



# AD18F06

---

## 用户手册

版本号: V1.0.0.2

---

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	Date
1	V1.0.0.0	初版	sunguoding	2022.7.20
2	V1.0.0.1	修正 ADC 章节通道描述	Aaron	2022.9.5
3	V1.0.0.2	1) 修改 LCD 控制寄存器、IIC 状态/控制寄存器描述 2) 增加 IO 双沿中断的使用说明 3) 增加 CMP 双沿中断的使用说明 4) 更新 CMP2 相关寄存器 5) 更新 IO 结构图 6) 更新中断寄存器定义	Aaron	2022.12.12

# 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 .....	13
1.3 引脚排列 .....	13
1.4 引脚说明 .....	15
1.5 极限参数 .....	20
1.6 烧录引脚说明 .....	21
2 中央处理器 .....	22
2.1 指令集 .....	22
2.2 程序存储 .....	22
2.2.1 程序计数器 .....	22
2.2.2 返回地址堆栈 .....	22
2.2.3 栈顶访问 .....	23
2.2.4 快速寄存器堆栈 .....	24
2.2.5 程序存储器 .....	25
2.3 数据存储器 RAM .....	26
2.4 数据寻址方式 .....	26
2.4.1 固有和立即数寻址 .....	27
2.4.2 直接寻址 .....	27
2.4.3 间接寻址 .....	27
2.5 配置选项 .....	29
3 EEPROM 和 FLASH .....	34
3.1 EEPROM .....	34
3.1.1 EECON1 和 EECON2 寄存器 .....	35
3.1.2 读数据 EEPROM 存储器 .....	36
3.1.3 写数据 EEPROM 存储器 .....	37
3.1.4 写校验 .....	38
3.1.5 避免误写操作 .....	38
3.2 FLASH .....	39

4	系统时钟源.....	43
4.1	系统时钟相关寄存器.....	43
4.2	频率与最小工作电压的关系.....	45
4.3	HIRC 频率微调.....	46
4.4	双速时钟启动模式.....	47
4.5	双速启动模式配置.....	47
4.6	双速启动顺序.....	48
4.7	故障保护时钟监控器.....	48
4.7.1.	故障保护操作.....	48
4.7.2.	故障检测完成.....	48
4.7.3.	故障保护条件清除.....	49
4.8	系统时钟相关寄存器定义.....	49
5	复位和电源电压检测.....	50
5.1	上电复位.....	51
5.2	低电压复位.....	51
5.3	上电复位延时.....	51
5.4	非法指令复位.....	51
5.5	软件复位.....	52
5.6	EMC 复位.....	52
5.7	LVD 检测.....	52
5.8	/TO /PD 状态.....	55
5.9	相关寄存器定义.....	55
6	I/O 端口.....	56
6.1	IO 工作模式.....	57
6.2	下拉电阻开漏.....	59
6.3	IO 中断寄存器.....	61
6.4	IO 模拟控制寄存器.....	62
6.5	外设功能引脚全映射控制.....	65
6.5.1.	外设功能引脚映射控制寄存器.....	65
6.6	IO 相关寄存器定义.....	68
7	定时器.....	69

7.1	Timer0 8 位定时/计数器 .....	69
7.1.1.	Timer0 计数/定时 .....	69
7.1.2.	使用内部时钟: 定时模式 .....	70
7.1.3.	使用外部时钟/内部 32K 时钟/: 计数模式 .....	70
7.1.4.	Prescaler (预置器) .....	70
7.1.5.	BUZZER (BUZZER 输出) .....	70
7.1.6.	TMR0 与 1 路 PWM .....	70
7.1.7.	TMR0 工作模式 .....	71
7.1.8.	寄存器列表 .....	75
7.2	TIMER1 16 位定时/计数器 .....	76
7.3	TIMER2 12 位定时器 .....	82
7.4	TIMER3 12 位定时器和 4 路 12 位 PWM .....	85
7.4.1.	TIMER3 定时器 .....	85
7.4.2.	4 路 12 位 PWM .....	87
7.4.3.	4 路 PWM 的输出 .....	88
7.4.4.	4 路 PWM 的周期 .....	88
7.4.5.	PWM 的占空比 .....	89
7.4.6.	PWM 的分辨率 .....	89
7.4.7.	PWM 的工作设置 .....	89
7.4.8.	PWM 中心对齐模式 .....	90
7.4.9.	TMR3 相关寄存器定义 .....	94
7.5	看门狗定时器 .....	94
8	捕获/比较/PWM 模块 .....	96
8.1	捕捉模式 .....	100
8.2	比较模式 .....	102
8.3	HBRIDGE 单元 .....	103
8.3.1.	PWM 调制 .....	108
8.3.2.	死区时间 .....	109
8.3.3.	互补式输出控制防呆电路 .....	110
8.4	CCP PWM 控制三路 HBRIDGE .....	112
8.5	TIMER3 PWM 控制三路 HBRIDGE .....	113

8.6	PWM 相关寄存器介绍 .....	114
8.7	PWM 使用说明 .....	118
	增强型六路 PWM .....	118
8.8	相关寄存器定义 .....	125
9	通用同步/异步收发器 (USART0 & UART1) .....	126
9.1	UART0 .....	126
	SPBRG .....	126
	RCREG .....	126
	RXREG .....	126
	TXSTA .....	126
	RCSTA .....	127
	UARTCON .....	129
9.2	波特率发生器 .....	130
9.3	异步发送器 .....	130
9.4	异步接收 .....	132
9.5	同步模式 .....	133
	9.5.1. 同步主机发送 .....	133
	9.5.2. 同步主机接收 .....	135
9.6	同步从机模式 .....	136
	9.6.1. 同步从机发送 .....	136
	9.6.2. 同步从机接收 .....	137
9.7	UART1 .....	138
	SPBRG1 .....	138
	RCREG1 .....	138
	RXREG1 .....	138
	TXSTA1 .....	138
	RCSTA1 .....	139
	UARTCON1 .....	140
9.8	UART0 & UART1 寄存器定义 .....	141
10	I2C 控制器 .....	142
10.1	I <sup>2</sup> C 相关寄存器: .....	143

I2CACKDLY (I2C 的延时控制寄存器)	143
I2CTX (I2C 的发送寄存器)	143
I2CRC (I2C 的接收寄存器)	143
I2CADD0 (I2C 的地址寄存器)	143
I2CADDMASK (I2C 的寄存控制器)	144
I2CSTAT (I2C 的控制寄存器)	144
I2CCON0 (I2C 的控制寄存器)	145
I2CCON1 (I2C 的控制寄存器)	146
I2CIE 寄存器	147
I2CIF 寄存器	147
I2CIP 寄存器	148
10.2    从机模式	149
10.3    支持广播呼叫地址	150
10.4    I2C 相关寄存器	151
11 SPI 接口	152
11.1    概述	152
11.2    模块框图	153
11.3    功能描述	153
11.3.1.    TR 接口传输格式	153
11.3.2.    主机模式传输格式	154
11.3.3.    中断功能	157
11.4    寄存器列表	157
SPCON 寄存器	157
SPSTA 寄存器	159
SPDAT 寄存器	160
11.5 SPI 相关寄存器定义	160
12    中断	161
12.1    外部中断	162
12.2    Timer0 中断	163
12.3    Timer1 中断	163
12.4    Timer2 中断	163

12.5	Timer3 中断 .....	163
12.6	PortA 输入改变中断.....	163
12.7	PortB 输入改变中断.....	164
12.8	PortC 输入改变中断.....	164
12.9	PortD 输入改变中断 .....	164
12.10	低电压、高电压中断.....	164
12.11	比较器中断.....	165
12.12	运放中断.....	165
12.13	ADC 中断.....	165
12.14	中断的相关寄存器 .....	165
12.15	中断相关寄存器定义.....	177
13	省电模式 (SLEEP).....	178
13.1	睡眠唤醒.....	178
13.2	SLEEP 相关寄存器定义.....	181
14	固定参考电压 (FVR) .....	182
14.1	FVR 相关寄存器定义 .....	183
15	数模转换器(DAC) .....	184
15.1	DAC 参考电压选择寄存器 .....	184
15.2	6Bit D/A 转换器 .....	186
15.3	DAC 相关寄存器定义.....	186
16	模数转换器 (ADC) .....	187
16.1	TIMER0 和 TIMER1 定时启动 ADC.....	187
16.2	ADC 的多路采集.....	187
16.2.1.	多路通道采集使用用方法: .....	187
16.3	A/D 转换步骤: .....	188
16.4	ADC 相关寄存器定义.....	200
17	运放.....	207
17.1	运放 OP .....	207
17.2	OP0 的输入失调校准.....	210
17.2.1.	OP0OFFSET 校准:.....	210
17.3	OP 相关寄存器定义 .....	211



18	比较器(CMP0&CMP1&CMP2&CMP3).....	212
18.1	比较器 CMP0.....	213
18.2	比较器 CMP1.....	216
18.3	比较器 CMP2.....	219
18.4	比较器 CMP3.....	222
18.5	CMP 相关寄存器定义.....	224
19	LCD 驱动和 LED 驱动.....	225
19.1	LCD 驱动.....	225
19.1.1.	概述.....	225
19.1.2.	LCD/LED 显示数据存储.....	226
19.1.3.	LCD 时钟源.....	226
19.1.4.	LCD 寄存器.....	227
19.1.5.	LCD 电压源和偏压.....	231
19.1.6.	LCD 的复位状态.....	231
19.1.7.	LCD 驱动输出.....	232
19.1.8.	编程注意事项.....	235
19.2	LED 驱动器.....	235
19.3	LCD/LED 驱动器相关寄存器定义.....	236
20	8X8 硬件乘法器和 16/16 硬件除法器.....	237
20.1	相关寄存器定义.....	239
21	电气参数.....	240
22	封装信息.....	244
23	订购信息.....	246

# 1 产品简介

## 1.1 功能特性

- RISC18 指令集，支持 83 条指令，内置 8\*8BIT 硬件乘法器和 16 位硬件除法器，拥有高低优先级中断：
  - ✧ 程序空间：8K\*16 BIT FLASH，支持 100K 次擦除写入，一页为 256 字节，支持 IAP；
  - ✧ EEPROM：256\*8BIT，支持字节擦除，PAGE（16Byte）擦除，擦除时间 1mS，写入 BYTE 时间 8uS；
  - ✧ 数据空间：512\*8 BIT SRAM
  - ✧ 堆栈：硬件 16 级堆栈，支持 16 层硬件堆栈嵌套
  - ✧ 支持 2T 和 4T 的 CPU 运算模式；
  - ✧ 支持在线调试，4 个硬件断点，无限个软件断点；
- 振荡器
  - ✧ 内部 16MHZ RC 振荡器，可以倍频到 32MHZ，用于系统时钟，支持软件微调。
  - ✧ 内部 1MHz RC 振荡器，用于系统时钟。
  - ✧ 内部 32KHz RC 超低功耗振荡器，可以用于计数和看门狗计数。
  - ✧ 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
  - ✧ 晶体时钟缺失检测
  - ✧ 晶体时钟配置的双速时钟启动
  - ✧ 慢时钟周期测量
- GPIO
  - ✧ 支持 26 个 IO+4 个 IO（普通），支持开漏、上拉、下拉独立控制
  - ✧ IO 口的电流档是可选的。
  - ✧ 支持 IO 映射模块 PMUX，可以将 UART0, UART1, I2C, PWM0, PWM10, PWM11, PWM20, PWM21, PWM30, PWM31, PWM2, PWM3, PWM4, PWM5 输入输出映射到所有 GPIO 引脚
- 内置三个时钟，高速 HIRC 可选 16M/8M/4M/2M/1M（支持倍频到 32MHz）；低速 LIRC 时钟 1M；超低功耗 LPIRC 32K 时钟；外部支持高速晶振 LXTH 和低速晶振时钟 LXTL；

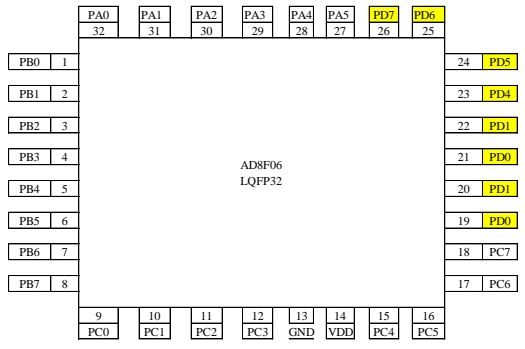
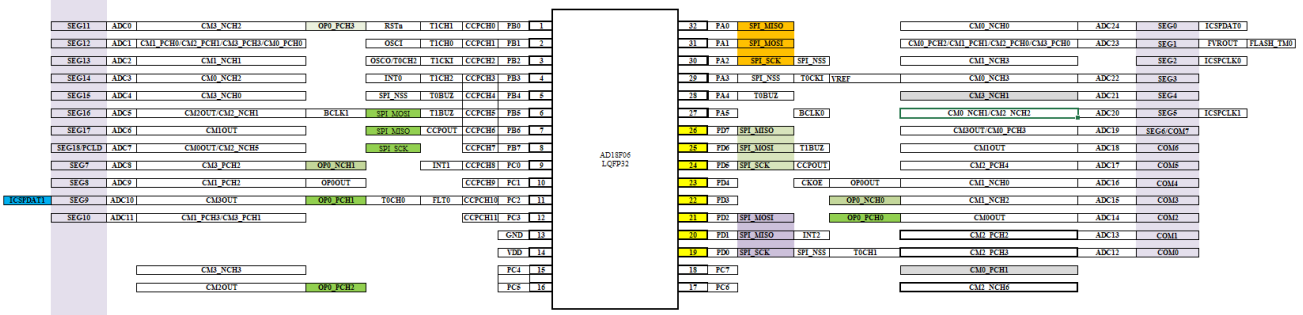
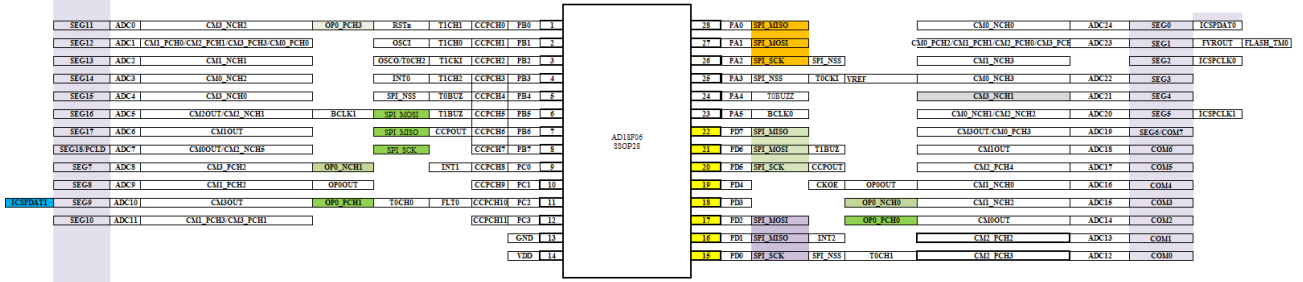
这些时钟可以作为 FCPU 时钟；通过寄存器可实时切换 CPU 时钟；CPU 有 IDLE, PWSAVE, DEEPPWSAVE, PWOFF 四种工作模式；

- ◇ IDLE: CPU 停止工作，外设工作正常；所有中断可以唤醒，唤醒后继续从当前 PC 运行；
- ◇ PWSAVE: CPU 停止工作，高速 32M 时钟停止工作，低速 32K 时钟工作；支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 32K 定时唤醒，唤醒后继续从当前 PC 运行；LCD/LED 可以运行，LCD 可以选择 10uA 驱动；
- ◇ DEEPPWSAVE: CPU 停止工作，高速 32M 时钟停止工作，低速 32K 时钟工作，SRAM 数据保持；支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 32K 定时唤醒，唤醒后继续从当前 PC 运行；LCD/LED 可以运行，LCD 可以选择 10uA 驱动；
- ◇ PWOFF: 全部外设和模拟停止工作，支持外部中断、IO 中断、复位，唤醒后继续从当前 PC 运行；
- TIMER0, 8 位定时/计数器，有多个时钟输入选择作为 TIMER0 时钟，同时支持 BUZZER 输出模式；支持一路 8bit 分辨率的 PWM 输出；支持 4 个通道输入捕捉功能；可以硬件启动 ADC；PWM 输出可以映射到任意 IO；
- TIMER1, 16 位定时/计数器，有多个时钟输入选择，支持 BUZZER 输出；可以工作在异步时钟工作模式，同时支持 IO 管脚捕捉模式的输入功能，可以硬件自动启动 ADC；可以输出一路 16 位 PWM 可调占空比的 PWM 输出；
- TIMER2, 12 位定时计数器，支持 CPU 中断，时钟源为 FCPU；支持调频，PR2 写入值更新为 TIMER2 溢出时间；
- TIMER3, 12 位定时计数器，支持 CPU 中断，时钟源为 FCPU；支持调频，PR3 写入值更新为 TIMER3 溢出时间；支持 4 路独立的可调占空比的 12 位 PWM 输出；PWM 输出可以映射到任意 IO；
- CCP, 与 Timer1 构成捕获和比较模块，支持一路捕获、一路比较；捕获支持三路霍尔传感器输入捕获和相位值；捕获可以选择 TIMER1 或者 TIMER2 做为采集输入源；与 Timer2 构成一路 12bit 分辨率独立 PWM 功能，支持 ADC 采集和自启动自关闭；
- 3 路独立 HBRIDGE PWM 输出控制单元，可以控 PWM 死区和互补输出；可以与 CCP 的 PWM 构成三组全桥互补 PWM 输出；与 TIMER3 的 PWM2 构成三组全桥互补 PWM 输

- 出；与 TIMER3 的 PWM2/PWM3/PWM4 构成三组独立可调占空比的全桥互补 PWM 输出；PWM 同时支持中心对齐模式，支持中心点启动 ADC 采集；支持比较器保护模式，可以自动启动 PWM 和关闭 PWM；PWM 输出可以映射到任意 IO；
- 2 路 UART：同步、异步、半双工和单线串口，支持 1bit、2bit、3bit 停止位，两路 UART 输出可以映射到任意 IO；
  - 硬件从 I2C 接口，支持 100K, 400K, 1M BPS，支持两个 7 位从地址应答寄存器，可设定是否产生应答；与调试接口服用，启用调试接口，I2C 寄存器不能配置，否则影响调试接口使用；I2C 接口可以映射到任意 IO；
  - 一路硬件主从 SPI 接口，最大支持 8MHz 传输率；可以映射到 4 组 IO；
  - 内置高精度电压源，提供电压 1.0V、2.0V、3.0V；
  - 内置温度传感器（NTC），支持 ADC 采集；
  - 12 BIT 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；ADC 支持 4 路通道连续采集设定，内部支持 4\*12 的 ADC 值缓存；
  - 1 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、DAC、FVR 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；支持单独配置使能功能；DAC 基准可以分开控制；
  - 4 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~VDD-1.5V；CMP 输出可以输出 IO；比较器输入支持 IO、DAC、OP0、FVR；支持单独配置使能功能；支持双沿产生中断；CMP0/CMP1/CMP3 支持迟滞和输入失调电压调整；
  - 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；
  - LCD 驱动功能：
    - ◇ LCD 支持：8Com\*18Seg, 7Com\*19Seg, 6Com\*19Seg, 5Com\*19Seg, 4Com\*19Seg, 3Com\*19Seg；
    - ◇ 支持 LCD R 型 1/2 BIAS 和 1/3 BIAS；
    - ◇ 偏压类型为 R 型；波形为 A 型或者 B 型；

- ◇ 支持 LCD 显示频率分频 32KHz 的 1/1、1/2、1/3、1/4、1/5 至 1/32 可调，最低 3.90625Hz，最高 666.7Hz；
- ◇ LCD 1/3 BIAS 占空比为 1/8、1/7、1/6、1/5、1/4、1/3；
- ◇ LCD 1/2 BIAS 占空比为 1/4、1/3
- LED 驱动功能:支持共阴级驱动，SEGxCOM8\*18,7\*19 模式；
- 超强程序加密算法，保证芯片程序内容唯一性；
- 除跳转指令为两个周期指令以外其余为单周期指令；
- 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；
- 中断：
  - ◇ 四个内部计数/定时器中断源；
  - ◇ 三个外部 IO 管脚中断源：INT 管脚；
  - ◇ PortA、PortB、PortC 和 PortD 的输入改变中断源；
  - ◇ CCP 中断；
  - ◇ 低电压、高电压 LVD 中断；
  - ◇ 比较器中断；
  - ◇ 运放中断
  - ◇ UART0 中断/UART1 中断
  - ◇ I2C 中断
  - ◇ SPI 中断
  - ◇ ADC 中断；
- 通过外部中断、PortA 中断、PortB 中断、PortC 中断、PortD 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；
- 有可靠的保证使得程序代码不被读出；
- CPU 支持 IAP 自编程功能；并支持整个程序空间读取和写入；
- 提供 96 BIT UID ；
- VDD 工作电压范围：2.4V - 5.5V，
  - ◇ CPU 工作的 16MHz(2T 和 4T 模式,电压 2.4V~5.5V)；
  - ◇ CPU 工作的 32MHz(4T 模式,电压 3.0V~5.5V) ；





注：PD0~PD7 为大电流口

## 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
	SPI_MISO	-	CMOS	第 1 路 SPI 的数据引脚		
	ADC24	AN		ADC 的通道		
	CM0_NCH0	AN		CMP0 的负端输入		
	SEG0		CMOS	LCD/LED 的 SEG 口		
	ICSPDAT0	SMT		SWD 模型下烧录数据引脚		
PA1	PA1	SMT	CMOS	GPIO	Y	Y
	SPI_MOSI	-	CMOS	第 1 路 SPI 的数据引脚		
	ADC23	AN		ADC 的输入引脚		
	CM0_PCH2	AN		CMP0 的正端输入		
	CM1_PCH1	AN		CMP1 的正端输入		
	CM2_PCH0	AN		CMP2 的正端输入		
	CM3_PCH0	AN		CMP3 的正端输入		
	SEG1		CMOS	LCD/LED 的 SEG 口		
	FVROUT	AN		FVR 的输出引脚		
PA2	PA2	SMT	CMOS	GPIO	Y	Y
	SPI_SCK	SMT		第 1 路 SPI 的输入时钟		
	SPI_NSS	SMT		SPI 的片选引脚		
	ADC26	AN	-	ADC 的输入通道		
	CM1_NCH3	AN		CMP1 的正端输入		
	SEG2	AN		LCD/LED 的 SEG 口		
	ICSPCLK0	SMT		SWD 模型下烧录时钟引脚		
PA3	PA3	SMT	CMOS	GPIO	Y	Y
	SPI_NSS	SMT		SPI 的片选引脚		
	T0CKI	SMT		TIMER0 的外部计数时钟源		
	ADC22	AN	-	ADC 的输入通道		
	VREF	AN		外部电压基准		
	CM0_NCH3	AN		CMP0 的负端输入		
	CM3_NCH1	AN		CMP3 的负端输入		
	SEG3		CMOS	LCD/LED 的 SEG 口		
PA4	PA4	SMT	CMOS	GPIO	Y	Y
	T0BUZZ		CMOS	TIMER0 的 buzz 的输出		
	CM2_NCH2	AN		CMP2 的负端输入		
	ADC21	AN	-	ADC 的输入通道		
	SEG4		CMOS	LCD/LED 的 SEG 口		
PA5	PA5	SMT	CMOS	GPIO	Y	Y
	BCLK0	SMT		UART0 的外部时钟引脚		



	<b>ADC20</b>	AN	-	ADC 的输入通道		
	<b>CM0_NCH1</b>	AN		CMP0 的负端通道		
	<b>CM2_NCH2</b>	AN		CMP2 的负端通道		
	<b>SEG5</b>		CMOS	LCD/LED 的 SEG 口		
	<b>ICSPCLK1</b>	SMT		SWD 模型下烧录时钟引脚		
<b>PB0</b>	<b>PB0</b>	SMT	CMOS	GPIO	Y	Y
	<b>T1CH1</b>	SMT		TIMER1 的脉宽测量通道		
	<b>RSTn</b>	SMT		芯片复位引脚		
	<b>CCP0CH0</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH0</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH0</b>	SMT		CCP 第 3 路通道		
	<b>ADC0</b>	AN		ADC 通道输入		
	<b>OP0_PCH3</b>	AN		OP0 的正端输入		
	<b>CM2_NCH5</b>	AN		CMP2 的负端通道		
	<b>CM3_NCH2</b>	AN		CMP3 的负端通道		
	<b>SEG11</b>		CMOS	LCD/LED 的 SEG 口		
<b>PB1</b>	<b>PB1</b>	SMT	CMOS	GPIO	Y	Y
	<b>T1CH0</b>	SMT		TIMER1 的脉宽测量通道		
	<b>OSCI</b>	SMT		外部晶振输入引脚		
	<b>CCP0CH1</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH1</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH1</b>	SMT		CCP 第 3 路通道		
	<b>ADC1</b>	AN		ADC 的模拟通道		
	<b>CM0_PCH0</b>	AN		CMP0 的正端输入		
	<b>CM1_PCH0</b>	AN		CMP1 的正端输入		
	<b>CM3_PCH3</b>	AN		CMP2 的正端输入		
	<b>SEG12</b>		CMOS	LCD/LED 的 SEG 口		
<b>PB2</b>	<b>PB2</b>	SMT	CMOS	GPIO	Y	Y
	<b>T1CK1</b>	SMT		TIMER1 的外部计数时钟源		
	<b>OSCO</b>		CMOS	外部晶振的时钟输出		
	<b>T0CH2</b>	SMT		TMR0 的脉宽检测通道		
	<b>CCP0CH2</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH2</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH2</b>	SMT		CCP 第 3 路通道		
	<b>ADC2</b>	AN		ADC 通道		
	<b>CM1_NCH1</b>	AN		CMP1 的负端输入		
	<b>SEG13</b>		CMOS	LCD/LED 的 SEG 口		
<b>PB3</b>	<b>PB3</b>	SMT	CMOS	GPIO	Y	Y
	<b>T1CH2</b>	SMT		TIMER1 的脉宽测量通道		
	<b>INT0</b>	SMT		外部中断 0		
	<b>CCP0CH3</b>	SMT		CCP 第 1 路通道		

	<b>CCP1CH3</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH3</b>	SMT		CCP 第 3 路通道		
	<b>ADC3</b>	AN		ADC 的模拟通道		
	<b>CM0_NCH2</b>	AN		CMP0 的负端输入		
	<b>SEG14</b>		CMOS	LCD/LED 的 SEG 口		
<b>PB4</b>	<b>PB4</b>	SMT	CMOS	GPIO	Y	Y
	<b>SPI_NSS</b>	SMT		SPI 的片选信号		
	<b>CCP0CH4</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH4</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH4</b>	SMT		CCP 第 3 路通道		
	<b>ADC4</b>	AN		ADC 的模拟通道		
	<b>CM2_NCH1</b>	AN		CMP2 的负端输入		
	<b>CM3_NCH0</b>	AN		CMP3 的负端输入		
	<b>SEG15</b>		CMOS	LCD/LED 的 SEG 口		
<b>PB5</b>	<b>PB5</b>	SMT	CMOS	GPIO	Y	Y
	<b>SPI_MOSI</b>	-	CMOS	第四路 SPI 的数据引脚		
	<b>BCLK1</b>	SMT		UART1 的外部波特率时钟引脚		
	<b>CCP0CH5</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH5</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH5</b>	SMT		CCP 第 3 路通道		
	<b>CM2_NCH1</b>	AN		CMP2 的负端通道		
	<b>ADC5</b>	AN		ADC 的模拟通道		
	<b>CM2OUT</b>	AN		CMP2 的输出		
<b>PB6</b>	<b>SEG16</b>		CMOS	LCD/LED 的 SEG 口		
	<b>PB6</b>	SMT	CMOS	GPIO	Y	Y
	<b>SPI_MISO</b>		CMOS	第 4 路 SPI 的数据输出		
	<b>CCP0CH6</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH6</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH6</b>	SMT		CCP 第 3 路通道		
	<b>ADC6</b>	AN		ADC 的模拟通道		
	<b>CM1OUT</b>			CMP1 的输出		
<b>PB7</b>	<b>SEG17</b>			LCD/LED 的 SEG 口		
	<b>PB7</b>	SMT	CMOS	GPIO	Y	Y
	<b>SPI_SCK</b>	SMT		第 4 路 SPI 的时钟信号		
	<b>CCP0CH7</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH7</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH7</b>	SMT		CCP 第 3 路通道		
	<b>ADC7</b>	AN		ADC 的模拟通道		
	<b>CM2_NCH5</b>	AN		CMP2 的负端通道		
	<b>PLCD</b>	AN		LCD 的外部供电电源		
	<b>CM0OUT</b>			CMP0 的输出		

	<b>SEG18</b>			LCD/LED 的 SEG 口		
<b>PC0</b>	<b>PC0</b>	SMT	CMOS	GPIO	Y	Y
	<b>INT1</b>	SMT		外部中断 1		
	<b>CCP0CH8</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH8</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH8</b>	SMT		CCP 第 3 路通道		
	<b>ADC8</b>	AN		ADC 的模拟通道		
	<b>OP0_NCH1</b>			OP0 的负端输入		
	<b>CM2_PCH2</b>			CMP2 的正端输入		
	<b>CM3_PCH2</b>			CMP3 的正端输入		
	<b>SEG7</b>			LCD/LED 的 SEG 口		
<b>PC1</b>	<b>PC1</b>	SMT	CMOS	GPIO	Y	Y
	<b>CCP0CH9</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH9</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH9</b>	SMT		CCP 第 3 路通道		
	<b>ADC9</b>	AN		ADC 的模拟通道		
	<b>CM3_PCH1</b>			CMP3 的正端输入		
	<b>OP0OUT</b>			OP0 的输出引脚		
	<b>CM1_PCH3</b>	AN		CMP1 的正端输入		
	<b>SEG8</b>			LCD/LED 的 SEG 口		
<b>PC2</b>	<b>PC2</b>	SMT	CMOS	GPIO	Y	Y
	<b>FLT0</b>	SMT		故障检测输入引脚		
	<b>T0CH0</b>	SMT		TMR1 的脉宽检测测量通道		
	<b>CCP0CH10</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH10</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH10</b>	SMT		CCP 第 3 路通道		
	<b>ADC10</b>	AN		ADC 的模拟通道		
	<b>OP0_PCH1</b>			OP0 的正端输入		
	<b>CM3OUT</b>	AN		CMP3 的输出		
	<b>SEG9</b>			LCD/LED 的 SEG 口		
<b>PC3</b>	<b>ICSPDAT1</b>			SWD 模式下烧录的数据引脚		
	<b>PC3</b>	SMT	CMOS	GPIO	Y	Y
	<b>CCP0CH11</b>	SMT		CCP 第 1 路通道		
	<b>CCP1CH11</b>	SMT		CCP 第 2 路通道		
	<b>CCP2CH11</b>	SMT		CCP 第 3 路通道		
	<b>ADC11</b>	AN		ADC 的模拟通道		
	<b>CM1_PCH3</b>			CMP1 的正端输入		
	<b>CM2_PCH1</b>			CMP2 的正端输入		
<b>SEG10</b>			LCD/LED 的 SEG 口			
	<b>PC4</b>	SMT	CMOS	GPIO	Y	Y

PC4	ADC25	AN		ADC 的模拟通道		
	CM3_NCH3	AN		CMP3 的负端输出		
	CM2_NCH6	AN		CMP2 的负端输出		
PC5	PC5	SMT	CMOS	GPIO	Y	Y
	CM2_NCH6	AN		CMP2 的负端输入		
	OP0_PCH2	AN		OP0 的正端输入		
PC6	PC6	SMT	CMOS	GPIO	Y	Y
	CM2_NCH6	AN		CMP2 的负端输入		
PC7	PC7	SMT	CMOS	GPIO	Y	Y
	CM0_PCH1	AN		CMP0 的正端输入		
PD0	PD0	SMT	CMOS	GPIO	Y	Y
	SPI_SCK	SMT		第 3 路 SPI 的时钟引脚		
	SPI_NSS	SMT		SPI 的片选信号		
	T0CH1	SMT		TMR0 的脉宽测量通道		
	ADC12	AN		ADC 的模拟通道		
	CM2_PCH3	AN		CMP2 的正端输入		
	COM0		CMOS	LCD/LED 的 COM 口		
PD1	PD1	SMT	CMOS	GPIO	Y	Y
	SPI_MISO		CMOS	第 3 路 SPI 的数据引脚		
	INT2	SMT		外部中断 2		
	ADC13	AN		ADC 的模拟通道		
	CM2_PCH2	AN		CMP2 的正端输入		
	COM1		CMOS	LCD/LED 的 COM 口		
PD2	PD2	SMT	CMOS	GPIO	Y	Y
	SPI_MOSI		CMOS	第 3 路 SPI 的数据引脚		
	ADC14	AN		ADC 的模拟通道		
	OP0_PCH0			OP0 的正端输入		
	CM0OUT	AN		CMP0 的输出		
	COM2		CMOS	LCD/LED 的 COM 口		
PD3	PD3	SMT	CMOS	GPIO	Y	Y
	ADC15	AN		ADC 的模拟通道		
	OP0_NCH0			OP0 的负端输入		
	CM1_NCH2	AN		CMP1 的负端输入		
	COM3		CMOS	LCD/LED 的 COM 口		
PD4	PD4	SMT	CMOS	GPIO	Y	Y
	CKOE		CMOS	系统时钟输出引脚		
	ADC16	AN		ADC 的模拟通道		
	OP0OUT			OP0 的输出		
	CM1_NCH0	AN		CMP1 的负端输入		
	COM4		CMOS	LCD/LED 的 COM 口		
	PD5	SMT	CMOS	GPIO	Y	Y
	SPI_SCK	SMT		第 2 路 SPI 的时钟引脚		

PD5	CCPOUT	SMT		CCP 的输出引脚		
	ADC17	AN		ADC 的模拟通道		
	CM2_PCH4	AN		CMP2 的正端输入		
	COM5		CMOS	LCD/LED 的 COM 口		
PD6	PD6	SMT	CMOS	GPIO	Y	Y
	SPI_MOSI		CMOS	第 2 路 SPI 的数据引脚		
	T1BUZZ		CMOS	TIMER1 的 buzz 输出		
	ADC18	AN		ADC 的模拟通道		
	CM1OUT	AN		CMP1 的输出		
	COM6		CMOS	LCD/LED 的 COM 口		
PD7	PD7	SMT	CMOS	GPIO	Y	Y
	SPI_MISO		CMOS	第 2 路 SPI 的数据引脚		
	CM0_PCH3	AN		CMP0 的正端输入		
	ADC19	AN		ADC 的模拟通道		
	CM3OUT	AN		CMP3 的输出		
	COM7		CMOS	LCD/LED 的 COM 口		
	SEG6		CMOS	LCD/LED 的 SEG 口		

## 1.5 极限参数

电源供应电压:	-----	$V_{SS}-0.3V \sim 5.5V$
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:	-----	120mA
IOH 总电流:	-----	-120mA
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。

## 1.6 烧录引脚说明

AD18F06烧录引脚说明			
序号	烧录引脚	烧录信号	在板说明
1	VDD	VDD (4.5v)	VDD Pin 和 GND Pin 在烧录时，电压会变化，因此必须由烧录器VDD 直接连接到芯片VDD。在烧录时，烧录器会多次断电再上电做检查，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	GND	GND	
3	PA0	DAT	CLK, DAT, 为通信线，不允许接入0.1uF以上的电容，不允许接入上下拉电阻。
4	PA2	CLK	

## 2 中央处理器

### 2.1 指令集

AD18F06 具有一个支持 84 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

AD18F06 包含一个 8K\*16 的 FLASH，支持 100K 次擦除写入，一页 256 字节

AD18F06 包含一个 256 字节的 EEPROM，支持字节擦除，PAGE（16BYTE）擦除，擦除时间 1ms,写 BYTE 时间 8 μs; 支持字节写入

AD18F06 有高低优先级的两个中断向量，中断向量地址为 0008H 和 0018H。

AD18F06 复位向量地址为 0000H。

#### 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（STKPTR）来实现 16 级的硬件堆栈操作。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。

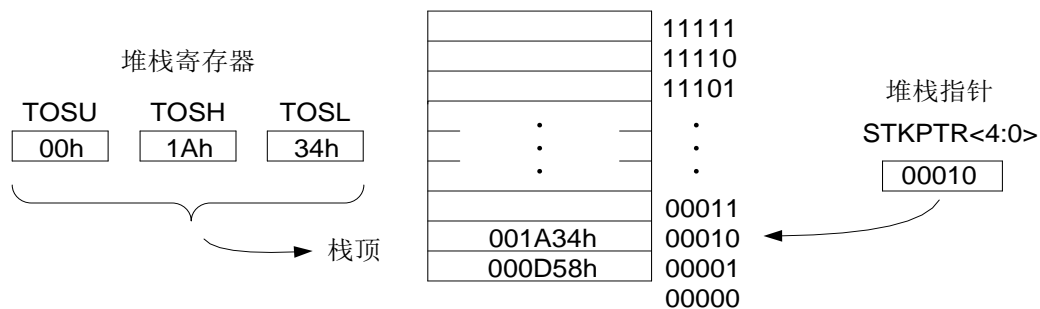
执行 CALL 类型指令引起进栈操作：堆栈指针首先加 1，并且将 PC 的内容写入堆栈指针指向的地址单元（PC 已经指向 CALL 的下一条指令）。执行 RETURN 类型指令时，引起出栈操作：STKPTR 寄存器所指向的地址单元的内容被传送给 PC，然后堆栈指针减 1。

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）时可读写的。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

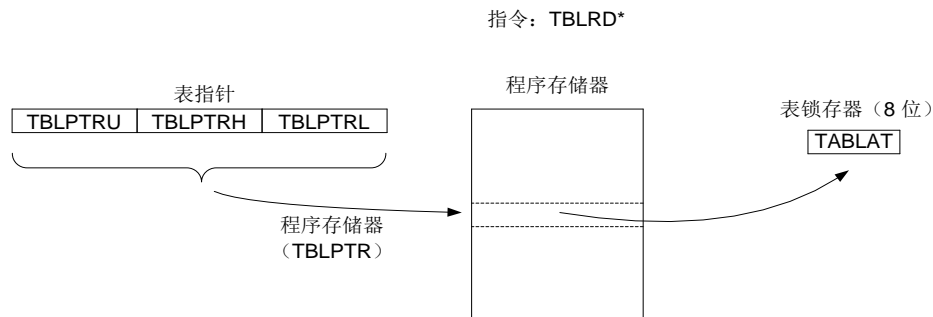
为 Status、WREG 和 BSR 寄存器提供的快速寄存器堆栈具有从中断“快速返回”的功能。每个寄存器堆栈的深度仅为 1 级，并且不可读写。当处理器转入中断向量处执行指令时，此堆栈装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用 RETFIE, FAST 指令从中断返回，这些寄存器中的值会被重新装回对应的寄存器。

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，AD18F06 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。



表读操作

表锁寄存器（Table Latch, TABLAT）是映射到 SFR 空间的一个 8 位寄存器。表锁寄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：

```

MOVLW      CODE_ADDR_UPPER      ;Load TBLPR with the base
MOVWF     TBLPTRU                ;address of the word
MOVLW     CODE_ADDR_HIGH
MOVWF     TBLPTRH
    
```

```

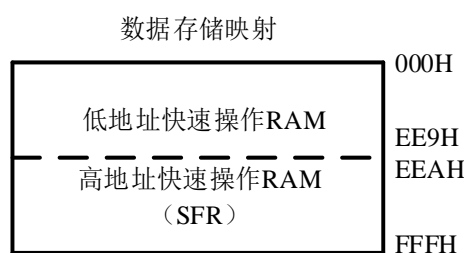
        MOVLW        CODE_ADDR_LOW
        MOVWF        TBLPTRL
    READ_WORD
        TBLRD*+                                ;read into TABLAT and increment
        MOVF         TABLAT, W                  ;get data
        MOVWF        WORD_EVEN
        TBLRD*+
        MOVFW        TABLAT, W                  ;get data
        MOVF         WORD_ODD
    
```

## 2.3 数据存储器 RAM

AD18F06 的数据存储器是用静态 RAM 实现的。存储空间包含 512 个字节，地址为 000H~1FFH;

LCD 的部分寄存器地址为 400H-412H,如果不使用 LCD,该地址区可以做 SRAM 空间使用;

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;SFR 的地址 EEAH~FFFH。



SRAM 数据存储器映射图

## 2.4 数据寻址方式

AD18F06 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

很多 AD18F06 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	LVR[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(默认) 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 0: 禁止使能 (默认)
1	RSEL	端口上/下拉电阻选择 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 (CKSUM 计算 FLASH CP 设定区域, 不保护时, 计算整个区域) 1: 使能校验 0: 关闭校验(默认)
		模拟 IO 打开, 数据读取控制位。

5	IDIS	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_CP0	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	FLASH_CP4	4~5K*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)
13	FLASH_CP5	5K*16~6K*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)
		6K*16~7K*16 FLASH 保护位(CBP 为 1 时有效), 保护后不可擦写



14	FLASH_CP6	1: 使能 FLASH 保护 0: 禁止 FLASH 保护(默认)
15	FLASH_CP7	7K*16~8K*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)

## 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) = 8.32s(默认) 110: TWDT (no Prescaler) = 4.224s 101: TWDT (no Prescaler) = 2.176s 100: TWDT (no Prescaler) = 1.152s 011: TWDT (no Prescaler) = 640ms 010: TWDT (no Prescaler) = 384ms 001: TWDT (no Prescaler) = 256ms 000: TWDT (no Prescaler) = 128ms
3	WDTEN	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
6:5	PSUT[1:0]	上电复位时间选择 11: PWRT = 18.05ms (默认) 10: PWRT = 288.05ms 01: PWRT = 864.05ms 00: PWRT = 0.55ms
	DBGEN	DEBUG 模式使能位,

7		1: 使能 0: 禁止(默认)
---	--	--------------------

## 配置字 3

位	名称	说明
0	DBGS_IOSEL	DEBUG 模式使能位以及下载脚的选择 1: PA0(ICSPDAT0)和 PA2(ICSPCLK0) 0: PC2(ICSPDAT1)和 PA5(ICSPCLK1)
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)
3	FLDIS	SWD 模式下读 FLASH 控制 1: 允许读 0: 禁止读 (默认)
4	EEDIS	SWD 模式下读 EEPROM 控制 1: 允许读 0: 禁止读 (默认)
5	ENVDDL	FLASH 的低电压读使能位 1: 使能读 (默认) 0: 禁止读
7~6	OSCM	内部芯片工作时钟选择 00: 选择内部低速时钟 32KHz 01: 选择外部高速晶振时钟 CRYH 10: 选择内部低速时钟 1MHz 11: 选择内部高速时钟 16MHz (默认)

## 3 EEPROM 和 FLASH

### 3.1 EEPROM

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）
- EEDAT
- EEADR

EEDAT 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。AD18F06 具有 256 字节的数据 EEPROM，地址范围从 00H 到 FFH，写入电压为芯片工作电压。

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。

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。

#### EE DAT 寄存器

地址：0XF48

Bit	Name	Description	Attribute	Reset
7:0	EEDAT[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：0XF47

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，WERERR 将被置 1。在这些情况下，用户可以在复位后检查 WERERR 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 EEDAT 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、EEDAT 和 EEADRH 和 EEADRL 寄存器。**

#### EECON1 寄存器

地址：0XF46

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	ER16EN	EEPROM 16Bytes 擦除使能 1: 使能 0: 禁止	R/W	0
5	ERBEN	EEPROM 1 Byte 擦除使能 1: 使能 0: 禁止	R/W	0
4	WRBEN	EEPROM 1Byte 写使能 1: 使能 0: 禁止	R/W	0
3	WERERR	EEPROM 错误标志位	R/W	0

		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		
2	WREN	EEPROM 正常写使能位 1: 允许正常写 (正常写包含 1Byte 擦除和 1Byte 写入) 0: 禁止正常写	R/W	0
1	WR	写控制位/擦除控制位 1: 启动正常写/擦除(一旦完成写操作或擦除操作, 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 的操作完成	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: 0XF45

Bit	Name	Description	Attribute	Reset
7:0	EECON2	写入数据 0X18 和 0XF6, 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, EEDAT 寄存器就有数据了。因此该数据可由下一条指令读取。EEDAT 将保持这个值直到用户下一次从该单元读取或向该单元写入数据时 (在写操作过程中)。

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 EEDAT 寄存器。然后用户必须按特定顺序开始写入每个字节。

写入 EEPROM 有两种方式：

第一种是正常写，正常写包含硬件实现的 EEPROM 1BYTE 擦除和写入逻辑，直接按例子写入即可，无需关心擦除的问题。

例：正常写数据 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 = 0XF6;	
EECON1 = 0X06;	使能 WR
while(EEIF==0)	等待写成功中断标志

第二种是 1Byte 擦除和 1Byte 写入，擦除和写入逻辑分开，需要先擦除再写入。

例：按 Byte 写数据 EEPROM

EEADR = data_addr;	写 EEPROM 数据地址
EEIF = 0;	清中断标志
EECON1 = 0X20;	使能 ERBEN
EECON2 = 0X18;	
EECON2 = 0XF6;	
EECON1 = 0X22;	使能 WR 擦除
while(EEIF==0)	等待擦除成功中断标志
EEDAT = data;	写 EEPROM 数据
EEIF = 0;	清中断标志
EECON1 = 0X10;	使能 WREN

```

EECON2 = 0X18;

EECON2 = 0XF6;

EECON1 = 0X12;      使能 WR 写操作

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

checkerror;        (伪代码)

else

checkok            (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 容量为 8Kx16，可以自编程的地址空间从 0000H 到 1FFFH；IAPADDR[13:0]（由 IAPADDRH 高 6 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℃。

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,0x18,0xf6，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-3FFFH 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[12:0]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。

### IAPTRIG 寄存器

地址：0XF44

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

### IAPCTRL 寄存器

地址：0XF43

Bit	Name	Description	Attribute	Reset
7:4	Reserved			



3	ERR	FLASH 的写/擦除错误标志位 1: 写/擦除操作异常（在开启代码保护状态下，写入或者擦除数据） 0: 写/擦除正常	R/W	0
2	LOCK	IAP 锁定状态位 1: 锁定 0: 未锁定	R/W	0
1	ER	FLASH 页擦除（Fosc 为 16Mhz 时需要耗时 2.1ms，无需手动延时；每页为 256x8） 1: 使能擦除 0: 未擦除	R/W	0
0	PG	写入 1: 写入（Fosc 为 16Mhz 需要耗时 21us，无需手动延时） 0: 未写入	R/W	0

### IAPADDRH 寄存器

地址：0XF42

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	IAPADDR[13:8]	IAP 地址高六位	R/W	0

### IAPADDRL 寄存器

地址：0XF41

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

### IAPDATH 寄存器

地址：0XF40

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

### IAPDATL 寄存器

地址：0XF3F

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

## IAPWAIT 寄存器

地址：0XF3E

Bit	Name	Description	Attribute	Reset
7:0	IAPWAIT[7:0]	$Delay\ Time = (IAPWAIT * 2 + 1) * 4 / F_{sys}$ 注：Delay Time 时间合理范围为 8us~12us，越接近 8us 越好，不能低于 8us	R/W	0x10

## SIGCON 寄存器

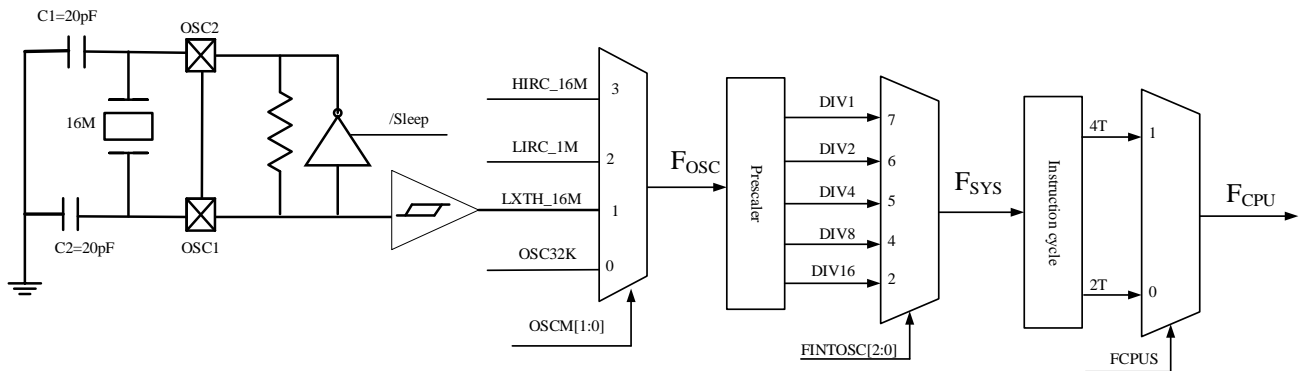
地址：0XF11

Bit	Name	Description	Attribute	Reset
7	ICE_TMEREN	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
6	PLL	时钟的倍频选择 1：系统时钟为 32M 0：系统时钟为 16M	R/W	0
5	LXTL_SEEPUP	低速晶振加速起振使能 1：允许使能加速起振（低速晶振起振后需关闭） 0：禁止使能加速起振	R/W	0
4:2	Reserved			
1	ENVDDL	FLASH 的低电压读使能位 1：使能读（默认） 0：禁止读	R/W	1
0	SIGS	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，不应使能配置自中的 CKSUM 功能。

## 4 系统时钟源

### 4.1 系统时钟相关寄存器



时钟原理图

注:OSC1 和 OSC2 为外部高速晶振和外部低速晶振

#### OSCCON 寄存器

地址: 0XF12

Bit	Name	Description	Attribute	Reset
7:5	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 010: 1:16 分频(默认)	R/W	010
4	CKOE	系统时钟输出引脚 1: 允许使能系统时钟输出引脚 0: 禁止使能系统时钟输出引脚	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: 表示外部高速晶振工作 (LXTH) 10: 表示内部低速时钟工作 (1MHz) 11: 表示内部高速时钟工作 (16MHz)	R	11

1:0	OSCM[1:0]	内部芯片工作时钟选择 00: 选择内部低速时钟 32KHz 01: 选择外部高速晶振时钟 LXTH 10: 选择内部低速时钟 1MHz 11: 选择内部高速时钟 16MHz (默认)	R/W	11
-----	-----------	---	-----	----

### CLKCFG1 寄存器

地址: 0XF4E

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	UART1CLKEN	UART1 的时钟使能, 使能后 UART1 才可以工作 1: 允许使能 0: 禁止使能	R/W	0
5	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
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	0
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	0
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	0
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	0
0	T0CLKEN	TIMER0 的时钟使能, 使能后 TIMER0 才可以工作	R/W	0

		1: 允许使能 0: 禁止使能		
--	--	--------------------	--	--

## CLKCFG2 寄存器

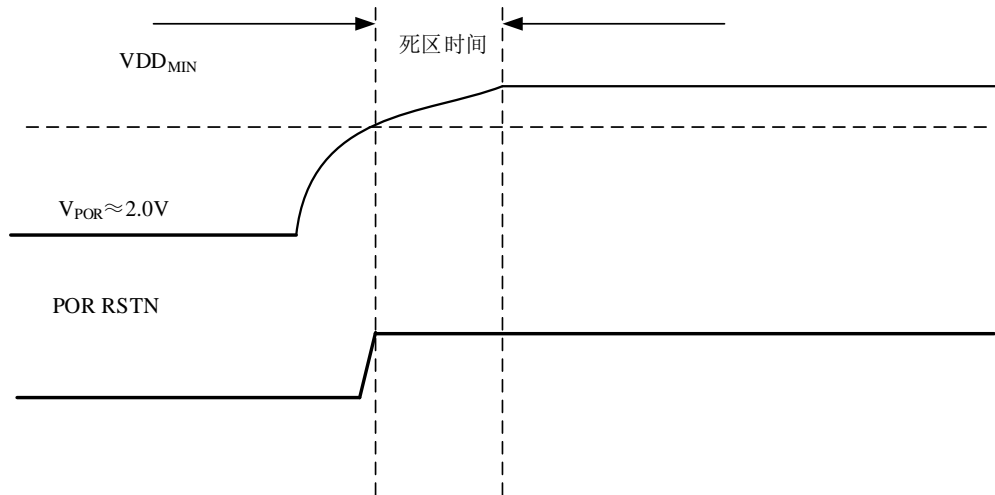
地址: 0XF4D

Bit	Name	Description	Attribute	Reset
7	SPICLKEN	SPI 的时钟使能, 使能后 SPI 才可以工作 1: 允许使能 0: 禁止使能	R/W	0
6:5	Reserved			
4	CMP2CLKEN	CMP2 的时钟使能, 使能后 CMP2 才可以工作 1: 允许使能 0: 禁止使能	R/W	0
3	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	0
2	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	0
1	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	0
0	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	0

## 4.2 频率与最小工作电压的关系

系统时钟频率越高, MCU 的对最小工作电压 (安全工作电压) 的要求就越高, 例如, 2T 工作模式下如果要跑 16M, VDD 应至少在 2.2V 以上。AD18F06 的上电复位电压 POR 在 2.0V 左右, 即当上电过程 VDD 超过 POR 之后复位释放, 再经过约 1ms 延时之后完成上电配置,

然后开始执行程序指令。对于一些慢上电且要跑 16M/2T 的应用，如果 VDD 从 POR 上升到 2.2V 的  $V_{DD_{min}}$  的时间过长，而恰好在这段“死区时间”内程序切换到最高的系统时钟 16M 的话，MCU 将有可能跑飞。



上电过程的“死区时间”

对于慢上电的应用，有以下几种方法可避免这种情况发生：

1. 烧录选项 LVR 必须使能且设置合适的值，比如 16M/2T 应该设置 2.2V 的复位电压；
2. 上电复位后，软件可延时足够长的时间让 VDD 升到安全工作电压后再切换到 16M 的系统时钟，即延时一段时间再做时钟的初始化；
3. 使能 PWRT 选项，PWRT 时间大约为 64ms,这段额外的复位时间有利于让 VDD 爬升到最小工作电压；

以上 3 点中，强烈建议采用第 1 种方式，因为它不仅能解决上电速度过慢的问题，还能监测在正常工作时的 VDD 意外跌落；

### 4.3 HIRC 频率微调

内建的高精度 HIRC 出厂时被校准至 16MHz @ 2.5V/25°C。校准过程是过滤掉制程上的偏差对精度造成的影响，此 HIRC 还会到受工作环境温度和工作电压的影响，其频率会有一些的漂移。除了出厂校准外，还提供了一种方式供用户对 HIRC 进行微调：通过对 OSCTUNE 寄存器的值进行改写。OSCTUNE 的初始值确保 HIRC 在上电后工作在 16MHz，该值在每颗 IC 上会有差异。设初始值为 OSCTUNE[X]，此时芯片工作在 16M，每改变 1 个 LSB 则 HIRC 频率变化约为 40kHz。OSCTUNE[5:0]和 HIRC 输出的关系如下

OSCTUNE[5:0]值	HIRC 实际输出频率（16M 为例）
---------------	---------------------

OSCTUNE[X]-n	(16000-n*40)
.....	.....
OSCTUNE[X]-2	(16000-2*40)=15960
OSCTUNE[X]-1	(16000-1*40)=15980
OSCTUNE[X]	16000
OSCTUNE[X]+1	(16000+1*40)=16020
OSCTUNE[X]+2	(16000+2*40)=16040
.....	.....
OSCTUNE[X+n	(16000+n*40)

### OSCTUNE(HIRC 微调寄存器)

地址：0XFA0

Bit	Name	Description	Attribute	Reset
7	OSCTUNE_EN	频率微调使能位 1: 允许使能 0: 禁止使能	R/W	0
6:0	OSCTUNE[6:0]	HIRC 微调值,	R/W	0X00

## 4.4 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。此时 OSCI/OSCO 映射成为数字 IO，晶振 IO 功能关闭。**

## 4.5 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；



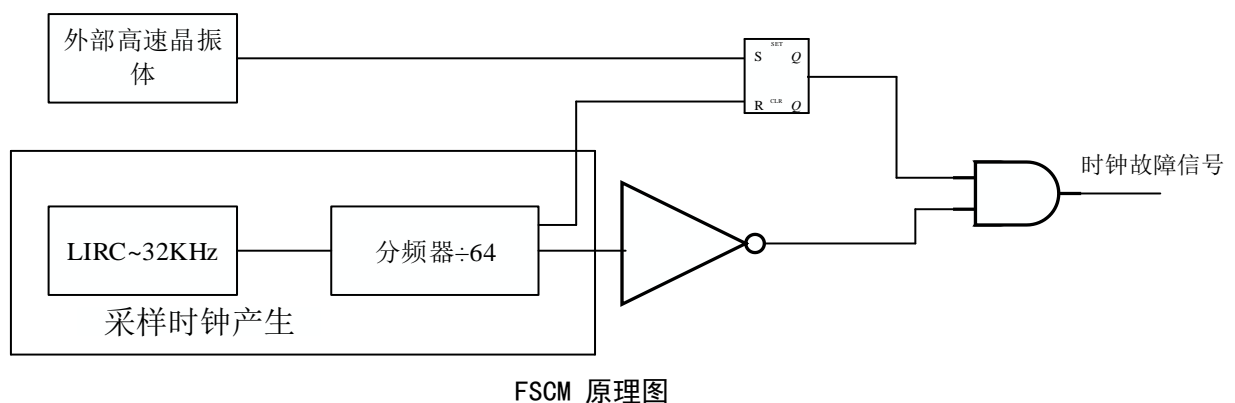
- ◇ 配置字选项 1 中的位 CRYHEN = 1，使能外部晶体高频振荡器。
- ◇ 在下列操作之后，进入双速启动模式：
- ◇ 从休眠状态唤醒。

## 4.6 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。

## 4.7 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。



### 4.7.1. 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 PIR1 寄存器的 OSFIF 标志位置 1。如果在 PIE1 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 OSCCON 寄存器 OSCM 决定，内部振荡器应该在故障发生前配置。

### 4.7.2. 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。

### 4.7.3. 故障保护条件清除

复位或者 OSCCON 寄存器的 OSCM 位的切换配置后，故障保护条件被清除。必须先清除故障保护条件，才能清零 OSFIF 标志位。

**注：经过一段时间后，读 OSCCON 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。**

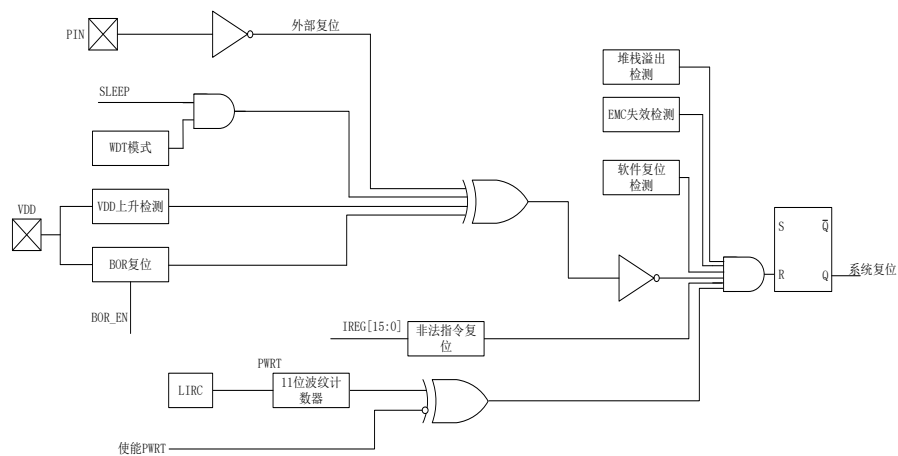
## 4.8 系统时钟相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xf4e	CLKCFG1	I2CCLKEN	UART1CLKEN	JART0CLKEN	CCPCLKEN	T3CLKEN	T2CLKEN	T1CLKEN	TOCLKEN	0000 0000	0000 0000
0xf4d	CLKCFG2	SPICLKEN			TOUCHCLKEN	DACCLKEN	ADCCLKEN	CMPCLKEN	OPCLKEN	0000 0000	0000 0000
0xf12	OSCCON	FINTOSC[2:0]			CKOE	OSCF[1:0]		OSCM[1:0]		0010 0001	0010 0001
0xf11	SIGCON	ICE_TMEREN	PLL					ENVDDL	SIGS	0--- ---0	0--- ---0
0xfa0	OSCTUNE	OSCTUNE[50]								--00 0000	--00 0000
0xfad	JPR1			OSFIP						1111 1111	1111 1111
0xfac	PIR1			OSFIF						0000 0000	0000 0000
0xfab	PIE1			OSFIE						0000 0000	0000 0000

## 5 复位和电源电压检测

AD18F06 单片机能通过以下方式复位:

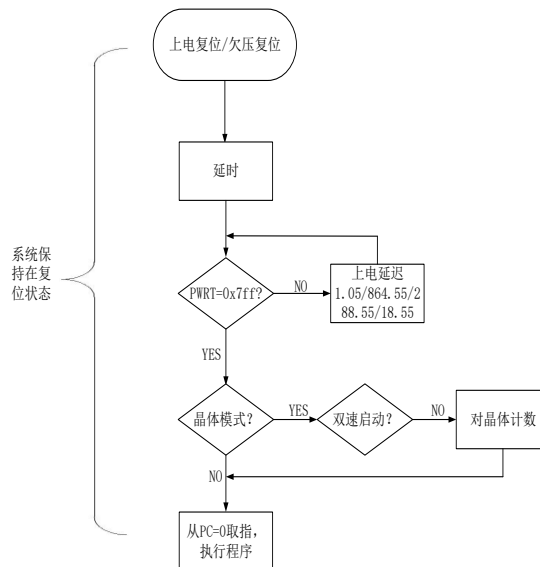
- 上电复位(POR)
- 掉电复位(Brown-out Reset BOR)
- 低电压复位
- 看门狗复位
- 非法指令复位
- 软件复位
- EMC 复位
- 外部管脚复位
- 堆栈溢出复位



复位框图

## 5.1 上电复位

片上的 POR 电路会将芯片保持在复位状态，直到 VDD 电源达到足够高。



上电复位流程图

## 5.2 低电压复位

低电压复位由配置字 0 中 LVR[3:0]和 LVREN 位来控制。低电压复位就是指当电源电压低于设定值时所产生的复位。

如果 LVREN 使能，电源电压在设定值以下，LVR 电路会将芯片控制在复位状态，直到 VDD 到达设定值以上，解除复位状态。

## 5.3 上电复位延时

复位模块内置了一个 11 位的上电复位定时器 PWRT 模块，上电复位计数器提供一个 (18.05/288.05/864.05/0.55ms)延迟时间，该延迟时间由配置字 2 的[6:5](PSUT[1:0])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 或看门狗溢出复位触发延迟时间。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。

## 5.4 非法指令复位

IRFEN 使能，当 CPU 的指令寄存器取指非法指令（未定义的操作码）时，复位标志位 RSTF[1:0]=2b11，同时系统将进行复位。利用此功能可增加系统的抗干扰能力。

## 5.5 软件复位

增强型内核实现了一条软件复位指令，助记符为 RESET，它提供了软件执行硬件的方法，复位标志位 RSTF[1:0]=2b01。

## 5.6 EMC 复位

上电后，内置了 EMC 关键寄存器对比电路，实时监控寄存器值是否产生变化，当寄存器初始值和监控值不一致时，EMC 硬件检测逻辑会发出一次复位，标志位为 RSTF[1:0]=2b10。EMCEN 控制该功能使能。

## 5.7 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。低电压复位后，电压高于 LVR 设定的电压值后最多 16ms CPU 工作。

### PCON(寄存器)

地址：0XFF1

Bit	Name	Description	Attribute	Reset
7	IPEN	高低优先级中断使能位 1: 允许高低优先级 0: 只允许高优先级中断	R/W	1
6	PB0ST	LVDM=11 时, PB0 输出值	R/W	0
5:4	LVDM[1:0]	电压比较中断 00: 禁止电压比较器 01: VDD 低于阈值电压产生中断 10: VDD 高于阈值电压产生中断 11: VDD 高于阈值电压产生中断, 且强制 PB0 输出为 PB0ST 值	R/W	00
	LVD[3:0]	LVR 电压选择		

3:0		0: 保留, 勿配置 1: 保留, 勿配置 2: 2.2V 3: 2.4V(默认) 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 LVT 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。

### RCON(寄存器)

地址: 0XFD2

Bit	Name	Description	Attribute	Reset
7	IRFEN	非法指令复位使能位 1: 使能非法指令复位 0: 禁止非法指令复位	R/W	0

6	EMCEN	EMC 复位使能位 1: 使能 EMC 复位 0: 禁止 EMC 复位	R/W	0
5:4	RSTF[1:0]	复位标志位 00:未发生复位 01: 软件指令复位 10: 表示发生 EMC 复位 11: 表示发生非法指令复位	R/W	00
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	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	1
1	POR	上电复位状态位 1: 未发生上电复位 0: 已发生上电复位（必须在上电复位后由软件置 1） 注：软件只能写 1	R/W	0
0	LVR	LVR 复位状态位 1: 未发生 LVR 复位 0: 已发生 LVR 复位（必须在 LVR 复位后由软件置 1） 注：软件只能写 1	R/W	0

## 5.8 /TO /PD 状态

/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

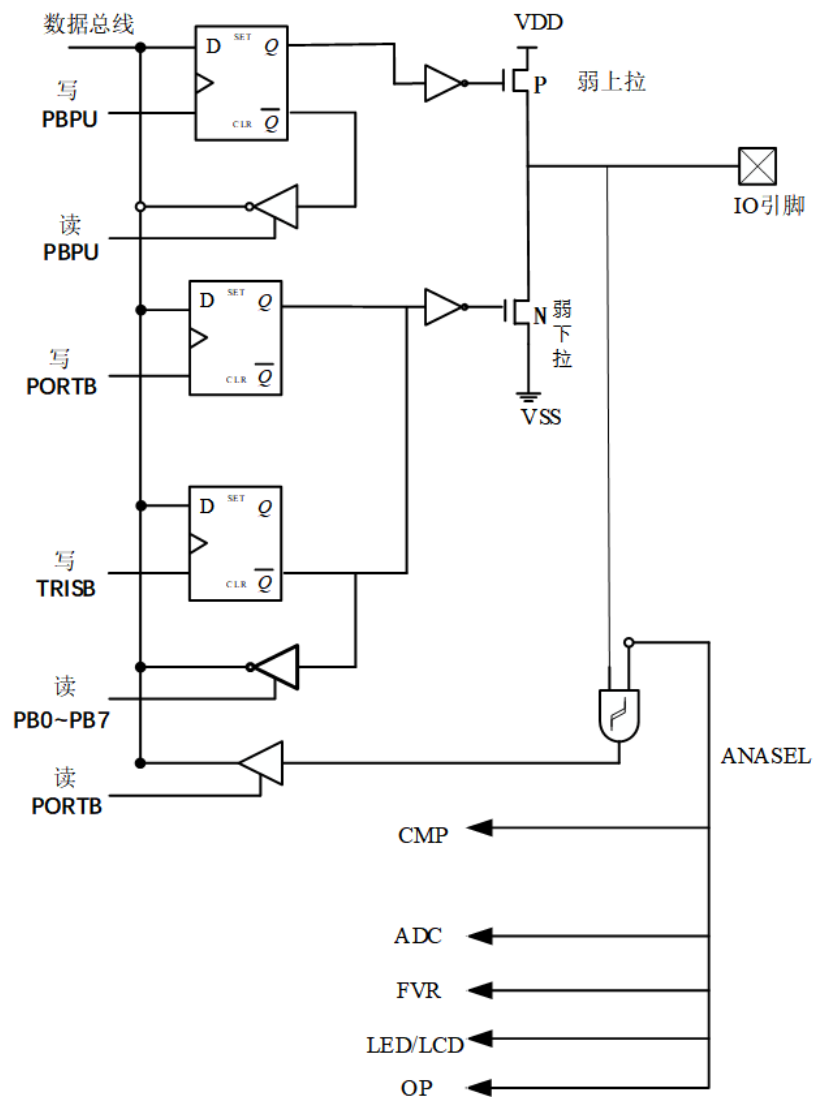
## 5.9 相关寄存器定义

B	C	D	E	F	G	H	I	J	K	L	M
address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0x0f2	INTCON	GIE/GIEH	PEIE/GIEL							00- ---	00- ---
0x0f1	PCON	IPEN	PB0ST	LVDLM[1:0]		LVT[3:0]				1000 0000	1000 0000
0x0f2	RCON	IRFEN	EMCEN	RSTF[1:0]		TO	PD	POR	BOR	0000 1100	0000 1100
0x0a7	IPR3	PDIP	PCIP	PBIP	PAIP	INT2IP	INT1IP	INT0IP	LVDIP	1111 1111	1111 1111
0x0a6	PIR3	PDIF	PCIF	PBIF	PAIF	INT2IF	INT1IF	INT0IF	LVDIF	0000 0000	0000 0000
0x0a5	PIE3	PDIE	PCIE	PBIE	PAIE	INT2IE	INT1IE	INT0IE	LVDIE	0000 0000	0000 0000



## 6 I/O 端口

PortA、PortB、PortC、PortD 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、TRISD)设置。PA、PB、PC 和 PD 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、PC 和 PD 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、PC 和 PD 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。



IO 结构图

## 6.1 IO 工作模式

### 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

### PORTB (Port 寄存器)

地址: 0XF81

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

### PORTC (Port 寄存器)

地址: 0XF82

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

### PORTD (Port 寄存器)

地址: 0XF83

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

### 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 0:IO 的输出模式	R/W	0x3F

**TRISB (I/O 口方向控制寄存器)**

地址: 0XF91

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

**TRISC (I/O 口方向控制寄存器)**

地址: 0XF92

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

**TRISD (I/O 口方向控制寄存器)**

地址: 0XF93

Bit	Name	Description	Attribute	Reset
7:0	TRISD[7:0]	1:IO 的输入模式 0:IO 的输出模式	R/W	0xFF

**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

**PINB (数据锁存寄存器)**

地址: 0XF89

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

**PINC (数据锁存寄存器)**

地址: 0XF8A

Bit	Name	Description	Attribute	Reset
7:0	PINC[7:0]	读: 读取当前 IO 状态; 写: 写入 PORTC 寄存器;	R/W	0x00

**PIND (数据锁存寄存器)**

地址: 0XF8B

Bit	Name	Description	Attribute	Reset
7:0	PIND[7:0]	读: 读取当前 IO 状态; 写: 写入 PORTD 寄存器;	R/W	0x00

## 6.2 下拉电阻开漏

**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

**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

**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

**PBPD(I/O 下拉控制寄存器)**

地址: 0XF8D

Bit	Name	Description	Attribute	Reset
7:0	PBnPD[7:0]	1: 关闭内部下拉 0: 使能内部下拉	R/W	0XFF

**PBPU (I/O 上拉控制寄存器)**

地址: 0XF85

Bit	Name	Description	Attribute	Reset
7:0	PBnPU[7:0]	1:关闭内部上拉 0:使能内部上拉	R/W	0xFF

**PBOD (I/O 开漏控制寄存器)**

地址: 0XF95

Bit	Name	Description	Attribute	Reset
7:0	PBnPD[7:0]	1:关闭内部下拉 0:使能内部下拉	R/W	0x00

**PCPD(I/O 下拉控制寄存器)**

地址: 0XF8E

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

**PCPU (I/O 上拉控制寄存器)**

地址: 0XF86

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

**PCOD (I/O 开漏控制寄存器)**

地址: 0XF96

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

**PDPD(I/O 下拉控制寄存器)**

地址: 0XF8F

Bit	Name	Description	Attribute	Reset
7:0	PDnPD[7:0]	1: 关闭内部下拉 0: 使能内部下拉	R/W	0XFF

**PDPU (I/O 上拉控制寄存器)**

地址: 0XF87

Bit	Name	Description	Attribute	Reset
7:0	PDnPU[7:0]	1:关闭内部上拉 0:使能内部上拉	R/W	0XFF

**PDOD (I/O 开漏控制寄存器)**

地址: 0XF957

Bit	Name	Description	Attribute	Reset
7:0	PDnPD[7:0]	1:关闭内部下拉 0:使能内部下拉	R/W	0X00

## 6.3 IO 中断寄存器

**PAINTMASK (Port A 端口电平变化中断掩膜位)**

地址: 0XF4F

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

**PBINTMASK (Port B 端口电平变化中断掩膜位)**

地址: 0XF4E

Bit	Name	Description	Attribute	Reset
7:0	PINTMASKB[7:0]	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

**PCINTMASK (Port C 端口电平变化中断掩膜位)**

地址：0XF4D

Bit	Name	Description	Attribute	Reset
7:0	PINTMASKC[7:0]	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

PDINTMASK (Port D 端口电平变化中断掩膜位)

地址：0XF4C

Bit	Name	Description	Attribute	Reset
7:0	PINTMASKD[7:0]	1: 使能 PortD 端口电平变化中断允许位 0: 禁止 PortD 端口电平变化中断允许位	R/W	0x00

## 6.4 IO 模拟控制寄存器

ANASEL0 (IO 的模拟通道寄存器)

地址：0XF9F

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

ANASEL1 (IO 的模拟通道寄存器)

地址：0XF9E

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

ANASEL2 (IO 的模拟通道寄存器)

地址：0XF9D

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

**ANASEL3 (IO 的模拟通道寄存器)**

地址: 0XF9C

Bit	Name	Description	Attribute	Reset
7:0	PDnAEN[7:0]	1: PD 口打开模拟通道 0: PD 口关闭模拟通道	R/W	0X00

**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

**SMTVB (PB 口施密特寄存器)**

地址: 0XF99

Bit	Name	Description	Attribute	Reset
7:0	SMTVB[7:0]	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	0XFF

**SMTVC(PC 口施密特寄存器)**

地址: 0XF9A

Bit	Name	Description	Attribute	Reset
7:0	SMTVC[7:0]	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	0XFF



**SMTVD (PD 口施密特寄存器)**

地址: 0XF9B

Bit	Name	Description	Attribute	Reset
7:0	SMTVD[7:0]	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	0XFF

**PDSINKCUR(IO 控制寄存器)**

地址: 0XF0F

Bit	Name	Description	Attribute	Reset
7	PD7CUR	PD7 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0
6	PD6CUR	PD6 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0
5	PD5CUR	PD5 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0
4	PD4CUR	PD4 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0
3	PD3CUR	PD3 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0
2	PD2UR	PD2 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0
1	PD1CUR	PD1 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0
0	PD0CUR	PD0 口的电流驱动(SINK 电流) 0: 低档位 1: 高档位	R/W	0

**CURCON(IO 控制寄存器)**

地址: 0XF0E

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:4	PCCUR	PC 口的电流驱动(SOURCE 电流) X0: 低档位 01: 中档位 11: 高档位	R/W	11
3:2	PBCUR	PB 口的电流驱动(SOURCE 电流) X0: 低档位 01: 中档位 11: 高档位	R/W	11
1:0	PACUR	PA 口的电流驱动(SOURCE 电流) X0: 低档位 01: 中档位 11: 高档位	R/W	11

## 6.5 外设功能引脚全映射控制

### 6.5.1. 外设功能引脚映射控制寄存器

寄存器地址	寄存器名称	寄存器地址	寄存器名称
0XEFF	TX0_MAP	0XEF5	PWM11_MAP
0XEFE	RX0_MAP	0XEF4	PWM20_MAP
0XEFD	TX1_MAP	0XEF3	PWM21_MAP
0XEFC	RX1_MAP	0XEF2	PWM2_MAP
0XEFB	SDA_MAP	0XEF1	PWM3_MAP
0XEFA	SCL_MAP	0XEF0	PWM4_MAP
0XEF9	PWM0_MAP	0XEEF	PWM5_MAP
0XEF8	PWM00_MAP		

0XEF7	PWM01_MAP		
0XEF6	PWM10_MAP		

### TX0\_MAP(IO 映射寄存器)

地址：0XEFF

Bit	Name	Description	Attribute	Reset
7:5	Reserved			
4:3	TX0_MPORT[1:0]	TX0 的映射端口选择  00: PA 01: PB 10: PC 11: PD	R/W	00
2:0	TX0_MPIN[2:0]	TX0 的映射端口输出引脚选择  TX0_MPIN[2:0] =n (n=0...7), 表示选择对应端口名的 n (n=0...7)脚  注：选择 PA 口时(n=0...5)	R/W	000

注意：1、输出功能，系统将选择优先级最高的进行多对一映射；但输入允许其映射到多个模块内部。

2、以上寄存器的复位为 0X00;这样复位后 IO 都为 GPIO,用户在使用外设功能引脚之前必须先配置上面寄存器，否则外设功能无法使用；

举例说明：

1、将 PWM0 的输出映射到 PB1 上，用户在启动 PWM0 之前应该配置下面寄存器：

PWM0\_MAP=0X09;PWM0→PB1;

2、如果想在下一次使 PWM0 输出映射到 PD2 上，需要将 PWM0\_MAP 寄存器设置为：

PWM0\_MAP=0X1A;

多个输出映射到一个端口上时，只能有一个输出有效，下面是默认的优先级：

优先级顺序	复用端口功能
1	TX0
2	RX0
3	TX1
4	RX1
5	SDA
6	SCL
7	PWM0
8	PWM00
9	PWM01
10	PWM10
11	PWM11
12	PWM20
13	PWM21
14	PWM2
15	PWM3
16	PWM4
17	PWM5

## 6.6 IO 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xf9f	ANASEL0			PA5AEN	PA4AEN	PA3AEN	PA2AEN	PA1AEN	PA0AEN	--00 0000	--00 0000
0xf9e	ANASEL1	PB7AEN	PB6AEN	PB5AEN	PB4AEN	PB3AEN	PB2AEN	PB1AEN	PB0AEN	0000 0000	0000 0000
0xf9d	ANASEL2	PC7AEN	PC6AEN	PC5AEN	PC4AEN	PC3AEN	PC2AEN	PC1AEN	PC0AEN	1111 1111	1111 1111
0xf9c	ANASEL3	PD7AEN	PD6AEN	PD5AEN	PD4AEN	PD3AEN	PD2AEN	PD1AEN	PD0AEN	0000 0000	0000 0000
0xf9b	SMTD	PD7SMT	PD6SMT	PD5SMT	PD4SMT	PD3SMT	PD2SMT	PD1SMT	PD0SMT	1111 1111	1111 1111
0xf9a	SMTC	PC7SMT	PC6SMT	PC5SMT	PC4SMT	PC3SMT	PC2SMT	PC1SMT	PC0SMT	1111 1111	1111 1111
0xf99	SMTB	PB7SMT	PB6SMT	PB5SMT	PB4SMT	PB3SMT	PB2SMT	PB1SMT	PB0SMT	1111 1111	1111 1111
0xf98	SMTA			PA5SMT	PA4SMT	PA3SMT	PA2SMT	PA1SMT	PA0SMT	--11 1111	--11 1111
0xf97	PDOD	PD7OD	PD6OD	PD5OD	PD4OD	PD3OD	PD2OD	PD1OD	PD0OD	0000 0000	0000 0000
0xf96	PCOD	PC7OD	PC6OD	PC5OD	PC4OD	PC3OD	PC2OD	PC1OD	PC0OD	0000 0000	0000 0000
0xf95	PBOD	PB7OD	PB6OD	PB5OD	PB4OD	PB3OD	PB2OD	PB1OD	PB0OD	0000 0000	0000 0000
0xf94	PAOD			PA5OD	PA4OD	PA3OD	PA2OD	PA1OD	PA0OD	--00 0000	--00 0000
0xf93	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111
0xf92	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
0xf91	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
0xf90	TRISA			TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
0xf8f	PDPD	PD7PD	PD6PD	PD5PD	PD4PD	PD3PD	PD2PD	PD1PD	PD0PD	1111 1111	1111 1111
0xf8e	PCPD	PC7PD	PC6PD	PC5PD	PC4PD	PC3PD	PC2PD	PC1PD	PC0PD	1111 1111	1111 1111
0xf8d	PBPD	PB7PD	PB6PD	PB5PD	PB4PD	PB3PD	PB2PD	PB1PD	PB0PD	1111 1111	1111 1111
0xf8c	PAPD			PA5PD	PA4PD	PA3PD	PA2PD	PA1PD	PA0PD	--11 1111	--11 1111
0xf8b	PIND			PORTD PIN Status[7:0]						0000 0000	uuuu uuuu
0xf8a	PINC			PORTC PIN Status[7:0]						0000 0000	uuuu uuuu
0xf89	PINB			PORTB PIN Status[7:0]						0000 0000	uuuu uuuu
0xf88	PINA			PORTA PIN Status[5:0]						0000 0000	uuuu uuuu
0xf87	PDPU	PD7PU	PD6PU	PD5PU	PD4PU	PD3PU	PD2PU	PD1PU	PD0PU	1111 1111	1111 1111
0xf86	PCPU	PC7PU	PC6PU	PC5PU	PC4PU	PC3PU	PC2PU	PC1PU	PC0PU	1111 1111	1111 1111
0xf85	PBPU	PB7PU	PB6PU	PB5PU	PB4PU	PB3PU	PB2PU	PB1PU	PB0PU	1111 1111	1111 1111
0xf84	PAPU			PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU	--11 1111	--11 1111
0xf83	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	00000 0000	uuuu uuuu
0xf82	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	00000 0000	uuuu uuuu
0xf81	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	00000 0000	uuuu uuuu
0xf80	PORTA			PA5	PA4	PA3	PA2	PA1	PA0	--00 0000	uuuu uuuu
0xfa7	IPR3	PDIP	PCIP	PBIP	PAIP	INT2IP	INT1IP	INT0IP	LVDIP	1111 1111	1111 1111
0xfa6	PIR3	PDIF	PCIF	PBIF	PAIF	INT2IF	INT1IF	INT0IF	LVDIF	0000 0000	0000 0000
0xfa5	PIE3	PDIE	PCIE	PBIE	PAIE	INT2IE	INT1IE	INT0IE	LVDIE	0000 0000	0000 0000
0xf4c	PAINTMASK			PA5MSK	PA4MSK	PA3MSK	PA2MSK	PA1MSK	PA0MSK	--00 0000	--00 0000
0xf4b	PBINTMASK	PB7MSK	PB6MSK	PB5MSK	PB4MSK	PB3MSK	PB2MSK	PB1MSK	PB0MSK	0000 0000	0000 0000
0xf4a	PCINTMASK	PC7MSK	PC6MSK	PC5MSK	PC4MSK	PC3MSK	PC2MSK	PC1MSK	PC0MSK	0000 0000	0000 0000
0xf49	PDINTMASK	PD7MSK	PD6MSK	PD5MSK	PD4MSK	PD3MSK	PD2MSK	PD1MSK	PD0MSK	0000 0000	0000 0000
0xf0f	PDSINKCUR	PD7SCUR	PD6SCUR	PD5SCUR	PD4SCUR	PD3SCUR	PD2SCUR	PD1SCUR	PD0SCUR	0000 0000	0000 0000
0xf0e	CURCON			PCCUR[1:0]		PBCUR[1:0]		PACUR[1:0]		--11 1111	--11 1111

## 7 定时器

### 7.1 Timer0 8 位定时/计数器

注：TIMER0、在工作之前需要开启模块时钟，设置 CLKCFG1[0]即 T0CLKEN=1；

#### 7.1.1. Timer0 计数/定时

Timer0 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 OSC32K 时钟源、内部 OSC1M 时钟源、T1BUZZ 输出以及比较器 CMP0OUT、CMP1OUT、CMP2OUT 的输出，使用外部时钟需要设置 T0CON 的 T0CS0、T0CS1、T0CS2 共同决定。若需要重新装载 TMR0 的初值，则需要中断函数中进行初值的重新装载：

注：4T 模式下定时：Time = (PR0+1) \* {PS2:PS0} \* 4 / Fsys; //TMR0 的初值为 0 时

Time = (0xFF - [TMR0] + 1) \* {PS2:PS0} \* 4 / Fsys; //TMR0 设定初值

2T 模式下定时：Time = (PR0+1) \* {PS2:PS0} \* 2 / Fsys; //TMR0 的初值为 0 时

Time = (0xFF - [TMR0] + 1) \* {PS2:PS0} \* 2 / Fsys; //TMR0 设定初值

```

//4T模式下
T0CLKEN=1;
TMR0IE=1;
TMR0IF=0
TMR0IP =1;

PR0=0xf9;
TMR0=0x00;

T0CON0=0x03;//{PS2:PS0}=3
    
```

---

```

//2T模式下
T0CLKEN=1;
TMR0IE=1;
TMR0IF=0
TMR0IP =1;

PR0=0xf9;
TMR0=0x00;

T0CON0=0x04;//{PS2:PS0}=4
    
```

### 7.1.2. 使用内部时钟: 定时模式

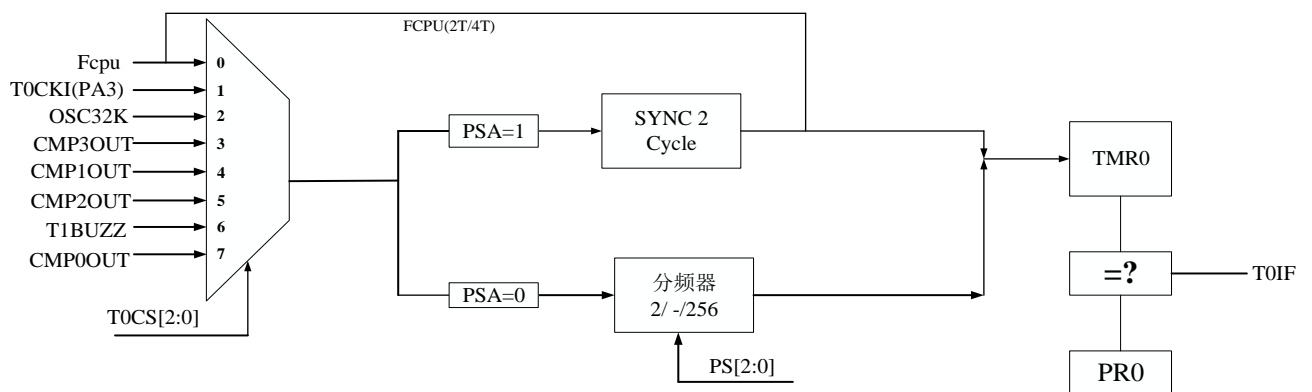
定时模式在没有预置器的情况下, 定时寄存器每个指令周期自动加 1, 设置 TMR0 以后, 定时器将在两个时钟周期以后开始自增。

### 7.1.3. 使用外部时钟/内部 32K 时钟/: 计数模式

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

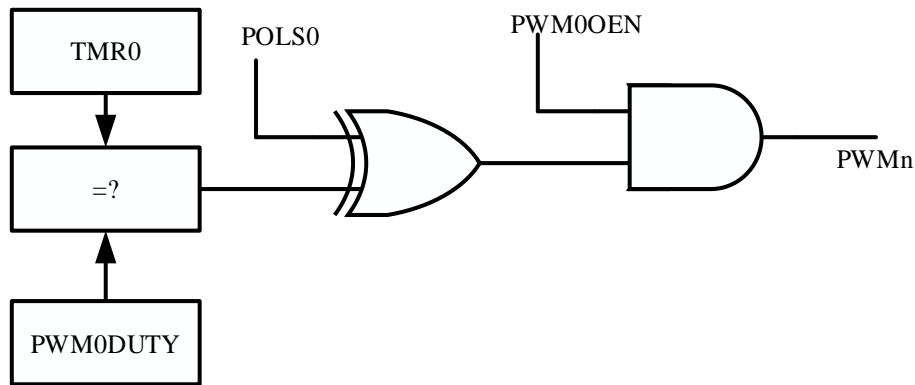
### 7.1.5. BUZZER (BUZZER 输出)

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, TMR0IF 有效, 且当 TMR0IE=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 1 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 的输出映射所有 IO，参考 6.4 外设功能引脚全映章节的 PWM0\_MAP 寄存器。

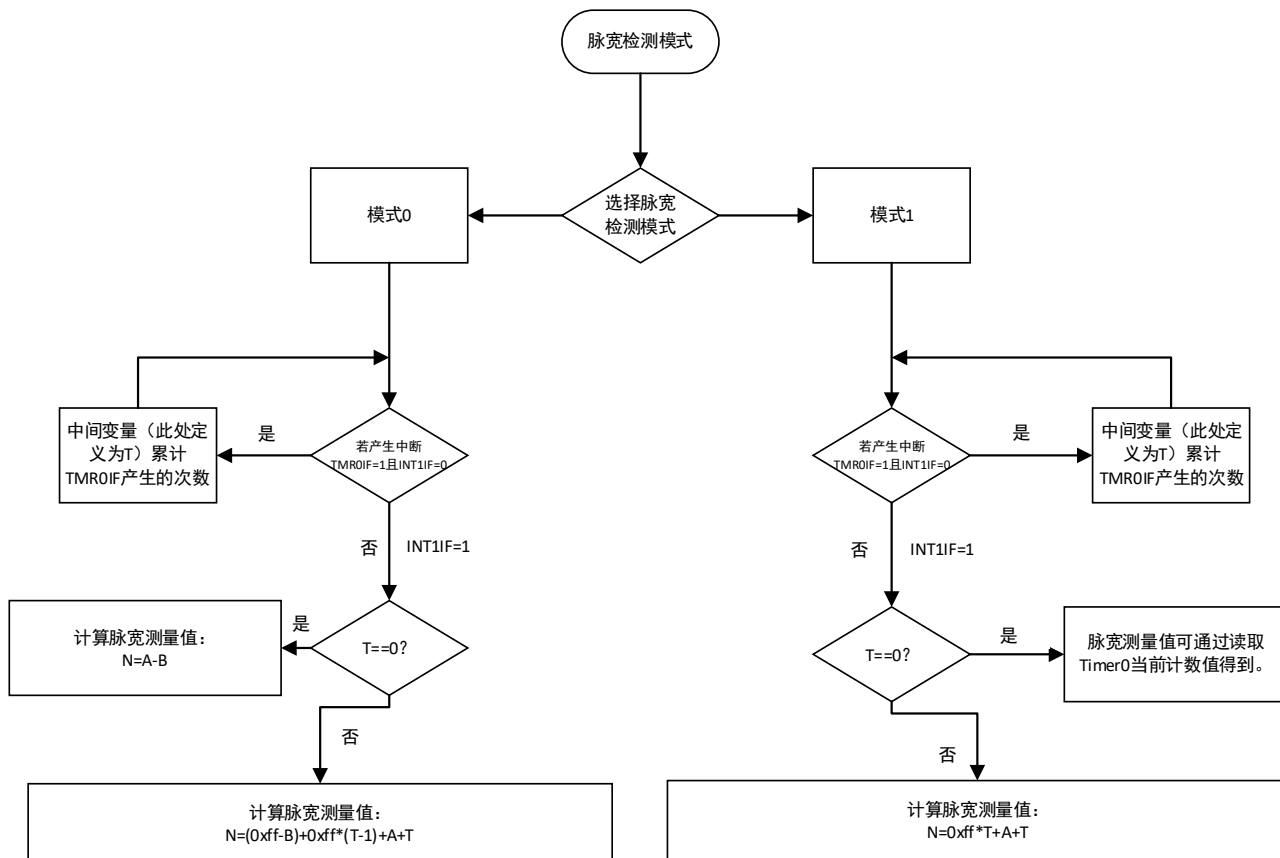
### 7.1.7. TMR0 工作模式

TIMER0 支持定时/计数器工作模式，支持 IO 脉宽测量模式、支持定时启动 ADC 工作模式。

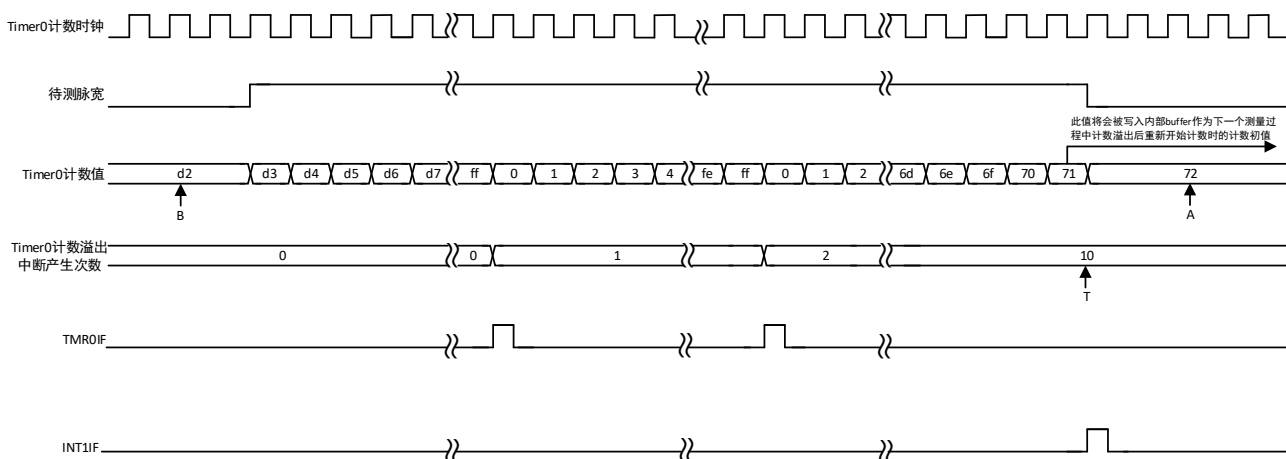
Timer0 支持两种脉宽检测模式。脉宽测量过程结束将产生 INT1 中断，可由此判断脉宽测量过程结束与否。下图为脉宽测量模式的参考应用流程。其中，A 表示脉宽测量过程结束后的 8 位 Timer0 计数值，B 表示脉宽测量过程开始前的 8 位 Timer0 计数值，T 表示脉宽测量过程中 Timer0 计数溢出中断产生的次数。

测量过程结束后，通过读取 8 位 Timer0 计数值，结合测量过程中的计数溢出中断产生次数，以及测量开始前的 8 位 Timer0 值计算得到实际测量值。





参考例程：时序图如下，此例程中，使用脉宽测量模式 1，在待测脉宽的上升沿启动计数，下降沿停止计数，此待测脉宽上升沿与下降沿之间的持续时间为 1000us，Timer0 计数时钟频率为 4MHz，即计数一次所用时间为 0.25us。查看波形，参考上述应用流程，得  $A=0x72$ ， $B=0xd2$ ， $T=0x10$ ，则得脉宽测量值  $N=(0xff-0xd2)+0xff*(0x10-1)+0x72+0x10=0xfa0$ ， $0xfa0$  换算为十进制为 4000， $4000*0.25=1000$  (us)，测量结果正确。



**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	0X00

**T0CON0 Register (TMR0 控制寄存器 0)**

地址: 0XFD7

Bit	Name	Description	Attribute	Reset
7	T0OUT	溢出输出 BUZZEE (PA4) 0: 禁止 BUZZEE 模式 1: 启动 BUZZEE 模式	R/W	0
6:4	T0CS[2:0]	TMR0 时钟源选择控制位 000: TMR0 时钟源为 CPU 运行时钟 001: TMR0 时钟源为 T0CKI(PA3) 010: TMR0 时钟源为内部 32K 011: TMR0 时钟源是 CMP3OUT 输出 100: TMR0 时钟源是 CMP1OUT 输出 101: TMR0 时钟源是 CMP2OUT 输出 110: TMR0 时钟源是 T1BUZZ 111: TMR0 时钟源是 CMP0OUT 输出	R/W	111

3	PSA	TMR0 的时钟分频使能 1:使能 TMR0 的时钟且不分频; 0:使能 TMR0 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  $F_{CPU}/8$ ;

### T0CON1 (TMR0 控制寄存器 1)

地址: 0XFD3

Bit	Name	Description	Attribute	Reset
7	T1BUZZEN	TIMER0 的 PWM0 输出信号选择 0: 选择 PWM0 的输出 1: 选择 T1BUZZ(PD6)的输出	R/W	0
6	T0EDGE	在 TOM[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
5:4	T0CH[1:0]	TIMER0 脉宽信号输入选择 00: T0CH0 作为脉宽检测输入信号(PC2) 01: T0CH1 作为脉宽检测输入信号(PD0) 10: T0CH2 作为脉宽检测输入信号(PB2) 11: CMP3OUT 作为脉宽检测输入信号	R/W	00
3	POLS0	PWM0 通过 IO 输出的极性 1: 反向输出 0: 正向输出	R/W	0
2	PWM0EN	PWM0 输出使能,通过 IO 输出 1: 使能 PWM0 输出 0: 禁止 PWM0 输出	R/W	0

1:0	T0M[1:0]	<p>TIMER0 工作模式</p> <p>00: TIMER0 工作在普通模式, 溢出时产生中断标志位。</p> <p>01: 脉宽测量模式 0, 测量上升沿与下降沿之间的时间, TOEDGE=1 时, 在脉冲的上升沿开始计数, 脉冲下降沿停止计数, 并在脉冲下降沿触发 INT1 中断。TOEDGE=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 INT1 中断。产生 INT1 中断标志后, 可以直接读取 TIMER0 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, TOEDGE=1 时在脉冲的上升沿 TIMER0 数据寄存器发生复位从 0x00 开始计数, 并在下降沿触发 INT1 中断。TOEDGE=0 时, 在脉冲的下降沿 TIMER0 数据寄存器发生复位从 0x00 开始计数, 在脉冲的上升沿触发 INT1 中断。产生 INT1 中断标志后, 可以直接读取 TIMER0 内容, 就可以得到测量值。</p> <p>11: 模式 2, TIMER0 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
-----	----------	---	-----	----

### 7.1.8. 寄存器列表

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por&bor reset value	other reset value
0xfd7	TOCON	TOOUT	TOCS2	TOCS1	TOCS0	PSA	PS2	PS1	PS0	0111 1111	0111 1111
0xfd6	PR0	PR0[7:0]								1111 1111	1111 1111
0xfd5	TMR0	TMR0[7:0]								0000 0000	0000 0000
0xfd4	PWM0DUTY	PWM0DUTY[7:0]								0000 0000	0000 0000
0xfd3	TOCON1	T1BUZZEN	TOEDGE	TOCH		POLS0	PWM0EN	T0M1	T0M0	0000 0000	0000 0000
0xfad	IPR1								TMR0IP	---- --1	---- --1
0xfac	PIR1								TMR0IF	---- --0	---- --0
0xfab	PIE1								TMR0IE	---- --0	---- --0

## 7.2 TIMER1 16 位定时/计数器

注：TIMER1 在工作之前需要开启模块时钟，设置 CLKCFG1[1]即 T1CLKEN=1；

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

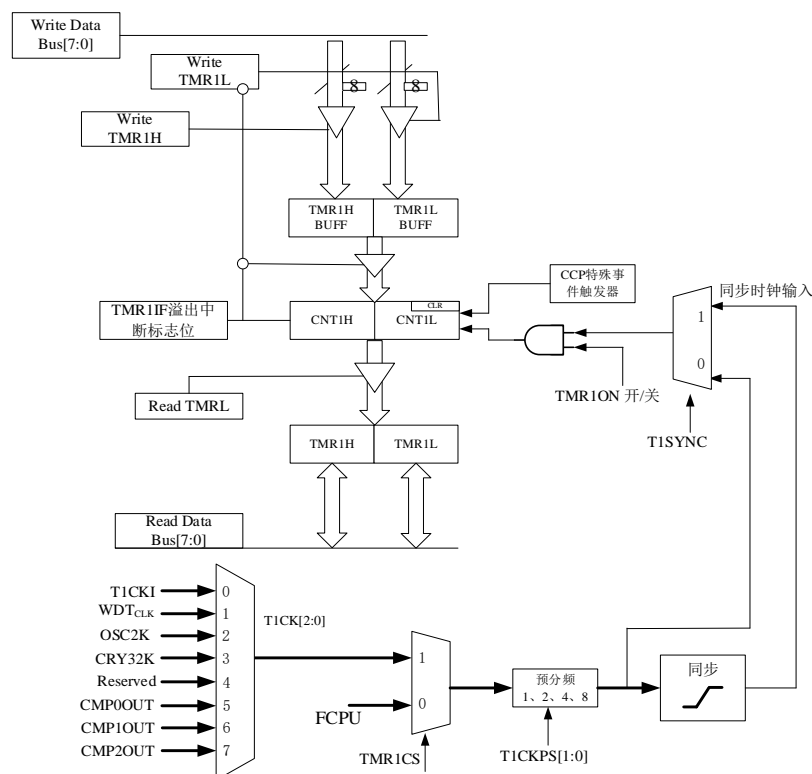
TIMER1 为 16 位定时/计数器，内部拥有写入缓冲区，当溢出时，自动从写入缓冲区装载到 TIMER1 计数器作为初值，然后每个 TIMER1 时钟周期，计数递增。当 TIMER1 计数值从 0XFFFF 变为 0X0000 时产生溢出中断，并自动重载写入缓冲区数值。TIMER1 的时钟源可以是外部时钟源(TICKI pin)、内部 WDT 时钟、内部 32K 时钟、外部低速晶振 LXTL、CMP 输出等。

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：4T 模式下定时： $Time = (0xFFFF - [TMR1] + 1) * \{T1CKPS1: T1CKPS0\} * 4 / F_{sys}$ ;  
//TMR1 的初值为设定值

2T 模式下定时： $Time = (0xFFFF - [TMR1] + 1) * \{T1CKPS1: T1CKPS0\} * 2 / F_{sys}$ ;

//TMR1 的初值为设定值

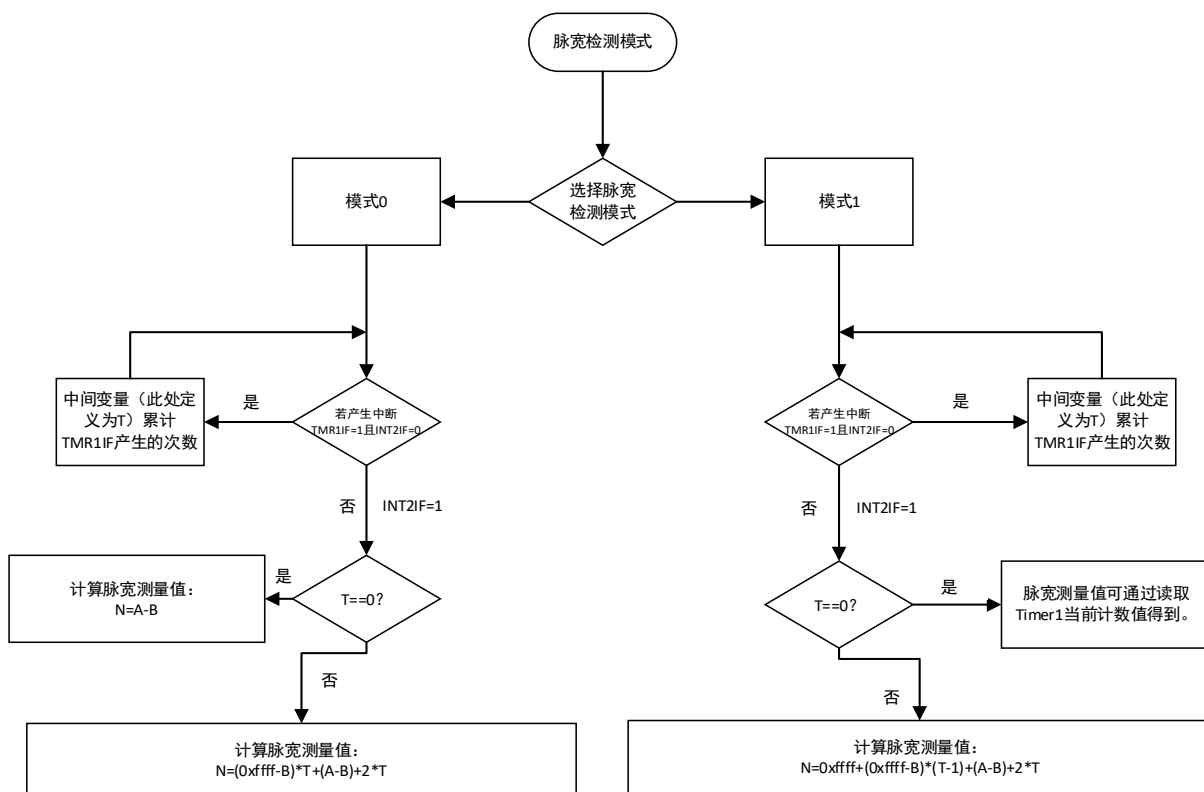


Timer1 的原理图

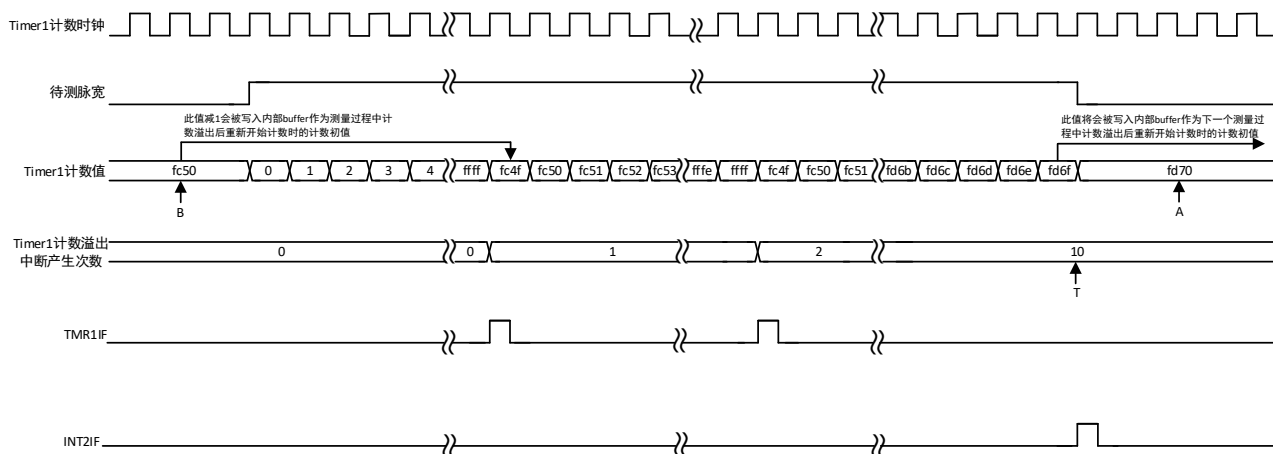
注：CNT1H 和 CNT1L 是 TIMER1 的内部寄存器。

Timer1 支持两种脉宽检测模式。脉宽测量过程结束将产生 INT2 中断，可由此判断脉宽测量过程结束与否。下图为脉宽测量模式的参考应用流程。其中，A 表示脉宽测量过程结束后的 16 位 Timer1 计数值，B 表示脉宽测量过程开始前的 16 位 Timer1 计数值，T 表示脉宽测量过程中 Timer1 计数溢出中断产生的次数。

测量过程结束后，通过读取 16 位 Timer1 计数值，结合测量过程中的计数溢出中断产生次数，以及测量开始前的 16 位 Timer1 值计算得到实际测量值。



参考例程：时序图如下，此例程中，使用脉宽测量模式 1，在待测脉宽的上升沿启动计数，下降沿停止计数，此待测脉宽上升沿与下降沿之间的持续时间为 10000us，Timer1 计数时钟频率为 8MHz，即计数一次所用时间为 0.125us。查看波形，参考上述应用流程，得 A=0xfd70，B=0xfc50，T=0x10，则得脉宽测量值  $N=0xffff+(0xffff-0xfc50)*(0x10-1)+(0xfd70-0xfc50)+2*0x10=0x13880$ ，0x13880 换算为十进制为 80000， $80000*0.125=10000$  (us)，测量结果正确。



### 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	0X00

### 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	0X00

**TMR1 寄存器在写寄存器时，必须先写高字节再写低字节；读先读低字节在读高字节；**

### T1CON0 (Timer1 控制寄存器)

地址: 0XFCF

Bit	Name	Description	Attribute	Reset
7:5	T1CK[2:0]	Timer1 的计数时钟源选择 000: T1CKI(PB2)作为计数时钟 001: WDT <sub>CLK</sub> 作为计数时钟 010: 内部 32K 作为计数时钟 011: 外部晶振 LXTL 作为计数时钟 100: Reserved 101: Reserved 110: CMP1OUT 的输出作为计数时钟 111: CMP2OUT 的输出作为计数时钟	R/W	000

		注意：使用比较器输出作为时钟时，需要使能相应比较器。		
4:3	T1CKPS[1:0]	Timer1 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
2	T1SYNC	1: 使用 FCPU 同步分频后时钟作为 TIMER1 时钟（注意：T1SYNC 置 1 时，不支持选择内部时钟，同时也不支持 1:1 分频） 0: 使用选择分频时钟作为 TIMER1 时钟	R/W	0
1	TMR1CS	1: 选择 T1CK[2:0]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：在 4T 和 2T 模式下，T1CKI 的时钟选择必须小于  $F_{CPU}/8$ ；

### 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,外部时钟引脚(PD1) 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,外部时钟引脚(PC0) 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,外部时钟引脚(PB3) 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:T1CH0	TIMER1 脉宽信号输入选择 00: T1CH0 作为脉宽检测输入信号(PB1) 01: T1CH1 作为脉宽检测输入信号(PB0) 10: T1CH2 作为脉宽检测输入信号(PB3) 11:TMR2CLK 作为脉宽检测输入信号(配置 T2CON0<2>=1)	R/W	00
1:0	T1M[1:0]	T TIMER1 工作模式 00: TIMER1 工作在普通模式, 溢出时间产生中断标志位。 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, 并在脉冲下降沿触发 INT2 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, 并触发 INT2 中断。INT2 中断产生后, 通过读取 TIMER1 内容结合测量过程中 TMR1IF 产生的次数以及测量开始前的 T1BUFF 值进行计算便可以得到测量值。 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, 并在下降沿触发 INT2 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 INT2 中断。INT2 中断产生后, 通过读取 TIMER1 内容结合测量过程中 TMR1IF 产生的次数以及测量开始前的 T1BUFF 值进行计算便可以得到测量值。 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。	R/W	00

注: TIBUZZ:在 T1M[1:0]=11 且 TMR1 计数到 0XFFFF 后, T1BUZZ 输出通过 PD6 引脚;

## TIMER1 相关寄存器定义

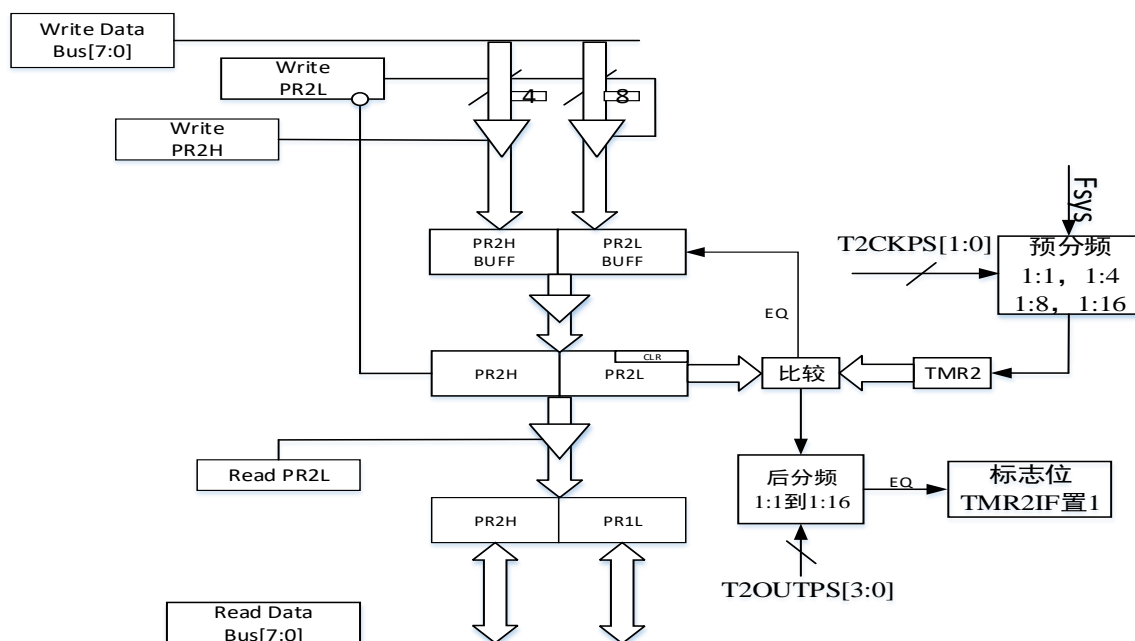
address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xfd1	TMR1H	Timer1 Register High Byte								0000 0000	uuuu uuuu
0xf0	TMR1L	Timer1 Register Low Byte								0000 0000	uuuu uuuu
0xfc	TICON0	TICK2	TICK1	TICK0	TICKPS1	TICKPS0	TISYNC	TMR1CS0	TMR1ON	0000 0000	0000 0000
0xfc	TICON1	INT2EDGE	INT1EDGE	INT0EDGE	TLEDGE	T1CH1	T1CH0	TMR1IP	TIM0	0000 0000	0000 0000
0xfad	IPR1							TMR1IF		1111 1111	1111 1111
0xfac	PIR1							TMR1IF		0000 0000	0000 0000
0xfab	PIE1							TMR1IE		0000 0000	0000 0000

## 7.3 TIMER2 12 位定时器

注：TIMER2 在工作之前需要开启模块时钟，设置 CLKCFG1[2]即 T2CLKEN=1；

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）
- 可软件编程的后分频器（分频比为 1:1 至 1:16）
- 当 TMR2（TMR2H, TMR2L）与 PR2（PR2H, PR2L）匹配时产生中断
- 采用系统时钟 Fosc 控制
- 当 TMR2 运行时，写入 PR2 会先写入 PR2 BUFFER 中，当 TMR2 与 PR2 相等时，PR2 BUFFER 更新到 PR2 中，可以用于调频



TIMER2 结构图

PR2L（timer2 的周期寄存器）

地址：0XFCB

Bit	Name	Description	Attribute	Reset
7:0	PR2[7:0]	Timer2 的周期寄存器的低 8 位	R/W	0XFF

**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	0X0F

**TMR2L (Timer2 的低八位寄存器)**

地址:0XFCD

Bit	Name	Description	Attribute	Reset
7:0	TMR2[7:0]	Timer2 定时/计数器的低 8 位	R/W	0X00

**TMR2H (Timer2 的高八位寄存器)**

地址: 0XFCC

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	TMR2[11:8]	Timer2 定时/计数器的高 4 位	R/W	0X00

- 注: 1、当 TMR2 运行时, 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;  
 2、当 TMR2 运行时, TMR2 寄存器读取时, 必须先读低字节在读高字节;

**T2CON(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	R/W	0000

		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		
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	00

### TMR2 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xfcd	TMR2L	TMR2L[7:0]								0000 0000	0000 0000
0xfce	TMR2H	TMR2H[11:8]								---- 0000	---- 0000
0xfcb	PR2L	PR2L[7:0]								1111 1111	1111 1111
0xfca	PR2H	PR2H[11:8]								---- 1111	---- 1111
0xfad	IPR1						TMR2IP			1111 1111	1111 1111
0xfac	PIR1						TMR2IF			0000 0000	0000 0000
0xfab	PIE1						TMR2IE			0000 0000	0000 0000



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；

T3CON(timer3 控制寄存器)

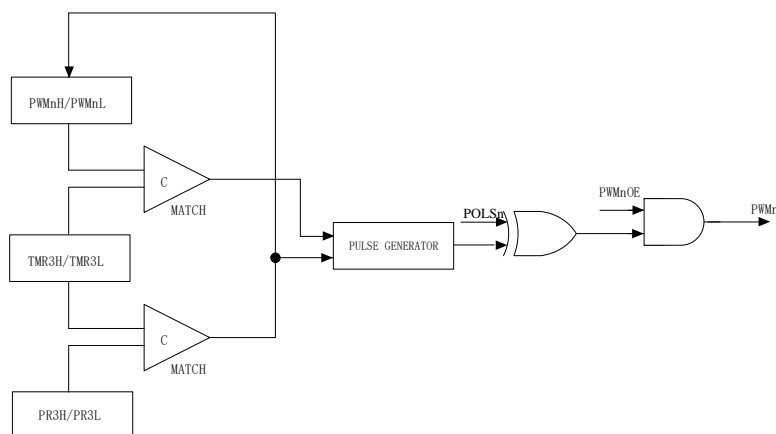
地址：0XFBC

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	R/W	0

		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		
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。

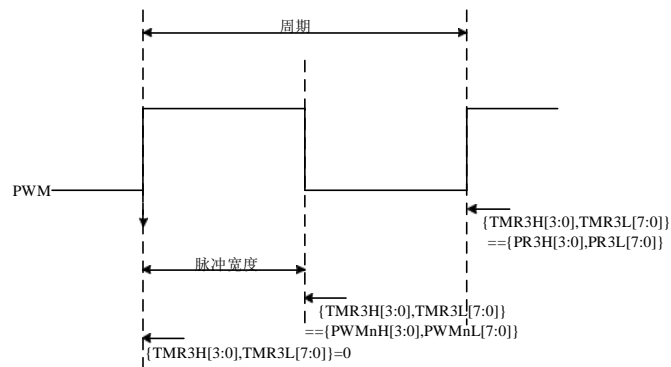


PWM 原理框图 (n=2/3/4/5)



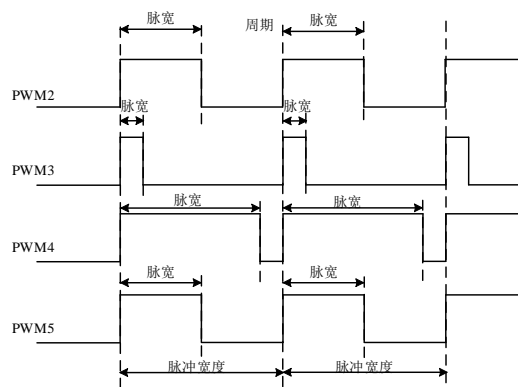
### 7.4.3.4 路 PWM 的输出

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 输出框图。



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<sub>sys</sub> (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

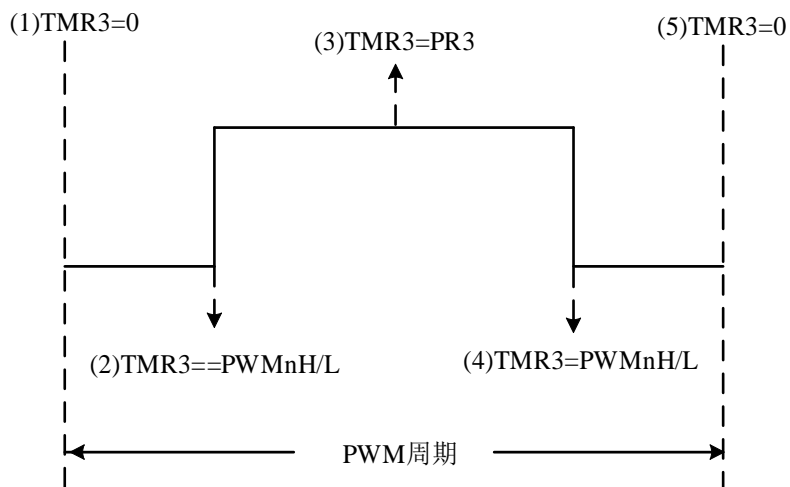
PWM 频率来自系统工作频率。系统工作频率的任何改变将使 PWM 频率的改变。

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。

### 7.4.8. PWM 中心对齐模式



PWM 中心对齐示意图

当 PWM3CAEN (T3CON<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, TMR3开始从0递增;
- (2) 当TMR3与PWMnH/L相等时, PWM开始第一次高低电平变化, TMR3继续递增;
- (3) 当TMR3与PR3相等时, TMR3开始自减;
- (4) 当TMR3再次与PWMnH/L相等时, PWM再次变化高低电平;
- (5) 当TMR3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  $2 \cdot \{ (PR3H, PR3L) + 1 \} / F_{sys} (\text{系统工作时钟}) \cdot (\text{TMR3 的预分频值})$ 。

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。

#### PWM2CON0(PWM 的控制寄存器)

地址: 0XF0F

Bit	Name	Description	Attribute	Reset
7	PWM5OEN	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	R/W	0

		1: 使能 PWM3 映射的 IO 输出 0: 禁止 PWM3 映射的 IO 输出		
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: 0XF0AE

Bit	Name	Description	Attribute	Reset
7	CMP3FLT	CMP3OUT 为故障输入, 当从 1 变为 0 后, 禁止 CCP 中 PWM 输出 1: 使能 CMP3OUT 作为故障输入 0: 禁止 CMP3OUT 作为故障输入	R/W	0
6	CMP2FLT	CMP2OUT 为故障输入, 当从 1 变为 0 后, 禁止 CCP 中 PWM 输出 1: 使能 CMP2OUT 作为故障输入 0: 禁止 CMP2OUT 作为故障输入	R/W	0

5	CMP1FLT	CMP1OUT 为故障输入，当从 1 变为 0 后，禁止 CCP 中 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
4	CMP0FLT	CMP0OUT 为故障输入，当从 1 变为 0 后，禁止 CCP 中 PWM 输出 1: 使能 CMP0OUT 作为故障输入 0: 禁止 CMP0OUT 作为故障输入	R/W	0
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

### 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

注: 1、当 PWM 运行时, PWMH/PWML 寄存器在写寄存器时, 必须先 PWMH 再写 PWML

## 7.4.9. TMR3 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value	
0xfb7	T3CON	PWM3CAEN	T3OUTPS3	T3OUTPS2	T3OUTPS1	T3OUTPS0	TMR3ON	T3CKPS1	T3CKPS0	0000 0000	0000 0000	
0xfb6	TMR3H						TMR3H[11:8]			---- 0000	---- 0000	
0xfb5	TMR3L	TMR3L[7:0]									0000 0000	0000 0000
0xfb4	PR3H						PR3H[11:8]			---- 1111	---- 1111	
0xfb3	PR3L	PR3L[7:0]									1111 1111	1111 1111
0xfb2	PWM5H						PWM5[11:8]			---- 0000	---- 0000	
0xfb1	PWM5L	PWM5[7:0]									0000 0000	0000 0000
0xfb0	PWM4H						PWM4[11:8]			---- 0000	---- 0000	
0xfb7	PWM4L	PWM4[7:0]									0000 0000	0000 0000
0xfb6	PWM3H						PWM3[11:8]			---- 0000	---- 0000	
0xfb5	PWM3L	PWM3[7:0]									0000 0000	0000 0000
0xfb4	PWM2H						PWM2[11:8]			---- 0000	---- 0000	
0xfb3	PWM2L	PWM2[7:0]									0000 0000	0000 0000
0xfaf	PWM2CON0	PWM5OEN	PWM4OEN	PWM3OEN	PWM2OEN	PWM5EN	PWM4EN	PWM3EN	PWM2EN	0000 0000	0000 0000	
0xfae	PWM2CON1	CMP3FLT	CMP2FLT	CMP1FLT	CMP0FLT	POLS5	POLS4	POLS3	POLS2	0000 0000	0000 0000	
0xfad	IPR1					TMR3IP				1111 1111	1111 1111	
0xfac	PIR1					TMR3IF				0000 0000	0000 0000	
0xfab	PIE1					TMR3IE				0000 0000	0000 0000	

## 7.5 看门狗定时器

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时间为128/256/384/640ms/(1.152/2.176/4.224/8.32s), 这个时间可以通过配置字TWDT位 (配置选项<2: 0>) 设置。

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

芯片处于调试模式中, WDT 被禁止使用

**WDTCON (看门狗的控制寄存器)**

地址: 0XFA1

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	000
0	WDTEN	WDT 使能 1: 使能 WDT 0: 禁止 WDT	R/W	

**WDT 相关寄存器定义**

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xfa1	WDTCON					PREDIV[2:0]			SWDTEN	---- 0000	---- 0000



## 8 捕获/比较/PWM 模块

注：CCP 在工作之前需要开启模块时钟，设置 CLKCFG1[4]即 CCPCLKEN=1；

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 CCP 的操作。捕捉和比较均是和 TMR1 或 TMR2 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 或 TMR2H 和 TMR2L 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	T2CAPEN	捕捉模块选择控制位 1: 捕捉 TMR2 计数值; 0: 捕捉 TMR1 计数值	R/W	0
6:4	CAP_PHASE[2:0]	存储捕捉三路霍尔通道相位值	R/W	000
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	0000

### CCPCON0 寄存器

地址：0XFBF

Bit	Name	Description	Attribute	Reset
7:4	CCP0CH[3:0]	CCP 捕获和霍尔捕捉通道 0 触发信号选择 0000: 外部 CCP0CH0 引脚输入信号(PB0) 0001: 外部 CCP0CH1 引脚输入信号(PB1) 0010: 外部 CCP0CH2 引脚输入信号(PB2) 0011: 外部 CCP0CH3 引脚输入信号(PB3) 0100: 外部 CCP0CH4 引脚输入信号(PB4) 0101: 外部 CCP0CH5 引脚输入信号(PB5) 0110: 外部 CCP0CH6 引脚输入信号(PB6) 0111: 外部 CCP0CH7 引脚输入信号(PB7) 1000: 外部 CCP0CH8 引脚输入信号(PC0) 1001: 外部 CCP0CH9 引脚输入信号(PC1) 1010: 外部 CCP0CH10 引脚输入信号(PC2) 1011: 外部 CCP0CH11 引脚输入信号(PC3) 1100: CMP0OUT 的输出 1101: 内部 OSC32K 1110: 外部晶振 LXTL 1111 : 当 T1M[1:0] 等于 0x3 时, 触发信号为 TMR1BUZZ(PD6); 当 T1M[1:0]不等于 0x3 时, 触发信号为 OP0OUT	R/W	0000
3:0	CCPR1M[3:0]	CCP 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 PD5, 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 PD5, TIMER1 溢出时, PD5 为高电平	R/W	0000

		<p>0110: 比较模式, 选择 CCPR1 匹配时将产生软件中断 (CCPIF 位置 1, 而 PD5 引脚不受影响); 并启动 ADC 采集 (如果 ADCON=1)</p> <p>0111: 比较模式, 选择 CCPR1 触发特殊事件 (CCPIF 位置 1, PD5 引脚不受影响); CCP1 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1)</p> <p>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</p> <p>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</p> <p>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</p> <p>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</p> <p>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</p> <p>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</p> <p>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</p> <p>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</p> <p>0011: 霍尔捕捉模式, 启动 3 路捕获, 捕获值存储 CCPR1, 并将 TMR1 清 0, 产生中断标志 CCPIF, 并存储相位值</p> <p>0010: PWM 模式, 启动 CCP PWM 模式</p>		
--	--	---	--	--

## CCPCON1 寄存器

地址: 0XFC0

Bit	Name	Description	Attribute	Reset
7:4	CCP2CH[3:0]	霍尔捕捉通道 2 触发信号选择  0000: 外部 CCP2CH0 引脚输入信号(PB0) 0001: 外部 CCP2CH1 引脚输入信号(PB1) 0010: 外部 CCP2CH2 引脚输入信号(PB2) 0011: 外部 CCP2CH3 引脚输入信号(PB3) 0100: 外部 CCP2CH4 引脚输入信号(PB4) 0101: 外部 CCP2CH5 引脚输入信号(PB5) 0110: 外部 CCP2CH6 引脚输入信号(PB6) 0111: 外部 CCP2CH7 引脚输入信号(PB7) 1000: 外部 CCP2CH8 引脚输入信号(PC0) 1001: 外部 CCP2CH9 引脚输入信号(PC1) 1010: 外部 CCP2CH10 引脚输入信号(PC2) 1011: 外部 CCP2CH11 引脚输入信号(PC3) 1100: CMP2OUT 的输出 1101: 内部 OSC32K 1110: 外部晶振 LXTL 1111: 当 T1M[1:0] 等于 0x3 时, 触发信号为 TMR1BUZZ(PD6); 当 T1M[1:0]不等于 0x3 时, 触发信号为 OP0OUT	R/W	0000
3:0	CCP1CH[3:0]	霍尔捕捉通道 1 触发信号选择  0000: 外部 CCP1CH0 引脚输入信号(PB0) 0001: 外部 CCP1CH1 引脚输入信号(PB1) 0010: 外部 CCP1CH2 引脚输入信号(PB2) 0011: 外部 CCP1CH3 引脚输入信号(PB3) 0100: 外部 CCP1CH4 引脚输入信号(PB4) 0101: 外部 CCP1CH5 引脚输入信号(PB5)	R/W	0000

	0110: 外部 CCP1CH6 引脚输入信号(PB6) 0111: 外部 CCP1CH7 引脚输入信号(PB7) 1000: 外部 CCP1CH8 引脚输入信号(PC0) 1001: 外部 CCP1CH9 引脚输入信号(PC1) 1010: 外部 CCP1CH10 引脚输入信号(PC2) 1011: 外部 CCP1CH11 引脚输入信号(PC3) 1100: CMP1OUT 的输出 1101: 内部 OSC32K 1110: 外部晶振 LXTL 1111: 当 TIM[1:0] 等于 0x3 时, 触发信号为 TMR1BUZZ(PD6); 当 TIM[1:0] 不等于 0x3 时, 触发信号为 OP0OUT		
--	---	--	--

注: 1、配置为霍尔捕捉模式时, 先配置通道, 再使能霍尔捕捉模式;

2、配置为霍尔捕捉模式时, 切换通道时, 先关闭之前捕捉通道, 再使能需要捕捉的通道。

## 8.1 捕捉模式

在捕捉模式下, 当引脚 CCP 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 CCPCON0[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

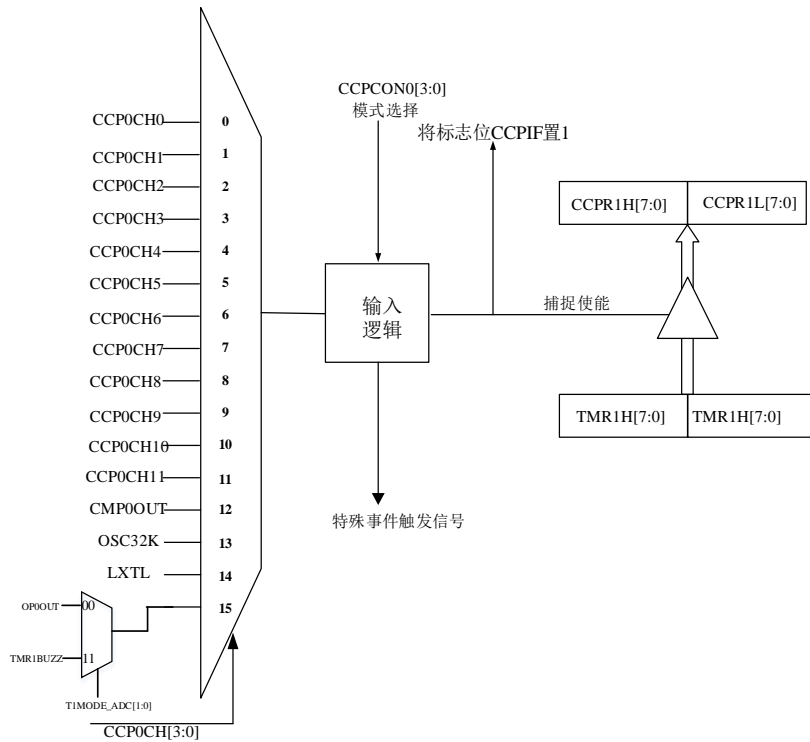
## 志 CCPIF

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

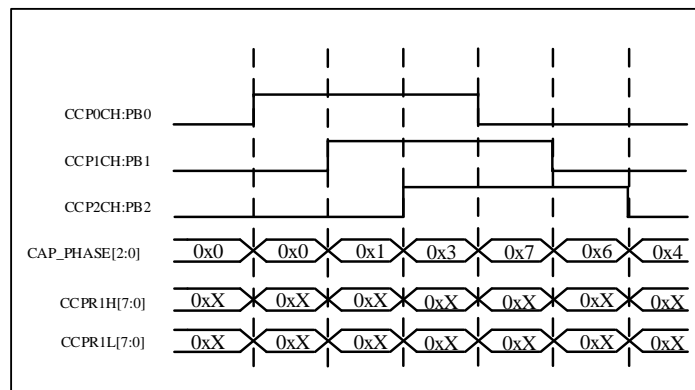
进行捕捉后, 中断请求标志位 CCPIF 被置 1。该中断标志位必须用软件清零。如果在 CCPR1H 和 CCPR1L 寄存器对中的值被读出之前又发生另一次捕捉, 那么原来的捕捉值会被新捕捉值覆盖。输入捕获通道有 CCP0CH[3:0]进行配置:

- ◇ 0000: 外部 CCP0CH0 引脚输入信号(PB0)
- ◇ 0001: 外部 CCP0CH1 引脚输入信号(PB1)
- ◇ 0010: 外部 CCP0CH2 引脚输入信号(PB2)
- ◇ 0011: 外部 CCP0CH3 引脚输入信号(PB3)
- ◇ 0100: 外部 CCP0CH4 引脚输入信号(PB4)
- ◇ 0101: 外部 CCP0CH5 引脚输入信号(PB5)
- ◇ 0110: 外部 CCP0CH6 引脚输入信号(PB6)
- ◇ 0111: 外部 CCP0CH7 引脚输入信号(PB7)
- ◇ 1000: 外部 CCP0CH8 引脚输入信号(PC0)
- ◇ 1001: 外部 CCP0CH9 引脚输入信号(PC1)
- ◇ 1010: 外部 CCP0CH10 引脚输入信号(PC2)
- ◇ 1011: 外部 CCP0CH11 引脚输入信号(PC3)
- ◇ 1100: CMP0OUT 的输出
- ◇ 1101: 内部 OSC32K
- ◇ 1110: 外部晶振 LXTL
- ◇ 1111: 当 T1M[1:0]等于 0x3 时, 触发信号为 TMR1BUZZ(PD6); 当 T1M[1:0]不等于 0x3 时, 触发信号为 OP0OUT



CCP 捕捉模式工作原理图

注：当 T2CAPEN (CCPR1LH<7>) 等于 1 时，CCPR1H:CCPR1L 将捕捉 TMR2 的计数值。



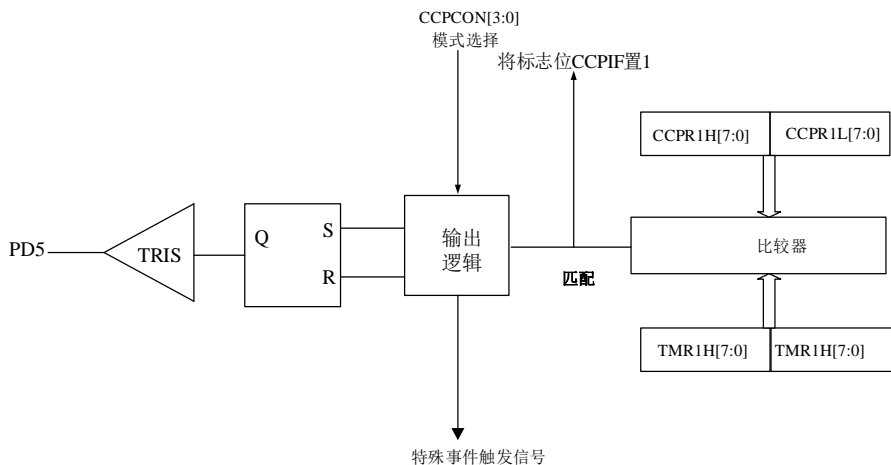
霍尔捕捉框图 (注: X 表示捕捉到 TMR1H/TMR1L 的计数值)

## 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 CCPCON0[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 PD5；TIMER1 溢出时，PD5 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 PD5; TIMER1 溢出时, PD5 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 PD5 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, PD5 引脚不受影响); CCP1 清零 TIMER1; 并启动 ADC 采集 (如果 ADON=1)

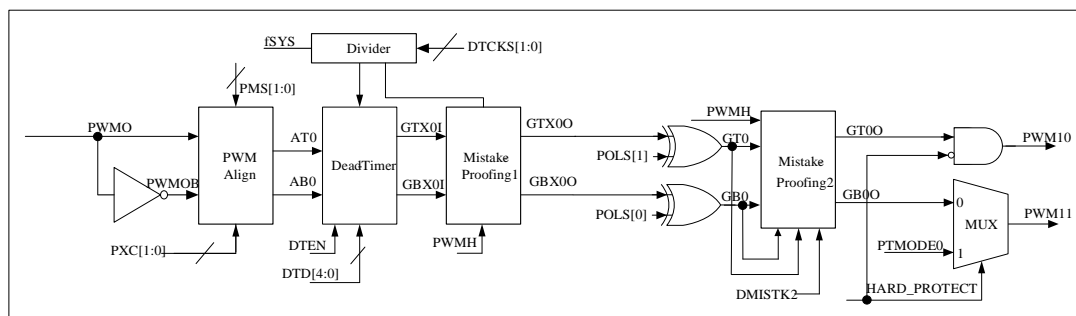


比较模式工作原理图

### 8.3 HBRIDGE 单元

HBRIDGE 是由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。

共有三组 HBRIDGE, 输出分别是 PWM00 和 PWM01、PWM10 和 PWM11、PWM20 和 PWM21, 其中 PWM10 和 PWM11 的原理图如下图, PWM00 和 PWM01、PWM20 和 PWM21 原理相同。



HBRIDGE 驱动结构图

注：上臂对应 PWM00、PWM10、PWM20

下臂对应 PWM01、PWM11、PWM21



**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]	PWM20、PWM21 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: PWM20、PWM21 控制 (由 PXC2 控制上/下臂输出)	R/W	00
3:2	PMS1[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 PXC1 控制上/下臂输出。)	R/W	00
1:0	PMS0[1:0]	PWM00、PWM01 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	R/W	00

		11: PWM00、PWM01 控制（由 PXC0 控制上/下臂输出。）		
--	--	--------------------------------------	--	--

### PXC(PWM 控制输出寄存器)

地址：0XFC4

Bit	Name	Description	Attribute	Reset
7	T3PWMEN	使能 TIMER3 PWM2,PWM3,PWM4 控制 CCP 的 HBRIDGE 单元 1: 允许使能 0: 禁止使能	R/W	0
6	T3PWMMODE	TIMER3 中 PWM 映射到 HBRIDGE 选择为 0: 选择 PWM2、PWM3、PWM4 映射到 3 个 HBRIDGE 单元; 1、选择 PWM2 同时映射到 3 个 HBRIDGE 单元		
5:4	PXC2[1:0]	PMS2[1:0]==2'b11 时, PWM20、PWM21 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, 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
1:0	PXC0[1:0]	PMS0[1:0]==2'b11 时, PWM00、PWM01 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	R/W	00

		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)		
--	--	--	--	--

### 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

### POLS (极性选择寄存器)

地址: 0XFC2

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	PTMODE2	保护 PWM21 的输出 1: PWM21 输出 1 0: PWM21 输出 0	R/W	0
5	POLS21	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	POLS20	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	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
2	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
1	POLS01	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	POLS00	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

### PME 寄存器

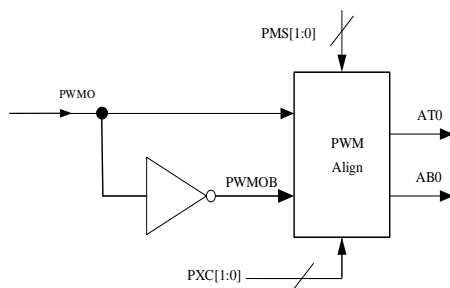
地址: 0XFC1

Bit	Name	Description	Attribute	Reset
7	PT1MODE	保护 PWM11 的输出 1: PWM11 输出 1 0: PWM11 输出 0	R/W	0
6	PT0MODE	保护 PWM01 的输出 1: PWM01 输出 1 0: PWM01 输出 0	R/W	0
5	PWM21EN	PWM21 映射的 IO 输出使能 1: 使能输出 0: 禁止输出	R/W	0
4	PWM20EN	PWM20 映射的 IO 输出使能 1: 使能输出 0: 禁止输出	R/W	0
3	PWM11EN	PWM11 映射的 IO 输出使能	R/W	0

		1: 使能输出 0: 禁止输出		
2	PWM10EN	PWM10 映射的 IO 输出使能 0: 禁止输出 1: 使能输出	R/W	0
1	PWM01EN	PWM01 映射的 IO 输出使能 1: 使能输出 0: 禁止输出	R/W	0
0	PWM00EN	PWM00 映射的 IO 输出使能 1: 使能输出 0: 禁止输出	R/W	0

### 8.3.1. 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

PWMO	PWMOB
0	1
1	0

校准电路输出表 1

PMS[1:0]	PWM 调制模式	AT0	AB0
00	互补式控制	PWMO	PWMOB
01	非互补式上臂调制	PWMO	0

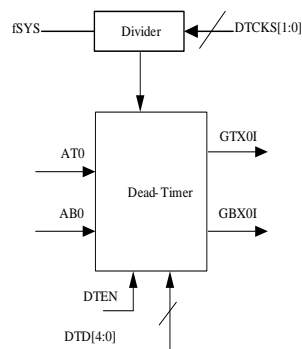
10	非互补式下臂调制	0	PWMO
11	由 PXC 寄存器控制上下臂输出	PXC[1]	PXC[0]

校准电路输出表 2

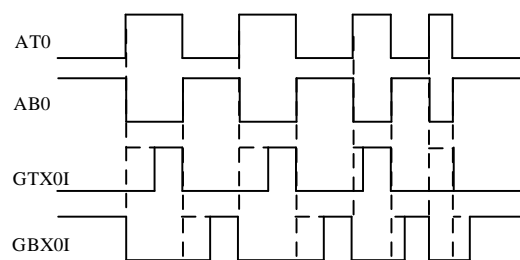
### 8.3.2. 死区时间

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



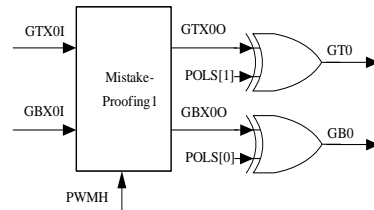
死区时间时序图 (n=0)

DTEN	GTX0I	GBX0I
1	AT0&DTD	AB0&DTD
0	AT0	AB0

死区时间电路输出表 1

### 8.3.3. 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	GTX0O	GBX0O
0	0	PWMH	0
0	1	0	1
1	0	1	0
1	1	PWMH	0

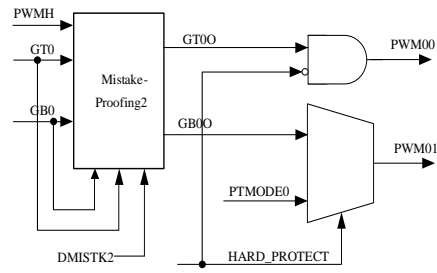
一级防呆电路输出表 1

GTX0O	POLS[1]	GT0
0	0	0
0	1	1
1	0	1
1	1	0

一级防呆电路输出表 2

GBX0O	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

GT0	GB0	DMISK2	GT0O	GB0O
0	0	0	PWMH	0
0	1	0	0	1
1	0	0	1	0
1	1	0	PWMH	0

GT0	GB0	DMISK2	GT0O	GB0O
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

GT0O	GB0O	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

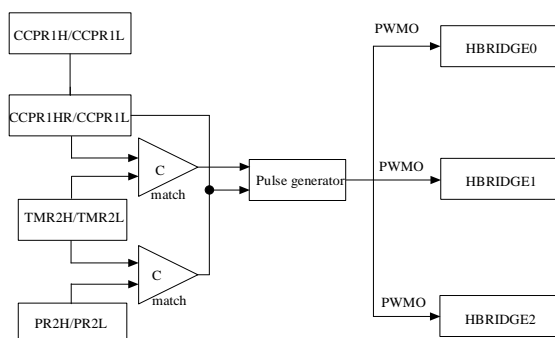
PR2H=0xf;
PR2L=0xff;//周期配置
TMR2H=0x00;
TMR2L=0x00;//计数值初始值

CCPR1H=0x8;
CCPR1L=0x00;//占空比配置

PXC=0x00;//上下臂及PWM IO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使能及保护选择
CCPCON=0x02;
T2CON=0x04;
    
```

## 8.4 CCP PWM 控制三路 HBRIDGE

通过 T3PWMEN(PXC[7])选择可以将 Timer2 或 Timer3 的 PWM 映射到 CPP 的 HBRIDGE 单元。当 T3PWMEN (PXC[7]) 为 0 时，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。



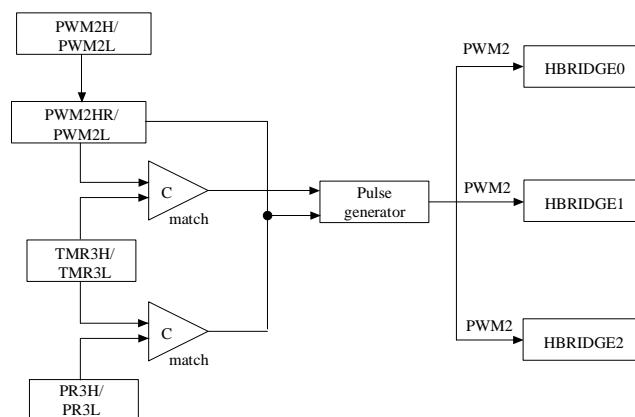
CCP PWM 驱动 3 路 HBRIDGE

## 8.5 TIMER3 PWM 控制三路 HBRIDGE

当 T3PWMEN (PXC[7]) 为 1 时, TMR3 PWM 输出到 3 个 HBRIDGE 单元, 可控制 3 组互补 PWM 输出。通过 T3PWMMODE (PXC[6]) 决定 TMR3 PWM 如何输出到 3 个 HBRIDGE 单元。

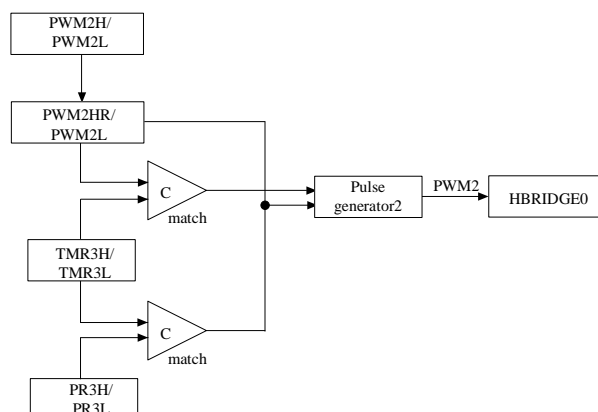
T3PWMMODE (PXC[6]) 设置为 1, Timer3 的 PWM2 同时输出到 3 个 HBRIDGE 单元, 可控制 3 组互补 PWM 输出。此时当 Timer3 的 PWMEN 寄存器使能时, PWM2 输出分别控制 PWM00 和 PWM01、PWM10 和 PWM11 和 PWM20 和 PWM21。如下图 1 所示。

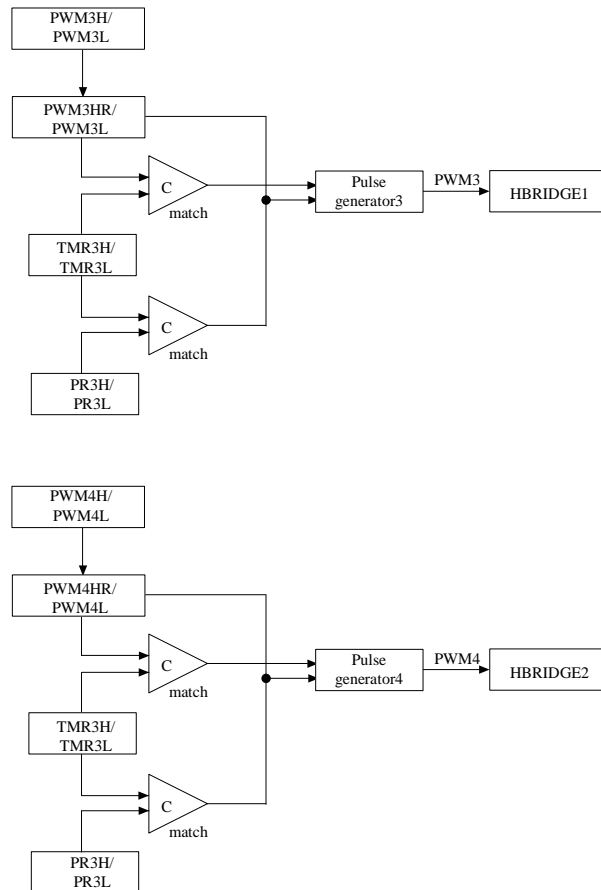
T3PWMMODE (PXC[6]) 设置为 0, Timer3 的 PWM2, PWM3, PWM4 的三路 PWM 信号, 分别输出到 3 个 HBRIDGE 单元, 可以构成 3 组独立占空比可调的互补 PWM 输出, 用于复杂 PWM 互补输出控制。此时当 Timer3 的 PWMEN 使能时, PWM2 输出控制 PWM00 和 PWM01, PWM3 输出控制 PWM10 和 PWM11, PWM4 输出控制 PWM20 和 PWM21。如下图 2 所示。



TMR3 PWM2 驱动 3 路 HBRIDGE (PXC[6] 置 1)

下图是 PXC[6] 置 0, TMR3 PWM2/3/4 输出到 HBRIDGE0/1/2。





TMR3 PWM2/3/4 分别驱动 HBRIDGE2/3/4 (PXC[6]置 0)

## 8.6 PWM 相关寄存器介绍

PWM1CON1(PWM 的死区时间控制寄存器)

地址: 0XFBD

Bit	Name	Description	Attribute	Reset
7:0	PWMADDLY[7:0]	ADC 采集延时时间: $T = \{PWMADDLY[7:0]\} * F_{sys}$ , $F_{sys}$ 为系统时钟	R/W	0x00

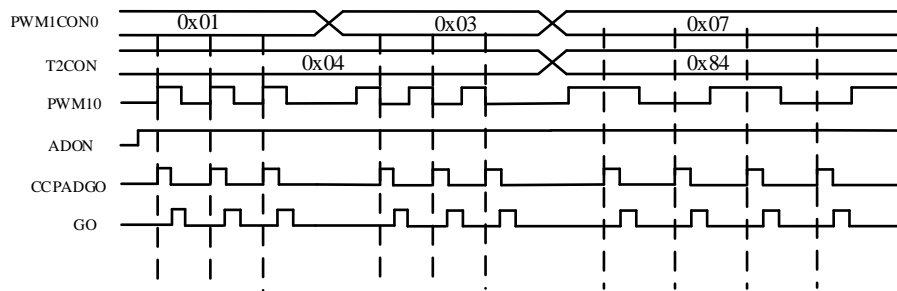
PWM1CON0(PWM 的控制寄存器)

地址: 0XFBE

Bit	Name	Description	Attribute	Reset
7	OPFLT	OP0OUT 为故障输入, 当从 1 变为 0 后, 禁止 PWM 输出 1: 使能 OP0OUT 作为故障输入 0: 禁止 OP0OUT 作为故障输入	R/W	0

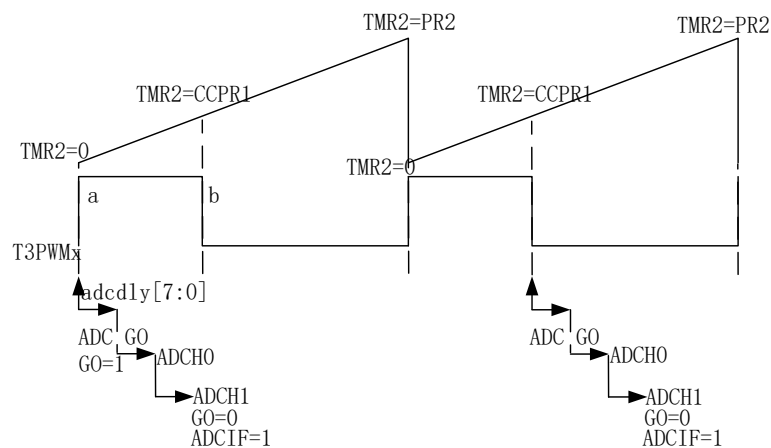
6	EFTFLT	<p>EFT 为故障输入，当从 1 变为 0 后，禁止 PWM 输出</p> <p>1: 使能 EFT 作为故障输入</p> <p>0: 禁止 EFT 作为故障输入</p>	R/W	0
5	ASTART	<p>发生故障后(ACLOSE=1)，当使能的故障都解除后，自动启动 PWM 输出</p> <p>1:使能启动 PWM 输出</p> <p>0:禁止启动 PWM 输出</p>	R/W	0
4	ALOSE	<p>发生故障后，自动关闭 PWM 输出</p> <p>1:使能关闭 PWM 输出</p> <p>0:禁止关闭 PWM 输出</p>	R/W	0
3	IOFLT	<p>IO 为故障输入，当从 1 变为 0 后，禁止 PWM 输出</p> <p>1: 使能 IO 作为故障输入</p> <p>0: 禁止 IO 作为故障输入</p> <p>注：故障 IO 映射为 PC2;</p>	R/W	0
2:1	PWMADPOS	<p>PWM 跳变沿使能 ADC 采集</p> <p>1x: 中心对齐模式：TMR2==PR2 及 TMR2==12'H0 时，TMR3==PR3 及 TMR3==12'H0 时，启动 PWMADDLY 计数器，当计数器为 0 时启动 ADC 采样</p> <p>01：PWM10 沿跳变时，TMR2==CCPR1 或者 TMR3=={PWM2H,PWM2L}时，启动 PWMADDLY 计数器，当计数器为 0 时启动 ADC 采样</p> <p>00：PWM10 沿跳变时，TMR2==PR2 或者 TMR3==PR3 时，启动 PWMADDLY 计数器，当计数器为 0 时启动 ADC 采样。</p> <p>沿跳变时 ADC 才会采集，如果沿不跳变 ADC 不会采集</p>	R/W	0
0	PWMADEN	<p>使能 PWM 输出跳变时，自动启动 ADC 采集功能</p> <p>1: 使能 ADC 采集</p> <p>0: 禁止 ADC 采集</p>	R/W	0

下图为 PWM10 跳变时，ADC 采集时序图。



ADC 采集时序图

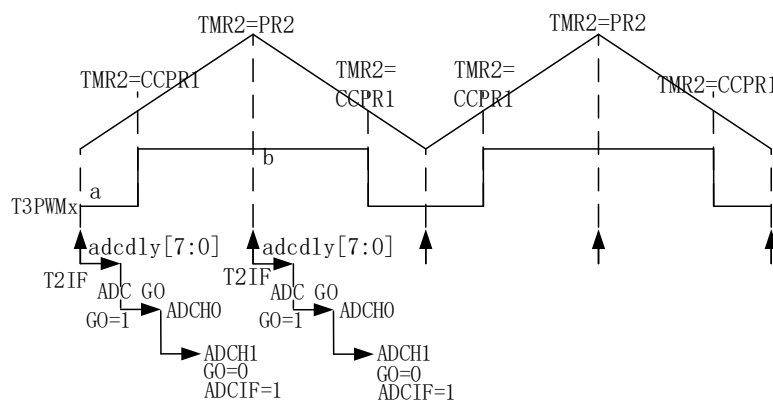
下图为 PWM 普通模式 ADC 采集图。



ADC 采集简图

注：a 点和 b 点 PWM1CON0 寄存器配置参考 ADC 采集时序图中配置。

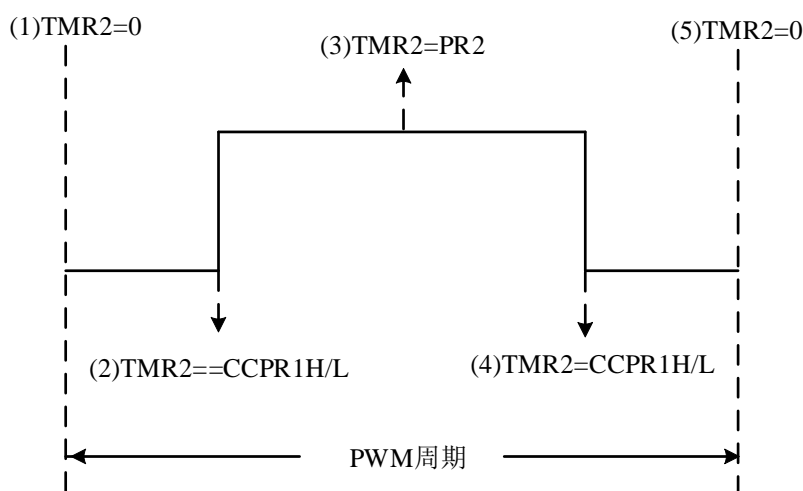
下图为 PWM 中心对齐模式 ADC 采集图。



## ADC 采集简图

注：当 ADC 采集 PWM 的 a 点时，ADC 采集延时时间（PWM1CON1 配置时间）不能超过 PWM 占空比时间；当 ADC 采集 PWM 的 b 点时，PWM 普通模式中 ADC 采集延时时间（PWM1CON1 配置时间）不能超过 PWM 下个周期开始时间，PWM 中心对齐模式中 ADC 采集延时时间（PWM1CON1 配置时间）不能超过 TMR2 递减到和 CCPR1 相等的时间。

## 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：

- (6) PWM周期开始，TMR2开始从0递增；
- (7) 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；
- (8) 当TMR2与PR2相等时，TMR2开始自减；
- (9) 当TMR2再次与CCPR1H/L相等时，PWM再次变化高低电平；
- (10) 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  $2 \cdot \{ (PR2H, PR2L) + 1 \} / F_{sys}(\text{系统工作时钟}) \cdot (\text{TMR2 的预分频值})$ 。

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。

## 8.7 PWM 使用说明

### 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

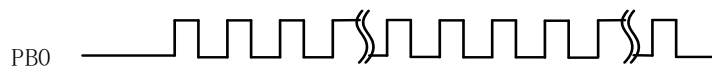
PWM00/PWM01/PWM10/PWM11/PWM20/PWM21 依次映射全部 IO。PWM 引脚的极性可配置，可通过配置 POLS 寄存器选择极性。每两个 I/O 为一组半桥。全桥为二组半桥。

#### 注意：

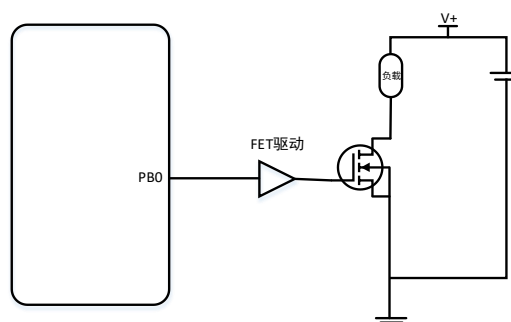
- 必须正确配置每个 PWM 输出的 TRIS 寄存器值；
- 清零 CCPCON 寄存器将放弃所有 PWM 输出引脚；
- 当 PWMnOE 不使能时，增强型 PWM 模式所不使用的任何引脚均可用于其他引脚功能。  
(n=00/01/10/11/20/21)

### 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。下图以 PB0 为例。



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

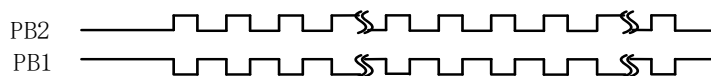
PR2H=0xf;
PR2L=0xff;//周期配置
TMR2H=0x00;
TMR2H=0x00;//计数值初始值

CCPR1H=0x8;
CCPR1L=0x00;//占空比配置

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

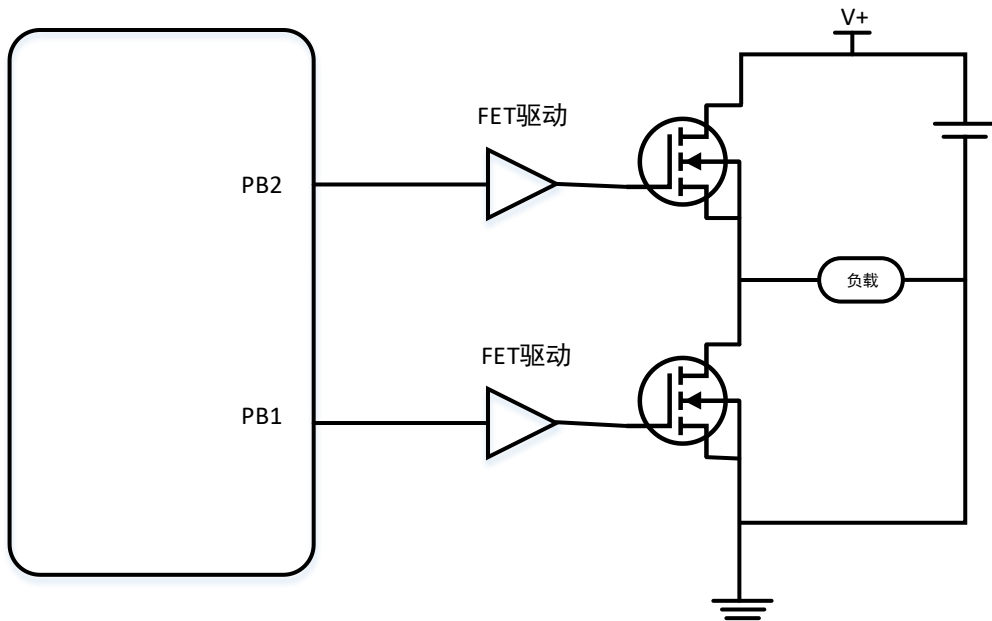
## 半桥模式

在半桥模式下，有两个引脚用作输出以驱动负载。配置如下 IO 作为 PWM 映射 IO，PB2 和 PB1 为一组半桥。PWM 输出信号被输出到 PB2 引脚，而互补 PWM 输出信号被输出到 PB1。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。

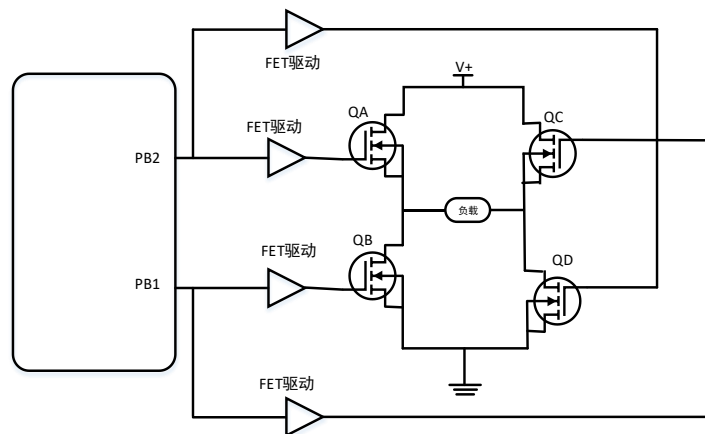


半桥模式输出图

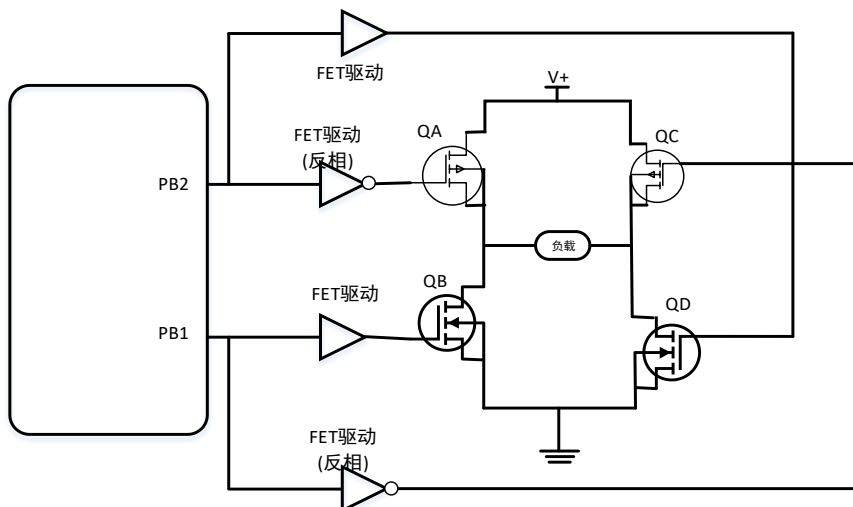




标准半桥电路图



半桥输出驱动全桥电路 (4NMOS)



半桥输出驱动全桥电路 (2PMOS+2NMOS)

## 标准半桥例子

```

CCPIE=1;
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

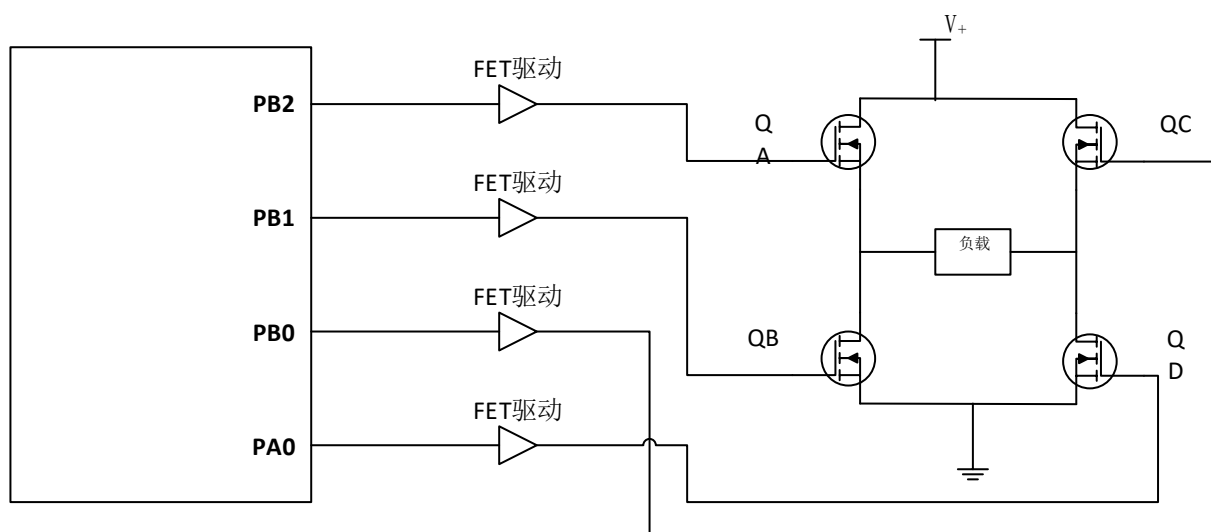
PMS=0x00; //互补输出
PXC=0x00; //上下臂选择
POLS=0x00; //极性选择
PME=0x3f; //PWM映射IO使能及保护选择
PCS=0x00; //PWM映射IO
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能TMR2
    
```

## 全桥模式

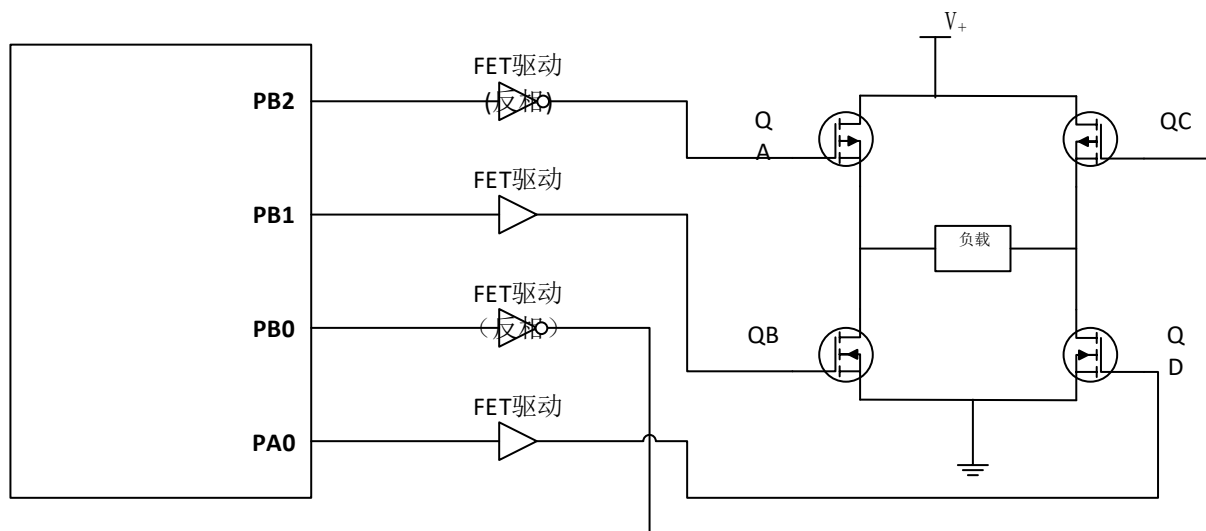
在全桥模式下，所有四个引脚用作输出。配置如下 IO 作为 PWM 映射 IO，这里以第一组半桥 PB2 和 PB1，第二组半桥 PB0 和 PA0 为例。三组半桥可以自由两两组合为全桥。

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。

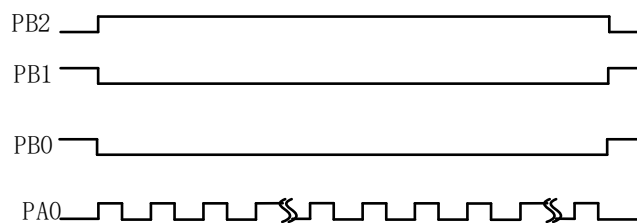


全桥应用示例 a



全桥应用示例 b

使用4个NMOS时，在全桥正向模式下，PB2引脚驱动为有效状态，PA0引脚为调制输出，而PB1和PB0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=1;
CCPIF=0;
CCPIP=1;
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

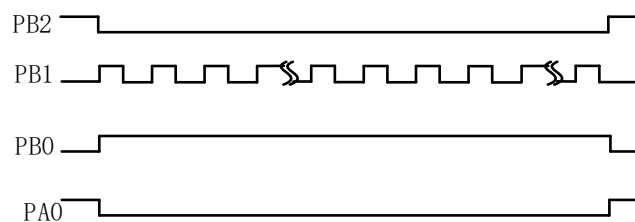
CCPR1H=0x8;
CCPR1L=0x00;//占空比配置

PMS=0x0b;//第一组半桥PXC控制上下臂输出，第二组半桥非互补式上臂调制
PXC=0x22;//第一组半桥上下臂选择
    
```

```

POLLS=0x00; //极性选择
PME=0x3f; //PWM映射IO使能及保护选择
PCS=0x00; //PWM映射IO
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，PB0 驱动为有效状态，PB1 引脚为调制输出，而 PB2 和 PA0 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：

```

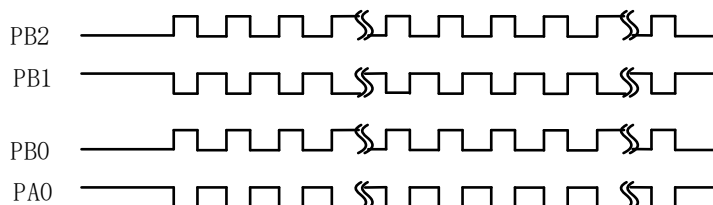
CCPIE=1;
CCPIF=0;
CCPIP=0;
TMR2IE=0; //中断相关配置

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; //第一组半桥非互补式下臂调制，第二组半桥PXC控制上下臂输出
PXC=0x08; //上下臂选择
POLLS=0x00; //极性选择
PME=0x3f; //PWM映射IO使能及保护选择
PCS=0x00; //PWM映射IO
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=1;
CCPIF=0;
CCPIP=0;
TMR2IE=0;//中断相关配置

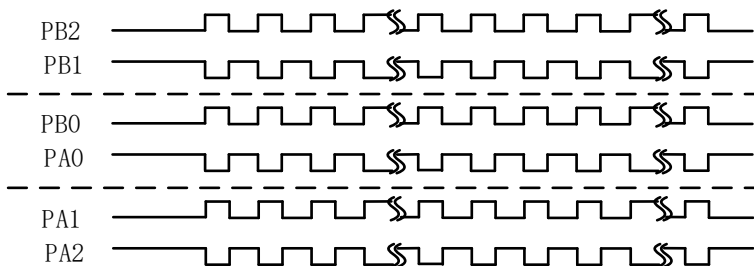
PR2H=0xf;
PR2L=0xff;//周期配置
TMR2H=0x00;
TMR2L=0x00;//计数值初始值

CCPR1H=0x8;
CCPR1L=0x00;//占空比配置

PMS=0x00;//互补式输出
PXC=0x00;
POLS=0x00;//极性选择
PME=0x3f;//PWM映射IO使能及保护选择
PCS=0x00;
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能 TMR2
    
```

### 3 组 PWM 输出

配置如下 IO 作为 PWM 映射 IO, 这里以第一组半桥 PB2 和 PB1, 第二组半桥 PB0 和 PA0, 第三组半桥为 PA1 和 PA2。



3 组 PWM 输出

如上图所示，PB2 和 PB1 是第一组半桥 PWM 输出，PB0 和 PA0、PA1 和 PA2 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

## 8.8 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & uor reset value	user reset value	
0xf08	CCPR1H	Capture Compare/PWM Register 1 High Byte								0000 0000	0000 0000	
0xf07	CCPR1L	Capture Compare/PWM Register 1 Low Byte								0000 0000	0000 0000	
0xf06	CCPR1H	T2CAPEN	CAP_PHASE[2:0]			CCPR1L[11:8]					0000 0000	0000 0000
0xf05	PMS	DMISTK2	PWMH	PMS2[1:0]		PMS1[1:0]		PMS0[1:0]		0000 0000	0000 0000	
0xf04	PXC	T3PWMEN	T3PWMMODE	PXC2[1:0]		PXC1[1:0]		PXC0[1:0]		0000 0000	0000 0000	
0xf03	DTC	DTCKS1	DTCKS0	DTEN	DTD4	DTD3	DTD2	DTD1	DTD0	0000 0000	0000 0000	
0xf02	POLS	PT2MODE		POLS2[1:0]		POLS1[1:0]		POLS0[1:0]		-000 0000	-000 0000	
0xf01	PME	PT1MODE	PT0MODE	PWM2EN[1:0]		PWM1EN[1:0]		PWM0EN[1:0]		0000 0000	0000 0000	
0xf00	CCPCON1	CCP2CH[3:0]			CCP1CH[3:0]			CCP1M[3:0]			0000 0000	0000 0000
0xf0f	CCPCON0	CCP0CH[3:0]			CCP1M[3:0]			CCP1M[3:0]			0000 0000	0000 0000
0xf0e	PWM1CON0	OPFLT	ASTART		ACLOSE	IOFLT	PWMADPOS[1:0]		PWMADEN	0000 0000	0000 0000	
0xf0d	PWM1CON1	PWMADDLY[7:0]								0000 0000	0000 0000	

## 9 通用同步/异步收发器 (USART0 & UART1)

注：UART 在工作之前需要开启模块时钟，设置 CLKCFG1[5]即 UARTCLKEN=1；

UART1 在工作之前需要开启模块时钟，设置 CLKCFG2[6]即 UART1CLKEN=1；

通用同步/异步收发器 (Universa 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）。AD18F06 的 USART 支持全双工异步通信。

### 9.1 UART0

#### SPBRG

地址：0XF2E

Bit	Name	Description	Attribute	Reset
7:0	SPBRG	波特率发生器	R/W	0x00

#### RCREG

地址：0XF2D

Bit	Name	Description	Attribute	Reset
7:0	RCREG	接收缓冲寄存器	R	0x00

#### RXREG

地址：0XF2C

Bit	Name	Description	Attribute	Reset
7:0	RXREG	发送缓冲寄存器	W	0x00

#### TXSTA

地址：0XF2B

Bit	Name	Description	Attribute	Reset
7	CSRC	时钟源选择位 异步模式： 忽略。 同步模式：	R/W	0

		1: 主机模式 (时钟来自内部 BRG) 0: 从机模式		
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	TX_PARITY	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	TMRT	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: 0XF2A

Bit	Name	Description	Attribute	Reset
7	SPEN	串口使能位	R/W	0



		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)		
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：0XF29

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，将 RX 引脚同时使能为半双工引脚 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

## 9.2 波特率发生器

BRG 是一个 8 位的发生器，受 BRGH0(TXSTA0<2>)位控制。波特率计算公式如下：

BCLK	BRGM	BRGH	波特率公式
0	1	0	$F_{OSC}/[4*(SPBRG)]$
0	1	1	$F_{OSC}/[2*(SPBRG)]$
0	0	0	$F_{OSC}/[64*(SPBRG)]$
0	0	1	$F_{OSC}/[16*(SPBRG)]$
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：

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

求解 SPBRG：

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

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

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

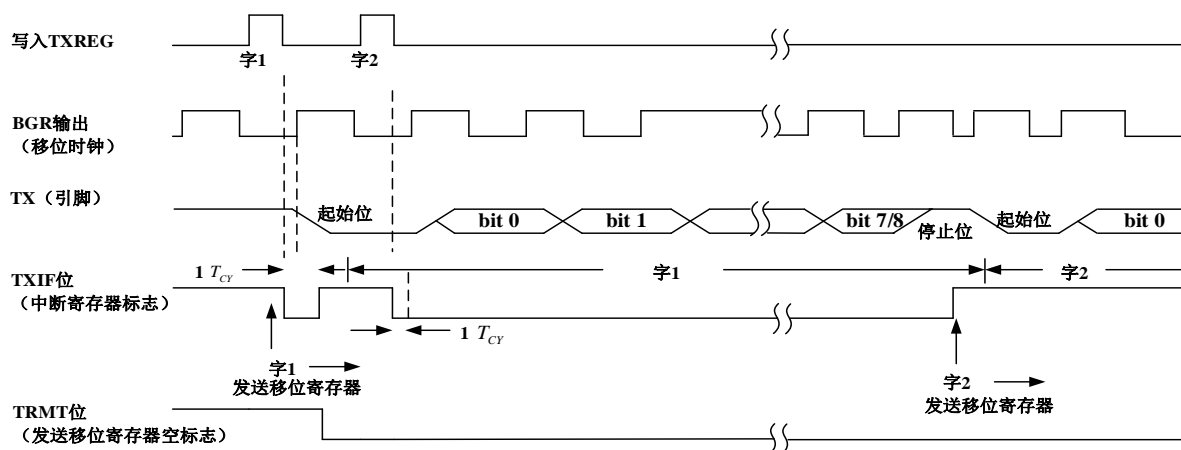
针对工作在同步模式下，工作频率为 BCLK，波特率计算与异步模式相同。

## 9.3 异步发送器

发送器的核心是发送（串行）移位寄存器（Transmit Shift Register, TSR0）。移位寄存器从读/写发送缓冲寄存器 TXREG0 中获取数据。TXREG0 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 TSR0 寄存器装入新数据。一旦停止位发送完毕，TXREG0 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





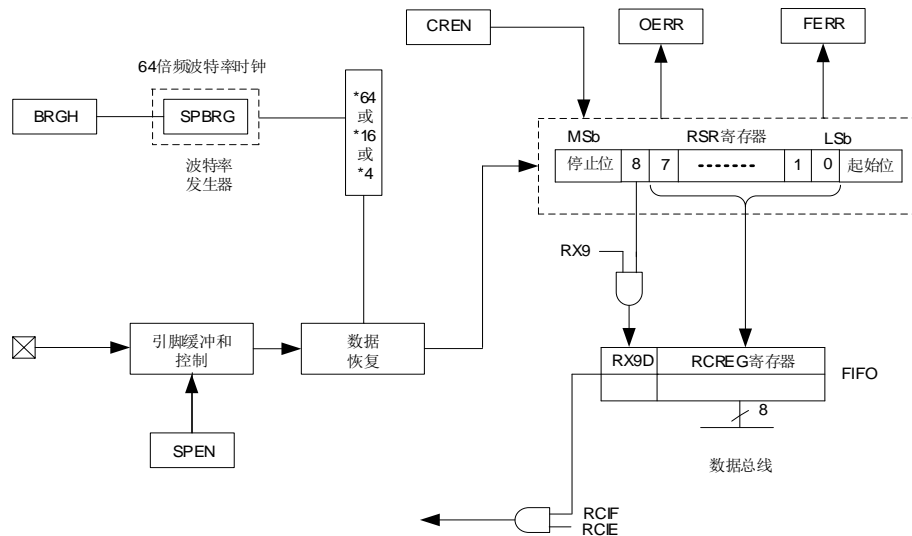
异步发送（背对背）

## 9.4 异步接收

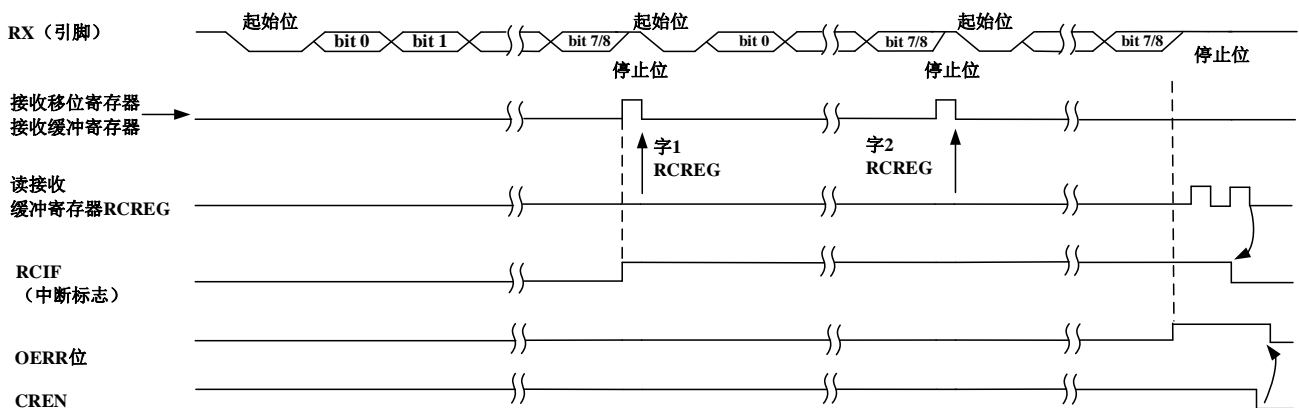
如 UART 接收原理图所示，在 RX 引脚上接收数据，并驱动数据恢复电路。数据恢复电路实际上是一个以 16 倍波特率为工作频率的高速移位器，而主接收串行移位器的工作频率等于比特率或 FOSC。此模式通常用于 RS-232 系统。

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 INTCON 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

## 9.5 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。

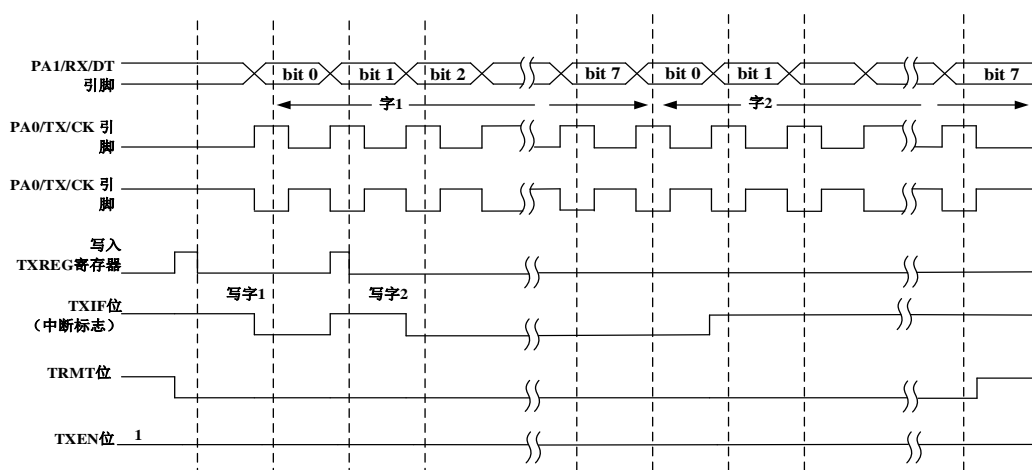
### 9.5.1. 同步主机发送

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：

- 1) 初始化SPBRG 寄存器，设置合适的波特率。
- 2) 通过将SYNC、SPEN 和CSRC 位置1，使能同步主机串口。
- 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TXEN 位置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INTCON寄存器中的GIEH(INTCON<7>)或GIEL(INTCON<6>)置1。

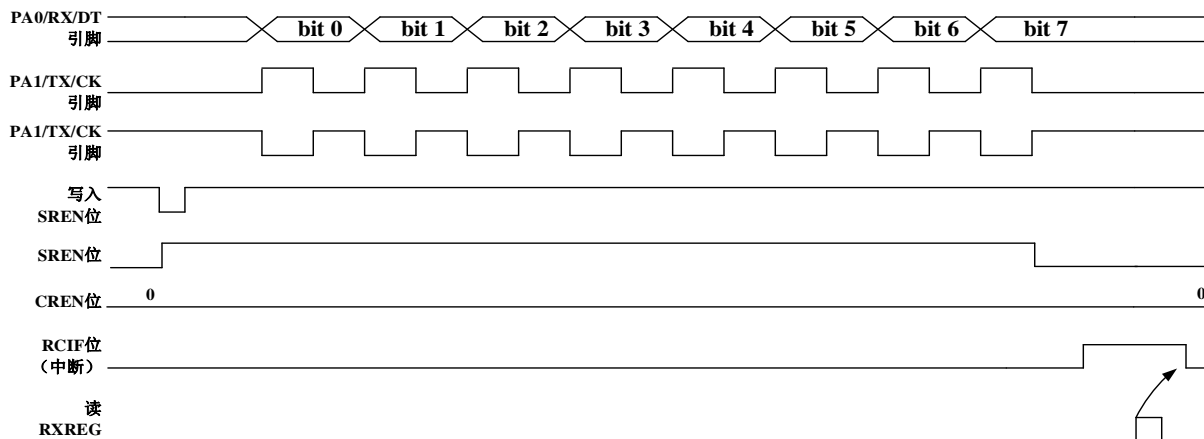


注：同步主控模式，SPBRG=0，连续发送两个8位字节

### 同步发送时序







主机模式同步接收的时序（由SREN位控制）

## 9.6 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。

### 9.6.1. 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：

- 第一个字立即传送到TSR 寄存器进行发送。
- 第二个字仍保留在TXREG 寄存器中。
- 不会将标志位TXIF 置1。
- 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。
- 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 INTCON 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。

### 9.6.2. 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 INTCON 寄存器中的 GIEH (INTCON<7>) 或 GIEL (INTCON<6>) 置 1。

通用同步/异步收发器 (Universa 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）。AD18F06 的 USART 支持全双工异步通信。

## 9.7 UART1

### SPBRG1

地址：0XF28

Bit	Name	Description	Attribute	Reset
7:0	SPBRG1	波特率发生器	R/W	0x00

### RCREG1

地址：0XF27

Bit	Name	Description	Attribute	Reset
7:0	RCREG1	接收缓冲寄存器	R	0x00

### RXREG1

地址：0XF26

Bit	Name	Description	Attribute	Reset
7:0	RXREG1	发送缓冲寄存器	W	0x00

### TXSTA1

地址：0XF25

Bit	Name	Description	Attribute	Reset
7	CSRC1	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	TX19	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	R/W	0

		0: 禁止发送		
4	SYNC1	USART 模式选择位 1: 同步模式 0: 异步模式	R/W	0
3	TX1_PARITY	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	BRGH1	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	TMRT1	发送移位寄存器状态位 1: TSR 空 0: TSR 满	R/W	1
0	TX19D	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

## RCSTA1

地址: 0XF24

Bit	Name	Description	Attribute	Reset
7	SPEN1	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	RX19	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	SREN1	单字节接收使能位	R/W	0

		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。		
4	CREN1	连续接收使能位 1：使能接收器 0：禁止接收器	R/W	0
3	BRGM1	与 BRGH 组成 UART 时钟选择位	R/W	0
2	FERR1	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	OERR1	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	RX19D	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

## UARTCON1

地址：0XF23

Bit	Name	Description	Attribute	Reset
7	INVMODE1	数据电平反向接受发送使能位 异步： 1：使能	R/W	0

		0: 禁止 同步: 忽略		
6	PARITY1	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	ODD1_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	BCLK1	波特率发生器时钟源选择位 1: 外部时钟 (PB5) 0: 内部系统时钟	R/W	0
3	HALF1_DUPLEX	单线半双工使能位 1: 使能, 将 RX 引脚同时使能为半双工引脚 0: 禁止	R/W	0
2	DLSB1	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	STOP1_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

## 9.8 UART0 & UART1 寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value	
0xf2e	SPBRG	Uart Baud Rate Generator Register Byte									0000 0000	0000 0000
0xf2d	RCREG	Uart Receive Register									0000 0000	0000 0000
0xf2c	TXREG	Uart Transmit Register									0000 0000	0000 0000
0xf2b	TXSTA	CSRC	TX9	TXEN	SYNC	TX_PARITY	BRGH	TMRT	TX9D	0000 0000	0000 0000	
0xf2a	RCSTA	SPEN	RX9	SREN	CREN	BGRM	FERR	OERR	RX9D	0000 0000	0000 0000	
0xf29	UARTCON	INVMODE	PARITY	ODD_EVEN	BCLK	HALF_DUPLEX	DLSB	STOP_BIT[1:0]		0000 0000	0000 0000	
0xf28	SPBRG1	Uart Baud Rate Generator Register Byte									0000 0000	0000 0000
0xf27	RCREG1	Uart Receive Register									0000 0000	0000 0000
0xf26	TXREG1	Uart Transmit Register									0000 0000	0000 0000
0xf25	TXSTA1	CSRC1	TX19	TXEN1	SYNC1	TX1_PARITY	BRGH1	TMRT1	TX19D	0000 0000	0000 0000	
0xf24	RCSTA1	SPEN1	RX19	SREN1	CREN1	BGRM1	FERR1	OERR1	RX19D	0000 0000	0000 0000	
0xf23	UARTCON1	INVMODE1	PARITY1	ODD1_EVEN	BCLK1	HALF1_DUPLEX	DLSB1	STOP1_BIT[1:0]		0000 0000	0000 0000	
0xfaa	IPR2					RCIP	TXIP	RC1IP	TX1IP	1111 1111	1111 1111	
0xfa9	PIR2					RC1F	TX1F	RC1IF	TX1IF	0000 0000	0000 0000	
0xfa8	PIE2					RC1E	TX1E	RC1IE	TX1IE	0000 0000	0000 0000	

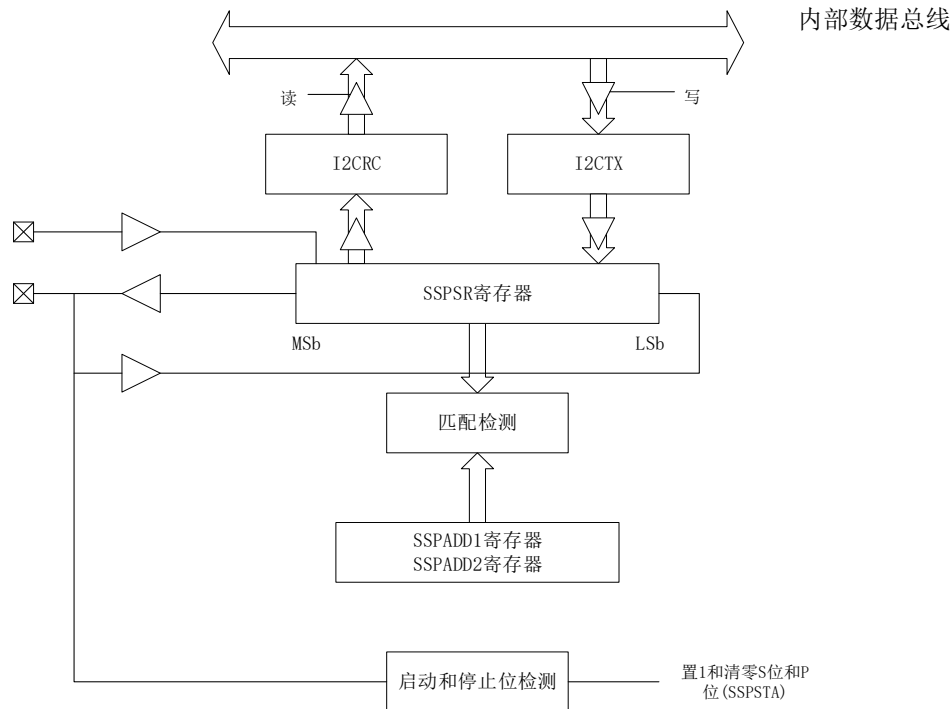
## 10 I2C 控制器

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）-- 可以映射到所有 IO
- 串行数据（SDA）-- 可以映射到所有 IO

**I2C 作为调试接口时间，禁止配置 I2C 寄存器，否则会影响断点调试功能。**

**I2C 工作时间，需要打开引脚开漏功能才能正常工作。**



I<sup>2</sup>C 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。I2CRC 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，SSPADD 寄存器将保存器件的地址。SSPADD 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、I2CTX 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 I2CRC 寄存器，同时将中断标志位 I2C\_RCIF 置 1；同时如果 I2CTX 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，I2CTX 与 SSPSR 构成双重缓冲发送。

## 10.1 I<sup>2</sup>C 相关寄存器：

### I2CACKDLY (I2C 的延时控制寄存器)

地址：0XF22

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 $T_{ACKDLY}=ACKDLY * 8 / F_{SYS}$	R/W	0

### I2CTX (I2C 的发送寄存器)

地址：0XF21

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

### I2CRC (I2C 的接收寄存器)

地址：0XF20

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

### I2CADD0 (I2C 的地址寄存器)

地址：0XF1F

Bit	Name	Description	Attribute	Reset
7	ADD0EN	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

### I2CADD1 (I2C 的地址寄存器)

地址：0XF1E

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

## I2CADDMASK (I2C 的寄存控制器)

地址: 0XF1D

Bit	Name	Description	Attribute	Reset
7:0	I2CADDMASK[7:0]	I2CADDMASK[7]是 I2C 地址掩码的使能 1: I2C 的 7 位器件地址位与 I2CADDMASK[6:0] 进行比对 0: I2C 的 7 位器件地址位与 I2CADDMASK[6:0] 不进行比对 在从机模式下: I2C 的 7 位器件地址位与 I2CADDMASK[6:0] 进行匹配	R/W	0

## I2CSTAT (I2C 的控制寄存器)

地址: 0XF1C

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5	I2C_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 <b>注:</b> 当复位或 SSPEN 被清零时, 该位被清零。	R	0
3	S	启动位	R	0

		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		
2	R/W	读/写信息位（仅用于 I <sup>2</sup> 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

## I2CCON0（I2C 的控制寄存器）

地址：0XF1B

Bit	Name	Description	Attribute	Reset
7	I2C_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） 0: 未发生冲突 在接收模式（主机或从机模式）下：在此忽略。	R/W	0
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	I2C 引脚使能位 1: 使能 SDA 和 SCL 引脚	R/W	0

		0: 禁止 SDA 和 SCL 引脚 <b>注:</b> 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		
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			

## I2CCON1 (I2C 的控制寄存器)

地址: 0XF1A

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 I<sup>2</sup>C 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)。

## I2CIE 寄存器

地址：0XF19

Bit	Name	Description	Attribute	Reset
7:5	Reserved			
4	I2C_FLOATIE	I2C 总线低时间超过 64ms，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

## I2CIF 寄存器

地址：0XF18

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	R/W	0

		1: 产生中断, 软件清 0 0: 没有中断		
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 64ms。

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。

## I2CIP 寄存器

地址: 0XF17

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

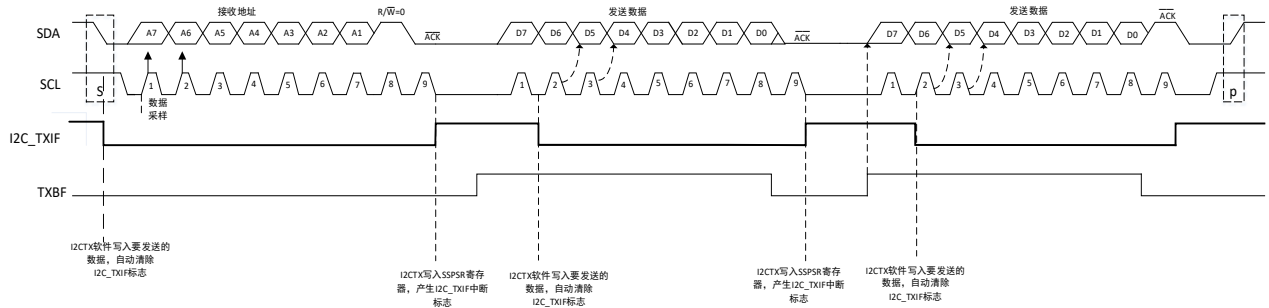
在从机模式下，一旦使能了 I2C 从机模块，它就会等待启动条件出现。启动条件出现后，就会向 SSPSR 寄存器移入 8 位数据。在时钟信号（SCL）的上升沿采样所有的输入位。在第 8 个时钟（SCL）脉冲的下降沿，寄存器 SSPSR<7:1>的值会和 I2CADD0 地址寄存器的值进行比较。如果地址匹配，并且 RCBF 位和 I2COV 位都被清零，会发生下列事件：

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，I2CSTAT 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）。

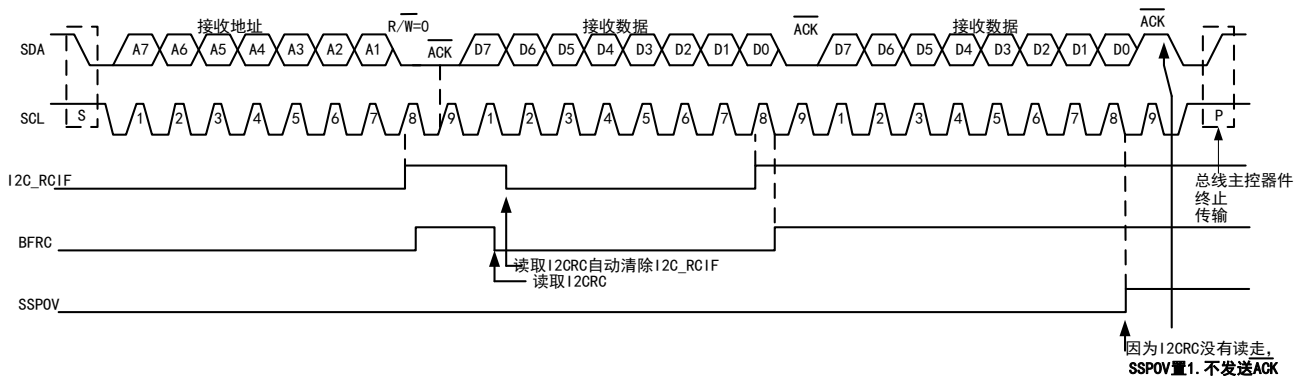
当接收的地址字节的 R/W 位为 1 且地址匹配时，I2CSTAT 寄存器的 R/W 位置 1。接收到的地址被装入 I2CRC 寄存器。ACK 脉冲在第 9 位上发送。从器件可以采用 I2CTX 提前准备好发送数据，并进行发送（可以在 I2C 配置时，写入一个 I2CTX 寄存器），当发送时被装入了 SSPSR 寄存器。8 个数据位在 SCL 时钟输入的下降沿被移出。

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

### 10.3 支持广播呼叫地址

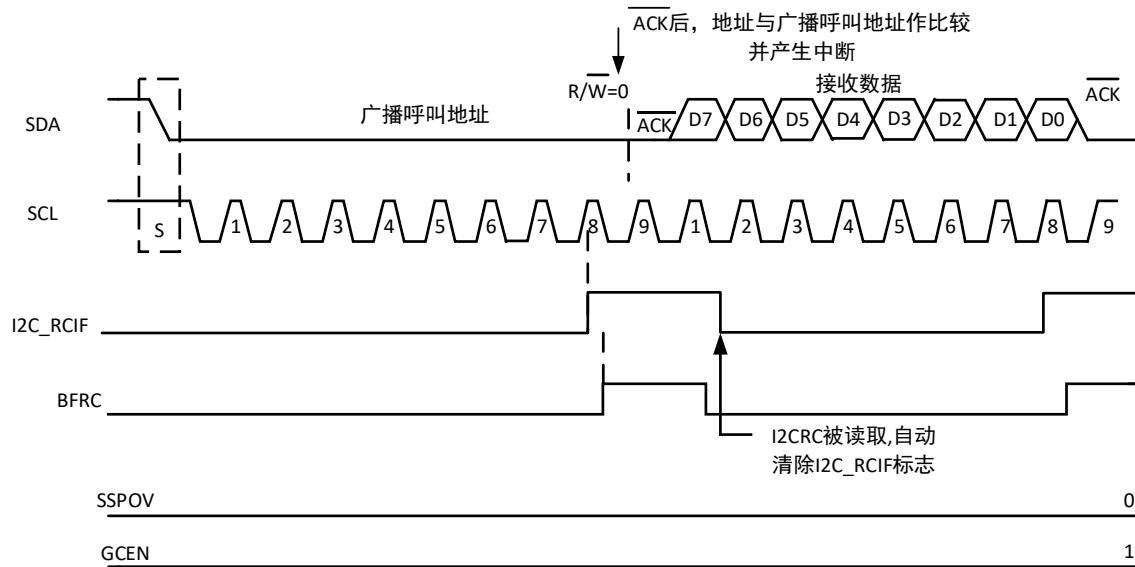
在 I<sup>2</sup>C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答信号来响应。同时支持两个地址选择，并支持 MASK 屏蔽选择。

广播呼叫地址是由 I<sup>2</sup>C 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

## 10.4 I2C 相关寄存器

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value		
0xf22	I2CACKDLY	ACKDLYON	ACKDLY[6:0]								0000 0000	0000 0000	
0xf21	I2CTX	I2C Transmit Register									0000 0000	uuuu uuuu	
0xf20	I2CRC	I2C Receive Register									0000 0000	uuuu uuuu	
0xf1f	I2CADD0	ADD0EN	I2C SLAVE Address Register									0000 0000	0000 0000
0xf1e	I2CADD1	ADD1EN	I2C SLAVE Address Register									0000 0000	0000 0000
0xf1d	I2CADDMASK	MSSP Address MASK Register in i2c Slave Mode.									0000 0000	0000 0000	
0xf1c	I2CSTAT	SMP1	SMP0	D/A	P	S	R/W	BPRX	BPTX	0000 0000	0000 0000		
0xf1b	I2CCON0	WCOL	I2COV	I2CEN	I2CSTEN					0000 0000	0000 0000		
0xf1a	I2CCON1	GCEN	ACKSTAT	ACKGET	ACKEN					0000 0000	0000 0000		
0xf19	I2CIE				I2C_FLOATIE	I2C_STOPIE	I2C_STARTIE	I2C_TXIE	I2C_RCIE	--00 0000	--00 0000		
0xf18	I2CIF				I2C_FLOATIF	I2C_STOPIF	I2C_STARTIF	I2C_TXIF	I2C_RCIF	--00 0000	--00 0000		
0xf17	I2CIP				I2C_FLOATIP	I2C_STOPIP	I2C_STARTIP	I2C_TXIP	I2C_RCIP	--11 1111	--11 1111		



# 11 SPI 接口

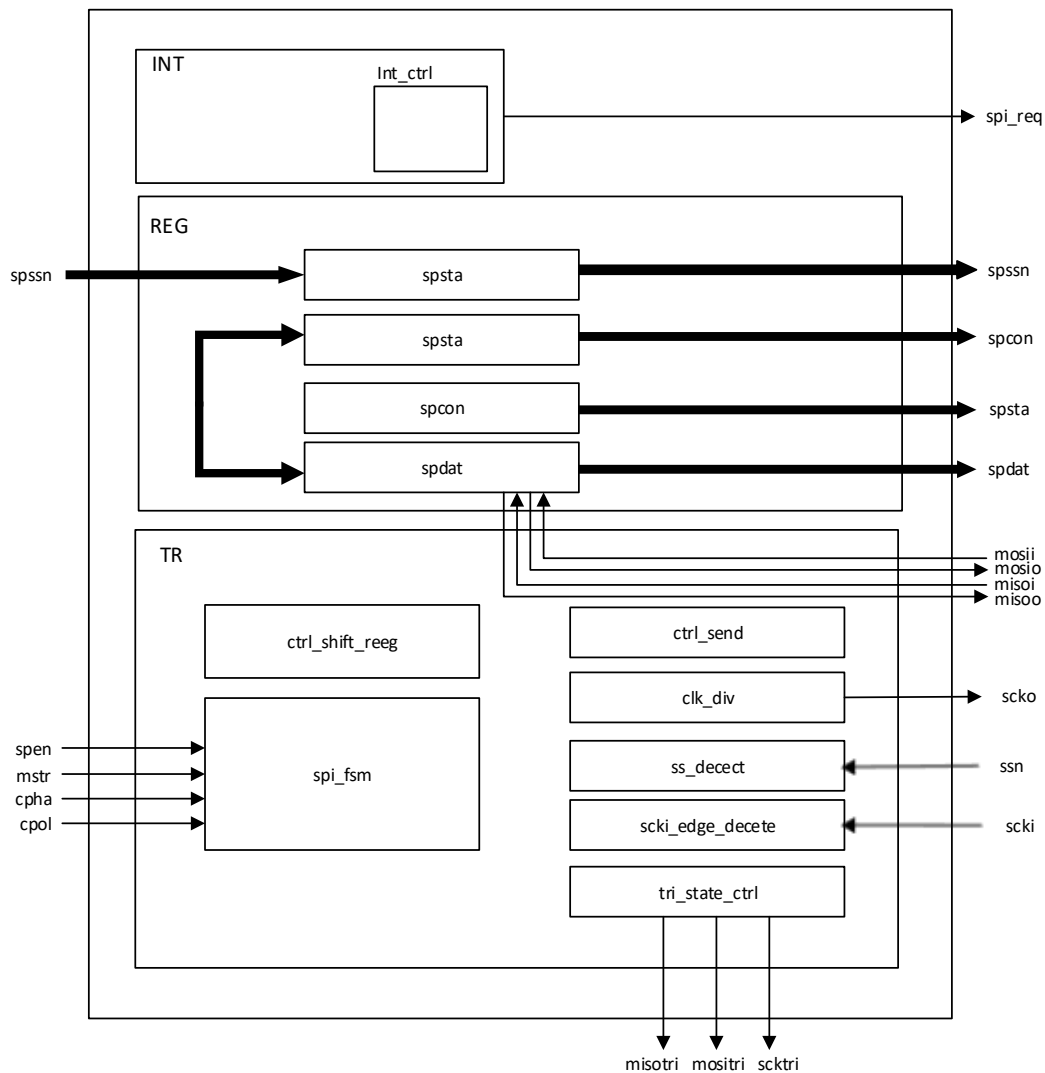
SPI 模块工作之前需要设置 CLKCFG2[7],即 SPICLKEN =1;

## 11.1 概述

芯片包含一个 SPI 接口，可以实现在 MCU 和外围设备（包含外部 MCU）之间的全双工同步串行通讯。SPI 模块可以编程实现以主模式或从模式工作，同时包含下列特征：

- ◇ 全双工模式，三线同步传输
- ◇ 主机和从机模式
- ◇ 通信速率可设置，最高可设置为  $f_{clk}/2$ ，最低为  $f_{clk}/128$
- ◇ 极性和相位可编程的串行时钟
- ◇ 具有写冲突处理机制
- ◇ 8 位数据传输，高字节在前，低字节在后
- ◇ 3 位从机选择接口，控制外部从机

## 11.2 模块框图



SPI 功能框图

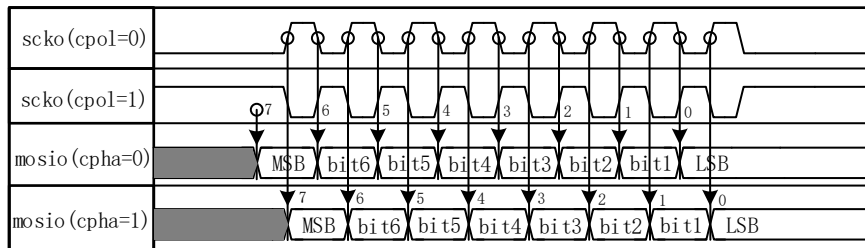
## 11.3 功能描述

串行外设接口(SPI)允许芯片与其他设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为从设备提供通信时钟(SCK)。

### 11.3.1. TR 接口传输格式

下图显示了数据传输的主要格式。根据 `SPI_MS` 模块的设置，数据的每一位在主时钟(SCKO)的上升沿(CPOL=0)或者下降沿(CPOL=1)被传送。数据在主时钟(SCKO)的下降沿(CPOL=0)或者上升沿(CPOL=1)被接收。这适用于主模式或从模式的传输器/接收器，前提是 SCKO 是传输工程中的主时钟。如果 `CPHA` 被置位，第一位(MSB)将在 SCKO 的第一个动态沿时通过 `MOSIO/MISOO` 被发送。如果 `CPHA` 被清零，第一位(MSB)将在 SCKO 的第一个动

态沿之前半个周期被发送。除此之外，输入数据(主模式时 MISOI，从模式是 MOSII)在每一位传输一半时被取样，在这个时钟周期的相反的电平上，数据被移位到输出信号 MOSIO 上。

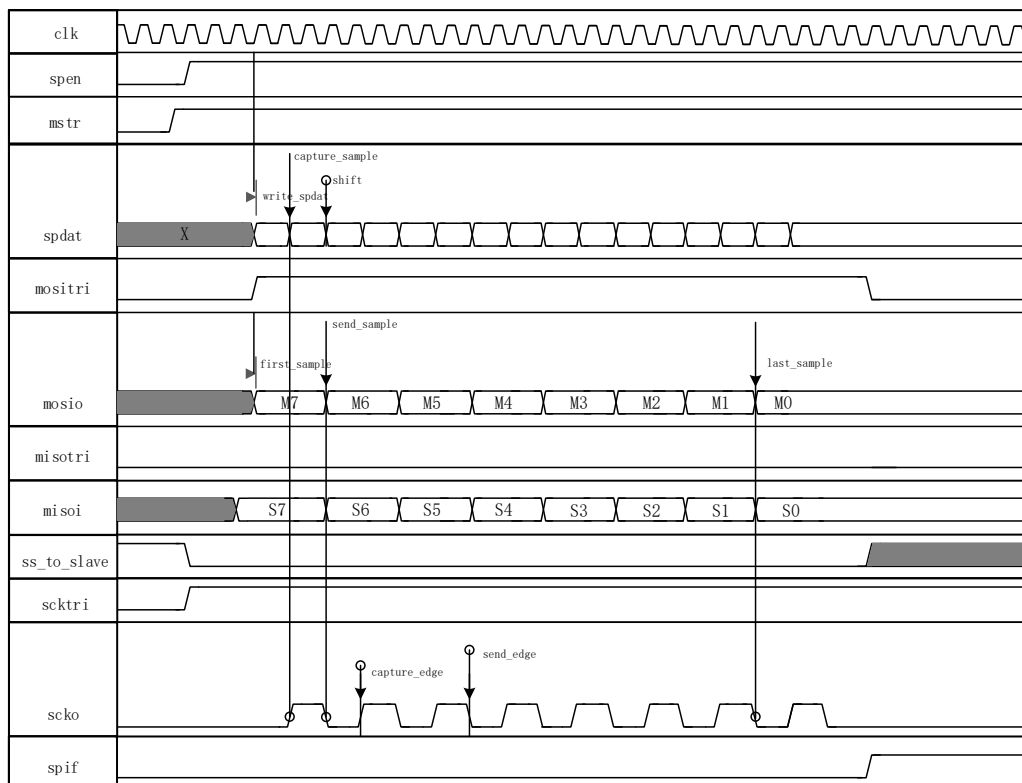


SPI\_MS 数据传输格式

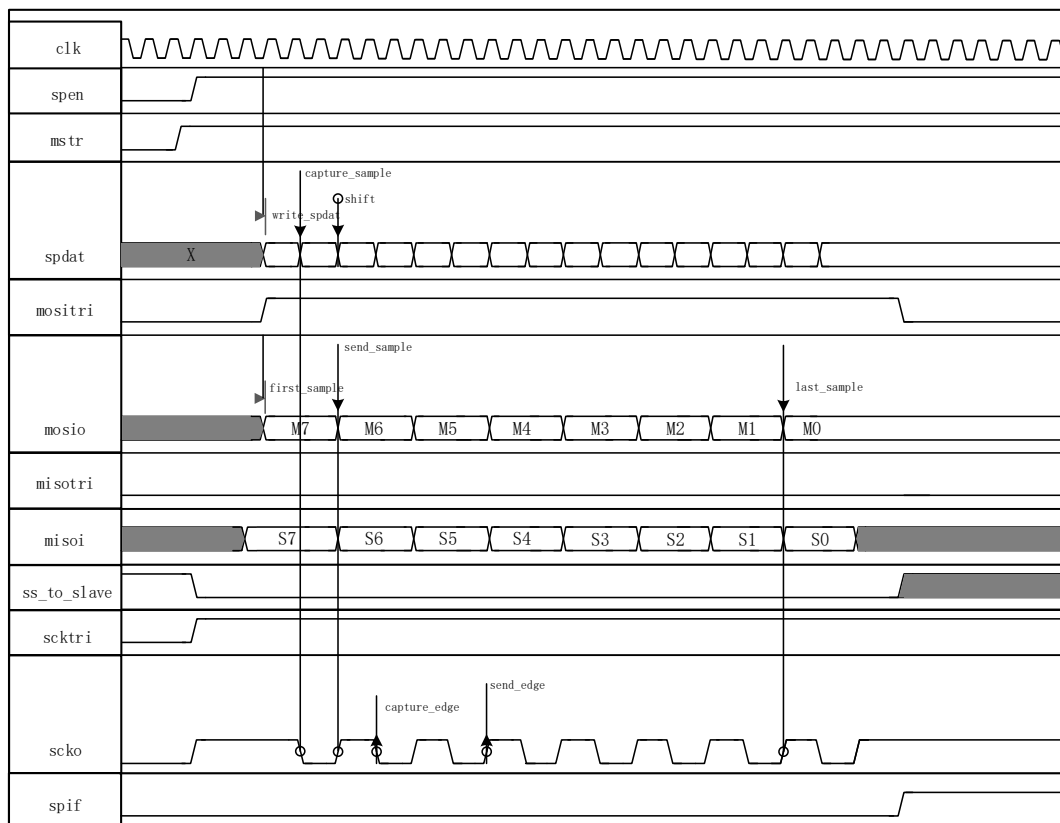
### 11.3.2. 主机模式传输格式

#### 11.3.2.1. SPI\_MS 默认主机模式：

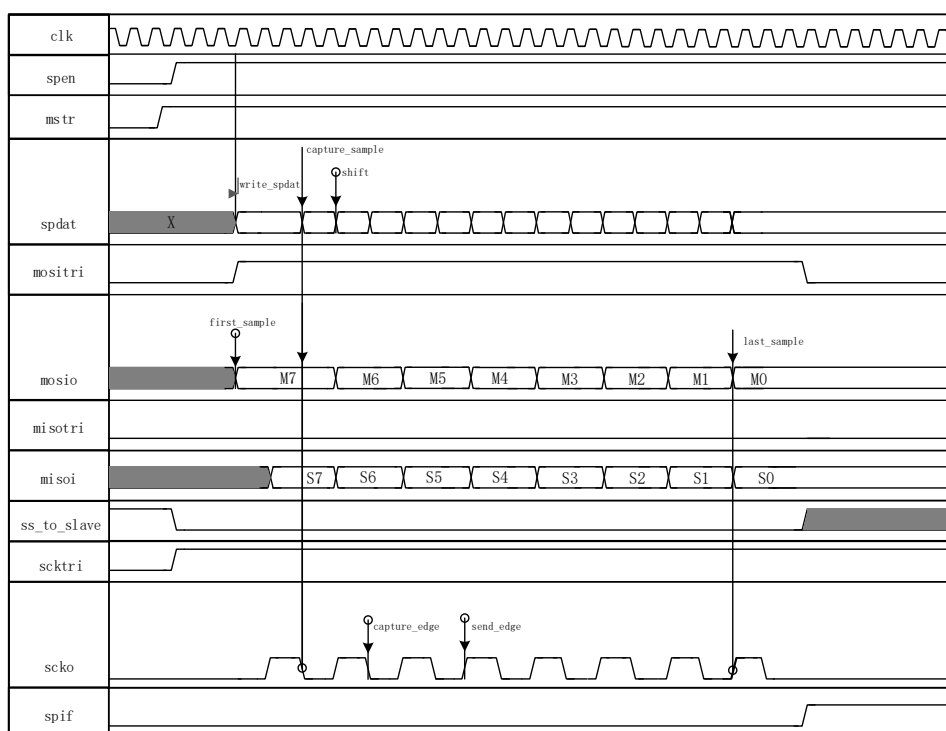
在主机模式中，SPI\_MS 等待程序向寄存器 SPDAT 中写入数据。如果向 SPDAT 的写入动作完成，传输就开始。在时钟 SCKO 的发送沿，数据被移位到输出引脚 MOSIO 上。同时，从从机传过来的另一个字节的数据被移位到主机的输入引脚 MISOI 上。



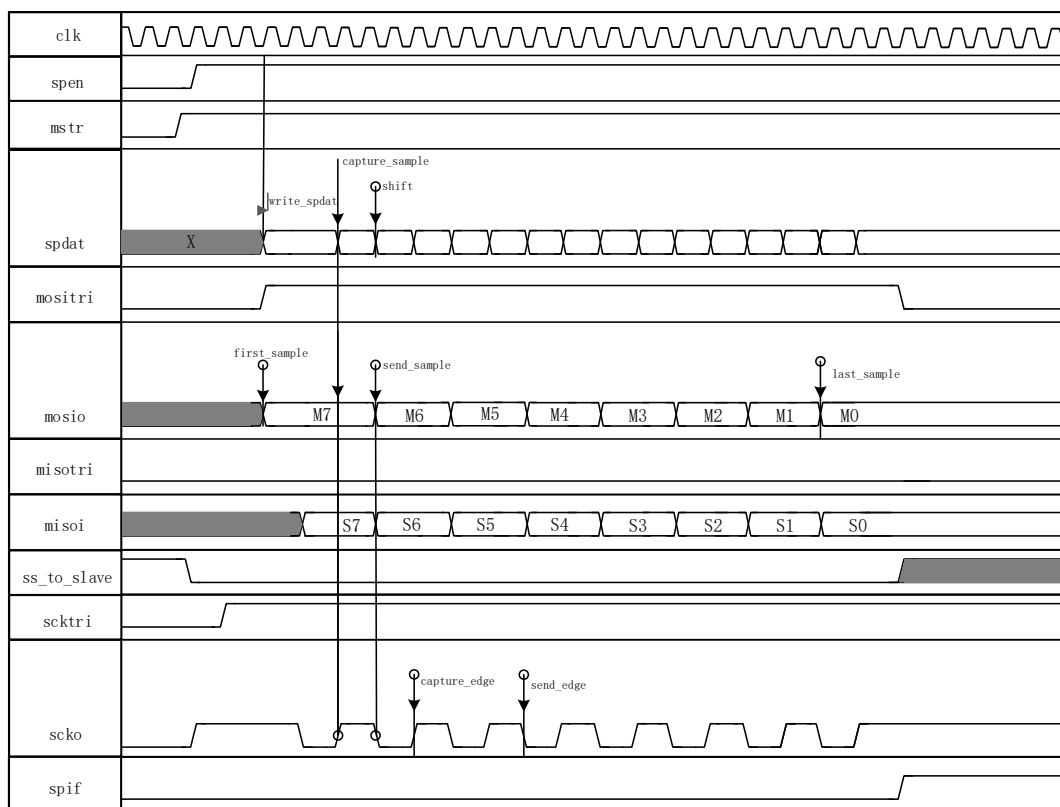
主机模式数据传输格式 (CPHA=0, CPOL=0)



主机模式数据传输格式 (CPHA=0, CPOL=1)



主机模式数据传输格式 (CPHA=1, CPOL=0)

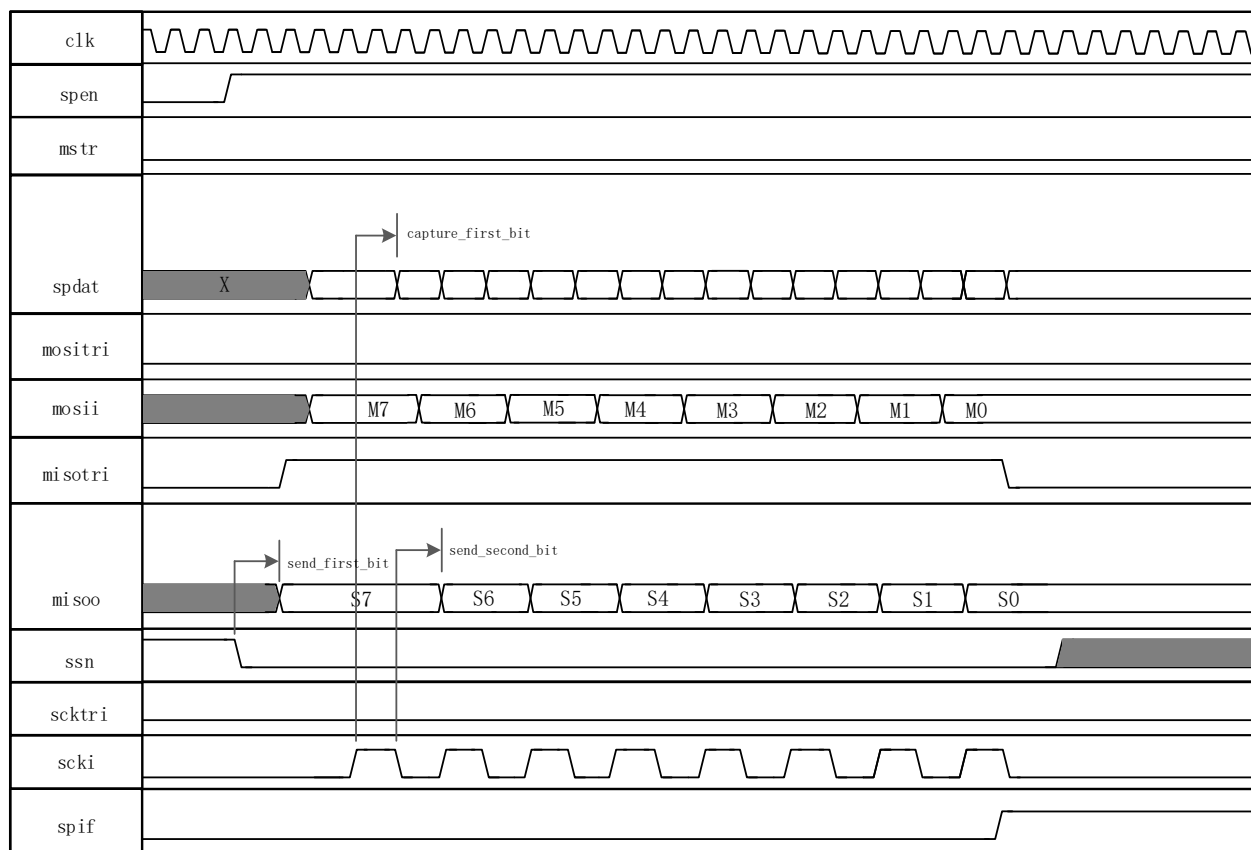


主机模式数据传输格式 (CPHA=1, CPOL=1)

### 11.3.2.2. 从机模式传输格式

首先，需要写寄存器 SPCON 中的 MSTR=0，以配置 PSI\_MS 进入从机模式，另外配置 SPEN=1 以打开 SPI\_MS 模块使能。

在从机模式中 SPI\_MS 等待输入信号 SSN 的低电平，当抓到 SSN 的下降沿时传输开始，直到传输完成，SSN 都需要保持低电平状态。寄存器 SPCON 中 CPHA 的状态决定传输的开始位置，当 CPHA 被清零时，从机必须在 SCKI 信号的第一个下降沿之前开始传输，当 CPHA 被置位，从机会把 SCKI 信号的第一个下降沿做为传输的开始标志。



从模式数据传输格式

### 11.3.3. 中断功能

SPI\_MS 提供 SPI 中断输出信号 INTSPI, 有两种状态产生中断请求。

名称	SPI_MS 中断标志描述
SPIF	当传输完成, 该标志位被硬件置位
MODF	当 SSN 与主从模式设置有冲突

## 11.4 寄存器列表

### SPCON 寄存器

地址: 0XF16

Bit	Name	Description	Attribute	Reset
7	SPIR2	SPI 时钟速率控制位 主模式时, SPI 时钟速率 SPR2、SPR1、SPR0 控制	R/W	0
6	SPIEN	SPI 使能位	R/W	0

		1: 使能 SPI 模块                      0: 关闭 SPI 模块		
5	SSDIS	SS 控制位  0: 在主/从模式打开 SSN 输入; 1: 在主从模式下关闭 SSN 输入, 该情况下不会产生 MODF 中断请求;  在从机模式中, 若 CPHA=0, 则该位无效	R/W	0
4	MSTR	SPI模式选择位  0: 从机模式 1: 主机模式	R/W	1
3	CPOL	时钟极性  0: SCK 在空闲状态时被设置为低电平 1: SCK 在空闲状态时被设置为高电平	R/W	0
2	CPHA	时钟相位  0: 表示高位 MSB 将 SCKO 动态沿之前半个周期被发送 1: 表示高位 MSB 将 SCKO 的第一个动态沿通过 MOSIO/MISOO 被发送	R/W	1
1: 0	SPIR[1:0]	SPI 时钟速率控制位  主模式时, SPI 时钟速率 SPR2、SPR1、SPR0 控制  000: Fsys/2 001: Fsys/4 010: Fsys/8 011: Fsys/16 100: Fsys/32 101: Fsys/64 110: Fsys/128 111: 不产生主时钟	R/W	00

## SPSTA 寄存器

地址：0XF15

Bit	Name	Description	Attribute	Reset																
7	Reserved																			
6	WCOL	写冲突标志位 当写 SPDAT 冲突时由硬件置位；当传输完成无冲突发生时由硬件复位，可通过访问寄存器 SPSTA 和 SPDAT 来复位；	R	0																
5	SSERR	同步从机错误标志位 在接收完成前当 SSN 输入有效时，被硬件置位；关闭 SPI 模块可清除该位，（设置 SPEN=0）	R	0																
4	MODF	模式故障标志位 当 SSN 引脚状态与设置的模式有冲突时，硬件自动置位；当 SSN 引脚恢复合适的电平状态时，硬件自动复位； 也可以由软件读 SPSTA 寄存器来复位	R	0																
3:2	SPINSS_MAP[1:0]	SPI 片选信号的 IO 映射选择  00: PA3 01: PA2 10: PD0 11: PB4	R/W	00																
1:0	SPI_MAP[1:0]	SPI 数据引脚和时钟引脚的 IO 映射选择  <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td></td> <td style="text-align: center;">00</td> <td style="text-align: center;">01</td> <td style="text-align: center;">11</td> </tr> <tr> <td>SPI_SCK</td> <td style="text-align: center;">PA2</td> <td style="text-align: center;">PD5</td> <td style="text-align: center;">PB7</td> </tr> <tr> <td>SPI_MOSI</td> <td style="text-align: center;">PA1</td> <td style="text-align: center;">PD6</td> <td style="text-align: center;">PB5</td> </tr> <tr> <td>SPI_MISO</td> <td style="text-align: center;">PA0</td> <td style="text-align: center;">PD7</td> <td style="text-align: center;">PB6</td> </tr> </table>		00	01	11	SPI_SCK	PA2	PD5	PB7	SPI_MOSI	PA1	PD6	PB5	SPI_MISO	PA0	PD7	PB6	R/W	00
	00	01	11																	
SPI_SCK	PA2	PD5	PB7																	
SPI_MOSI	PA1	PD6	PB5																	
SPI_MISO	PA0	PD7	PB6																	



## SPDAT 寄存器

地址：0XF14

Bit	Name	Description	Attribute	Reset
7:0	SPDAT[7:0]	数据寄存器	R/W	0x00

## 11.5 SPI 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xfad	IPR1	SPIP	EEIP	OSFIP	CCPIP	TMR3IP	TMR2IP	TMR1IP	TMR0IP	1111 1111	1111 1111
0xfac	PIR1	SPIF	EEIF	OSFIF	CCPIF	TMR3IF	TMR2IF	TMR1IF	TMR0IF	0000 0000	0000 0000
0xfab	PIE1	SPIE	EEIE	OSFIE	CCPIE	TMR3IE	TMR2IE	TMR1IE	TMR0IE	0000 0000	0000 0000
0xf16	SPCON	SPIR2	SPIEN	SSDIS	MSTR	CPOL	CPHA	SPIR[1:0]		0001 0100	0001 0100
0xf15	SPSTA		WCOL	SSERR	MODF	SPINSS_MAP[1:0]		SPI_MAP[1:0]		---- 0000	---- 0000
0xf14	SPDAT	SPDAT[7:0]								0000 0000	0000 0000

## 12 中断

AD18F06 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 和 PORTD 输入改变中断
- CCP 中断
- ADC 中断
- CMP 中断
- OP 中断
- 时钟故障检测中断

将 IPEN 位置位，可启用中断优先级功能，当中断优先级使能时，有 2 位允许全局中断。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 (AD18F06 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。



条指令。

## 12.2 Timer0 中断

TMR0 发生溢出 TMR0=PR0 时 TMR0IF 标志位置 1，标志位软件清 0；当 TMR0IE 位清零，该中断被屏蔽。

当 TMR0 溢出且 TMR0=PR0 时，将产生溢出中断，TMR0IF 标志位被置 1，并自动重新写入计数初值；当 TMR0IE 位清零时，该中断被屏蔽。

## 12.3 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，TMR1IF 标志位置 1，标志位软件清 0；TMR1IE 位清零，该中断被屏蔽。

## 12.4 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，TMR2IF 标志位置 1；TMR2IE 位清零，该中断被屏蔽。

## 12.5 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，TMR3IF 标志位置 1；TMR3IE 位清零，该中断被屏蔽。

## 12.6 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 (PIR3<4>). PAIE 位(PIE3<4>)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。上升沿和下降沿都可以触发中断。

**使能 PAINTMASK**，输入改变可以产生上升沿中断；

**使能 PAINTMASK 且在中断函数中读取对应中断的 PIN**，可以产生双沿中断（即上升沿和下降沿都可以触发中断。）

## 12.7 PortB 输入改变中断

输入改变中断触发时 PB<7:0> PBIF 标志位置 1 (PIR3<5>). PBIE 位(PIE3<5>)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。上升沿和下降沿都可以触发中断。

使能 PBINTMASK, 输入改变可以产生上升沿中断;

使能 PBINTMASK 且在中断函数中读取对应中断的 PIN, 可以产生双沿中断 (即上升沿和下降沿都可以触发中断。)

## 12.8 PortC 输入改变中断

输入改变中断触发时 PC<7:0> PCIF 标志位置 1 (PIR3<6>). PCIE 位(PIE3<6>)清零, 该中断被屏蔽。PCIE 在睡眠之前置 1, Port C 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。上升沿和下降沿都可以触发中断。

使能 PCINTMASK, 输入改变可以产生上升沿中断;

使能 PCINTMASK 且在中断函数中读取对应中断的 PIN, 可以产生双沿中断 (即上升沿和下降沿都可以触发中断。)

## 12.9 PortD 输入改变中断

输入改变中断触发时 PD<7:0> PDIF 标志位置 1 (PIR3<7>). PDIE 位(PIE3<7>)清零, 该中断被屏蔽。PDIE 在睡眠之前置 1, Port D 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。上升沿和下降沿都可以触发中断。

使能 PDINTMASK, 输入改变可以产生上升沿中断;

使能 PDINTMASK 且在中断函数中读取对应中断的 PIN, 可以产生双沿中断 (即上升沿和下降沿都可以触发中断。)

## 12.10 低电压、高电压中断

AD18F06 提供 16 组电压选择, 如下所示。当 LVDM[1:0]==2'b01 时, 系统 VDD 电压低于设定的 LVDM 电压值, LVDIF 位置为 1。LVDIE 位(PIE3<0>)清零, 该中断被屏蔽。

## 12.11 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(PIE4<2>)清零, CMP0 中断被屏蔽。

CMP1、CMP2、CMP3 的中断原理同 CMP0;

**双沿中断:** 需要在中断函数中读取 CMPxOUT 该比特位, 可以产生双沿中断。(即上升沿和下降沿都可以触发中断。)

## 12.12 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(PIE4<6>)清零, OP0 中断被屏蔽。

## 12.13 ADC 中断

当 GO 从 1 变为 0 时, ADIF 置位为 1. ADIE 位(PIE2<4>)清零, 该中断被屏蔽

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该中断标志; 当 ACEN 为 1 时, 只生成 ADCMP0IF 中断标志; 采集值大于 ADCMP1H 产生中断 ADCMP1IF 或者小于 ADCMP0H 产生中断 ADCMP0IF; 采集值小于 ADCMP1H 且大于 ADCMP0H, 产生中断 ADCMPIF0;

## 12.14 中断的相关寄存器

INTCON 寄存器

地址:0XFF2

Bit	Name	Description	Attribute	Reset
7	GIE/GIEH	全局中断使能位	R/W	0

		<p><u>当 IPEN=1 时:</u></p> <p>1: 允许所有高优先级中断</p> <p>0: 禁止所有高优先级中断</p> <p><u>当 IPEN=0 时:</u></p> <p>1: 允许所有未屏蔽的中断</p> <p>0: 禁止所有中断</p>		
6	PEIE/GIEL	<p>外设中断使能位</p> <p><u>当 IPEN=1 时:</u></p> <p>1: 允许所有低优先级的外设中断</p> <p>0: 禁止所有低优先级的外设中断</p> <p><u>当 IPEN=0 时:</u></p> <p>1: 允许所有未屏蔽的中断</p> <p>0: 禁止所有外设中断</p>	R/W	0
5:0	Reserved			

### IPR1 寄存器

地址:0XFAD

Bit	Name	Description	Attribute	Reset
7	SPIIP	SPI 中断优先级 1: 高优先级 0: 低优先级	R/W	1
6	EEIP	EEPROM 中断优先级 1: 高优先级 0: 低优先级	R/W	1
5	OSFIP	OSF 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	R/W	1

		0: 低优先级		
3	TMR3IP	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	TMR0IP	TMR0 中断优先级 1: 高优先级 0: 低优先级	R/W	1

### PIR1 寄存器

地址:0XFAC

Bit	Name	Description	Attribute	Reset
7	SPIIF	SPI 中断标志位 1: SPI 传送产生中断, 软件设置清零或者读寄存器 PIR1 并且访问 SPDAT 寄存器清除 0: 未产生 SPI 中断	R/W	0
6	EEIF	EEPROM 写操作中断标志位 1: EEPROM 产生中断, 软件设置清零 0: 未产生 EEPROM 中断	R/W	0
5	OSFIF	时钟检测中断标志位 1: 时钟检测产生中断, 软件设置清零 0: 未产生时钟检测中断	R/W	0
4	CCPIF	CCP 中断标志位 1: CCP 产生中断, 软件设置清零	R/W	0



		0: 未产生 CCP 中断		
3	TMR3IF	Timer3 溢出中断标志位 1: 产生 Timer3 溢出中断, 软件设置清零 0: 未产生 Timer3 溢出中断	R/W	0
2	TMR2IF	Timer2 溢出中断标志位 1: 产生 Timer2 溢出中断, 软件设置清零 0: 未产生 Timer2 溢出中断	R/W	0
1	TMR1IF	Timer1 溢出中断标志位 1: 产生 Timer1 溢出中断, 软件设置清零 0: 未产生 Timer1 溢出中断	R/W	0
0	TMR0IF	Timer0 溢出中断标志位 1: 产生 Timer0 溢出中断, 软件设置清零 0: 未产生 Timer0 溢出中断	R/W	0

### PIE1 寄存器

地址:0XFAB

Bit	Name	Description	Attribute	Reset
7	SPIIE	SPI 中断允许位 1: 使能 SPI 中断 0: 禁止 SPI 中断	R/W	0
6	EEIE	EEPROM 中断允许位 1: 使能 EEPROM 中断 0: 禁止 EEPROM 中断	R/W	0
5	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
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	TMR0IE	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

## IPR2 寄存器

地址:0XFAA

Bit	Name	Description	Attribute	Reset
7	Reserve			
6	ADCMP1IP	ADCMP1 中断优先级 1: 高优先级 0: 低优先级	R/W	1
5	ADCMP0IP	ADCMP0 中断优先级 1: 高优先级 0: 低优先级	R/W	1
4	ADIP	ADC 中断优先级 1: 高优先级 0: 低优先级	R/W	1
3	RCIP	UART 接收中断优先级 1: 高优先级 0: 低优先级	R/W	1

2	TXIP	UART 发送中断优先级 1: 高优先级 0: 低优先级	R/W	1
1	RC1IP	UART1 接收中断优先级 1: 高优先级 0: 低优先级	R/W	1
0	TX1IP	UART1 发送中断优先级 1: 高优先级 0: 低优先级	R/W	1

## PIR2 寄存器

地址:0XFA9

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	ADCMPIF	ADC 采集值比较器 1 中断标志位 1: 产生 ADC 采集值比较器 1 中断, 软件设置清零 0: 未产生 ADC 采集值比较器 1 中断	R/W	0
5	ADCMP0IF	ADC 采集值比较器 0 中断标志位 1: 产生 ADC 采集值比较器 0 中断, 软件设置清零 0: 未产生 ADC 采集值比较器 0 中断	R/W	0
4	ADIF	A/D 转换产生中断标志位 1: 产生 A/D 转换中断, 软件设置清零 0: 未产生 A/D 转换中断	R/W	0
3	RCIF	UART0 接收中断标志位 1: 产生 UART0 接收中断, 读取 RCREG 清除中断标志 0: 未产生 UART0 接收中断	R/W	0
2	TXIF	UART0 发送中断标志位 1: 产生 UART0 发送中断, 写入 TXREG 清除中断标志 0: 未产生 UART0 发送中断	R/W	0

1	RC1IF	UART1 接收中断标志位 1: 产生 UART1 接收中断, 读取 RCREG1 清除中断标志 0: 未产生 UART1 接收中断	R/W	0
0	TX1IF	UART1 发送中断标志位 1: 产生 UART1 发送中断, 写入 TXREG1 清除中断标志 0: 未产生 UART1 发送中断	R/W	0

## PIE2 寄存器

地址:0XFA8

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	ADCMP1IE	ADCMP1 中断允许位 1: 使能 ADCMP1 中断 0: 禁止 ADCMP1 中断	R/W	0
5	ADCMP0IE	ADCMP0 中断允许位 1: 使能 ADCMP0 中断 0: 禁止 ADCMP0 中断	R/W	0
4	ADIE	ADC 中断允许位 1: 使能 ADC 中断 0: 禁止 ADC 中断	R/W	0
3	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
2	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
1	RC1IE	UART1 接收中断允许位 1: 使能外部中断 0: 禁止外部中断	R/W	0

0	TX1IE	UART1 发送中断允许位 1: 使能外部中断 0: 禁止外部中断	R/W	0
---	-------	---	-----	---

### IPR3 寄存器

地址:0XFA7

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

		1: 高优先级 0: 低优先级		
--	--	--------------------	--	--

### PIR3 寄存器

地址:0XFA6

Bit	Name	Description	Attribute	Reset
7	PDIF	PortD 输入改变中断标志位 1: 产生 PortD 输入改变, 软件设置清零 0: 未产生 PortD 输入改变	R/W	0
6	PCIF	PortC 输入改变中断标志位 1: 产生 PortC 输入改变, 软件设置清零 0: 未产生 PortC 输入改变	R/W	0
5	PBIF	PortB 输入改变中断标志位 1: 产生 PortB 输入改变, 软件设置清零 0: 未产生 PortB 输入改变	R/W	0
4	PAIF	PortA 输入改变中断标志位 1: 产生 PortA 输入改变, 软件设置清零 0: 未产生 PortA 输入改变	R/W	0
3	INT2IF	INT2 外部中断标志位 1: 产生 INT2 外部中断标志位 0: 未产生 INT2 外部中断标志位	R/W	0
2	INT1IF	INT1 外部中断标志位 1: 产生 INT1 外部中断标志位 0: 未产生 INT1 外部中断标志位	R/W	0
1	INT0IF	INT0 外部中断标志位 1: 产生 INT0 外部中断标志位 0: 未产生 INT0 外部中断标志位	R/W	0
0	LVDIF	电压检测中断标志位 1: 产生电压检测中断, 软件设置清零	R/W	0

		0: 未产生电压检测中断		
--	--	--------------	--	--

### PIE3 寄存器

地址:0XFA5

Bit	Name	Description	Attribute	Reset
7	PDIE	PortD 输入改变中断允许位 1: 使能 PortD 输入改变中断 0: 禁止 PortD 输入改变中断	R/W	0
6	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
5	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
4	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
3	INT2IE	外部中断 2 允许位 1: 使能 INT2 中断 0: 禁止 INT2 中断	R/W	0
2	INT1IE	外部中断 1 允许位 1: 使能 INT1 中断 0: 禁止 INT1 中断	R/W	0
1	INT0IE	外部中断 0 允许位 1: 使能 INT0 中断 0: 禁止 INT0 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

**IPR4 寄存器**

地址:0XFA4

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	OP0IP	OP0 中断优先级 1: 高优先级 0: 低优先级	R/W	1
5	CMP3IP	CMP3 中断优先级 1: 高优先级 0: 低优先级	R/W	1
4	CMP2IP	CMP2 中断优先级 1: 高优先级 0: 低优先级	R/W	1
3	CMP1IP	CMP1 中断优先级 1: 高优先级 0: 低优先级	R/W	1
2	CMP0IP	CMP0 中断优先级 1: 高优先级 0: 低优先级	R/W	1
1	Reserved			
0	Reserved			

**PIR4 寄存器**

地址:0XFA3

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	OP0IF	OP0 中断标志位 1: 产生 OP0 中断标志位 0: 未产生 OP0 中断标志位	R/W	0



5	CMP3IF	CMP3 中断标志位 1: 产生 CMP3 中断标志位 0: 未产生 CMP3 中断标志位	R/W	0
4	CMP2IF	CMP2 中断标志位 1: 产生 CMP2 中断标志位 0: 未产生 CMP2 中断标志位	R/W	0
3	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
2	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
1	Reserved			
0	Reserved			

#### PIE4 寄存器

地址:0XFA2

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	OP0IE	OP0 中断允许位 1: 使能 OP0 中断 0: 禁止 OP0 中断	R/W	0
5	CMP3IE	CMP3 中断允许位 1: 使能 CMP3 中断 0: 禁止 CMP3 中断	R/W	0
4	CMP2IE	CMP2 中断允许位 1: 使能 CMP2 中断 0: 禁止 CMP2 中断	R/W	0
3	CMP1IE	CMP1 中断允许位	R/W	0

		1: 使能 CMP1 中断 0: 禁止 CMP1 中断		
2	CMP0IE	CMP0 中断允许位 1: 使能 CMP0 中断 0: 禁止 CMP0 中断	R/W	0
1	Reserved			
0	Reserved			

## 12.15 中断相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xf2	INTCON	GIE/GIEH	PEIE/GIEL							00-- ----	00-- ----
0xf1	PCON	IPEN	PB0ST		LVDM[1:0]			LVT[3:0]		1000 0000	1000 0000
0xfad	IPR1	SPIP	EIIP	OSFIP	CCPIP	TMR3IP	TMR2IP	TMR1IP	TMR0IP	1111 1111	1111 1111
0xfac	PIR1	SPIF	EEIF	OSFIF	CCPIF	TMR3IF	TMR2IF	TMR1IF	TMR0IF	0000 0000	0000 0000
0xfab	PIE1	SPIE	EEIE	OSFIE	CCPIE	TMR3IE	TMR2IE	TMR1IE	TMR0IE	0000 0000	0000 0000
0xfaa	IPR2		ADCMPIP	ADCMPOIP	ADIP	RCIP	TXIP	RC1IP	TX1IP	-111 1111	-111 1111
0xfa9	PIR2		ADCMPIF	ADCMPOIF	ADIF	RCIF	TXIF	RC1IF	TX1IF	-000 0000	-000 0000
0xfa8	PIE2		ADCMPIE	ADCMPOIE	ADIE	RCIE	TXIE	RC1IE	TX1IE	-000 0000	-000 0000
0xfa7	IPR3	PDIP	PCIP	PBIP	PAIP	INT2IP	INT1IP	INT0IP	LVDIP	1111 1111	1111 1111
0xfa6	PIR3	PDIF	PCIF	PBIF	PAIF	INT2IF	INT1IF	INT0IF	LVDIF	0000 0000	0000 0000
0xfa5	PIE3	PDIE	PCIE	PBIE	PAIE	INT2IE	INT1IE	INT0IE	LVDIE	0000 0000	0000 0000
0xfa4	IPR4		OP0IP	CMP3IP	CMP2IP	CMP1IP	CMP0IP			-111 1111	-111 1111
0xfa3	PIR4		OP0IF	CMP3IF	CMP2IF	CMP1IF	CMP0IF			-000 0000	-000 0000
0xfa2	PIE4		OP0IE	CMP3IE	CMP2IE	CMP1IE	CMP0IE			-000 0000	-000 0000

## 13 省电模式 (SLEEP)

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

- 000: IDLE 模式, CPU 停止工作, 外设工作正常; 所有中断可以唤醒, 唤醒后继续从当前 PC 运行;
- 001: PWSAVE 模式, CPU 停止工作, 高速 16M 时钟停止工作, 低速 32K 时钟工作, 支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 32K 定时唤醒, 唤醒后继续从当前 PC 运行;
- 010: DEEPPWSAVE 模式, CPU 停止工作, 高速 16M 时钟停止工作, 低速 32K 时钟工作, SRAM 数据保持; 支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 32K 定时唤醒, 唤醒后继续从当前 PC 运行;
- 011: PWOFF 模式, 全部外设和模拟停止工作, 支持外部中断、IO 中断、复位, 唤醒后继续从当前 PC 运行;

注: SLEEP 语句之后需加一条 NOP 指令;

### 13.1 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	√	√		√

CMP	√			
OP	√			
ADC	√	√		√

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

	ACTIVE	IDLE	PWSAVE	DEEPPWSAVE	PWOFF
<b>HIRC</b>	Y	Y	N	N	N
<b>LIRC</b>	Y	Y	Y	Y	N
<b>CPU</b>	Y	N	N	N	N
<b>SRAM</b>	Y	Y	Y	Y	Y
<b>FLASH</b>	Y	Y	N	N	N
<b>Timer0/2/3</b>	Y	Y	N	N	N
<b>Timer1</b>	Y	Y	Y	Y	N
<b>CCP</b>	Y	Y	N	N	N
<b>WDT</b>	Y	Y	Y	Y	N
<b>External Interrupt</b>	Y	Y	Y	Y	Y
<b>PAIF/PBIF/ PCIF/PDIF</b>	Y	Y	Y	Y	Y
<b>BGR</b>	Y	Y	Y	Y	N
<b>LVD/LVR</b>	Y	Y	Y	Y	N
<b>ADC</b>	Y	Y	Y(LIRC)	Y(LIRC)	N
<b>DAC</b>	Y	Y	N	N	N
<b>LCD</b>	Y	Y	Y	Y	N
<b>CMP</b>	Y	Y	N	N	N
<b>IO</b>	Y	Y	Y	Y	Y

<b>RESET</b>	Y	Y	Y	Y	Y
<b>备注</b>	1.LCD 进入 SLEEP 前应主动关闭 2.BGR LVR LVT 按 10%占空比定时开始工作（10%时间工作，90%时间睡眠）				

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向量。

### SMCR（状态控制寄存器）

地址：0XF10

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:1	SM[2:0]	休眠模式选择 000: IDLE 模式, CPU 停止工作, 外设工作正常; 所有中断可以唤醒, 唤醒后继续从当前 PC 运行; 001: PWSAVE 模式, CPU 停止工作, 高速 16M 时钟停止工作, 低速 32K 时钟工作, 支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 32K 定时唤醒, 唤醒后继续从当前 PC 运行; 010: DEEPPWSAVE 模式, CPU 停止工作, 高速 16M 时钟停止工作, 低速 32K 时钟工作; 支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 32K 定时唤醒, 唤醒后继续从当前 PC 运行; 011: PWOFF 模式, 全部外设和模拟停止工作, 支持外部中断、IO 中断、复位, 唤醒后继续从当前 PC 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	R/W	0

		0: 禁止休眠模式		
--	--	-----------	--	--

实例代码:

```

#define SLEEP_PWIDLE() SMCR = 0X01; SLEEP();NOP()

unsigned char t0;
Void init()
{
    PR0=0xaa;
    TMR0=0x00;
    T0CON=0x00;
    TMR0IE=1;
    TMR0IF=0;
    TMR0IP=1;
}

void main(void)
{
    init();
    t0=0;
    GIEH=1;
    while(1)
    {
        SLEEP_PWIDLE();
        if(t0==24)
        {
            PORTB=0x7f;
        }
    }
}

void __interrupt(high_priority) ISR_h(void)
{
    if(TMR0IF)
    {
        TMR0IF=0;
        t0++;
    }
}
    
```

## 13.2 SLEEP 相关寄存器定义

bin	address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
240	0xf10	SMCR					SM2	SM1	SM0	SE	---- 0000	---- 0000

## 14 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 1.0V、2.0V、3.0V。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 CMP 正端和负端参考电压
- OP0 正端和负端参考电压
- DAC0 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；

### FVRCON0 寄存器

地址：0XF70

Bit	Name	Description	Attribute	Reset
7	FVR_A2D_OUT	FVR 输出 IO 的状态	R	0
6	FVROUTEN	FVR 输出 IO 使能.通过 PA1 输出 1: 使能 FVR 输出 0: 禁止 FVR 输出	R/W	0
5	FVREN	FVR 使能 1: 使能 FVR 0: 禁止 FVR	R/W	0
4:3	FVRPGA[1:0]	FVR 电压放大倍数选择位 0X: 1 倍 10: 2 倍 11: 3 倍	R/W	00
2:0	FVR_SEL[2:0]	FVR 电压选择位 000: VREF1P0(1.0V) 001: PA3 010: NTC 011: DAC0_OUT 100: DAC1_OUT	R/W	000

		101: 1/4*VDD(来自 DAC1)		
		110: Reserved		
		111: Reserved		

### FVRCON1 寄存器

地址: 0XF6F

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5	FVR_OFFSET_EN	FVROFFSET CANCEL 使能信号 1: 允许使能 0: 禁止使能	R/W	0
4:0	FVR_TRIM[4:0]	FVR 的校准值	R/W	10000

## 14.1 FVR 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xf70	FVRCON0	FVR_A2D_OUT	FVROUTEN	FVREN	FVRPGA[1:0]		FVR_SEL[2:0]			0000 0000	0000 0000
0xf6f	FVRCON1			FVR_OFFSET	FVR_TRIM[4:0]					0000 0000	0000 0000



## 15 数模转换器(DAC)

使用 DAC0 模块 DAC1 模块之前需要配置 CLKCFG2[3]，即 DACCLKEN=1；

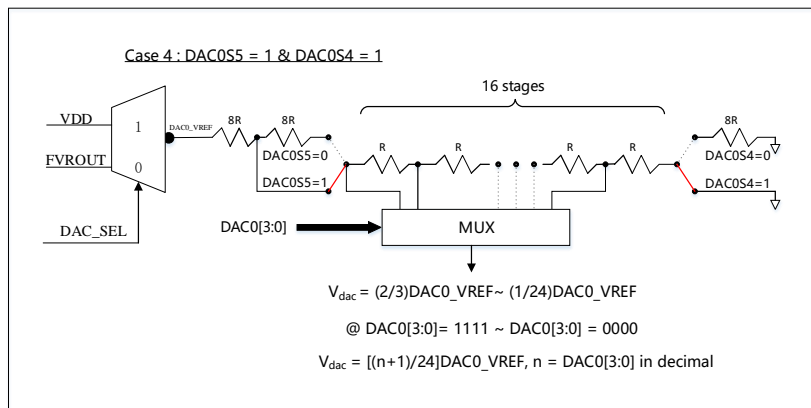
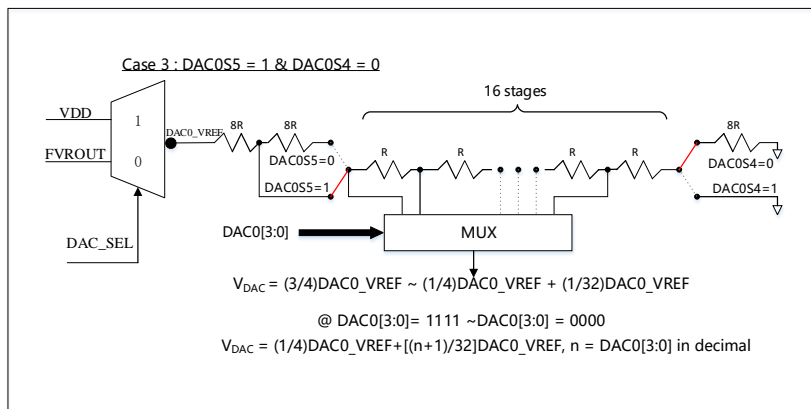
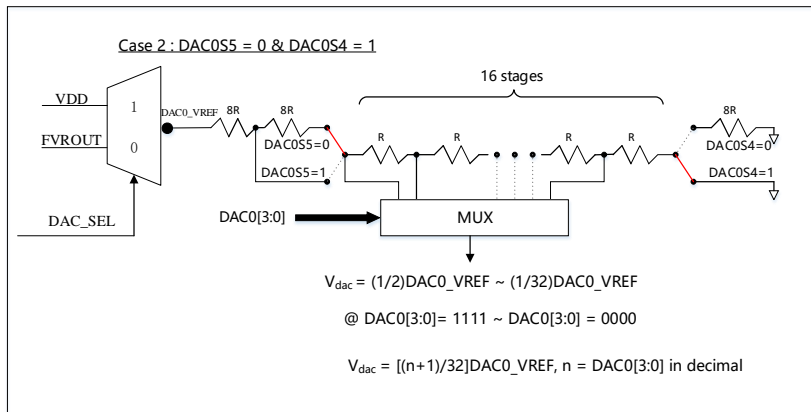
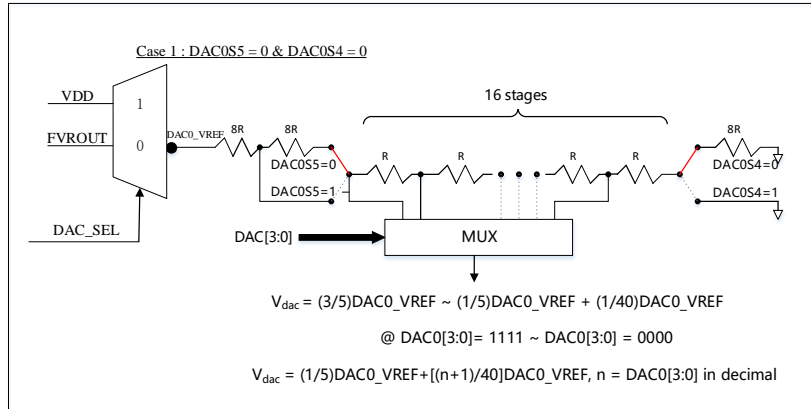
AD18F06 包含两个数模转换器 DAC。DAC0 是由一串电阻所组成，可以产生不同层次的参考电压，DAC0CON 寄存器的 4 和 5 位用来选择电阻串的最高和最低值；DAC0[3:0]用于选择所要的电压值，该值由 DAC0S5，DAC0S4 来决定。下图显示了四个不同选择时，内部参考电压值的计算。DAC0 输出电压范围可以从  $(1/32) * VDD$  到  $(3/4) * VDD$ 。

### 15.1 DAC 参考电压选择寄存器

DAC0CON

地址：0XF76

Bit	Name	Description	Attribute	Reset
7	DAC0VREF	DAC0 参考电压选择位 1: VDD      0: FVROUT	R/W	1
6	DAC0EN	DAC0 的使能 1: 允许使能   0: 禁止使能	R/W	0
5	DAC0S5	DAC0 正端电阻抽头选择	R/W	0
4	DAC0S4	DAC0 负端电阻抽头选择	R/W	0
3:0	DAC0[3:0]	DAC0 输出选择	R/W	0000

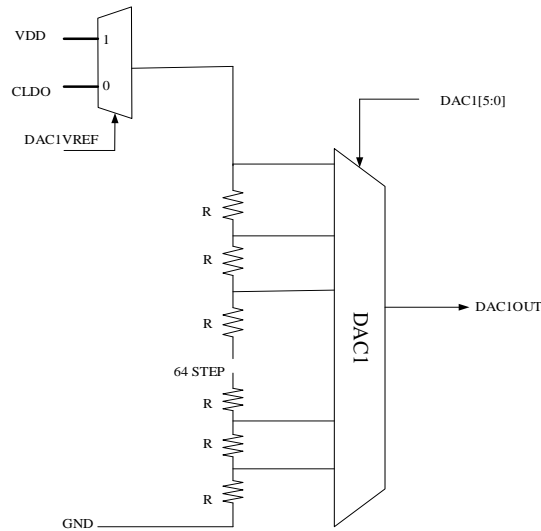


## 15.2 6Bit D/A 转换器

D/A 转换器提供了一个可变参考电压，与输入源成正比，具有 64 个可选输出电压。可通过 DAC1CON 的 DAC1VREF[7]位选择 DAC1 的输入为 VDD 或是 CLDO。

DAC 输出计算公式：

$$V_{out} = (V_{source}) * \frac{DAC1[5:0] + 1}{64}$$



D/A 结构图

### DAC1CON 寄存器

地址：0XF77

Bit	Name	Description	Attribute	Reset
7	DAC1_VREF	DAC1 的电压选择 1: VDD 0: CLDO	R/W	1
6	DAC1EN	DAC1 使能 1: 允许使能 0: 禁止使能	R/W	0
5:0	DAC1[5:0]	DAC1 电压输出选择位	R/W	0

## 15.3 DAC 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value	
0xf77	DAC1CON	DAC1VREF	DAC1EN	DAC1[5:0]							1000 0000	1000 0000
0xf76	DAC0CON	DAC0VREF	DAC0EN	DAC0S5	DAC0S4	DAC0[3:0]				1000 0000	1000 0000	
0xf4d	CLKCFG2					DACCLKEN				0000 0000	0000 0000	

## 16 模数转换器 (ADC)

使用 ADC 模块之前，需要设置 CLKCFG2[2],即 ADCCLKEN=1;

AD18F06 包含一个 32 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。

### 16.1 TIMER0 和 TIMER1 定时启动 ADC

TIMER0 和 TIMER1 分别可以定时启动 ADC 的自采集

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:ADRES 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。

### 16.2 ADC 的多路采集

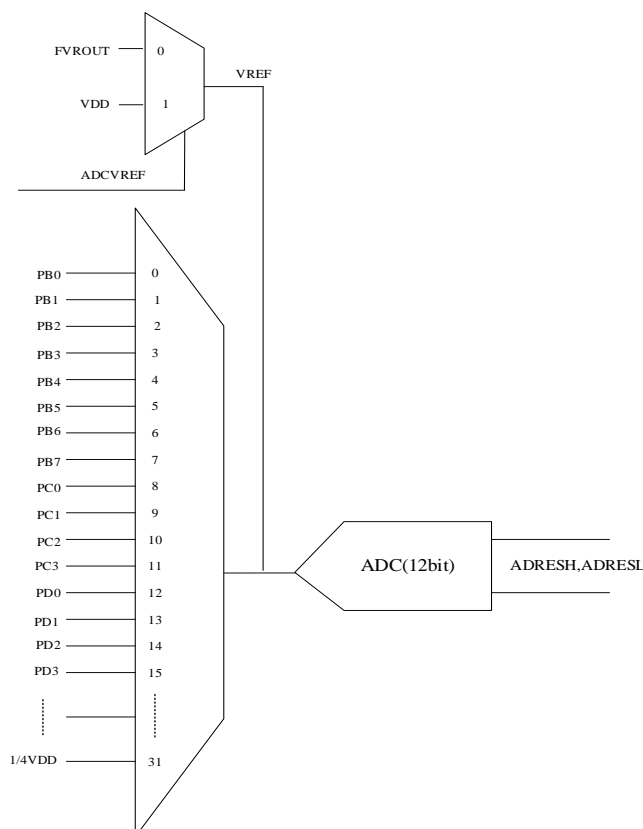
ADC 支持 4 路通道依次采集设定，内部支持 4 路\*12BIT 的 ADC 值缓存。

#### 16.2.1. 多路通道采集使用用方法：

- 1、采集 1 路时，仅支持采集 CH0[4:0]通道；仅支持上述通道组合的采集，不支持其他通道组合采集；采集完成后可产生中断；采集次数可以通过 ADCRPT[1:0]设置，结果寄存器存取最后一一次的采集值；
- 2、采集 2 路时，仅支持采集 CH0[4:0]通道和 CH1[4:0]通道，即需要使能 AD1EN，仅支持上述 2 路通道组合的采集，不支持其他通道组合采集；采集 2 路完成后可产生中断；采集次数可以通过 ADCRPT[1:0]设置，结果寄存器存取最后一一次的采集值；
- 3、采集 3 路时，仅支持采集 CH0[4:0]通道和 CH1[4:0]通道以及 CH2[4:0]，即需要使能 AD1EN 和 AD2EN；仅支持上述通道组合的采集，不支持其他通道组合采集；采集 3 路完成后可产生中断；采集次数可以通过 ADCRPT[1:0]设置，结果寄存器存取最后一一次的采集值；
- 4、采集 4 路时，仅支持采集 CH0[4:0]通道、CH1[4:0]通道、CH2[4:0]通道和 CH3[4:0]通

道，即需要使能 AD1EN 和 AD2EN 以及 AD3EN；仅支持上述通道组合的采集，不支持其他通道组合采集；采集 4 路完成后可产生中断；采集次数可以通过 ADCRPT[1:0] 设置，结果寄存器存取最后一次的采集值；

5、ADC 完整通道，查看寄存器 ADRESH0、ADRESH1、ADRESH2、ADRESH3 和 ADCON0 寄存器



A/D 结构图

### 16.3 A/D 转换步骤：

配置 A/D 模块

- 选择参考电压（通过 ADCON1[7]寄存器）
- 选择 A/D 输入通道（通过 ADRESH0[7:4]寄存器与 ADCON0[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。

#### ADCMP0H (AD 比较寄存器 0 高字节)

地址: 0XF5F

Bit	Name	Description	Attribute	Reset
7:0	ADCMP0H	AD 比较寄存器 0 的高 8 位	R/W	0X00

#### ADCMP1H(AD 比较寄存器 1 高字节)

地址: 0XF5E

Bit	Name	Description	Attribute	Reset
7:0	ADCMP1H	AD 比较寄存器 1 的高 8 位	R/W	0X00

#### ADCMP01L(AD 比较寄存器 0 和 1 低字节)

地址: 0XF5D

Bit	Name	Description	Attribute	Reset
7:4	ADCMP1L	AD 比较寄存器 1 的低 4 位	R/W	0000
3:0	ADCMP0L	AD 比较寄存器 0 的低 4 位	R/W	0000

#### ADRESH3 (AD 转换结果的高四位)

地址: 0XF5C

Bit	Name	Description	Attribute	Reset

7:4	AD3CH[3:0]	AD3CH[3:0]与 ADCON0 寄存器中的 AD3CH[4]组成 ADC 的 ADC3CH[4:0]种通道 AD3CH4:AD3CH0 – 第 4 路模拟通道选择位 00000: ADC 通道 0(PB0) 00001: ADC 通道 1(PB1) 00010: ADC 通道 2(PB2) 00011: ADC 通道 3(PB3) 00100: ADC 通道 4(PB4) 00101: ADC 通道 5(PB5) 00110: ADC 通道 6(PB6) 00111: ADC 通道 7(PB7) 01000: ADC 通道 8(PC0) 01001: ADC 通道 9(PC1) 01010: ADC 通道 10(PC2) 01011: ADC 通道 11(PC3) 01100: ADC 通道 12(PD0) 01101: ADC 通道 13(PD1) 01110: ADC 通道 14(PD2) 01111: ADC 通道 15(PD3) 10000: ADC 通道 16(PD4) 10001: ADC 通道 17(PD5) 10010: ADC 通道 18(PD6) 10011: ADC 通道 19(PD7) 10100: ADC 通道 20(PA5) 10101: ADC 通道 21(PA4) 10110: ADC 通道 22(PA3) 10111: ADC 通道 23(PA1) 11000: ADC 通道 24(PA0) 11001: ADC 通道 25(PC4)	R/W	00000
-----	------------	--	-----	-------

		11010: ADC 通道 26(PA2) 11011: ADC 通道 27(FVROUT) 11100: ADC 通道 28(DAC0OUT) 11101: ADC 通道 29(DAC1OUT) 11110: ADC 通道 30(OP0OUT) 11111: ADC 通道 31(1/4*VDD) 1/4*VDD 来自于 DAC1_OUT, 打开 DACCLKEN=1 和 DAC1EN 置 1 即可, 不受 DAC1 电压输出选择 位的影响;		
3:0	AD3RESH[3:0]	ADC 第 4 路转换结果的高四位	R/W	0000

### ADRESL3(AD 转换结果的低八位)

地址: 0XF5B

Bit	Name	Description	Attribute	Reset
7:0	AD3RESL	ADC 第 4 路转换结果的低 8 位	R/W	0X00

### ADRESH2 (AD 转换结果的高四位)

地址: 0XF5A

Bit	Name	Description	Attribute	Reset
7:4	AD2CH[3:0]	AD2CH[3: 0]与 ADCON0 寄存器中的 AD2CH[4] 组成 ADC 的 ADC2CH[4:0]种通道 AD2CH4:AD2CH0 – 第 4 路模拟通道选择位 00000: ADC 通道 0(PB0) 00001: ADC 通道 1(PB1) 00010: ADC 通道 2(PB2) 00011: ADC 通道 3(PB3) 00100: ADC 通道 4(PB4)	R/W	00000



	00101: ADC 通道 5(PB5) 00110: ADC 通道 6(PB6) 00111: ADC 通道 7(PB7) 01000: ADC 通道 8(PC0) 01001: ADC 通道 9(PC1) 01010: ADC 通道 10(PC2) 01011: ADC 通道 11(PC3) 01100: ADC 通道 12(PD0) 01101: ADC 通道 13(PD1) 01110: ADC 通道 14(PD2) 01111: ADC 通道 15(PD3) 10000: ADC 通道 16(PD4) 10001: ADC 通道 17(PD5) 10010: ADC 通道 18(PD6) 10011: ADC 通道 19(PD7) 10100: ADC 通道 20(PA5) 10101: ADC 通道 21(PA4) 10110: ADC 通道 22(PA3) 10111: ADC 通道 23(PA1) 11000: ADC 通道 24(PA0) 11001: ADC 通道 25(PC4) 11010: ADC 通道 26(PA2) 11011: ADC 通道 27(FVROUT) 11100: ADC 通道 28(DAC0OUT) 11101: ADC 通道 29(DAC1OUT) 11110: ADC 通道 30(OP0OUT) 11111: ADC 通道 31(1/4*VDD) 1/4*VDD 来自于 DAC1_OUT, 打开 DACCLKEN=1 和 DAC1EN 置 1 即可, 不受 DAC1 电压输出选择		
--	--	--	--

		位的影响;		
3:0	AD2RESH[3:0]	ADC 第 3 路转换结果的高四位	R/W	0000

### ADRESL2(AD 转换结果的低八位)

地址: 0XF59

Bit	Name	Description	Attribute	Reset
7:0	AD2RESL	ADC 第 3 路转换结果的低 8 位	R/W	0X00

### ADRESH1 (AD 转换结果的高四位)

地址: 0XF58

Bit	Name	Description	Attribute	Reset
7:4	AD1CH[3:0]	AD1CH[3: 0]与 ADCON0 寄存器中的 AD1CH[4] 组成 ADC 的 ADC1CH[4:0]种通道 AD1CH4:AD1CH0 – 第 2 路模拟通道选择位 00000: ADC 通道 0(PB0) 00001: ADC 通道 1(PB1) 00010: ADC 通道 2(PB2) 00011: ADC 通道 3(PB3) 00100: ADC 通道 4(PB4) 00101: ADC 通道 5(PB5) 00110: ADC 通道 6(PB6) 00111: ADC 通道 7(PB7) 01000: ADC 通道 8(PC0) 01001: ADC 通道 9(PC1) 01010: ADC 通道 10(PC2) 01011: ADC 通道 11(PC3) 01100: ADC 通道 12(PD0) 01101: ADC 通道 13(PD1) 01110: ADC 通道 14(PD2) 01111: ADC 通道 15(PD3)	R/W	00000

		10000: ADC 通道 16(PD4) 10001: ADC 通道 17(PD5) 10010: ADC 通道 18(PD6) 10011: ADC 通道 19(PD7) 10100: ADC 通道 20(PA5) 10101: ADC 通道 21(PA4) 10110: ADC 通道 22(PA3) 10111: ADC 通道 23(PA1) 11000: ADC 通道 24(PA0) 11001: ADC 通道 25(PC4) 11010: ADC 通道 26(PA2) 11011: ADC 通道 27(FVROUT) 11100: ADC 通道 28(DAC0OUT) 11101: ADC 通道 29(DAC1OUT) 11110: ADC 通道 30(OP0OUT) 11111: ADC 通道 31(1/4*VDD)  1/4*VDD 来自于 DAC1_OUT, 打开 DACCLKEN=1 和 DAC1EN 置 1 即可, 不受 DAC1 电压输出选择 位的影响;		
3:0	AD1RESH	ADC 第 2 路转换结果的高四位	R/W	0000

### ADRESL1(AD 转换结果的低八位)

地址: 0XF57

Bit	Name	Description	Attribute	Reset
7:0	AD1RESL	ADC 第 2 路转换结果的低 8 位	R/W	0X00

## ADRESH0 (AD 转换结果的高四位)

地址: 0XF56

Bit	Name	Description	Attribute	Reset
7:4	AD0CH[3:0]	AD0CH[3: 0]与 ADCON0 寄存器中的 AD0CH[4] 组成 ADC 的 ADC0CH[4:0]种通道 AD0CH4:AD0CH0 – 第 1 路模拟通道选择位  00000: ADC 通道 0(PB0) 00001: ADC 通道 1(PB1) 00010: ADC 通道 2(PB2) 00011: ADC 通道 3(PB3) 00100: ADC 通道 4(PB4) 00101: ADC 通道 5(PB5) 00110: ADC 通道 6(PB6) 00111: ADC 通道 7(PB7) 01000: ADC 通道 8(PC0) 01001: ADC 通道 9(PC1) 01010: ADC 通道 10(PC2) 01011: ADC 通道 11(PC3) 01100: ADC 通道 12(PD0) 01101: ADC 通道 13(PD1) 01110: ADC 通道 14(PD2) 01111: ADC 通道 15(PD3) 10000: ADC 通道 16(PD4) 10001: ADC 通道 17(PD5) 10010: ADC 通道 18(PD6) 10011: ADC 通道 19(PD7) 10100: ADC 通道 20(PA5) 10101: ADC 通道 21(PA4) 10110: ADC 通道 22(PA3)	R/W	0000

		10111: ADC 通道 23(PA1) 11000: ADC 通道 24(PA0) 11001: ADC 通道 25(PC4) 11010: ADC 通道 26(PA2) 11011: ADC 通道 27(FVROUT) 11100: ADC 通道 28(DAC0OUT) 11101: ADC 通道 29(DAC1OUT) 11110: ADC 通道 30(OP0OUT) 11111: ADC 通道 31(1/4*VDD) 1/4*VDD 来自于 DAC1_OUT, 打开 DACCLKEN=1 和 DAC1EN 置 1 即可, 不受 DAC1 电压输出选择 位的影响;		
3:0	AD0RESH	ADC 第 1 路转换结果的高四位	R/W	0000

### ADRESL0(AD 转换结果的低八位)

地址: 0XF55

Bit	Name	Description	Attribute	Reset
7:0	AD0RESL	ADC 第 1 路转换结果的低 8 位	R/W	0X00

### ADCON0 (ADC 控制寄存器 0)

地址: 0XF54

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],CH[3:0]} 0: 右对齐 ADRESH = {CH[3:0],adc[11:8]} 右对齐 ADRESL = adc[7:0]	R/W	0
6	ADCMPMODE	AD 采集值的比较中断模式选择;	R/W	0

		0: 采集值大于 ADCMP1H 产生中断 ADCMP1IF 或者小于 ADCMP0H 产生中断 ADCMP0IF; 1: 采集值小于 ADCMP1H 且大于 ADCMP0H, 产生中断 ADCMPIF0;		
5	AD3CH[4]	与 ADRESH3 中 AD3CH[3:0]组成 5 位的 ADC 通道	R/W	0
4	AD2CH[4]	与 ADRESH2 中 AD2CH[3:0]组成 5 位的 ADC 通道	R/W	0
3	AD1CH[4]	与 ADRESH1 中 AD1CH[3:0]组成 5 位的 ADC 通道	R/W	0
2	AD0CH[4]	与 ADRESH0 中 AD0CH[3:0]组成 5 位的 ADC 通道	R/W	0
1	GO/DONE	GO/DONE – 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

### ADCON1 (ADC 控制寄存器 1)

地址: 0XF53

Bit	Name	Description	Attribute	Reset
7	ADVREF	A/D 参考电压选择位  0: FVROUT 1: VDD	R/W	1
6	ADSP	ADC 采集速度选择位  1: 高速 (默认) 0: 低速	R/W	1
5:3	ACQT[2:0]	A/D 延时采集时间选择  111: 15 TAD 110: 13TAD 101: 11TAD	R/W	0

		100: 9TAD 011: 7 TAD 010: 5TAD 001: 3 TAD 000: 1TAD		
2:0	ADCS[2:0]	A/D 转换时钟选择位 111: 内部 32K 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	000

### ADCON2 (ADC 控制寄存器 2)

地址: **0XF52**

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 ADCMP0IF 00: 1 次 01: 2 次 10: 4 次 11: 8 次	R/W	00

1	ACFVR	<p>当 ACEN 使能时间， FVR 做 ADC 的参考时，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</p> <p>1: 使能 FVR 自动工作</p> <p>0: 禁止 FVR 自动工作</p> <p>注：1、FVREN 与 ACFVR 为或的关系</p> <p>2、VDD 做参考时自启动只需要使能 ACEN；</p> <p>3、FVR 作为 ADC 参考时，使能 ACFVR，同时 FVR 寄存器的配置按需设置即可</p>	R/W	0
0	ACEN	<p>自动采集使能</p> <p>1: 使能自动采集</p> <p>0: 禁止自动采集</p> <p>注：VDD 做参考时自启动只需要使能 ACEN；</p>	R/W	0

### ADCON3 (ADC 控制寄存器 3)

地址：0XF51

Bit	Name	Description	Attribute	Reset
7:5	Reserved			
4:3	ADCRPT[1:0]	<p>ADC 每个通道采集次数</p> <p>00: 1 次</p> <p>01: 2 次</p> <p>10: 3 次</p> <p>11: 4 次</p> <p>说明：例如设置每个通道采集 4 次，则前三次为空采集，不保留数据，仅保留最后一次采集的数据。</p>	R/W	00
2	AD3EN	<p>ADC 第 4 路的采集使能</p> <p>1: 允许使能</p>	R/W	0



		0: 禁止使能		
1	AD2EN	ADC 第 3 路的采集使能 1: 允许使能 0: 禁止使能	R/W	0
0	AD1EN	ADC 第 2 路的采集使能 1: 允许使能 0: 禁止使能	R/W	0

## 16.4 ADC 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xf5f	ADCMPOH	A/D Result Compare Register 0 High Byte								0000 0000	0000 0000
0xf5e	ADCMP1H	A/D Result Compare Register 1 High Byte								0000 0000	0000 0000
0xf5d	ADCMP01L	A/D Result Compare Register 1 Low[3:0]				A/D Result Compare Register 3 Low[3:0]				0000 0000	0000 0000
0xf5c	ADRESH3	AD3CH[3:0]				A/D Result Register 3 High Byte				---- 0000	---- 0000
0xf5b	ADRESL3	A/D Result Register Low Byte								0000 0000	0000 0000
0xf5a	ADRESH2	AD2CH[3:0]				A/D Result Register 2 High Byte				---- 0000	---- 0000
0xf59	ADRESL2	A/D Result Register 2 Low Byte								0000 0000	0000 0000
0xf58	ADRESH1	AD1CH[3:0]				A/D Result Register 1 High Byte				---- 0000	---- 0000
0xf57	ADRESL1	A/D Result Register 1 Low Byte								0000 0000	0000 0000
0xf56	ADRESH0	AD0CH[3:0]				A/D Result Register 0 High Byte				---- 0000	---- 0000
0xf55	ADRESL0	A/D Result Register 0 Low Byte								0000 0000	0000 0000
0xf54	ADCON0	ADFM	ADCMPOH	AD3CH[4]	AD2CH[4]	AD1CH[4]	AD0CH[4]	GO/DONE	ADON	0000 0000	0000 0000
0xf53	ADCON1	ADVREF	ADCSP	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	1100 0000	1100 0000
0xf52	ADCON2	ACQT[6:3]				ANC[1:0]		ACFVR	ACEN	0000 0000	0000 0000
0xf51	ADCON3				ADCRPT[1:0]		AD3EN	AD2EN	AD1EN	--0 0000	--0 0000
0xf5a	IPR2		ADCMP1IP	ADCMP0IP	ADIP					-111 1111	-111 1111
0xf59	PIR2		ADCMP1IF	ADCMP0IF	ADIF					-000 0000	-000 0000
0xf58	PIE2		ADCMP1IE	ADCMP0IE	ADIE					-000 0000	-000 0000

注意:

- 1、使用 ADC 自动采集功能时必须设置 **TIMER0** 或 **TIMER1** 中任意一个定时器，且自动采集时间间隔由 **TIMER0** 或者 **TIMER1** 定时时间确定。
- 2、在 **SLEEP\_DEEPPWSAVE**、**SLEEP\_PWSAVE** 模式下，**ADCS** 无法设置为内部 **OSC32K OSC**
- 3、ADC 的转换时间=ACQT(延时采集时间)+13TAD
- 4、完成一次的转换的时间定义为 **TAD**

例子:

ADC 通过 **TIMER1** 自启动 ADC 的采集部分配置示例

```

unsigned char adc0int;
void main (void)
{

```

```
unsigned char i,j;
unsigned short val;

ADCCLKEN =1;
T1CLKEN = 1;
//-----定时-----
TMR1H = 0xFF;
TMR1L = 0xF0;
T1CON0 = 0x43;//timer1的配置
T1CON1 = 0x1F;//timer1的定时启动ADC的模式

GIEH      = 1;
IPEN      = 1;

TMR1IP    = 1;
TMR1IE    = 0;
TMR1IF    = 0;

ADCMP0IE  = 1;
ADCMP0IP  = 1;
ADCMP0IF  = 0;

ADCMP1IE  = 1;
ADCMP1IP  = 1;
ADCMP1IF  = 0;

ADCMP0H   = 0x55;
ADCMP1H   = 0xaa;
ADCON1    = 0x40;
ADCON0    = 0x01;
ADCON2    = 0x1F;//使能ADC的自启动配置
GO = 1;
While (GO) ;
}
void __interrupt(high_priority) ISR_h(void) {
    if (ADCMP0IE && ADCMP0IF) {

        ADCMP0IF = 0;
        adc0int=adc0int+1;
        TMR0=adc0int;
    }
    if (ADCMP1IE && ADCMP1IF) {
        ADCMP1IF = 0;
    }
}
```

```

    }
    if (TMR1IE && TMR1IF) {
        TMR1IF = 0;
    }
}
    
```

### ADC 通过 TIMER0 自启动 ADC 的采集部分配置示例

```

unsigned char adc0int;
void main (void)
{
    unsigned char i,j;
    unsigned short val;

    ADCCLKEN =1;
    TOCLKEN = 1;

    TMR0     = 0Xaa;
    TOCON0   = 0X00; //timer0的配置

    GIEH     = 1;
    IPEN     = 1;

    TMR0IP   = 1;
    TMR0IE   = 0;
    TMR1OF   = 0;

    ADCMP0IE = 1;
    ADCMP0IP = 1;
    ADCMP0IF = 0;

    ADCMP1IE = 1;
    ADCMP1IP = 1;
    ADCMP1IF = 0;

    ADCMP0H  = 0X55;
    ADCMP1H  = 0Xaa;

    ADCON1   = 0X40;
    ADCON0   = 0X01;
    ADCON2   = 0X1F; //使能ADC的自启动配置
    GO = 1;
    While (GO) ;
}
void __interrupt(high_priority) ISR_h(void){
    
```

```
    if (ADCMP0IE && ADCMP0IF) {  
  
        ADCMP0IF = 0;  
        adc0int=adc0int+1;  
        TMR0=adc0int;  
    }  
    if (ADCMP1IE && ADCMP1IF) {  
        ADCMP1IF = 0;  
    }  
    if (TMR1IE && TMR1IF) {  
        TMR1IF = 0;  
    }  
}
```

#### ADC 的 4 路采集和通过硬件控制采集次数部分配置示例

```
    ADCON3    = 0X00;  
    ADRESH0   = 0X00;  
    GO = 1;  
    while(GO);  
    val = ((ADRESH0 & 0xf)<<8)|ADRESL0;
```

```
    if(val !=1630)
        PORTB =0X01;

    ADCON3      = 0X01 | (1<<3);
    ADRESH1 = 0X10;

    while(GO);
        val = ((ADRESH1 & 0xf)<<8)|ADRESL1;
        if(val !=2450)
            PORTB =0X02;

    ADCON3      = 0X03 | (2<<3);
    ADRESH2 = 0X20;

    while(GO);
        val = ((ADRESH2 & 0xf)<<8)|ADRESL2;
        if(val !=3270)
            PORTB =0X03;

    ADCON3      = 0X07 | (3<<3);
    ADRESH3 = 0X30;
    GO = 1;
    while(GO);
        val = ((ADRESH3 & 0xf)<<8)|ADRESL3;
        if(val !=1638)
            PORTB =0X04;

    ADCON3      = 0X03 | (2<<3);
    ADRESH3 = 0X30;

    while(GO);

    ADCON3      = 0X01 | (1<<3);
    ADRESH3 = 0X30;

    while(GO);

    ADCON3      = 0X00;
    ADRESH3 = 0X30;

    while(GO);

    ADCON3      = 0X00 | (3<<3);
```

```
ADRESH0 = 0X00;

while(GO);
    val = ((ADRESH0 & 0xf)<<8)|ADRESL0;
    if(val !=1630)
        PORTB =0X01;

ADCON3    = 0X01 | (2<<3);
ADRESH1 = 0X10;

while(GO);
    val = ((ADRESH1 & 0xf)<<8)|ADRESL1;
    if(val !=2450)
        PORTB =0X02;

ADCON3    = 0X03 | (1<<3);
ADRESH2 = 0X20;

while(GO);
    val = ((ADRESH2 & 0xf)<<8)|ADRESL2;
    if(val !=3270)
        PORTB =0X03;

ADCON3    = 0X07 | (0<<3);
ADRESH3 = 0X30;

while(GO);
    val = ((ADRESH3 & 0xf)<<8)|ADRESL3;
    if(val !=1638)
        PORTB =0X04;

ADCON3    = 0X03 | (1<<3);
ADRESH3 = 0X30;

while(GO);

ADCON3    = 0X01 | (2<<3);
ADRESH3 = 0X30;

while(GO);
```

```
ADCON3 = 0X00 | (3<<3);
```

```
ADRESH3 = 0X30;
```

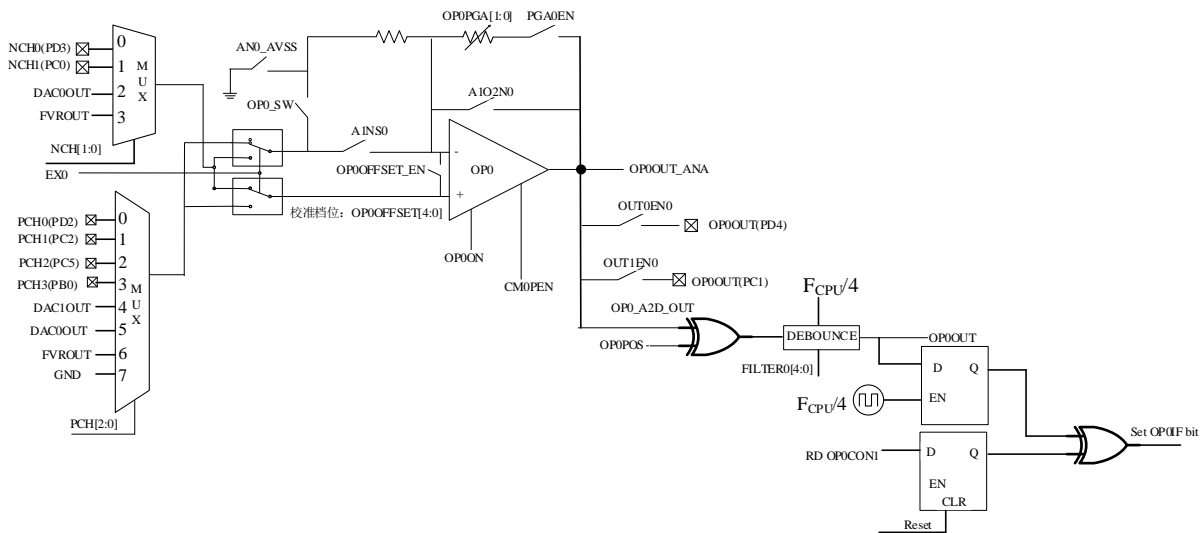
```
while (GO);
```

# 17 运放

使用 OP 模块之前需要配置 CLKCFG2[0],即 OPCLKEN=1;

AD18F06 提供一个轨到轨运放, 可以选取多个输入作为运放输入源。

## 17.1 运放 OP



OP0 电路图

注: OUT0EN0 和 OUT1EN0 使能时, PD4 引脚和 PC1 引脚与 OP0OUT\_ANA 的连接关系不受 OP0ON 控制; 可以直接通过 PD4 或者 PC1 外灌电压从 OP0OUT\_ANA 输出, 通过 ADC 采集;

OP0CON0(OP 控制寄存器)

地址: 0XF75

Bit	Name	Description	Attribute	Reset
7	OP0ON	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	R/W	0



		0: 断开		
4:2	PCH0[2:0]	运放 P 输入选择: 000: PCH0(PD2) 001: PCH1(PC2) 010:PCH2(PC5) 011: PCH3(PB0) 100:DAC1OUT 101: DAC0OUT 110: FVR0UT 111: GND	R/W	000
1:0	NCH0[1:0]	运放 N 输入选择 00: NCH0(PD3) 01: NCH1(PC0) 10:DAC0OUT 11: FVRCON	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。

### OP0CON1(OP 控制寄存器)

地址: 0XF74

Bit	Name	Description	Attribute	Reset
7	OP0OUT	运放输出信号 1: 运放输出为 0 0: 运放输出为 1	R/W	0
6	OUT1EN0	OP0 输出信号映射使能 IO(PD4) 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	A1O2N0	运放 buffer 模式 1: 负端与输出短接, 形成 BUFFER 0: 禁止	R/W	00
3	PGAEN0	内部使能放大 1: 使能 0: 禁止	R/W	0
2	OP0POS	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	OUT0EN0	OP0 输出信号映射使能 IO(PC1) 1: 允许使能输出到 IO 0: 禁止使能输出到 IO	R/W	0

### OP0CON2(OP 控制寄存器)

地址: 0XF73

Bit	Name	Description	Attribute	Reset
7:3	FILTER0[4:0]	OP 的滤波时间 $T = \text{FILTER0}[4:0] / F_{cpu}$ 有效滤除 OP0OUT 在 T 时间内的毛刺 FILTER0[4:0]==0x00 时没有滤波	R/W	0
2	OP0_A2D_OUT	运放数字输出使能	R	0
1:0	OP0PGA[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

## OP0CON3(OP 控制寄存器)

地址: 0XF52

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	OP0_SW	OP0 防扰开关 1: 允许使能 0: 禁止使能	R/W	0
5	OP0OFFSET_EN	OP0 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	OP0OFFSET[4:0]	OP0 输入失调电压校准模式选择位	R/W	00000

## 17.2 OP0 的输入失调校准

### 17.2.1. OP0OFFSET 校准:

步骤 1. 设置 OP0OFFSET\_EN=1 和 OP0ON=1, OP0\_SW=0, CMPMODE0=0, PGAEN0=0, A1O2N0=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 OP0OFFSET[4:0]=00000, 判断 OP0OUT 的值, 如果是低电平“0”设置 OP0OFFSET[4]=0; 如果是高电平“1”设置 OP0OFFSET[4]=1。

步骤 3. 设置 OP0OFFSET[3:0]=0000, 读取 OP0OUT 的值。

步骤 4. 将 OP0OFFSET[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 OP0OFFSET[3:0]值为 VOOS1, 转到执行步骤 5。

步骤 5. 设置 OP0OFFSET[3:0]= 1111, 读取 OP0OUT 的值。

步骤 6. 将 OP0OFFSET[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 OP0OFFSET[3:0]值为 VOOS2, 转到执行步骤 7。

步骤 7. 重新存储  $OP0OFFSET[3:0]=VOOS=(VOOS1+VOOS2)/2$ ，校准完成。若  $(VOOS1+VOOS2)/2$  非整数，则只取整数部分，完成后需关闭  $OP0OFFSET\_EN$ 。

### 17.3 OP 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xf75	OP0CON0	OP0ON	EX0	A1NS0	PCH0[2:0]		NCH0[1:0]			0000 0000	0000 0000
0xf74	OP0CON1	OP0OUT	OUTIEN0	AN_AVSS0	A1O2N0	PGAEN0	OP0POS	CMPMODE0	OUT0EN0	0000 0000	0000 0000
0xf73	OP0CON2	FILTER0[4:0]			OP0A2DEN		OP0PGA1	OP0PGA0		0000 0000	0000 0000
0xf72	OP0CON3	OP0_SW0	OP0_OFFSET0	OP0_OFFSET[4:0]						--00 0000	--00 0000
0xf71	OP0CON4							OP0CSEL[1:0]		0000 0000	0000 0000
0xfa4	IPR4			OP0IP						1111 1111	1111 1111
0xfa3	PIR4			OP0IF						0000 0000	0000 0000
0xfa2	PIE4			OP0IE						0000 0000	0000 0000

## 18 比较器(CMP0&CMP1&CMP2&CMP3)

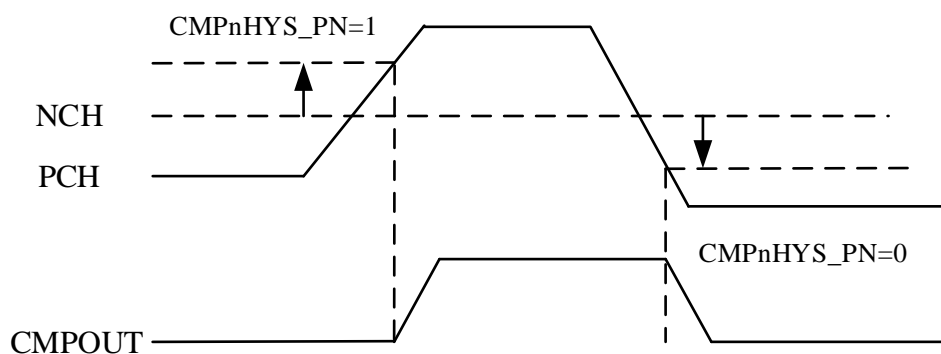
使用 CMP0、CMP1、CMP2、CMP3 模块之前，需要设置 CLKCFG2[1],即 CMPCLKEN=1; AD18F06 提供四个比较器，可以选取多个输入作为比较器输入源。

比较器的迟滞可以通过软件可编程比较器控制寄存器 CMPnCON2 设置。比较器迟滞可以使用比较器控制寄存器 CMPnCON2 中的 CMPnHYS\_SEL[1:0]进行设置，可以编程设置 0 mV、17 mV、32mV 或 55mv 的电压迟滞；寄存器 CMPnCON1 中的 CMPn\_HYSHEN 设置是否选择硬件控制迟滞使能，控制逻辑如下：

- 当 CMPn\_HYSHEN=1 时，此时寄存器 CMPnHYS\_PN 位将失效，迟滞正负端选择将由硬件自动控制，硬件控制逻辑为：CMPnPOS 配置为 0 时，当检测到 CMPnOUT 为 0 时，自动配置为上升沿迟滞，当检测到 CMPnOUT 为 1 时，自动配置为下降沿迟滞；CMPnPOS 配置为 1 时则相反，当检测到 CMPnOUT 为 0 时，自动配置为下降沿迟滞，当检测到 CMPnOUT 为 1 时，自动配置为上升沿迟滞。
- 当 CMPn\_HYSHEN=0 时，此时迟滞正负端使能将由寄存器 CMPnHYS\_PN 位控制；此时将 CMPnHYS\_PN 配置为 1，可使能上升沿迟滞，将 CMPnHYS\_PN 配置为 0，可使能下降沿迟滞，如下图所示。

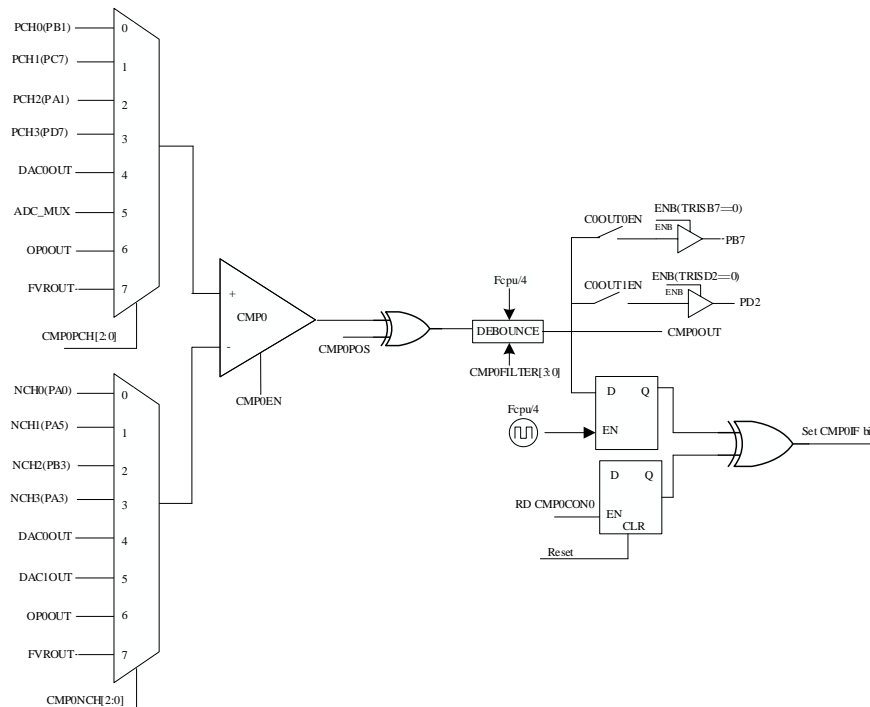
比较器中断可以在上升沿和下降沿输出转换时产生。当比较器 CMPnOUT 下降沿出现时，CMPnIF 标志设置为逻辑 1;当比较器 CMPnOUT 上升沿出现时，CMPnIF 标志设置为逻辑 1。一旦设置好，这些位将保持设置，直到被软件清除。

以上内容中的 n= (0、1、3)。



比较器迟滞图

## 18.1 比较器 CMP0



CMP0 电路图

### CMP0CON0(比较器控制寄存器)

地址：0XF7F

Bit	Name	Description	Attribute	Reset
7	CMP0EN	比较器的使能 1: 使能 CMP0 0: 禁止 CMP0	R/W	0
6	CMP0POS	比较器输出信号是否取反 1: 取反 0: 同向	R/W	0
5:3	CMP0PCH[2:0]	比较器的正端输入 000: C0P0(PB1) 001: C0P1(PC7) 010: C0P2(PA1) 011: C0P3(PD7) 100: DAC0OUT	R/W	000

		101: ADC_MUX 110: OP0OUT 111: FVR0UT		
2:0	CMP0NCH[2:0]	比较器的负端输入  000: C0N0(PA0) 001: C0N1(PA5) 010: C0N2(PB3) 011: C0N3(PA3)  100: DAC0OUT 101: DAC1OUT 110: OP0OUT 111: FVR0UT	R/W	000

### CMP0CON1(比较器控制寄存器)

地址: 0XF7E

Bit	Name	Description	Attribute	Reset
7	CMP0_HYSEN	CMP0 的硬件迟滞使能  1: 允许使能 (此时寄存器 CMP0HYS_PN 位将失效, 迟滞正负端将由硬件自动选择)  0: 禁止使能 (此时迟滞正负端使能将由寄存器 CMP0HYS_PN 位控制)	R/W	0
6:3	CMP0FILTER[3:0]	CMP0 的滤波时间  $T=(CMP0FILTER[3:0] \ll 4 + 0X0F) / F_{cpu}$ 有效滤除 CMP0OUT 在 T 时间内的毛刺 CMP0FILTER[3:0]=0x00 时没有滤波	R/W	0
2	C0OUT1EN	使能比较器 CMP0OUT 由 PD2 输出, 此时使能 PD2 IO 状态为输出  1: 使能 0: 禁止	R/W	0
1	C0OUT0EN	使能比较器 CMP0OUT 由 PB7 输出, 此时使能	R/W	0

		PB7 IO 状态为输出 1: 使能 0: 禁止		
0	CMP0OUT	CMP0 比较器的输出	R	0

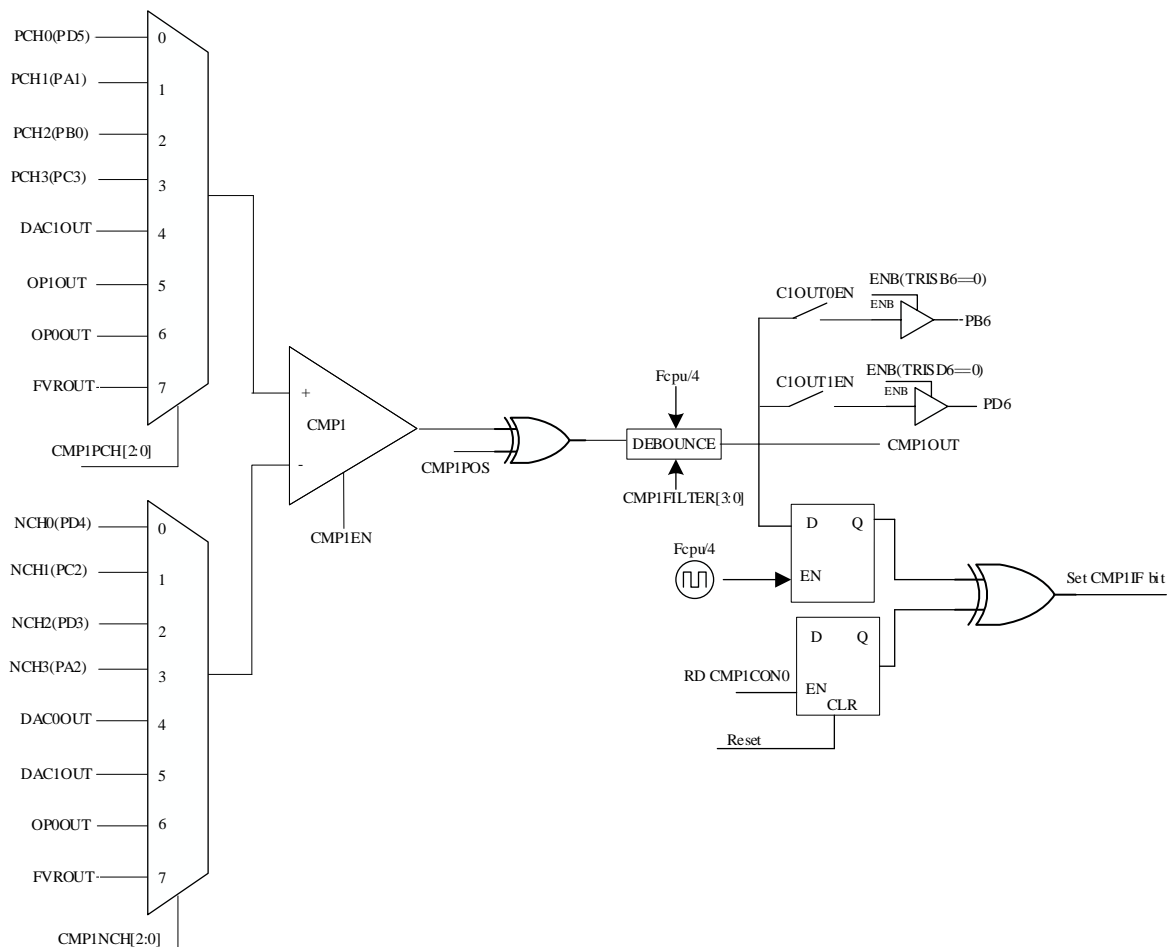
### CMP0CON2(比较器控制寄存器)

地址: 0XEEC

Bit	Name	Description	Attribute	Reset
7:6	CMP0HYS_SEL[1:0]	CMP0 迟滞电压选择信号 00: 0mV 01: 17mV 10: 32mV 11: 55mV	R/W	00
5	CMP0HYS_PN	CMP0 的上升/下降迟滞选择信号 1: 上升沿迟滞 0: 下降沿迟滞	R/W	0
4	CMP0_OFFSETEN	CMP0 的失调校准使能信号 0: 禁止使能 1: 允许使能	R/W	0
3:0	CMP0_TRIM[3:0]	CMP0 的失调校准值	R/W	0000



## 18.2 比较器 CMP1



CMP1 电路图

### CMP1CON0(比较器控制寄存器)

地址：0XF7D

Bit	Name	Description	Attribute	Reset
7	CMP1EN	比较器的使能 1: 使能 CMP1 0: 禁止 CMP1	R/W	0
6	CMP1POS	比较器输出信号是否取反 1: 取反 0: 同向	R/W	0
5:3	CMP1PCH[2:0]	比较器的正端输入 000: C1P0(PB1)	R/W	000

		001: C1P1(PA1) 010: C1P2(PC1) 011: C1P3(PC3) 100: DAC1OUT 101: ADC_MUX 110: OP0OUT 111: FVR0UT		
2:0	CMP1NCH[2:0]	比较器的负端输入 000: C1N0(PD4) 001: C1N1(PB2) 010: C1N2(PD3) 011: C1N3(PA2) 100: DAC0OUT 101: DAC1OUT 110: ADC_MUX 111: FVR0UT	R/W	000

### CMP1CON1(比较器控制寄存器)

地址: 0XF7C

Bit	Name	Description	Attribute	Reset
7	CMP1_HYSEN	CMP1 的硬件迟滞使能 1: 允许使能 (此时寄存器 CMP1HYS_PN 位将失效, 迟滞正负端将由硬件自动选择) 0: 禁止使能 (此时迟滞正负端使能将由寄存器 CMP1HYS_PN 位控制)	R/W	0
6:3	CMP1FILTER[3:0]	CMP1 的滤波时间 $T=(CMP1FILTER[3:0] \ll 4 + 0X0F) / F_{cpu}$ 有效滤除 CMP1OUT 在 T 时间内的毛刺 CMP1FILTER[3:0]=0x00 时没有滤波	R/W	0

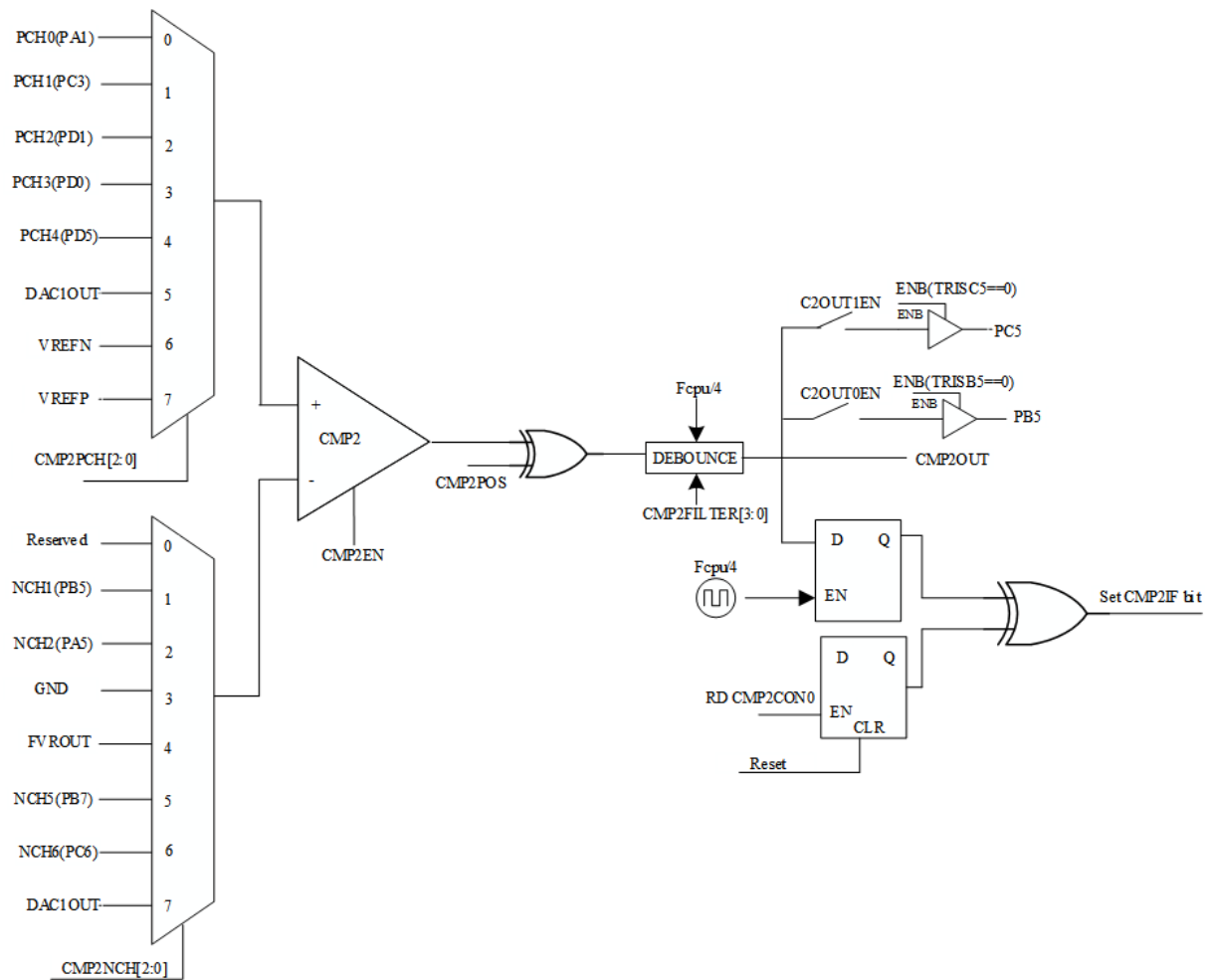
2	C1OUT1EN	使能比较器 CMP1OUT 由 PD6 输出，此时使能 PD6 IO 状态为输出 1: 使能 0: 禁止	R/W	0
1	C1OUT0EN	使能比较器 CMP1OUT 由 PB6 输出，此时使能 PB6 IO 状态为输出 1: 使能 0: 禁止	R/W	0
0	CMP1OUT	CMP1 比较器的输出	R	0

### CMP1CON2(比较器控制寄存器)

地址: 0XEED

Bit	Name	Description	Attribute	Reset
7:6	CMP1HYS_SEL[1:0]	CMP1 迟滞电压选择信号 00: 0mV 01: 17mV 10: 32mV 11: 55mV	R/W	00
5	CMP1HYS_PN	CMP1 的上升/下降迟滞选择信号 1: 上升沿迟滞 0: 下降沿迟滞	R/W	0
4	CMP1_OFFSETEN	CMP1 的失调校准使能信号 0: 禁止使能 1: 允许使能	R/W	0
3:0	CMP1_TRIM[3:0]	CMP1 的失调校准值	R/W	0000

## 18.3 比较器 CMP2



CMP2 电路图

### CMP2CON0(比较器控制寄存器)

地址：0XF7B

Bit	Name	Description	Attribute	Reset
7	CMP2EN	比较器的使能 1: 使能 CMP2 0: 禁止 CMP2	R/W	0
6	CMP2POS	比较器输出信号是否取反 1: 取反 0: 同向	R/W	0
5:3	CMP2PCH[2:0]	比较器的正端输入	R/W	000

		000: PCH0(PA1) 001: PCH1(PC3) 010: PCH2(PD1) 011: PCH3(PD0) 100: PCH4(PD5) 101: DAC1OUT 110: VREFN 111: VREFP		
2:0	CMP2NCH[2:0]	比较器的负端输入 000: Reserved 001: NCH1(PB5) 010: NCH2(PA5) 011: GND 100: FVROUT 101: NCH5(PB7) 110: NCH6(PC6) 111: DAC1OUT	R/W	000

### CMP2CON1(比较器控制寄存器)

地址: 0XF7A

Bit	Name	Description	Attribute	Reset
7	CMP2_HYSEN	CMP2 的迟滞使能 1: 允许使能      0: 禁止使能	R/W	0
6:3	CMP2FILTER[3:0]	CMP2 的滤波时间 $T=(CMP2FILTER[3:0] \ll 4 + 0X0F) / F_{cpu}$ 有效滤除 CMP2OUT 在 T 时间内的毛刺 CMP2FILTER[3:0]=0x00 时没有滤波	R/W	0
2	C2OUT1EN	➤ 使能比较器 CMP2OUT 由 PC5 输出, 此时使能 PC5 IO 状态为输出	R/W	0

		1: 使能 0: 禁止		
1	C2OUT0EN	使能比较器 CMP2OUT 由 PB5 输出, 此时使 能 PB5 IO 状态为输出  1: 使能 0: 禁止	R/W	0
0	CMP2OUT	CMP2 比较器的输出	R	0

### CMP2CON2(比较器控制寄存器)

地址: 0XF79

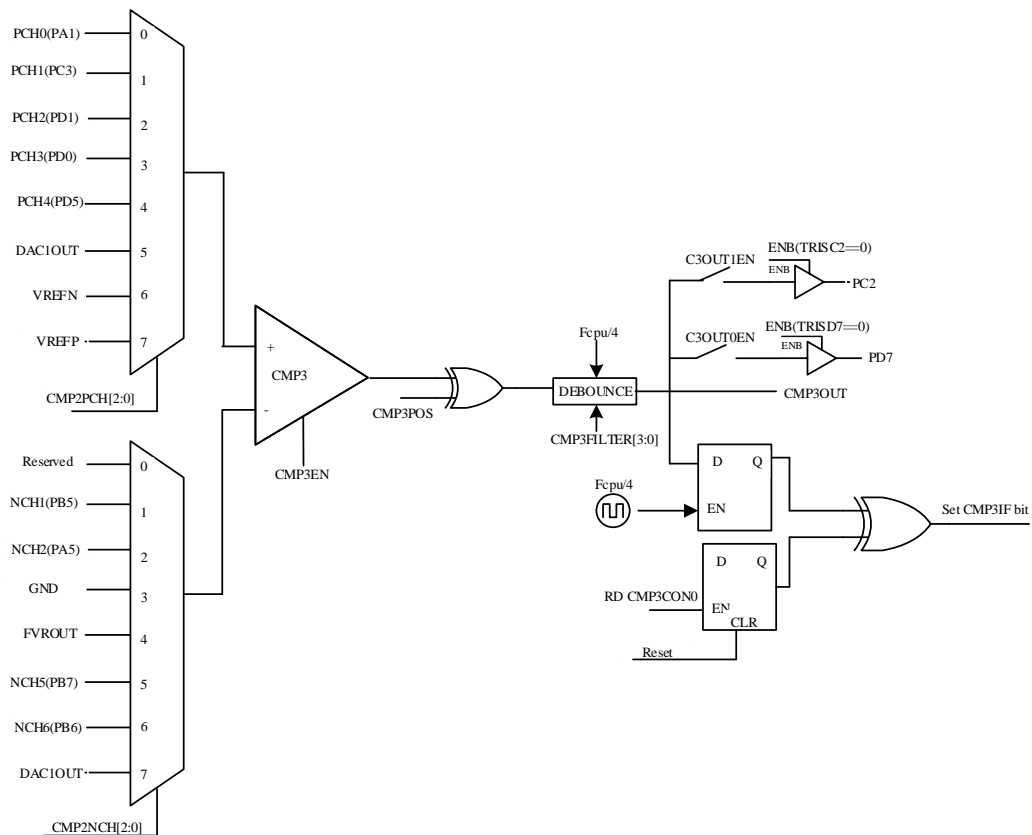
Bit	Name	Description	Attribute	Reset
7:2	Reserved			
1	PCH0_SEL	1: 作为 CMP2 时此位必须置 1	R/W	0
0	C2DEN	CMP2OUT 数据选择位  0: 滤波后信号 1: 模拟输出信号	R/W	0

### CMP2CON3(比较器控制寄存器)

地址: 0XF78

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	CMRSEL[5:0]	CMP2OUT 的输出电流控制	R/W	0

## 18.4 比较器 CMP3



### CMP3CON0(比较器控制寄存器)

地址: 0XF50

Bit	Name	Description	Attribute	Reset
7	CMP3EN	比较器的使能 1: 使能 CMP3 0: 禁止 CMP3	R/W	0
6	CMP3POS	比较器输出信号是否取反 1: 取反 0: 同向	R/W	0
5:3	CMP3PCH[2:0]	比较器的正端输入 000: C3P0(PA1) 001: C3P1(PC3) 010: C3P2(PC0) 011: C3P3(PB1)	R/W	000

		100: DAC1OUT 101: ADC_MUX 110: OP0OUT 111: FVROUT		
2:0	CMP3NCH[2:0]	比较器的负端输入 000: C3N0(PB4) 001: C3N1(PA2) 010: C3N2(PB0) 011: C3N3(PC4) 100: DAC0OUT 101: DAC1OUT 110: ADC_MUX 111: FVROUT	R/W	000

### CMP3CON1(比较器控制寄存器)

地址: 0XF4F

Bit	Name	Description	Attribute	Reset
7	CMP3_HYSEN	CMP3 的迟滞使能 1: 允许使能      0: 禁止使能	R/W	0
6:3	CMP3FILTER[3:0]	CMP3 的滤波时间 $T=(CMP3FILTER[3:0] \ll 4 + 0X0F) / F_{cpu}$ 有效滤除 CMP3OUT 在 T 时间内的毛刺 CMP3FILTER[3:0]==0x00 时没有滤波	R/W	0
2	C3OUT1EN	使能比较器 CMP3OUT 由 PC2 输出, 此时使 能 PC2 IO 状态为输出 1: 使能 0: 禁止	R/W	0
1	C3OUT0EN	使能比较器 CMP3OUT 由 PD7 输出, 此时使 能 PD7 IO 状态为输出 1: 使能 0: 禁止	R/W	0



0	CMP3OUT	CMP3 比较器的输出	R	0
---	---------	-------------	---	---

### CMP3CON2(比较器控制寄存器)

地址: 0XEAA

Bit	Name	Description	Attribute	Reset
7:6	CMP3HYS_SEL[1:0]	CMP3 迟滞电压选择信号  00: 0mV  01: 17mV  10: 32mV  11: 55mV  注: 此值为实际翻转	R/W	00
5	CMP3HYS_PN	CMP3 的上升/下降迟滞选择信号  1: 上升沿迟滞  0: 下降沿迟滞	R/W	0
4	CMP3_OFFSETEN	CMP3 的失调校准使能信号  0: 禁止使能 1: 允许使能	R/W	0
3:0	CMP3_TRIM[3:0]	CMP3 的失调校准值	R/W	0000

## 18.5 CMP 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	por & bor reset value	other reset value
0xf7f	CMP0CON0	CMP0EN	CMP0POS	CMP0PCH[2:0]			CMP0NCH[2:0]			0000 0000	0000 0000
0xf7e	CMP0CON1	CMP0HYSHEN	CMP0FILTER[3:0]			C0OUT1EN	C0OUT0EN	CMP0OUT	0000 0000	0000 0000	0000 0000
0xf7d	CMP1CON0	CMP1EN	CMP1POS	CMP1PCH[2:0]			CMP1NCH[2:0]			0000 0000	0000 0000
0xf7c	CMP1CON1	CMP1HYSHEN	CMP1FILTER[3:0]			C1OUT1EN	C1OUT0EN	CMP1OUT	0000 0000	0000 0000	0000 0000
0xf7b	CMP2CON0	CMP2EN	CMP2POS	CMP2PCH[2:0]			CMP2NCH[2:0]			0000 0000	0000 0000
0xf7a	CMP2CON1	CMP2HYSHEN	CMP2FILTER[3:0]			C2OUT1EN	C2OUT0EN	CMP2OUT	0000 0000	0000 0000	0000 0000
0xf79	CMP2CON2	VP1[2:0]		VP0[2:0]			PCH0_SEL	C2DEN	1111 0000	1111 0000	1111 0000
0xf78	CMP2CON3	CMRSEL[5:0]							-00 0000	-00 0000	
0xfa4	IPR4			CMP3IP	CMP2IP	CMP1IP	CMP0IP			-111 1111	-111 1111
0xfa3	PIR4			CMP3IF	CMP2IF	CMP1IF	CMP0IF			-000 0000	-000 0000
0xfa2	PIE4			CMP3IE	CMP2IE	CMP1IE	CMP0IE			-000 0000	-000 0000
0xec	CMP0CON2	CMP0HYS_SEL[1:0]	CMP0HYS_PN	CMP0_OFFSETEN	CMP0_TRIM[3:0]					00-0 0000	00-0 0000
0xeb	CMP1CON2	CMP1HYS_SEL[1:0]	CMP1HYS_PN	CMP1_OFFSETEN	CMP1_TRIM[3:0]					00-0 0000	00-0 0000
0xea	CMP3CON2	CMP3HYS_SEL[1:0]	CMP3HYS_PN	CMP3_OFFSETEN	CMP3_TRIM[3:0]					00-0 0000	00-0 0000

## 19 LCD 驱动和 LED 驱动

### 19.1 LCD 驱动

#### 19.1.1. 概述

对于设计中带有 LCD 功能的大批量应用，选择定制而非较昂贵的基于字符的显示方式可以有效地降低成本。然而，驱动此类定制的显示器需要振幅及时间可变的 COM 和 SEG 信号，且需要很多特殊的考虑以正确地操作 LCD。此单片机有内部 LCD 信号产生电路，可以自动地产生时间与振幅可变的信号直接驱动 LCD，与用户 LCD 的接口连接也相当容易。设置 I/O 口方向和数据寄存器，其中 LCD 功能设置相应管脚为输入态，LED 功能设置相应管脚为输出态并输出低电平。

驱动数目	占空比	偏压	COMSEL[2:0]	波形类型
18*8	1/8	1/3	000	A 或 B
19*7	1/7	1/3	001	A 或 B
19*6	1/6	1/3	010	A 或 B
19*5	1/5	1/3	011	A 或 B
19*4	1/4	1/3	100	A 或 B
19*3	1/3	1/3	101	A 或 B
19*3	1/3	1/2	110	A
19*4	1/4	1/2	111	A

特性：

- 支持 LCD R 型 1/2 BIAS 和 1/3 BIAS；
- 支持 LCD 8Com\*18Seg，7Com\*19Seg，6Com\*19Seg，5Com\*19Seg，4Com\*19Seg，3Com\*19Seg，
- 支持 LCD/LED 显示频率分频 32KHZ 的 1/1、1/2、1/3、1/4、1/5 至 1/32 可调，最低 3.90625Hz，最高 666.7Hz；
- LCD 1/3 BIAS 占空比为 1/8、1/7、1/6、1/5、1/4、1/3；
- LCD 1/2 BIAS 占空比为 1/4、1/3
- LCD/LED 存储阵列，MCU 直接方式访问；

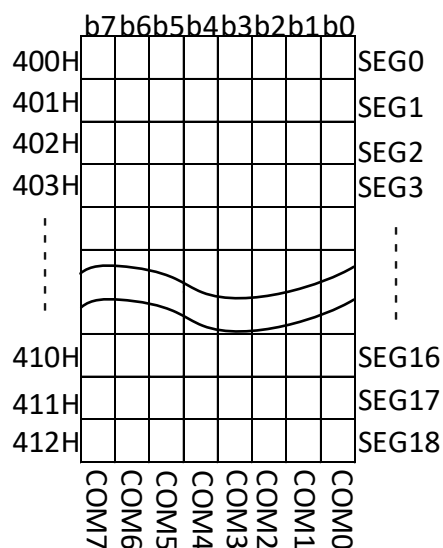
- 支持 LCD/LED 功能，可选 LCD 或 LED，不能同时使用。
- LED 仅支持共阴驱动

### 19.1.2. LCD/LED 显示数据存储

数据存储中有一部分区域是专门为 LCD 的显示数据而保留，即 LCD 显示数据存储。单片机内部显示驱动电路会自动读取任何写入此处的数据并据此产生 LCD 驱动信号。因此任何写入 LCD 存储器的数据，会立即映射到连接单片机的 LCD 显示器上。应注意的是，未引出的 LCD SEG 引脚所对应的 LCD 显示存储数据位不可用。

该单片机为 LCD 显示提供一个嵌入式数据存储区域。LCD 存储器的地址与通用数据存储重叠，这个区域在 Sector 4。当要存取 LCD 存储器时，首先要将 XRAM 的值设为“0400H”来选择对 LCDRAM 操作。此后，用户可以通过“0400H+N(N≤18)”来对存储器进行操作。就可以直接对显示存储器进行读或者写的操作了。

下方的 LCD 存储器映射图显示了内部 LCD 存储如何映射到单片机显示的 SEG 和 COM 引脚。应注意，未使用的 LCD RAM 可作为通用数据存储使用。



注：COM7 和 SEG6 复用 PD7，故选中 COM7 上图中无 SEG6 行，反之选中 SEG6 上图中无 COM7 列。

LCD 存储器映射图

### 19.1.3. LCD 时钟源

LCD 时钟来自内部 LIRC 32k 振荡器。该方法可产生理想的 LCD 时钟，以获得更好的 LCD

显示效果。

#### 19.1.4. LCD 寄存器

LCD 控制寄存器 LCDCON0,LCDCON1,COMEN,SEGEN0,SEGEN1 与 SEGEN2 用于设置 LCD 驱动器的各种特性。

LCDCON0 寄存器中的比特位用于整个 LCD 的使能/禁止使能控制，工作频率设置与偏压电流选择。LCDCON1 寄存器中的比特位用于控制 LCD 波形类型，LCD 清屏，设置 COM 口工作数与 LCD 电压源。若单片机处于休眠模式，则显示器将始终不使能。

##### LCDCON0(LCD 控制寄存器)

地址：0XF6E

Bit	Name	Description	Attribute	Reset
7:3	LCD_DIV[4:0]	LCD/LED 工作频率分频  00000: 1/1  00001: 1/2  .....  11110: 1/31  11111: 1/32  LCD 工作频率:  B 型波: $F_{32k}/(DIV[4:0]+1)/16/\text{工作 COM 数}$ A 型波: $F_{32k}/(DIV[4:0]+1)/16/(\text{工作 COM 数} * 2)$  LED 工作频率:  $F_{32k}/(DIV[4:0]+1)/16/\text{工作 COM 数}$	R/W	0
2	PLCDEN	VPLCD 的电压选择  0: VDD  1: PB7 引脚输入电压	R/W	0
1	LEDEN	LED 使能控制  1: 允许使能  0: 禁止使能	R/W	0
0	LCDEN	LCD 使能控制	R/W	0

		1: 允许使能		
		0: 禁止使能		

**LCDCON1 (LCD 控制寄存器)**

地址: 0XF6D

Bit	Name	Description	Attribute	Reset
7:6	LCDIS[1:0]	R 型 LCD 偏压电流选择 ( $V_{DD,1/3bias}$ $1/2bias$ ) 00:10 $\mu$ A 01:50 $\mu$ A 10:100 $\mu$ A 11:200 $\mu$ A 注: 此处为 PLCD 电流 ( $V_{DD}$ 和 PB7 电流)	R/W	00
5:3	COMSEL[2:0]	LCD/LED COM 端口和 BIAS 选择  LCD: 000:1/3 占空比    bias 1/3 001:1/4 占空比    bias 1/3 010:1/5 占空比    bias 1/3 011:1/6 占空比    bias 1/3 100:1/7 占空比    bias 1/3 101:1/8 占空比    bias 1/3 110:1/3 占空比    bias 1/2 111:1/4 占空比    bias 1/2  LED: 000:1/1 占空比 001:1/2 占空比 010:1/3 占空比 011:1/4 占空比 100:1/5 占空比 101:1/6 占空比	R/W	000

		110:1/7 占空比		
		111:1/8 占空比		
2:0	Reserved			

### COMEN (LCD 控制寄存器)

地址: 0XF6C

Bit	Name	Description	Attribute	Reset
7:0	COMxEN	COM 口功能设置; 0: 对应 COMx 口为普通 I/O 口(x=0-7); 1: 对应 COMx 口为 LCD/LED 功能的 COM 口(x=0-7)。 注: x=0-7 对应 0-7bit。	R/W	0x00

### SEGEN0 (LCD 控制寄存器)

地址: 0XF6B

Bit	Name	Description	Attribute	Reset
7:0	SEGxEN	SEGx 口功能设置; 0: 对应 SEGx 口为普通 I/O 口(x=0-7); 1: 对应 SEGx 口为 LCD/LED 功能的 SEGx 口 (x=0-7)。 注: x=0-7 对应 0-7bit。	R/W	0x00

### SEGEN1 (LCD 控制寄存器)

地址: 0XF6A

Bit	Name	Description	Attribute	Reset
7:0	SEGxEN	SEGx 口功能设置; 0: 对应 SEGx 口为普通 I/O 口(x=8-15); 1: 对应 SEGx 口为 LCD/LED 功能的 SEGx 口 (x=8-15)。 注: x=8-15 对应 0-7bit。	R/W	0x00

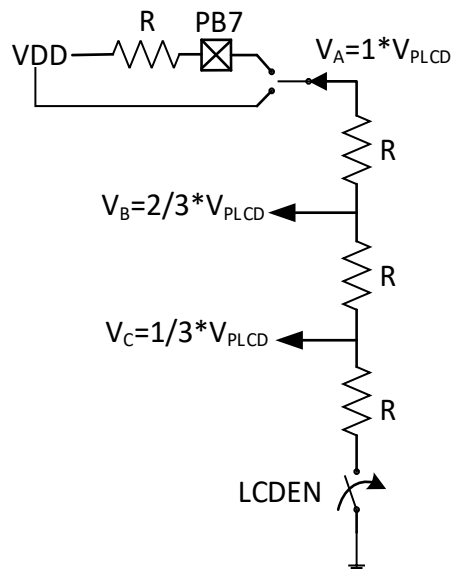
## SEGEN2 (LCD 控制寄存器)

地址: 0XF69

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5	SEGOFF	SEG 口输出模式选择(LCD 模式); 1: 当 LCD 刷新完一帧后, SEG 口输出为 LCDDATA 中的数据; 0: SEG 口立即输出全为 0, 清除 LCD 显示屏; SEG 口输出模式选择(LED 模式); 1: 当 LED 刷新完一帧后, SEG 口输出为 LCDDATA 中的数据; 0: SEG 口立即输出全为 0, 清除 LED 显示屏;	R/W	1
4	TYPE	LCD 波形类型选择 1: B 型    0: A 型	R/W	0
3	Reserved			
2:0	SEGxEN	SEGx 口功能设置; 0: 对应 SEGx 口为普通 I/O 口(x=16-18); 1: 对应 SEGx 口为 LCD/LED 功能的 SEGx 口(x=16-18)。 注: x=16-18 对应 0-2bit。	R/W	0

### 19.1.5. LCD 电压源和偏压

对于 1/3 偏压的结构， $V_{PLCD} = V_{DD}$ （ $PLCDEN == 0$ ）或者 PB7（ $PLCDEN == 1$ ）的电压



LCD 偏压配置

### 19.1.6. LCD 的复位状态

LCD 具有内部复位功能，通过对 LCDCON0 寄存器中 LCDEN 位状态与休眠功能执行或运算确定。清零 LCDEN 位将复位 LCD 功能。当单片机进入休眠模式，即使 LCDEN 位被置位以使能 LCD 驱动功能，LCD 仍将被复位。

当 LCDEN 被置位以使能 LCD 驱动功能，若此时发生单片机复位，则 LCD 将被复位，复位期间 COM 与 SEG 输出将处于浮空状态。复位操作将持续一个  $t_{RSTD} + t_{SST}$  时间。 $t_{RSTD} + t_{SST}$  的详细信息参考系统上电时间特性表。

MCU 复位	休眠模式	LCDEN	LCD 复位	COM 与 SEG 电压电平
否	OFF	1	否	正常运行
否	OFF	0	是	低
否	ON	-	是	低
是	-	-	是	浮空

注：1：此处的单片机复位条件不包括空闲或休眠模式下的看门狗定时器溢出复位。

2：“-”：无关。

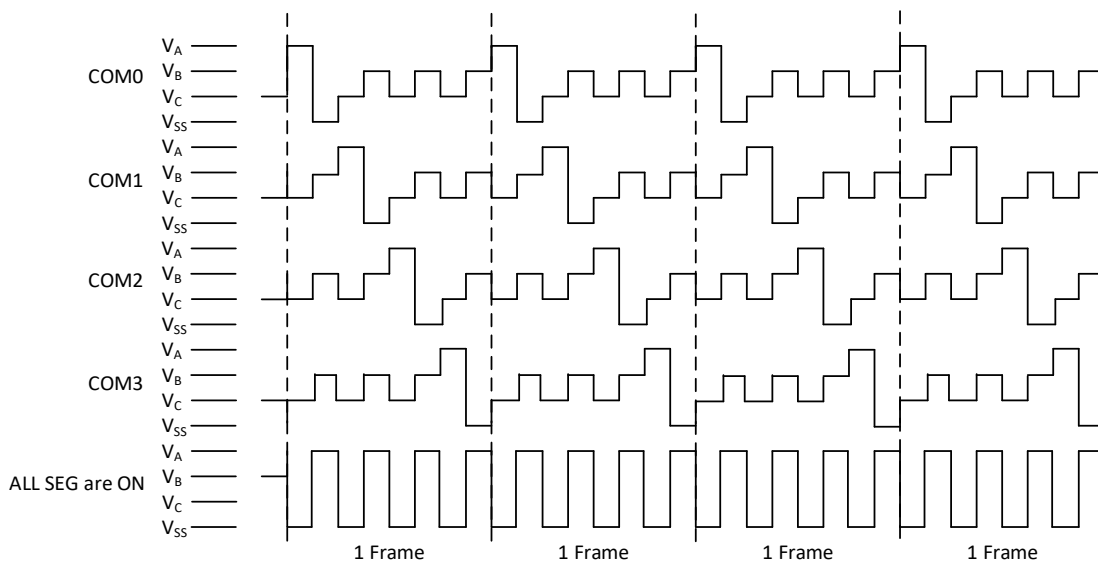


### 19.1.7. LCD 驱动输出

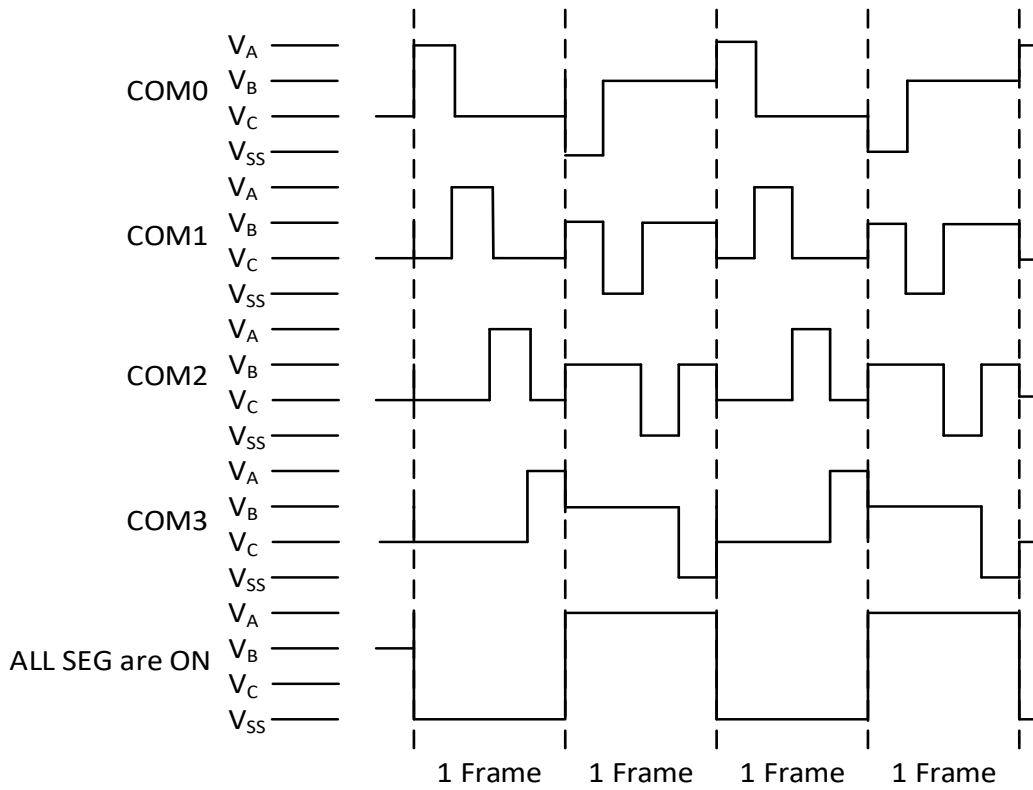
LCD 驱动的输出结构为  $18 \times 8$  或  $19 \times 7$ 。LCD 驱动偏压类型为 R 型，且偏压值为  $1/2$  和  $1/3$ 。

由于 LCD 基本性质的缘故，它们的像素点只能加上 AC 电压，如果加上 DC 电压，将会引起永久性的损害。因此 LCD 显示器的对比度由提供到每个像素的实际 RMS 电压控制，这个值等于 COM 引脚上的电压值减去 SEG 引脚上电压值的结果的 RMS 值。RMS 电压必须大于 LCD 的饱和电压，以便能打开像素点，但同时也要小于阈值电压，以便能关闭像素点。

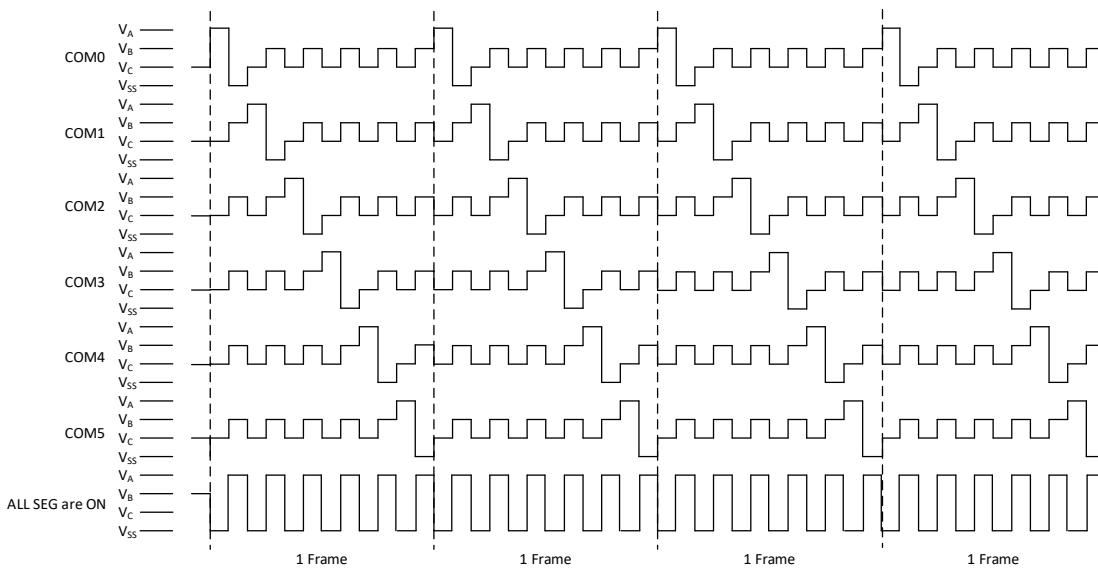
因为要将 DC 电压限制为 0 且以最少的连接数来控制尽可能多的像素点，因此需要产生时间振幅可变的信号供给 LCD 使用。这些时间与振幅都可变的信号由单片机内的 LCD 驱动电路自动产生。占空比决定使用 COM 口的个数，也称为底板或 COM。例如，占空比为  $1/4$ ，表示 COM 的数目为 4，因此该值定义了每个 LCD 信号帧内的时间片数。单片机提供两种类型的信号即 A 型和 B 型，通过寄存器 SEGEN3 中的 TYPE 位加以选择。A 型提供较低频率的信号，然而，较低的频率可能引起闪烁，从而影响显示的清晰度。



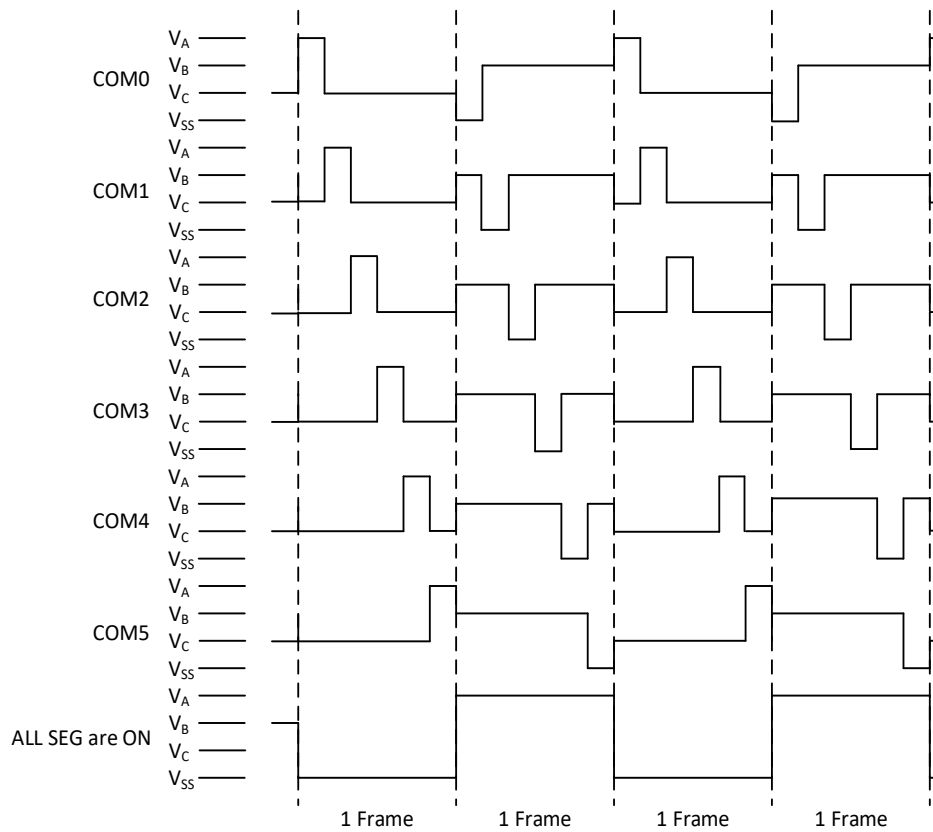
LCD 驱动输出-A 型,  $1/4$ Duty,  $1/3$ Bias



LCD 驱动输出-B 型, 1/4 Duty, 1/3Bias



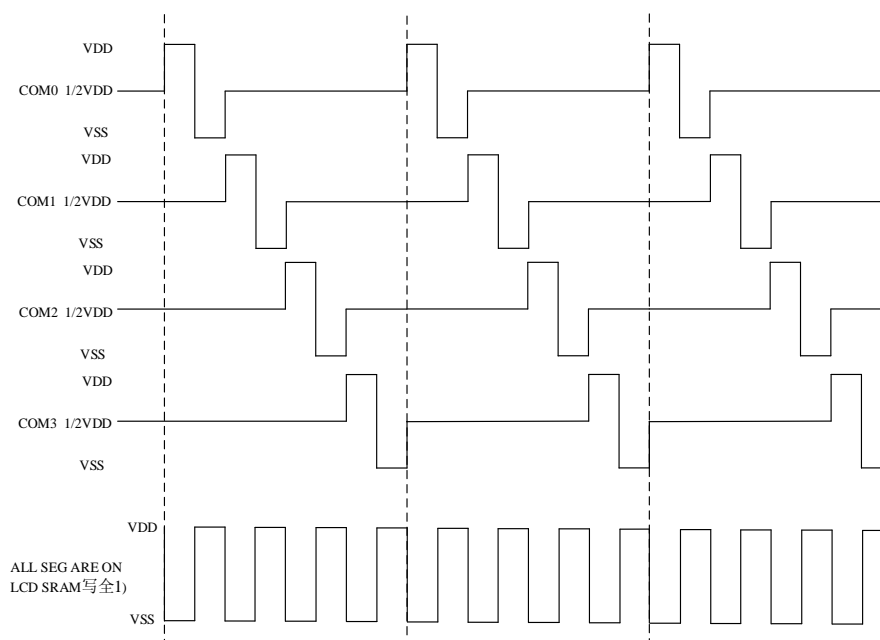
LCD 驱动输出-A 型, 1/6 Duty, 1/3Bias



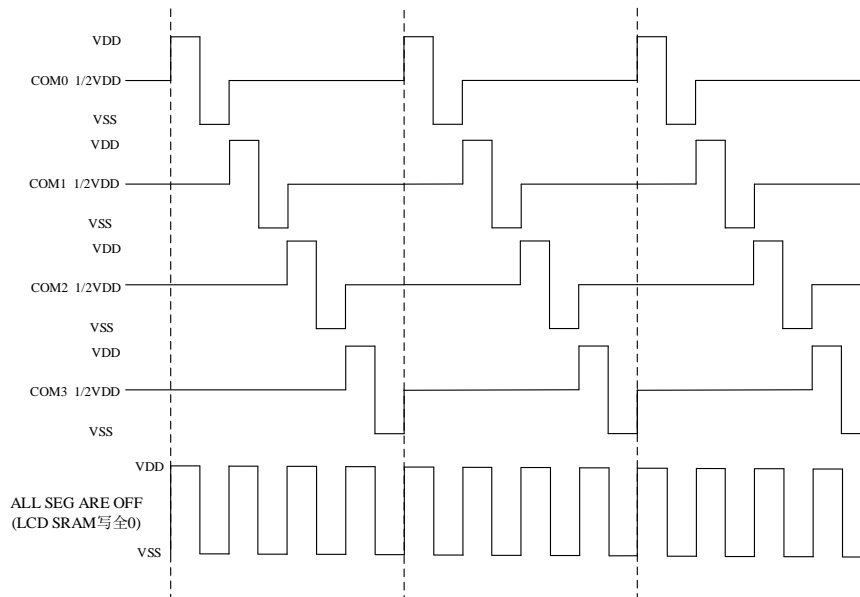
LCD 驱动输出-B 型, 1/6 Duty, 1/3Bias

注：以上 1/3Bias 波形图中，当 ALL SEG ARE OFF 时，ALL SEG 信号取反（同下 1/2Bias 波形图）。

下图为 1/2Bias 波形图。



LCD 驱动输出-A 型, 1/4Duty, 1/2Bias (ALL SEG ARE ON)



LCD 驱动输出-A 型, 1/4Duty, 1/2Bias (ALL SEG ARE OFF)

### 19.1.8. 编程注意事项

LCD 编程时要注意几点, 其中之一就是在单片机上电后, 要保证 LCD 存储器正确地初始化。与通用数据存储器一样, 在上电后, LCD 存储器的内容是未知的。由于 LCD 存储器的内容会映射到实际的 LCD 显示, 所以在上电后, 为获得正确的显示图形, 初始化此存储器内容是非常重要的。

在实际应用中, 必须要考虑 LCD 的实际容性负载。对于单片机来说, LCD 的像素点一般可以看作电容性的负载, 要确保所连接的像素点不能过多。这点对可以连接多个 LCD 像素点的 COM 口来说尤为重要。下方图示描述 LCD 的等效电路。

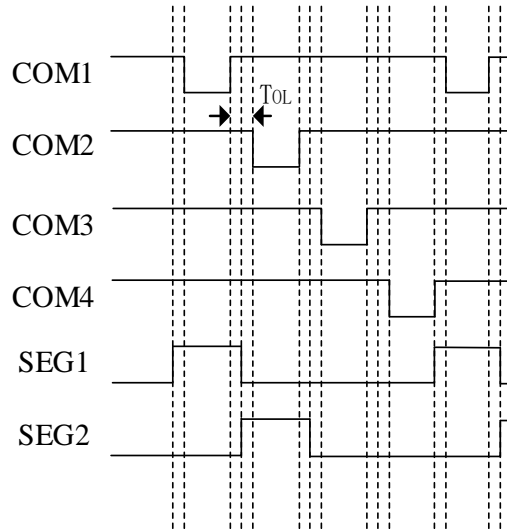
另外还有一个要注意的就是当单片机进入空闲模式或休眠模式后所发生的变化, 可以控制 LCDIS[1:0] 选择 LCD 控制电流, 当低功耗模式时, 进入超低功耗显示, 同样, 也可以关闭 LCD 进入更低功耗运行。

要注意当上电复位后, LCDEN 位会被清零, 显示功能关闭。

## 19.2 LED 驱动器

LED 最大支持 18Com\*8 Seg 共 144 个点的驱动, 与 LCD 共用存储器和引脚, 共 26 个输出, 可通过调节 COM 的占空比来调节 LED 亮度, 仅支持共阴驱动, 每个 SEG 管脚最大提供 12mA/20mA 电流。

LED 波形图(LED\_MOD=0, 1/4 duty):



注：TOL 为 LED COMMON 信号间的重叠时间， $T_{OL}=62.5\mu s$

### 19.3 LCD/LED 驱动器相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	reset value	value
0xffe	LCDCON0	DIV[4:0]					PLCDEN	LEDEN	LCDEN	0000 0000	0000 0000
0xffd	LCDCON1	LCDIS[1:0]		COMSEL[2:0]						0000 0000	0000 0000
0xffc	COMEN	COM7EN	COM6EN	COM5EN	COM4EN	COM3EN	COM2EN	COM1EN	COM0EN	0000 0000	0000 0000
0xffb	SEGEN0	SEG7EN	SEG6EN	SEG5EN	SEG4EN	SEG3EN	SEG2EN	SEG1EN	SEG0EN	0000 0000	0000 0000
0xffa	SEGEN1	SEG15EN	SEG14EN	SEG13EN	SEG12EN	SEG11EN	SEG10EN	SEG9EN	SEG8EN	0000 0000	0000 0000
0xff9	SEGEN2			SEGOFF	TYPE		SEG18EN	SEG17EN	SEG16EN	--10 -000	--10 -000

## 20 8X8 硬件乘法器和 16/16 硬件除法器

AD18F06 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 AD18F06 器件。

AD18F06 包含一个 16/16 硬件除法器。该除法器可执行无符号运算并产生一个 16 位的商和余数，该结果存储在—对商寄存器余数寄存器中。该除法器器执行的运算不会影响状态寄存器中的任何标志。

通过硬件执行除法运算只需要 9 个指令周期。硬件除法器具有更高的计算吞吐量并减少了除法算法的代码长度，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 AD18F06 器件。

### DIVIDENDH 寄存器

地址：0XF68

Bit	Name	Description	Attribute	Reset
7:0	DIVDENDH[15:8]	被除数高 8 位寄存器	R/W	0X00

### DIVIDENDL 寄存器

地址：0XF67

Bit	Name	Description	Attribute	Reset
7:0	DIVDENDL[7:0]	被除数低 8 位寄存器	R/W	0X00

### DIVISORH 寄存器

地址：0XF66

Bit	Name	Description	Attribute	Reset
7:0	DIVISORH[15:8]	除数高 8 位寄存器	R/W	0XFF

### DIVISORL 寄存器

地址：0XF65

Bit	Name	Description	Attribute	Reset
7:0	DIVISORL[7:0]	除数低 8 位寄存器	R/W	0XFF

### QUOTIENTH 寄存器

地址：0XF64

Bit	Name	Description	Attribute	Reset
7:0	QUOTIENTH[15:8]	商的高 8 位寄存器	R	0X00

### QUOTIENTL 寄存器

地址：0XF63

Bit	Name	Description	Attribute	Reset
7:0	QUOTIENTL[7:0]	商的低 8 位寄存器	R	0X00

### REMAINDERH 寄存器

地址：0XF62

Bit	Name	Description	Attribute	Reset
7:0	REMAINDERH[15:8]	余数高 8 位寄存器	R	0X00

### REMAINDERL 寄存器

地址：0XF61

Bit	Name	Description	Attribute	Reset
7:0	REMAINDERL[7:0]	余数低 8 位寄存器	R	0X00

### DIVCON 寄存器

地址：0XF60

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3	DIV0	除数为 0 标志 1: 为 0      0: 不为 0	R	0
2	COMPLETE	除法完成标志 1: 完成                      0: 运行中	R	0
1	BUSY	除法运算忙标志	R	0

		1: 忙 0: 完成		
0	START	除法运算启动控制位 1: 启动, 硬件自动清0 0: 不启动	R/W	0

### PRODH 寄存器

地址: 0XFF4

Bit	Name	Description	Attribute	Reset
7:0	PRODH[7:0]	乘积寄存器高字节	R/W	0X00

### PRODL 寄存器

地址: 0XFF3

Bit	Name	Description	Attribute	Reset
7:0	PRODL[7:0]	乘积寄存器低字节	R/W	0X00

## 20.1 相关寄存器定义

address	register name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	pur & our reset value
0xff4	PRODH				Product Register High Byte					0000 0000
0xff3	PRODL				Product Register Low Byte					0000 0000
0xf68	DIVIDENDH				DIVIDENDH[7:0]					0000 0000
0xf67	DIVIDENDL				DIVIDENDL[7:0]					0000 0000
0xf66	DIVISORH				DIVISORH[7:0]					1111 1111
0xf65	DIVISORL				DIVISORL[7:0]					1111 1111
0xf64	QUOTIENTH				QUOTIENTH[7:0]					0000 0000
0xf63	QUOTIENTL				QUOTIENTL[7:0]					0000 0000
0xf62	REMAINDERH				REMAINDERH[7:0]					0000 0000
0xf61	REMAINDERL				REMAINDERL[7:0]					0000 0000
0xf60	DIVCON					DIV0	COMPLETE	BUSY	START	---- 0000



## 21 电气参数

### 直流特性

符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件 (常温 25°C)				
VDD	工作电压	—	Fsys=16MHz, 2T/4T	2.4		5.5	V
		—	Fsys=32MHz, 2T/4T	3.0		5.5	V
IDD1	工作电流	5.0V	Fsys = 16MHz, 4T, 高频模式, WDT 禁止, 无负载	—	3.3	—	mA
IDD2	工作电流	5.0V	IDLE 模式下电流	—	1.70	—	mA
IDD3	工作电流	5.0V	PWSAVE 模式下电流	—	30.0	—	uA
IDD4	工作电流	5.0V	DEEPPWSAVE 模式下电流	—	16.0	—	uA
IDD5	工作电流	5.0V	PWOFF 模式下电流	—	1.0	—	μ A
VIL1	低电平输入电压	5.0V	施密特输入口	VSS		0.2*VD D	V
VIH1	高电平输入电压	5.0V	施密特输入口	0.4*VDD		VDD	V
VIL2	低电平输入电压	5.0V	施密特输入口	VSS		0.3*VD D	V
VIH2	高电平输入电压	5.0V	施密特输入口	0.7*VDD		VDD	V
RPH1	内部上拉电阻	5.0V	PA0~PA5, PB0~PB7, PC0~PC7, PD0~PD7	—	30	—	kΩ
RPD1	内部下拉电阻	5.0V	PA0~PA5, PB0~PB7, PC0~PC7, PD0~PD7	—	30	—	kΩ
RPH2	内部上拉电阻	5.0V	PA0~PA5, PB0~PB7, PC0~PC7, PD0~PD7	—	190	—	kΩ
RPD2	内部下拉电阻	5.0V	PA0~PA5, PB0~PB7, PC0~PC7, PD0~PD7	—	300	—	kΩ

I <sub>OL1</sub>	输出口灌电流	5.0V	VOL=0.1*VDD, PA0~PA5, PB0~PB1	—	20	—	mA
I <sub>OH1</sub>	低档输出口拉电流	5.0V	VOH=0.9*VDD, PA0~PA5, PB0~PB1	—	3	—	mA
	中档输出口拉电流	5.0V	VOL=0.9*VDD, PA0~PA5, PB0~PB1	—	6	—	mA
	高档输出口拉电流	5.0V	VOL=0.9*VDD, PA0~PA5, PB0~PB1	—	20	—	mA
I <sub>OL2</sub>	输出口灌电流	5.0V	VOL=0.1*VDD, PB2~PB7, PC0~PC7	—	20	—	mA
I <sub>OH2</sub>	低档输出口拉电流	5.0V	VOH=0.9*VDD, PB2~PB7, PC0~PC7	—	3	—	mA
	中档输出口拉电流	5.0V	VOH=0.9*VDD, PB2~PB7, PC0~PC7	—	6	—	mA
	高档输出口拉电流	5.0V	VOH=0.9*VDD, PB2~PB7, PC0~PC7	—	12	—	mA
I <sub>OL3</sub>	低档输出口灌电流	5.0V	VOL=0.1*VDD, PD0~PD7	—	20	—	mA
	高档输出口灌电流	5.0V	VOL=0.1*VDD, PD0~PD7	—	50	—	mA
I <sub>OH3</sub>	输出口拉电流	5.0V	VOH=0.9*VDD, PD0~PD7	—	12	—	mA
V <sub>POR</sub>	上电复位电压	—		—	—	0.1	V
V <sub>BOR</sub>	掉电复位电压	—		—	1.85	—	V
V <sub>EEPROM</sub>	EEPROM 模块工作电压	—		2.4	—	5.5	V
V <sub>LVR</sub>	低电压复位电压	—	LVR 使能, 选择 2.2V	-5%	2.2	+5%	V
			LVR 使能, 选择 2.4V		2.4		
			LVR 使能, 选择 2.6V		2.6		
			LVR 使能, 选择 2.7V		2.7		
			LVR 使能, 选择 2.9V		2.9		

			LVR 使能, 选择 3.0V		3.0		
			LVR 使能, 选择 3.1V		3.1		
			LVR 使能, 选择 3.3V		3.3		
			LVR 使能, 选择 3.6V		3.6		
			LVR 使能, 选择 3.7V		3.7		
			LVR 使能, 选择 3.8V		3.8		
			LVR 使能, 选择 4.1V		4.1		
			LVR 使能, 选择 4.2V		4.2		
			LVR 使能, 选择 4.3V		4.3		
V <sub>FVR</sub>	固定基准电压	5.0V		0.995	1.0	1.005	V
		5.0V	LCDIS<1:0>=00	1.990	2.0	2.010	V
		5.0V	LCDIS<1:0>=01	2.985	3.0	3.015	V
I <sub>BIAS_LCD</sub>	LCD 偏压电 流	5.0V	LCDIS<1:0>=10	—	10	—	uA
		5.0V	LCDIS<1:0>=11	—	50	—	uA
		5.0V	CMP0/1/3	—	100	—	uA
		5.0V	CMP0/1/3	—	200	—	uA
V <sub>CM_CMP1</sub>	共模电压范围	5.0V	CMP2	VSS	—	VDD- 1.5	V
T <sub>RP_CMP1</sub>	响应时间	5.0V	OP 输入共模电压	—	0.2	—	us
V <sub>CM_CMP2</sub>	共模电压范围	5.0V	OP 最大输出电压范围	VSS	—	VDD	V
V <sub>CM_OP</sub>	共模电压范围	5.0V	OP 输入失调校准后	VSS	—	VDD	V
V <sub>OR</sub>	输出电压范围	5.0V		VSS+0.2	—	VDD- 0.2	V
V <sub>OS_OP</sub>	输入失调电压	5.0V			±2		mV
T <sub>ADCK</sub>	ADC 时钟周 期	5.0V	—	62.5			ns
T <sub>ADC</sub>	ADC 转换时 间	5.0V		16			T <sub>AD</sub>

N <sub>R</sub>	分辨率	5.0V				12	Bit
----------------	-----	------	--	--	--	----	-----

### 交流特性

符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件 (常温 25°C)				
F <sub>RCH</sub>	高频内部 RC 振荡器	5.0V	---	15.96	16	16.04	MHz
		5.0V	-40°C~85°C	15.68	16	16.32	MHz
		—	2.4~5.5V	15.92	16	16.08	MHz
F <sub>RCL</sub>	低频内部 RC 振荡器	5.0V			32		KHz
		5.0V	-40°C~85°C	30.40	32	33.60	KHz
		—	2.4~5.5V, -40°C~85°C	27.20	32	36.80	KHz
F <sub>OSH</sub>	外部高频晶振	—	2.4~5.5V	4	—	16	MHz
F <sub>OSL</sub>	外部低频晶振	—	2.4~5.5V	—	32.768	—	KHz
PFG	PFG 时钟	5.0V	-40°C~85°C	1.67	1.7	1.73	MHz
PFG	PFG 时钟	5.0V	-40°C~85°C	1.96	2.0	2.04	MHz
PFG	PFG 时钟	5.0V	-40°C~85°C	2.94	3.0	3.06	MHz
PFG	PFG 时钟	5.0V	-40°C~85°C	3.23	3.3	3.37	MHz

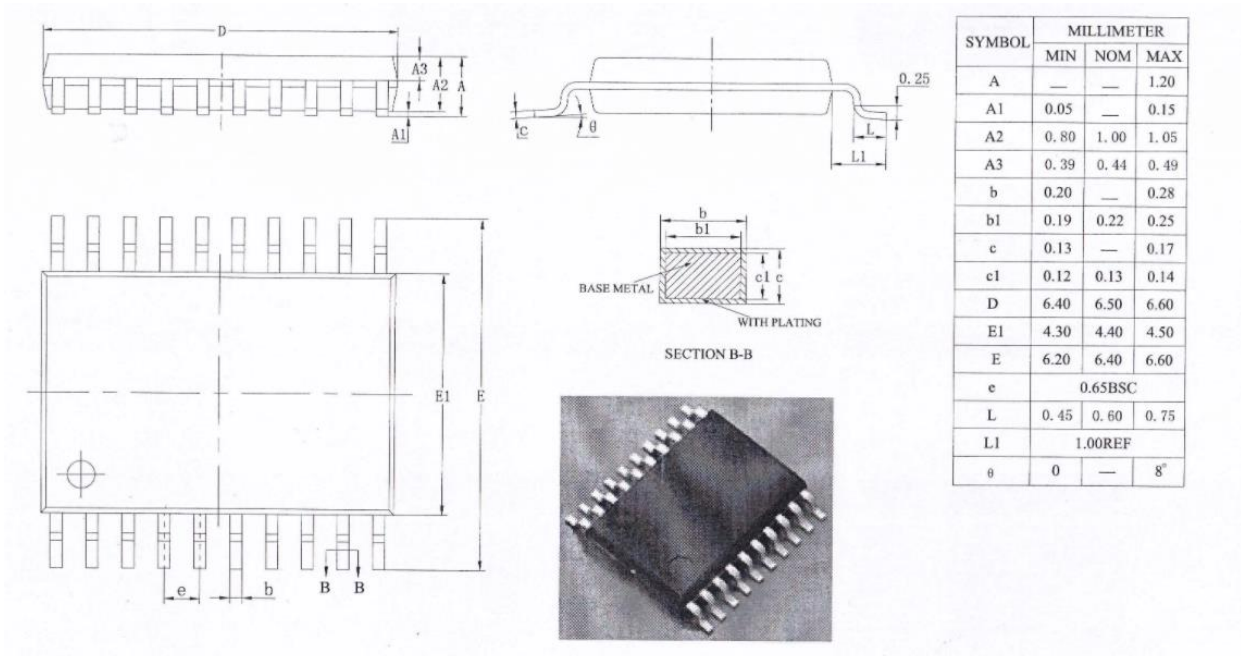
### FLASH & EEPROM

温度: -40°C~85°C

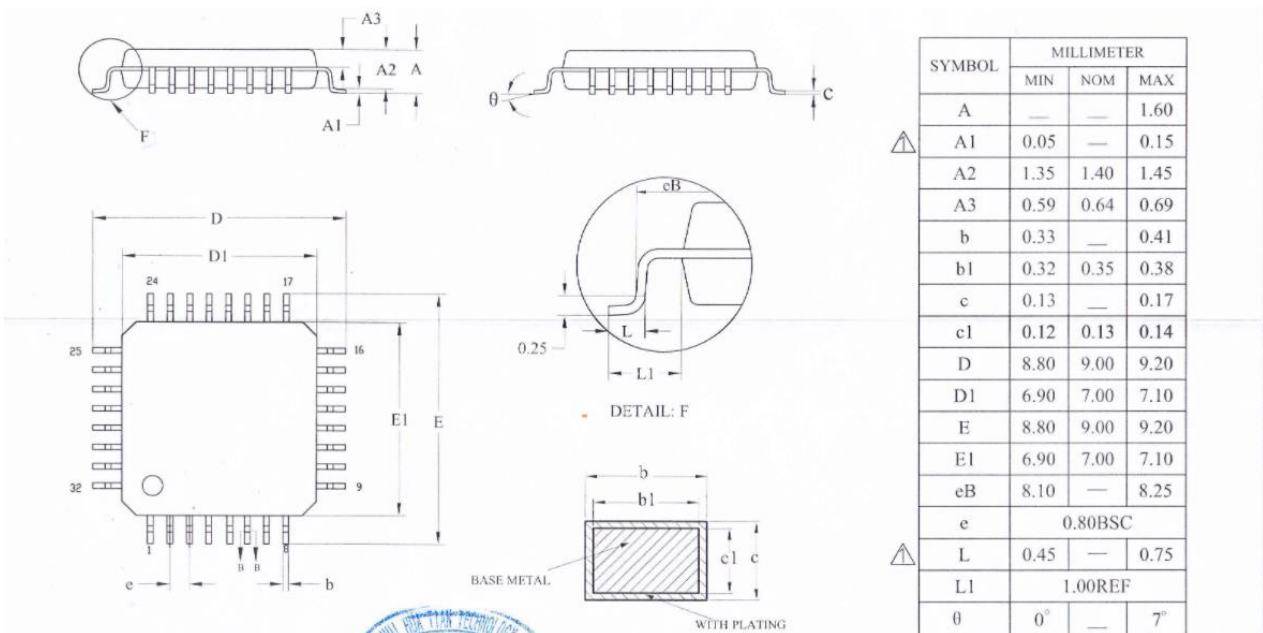
符号	参数	最小值	典型值	最大值	单位
VDD	电源电压	2.4	5.0	5.5	V
VSS	地	0	0	0	V

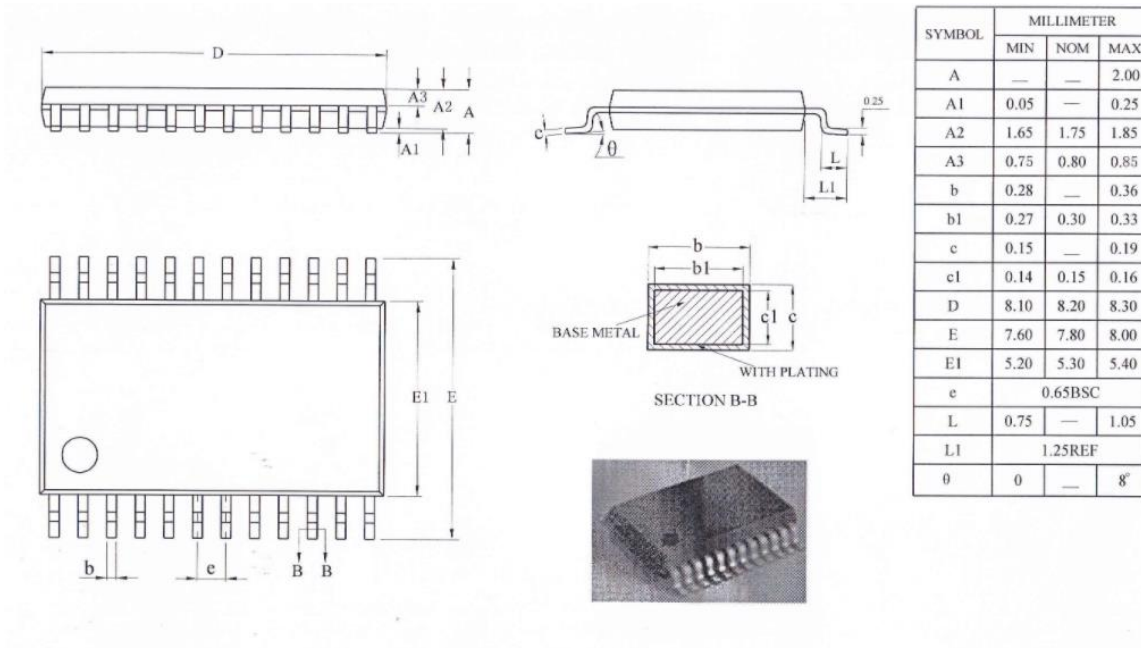
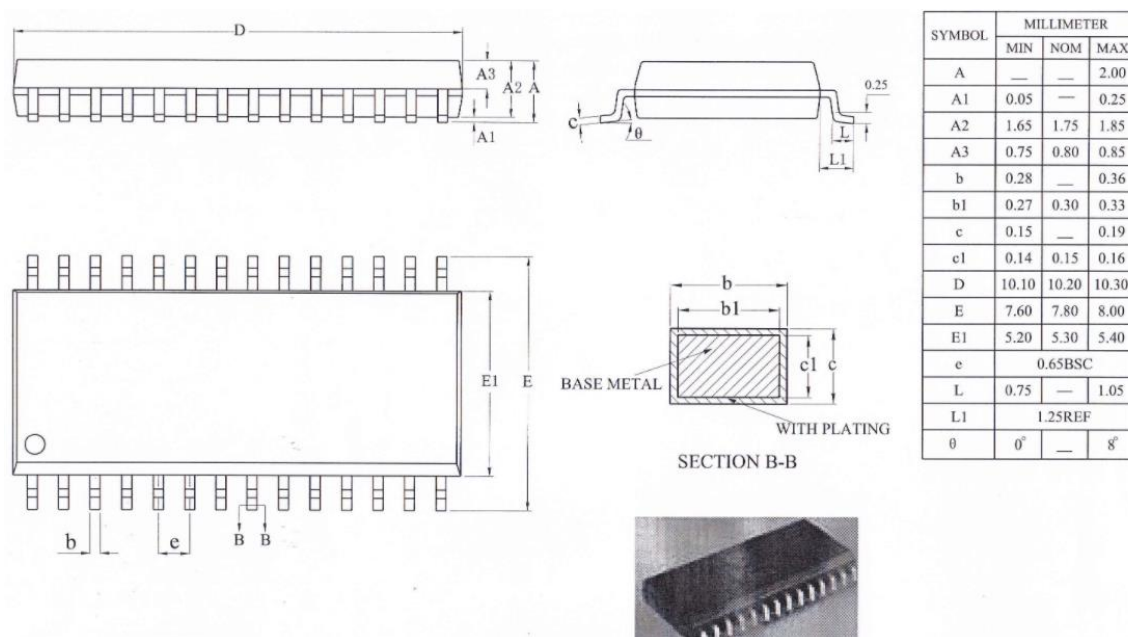
## 22 封装信息

TSSOP20(6.50X4.40X1.00e-0.65)



LQFP32(7.00X7.00X1.40e-0.80)



**SSOP24(8.20X5.30X1.75e-0.65)**

**SSOP28(10.20X5.30X1.75e-0.65)**


## 23 订购信息

### 丝印信息

现行ADUC的单片机表面印有一栏信息：产品代码和日期码。

Marking

F806	2	12	8
Device code	Year	Week	SeriesNo
Year: 0:2020; 2: 2022			
Week: 01:第1周; 12:第12周			
SeriesNo: 序列号0-Z, 1:当前周第2个工单			

### 标签信息

货品内外包装上粘贴的标签上包含：产品名称，封装信息，芯片批号，丝印信息，出货日期及包装数量。

产品名称

Part No: AD18F06P28-XT

封装信息

Package:SSOP28

芯片批号

Lot No: R2H400820

丝印信息

Marking:F80621210

出货日期

Date: 2022-03-29

包装数量

QTY: 48000pcs

**空片**

### 采购信息

AD18F06			
产品名称	封装信息	工作温度	包装方式
AD18F06P28-XT	SSOP28, 绿色封装	-40-85°C	Tube 48/tube
AD18F06L32-X	LQFP32, 绿色封装	-40-85°C	Tray 250/tray