



西安恩狄集成电路有限公司

Analog & Digital Microelectronics Limited

AD18E211 用户手册

版权所有©

西安恩狄集成电路有限公司

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

西安恩狄集成电路有限公司

地 址：陕西省西安市高新区高新一路 19 号思安大厦 401

地 址：深圳市 宝安区 西乡大道 230 号 满京华艺峦大厦 4 座 1505 室

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

微信号：恩狄 ADUC



目录

| | |
|----------------------------|----|
| 产品修改信息 | 4 |
| 1 产品简介 | 5 |
| 1.1 功能特性 | 5 |
| 1.2 系统结构 | 7 |
| 1.3 引脚排列 | 8 |
| 1.4 引脚说明 | 9 |
| 1.5 绝对最大额定值 | 12 |
| 2 中央处理器 | 13 |
| 2.1 指令集 | 13 |
| 2.2 程序存储 | 13 |
| 2.2.1 程序计数器 | 13 |
| 2.2.2 返回地址堆栈 | 13 |
| 2.2.3 栈顶访问 | 13 |
| 2.2.4 返回堆栈指针(STKPTR) | 14 |
| 2.2.5 快速寄存器堆栈 | 14 |
| 2.2.6 程序存储器 | 15 |
| 2.3 数据存储器 RAM | 15 |
| 2.4 数据寻址方式 | 16 |
| 2.4.1 固有和立即数寻址 | 16 |
| 2.4.2 直接寻址 | 16 |
| 2.4.3 间接寻址 | 16 |
| 2.5 配置选项 | 17 |
| 3 MTP | 20 |
| 3.1 MTP 程序存储器 | 20 |
| 3.2 IAP | 21 |
| 3.3 注意事项 | 22 |
| 4 系统时钟源 | 23 |
| 4.1 系统时钟相关寄存器 | 23 |
| 5 复位 | 25 |
| 5.1 初始化配置时序 | 25 |
| 5.2 上电复位 | 26 |
| 5.3 低电压复位 | 27 |
| 5.4 上电复位延时 | 27 |
| 5.5 非法指令复位 | 27 |
| 5.6 软件复位 | 27 |

| | | |
|-------|------------------------------|----|
| 5.7 | 堆栈溢出复位 | 27 |
| 6 | I/O 端口 | 30 |
| 6.1 | IO 工作模式 | 30 |
| 6.1.1 | PA 口 | 30 |
| 6.1.2 | PB 口 | 33 |
| 6.1.3 | INT0/INT1/INT2/INT3 中断 | 35 |
| 7 | 定时器 | 39 |
| 7.1 | Timer0/Prescler/BUZZER | 39 |
| 7.1.1 | Prescaler | 39 |
| 7.1.2 | BUZZER | 39 |
| 7.2 | TIMER1 定时/计数和捕捉 | 42 |
| 7.2.1 | Timer1 的定时计数 | 42 |
| 7.2.2 | Timer1 捕捉模式 | 43 |
| 7.3 | TIMER2 定时器和 3 路独互补 PWM | 48 |
| 7.3.1 | Timer2 计数/定时 | 48 |
| 7.3.2 | PWM | 48 |
| 7.3.3 | HBRIDGE | 52 |
| 7.3.4 | 寄存器 | 57 |
| 7.4 | 看门狗定时器(WDT) | 63 |
| 8 | UART | 65 |
| 8.1 | 相关寄存器 | 65 |
| 8.2 | 波特率发生器 | 67 |
| 8.3 | 异步发送 | 68 |
| 8.4 | 异步接收 | 69 |
| 8.5 | 单线半双工 | 70 |
| 9 | 固定参考电压(FVR) | 72 |
| 10 | 模数转换器(ADC) | 74 |
| 11 | 数模转换器(DAC) | 79 |
| 11.1 | DAC 参考电压选择寄存器 | 79 |
| 12 | 比较器 | 82 |
| 12.1 | 比较器 CMP0 | 82 |
| 13 | 运放 OP | 85 |
| 13.1 | 运放 OP | 85 |
| 13.2 | OP 比较器模式 | 88 |
| 14 | 中断 | 89 |
| 14.1 | 外部中断 | 89 |
| 14.2 | Timer0 中断 | 90 |

| | | |
|--------|-------------------------------|-----|
| 14.3 | Timer1 溢出中断以及上升沿和下降捕捉中断 | 90 |
| 14.4 | Timer2 溢出中断 | 90 |
| 14.5 | PWM20、PWM22、PWM24 中断 | 90 |
| 14.6 | UART 接收、发送中断 | 90 |
| 14.7 | 比较器中断 | 91 |
| 14.8 | OP 中断 | 91 |
| 14.9 | ADC 中断 | 91 |
| 14.10 | 中断的相关寄存器 | 91 |
| 15 | 8X8 硬件乘法器 | 97 |
| 16 | 省电模式(SLEEP) | 98 |
| 16.1 | 睡眠唤醒 | 98 |
| 17 | 电气特性 | 100 |
| 17.1 | 直流交流电气特性 | 100 |
| 17.2 | HIRC 频率微调参数说明 | 101 |
| 17.3 | HIRC 温漂曲线 | 104 |
| 17.4 | LIRC 温漂曲线 | 105 |
| 18 | 特别注意事项 | 106 |
| 18.1 | 看门狗 | 106 |
| 18.2 | IC 烧录方法 | 106 |
| 18.2.1 | 烧录引脚及方法 | 106 |
| 18.2.2 | 烧录电压条件 | 107 |
| 18.2.3 | 烧录注意事项 | 107 |
| 19 | 封装尺寸 | 108 |
| 19.1 | SOP16 | 108 |
| 19.2 | QFN16 | 108 |
| 19.3 | SO14 | 109 |
| 19.4 | SOP8 | 109 |
| 19.5 | SOT23-6 | 110 |
| 20 | 订购信息 | 111 |
| | 免责声明 | 112 |

产品修改信息

| Bin | version | Change List | Owner |
|-----|---------|---|-------|
| 1 | 1.0.0 | 初版手册 | Minjw |
| 2 | 1.0.1 | 1、修改 Timer2 的原理结构图 2、修改 OSCF 和 OSCM 的描述 3、修改 FVR 结构图和 OP 中描述 4、修改烧录注意事项中 VPP 的电阻大小，将 200 Ω 修改为 100 Ω 5、修改 CMP0CON2 的默认值 | Minjw |
| 3 | 1.0.2 | 1、修改 SOP14 的引脚顺序 2、修改级联烧录引脚图 | Minjw |
| 4 | 1.0.3 | 修改 SOP8 的引脚顺序 | Minjw |

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*16BIT MTP，支持 9.5V VPP 电压下，100 次擦除写入；
 - ✧ 支持非法指令复位功能；支持复位指令；
 - ✧ 支持 TypeC 口带电在线升级程序，无限断点设定和调试
 - ✧ 支持 TypeC 口双芯片级联带电在线升级程序，支持级联烧录，支持 VPP 单线更新 IAP 数据
 - ✧ 支持支持最后 32*8BIT 区域的 IAP 写入
 - ✧ 数据空间：192*8BIT SRAM
 - ✧ 堆栈：支持 8 层硬件堆栈嵌套
- 支持 2T 和 4T 的 CPU 运算模式；
- 内置双时钟模式，高速可以倍频到 32MHz，可选 16M/8M/4M/2M/1M；超低功耗 32K 时钟；通过寄存器可实时切换 CPU 时钟；
- CPU 有 PWDEEP, PWOFF 两种低功耗工作模式；
 - ✧ PWDEEP: CPU 停止工作，高速 16M 时钟停止工作，低速 32K 时钟工作，SRAM 数据保持；支持外部中断、比较器、看门狗溢出和 TIMER1 的 32K 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，支持外部中断，唤醒后继续从当前 PC 运行；
- TIMER0, 8 位定时/计数器，可选 FCPU、OSC32K(32K 时钟)、T0CKI、比较器输出作为 TIMER0 时钟，同时支持 BUZZER 输出模式
- TIMER1, 16 位定时/计数器，低功耗定时器(Power Deep 32K 定时)，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 管脚捕捉模式和脉宽测量功能；支持自启动 ADC 功能；
- TIMER2, 10 位定时计数器，支持 CPU 中断，时钟源为 Fsys，最高可以跑 32MHz；支持调频，PR2 写入值更新为 TIMER2 溢出时间；三路同周期独立占空比的 PWM，支持多种死区时间设置的 3 路互补 PWM 输出；可以根据 PWM 边沿，延时启动 ADC 采集；
- 一路 UART，支持全双工、单线半双工，含波特率发生器，可以配置 8/9 比特收发功能
- 内置高精度电压源，提供电压 1.2V、2.0V、3.0V 和 4.0V；
- 16 通道的，12bit 高精度的 ADC，支持 14 个 IO 引脚采集，支持 FVR 采集；
- LVR 提供 7 种低电压选择：2.0V、2.1V、2.2V、2.4V、2.6V、2.8V 和 3.0V；
- 一路比较器，支持迟滞，支持 offset 校准，用于比较电压的大小；支持最长 32us(16M/2T)数字滤波输出；
- 支持一路 DAC，可以选择 VDD 和 FVR 做参考源；
- 一路模拟运放，支持信号单端和差分放大；支持配置成比较器模式
- 超强程序加密算法，保证芯片程序内容唯一性；
- 除跳转指令为两个周期指令以外其余为单周期指令；
- 运行速度:支持最高 32MHz/4T 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;
- 15 个通用 GPIO
 - ✧ PB6 为全 IO, 支持 VPP 输入;
 - ✧ 所有 IO 内置上下拉电阻, 支持 SMT/TTL 设定; 支持开漏设定, 并支持 $30\%*VDD/70\%*VDD$ 施密特输入和 TTL 输入;
 - ✧ 所有 IO 支持 INT 中断;
 - ✧ 所有 IO 支持 $VDD=5V$, $I_{Source}=20mA@V_{OH}=90\%*VDD$, $I_{Sink}=60mA@V_{OL}=10\%*VDD$
- 中断:
 - ✧ 3 个内部计数/定时器中断;
 - ✧ 2 个捕捉中断和 1 个门控中断;
 - ✧ 四个外部 INT 中断, 可以映射所有 IO;
 - ✧ PWM20、PWM22、PWM24 占空比中断;
 - ✧ 串口收发中断;
 - ✧ 1 个模拟比较器中断;
 - ✧ 1 个模拟运放中断;
 - ✧ ADC 中断;
- 通过外部中断、TIMER1 中断、WDT 溢出实现睡眠模式唤醒;
- 有可靠的保证使得程序代码不被读出;
- CPU 工作频率 16MHz/4T 模式, VDD 工作电压范围: 2.0V-5.5V; CPU 工作的最高频率为 16MHz/2T 模式或者 32MHz/4T 模式, VDD 工作电压范围: 2.6V-5.5V;
- 封装类型: QFN16/SOP14/SOP16/SOP8/SOT236;

1.2 系统结构

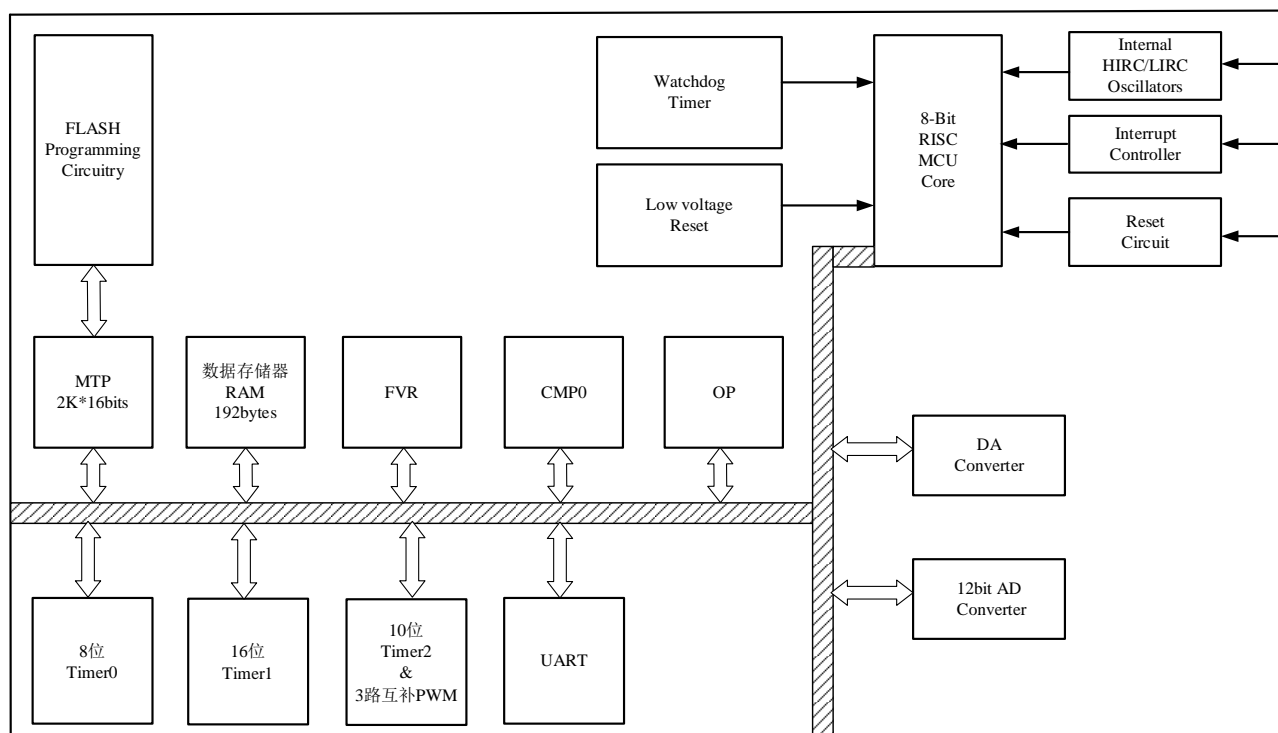


图1 系统结构图

1.3 引脚排列

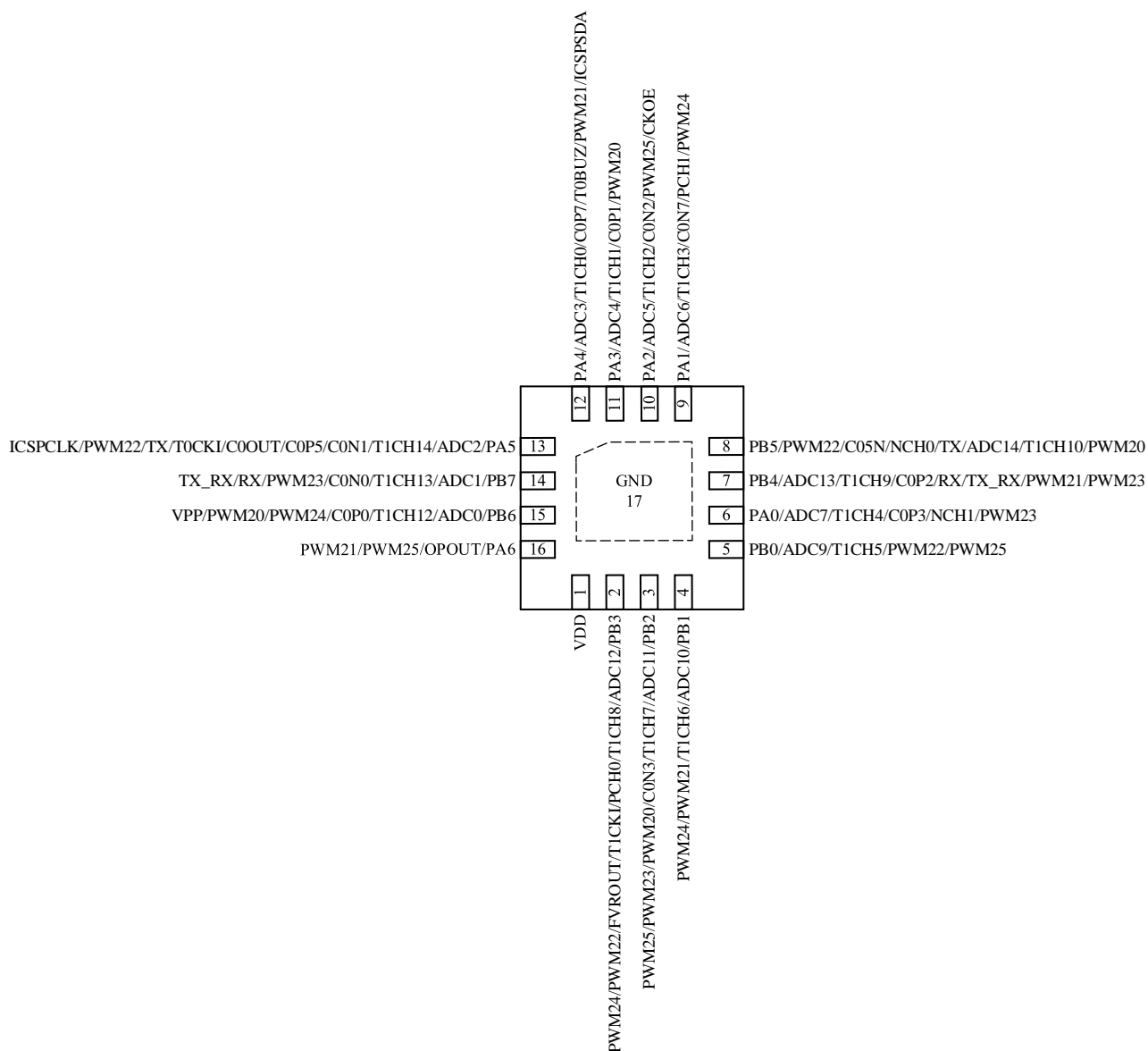


图2 AD18E211 Q16

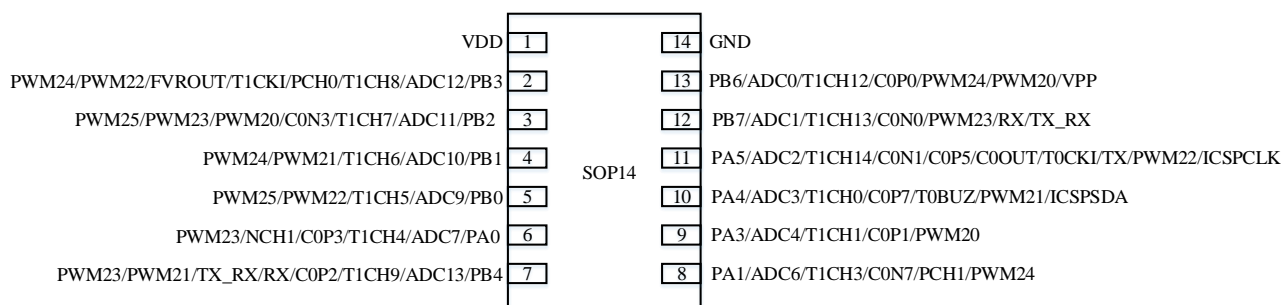


图3 AD18E211 SOP14

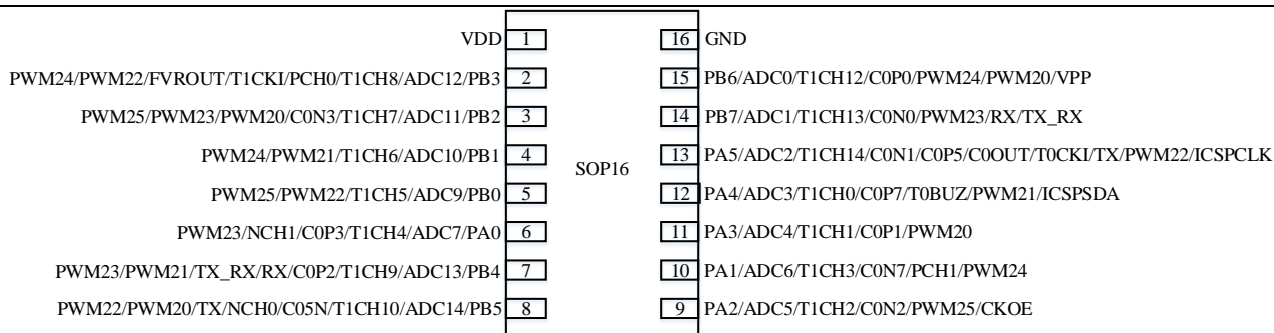


图4 AD18E11 SOP16

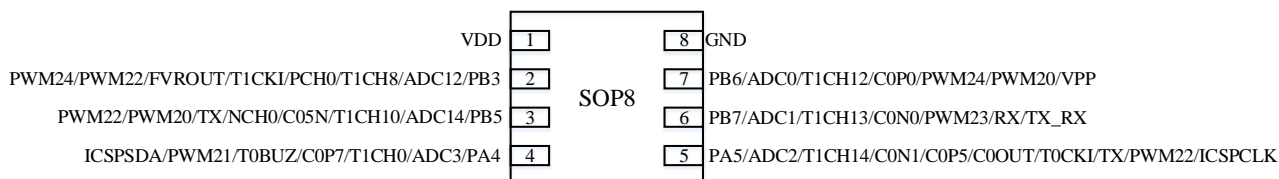


图5 AD18E211 SOP8

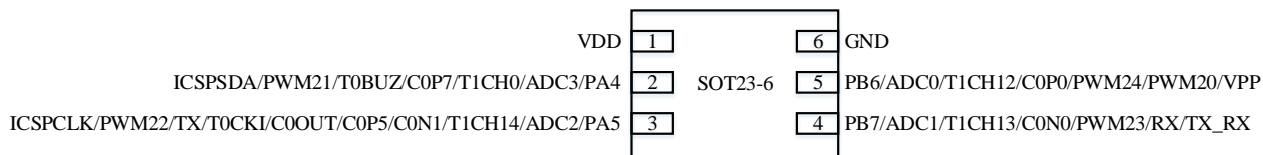


图6 AD18E211 SOT23-6

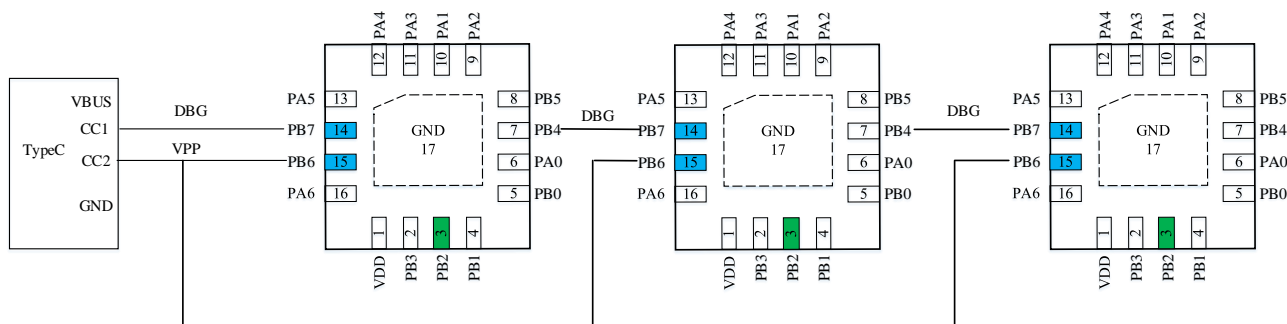


图7 AD18E211 多芯片级联烧录

1.4 引脚说明

| 管脚名 | QFN16引脚 | 功能名 | 输入类型 | 输出类型 | 具体描述 |
|-----|---------|---------|------|------|--------------|
| PA0 | 8 | PA0 | SMT | CMOS | GPIO |
| | | HBPWM23 | | CMOS | HBPWM23 输出引脚 |
| | | ADC7 | AN | | ADC 通道 7 |
| | | T1CH4 | SMT | | Timer1 的捕捉通道 |
| | | C0P3 | AN | | CMP0 正端输入 |
| | | NCH1 | AN | | OP 负端输入 |
| PA1 | 9 | PA1 | SMT | CMOS | GPIO |

| | | | | | |
|-----|----|----------|-----|------|----------------|
| | | HBPWM24 | | CMOS | HBPWM24 输出引脚 |
| | | C0N7 | AN | | CMP0 负端输入 |
| | | PCH1 | AN | | OP 正端输入 |
| | | ADC6 | AN | | ADC 通道 6 |
| | | T1CH3 | | | Timer1 的捕捉通道 |
| PA2 | 10 | PA2 | SMT | CMOS | GPIO |
| | | HBPWM25 | | CMOS | HBPWM25 输出引脚 |
| | | C0N2 | AN | | CMP0 的负端输入 |
| | | ADC5 | AN | | ADC 通道 5 |
| | | T1CH2 | SMT | | Timer1 的捕捉通道 |
| | | CKOE | SMT | | CKOE 输出 |
| PA3 | 11 | PA3 | SMT | CMOS | GPIO |
| | | HBPWM20 | AN | | HBPWM20 输出引脚 |
| | | C0P1 | AN | | CMP0 正端输入 |
| | | ADC4 | AN | | ADC 通道 4 |
| | | T1CH1 | SMT | | Timer1 的捕捉通道 |
| PA4 | 12 | PA4 | SMT | CMOS | GPIO |
| | | HBPWM21 | - | CMOS | HBPWM21 输出引脚 |
| | | C0P7 | AN | | CMP0 的正端引脚输入 |
| | | ADC3 | AN | | ADC 通道 3 |
| | | T0BUZZER | | CMOS | TMR0 BUZZER 输出 |
| | | T1CH0 | SMT | - | Timer1 的捕捉通道 |
| | | SDA | SMT | CMOS | 烧录数据引脚 |
| PA5 | 13 | PA5 | SMT | CMOS | GPIO |
| | | HBPWM22 | - | CMOS | HBPWM22 输出引脚 |
| | | C0N1 | AN | | CMP0 的负端引脚输入 |
| | | C0P5 | AN | | CMP0 的正端引脚输入 |
| | | ADC2 | AN | | ADC 通道 2 |
| | | C0OUT | | CMOS | CMP0 的输出引脚 |
| | | T1CH14 | SMT | | Timer1 的捕捉通道 |
| | | T0CKI | SMT | | TMR0 时钟输入引脚 |
| | | TX | | CMOS | UART 发送引脚 |
| PA6 | 16 | SCL | SMT | CMOS | 烧录时钟引脚 |
| | | PA6 | SMT | CMOS | GPIO |
| | | HBPWM25 | - | CMOS | HBPWM25 输出引脚 |
| | | HBPWM21 | - | CMOS | HBPWM21 输出引脚 |
| PB0 | 7 | OP_OUT | AN | | OP 输出引脚 |
| | | PB0 | SMT | CMOS | GPIO |
| | | HBPWM22 | - | CMOS | HBPWM22 输出引脚 |
| | | HBPWM25 | - | CMOS | HBPWM25 输出引脚 |
| | | ADC9 | AN | | ADC 通道 9 |
| PB1 | 6 | T1CH5 | | | Timer1 的捕捉通道 |
| | | PB1 | SMT | CMOS | GPIO |
| | | HBPWM21 | | CMOS | HBPWM21 输出 |

| | | | | | |
|-----|----|---------|-----|------|---------------------|
| | | HBPWM24 | | CMOS | HBPWM24 输出 |
| | | ADC10 | AN | | ADC 通道 10 |
| | | T1CH6 | | | Timer1 的捕捉通道 |
| PB2 | 5 | PB2 | SMT | CMOS | GPIO |
| | | HBPWM20 | | CMOS | HBPWM20 输出 |
| | | HBPWM23 | | CMOS | HBPWM23 输出 |
| | | HBPWM25 | | CMOS | HBPWM25 输出 |
| | | C0N3 | AN | | CMP0 正端输入 |
| | | T1CH7 | SMT | | Timer1 的捕捉通道 |
| | | ADC11 | AN | | ADC 通道 11 |
| PB3 | 4 | PB3 | SMT | CMOS | GPIO |
| | | HBPWM22 | | CMOS | HBPWM25 输出 |
| | | HBPWM24 | | CMOS | HBPWM25 输出 |
| | | PCH0 | AN | | OP 正端输入 |
| | | FVR_OUT | AN | | FVR 的输出 |
| | | T1CKI | SMT | | TMR1 时钟输入 |
| | | T1CH8 | SMT | | Timer1 的捕捉通道 |
| PB4 | 3 | PB4 | SMT | CMOS | GPIO |
| | | HBPWM21 | | CMOS | HBPWM21 输出 |
| | | HBPWM23 | | CMOS | HBPWM23 输出 |
| | | C0P2 | AN | | CMP0 正端输入 |
| | | ADC13 | AN | | ADC 通道 13 |
| | | RX | SMT | | UART 的接收引脚 |
| | | TX_RX | | CMOS | UART 的单线半双工通信 IO 引脚 |
| | | T1CH9 | SMT | | Timer1 的捕捉通道 |
| PB5 | 2 | PB5 | SMT | CMOS | GPIO |
| | | HBPWM20 | | CMOS | HBPWM20 输出 |
| | | HBPWM22 | | CMOS | HBPWM22 输出 |
| | | C0N5 | AN | | CMP0 负端输入 |
| | | NCH0 | AN | | OP 负端输入 |
| | | T1CH10 | SMT | | Timer1 的捕捉通道 |
| | | TX | SMT | CMOS | UART 输出引脚 |
| | | ADC14 | AN | | ADC 通道 14 |
| PB6 | 15 | PB6 | SMT | CMOS | GPIO |
| | | HBPWM24 | | CMOS | HBPWM24 输出 |
| | | HBPWM20 | | CMOS | HBPWM20 输出 |
| | | C0P0 | AN | | CMP0 正端输入 |
| | | VPP | AN | | 烧录电压输入引脚 |
| | | T1CH12 | SMT | | Timer1 的捕捉通道 |
| | | ADC0 | AN | | ADC 通道 0 |
| PB7 | 14 | PB7 | SMT | CMOS | GPIO |
| | | HBPWM23 | | CMOS | HBPWM23 输出 |
| | | C0N0 | AN | | CMP0 负端输入 |
| | | RX | SMT | | UART 的接收引脚 |

| | | | | | |
|-----|----|--------|-----|------|---------------------|
| | | TX_RX | | CMOS | UART 的单线半双工通信 IO 引脚 |
| | | T1CH13 | SMT | | Timer1 的捕捉通道 |
| | | ADC1 | AN | | ADC 通道 1 |
| VDD | 1 | VDD | | | 供电电源 |
| GND | 17 | GND | | | 地 |

1.5 绝对最大额定值

表 1-1 绝对最大额定值

| 符号 | 描述 | 最小值 | 最大值 | 单位 |
|------------------|-----------|---------|---------|----|
| VDD-GND | 电源电压 | -0.3 | +6.0 | V |
| V _{IN} | 端口输入信号电压 | GND-0.3 | VDD+0.3 | V |
| I _{VDD} | VDD 的最大电流 | - | +100 | mA |
| I _{GND} | GND 的最大电流 | - | -100 | mA |
| T _J | 最大结温 | - | +150 | °C |
| T _{STG} | 存储温度范围 | -55 | +150 | °C |
| T _A | 工作温度 | -40 | +85 | °C |
| ESD | 人体模式 | 2K | - | V |

注：这是一个额定值，如果对芯片的操作超过绝对最大额定值所规定的范围，将对芯片造成损坏。

2 中央处理器

2.1 指令集

AD18E211 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

AD18E211 包含一个 2K*16 的 MTP 用于存储用户程序和数据。

AD18E211 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 传送到程序计数器。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 12 位的 RAM 和 3 位的堆栈指针(STKPTR)来实现 8 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。

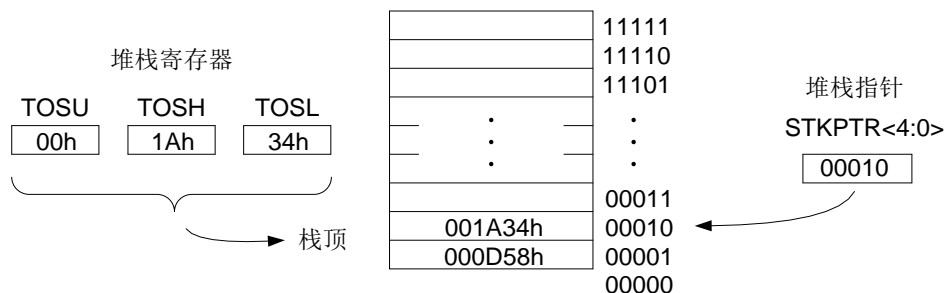


图8 返回地址堆栈

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 7 次(且没有值从堆栈弹出)后，STKFUL 位就会置 1。通过软件或 POR 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 7 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 7 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 7 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 | 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 | 0 |
| 5:3 | Reserved | | | |
| 2:0 | SP[2:0] | 堆栈指针地址位 | R/W | 000 |

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, AD18E211 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。

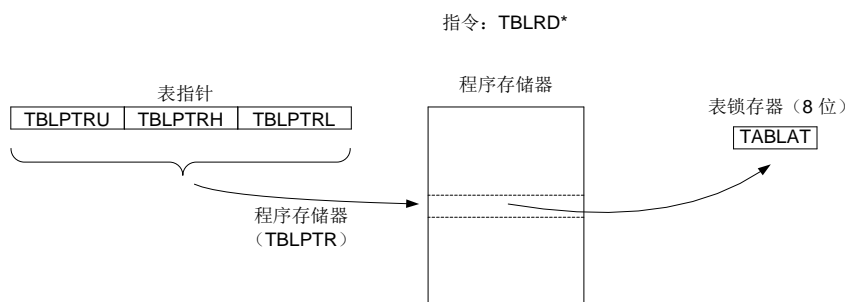


图9 表读操作

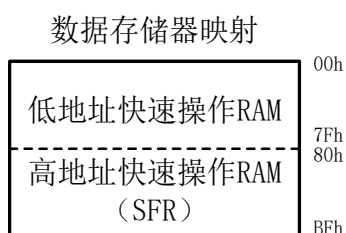
表锁存器(Table Latch,TABLAT)是映射到 SFR 空间的一个 8 位寄存器。表锁存器用于在程序存储器和数据 RAM 之间传输数据时保存 8 位数据。

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。

2.3 数据存储器 RAM

AD18E211 的数据存储器是用静态 RAM 实现的。存储空间包含 192 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。



2.4 数据寻址方式

AD18E211 支持 4 种寻址方式：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

2.4.1 固有和立即数寻址

很多 AD18E211 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。

2.4.2 直接寻址

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

2.4.3 间接寻址

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

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

2.4.3.1 FSR 寄存器和 INDF 操作数

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

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

由于间接寻址使用完整的 12 位地址，因此没有必要进行数据 RAM 分区。因此 BSR 的当前内容和快速操作 RAM 位对于确定目标地址没有影响。

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

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

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

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

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

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

2.4.3.3 通过 FSR 对其他 FSR 进行操作

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

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

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

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

2.5 配置选项

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] | <p>LVR 电压选择</p> <p>000: Reserved</p> <p>001: 2.0V</p> <p>010: 2.1V</p> <p>011: 2.2V</p> <p>100: 2.4V</p> <p>101: 2.6V</p> <p>110: 2.8V</p> <p>111: 3.0V</p> |
| 3 | LVREN | <p>LVR 使能选择</p> <p>1: 允许使能</p> <p>0: 禁止使能</p> |
| 4 | FCPU | <p>指令周期选择</p> <p>1: 1 个指令周期为 4 个机器周期</p> <p>0: 1 个指令为周期 2 个机器周期</p> |
| 7:5 | FINTOSC[2:0] | <p>内部 RC 振荡器频率选择</p> <p>111: 1:1 分频</p> <p>110: 1:2 分频</p> <p>101: 1:4 分频</p> <p>100: 1:8 分频</p> <p>01x: 1:16 分频</p> |

配置字 1

| 位 | 名称 | 说明 |
|-----|------------|--|
| 1:0 | WDTPS[1:0] | <p>看门狗溢出时间及上电复位时间选择</p> <p>11: TWDT(no Prescaler)=31.5ms</p> <p>10: TWDT(no Prescaler)=4.03ms</p> <p>01: TWDT(no Prescaler)=246.56ms</p> <p>00: TWDT(no Prescaler)=62.24ms</p> |
| 2 | WDTPS[2] | <p>上电复位延时时间选择(不需要使能 WDTE 或 WDTEN)</p> <p>1: 按照 WDTPS[1:0]位配置的分频时间进行上电延时</p> <p>0: 上电复位延时时间为 15 个时钟周期</p> |

| | | |
|-----|-----------------|---|
| 3 | WDTE | WDT 使能选择 1: 允许使能 WDT 0: 禁止使能 WDT |
| 4 | SMT | 电平选择位 1: 选择施密特 0: 选择 TTL 电平 |
| 6:5 | DBGEN_UART[1:0] | UART 调试模式使能选择 00: 禁止 DEBUG_UART 调试 01: 使能 1M 的波特率进行 DEBUG_UART 调试 10: 使能 115200 的波特率进行 DEBUG_UART 调试 11: 使能 VPP 为高, 切换成 PB7 为 DEBUG_UART 调试, 波特率 1M |
| 7 | DBGEN_I2C | I2C 调试模式使能选择 1: 使能 DBGEN_I2C 调试 0: 禁止 DBGEN_I2C 调试 |

3 MTP

MTP 可以自编程的地址空间从 0000H 到 1FFFH。
MTP 写入电压为 VPP：9.5V，支持 100 次擦除写入。

3.1 MTP 程序存储器

MTP 程序存储器用来存放执行的程序指令。MTP 程序存储器可以存储数据，包含：数据，表格和中断入口，程序从 0X0000 地址开始，中断入口地址为 0x0008H 和 0X0018H，MTP 程序存储器最后 16 个地址空间是保留给系统使用的,如校验、ID 号等。AD18E211 的存储空间为 2K*16；MTP 存储器从地址“0xFF0~0xFFF”供系统使用，从“0x0001~0x0007”和“0x0009~0X0017”和“0x0019~0xFEf”地址空间是用户的程序空间。

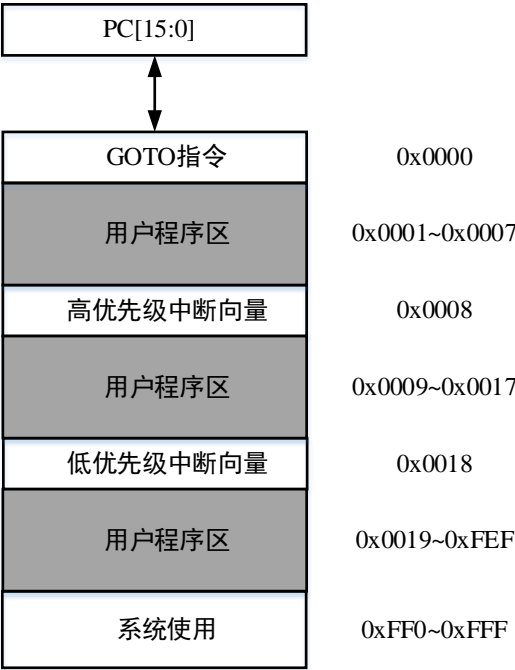


图11 AD18E211 存储器映射图

表 3-1 AD18E211 程序存储器结构

| 地址 | 功能 |
|---------------|------------|
| 0X0000 | GOTO 指令 |
| 0X0001~0X0007 | 用户程序区 |
| 0X0008 | 中断高优先级入口地址 |
| 0X0009~0X0017 | 用户程序区 |
| 0X0018 | 中断低优先级入口地址 |
| 0X0019~0XFEF | 用户程序区 |
| 0XFF0~0XFFF | 系统使用 |

3.2 IAP

MTP 支持 IAP 的地址空间为 16*16。

FLASH 写入数据操作步骤:

- (1) 写入电压为芯片工作电压; 进行读操作的电压为: 2.4V-5.5V; 进行写操作的电压为: 2.4V-5.5V, 且 VPP 输入 9.5V。
- (2) 对 IAPCTRL[7] 写入 0x1, 解除锁定状态, 进入到 IAP mode。注意: IAP 操作结束后应写入 0, 使芯片进入 IAPLOCK 状态。
- (3) 设置 IAPADDR, 设置数据要写入 FLASH 对应 00H-0FH 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) 将 PG 位置 1, CPU 将 IAPDAT[15:0]数据写入 IAPADDR 对应的 FLASH 地址中, 写入完成后 PG 位会自动清 0。
- (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 IAPCTRL[7]写入 0x0, 便可以退出写操作。

IAPCTRL 寄存器

地址: 0XF9E

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|--|-----------|-------|
| 7 | LOCK | IAP 锁定控制位 1: 解锁 0: 锁定 | R/W | 0 |
| 6 | Reserved | | | |
| 5 | PG | 写入控制位(写入完成后硬件自动清 0) 1: 使能写入控制 0: 写入完成 | R/W | 0 |
| 4 | ERR | FLASH 的擦写错误标志位 1: 以下情况发生时, 产生擦写错误标志位 一页擦除 ER 和写操作 PG 同时执行 一代码加密时进行页擦除操作或写操作 0: 未发生擦写错误 | R/W | 0 |
| 3:0 | IAPADDR[3:0] | IAP 地址 | R/W | 0000 |

IAPDATH 寄存器

地址: 0XF9D

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

| | | | | |
|-----|---------------|-----------|-----|---|
| 7:0 | IAPDATH[15:8] | IAP 数据高八位 | R/W | 0 |
|-----|---------------|-----------|-----|---|

IAPDATL 寄存器

地址：0XF 9C

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

3.3 注意事项

- 对 FLASH 编程时，应保证 VPP 输入电压 9.7V。VPP 引脚需要串联 200 欧电阻。
- IAP 操作时不建议使用于 AC 阻容降压供电或有高 EFT 要求的应用；
- IAP 写入时间要求芯片供电电压 5V;

4 系统时钟源

4.1 系统时钟相关寄存器

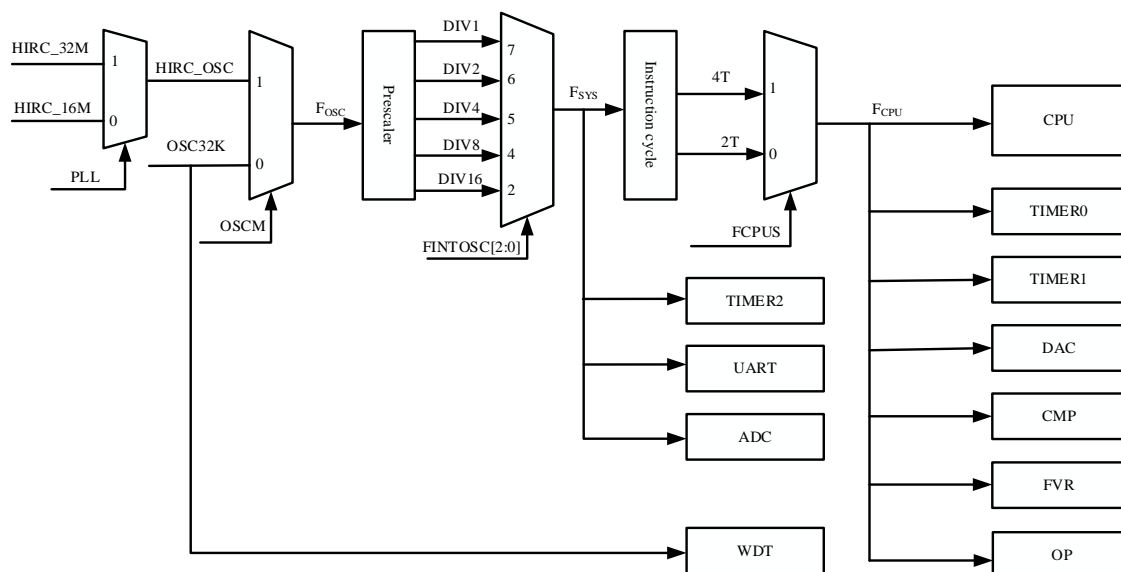


图12 系统时钟结构图

OSCCON 寄存器

地址：0XFF0

| 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 |
| 6:4 | 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 | 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 访问地址 0x00~0x1f | R/W | 0 |
| 2 | CKOE | PA2 输出当前系统时钟 | R/W | 0 |

| | | | | |
|---|------|--|-----|---|
| | | 1: 使能系统时钟输出 0: 禁止系统时钟输出 | | |
| 1 | OSCF | 工作时钟状态位 1: 表示内部高速时钟工作(HIRC) 0: 表示内部低速时钟工作(32KHz) | R | 0 |
| 0 | OSCM | 系统时钟选择 1: 选择内部高速时钟 HIRC 0: 选择内部低速时钟 32KHz | R/W | 0 |

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 | PLL | 时钟倍频选择 1: 系统时钟为 32MHz 0: 系统时钟为 16MHz | R/W | 0 |
| 5:4 | Reserved | | | - |
| 3:1 | PREDIV[2:0] | WDT 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 使能位: 1: 使能 WDT 0: 禁止 WDT | R/W | 0 |

5 复位

AD18E211 单片机能通过以下方式复位:

- 上电复位(POR)
- 掉电复位(BOR)
- 低电压复位
- 看门狗复位
- 非法指令复位
- 软件复位
- 堆栈溢出复位

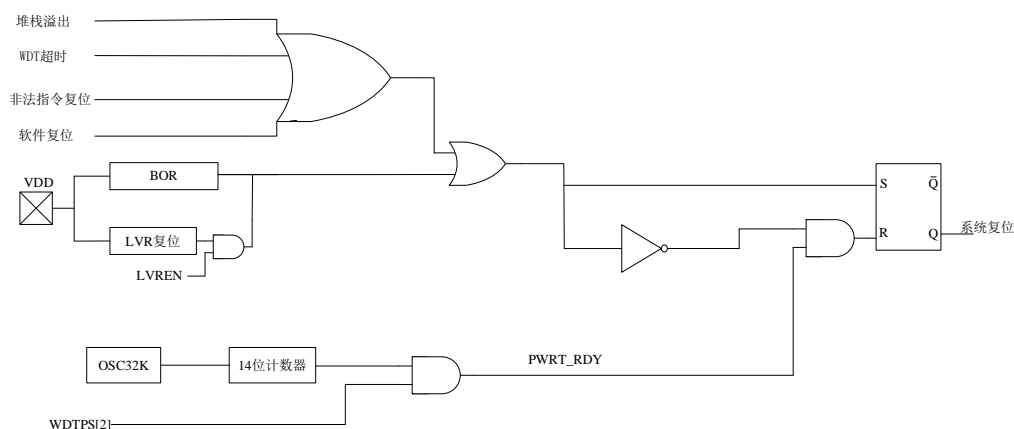


图13 复位框图

5.1 初始化配置时序

表 5-1 初始化配置

| 名称 | 功能 | 默认 |
|------|---|----|
| PWRT | 上电延时定时器，初始化配置完成后额外延时 4.03ms/31.5ms/62.24ms/246.56ms | 关闭 |

以上初始化配置，由 IDE 界面设置，不能通过指令修改，初始化配置过程：

- 1、CPU 空闲等待 0.5ms;
- 2、从非易失性存储器中加载初始化配置寄存器值，该过程 0.2ms。这些寄存器值由 IDE 预先设置，不受指令影响;
- 3、如果使能上电延时定时器(Power-On-Timer,PWRT)，CPU 将额外空闲等待时间通过配置字 1[2:0]选择;

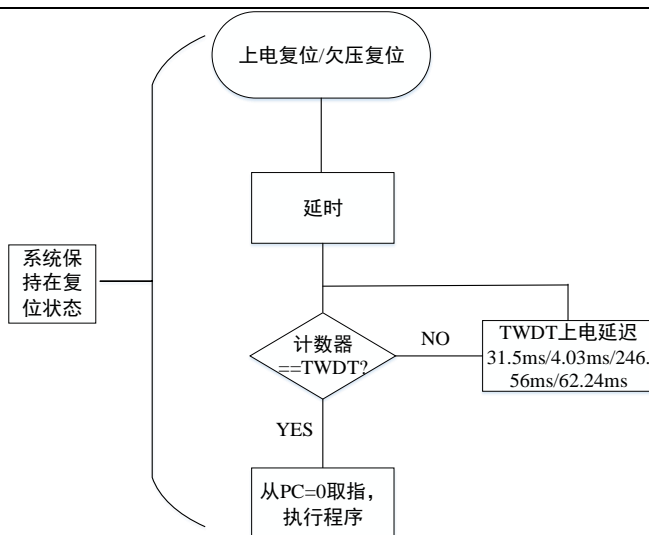


图16 上电复位流程图

5.3 低电压复位

低电压复位由配置字 0 控制，当 LVREN 使能，电源电压在设定值以下时，芯片发生低电压复位，直到 VDD 到达设定值以上，解除复位状态。

5.4 上电复位延时

内置一个 14 位的上电复位定时器 PWRT，PWRT 提供延迟时间(该延迟时间由 WDTPS[1:0](配置字 1)设置，只要 PWRT 在运行，设备就处于复位状态)(基于不同的振荡源和复位条件)在 Power-on Reset(POR)，看门狗溢出复位。VDD、温度和其他变化会影响设备延迟时间。

5.5 非法指令复位

IRF_STVREN 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]=1，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。

5.6 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[0]=1。

5.7 堆栈溢出复位

配置 IRF_STVREN(RCON)使能堆栈功能，溢出后复位，复位标志位 RSTF[2]=1。

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 | PLL | 时钟倍频选择 | R/W | 0 |

| | | | | |
|-----|-------------|--|-----|-----|
| | | 1: 系统时钟为 32MHz 0: 系统时钟为 16MHz | | |
| 5:4 | Reserved | | | - |
| 3:1 | PREDIV[2:0] | WDT 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 使能位: 1: 使能 WDT 0: 禁止 WDT | R/W | 0 |

RCON(寄存器)

地址: 0XFD2

| Bit | Name | Description | Attribute | Reset |
|-----|------------|--|-----------|-------|
| 7 | IRF_STVREN | 非法指令和堆栈溢出复位使能位 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 6 | RSTF[2] | 堆栈复位标志位 0: 未发生复位 1: 表示发生复位 | R/W | 0 |
| 5 | RSTF[1] | 非法指令复位标志位 0: 未发生复位 1: 表示发生复位 | R/W | 0 |
| 4 | RSTF[0] | 软件复位标志位 0: 未发生复位 1: 表示发生复位 | R/W | 0 |
| 3 | TO | 看门狗超时标志位 1: 通过上电、CLRWDT 指令或 SLEEP 指令置 1 | R/W | 1 |

| | | | | |
|---|-----|---|-----|---|
| | | 0: 发生了 WDT 溢出 | | |
| 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 | BOR | 掉电复位状态位 1: 未发生掉电复位 0: 已发生掉电复位(必须在掉电复位后由软件置 1) | R/W | 0 |

6 I/O 端口

PortA 和 PortB 双向三态 I/O 口。所有 I/O 的输入/输出方式由 I/O 控制寄存器(TRISA、TRISB)设置。PA、PB 通过上拉控制位(上拉寄存器)来设置使能内部上拉和下拉控制位(下拉寄存器)设置使能内部下拉，如果设置为输出模式，内部上拉功能会自动关闭，内部下拉功能不会自动关闭，需要自行关闭。PB 不仅有上/下拉控制位，还有开漏控制位(开漏寄存器)设置输出为开漏输出。

6.1 IO 工作模式

6.1.1 PA 口

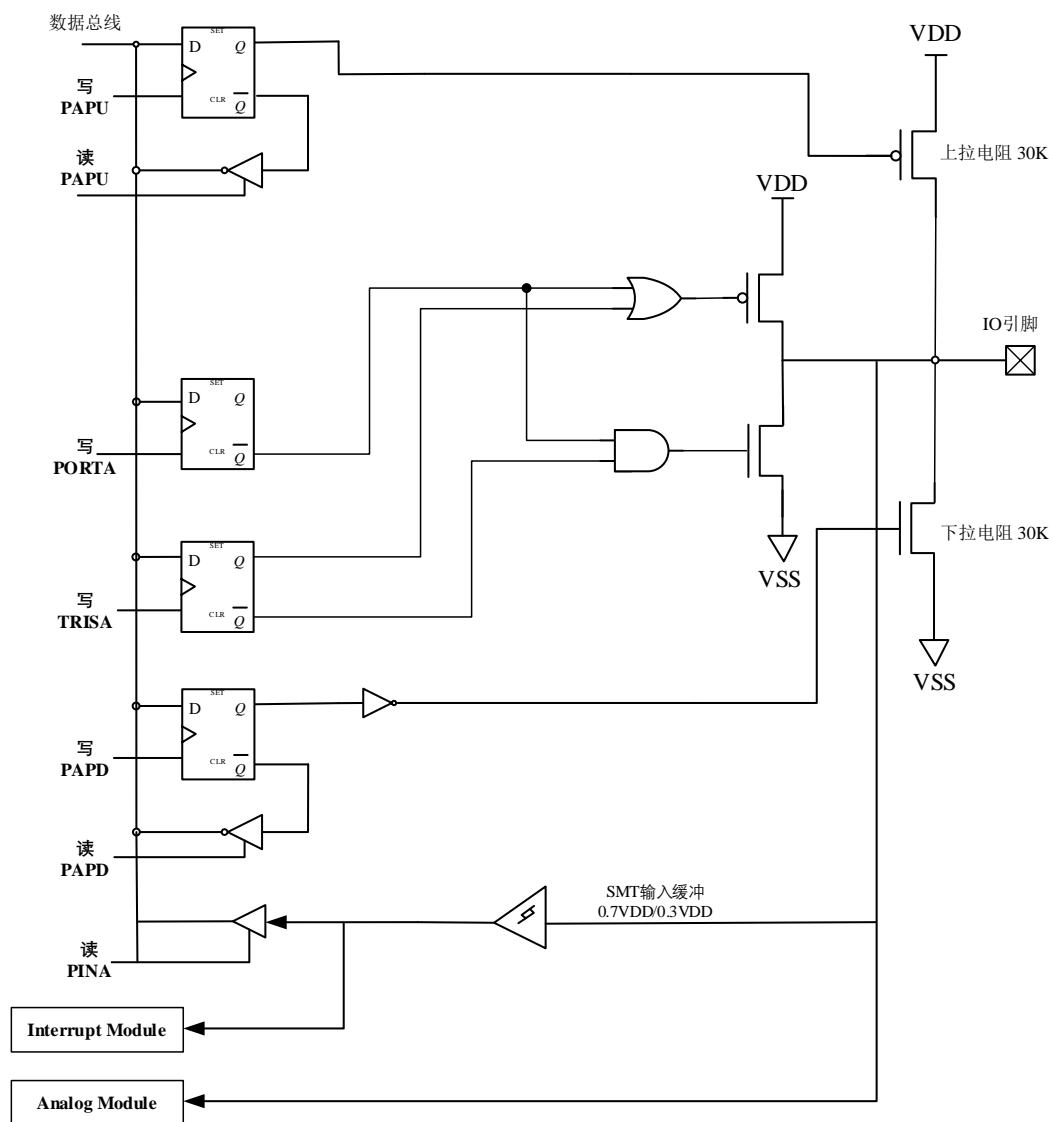


图17 PA0~3、PA6 口结构图

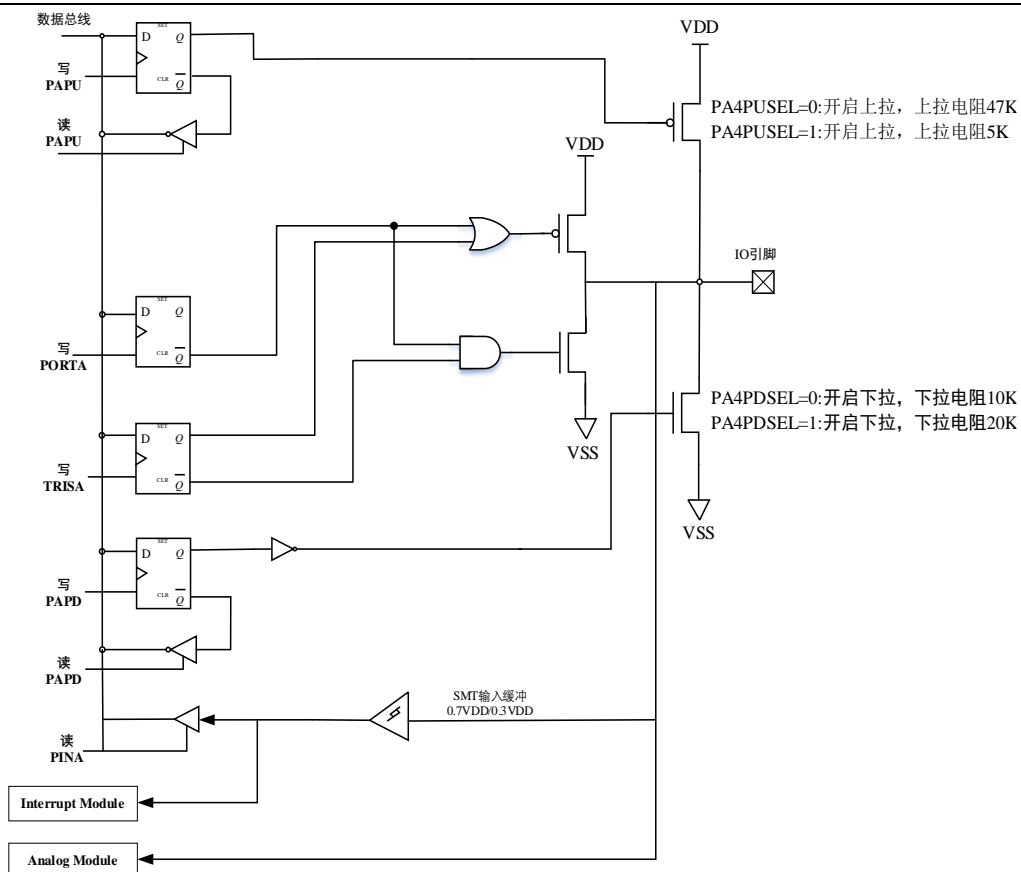


图18 PA4 结构图

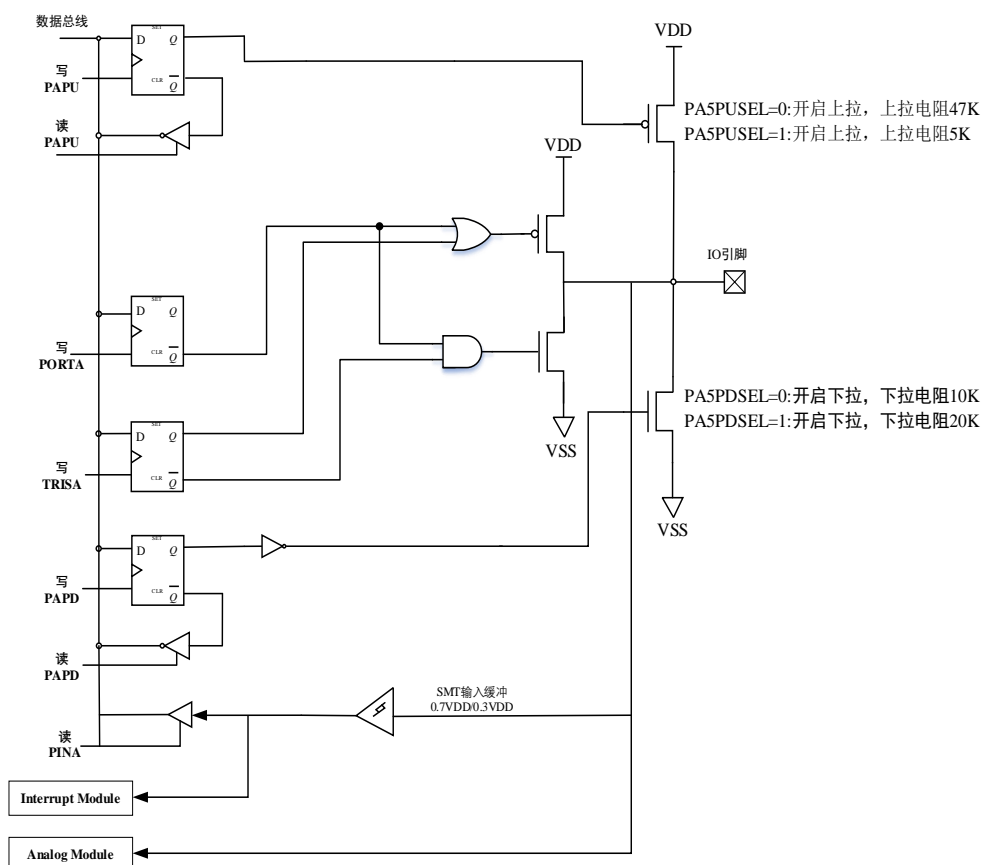


图19 PA5 口结构图

PORTA(Port 寄存器)

地址: 0XF80

| Bit | Name | Description | Attribute | Reset |
|-----|------------|---|-----------|-------|
| 7 | Reserved | | | |
| 6:0 | PORTA[6:0] | 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 | 0x00 |

TRISA(I/O 口方向控制寄存器)

地址: 0XF89

| Bit | Name | Description | Attribute | Reset |
|-----|------------|---|-----------|-------|
| 7 | Reserved | | | |
| 6:0 | TRISA[6:0] | PA 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 | 0x7F |

PINA(数据锁存寄存器)

地址: 0XF86

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

PAPD(I/O 下拉控制寄存器)

地址: 0XF8C

| Bit | Name | Description | Attribute | Reset |
|-----|------------|---------------------------------------|-----------|-------|
| 7 | Reserved | | | |
| 6:0 | PAnPD[6:0] | PA 口下拉使能位选择 1: 关闭内部下拉 0: 使能内部下拉 | R/W | 0x3F |

PAPU(I/O 上拉控制寄存器)

地址: 0XF83

| Bit | Name | Description | Attribute | Reset |
|-----|------------|-------------|-----------|-------|
| 7 | Reserved | | | |
| 6:0 | PAnPU[6:0] | PA 口上拉使能位选择 | R/W | 0x7F |

| |
|-----------|
| 1: 关闭内部上拉 |
| 0: 使能内部上拉 |

6.1.2 PB 口

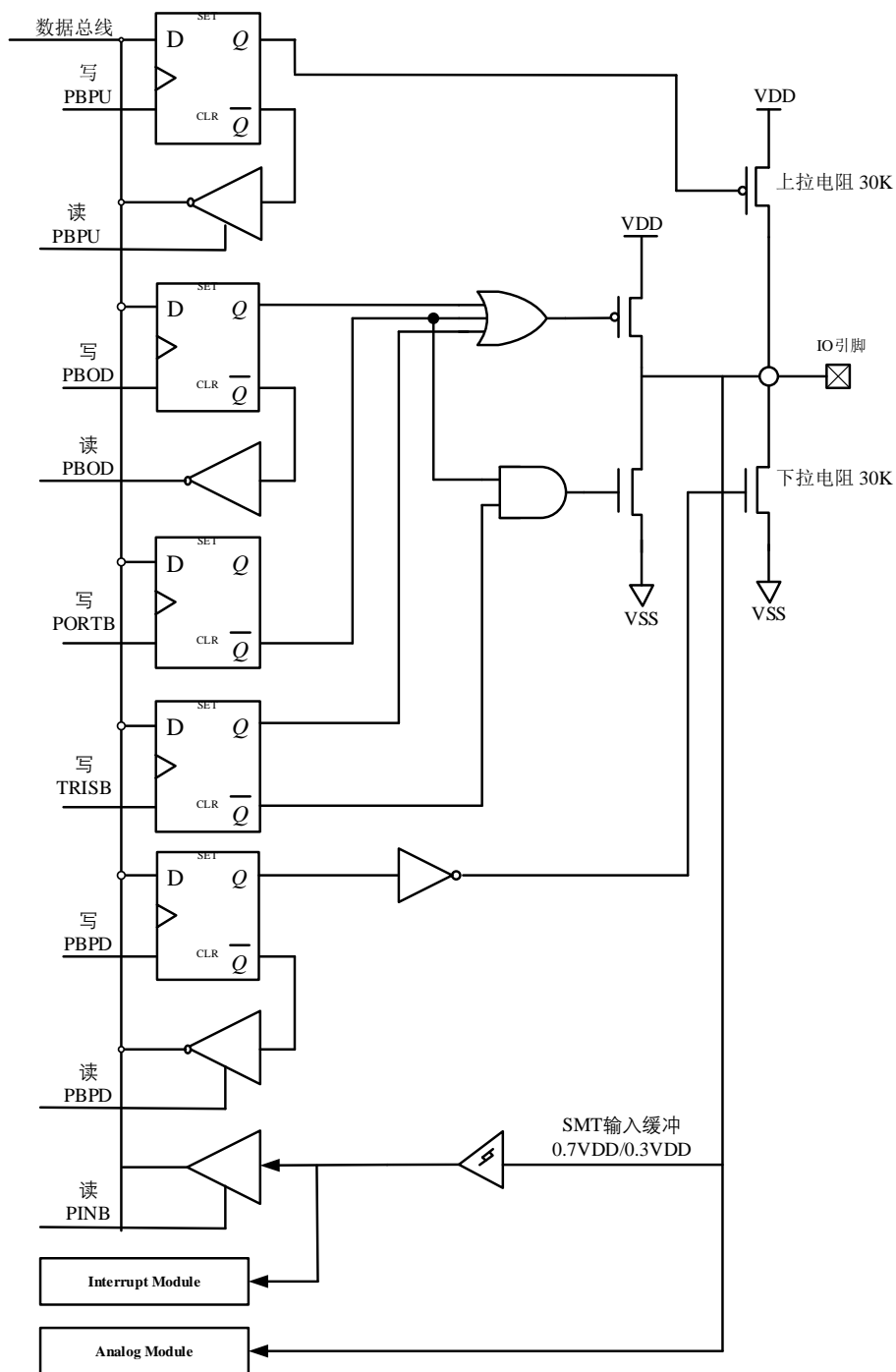


图20 PB0~PB5 口结构图

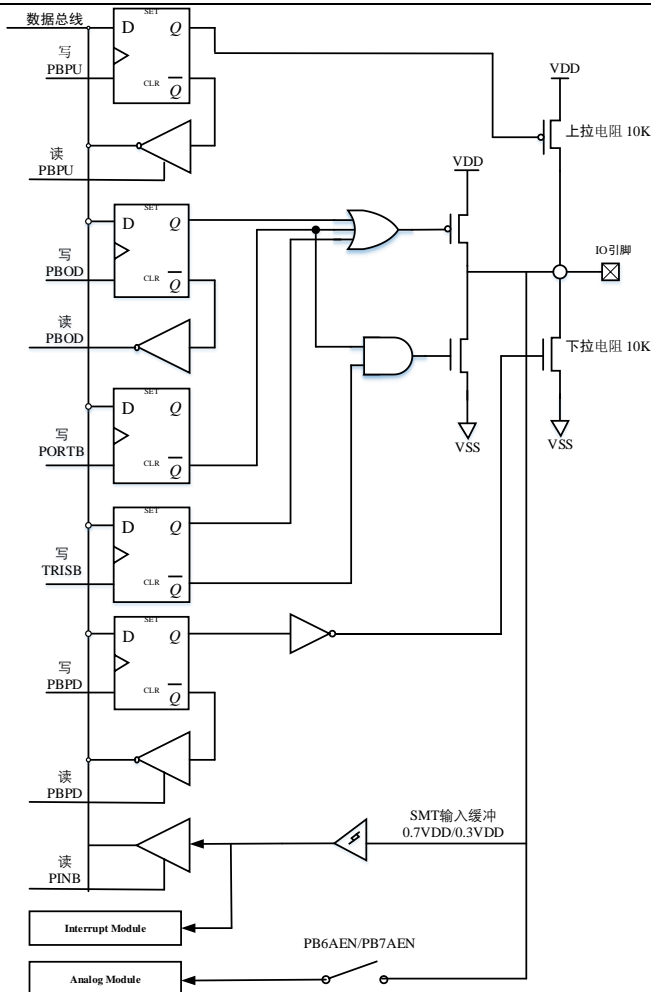


图21 PB6~PB7 口结构图

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 |

TRISB(I/O 口方向控制寄存器)

地址：0XF8A

| 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 |

PINB(数据锁存寄存器)

地址：0XF87

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

PBPU(I/O 上拉控制寄存器)

地址：0XF84

| 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 PB0~5 口的上拉电阻为 30K PB6~7 口的上拉电阻为 10K | R/W | 0xFF |

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 PB0~5 口的下拉电阻为 30K PB6~7 口的下拉电阻为 10K | R/W | 0xFF |

PBOD(I/O 开漏控制寄存器)

地址：0XF90

| Bit | Name | Description | Attribute | Reset |
|-----|------------|---------------------------------------|-----------|-------|
| 7:0 | PBnOD[7:0] | PB 口开漏使能位选择 1: 使能内部开漏 0: 关闭内部开漏 | R/W | 0x00 |

6.1.3 INT0/INT1/INT2/INT3 中断

INT0_MAP 寄存器

地址：0XFBA

| Bit | Name | Description | Attribute | Reset |
|-----|----------|---|-----------|-------|
| 7:5 | Reserved | | | |
| 4 | INT0EDGE | 外部中断 0 的边沿选择位 1: 下降沿触发中断 0: 上升沿触发中断 | R/W | 0 |

| | | | | |
|-----|----------------|--|-----|-----|
| 3 | INT0_MPORT | IO 端口的映射 INT0 中断的选择 0: PA 1: PB | R/W | 0 |
| 2:0 | INT0_MPIN[2:0] | INT0 的映射端口输出引脚选择 INT0_MPIN[2:0]=n(n=0....7),表示选择对应端口名的 n(n=0....7)脚 注: 选择 PA 口(n=0....6) | R/W | 000 |

INT1_MAP 寄存器

地址: 0XFB9

| Bit | Name | Description | Attribute | Reset |
|-----|----------------|--|-----------|-------|
| 7 | Reserved | | | |
| 6 | PA4PDSEL | PA4 下拉电阻选择位 0: PA4PD=0, PA4 口下拉电阻 10K; 1: PA4PD=0, PA4 口下拉电阻 20K; | R/W | 0 |
| 5 | PA4PUSEL | PA4 上拉电阻选择位 0: PA4PU=0, PA4 口上拉电阻 47K; 1: PA4PU=0, PA4 口上拉电阻 5K; | R/W | 0 |
| 4 | INT1EDGE | 外部中断 1 的边沿选择位 1: 下降沿触发中断 0: 上升沿触发中断 | R/W | 0 |
| 3 | INT1_MPORT | IO 端口的映射 INT1 中断的选择 0: PA 1: PB | R/W | 00 |
| 2:0 | INT1_MPIN[2:0] | INT1 的映射端口输出引脚选择 INT1_MPIN[2:0]=n(n=0....7),表示选择对应端口名的 n(n=0....7)脚 注: 选择 PA 口(n=0....6) | R/W | 000 |

INT2_MAP 寄存器

地址: 0XFB8

| Bit | Name | Description | Attribute | Reset |
|-----|----------|---|-----------|-------|
| 7 | Reserved | | | |
| 6 | PA5PDSEL | PA5 下拉电阻选择位 0: PA5PD=0, PA5 口下拉电阻 10K; 1: PA5PD=0, PA5 口下拉电阻 20K; | R/W | 0 |

| | | | | |
|-----|----------------|--|-----|-----|
| 5 | PA5PUSEL | PA5 上拉电阻选择位 0: PA5PU=0, PA5 口上拉电阻 47K; 1: PA5PU=0, PA5 口上拉电阻 5K; | R/W | 0 |
| 4 | INT2EDGE | 外部中断 2 的边沿选择位 1: 下降沿触发中断 0: 上升沿触发中断 | R/W | 0 |
| 3 | INT2_MPORT | IO 端口的映射 INT2 中断的选择 0: PA 1: PB | R/W | 0 |
| 2:0 | INT2_MPIN[2:0] | INT2 的映射端口输出引脚选择 INT2_MPIN[2:0]=n(n=0....7),表示选择对应端口名的 n(n=0....7)脚 注: 选择 PA 口(n=0....6) | R/W | 000 |

INT3_MAP 寄存器

地址: 0XFB7

| Bit | Name | Description | Attribute | Reset |
|-----|----------------|--|-----------|-------|
| 7 | Reserved | | | |
| 6 | PB7AEN | PB7 模拟通道使能位 1: 模拟通道开启 0: 模拟通道关闭 | R/W | 0 |
| 5 | PB6AEN | PB6 模拟通道使能位 1: 模拟通道开启 0: 模拟通道关闭 | R/W | 0 |
| 4 | INT3EDGE | 外部中断 3 的边沿选择位 1: 下降沿触发中断 0: 上升沿触发中断 | R/W | 0 |
| 3 | INT3_MPORT | IO 端口的映射 INT3 中断的选择 0: PA 1: PB | R/W | 0 |
| 2:0 | INT3_MPIN[2:0] | INT3 的映射端口输出引脚选择 INT3_MPIN[2:0]=n(n=0....7),表示选择对应端口名的 n(n=0....7)脚 注: 选择 PA 口(n=0....6) | R/W | 000 |

7 定时器

7.1 Timer0/Prescaler/BUZZER

TMR0 是一个 8 位定时/计数器寄存器，Timer0 的时钟源可来自于指令周期、外部实时钟(T0CKI)、内部 OSC32K 时钟源、比较器 CMP0 的输出，由 T0CON 的 T0CS0、T0CS1 共同决定。

定时模式：使用内部时钟，设置 TMR0 寄存器和 PR0 寄存器以后，定时器在两个时钟周期后开始自增，在没有预置器的情况下，每个指令周期加 1。

计数模式：通过 T0CKI(由 T0SE 位(T0CON<4>)决定上升下降触发)、OSC32K 时钟触发 Timer0 寄存器的计数，在没有预置器的情况下，外部时钟输入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。

7.1.1 Prescaler

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。

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

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

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

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

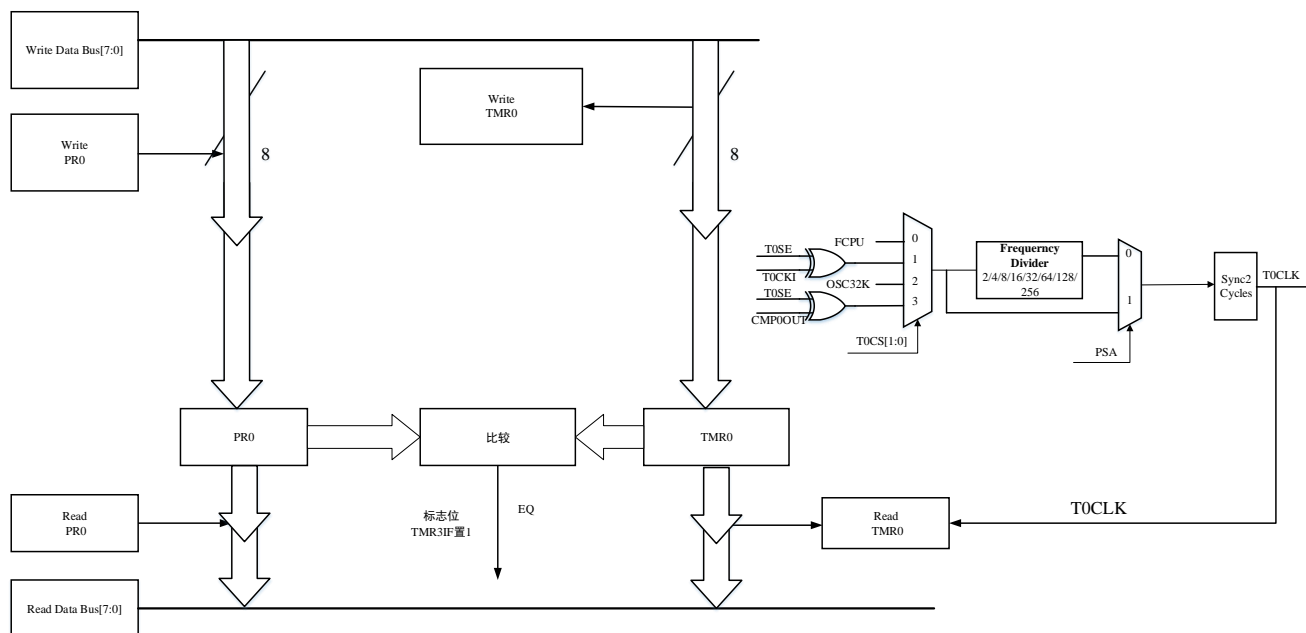


图22 Timer0 结构图

7.1.2 BUZZER

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。

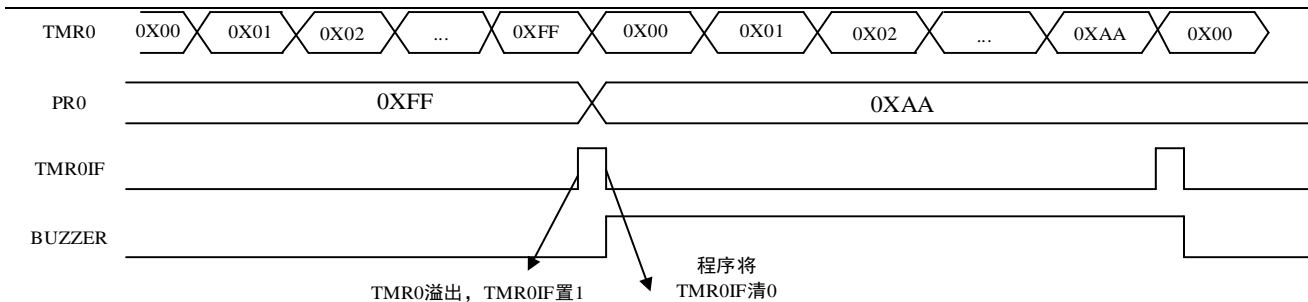


图23 BUZZEE 时序图

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/计数器寄存器)

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 | 输出 BUZZEE 0: 禁止 BUZZEE 模式 1: 启动 BUZZEE 模式(PA4) | R/W | 0 |
| 6:5 | T0CS[1:0] | Timer0 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 32K 11: TMR0 时钟源是 CMPOUT 输出 | R/W | 11 |
| 4 | T0SE | Timer0 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 | Timer0 的时钟分频使能 1: 使能 Timer0 的时钟且不分频; 0: 使能 Timer0 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 | R/W | 111 |

| | | | | | |
|--|--|-----|-------|--|--|
| | | 011 | 1:16 | | |
| | | 100 | 1:32 | | |
| | | 101 | 1:64 | | |
| | | 110 | 1:128 | | |
| | | 111 | 1:256 | | |

在 4T 和 2T 模式下，T0CKI 的时钟选择必须小于 $F_{CPU}/8$;

7.2 TIMER1 定时/计数和捕捉

7.2.1 Timer1 的定时计数

Timer1 为 16 位定时/计数器，时钟源可以是外部时钟源(TICKI)、内部 32K 时钟、CMP0 的输出等，并且支持定时/计数工作模式、捕捉模式。其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(并自动重载写入缓冲区数值)，然后每个 Timer1 时钟周期，计数递增。

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。

Timer1 支持不同模式的捕捉和 ADC 的自启动功能；

注：4T 模式下定时：

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

$Time = (PR1 + 1) * \{TICKPS1: TICKPS0\} * 4 / F_{sys}$; //TMR1 的初值为 0 时

2T 模式下定时：

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

$Time = (PR1 + 1) * \{TICKPS1: TICKPS0\} * 2 / F_{sys}$; //TMR1 的初值为 0 时

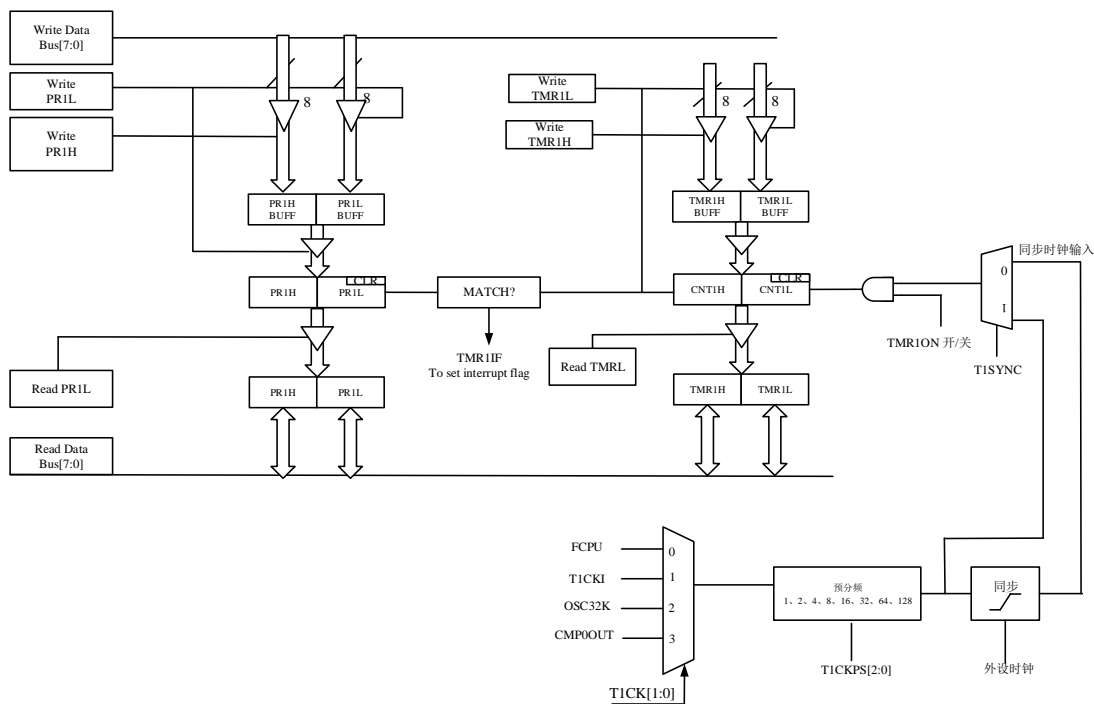


图24 Timer1 模块结构图

注：PR1HBUFF、PR1LBUFF、TMR1HBUFF、TMR1LBUFF 是 Timer1 的内部寄存器。

7.2.2 Timer1 捕捉模式

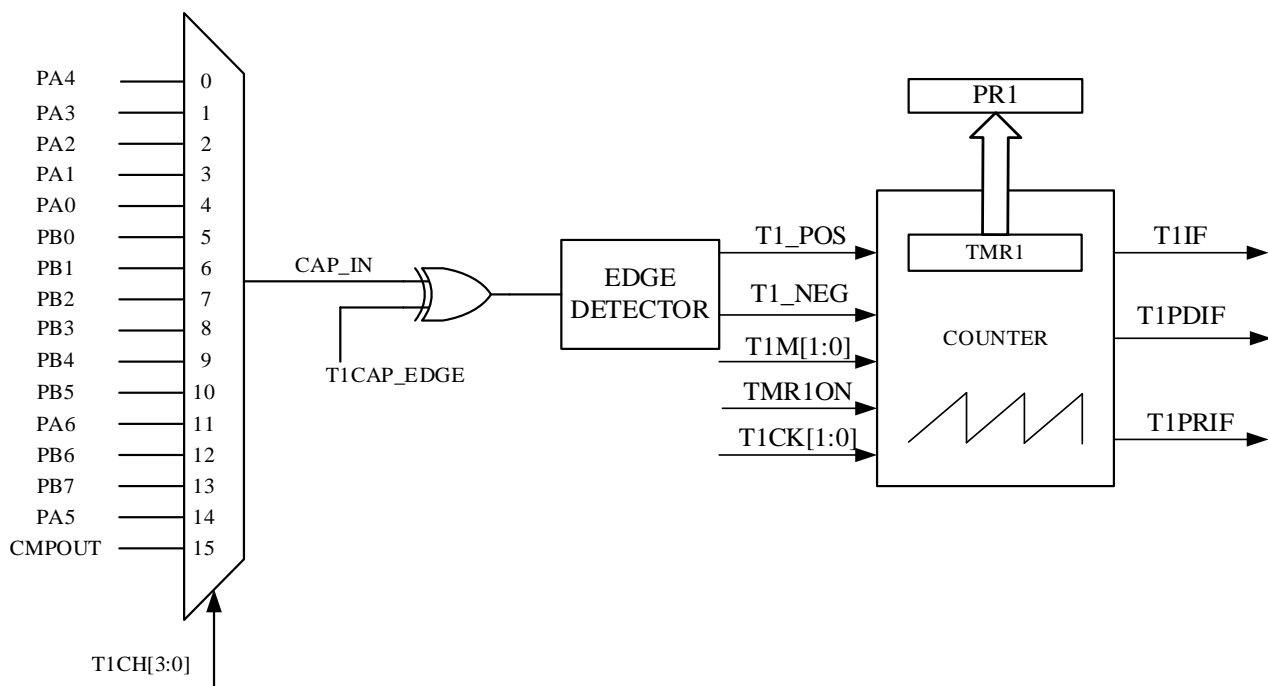


图25 捕捉模式结构图

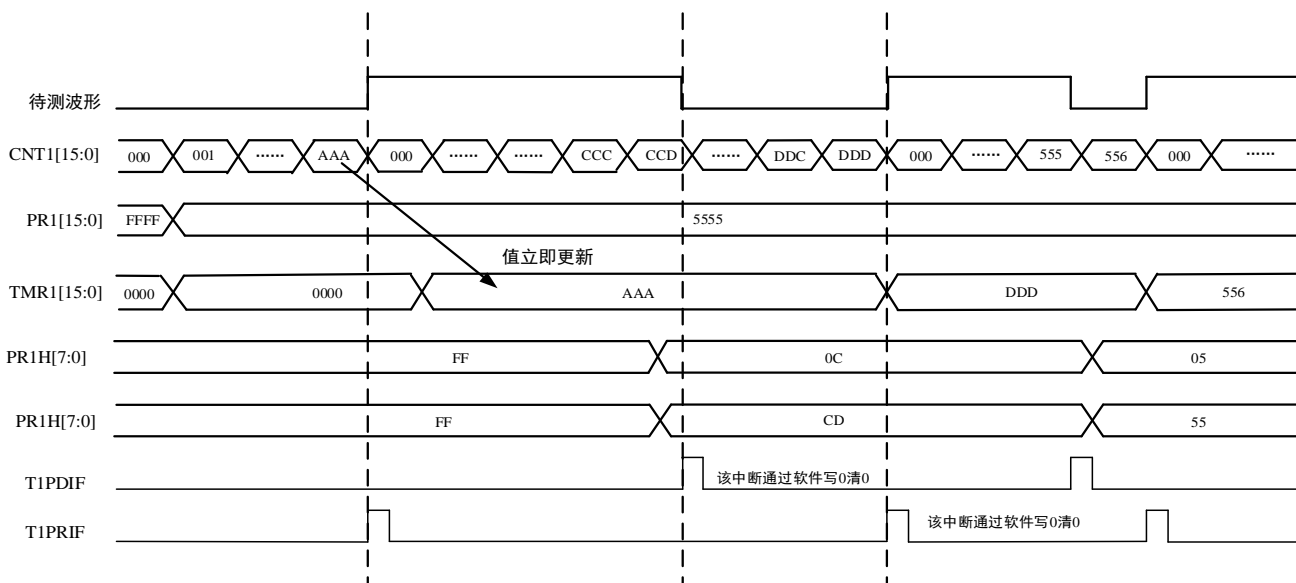


图26 上升沿捕捉模式 1 时序图

配置 T1CON1 寄存器的 T1M[1:0]=2'B01 和 T1CON1[6]=0，即 Timer1 工作模式选择上升沿捕捉模式 1，输入第一个上升沿时，CNT1 寄存器清零，下降沿时将捕捉的值读入 PR1 寄存器，并产生 T1PDIF 中断标志，输入第二个上升沿时将捕捉的值立即读入 TMR1 寄存器中，并产生 T1PRIF；

配置 T1CON1 寄存器的 T1M[1:0]=2'B01 和 T1CON1[6]=1，即 Timer1 工作模式选择下降沿捕捉模式 1，输入第一个下降沿时，CNT1 寄存器清零，上升沿时将捕捉的值读入 PR1 寄存器，并产生 T1PDIF 中断标志，输入第二个下降沿时将捕捉的值立即读入 TMR1 寄存器中，并产生 T1PRIF；

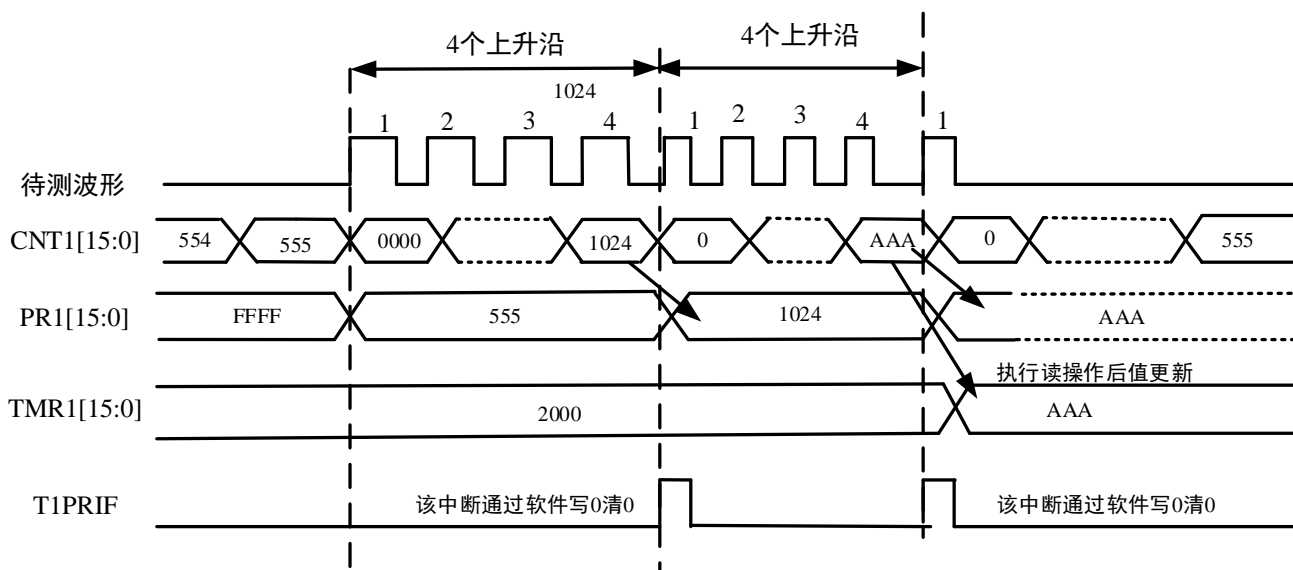


图27 上升沿捕捉模式 2 时序图

配置 T1CON1 寄存器的 T1M[1:0]=2'B10 和 T1CON1[6]=0, 即 Timer1 工作模式选择上升沿捕捉模式 2, 当检测到输入的第四个上升沿时, 将捕捉的值读取到 PR1 寄存器, 并产生 T1PRIF。

当 Timer1 尚未检测到输入的第四个上升沿时, 计数值 TMR1 达到 0xFFFF, 发生上溢事件, 中断标记 T1IF 置 1(T1IF 需要手动清零), TMR1 寄存器清零, TMR1 寄存器继续计数。

配置 T1CON1 寄存器的 T1M[1:0]=2'B10 和 T1CON1[6]=1, 即 Timer1 工作模式选择下降沿捕捉模式 2, 当检测到输入的第四个下降沿时, 将捕捉的值读取到 PR1 寄存器, 并产生 T1PRIF。

当 Timer1 尚未检测到输入的第四个下降沿时, 计数值 TMR1 达到 0xFFFF, 发生上溢事件, 中断标记 T1IF 置 1(T1IF 需要手动清零), TMR1 寄存器清零, TMR1 寄存器继续计数。

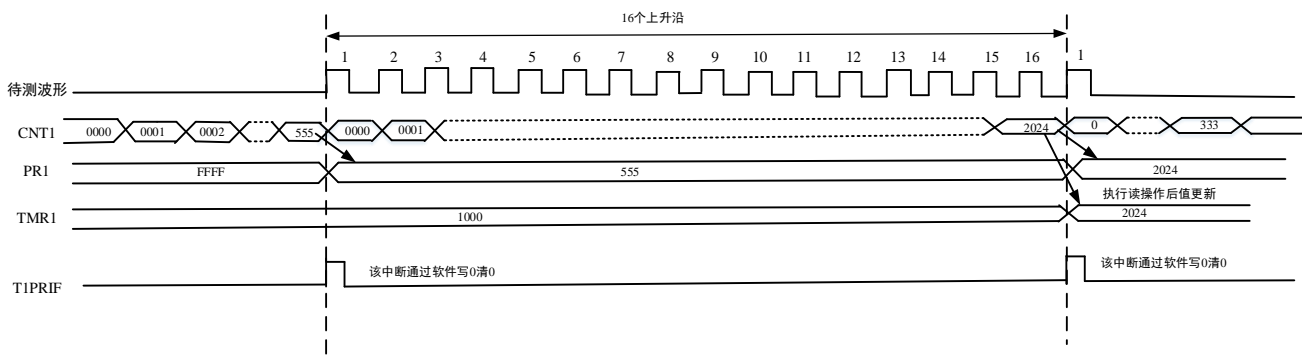


图28 上升沿捕捉模式 3 时序图

配置 T1CON1 寄存器的 T1M[1:0]=2'B11 和 T1CON1[6]=0, 即 Timer1 工作模式选择上升沿捕捉模式 3, 当检测到输入的第十六个上升沿时, 将捕捉的值读取到 PR1 寄存器, 并产生 T1PRIF。

当 Timer1 尚未检测到输入的第十六个上升沿时, 计数值 TMR1 达到 0xFFFF, 发生上溢事件, 中断标记 T1IF 置 1(T1IF 需要手动清零), TMR1 寄存器清零, TMR1 寄存器继续计数。

配置 T1CON1 寄存器的 TIM[1:0]=2'B11 和 T1CON1[6]=1,即 Timer1 工作模式选择下降沿捕捉模式 3,当检测到输入的第十六个下降沿时,将捕捉的值读取到 PR1 寄存器,并产生 T1PRIF。

当 Timer1 尚未检测到输入的第十六个下降沿时,计数值 TMR1 达到 0xFFFF,发生上溢事件,中断标记 T1IF 置 1(T1IF 需要手动清零),TMR1 寄存器清零,TMR1 寄存器继续计数。

PR1L(TIMER1 周期寄存器)

地址: 0XFCE

| Bit | Name | Description | Attribute | Reset |
|-----|-----------|------------------------|-----------|-------|
| 7:0 | PR1L[7:0] | 16位捕获/比较/定时/计数周期寄存器低8位 | R/W | 0XFF |

PR1H(TIMER1 周期寄存器)

地址: 0XFCF

| Bit | Name | Description | Attribute | Reset |
|-----|-----------|------------------------|-----------|-------|
| 7:0 | PR1H[7:0] | 16位捕获/比较/定时/计数周期寄存器高8位 | R/W | 0XFF |

TMR1L(Timer1 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 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: 0XFCD

| Bit | Name | Description | Attribute | Reset |
|-----|-------------|--|-----------|-------|
| 7:6 | T1CK[1:0] | Timer1 的时钟选择 00: FCPU 作为计数时钟 01: T1CKI 作为时钟(PB3) 10: 内部 32K 作为时钟 11: CMP0OUT 作为时钟 | R/W | 0 |
| 5 | T1ADCEN | 定时器 1 自启动 ADC 使能 1: 使能自启动 0: 禁止自启动 | R/W | 0 |
| 4:2 | T1CKPS[2:0] | Timer1 的输入时钟分频 000: Timer1 输入时钟 1:1 分频 001: Timer1 输入时钟 1:2 分频 010: Timer1 输入时钟 1:4 分频 011: Timer1 输入时钟 1:8 分频 | R/W | 000 |

| | | | | |
|---|--------|---|-----|---|
| | | 100: Timer1 输入时钟 1:16 分频 101: Timer1 输入时钟 1:32 分频 110: Timer1 输入时钟 1:64 分频 111: Timer1 输入时钟 1:128 分频 | | |
| 1 | T1SYNC | Timer 外部时钟输入同步选择位 1: 不同步外部时钟输入 0: 同步外部时钟输入 | R/W | 0 |
| 0 | TMR1ON | Timer1 的使能位 1: 使能 Timer1 定时计数器 0: 关闭 Timer1 定时计数器 | R/W | 0 |

- 1、在 4T 和 2T 模式下，T1CKI 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: 0XFCC

| Bit | Name | Description | Attribute | Reset |
|-----|------------|--|-----------|-------|
| 7 | Reserved | | | |
| 6 | T1CAP_EDGE | Timer1 的起始捕捉沿的选择 1: 选择下降沿 0: 选择上升沿 | R/W | 0 |
| 5:2 | T1CH[3:0] | Timer1 脉宽信号输入选择 0000: T1CH0 作为脉宽检测输入信号(PA4) 0001: T1CH1 作为脉宽检测输入信号(PA3) 0010: T1CH2 作为脉宽检测输入信号(PA2) 0011: T1CH3 作为脉宽检测输入信号(PA1) 0100: T1CH4 作为脉宽检测输入信号(PA0) 0101: T1CH5 作为脉宽检测输入信号(PB0) 0110: T1CH6 作为脉宽检测输入信号(PB1) 0111: T1CH7 作为脉宽检测输入信号(PB2) 1000: T1CH8 作为脉宽检测输入信号(PB3) 1001: T1CH9 作为脉宽检测输入信号(PB4) 1010: T1CH10 作为脉宽检测输入信号(PB5) 1011: T1CH11 作为脉宽检测输入信号(OSC32K) 1100: T1CH12 作为脉宽检测输入信号(PB6) 1101: T1CH13 作为脉宽检测输入信号(PB7) | R/W | 0000 |

| | | | | |
|-----|----------|--|-----|----|
| | | 1110: T1CH14 作为脉宽检测输入信号(PA5) 1111: T1CH15 作为脉宽检测输入信号(CMPOUT) | | |
| 1:0 | T1M[1:0] | <p>Timer1 工作模式</p> <p>00: Timer1 工作在普通模式, 溢出时间产生中断标志位。</p> <p>01: 捕捉模式 1: 当 T1CAP_EDGE=0 时输入第一个上升沿时, CNT1 寄存器清零, 下降沿时将捕捉的值读取到 PR1 寄存器, 并产生 T1PDIF 中断标志, 输入第二个上升沿时将捕捉的值读取到 TMR1 寄存器中, 并产生 T1PRIF;</p> <p>当 T1CAP_EDGE=1 时输入第一个下降沿时, CNT1 寄存器清零, 上升沿时将捕捉的值读取到 PR1 寄存器, 并产生 T1PDIF 中断标志, 输入第二个下降沿时将捕捉的值读取到 TMR1 寄存器中, 并产生 T1PRIF;</p> <p>10: 捕捉模式 2: 当 T1CAP_EDGE=0 时测量 4 次上升沿的时间, 当检测到输入的第四个上升沿时, 将捕捉的值读取到 PR1 寄存器, 并产生 T1PRIF。</p> <p>当 T1CAP_EDGE=1 时测量 4 次下降沿的时间, 当检测到输入的第四个下降沿时, 将捕捉的值读取到 PR1 寄存器, 并产生 T1PRIF。</p> <p>11: 捕捉模式 3: 当 T1CAP_EDGE=0 测量 16 次上升沿的时间, 当检测到输入的第十六个上升沿时, 将捕捉的值读取到 PR1 寄存器, 并产生 T1PRIF。</p> <p>当 T1CAP_EDGE=1 测量 16 次下降沿的时间, 当检测到输入的第十六个下降沿时, 将捕捉的值读取到 PR1 寄存器, 并产生 T1PRIF。</p> | R/W | 00 |

7.3 TIMER2 定时器和 3 路独立互补 PWM

Timer2 是 10 位定时计数器，支持定时、沿启动 ADC 采集和带死区的 3 路独立互补 PWM 输出。同时支持 CPU 中断，时钟源为 Fsys，通过预分频器和分频器可调整 Timer2 时钟。

7.3.1 Timer2 计数/定时

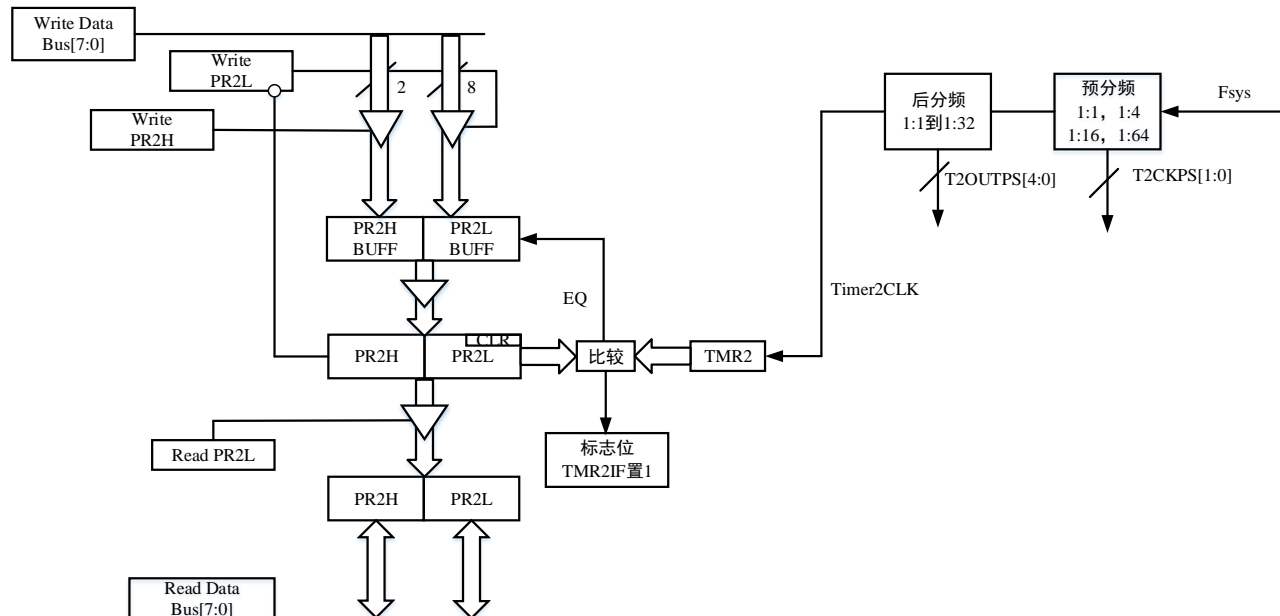


图29 定时模式原理框图

Timer2 定时操作步骤：

- Timer2 时钟源为 Fsys，通过配置预分频器 T2CKPS[1:0]，可对 Fsys 进行 1/4/16/64 分频，再配置分频器 T2OUTPS[4:0]，对预分频的时钟再分频(1 分频-32 分频)后，作为 Timer2 计数时钟
- 通过配置 TMR2 和 PR2，控制定时时间
- 使用中断，需将全局中断 GIE(INTCON<7>)、外设中断 PEIE(INTCON<6>)和使能位 TMR2IE 置 1，同时可配置 TMR2IP 设置中断优先级
- 配置完相应的寄存器后，TMR2ON 置 1，使能计数器开始计数

Timer2 的定时时间计算公式如下：

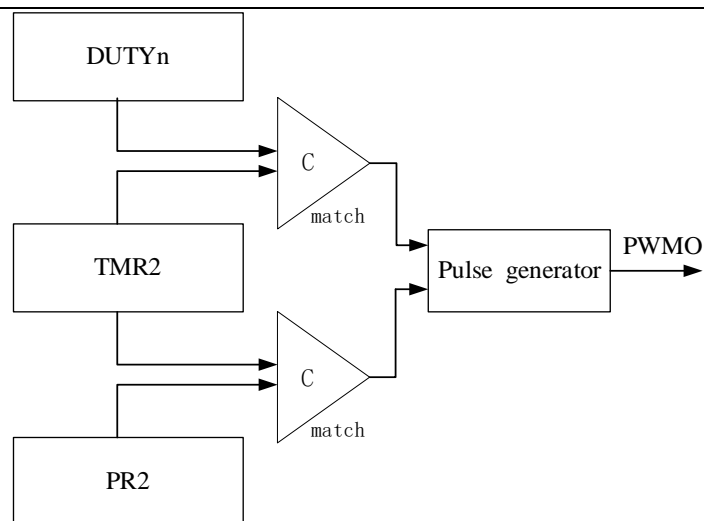
$$\text{Time2} = (\text{PR2} - \text{TMR2} + 1) * \{ \text{T2CKPS}[1:0] \} * \{ \text{T2OUTPS}[4:0] \} / \text{Fsys}$$

注：1、T2CKPS[1:0]对应 1/4/16/64 T2OUTPS[4:0]对应 1-32

2、TMR2 一般使用默认值 0，若想通过给 TMR2 赋初值来控制定时时间，须在中断中给 TMR2 赋值。

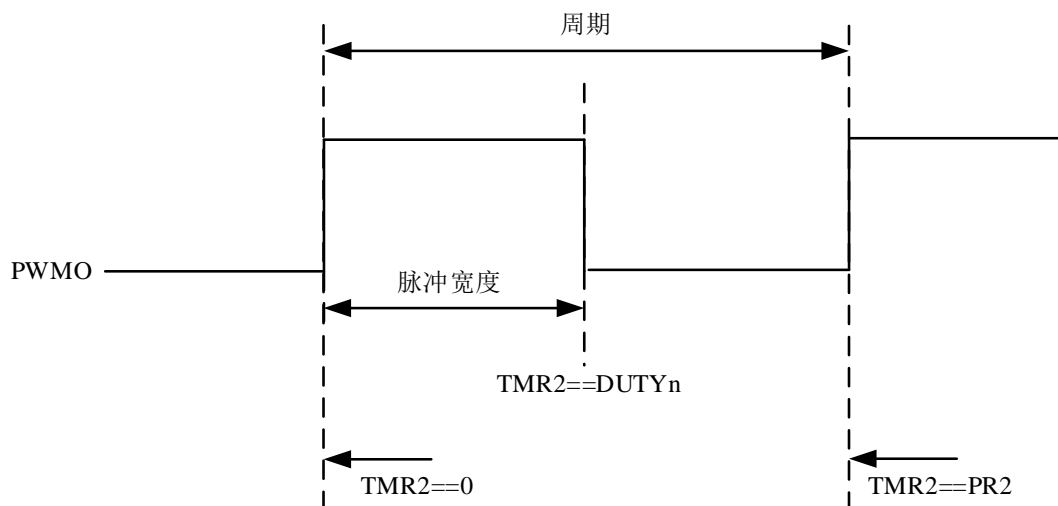
7.3.2 PWM

Timer2 模块通过配置 TMR2、PR2 和 DUTY_n(DUTY20, DUTY22, DUTY24)寄存器，可以生成占空比为 10 位分辨率的 PWM。该 PWM 无法直接映射到 IO，是作为 HBRIDGE 单元的输入，可生成 3 路独立互补 PWM。



注：n=20/22/24。

图30 PWM 原理框图



注：n=20/22/24。

图31 PWM 输出原理图

7.3.2.1 PWM 的周期

PWM 周期由 Timer2 的 PR2 寄存器指定。PWM 周期公式

PWM 周期为 $(PR2+1) \times (TMR2 \text{ 预分频值}) \times (TMR2 \text{ 分频值}) / F_{sys}$ (系统工作时钟)

当 TMR2 等于 PR2 时，下一次递增周期将发生以下事件：

1. TMR2 清零；
2. PWM 输出引脚置 1 (例外：若 PWM 占空比=0%，引脚不置 1)。

7.3.2.2 占空比

通过给 DUTYn 寄存器写入 10 位值可指定 PWM 占空比。

PWM 脉冲宽度 = $(DUTY_n+1) \times (TMR2 \text{ 预分频值}) \times (TMR2 \text{ 分频值}) / F_{sys}$ (系统工作时钟)

$$PWM \text{ 占空比} = \frac{DUTY_n + 1}{PR2}$$

注：n=20/22/24。

7.3.2.3 分辨率

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

$$\text{分辨率} = \text{Log}_2\{PR2+1\} \text{ 位}$$

7.3.2.4 PWM 使能 ADC 采集

下图为 PWM20 跳变时，ADC 采集时序图：

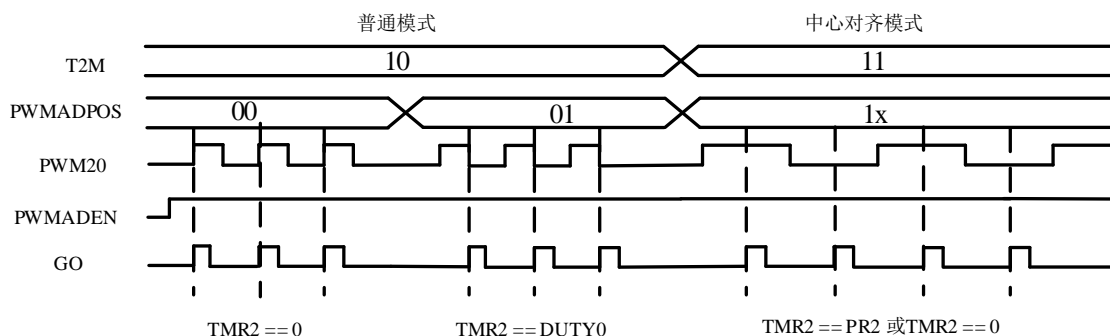


图32 ADC 采集时序图

下图为 PWM 普通模式，ADC 采集图：

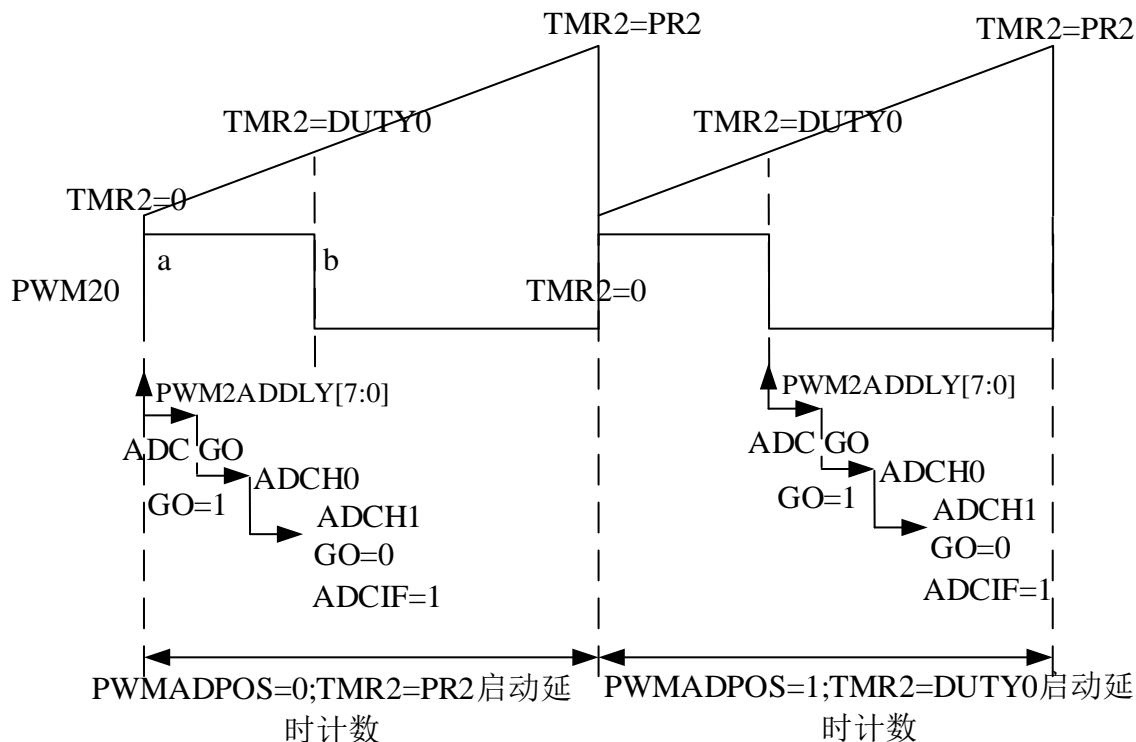
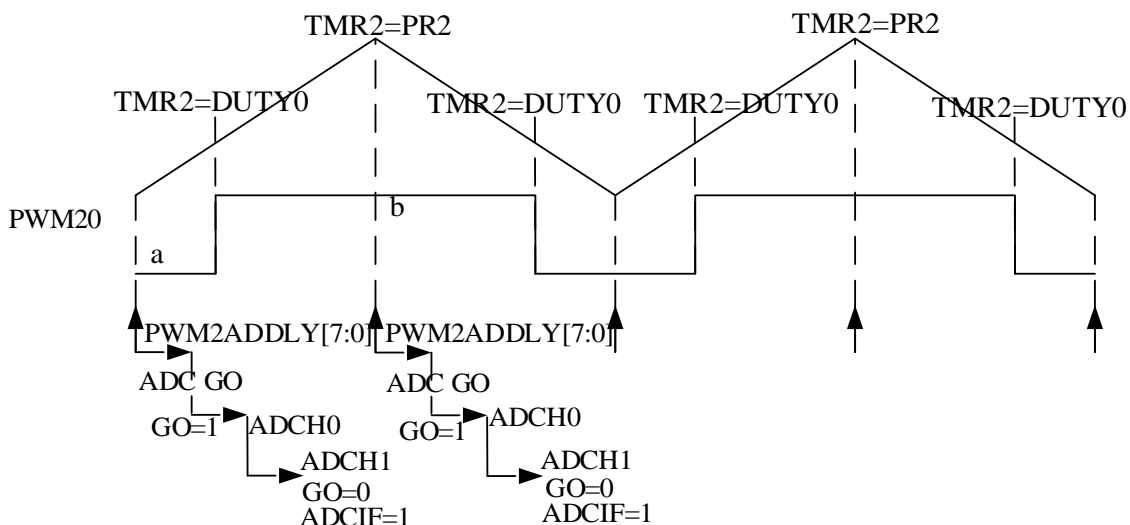


图33 PWM 普通模式下，ADC 采集框图

下图为PWM中心对齐模式，ADC采集图：

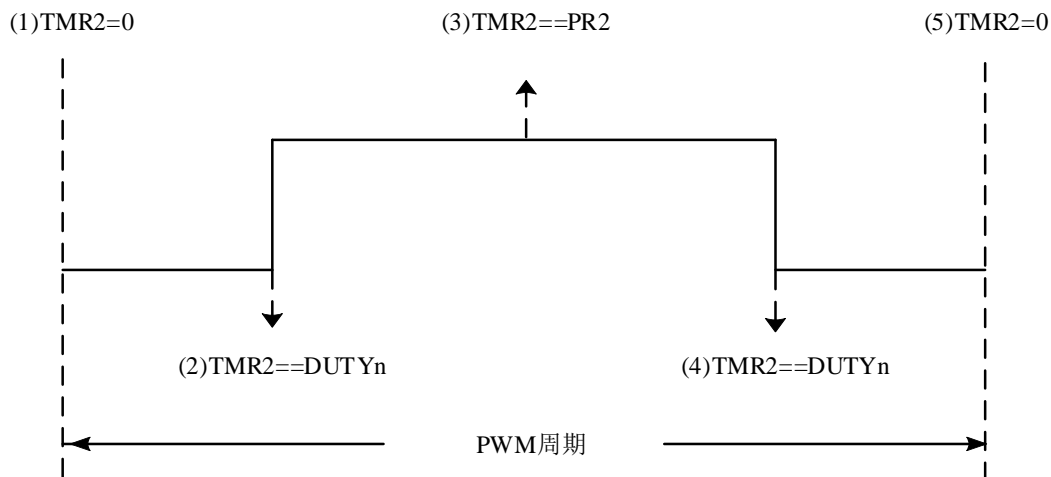


PWMADPOS=1x, TMR2=8'H0启动延时计数 TMR2=PR2启动延时计数

图34 PWM 中心模式下，ADC 采集框图

注：当 ADC 采集 PWM20 的 a 点时，ADC 采集延时时间不能超过 PWM20 占空比时间；当 ADC 采集 PWM20 的 b 点时，PWM 普通模式中 ADC 采集延时时间不能超过 PWM20 下个周期开始时间，PWM 中心对齐模式中 ADC 采集延时时间不能超过 TMR2 递减到和 DUTY0 相等的时间。

7.3.2.5 PWM 中心对齐模式



注：n=20/22/24。

图35 PWM 中心对齐示意图

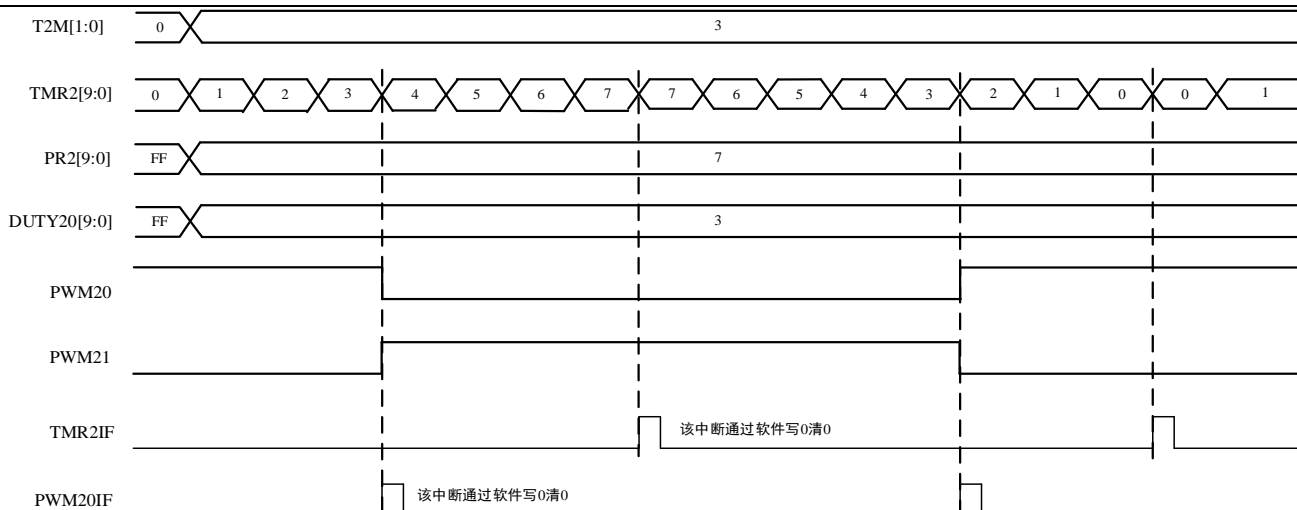


图36 中心对齐模式下，中断产生时序图(以 PWM20、PWM21 为例)

配置 T2M=11，PWM 使能中心对齐模式，此时 PWM 工作过程如下：

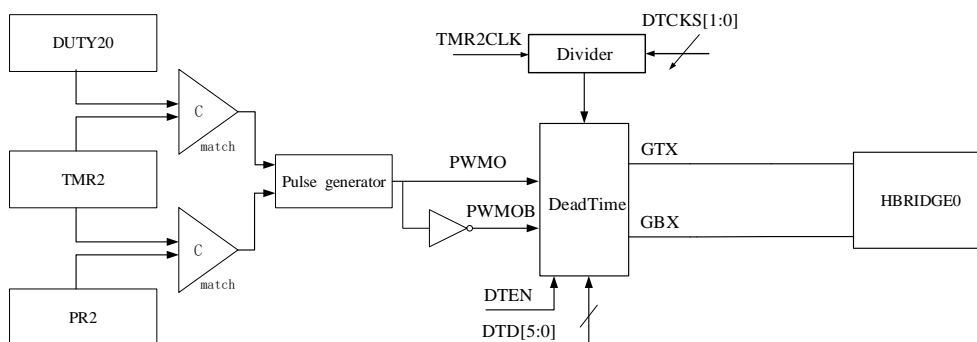
- (1) PWM周期开始，TMR2开始从0递增；
- (2) 当TMR2与DUTY_n相等时，PWM开始第一次高低电平变化，TMR2继续递增；
- (3) 当TMR2与PR2相等时，TMR2开始自减；
- (4) 当TMR2再次与DUTY_n相等时，PWM再次变化高低电平；
- (5) 当TMR2自减为0时，此时PWM周期结束，开始下一个PWM周期。

此时，实际上 PWM 周期为 $2 * (PR2 + 1) * (TMR2 \text{ 预分频值}) * (TMR2 \text{ 预分频值}) / F_{sys}$ (系统工作时钟)

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；n=20/22/24。

7.3.3 HBRIDGE

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 死区时间设置以及输出极性控制等。通过将 Timer2 的 PWM 同时输出到 3 个 HBRIDGE 单元，可控制 3 组独立互补 PWM(PWM20 和 PWM21、PWM22 和 PWM23、PWM24 和 PWM25)输出。PWM20 和 PWM21、WM22 和 PWM23、PWM24 和 PWM25 原理图如下图。



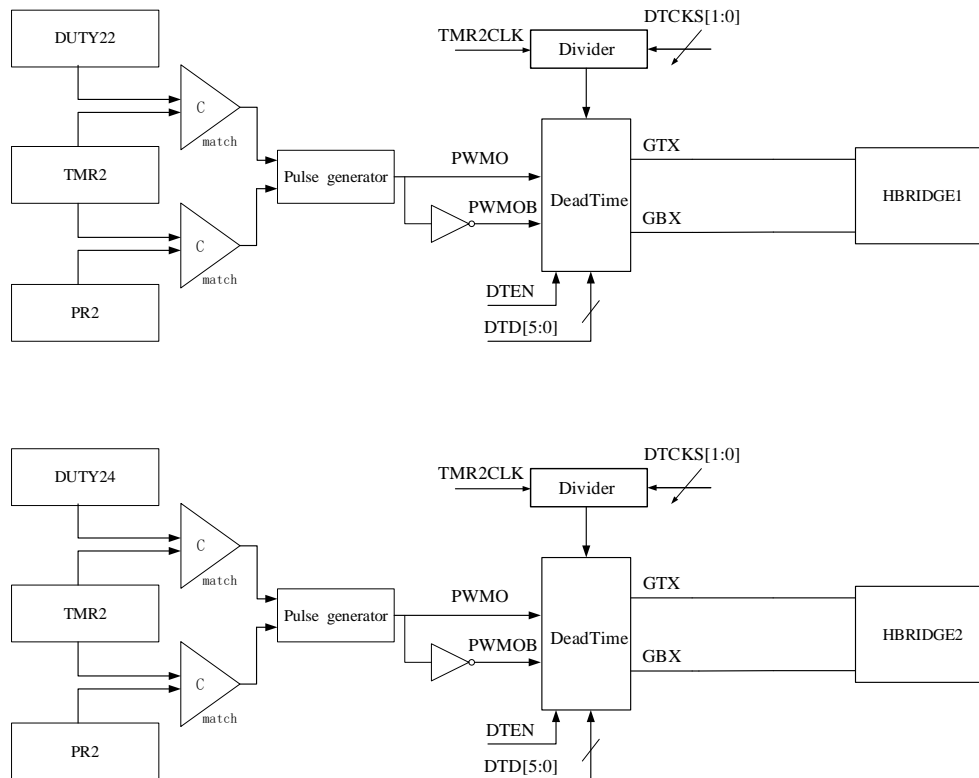
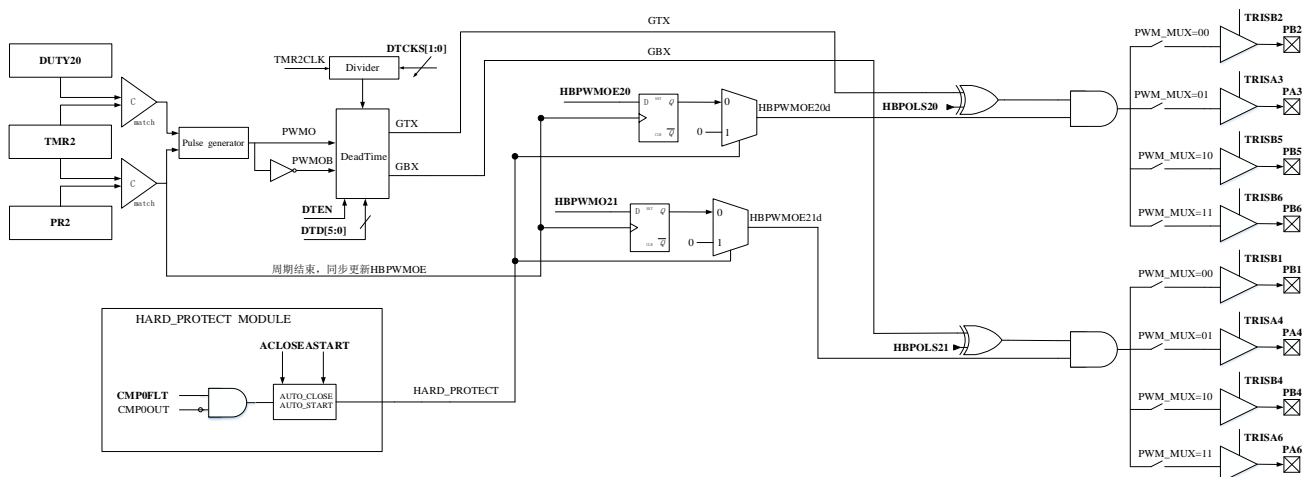
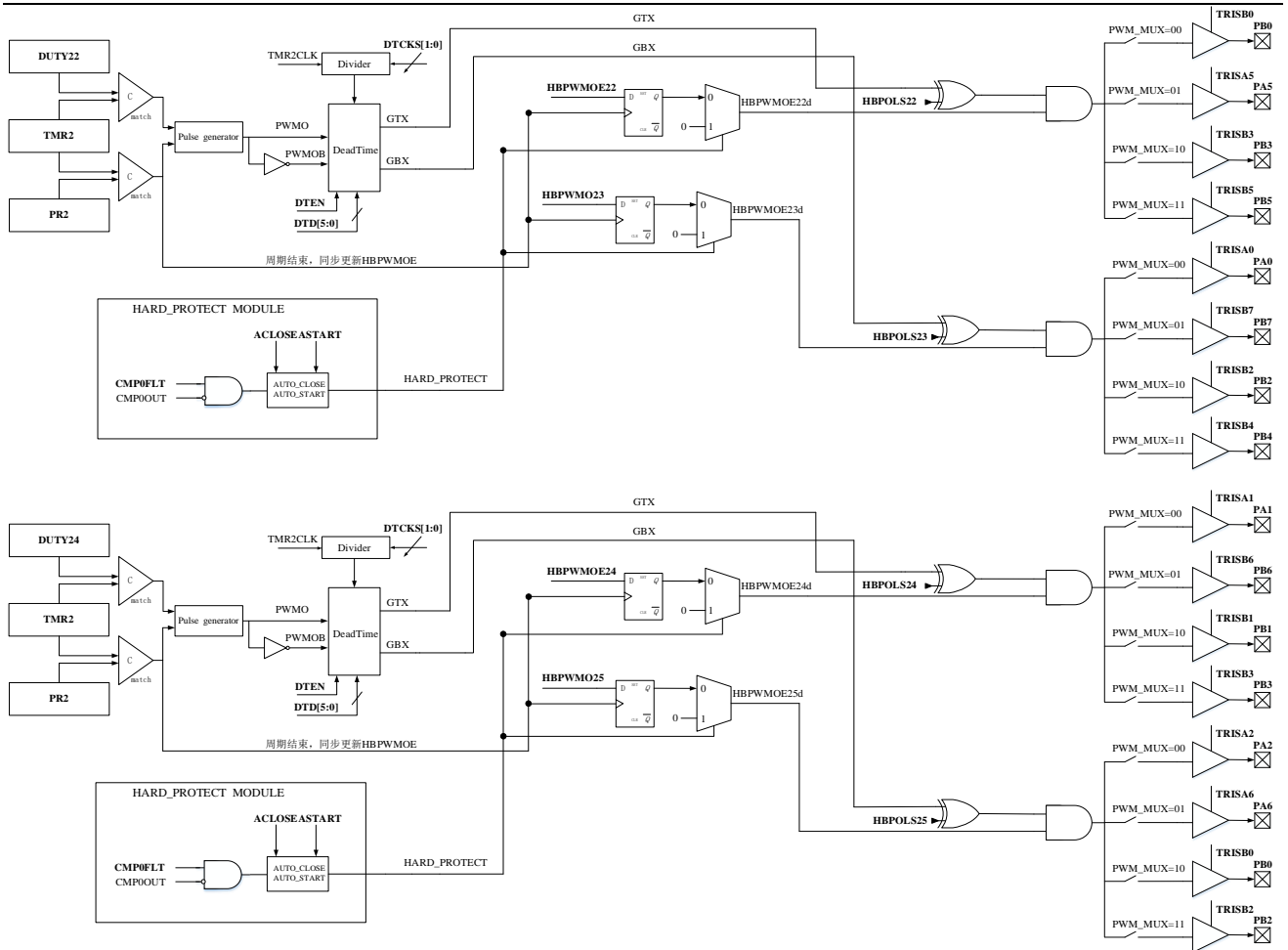


图37 带有死区的互补 PWM 驱动 3 路 HBRIDGE





注：HBPWMOEnd 为内部信号；n=20/21/22/23/24/25。

图38 HBRIDGE 输出原理图

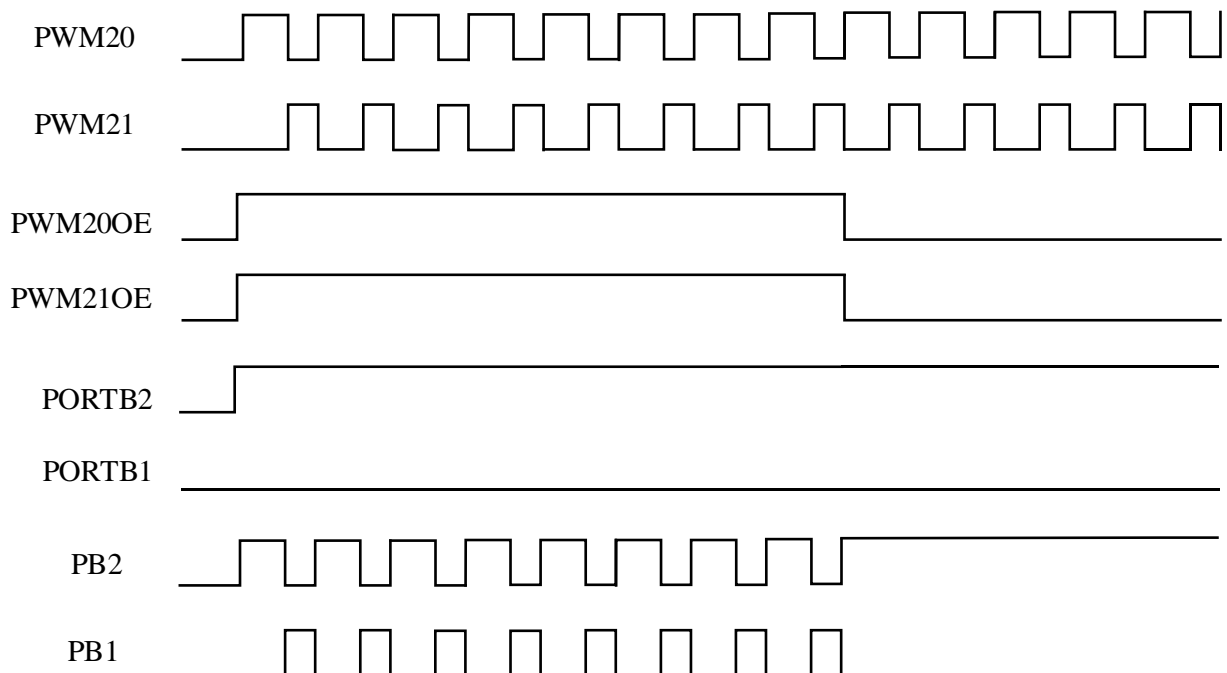


图39 PWM 映射 IO 时序图

7.3.3.1 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 PME 寄存器的 DTEN 位控制。死区时间要控制在 $0\mu\text{s}\sim 100\mu\text{s}$ 左右，可通过 DTCKS1~DTCKS0 位选择死区时钟源，并通过 DTD5~DTD0 位对插入的死区时间进行调整。

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

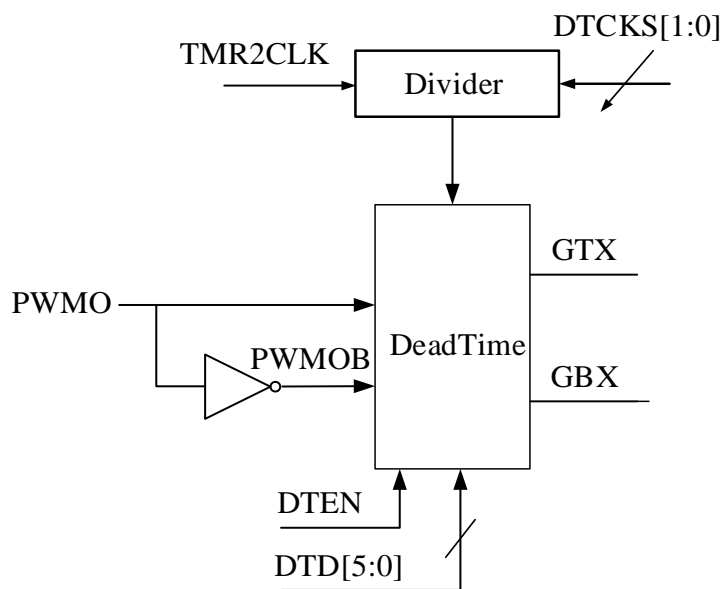


图40 死区时间方框图

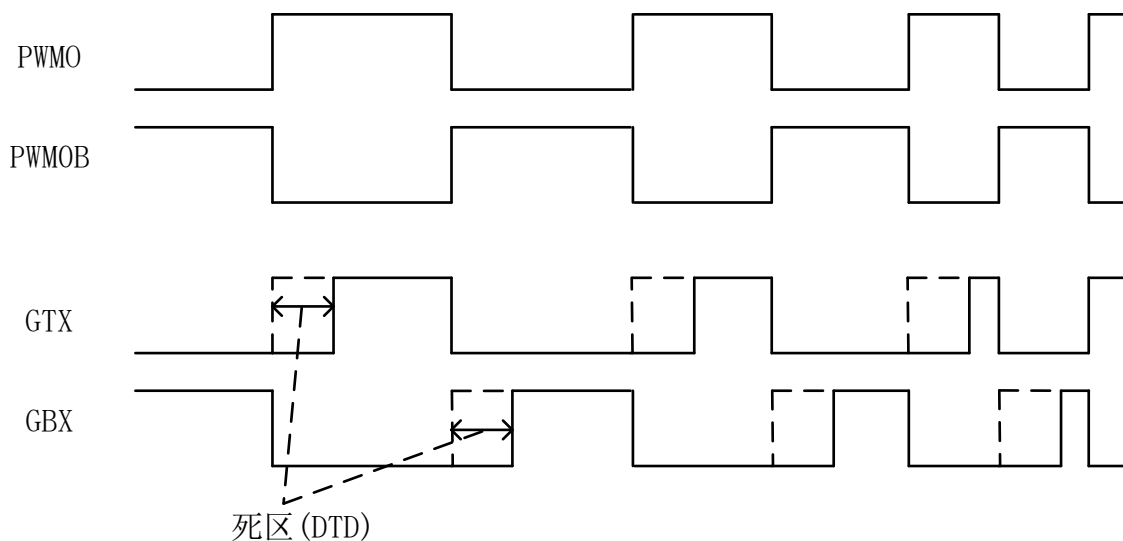


图41 死区时间时序图

表 7-1 死区时间电路输出表

| DTEN | GTX | GBX |
|------|----------|-----------|
| 1 | PWMO&DTD | PWMOB&DTD |
| 0 | PWMO | PWMOB |

7.3.3.2 故障刹车

6 路 HBRIDGE 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 映射引脚将输出 PORT 口状态。

故障刹车事件可为 CMP0FLT 或 CMP1FLT；

故障刹车时，HBPWM20/HBPWM21/HBPWM22/HBPWM23/HBPWM24/HBPWM25 输出关闭。此时输出逻辑由 PORT 口控制，输出逻辑高或逻辑低。

ACLOSE 自动关闭-当故障事件有效时，PWM 输出关闭。

ASTART 自启动-当故障事件结束后，PWM 将在更新周期时，使能 PWM 输出。

下图以 HBPWM20 和 HBPWM21 举例说明故障刹车图。HBPWM22 和 HBPWM23、HBPWM24 和 HBPWM25 同理。

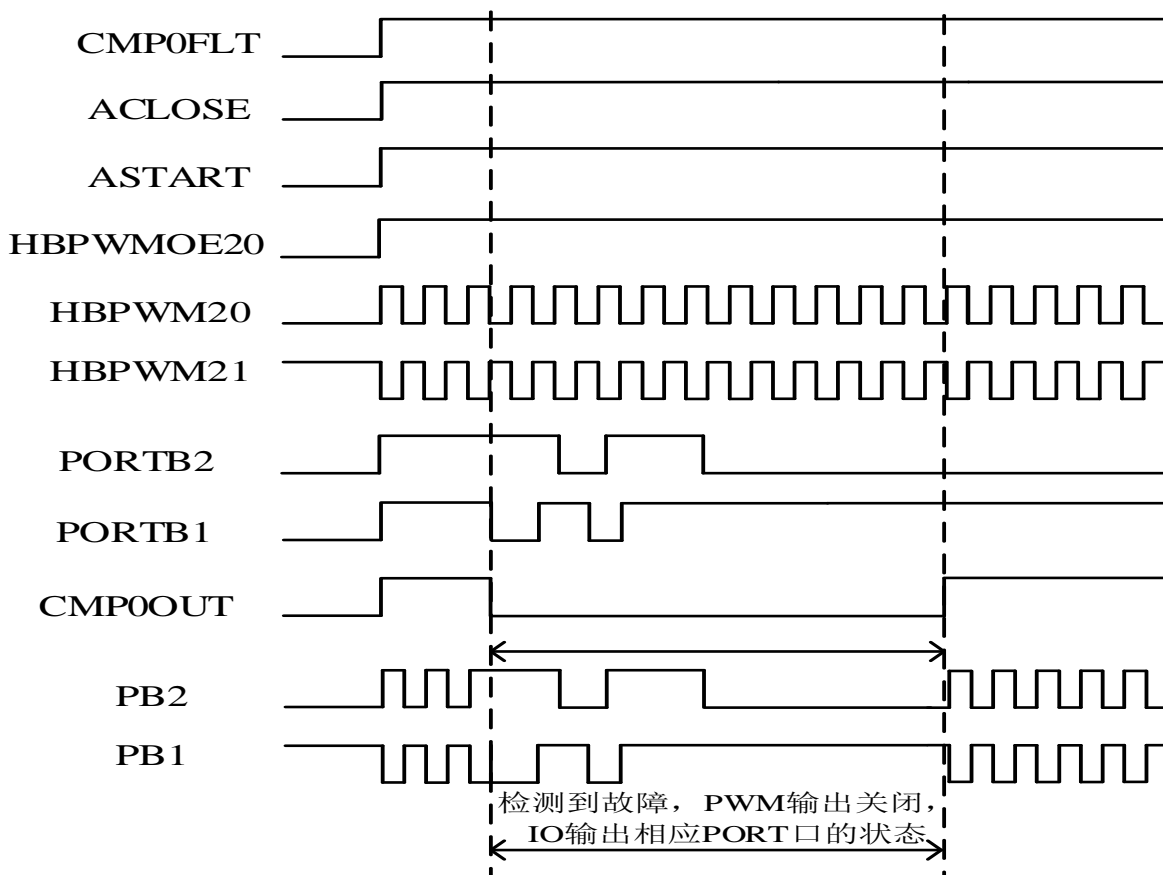


图42 故障刹车图

7.3.4 寄存器

PR2H(timer2 的周期寄存器高 2 位)

地址: 0XFC9

| Bit | Name | Description | Attribute | Reset |
|-----|-----------|-----------------|-----------|-------|
| 7:2 | reserved | | | |
| 1:0 | PR2H[1:0] | TMR2 周期寄存器高 2 位 | R/W | 11 |

PR2L(timer2 的周期寄存器)

地址: 0XFC8

| Bit | Name | Description | Attribute | Reset |
|-----|-----------|-----------------|-----------|-------|
| 7:0 | PR2L[7:0] | TMR2 周期寄存器低 8 位 | R/W | 0XFF |

TMR2H(Timer2 的高 2 位寄存器)

地址:0XFCB

| Bit | Name | Description | Attribute | Reset |
|-----|------------|-----------------|-----------|-------|
| 7:2 | reserved | | | |
| 1:0 | TMR2H[1:0] | 10 位定时/计数器高 2 位 | R/W | 00 |

TMR2L(Timer2 的低八位寄存器)

地址:0XFCA

| Bit | Name | Description | Attribute | Reset |
|-----|-----------|-----------------|-----------|-------|
| 7:0 | TMR2[7:0] | 10 位定时/计数器低 8 位 | R/W | 0X00 |

DUTY20H(PWM20, PWM21 的占空比寄存器高 2 位)

地址:0XFC7

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|-----------------|-----------|-------|
| 7:2 | reserved | | | |
| 1:0 | DUTY20H[1:0] | 10 位占空比寄存器高 2 位 | R/W | 11 |

DUTY20L(PWM20, PWM21 的占空比寄存器低 8 位)

地址:0XFC6

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|-----------------|-----------|-------|
| 7:0 | DUTY20L[7:0] | 10 位占空比寄存器低 8 位 | R/W | 0X00 |

DUTY22H(PWM22, PWM23 的占空比寄存器高 2 位)

地址:0XFC5

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|-----------------|-----------|-------|
| 7:2 | reserved | | | |
| 1:0 | DUTY22H[1:0] | 10 位占空比寄存器高 2 位 | R/W | 11 |

DUTY22L(PWM22, PWM23 的占空比寄存器)

地址: 0XFC4

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

| | | | | |
|-----|--------------|-----------------|-----|------|
| 7:0 | DUTY22L[7:0] | 10 位占空比寄存器低 8 位 | R/W | 0X00 |
|-----|--------------|-----------------|-----|------|

DUTY24H(PWM24, PWM25 的占空比寄存器)

地址: 0XFC3

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|-----------------|-----------|-------|
| 7:2 | reserved | | | |
| 1:0 | DUTY24L[1:0] | 10 位占空比寄存器高 2 位 | R/W | 11 |

DUTY24L(PWM24, PWM25 的占空比寄存器)

地址: 0XFC2

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|-----------------|-----------|-------|
| 7:0 | DUTY24L[7:0] | 10 位占空比寄存器低 8 位 | R/W | 0X00 |

注: TMR2、PR2 和 DUTYn 寄存器在写寄存器时, 必须先写高字节再写低字节; n=20/22/24;

T2CON0(timer2 控制寄存器)

地址: 0XFC1

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|---|-----------|-------|
| 7:3 | T2OUTPS[4:0] | Timer2 分频选择位 00000: 1:1 分频 10000: 1:17 分频 00001: 1:2 分频 10001: 1:18 分频 00010: 1:3 分频 10010: 1:19 分频 00011: 1:4 分频 10011: 1:20 分频 00100: 1:5 分频 10100: 1:21 分频 00101: 1:6 分频 10101: 1:22 分频 00110: 1:7 分频 10110: 1:23 分频 00111: 1:8 分频 10111: 1:24 分频 01000: 1:9 分频 11000: 1:25 分频 01001: 1:10 分频 11001: 1:26 分频 01010: 1:11 分频 11010: 1:27 分频 01011: 1:12 分频 11011: 1:28 分频 01100: 1:13 分频 11100: 1:29 分频 01101: 1:14 分频 11101: 1:30 分频 01110: 1:15 分频 11110: 1:31 分频 01111: 1:16 分频 11111: 1:32 分频 | R/W | 0 |
| 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 | R/W | 0 |

| | | | | |
|--|--|--|--|--|
| | | 00: 预分频值为 1 01: 预分频值为 4 10: 预分频值为 16 11: 预分频值为 64 | | |
|--|--|--|--|--|

T2CON1 (Timer2 控制寄存器)

地址: 0XFC0

| Bit | Name | Description | Attribute | Reset |
|-----|----------|--|-----------|-------|
| 7:2 | Reserved | | | |
| 1:0 | T2M[1:0] | Timer2 工作模式 00: Timer2 工作在定时模式, TMR2==PR2 时, 产生中断标志位。 10: Timer2 PWM 工作在普通边沿模式 11: Timer2 PWM 工作在中心对齐模式 | R/W | 00 |

DTC(死区时间控制寄存器)

地址: 0XFBF

| 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: $F_{DT}=F_{sys}/4$ 10: $F_{DT}=F_{sys}/16$ 11: $F_{DT}=F_{sys}/64$ | R/W | 00 |
| 5:0 | DTD[5:0] | 死区时间计数器死区时间 $=(DTD[5:0]+1)/F_{DT}$ | R/W | 0x00 |

POLS(极性选择寄存器)

地址: 0XFB

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|--|-----------|-------|
| 7:6 | PWM_MUX[1:0] | PWM 引脚映射 00: PWM20,PWM21 映射至 PB2,PB1; PWM22,PWM23 映射至 PB0,PA0; PWM24,PWM25 映射至 PA1,PA2; 01: PWM20,PWM21 映射至 PA3,PA4; PWM22,PWM23 映射至 PA5,PB7; PWM24,PWM25 映射至 PB6,PA6; | R/W | 00 |

| | | | | |
|---|----------|--|-----|---|
| | | 10: PWM20,PWM21 映射至 PB5,PB4; PWM22,PWM23 映射至 PB3,PB2; PWM24,PWM25 映射至 PB1,PB0; 11: PWM20,PWM21 映射至 PB6,PA6; PWM22,PWM23 映射至 PB5,PB4; PWM24,PWM25 映射至 PB3,PB2; | | |
| 5 | HBPOLS25 | HBPWM25 输出极性控制 1: 反相输出 0: 同相输出 | R/W | 0 |
| 4 | HBPOLS24 | HBPWM24 输出极性控制 1: 反相输出 0: 同相输出 | R/W | 0 |
| 3 | HBPOLS23 | HBPWM23 输出极性控制 1: 反相输出 0: 同相输出 | R/W | 0 |
| 2 | HBPOLS22 | HBPWM22 输出极性控制 1: 反相输出 0: 同相输出 | R/W | 0 |
| 1 | HBPOLS21 | HBPWM21 输出极性控制 1: 反相输出 0: 同相输出 | R/W | 0 |
| 0 | HBPOLS20 | HBPWM20 输出极性控制 1: 反相输出 0: 同相输出 | R/W | 0 |

PME 寄存器

地址: 0XFBD

| Bit | Name | Description | Attribute | Reset |
|-----|----------|---|-----------|-------|
| 7 | UART_MUX | UART 的引脚映射 1: PB5(TX)和 PB4(RX) 0: PA5(TX)和 PB7(RX) 注意: 配置字 DEBUG_UART=2'b11 时; | R/W | 0 |

| | | | | |
|---|-----------|---|-----|---|
| | | 当 VPP 为高电平时，硬件自动将 UART_MUX 功能映射成 0，并同时使能调试模式； 当 VPP 为低电平，硬件恢复 UART_MUX 寄存器功能，CPU 可以正常控制 UART 的寄存器。 | | |
| 6 | DTEN | 死区时间使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 5 | HBPWMOE25 | HBPWM25 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 4 | HBPWMOE24 | HBPWM24 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 3 | HBPWMOE23 | HBPWM23 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 2 | HBPWMOE22 | HBPWM22 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 1 | HBPWMOE21 | HBPWM21 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 0 | HBPWMOE20 | HBPWM20 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |

PWM2ADDLY(PWM 的控制寄存器)

地址: 0XFBB

| Bit | Name | Description | Attribute | Reset |
|-----|----------------|---|-----------|-------|
| 7:0 | PWM2ADDLY[7:0] | ADC 采集延时时间: $T = \{PWM2ADDLY[7:0]\} * TMR2CLK$ | R/W | 0 |

HBCON(PWM 的控制寄存器)

地址: 0XFBC

| Bit | Name | Description | Attribute | Reset |
|-----|----------|-------------|-----------|-------|
| 7 | Reserved | | | |

| | | | | |
|-----|----------|---|-----|---|
| 6 | CMP0FLT | CMP0OUT 为故障输入，当从 1 变为 0 后，禁止 PWM 输出 1: 使能 CMP0OUT 作为故障输入 0: 禁止 CMP0OUT 作为故障输入 | R/W | 0 |
| 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 | PWMADEN | 使能 PWM 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 |
| 2:1 | PWMADPOS | PWM 跳变沿使能 ADC 采集 1x: 中心对齐模式: TMR2==PR2 或者 TMR2==8'H0 时，启动 PWMADDLY 计数器，当计数器为 0 时启动 ADC 采样 01: PWM20 边沿跳变时，TMR2==DUTY20 时，启动 PWMADDLY 计数器，当计数器为 0 时启动 ADC 采样 00: PWM20 边沿跳变时,TMR2==PR2 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 | Reserved | | | |

7.4 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[31])位被清零。

配置字 WDTE 位(配置选项 1[2])与 PCON 的 WDTEN 位(PCON[0])都可以单独控制看门狗定时器。

如 WDTEN 位(WDTON[0])与配置字 WDTE 位(配置选项 1[2])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 31.5/4.03/246.56/62.2ms，这个时间可以通过配置字 1 的 WDTPS 位 (配置选项 1[1:0])设置。

CLRWDI 指令能使 WDT 和 TO 清零，启用看门狗可以防止超时，如果超时 MCU 能复位。

芯片处于调试模式中，WDT 被禁止使用。

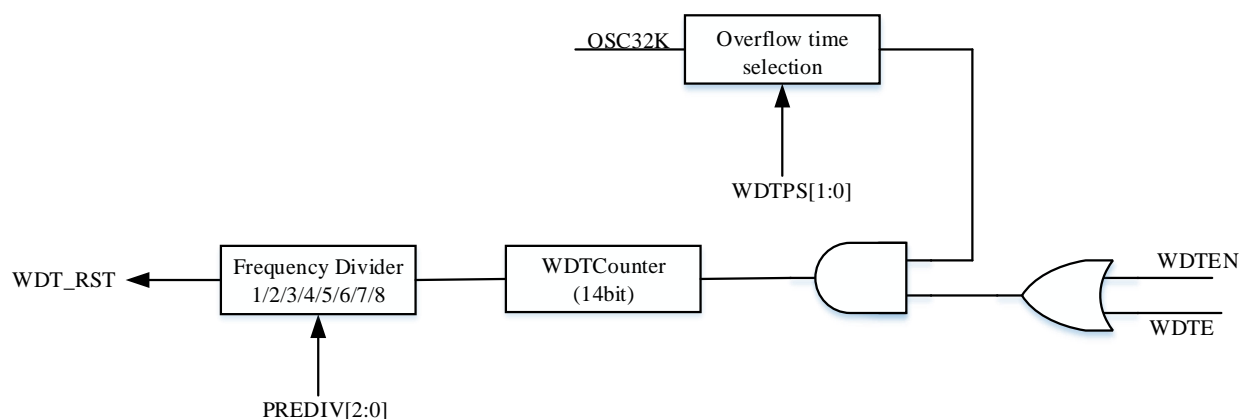


图43 WDT 结构图

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 | PLL | 时钟倍频选择 1: 系统时钟为 32M 0: 系统时钟为 16M | R/W | 0 |
| 5:4 | Reserved | | | - |
| 3:1 | PREDIV[2:0] | WDT 分频 000: 1:1 001: 1:2 010: 1:3 011: 1:4 | R/W | 000 |

| | | | | |
|---|-------|--|-----|---|
| | | 100: 1:5 101: 1:6 110: 1:7 111: 1:8 | | |
| 0 | WDTEN | WDT 使能选择 1: 允许使能 WDT 0: 禁止使能 WDT | R/W | 0 |

8 UART

通用异步收发器(Universa Asynchronous Receiver Transmitter, UART)模块是两个串行 I/O 模块之一。(通常, UART 也被称为串行通信接口)。支持异步通信正常模式, 可编程 8 位或者 9 位字符长度以及 1M 波特率下收发通信。支持单线半双工模式。

8.1 相关寄存器

表 8-1 UART 的 IO 引脚映射

| UART_MUX | TX | RX |
|----------|-----|-----|
| 0 | PA5 | PB7 |
| 1 | PB5 | PB4 |

UART 的 IO 引脚映射: PA5 和 PB7 为一组; PB5 和 PB4 为另外一组

UARTCON 寄存器

地址: 0XFB7

| Bit | Name | Description | Attribute | Reset |
|-----|-------------|---|-----------|-------|
| 7 | SPEN | 串口使能位 1: 使能串口(将 RX/DT 和 TX/CK 引脚配置为串口引脚) 0: 禁止串口(保持在复位状态) | R/W | 0 |
| 6 | MODE9 | UART 的 9bit 模式的选择 1: 使能选择 9bit 模式 0: 禁止选择 9bit 模式 | R/W | 0 |
| 5 | HALF_DUPLEX | 单线半双工使能位 1: 使能 0: 禁止 | R/W | 0 |
| 4 | TX_EN | 发送使能位 1: 使能发送 0: 禁止发送 | R/W | 0 |
| 3 | BRGM | 与 BRGH 组成 UART 时钟选择位 | R/W | 0 |
| 2 | BRGH | 高波特率选择位 异步模式: 1: 高速 0: 低速 同步模式: | R/W | 0 |

| | | | | |
|---|---------|--|-----|---|
| | | 在此模式下未使用。 | | |
| 1 | TX9_RX9 | 发送数据时：9 位发送使能位 1：选择 9 位发送 0：选择 8 位发送 接收数据时：9 位接收使能位 1：选择 9 位接收 0：选择 8 位接收 | R/W | 0 |
| 0 | BUSY | UART 的工作状态位 1：处于工作中状态 0：处于空闲状态 | R | 0 |

SPBRG

地址：0XFB6

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

RCREG

地址：0XFB5

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

TXREG

地址：0XFB4

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

PME 寄存器

地址：0XFBD

| Bit | Name | Description | Attribute | Reset |
|-----|----------|---|-----------|-------|
| 7 | UART_MUX | UART 的引脚映射 1：PB5(TX)和 PB4(RX) 0：PA5(TX)和 PB7(RX) 注意：配置字 DEBUG_UART=2'b11 时； 当 VPP 为高电平时，硬件自动将 UART_MUX 功能映射成 0，并同时使能调试模式； 当 VPP 为低电平，硬件恢复 UART_MUX 寄存器功能，CPU 可以正常控制 UART 的寄存器。 | R/W | 0 |
| 6 | DTEN | 死区时间使能 1：允许使能 | R/W | 0 |

| | | | | |
|---|-----------|---|-----|---|
| | | 0: 禁止使能 | | |
| 5 | HBPWMOE25 | HBPWM25 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 4 | HBPWMOE24 | HBPWM24 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 3 | HBPWMOE23 | HBPWM23 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 2 | HBPWMOE22 | HBPWM22 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 1 | HBPWMOE21 | HBPWM21 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 0 | HBPWMOE20 | HBPWM20 映射的 IO 输出使能 1: 允许使能 0: 禁止使能 | R/W | 0 |

8.2 波特率发生器

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

表 8-2 波特率计算表格

| BRGM | BRGH | 波特率公式 |
|------|------|------------------------|
| 1 | 0 | $F_{sys}/[4*(SPBRG)]$ |
| 1 | 1 | $F_{sys}/[2*(SPBRG)]$ |
| 0 | 0 | $F_{sys}/[64*(SPBRG)]$ |
| 0 | 1 | $F_{sys}/[16*(SPBRG)]$ |

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

目标波特率= $F_{sys}/(64*[SPBRG])$

求解 SPBRG：

$$X=((F_{sys}/\text{目标波特率})/64)$$

$$=((16000000)/9600/64)$$

$$=[26.042]=26$$

计算得到的波特率= $16000000/(64 \times 26)$

$$=9615$$

误差=(波特率计算结果-目标波特率)/目标波特率

$$=(9615-9600)/9600=0.16\% \text{ 误差}$$

目标波特率为 115200bps 的器件:

$$\text{目标波特率} = F_{\text{sys}} / (2 \times [\text{SPBRG}])$$

求解 SPBRG:

$$X = (F_{\text{sys}} / \text{目标波特率}) / 2$$

$$= ((16000000) / 115200) / 2$$

$$= [69.444] = 69$$

计算得到的波特率 = $16000000 / (2 \times 69)$

$$= 115942$$

误差=(波特率计算结果-目标波特率)/目标波特率

$$= (115942 - 115200) / 115200 = 0.64\% \text{ 误差}$$

8.3 异步发送

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

一旦 TXREG 寄存器向 TSR 寄存器传输了数据(在 1 个 T_{CY} 内发生), TXREG 寄存器就为空,同时标志位 TXIF(PIR2<1>)置 1。可以通过将中断使能位 TXIE(PIE2<1>)置 1 或清零来使能/禁止该中断。不管 TXIE 的状态如何,只要中断发生, TXIF 就会置 1 并且不能用软件清零,而是在 TXREG 装入新数据时被清零。标志位 TXIF 指示的是 TXREG 寄存器的状态,而另一位 BUSY(UARTCON<0>)则指示 TSR 寄存器的状态。BUSY 是只读位,它在 TSR 寄存器为空时被置 1。BUSY 位与任何中断均无关联,因此要确定 TSR 寄存器是否为空,用户只能对此位进行轮询。

设置异步发送的操作步骤如下:

- 初始化 SPBRG, 设置合适的波特率。按需要将 BRGH 位置 1 或清零,以获得目标波特率。
- 若需要中断,将使能位 TXIE 置 1。
- 若需要发送 9 位数据,需将 MODE9 置 1。
- 通过将 TX_EN 位置 1 使能发送,此操作同时也会将 TXIF 位置 1。
- 如果选择发送 9 位数据,应该将第 9 位数据装入 TX9_RX9 位。
- 将数据装入 TXREG 寄存器(启动发送)。
- 若想使用中断,请确保将 INTCON 寄存器中的 GIEH/GIEL(INTCON<7>/INTCON<6>)置 1。

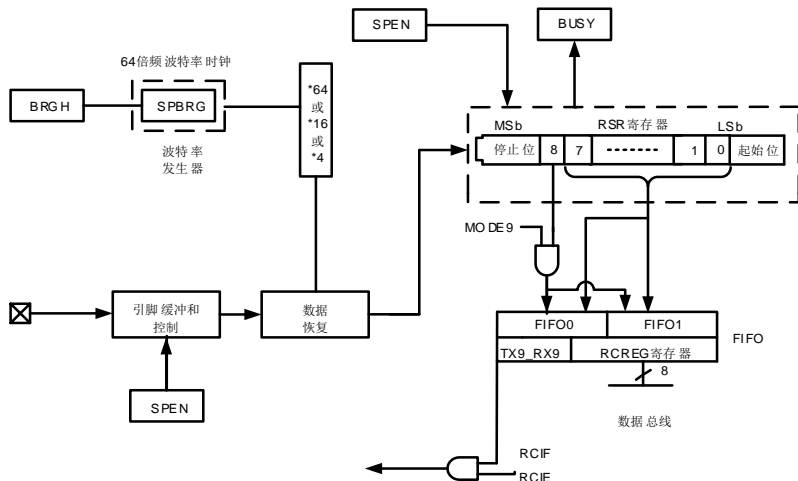


图46 UART接收原理图

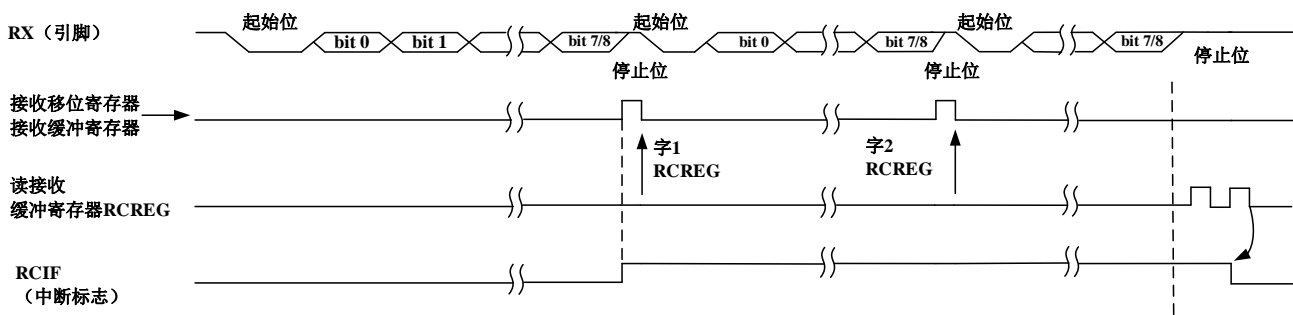


图47 异步接收时序

8.5 单线半双工

单线半双工模式通过设置 UARTCON 寄存器的 HALF_DUPLEX 位选择。当 HALF_DUPLEX 为'1'时，UART 遵循单线半双工协议。在该模式下，RX 不在被使用，当没有数据传输时，TX 总是被释放。因此，在空闲状态或接收状态时表现为一个标准 I/O 口。该 I/O 在不被 UART 驱动时，应配置成悬空输入。除此之外，通信与正常 UART 模式类似，由软件来管理线上的冲突。当 TX_EN 置 1 时，只要数据写入到 TXREG，发送就继续。单线工作时，要求外部总线有上拉电阻或者打开内部电阻。

设置单线半双工模式的操作步骤如下：

- 初始化 SPBRG，设置合适的波特率。按需要将 BRGH 位置 1 或清零，以获得目标波特率。
- 如果需要中断，将使能位 TXIE、RCIE 置位。
- 若需要发送或接收 9 位数据，需将 MODE9 置 1。
- 发送时，需将 TX 口配置为输出，TX_EN 置 1；接收时，需将 TX 口配置为输入，TX_EN 置 0。
- 当发送或接收完成时标志位 TXIF 或 RCIF 将被置 1，此时如果使能位 TXIE 或 RCIE 已置 1，还将产生相应的中断。
- 读 TX9_RX9 以获取第 9 位数据(如果已使能)，并判断是否在接收过程中是否发生了错误。
- 将数据装入 TXREG 寄存器(启动发送)，并通过读 RCREG 寄存器来读取接收到的 8 位数据。

-
- 若想使用中断，请确保将 INTCON 寄存器中的 GIEH/GIEL(INTCON<7>/INTCON<6>)置 1。

9 固定参考电压(FVR)

FVR 是稳定的参考电压，独立于 VDD；输入通道选择 VBG 时，输出可配置成 1.2V、2.0V、3.0V、4.0V。

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；

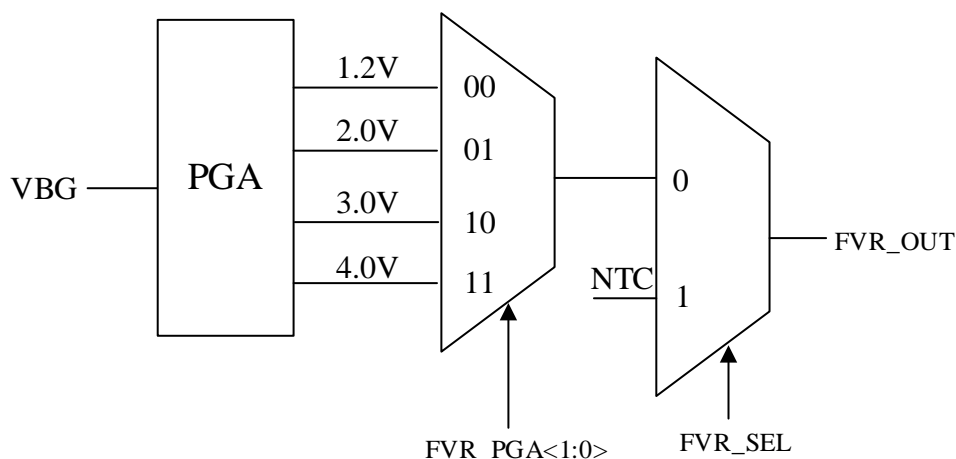


图48 FVR 结构图

FVRCON 寄存器

地址：0XFD3

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|---|-----------|-------|
| 7:6 | Reserved | | | |
| 5 | NTC_EN | NTC 使能 1：使能 NTC 0：NTC 关闭 | | |
| 4 | FVR_SEL | FVR 输入通道选择位 0：VBG 1：NTC | R/W | 0 |
| 3 | FVR_OUTEN | FVR 输出 IO 使能 1：允许使能 0：禁止使能 通过 PB3 输出 | R/W | 0 |
| 2:1 | FVR_PGA[1:0] | FVR 电压放大倍数选择位 | R/W | 00 |

| | | | | |
|---|--------|--|-----|---|
| | | 00: 1.2V 01: 2.0V 10: 3.0V 11: 4.0V | | |
| 0 | FVR_EN | FVR 使能 1: 使能 FVR 0: 禁止 FVR | R/W | 0 |

NTC 说明: NTC 为内部负温度系数电压, 斜率为-6.24mV/°C。NTC 电压与温度对应关系如下(该数据仅为数据参考值, 并不是每个芯片测试):

表 9-1 NTC 电压与温度对应关系

| Temperature(°C) | NTC(V) | Temperature(°C) | NTC(V) |
|-----------------|--------|-----------------|--------|
| -40 | 2.2208 | +30 | 1.7842 |
| -30 | 2.1588 | +40 | 1.7215 |
| -20 | 2.0966 | +50 | 1.6589 |
| -10 | 2.0342 | +60 | 1.5963 |
| 0 | 1.9718 | +70 | 1.5336 |
| +10 | 1.9093 | +80 | 1.4710 |
| +20 | 1.8468 | | |

NTC 可通过 FVR 模块配置 FVR_SEL=1'b1 进行输出, 经 ADC 采集 FVR 输出并查询上表实现实时温度读取。

10 模数转换器(ADC)

AD18E211 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, $\overline{\text{GO/DONE}}$ 位被硬
件清零且 A/D 中断标志位 ADIF 被置 1。

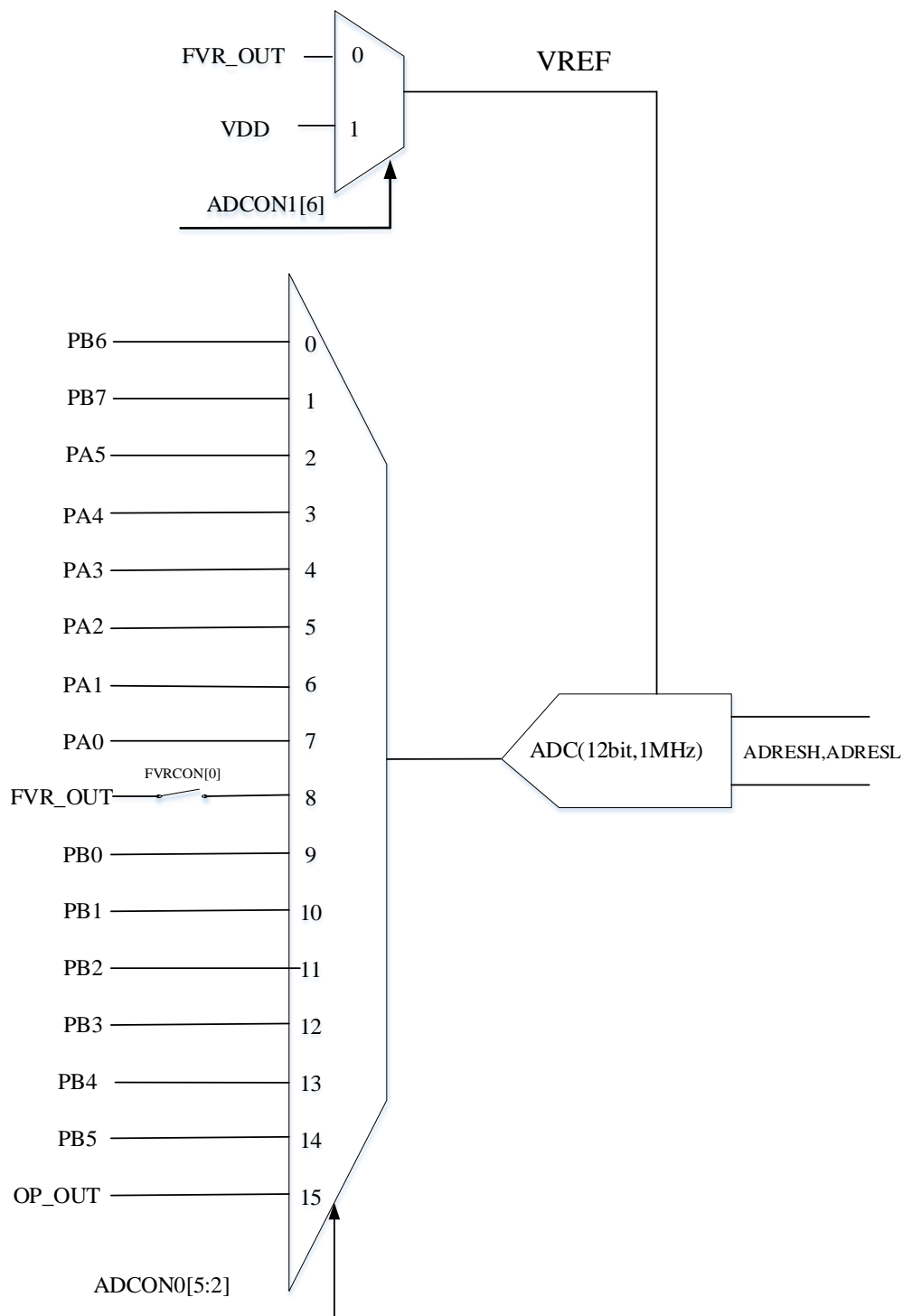


图49 A/D 结构图

执行 A/D 转换步骤:

配置 A/D 模块

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

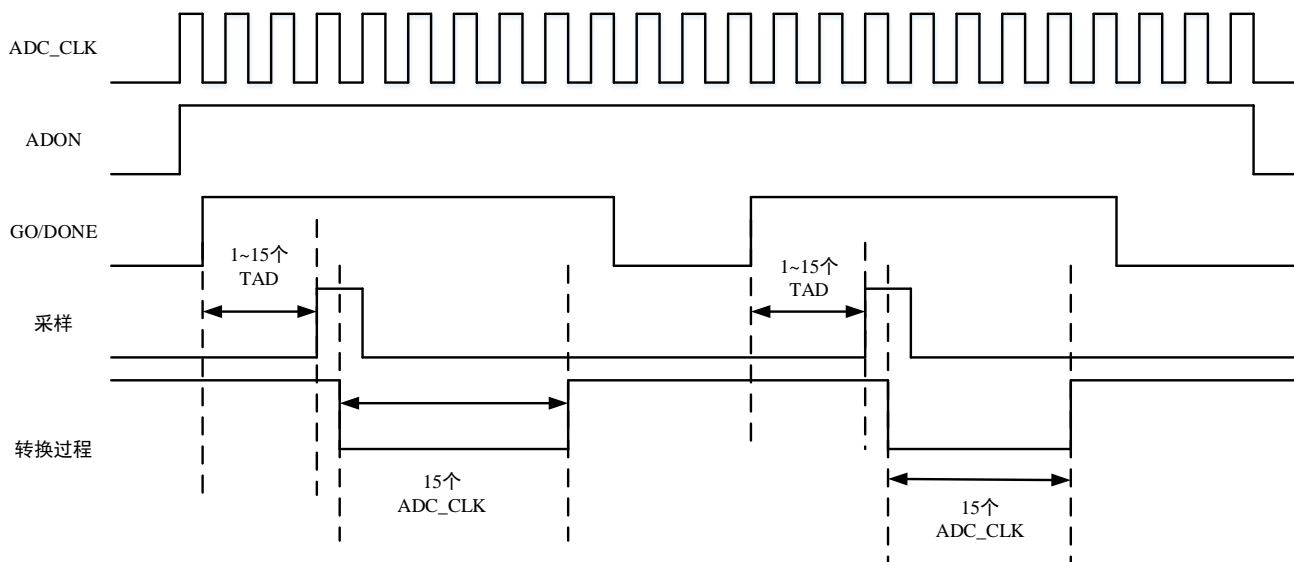


图50 ADC 时序转换图

外部输入最大阻抗计算公式:

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

该公式用于确定允许误差低于 1/4LSB 的最大外部阻抗，N=12；

$$f_{ADC}=8\text{MHz}; C_{ADC}=2.433\text{pF};$$

表 10-1 外部信号源与时钟频率

| VDD(V) | R _{ADC} (KΩ) | T _s (Cycles) | t _s (μs) | R _{AIN max} (KΩ) |
|--------|-----------------------|-------------------------|---------------------|---------------------------|
| 5.0 | 2.501 | 3 | 0.375 | 13.382 |
| 3.3 | 3.276 | 3 | 0.375 | 12.607 |

ADRESH(AD 转换结果的高 4 位)

地址：0XFA3

| 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：0XFA2

| 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：0XFA1

| 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 | Reserved | | | |
| 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(FVR_OUT) | R/W | 0000 |

| | | | | |
|---|---------|--|-----|---|
| | | 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) 1111: ADC 通道 15(OP_OUT) | | |
| 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: 0XFA0

| Bit | Name | Description | Attribute | Reset |
|-----|-----------|--|-----------|-------|
| 7 | Reserved | | | |
| 6 | ADVREF | A/D 参考电压选择位 0: FVR_OUT 1: VDD | R/W | 0 |
| 5:3 | ACQT[2:0] | A/D 延时采集时间选择 111: 15TAD 110: 13TAD 101: 11TAD 100: 9TAD 011: 7TAD 010: 5TAD 001: 3TAD 000: 1TAD 1 个 TAD 为 1 个 ADC_CLK | R/W | 0 |
| 2:0 | ADCS[2:0] | A/D 转换时钟选择位 111: OSC32K 110: Fsys/64 | R/W | 000 |

| | | | | |
|--|--|--|--|--|
| | | 101: Fsys/16 100: Fsys/4 011: OSC32K 010: Fsys/32 001: Fsys/8 000: Fsys/2 | | |
|--|--|--|--|--|

11 数模转换器(DAC)

AD18E211 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 $(3/4)*VDD$ 。

11.1 DAC 参考电压选择寄存器

DACCON

地址：0XFA4

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

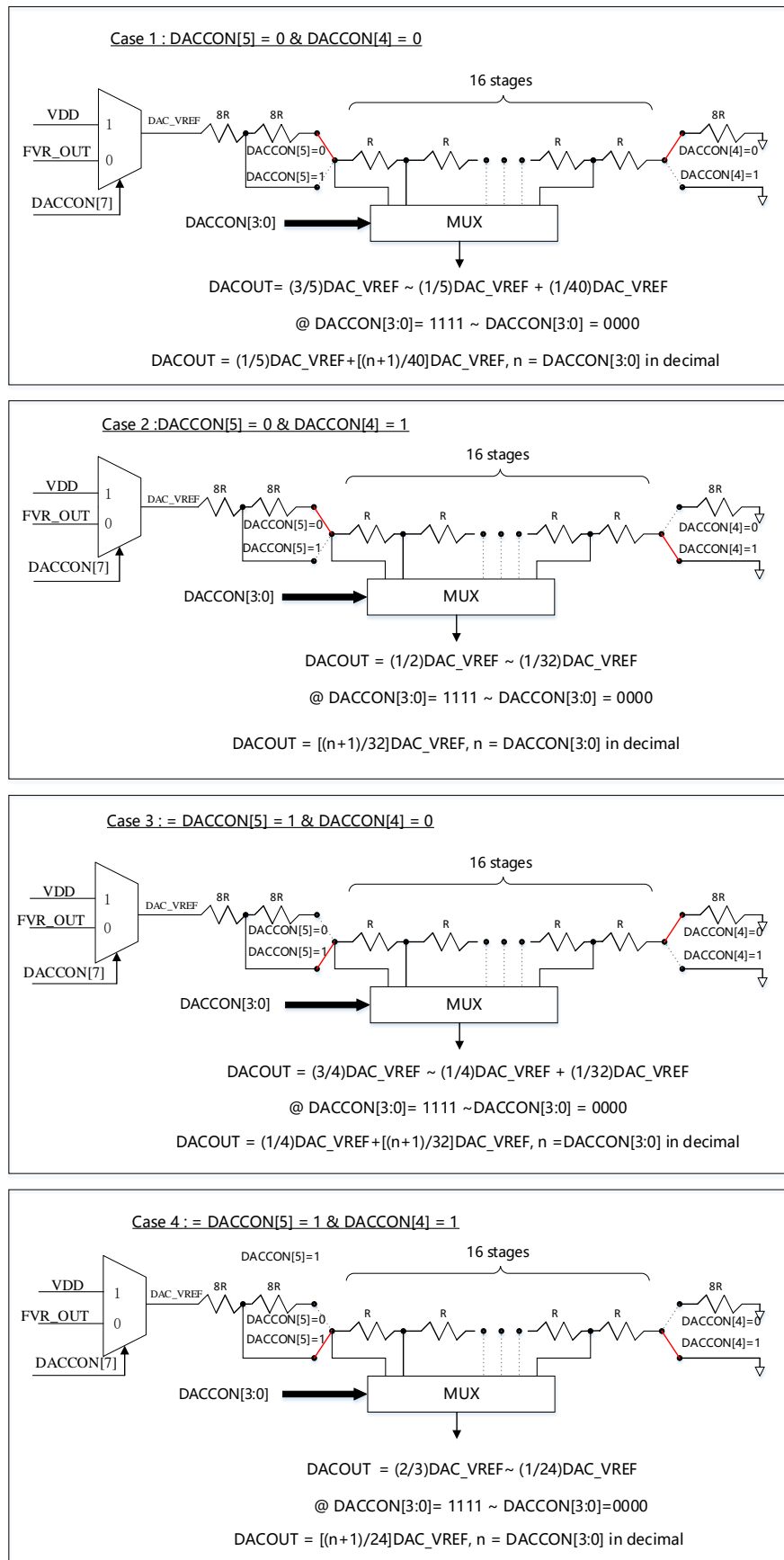


图51 DAC 框图

12 比较器

AD18E211 提供一个比较器，可以选取多个输入作为比较器输入源。CMP0 为轨对轨类型输入，支持输入信号范围 0~VDD。

比较器的迟滞可以通过软件可编程比较器控制寄存器 CMP0CON1 设置，比较器迟滞电压通过 CMP0CON1[0]使能，上升沿和下降沿的迟滞电压都为 20mV。

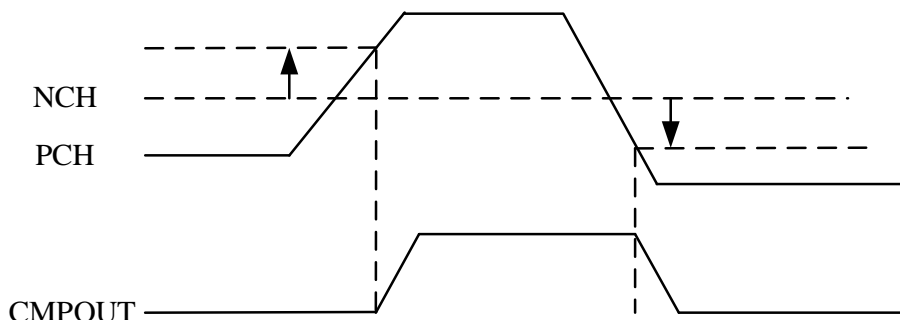


图52 比较器迟滞图

12.1 比较器 CMP0

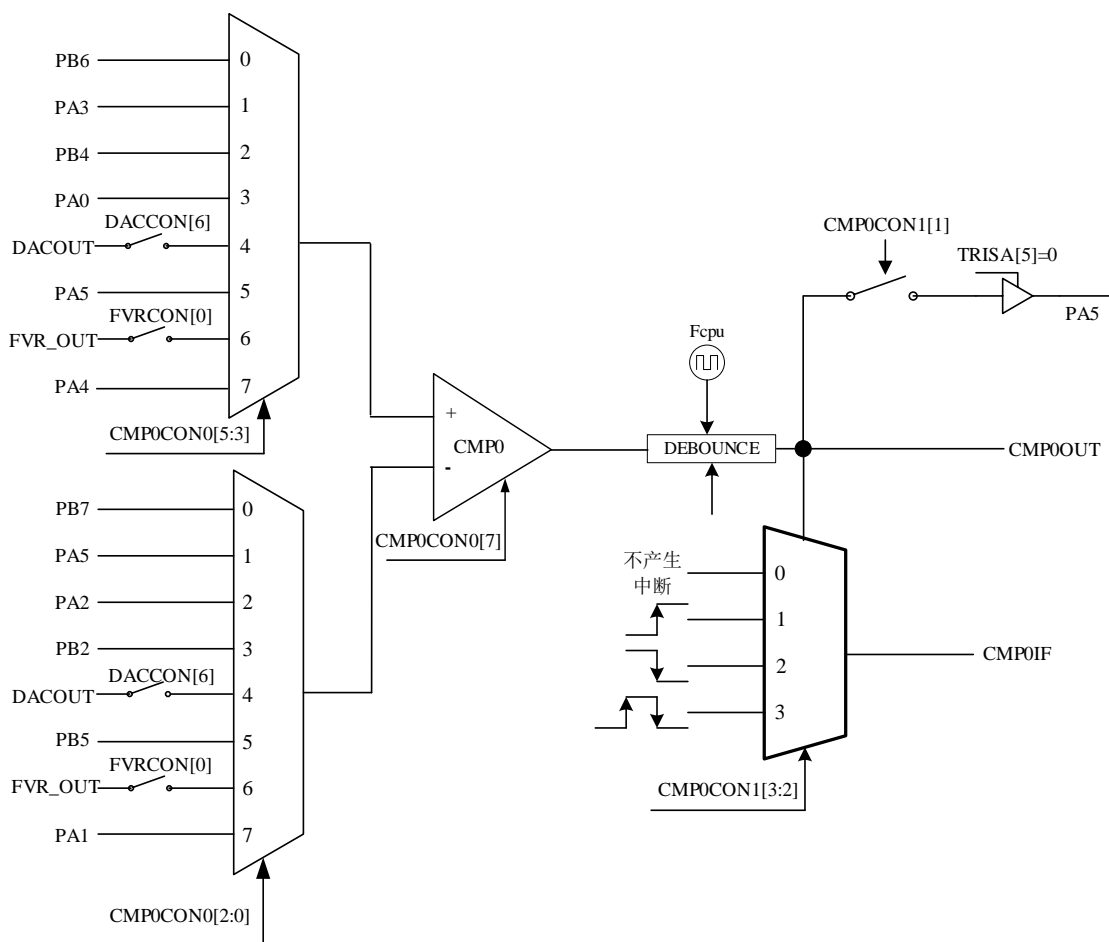


图53 CMP0 电路图

注意事项

1、CMP0 的输出极性见 T2CON1[3]位，工作时钟源的选择见 T2CON1[2]位；

CMP0CON0(比较器控制寄存器)

地址：0XFA8

| Bit | Name | Description | Attribute | Reset |
|-----|--------------|--|-----------|-------|
| 7 | CMP0EN | 比较器的使能 1: 使能 CMP0 0: 禁止 CMP0 | R/W | 0 |
| 6 | CMP0OUT | CMP0 比较器的输出 | R | 0 |
| 5:3 | CMP0PCH[2:0] | CMP0 比较器的正端输入 000: PB6 001: PA3 010: PB4 011: PA0 100: DACOUT 101: PA5 110: FVR_OUT 111: PA4 | R/W | 000 |
| 2:0 | CMP0NCH[2:0] | CMP0 比较器的负端输入 000: PB7 001: PA5 010: PA2 011: PB2 100: DACOUT 101: PB5 110: FVR_OUT 111: PA1 | R/W | 000 |

CMP0CON1(比较器控制寄存器)

地址：0XFA7

| Bit | Name | Description | Attribute | Reset |
|-----|-----------------|--|-----------|-------|
| 7:4 | CMP0FILTER[3:0] | CMP0 的滤波时间 $T = (CMP0FILTER[3:0] \ll 4 + 0X0F) * F_{cpu}$ 有效滤除 CMP0OUT 在 T 时间内的毛刺 CMP0FILTER[3:0]==0x00 时没有滤波 | R/W | 1000 |
| 3:2 | CMP0INT[1:0] | CMP0 中断产生沿选择位 | R/W | 00 |

| | | | | |
|---|-----------|---|-----|---|
| | | 00: 不产生中断 01: 上升沿触发中断 10: 下降沿触发中断 11: 双沿触发中断 | | |
| 1 | CMP0OUTEN | 使能 CMP0OUT 通过 PA5 输出 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 0 | CMP0POSE | CMP0 的输出极性 1: 反向输出 0: 正向输出 | R/W | 0 |

CMP0CON2(比较器控制寄存器)

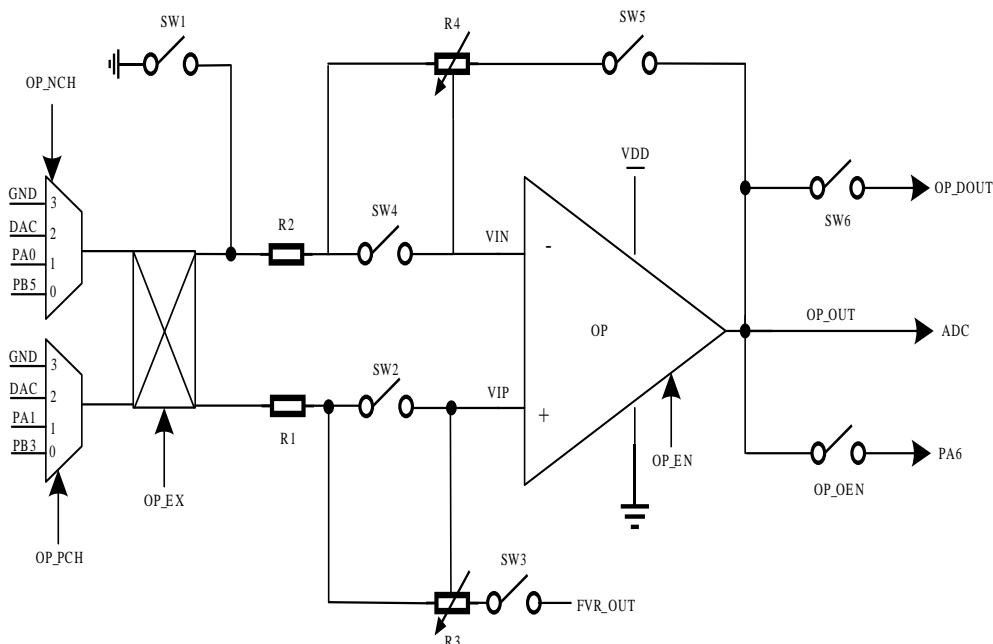
地址: 0XFA6

| Bit | Name | Description | Attribute | Reset |
|-----|----------------|--|-----------|-------|
| 7:4 | CMP0OFFTR[3:0] | CMP0 失调校准位 | R/W | 1000 |
| 3 | CMP0OFFEN | CMP0 失调校准使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 2 | CMP0NHSEN | 下降沿迟滞使能 1: 使能下降沿迟滞 20mV 0: 关闭下降沿迟滞 20mV | R/W | 0 |
| 1 | CMP0PHYSEN | 上升沿迟滞使能 1: 使能上升沿迟滞 20mV 0: 关闭上升沿迟滞 20mV | R/W | 0 |
| 0 | CMP0CLK_CS | CMP0 的工作时钟选择 1: 选择 OSC32K 做工作时钟 0: 选择 F _{CPU} 做工作时钟 | R/W | 0 |

13 运放 OP

AD18E211 提供差分运放 OP，可以选取多个输入作为运放输入源。

13.1 运放 OP



差分放大(OP_DIFF_EN=1)→SW1=0, SW2=0, SW3=1, SW4=0, SW5=1, SW6=0
正向放大(OP_PGA_EN=1)→SW1=1, SW2=1, SW3=0, SW4=0, SW5=1, SW6=0
比较器模式(OP_CMP_EN=1)→SW1=0, SW2=1, SW3=0, SW4=1, SW5=0, SW6=1

图54 OP 电路图

OP 输出计算:

- 正向放大计算公式: $V_{out}=V_{IP}*\beta$
- 差分放大计算公式: $V_{out}=FVR_OUT/2+(\beta-1)*(V_{IP}-V_{IN})$

V_{IN} : 反相端输入电压; V_{IP} : 正相端输入电压; β : 放大倍数($\beta=1/10/20/40$, 差分放大不支持 $\beta=1$)。

注:

1. 使用差分功能时, 需要提前开启 FVR, 且此时 ADC 的参考电压推荐选择 FVR_OUT;
2. 使能差分放大后, 可以使能 OP_EX 交换正负端进行差分放大, 但 PCH<2>、NCH<2>不支持正负端交换;
3. 使能正向放大后, 只能选择正端进行放大, 不支持正负端交换;

OPCON0(OP 控制寄存器 0)

地址: 0XFAA

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

| | | | | |
|-----|--------------------------|--|-----|---|
| 7 | OP_EN | OP 的运放使能 1: 允许使能 0: 禁止使能 | R/W | 0 |
| 6 | OP_EX/OP_CMP_INT[1] | OP_CMP_EN=0 时为 OP_EX: OP 的正负端交换 1: 使能交换 0: 关闭交换 OP_CMP_EN=1 时为 OP_CMP_INT[1]: 配合 OP_CMP_INT[0]使用 | R/W | 0 |
| 5 | OP_DIFF_EN/OP_CMP_INT[0] | OP_CMP_EN=0 时为 OP_DIFF_EN: OP 差分模式使能 1: 使能差分放大模式 0: 关闭差分放大模式 OP_CMP_EN=1 时为 OP_CMP_INT[0]: 配合 OP_CMP_INT[1]使用 00: 不产生中断标志位 01: OPOUT 上升沿中断标志位置 1 10: OPOUT 下降沿中断标志位置 1 11: OPOUT 上升/下降沿中断标志位置 1 | R/W | 0 |
| 4 | OP_CMP_EN | OP 比较模式使能 1: 使能比较器模式 0: 关闭比较器模式 | R/W | 0 |
| 3:2 | OP_PCH | OP 正端输入信号选择 00: PB3 01: PA1 10: DAC 11: AVSS | R/W | 0 |
| 1:0 | OP_NCH | OP 负端输入信号选择 00: PB5 01: PA0 10: DAC 11: AVSS | R/W | 0 |

OPCON1(OP 控制寄存器 1)

地址: 0XFA9

| Bit | Name | Description | Attribute | Reset |
|-----|----------------------------|---|-----------|-------|
| 7 | OPIF | OP 比较模式中断标志位 1: 产生 OP 比较模式中断标志位 0: 未产生 OP 比较模式中断 | R/W | 0 |
| 6 | OPIE | OP 比较模式中断允许位。 1: 使能 OP 比较模式中断 0: 禁止 OP 比较模式中断 | R/W | 0 |
| 5 | OPIP | OP 比较模式中断优先级 1: 高优先级 0: 低优先级 | R/W | 0 |
| 4 | OP_OEN | OP 输出信号映射使能 IO 1: 允许使能输出到 IO 0: 禁止使能输出到 IO PA6 输出 | R/W | 0 |
| 3 | OP_OUT | OP 输出信号(比较器模式有效) | R | 0 |
| 2 | OP_PGA_EN/OP_FILTER[2] | OP_CMP_EN=0 时为 OP_PGA_EN: OP 正向放大使能 1: 使能正向放大模式 0: 关闭正向放大模式 OP_CMP_EN=1 时为 OP_FILTER[2]: 配合 OP_FILTER[1:0]使用 | R/W | 0 |
| 1:0 | OP_PGA[1:0]/OP_FILTER[1:0] | OP_CMP_EN=0 时为 OP_PGA[1:0]: OP 放大倍数选择 00: 1 倍 01: 10 倍 10: 20 倍 11: 40 倍 OP_CMP_EN=1 时为 OP_FILTER[1:0]: 配合 OP_FILTER[2]使用,OP 的滤波时间 $T=(OP_FILTER[2:0] \ll 2 + 0x3) * F_{cpu}$ 有效滤除 OPOUT 在 T 时间内的毛刺 OP_FILTER[2:0]==0x00 时没有滤波 | R/W | 10 |

13.2 OP 比较器模式

步骤 1.使能 OPEN=1、OP_CMP_EN=1，开启 OP 比较器模式；

步骤 2.通过 OP_PCH[1:0]、OP_NCH[1:0]分别选择比较器正负端输入信号；

步骤 3.使能 OP_CMP_EN=1 后，OP_PGA_EN 与 OP_PGA[1:0]自身功能立即失效，此时{OP_PGA_EN，OP_PGA[1:0]}组成滤波时间复用寄存器，即 $T = (\{OP_PGA_EN, OP_PGA[1:0]\} \ll 2 + 2'b11) * F_{cpu}$ ；有效滤除 OP_OUT 在 T 时间内的毛刺；且{OP_PGA_EN，OP_PGA[1:0]}=0x00 时没有滤波；

步骤 4. 使能 OP_CMP_EN=1 后，OP_EX 与 OP_DIFF_EN 自身功能立即失效，此时{OP_EX，OP_DIFF_EN}作为 OP 比较模式产生中断标志位的模式选择位，细节如下；

- (1){OP_EX，OP_DIFF_EN}=2'b00，不产生中断标志位；
- (2){OP_EX，OP_DIFF_EN}=2'b01，OPOUT 上升沿中断标志位置 1；
- (3){OP_EX，OP_DIFF_EN}=2'b10，OPOUT 下降沿中断标志位置 1；
- (4){OP_EX，OP_DIFF_EN}=2'b11，OPOUT 上升/下降沿中断标志位置 1；

步骤 5.可通过 OP_OEN=1，将 OP_OUT 输出到 PA6 上查看比较结果；

14 中断

AD18E211 系统具备以下中断源:

- INT 管脚的外部中断
- Timer0 溢出中断
- Timer1 溢出中断以及上升沿和下降沿捕捉中断
- Timer2 溢出中断
- ADC 中断
- CMP0 中断
- OP 中断
- PWM20、PWM22、PWM24 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(AD18E211 区分中断优先级), 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。

14.1 外部中断

外部中断 INT0 管脚上升沿还是下降沿触发由 INT0EDG 位(T2CON1 寄存器)决定, 当一个有效的跳变发生时标志位 INT0IF 置 1。INT0IE 位清零, 该中断被屏蔽。

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

外部中断 INT1 管脚上升沿还是下降沿触发由 INT1EDG 位(T2CON1 寄存器)决定, 当一个有效的跳变发生时标志位 INT1IF 置 1。INT1IE 位清零, 该中断被屏蔽。

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

外部中断 INT2 管脚上升沿还是下降沿触发由 INT2EDG 位(T2CON1 寄存器)决定, 当一个有效的跳变发生时标志位 INT2IF 置 1。INT2IE 位清零, 该中断被屏蔽。

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

外部中断 INT3 管脚上升沿还是下降沿触发由 INT3EDG 位(T2CON1 寄存器)决定, 当一个有效的跳变发生时标志位 INT3IF 置 1。INT3IE 位清零, 该中断被屏蔽。

在睡眠之前 INT3IE 位已被置 1, INT3 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。

14.2 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。

14.3 Timer1 溢出中断以及上升沿和下降捕捉中断

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 PR1H[15:8]和 PR1L[7:0]组成的 16 位数值与[TIMER1H:TIMER1L]寄存器的值相等时，TMR1IF 标志位置 1。TMR1IE 位清零，该中断被屏蔽。

配置 T1CON1 寄存器的 T1M[1:0]=2'B01，T1CAP_EDGE=0，选择捕捉模式 1，输入第一个上升沿时，CNT1 寄存器清零，下降沿时将捕捉的值读取到 PR1 寄存器，并产生 T1PDIF 中断标志，输入第二个上升沿时将捕捉的值立即读取到 TMR1 寄存器中，并产生 T1PRIF；

配置 T1CON1 寄存器的 T1M[1:0]=2'B01，T1CAP_EDGE=1，选择捕捉模式 1，输入第一个下降沿时，CNT1 寄存器清零，上升沿时将捕捉的值读取到 PR1 寄存器，并产生 T1PDIF 中断标志，输入第二个下降沿时将捕捉的值立即读取到 TMR1 寄存器中，并产生 T1PRIF；

配置 T1CON1 寄存器的 T1M[1:0]=2'B10，T1CAP_EDGE=0，选择捕捉模式 2，当检测到输入的第四个上升沿时，将捕捉的值读取到 PR1 寄存器，并产生 T1PRIF。

配置 T1CON1 寄存器的 T1M[1:0]=2'B10，T1CAP_EDGE=1，选择捕捉模式 2，当检测到输入的第四个下降沿时，将捕捉的值读取到 PR1 寄存器，并产生 T1PRIF。

配置 T1CON1 寄存器的 T1M[1:0]=2'B11，T1CAP_EDGE=0，选择捕捉模式 3，当检测到输入的第十六个上升沿时，将捕捉的值读取到 PR1 寄存器，并产生 T1PRIF。

配置 T1CON1 寄存器的 T1M[1:0]=2'B11，T1CAP_EDGE=1，选择捕捉模式 3，当检测到输入的第十六个下降沿时，将捕捉的值读取到 PR1 寄存器，并产生 T1PRIF。

14.4 Timer2 溢出中断

当 TMR2ON=1 时，Timer2 定时器开始从零计数，在计数的过程中 PR2[9:0]组成的 10 位数值与 TIMER2 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。

14.5 PWM20、PWM22、PWM24 中断

当 T2M[1:0]=2'b10 开启 PWM 模式，当 TMR2ON=1 时，TMR2 寄存器开始计数，在计数的过程中 DUTYn[9:0]组成的 10 位数值与 TMR2 寄存器的值相等时，PWMnIF(PIR1[7:5])标志位置 1。PWMnIE 位(PIE1[7:5])清零，该中断被屏蔽。

当 T2M[1:0]=2'b11 开启 PWM 中心对齐模式，当 TMR2ON=1 时，TMR2 寄存器开始计数，在计数的过程中 DUTYn[9:0]组成的 10 位数值与 TMR2 寄存器的值相等，或 TMR2 寄存器的值为 0 时；PWMnIF(PIR1[7:5])标志位置 1。PWMnIE 位(PIE1[7:5])清零，该中断被屏蔽。

注：n=20/22/24；

14.6 UART 接收、发送中断

当 UART 接收到一个字节数据时，RCIF 标志位置 1。RCIE 位清零，该中断被屏蔽。

当 UART 发送一个字节数据时，TXIF 标志位置 1。TXIE 位清零，该中断被屏蔽。

14.7 比较器中断

通过寄存器 CMP0CON1[3:2]选择不同模式的中断，当 CMP0OUT 从 0 变为 1 时，CMP0IF 置位 (CMP0IF=1)。读取 CMP0CON0 寄存器后，当 CMP0OUT 输出发生改变，CMP0IF 置位 (CMP0IF=1)。CMP0IE 位 (PIE1[6]) 清零，CMP0 中断被屏蔽。

14.8 OP 中断

使能 OPCMPEN=1 开启 OP 比较器模式后：

- (1){OP_EX, OP_DIFF_EN}=2'b00, 不产生 OPIF 标志位；
- (2){OP_EX, OP_DIFF_EN}=2'b01, OPOUT 上升沿 OPIF 标志位置 1, OPIE 位清零，该中断被屏蔽；
- (3){OP_EX, OP_DIFF_EN}=2'b10, OPOUT 下降沿 OPIF 标志位置 1, OPIE 位清零，该中断被屏蔽；
- (4){OP_EX, OP_DIFF_EN}=2'b11, OPOUT 上升/下降沿 OPIF 标志位置 1, OPIE 位清零，该中断被屏蔽；

14.9 ADC 中断

当 GO 从 1 变为 0 时，ADIF 置位为 1，ADIE 位 (PIE3[2]) 清零，该中断被屏蔽

14.10 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=0 时:</u> 1: 使能所有中断 0: 禁止所有中断 <u>当 IPEN=1 时:</u> 1: 使能所有高优先级中断 0: 禁止所有高优先级中断 | R/W | 0 |
| 6 | PEIE/GIEL | 外设中断使能位 <u>当 IPEN=0 时:</u> 1: 使能所有外设中断 0: 禁止所有外设中断 <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 | 0 |
| 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:0XFB1

| Bit | Name | Description | Attribute | Reset |
|-----|---------|---|-----------|-------|
| 7 | PWM24IP | PWM24 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 6 | PWM22IP | PWM22 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 5 | PWM20IP | PWM20 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 4 | TMR2IP | Timer2 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 3 | T1PRIP | Timer1 捕捉上升沿中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 2 | T1PDIP | Timer1 捕捉下降沿中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 1 | TMR1IP | Timer1 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 0 | TMR0IP | Timer0 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |

PIR1 寄存器

地址:0XFB0

| Bit | Name | Description | Attribute | Reset |
|-----|----------|---------------------------------------|-----------|-------|
| 7 | PWM24PIF | PWM24 中断标志, 发生中断置 1, 通过软件写 0 清零 | R/W | 0 |
| 6 | PWM22PIF | PWM22 中断标志, 发生中断置 1, 通过软件写 0 清零 | R/W | 0 |
| 5 | PWM20PIF | PWM20 中断标志, 发生中断置 1, 通过软件写 0 清零 | R/W | 0 |
| 4 | TMR2IF | 溢出中断标志, 发生 Timer2 溢出中断置 1, 通过软件写 0 清零 | R/W | 0 |
| 3 | T1PRIF | Timer1 捕获上升沿中断标志位 | R/W | 0 |

| | | | | |
|---|--------|--|-----|---|
| | | 1: 产生捕捉上升沿中断标志 0: 未产生捕捉上升沿中断 通过软件写 0 清零 | | |
| 2 | T1PDIF | Timer1 捕捉下降沿中断标志位 1: 产生捕捉下降沿中断标志 0: 未产生捕捉下降沿中断 通过软件写 0 清零 | R/W | 0 |
| 1 | TMR1IF | 溢出中断标志, 发生 Timer1 溢出中断置 1, 通过软件写 0 清零 | R/W | 0 |
| 0 | TMR0IF | 溢出中断标志, 发生 Timer0 溢出中断置 1, 通过软件写 0 清零 | R/W | 0 |

PIE1 寄存器

地址:0XF AF

| Bit | Name | Description | Attribute | Reset |
|-----|---------|---|-----------|-------|
| 7 | PWM24IE | PWM24 中断允许位。 1: 使能外部中断 0: 禁止外部中断. | R/W | 0 |
| 6 | PWM22IE | PWM22 中断允许位。 1: 使能外部中断 0: 禁止外部中断. | R/W | 0 |
| 5 | PWM20IE | PWM20 中断允许位。 1: 使能外部中断 0: 禁止外部中断 | R/W | 0 |
| 4 | 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 |
| 3 | T1PRIE | Timer1 捕捉上升沿中断使能 1: 使能捕捉上升沿中断 0: 禁止捕捉上升沿中断 | R/W | 0 |
| 2 | T1PDIE | Timer1 捕捉下降沿中断使能 1: 使能捕捉下降沿中断 0: 禁止捕捉下降沿中断 | 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。 | R/W | 0 |

| | | | | |
|--|--|--|--|--|
| | | 1: 使能 Timer0 溢出中断 0: 禁止 Timer0 溢出中断 | | |
|--|--|--|--|--|

IPR2 寄存器

地址:0XFAE

| Bit | Name | Description | Attribute | Reset |
|-----|--------|-------------------------------------|-----------|-------|
| 7 | CMP0IP | CMP0 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 6 | ADIP | ADC 中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 5 | INT3IP | 外部中断 3 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 | TXIP | UART 的发送中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |
| 0 | RCIP | UART 的接收中断优先级 1: 高优先级 0: 低优先级 | R/W | 1 |

PIR2 寄存器

地址:0XFAD

| Bit | Name | Description | Attribute | Reset |
|-----|--------|-------------|-----------|-------|
| 7 | CMP0IF | CMP0 的中断标志位 | R/W | 0 |

| | | | | |
|---|--------|--|-----|---|
| | | 1: 产生 CMP0 中断标志位(必须用软件清零, 写 0 清 0) 0: 未产生 CMP0 中断 | | |
| 6 | ADIF | ADC 的中断标志位 1: 产生 ADC 中断标志位(必须用软件清零, 写 0 清 0) 0: 未产生 ADC 中断 | R/W | 0 |
| 5 | INT3IF | INT3 外部中断标志位 1: 产生 INT3 外部中断标志位(必须由软件清零, 写 0 清 0) 0: 未产生 INT3 外部中断 | R/W | 0 |
| 4 | INT2IF | INT2 外部中断标志位 1: 产生 INT2 外部中断标志位(必须由软件清零, 写 0 清 0) 0: 未产生 INT2 外部中断 | R/W | 0 |
| 3 | INT1IF | INT1 外部中断标志位 1: 产生 INT1 外部中断标志位(必须由软件清零, 写 0 清 0) 0: 未产生 INT1 外部中断 | R/W | 0 |
| 2 | INT0IF | INT0 外部中断标志位 1: 产生 INT0 外部中断标志位(必须由软件清零, 写 0 清 0) 0: 未产生 INT0 外部中断 | R/W | 0 |
| 1 | TXIF | UART 的发送中断标志位 1: UART 发送缓冲器 TXREG 为空(写入 TXREG 时清零) 0: UART 发送缓冲器已满 | R/W | 0 |
| 0 | RCIF | UART 的接收中断标志位 1: UART 接收缓冲器 RCREG 已满(读取 RCREG 时清零) 0: UART 接收缓冲器为空 | R/W | 0 |

PIE2 寄存器

地址:0XFAC

| Bit | Name | Description | Attribute | Reset |
|-----|--------|--|-----------|-------|
| 7 | CMP0IE | CMP0 中断允许位 1: 使能 CMP0 中断 0: 禁止 CMP0 中断 | R/W | 0 |
| 6 | ADIE | ADC 中断允许位 1: 使能 ADC 中断 0: 禁止 ADC 中断 | R/W | 0 |
| 5 | INT3IE | 外部中断 3 允许位 1: 使能 INT3 中断 | R/W | 0 |

| | | | | |
|---|--------|---|-----|---|
| | | 0: 禁止 INT3 中断 | | |
| 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 | TXIE | UART 的发送中断允许位 1: 使能 UART 的发送中断 0: 禁止 UART 的发送中断 | R/W | 0 |
| 0 | RCIE | UART 的接收中断允许位 1: 使能 UART 的接收中断 0: 禁止 UART 的接收中断 | R/W | 0 |

15 8X8 硬件乘法器

AD18E211 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 AD18E211 器件。

16 省电模式(SLEEP)

拥有两种睡眠模式：(PWDEEP、PWOFF)

- 010: PWDEEP 模式，CPU 停止工作，高速 16M 时钟停止工作，低速 32K 时钟工作，SRAM 数据保持；支持外部中断、复位、看门狗溢出、TIMER1 的 32K 定时唤醒以及 CMP0 选择 OSC32K 工作时可以唤醒，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、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；

16.1 睡眠唤醒

在睡眠状态下，两种模式，单片机能通过以下方式唤醒：如下图所示

表 16-1 两种模式下唤醒方式

| 模块 | 睡眠模式 | |
|--------|-------|--------|
| | PWOFF | PWDEEP |
| TIMER1 | | √ |
| WDT | | √ |
| INT | √ | √ |
| CMP0 | | √ |

注：√表示可唤醒的方式

在睡眠状态下，两种模式可以工作的模块如下图所示

表 16-2 两种模式可以工作的模块

| 模块 | | | |
|--------------------|--------|-------|--------|
| | ACTIVE | PWOFF | PWDEEP |
| OSC16M | √ | | |
| OSC32K | √ | | √ |
| CPU | √ | | |
| SRAM | √ | √ | √ |
| Timer0/2 | √ | | |
| Timer1 | √ | | √ |
| WDT | √ | | √ |
| External Interrupt | √ | √ | √ |
| 内部基准 | √ | | |
| LVR | √ | | |
| ADC | √ | | |
| DAC | √ | | |
| CMP0 | √ | | √ |
| OP | √ | | |
| IO | √ | √ | |

看门狗溢出能使机器复位.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：0XFD4

| Bit | Name | Description | Attribute | Reset |
|-----|----------|---|-----------|-------|
| 7:4 | Reserved | | | |
| 3:1 | SM[2:0] | 休眠模式选择 010: PWDEEP 模式，CPU 停止工作，高速 16M 时钟停止工作，低速 32K 时钟工作；支持外部中断、复位、看门狗溢出、TIMER1 的 32K 定时唤醒，CMP0 选择 OSC32K 工作时可以唤醒，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、复位，唤醒后继续从当前 PC 运行； | R/W | 000 |
| 0 | SE | 休眠模式使能位 1: 使能休眠模式,硬件自动清零 0: 禁止休眠模式 | R/W | 0 |

表 16-3 SLEEP 不同工作模式下的唤醒时间

| 工作模式 | 唤醒时间 |
|-----------|------|
| PWDEEP 模式 | 1ms |
| PWOFF 模式 | 1ms |

17 电气特性

17.1 直流交流电气特性

下列所有数据除特别列明外，皆于 Ta=25°C，VDD=5.0V 条件下获得

| 符号 | 描述 | 最小值 | 典型值 | 最大值 | 单位 | 条件 |
|---------------------|-------------|---------|------|---------|-----|-----------------------|
| VDD | 工作电压 | 2.0 | 5.0 | 5.5 | V | |
| ISAVE | PWDEEP 模式 | | 4 | | uA | VDD=5.0V |
| IOFF | PWOFF 模式 | | 1.5 | | uA | VDD=5.0V |
| VDR | RAM 保留数据电压* | 1.5 | | | V | |
| ILKG | 引脚漏电流* | -2 | - | +2 | uA | |
| VIL | 输入低电压 | 0 | | 0.3*VDD | V | |
| VIH1 | 输入高电压 | 0.7*VDD | | VDD | V | |
| IOH | IO 输出拉电流 | - | 10 | - | mA | VDD=3.3V, VIO=0.9*VDD |
| | | - | 20 | - | | VDD=5.0V, VIO=0.9*VDD |
| IOL | IO 输出灌电流 | - | 30 | - | mA | VDD=3.3V, VIO=0.1*VDD |
| | | - | 60 | - | | VDD=5.0V, VIO=0.1*VDD |
| RPU | I/O 口上拉电阻 | - | 30 | - | KΩ | 除 PA4、PA5、PB6、PB7 |
| | | | 47 | | | PA4、PA5 |
| | | | 5 | | | |
| | | | 10 | | | PB6、PB7 |
| RPD | I/O 口下拉电阻 | - | 30 | - | KΩ | 除 PA4、PA5、PB6、PB7 |
| | | | 10 | | | PA4、PA5 |
| | | | 20 | | | |
| | | | 10 | | | PB6、PB7 |
| fLIRC | 校准后* | -10% | 32 | +10% | KHz | 25°C, 2.2V~5.5V |
| | | -30% | | +30% | | -40°C~85°C, 2.2V~5.5V |
| fHIRC | 校准后* | -2.0% | 16 | +2.0% | MHz | 25°C, 2.5V~5.5V |
| | | -2.0% | | +2.0% | | 0°C~70°C, 2.5V~5.5V |
| CMP _{OS} | 比较器输入失调电压* | - | ±10 | - | mV | |
| CMP _{CM} | 比较器共模输入电压* | 0 | | VDD | V | |
| CMP _{RT} | 比较器响应时间* | - | 100 | - | ns | |
| CMP _{HYS+} | 比较器正端迟滞电压* | - | +20 | - | mV | |
| CMP _{HYS-} | 比较器负端迟滞电压* | - | -20 | - | mV | |
| OP _{OS} | 运放输入失调电压* | | ±4.6 | | mV | |
| OP _{CM} | 运放共模输入电压 | 0 | | VDD-1.5 | V | -40°C~85°C |
| OP _{OH} | 运放最大输出电压 | | | VDD-0.3 | V | -40°C~85°C |
| OP _{OL} | 运放最小输出电压 | 0.3 | | | V | -40°C~85°C |

| | | | | | | |
|---------------------|--|-----|------|------|--------------------|---|
| OP _{AV} | 运放开环增益* | | 90 | | dB | |
| OP _{BW} | 运放带宽* | | 6.75 | | MHz | C _{LOAD} =30pF |
| OP _{PSRR} | 运放电源抑制比* | | 70 | | dB | VDD=2.5~5.5V, V _{in} =1V |
| OP _{CMRR} | 运放共模抑制比* | | 110 | | dB | V _{in} =0.3V~(VDD-1.5V) |
| OP _{SR} | 运放摆率* | | ±3.4 | | V/μs | C _{LOAD} =30pF |
| | | | ±2.8 | | V/μs | R _{LOAD} =2K, C _{LOAD} =100pF |
| OP _{TSTB} | 运放稳定时间 | | | 2 | μs | |
| V _{ADC} | ADC 工作电压 | 2.7 | | VDD | V | |
| V _{AD} | ADC 的输入电压 | 0 | | Vref | V | |
| AD _{TS} | ADC 分辨率 | | | 12 | Bit | |
| T _{ADCONV} | ADC 转换时间(T _{ADCLK} 是选定 AD 转换时钟周期) | | 15 | | T _{ADCLK} | |
| V _{REFH} | ADC 参考电压 1.2V* | -1% | 1.2 | +1% | V | |

注：*这些参数是设计参考值，并不是每个芯片测试。

17.2 HIRC 频率微调参数说明

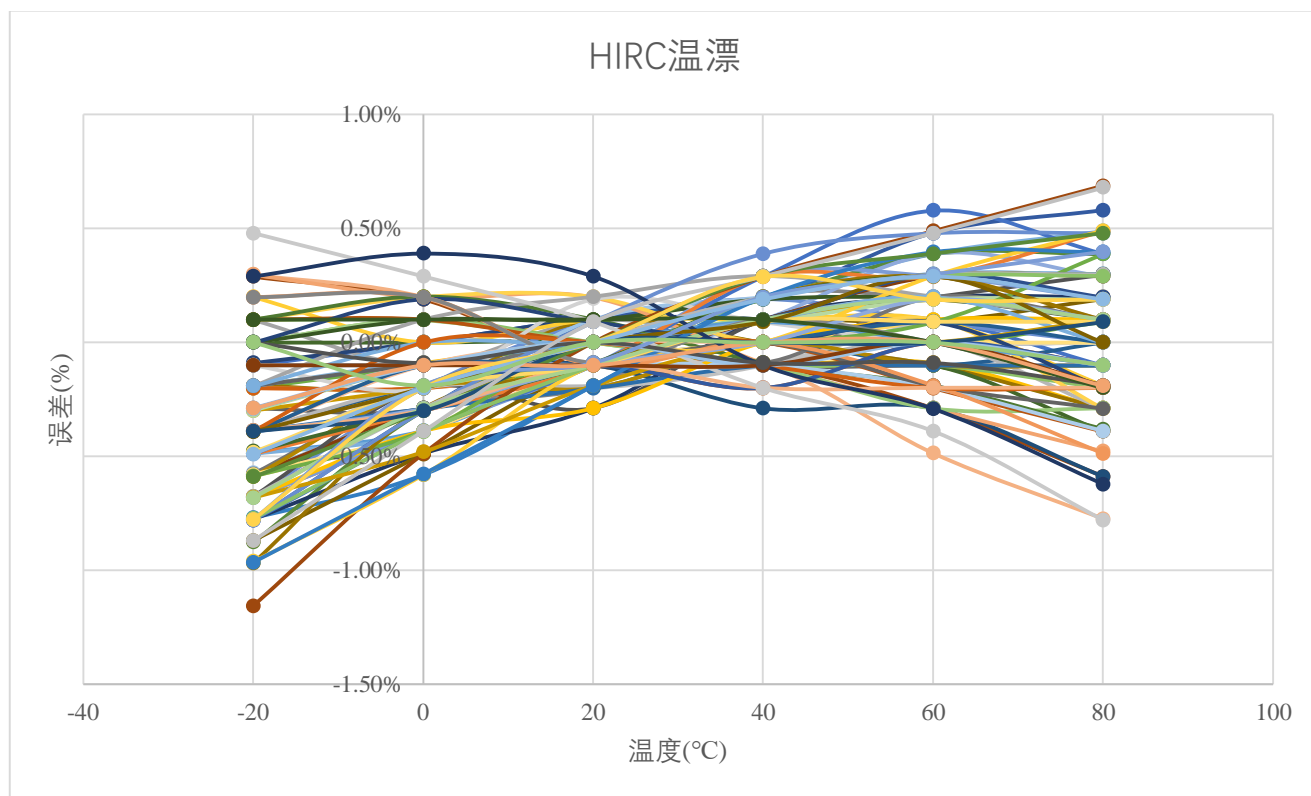
| HIRCCAL 修调值 (十六进制) | 频率(MHz) | 修调偏差(HIRCCAL=0x7F) |
|-----------------------|---------|--------------------|
| 00 | 10.18 | -36.37% |
| 01 | 10.25 | -35.93% |
| 02 | 10.32 | -35.48% |
| 03 | 10.40 | -34.98% |
| 04 | 10.47 | -34.54% |
| 05 | 10.54 | -34.09% |
| 06 | 10.62 | -33.59% |
| 07 | 10.70 | -33.14% |
| 08 | 10.78 | -32.64% |
| 09 | 10.85 | -32.19% |
| 0A | 10.92 | -31.75% |
| 0B | 11.00 | -31.25% |
| 0C | 11.07 | -30.80% |
| 0D | 11.14 | -30.36% |
| 0E | 11.22 | -29.85% |
| 0F | 11.30 | -29.35% |
| 10 | 11.38 | -28.84% |
| 11 | 11.45 | -28.40% |
| 12 | 11.54 | -27.89% |
| 13 | 11.62 | -27.39% |
| 14 | 11.70 | -26.88% |
| 15 | 11.77 | -26.44% |
| 16 | 11.86 | -25.87% |

| | | |
|----|-------|---------|
| 17 | 11.94 | -25.36% |
| 18 | 12.03 | -24.79% |
| 19 | 12.10 | -24.35% |
| 1A | 12.18 | -23.85% |
| 1B | 12.26 | -23.34% |
| 1C | 12.35 | -22.83% |
| 1D | 12.44 | -22.26% |
| 1E | 12.52 | -21.76% |
| 1F | 12.60 | -21.25% |
| 20 | 12.69 | -20.68% |
| 21 | 12.77 | -20.18% |
| 22 | 12.86 | -19.61% |
| 23 | 12.95 | -19.04% |
| 24 | 13.03 | -18.53% |
| 25 | 13.12 | -17.96% |
| 26 | 13.23 | -17.33% |
| 27 | 13.32 | -16.76% |
| 28 | 13.42 | -16.13% |
| 29 | 13.51 | -15.56% |
| 2A | 13.60 | -14.99% |
| 2B | 13.70 | -14.36% |
| 2C | 13.79 | -13.79% |
| 2D | 13.89 | -13.16% |
| 2E | 13.99 | -12.52% |
| 2F | 14.11 | -11.83% |
| 30 | 14.21 | -11.20% |
| 31 | 14.31 | -10.56% |
| 32 | 14.41 | -9.93% |
| 33 | 14.52 | -9.23% |
| 34 | 14.62 | -8.60% |
| 35 | 14.73 | -7.91% |
| 36 | 14.84 | -7.21% |
| 37 | 14.96 | -6.51% |
| 38 | 15.07 | -5.82% |
| 39 | 15.18 | -5.12% |
| 3A | 15.29 | -4.43% |
| 3B | 15.40 | -3.73% |
| 3C | 15.52 | -2.97% |
| 3D | 15.63 | -2.28% |
| 3E | 15.76 | -1.52% |
| 3F | 15.88 | -0.76% |
| 40 | 16.00 | 0.00% |
| 41 | 16.12 | 0.76% |
| 42 | 16.25 | 1.58% |
| 43 | 16.37 | 2.34% |
| 44 | 16.50 | 3.16% |
| 45 | 16.64 | 3.98% |
| 46 | 16.77 | 4.81% |

| | | |
|----|-------|--------|
| 47 | 16.90 | 5.63% |
| 48 | 17.03 | 6.45% |
| 49 | 17.16 | 7.27% |
| 4A | 17.29 | 8.10% |
| 4B | 17.42 | 8.92% |
| 4C | 17.57 | 9.80% |
| 4D | 17.71 | 10.69% |
| 4E | 17.85 | 11.57% |
| 4F | 17.99 | 12.46% |
| 50 | 18.14 | 13.41% |
| 51 | 18.29 | 14.29% |
| 52 | 18.44 | 15.24% |
| 53 | 18.59 | 16.19% |
| 54 | 18.74 | 17.14% |
| 55 | 18.89 | 18.09% |
| 56 | 19.05 | 19.10% |
| 57 | 19.22 | 20.11% |
| 58 | 19.38 | 21.13% |
| 59 | 19.53 | 22.07% |
| 5A | 19.69 | 23.09% |
| 5B | 19.85 | 24.10% |
| 5C | 20.02 | 25.11% |
| 5D | 20.19 | 26.19% |
| 5E | 20.36 | 27.26% |
| 5F | 20.53 | 28.34% |
| 60 | 20.71 | 29.48% |
| 61 | 20.89 | 30.55% |
| 62 | 21.06 | 31.63% |
| 63 | 21.24 | 32.76% |
| 64 | 21.42 | 33.90% |
| 65 | 21.61 | 35.10% |
| 66 | 21.80 | 36.24% |
| 67 | 21.99 | 37.44% |
| 68 | 22.19 | 38.71% |
| 69 | 22.38 | 39.91% |
| 6A | 22.57 | 41.05% |
| 6B | 22.77 | 42.31% |
| 6C | 22.97 | 43.58% |
| 6D | 23.17 | 44.85% |
| 6E | 23.37 | 46.11% |
| 6F | 23.59 | 47.44% |
| 70 | 23.81 | 48.83% |
| 71 | 24.01 | 50.09% |
| 72 | 24.22 | 51.42% |
| 73 | 24.45 | 52.81% |
| 74 | 24.67 | 54.21% |
| 75 | 24.89 | 55.60% |
| 76 | 25.12 | 56.99% |

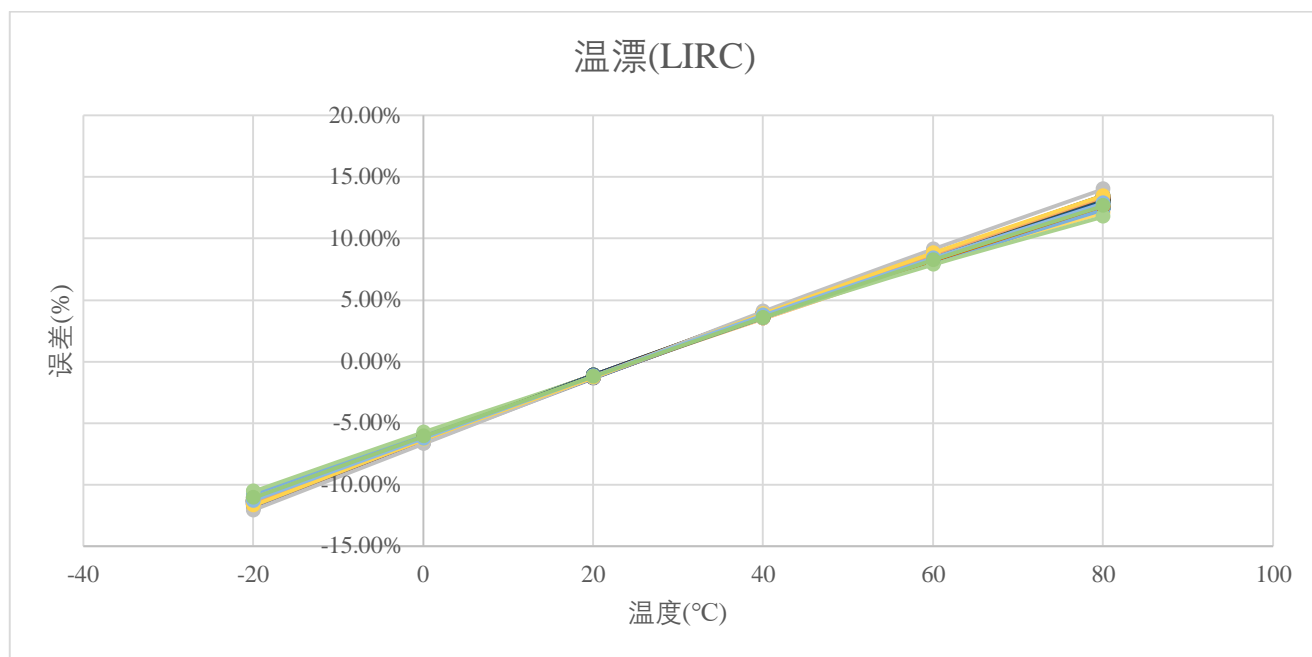
| | | |
|----|-------|--------|
| 77 | 25.35 | 58.44% |
| 78 | 25.58 | 59.90% |
| 79 | 25.81 | 61.35% |
| 7A | 26.05 | 62.81% |
| 7B | 26.29 | 64.33% |
| 7C | 26.53 | 65.84% |
| 7D | 26.78 | 67.43% |
| 7E | 27.03 | 68.94% |
| 7F | 27.29 | 70.59% |

17.3 HIRC 温漂曲线



注：仅供参考，并不是每个芯片测试数据。

17.4 LIRC 温漂曲线



注：仅供参考，并不是每个芯片测试数据。

18 特别注意事项

此章节提醒使用者在使用该 IC 时避免常犯的一些错误；

18.1 看门狗

当 LIRC 关闭时，看门狗也会失效；

18.2 IC 烧录方法

18.2.1 烧录引脚及方法

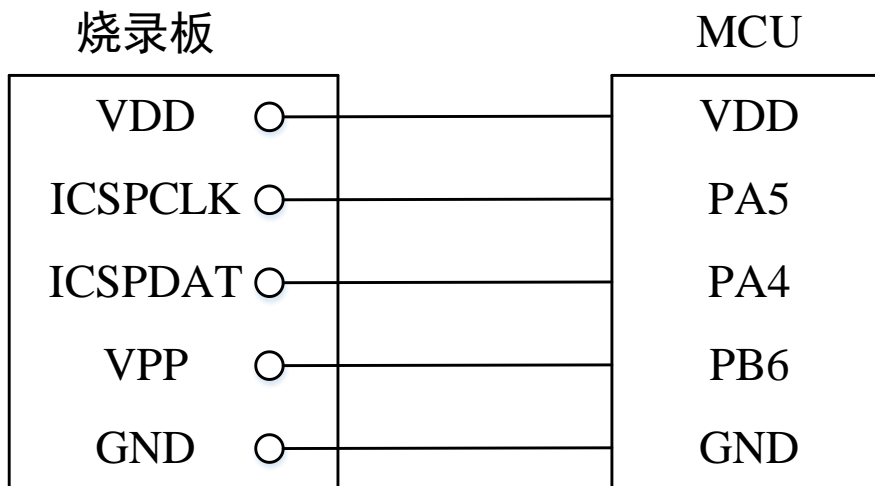


图55 五线烧录示意图

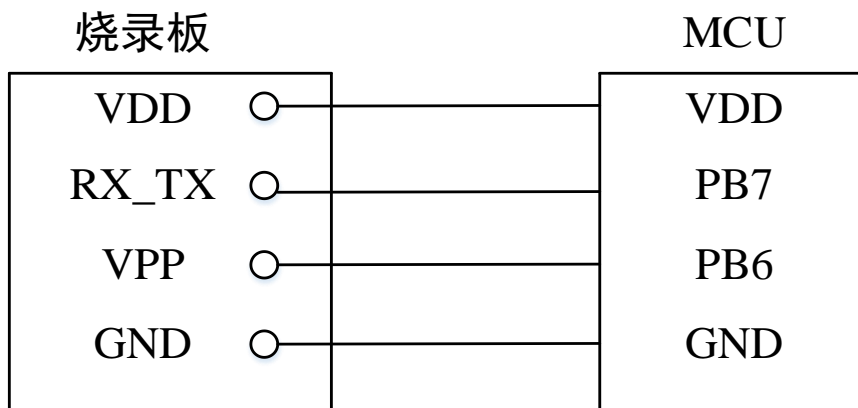


图56 四线烧录示意图

1、烧录引脚为 PA5(ICSPCLK)和 PA4(ICSPDAT)，烧录时 VPP 需要加高压(9.5V~9.8V)，芯片电源电压范围应为(3.3V~5.5V)。如需校准，则根据校准方案引出相应引脚。

- a) I2C 的调试通过配置字 DBGEN_I2C 使能，此时芯片运行过程中 PA5(ICSPCLK)和 PA4(ICSPDAT)无法用作其他功能。

2、芯片时钟校准后可以使用 UART 单线通讯；烧录引脚为 PB7(RX_TX)；烧录时 VPP 需要加高压 (9.5V~9.8V)。

- a) UART 的调试模式通过配置字 DBGEN_UART 选择不同的波特率进行调试，详见配置字 1。
- b) UART 支持多芯片级联烧录，连接方法见章节 1 图 6。

18.2.2 烧录电压条件

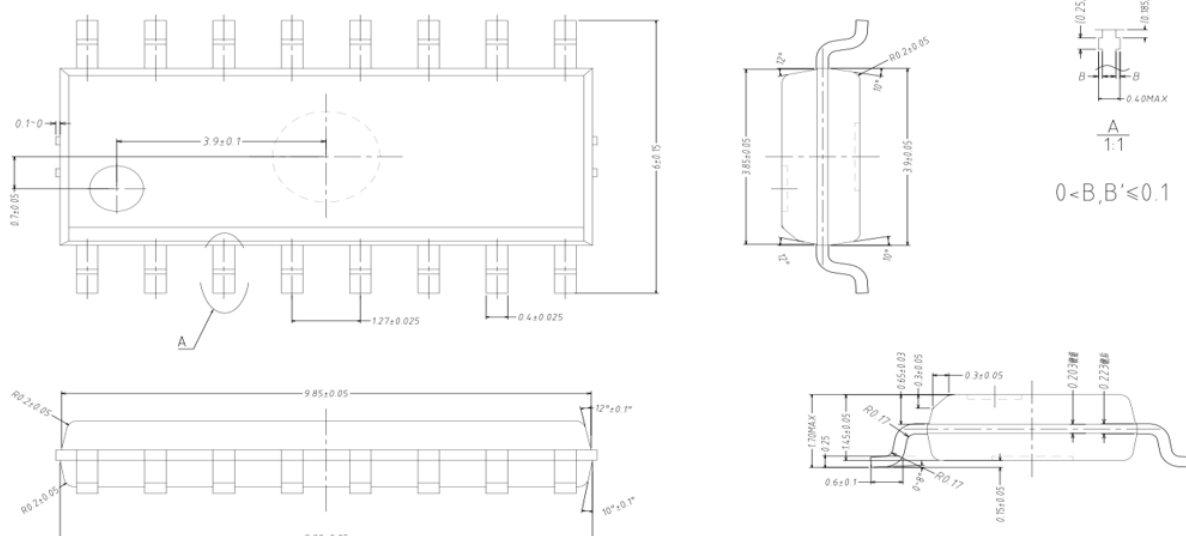
VPP(PB6 引脚)需要 9.5V~9.8V 的电压。

18.2.3 烧录注意事项

- 1、单线下载时会占用串口。
- 2、VPP 电压在写入 MTP 时供给。
- 3、VPP 需要串联 100 欧电阻。

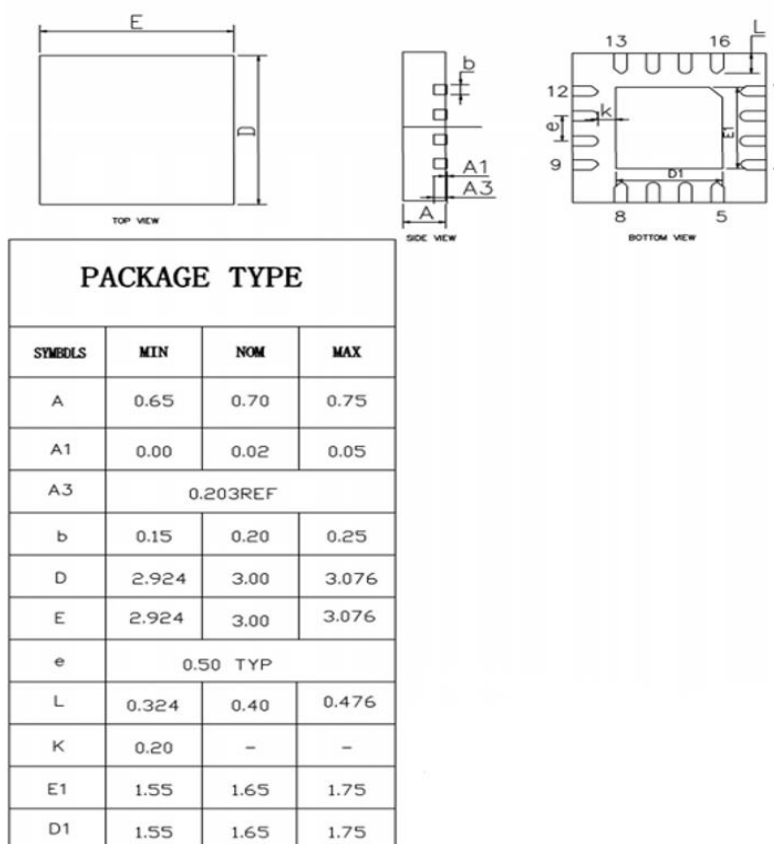
19 封装尺寸

19.1 SOP16

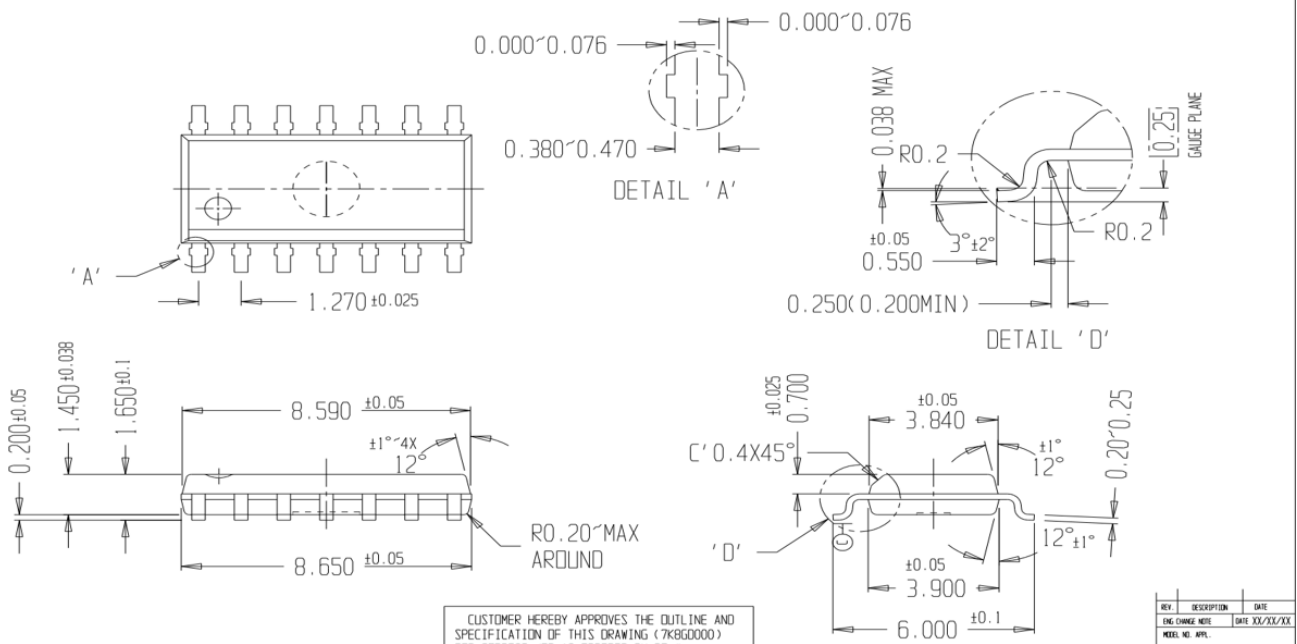


19.2 QFN16

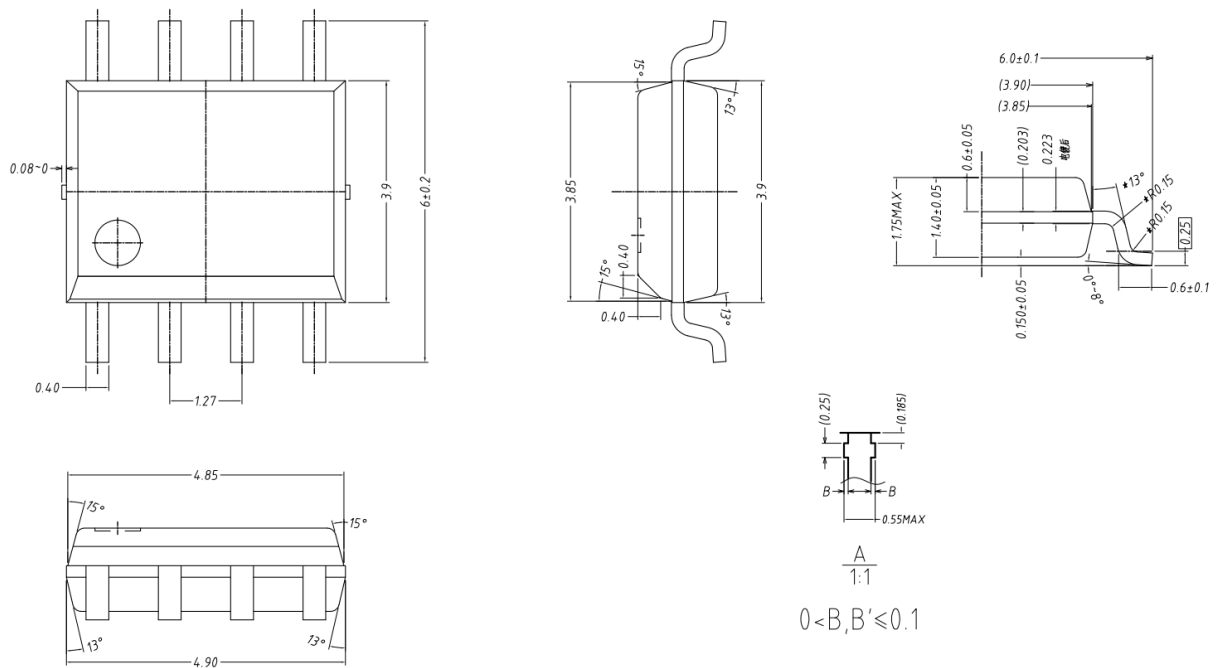
QFN3*3-16L (P0. 5T0. 65/0. 70/0. 75)



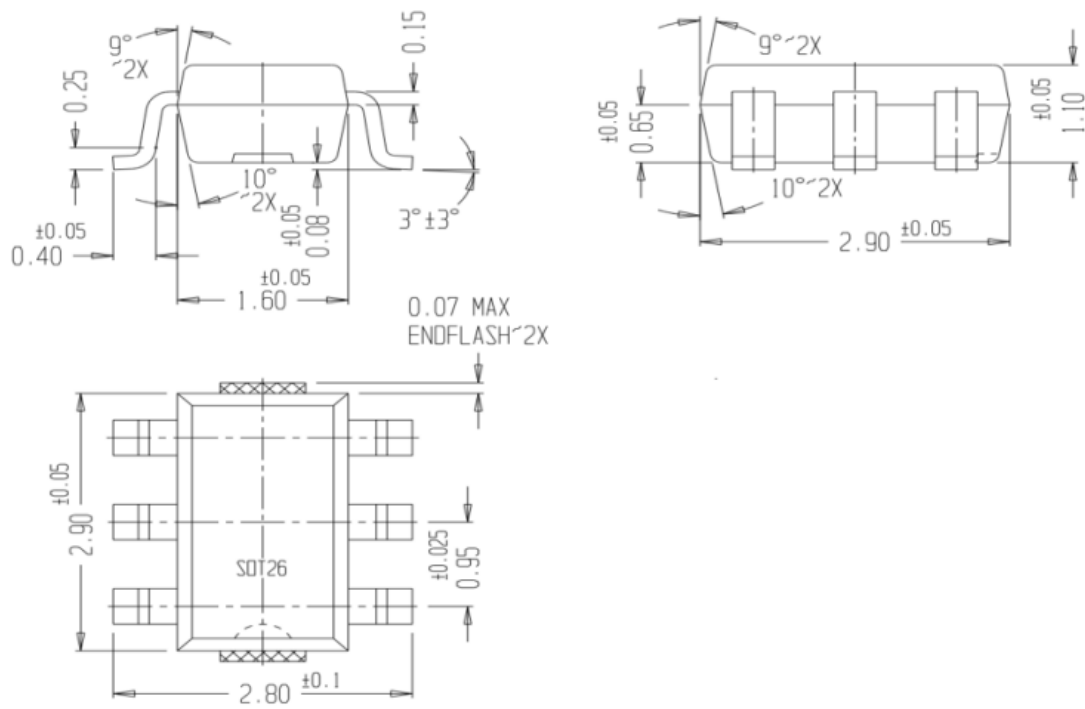
19.3 SO14



19.4 SOP8



19.5 SOT23-6



20 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

| | | | | |
|-------------------------------|------|------|----------|-------------|
| AD | 5 | 51 | 0 | X |
| Device code | Year | Week | SeriesNo | Internal No |
| Year: 1: 2021; 2 : 2022 | | | | |
| Week: 01:第1周; 23:第23周 | | | | |
| SeriesNo: 序列号0-Z, 1:当前周别第3个工单 | | | | |

标签信息

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

产品名称

Part No: AD18E211Q16

封装信息

Package: QFN16(3*3)

芯片批号

Lot No: NKK555000

丝印信息

Marking: AD55100

出货日期

Date: 2025-11-23

包装数量

QTY: *****

采购信息

| AD18E211 | | |
|-------------|--------------------|---------|
| 产品名称 | 封装信息 | 工作温度 |
| AD18E211Q16 | QFN16 (3*3) , 绿色封装 | -40~85℃ |
| AD18E211S16 | SOP16, 绿色封装 | -40~85℃ |
| AD18E211S14 | SOP14, 绿色封装 | -40~85℃ |
| AD18E211S8 | SOP8, 绿色封装 | -40~85℃ |
| AD18E211T6 | SOT23-6, 绿色封装 | -40~85℃ |

免责声明

本资料内容为西安恩狄集成电路有限公司(以下简称“恩狄”)版权所有。

恩狄将力求本资料的内容做到准确无误,但同时保留在不通知用户的情况下,对本资料内容的修改权。

如您需要获得最新的资料,请及时联系恩狄。

恩狄将尽最大努力为您提供高品质、高稳定性的产品。尽管如此,由于一般半导体器件的电气敏感性及易受到外部物理损伤等固有特性存在,所以难免造成半导体器件出现故障或失效的可能。当您使用恩狄产品时,有责任按照本资料以及相关关联资料中提到的“规则”来设计一个稳定及安全可靠的系统环境。另外,在遇到超规格(本资料中未描述到内容)的使用,请您提前咨询恩狄,以免因恩狄产品在一些特殊设备中或者特殊环境下的使用,导致财产损失、人员伤亡等严重后果的发生。因超规格的使用、未经咨询授权恩狄产品的使用,恩狄对此不承担任何法律责任。