



# AD18F08

---

## 用户手册

版本号: V1.1.0.2

---

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使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
1	V1.1.0.0	初版	Adler
2	V1.1.0.1	产品迭代，工作电压 2.4V~5.5V 更新为 2.0V~5.5V 注：该特性不适用于旧芯片版本，具体芯片版本由芯片丝印确认或咨询销售人员	Adler
3	V1.1.0.2	1.修改引脚排列图：PA0 的 PWM10 改为 PWM11 2.删除 MSOP10 封装 3.删除 FVRCON1(出厂校准，不需要手动校准) 4.增加寄存器列表 5.修改订购信息	SGD

# 目 录

版本修订记录 .....	1
1 产品简介 .....	5
1.1 功能特性 .....	5
1.2 引脚排列 .....	8
1.3 引脚说明 .....	11
1.4 绝对最大额定值 .....	14
1.5 烧录引脚说明 .....	15
2 中央处理器 .....	16
2.1 指令集 .....	16
2.2 程序存储 .....	16
2.3 数据存储器 RAM .....	20
2.4 寄存器列表 .....	21
2.5 数据寻址方式 .....	23
2.6 配置选项 .....	25
3 FLASH .....	29
3.1 注意事项 .....	31
4 系统时钟源 .....	32
4.1 系统时钟相关寄存器 .....	32
4.2 HIRC 频率微调 .....	34
5 复位和电源电压检测 .....	36
5.1 上电复位 .....	37
5.2 低电压复位 .....	37
5.3 上电复位延时 .....	37
5.4 非法指令复位 .....	38
5.5 软件复位 .....	38
5.6 外部管脚复位 .....	38
5.7 堆栈溢出复位 .....	38
5.8 EMC 复位 .....	38
5.9 LVD 检测 .....	38
5.10 /TO /PD 状态 .....	40

6 I/O 端口 .....	42
6.1 IO 工作模式 .....	42
6.2 下拉电阻开漏 .....	43
6.3 模拟 IO 寄存器 .....	45
7 定时器 .....	48
7.1 Timer0/Prescler/BUZZER/PWM .....	48
7.2 TIMER1 16 位定时/计数器 .....	53
7.3 TIMER2 定时器 .....	58
7.4 TIMER3 定时器和 4 路 12 位 PWM .....	61
7.5 看门狗定时器 (WDT) .....	71
8 捕获/比较/PWM 模块 .....	73
8.1 捕捉模式 .....	75
8.2 比较模式 .....	77
8.3 故障刹车功能 .....	79
8.4 3 组互补 PWM 输出 .....	79
9 中断 .....	100
9.1 外部中断 .....	100
9.2 Timer0 中断 .....	101
9.3 Timer1 中断 .....	101
9.4 Timer2 中断 .....	101
9.5 Timer3 中断 .....	101
9.6 PortA 输入改变中断 .....	101
9.7 PortB 输入改变中断 .....	102
9.8 低电压、高电压中断 .....	102
9.9 比较器中断 .....	103
9.10 ADC 中断 .....	103
9.11 CCP 中断 .....	104
9.12 中断的相关寄存器 .....	104
10 省电模式 (SLEEP) .....	110
10.1 睡眠唤醒 .....	110
11 固定参考电压 (FVR) .....	114

12 模数转换器 (ADC) .....	115
13 数模转换器(DAC).....	120
13.1 DAC 参考电压选择寄存器.....	120
14 比较器.....	122
14.1 比较器 CMP.....	122
15 LCD 驱动模块.....	126
15.1 LCD 管脚设置.....	126
15.2 LCD 相关寄存器.....	126
16 8X8 硬件乘法器.....	128
17 电气特性.....	129
17.1 直流交流电气特性.....	129
17.2 FLASH.....	131
18 封装信息.....	132
19 订购信息.....	137

# 1 产品简介

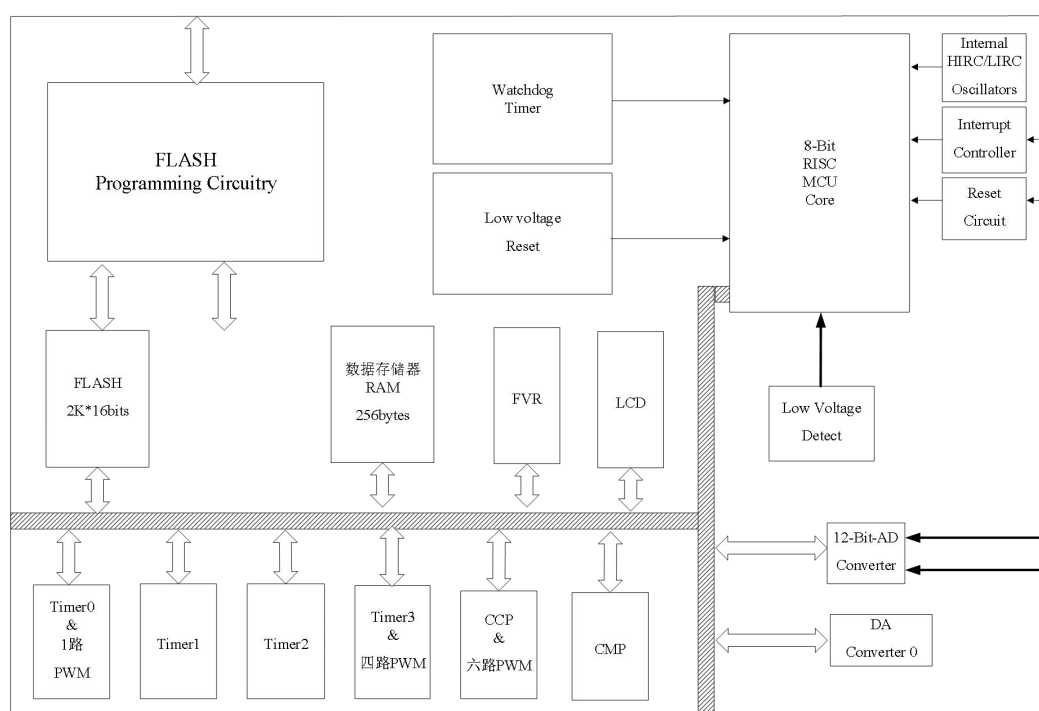
## 1.1 功能特性

- RISC18 指令集，支持 83 条指令，内置 8\*8BIT 硬件乘法器，拥有高低优先级中断：
  - ◇ 程序空间：2K\*16 BIT FLASH，支持 20K 次擦除写入，一页为 128 字节，支持 IAP；支持非法指令复位功能；支持复位指令；支持 EMC 侦测复位功能
  - ◇ 数据空间：256\*8 BIT SRAM
- 堆栈：硬件 8BIT 堆栈，支持 8 层硬件堆栈嵌套
- 支持 2T 和 4T 的 CPU 运算模式；
- 支持在线调试，2 个硬件断点和无限软件断点；
- 内置双时钟模式，高速可选 16M/8M/4M/2M/1M；超低功耗 2K 时钟；并可以作为 FCPU 时钟；通过寄存器可实时切换 CPU 时钟；CPU 有 IDLE, PWSAVE, DEEPPWSAVE, PWOFF 四种低功耗工作模式：
  - ◇ 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 16M 时钟停止工作，低速 2K 时钟工作，工作功耗 30uA；支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 2K 定时唤醒，唤醒后继续从当前 PC 运行；
  - ◇ DEEPPWSAVE: CPU 停止工作，高速 16M 时钟停止工作，低速 2K 时钟工作，SRAM 数据保持，功耗 3uA；支持外部中断、IO 中断、复位、看门狗溢出、LVD 低压唤醒和 TIMER1 的 2K 定时唤醒，唤醒后继续从当前 PC 运行；
  - ◇ PWOFF: 全部外设和模拟停止工作，功耗 1.5uA，支持外部中断、IO 中断、复位，唤醒后继续运行；
- TIMER0, 8 位定时/计数器，可选 FCPU、OSC2K(2K 时钟)、T0CKI、运放输出作为 TIMER0 时钟，同时支持 BUZZER 输出模式；支持四路 8bit 分辨率的 PWM 输出；
- TIMER1, 16 位定时/计数器，有多个时钟输入选择，可以工作在异步时钟工作模式，同时支持 IO 管脚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

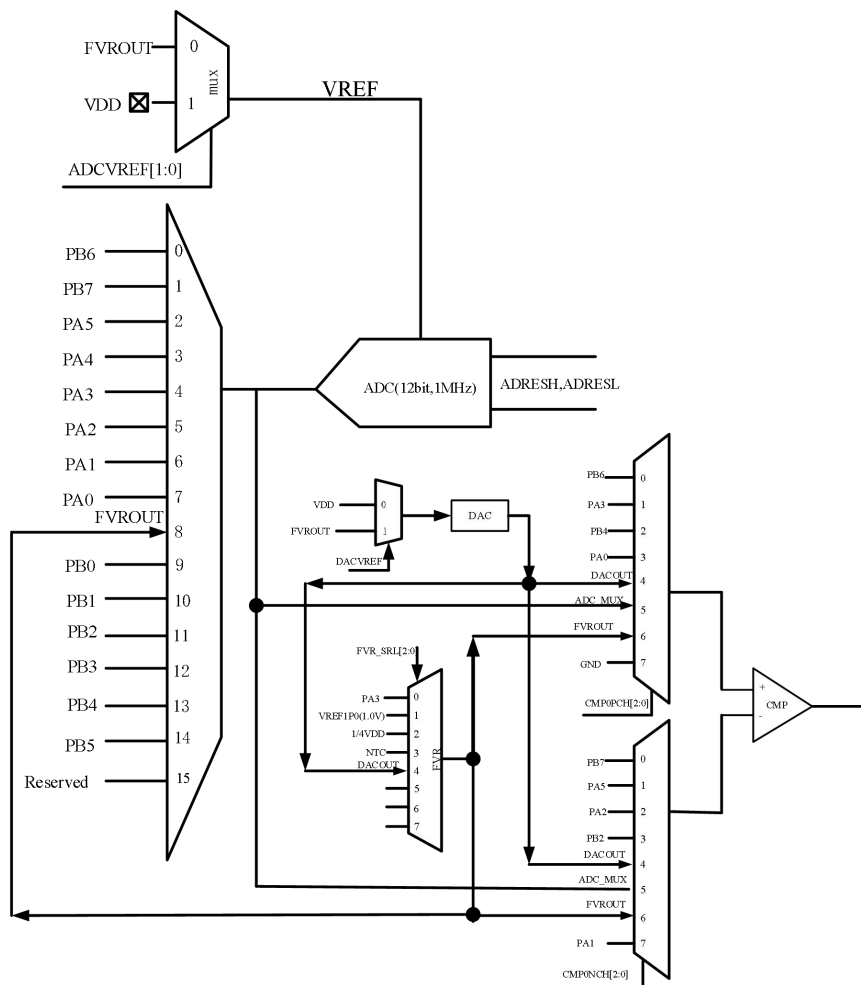
更新为 TIMER2 溢出时间；

- TIMER3, 12 位定时计数器, 支持 CPU 中断, 时钟源为 F<sub>sys</sub>; 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; 与 CCP 的 HBRIDGE 构成 3 路互补可调占空比的三组全桥输出 PWM 信号; 支持中心对齐 PWM 和 ADC 自启动采集;
- CCP, 12 位 CCP 模块, 支持一路捕获, 捕捉可以选择 TIMER1 或者 TIMER2; 一路比较输出, 可以控制 IO 输出和启动 ADC 采集; 支持扩展成六路 12bit 分辨率独立 PWM 功能, 支持 ADC 采集和自启动自关闭; 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;
- 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;
- 12bit 高精度的 ADC, 支持多个 IO 引脚采集, 同时支持内部模拟信号采集; 支持采集比较模式, 可用于低功耗下信号采集;
- LVR 提供 8 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;
- 一路比较器, 用于比较电压的大小;
- 1 路 6 位精度的 DAC, 可以输出到 CMP、FVR; 支持单独配置使能功能, 四种抽头可选择; DAC 基准可以分开控制;
- 支持 1/2 偏压 LCD 驱动;
- 支持 13 路 TOUCH KEY 的输入;
- 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;
- 运行速度: 支持最高 16MHz CPU 运行工作频率和外设运行频率;
- 上电复位计数器 (PWRT) 和振荡启动计数器 (Oscillator Start-up Timer OST);
- 内部振荡器集成了一个看门狗保证了可靠的操作同时软件使能看门狗操作;
- IO 口的驱动能力可配置; PA1、PA2 大电流驱动可达 60mA(Sink);
- 通过编程控制 I/O 端口的上拉/下拉、开漏等状态, 上拉电阻支持两类选择。上电默认所有上下拉电阻关闭;

- 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 和 PortB 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；
  - ✧ ADC 中断；
- 通过外部中断、PortA 中断、PortB 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；
- VDD 工作电压范围：2.0V ~ 5.5V，CPU 工作的最高频率为 16MHz(2T 和 4T 模式)，32MHz(4T 模式)；
- **不建议使用于 AC 阻容降压供电或有高 EFT 要求的应用**
- 封装类型： QFN16(3\*3)、SOP16、SOP8、DFN8、SOP14-2
- 系统结构

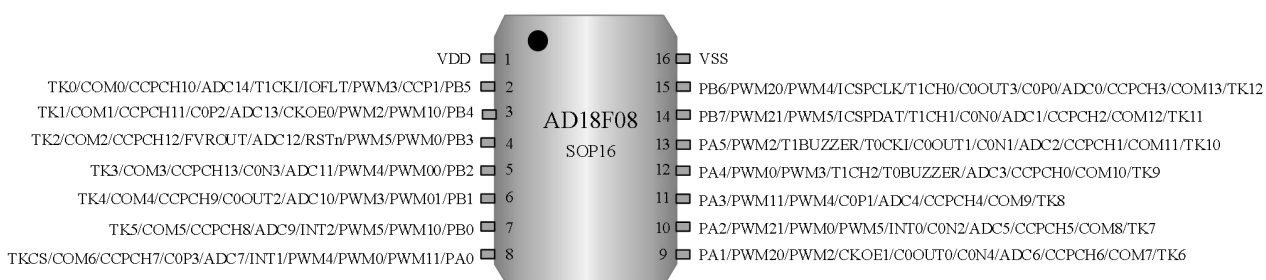


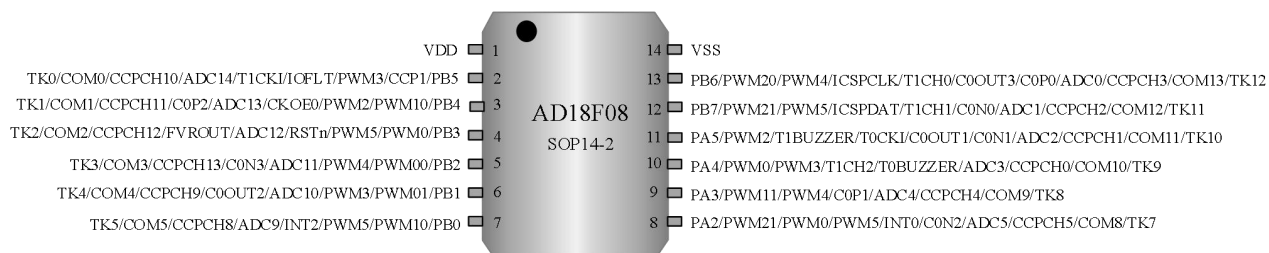


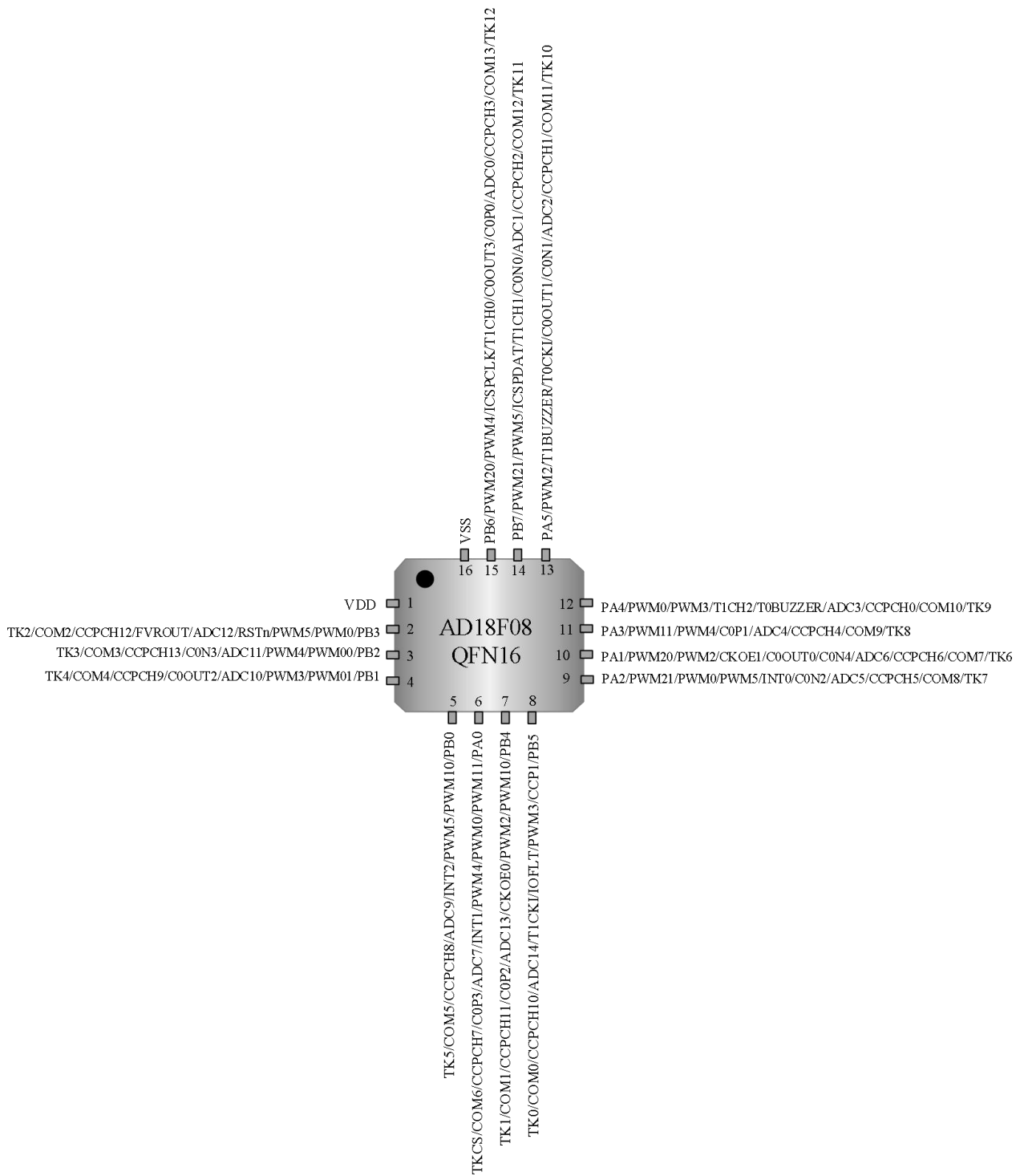


模块关系图

## 1.2 引脚排列







### 1.3 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
	PWM11	-	CMOS	PWM11 的输出		
	PWM4	-	CMOS	PWM4 的输出		
	PWM0	-	CMOS	PWM0 的输出		
	INT1	SMT		外部中断 1		
	CCPCH7	SMT		CCP 的输入通道		
	ADC7	AN	-	ADC 通道输入		
	C0P3	AN	-	CMP 的正端输入		
	TKCS	AN		接片外电容		
	COM6	AN		LCD 的 COM 输出口		
PA1	PA1	SMT	CMOS	GPIO	Y	Y
	PWM20	-	CMOS	PWM20 的输出		
	PWM2	-	CMOS	PWM2 的输出		
	CKOE1	-	CMOS	系统时钟的输出		
	CCPCH6	SMT		CCP 的输入通道		
	C0OUT0	-	CMOS -	CMP 的输出引脚		
	C0N4	AN		CMP 的负端输入		
	ADC6	AN	-	ADC 的输入通道		
	TK6	AN		TOUCH 的输入通道		
	COM7	AN		LCD 的 COM 输出口		
PA2	PA2	SMT	CMOS	GPIO	Y	Y
	PWM21	-	CMOS	PWM21 的输出		
	PWM5	-	CMOS	PWM5 的输出		
	PWM0	-	CMOS	PWM0 的输出		
	INT0	SMT	CMOS	外部中断 0		
	CCPCH5	SMT	CMOS	CCP 的输入通道		
	C0N2	AN	-	CCP 的负端输入		
	ADC5	AN	-	ADC 的输入通道		
	TK7	AN		TOUCH 的输入通道		
	COM8	AN		LCD 的 COM 输出口		
PA3	PA3	SMT	CMOS	GPIO	Y	Y
	PWM11	-	CMOS	PWM11 的输出		
	PWM4	-	CMOS	PWM 的输出		
	CCPCH4	SMT	-	CCP 的输入通道		
	C0P1	AN	-	CMP 的正端输入		
	ADC4	SMT	-	ADC 的输入通道		
	TK8	AN		TOUCH 的输入通道		
	COM9	AN		LCD 的 COM 输出口		

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
PA4	PA4	SMT	CMOS	GPIO	Y	Y
	PWM0	-	CMOS	PWM0 的输出		
	PWM3	-	CMOS	PWM3 的输出		
	T1CH2	SMT		TIMER1 的脉宽输入		
	T0BUZZER		CMOS	TIMER0 的 BUZZ 输出		
	CCPCH0	SMT		CCP 的输入通道		
	ADC3	AN		ADC 的输入通道		
	TK9	AN		TOUCH 的输入通道		
	COM10	AN		LCD 的 COM 输出口		
PA5	PA5	SMT	CMOS	GPIO	Y	Y
	PWM2	-	CMOS	PWM2 的输出		
	T0CKI	S-MT		TIMER0 的时钟源输入		
	T1BUZZER		CMOS	TIMER1 的 BUZZ 输出		
	CCPCH1	SMT		CCP 的输入通道		
	C0OUT1	-	CMOS	CMP 的输出		
	C0N1	AN		CMP 的负端输入		
	ADC2	AN		ADC 的输入通道		
	TK10	AN		TOUCH 的输入通道		
	COM11	AN		LCD 的 COM 输出口		
PB0	PB0	SMT	CMOS	GPIO	Y	Y
	PWM10	-	CMOS	PWM10 的输出		
	PWM5	-	CMOS	PWM5 的输出		
	INT2	SMT	-	外部中断 2		
	CCPCH8	SMT	-	CCP 的输入通道		
	ADC9	AN	-	ADC 的输入通道		
	TK5	AN		TOUCH 的输入通道		
	COM5	AN		LCD 的 COM 输出口		
PB1	PB1	SMT	CMOS	GPIO	Y	Y
	PWM01	-	CMOS	PWM01 的输出		
	PWM3	-	CMOS	PWM3 的输出		
	CCPCH9	SMT		CCP 的输入通道		
	C0OUT2	-	CMOS	CMP 的输出		
	ADC10	AN	-	ADC 的输入通道		
	TK4	AN		TOUCH 的输入通道		
	COM4	AN		LCD 的 COM 输出口		
PB2	PB2	SMT	CMOS	GPIO	Y	Y
	PWM00	-	CMOS	PWM00 的输出		
	PWM4	-	CMOS	PWM4 的输出		
	CCPCH13	SMT	-	CCP 的输入通道		
	C0N3	AN	-	CMP 的负端输入		
	ADC11	AN	-	ADC 的输入通道		

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
	<b>TK3</b>	AN		TOUCH 的输入通道		
	<b>COM3</b>	AN		LCD 的 COM 输出口		
<b>PB3</b>	<b>PB3</b>	SMT	CMOS	GPIO	Y	Y
	<b>PWM0</b>	-	CMOS	PWM0 的输出		
	<b>PWM5</b>	-	CMOS	PWM5 的输出		
	<b>RSTn</b>	SMT	-	芯片复位引脚		
	<b>CCPCH12</b>	SMT	-	CCP 的输入通道		
	<b>FVROUT</b>	AN		FVR 的输出		
	<b>ADC12</b>	AN		ADC 的输入通道		
	<b>TK2</b>	AN		TOUCH 的输入通道		
	<b>COM2</b>	AN		LCD 的 COM 输出口		
<b>PB4</b>	<b>PB4</b>	SMT	CMOS	GPIO	Y	Y
	<b>PWM10</b>	-	CMOS	PWM10 的输出		
	<b>PWM2</b>	-	CMOS	PWM2 的输出		
	<b>CKOE0</b>	-	CMOS	系统时钟输出		
	<b>CCPCH11</b>	SMT	-	CCP 的输入通道		
	<b>C0P2</b>	AN		CMP 的正端输入		
	<b>ADC13</b>	AN		ADC 的输入通道		
	<b>TK1</b>	AN		TOUCH 的输入通道		
	<b>COM1</b>	AN		LCD 的 COM 输出口		
<b>PB5</b>	<b>PB5</b>	SMT	CMOS	GPIO	Y	Y
	<b>CCP1</b>	-	CMOS	CCP 的输出		
	<b>PWM3</b>	-	CMOS	PWM3 的输出		
	<b>IOFLT</b>	SMT	-	IO 故障检测引脚		
	<b>T1CKI</b>	SMT	-	TIMER1 的时钟源输入		
	<b>CCPCH10</b>	SMT	-	CCP 的输入通道		
	<b>ADC14</b>	AN		ADC 的输入通道		
	<b>TK0</b>	AN		TOUCH 的输入通道		
<b>COM0</b>	AN		LCD 的 COM 输出口			
<b>PB6</b>	<b>PB6</b>	SMT	CMOS	GPIO	Y	Y
	<b>PWM20</b>	-	CMOS	PWM20 的输出		
	<b>PWM4</b>	-	CMOS	PWM4 的输出		
	<b>ICSPCLK</b>	SMT	-	烧录芯片的时钟引脚		
	<b>T1CH0</b>	SMT	-	TIMER1 的脉宽输入		
	<b>CCPCH3</b>	SMT	-	CCP 的输入通道		
	<b>C0OUT3</b>	-	CMOS	CMP 的输出引脚		
	<b>C0P0</b>	AN		CMP 的正端输入		
	<b>ADC0</b>	AN		ADC 的输入通道		
	<b>TK12</b>	AN		TOUCH 的输入通道		
	<b>COM13</b>	AN		LCD 的 COM 输出口		
<b>PB7</b>	<b>PB7</b>	SMT	CMOS	GPIO	Y	Y

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
	<b>PWM21</b>		CMOS	PWM21 的输出		
	<b>PWM5</b>		CMOS	PWM5 的输出		
	<b>ICSPDAT</b>	SMT	CMOS	烧录芯片的数据引脚		
	<b>T1CH1</b>	SMT	-	TIMER1 的脉宽输入		
	<b>CCPCH2</b>	SMT	-	CCP 的输入通道		
	<b>C0N0</b>	AN		CMP 的负端输入		
	<b>ADC1</b>	AN		ADC 的输入通道		
	<b>TK11</b>	AN		TOUCH 的输入通道		
	<b>COM12</b>	AN		LCD 的 COM 输出口		

## 1.4 绝对最大额定值

符号	描述	最小值	最大值	单位
VCC-VSS	电源电压	-0.3	+6.0	V
VIN	端口输入信号电压	VSS-0.3	VCC+0.3	V
IVCC	VCC 的最大电流	-	+100	mA
IVSS	VSS 的最大电流	-	-100	mA
TJ	最大结温	-	+150	°C
TSTG	存储温度范围	-55	+150	°C
TA	工作温度	-40	+85	°C

注：如果运行条件超过了上述“绝对最大额定值”，即可能对器件造成永久性损坏。上述值仅为运行条件的极大值，我们不建议器件运行在该规范范围以外。器件长时间工作在绝对最大额定值条件下，其稳定性可能受到影响。

## 1.5 烧录引脚说明

AD18F08 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	PB7	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	PB6	CLK	



## 2 中央处理器

### 2.1 指令集

AD18F08 具有一个支持 83 条内核指令的标准指令集。其中包含 8 条针对优化递归和软件堆栈代码的扩展指令。

### 2.2 程序存储

AD18F08 包含一个 2K\*16 的 FLASH 用于存储用户程序和数据。

AD18F08 有两个中断向量。复位向量地址为 0000h,中断向量地址为 0008h 和 0018h。

#### 2.2.1. 程序计数器

程序计数器 (Program Counter,PC), 指定要取出执行的指令地址。PC 内的地址为 16 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<15:8>位, 不可直接读写。可以通过 PCLATH 寄存器更新 PCH 寄存器。

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

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

#### 2.2.2. 返回地址堆栈

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

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

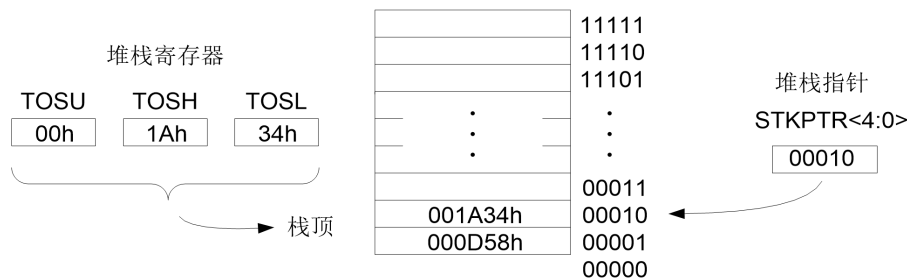
执行 CALL 类型指令引起进栈操作: 堆栈指针首先加 1, 并且将 PC 的内容写入堆栈指针指向的地址单元 (PC 已经指向 CALL 的下一条指令)。执行 RETURN 类型指令时, 引起出栈操作: STKPTR 寄存器所指向的地址单元的内容被传送给 PC, 然后堆栈指针减 1。

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

### 2.2.3. 栈顶访问

只有栈顶 (Top-of-Stack, TOS) 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。



### 2.2.4. 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	00000

### 2.2.5. 快速寄存器堆栈

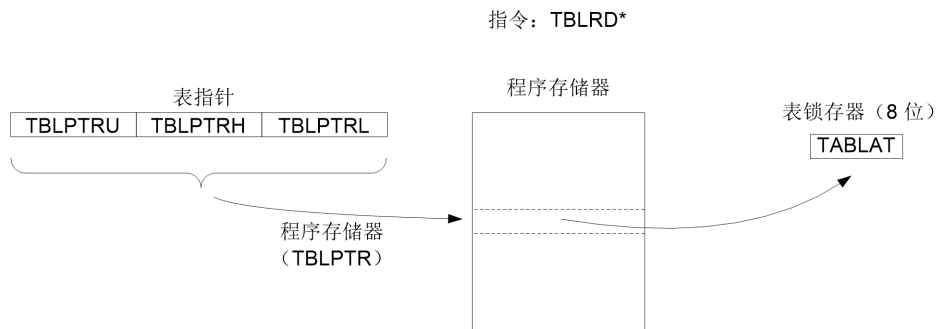
为 Status、WREG 和 BSR 寄存器提供的快速寄存器堆栈具有从中断“快速返回”的功能。每个寄存器堆栈的深度仅为 1 级，并且不可读写。当处理器转入中断向量处执行指令时，此堆栈装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用 RETFIE，FAST 指令从中断返回，这些寄存器中的值会被重新装回对应的寄存器。

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。

### 2.2.6. 程序存储器

为了读取程序存储器，AD18F08 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

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

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

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

示例	表指针操作
TBLRD*	不修改 TBLPTR
TBLRD*+	TBLPTR 在读后递增
TBLRD*-	TBLPTR 在读后递减
TBLRD+*	TBLPTR 在读前递增

### 使用 TBLRD 指令执行表指针操作

读取一个闪存程序存储器字:

```

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

AD18F08 的数据存储器是用静态 RAM 实现的。存储空间包含 256 个字节；

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

数据存储器映射



SRAM 数据存储器映射图

## 2.4 寄存器列表

add	register	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xff	TOSU								
0xffe	TOSH						TOS[12:8]		
0xffd	TOSL	TOS[7:0]							
0xffc	SIKPIR	SIKFUL	SIKUNF				SP2	SP1	SP0
0xffb	PCLATU								
0xffa	PLCAIH						PC[12:8]		
0xff9	PCL	PC[7:0]							
0xff8	TBLPIRU								
0xff7	TBLPTRH						TBLPTR[12:8]		
0xff6	TBLPIRL	TBLPTR[7:0]							
0xff5	TABLAT	Program Memory TaLEB Lath							
0xff4	PRODH	Product Register High Byte							
0xff3	PRODL	Product Register Low Byte							
0xff2	INTCON	GIE/GIEH	PEIE/GIEL						
0xff1	PCON	IPEN	PB0ST	LVDM[1:0]				LVD[2:0]	
0xff0									
0xfef	INDF	not a physical register							
0xfef	POSTINC0	not a physical register							
0xfed	POSTDEC0	not a physical register							
0xfec	PREINC0	not a physical register							
0xfef	PLUSW0	not a physical register							
0xfea	FSR0H						FSR0[11:8]		
0xfe9	FSR0L	FSR0[7:0]							
0xfe8	WREG	WREG[7:0]							
0xfe7	INDF1	not a physical register							
0xfe6	POSTINC1	not a physical register							
0xfe5	POSTDEC1	not a physical register							
0xfe4	PREINC1	not a physical register							
0xfe3	PLUSW1	not a physical register							
0xfe2	FSR1H						FSR1[11:8]		
0xfe1	FSR1L	FSR1[7:0]							
0xfe0	BSR						BSR[3:0]		
0xfdf	INDF2	not a physical register							
0xfde	POSTINC2	not a physical register							
0xfdd	POSTDEC2	not a physical register							
0xfdc	PREINC2	not a physical register							
0xfdb	PLUSW2	not a physical register							
0xfda	FSR2H						FSR2[11:8]		
0xfd9	FSR2L	FSR2[7:0]							
0xfd8	STATUS				N	OV	Z	DC	C
0xfd7	T0CON0	T0OUT	T0CS1	T0CS0	T0SE	PSA	PS2	PS1	PS0
0xfd6	PR0	PR0[7:0]							
0xfd5	TMR0	TMR0[7:0]							
0xfd4	PWM0DUTY	PWM0DUTY[7:0]							
0xfd3	PWM0CON	POLS_03	POLS_02	POLS_01	POLS_00	PWM03EN	PWM02EN	PWM01EN	PWM00EN
0xfd2	RCON	IRFEN	EMCEN	RSTIE[1:0]		TO	PD	POR	BOR
0xfd1	TMR1H	Timer1 Register High Byte							
0xfd0	TMR1L	Timer1 Register Low Byte							
0xfef	T1CON0	T1CK1	T1CK0	T1CKPS1	T1CKPS0	T1SE	T1SYNC	TMR1CS0	TMR1ON
0xfef	T1CON1	INT2EDGE	INT1EDGE	INT0EDGE	T1EDGE	T1CH1	T1CH0	T1M1	T1M0
0xfed	TMR2L	TMR2L[7:0]							
0xfec	TMR2H						TMR2H[11:8]		
0xfec	PR2L	PR2L[7:0]							
0xfca	PR2H						PR2H[11:8]		
0xf9	T2CON	PWM2CAEN	T2OUIPS3	T2OUIPS2	T2OUIPS1	T2OUIPS0	TMR2ON	T2CKPS1	T2CKPS0
0xf8	CCPR1H	Capture/Compare/PWM Register 1 High Byte							
0xf7	CCPR1L	Capture/Compare/PWM Register 1 Low Byte							
0xf6	CCPR1LH	T2CAPEN					CCPR1L[11:8]		
0xf5	PMS	DMISTK2	PWMH	PMS2[1:0]		PMS1[1:0]		PMS0[1:0]	
0xf4	PXC	T3PWMMEN	T3PWMMODE	PXC2[1:0]		PXC1[1:0]		PXC0[1:0]	
0xf3	DTC	DTCKS1	DTCKS0	DTEN	DID4	DID3	DID2	DID1	DID0
0xf2	POLS	PT2MODE		POLS2[1:0]		POLS1[1:0]		POLS0[1:0]	
0xf1	PME	PT1MODE	PT0MODE	PWM2EN[1:0]		PWM1EN[1:0]		PWM0EN[1:0]	
0xf0	PCS					PWM21CS	PWM20CS	PWM11CS	PWM10CS

add	register	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
0xfbf	CCPCON	CCP1CH[3:0]			CCP1M[3:0]					
0xfbe	PWMCON0	TKSEL	CMPFLT	ASTART	ACLOSE	IOFLT	PWMADPOS[1:0]		PWMADEN	
0xfbd	PWMCON1	PWMADDLY[7:0]								
0xfbc	T3CON	PWMBCAEN	T3OUTPS3	T3OUTPS2	T3OUTPS1	T3OUTPS0	TMR3ON	T3CKPS1	T3CKPS0	
0xfbb	TMR3H	TMR3H[11:8]								
0xfba	TMR3L	TMR3L[7:0]								
0xfb9	PR3H	PR3H[11:8]								
0xfb8	PR3L	PR3L[7:0]								
0xfb7	PWMEH	PWME[11:8]								
0xfb6	PWML	PWML[7:0]								
0xfb5	PWMEH	PWME[11:8]								
0xfb4	PWML	PWML[7:0]								
0xfb3	PWMEH	PWME[11:8]								
0xfb2	PWML	PWML[7:0]								
0xfb1	PWMEH	PWME[11:8]								
0xfb0	PWML	PWML[7:0]								
0xfaf	PWMCON0	PWMOEN	PWMOEN	PWMOEN	PWMOEN	PWMEEN	PWMEEN	PWMBEN	PWMEEN	
0xfae	PWMCON1	POL55								
0xfad	PWMCON3	PSCS[1:0]		P4CS[1:0]		P3CS[1:0]		P2CS[1:0]		
0xfab	ADCMF0H	A/D Result Compare Register 0 High Byte								
0xfaa	ADCMF1H	A/D Result Compare Register 1 High Byte								
0xfa9	ADCMF0L	A/D Result Compare Register 1 Low[3:0]				A/D Result Compare Register 0 Low[3:0]				
0xfa8	ADRESH	A/D Result Register High Byte								
0xfa7	ADRESL	A/D Result Register Low Byte								
0xfa6	ADCON0	ADFM	ADCMPODE	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	
0xfa5	ADCON1	ADVREF1	ADVREF0	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	
0xfa4	ADCON2	ACQT[6:3]				ANC[1:0]		ACFVR	ACEN	
0xfa3	IPR1	ADCMPI1P	ADCMPO1P	ADIP	CC1P	TMR31P	TMR21P	TMR11P	TRM01P	
0xfa2	PIR1	ADCMPI1F	ADCMPO1F	ADIF	CC1F	TMR31F	TMR21F	TMR11F	TRM01F	
0xfa1	PIE1	ADCMPI1E	ADCMPO1E	ADIE	CC1E	TMR31E	TMR21E	TMR11E	TRM01E	
0xfa0	IPR2		PB1P	PA1P	INT2IP	INT1IP	INT0IP	LVDIP	CMP1P	
0xf9f	PIR2		PB1F	PA1F	INT2IF	INT1IF	INT0IF	LVDIF	CMP1F	
0xf9e	PIE2		PB1E	PA1E	INT2IE	INT1IE	INT0IE	LVDIE	CMP1E	
0xf99	SMTB	SMTB7	SMTB6	SMTB5	SMTB4	SMTB3	SMTB2	SMTB1	SMTB0	
0xf98	SMTA		SMTA5	SMTA4	SMTA3	SMTA2	SMTA1	SMTA0		
0xf95	PBOD	PB7OD	PB6OD	PB5OD	PB4OD	PB3OD	PB2OD	PB1OD	PB0OD	
0xf94	PAOD		PA5OD	PA4OD	PA3OD	PA2OD	PA1OD	PA0OD		
0xf91	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	
0xf90	TRISA		TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0		
0xf8d	PBPD	PB7PD	PB6PD	PB5PD	PB4PD	PB3PD	PB2PD	PB1PD	PB0PD	
0xf8c	PAPD		PA5PD	PA4PD	PA3PD	PA2PD	PA1PD	PA0PD		
0xf89	PINB	FORIB PIN Status[7:0]								
0xf88	PINA	PORTA PIN Status[5:0]								
0xf85	PBPU	PB7PU	PB6PU	PB5PU	PB4PU	PB3PU	PB2PU	PB1PU	PB0PU	
0xf84	PAPU		PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU		
0xf81	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
0xf80	PORTA		PA5	PA4	PA3	PA2	PA1	PA0		
0xf78	CMPCON0	CMPEN	CMPPOS	CMPFCH[2:0]			CMPNCH[2:0]			
0xf77	CMPCON1	HYSN_VOL[1:0]		HYSN_VOL[1:0]		CMPOUT				
0xf76	CMPCON2	CMPFLLIER[3:0]			C0OUT3EN		C0OUT2EN		C0OUT1EN	
0xf74	DACCON	DACVREF	DACEN	DACS5	DACS4	DAC[3:0]				
0xf72	FVRCON0	FVR_A2D_OUI	FVROUTIEN	FVREN	FVRPGA[1:0]		FVR_SEL[2:0]			
0xf70	LCDCON0	LCDEN	LCDCSEL[1:0]							
0xf6f	LCDCON1	COM7EN	COM6EN	COM5EN	COM4EN	COM3EN	COM2EN	COM1EN	COM0EN	
0xf6e	LCDCON2		COM3EN	COM2EN	COM1EN	COM0EN	COM0EN	COM0EN	COM0EN	
0xf6c	IAPTRIG	0X18,0XF0,0X70								
0xf6b	IAPCTRL				ERR	LOCK	ER	PG		
0xf6a	IAPADDRH	IAPADDR[11:8]								
0xf69	IAPADDRL	IAPADDR[7:0]								
0xf68	IAPDAIH	IAPDAI[11:8]								
0xf67	IAPDAIL	IAPDAI[7:0]								
0xf66	IAPWAIT	IAPWAIT[7:0]								
0xf65	SMCR					SMD	SMD	SMD	SE	
0xf64	OSCCON	PLL	FINTOSC[2:0]		CKOE1	CKOE0	OSCF	OSCM		
0xf63	SIGCON	ICE_IMEREN	SIGS							
0xf62	CLKCFG1	CMPCLKEN	DACCLKEN	ADCLKEN	CCCLKEN	T3CLKEN	T2CLKEN	T1CLKEN	T0CLKEN	
0xf60	PAINTMASK		PAINMASK5	PAINMASK4	PAINMASK3	PAINMASK2	PAINMASK1	PAINMASK0		
0xf5f	PBINTMASK	PBINTMASK7	PBINTMASK6	PBINTMASK5	PBINTMASK4	PBINTMASK3	PBINTMASK2	PBINTMASK1	PBINTMASK0	
0xf58	WDTCON	PREDIV[2:0]							WDIEN	
0xf55	CURCON				PBCUR	PACUR	PA2CUR	PA1CUR		
0xf54	ANASEL0		PA5AEN	PA4AEN	PA3AEN	PA2AEN	PA1AEN	PA0AEN		
0xf53	ANASEL1	PB7AEN	PB6AEN	PB5AEN	PB4AEN	PB3AEN	PB2AEN	PB1AEN	PB0AEN	

## 2.5 数据寻址方式

AD18F08 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

### 2.5.1. 固有和立即数寻址

很多 AD18F08 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 16 位的程序存储器地址。

### 2.5.2. 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。

### 2.5.3. 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。

#### 2.5.3.1. 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。

### 2.5.3.2. 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。

### 2.5.3.3. 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。

## 2.6 配置选项

**注:不经过配置的配置字默认为 1;**

### 配置字 0

位	名称	说明
2:0	LVR [2:0]	LVR 电压选择 000: 2.4V 001: 2.7V 010: 3.0V 011: 3.3V 100: 3.6V 101: 3.8V 110: 4.2V 111: 4.3V
3	LVREN	LVR 使能控制位 1: 使能 0: 禁止

4	DBGEN	DEBUG 模式使能位, 1: 使能 0: 禁止
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 010: 1:16 分频

### 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
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: 100K 下拉电阻: 300K
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
3	CBP	代码保护选择位 1: 代码加密 0: 代码不加密
4	CKSUM	校验使能位 1: 使能校验

		0: 关闭校验
5	IDIS	模拟 IO 打开，数据读取控制位。 1: 模拟 IO 打开，能正常读取 IO 输入 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	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
7	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

## 配置字 2

位	名称	说明
2:0	TWDT [2:0]	看门狗溢出时间选择 111: TWDT (no Prescaler) = 8.32s 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 注：若配置字中使能了WDT，则上电复位延时时间不可大于WDT时间，否则无法进入主函数。如选择延时为864.55ms时，WDT溢出时间只可选择1.152s及以上。
3	WDTE	WDT 使能 1: 使能 WDT

		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 0: 1 个指令周期为 2 个机器周期
6:5	PSUT [1:0]	上电复位延时时间选择 11: PWRT = 25.8ms 10: PWRT = 288.55ms 01: PWRT = 864.55ms 00: PWRT = 8.05ms
7	Reserved	

### 配置字 3

位	名称	说明
0	ADCSP	ADC 采集速度选择位 1: 高速 0: 低速
1	MTPDIS	SWD 模式下读 MTP 控制 1: 允许读 0: 禁止读
6:2	Reserved	
7	OSCM	内部芯片工作时钟选择 0: 选择内部低速时钟 2KHz 1: 选择内部高速时钟 16MHz

### 3 FLASH

FLASH 可以自编程的地址空间从 000H 到 FFFH，IAPADDR[11:0]（由 IAPADDRH 高 4 位和 IAPADDRL 的低 8 位寄存器组成）对应 FLASH 的地址。IAPDATH[15:8]和 IAPDATL[7:0]组成 16 位数据。

FLASH 写入数据操作步骤：

- (1) 写入电压为芯片工作电压；进行读写操作的电压为：2.4V~5.5V；
- (2) 对 IAPTRIG 连着写入 0xAD,0x18,0xF8，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 000H~FFFH 地址内的某个地址。
- (4) 设置 IAPDATH（高位数据）和 IAPDATL（低位数据），即把需要写入数据寄存器的数据暂放在 IAPDAT 寄存器中。
- (5) CPU 将 IAPDAT[15:0]数据写入 IAPADDRH 和 IAPADDRL 对应的 FLASH 地址中。
- (6) 如果要继续写，则回到步骤（3）开始。如果要退出写入操作，则到步骤（7）。
- (7) 对 IAPTRIG 写入 0，便可以退出写操作。

#### IAPTRIG 寄存器

地址：0XF6C

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

#### IAPCTRL 寄存器

地址：0XF6B

Bit	Name	Description	Attribute	Reset
7:4	Reserve			
3	ERR	FLASH 的擦写错误标志位 1：以下情况发生时，产生擦写错误 a) 页擦除 ER 和写操作 PG 同时执行 b) 代码加密时进行页擦除操作或写操作	R/W	0

Bit	Name	Description	Attribute	Reset
		0: 未发生擦写错误		
2	LOCK	IAP 锁定状态位 1: 锁定 0: 未锁定	R	0
1	ER	FLASH 页擦除 (Fsys 为 16Mhz 时需要耗时 100ms, 无需手动延时, 擦除完成后硬件自动清 0) 1: 使能擦除 0: 擦除完成	R/W	0
0	PG	写入控制位 (写入完成后硬件自动清 0) 1: 使能写入 0: 写入完成	R/W	0

### IAPADDRH 寄存器

地址: 0XF6A

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	IAPADDRH [11:8]	IAP 地址高 4 位	R/W	0000

### IAPADDRL 寄存器

地址: 0XF69

Bit	Name	Description	Attribute	Reset
7:0	IAPADDRL [7:0]	IAP 地址低 8 位	R/W	0x00

### IAPDATH 寄存器

地址: 0XF68

Bit	Name	Description	Attribute	Reset
7:0	IAPDATH [15:8]	IAP 数据高 8 位	R/W	0x00

### IAPDATL 寄存器

地址: 0XF67

Bit	Name	Description	Attribute	Reset
7:0	IAPDATL [7:0]	IAP 数据低 8 位	R/W	0x00

## IAPWAIT 寄存器

地址：0XF66

Bit	Name	Description	Attribute	Reset
7:0	IAPWAIT [7:0]	IAP 延时位（根据 Fcpu 来调整 FLASH 相关操作延时） 典型值速算： Fcpu= 8MHz, IAPWAIT=15 Fcpu= 4MHz, IAPWAIT=7 Fcpu= 2MHz, IAPWAIT=4 Fcpu= 1MHz, IAPWAIT=1 Fcpu= 500KHz, IAPWAIT=0 （Fcpu 见章节 4）	R/W	0x00

### 3.1 注意事项

- 对 FLASH 进行 IAP 操作时，不应使能配置字中的 CKSUM 功能。
- 对 FLASH 编程时，应保证电压范围 2.4V~5.5V。
- 当 HIRC 倍频至 32MHz 时，仅支持 4T 模式运行，且电压需高于 2.5V。
- 当工作电压低于 2.5V 时，应主动将 Fsys 频率降低一半。

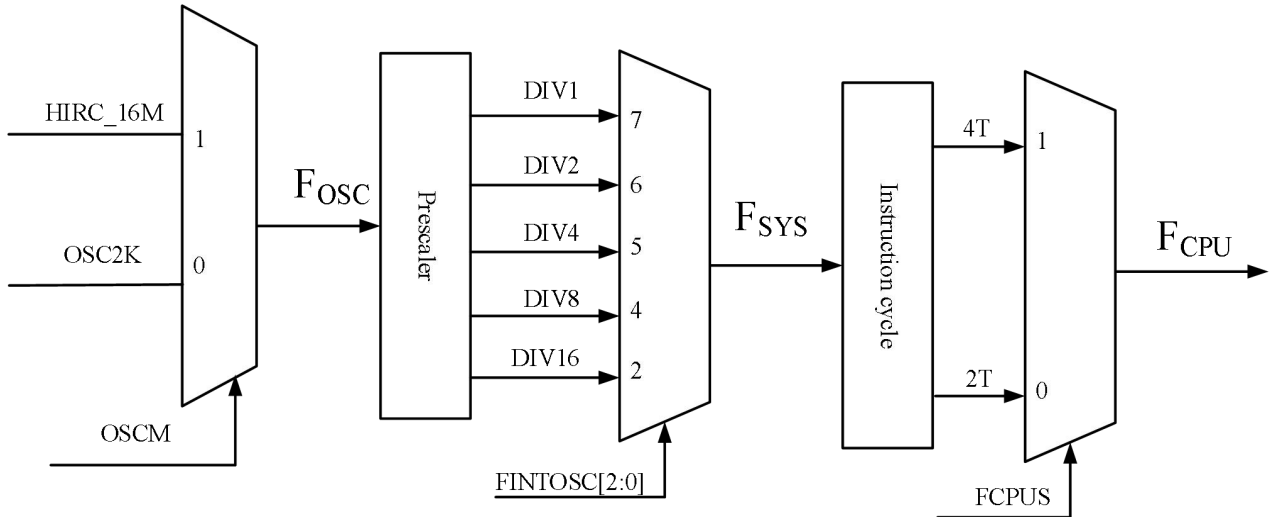
例如：

压	FCPU 电	4T	2T
	≥2.5V	32MHz	16MHz
<2.5V	16MHz	8MHz	



## 4 系统时钟源

### 4.1 系统时钟相关寄存器



#### OSCCON 寄存器

地址：0XF64

Bit	Name	Description	Attribute	Reset
7	PLL	时钟的倍频选择 1: 系统时钟为 32M 0: 系统时钟为 16M 注：系统时钟为 32M 时，指令周期支持 2T(工作电压 3.5V~5V)和 4T 模式； 系统时钟为 16M 时，指令周期 2T 和 4T 模式都支持；	R/W	0
6:4	FINTOSC	内部振荡器频率选择位 111: 1:1 分频 110: 1:2 分频 101: 1:4 分频 100: 1:8 分频 010: 1:16 分频	R/W	010
3	CKOE1	PA1 输出当前系统时钟	R/W	0

Bit	Name	Description	Attribute	Reset
		1: 使能系统时钟输出 0: 禁止系统时钟输出		
2	CKOE0	PB4 输出当前系统时钟 1: 使能系统时钟输出 0: 禁止系统时钟输出	R/W	0
1	OSCF	工作时钟状态位 1: 表示内部高速时钟工作 (16MHz) 0: 表示内部低速时钟工作 (2KHz)	R	0
0	OSCM	系统时钟选择 1: 选择内部高速时钟 16MHz 0: 选择内部低速时钟 2KHz	R/W	0

### CLKCFG1 寄存器

地址: 0XF62

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

Bit	Name	Description	Attribute	Reset
		1: 允许使能 0: 禁止使能		
2	T2CLKEN	TIMER2 的时钟使能, 使能后 TIMER2 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
1	T1CLKEN	TIMER1 的时钟使能, 使能后 TIMER1 才可以工作 1: 允许使能 0: 禁止使能	R/W	1
0	T0CLKEN	TIMER0 的时钟使能, 使能后 TIMER0 才可以工作 1: 允许使能 0: 禁止使能	R/W	1

### SIGCON 寄存器

地址: 0XF63

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
6:1	Reserved			
0	SIG	读取配置字中的校准信息 1: 读取配置字中的校准信息 0: 禁止读取配置字中的校准信息	R/W	0

## 4.2 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[6:0]和 HIRC 输出的关系如下

OSCTUNE[6:0]值	HIRC 实际输出频率（16M 为例）
OSCTUNE[X]+n	(16000+n*40)
.....	.....
OSCTUNE[X]+2	(16000+2*40) = 16080
OSCTUNE[X]+1	(16000+1*40) = 16040
OSCTUNE[X]	16000
OSCTUNE[X]-1	(16000-1*40) = 15960
OSCTUNE[X]-2	(16000-2*40) = 15920
.....	.....
OSCTUNE[X]-n	(16000-n*40)

#### OSCTUNE(HIRC 微调寄存器)

地址：0XF06

Bit	Name	Description	Attribute	Reset
7	Reserved			
6:0	OSCTUNE [6:0]	HIRC 微调值	R/W	0X00

#### OSCT

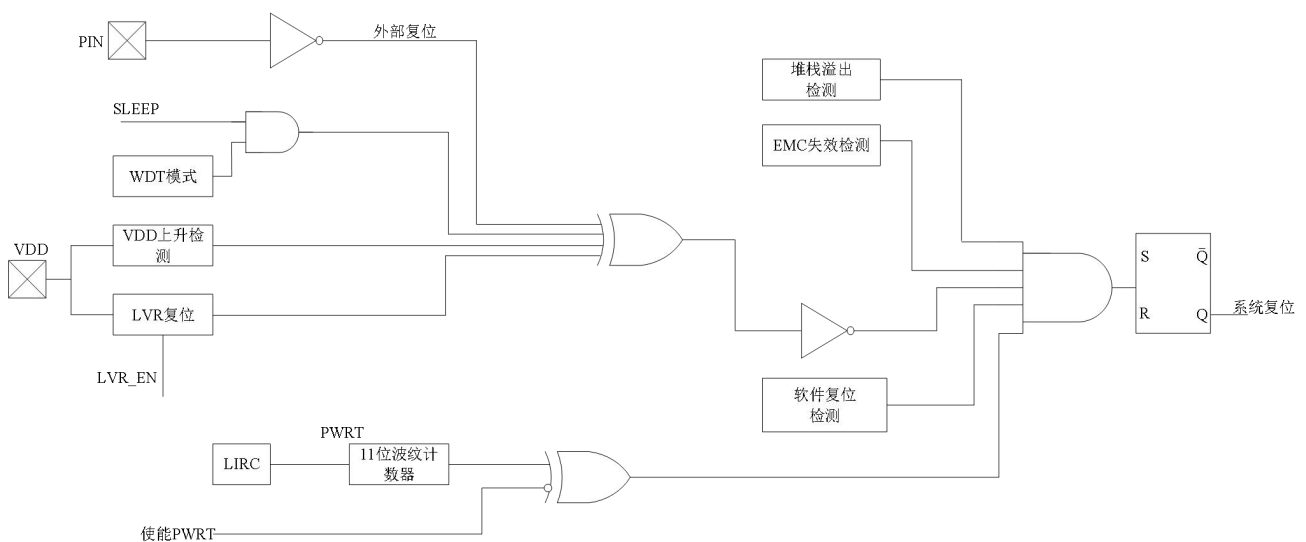
地址：0XF07

Bit	Name	Description	Attribute	Reset
7:1	Reserved			
0	OSCTEN	OSCTUNE 寄存器读写使能位 1: OSCTUNE 寄存器读写使能有效 0: OSCTUNE 寄存器读写使能无效	R/W	1

## 5 复位和电源电压检测

AD18F08 单片机能通过以下方式复位:

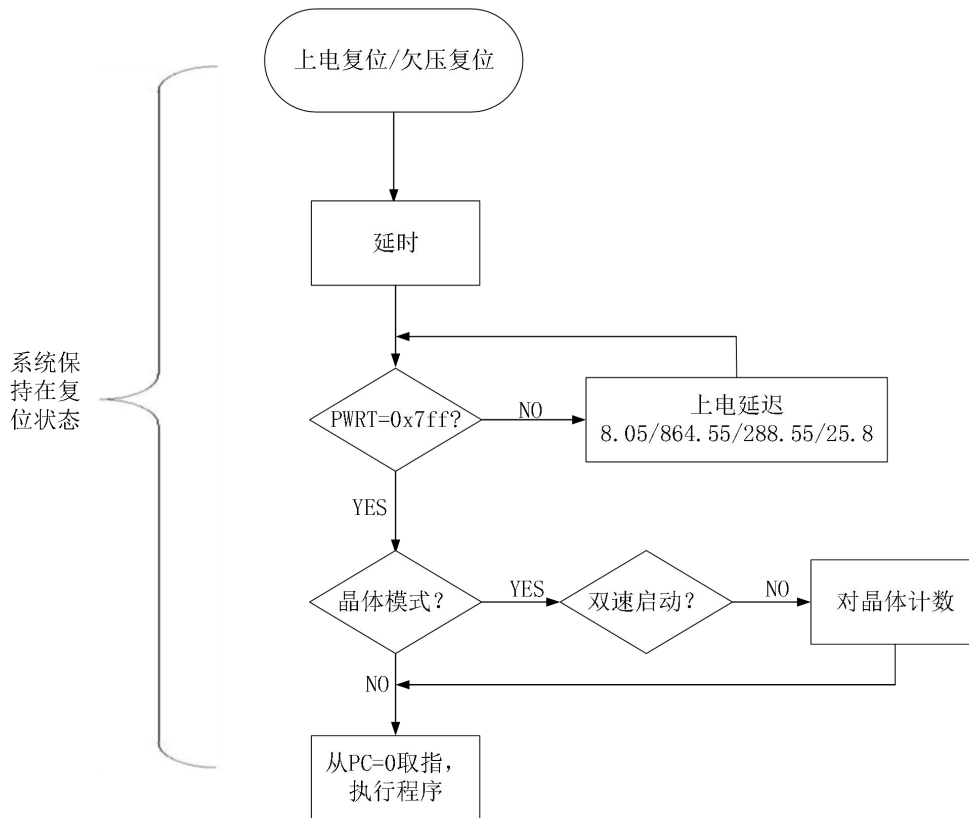
- 上电复位(POR)
- 低电压复位
- 看门狗复位
- 非法指令复位
- 软件复位
- EMC 复位
- 外部管脚低电平复位
- 堆栈溢出复位



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

低电压复位有 LVD[2:0]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 (25.8ms/288.55ms/864.55ms/8.05ms)延迟时间 (该延迟时间由配置字 PSUT<6:5>设置) (基于不同的振荡源和复位条件) 在 Power-on Reset (POR)，RSTn 或看门狗溢出复位。只要 PWRT 在运行，设备就一直保持的复位状态。VDD、温度和其他变化会影响其控制的设备延迟时间。

## 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 外部管脚复位

RSTn 引脚为低电平时复位，复位脚为 PB3 引脚，使能配置字 1 的 BIT2；

**注：使用外部引脚复位功能时，复位脚需接上拉电阻。**

## 5.7 堆栈溢出复位

设置配置字 1 的 BIT0,使能堆栈功能，溢出后复位

## 5.8 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。

## 5.9 LVD 检测

LVD 可通过 PCON[2:0]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	R/W	1

Bit	Name	Description	Attribute	Reset
		0: 只允许高优先级中断		
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
3	Reserved			
2:0	LVD [2:0]	LVD 电压选择 000: 2.4V 001: 2.7V 010: 3.0V 011: 3.3V 100: 3.6V 101: 3.8V 110: 4.2V 111: 4.3V	R/W	000

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

### 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	R/W	0



Bit	Name	Description	Attribute	Reset
		1: 使能 EMC 复位 0: 禁止 EMC 复位		
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/W	1
2	PD	掉电检测标志位 1: 通过上电或 CLRWDT 指令置 1 0: 通过执行 SLEEP 指令置 0	R/W	1
1	POR	上电复位状态位 1: 未发生上电复位 0: 已发生上电复位 (必须在上电复位后由软件置 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 (必须在欠压复位后由软件置 1)	R/W	0

## 5.10 /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

## 6 I/O 端口

PortA、PortB 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)设置。PA 和 PB 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 和 PB 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 和 PB 有相应的开漏控制位(开漏寄存器)来设置使能开漏来设置输出为开漏输出。

### 6.1 IO 工作模式

#### PORTA (Port 寄存器)

地址：0XF80

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PORTA [5:0]	输出模式下，PA 口输出控制 1: 输出高电平 0: 输出低电平	R/W	000000

#### PORTB (Port 寄存器)

地址：0XF81

Bit	Name	Description	Attribute	Reset
7:0	PORTB [7:0]	输出模式下，PB 口输出控制 1: 输出高电平 0: 输出低电平	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]	PA 口的数据方向控制选择 1: IO 的输入模式	R/W	111111

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

### TRISB (I/O 口方向控制寄存器)

地址: 0XF91

Bit	Name	Description	Attribute	Reset
7:0	TRISB [7:0]	PB 口的数据方向控制选择 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: 输入模式下, 读取当前 IO 状态(读 SMT 输出); 写: 写入 PORTA 寄存器;	R/W	000000

### PINB (数据锁存寄存器)

地址: 0XF89

Bit	Name	Description	Attribute	Reset
7:0	PINB [7:0]	读: 输入模式下, 读取当前 IO 状态(读 SMT 输出); 写: 写入 PORTB 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]	PA 口下拉使能位选择	R/W	0x3F

		1: 关闭内部下拉 0: 使能内部下拉		
--	--	------------------------	--	--

**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]	PA 口上拉使能位选择 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]	PA 口的开漏选择 1: 使能内部开漏 0: 关闭内部开漏	R/W	00000

**PBPD(I/O 下拉控制寄存器)**

地址: 0XF8D

Bit	Name	Description	Attribute	Reset
7:0	PBnPD [7:0]	PB 口下拉使能位选择 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]	PB 口上拉使能位选择 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	PBnOD [7:0]	PB 口的开漏选择 1:使能内部开漏 0:关闭内部开漏	R/W	0x00

### PAINTMASK (Port A 端口电平变化中断掩膜位)

地址: 0XF60

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	PINTMASKA [5:0]	PA 口的中断控制位 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	000000

### PBINTMASK (Port B 端口电平变化中断掩膜位)

地址: 0XF5F

Bit	Name	Description	Attribute	Reset
7:0	PINTMASKB [7:0]	PB 口的中断控制位 1: 使能 PortB 端口电平变化中断允许位 0: 禁止 PortB 端口电平变化中断允许位	R/W	0x00

## 6.3 模拟 IO 寄存器

### ANASEL0 (IO 的模拟通道寄存器)

地址: 0XF54

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

7:6	Reserved			
5:0	PAnAEN [5:0]	PA 口的模拟通道选择 1: PA 口打开模拟通道 0: PA 口关闭模拟通道	R/W	000000

### ANASEL1 (IO 的模拟通道寄存器)

地址: 0XF53

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

### SMTA (PA 口施密特寄存器)

地址: 0XF98

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5:0	SMTA [5:0]	PA 口施密特选择 1: $0.3 \times VDD / 0.7 \times VDD$ (IO 低电平变为高电平时为 $0.7 \times VDD$ , 高电平变为低电平时为 $0.3 \times VDD$ ) 0: $0.2 \times VDD / 0.4 \times VDD$ (IO 低电平变为高电平时为 $0.4 \times VDD$ , 高电平变为低电平时为 $0.2 \times VDD$ )	R/W	0X3F

### SMTB (PB 口施密特寄存器)

地址: 0XF99

Bit	Name	Description	Attribute	Reset
7:0	SMTB [7:0]	PB 口施密特选择 1: $0.3 \times VDD / 0.7 \times VDD$ (IO 低电平变为高电平时为 $0.7 \times VDD$ , 高电平变为低电平时为 $0.3 \times VDD$ )	R/W	0XFF

		0:0.2*VDD/0.4V*DD (I0 低电平变为高电平时为 0.4*VDD, 高电平变为低电平时为 0.2*VDD)		
--	--	---	--	--

**CURCON(寄存器读写)**

地址: 0XF55

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3	PBCUR	PB 口的电流驱动(SINK 电流) 1: 20mA 0: 3.5mA SOURCE 电流为 10mA	R/W	0
2	PACUR	PA 口的电流驱动(SINK 电流) 1: 20mA 0: 3.5mA SOURCE 电流为 10mA	R/W	0
1	PA2CUR	PA2 的电流驱动(SINK 电流) 1: 60mA 0: 20mA SOURCE 电流为 10mA	R/W	0
0	PA1CUR	PA1 的电流驱动(SINK 电流) 1: 60mA 0: 20mA SOURCE 电流为 10mA	R/W	0



## 7 定时器

### 7.1 Timer0/Prescaler/BUZZER/PWM

#### 7.1.1. Timer0 计数/定时

TMR0 是一个 8 位定时/计数器寄存器，Timer0 的时钟源可以取值于指令周期、外部实时钟（T0CKI pin）、内部 OSC2K 时钟源、比较器的输出，使用外部时钟需要设置 T0CON 的 T0CS0、T0CS1 共同决定

#### 7.1.2. 使用内部时钟: 定时模式

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

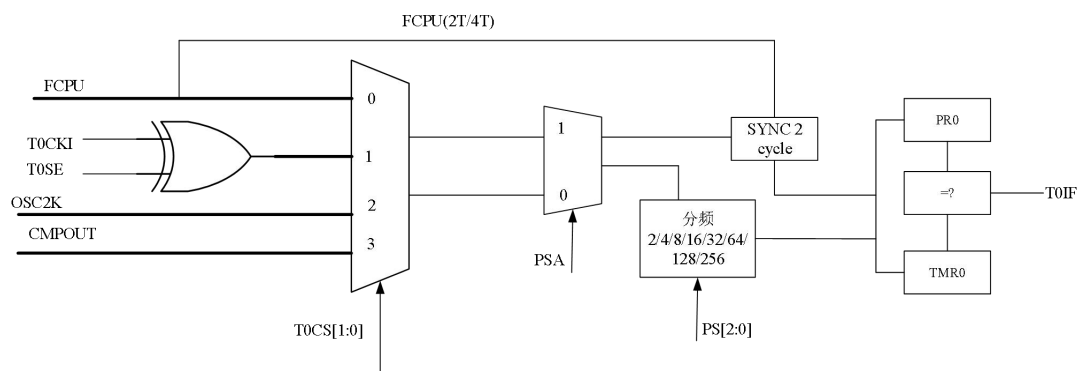
#### 7.1.3. 使用外部时钟/内部 OSC2K 时钟/: 计数模式

通过 T0CKI 上升或下降沿、OSC2K 时钟，触发 Timer0 寄存器的增加，T0CKI 由 T0SE 位 (T0CON<4>) 决定上升下降触发。

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

#### 7.1.4. Prescaler（预置器）

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



Timer0 结构图

注：4T 模式下定时： $Time = (PR0+1) * \{PS2:PS0\} * 4 / F_{sys}$ ；//TMR0 的初值为 0 时  
 $Time = (0XFF - [TMR0] + 1) * \{PS2:PS0\} * 4 / F_{sys}$ ；//TMR0 设定初值  
 2T 模式下定时： $Time = (PR0+1) * \{PS2:PS0\} * 2 / F_{sys}$ ；//TMR0 的初值为 0 时  
 $Time = (0XFF - [TMR0] + 1) * \{PS2:PS0\} * 2 / F_{sys}$ ；//TMR0 设定初值

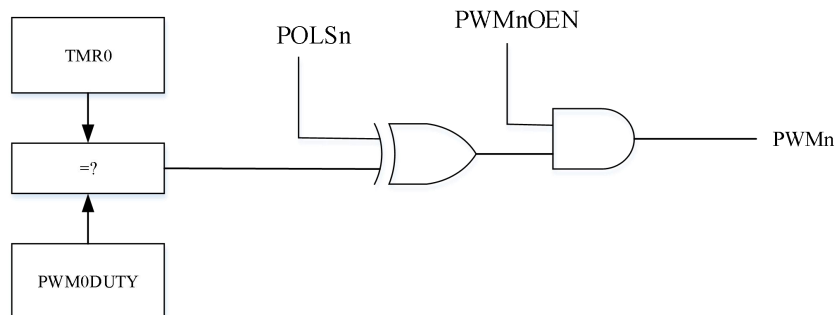
### 7.1.5. BUZZER (BUZZER 输出)

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 4 路 PWM

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



PWM 原理框图 (n=0)

注：PWM0 的输出复用的 IO 有：PA0、PA2、PA4、PB3

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

**T0CON (TMR0 控制寄存器)**

地址: 0XFD7

Bit	Name	Description	Attribute	Reset
7	T0OUT	PA4 溢出输出 BUZZER 0: 禁止 BUZZER 模式 1: 启动 BUZZER 模式	R/W	0
6:5	T0CS [1:0]	TMR0 的时钟源选择 00: TMR0 时钟源为 CPU 运行时钟 01: TMR0 时钟源为 T0CKI(PA5) 10: TMR0 时钟源为内部 2K 11: TMR0 时钟源是 CMPOUT 输出	R/W	11
4	T0SE	TMR0 触发方式控制位 1: T0CKI 脚下降沿触发计数 0: T0CKI 脚上升沿触发计数	R/W	1
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, T0CKI 的时钟选择必须小于  $F_{CPU}/8$ ;

## PWM0CON(PWM 的控制寄存器)

地址: 0XFD3

Bit	Name	Description	Attribute	Reset
7	POLS_03	PWM0 通过 PB3 输出的极性 1: 反向输出 0: 正向输出	R/W	0
6	POLS_02	PWM0 通过 PA4 输出的极性 1: 反向输出 0: 正向输出	R/W	0
5	POLS_01	PWM0 通过 PA2 输出的极性 1: 反向输出 0: 正向输出	R/W	0
4	POLS_00	PWM0 通过 PA0 输出的极性 1: 反向输出 0: 正向输出	R/W	0
3	PWM03OEN	PWM0 输出使能,通过 PB3 输出 1: 使能 PWM0 输出 0: 禁止 PWM0 输出	R/W	0
2	PWM02OEN	PWM0 输出使能, 通过 PA4 输出 1: 使能 PWM0 输出 0: 禁止 PWM0 输出	R/W	0
1	PWM01OEN	PWM0 输出使能, 通过 PA2 输出 1: 使能 PWM0 输出 0: 禁止 PWM0 输出	R/W	0
0	PWM00OEN	PWM0 输出使能, 通过 PA0 输出 1: 使能 PWM0 输出 0: 禁止 PWM0 输出	R/W	0

## PWM1CON0

地址: 0XFBE

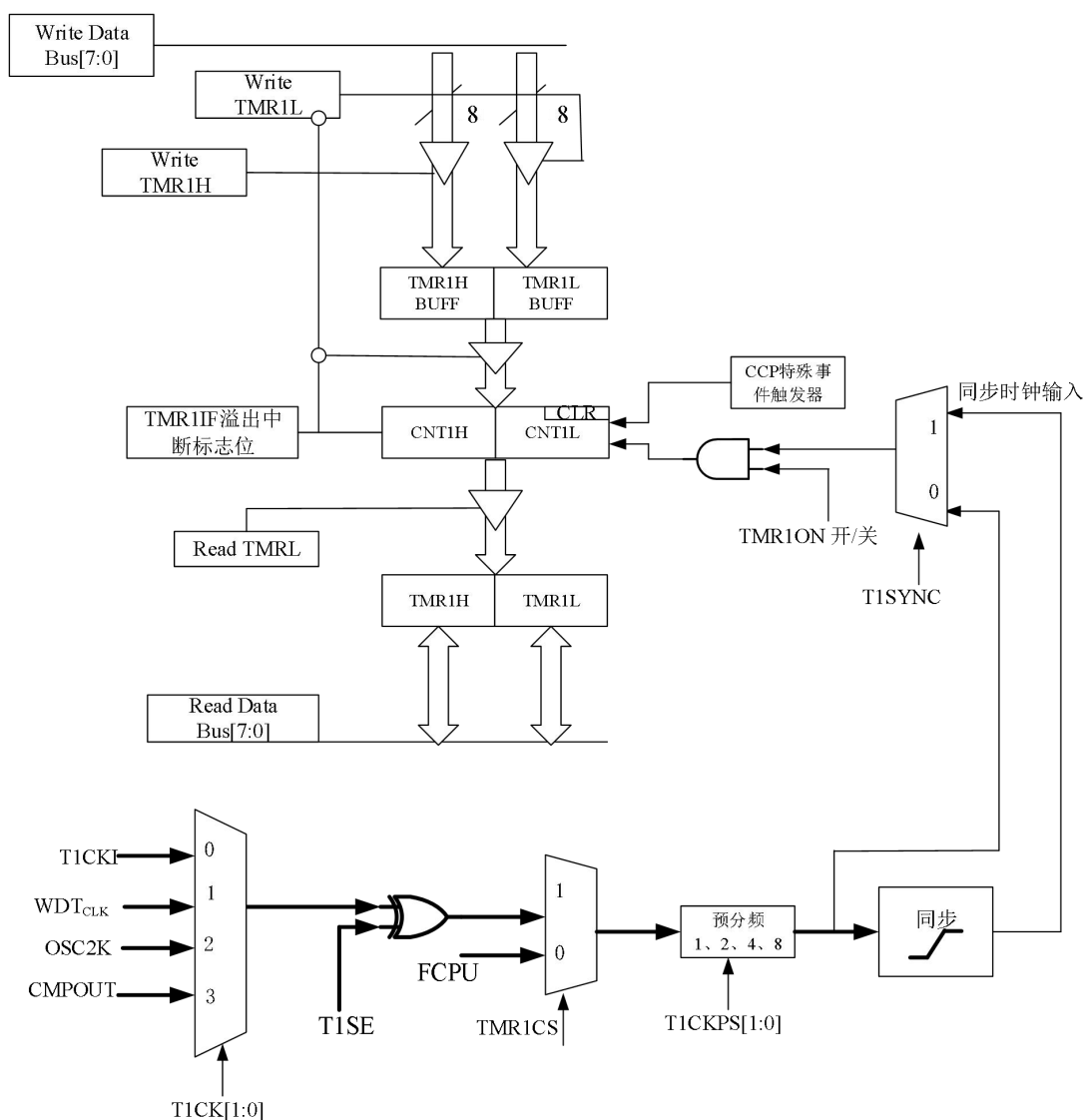
Bit	Name	Description	Attribute	Reset
7	-	-	R/W	0
6	CMPFLT	CMPOUT 为故障输入, 当从 1 变为 0 后, 禁止 PWM 输出 1: 使能 CMPOUT 作为故障输入 0: 禁止 CMPOUT 作为故障输入	R/W	0
5	ASTART	发生故障后(ACLOSE=1), 当使能的故障都解除后, 自动启动 PWM 输出 1:使能启动 PWM 输出 0:禁止启动 PWM 输出	R/W	0
4	ACLOSE	发生故障后, 自动关闭 PWM 输出 1:使能关闭 PWM 输出 0:禁止关闭 PWM 输出	R/W	0
3	IOFLT	IO 为故障输入, 当从 1 变为 0 后, 禁止 PWM 输出 1: 使能 IO 作为故障输入 0: 禁止 IO 作为故障输入  注: 故障 IO 映射为 PA0;	R/W	0
2:1	PWMADPOS	PWM 跳变沿使能 ADC 采集 1x: 中心对齐模式: TMR2==PR2 或者 TMR2==12'H0 时, TMR3==PR3 或者 TMR3==12'H0 时, 启动 PWMADDLY 计数器, 当计数器为 0 时启动 ADC 采样 01:PWM 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 (下降沿, PLOS 默认选 0 时) 00: PWM 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。 沿跳变时 ADC 才会采集, 如果沿不跳变 ADC 不会采集(上升沿, POLS 默认选 0 时)	R/W	0
0	PWM10ADEN	使能 PWM10 输出跳变时, 自动启动 ADC 采集功能 1: 使能 ADC 采集 0: 禁止 ADC 采集	R/W	0

## 7.2 TIMER1 16 位定时/计数器

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 由计数值从 0XFFFF 变为 0X0000 时产生时钟溢出中断，并自动重载写入缓冲区数值。TIMER1 的时钟源可以是  $F_{CPU}$ 、内部 WDT 时钟、外部时钟源(TICKI pin)、内部 2K 时钟、CMP 输出。

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



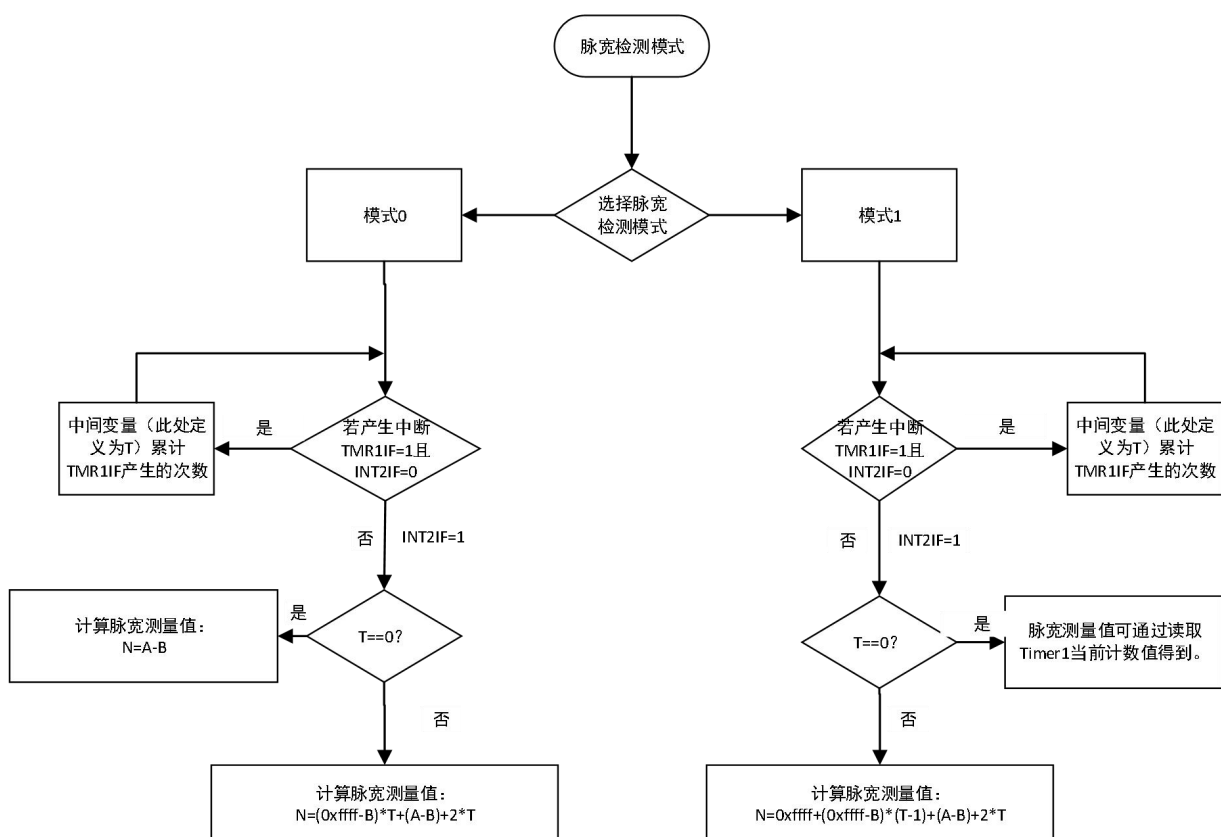
注：CNT1H 和 CNT1L 是 TIMER1 的内部寄存器。

4T 模式下定时： $Time = (0xFFFF - [TMR1] + 1) * \{TICKPS1; TICKPS0\} * 4 / F_{sys}$ ；//TMR1 的初值为设定值

2T 模式下定时： $Time = (0xFFFF - [TMR1] + 1) * \{TICKPS1; TICKPS0\} * 2 / F_{sys}$ ；//TMR1 的初值为设定值

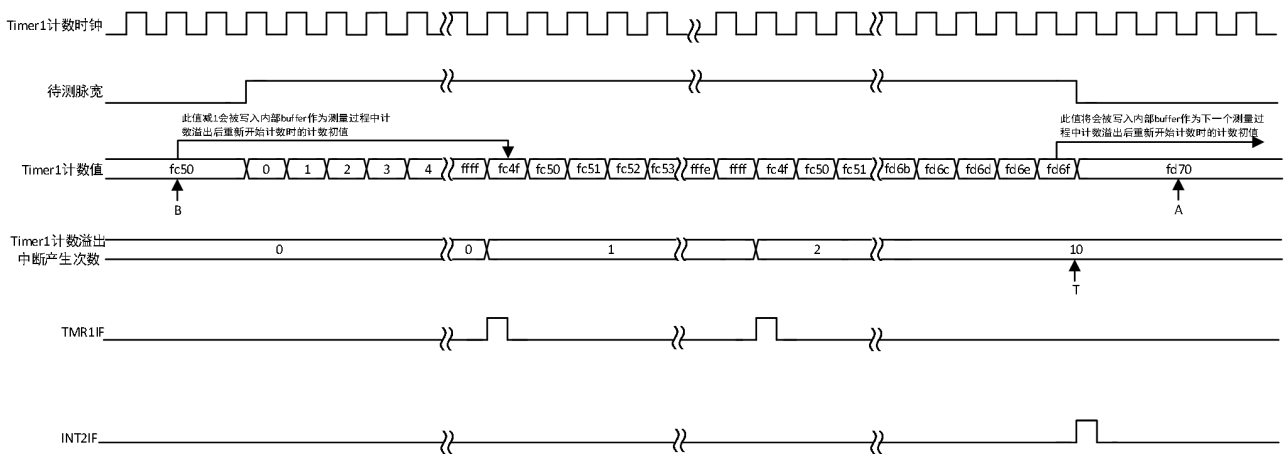
Timer1 支持两种脉宽检测模式。脉宽测量过程结束将产生 INT2 中断，可由此判断脉宽测量过程结束与否。下图为脉宽测量模式的参考应用流程。其中，A 表示脉宽测量过程结束后的 16 位 Timer1 计数值，B 表示脉宽测量过程开始前的 16 位 Timer1 计数值，T 表示脉宽测量过程中 Timer1 计数溢出中断产生的次数。

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

### T1CON0 (Timer1 控制寄存器)

地址: 0XFCF

Bit	Name	Description	Attribute	Reset
7:6	T1CK [1:0]	T1 的时钟选择 00: T1CKI 作为时钟(PB5) 01: WDT <sub>CLK</sub> (WDT 的溢出时间) 10: 内部 2K 作为时钟 11: CMPOUT 输出	R/W	00
5:4	T1CKPS [1:0]	T1 的输入时钟分频	R/W	00



Bit	Name	Description	Attribute	Reset
		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		
3	T1SE	TIMER1 的时钟异或输入选项	R/W	0
2	T1SYNC	1: 使用 FCPU 同步分频后时钟作为 TIMER1 时钟（注意不支持选择内部时钟，同时 1:1 分频，并且 T1SYNC 为 1） 0: 使用选择分频时钟作为 TIMER1 时钟	R/W	0
1	TMR1CS	1: 选择 TICK[1:0]作为外设 TIMER1 的时钟 0: 选择 FCPU 时钟外设 TIMER1 的时钟	R/W	0
0	TMR1ON	1: 使能 Timer1 定时计数器 0: 关闭 Timer1 定时计数器	R/W	0

注：

- 1、在 4T 和 2T 模式下，TICKI 的时钟选择必须小于  $F_{CPU}/8$ ;
- 2、TMR1 寄存器在写寄存器时，必须先写高字节再写低字节；

### T1CON1 (Timer1 控制寄存器)

地址：0XFCE

Bit	Name	Description	Attribute	Reset
7	INT2EDGE	外部中断 2 边沿选择位 0: 上升沿触发中断 1: 下降沿触发中断	R/W	0
6	INT1EDGE	外部中断 1 边沿选择位 0: 上升沿触发中断 1: 下降沿触发中断	R/W	0
5	INT0EDGE	外部中断 0 边沿选择位 0: 上升沿触发中断 1: 下降沿触发中断	R/W	0
4	T1EDEG	在 T1M[1:0]在脉宽测量模式时 0: 在下降沿启动计数，上升沿停止计数 1: 在上升沿启动计数，下降沿停止计数	R/W	0

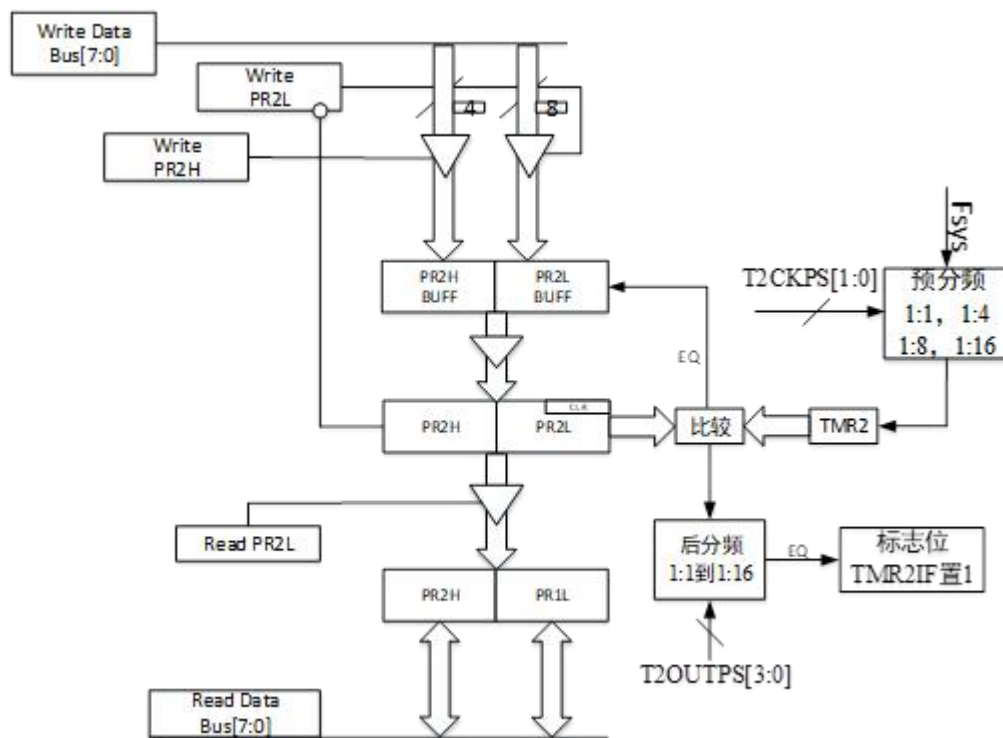
Bit	Name	Description	Attribute	Reset
3:2	T1CH1:T1CH0	TIMER1 脉宽信号输入选择 00: T1CH0 作为脉宽检测输入信号(PB6) 01: T1CH1 作为脉宽检测输入信号(PB7) 10: T1CH2 作为脉宽检测输入信号(PA4) 11: TMR2CLK 作为脉宽检测输入信号(配置 T2CON<2>=1)	R/W	00
1:0	T1M[1:0]	TIMER1 工作模式 00: TIMER1 工作在普通模式, 溢出时间产生中断标志位。 01: 脉宽测量模式 0, 测量上升沿与下降沿之间的时间, T1EDGE=1 时, 在脉冲的上升沿开始计数, 脉冲下降沿停止计数, 并在脉冲下降沿沿触 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, 在脉冲上升沿触发 INT2 中断。产生 INT2 中断标志后, 可以直接读取 TIMER1 内容, 就可以得到测量值。 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 <b>复位</b> 从 0x0000 开始计数, 并在下降沿沿触 INT2 中断。T1EDGE=0 时, 在脉冲的下降沿 TIMER1 数据寄存器发生 <b>复位</b> 从 0x0000 开始计数, 在脉冲的上升沿触发 INT2 中断。产生 INT2 中断标志后, 可以直接读取 TIMER1 内容, 就可以得到测量值。 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: 1. T1 BUZZER:在 T1M[1:0]=11 且 TMR1 计数到 0xFFFF 后, T1BUZZER 输出通过 PA5 引脚。**

## 7.3 TIMER2 定时器

Timer2 定时器模块具有以下特征：

- 12 位定时器和周期寄存器（分别为 TMR2L、TMR2H 和 PR2L、PR2H）
- 可读写（以上四个寄存器）
- 可软件编程的预分频器（分频比为 1:1、1:4、1:8、1:16）
- 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 F<sub>sys</sub> 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



T2 结构图

PR2L（timer2 的周期寄存器）

地址：0XFCB

Bit	Name	Description	Attribute	Reset
7:0	PR2[7:0]	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	1111

**TMR2L (Timer2 的低八位寄存器)**

地址:0XFCD

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

**TMR2H (Timer2 的高八位寄存器)**

地址: 0XFCC

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	TMR2H [11:8]	Timer2 定时/计数器的高 4 位	R/W	0000

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

**T2CON(timer2 控制寄存器)**

地址: 0XFC9

Bit	Name	Description	Attribute	Reset
7	PWM2CAEN	Timer2 中心对齐模式使能 1: 使能 Timer2 中心对齐模式。 0: 禁止 Timer2 中心对齐模式。	R/W	0
6:3	T2OUTPS [3:0]	Timer2 输出后分频选择位 0000: 1:1 后分频值 0001: 1:2 后分频值 0010: 1:3 后分频值 0011: 1:4 后分频值 0100: 1:5 后分频值	R/W	0000

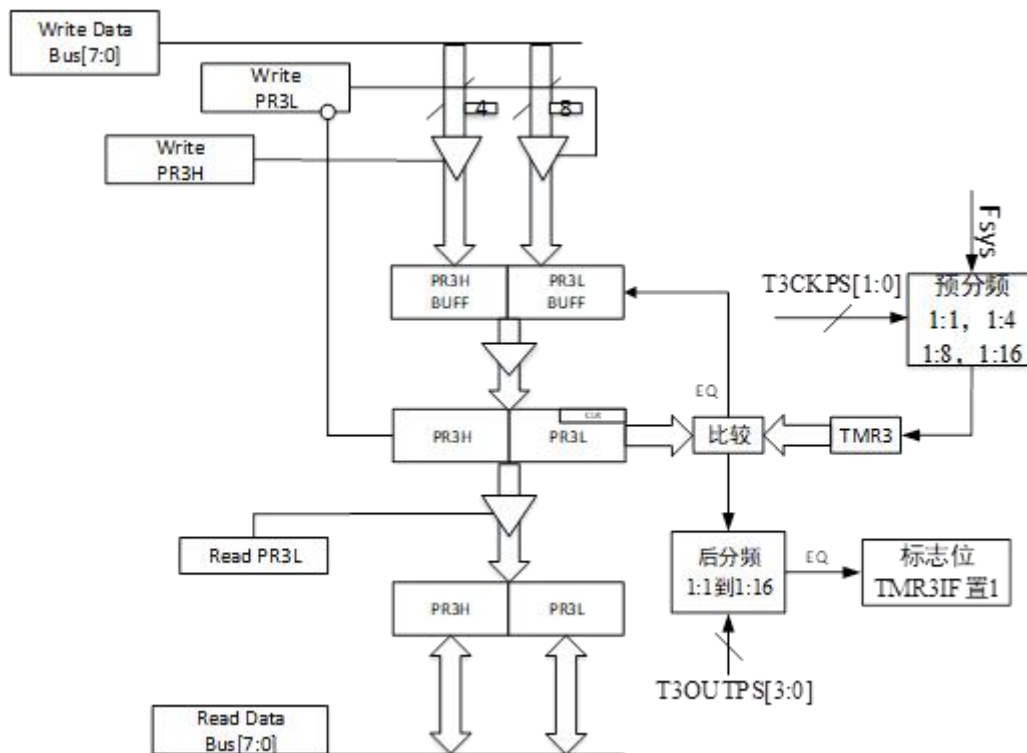
Bit	Name	Description	Attribute	Reset
		0101: 1:6 后分频值 0110: 1:7 后分频值 0111: 1:8 后分频值 1000: 1:9 后分频值 1001: 1:10 后分频值 1010: 1:11 后分频值 1011: 1:12 后分频值 1100: 1:13 后分频值 1101: 1:14 后分频值 1110: 1:15 后分频值 1111: 1:16 后分频值		
2	TMR2ON	Timer2 使能位 1: 使能 Timer2 0: 禁止 Timer2	R/W	0
1:0	T2CKPS [1:0]	Timer2 时钟预分频选择位 00: 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

## 7.4 TIMER3 定时器和 4 路 12 位 PWM

### 7.4.1. TIMER3 定时器

Timer3 定时器模块具有以下特征：

- 12 位定时器和周期寄存器（分别为 TMR3L、TMR3H 和 PR3L、PR3H）
- 可读写（以上四个寄存器）
- 可软件编程的预分频器（分频比为 1:1、1:4、1:8、1:16）
- 可软件编程的后分频器（分频比为 1:1 至 1:16）
- 当 TMR3（TMR3H, TMR3L）与 PR3（PR3H, PR3L）匹配时产生中断
- 采用系统时钟 F<sub>sys</sub> 控制
- 当 TMR3 运行时，写入 PR3 会先写入 PR3BUFFER 中，当 TMR3 与 PR3 相等时，PR3BUFFER 更新到 PR3 中，可以用于调频



T3 结构图

PR3L（timer3 的周期寄存器）

地址：0XFB8

Bit	Name	Description	Attribute	Reset
7:0	PR3 [7:0]	周期寄存器的低 8 位	R/W	0XFF

### PR3H (timer3 的周期寄存器)

地址: 0XFB9

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PR3 [11:8]	周期寄存器的高 4 位	R/W	1111

### 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: 在使能 TMR3ON 之前, TMR3 寄存器的高字节与低字节任意写, TMR3ON 使能之后, TMR3 寄存器读写时, 必须先写高字节再写低字节;**

### T3CON(timer3 控制寄存器)

地址: 0XFBC

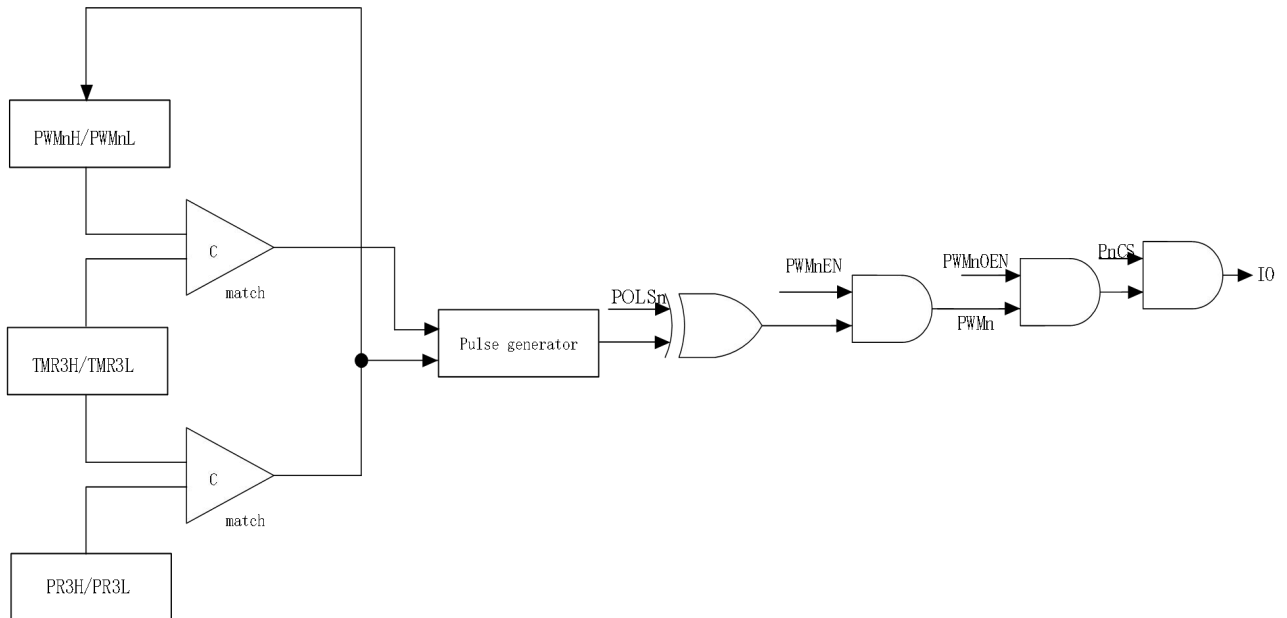
Bit	Name	Description	Attribute	Reset
7	PWM3CAEN	Timer3 中心对齐模式使能 1: 允许使能 Timer3 中心对齐模式。 0: 禁止使能 Timer3 中心对齐模式。	R/W	0
6:3	T3OUTPS [3:0]	Timer3 输出后分频选择位 0000: 1:1 后分频值 0001: 1:2 后分频值 0010: 1:3 后分频值	R/W	0

Bit	Name	Description	Attribute	Reset
		0011: 1:4 后分频值 0100: 1:5 后分频值 0101: 1:6 后分频值 0110: 1:7 后分频值 0111: 1:8 后分频值 1000: 1:9 后分频值 1001: 1:10 后分频值 1010: 1:11 后分频值 1011: 1:12 后分频值 1100: 1:13 后分频值 1101: 1:14 后分频值 1110: 1:15 后分频值 1111: 1:16 后分频值		
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

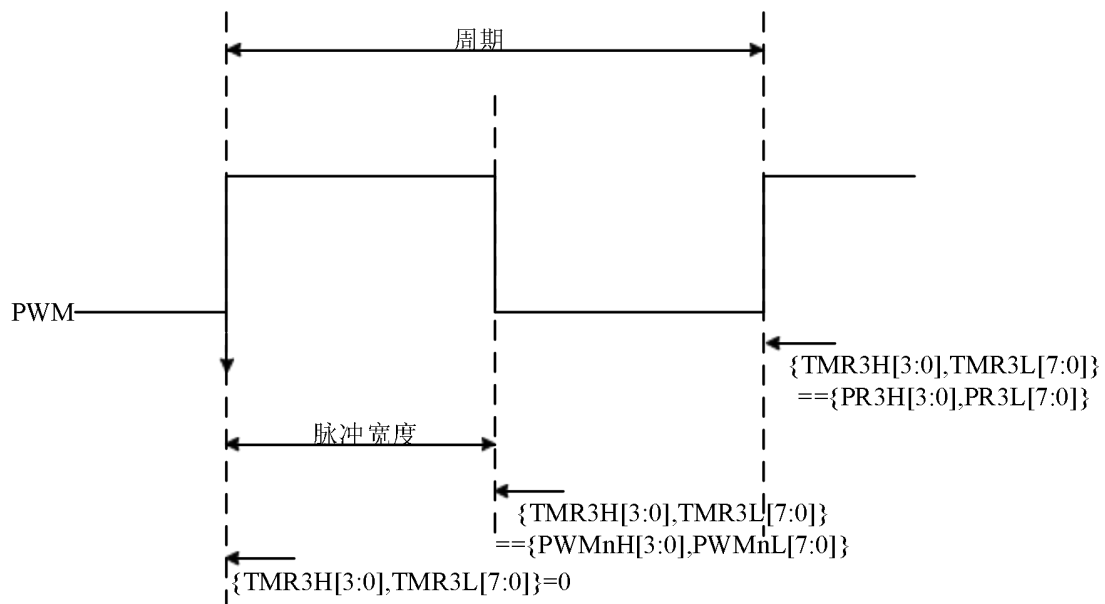
PWM 原理框图如下图所示。



PWM 原理框图 (n=2/3/4/5)

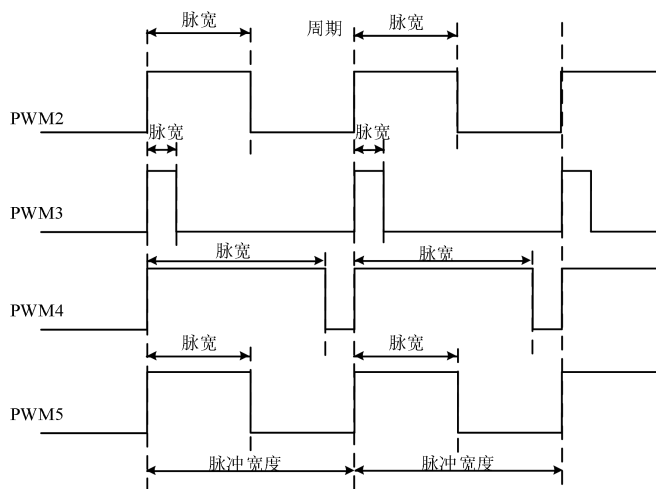
### 7.4.3.4 路 PWM 的输出

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<sub>sys</sub> (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

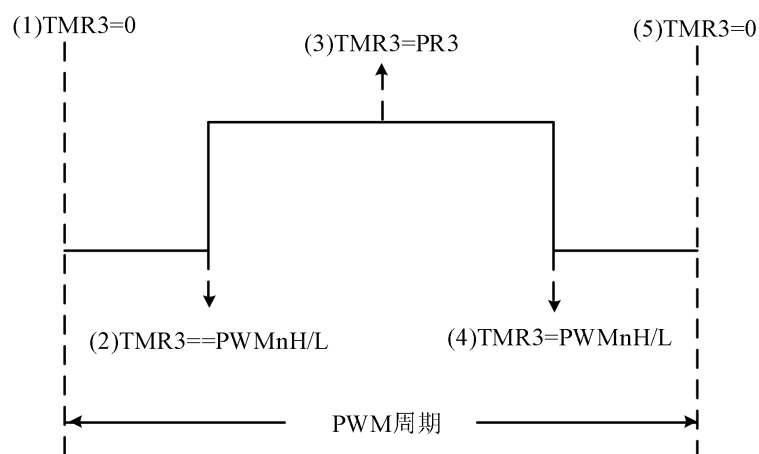
PWM 频率来自系统工作频率。系统工作频率的任何改变将使 PWM 频率的改变。

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 T3CON 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  $2 \cdot \{ (PR3H, PR3L) + 1 \} / F_{sys}$  (系统工作时钟) \* (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 1: 使能 PWM3 映射的 IO 输出 0: 禁止 PWM3 映射的 IO 输出	R/W	0
4	PWM2OEN	PWM2 映射的 IO 输出使能 1: 使能 PWM2 映射的 IO 输出 0: 禁止 PWM2 映射的 IO 输出	R/W	0
3	PWM5EN	PWM5 功能使能 1: 使能 PWM5 功能 0: 禁止 PWM5 功能	R/W	0
2	PWM4EN	PWM4 功能使能 1: 使能 PWM4 功能 0: 禁止 PWM4 功能	R/W	0
1	PWM3EN	PWM3 功能使能 1: 使能 PWM3 功能 0: 禁止 PWM3 功能	R/W	0
0	PWM2EN	PWM2 功能使能 1: 使能 PWM2 功能 0: 禁止 PWM2 功能	R/W	0

**PWM2CON1(PWM 的控制寄存器)**

地址: 0XFAE

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

**PWM2CON3(PWM 的控制寄存器)**

地址: 0XFAD

Bit	Name	Description	Attribute	Reset
7:6	P5CS [1:0]	PWM5 映射的 IO 输出 00:PA2 输出 01:PB7 输出 10:PB3 输出 11:PB0 输出	R/W	0
5:4	P4CS [1:0]	PWM4 对映射的 IO 输出 00:PA3 输出 01:PB6 输出 10:PB2 输出 11:PA0 输出	R/W	0

Bit	Name	Description	Attribute	Reset
3:2	P3CS [1:0]	PWM3 映射的 IO 输出 00:PB1 输出 01:PA4 输出 10:PB5 输出 11:保留	R/W	0
1:0	P2CS [1:0]	PWM2 映射的 IO 输出 00:PA1 输出 01:PA5 输出 10:PB4 输出 11:保留	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	0000

#### 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
-----	------	-------------	-----------	-------

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	PWM3H [3:0]	PWM3 高位占空比控制	R/W	0000

#### 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	0000

#### 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	0000

## 7.5 看门狗定时器 (WDT)

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

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

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

在没有预置器时看门狗的溢出约为 128/256/384/640ms(或 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<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 (WDTON<3:1>)进行分频, 因此最长的看门狗溢出周期为 66.56 秒。

CLRWDT 指令能使 WDT 和预置器清零, 防止看门狗超时, 如果超时 MCU 将复位。

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: 0XF58

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



Bit	Name	Description	Attribute	Reset
		1: 使能 WDT 0: 禁止 WDT		

## 8 捕获/比较/PWM 模块

捕获/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。捕获/比较的寄存器(CCPR1) 由两个 8 位寄存器组成：CCPR1L (低字节) 和 CCPR1H (高字节)。CCPCON 寄存器控制 CCP1 的操作。捕获和 Timer1 与 Timer2 均相关；比较和 Timer1 相关，比较匹配将产生特殊事件触发信号，该信号会使 TMR1H 和 TMR1L 寄存器清零。PWM 和 Timer2 相关。Timer3 中的 PWM 和 CCP 中 PWM 可以映射到 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	Reserved			
3:0	CCPR1LH [11:8]	CCPR1 寄存器低字节中的高 4 字节, 用于 PWM 的高 4 位	R/W	0X00

## CCPCON 寄存器

地址: 0XFBF

Bit	Name	Description	Attribute	Reset
7:4	CCPR1CH [3:0]	CCP1 捕获触发信号选择 0000: 外部 CCPCH0 引脚输入信号(PA4) 0001: 外部 CCPCH1 引脚输入信号(PA5) 0010: 外部 CCPCH2 引脚输入信号(PB7) 0011: 外部 CCPCH3 引脚输入信号(PB6) 0100: 外部 CCPCH4 引脚输入信号(PA3) 0101: 外部 CCPCH5 引脚输入信号(PA2) 0110: 外部 CCPCH6 引脚输入信号(PA1) 0111: 外部 CCPCH7 引脚输入信号(PA0) 1000: 外部 CCPCH8 引脚输入信号(PB0) 1001: CMPOUT 的输出 1010: 内部 2K 1011: 外部 CCPCH9 引脚输入信号(PB1) 1100: 外部 CCPCH10 引脚输入信号(PB5) 1101: 外部 CCPCH11 引脚输入信号(PB4) 1110: 外部 CCPCH12 引脚输入信号(PB3) 1111: 外部 CCPCH13 引脚输入信号(PB2)	R/W	0X00
3:0	CCPR1M [3:0]	CCP1 模式选择位 0000: 禁止捕捉/比较/PWM 0100: 比较模式, 选择 CCPR1 匹配时将输出置为高电平 (CCPIF 位置 1), 输出引脚为 PB5, TIMER1 溢出时, PB5 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 PB5, TIMER1 溢出时, PB5 为高电平 0110: 比较模式, 选择 CCPR1 匹配时将产生软件中断	R/W	0000

Bit	Name	Description	Attribute	Reset
		(CCPIF 位置 1, 而 PB5 引脚不受影响); 并启动 ADC 采集 (如果 ADCON=1) 0111: 比较模式, 选择 CCPR1 触发特殊事件 (CCPIF 位置 1, PB5 引脚不受影响); CCP1 清零 Timer1; 并启动 ADC 采集 (如果 ADCON=1) 1000: 捕捉模式, 在每个下降沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1001: 捕捉模式, 在每个上升沿发生, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1010: 捕捉模式, 在每 4 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1011: 捕捉模式, 在每 16 个上升沿发生一次, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1101: 捕捉模式, 上升沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF 1110: 捕捉模式, 下降沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCPIF 1111: 捕捉模式, 上升沿, 捕捉值存入 CCPR1 寄存器, 不产生中断标志 CCPIF 0010: PWM 模式, 启动 CCP PWM 模式		

## 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 CCPCON[3:0]进行配置:

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

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

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

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

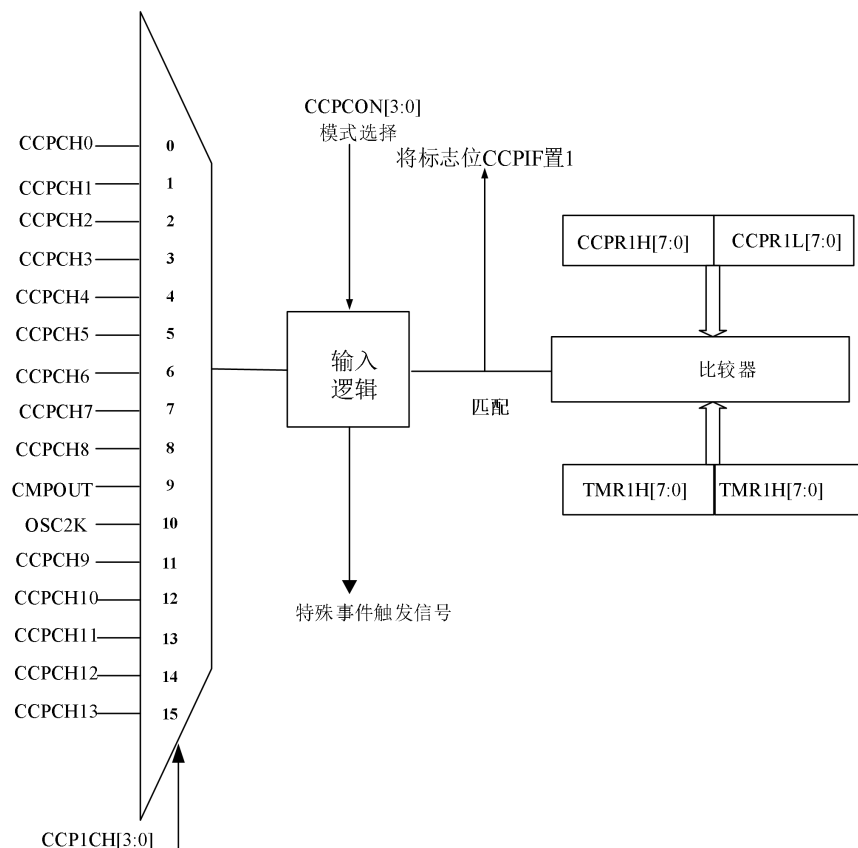
1100: 捕捉模式, 下降沿, 复位 TIMER1, 捕捉值存入 CCPR1 寄存器, 并产生中断标志 CCPIF

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

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

进行捕捉后, 中断请求标志位 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 CCP1CHI[3:0]进行配置:

- ✧ 0000: 外部 CCPCH0 引脚输入信号(PA4)
- ✧ 0001: 外部 CCPCH1 引脚输入信号(PA5)
- ✧ 0010: 外部 CCPCH2 引脚输入信号(PB7)
- ✧ 0011: 外部 CCPCH3 引脚输入信号(PB6)
- ✧ 0100: 外部 CCPCH4 引脚输入信号(PA3)
- ✧ 0101: 外部 CCPCH5 引脚输入信号(PA2)
- ✧ 0110: 外部 CCPCH6 引脚输入信号(PA1)
- ✧ 0111: 外部 CCPCH7 引脚输入信号(PA0)
- ✧ 1000: 外部 CCPCH8 引脚输入信号(PB0)
- ✧ 1001: CMPOUT 的输出
- ✧ 1010: 内部 2K
- ✧ 1011: 外部 CCPCH9 引脚输入信号(PB1)
- ✧ 1100: 外部 CCPCH10 引脚输入信号(PB5)
- ✧ 1101: 外部 CCPCH11 引脚输入信号(PB4)
- ✧ 1110: 外部 CCPCH12 引脚输入信号(PB3)
- ✧ 1111: 外部 CCPCH13 引脚输入信号(PB2)



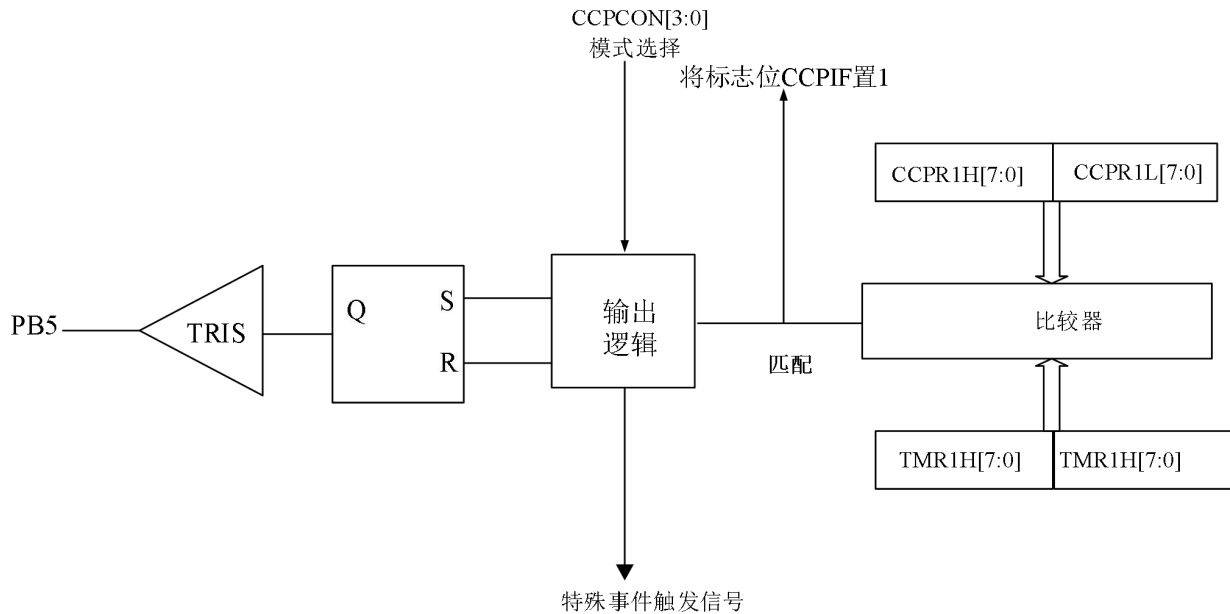
CCP1 捕捉模式工作原理图

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

## 8.2 比较模式

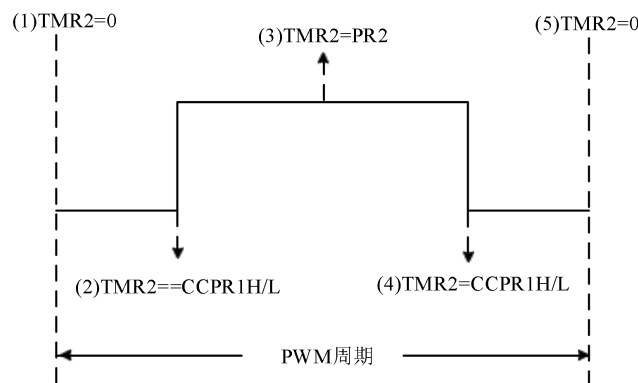
在比较模式下，CCPR1 寄存器的 16 位值不断与 TMR1 寄存器的值进行比较。事件定义如下，由 CCPCON[3:0] 进行配置：

- ✧ 0100：比较模式，选择 CCPR1 匹配时将输出置为高电平（CCPIF 位置 1），输出引脚为 PB5；TIMER1 溢出时，PB5 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 PB5；TIMER1 溢出时，PB5 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 PB5 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，PB5 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

### 8.2.1. PWM 中心对齐模式



PWM 中心对齐示意图

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

- (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  $2 \cdot \{ (PR2H, PR2L) + 1 \} / F_{sys} (\text{系统工作时钟}) \cdot (TMR2 \text{ 的预分频值})$ 。

注意: 普通 PWM 模式切换中心对齐模式需要将 PWM 关闭后重新打开。

### 8.3 故障刹车功能

3 路 PWM 均支持故障刹车功能。一旦发生故障刹车事件，且只要故障条件一直存在，PWM 输出引脚将根据其设置一直输出预定状态。故障刹车事件可以为下列条件之一：

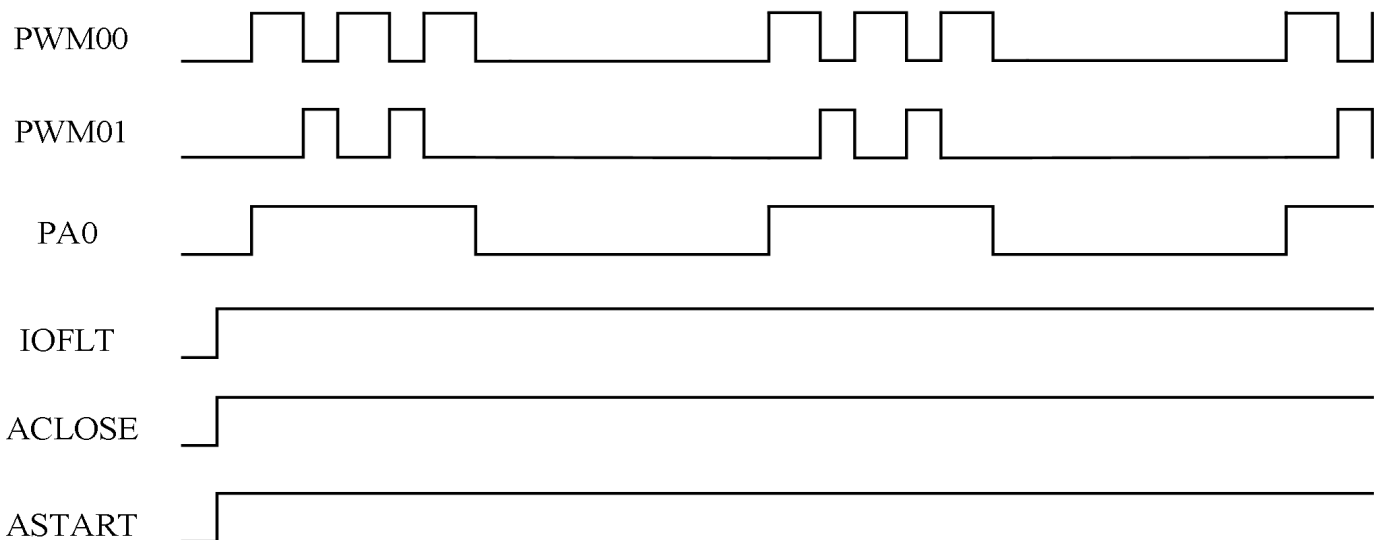
IO (PA0) =0

CMPOUT=0

故障刹车时的输出状态,故障刹车时，三组互补 PWM 输出关闭；此时输出逻辑由 PORTA 寄存器控制，输出逻辑高或逻辑低。

ACLOSE 故障清除,只要故障条件有效，PWM 输出便不能由指令打开。只有当故障条件被清除时，ACLOSE 才能再次启动。

ASTART 自动重启模式,故障刹车时，PWM 输出使能关闭。当故障事件结束后，PWM 将在 TIMER2 或 TIMER3 更新周期时使能 PWM 输出。3 路 PWM 输出可同时配置成自动重启模式，否则 PWM 输出必须由指令重启。



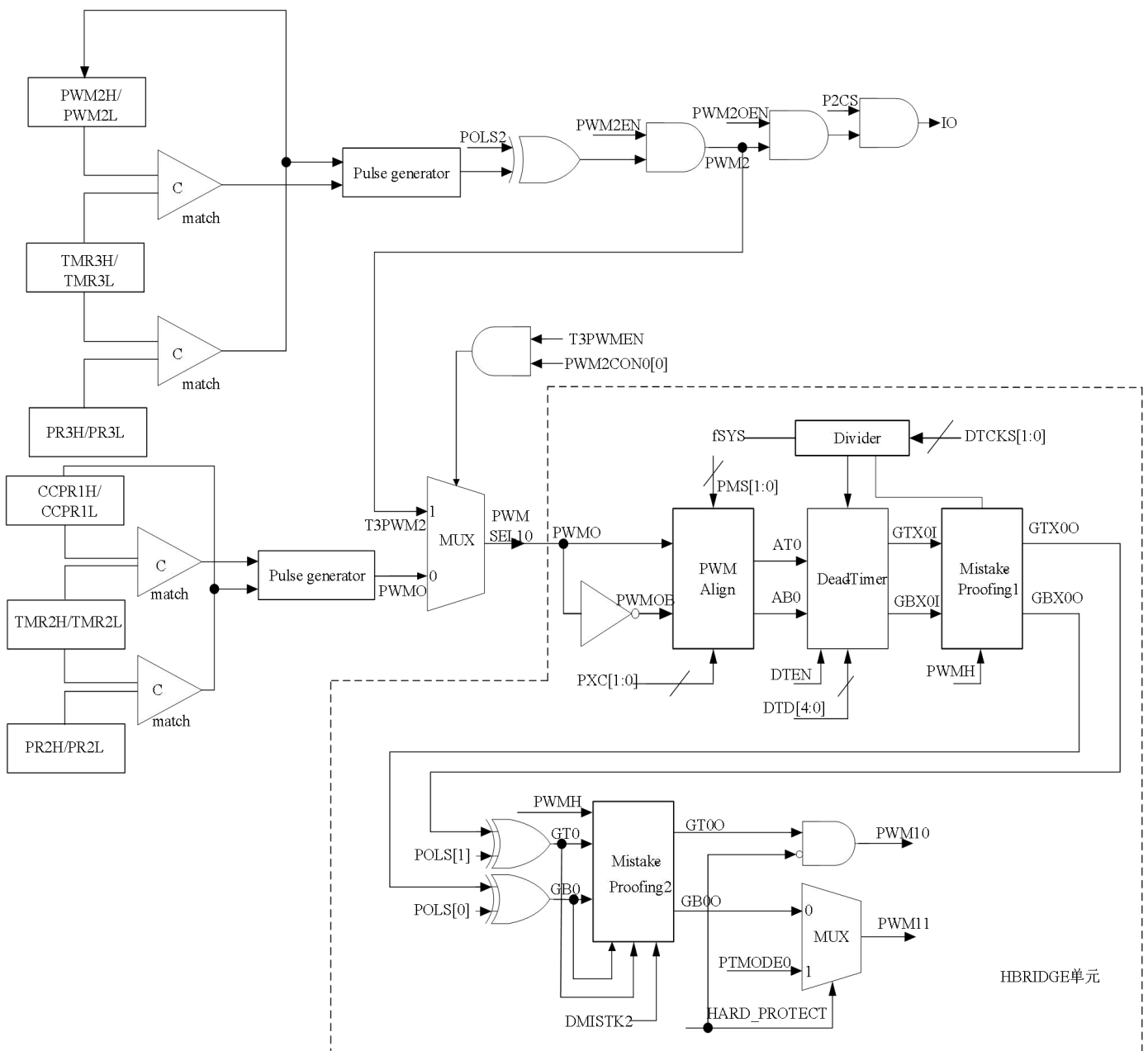
IO 作故障输入时序图

### 8.4 3 组互补 PWM 输出

互补式输出控制是通过一系列寄存器实现的。这些寄存器可用于选择 PWM 调制模式、死区时间设置以及输出极性控制等。通过选择可以将 Timer2 或 Timer3 的 PWM 映射到 CCP 的 HBRIDGE 单元。CCP 和 Timer2 构成一路 PWM0 信号，同时输出到 3 个 HBRIDGE 单元，



可控制 3 组互补 PWM 输出。Timer3 的 PWM2, PWM3, PWM4 的三路 PWM 信号, 分别输出到 3 个 HBRIDGE 单元, 可以构成 3 组独立占空比可调的互补 PWM 输出, 用于复杂 PWM 互补输出控制。当 Timer3 的 PWMMEN 寄存器使能时, PWM2 输出控制 PWM00 和 PWM01, PWM3 输出控制 PWM10 和 PWM11, PWM4 输出控制 PWM20 和 PWM21。PWM00 和 PWM01、PWM10 和 PWM11、PWM20 和 PWM21 分别是三组输出, 其中 PWM00 和 PWM01 的原理图如下图, PWM10 和 PWM11、PWM20 和 PWM21 原理相同。



PWM 输出原理图

注：上臂对应 PWM00、PWM10、PWM20

### 下臂对应 PWM01、PWM11、PWM21

#### PWM1CON1(PWM 的控制寄存器)

地址：0XFBD

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

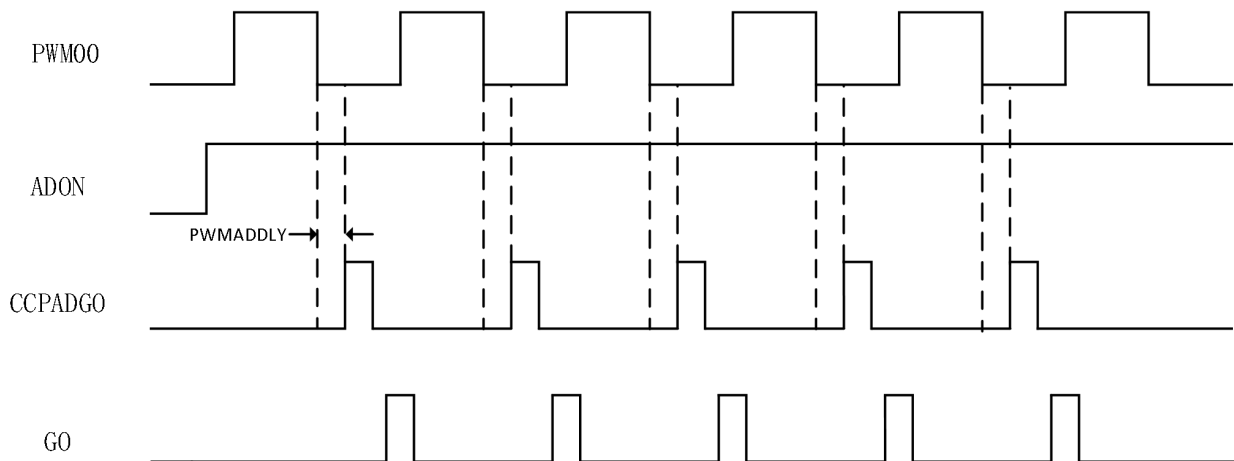
#### PWM1CON0(PWM 的控制寄存器)

地址：0XFBE

Bit	Name	Description	Attribute	Reset
7	-	-	R/W	0
6	CMPFLT	CMPOUT 为故障输入，当从 1 变为 0 后，禁止 PWM 输出 1: 使能 CMPOUT 作为故障输入 0: 禁止 CMPOUT 作为故障输入	R/W	0
5	ASTART	发生故障后(ACLOSE=1)，当使能的故障都解除后，自动启动 PWM 输出 1:使能启动 PWM 输出 0:禁止启动 PWM 输出	R/W	0
4	ACLOSE	发生故障后，自动关闭 PWM 输出 1:使能关闭 PWM 输出 0:禁止关闭 PWM 输出	R/W	0
3	IOFLT	IO 为故障输入，当从 1 变为 0 后，禁止 PWM 输出 1: 使能 IO 作为故障输入 0: 禁止 IO 作为故障输入 <span style="color: red;">注：故障 IO 映射为 PA0;</span>	R/W	0
2:1	PWMADPOS	PWM 跳变沿使能 ADC 采集 1x: 中心对齐模式：TMR2==PR2 或者 TMR2==12'H0 时， TMR3==PR3 或者 TMR3==12'H0 时，启动 PWMADDLY 计数器，当计数器为 0 时启动 ADC 采样	R/W	0

		01: PWM00 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  00: PWM00 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。  沿跳变时 ADC 才会采集, 如果沿不跳变 ADC 不会采集 (上升沿, PLOS 默认选 0 时)		
0	PWM10ADEN	使能 PWM00 输出跳变时, 自动启动 ADC 采集功能  1: 使能 ADC 采集  0: 禁止 ADC 采集	R/W	0

下图为 PWM00 跳变时, ADC 采集时序图。



ADC 采集时序图

## 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	R/W	0

Bit	Name	Description	Attribute	Reset
		0: 防呆输出 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 PXC20 和 PXC21 位分别控制上/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 PXC11 和 PXC10 位分别控制上/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 11: PWM00、PWM01 控制（由 PXC00 和 PXC01 位分别控制上/下臂输出。）	R/W	00

### PXC(PWM 控制输出寄存器)

地址：0XFC4

Bit	Name	Description	Attribute	Reset
7	T3PWMEN	使能 TIME3 PWM2,PWM3,PWM4 控制 CCP 的 HBRIDGE 单元	R/W	0

Bit	Name	Description	Attribute	Reset
		1: 允许使能 0: 禁止使能		
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	R/W	0
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 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

**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

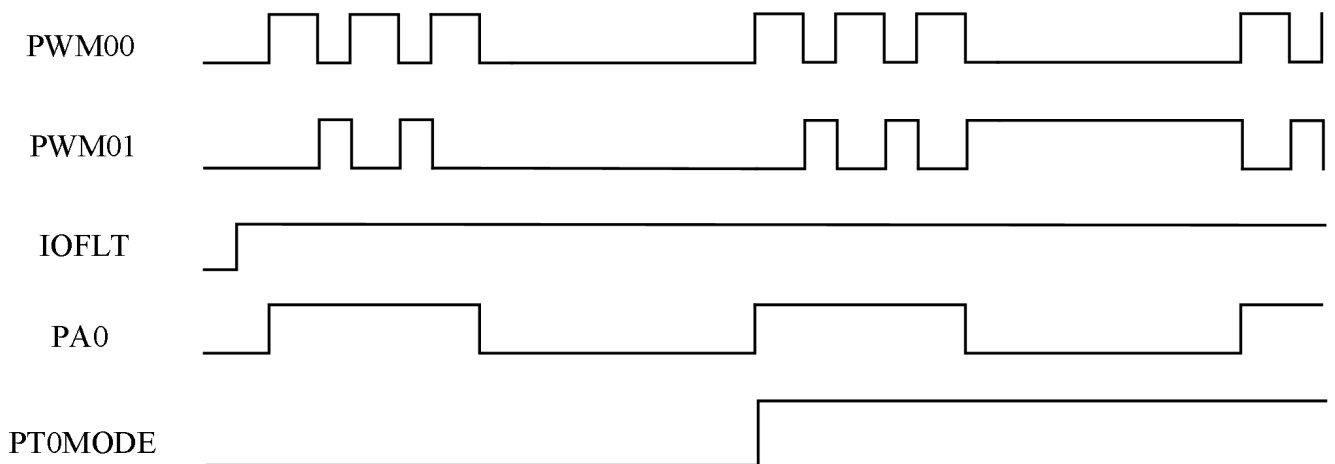
**POLS (极性选择寄存器)**

地址: 0XFC2

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	PT2MODE	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

Bit	Name	Description	Attribute	Reset
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

下图为 PWM01 保护输出时序图：



## 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

Bit	Name	Description	Attribute	Reset
5	PWM21EN	PWM21 映射的 IO 输出使能 1: 使能 PB7 和 PA2 输出 0: 禁止 PB7 和 PA2 输出	R/W	0
4	PWM20EN	PWM20 映射的 IO 输出使能 1: 使能 PB6 和 PA1 输出 0: 禁止 PB6 和 PA1 输出	R/W	0
3	PWM11EN	PWM11 映射的 IO 输出使能 1: 使能 PA0 和 PA3 输出 0: 禁止 PA0 和 PA3 输出	R/W	0
2	PWM10EN	PWM10 映射的 IO 输出使能 0: 禁止 PB0 和 PB4 输出 1: 使能 PB0 和 PB4 输出	R/W	0
1	PWM01EN	PWM01 映射的 IO 输出使能 1: 使能 PB1 输出 0: 禁止 PB1 输出	R/W	0
0	PWM00EN	PWM00 映射的 IO 输出使能 1: 使能 PB2 输出 0: 禁止 PB2 输出	R/W	0

注: PWM20、PWM21、PWM10、PWM11 的 IO 映射选择寄存器见 PCS 寄存器

### PCS 寄存器

地址: 0XFC0

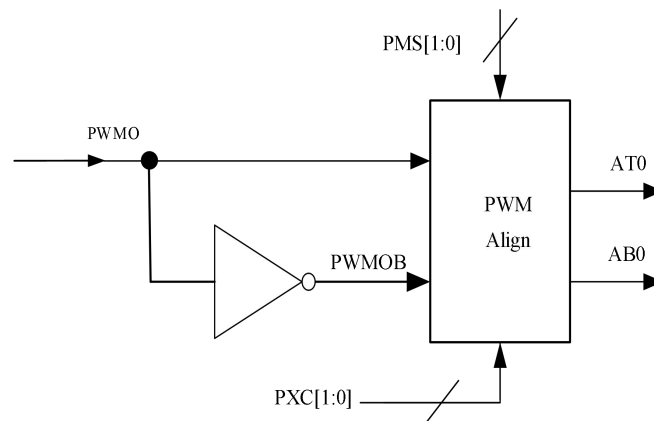
Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3	PWM21CS	PWM21 的 IO 映射 1:PB7 输出 0: PA2 输出	R/W	0
2	PWM20CS	PWM20 的 IO 映射 1:PB6 输出 0: PA1 输出	R/W	0



1	PWM11CS	PWM11 的 IO 映射 1:PA3 输出 0: PA0 输出	R/W	0
0	PWM10CS	PWM10 的 IO 映射 1:PB4 输出 0: PB0 输出	R/W	0

### 8.4.1. 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

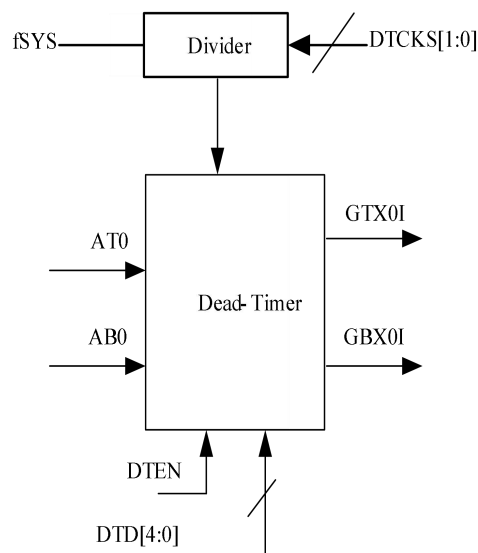
PMS [1:0]	PWM 调制模式	AT0	AB0
00	互补式控制	PWMO	PWMOB
01	非互补式上臂调制	PWMO	0
10	非互补式下臂调制	0	PWMO
11	由 PXC 寄存器控制上下臂输出	PXC [1]	PXC [0]

校准电路输出表 2

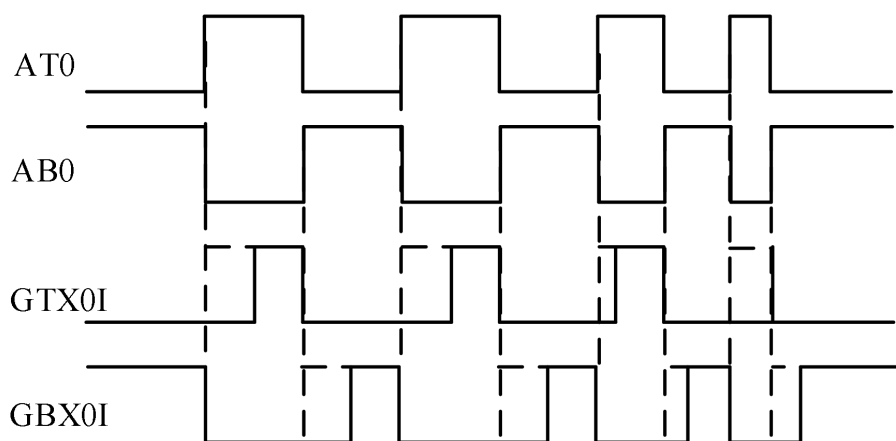
### 8.4.2. 死区时间

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



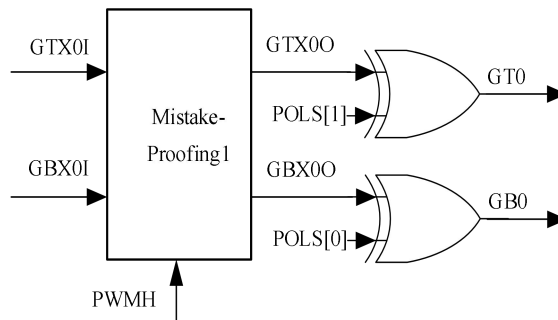
死区时间时序图 (n=0)

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

死区时间电路输出表 1

### 8.4.3. 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

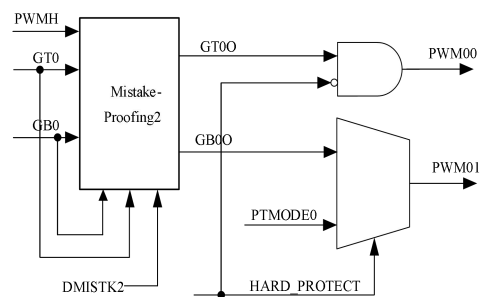
一级防呆电路输出表 1

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

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

一级防呆电路输出表 3



二级防呆电路

GT0	GB0	DMISK2	GT00	GB00
0	0	0	PWMH	0
0	1	0	0	1
1	0	0	1	0
1	1	0	PWMH	0

GT0	GB0	DMISK2	GT00	GB00
0	0	1	0	0
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1

二级防呆电路输出表 1

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

GT00	GB00	HARD_PROTECT	PWM00	PWM01
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=0x40;//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;
    
```

#### 8.4.4. 增强型六路 PWM

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

- 单 PWM
- 半桥 PWM

- 全桥 PWM，正向模式
- 全桥 PWM，反向模式

PWM 输出与 I/O 引脚复用，并被指定为

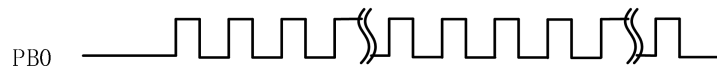
PWM00/PWM01/PWM10/PWM11/PWM20/PWM21 依次复用 PB2、PB1、PB0/PB4、PA0/PA3、PA1/PB6、PA2/PB7。PWM 引脚的极性可配置，可通过配置 POLS 寄存器选择极性。每两个 I/O 为一组半桥。全桥为二组半桥。

### 注意：

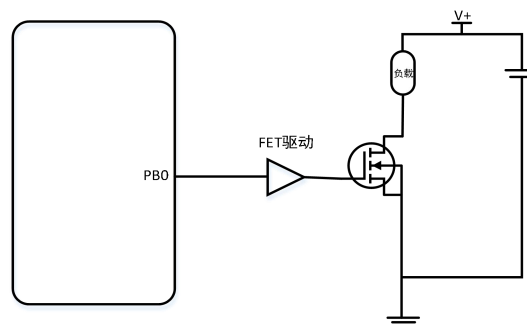
- 必须正确配置每个 PWM 输出的 TRIS 寄存器值；
- 清零 CCPCON 寄存器将放弃所有 PWM 输出引脚；
- 当 PWMnOE 不使能时，增强型 PWM 模式所不使用的任何引脚均可用于其他引脚功能。  
(n=00/01/10/11/20/21)

#### 8.4.4.1. 单 PWM 模式

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



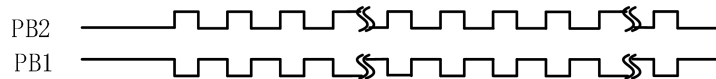
单 PWM 模式输出图



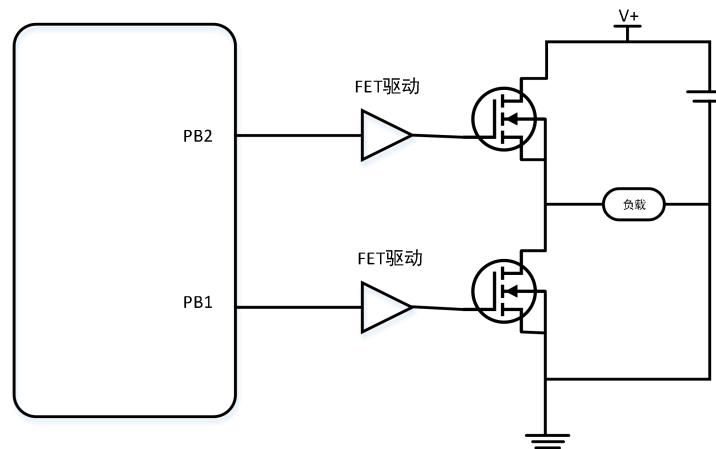
单 PWM 电路

#### 8.4.4.2. 半桥模式

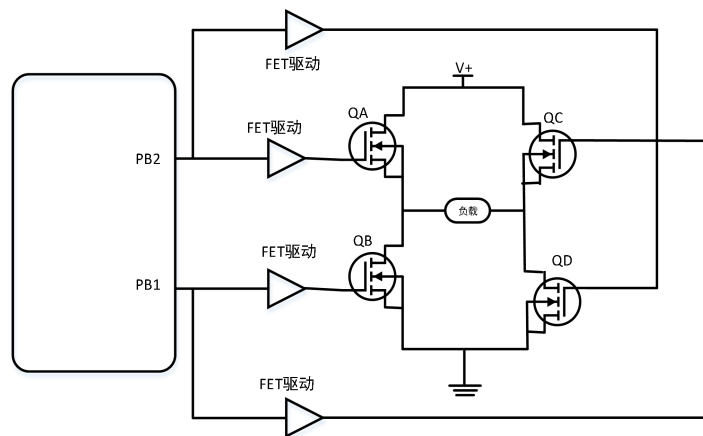
在半桥模式下，有两个引脚用作输出以驱动负载。PB2 和 PB1 为一组半桥，PB0/PB4 和 PA0/PA3 为一组半桥，PA2/PB7 和 PA1/PB6 为一组半桥。PWM 输出信号被输出到 PB2、PB0、PA2 引脚，而互补 PWM 输出信号被输出到 PB1、PB0、PA1。下图是一组半桥输出图，其它两组半桥同理。半桥模式可以运用为全桥模式。



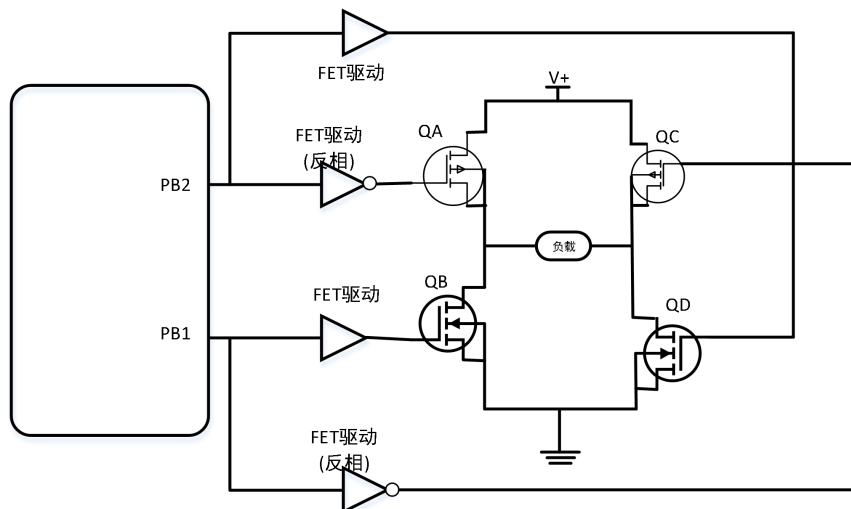
半桥模式输出图



标准半桥电路图



半桥输出驱动全桥电路（4NMOS）注：PB0 和 PA0, PA1 和 PA2 同图上一样



半桥输出驱动全桥电路（2PMOS+2NMOS）注：PB0 和 PA0, PA1 和 PA2 同图上一样

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

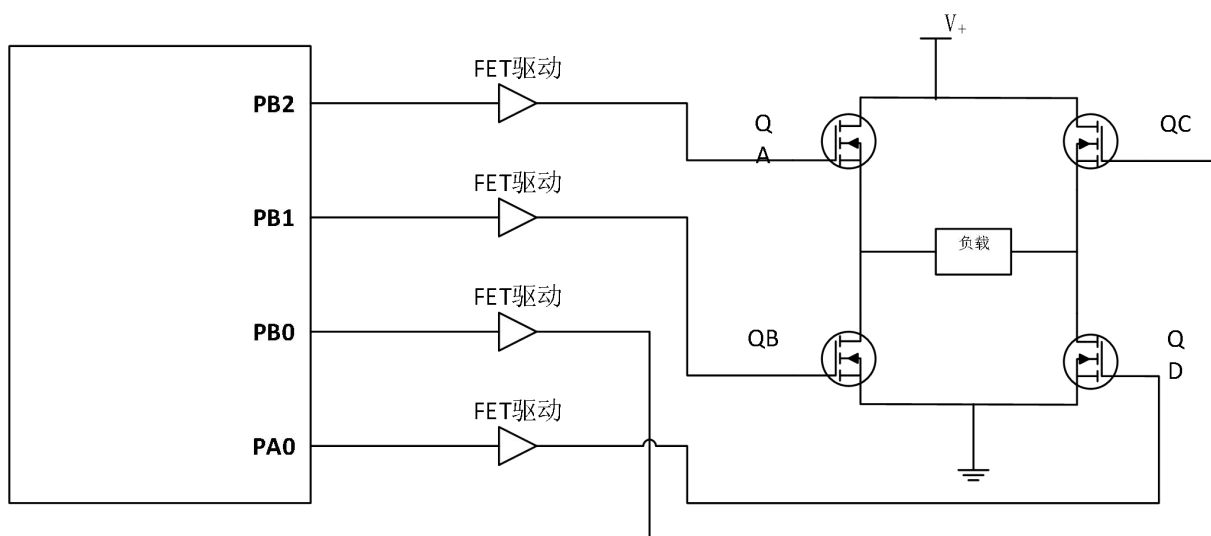
PMS=0x00; //互补输出
PXC=0x00; //上下臂选择
POLLS=0x00; //极性选择
PME=0x3f; //PWM映射IO使能及保护选择
PCS=0x00; //PWM映射IO
CCPCON=0x02; //使能PWM模式
T2CON=0x04; //使能TMR2
    
```

#### 8.4.4.3. 全桥模式

在全桥模式下，所有四个引脚用作输出。这里以第一组半桥 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。

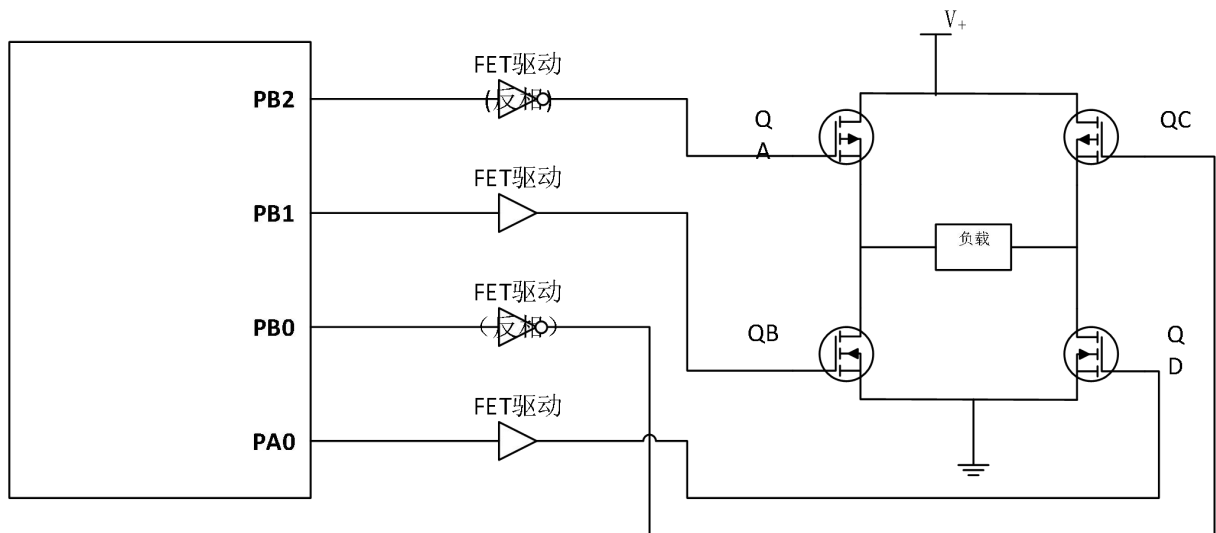
全桥应用示例 a 所示为一个全桥应用示例，使用 4 个 NMOS。

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。



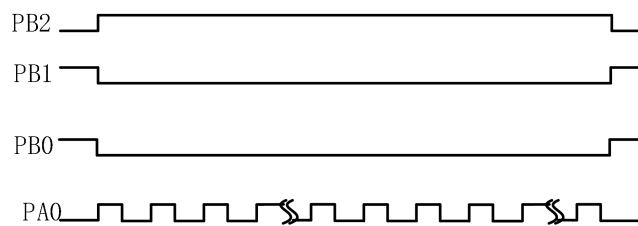
全桥应用示例 a





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

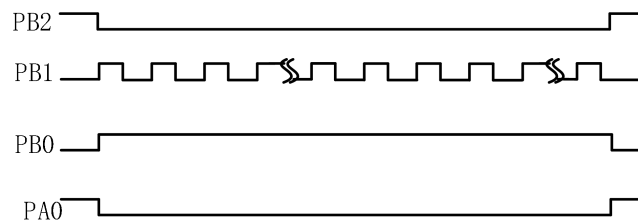
PMS=0x0b;//第一组半桥PXC控制上下臂输出，第二组半桥非互补式上臂
    
```

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

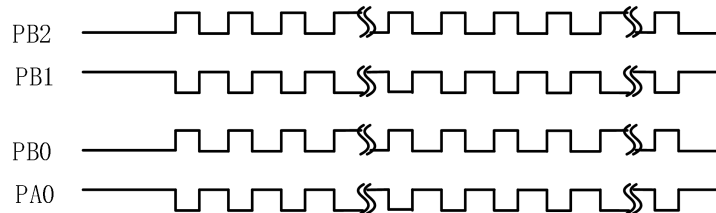
PR2H=0xf;
PR2L=0xff; //周期配置
TMR2H=0x00;
TMR2L=0x00; //计数值初始值

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:

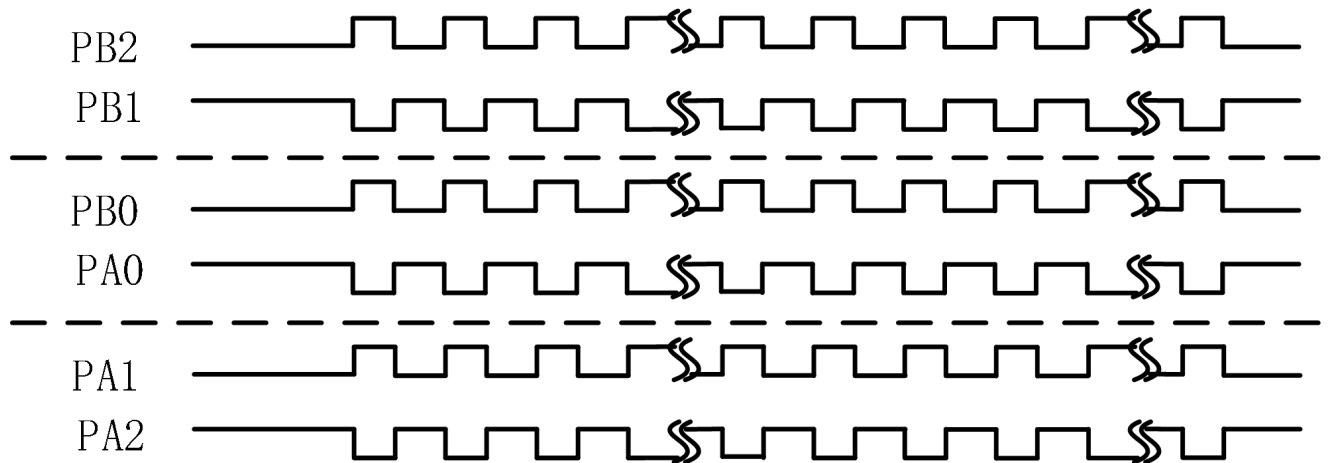
```
CCPIE=1;
CCPIF=0;
CCPIP=0;
TMR2IE=0; //中断相关配置

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
```

## 8.4.4.4. 3组 PWM 输出



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。

## 9 中断

AD18F08 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 输入改变中断
- CCP 中断
- 低电压中断
- ADC 中断
- CMP 中断

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

中断发生时 GIEH(GIEL)位 (在中断发生前 GIEH(GIEL)位和该中断相关的中断屏蔽位置 1) 被硬件清零从而禁止进一步中断 (AD18F08 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。

### 9.1 外部中断

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

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

外部中断 INT1 管脚上升沿还是下降沿触发由 INT1EDG 位 (T1CON1 寄存器)决定, 当

一个有效的跳变发生时标志位 INT1IF 置 1，如 INT1IE 位清零，该中断被屏蔽。

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

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

在睡眠之前 INT2IE 位已被置 1，INT2 管脚可以作为系统睡眠唤醒条件。在睡眠之前 GIEH(GIREL)位被置 1，CPU 唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

## 9.2 Timer0 中断

TMR0 发生溢出 TMR0=PR0 时 TMR0IF 标志位置 1，TMR0IE 位清零，该中断被屏蔽。

## 9.3 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，TMR1IE 位清零，该中断被屏蔽。

## 9.4 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。

## 9.5 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。

## 9.6 PortA 输入改变中断

输入改变中断触发时 PA<5:0> PAIF 标志位置 1 (PIR2<5>). PAIE 位(PIE2<5>)清零，该中断被屏蔽。 PAIE 在睡眠之前置 1， Port A 输入脚改变中断也可以作为睡眠唤醒条件。在睡

眠之前 GIE 位已被置 1 机器唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

**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。）

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

**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。）

## 9.8 低电压、高电压中断

- 当 LVDM[1:0]==2'b01 时，系统 VDD 电压低于设定的 LVD 电压值，LVDIF 位置为 1。LVDIE 位(PIE2<2>)清零，该中断被屏蔽，LVD 中断的触发时间大约需要 8ms。
- 当 LVDM[1:0]==2'b10 时，系统 VDD 电压高于设定的 LVD 电压值，LVDIF 位置为 1。LVDIE 位(PIE2<2>)清零，该中断被屏蔽，LVD 中断的触发时间大约需要 8ms。

### 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	R/W	00

Bit	Name	Description	Attribute	Reset
		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, 并强制 PB0 输出为 PB0ST 值		
3	Reserved			
2:0	LVD [2:0]	VDD 电压阈值选择 0: 2.4V(默认) 1: 2.7V 2: 3.0V 3: 3.3V 4: 3.6V 5: 3.8V 6: 4.2V 7: 4.3V	R/W	000

## 9.9 比较器中断

上升沿中断,当 CMPOUT 从 0 变为 1 时, CMPIF 置位(CMPIF=1)。

双沿中断,读取 CMPCON 寄存器后,当 CMPOUT 输出发生改变,CMPIF 置位(CMPIF=1)。

CMPIE 位(PIE2<0>)清零, CMP 中断被屏蔽。

## 9.10 ADC 中断

1、当 GO 从 1 变为 0 时, ADIF 置位为 1. ADIE 位(PIE1<5>)清零, 该中断被屏蔽

2、支持采集电压比较功能, 并可以产生如下描述的对应中断标志;

✧ 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 ADCMP0IF;



## 9.11 CCP 中断

中断详细描述见第 8 章节

## 9.12 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 <u>当 IPEN=1 时:</u> 1: 使能所有高优先级中断 0: 禁止所有中断	R/W	1
6	PEIE/GIEL	外设中断使能位 <u>当 IPEN=1 时:</u> 1: 允许所有低优先级的外设中断 0: 禁止所有低优先级的外设中断	R/W	1
5:0	Reserved			

**注意:** IPEN=1 时, GIE 与 GIEH 功能相同, PEIE 与 GIEL 功能相同; IPEN=0 时, 默认进入高优先级中断服务函数, 无低优先级中断服务函数, 关闭总中断 PEIE=0。

### IPR1 寄存器

地址:0XFA3

Bit	Name	Description	Attribute	Reset
7	ADCMP1IP	ADCMP1 中断优先级 1: 高优先级 0: 低优先级	R/W	1
6	ADCMP0IP	ADCMP0 中断优先级 1: 高优先级 0: 低优先级	R/W	1
5	ADIP	ADC 中断优先级	R/W	1

Bit	Name	Description	Attribute	Reset
		1: 高优先级 0: 低优先级		
4	CCPIP	CCP 中断优先级 1: 高优先级 0: 低优先级	R/W	1
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:0XFA2

Bit	Name	Description	Attribute	Reset
7	ADCMP1IF	ADC 采集值比较器 1 中断标志位, 软件设置清零	R/W	0
6	ADCMP0IF	ADC 采集值比较器 0 中断标志位, 软件设置清零	R/W	0
5	ADIF	A/D 转换产生中断标志, 软件设置清零	R/W	0
4	CCPIF	CCP 中断标志, 发生中断置 1, 软件设置清零	R/W	0
3	TMR3IF	溢出中断标志, 发生 Timer3 溢出中断置 1, 软件设置清零	R/W	0
2	TMR2IF	溢出中断标志, 发生 Timer2 溢出中断置 1, 软件设置清零	R/W	0
1	TMR1IF	溢出中断标志, 发生 Timer1 溢出中断置 1, 软件设置清零	R/W	0

Bit	Name	Description	Attribute	Reset
0	TMR0IF	溢出中断标志，发生 Timer0 溢出中断置 1，软件设置清零	R/W	0

## PIE1 寄存器

地址:0XFA1

Bit	Name	Description	Attribute	Reset
7	ADCMP1IE	ADCMP1 中断允许位 1: 使能 ADCMP1 中断 0: 禁止 ADCMP1 中断	R/W	0
6	ADCMP0IE	ADCMP0 中断允许位 1: 使能 ADCMP0 中断 0: 禁止 ADCMP0 中断	R/W	0
5	ADIE	ADC 中断允许位 1: 使能 ADC 中断 0: 禁止 ADC 中断	R/W	0
4	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	R/W	0

Bit	Name	Description	Attribute	Reset
		0: 禁止 Timer0 溢出中断		

## IPR2 寄存器

地址:0XFA0

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

## PIR2 寄存器

地址:0XF9F

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	PBIF	<p>PB 口的中断标志位</p> <p>1:至少一个 PORTB 引脚的电平状态发生了改变 (必须用软件清零, 写 0 清 0)</p> <p>0:没有一个 PORTB 引脚的电平状态发生改变</p>	R/W	0
5	PAIF	<p>PA 口的中断标志位</p> <p>1:至少一个 PORTA 引脚的电平状态发生了改变 (必须用软件清零, 写 0 清 0)</p> <p>0:没有一个 PORTA 引脚的电平状态发生改变</p>	R/W	0
4	INT2IF	<p>INT2 外部中断标志位</p> <p>1: 产生 INT2 外部中断标志位(必须由软件清零, 写 0 清 0)</p> <p>0: 未产生 INT2 外部中断</p>	R/W	0
3	INT1IF	<p>INT1 外部中断标志位</p> <p>1: 产生 INT1 外部中断标志位(必须由软件清零, 写 0 清 0)</p> <p>0: 未产生 INT1 外部中断</p>	R/W	0
2	INT0IF	<p>INT0 外部中断标志位</p> <p>1: 产生 INT0 外部中断标志位(必须由软件清零, 写 0 清 0)</p> <p>0: 未产生 INT0 外部中断</p>	R/W	0
1	LVDIF	<p>LVD 的中断标志</p> <p>当 LVDM[1:0]=2'b01 时</p> <p>1: LVD 检测电压低于设置阈值电压(必须由软件清零)</p> <p>0: LVD 检测电压高于设置阈值电压, 或已经由软件清 0</p> <p>当 LVDM[1:0]=2'b10 时</p> <p>1: LVD 检测电压高于设置阈值电压(必须由软件清零)</p> <p>0: LVD 检测电压低于设置阈值电压, 或已经由软件清 0</p> <p>LVDM[1:0]在 PCON 寄存器中</p>	R/W	0
0	CMPIF	CMP 中断标志位	R/W	0

Bit	Name	Description	Attribute	Reset
		1: 产生 CMP 中断标志位(必须由软件清零, 写 0 清 0) 0: 未产生 CMP 中断标志位		

## PIE2 寄存器

地址:0XF9E

Bit	Name	Description	Attribute	Reset
7	Reserved			
6	PBIE	PortB 输入改变中断允许位 1: 使能 PortB 输入改变中断 0: 禁止 PortB 输入改变中断	R/W	0
5	PAIE	PortA 输入改变中断允许位 1: 使能 PortA 输入改变中断 0: 禁止 PortA 输入改变中断	R/W	0
4	INT2IE	外部中断 2 允许位 1: 使能 INT2 中断 0: 禁止 INT2 中断	R/W	0
3	INT1IE	外部中断 1 允许位 1: 使能 INT1 中断 0: 禁止 INT1 中断	R/W	0
2	INT0IE	外部中断 0 允许位 1: 使能 INT0 中断 0: 禁止 INT0 中断	R/W	0
1	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
0	CMPIE	CMP 中断允许位 1: 使能 CMP 中断 0: 禁止 CMP 中断	R/W	0

## 10 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 2K 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 2K 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 2K 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 2K 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；

### 10.1 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	DEEPPWSAVE	PWOFF
CCP	√			
TIMER0	√			
TIMER1	√	√	√	
TIMER2	√			
TIMER3	√			
WDT	√	√	√	
RST	√	√	√	√
INT	√	√	√	√
IO	√	√	√	√
LVD	√	√	√	

注：√表示可唤醒的方式

在睡眠状态下，四种模式可以工作的模块如下图所示

模块	工作模式				
	ACTIVE	IDLE	POWER SAVE	DEEPPWSAVE	POWER OFF
OSC16M	√	√			
OSC2K	√	√	√	√	
CPU	√				
SRAM	√	√	√	√	√
Timer0/2/3	√	√			
Timer1	√	√	√	√	
CCP	√	√			
WDT	√	√	√	√	
External Interrupt	√	√	√	√	√
PAIF/ PBIF	√	√	√	√	√
BGR	√	√	√	√	
LVT	√	√	√	√	
ADC	√	√	√	√	
DAC	√	√			
IO	√	√	√	√	√
RESET	√	√	√	√	√

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

### SMCR（状态控制寄存器）

地址：0XF65

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:1	SM[2:0]	休眠模式选择 000: IDLE 模式，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 2K 时钟工作，支持外部中断、IO 中断、复位、	R/W	000



Bit	Name	Description	Attribute	Reset
		看门狗溢出、LVD 低压唤醒和 TIMER1 的 2K 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 2K 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 2K 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；		
0	SE	休眠模式使能位  1: 使能休眠模式,硬件自动清零  0: 禁止休眠模式	R/W	0

### 实例代码:

```

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

unsigned char t0;
void init()
{
    PR0=0xaa;
    TMR0=0x00;
    TOCON=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++;  
  }  
}
```

## 11 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
- DAC 参考电压

**注：**本章涉及到的具体电压均为内部参考电压产生，无需单独开关；

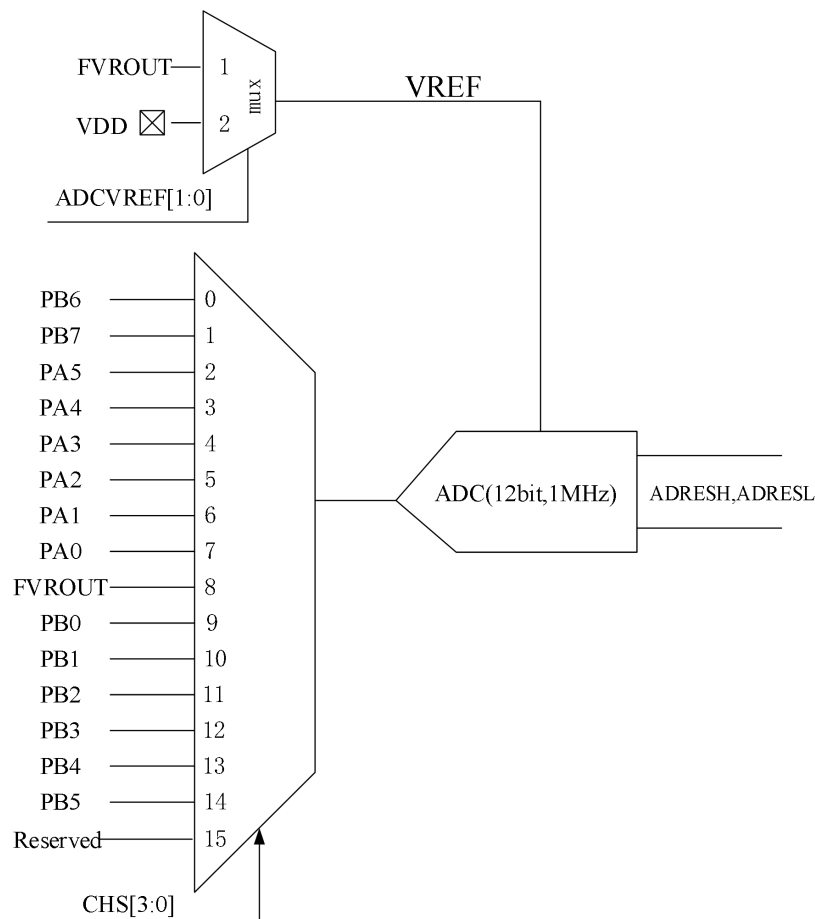
### FVRCON0 寄存器

地址：0XF72

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 PB3 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: Reserved 001: VREF1P0(1.0V) 010: 1/4VDD 011: NTC 100: DACOUT	R/W	000

## 12 模数转换器 (ADC)

AD18F08 包含一个 16 通道输入的 12 位 ADC，能够将一个模拟输入转换成 12 位数字信号。在根据需要配置好 A/D 模块之后，必须在转换开始之前对选定的通道进行采样。当采集启动延时 ACQT 计数完成后，硬件启动 A/D 转换。A/D 转换完成之后，转换结果被装入 ADRESH:ADRESL 寄存器对，GO/DONE 位被硬件清零且 A/D 中断标志位 ADIF 被置 1。



A/D 结构图

执行 A/D 转换步骤：

配置 A/D 模块

- 选择参考电压 (通过 ADCON1[7:6]寄存器)
- 选择 A/D 输入通道 (通过 ADCON0[5:2]寄存器)
- 选择 A/D 采集时间 (通过 ADCON1[5:3]寄存器)
- 选择 A/D 转换时间 (通过 ADCON1[2:0]寄存器)
- 使能 A/D 模块 (通过 ADCON0[0]寄存器)

1) 需要时，配置 A/D 中断

- 清零 ADIF 位
  - 将 ADIE 位置 1
  - 将 GIE 位置 1
- 2) 如果需要, 需等待所需的采集时间。
- 3) 启动转换:
- 将  $\overline{\text{GO/DONE}}$  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。
- 7) 外部输入最大阻抗计算公式:

$$R_{AIN} < \frac{T_s}{f_{ADC} * C_{ADC} * \ln(2^{N+2})} - R_{ADC}$$

- 8) 该公式用于确定允许误差低于 1/4LSB 的最大外部阻抗, N=12;

$$f_{ADC}=8\text{MHz}; C_{ADC}=2.67\text{pF};$$

VDD(V)	R <sub>ADC</sub> (KΩ)	T <sub>s</sub> (Cycles)	t <sub>s</sub> (μs)	R <sub>AIN</sub> max(KΩ)
5.0	2.501	3	0.375	13.82
3.3	3.276	3	0.375	12.607

#### ADCMP0H (AD 比较寄存器 0 高字节)

地址: 0XFAB

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

#### ADCMP1H(AD 比较寄存器 1 高字节)

地址: 0XFAA

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

#### ADCMP01L(AD 比较寄存器 0 和 1 低字节)

地址：0XFA9

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

ADRESH (AD 转换结果的高 4 位)

地址：0XFA8

Bit	Name	Description	Attribute	Reset
7:4	Reserved			
3:0	ADRESH	AD 转换结果的高 4 位	R/W	0000

ADRESL(AD 转换结果的低 8 位)

地址：0XFA7

Bit	Name	Description	Attribute	Reset
7:0	ADRESL	AD 转换结果的低 8 位	R/W	0X00

ADCON0 (ADC 控制寄存器 0)

地址：0XFA6

Bit	Name	Description	Attribute	Reset
7	ADFM	AD 结果对齐标志 1: 左对齐 ADRESH = adc[11:4] 左对齐 ADRESL = {adc[3:0],4'b0} 0: 右对齐 ADRESH = {4'b0,adc[11:8]} 右对齐 ADRESL = adc[7:0]	R/W	0
6	ADCMPMODE	AD 采集值的比较模式；支持采集电压比较功能，可以产生如下描述的对应中断标志 0: 采集值大于 {ADCMP1H:ADCMP01L[7:4]} 产生中断 ADCMP1IF 或者小于 {ADCMP0H:ADCMP01L[3:0]} 产生中断 ADCMP0IF;	R/W	0

Bit	Name	Description	Attribute	Reset
		1: 采集值小于{ADCMP1H:ADCMP01L[7:4]}且大于{ADCMP0H:ADCMP01L[3:0]}, 产生中断 ADCMP0IF;		
5:2	CHS [3:0]	CHS3:CHS0 – 模拟通道选择位  0000: ADC 通道 0(PB6) 0001: ADC 通道 1(PB7) 0010: ADC 通道 2(PA5) 0011: ADC 通道 3(PA4) 0100: ADC 通道 4(PA3) 0101: ADC 通道 5(PA2) 0110: ADC 通道 6(PA1) 0111: ADC 通道 7(PA0)  1000: ADC 通道 8(FVROUT) 1001: ADC 通道 9(PB0) 1010: ADC 通道 10(PB1) 1011: ADC 通道 11(PB2) 1100: ADC 通道 12(PB3) 1101: ADC 通道 13(PB4) 1110: ADC 通道 14(PB5)	R/W	0000
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：0XFA5

Bit	Name	Description	Attribute	Reset
7:6	ADVREF [1:0]	A/D 参考电压选择位 01: FVROUT 10: VDD	R/W	00
5:3	ACQT [2:0]	A/D 延时采集时间选择 111: 15 TAD 110: 13TAD 101: 11TAD 100: 9TAD 011: 7 TAD 010: 5TAD 001: 3 TAD 000: 1TAD	R/W	000
2:0	ADCS [2:0]	A/D 转换时钟选择位 111: Reserved 110: FSYS/512(ADSP=0), FSYS /64(ADSP=1) 101: FSYS /128(ADSP=0), FSYS /16(ADSP=1) 100: FSYS /32(ADSP=0), FSYS /4(ADSP=1) 011: Reserved 010: FSYS /256(ADSP=0), FSYS /32(ADSP=1) 001: FSYS /32 (ADSP=0), FSYS /8(ADSP=1) 000: FSYS /16(ADSP=0), FSYS /2(ADSP=1)	R/W	000

**注意：**

- 1、使用 ADC 自动采集功能时必须设置 TMR1，自动采集时间间隔由 TMR1 定时时间确定。
- 2、ADC 的转换时间=3TAD(输入采样时间)+12TAD
- 3、完成一位的转换的时间定义为 TAD



## 13 数模转换器(DAC)

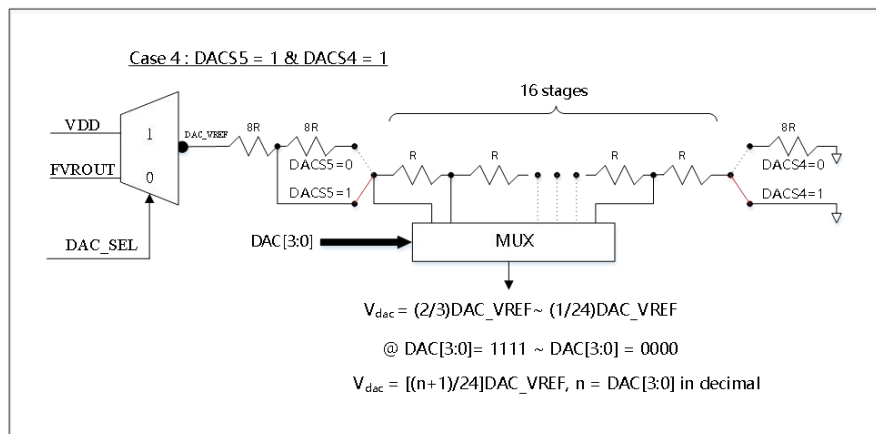
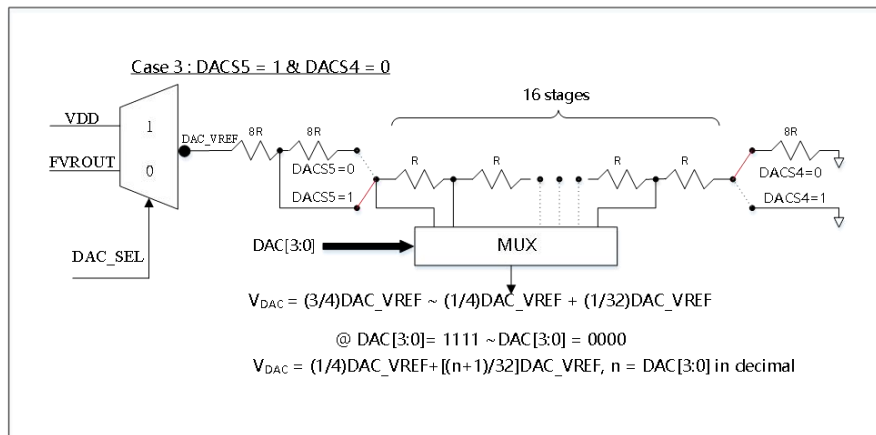
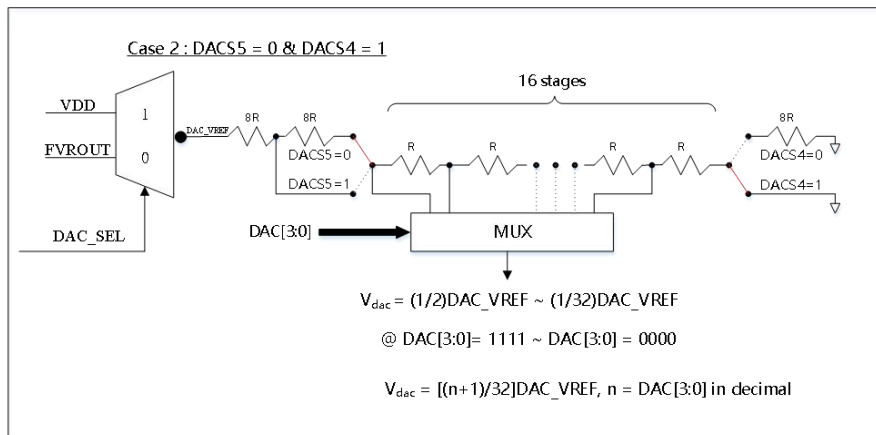
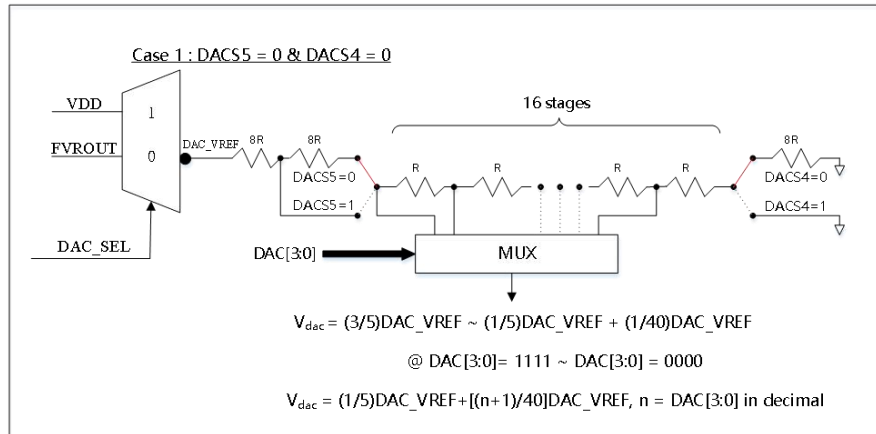
AD18F08 包含一个数模转换器 DAC。DAC 是由一串电阻所组成，可以产生不同层次的参考电压，DACON 寄存器的 4 和 5 位用来选择电阻串的最高和最低值；DAC[3:0]用于选择所要的电压值，该值由 DACS5，DACS4 来决定。下图显示了四个不同选择时，内部参考电压值的计算。DAC 输出电压范围可以从  $(1/32) * VDD$  到  $(3/4) * VDD$ 。

### 13.1 DAC 参考电压选择寄存器

DACCON

地址：0XF74

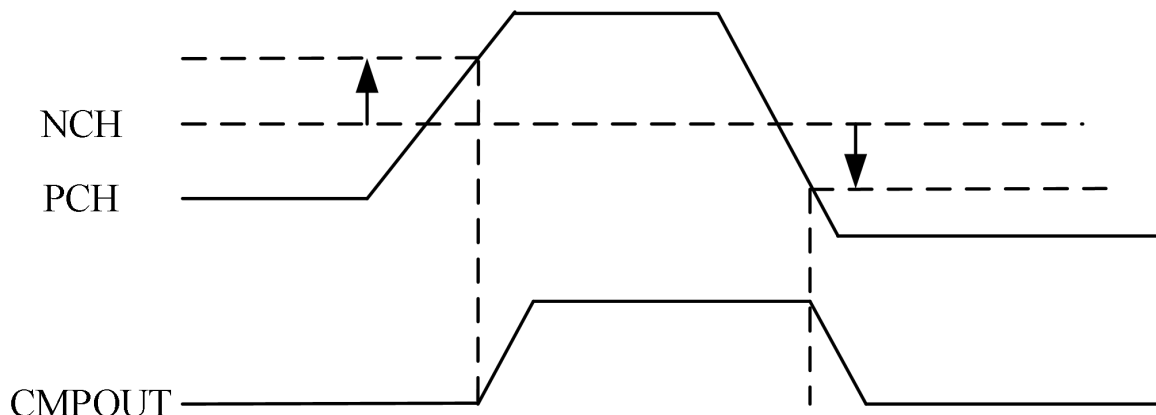
Bit	Name	Description	Attribute	Reset
7	DAC_SEL	DAC 参考电压选择位 1: VDD 0: FVROUT	R/W	0
6	DACEN	DAC 的使能 1: 允许使能 0: 禁止使能	R/W	0
5	DACS5	DAC0 正端电阻抽头选择	R/W	0
4	DACS4	DAC0 负端电阻抽头选择	R/W	0
3:0	DAC [3:0]	DAC0 输出选择	R/W	0000



# 14 比较器

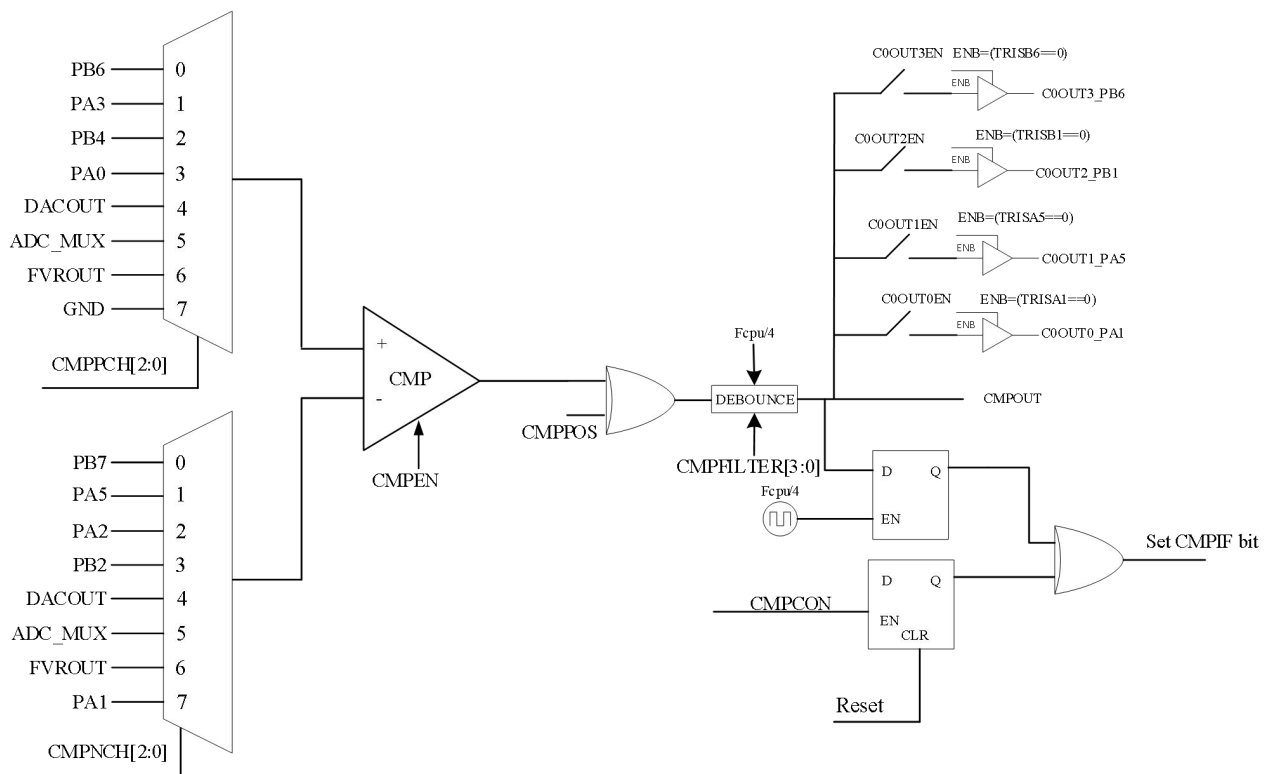
AD18F08 提供一个比较器，可以选取多个输入作为比较器输入源。

比较器的迟滞可以通过软件可编程比较器控制寄存器 CMPCON1 设置，比较器迟滞电压正端由 CMPHYSP\_VOL[1:0] 设置，可编程设置 0 mV、12mV、18mV、38mV；比较器迟滞电压负端由 CMPHYSN\_VOL[1:0] 设置，可编程设置 0 mV、14mV、21mV、41mV。



比较器迟滞图

## 14.1 比较器 CMP



CMP 电路图

**CMPCON0(比较器控制寄存器)**

地址: 0XF78

Bit	Name	Description	Attribute	Reset
7	CMPEN	比较器的使能 1: 使能 CMP 0: 禁止 CMP	R/W	0
6	CMPPPOS	比较器输出信号是否取反 1: 取反 0: 同向	R/W	0
5:3	CMPPCH[2:0]	比较器的正端输入 000: C0P0(PB6) 001: C0P1(PA3) 010: C0P2(PB4) 011: C0P3(PA0) 100: DACOUT 101: ADC_MUX 110: FVROUT 111: GND	R/W	000
2:0	CMPNCH[2:0]	比较器的负端输入 000: C0N0(PB7) 001: C0N1(PA5) 010: C0N2(PA2) 011: C0N3(PB2) 100: DACOUT 101: ADC_MUX 110: FVROUT 111: C0N4(PA1)	R/W	000

**CMPCON1(比较器控制寄存器)**

地址: 0XF77

Bit	Name	Description	Attribute	Reset
7:6	CMPHYSN_VOL	CMP 下降沿迟滞电压选择信号: 00:0mV 01:14mV 10:21mV 11:41mV	R/W	00
5:4	CMPHYSP_VOL	CMP 上升沿迟滞电压选择信号 00:0mV 01:12mV 10:18mV 11:38mV	R/W	00
3:1	Reserved			
0	CMPOUT	CMP 比较器的输出	R	0

**CMPCON2(比较器控制寄存器)**

地址: 0XF76

Bit	Name	Description	Attribute	Reset
7:4	CMPFILTER[3:0]	CMP 的滤波时间 $T=(\text{CMPFILTER}[3:0] \ll 4 + 0X0F) / F_{\text{cpu}}$ 有效滤除 CMP0OUT 在 T 时间内的毛刺 CMPFILTER[3:0]==0x00 时没有滤波	R/W	0000
3	C0OUT3EN	使能比较器 CMP0OUT 由 PB6 输出, 此时使能 PB6 的 IO 状态为输出 1: 使能 0: 禁止	R/W	0
2	C0OUT2EN	使能比较器 CMPOUT 由 PB1 输出, 此时使能 PB1 IO 状态为输出 1: 使能 0: 禁止	R/W	0

Bit	Name	Description	Attribute	Reset
1	C0OUT1EN	使能比较器 CMPOUT 由 PA5 输出，此时使能 PA5 IO 状态为输出 1：使能 0：禁止	R/W	0
0	C0OUT0EN	使能比较器 CMPOUT 由 PA1 输出，此时使能 PA1 IO 状态为输出 1：使能 0：禁止	R/W	0

## 15 LCD 驱动模块

AD18F08 可驱动 1/2Bias 的 LCD，使能 LCD 控制位后，芯片需程序控制输出驱动 LCD。

### 15.1 LCD 管脚设置

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

### 15.2 LCD 相关寄存器

LCDCON0(LCD 使能寄存器)

地址:0XF70

Bit	Name	Description	Attribute	Reset
7	LCDEN	LCD 使能寄存器 1: 使能 0: 禁止	R/W	0
6:2	Reserved			
1:0	LCDISEL	LCD 输出电流选择位 00: 100uA@5V 01: 200uA@5V 10: 400uA@5V 11: 800uA@5V	R/W	00

LCDCON1(LCD 控制寄存器)

地址: 0XF6F

Bit	Name	Description	Attribute	Reset
7	COM7EN	COM7 口使能(PA1) 1: 使能 0: 禁止	R/W	0
6	COM6EN	COM6 口使能(PA0) 1: 使能 0: 禁止	R/W	0
5	COM5EN	COM5 口使能(PB0) 1: 使能 0: 禁止	R/W	0

Bit	Name	Description	Attribute	Reset
4	COM4EN	COM4 口使能(PB1) 1: 使能 0: 禁止	R/W	0
3	COM3EN	COM3 口使能(PB2) 1: 使能 0: 禁止	R/W	0
2	COM2EN	COM2 口使能(PB3) 1: 使能 0: 禁止	R/W	0
1	COM1EN	COM1 口使能(PB4) 1: 使能 0: 禁止	R/W	0
0	COM0EN	COM0 口使能(PB5) 1: 使能 0: 禁止	R/W	0

### LCDCON2(LCD 控制寄存器)

地址: 0XF6E

Bit	Name	Description	Attribute	Reset
7:6	Reserved			
5	COM13EN	COM13 口使能(PB6) 1: 使能 0: 禁止	R/W	0
4	COM12EN	COM12 口使能(PB7) 1: 使能 0: 禁止	R/W	0
3	COM11EN	COM11 口使能(PA5) 1: 使能 0: 禁止	R/W	0
2	COM10EN	COM10 口使能(PA4) 1: 使能 0: 禁止	R/W	0
1	COM9EN	COM9 口使能(PA3)	R/W	0



Bit	Name	Description	Attribute	Reset
		1: 使能 0: 禁止		
0	COM8EN	COM8 口使能(PA2) 1: 使能 0: 禁止	R/W	0

## 16 8X8 硬件乘法器

AD18F08 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 AD18F08 器件。

# 17 电气特性

## 17.1 直流交流电气特性

符号	描述	最小值	典型值	最大值	单位	条件(25°C)
VDD	工作电压	2.0	5	5.5	V	
F <sub>sys</sub>	系统时钟(CLK)					
	HIRC/2	0		8M	Hz	VDD=5V
	HIRC/4	0		4M		
	HIRC/8	0		2M		
	HIRC/16	0		1M		
	LIRC		2K			
I	PWSAVE 模式下电流		30			
I	DEEPPWSAVE 模式下电流		4		μA	VDD=5V
I	PWOFF 模式下电流		1.5		μA	VDD=5V
V <sub>IL</sub>	输入低电压	0		0.2*VDD	V	SMTV=0
		0		0.3*VDD		SMTV=1
V <sub>IH</sub>	输入高电压	0.4*VDD		VDD	V	SMTV=0
		0.7*VDD		VDD		SMTV=1
I <sub>OH</sub>	IO 输出拉电流		10		mA	VIO =4.5V
I <sub>OL</sub>	IO 输出灌电流 (除 PA1、PA2 外)		3.5		mA	VIO=0.5V, CUR=0
			20		mA	VIO=0.5V, CUR=1
	PA1, PA2		20		mA	VIO=0.5V, CUR=0
			60		mA	VIO=0.5V, CUR=1
R <sub>PH</sub>	上拉电阻		30		KΩ	RSEL=1
			100		KΩ	RSEL=0
R <sub>PL</sub>	下拉电阻		30		KΩ	RSEL=1
			300		KΩ	RSEL=0
f <sub>HIRC</sub>	校准后的 HIRC 频率	-2.0%*	16	+2.0%*	MHz	25°C, 2.4V~5.5V

符号	描述	最小值	典型值	最大值	单位	条件(25°C)
		-2.0%*	16	+2.0%*		0°C~70°C 2.4V~5.5V
V <sub>ADC</sub>	ADC 工作电压	2.7		VDD	V	
V <sub>AD</sub>	ADC 的输入电压	0		VDD	V	
AD <sub>rs</sub>	ADC 分辨率		12		Bit	
T <sub>ADCONV</sub>	ADC 转换时间(T <sub>ADCLK</sub> 是选定 AD 转换时钟周期)		15		T <sub>ADCLK</sub>	
V <sub>REFH</sub>	ADC 的参考电压				V	VDD=5V
	1V	0.99*	1	1.01*		
	2V	1.98*	2	2.02*		
	3V	2.97*	3	3.03*		
T <sub>WDT</sub>	看门狗超时溢出时间		8.32		S	111
			4.224			110
			2.176			101
			1.152			100
			0.640			011
			0.384			010
			0.256			001
			0.128			000

注：\*参数是设计参考值，并未在生产时对每个芯片进行测试。

实测功耗：

工作模式	电压点/V		3.3V	5V
	2T	16MHz		2327uA
8MHz			1400uA	2093uA
4MHz			909uA	1302uA
2MHz			650uA	879uA
1MHz			532uA	674uA

	4T	16MHz	1554uA	2285uA
		8MHz	1019uA	1429uA
		4MHz	749uA	1006uA
		2MHz	615uA	779uA
		1MHz	560uA	670uA
POWER IDLE 模式	-	-	508uA	703uA
POWER SAVE 模式	-	-	25.5uA	27.8uA
DeepSleep 模式	-	-	1.3uA	1.8uA
PowerOff 模式	-	-	0.8uA	1.3uA

注:\*参数是设计参考值，并未在生产时对每个芯片进行测试。

## 17.2 FLASH

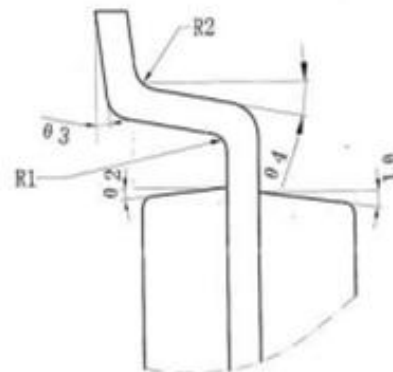
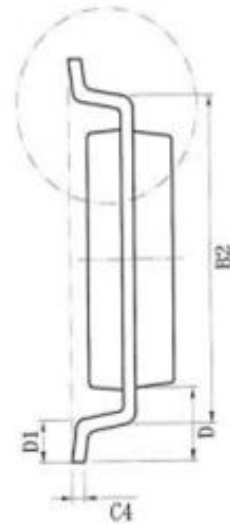
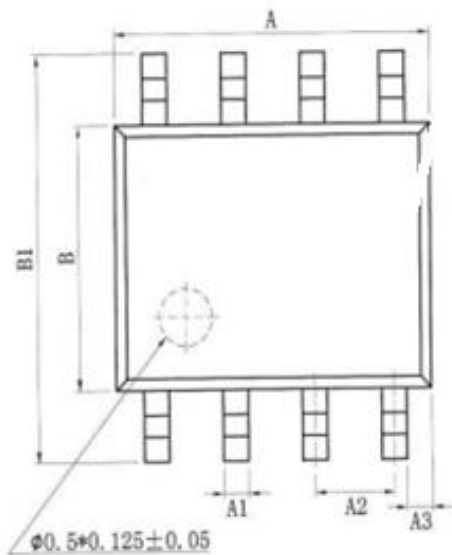
温度：-40°C~85°C

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

# 18 封装信息

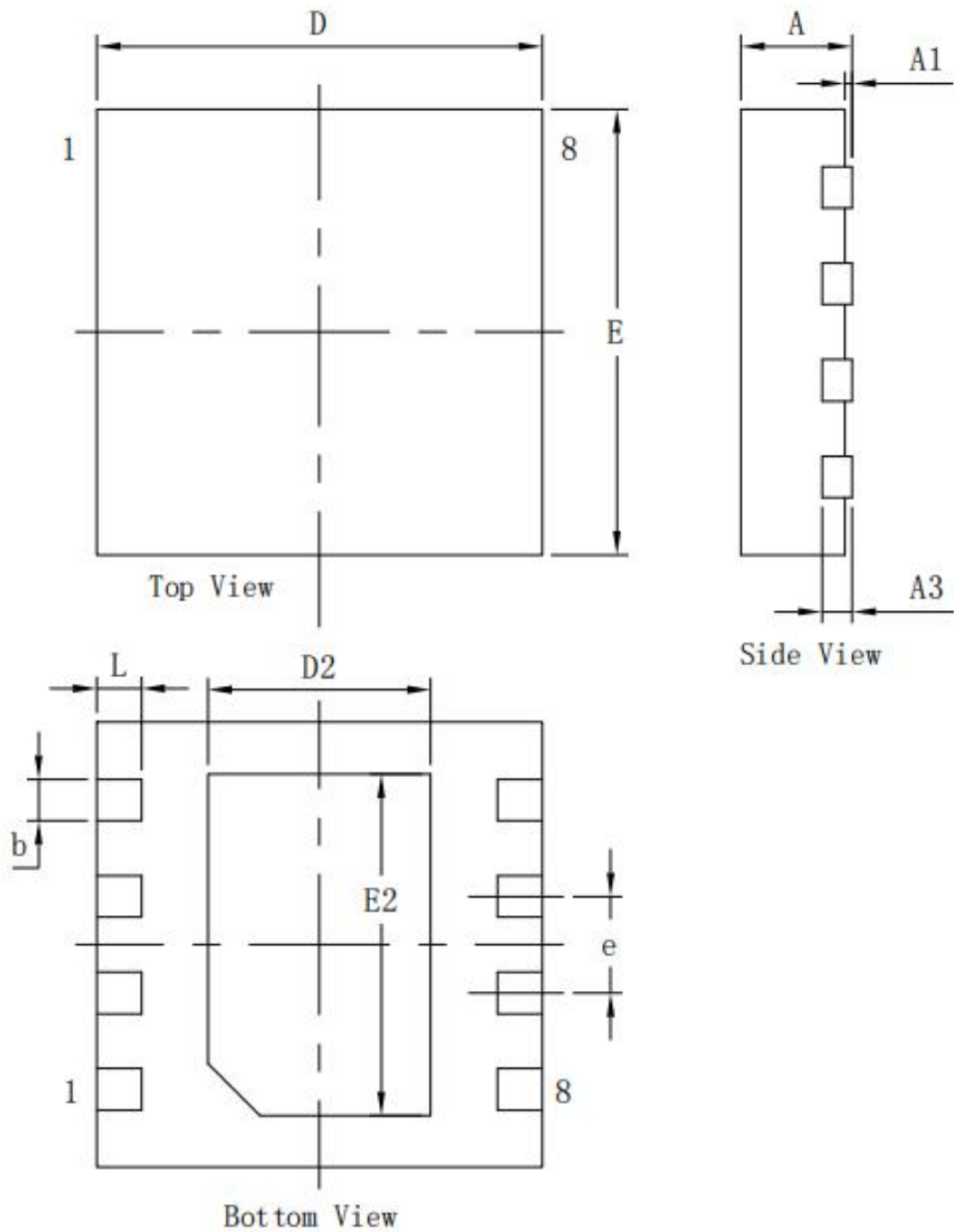
SOP8

标注	尺寸	最小 (MM)	最大 (MM)	标注	尺寸	最小 (MM)	最大 (MM)
A		4.80	5.00	C4		0.2	0.25
A1		0.380	0.470	D		1	1.1
A2		1.245	1.295	D1		0.50	0.70
A3		0.345TYP		R1		0.20TYP	
B		3.80	4.00	R2		0.20TYP	
B1		5.90	6.10	θ 1		11°	13°
B2		5.00TYP		θ 2		11°	13°
C		1.42	1.48	θ 3		0° -8°	
C1		0.675	0.725	θ 4		4° -12°	
C2		0.675	0.725				
C3		0.05	0.10				



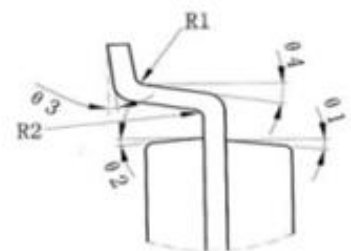
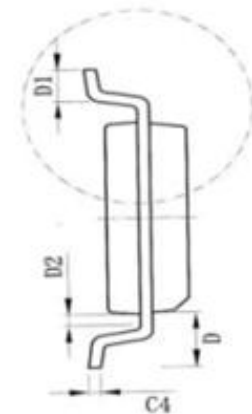
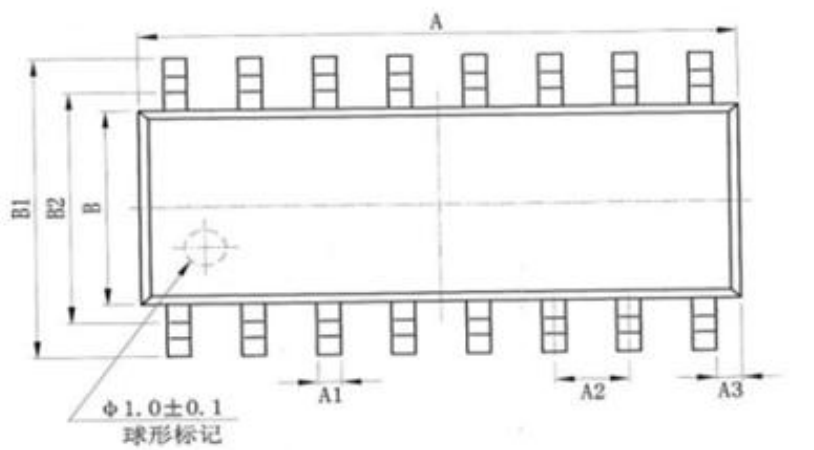
## DFN8

标注	尺寸	最小 (mm)	标准 (mm)	最大 (mm)	标注	尺寸	最小 (mm)	标准 (mm)	最大 (mm)
A		0.70	0.75	0.80	E		2.90	3.00	3.10
A1		-	-	0.05	D2		1.40	1.50	1.60
A3		0.203 REF			E2		2.20	2.30	2.40
b		0.23	0.28	0.33	e		0.65 TYP		
D		2.90	3.00	3.10	L		0.25	0.30	0.35



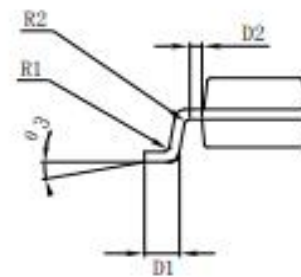
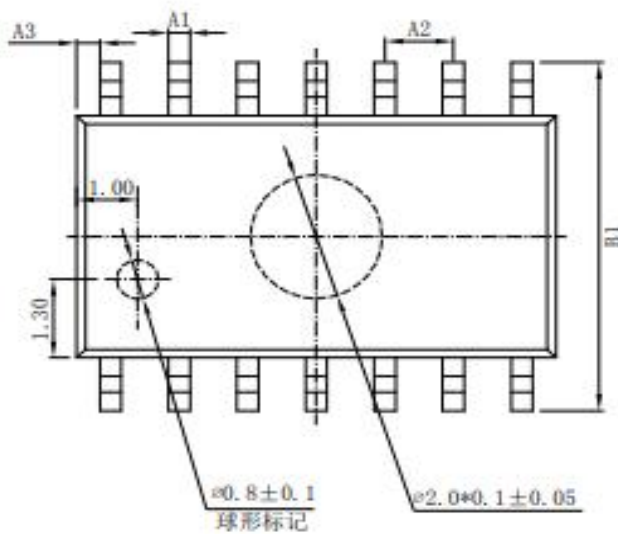
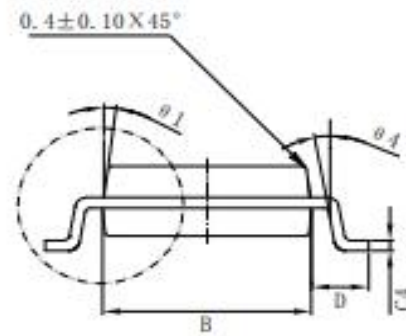
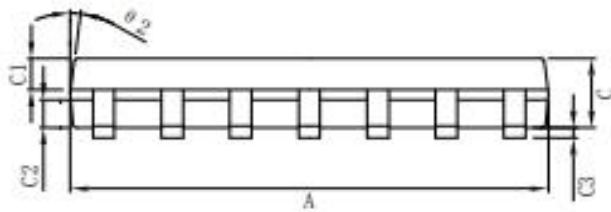
## SOP16

标注	尺寸	最小(MM)	最大(MM)	标注	尺寸	最小(MM)	最大(MM)
A		9.85	9.95	C4		0.2	0.25
A1		0.380	0.470	D		1	1.1
A2		1.245	1.295	D1		0.50	0.70
A3		0.302TYP		D2		0.15	0.25
B		3.875	3.925	R1		0.20TYP	
B1		5.90	6.10	R2		0.20TYP	
B2		5.00TYP		θ 1		11°	13°
C		1.42	1.48	θ 2		11°	13°
C1		0.675	0.725	θ 3		0° -8°	
C2		0.675	0.725	θ 4		4° -12°	
C3		0.15	0.25				



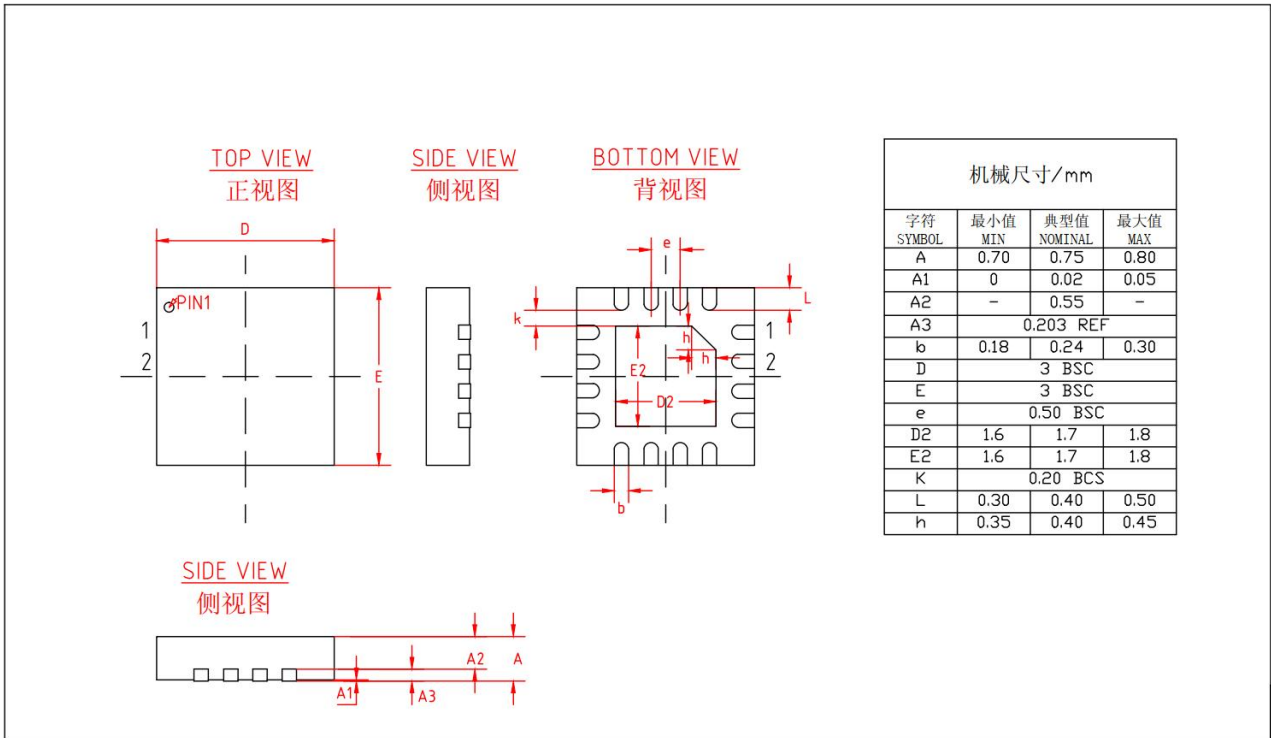
## SOP14

标注	尺寸	最小(mm)	最大(mm)	标注	尺寸	最小(mm)	最大(mm)
A		8.55	8.75	C4		0.193	0.213
A1		0.356	0.456	D		0.95	1.15
A2		1.27TYP		D1		0.40	0.70
A3		0.312TYP		D2		0.20TYP	
B		3.80	4.00	R1		0.20TYP	
B1		5.80	6.20	R2		0.20TYP	
C		1.40	1.60	θ1		8° ~ 12° TYP4	
C1		0.60	0.70	θ2		8° ~ 12° TYP4	
C2		0.55	0.65	φ3		0° ~ 8°	
C3		0.05	0.25	φ4		4° ~ 12°	





QFN16(3\*3)



## 19 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

F808	2	28	1	x
Device code	Year	Week	SeriesNo	Internal No
Year: 2:2022; 3: 2023				
Week: 01:第1周; 28:第28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

封装信息

芯片批号

丝印信息

出货日期

包装数量

Part No: AD18F08Q16

Package:QFN16

Lot No: NGE703000

Marking: F808 32812

Date: 2023-09-27

QTY: \*\*\*\*\*

**空片**

### 采购信息

AD18F08		
产品名称	封装信息	工作温度
AD18F08S16-XT	SOP16, 绿色封装	-40~85°C
AD18F08S8-XT	SOP8, 绿色封装	-40~85°C
AD18F08D8	DFN8(3*3), 绿色封装	-40~85°C
AD18F08Q16	QFN16(3*3), 绿色封装	-40~85°C
AD18F08S14-2	SOP14-2, 绿色封装	-40~85°C