



## CSU8RF3111/CSU8RF3112 用户手册

带 12-bit ADC 的 8 位 RISC FLASH MCU

Rev. 1.2.0

通讯地址：深圳市南山区蛇口南海大道 1079 号花园城数码大厦 A 座 9 楼  
邮政编码：518067  
公司电话：+(86 755)86169257  
传真：+(86 755)86169057  
公司网站：[www.chipsea.com](http://www.chipsea.com)

## 带 12-bit ADC 的 8 位 Flash 单片机产品介绍

### 主要特性

#### 高性能的 RISC CPU

- 8 位单片机 MCU
- 内置 1K×16 位程序存储器 E2PROM
- 96 字节数据存储器 (SRAM)
- 54 字节的 E2PROM, 用于数据存储
- 只有 43 条单字指令
- 6 级存储堆栈
- 支持 ISP

#### 振荡器

- 内带 16MHz 振荡器, 精度为 ±2%
- 外部 32768Hz 晶振 (RTC) 或 4MHz~16MHz 晶振
- ERC 3V DC~4M 5V DC~8M

#### 外设特性

- 11 位双向 I/O 口, 1 位输入口
- 2 路蜂鸣器输出
- 2 路 PWM 输出
- 4 个内部中断, 2 个外部中断
- 5 个具有唤醒功能的输入口
- 5+3 路 12-bit ADC
- 内部 1.4V、VDD、外部输入三种参考电压选择
- 带数字比较器
- 低电压检测 (LVD) 引脚, 内部提供 2.4V、3.6V 电压比较
- 3 个开漏输出口

#### 专用微控制器的特性

- 上电复位 (POR)
- 上电复位和硬件复位延迟定时器 (40ms)
- 内带低电压复位 (LVR)
- 定时器 0
  - 8 位可编程预分频的 8 位的定时计数器
- 定时/计数器 2
  - 8 位可编程预分频的 8 位的分频器
- 定时/计数器 3
  - 8 位可编程预分频的 8 位的分频器
- 扩展型看门狗定时器 (32K WDT)
  - 可编程的时间范围

#### CMOS 技术

- 电压工作范围
  - VDD 2.3V~5.5V

#### 低功耗特性

- MCU 工作电流
  - 正常模式 0.85mA@4MHz, 3V
  - 正常模式 8uA@32KHz, 3V
  - 休眠模式下的电流小于 1 μ A

#### 封装

- 8/14-pin SOP DIP

## 历史修改记录

| 历史版本.        | 修改内容                                                                                                               | 版本日期       |
|--------------|--------------------------------------------------------------------------------------------------------------------|------------|
| Rev. 1. 0. 0 | 1、换新 LOGO 初始版本                                                                                                     | 2012-12-24 |
| Rev. 1. 1. 0 | 1、更新存储器框图<br>2、增加 E2PROM 41AH 处所存放数据的说明<br>3、增加 1.4V 校准方法<br>4、删除原第 31、42、47、75 页的空白页<br>5、修改 MCU 指令集章节对 f 地址的错误定义 | 2013-03-14 |
| Rev. 1. 1. 1 | 1、更改 CPU 核的功能模块图                                                                                                   | 2013-04-11 |
| Rev. 1. 2. 0 | 1、更新 3.7 烧录模块的说明<br>增加 AD 校正信号线的描述                                                                                 | 2013-04-25 |

## 目 录

|                          |           |
|--------------------------|-----------|
| 历史修改记录.....              | 3         |
| 目 录.....                 | 4         |
| <b>1 PIN配置.....</b>      | <b>6</b>  |
| <b>2 标准功能.....</b>       | <b>8</b>  |
| 2.1 CPU核.....            | 8         |
| 2.1.1 存储器.....           | 10        |
| 2.1.2 状态寄存器.....         | 11        |
| 2.1.3 SFR.....           | 13        |
| 2.2 时钟系统.....            | 15        |
| 2.2.1 概述.....            | 15        |
| 2.2.2 时钟框图.....          | 15        |
| 2.2.3 寄存器.....           | 16        |
| 2.2.4 内部高速RC时钟.....      | 16        |
| 2.2.5 内部低速wdt时钟.....     | 17        |
| 2.2.6 外部高速晶振时钟.....      | 17        |
| 2.2.7 外部低速晶振时钟.....      | 17        |
| 2.2.8 外部RC振荡器.....       | 17        |
| 2.2.9 外部时钟源.....         | 17        |
| 2.3 复位系统.....            | 18        |
| 2.3.1 上电复位.....          | 19        |
| 2.3.2 看门狗复位.....         | 19        |
| 2.3.3 掉电复位.....          | 19        |
| 2.3.4 外部硬件复位.....        | 20        |
| 2.4 中断.....              | 21        |
| 2.4.1 中断使能寄存器.....       | 22        |
| 2.4.2 中断标志寄存器.....       | 23        |
| 2.4.3 外部中断 0.....        | 24        |
| 2.4.4 外部中断 1.....        | 24        |
| 2.4.5 AD中断溢出.....        | 25        |
| 2.4.6 定时器 0 溢出中断.....    | 25        |
| 2.4.7 定时/计数器 2 溢出中断..... | 25        |
| 2.4.8 定时/计数器 3 溢出中断..... | 25        |
| 2.4.9 PUSH和POP处理.....    | 25        |
| 2.5 定时器 0.....           | 27        |
| 2.6 I/O PORT.....        | 30        |
| 2.6.1 PT1 口.....         | 30        |
| 2.6.2 PT3 口.....         | 32        |
| 2.6.3 PT5 口.....         | 33        |
| <b>3 增强功能.....</b>       | <b>35</b> |
| 3.1 HALT和SLEEP模式.....    | 35        |
| 3.2 看门狗(WDT).....        | 37        |
| 3.3 定时/计数器 2.....        | 39        |
| 3.3.1 寄存器描述.....         | 39        |

|          |                                                             |           |
|----------|-------------------------------------------------------------|-----------|
| 3.3.2    | 蜂鸣器 .....                                                   | 41        |
| 3.3.3    | PWM .....                                                   | 41        |
| 3.4      | 定时/计数器 3 .....                                              | 43        |
| 3.4.1    | 寄存器描述 .....                                                 | 43        |
| 3.4.2    | 蜂鸣器 .....                                                   | 45        |
| 3.4.3    | PWM .....                                                   | 45        |
| 3.5      | 模数转换器 (ADC) .....                                           | 47        |
| 3.5.1    | 寄存器描述 .....                                                 | 47        |
| 3.5.2    | 转换时间 .....                                                  | 51        |
| 3.5.3    | 使用内部参考电压 1.40V 的校准方法 .....                                  | 53        |
| 3.5.4    | AD失调电压校正 .....                                              | 54        |
| 3.5.5    | 数字比较器 .....                                                 | 55        |
| 3.5.6    | 内部测量VDD的电压 .....                                            | 57        |
| 3.6      | 数据E2PROM .....                                              | 58        |
| 3.7      | 烧录模块 .....                                                  | 59        |
| 3.8      | 代码选项 .....                                                  | 60        |
| <b>4</b> | <b>MCU指令集 .....</b>                                         | <b>61</b> |
| <b>5</b> | <b>电气特性 .....</b>                                           | <b>80</b> |
| 5.1      | 极限值 .....                                                   | 80        |
| 5.2      | 直流特性 (VDD = 5V, T <sub>A</sub> = 25°C, 如无其他说明则都是此条件) .....  | 80        |
| 5.3      | ADC特性 (VDD = 5V, T <sub>A</sub> = 25°C, 如无其他说明则都是此条件) ..... | 82        |
| 5.4      | RC时钟频率特性 .....                                              | 83        |
| 5.5      | WDT时钟频率特性 .....                                             | 83        |
| 5.6      | ERC频率特性 .....                                               | 83        |
| 5.7      | 2.0V掉电复位温度特性 .....                                          | 85        |
| 5.8      | 2.4V低电压复位温度特性 .....                                         | 85        |
| 5.9      | 3.6V低电压复位温度特性 .....                                         | 86        |
| 5.10     | 1.4V内部参考电压温度特性 .....                                        | 86        |
| <b>6</b> | <b>封装图 .....</b>                                            | <b>87</b> |
| 6.1      | DIP-14PIN .....                                             | 87        |
| 6.2      | SOP-14PIN .....                                             | 88        |

1 PIN 配置

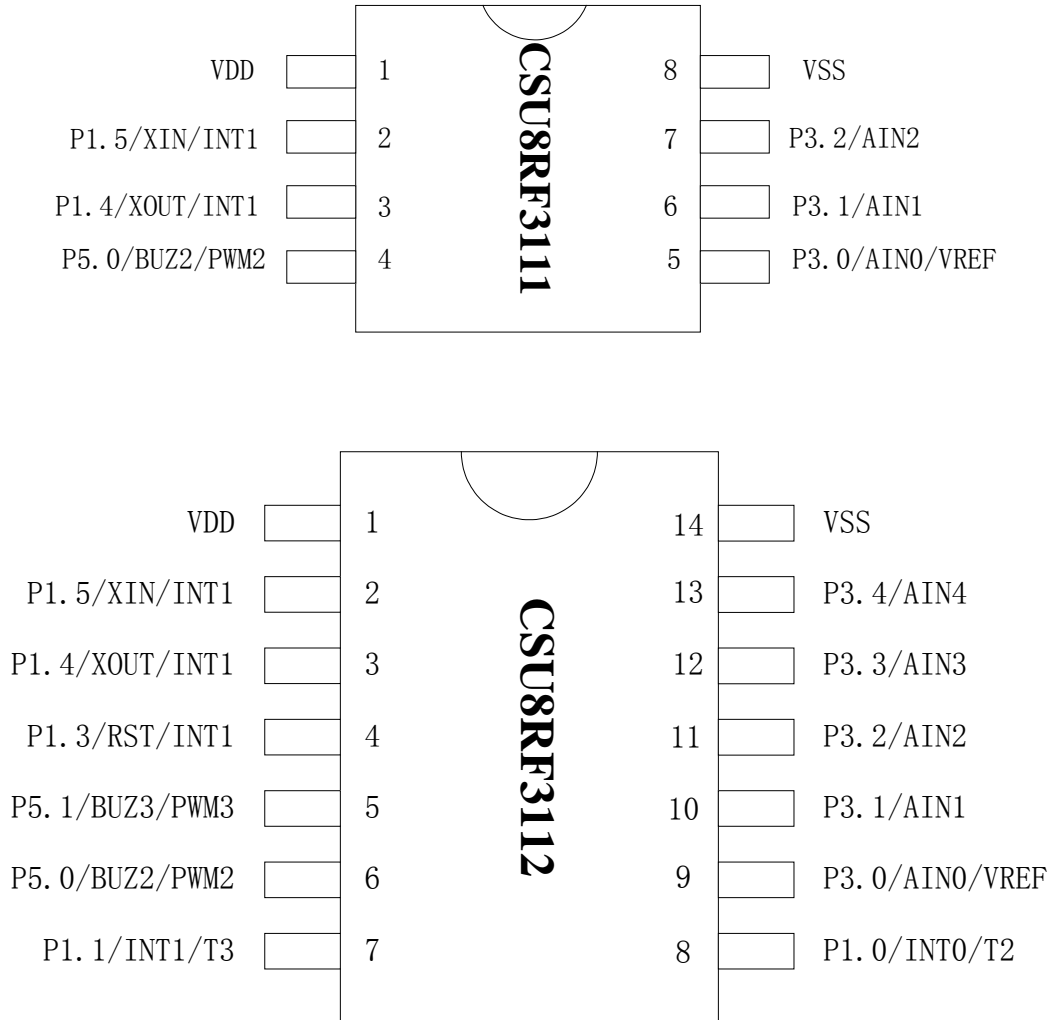


图1 8/14-pin 配置图

表 1 引脚说明表

| 管脚名称                         | 输入/输出 | 管脚序号 | 描述                                             |
|------------------------------|-------|------|------------------------------------------------|
| VDD                          | P     | 1    | 电源                                             |
| P1.5/XIN/INT1                | I/O   | 2    | IO; 外置晶振输入; 外部中断 1 输入, 具有唤醒功能                  |
| P1.4/XOUT/INT1               | I/O   | 3    | IO; 外置晶振输出; 外部中断 1 输入, 具有唤醒功能                  |
| P1.3/ $\overline{RST}$ /INT1 | I     | 4    | 普通输入口; 复位输入; 外部中断 1 输入, 具有唤醒功能                 |
| P5.1/BZ3/PWM3                | I/O   | 5    | IO; 蜂鸣器输出; PWM 输出; 具有开漏输出功能                    |
| P5.0/BZ2/PWM2                | I/O   | 6    | IO; 蜂鸣器输出; PWM 输出; 具有开漏输出功能                    |
| P1.1/INT1/T3                 | I/O   | 7    | IO; 外部中断 1 输入; 具有唤醒功能; 具有开漏输出功能; 定时/计数器 3 外部输入 |
| P1.0/INT0/T2                 | I/O   | 8    | IO; 外部中断 0 输入; 具有唤醒功能; 定时/计数器 2 外部输入           |
| P3.0/AIN0/VREF               | I/O   | 9    | IO; ADC 输入 0; ADC 参考电压输入                       |
| P3.1/AIN1                    | I/O   | 10   | IO; ADC 输入 1                                   |
| P3.2/AIN2                    | I/O   | 11   | IO; ADC 输入 2                                   |
| P3.3/AIN3                    | I/O   | 12   | IO; ADC 输入 3                                   |
| P3.4/AIN4                    | I/O   | 13   | IO; ADC 输入 4                                   |
| VSS                          | P     | 14   | 地                                              |

2 标准功能

2.1 CPU核

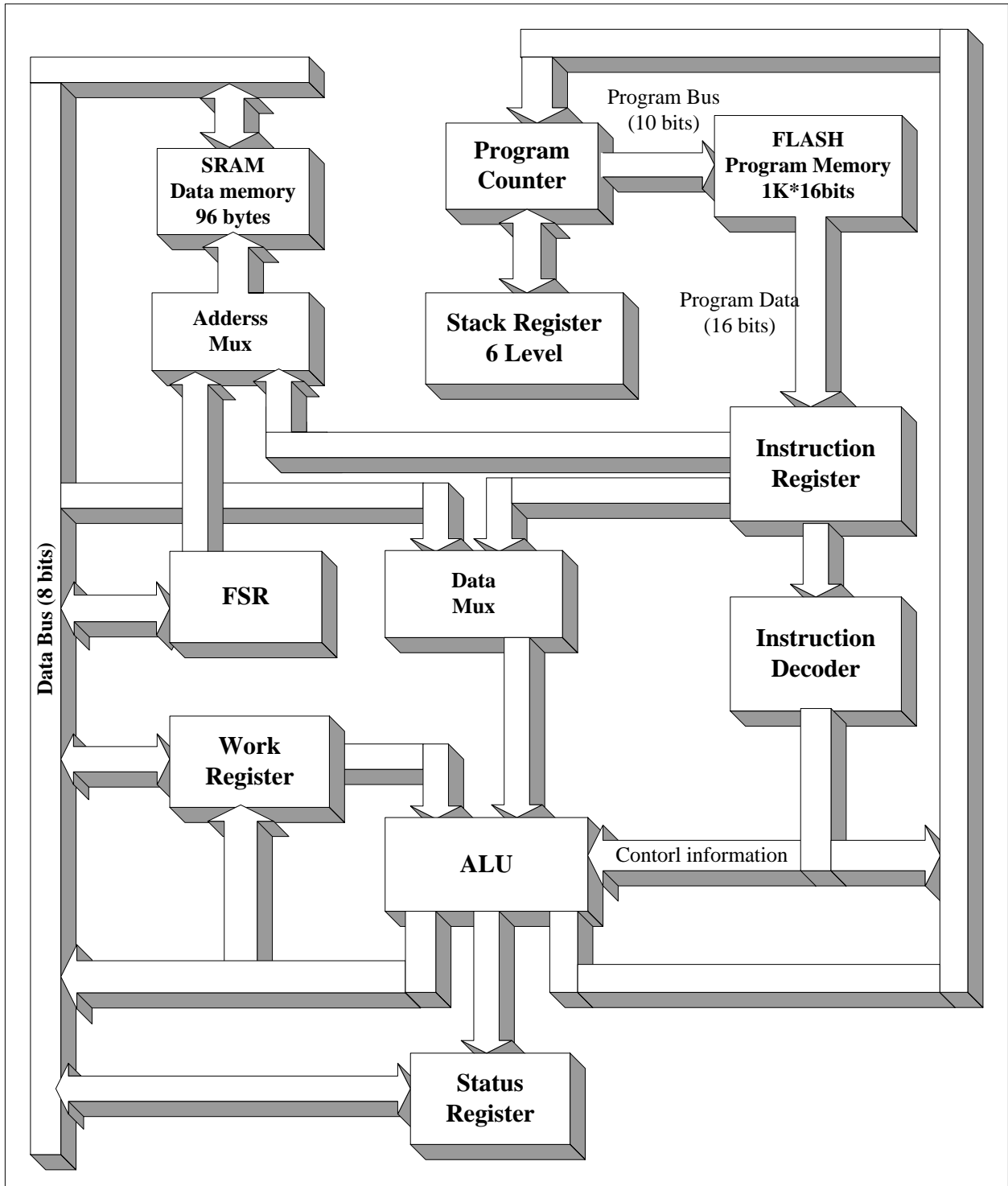


图2 CSU8RF3111/CSU8RF3112 CPU 核的功能模块图



从 CPU 核的功能模块图中，可以看到它主要包含 7 个主要寄存器及 2 个存储器单元。

表 2 MCU 架构说明

| 模块名称    | 描述                                                                                                                                                                                                                                                                                                                                                   |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 程序计数器   | 此寄存器在 CPU 的工作周期期间起到很重要的作用，它记录 CPU 每个周期处理程序存储器中指令的指针。在一个 CPU 周期中，程序计数器将程序存储器地址（10bits），指令指针推送到程序存储器，然后自动加 1 以进行下一次周期。                                                                                                                                                                                                                                 |
| 栈寄存器    | 堆栈寄存器是用来记录程序返回的指令指针。当程序调用函数，程序计数器会将指令指针推送到堆栈寄存器。在函数执行结束之后，堆栈寄存器会将指令指针送回到程序计数器以继续原来的程序处理。                                                                                                                                                                                                                                                             |
| 指令寄存器   | <p>程序计数器将指令指针（程序存储器地址）推送到程序存储器，程序存储器将程序存储器的数据（16bits）及指令推送到指令寄存器。</p> <p>CSU8RF3111/CSU8RF3112 的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。</p> <p>CPU 能将立即数推送到工作寄存器，或者进行某些处理后，根据控制信息，将立即数存储到直接地址所指向的数据存储器寄存器中。</p> <p>直接地址（8bits）<br/>数据寄存器的地址。CPU 能利用此地址来对数据存储器进行操作。</p> <p>直接数据（8bits）<br/>CPU 通过 ALU 利用此数据对工作寄存器进行操作。</p> <p>控制信息<br/>它记录着 ALU 的操作信息。</p> |
| 指令译码器   | 指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。                                                                                                                                                                                                                                                                                                       |
| 算术逻辑单元  | 算术逻辑单元不仅能完成 8 位二进制的加，减，加 1，减 1 等算术计算，还能对 8 位变量进行逻辑的与，或，异或，循环移位，求补，清零等逻辑运算。                                                                                                                                                                                                                                                                           |
| 工作寄存器   | 工作寄存器是用来缓存数据存储器中某些存储地址的数据。                                                                                                                                                                                                                                                                                                                           |
| 状态寄存器   | 当 CPU 利用 ALU 处理寄存器数据时，如下的状态会随着如下顺序变化：<br>PD, TO, DC, C 及 Z。                                                                                                                                                                                                                                                                                          |
| 文件选择寄存器 | 在 CSU8RF3111/CSU8RF3112 的指令集中，FSR 是用于间接数据处理（即实现间接寻址）。用户可以利用 FSR 来存放数据存储器中的某个寄存器地址，然后通过 IND 寄存器对这个寄存器进行处理。                                                                                                                                                                                                                                            |
| 程序存储器   | CSU8RF3111/CSU8RF3112 内带 1K×16 位的 FLASH 作为程序存储器。由于指令的操作码（OPCODE）是 16bits，用户最多只能编程 1K 的指令。程序存储器的地址总线是 10bits，数据总线是 16bits。                                                                                                                                                                                                                            |
| 数据存储器   | CSU8RF3111/CSU8RF3112 内带 96 bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 7bits，数据总线是 8bits。                                                                                                                                                                                                                                                                     |

2.1.1 存储器

(1) 程序存储器

程序存储器主要用于指令的存储，在 CSU8RF3111/CSU8RF3112 中，该程序存储器是 1K\*16bit 的程序 FLASH，对于程序员来说，该存储器只读，不可以写入。系统的 reset 地址为 000H，中断入口地址为 004H，需要注意的一点就是所有的中断共用同一个中断入口地址。数据 E2PROM 的地址范围为 400H~41AH。

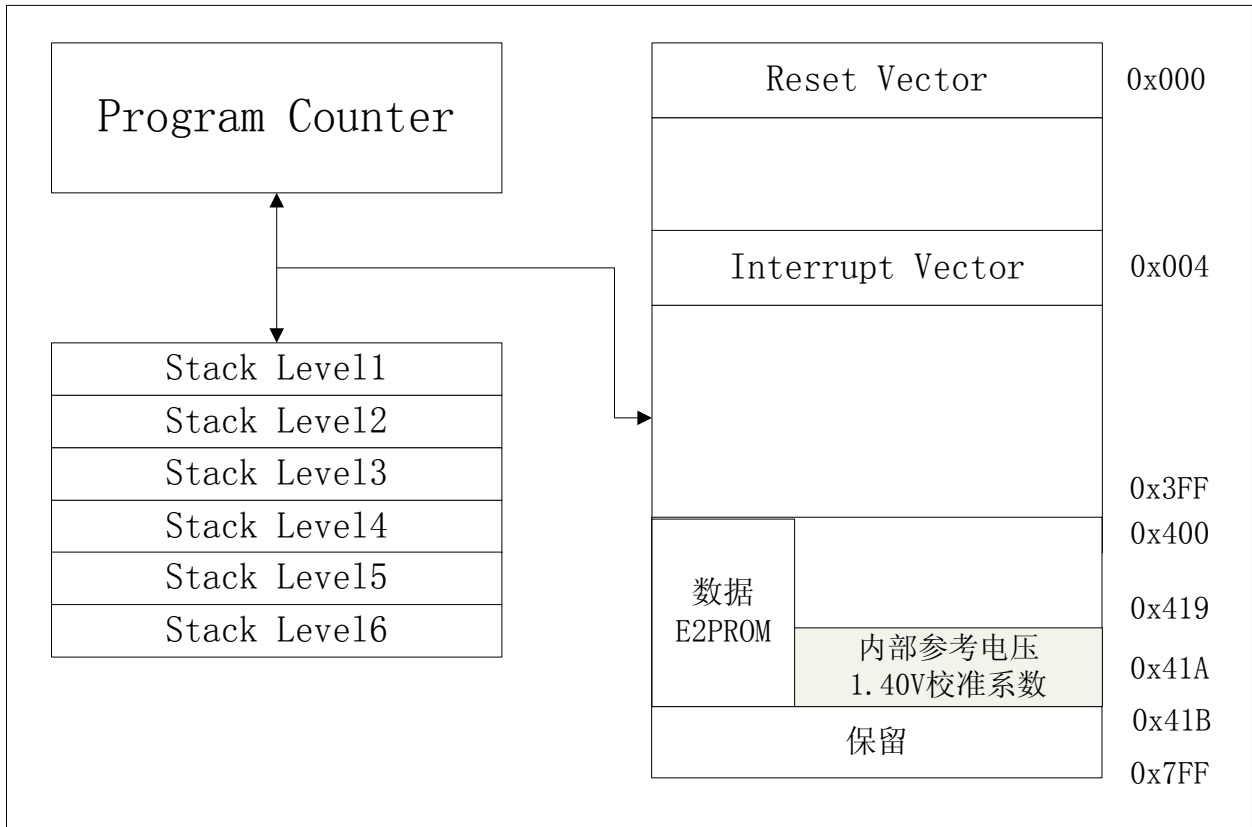


图3 程序存储器

(2) 数据存储器

数据存储器主要用于程序运行过程中，全局以及中间变量的存储。该存储器分为三个部分。地址的 00H 至 07H 是系统特殊功能寄存器，例如间接地址，间接地址指针，状态寄存器，工作寄存器，中断标志位，中断控制寄存器。地址的 08H 至 57H 外设特殊功能寄存器，例如 IO 端口，定时器，系统特殊功能寄存器和外设特殊功能寄存器是用寄存器实现，而通用数据存储器是 RAM 实现，可以读出也可以写入。

表 3 数据存储器地址分配

| 数据存储器     | 起始地址 | 结束地址 |
|-----------|------|------|
| 系统特殊功能寄存器 | 00H  | 07H  |
| 外设特殊功能寄存器 | 08H  | 57H  |
| 通用数据存储器   | 80H  | DFH  |

通过 **IND0** 以及 **FSR0** 这两个寄存器可以对数据存储器以及特殊功能寄存器进行间接访问。当从间接地址寄存器(**IND0**)读入数据时，MCU 实际上是以 **FSR0** 中的值作为地址去访问数据存储器得到数据。当向间接寄存器(**IND0**)写入数据时，MCU 实际上是以 **FSR0** 中的值作为地址去访问数据存储器将值存入该地址。其访问方式见。

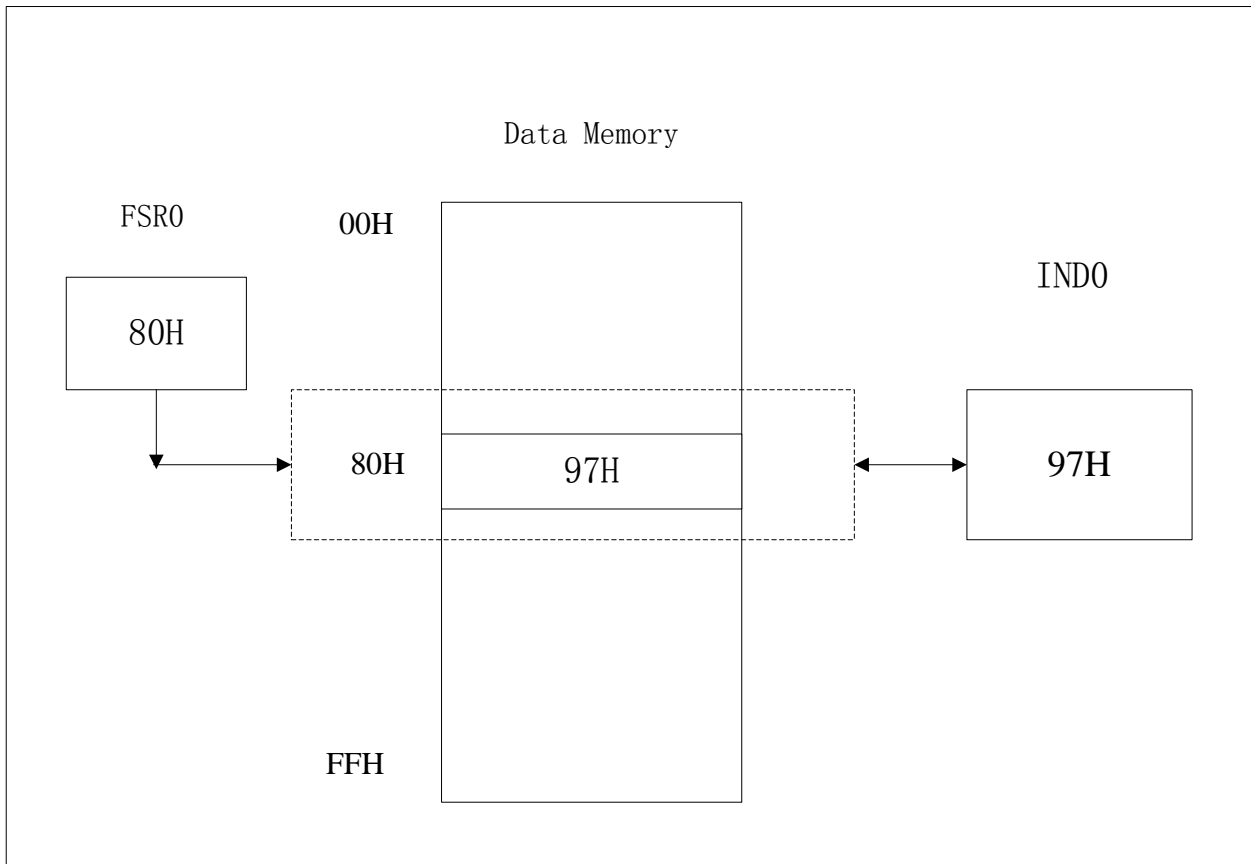


图4 间接地址访问

2.1.2 状态寄存器

状态寄存器包含 ALU 的算术状态及复位状态。状态寄存器类似于其它寄存器，可以作为任何指令的目标寄存器。如果状态寄存器是某条指令的目标寄存器，而且影响到 Z, DC 或 C 位，那么对这三个位的写是不使能。这些位是由器件逻辑进行置位或清零。T0 及 PD 位是不可写的。

**状态寄存器（地址为 04h）**

| 特性     | R-0   | R-0   | U-0  | R-0  | R-0  | R/W-0 | R/W-0 | R/W-0 |
|--------|-------|-------|------|------|------|-------|-------|-------|
| STATUS | LVD36 | LVD24 |      | PD   | T0   | DC    | C     | Z     |
|        | Bit7  | Bit6  | Bit5 | Bit4 | Bit3 | Bit2  | Bit1  | Bit0  |

Bit 7 LVD36: 3.6V LVD 工作电压标志，只有当代码选项 LVD\_SEL 为 2' b01 和 2' b10 有效

- 1: 系统工作电压低于 3.6V，说明低电压检测器已处于监控状态
- 0: 系统工作电压超过 3.6V，低电压检测器没有工作

Bit 6 LVD24: 2.4V LVD 工作电压标志，只有当代码选项 LVD\_SEL 为 2' b01 有效

- 1: 系统工作电压低于 2.4V，说明低电压检测器已处于监控状态
- 0: 系统工作电压超过 2.4V，低电压检测器没有工作

Bit 4 PD: 掉电标志位。通过对此位写 0 清零，sleep 后置此位

- 1: 执行 SLEEP 指令后
- 0: 上电复位后或硬件复位或 CLRWDT 指令之后

Bit 3 T0: 看门狗定时溢出标志。通过对此位写 0 清零，看门狗定时溢出设置此位

- 1: 看门狗定时溢出生
- 0: 上电复位后或硬件复位或 CLRWDT 指令后或 SLEEP 指令后

Bit 2 DC: 半字节进位标志/借位标志

- 用于借位时，极性相反
- 1: 结果的第 4 位出现进位溢出
- 0: 结果的第 4 位不出现进位溢出

Bit 1 C: 进位标志/借位标志

- 用于借位时，极性相反
- 1: 结果的最高位（MSB）出现进位溢出
- 0: 结果的最高位（MSB）不出现进位溢出

Bit 0 Z: 零标志

- 1: 算术或逻辑操作是结果为 0
- 0: 算术或逻辑操作是结果不为 0

**特性（Property）：**

R = 可读位                      W = 可写位                      U = 无效位  
 -n = 上电复位后的值      '1' = 位已设置              '0' = 位已清零              X = 不确定位

2.1.3 SFR

特殊功能寄存器（SFR）包含系统专用寄存器和辅助专用寄存器。

系统专用寄存器用于完成 CPU 核的功能，由间接地址，间接地址指针，状态寄存器，工作寄存器，中断标志及中断控制寄存器。

辅助专用寄存器是为辅助功能而设计，比如 I/O 口，定时器，信号的条件控制寄存器。

表 4 寄存器列表

| 地址  | 名称       | Bit7                     | Bit6        | Bit5          | Bit4   | Bit3        | Bit2       | Bit1         | Bit0    | 上电复位值     |
|-----|----------|--------------------------|-------------|---------------|--------|-------------|------------|--------------|---------|-----------|
| 00h | IND0     | 以 FSR0 中内容作为地址的数据存储器中的数据 |             |               |        |             |            |              |         | xxxxxxxx  |
| 02h | FSR0     | 间接数据存储器的地址指针 0           |             |               |        |             |            |              |         | xxxxxxxx  |
| 04h | STATUS   | LVD36                    | LVD24       |               | PD     | TO          | DC         | C            | Z       | xxu00xxx  |
| 05h | WORK     | 工作寄存器                    |             |               |        |             |            |              |         | xxxxxxxx  |
| 06h | INTF     |                          | TM2IF       |               | TM0IF  | SRADIF      |            | E1IF         | E0IF    | u0u0u0u0  |
| 07h | INTE     | GIE                      | TM2IE       |               | TM0IE  | SRADIE      |            | E1IE         | E0IE    | 00u0u0u0  |
| 0Ah | EADRH    |                          |             |               |        |             | PAR[10:8]  |              |         | uuuuuxxx  |
| 0Bh | EADRL    | PAR[7:0]                 |             |               |        |             |            |              |         | xxxxxxxx  |
| 0Ch | EDATH    | EDATH[7:0]               |             |               |        |             |            |              |         | xxxxxxxx  |
| 0Dh | WDTCON   | WDTEN                    |             |               |        |             | WTS[2:0]   |              |         | 0uuuuuxxx |
| 0Eh | WDTIN    | WDTIN[7:0]               |             |               |        |             |            |              |         | xxxxxxxx  |
| 0Fh | TM0CON   | T0EN                     | T0RATE[2:0] |               |        | T0RSTB      | T0SEL[1:0] |              |         | 0xxxx1xx  |
| 10h | TM0IN    | TM0IN[7:0]               |             |               |        |             |            |              |         | xxxxxxxx  |
| 11h | TM0CNT   | TM0CNT[7:0]              |             |               |        |             |            |              |         | 00000000  |
| 16h | MCK      | CST                      | CST_IN      | CST_WDT       | EO_SLP |             |            |              | CLKSEL  | 0010uuu0  |
| 17h | TM2CON   | T2EN                     | T2RATE[2:0] |               |        | T2CKS       | T2RSTB     | T2OUT        | PWM2OUT | 0xxxx1xx  |
| 18h | TM2IN    | TM2IN[7:0]               |             |               |        |             |            |              |         | xxxxxxxx  |
| 19h | TM2CNT   | TM2CNT[7:0]              |             |               |        |             |            |              |         | 00000000  |
| 1ah | TM2R     | TM2R[7:0]                |             |               |        |             |            |              |         | xxxxxxxx  |
| 1bh | TM3CON   | T3EN                     | T3RATE[2:0] |               |        | T3CKS       | T3RSTB     | T3OUT        | PWM3OUT | 0xxxx1xx  |
| 1ch | TM3IN    | TM3IN[7:0]               |             |               |        |             |            |              |         | xxxxxxxx  |
| 1dh | TM3CNT   | TM3CNT[7:0]              |             |               |        |             |            |              |         | 00000000  |
| 1eh | TM3R     | TM3R[7:0]                |             |               |        |             |            |              |         | xxxxxxxx  |
| 20h | PT1      |                          |             | PT1[5:3]      |        |             |            | PT1[1:0]     |         | uuxxxuxx  |
| 21h | PT1EN    |                          |             | PT1EN[5:3]    |        |             |            | PT1EN[1:0]   |         | uu000u00  |
| 22h | PT1PU    |                          |             | PT1PU[5:3]    |        |             |            | PT1PU[1:0]   |         | uu000u00  |
| 23h | PT1CON   | PT11OD                   | PT1W[3:0]   |               |        |             | E1M        | E0M[1:0]     |         | 00000000  |
| 28h | PT3      |                          |             |               |        | PT3[4:0]    |            |              |         | uuuxxxxx  |
| 29h | PT3EN    |                          |             |               |        | PT3EN[4:0]  |            |              |         | uuu00000  |
| 2ah | PT3PU    |                          |             |               |        | PT3PU[4:0]  |            |              |         | uuu00000  |
| 2bh | PT3CON   |                          |             |               |        | PT3CON[4:0] |            |              |         | uuu00000  |
| 30h | PT5      |                          |             |               |        |             |            | PT5[1:0]     |         | uuuuuuxx  |
| 31h | PT5EN    |                          |             |               |        |             |            | PT5EN[1:0]   |         | uuuuuu00  |
| 32h | PT5PU    |                          |             |               |        |             |            | PT5PU[1:0]   |         | uuuuuu00  |
| 33h | PT5CON   |                          |             |               |        |             |            | PT51OD       | PT50OD  | uuuuuu00  |
| 3ch | INTF2    |                          |             |               | TM3IF  |             |            |              |         | uuu0uuuu  |
| 3dh | INTE2    |                          |             |               | TM3IE  |             |            |              |         | uuu0uuuu  |
| 50h | SRADCON0 |                          |             | SRADACKS[1:0] |        |             |            | SRADCKS[1:0] |         | uuxxuuxx  |



|     |          |            |       |       |       |             |       |            |          |
|-----|----------|------------|-------|-------|-------|-------------|-------|------------|----------|
| 51h | SRADCON1 | SRADEN     | SRADS | OFTEN | CALIF | ENOV        | OFFEX | VREFS[1:0] | 00000000 |
| 52h | SRADCON2 | CHS[3:0]   |       |       |       |             |       |            | xxxxuuuu |
| 54h | SRADL    | SRAD[7:0]  |       |       |       |             |       |            | xxxxxxxx |
| 55h | SRADH    |            |       |       |       | SRAD[11:8]  |       |            | uuuuxxxx |
| 56h | SROFTL   | SROFT[7:0] |       |       |       |             |       |            | 00000000 |
| 57h | SROFTH   |            |       |       |       | SROFT[11:8] |       |            | uuuu0000 |

注：进行读操作时，无效位读出为 0

|                        |            |            |          |
|------------------------|------------|------------|----------|
| <b>特性 (Property) :</b> |            |            |          |
| R = 可读位                | W = 可写位    | U = 无效位    |          |
| -n = 上电复位后的值           | '1' = 位已设置 | '0' = 位已清零 | X = 不确定位 |

2.2 时钟系统

2.2.1 概述

芯片的时钟系统包括内置 16MHz 的 RC 振荡时钟 (IHRC)、外置高速晶振、内置低速 32KHz 的 WDT 时钟、外置低速的晶振时钟、外部 RC 时钟、外部时钟源。除去 WDT 时钟外，以上时钟都可以做为系统时钟源 Fosc。Fcpu 是 CPU 时钟频率。

普通模式 (高速时钟) :  $F_{cpu} = F_{osc} / N$ ,  $N = 4, 8, 16, 32$

低速模式 (低速时钟) :  $F_{cpu} = F_{osc} / N$ ,  $N = 4, 8, 16, 32$

2.2.2 时钟框图

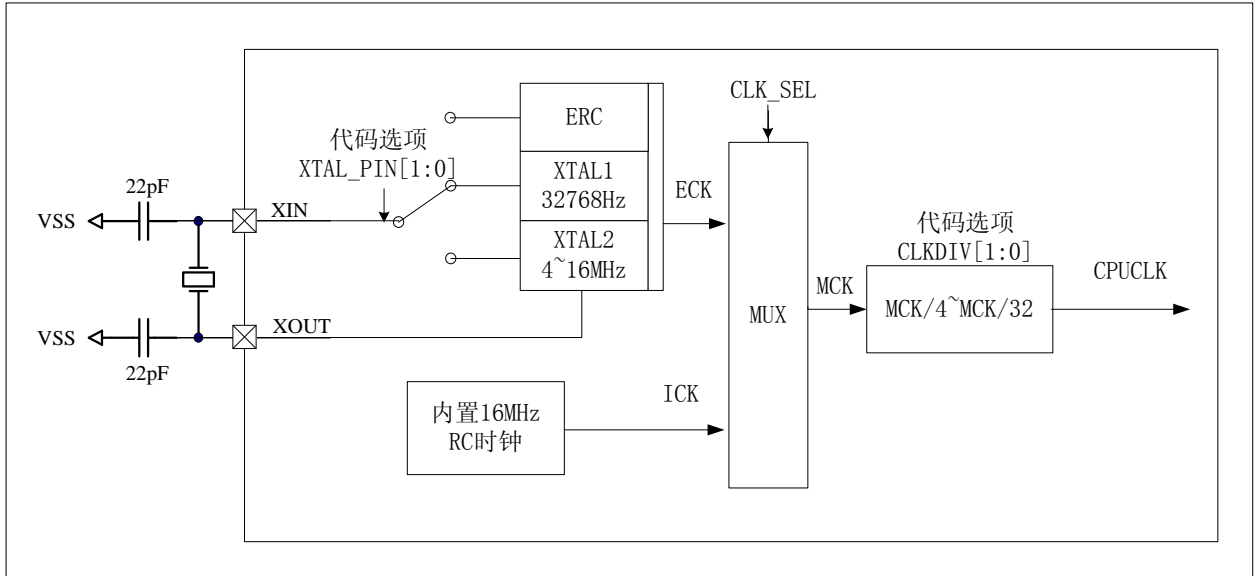


图5 CSU8RF3111/CSU8RF3112 振荡器状态框图 A

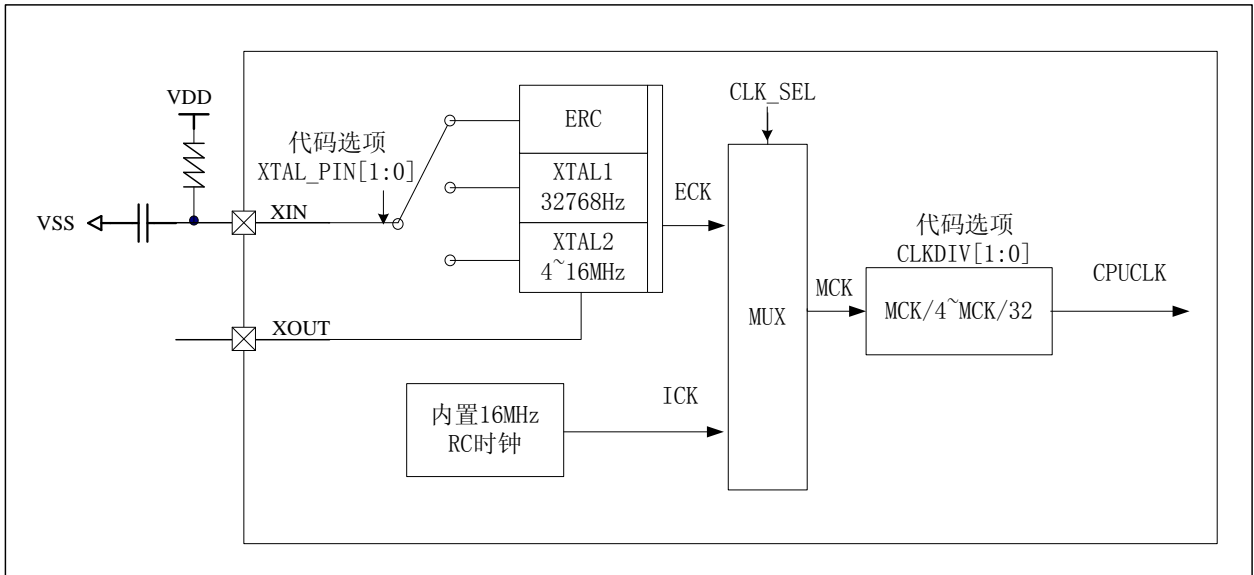


图6 CSU8RF3111/CSU8RF3112 振荡器状态框图 B

2.2.3 寄存器

表 5 CSU8RF3111/CSU8RF3112 时钟系统寄存器列表

| 地址  | 名称  | Bit7 | Bits6  | Bit5    | Bits4  | Bit3 | Bits2 | Bit1 | Bit0   | 上电复位值    |
|-----|-----|------|--------|---------|--------|------|-------|------|--------|----------|
| 16H | MCK | CST  | CST_IN | CST_WDT | EO_SLP |      |       |      | CLKSEL | 0010uuu0 |

表 6 MCK 寄存器各位功能表

| 位地址     | 标识符      | 功能                                                                                                                                                                                                                          |         |        |   |          |   |          |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|--------|---|----------|---|----------|
| 7       | CST      | 外部晶振启动开关<br>1: 外部晶振关闭<br>0: 外部晶振打开                                                                                                                                                                                          |         |        |   |          |   |          |
| 6       | CST_IN   | 内部晶振启动开关<br>1: 内部晶振关闭<br>0: 内部晶振打开                                                                                                                                                                                          |         |        |   |          |   |          |
| 5       | CST_WDT  | 内部 WDT 晶振启动开关<br>1: 内部 WDT 晶振关闭<br>0: 内部 WDT 晶振打开                                                                                                                                                                           |         |        |   |          |   |          |
| 4       | EO_SLP   | 外部低速晶振控制位<br>1: 如果选择的是外部低速晶振（32768Hz），在 sleep 模式下不关闭外部晶振<br>0: sleep 模式下关闭外部晶振                                                                                                                                              |         |        |   |          |   |          |
| 0       | CLKSEL   | 时钟源选择位 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CLK_SEL</th> <th>CPU 时钟</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>内部晶振系统时钟</td> </tr> <tr> <td>1</td> <td>外部晶振系统时钟</td> </tr> </tbody> </table> | CLK_SEL | CPU 时钟 | 0 | 内部晶振系统时钟 | 1 | 外部晶振系统时钟 |
| CLK_SEL | CPU 时钟   |                                                                                                                                                                                                                             |         |        |   |          |   |          |
| 0       | 内部晶振系统时钟 |                                                                                                                                                                                                                             |         |        |   |          |   |          |
| 1       | 外部晶振系统时钟 |                                                                                                                                                                                                                             |         |        |   |          |   |          |

对 MCK 寄存器进行写操作时，建议使用 bcf 或 bsf 指令。

注意：把 CPU 时钟由内部晶振切换到外部晶振，并把内部晶振关闭时应按照以下顺序执行

```

bcf mck, 7      ;打开外部晶振
call delay      ;低速 32768Hz 推荐延迟 2S；高速 16MHz 推荐延迟 15mS
bsf mck, 0      ;切换到外部晶振
nop
nop
bsf mck, 6      ;关闭内部晶振
...
    
```

2.2.4 内部高速RC时钟





内部高速 RC 时钟（16MHz），通过寄存器 CST\_IN 使能开关。当使用内部高速 RC 时钟做为系统的主时钟时，外部晶振引脚 PT1.4、PT1.5 可以通过代码选项配置做为普通的 GPIO 口。

### 2.2.5 内部低速wdt时钟

内部低速 wdt 时钟（32kHz），通过寄存器 CST\_WDT 使能开关。内部 wdt 时钟不能做为系统主时钟，只能做为 WDT 使用和定时器 0 使用。

### 2.2.6 外部高速晶振时钟

外部高速晶振时钟，通过代码选项配置为外部高速时钟，同时通过寄存器 CST 使能开关。此时，PT1.4、PT1.5 口做为晶振引脚。

### 2.2.7 外部低速晶振时钟

外部低速晶振时钟，通过代码选项配置为外部低速时钟，同时通过寄存器 CST 使能开关。此时，PT1.4、PT1.5 口做为晶振引脚。

### 2.2.8 外部RC振荡器

外部 RC 振荡器，通过代码选项配置为外部 RC 振荡器，同时通过寄存器 CST 使能开关。此时，PT1.5 口做为 RC 输入引脚，PT1.4 做为普通的 GPIO 口。外置 RC 振荡器的频率最高可以到 8MHz，最低可以到几 KHz，甚至更低。

### 2.2.9 外部时钟源

外部时钟源，通过代码选项配置为外部高速时钟或外部低速时钟或外部 RC 振荡器，同时通过寄存器 CST 使能开关。外部时钟源通过 PT1.5 口灌入时钟。当外部时钟源频率较快时，代码选项建议选择外部高速时钟或外部 RC 时钟；当外部时钟源频率较低时（与外部低速时钟频率相当），代码选项可以选择外部低速时钟或外部 RC 时钟。

2.3 复位系统

CSU8RF3111/CSU8RF3112 有以下方式复位：

- 1) 上电复位
- 2)  $\overline{RST}$  硬件复位（正常操作）
- 3)  $\overline{RST}$  硬件复位（从 Sleep 模式）
- 4) WDT 复位（正常操作）
- 5) WDT 复位（从 Sleep 模式）
- 6) 低电压复位（LVR）

上述任意一种复位发生时，所有系统寄存器恢复默认状态（WDT 复位 TO、PD 标志位除外），程序停止运行，同时程序计数器 PC 清零。复位结束后，系统从向量 000H 重新开始。各种复位情况下的 TO，PD 标志位如下表所示。

表 7 复位信号和状态寄存器关系

| 条件                                | TO | PD |
|-----------------------------------|----|----|
| 上电复位                              | 0  | 0  |
| $\overline{RST}$ 硬件复位（正常操作）       | 0  | 0  |
| $\overline{RST}$ 硬件复位（从 Sleep 模式） | 0  | 0  |
| WDT 复位（正常操作）                      | 1  | 不变 |
| WDT 复位（从 Sleep 模式）                | 1  | 不变 |
| 低电压复位                             | 0  | 0  |

下图给出了复位电路原理图。

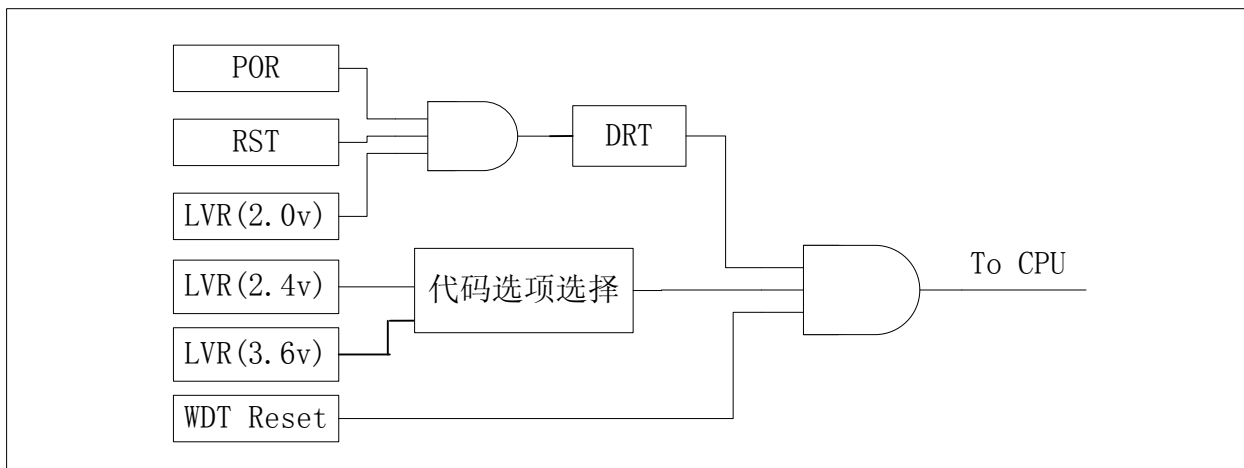


图7 复位电路原理图

任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。对于不同类型的振荡器起振的时间不同，所以完成复位的时间也有所不同。RC 振荡器起振时间最短，外置低速晶振起振时间最长。所以在有外部晶振电路应用的情况下，用户应在上电复位后，预留一定的时间再从内部 RC 时钟切换到外部晶振电路。用户在终端使用过程中，应注意考虑主机对上电复位的要求。

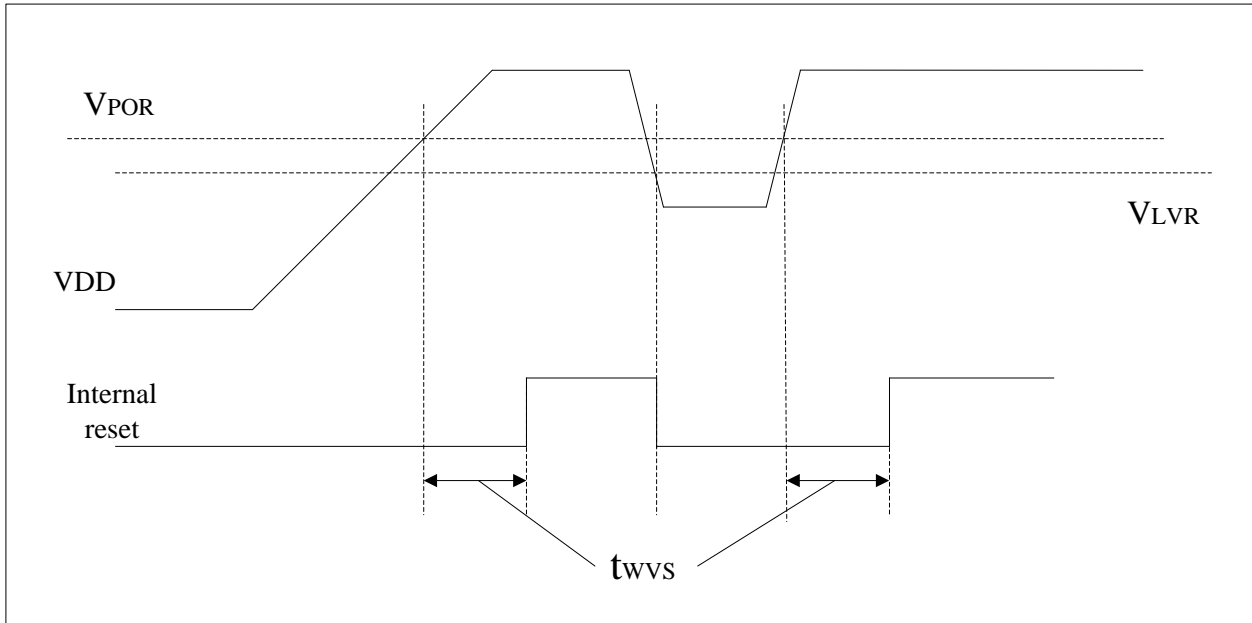


图8 上电复位电路示例及上电过程

| 参数                                         | 最小值  | 典型值  | 最大值  |
|--------------------------------------------|------|------|------|
| VPOR                                       | 1.8V | 2.0V | 2.2V |
| VLVR                                       | 1.8V | 2.0V | 2.2V |
| t <sub>wvs</sub><br>(测试条件: VDD=5V, T=25°C) | 32ms | 40ms | 48ms |

VPOR: 上电复位

VLVR: 低电压复位

t<sub>wvs</sub>: 等待电压稳定时间

### 2.3.1 上电复位

系统上电呈现逐渐上升的曲线形式，需要一定时间才能达到正常的工作电压（对于不同的指令周期所需工作电压是不同的，指令周期越快相应所需的工作电压就越高，见[5.2 直流特性](#)）。要求用户系统的上电速度要大于 0.15V/mS，尤其是要注意指令周期是 4MHz 时，因为他要求的工作电压最高。

### 2.3.2 看门狗复位

看门狗复位是一种系统的保护设置。在正常状态下，程序将看门狗定时器清零。如出错，系统处于未知状态，此时利用看门狗复位。看门狗复位后，系统重新进入正常状态。

### 2.3.3 掉电复位

掉电复位针对外部引起的系统电压跌落情况，例如受到干扰或者负载变化。系统掉电可能会引起系统工作状态不正常或者程序执行错误。

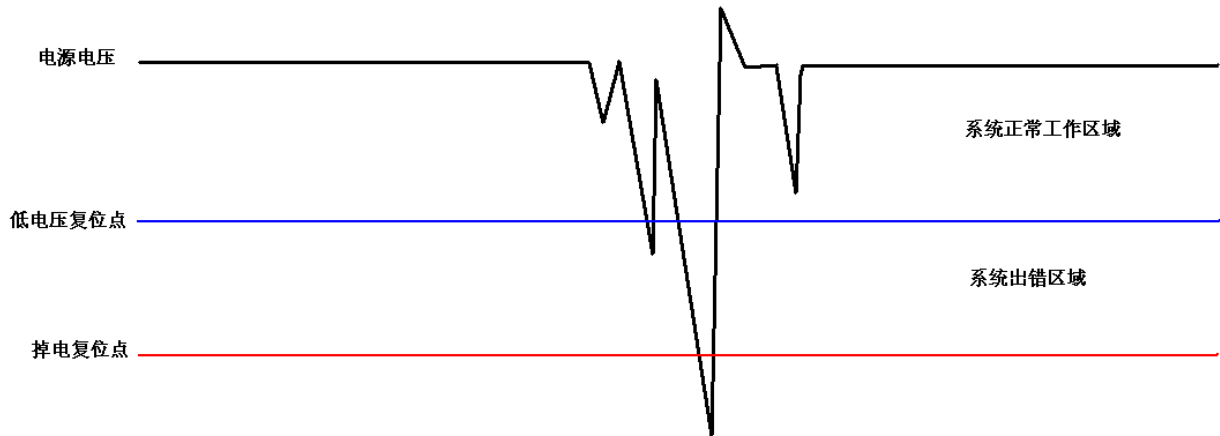


图9 系统掉电复位示意图

电压跌落可能会进入系统死区。进入系统死区，即电源电压不能满足系统的最小工作电压要求。系统掉电复位示意图如上图所示。芯片的掉电复位点在 2.0V，芯片的低电压复位点可以通过代码选项设置成 2.4V 或者 3.6V 或者不设置低电压复位点。

为避免进入系统死区，建议利用低电压复位（LVR）功能，尤其是指令周期是高速应用的情况。

不同指令周期的系统出错区域不同，取决于指令周期工作电压范围，[见 5.2](#)。如果指令周期是 4MHz 时，建议使用 3.6V 低电压复位。如果指令周期是 2Mhz 时，建议使用 2.4V 低电压复位。如果指令周期是 1MHz 或者 500KHz 时，可以不打开低电压复位，此时只有 2.0V 掉电复位。

掉电复位性能的改善可以通过如下几点实现：

- 1) 低电压复位（LVR）
- 2) 看门狗复位
- 3) 降低系统指令周期
- 4) 采用外部复位电路（稳压二极管复位电路；电压偏移复位电路；外部 IC 复位）

#### 2.3.4 外部硬件复位

外部复位由代码选项 RESET\_PIN 控制，[见 3.8](#)。通过设置该代码选项，可使能外部硬件复位功能。外部硬件复位引脚为施密特触发结构，低电平有效。硬件复位引脚为高电平时，系统正常工作；硬件复位引脚为低电平时，系统复位。

在芯片代码选项使能外部硬件复位功能后，需要注意的是：在系统上电完成后，外部复位需要输入高电平，否则，系统会一直复位，直到外部硬件复位结束。

外部硬件复位可以在上电过程中使用系统复位。良好的外部复位电路可以保护系统避免进入系统死区。

2.4 中断

CSU8RF3111/ CSU8RF3112 有 6 个中断源，只有 1 个中断入口地址 004H。与中断相关的 SFR：中断使能控制寄存器 INTE 和中断标志位寄存器 INTF。这 4 个中断源都各自有一个中断使能，和一个总使能位 GIE，并且它们的标志位硬件置位，软件清 0。

当响应中断时，会把当前的 PC 值入栈保护，并把 PC 置为 004H，同时把总使能位 GIE 清 0。执行完中断服务程序，并用 RETFIE 返回到之前的主程序，并把 GIE 置 1。

所有的中断都可以唤醒 sleep 睡眠模式和 halt 停止模式。

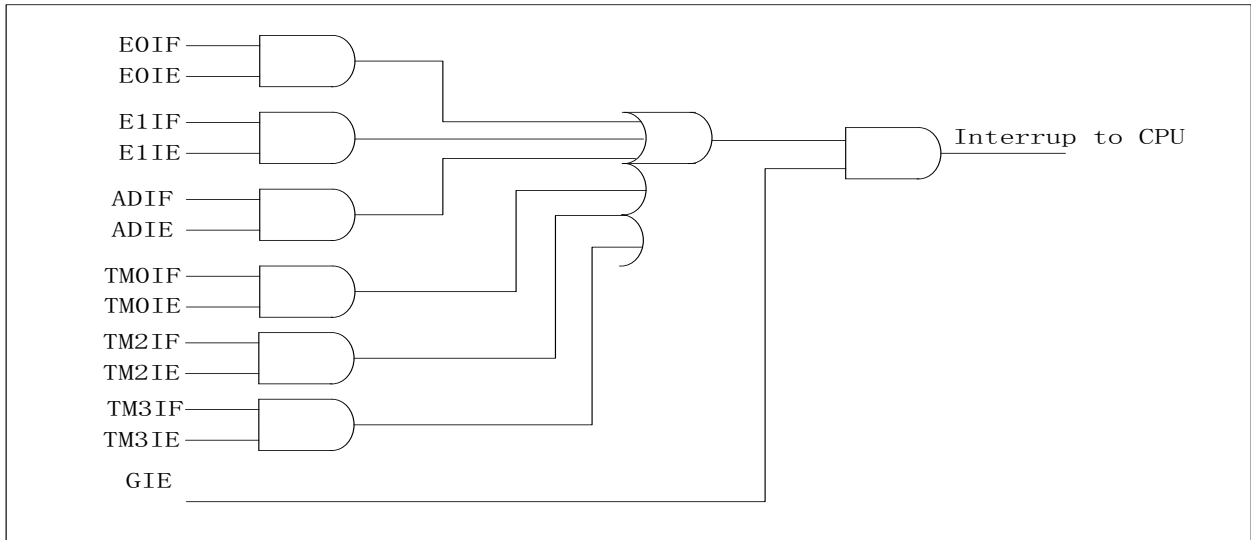


图10 中断逻辑

2.4.1 中断使能寄存器

**INTE 寄存器（地址为 07h）**

| 特性   | R/W-0 | R/W-0 | U-0  | R/W-0 | R/W-0  | U-0  | R/W-0 | R/W-0 |
|------|-------|-------|------|-------|--------|------|-------|-------|
| INTE | GIE   | TM2IE |      | TM0IE | SRADIE |      | E1IE  | E0IE  |
|      | Bit7  | Bit6  | Bit5 | Bit4  | Bit3   | Bit2 | Bit1  | Bit0  |

- Bit 7 GIE: 全局中断使能标志  
1 = 使能所有非屏蔽中断  
0 = 不使能所有中断
- Bit 6 TM2IE: 8-Bit 定时/计数器 2 中断使能标志  
1 = 使能定时/计数器 2 中断  
0 = 不使能定时/计数器 2 中断
- Bit 4 TM0IE: 8-Bit 定时 0 器中断使能标志  
1 = 使能定时器 0 中断  
0 = 不使能定时器 0 中断
- Bit 3 SRADIE: AD 中断使能标志  
1 = 使能 AD 中断  
0 = 不使能 AD 中断
- Bit 1 E1IE: 外部中断 1 使能标志  
1 = 使能外部中断 1  
0 = 不使能外部中断 1
- Bit 0 E0IE: 外部中断 0 使能标志  
1 = 使能外部中断 0  
0 = 不使能外部中断 0

**INTE2 寄存器（地址为 3dh）**

| 特性    | U-0  | U-0  | U-0  | R/W-0 | U-0  | U-0  | U-0  | U-0  |
|-------|------|------|------|-------|------|------|------|------|
| INTE2 |      |      |      | TM3IE |      |      |      |      |
|       | Bit7 | Bit6 | Bit5 | Bit4  | Bit3 | Bit2 | Bit1 | Bit0 |

- Bit 4 TM3IE: 8-Bit 定时/计数器 3 中断使能标志  
1 = 使能定时/计数器 3 中断  
0 = 不使能定时/计数器 3 中断

**特性 (Property) :**  
 R = 可读位                      W = 可写位                      U = 无效位  
 -n = 上电复位后的值      ‘1’ = 位已设置              ‘0’ = 位已清零              X = 不确定位

## 2.4.2 中断标志寄存器

中断标志位都是硬件置 1，软件清 0。某一个中断标志位只有在其对应的中断使能位为 1 的情况下，才有可能硬件置 1。

## INTF 寄存器（地址为 06h）

| 特性   | U-0  | R/W-0 | U-0  | R/W -0 | R/W -0 | U-0  | R/W -0 | R/W -0 |
|------|------|-------|------|--------|--------|------|--------|--------|
| INTF |      | TM2IF |      | TM0IF  | SRADIF |      | E1IF   | E0IF   |
|      | Bit7 | Bit6  | Bit5 | Bit4   | Bit3   | Bit2 | Bit1   | Bit0   |

Bit 6 TM2IF: 8-Bit 定时/计数器 2 中断标志，软件清零，硬件置高

1 = 发生定时中断，必须软件清 0

0 = 没发生定时中断

Bit 4 TM0IF: 8-Bit 定时器 0 中断标志，软件清零，硬件置高

1 = 发生定时中断，必须软件清 0

0 = 没发生定时中断

Bit 3 SRADIF: AD 中断标志，软件清零，硬件置高

1 = 发生 AD 中断，必须软件清 0

0 = 没发生 AD 中断

Bit 1 E1IF: 外部中断 1 中断标志，软件清零，硬件置高

1 = 外部中断 1 发生中断，必须软件清 0

0 = 外部中断 1 没发生中断

Bit 0 E0IF: 外部中断 0 中断标志，软件清零，硬件置高

1 = 外部中断 0 发生中断，必须软件清 0

0 = 外部中断 0 没发生中断

## INTF2 寄存器（地址为 3ch）

| 特性    | U-0  | U-0  | U-0  | R/W -0 | U-0  | U-0  | U-0  | U-0  |
|-------|------|------|------|--------|------|------|------|------|
| INTF2 |      |      |      | TM3IF  |      |      |      |      |
|       | Bit7 | Bit6 | Bit5 | Bit4   | Bit3 | Bit2 | Bit1 | Bit0 |

Bit 4 TM3IF: 8-Bit 定时/计数器 3 中断标志，软件清零，硬件置高

1 = 发生定时中断，必须软件清 0

0 = 没发生定时中断

## 特性 (Property) :

R = 可读位

W = 可写位

U = 无效位

-n = 上电复位后的值

'1' = 位已设置

'0' = 位已清零

X = 不确定位

2.4.3 外部中断 0

PT1.0 为外部中断 0 的输入端。触发方式由 PT1CON 寄存器中的 E0M[1:0]寄存器决定。INTE 寄存器中的 E0IE 为外部中断 0 的使能位，INTF 寄存器中的 E0IF 为中断标志位，硬件置 1，软件清 0。可唤醒 sleep 或 halt 模式。只有 E0IE 使能的情况下，PT1.0 被触发，中断标志位 E0IF 才会置 1。

PT1CON 寄存器（地址为 23h）

| 特性     | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0    | R/W-0 |
|--------|-------|-------|-------|-------|-------|-------|----------|-------|
| PT1CON |       |       |       |       |       | E1M   | E0M[1:0] |       |
|        | Bit7  | Bit6  | Bit5  | Bit4  | Bit3  | Bit2  | Bit1     | Bit0  |

Bit 2 E1M: 外部中断 1 触发模式

- 1 = 外部中断 1 为下降沿触发
- 0 = 外部中断 1 在状态改变时触发

Bit 1-0 E0M[1:0]: 外部中断 0 触发模式

- 11 = 外部中断 0 在状态改变时触发
- 10 = 外部中断 0 在状态改变时触发
- 01 = 外部中断 0 为上升沿触发
- 00 = 外部中断 0 为下降沿触发

2.4.4 外部中断 1

PT1.1、PT1.3、PT1.4 和 PT1.5 都可作为外部中断 1 的输入端。触发方式由 PT1CON 寄存器中的 E1M 寄存器决定。INTE 寄存器中的 E1IE 为外部中断 0 的使能位，INTF 寄存器中的 E1IF 为中断标志位，硬件置 1，软件清 0。只有 E1IE 使能的情况下，外部中断 1 被触发，中断标志位 E1IF 才会置 1。

PT1CON 寄存器（地址为 23h）

| 特性     | R/W-0 | R/W-0     | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|--------|-------|-----------|-------|-------|-------|-------|-------|-------|
| PT1CON |       | PT1W[3:0] |       |       |       |       |       |       |
|        | Bit7  | Bit6      | Bit5  | Bit4  | Bit3  | Bit2  | Bit1  | Bit0  |

Bit 6 PT1W[3]:PT1.5 外部中断 1 使能

- 0 = 禁止 PT1.5 外部中断 1
- 1 = 使能 PT1.5 外部中断 1

Bit 5 PT1W[2]:PT1.4 外部中断 1 使能

- 0 = 禁止 PT1.4 外部中断 1
- 1 = 使能 PT1.4 外部中断 1

Bit 4 PT1W[1]:PT1.3 外部中断 1 使能

- 0 = 禁止 PT1.3 外部中断 1
- 1 = 使能 PT1.3 外部中断 1

Bit 3 PT1W[0]:PT1.1 外部中断 1 使能

- 0 = 禁止 PT1.1 外部中断 1
- 1 = 使能 PT1.1 外部中断 1

|                        |            |            |          |
|------------------------|------------|------------|----------|
| <b>特性 (Property) :</b> |            |            |          |
| R = 可读位                | W = 可写位    | U = 无效位    |          |
| -n = 上电复位后的值           | '1' = 位已设置 | '0' = 位已清零 | X = 不确定位 |



#### 2.4.5 AD中断溢出

INTE 寄存器中的 SRADIE 为 ADC 中断的使能位，INTF 寄存器中的 SRADIF 为中断标志位，软件清 0。当 ADC 转换完成时，只有 SRADIE 使能的情况下，SRADIF 才能硬件置 1。

#### 2.4.6 定时器 0 溢出中断

INTE 寄存器中的 TM0IE 为定时器 0 中断的使能位，INTF 寄存器中的 TM0IF 为中断标志位，软件清 0。当定时器 0 溢出时，只有 TM0IE 使能的情况下，TM0IF 才能硬件置 1。

#### 2.4.7 定时/计数器 2 溢出中断

INTE 寄存器中的 TM2IE 为定时/计数器 2 中断的使能位，INTF 寄存器中的 TM2IF 为中断标志位，软件清 0。当定时/计数器 2 溢出时，只有 TM2IE 使能的情况下，TM2IF 才能硬件置 1。

#### 2.4.8 定时/计数器 3 溢出中断

INTE2 寄存器中的 TM3IE 为定时/计数器 3 中断的使能位，INTF2 寄存器中的 TM3IF 为中断标志位，软件清 0。当定时/计数器 3 溢出时，只有 TM3IE 使能的情况下，TM3IF 才能硬件置 1。

#### 2.4.9 PUSH和POP处理

CSU8RF3111/ CSU8RF3112 有 6 级的 PUSH 和 POP 堆栈。有中断请求被响应后，程序跳转到 004h 执行子程序。响应中断之前必须保存 WORK 和 STATUS 中的标志位(只保存 C, DC, Z)。芯片提供 PUSH 和 POP 指令进行入栈保存和出栈恢复，从而避免中断结束后程序运行错误。子程序中也可以使用 PUSH 和 POP 指令对 WORK 和 STATUS(C, DC, Z)进行保存和恢复。

```
...
org 004H
goto int_server
...
int_server:
    push
    btfsc intf,e0if    ;判断外部中断 0 标志
    goto ex0_int
    btfsc intf,elif    ;判断外部中断 1 标志
    goto ex1_int
    btfsc intf,tm0if   ;判断定时器 0 中断标志
    goto tm0_int
    btfsc intf,tm2if   ;判断定时/计数器 2 中断标志
    goto tm2_int
    btfsc intf,tm3if   ;判断定时/计数器 3 中断标志
    goto tm3_int
    ...
ex0_int:
    bcf intf, elif    ;清除 elif
    ...
    pop
    retfie
    ...
```

2.5 定时器 0

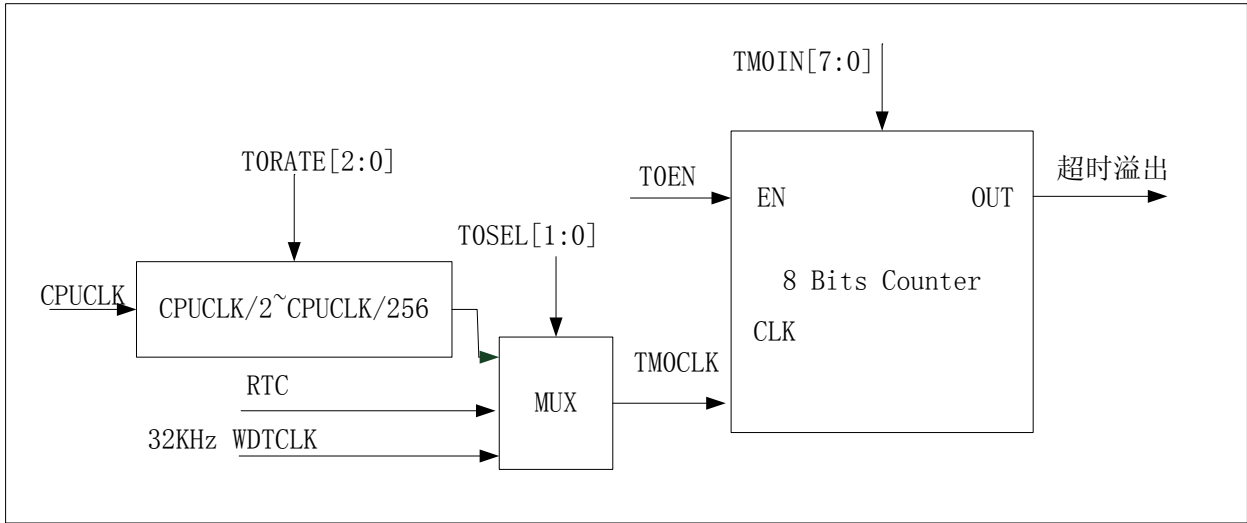


图11 定时器 0 功能框图

定时器 0 模块的输入为 CPUCLK。在定时器 0 模块集成了一个分频器，分频的时钟 TMOCLK 作为 8 bits 计数器的输入时钟。当用户设置了定时器 0 模块的使能标志，8 bits 计数器将启动，将会从 000H 递增到 TMOIN。用户需要设置 TMOIN（定时器 0 模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，中断标志位会自设置，程序计数器会跳转到 004H 以执行中断服务程序。

表 8 定时器 0 寄存器列表

| 地址  | 名称     | Bit7        | Bit6        | Bit5 | Bit4  | Bit3 | Bit2   | Bit1       | Bit0 | 上电复位值    |
|-----|--------|-------------|-------------|------|-------|------|--------|------------|------|----------|
| 06H | INTF   |             |             |      | TM0IF |      |        |            |      | uu00uu00 |
| 07H | INTE   | GIE         |             |      | TM0IE |      |        |            |      | 0u00uu00 |
| 0FH | TM0CON | TOEN        | TORATE[2:0] |      |       |      | TORSTB | T0SEL[1:0] |      | 0xxxu1xx |
| 10H | TM0IN  | TM0IN[7:0]  |             |      |       |      |        |            |      | xxxxxxxx |
| 11H | TM0CNT | TM0CNT[7:0] |             |      |       |      |        |            |      | 00000000 |

表 9 TM0CON 寄存器各位功能表

| 位地址          | 标识符                                           | 功能                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
|--------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-----------|-----|-----------|-----|-----------|-----|-----------------------------------------------|-----|------------------------------------|-----|------------|-----|------------|-----|-------------|-----|-------------|
| 7            | TOEN                                          | 定时器 0 使能位<br>1: 使能定时器 0<br>0: 禁止定时器 0                                                                                                                                                                                                                                                                                                                                                                                                                                                       |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 6:4          | TORATE[2:0]                                   | 定时器 0 时钟选择<br><table border="1"> <thead> <tr> <th>TORATE [2:0]</th> <th>TM0CLK</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>CPUCLK /2</td> </tr> <tr> <td>001</td> <td>CPUCLK /4</td> </tr> <tr> <td>010</td> <td>CPUCLK /8</td> </tr> <tr> <td>011</td> <td>CPUCLK /16</td> </tr> <tr> <td>100</td> <td>CPUCLK /32</td> </tr> <tr> <td>101</td> <td>CPUCLK /64</td> </tr> <tr> <td>110</td> <td>CPUCLK /128</td> </tr> <tr> <td>111</td> <td>CPUCLK /256</td> </tr> </tbody> </table> | TORATE [2:0] | TM0CLK    | 000 | CPUCLK /2 | 001 | CPUCLK /4 | 010 | CPUCLK /8                                     | 011 | CPUCLK /16                         | 100 | CPUCLK /32 | 101 | CPUCLK /64 | 110 | CPUCLK /128 | 111 | CPUCLK /256 |
| TORATE [2:0] | TM0CLK                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 000          | CPUCLK /2                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 001          | CPUCLK /4                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 010          | CPUCLK /8                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 011          | CPUCLK /16                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 100          | CPUCLK /32                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 101          | CPUCLK /64                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 110          | CPUCLK /128                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 111          | CPUCLK /256                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 2            | TORSTB                                        | 定时器 0 复位<br>1: 禁止定时器 0 复位<br>0: 使能定时器 0 复位<br>当将该位为 0 时，定时器 0 复位后，TORSTB 会自动置 1                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 1:0          | TOSEL[1:0]                                    | 时钟源选择<br><table border="1"> <thead> <tr> <th>TOSEL[1:0]</th> <th>定时器 0 时钟源</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>TM0CLK</td> </tr> <tr> <td>01</td> <td>TM0CLK</td> </tr> <tr> <td>10</td> <td>外部 32768Hz 晶振时钟，<br/>仅当外部接 32768Hz 晶振，且晶振打开时有效</td> </tr> <tr> <td>11</td> <td>内部 32K WDT 时钟，<br/>仅当内部 WDT 晶振打开时有效</td> </tr> </tbody> </table>                                                                                                                                   | TOSEL[1:0]   | 定时器 0 时钟源 | 00  | TM0CLK    | 01  | TM0CLK    | 10  | 外部 32768Hz 晶振时钟，<br>仅当外部接 32768Hz 晶振，且晶振打开时有效 | 11  | 内部 32K WDT 时钟，<br>仅当内部 WDT 晶振打开时有效 |     |            |     |            |     |             |     |             |
| TOSEL[1:0]   | 定时器 0 时钟源                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 00           | TM0CLK                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 01           | TM0CLK                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 10           | 外部 32768Hz 晶振时钟，<br>仅当外部接 32768Hz 晶振，且晶振打开时有效 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |
| 11           | 内部 32K WDT 时钟，<br>仅当内部 WDT 晶振打开时有效            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |              |           |     |           |     |           |     |                                               |     |                                    |     |            |     |            |     |             |     |             |

表 10 TM0IN 寄存器各位功能表

| 位地址  | 标识符        | 功能                   |
|------|------------|----------------------|
| 7: 0 | TM0IN[7:0] | 定时器 0 溢出值（溢出值：1~255） |

表 11 TM0CNT 寄存器各位功能表

| 位地址  | 标识符         | 功能             |
|------|-------------|----------------|
| 7: 0 | TM0CNT[7:0] | 定时器 0 计数寄存器，只读 |

操作：

- 1) 设置 TM0CLK，为定时器 0 模块选择输入。
- 2) 设置 TM0IN，选择定时器 0 溢出值。（溢出值：1~255）
- 3) 设置寄存器标志位：TMOIE 与 GIE，使能定时器 0 中断。
- 4) 清零寄存器标志位：TORSTB，复位定时器 0 模块的计数器。
- 5) 设置寄存器标志位：TMOEN，使能定时器 0 模块的 8 bits 计数器。
- 6) 当定时超时发生时，程序计数器会跳转到 004H。

定时器 0 溢出时间计算方法：

定时器 0 溢出时间= (TM0IN+1) /TM0CLK.

2.6 I/O PORT

表 12 I/O 口寄存器表

| 地址  | 名称     | Bit7   | Bit6      | Bit5       | Bit4 | Bit3        | Bit2   | Bit1       | Bit0 | 上电复位值    |
|-----|--------|--------|-----------|------------|------|-------------|--------|------------|------|----------|
| 20h | PT1    |        |           | PT1[5:3]   |      |             |        | PT1[1:0]   |      | uuxxxu00 |
| 21h | PT1EN  |        |           | PT1EN[5:3] |      |             |        | PT1EN[1:0] |      | uu000u00 |
| 22h | PT1PU  |        |           | PT1PU[5:3] |      |             |        | PT1PU[1:0] |      | uu000u00 |
| 23h | PT1CON | PT11OD | PT1W[3:0] |            |      |             | E1M    | E0M[1:0]   |      | 00000000 |
| 28h | PT3    |        |           |            |      | PT3[4:0]    |        |            |      | uuu00000 |
| 29h | PT3EN  |        |           |            |      | PT3EN[4:0]  |        |            |      | uuu00000 |
| 2ah | PT3PU  |        |           |            |      | PT3PU[4:0]  |        |            |      | uuu00000 |
| 2bh | PT3CON |        |           |            |      | PT3CON[4:0] |        |            |      | uuu00000 |
| 30h | PT5    |        |           |            |      |             |        | PT5[1:0]   |      | uuuuuu00 |
| 31h | PT5EN  |        |           |            |      |             |        | PT5EN[1:0] |      | uuuuuu00 |
| 32h | PT5PU  |        |           |            |      |             |        | PT5PU[1:0] |      | uuuuuu00 |
| 33h | PT5CON |        |           |            |      |             | PT51OD | PT50OD     |      | uuuuuu00 |

微控制器中的通用 I/O 口（GPIO）用于通用的输入与输出功能。用户可以通过 GPIO 接收数据信号或将数据传送给其它的数字设备。CSU8RF3111/CSU8RF3112 的部分 GPIO 可以被定义为其它的特殊功能。在本节，只说明 GPIO 的通用 I/O 口功能，特殊功能将会在接下来的章节中说明。

2.6.1 PT1 口

PT1 寄存器（地址为 20h）

| 特性  | U-0  | U-0  | R/W-X    | R/W-X | R/W-0 | U-0  | R/W-0    | R/W-0 |  |
|-----|------|------|----------|-------|-------|------|----------|-------|--|
| PT1 |      |      | PT1[5:3] |       |       |      | PT1[1:0] |       |  |
|     | Bit7 | Bit6 | Bit5     | Bit4  | Bit3  | Bit2 | Bit1     | Bit0  |  |

- Bit 5-0 PT1[5:0]: GPIO1 口数据标志
- PT1[5] = GPIO1 bit 5 数据标志位
- PT1[4] = GPIO1 bit 4 数据标志位
- PT1[3] = GPIO1 bit 3 数据标志位
- PT1[1] = GPIO1 bit 1 数据标志位
- PT1[0] = GPIO1 bit 0 数据标志位

PT1EN 寄存器（地址为 21h）

| 特性    | U-0  | U-0  | R/W-0      | R/W-0 | R-0  | U-0  | R/W-0      | R/W-0 |  |
|-------|------|------|------------|-------|------|------|------------|-------|--|
| PT1EN |      |      | PT1EN[5:3] |       |      |      | PT1EN[1:0] |       |  |
|       | Bit7 | Bit6 | Bit5       | Bit4  | Bit3 | Bit2 | Bit1       | Bit0  |  |

- Bit 5-0 PT1EN[5:0]: GPIO1 口输入/输出控制标志
- PT1EN[5] = GPIO1 bit 5 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口
- PT1EN[4] = GPIO1 bit 4 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口
- PT1EN[3] = GPIO1 bit 3 的 I/O 控制标志位；0 = 定义为输入口，只能为输入口，只读**
- PT1EN[1] = GPIO1 bit 1 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口
- PT1EN[0] = GPIO1 bit 0 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口

特性 (Property) :

R = 可读位            W = 可写位            U = 无效位  
 -n = 上电复位后的值    '1' = 位已设置    '0' = 位已清零            X = 不确定位



PT1PU 寄存器（地址为 22h）

| 特性    | U-0  | U-0  | R/W-0      | R/W-0 | R/W-0 | U-0  | R/W-0      | R/W-0 |
|-------|------|------|------------|-------|-------|------|------------|-------|
| PT1PU |      |      | PT1PU[5:3] |       |       |      | PT1PU[1:0] |       |
|       | Bit7 | Bit6 | Bit5       | Bit4  | Bit3  | Bit2 | Bit1       | Bit0  |

Bit 5-0 PT1PU[5:0]: GPIO1 口上拉电阻使能标志

PT1PU[5] = GPIO1 bit 5 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT1PU[4] = GPIO1 bit 4 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT1PU[3] = GPIO1 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT1PU[1] = GPIO1 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT1PU[0] = GPIO1 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT1CON 寄存器（地址为 23h）

| 特性     | R/W-0  | R/W-0     | R/W-0 | R/W-0 | R/W-0 | R/W-0    | R/W-0 | R/W-0 |
|--------|--------|-----------|-------|-------|-------|----------|-------|-------|
| PT1CON | PT1IOD | PT1W[3:0] |       |       | E1M   | E0M[1:0] |       |       |
|        | Bit7   | Bit6      | Bit5  | Bit4  | Bit3  | Bit2     | Bit1  | Bit0  |

Bit 7 PT1IOD: PT1.1 漏极开路使能位

0 = 禁止 PT1.1 漏极开路

1 = 使能 PT1.1 漏极开路

Bit 6 PT1W[3]:PT1.5 外部中断 1 使能

0 = 禁止 PT1.5 外部中断 1

1 = 使能 PT1.5 外部中断 1

Bit 5 PT1W[2]:PT1.4 外部中断 1 使能

0 = 禁止 PT1.4 外部中断 1

1 = 使能 PT1.4 外部中断 1

Bit 4 PT1W[1]:PT1.3 外部中断 1 使能

0 = 禁止 PT1.3 外部中断 1

1 = 使能 PT1.3 外部中断 1

Bit 3 PT1W[0]:PT1.1 外部中断 1 使能

0 = 禁止 PT1.1 外部中断 1

1 = 使能 PT1.1 外部中断 1

Bit 2 E1M: 外部中断 1 触发模式

1 = 外部中断 1 为下降沿触发

0 = 外部中断 1 在状态改变时触发

Bit 1-0 E0M[1:0]: 外部中断 0 触发模式

11 = 外部中断 0 在状态改变时触发

10 = 外部中断 0 在状态改变时触发

01 = 外部中断 0 为上升沿触发

00 = 外部中断 0 为下降沿触发

**特性 (Property) :**

R = 可读位

W = 可写位

U = 无效位

-n = 上电复位后的值 ‘1’ = 位已设置 ‘0’ = 位已清零

X = 不确定位

**2.6.2 PT3 口**

PT3 寄存器（地址为 28h）

| 特性  | U-0  | U-0  | U-0  | R/W-X    | R/W-X | R/W-X | R/W-X | R/W-X |  |
|-----|------|------|------|----------|-------|-------|-------|-------|--|
| PT3 |      |      |      | PT3[4:0] |       |       |       |       |  |
|     | Bit7 | Bit6 | Bit5 | Bit4     | Bit3  | Bit2  | Bit1  | Bit0  |  |

- Bit 4-0 PT3[4:0]: GPIO3 口数据标志位
- PT3[4] = GPIO3 bit 4 的数据标志位
- PT3[3] = GPIO3 bit 3 的数据标志位
- PT3[2] = GPIO3 bit 2 的数据标志位
- PT3[1] = GPIO3 bit 1 的数据标志位
- PT3[0] = GPIO3 bit 0 的数据标志位

PT3EN 寄存器（地址为 29h）

| 特性    | U-0  | U-0  | U-0  | R/W-0      | R/W-0 | R/W-0 | R/W-0 | R/W-0 |  |
|-------|------|------|------|------------|-------|-------|-------|-------|--|
| PT3EN |      |      |      | PT3EN[4:0] |       |       |       |       |  |
|       | Bit7 | Bit6 | Bit5 | Bit4       | Bit3  | Bit2  | Bit1  | Bit0  |  |

- Bit 4-0 PT3EN[4:0]: GPIO 3 口输入/输出控制标志
- PT3EN[4] = GPIO3 bit 4 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT3EN[3] = GPIO3 bit 3 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT3EN[2] = GPIO3 bit 2 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT3EN[1] = GPIO3 bit 1 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT3EN[0] = GPIO3 bit 0 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT3PU 寄存器（地址为 2ah）

| 特性    | U-0  | U-0  | U-0  | R/W-0      | R/W-0 | R/W-0 | R/W-0 | R/W-0 |  |
|-------|------|------|------|------------|-------|-------|-------|-------|--|
| PT2PU |      |      |      | PT3PU[4:0] |       |       |       |       |  |
|       | Bit7 | Bit6 | Bit5 | Bit4       | Bit3  | Bit2  | Bit1  | Bit0  |  |

- Bit 4-0 PT3PU[4:0]: GPIO3 口上拉电阻使能标志
- PT3PU[4] = GPIO3 bit 4 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT3PU[3] = GPIO3 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT3PU[2] = GPIO3 bit 2 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT3PU[1] = GPIO3 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT3PU[0] = GPIO3 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

**特性 (Property) :**  
 R = 可读位      W = 可写位      U = 无效位  
 -n = 上电复位后的值    ‘1’ = 位已设置    ‘0’ = 位已清零      X = 不定位



PT3CON 寄存器（地址为 2bh）

| 特性     | U-0  | U-0  | U-0  | R/W-0       | R/W-0 | R/W-0 | R/W-0 | R/W-0 |  |
|--------|------|------|------|-------------|-------|-------|-------|-------|--|
| PT3CON |      |      |      | PT3CON[4:0] |       |       |       |       |  |
|        | Bit7 | Bit6 | Bit5 | Bit4        | Bit3  | Bit2  | Bit1  | Bit0  |  |

Bit 4-0 PT3CON[4:0]: GPIO3 口模拟/数字端口使能标志

PT3CON[4] = GPIO3bit 4 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT3CON[3] = GPIO3bit 3 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT3CON[2] = GPIO3bit 2 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT3CON[1] = GPIO3bit 1 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT3CON[0] = GPIO3bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

### 2.6.3 PT5 口

PT5 寄存器（地址为 30h）

| 特性  | U-0  | U-0  | U-0  | U-0  | U-0  | U-0  | R/W-X    | R/W-X |
|-----|------|------|------|------|------|------|----------|-------|
| PT5 |      |      |      |      |      |      | PT5[1:0] |       |
|     | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1     | Bit0  |

Bit 1-0 PT5[1:0]: GPIO5 口数据标志位

PT5[1] = GPIO5 bit 1 的数据标志位

PT5[0] = GPIO5 bit 0 的数据标志位

PT5EN 寄存器（地址为 31h）

| 特性    | U-0  | U-0  | U-0  | U-0  | U-0  | U-0  | R/W-0      | R/W-0 |
|-------|------|------|------|------|------|------|------------|-------|
| PT5EN |      |      |      |      |      |      | PT5EN[1:0] |       |
|       | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1       | Bit0  |

Bit 1-0 PT5EN[1:0]: GPIO5 口输入/输出控制标志

PT5EN[1] = GPIO5 bit 1 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT5EN[0] = GPIO5 bit 0 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT5PU 寄存器（地址为 32h）

| 特性    | U-0  | U-0  | U-0  | U-0  | U-0  | U-0  | R/W-0      | R/W-0 |
|-------|------|------|------|------|------|------|------------|-------|
| PT5PU |      |      |      |      |      |      | PT5PU[1:0] |       |
|       | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1       | Bit0  |

Bit 1-0 PT5PU[1:0]: GPIO5 口上拉电阻使能标志

PT5PU[1] = GPIO5 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT5PU[0] = GPIO5 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

**特性 (Property) :**

R = 可读位

W = 可写位

U = 无效位

-n = 上电复位后的值 ‘1’ = 位已设置 ‘0’ = 位已清零

X = 不确定位



PT5CON 寄存器（地址为 33h）

| 特性     | U-0  | U-0  | U-0  | U-0  | U-0  | U-0  | R/W-0  | R/W-0  |
|--------|------|------|------|------|------|------|--------|--------|
| PT5CON |      |      |      |      |      |      | PT51OD | PT50OD |
|        | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1   | Bit0   |

Bit 1-0 PT5CON[1:0]: GPIO5 口控制标志

PT5CON[1] = GPIO5 bit 1 控制标志位; 0 = 禁止开漏输出, 1 = 使能开漏输出

PT5CON[0] = GPIO5 bit 0 控制标志位; 0 = 禁止开漏输出, 1 = 使能开漏输出

### 3 增强功能

#### 3.1 Halt和Sleep模式

CSU8RF3111/CSU8RF3112 支持低功耗工作模式。为了使 CSU8RF3111/CSU8RF3112 处于待机状态，可以让 CPU 停止工作使 CSU8RF3111/CSU8RF3112 进行停止或睡眠模式，减低功耗。这两种模式描述如下：

##### 停止模式

CPU 执行停止指令后，程序计数器停止计数直到出现中断指令。为了避免由中断返回（Interrupt Return）引起的程序错误，建议在停止指令之后加一 NOP 指令以保证程序返回时能正常运行。

##### 睡眠模式

CPU 执行睡眠指令后，所有的振荡器停止工作(EO\_SLP 为 0 时)直到出现一个外部中断指令复位 CPU。为了避免由中断返回（Interrupt Return）引起的程序错误，建议停止指令之后加一 NOP 指令以保证程序的正常运行。在睡眠模式下的功耗大约有 1uA。

为了保证 CPU 在睡眠模式下的功耗最小，在执行睡眠指令之前，需要把 IO 口的上拉电阻断开，并且保证所有的输入口是接到 VDD 或 VSS 电平。

##### 注：

芯片如果处于 sleep 状态，这时候降低电压，配置 2.4V 和 3.6V 低电压复位不会起作用，低于 2.0V 掉电复位点才会复位。如果 sleep 唤醒后，此时还处于低电压复位点以下，则会立即复位。

## Halt 示范程序:

```

...
movlw 01h
movwf pt1up ;断开 pt1 除 bit0(pt1[0])外的其他接口的上拉电阻
movlw feh
movwf pt1len ;pt1 口除 bit0(pt1[0])做输入口外, 其他接口作为输出口 (pt1.3 除外)
clrf pt1 ;将 pt1[4:1]输出为低
clrf pt3up ;断开 pt3 上拉电阻
clrf pt3en ;pt3 口用作输入口
clrf pt3con ;pt3 口用作数字口
clrf pt3 ;将 pt3 输出为低
clrf pt5up ;断开 pt5 上拉电阻
clrf pt5en ;pt5 口用作输入口
clrf pt5 ;将 pt5 输出为低
clrf intf ;清除中断标志位
movlw 81h
movwf inte ;使能外部中断 0
halt ;进入停止模式
nop ;保证 CPU 重启后程序能正常工作
...

```

## Sleep 示范程序:

```

...
movlw 01h
movwf pt1up ;断开 pt1 除 bit0(pt1[0])外的其他接口的上拉电阻
movlw feh
movwf pt1len ;pt1 口除 bit0(pt1[0])做输入口外, 其他接口作为输出口 (pt1.3 除外)
clrf pt1 ;将 pt1[4:1]输出为低
clrf pt3up ;断开 pt3 上拉电阻
clrf pt3en ;pt3 口用作输入口
clrf pt3con ;pt3 口用作数字口
clrf pt3 ;将 pt3 输出为低
clrf pt5up ;断开 pt5 上拉电阻
clrf pt5en ;pt5 口用作输入口
clrf pt5 ;将 pt5 输出为低
clrf intf ;清除中断标志位
movlw 81h
movwf inte ;使能外部中断 0
sleep ;进入睡眠模式
nop ;保证 CPU 重启后程序能正常工作
...

```

3.2 看门狗(WDT)

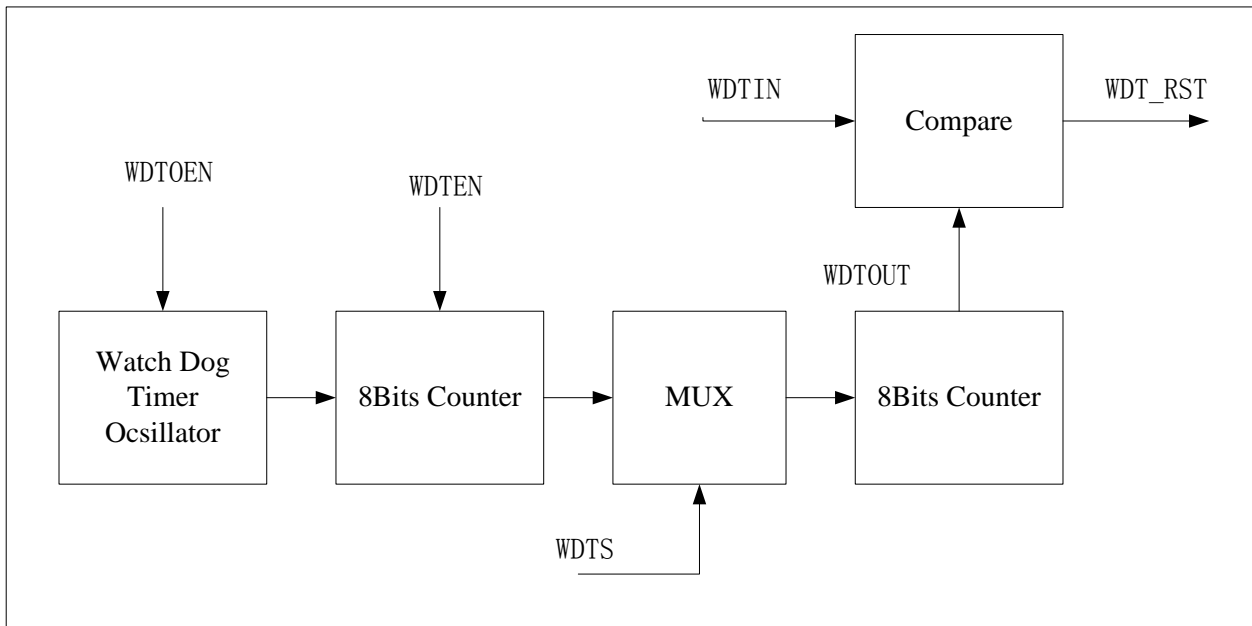


图12 看门狗定时器功能框图

看门狗定时器 (WDT) 用于防止程序由于某些不确定因素而失去控制。当 WDT 启动时，WDT 计时超时后将使 CPU 复位。在运行的程序一般在 WDT 复位 CPU 之前先复位 WDT。当出现某些故障时，程序会被 WDT 复位到正常状态下，但程序不会复位 WDT。

当用户把 CST\_WDT 清 0 时，则内部的看门狗定时器振荡器 (32KHz) 将会启动，产生的时钟被送到“8 bits 计数器 1”。当用户置位 WDTEN 时，“8 bits 计数器 1”开始计数，“8 bits 计数器 1”的输出是内部信号 WDTA[7:0]，被发送到一个受寄存器标志位 WDTS[2:0]控制的多路选择器，选择器的输出作为“8 bits 计数器 2”的时钟输入。当“8 bits 计数器 2”计数值与 WDTIN 数值相等时溢出，溢出时它会发送 WDTOUT 信号复位 CPU 及置位 TO 标志位。用户可以使用指令 CLRWDT 复位 WDT。

表 13 看门狗定时器寄存器表

| 地址  | 名称     | Bit7       | Bit6 | Bit5 | Bit4 | Bit3 | Bit2      | Bit1 | Bit0 | 上电复位值    |
|-----|--------|------------|------|------|------|------|-----------|------|------|----------|
| 04H | STATUS |            |      |      |      | TO   |           |      |      | 00u00xxx |
| 0DH | WDTCON | WDTEN      |      |      |      |      | WDTS[2:0] |      |      | 0uuuuxxx |
| 0EH | WDTIN  | WDTIN[7:0] |      |      |      |      |           |      |      | xxxxxxxx |

操作：

1. 设置 WDTS[3:0]，选择 WDT 时钟频率。
2. 设置 WDTIN，选择不同的溢出时间值
2. 置位寄存器标志位：WDTEN，使能 WDT。
3. 把 CST\_WDT 清 0，打开 WDT 的晶振。
4. 在程序中执行 CLRWDT 指令复位 WDT。

WDT 溢出时间计算公式:

$$\text{溢出时间} = \frac{2^{(8-\text{WDTS}[2:0])}}{32k} * (\text{WDTIN}[7:0] + 1)$$

WDTS[2:0]范围为 0~7, WDTIN[7:0]范围为 0~255。

| WDTS[2:0] | 计数器时钟    | 时间 (当 WDTIN==FFH) |
|-----------|----------|-------------------|
| 000       | WDTA [0] | 2048ms            |
| 001       | WDTA [1] | 1024ms            |
| 010       | WDTA [2] | 512ms             |
| 011       | WDTA [3] | 256ms             |
| 100       | WDTA [4] | 128ms             |
| 101       | WDTA [5] | 64ms              |
| 110       | WDTA [6] | 32ms              |
| 111       | WDTA [7] | 16ms              |

3.3 定时/计数器 2

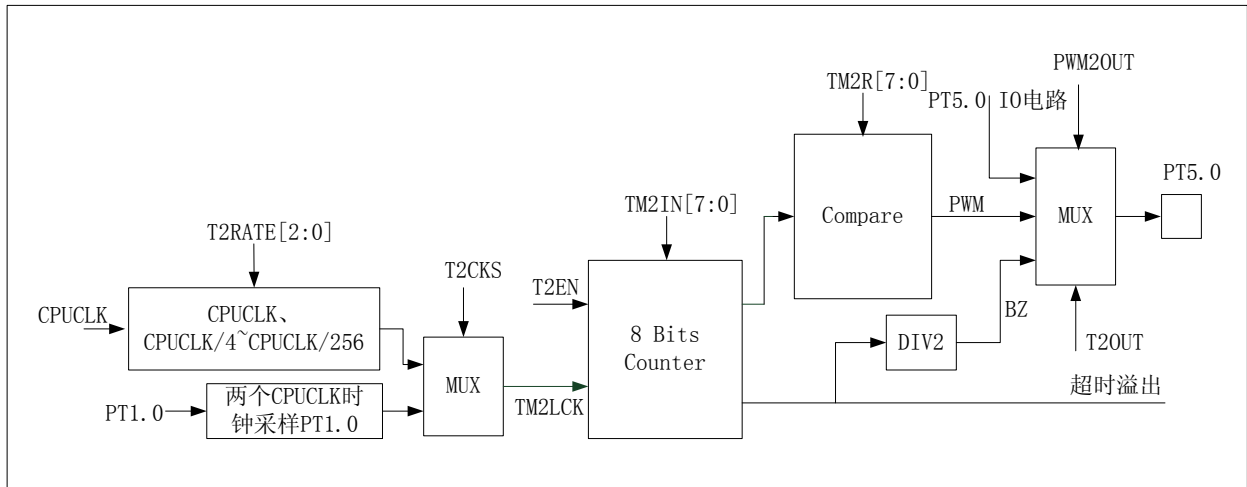


图13 定时/计数器 1 模块的功能框图

定时/计数器 2 模块的输入是 TM2CLK。当用户设置了定时/计数器 2 模块的使能标志，8 bits 计数器将启动，从 00h 递增到 TM2IN。用户需要设置 TM2IN（定时器模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，BZ 输出信号发生跳变。

主要功能：

- 1) 8 位可编程定时器；
- 2) 外部事件计数；
- 3) 蜂鸣器输出；
- 4) PWM2 输出；

3.3.1 寄存器描述

表 14 定时器寄存器列表

| 地址  | 名称     | Bit7        | Bit6        | Bit5 | Bit4  | Bit3   | Bit2  | Bit1    | Bit0 | 上电复位值    |
|-----|--------|-------------|-------------|------|-------|--------|-------|---------|------|----------|
| 06h | INTF   |             | TM2IF       |      |       |        |       |         |      | u0u00u00 |
| 07h | INTE   | GIE         | TM2IE       |      |       |        |       |         |      | 00u00u00 |
| 17h | TM2CON | T2EN        | T2RATE[2:0] |      | T2CKS | T2RSTB | T2OUT | PWM2OUT |      | 0xxxx1xx |
| 18h | TM2IN  | TM2IN[7:0]  |             |      |       |        |       |         |      | xxxxxxxx |
| 19h | TM2CNT | TM2CNT[7:0] |             |      |       |        |       |         |      | 00000000 |
| 1ah | TM2R   | TM2R[7:0]   |             |      |       |        |       |         |      | xxxxxxxx |

表 15 TM2CON 寄存器各位功能表

| 位地址                                | 标识符         | 功能                                                                                         |
|------------------------------------|-------------|--------------------------------------------------------------------------------------------|
| 7                                  | T2EN        | 定时/计数器 2 使能位<br>1: 使能定时器 2<br>0: 禁止定时器 2                                                   |
| 6:4                                | T2RATE[2:0] | 定时/计数器 1 时钟选择                                                                              |
|                                    |             | T2RATE [2:0]      TM2CLK                                                                   |
|                                    |             | 000                    CPUCLK                                                              |
|                                    |             | 001                    CPUCLK /4                                                           |
|                                    |             | 010                    CPUCLK /8                                                           |
|                                    |             | 011                    CPUCLK /16                                                          |
|                                    |             | 100                    CPUCLK /32                                                          |
|                                    |             | 101                    CPUCLK /64                                                          |
| 110                    CPUCLK /128 |             |                                                                                            |
| 111                    CPUCLK /256 |             |                                                                                            |
| 3                                  | T2CKS       | 定时/计数器 2 时钟源选择位<br>1: PT1.0 作为时钟<br>0: CPUCLK 的分频时钟                                        |
| 2                                  | T2RSTB      | 定时/计数器 2 复位<br>1: 禁止定时/计数器 2 复位<br>0: 使能定时/计数器 2 复位<br>当将该位为 0 时, 定时器 2 复位后, T2RSTB 会自动置 1 |
| 1                                  | T2OUT       | PT5.0 口输出控制                                                                                |
|                                    |             | T2OUT      PWM2OUT      PT5.0 输出控制, 仅当 PT5.0 配置为输出有效                                       |
| 0                                  | PWM2OUT     | 0                    0                    IO 输出                                            |
|                                    |             | 0                    1                    PWM2 输□                                          |
|                                    |             | 1                    0                    蜂鸣器输出                                            |
|                                    |             | 1                    1                    PWM2 输出                                          |

表 16 TM2IN 寄存器各位功能表

| 位地址   | 标识符        | 功能        |
|-------|------------|-----------|
| 7 : 0 | TM2IN[7:0] | 定时/计数器溢出值 |

表 17 TM2CNT 寄存器各位功能表

| 位地址   | 标识符         | 功能                 |
|-------|-------------|--------------------|
| 7 : 0 | TM2CNT[7:0] | 定时/计数器 2 计数寄存器, 只读 |

表 18 TM1CNT 寄存器各位功能表

| 位地址   | 标识符       | 功能                         |
|-------|-----------|----------------------------|
| 7 : 0 | TM2R[7:0] | 定时/计数器 2 的 PWM 高电平占空比控制寄存器 |



操作:

- 1) 设置 TM2CLK, 为定时器模块选择输入。
- 2) 设置 TM2IN, 选择定时器溢出值。
- 3) 设置寄存器标志位: TM2IE 与 GIE, 使能定时器中断。
- 4) 清零寄存器标志位: T2RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T2EN, 使能定时器模块的 8 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出; 程序计数器会跳转到 004H。

定时器 2 溢出时间计算方法:

$$\text{定时器 2 溢出时间} = (\text{TM2IN} + 1) / \text{TM2CLK.} \quad (\text{TM2IN 不为 0})$$

### 3.3.2 蜂鸣器

操作:

- 1) 把 PT5.0 配置为输出口。
- 2) 设置 TM2CLK, 为定时器模块选择输入。
- 3) 设置 TM2IN, 选择定时器溢出值。
- 4) 清零寄存器标志位: T2RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T2EN, 使能定时器模块的 8 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出。

蜂鸣器周期计算方法:

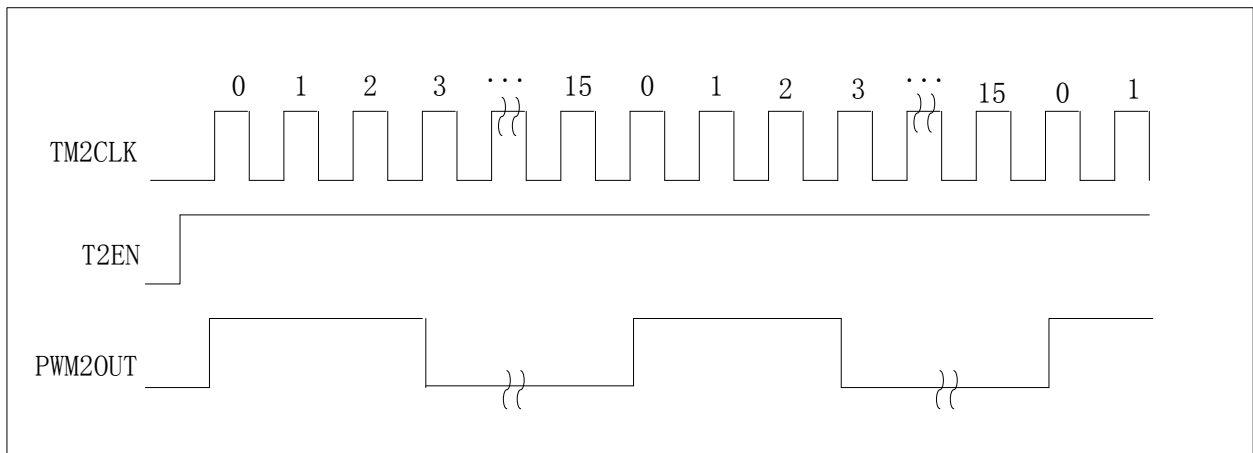
$$\text{蜂鸣器周期} = (\text{TM2IN} + 1) * 2 / \text{TM2CLK.} \quad (\text{TM2IN 不为 0})$$

### 3.3.3 PWM

操作:

- 1) 把 PT5.0 配置为输出口。
- 2) 设置 TM2CLK, 为定时/计数器 2 模块选择输入。
- 3) 设置 TM2IN 来配置 PWM2 的周期。
- 4) 设置 TM2R 来配置 PWM2 的高电平的脉宽。
- 5) 使能 PWM2OUT 输出, 配置 PT5.0 为输出口, 之后把 T2EN 置 1 启动定时器。
- 6) PWM 从 PT5.0 输出。

周期为 TM2IN+1, 高电平脉宽为 TM2R。如 TM2IN=0x0F, TM2R=0x03 的 PWM2 波形输出如下:



3.4 定时/计数器 3

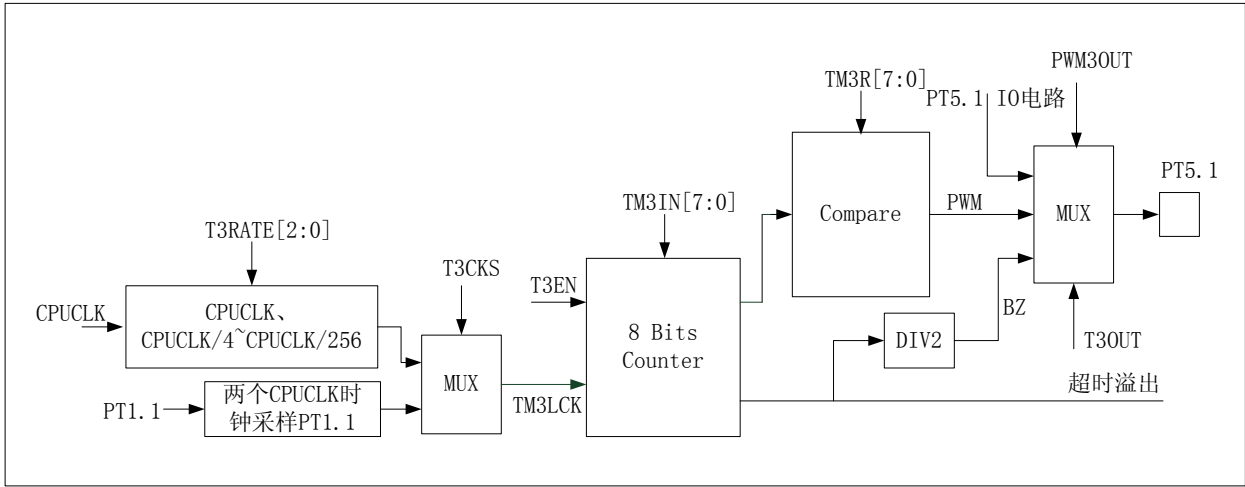


图14 定时/计数器 1 模块的功能框图

定时/计数器 3 模块的输入是 TM3CLK。当用户设置了定时/计数器 3 模块的使能标志，8 bits 计数器将启动，从 00h 递增到 TM3IN。用户需要设置 TM3IN（定时器模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，BZ 输出信号发生跳变。

主要功能：

- 1) 8 位可编程定时器；
- 2) 外部事件计数；
- 3) 蜂鸣器输出；
- 4) PWM 输出；

3.4.1 寄存器描述

表 19 定时器寄存器列表

| 地址  | 名称     | Bit7        | Bit6        | Bit5 | Bit4  | Bit3   | Bit2  | Bit1    | Bit0 | 上电复位值    |
|-----|--------|-------------|-------------|------|-------|--------|-------|---------|------|----------|
| 3ch | INTF2  |             |             |      | TM3IF |        |       |         |      | uuu0uuuu |
| 3dh | INTE2  |             |             |      | TM3IE |        |       |         |      | uuu0uuuu |
| 1bh | TM3CON | T3EN        | T3RATE[2:0] |      | T3CKS | T3RSTB | T3OUT | PWM3OUT |      | 0xxxx1xx |
| 1ch | TM3IN  | TM3IN[7:0]  |             |      |       |        |       |         |      | xxxxxxxx |
| 1dh | TM3CNT | TM3CNT[7:0] |             |      |       |        |       |         |      | 00000000 |
| 1eh | TM3R   | TM3R[7:0]   |             |      |       |        |       |         |      | xxxxxxxx |

表 20 TM3CON 寄存器各位功能表

| 位地址 | 标识符         | 功能                                                                                         |             |                              |
|-----|-------------|--------------------------------------------------------------------------------------------|-------------|------------------------------|
| 7   | T3EN        | 定时/计数器 3 使能位<br>1: 使能定时器 3<br>0: 禁止定时器 3                                                   |             |                              |
| 6:4 | T3RATE[2:0] | 定时/计数器 3 时钟选择                                                                              |             |                              |
|     |             | T3RATE [2:0]                                                                               | TM3CLK      |                              |
|     |             | 000                                                                                        | CPUCLK      |                              |
|     |             | 001                                                                                        | CPUCLK /4   |                              |
|     |             | 010                                                                                        | CPUCLK /8   |                              |
|     |             | 011                                                                                        | CPUCLK /16  |                              |
|     |             | 100                                                                                        | CPUCLK /32  |                              |
|     |             | 101                                                                                        | CPUCLK /64  |                              |
|     |             | 110                                                                                        | CPUCLK /128 |                              |
| 111 | CPUCLK /256 |                                                                                            |             |                              |
| 3   | T3CKS       | 定时/计数器 3 时钟源选择位<br>1: PT1.1 作为时钟<br>0: CPUCLK 的分频时钟                                        |             |                              |
| 2   | T3RSTB      | 定时/计数器 3 复位<br>1: 禁止定时/计数器 3 复位<br>0: 使能定时/计数器 3 复位<br>当将该位为 0 时, 定时器 3 复位后, T3RSBT 会自动置 1 |             |                              |
| 1   | T3OUT       | PT5.1 口输出控制                                                                                |             |                              |
|     |             | T3OUT                                                                                      | PWM3OUT     | PT5.1 输出控制, 仅当 PT5.1 配置为输出有效 |
| 0   | PWM3OUT     | 0                                                                                          | 0           | IO 输出                        |
|     |             | 0                                                                                          | 1           | PWM3 输出                      |
|     |             | 1                                                                                          | 0           | 蜂鸣器输出                        |
|     |             | 1                                                                                          | 1           | PWM3 输出                      |

表 21 TM3IN 寄存器各位功能表

| 位地址   | 标识符        | 功能        |
|-------|------------|-----------|
| 7 : 0 | TM3IN[7:0] | 定时/计数器溢出值 |

表 22 TM3CNT 寄存器各位功能表

| 位地址   | 标识符         | 功能                 |
|-------|-------------|--------------------|
| 7 : 0 | TM3CNT[7:0] | 定时/计数器 3 计数寄存器, 只读 |

表 23 TM3R 寄存器各位功能表

| 位地址   | 标识符       | 功能                         |
|-------|-----------|----------------------------|
| 7 : 0 | TM3R[7:0] | 定时/计数器 3 的 PWM 高电平占空比控制寄存器 |

操作:

- 1) 设置 TM3CLK, 为定时器模块选择输入。
- 2) 设置 TM3IN, 选择定时器溢出值。
- 3) 设置寄存器标志位: TM3IE 与 GIE, 使能定时器中断。
- 4) 清零寄存器标志位: T3RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T3EN, 使能定时器模块的 8 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出; 程序计数器会跳转到 004H。

定时器 3 溢出时间计算方法:

定时器 3 溢出时间 =  $(TM3IN+1) / TM3CLK$ . (TM3IN 不为 0)

### 3.4.2 蜂鸣器

操作:

- 1) 把 PT5.1 配置为输出口。
- 2) 设置 TM3CLK, 为定时器模块选择输入。
- 3) 设置 TM3IN, 选择定时器溢出值。
- 4) 清零寄存器标志位: T3RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T3EN, 使能定时器模块的 8 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出

蜂鸣器周期计算方法:

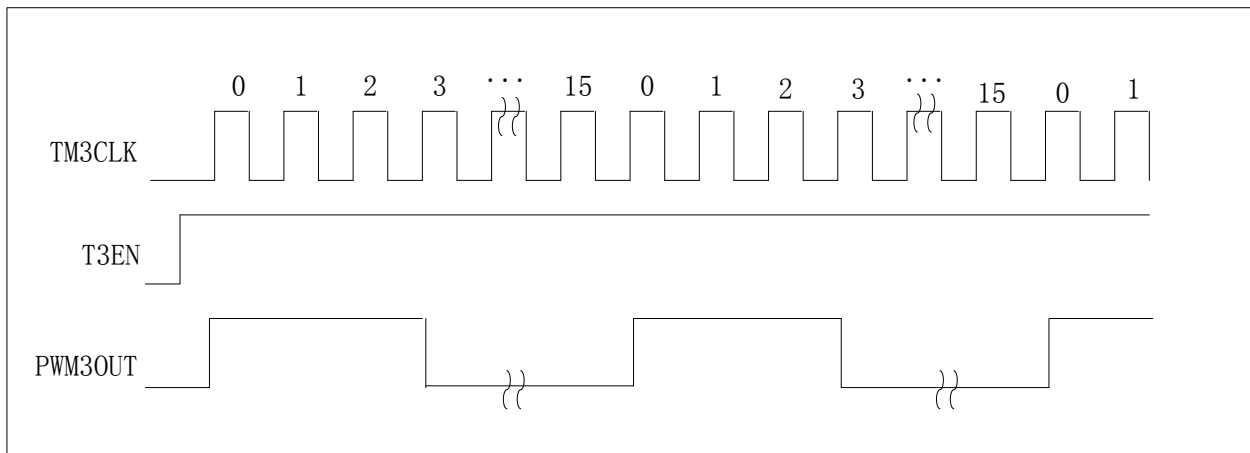
蜂鸣器周期 =  $(TM3IN+1) * 2 / TM3CLK$ . (TM3IN 不为 0)

### 3.4.3 PWM

操作:

- 1) 把 PT5.1 配置为输出口。
- 2) 设置 TM3CLK, 为定时/计数器 3 模块选择输入。
- 3) 设置 TM3IN 来配置 PWM3 的周期。
- 4) 设置 TM3R 来配置 PWM3 的高电平的脉宽。
- 5) 使能 PWM3OUT 输出, 配置 PT5.1 为输出口, 之后把 T3EN 置 1 启动定时器。
- 6) PWM3 从 PT5.1 输出。

周期为  $TM3IN+1$ , 高电平脉宽为  $TM3R$ 。如  $TM3IN=0x0F$ ,  $TM3R=0x03$  的 PWM3 波形输出如下:



3.5 模数转换器 (ADC)

CSU8RF3111/CSU8RF3112 模数转换模块共用 5 条外部通道 (AIN0~AIN4) 和 3 条特殊通道 (AIN5: 内部 1/8VDD; AIN6: 内部参考电压; AIN7: GND), 可以将模拟信号转换成 12 位数字信号。进行 AD 转换时, 首先要选择输入通道(AIN0~AIN5), 然后把 SRADEN 置 1 使能 ADC, 之后把 SRADS 置 1, 启动 AD 转换。转换结束后, 系统自动将 SRADS 清 0, 并将转换结果存入寄存器 SRADL 和 SRADH 中。

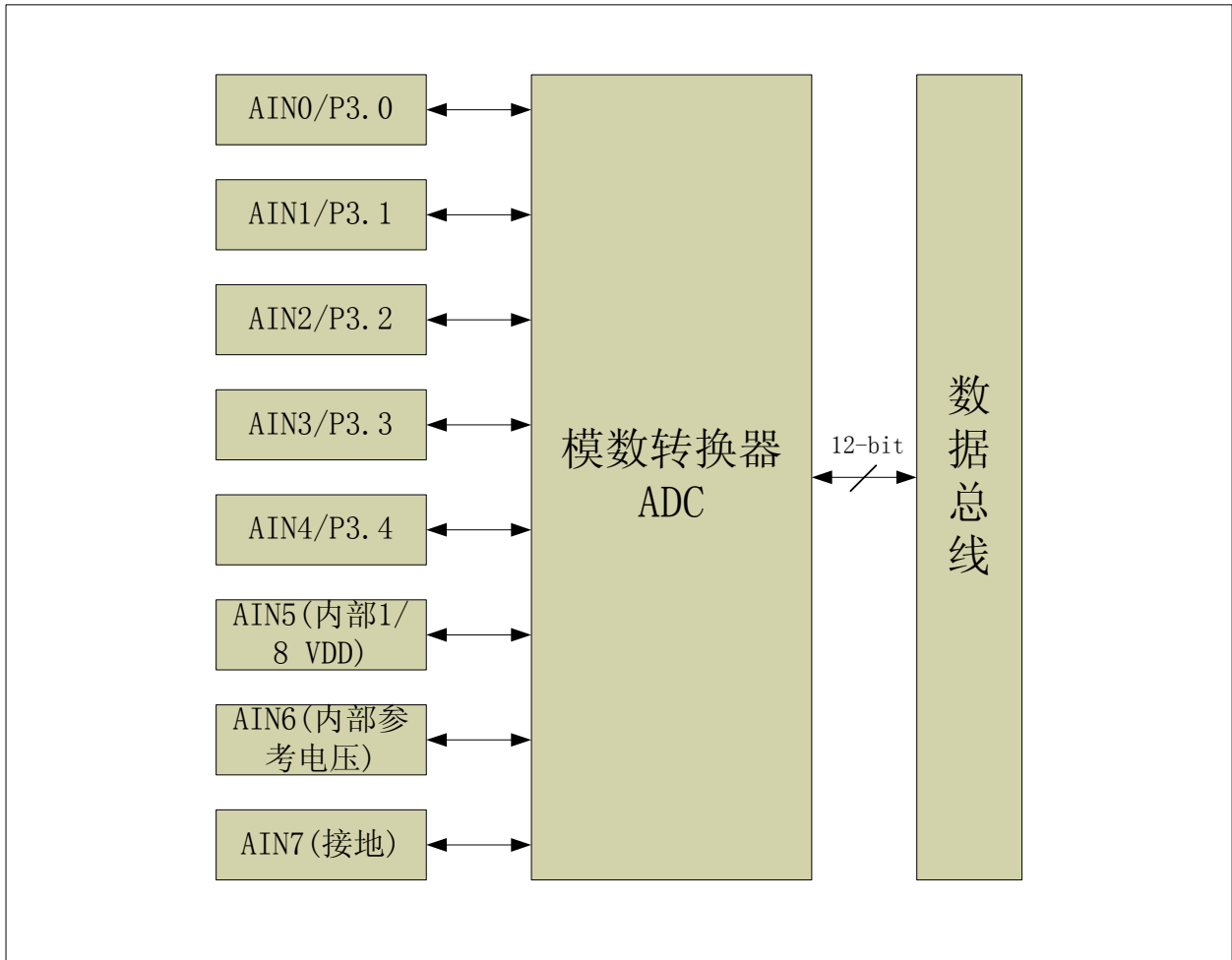


图15 模数转换器 ADC 功能框图

3.5.1 寄存器描述

表 24 ADC 寄存器列表

| 地址  | 名称       | Bit7      | Bit6  | Bit5          | Bit4  | Bit3 | Bit2  | Bit1         | Bit0 | 上电复位值    |         |
|-----|----------|-----------|-------|---------------|-------|------|-------|--------------|------|----------|---------|
| 06h | INTF     |           |       |               |       | ADIF |       |              |      | u0u00u00 |         |
| 07h | INTE     | GIE       |       |               |       | ADIE |       |              |      | 00u00u00 |         |
| 50h | SRADCON0 |           |       | SRADACKS[1:0] |       |      |       | SRADCKS[1:0] |      | uuxxuuxx |         |
| 51h | SRADCON1 | SRADEN    | SRADS | OFTEN         | CALIF | ENOV | OFFEX | VREFS[1:0]   |      | 00000000 |         |
| 52h | SRADCON2 | CHS[3:0]  |       |               |       |      |       |              |      | xxxxuuuu |         |
| 54h | SRADL    | SRAD[7:0] |       |               |       |      |       |              |      |          | xxxxxxx |

|     |        |            |  |  |  |             |          |          |
|-----|--------|------------|--|--|--|-------------|----------|----------|
| 55h | SRADH  |            |  |  |  | SRAD[11:8]  | uuuuxxxx |          |
| 56h | SROFTL | SROFT[7:0] |  |  |  |             |          | 00000000 |
| 57h | SROFTH |            |  |  |  | SROFT[11:8] | uuuu0000 |          |

表 25 SRADCON0 寄存器各位功能表

| 位地址  | 标识符           | 功能            |              |
|------|---------------|---------------|--------------|
| 5: 4 | SRADACKS[1:0] | ADC 输入信号获取时间  |              |
|      |               | SRADACKS[1:0] | ADC 输入信号获取时间 |
|      |               | 00            | 16 个 ADC 时钟  |
|      |               | 01            | 8 个 ADC 时钟   |
|      |               | 10            | 4 个 ADC 时钟   |
| 1: 0 | SRADCKS[1:0]  | ADC 时钟        |              |
|      |               | SRADCKS[1:0]  | ADC 采样时钟     |
|      |               | 00            | CPUCLK       |
|      |               | 01            | CPUCLK/2     |
|      |               | 10            | CPUCLK/4     |
|      |               | 11            | CPUCLK/8     |

表 26 SRADCON1 寄存器各位功能表

| 位地址 | 标识符        | 功能                                                                                                 |                |
|-----|------------|----------------------------------------------------------------------------------------------------|----------------|
| 7   | SRADEN     | ADC 使能位<br>1: 使能<br>0: 禁止                                                                          |                |
| 6   | SRADS      | ADC 启动位/状态控制位<br>1: 开始, 转换过程中<br>0: 停止, 转换结束<br>当置位后, 启动 ADC 转换, 转换完成会自动清 0                        |                |
| 5   | OFTEN      | 转换结果选择控制位<br>1: 转换结果放在 SROFT 寄存器中<br>0: 转换结果放在 SRAD 寄存器中                                           |                |
| 4   | CALIF      | 校正控制位(OFTEN 为 0 时有效)<br>1: 使能校正, 即 AD 转换的结果是减去了 SROFT 失调电压值<br>0: 禁止校正, 即 AD 转换结果是没有减去 SROFT 失调电压值 |                |
| 3   | ENOV       | 使能比较器溢出模式(CALIF 为 1 时有效)<br>1: 使能, 上溢或下溢直接是减去后的结果<br>0: 禁止, 下溢为 000h, 上溢为 fffh                     |                |
| 2   | OFFEX      | OFFSET 交换<br>1: 比较器两端信号交换<br>0: 比较器两端信号不交换 (正端为信号, 负端为参考电压)                                        |                |
| 1:0 | VREFS[1:0] | ADC 参考电源选择<br><b>注: 不同参考电压切换, 建议延迟 10uS 再做 AD 转换</b>                                               |                |
|     |            | VREFS[1:0]                                                                                         | AD 参考电压        |
|     |            | 00                                                                                                 | VDD            |
|     |            | 01                                                                                                 | PT3.0 外部参考电源输入 |



|  |  |    |                                              |
|--|--|----|----------------------------------------------|
|  |  | 10 | 内部参考电压 1.4V                                  |
|  |  | 11 | 内部参考电压 1.4V, PT3.0 可外接电容作为内置参考电压滤波使用, 以提高精度。 |

表 27 SRADCON2 寄存器各位功能表

| 位地址  | 标识符      | 功能          |                    |
|------|----------|-------------|--------------------|
| 7: 4 | CHS[3:0] | ADC 输入通道选择位 |                    |
|      |          | CHS[3:0]    | 输入通道               |
|      |          | 00□0        | AIN0 输入            |
|      |          | 0001        | AIN1 输入            |
|      |          | 0010        | AIN2 输入            |
|      |          | 0011        | AIN3 输入            |
|      |          | 0100        | AIN4 输入            |
|      |          | 0101        | AIN5 输入, 内部 1/8VDD |
|      |          | 0110        | AIN6 输入, 内部参考电压    |
|      |          | 0111        | AIN7 输入, 内部接地      |
|      | 其它       | 保留          |                    |

表 28 SRADL 寄存器各位功能表

| 位地址  | 标识符       | 功能                |
|------|-----------|-------------------|
| 7: 0 | SRAD[7:0] | ADC 数据的低 8 位, 只可读 |

表 29 SRADH 寄存器各位功能表

| 位地址  | 标识符        | 功能                |
|------|------------|-------------------|
| 3: 0 | SRAD[11:8] | ADC 数据的高 4 位, 只可读 |

表 30 SROFTL 寄存器各位功能表

| 位地址  | 标识符        | 功能          |
|------|------------|-------------|
| 7: 0 | SROFT[7:0] | 校正值数据的低 8 位 |

表 31 SROFTH 寄存器各位功能表

| 位地址  | 标识符         | 功能          |
|------|-------------|-------------|
| 3: 0 | SROFT[11:8] | 校正值数据的高 4 位 |

表 32 输入电压和 SRAD 输出数据的关系

| 输入电压        | SRAD[11:0] |    |   |   |   |   |   |   |   |   |   |   |
|-------------|------------|----|---|---|---|---|---|---|---|---|---|---|
|             | 11         | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0/4096*VREF | 0          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1/4096*VREF | 0          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ...         |            |    |   |   |   |   |   |   |   |   |   |   |

|                |   |   |   |   |   |   |   |   |   |   |   |   |
|----------------|---|---|---|---|---|---|---|---|---|---|---|---|
| ...            |   |   |   |   |   |   |   |   |   |   |   |   |
| 4094/4096*VREF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 4095/4096*VREF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

3.5.2 转换时间

|                                                                                               |
|-----------------------------------------------------------------------------------------------|
| $12 \text{ 位 AD 转换时间} = (1/\text{ADC 时钟频率}) \times (12 + \text{ADC 输入信号获取时间} + \text{CALIF})$ |
|-----------------------------------------------------------------------------------------------|

表 33 转换时间说明表<sup>(1)</sup>

| CLKDIV<br><small>(2)</small> | CALIF | SRADCKS | SRADACKS | AD 转换时间 <sup>(3)</sup>                                                  |
|------------------------------|-------|---------|----------|-------------------------------------------------------------------------|
|                              |       | 01      | 00       | $1 / ( (16\text{MHz} / 4) / 2 ) \times (12 + 0 + 16) = 14\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 4) / 2 ) \times (12 + 0 + 8) = 10\mu\text{s}$    |
|                              |       | 10      | 00       | $1 / ( (16\text{MHz} / 4) / 4 ) \times (12 + 0 + 16) = 28\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 4) / 4 ) \times (12 + 0 + 8) = 20\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 4) / 4 ) \times (12 + 0 + 4) = 16\mu\text{s}$    |
|                              |       | 11      | 00       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 0 + 16) = 56\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 0 + 8) = 40\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 0 + 4) = 32\mu\text{s}$    |
|                              |       |         | 11       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 0 + 2) = 28\mu\text{s}$    |
|                              |       | 01      | 00       | $1 / ( (16\text{MHz} / 4) / 2 ) \times (12 + 1 + 16) = 14.5\mu\text{s}$ |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 4) / 2 ) \times (12 + 1 + 8) = 10.5\mu\text{s}$  |
|                              |       | 10      | 00       | $1 / ( (16\text{MHz} / 4) / 4 ) \times (12 + 1 + 16) = 29\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 4) / 4 ) \times (12 + 1 + 8) = 21\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 4) / 4 ) \times (12 + 1 + 4) = 17\mu\text{s}$    |
|                              |       | 11      | 00       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 1 + 16) = 58\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 1 + 8) = 42\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 1 + 4) = 34\mu\text{s}$    |
|                              |       |         | 11       | $1 / ( (16\text{MHz} / 4) / 8 ) \times (12 + 1 + 2) = 30\mu\text{s}$    |
|                              |       | 01      | 00       | $1 / ( (16\text{MHz} / 8) / 2 ) \times (12 + 0 + 16) = 28\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 8) / 2 ) \times (12 + 0 + 8) = 20\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 8) / 2 ) \times (12 + 0 + 4) = 16\mu\text{s}$    |
|                              |       | 10      | 00       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 0 + 16) = 56\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 0 + 8) = 40\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 0 + 4) = 32\mu\text{s}$    |
|                              |       |         | 11       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 0 + 2) = 24\mu\text{s}$    |
|                              |       | 11      | 00       | $1 / ( (16\text{MHz} / 8) / 8 ) \times (12 + 0 + 16) = 112\mu\text{s}$  |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 8) / 8 ) \times (12 + 0 + 8) = 80\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 8) / 8 ) \times (12 + 0 + 4) = 64\mu\text{s}$    |
|                              |       |         | 11       | $1 / ( (16\text{MHz} / 8) / 8 ) \times (12 + 0 + 2) = 48\mu\text{s}$    |
|                              |       | 01      | 00       | $1 / ( (16\text{MHz} / 8) / 2 ) \times (12 + 1 + 16) = 29\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 8) / 2 ) \times (12 + 1 + 8) = 21\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 8) / 2 ) \times (12 + 1 + 4) = 17\mu\text{s}$    |
|                              |       | 10      | 00       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 1 + 16) = 58\mu\text{s}$   |
|                              |       |         | 01       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 1 + 8) = 42\mu\text{s}$    |
|                              |       |         | 10       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 1 + 4) = 34\mu\text{s}$    |
|                              |       |         | 11       | $1 / ( (16\text{MHz} / 8) / 4 ) \times (12 + 1 + 2) = 30\mu\text{s}$    |
|                              |       | 11      | 00       | $1 / ( (16\text{MHz} / 8) / 8 ) \times (12 + 1 + 16) = 116\mu\text{s}$  |

|  |  |    |    |                                                                       |
|--|--|----|----|-----------------------------------------------------------------------|
|  |  |    | 01 | $1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 8) = 84\mu\text{s}$    |
|  |  |    | 10 | $1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 4) = 68\mu\text{s}$    |
|  |  |    | 11 | $1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 2) = 60\mu\text{s}$    |
|  |  | 01 | 00 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 16) = 56\mu\text{s}$  |
|  |  |    | 01 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 8) = 40\mu\text{s}$   |
|  |  |    | 10 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 4) = 32\mu\text{s}$   |
|  |  |    | 11 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 2) = 28\mu\text{s}$   |
|  |  | 10 | 00 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 16) = 112\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 8) = 80\mu\text{s}$   |
|  |  |    | 10 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 4) = 64\mu\text{s}$   |
|  |  |    | 11 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 2) = 48\mu\text{s}$   |
|  |  | 11 | 00 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 16) = 224\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 8) = 160\mu\text{s}$  |
|  |  |    | 10 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 4) = 128\mu\text{s}$  |
|  |  |    | 11 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 2) = 96\mu\text{s}$   |
|  |  | 01 | 00 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 16) = 58\mu\text{s}$  |
|  |  |    | 01 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 8) = 42\mu\text{s}$   |
|  |  |    | 10 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 4) = 34\mu\text{s}$   |
|  |  |    | 11 | $1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 2) = 15\mu\text{s}$   |
|  |  | 10 | 00 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 16) = 116\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 8) = 84\mu\text{s}$   |
|  |  |    | 10 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 4) = 68\mu\text{s}$   |
|  |  |    | 11 | $1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 2) = 60\mu\text{s}$   |
|  |  | 11 | 00 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 16) = 232\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 8) = 168\mu\text{s}$  |
|  |  |    | 10 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 4) = 136\mu\text{s}$  |
|  |  |    | 11 | $1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 2) = 120\mu\text{s}$  |
|  |  | 01 | 00 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 0 + 16) = 112\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 0 + 8) = 80\mu\text{s}$   |
|  |  |    | 10 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 0 + 4) = 64\mu\text{s}$   |
|  |  |    | 11 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 0 + 2) = 56\mu\text{s}$   |
|  |  | 10 | 00 | $1 / ((16\text{MHz} / 32) / 4) \times (12 + 0 + 16) = 224\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 32) / 4) \times (12 + 0 + 8) = 160\mu\text{s}$  |
|  |  |    | 10 | $1 / ((16\text{MHz} / 32) / 4) \times (12 + 0 + 4) = 128\mu\text{s}$  |
|  |  |    | 11 | $1 / ((16\text{MHz} / 32) / 4) \times (12 + 0 + 2) = 96\mu\text{s}$   |
|  |  | 11 | 00 | $1 / ((16\text{MHz} / 32) / 8) \times (12 + 0 + 16) = 448\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 32) / 8) \times (12 + 0 + 8) = 320\mu\text{s}$  |
|  |  |    | 10 | $1 / ((16\text{MHz} / 32) / 8) \times (12 + 0 + 4) = 256\mu\text{s}$  |
|  |  |    | 11 | $1 / ((16\text{MHz} / 32) / 8) \times (12 + 0 + 2) = 192\mu\text{s}$  |
|  |  | 01 | 00 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 1 + 16) = 116\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 1 + 8) = 84\mu\text{s}$   |
|  |  |    | 10 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 1 + 4) = 68\mu\text{s}$   |
|  |  |    | 11 | $1 / ((16\text{MHz} / 32) / 2) \times (12 + 1 + 2) = 60\mu\text{s}$   |
|  |  | 10 | 00 | $1 / ((16\text{MHz} / 32) / 4) \times (12 + 1 + 16) = 232\mu\text{s}$ |
|  |  |    | 01 | $1 / ((16\text{MHz} / 32) / 4) \times (12 + 1 + 8) = 168\mu\text{s}$  |

|  |    |    |                                                                        |
|--|----|----|------------------------------------------------------------------------|
|  | 11 | 10 | $1 / ( (16\text{MHz} / 32) / 4) \times (12 + 1 + 4) = 136\mu\text{s}$  |
|  |    | 11 | $1 / ( (16\text{MHz} / 32) / 4) \times (12 + 1 + 2) = 120\mu\text{s}$  |
|  |    | 00 | $1 / ( (16\text{MHz} / 32) / 8) \times (12 + 1 + 16) = 464\mu\text{s}$ |
|  |    | 01 | $1 / ( (16\text{MHz} / 32) / 8) \times (12 + 1 + 8) = 336\mu\text{s}$  |
|  |    | 10 | $1 / ( (16\text{MHz} / 32) / 8) \times (12 + 1 + 4) = 272\mu\text{s}$  |
|  |    | 11 | $1 / ( (16\text{MHz} / 32) / 8) \times (12 + 1 + 2) = 240\mu\text{s}$  |

- (1) fosc=16MHz
- (2) 代码选项
- (3) AD 转换时间随 fosc 频率的改变而改变。

### 3.5.3 使用内部参考电压 1.40V 的校准方法

内部 1.4V 作参考时, 由于 1.4V 本身的浮动范围在 1.3-1.45V 之间, 所以需要对 1.4V 的参考进行校正。在烧录芯片的时候, 已经在 E2PROM 的 082fH(不同型号 MCU 的位置有差别, 注意参考用户手册)的地址里烧录了 1.4V 的校正值, 所以在上电的时候, 需要把这个校正值读出来, 在采到 AD 后, 把这个校正值与采到的 AD 值相乘然后除以 8000H, 得到的最后结果才是实际 AD 值。

如果有多个通道在使用时, 当检测其中任一通道时, 其它模拟口要设置为数字口。

```

movlf macro d1, f1
    movlw d1
    movwf f1
endm
movff macro f1, f2
    movfw f1
    movwf f2
endm
;*****
E2PROM_ROUT: ;上电先从EEPROM的82fH中读出1.4V参考的校准值
    movlf 1AH, eadr1 ;要读的EEPORM地址放入EADRL EADRH中
    movlf 04H, eadrh
    movp
    movwf M_D_TEMP12 ;低位W中的数据放在M_D_TEMP12中
    movff edath, M_D_TEMP11 ;高位EDATH的数据放在M_D_TEMP11中
    return
AD_init: ;CPUCLK = 16/4 =4MHZ
    movlw 00000001B ;B1:B0 01 ADC CLOCK = CPUCLK/2=2M
    movwf SRADCON0 ;B5 B4 = 0 0 ADC输入信号获取时间 16个ADC时钟
    movlw 00000010B ;B[5] 0 结果放在SRAD中
    movwf SRADCON1 ;B[1:0] 10 内部参考电压1.4V

    movlw 00000000B ;00H AINO输入
    movwf SRADCON2
    movlw 00000000B
    movwf SRADL ;[ADL7:0] ADC数据的低8位
    movwf SRADH ;[ADH11:8] ADC数据的高4位
    return
...
    
```



```

ADC_CONVERT:
    movlf  00001000B           ;把其它的模拟口设为数字口，打开当前的模拟口
    movwf  PT3CON
    movlf  30h, SRADCON2       ;转换第3通道的信号
    bsf    SRADCON1, ADEN
    call   delay40US
    bsf    SRADCON1, SRADS
    nop
    btfsc  SRADCON1, SRADS
    goto   $-1
    movfw  SRADL
    movwf  AD3_VALU_1, 1
    movfw  SRADH
    movwf  AD3_VALU_h, 1

    movff  AD3_VALU_h, M_D_TEMP3 ;采到的AD值作为被乘数
    movff  AD3_VALU_1, M_D_TEMP4
    movff  M_D_TEMP11, M_D_TEMP9
    movff  M_D_TEMP12, M_D_TEMP10 ;EEPROM里014AH读出的值为乘数
    call   F_Mu12_2             ;调用乘法运算
    movlf  80H, M_D_TEMP9
    movlf  00H, M_D_TEMP10      ;AD值与校准系数得到的积作被除数/8000H
    call   F_Div4_2             ;调用除法运算
    movff  M_D_TEMP3, AD3_VALU_h ;把除法结果重新放在AD保存寄存器里
    movff  M_D_TEMP4, AD3_VALU_1
    ...

```

### 3.5.4 AD失调电压校正

不同芯片由于离散性的原因，AD的失调电压可能有正有负。

校正失调电压的方法：

在AD转换过程中通过不断变换SRADCON1寄存器中的OFFEX的值。如第一次AD转换OFFEX置0，第二次AD转换OFFEX置1，然后将第一次和第二次测试的AD值求平均值。两次转换得到的平均值就是去掉失调电压的正确结果。

```

...
clrf sradcon1      ;VDD 为参考电压,often=0,calif=0;enov=0,offex=0,vrefs=00
movlw 20h
movwf sradcon2     ;chs[3:0]=0010, 选择通道 2
bsf sradcon1,7     ;使能 ADC 模块
call delay_10us
...
bsf sradcon1,6     ;srads=1, 开始转换
btfsc sradcon1,6   ;检测转换是否完成
goto $-1
movlw srادل
movwf adtmp1_1
movlw sradh
movwf adtmph_1
...
bsf sradcon1,2     ;offex=1
bsf sradcon1,6     ;srads=1, 开始转换
btfsc sradcon1,6   ;检测转换是否完成
goto $-1
movlw srادل
movwf adtmp1_2
movlw sradh
movwf adtmph_2
aver adtmph_1,adtmp1_1,adtmph_2,adtmp1_2 ;求两次 AD 值平均值, 并保存在
;adtmph_1,adtmp1_1
...

```

### 3.5.5 数字比较器

ADC 模块可作为一个数字比较器。被测信号的输入频率应小于转换频率的 1/2。比较器的速率是和 AD 转换频率相关的。

操作:

- 1) 通过 ADC 通道选择控制位 chs[3:0]选择比较器负端的信号输入, 之后把 OFTEN 置 1, CALIF 清 0, ENOV 置 0, 把 SRADEN 置 1 使能 ADC, SRADS 置 1 启动转换, 转换完成可把转换结果写入 SROFT 寄存器。  
也可以直接把负端信号的 AD 值直接写到 SROFT 寄存器中, 即人为指定负端电压值。
- 2) 通过 ADC 通道选择控制位 chs[3:0]选择比较器正端的信号输入, 之后把 OFTEN 置 0, CALIF 清 1, ENOV 置 1, 把 SRADEN 置 1 使能 ADC, SRADS 置 1 启动转换。
- 3) AD 数据的最高位 SRAD[11]则是比较器的结果, 为 0 时表示正端电压大于负端电压, 为 1 时表示正端电压小于负端电压。SRAD[11:0]为差值, 带符号位的补码。

比较通道 0 和通道 1 的电压值, 通道 0 接比较器正端, 通道 1 接比较器负端。

```

...
clrf sradcon1      ;VDD 为参考电压,often=0,calif=0;enov=0,offex=0,vrefs=00
bsf sradcon1,5    ;often=1,结果保存在 sroft 寄存器中
movlw 00h
movwf sradcon2    ;chs[3:0]=0000,选择通道 0 作为比较器负端
bsf sradcon1,7    ;使能 ADC 模块
call delay_10us
bsf sradcon1,6    ;srad=1,开始转换
btfsc sradcon1,6  ;检测转换是否完成
goto $-1
...
movlw 10h
movwf sradcon2    ;chs[3:0]=0001,选择通道 1 作为比较器正端
bcf sradcon1,5    ;often=0
bsf sradcon1,4    ;calif=1
bsf sradcon1,3    ;enov=1
bsf sradcon1,6    ;srad=1,开始转换
btfsc sradcon1,6  ;检测转换是否完成
goto $-1
btfsc sradh,3
goto le_cmp       ;正端电压小于负端电压
goto gt_cmp       ;正端大于等于负端电压
...

```

比较 1V 电压和通道 1 的电压，通道 1 接比较器正端，1V 接比较器负端，假设采用 5V 的 VDD 作为参考电压，那么 1V 的 AD 值为 0x333。

```

...
clrf sradcon1      ;VDD 为参考电压,often=0,calif=0;enov=0,offex=0,vrefs=00
movlw 10h
movwf sradcon2    ;chs[3:0]=0001,选择通道 1 作为比较器正端
bsf sradcon1,4    ;calif=1
bsf sradcon1,3    ;enov=1
movlw 03h
movwf sroft
movlw 33h
movwf sroftl      ;sroft 寄存器存入 333h,即 1V 作为比较器负端
bsf sradcon1,7    ;使能 ADC 模块
call delay_10us
bsf sradcon1,6    ;srad=1,开始转换
btfsc sradcon1,6  ;检测转换是否完成
goto $-1
btfsc sradh,3
goto le_cmp       ;正端电压小于负端电压
goto gt_cmp       ;正端大于等于负端电压
...

```



### 3.5.6 内部测量VDD的电压

用户可以通过使用内部参考电压或者外部参考电压输入（外部参考电压固定且不随 VDD 电压变化）两种方法来测试芯片内部 VDD 的电压。

使用外部参考电压，使用条件较多，需额外提供参考源。

使用内部参考电压不需要额外的硬件条件。但是，使用内部参考电压会由于本身内部参考电压值的不准而影响精度。可以通过内部参考电压校正来提高测试的精度。

外接 3V 作为参考电压，测 VDD 电压。选择通道 5，测出 1/8VDD 的 AD 值，之后乘以 8 得出 VDD 的 AD 值，再乘以参考电压则为 VDD 电压。

```

...
    clrf  sradcon1      ;often=0,calif=0;enov=0,offex=0,vrefs=00
    bsf  sradcon1,0    ;vrefs=01,选择外部参考电压,接3V
    movlw 50h
    movwf sradcon2     ;chs[3:0]=0101,选择通道5,1/8VDD
    bsf  sradcon1,7    ;使能ADC模块
    call delay_10us
    bsf  sradcon1,6    ;srad=1,开始转换
    btfsc sradcon1,6   ;检测转换是否完成
    goto $-1
    movlw sradl
    movwf adtmpl
    movlw sradh
    movwf adtmph
    bcf  status,c
    rlf  adtmpl
    rlf  adtmph        ;AD值乘以2
    rlf  adtmpl
    rlf  adtmph        ;AD值乘以4
    rlf  adtmpl
    rlf  adtmph        ;AD值乘以8,小数点在 adtmph 的 bit3 和 bit4 之间
...

```

3.6 数据E2PROM

CSU8RF3111/CSU8RF3112 具有 54 bytes 的 E2PROM，和用户程序存储器共同编址，具有掉电不丢失数据的特性。数据 E2PROM 的地址范围为 400H~41AH。

数据E2PROM地址为41AH的位置烧录时存放着内部参考电压1.40V的校准系数，若不需要用到内部参考电压1.40V时，41AH可用作数据E2PROM。

表 34 数据 E2PROM 寄存器列表

| 地址  | 名称    | Bit7       | Bit6 | Bit5 | Bit4 | Bit3 | Bit2      | Bit1 | Bit0 | 上电复位值    |
|-----|-------|------------|------|------|------|------|-----------|------|------|----------|
| 05H | WORK  | 工作寄存器      |      |      |      |      |           |      |      | XXXXXXXX |
| 0AH | EADRH |            |      |      |      |      | PAR[10:8] |      |      | uuuuuxxx |
| 0BH | EADRL | PAR[7:0]   |      |      |      |      |           |      |      | XXXXXXXX |
| 0CH | EDATH | EDATH[7:0] |      |      |      |      |           |      |      | XXXXXXXX |

EADRH/EADRL 提供写或者读操作的数据地址；

EDATH/WORK 提供写或读操作所用的数据。

E2PROM 的读或写操作都是基于一个字（16 bits）的。

执行写操作时，在地址和数据寄存器输入相应的值，之后执行 TBLP 指令(指令后面参数固定为 0，如 tblp 0)，便可在相应的 E2PROM 地址写入相应的数据。写操作要求指令周期为 1MHz~4MHz，或代码选项 CLKDIV[1:0]=2'b11 且指令周期 200KHz~800KHz。执行一次写操作大概需要 3ms 的时间。执行 TBLP 指令的时候，应关闭所有中断和 WDT。

执行读操作时，在地址寄存器输入相应的值，之后执行 MOVP 指令，便可在相应的 E2PROM 地址的数据读入到 EDATH/WORK 寄存器中。执行一次读操作大概需要 3 个指令周期。

```

movlw 04H
movwf EADRH ;给高字节地址赋值
movlw 00H
movwf EADRL ;给低字节地址赋值
movlw aaH
movwf EDATH ;给高字节数据赋值
movlw 55H ;给低字节数据赋值
tblp 0 ;执行写操作
nop
...
movp ;执行读操作
nop
...
    
```

3.7 烧录模块

烧写器的接口:

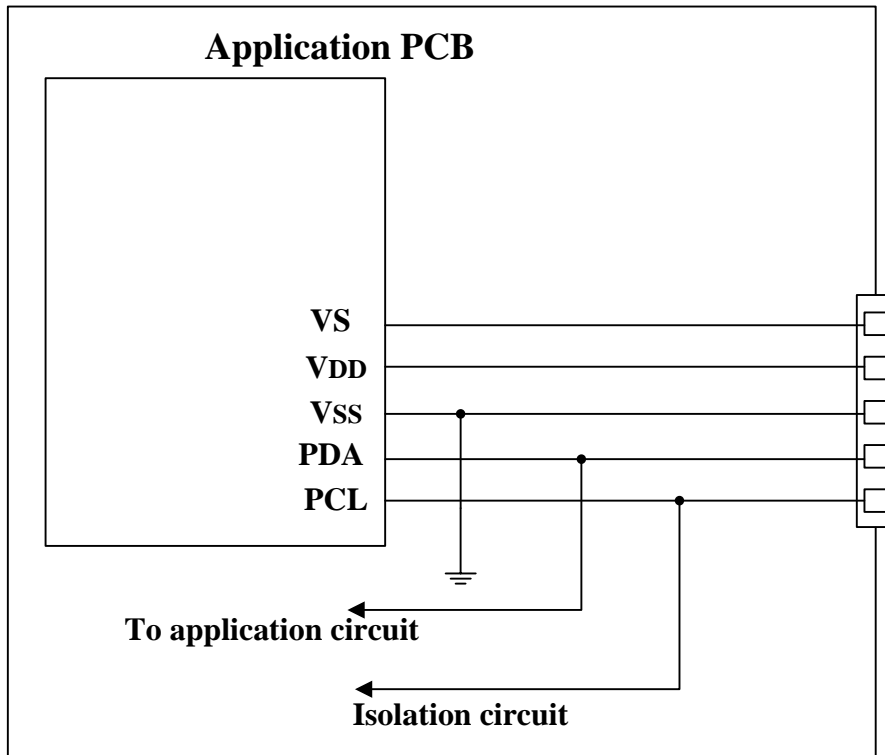


图16 烧写器接口图

表 35 烧录接口说明

| 端口名称 | 说明                                                                                                                         |
|------|----------------------------------------------------------------------------------------------------------------------------|
| VDD  | 输入                                                                                                                         |
| VSS  | 输入                                                                                                                         |
| PDA  | PT1[4]输入/输出                                                                                                                |
| PCL  | PT1[5]输入                                                                                                                   |
| VS   | PT3[0]AD 校正信号线，程序中若使用 AD 校正功能，则烧录时需连接此线，若无使用 1.4V 校正功能，则可以不连接，若连接，对烧录亦无影响。<br>(上述描述仅针对离线烧录器，使用仿真器烧录时无论是否使用 AD 校正功能，均不需要连接) |

3.8 代码选项

| 标识符                                                                                 | 功能                                                                                                                                                                                                                                                                                                                           |      |                         |                                                                                     |                                                  |                                                              |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------|
| CLKDIV                                                                              | 指令周期选择<br><table border="1"> <tr><td>指令周期</td></tr> <tr><td>指令周期=4 个时钟周期</td></tr> <tr><td>指令周期=8 个时钟周期</td></tr> <tr><td>指令周期=16 个时钟周期</td></tr> <tr><td>指令周期=32 个时钟周期</td></tr> </table>                                                                                                                                   | 指令周期 | 指令周期=4 个时钟周期            | 指令周期=8 个时钟周期                                                                        | 指令周期=16 个时钟周期                                    | 指令周期=32 个时钟周期                                                |
| 指令周期                                                                                |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 指令周期=4 个时钟周期                                                                        |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 指令周期=8 个时钟周期                                                                        |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 指令周期=16 个时钟周期                                                                       |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 指令周期=32 个时钟周期                                                                       |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| LVD_SEL                                                                             | LVD 配置<br><table border="1"> <tr><td>功能</td></tr> <tr><td>2.0V 上电/掉电复位。低电压检测关闭。</td></tr> <tr><td>2.0V 上电/掉电复位。<br/>STATUS 的 LVD24 作为 2.4V 的低电压检测器。<br/>STATUS 的 LVD36 作为 3.6V 的低电压检测器。</td></tr> <tr><td>2.4V 上电/掉电复位。<br/>STATUS 的 LVD36 作为 3.6V 的低电压检测器。</td></tr> <tr><td>3.6V 上电/掉电复位。</td></tr> </table>               | 功能   | 2.0V 上电/掉电复位。低电压检测关闭。   | 2.0V 上电/掉电复位。<br>STATUS 的 LVD24 作为 2.4V 的低电压检测器。<br>STATUS 的 LVD36 作为 3.6V 的低电压检测器。 | 2.4V 上电/掉电复位。<br>STATUS 的 LVD36 作为 3.6V 的低电压检测器。 | 3.6V 上电/掉电复位。                                                |
| 功能                                                                                  |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 2.0V 上电/掉电复位。低电压检测关闭。                                                               |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 2.0V 上电/掉电复位。<br>STATUS 的 LVD24 作为 2.4V 的低电压检测器。<br>STATUS 的 LVD36 作为 3.6V 的低电压检测器。 |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 2.4V 上电/掉电复位。<br>STATUS 的 LVD36 作为 3.6V 的低电压检测器。                                    |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| 3.6V 上电/掉电复位。                                                                       |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| RESET_PIN                                                                           | 复位引脚选择<br>PT1.3 作为复位引脚<br>PT1.3 作为普通输入口                                                                                                                                                                                                                                                                                      |      |                         |                                                                                     |                                                  |                                                              |
| XTAL_PIN                                                                            | 晶振引脚选择<br><table border="1"> <tr><td>晶振引脚</td></tr> <tr><td>PT1.4 和 PT1.5 作为普通 IO 口</td></tr> <tr><td>PT1.5 作为外部 RC 时钟输入；或者<br/>PT1.5 通过外部时钟源直接灌入时钟；<br/>PT1.4 还是普通 IO 口；</td></tr> <tr><td>PT1.4 和 PT1.5 接外部晶振为 32768Hz</td></tr> <tr><td>PT1.4 和 PT1.5 接外部晶振<br/>4M~16MHz； 还可通过 PT1.5 灌时钟，<br/>PT1.4 悬空</td></tr> </table> | 晶振引脚 | PT1.4 和 PT1.5 作为普通 IO 口 | PT1.5 作为外部 RC 时钟输入；或者<br>PT1.5 通过外部时钟源直接灌入时钟；<br>PT1.4 还是普通 IO 口；                   | PT1.4 和 PT1.5 接外部晶振为 32768Hz                     | PT1.4 和 PT1.5 接外部晶振<br>4M~16MHz； 还可通过 PT1.5 灌时钟，<br>PT1.4 悬空 |
| 晶振引脚                                                                                |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| PT1.4 和 PT1.5 作为普通 IO 口                                                             |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| PT1.5 作为外部 RC 时钟输入；或者<br>PT1.5 通过外部时钟源直接灌入时钟；<br>PT1.4 还是普通 IO 口；                   |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| PT1.4 和 PT1.5 接外部晶振为 32768Hz                                                        |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |
| PT1.4 和 PT1.5 接外部晶振<br>4M~16MHz； 还可通过 PT1.5 灌时钟，<br>PT1.4 悬空                        |                                                                                                                                                                                                                                                                                                                              |      |                         |                                                                                     |                                                  |                                                              |

## 4 MCU 指令集

表 36 表 MCU 指令集

| 指令         | 操作                                                                   | 指令周期 | 标志位    |
|------------|----------------------------------------------------------------------|------|--------|
| ADDLW k    | $[W] \leftarrow [W] + k$                                             | 1    | C,DC,Z |
| ADDPCW     | $[PC] \leftarrow [PC] + 1 + [W]$                                     | 2    | ~      |
| ADDWF f,d  | $[Destination] \leftarrow [f] + [W]$                                 | 1    | C,DC,Z |
| ADDWFC f,d | $[Destination] \leftarrow [f] + [W] + C$                             | 1    | C,DC,Z |
| ANDLW k    | $[W] \leftarrow [W] \text{ AND } k$                                  | 1    | Z      |
| ANDWF f,d  | $[Destination] \leftarrow [W] \text{ AND } [f]$                      | 1    | Z      |
| BCF f,b    | $[f \langle b \rangle] \leftarrow 0$                                 | 1    | ~      |
| BSF f,b    | $[f \langle b \rangle] \leftarrow 1$                                 | 1    | ~      |
| BTFSC f,b  | Jump if $[f \langle b \rangle] = 0$                                  | 1/2  | ~      |
| BTFSS f,b  | Jump if $[f \langle b \rangle] = 1$                                  | 1/2  | ~      |
| CALL k     | Push PC+1 and Goto K                                                 | 2    | ~      |
| CLRF f     | $[f] \leftarrow 0$                                                   | 1    | Z      |
| CLRWDT     | Clear watch dog timer                                                | 1    | ~      |
| COMF f,d   | $[f] \leftarrow \text{NOT}([f])$                                     | 1    | Z      |
| DAW        | Decimal Adjust W                                                     | 1    | C,DC   |
| DECF f,d   | $[Destination] \leftarrow [f] - 1$                                   | 1    | Z      |
| DECFSZ f,d | $[Destination] \leftarrow [f] - 1$ , jump if the result is zero      | 1/2  | ~      |
| GOTO k     | $PC \leftarrow k$                                                    | 2    | ~      |
| HALT       | CPU Stop                                                             | 1    | ~      |
| INCF f,d   | $[Destination] \leftarrow [f] + 1$                                   | 1    | Z      |
| INCFSZ f,d | $[Destination] \leftarrow [f] + 1$ , jump if the result is zero      | 1/2  | ~      |
| IORLW k    | $[W] \leftarrow [W] \text{ OR } k$                                   | 1    | Z      |
| IORWF f,d  | $[Destination] \leftarrow [W] \text{ OR } [f]$                       | 1    | Z      |
| MOVFW f    | $[W] \leftarrow [f]$                                                 | 1    | ~      |
| MOVLW k    | $[W] \leftarrow k$                                                   | 1    | ~      |
| MOVP       | Read eprom                                                           | 3    | ~      |
| MOVWF f    | $[f] \leftarrow [W]$                                                 | 1    | ~      |
| NOP        | No operation                                                         | 1    | ~      |
| POP        | Pop W and Status                                                     | 2    | ~      |
| PUSH       | Push W and Status                                                    | 2    | ~      |
| RETFIE     | Pop PC and GIE = 1                                                   | 2    | ~      |
| RETLW k    | RETURN and W=k                                                       | 2    | ~      |
| RETURN     | POP PC                                                               | 2    | ~      |
| RLF f,d    | $[Destination \langle n+1 \rangle] \leftarrow [f \langle n \rangle]$ | 1    | C,Z    |
| RRF f,d    | $[Destination \langle n-1 \rangle] \leftarrow [f \langle n \rangle]$ | 1    | C,Z    |
| SLEEP      | STOP OSC                                                             | 1    | PD     |
| SUBLW k    | $[W] \leftarrow k - [W]$                                             | 1    | C,DC,Z |
| SUBWF f,d  | $[Destinnation] \leftarrow [f] - [W]$                                | 1    | C,DC,Z |
| SUBWFC f,d | $[Destinnation] \leftarrow [f] - [W] - 1 + C$                        | 1    | C,DC,Z |
| SWAPF f,d  | swap f                                                               | 1    | ~      |
| TBLP k     | Write e2prom                                                         | -    | ~      |
| XORLW k    | $[W] \leftarrow [W] \text{ XOR } k$                                  | 1    | Z      |
| XORWF f,d  | $[Destination] \leftarrow [W] \text{ XOR } [f]$                      | 1    | Z      |

参数说明:

f: 数据存储器地址(00H ~DFH)

W: 工作寄存器

k: 立即数

d: 目标地址选择: d=0 结果保存在工作寄存器, d=1: 结果保存在数据存储器 f 单

元

b: 位选择(0~7)

[f]: f 地址的内容

PC: 程序计数器

C: 进位标志

DC: 半加进位标志

Z: 结果为零标志

PD: 睡眠标志位

TO: 看门狗溢出标志

WDT: 看门狗计数器

表 37 MCU 指令集描述

1

|                 |                                        |
|-----------------|----------------------------------------|
| ADDLW           | 加立即数到工作寄存器                             |
| 指令格式            | ADDLW K (0<=K<=FFH)                    |
| 操作              | (W)<←(W)+K                             |
| 标志位             | C, DC, Z                               |
| 描述              | 工作寄存器的内容加上立即数 K 结果保存到工作寄存器中            |
| 周期              | 1                                      |
| 例子<br>ADDLW 08H | 在指令执行之前:<br>W=08H<br>在指令执行之后:<br>W=10H |

2

|                |                                                        |
|----------------|--------------------------------------------------------|
| ADDPCW         | 将 W 的内容加到 PC 中                                         |
| 指令格式           | ADDPCW                                                 |
| 操作             | (PC)<←(PC)+1+(W) 当(W)<=7FH<br>(PC)<←(PC)+1+(W)-100H 其余 |
| 标志位            | 没有                                                     |
| 描述             | 将地址 PC+1+W 加载到 PC 中                                    |
| 周期             | 2                                                      |
| 例子 1<br>ADDPCW | 在指令执行之前:<br>W=7FH, PC=0212H<br>指令执行之后:<br>PC=0292H     |
| 例子 2<br>ADDPCW | 在指令执行之前:<br>W=80H, PC=0212H<br>指令执行之后:<br>PC=0193H     |
| 例子 3           | 在指令执行之前:                                               |

|         |                                         |
|---------|-----------------------------------------|
| ADDDPCW | W=FEH , PC=0212H<br>指令执行之后:<br>PC=0211H |
|---------|-----------------------------------------|

3

|                   |                                                                         |
|-------------------|-------------------------------------------------------------------------|
| ADDWF             | 加工作寄存器到 f                                                               |
| 指令格式              | ADDWF f,d 0<=f<=DFH d=0,1                                               |
| 操作                | [目标地址]<←(f)+(W)                                                         |
| 标志位               | C, CD, Z                                                                |
| 描述                | 将 f 的内容和工作寄存器的内容加到一起。<br>如果 d 是 0, 结果保存到工作寄存器中。<br>如果 d 是 1, 结果保存到 f 中。 |
| 周期                | 1                                                                       |
| 例子 1<br>ADDWF f 0 | 指令执行之前:<br>f=C2H W=17H<br>在指令执行之后<br>f=C2H W=D9H                        |
| 例子 2<br>ADDWF f 1 | 指令执行之前<br>f=C2H W=17H<br>指令执行之后<br>f=D9H W=17H                          |

4

|                   |                                                                      |
|-------------------|----------------------------------------------------------------------|
| ADDWFC            | 将 W f 和进位位相加                                                         |
| 指令格式              | ADDWFC f, d 0<=f<=DFH d=0,1                                          |
| 操作                | (目标地址)<←(f)+(W)+C                                                    |
| 标志位               | C, DC, Z                                                             |
| 描述                | 将工作寄存器的内容和 f 的内容以及进位位相加<br>当 d 为 0 时结果保存到工作寄存器<br>当 d 为 1 时结果保存到 f 中 |
| 周期                | 1                                                                    |
| 例子<br>ADDWFC f, 1 | 指令执行之前<br>C=1 f=02H W=4DH<br>指令执行之后<br>C=0 f=50H W=4DH               |

5

|                 |                                      |
|-----------------|--------------------------------------|
| ANDLW           | 工作寄存器与立即数相与                          |
| 指令格式            | ANDLW K 0<=K<=FFH                    |
| 操作              | (W)<←(W) AND K                       |
| 标志位             | Z                                    |
| 描述              | 将工作寄存器的内容与 8bit 的立即数相与, 结果保存到工作寄存器中。 |
| 周期              | 1                                    |
| 例子<br>ANDLW 5FH | 在指令执行之前<br>W=A3H<br>在指令执行之后<br>W=03H |



6

|                    |                                                                  |
|--------------------|------------------------------------------------------------------|
| ANDWF              | 将工作寄存器和 f 的内容相与                                                  |
| 指令格式               | ANDWF f, d 0<=f<=DFH d=0,1                                       |
| 操作                 | (目标地址)<←(W) AND (f)                                              |
| 标志位                | Z                                                                |
| 描述                 | 将工作寄存器的内容和 f 的内容相与<br>如果 d 为 0 结果保存到工作寄存器中<br>如果 d 为 1 结果保存到 f 中 |
| 周期                 | 1                                                                |
| 例子 1<br>ANDWF f, 0 | 在指令执行之前<br>W=0FH f=88H<br>在指令执行之后<br>W=08H f=88H                 |
| 例子 2<br>ANDWF f, 1 | 在指令执行之前<br>W=0FH f=88H<br>在指令执行之后<br>W=0FH f=08H                 |

7

|                  |                                            |
|------------------|--------------------------------------------|
| BCF              | 清除 f 的某一位                                  |
| 指令格式             | BCF f, b 0<=f<=DFH 0<=b<=7                 |
| 操作               | (f[b])<←0                                  |
| 标志位              | 无                                          |
| 描述               | F 的第 b 位置为 0                               |
| 周期               | 1                                          |
| 例子<br>BCF FLAG 2 | 指令执行之前:<br>FLAG=8DH<br>指令执行之后:<br>FLAG=89H |

8

|                  |                                            |
|------------------|--------------------------------------------|
| BSF              | F 的 b 位置 1                                 |
| 指令格式             | BSF f, b 0<=f<=DFH 0<=b<=7                 |
| 操作               | (f[b])<←1                                  |
| 标志位              | 无                                          |
| 描述               | 将 f 的 b 位置 1                               |
| 周期               | 1                                          |
| 例子<br>BSF FLAG 2 | 在指令执行之前<br>FLAG=89H<br>在指令执行之后<br>FLAG=8DH |

9

|                                         |                                                                                                               |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------|
| BTFSC                                   | 如果 bit 测试为 0 则跳转                                                                                              |
| 指令格式                                    | BTFSC f, b $0 \leq f \leq DFH$ $0 \leq b \leq 7$                                                              |
| 操作                                      | Skip if (f[b])=0                                                                                              |
| 标志位                                     | 无                                                                                                             |
| 描述                                      | 如果 f 的 bit 位是 0, 下一条取到的指令将被丢到, 然后执行一条空指令组成一个两周期的指令。                                                           |
| 周期                                      | 无跳转则为 1 个指令周期, 否则 2 个指令周期                                                                                     |
| 例子<br>NODE BTFSC FLAG 2<br>OP1:<br>OP2: | 在程序执行以前<br>PC=address(NODE)<br>指令执行之后<br>If(FLAG[2])=0<br>PC=address(OP2)<br>If(FLAG[2])=1<br>PC=address(OP1) |

10

|                                         |                                                                                                               |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------|
| BTFSS                                   | 如果 bit 测试为 1, 则跳转                                                                                             |
| 指令格式                                    | BTFSS f, b $0 \leq f \leq DFH$ $0 \leq b \leq 7$                                                              |
| 操作                                      | Skip if (f[b])=1                                                                                              |
| 标志位                                     | 无                                                                                                             |
| 描述                                      | 如果 f 的 bit 位是 1, 下一条取到的指令将被丢到, 然后执行一条空指令组成一个两周期的指令。                                                           |
| 周期                                      | 无跳转则为 1 个指令周期, 否则 2 个指令周期                                                                                     |
| 例子<br>NODE BTFSS FLAG 2<br>OP1:<br>OP2: | 在程序执行以前<br>PC=address(NODE)<br>指令执行之后<br>If(FLAG[2])=0<br>PC=address(OP1)<br>If(FLAG[2])=1<br>PC=address(OP2) |

11

|      |                                        |
|------|----------------------------------------|
| CALL | 子程序调用                                  |
| 指令格式 | CALL K $0 \leq K \leq 3FFH$            |
| 操作   | (top stack) ← PC+1<br>PC ← K           |
| 标志位  | 无                                      |
| 描述   | 子程序调用, 先将 PC+1 压入堆栈, 然后把立即数地址下载到 PC 中。 |
| 周期   | 2                                      |
|      |                                        |

12

|                 |                                            |
|-----------------|--------------------------------------------|
| CLRF            | 清除 f                                       |
| 指令格式            | CLRF f 0<=f<=DFH                           |
| 操作              | (f)←0                                      |
| 标志位             | Z                                          |
| 描述              | 将 f 的内容清零                                  |
| 周期              | 1                                          |
| 例子<br>CLRF WORK | 在指令执行之前<br>WORK=5AH<br>在指令执行之后<br>WORK=00H |

\*注。当 clrf 清除 status 寄存器时，标志位 Z 不会置高

13

|              |                 |
|--------------|-----------------|
| CLRWDT       | 清除看门狗定时器        |
| 指令格式         | CLRWDT          |
| 操作           | 看门狗计数器清零        |
| 标志位          | 无               |
| 描述           | 清除看门狗定时器        |
| 周期           | 1               |
| 例子<br>CLRWDT | 指令执行之后<br>WDT=0 |

14

|                   |                                                              |
|-------------------|--------------------------------------------------------------|
| COMF              | f 取反                                                         |
| 指令格式              | COMF f, d 0<=f<=DFH d=0,1                                    |
| 操作                | (目的地址)←NOT(f)                                                |
| 标志位               | Z                                                            |
| 描述                | 将 f 的内容取反，<br>当 d 为 0 时，结果保存到工作寄存器中，<br>当 d 为 1 时，结果保存到 f 中。 |
| 周期                | 1                                                            |
| 例子<br>COMF f, 0   | 在指令执行之前<br>W=88H, f=23H<br>在指令执行之后<br>W=DCH, f=23H           |
| 例子 2<br>COMF f, 1 | 在指令执行之前<br>W=88H, f=23H<br>在指令执行之后<br>W=88H, f=DCH           |

15

|                                    |                                                                                                                     |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| DAW                                | 十进制调整 W 寄存器                                                                                                         |
| 指令格式                               | DAW                                                                                                                 |
| 操作                                 | 十进制调整 W 寄存器                                                                                                         |
| 标志位                                | C,DC                                                                                                                |
| 描述                                 | 一般与加法一起使用。<br>如果低半字节的值大于 9 或 DC 为 1 时，低半字节加 6；<br>如果高半字节的值大于 9 或 C 为 1 时，高半字节加 6                                    |
| 周期                                 | 1                                                                                                                   |
| 例子<br>若 W=25H;<br>ADDLW 39H<br>DAW | 在 DAW 指令执行之前<br>W=25H+39H =64=5EH<br>在指令执行之后<br>W= (64) BCD<br><br><pre> 25H + 39H ----- 5EH + 06H ----- 64H </pre> |

16

|                   |                                                           |
|-------------------|-----------------------------------------------------------|
| DECF              | f 减 1                                                     |
| 指令格式              | DECF f, d 0<=f<=DFH d=0,1                                 |
| 操作                | (目的地址)<←(f)-1                                             |
| 标志位               | Z                                                         |
| 描述                | F 的内容减 1<br>当 d 为 0 时，结果保存到工作寄存器中<br>当 d 为 1 时，结果保存到 f 中。 |
| 周期                | 1                                                         |
| 例子<br>DECF f, 0   | 在指令执行之前<br>W=88H f=23H<br>在指令执行之后<br>W=22H f=23H          |
| 例子 2<br>DECF f, 1 | 在指令执行之前<br>W=88H f=23H<br>在指令执行之后<br>W=88H f=22H          |

17

|                                           |                                                                                                                              |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| DECFSZ                                    | f 减 1 如果为 0 则跳转                                                                                                              |
| 指令格式                                      | DECFSZ f, d 0<=f<=DFH d=0,1                                                                                                  |
| 操作                                        | (目的地址)<-(f)-1,如果结果为 0 跳转                                                                                                     |
| 标志位                                       | 无                                                                                                                            |
| 描述                                        | f 的内容减 1。<br>如果 d 为 0, 结果保存到工作寄存器中。<br>如果 d 为 1, 结果保存到 f 中<br>如果结果为 0, 下一条已经取到的指令将被丢掉, 然后插入一条 NOP 指令组成一个两个周期的指令。             |
| 周期                                        | 无跳转则为 1 个指令周期, 否则 2 个指令周期                                                                                                    |
| 例子<br>Node DECFSZ FLAG, 1<br>OP1:<br>OP2: | 在指令执行之前<br>PC=address(Node)<br>在指令执行之后<br>(FLAG)=(FLAG)-1<br>If(FLAG)=0<br>PC=address(OP2)<br>If(FLAG)!=0<br>PC=address(OP1) |

18

|      |                   |
|------|-------------------|
| GOTO | 无条件跳转             |
| 指令格式 | GOTO K 0<=K<=3FFH |
| 操作   | PC<-K             |
| 标志位  | 无                 |
| 描述   | 立即地址载入 PC         |
| 周期   | 2                 |

19

|      |                                       |
|------|---------------------------------------|
| HALT | 停止 CPU 时钟                             |
| 指令格式 | HALT                                  |
| 操作   | CPU 停止                                |
| 标志位  | 无                                     |
| 描述   | CPU 时钟停止, 晶振仍然工作, CPU 能够通过内部或者外部中断重启。 |
| 周期   | 1                                     |

20

|                   |                                                        |
|-------------------|--------------------------------------------------------|
| INCF              | f 加 1                                                  |
| 指令格式              | INCF f, d 0<=f<=DFH d=0,1                              |
| 操作                | (目的地址)<←(f)+1                                          |
| 标志位               | Z                                                      |
| 描述                | f 加 1<br>如果 d 为 0, 结果保存到工作寄存器中<br>如果 d 为 1, 结果保存到 f 中。 |
| 周期                | 1                                                      |
| 例子<br>INCF f, 0   | 在指令执行之前<br>W=88H f=23H<br>在指令执行之后<br>W=24H f=23H       |
| 例子 2<br>INCF f, 1 | 在指令执行之前<br>W=88H f=23H<br>在指令执行之后<br>W=88H f=24H       |

21

|                                           |                                                                                                                               |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| INCFSZ                                    | f 加 1, 如果结果为 0 跳转                                                                                                             |
| 指令格式                                      | INCFSZ f, d 0<=f<=DFH d=0,1                                                                                                   |
| 操作                                        | (目的地址)<←(f)+1 如果结果为 0 就跳转                                                                                                     |
| 标志位                                       | 无                                                                                                                             |
| 描述                                        | f 的内容加 1。<br>如果 d 为 0, 结果保存到工作寄存器中。<br>如果 d 为 1, 结果保存到 f 中<br>如果结果为 0, 下一条已经取到的指令将被丢掉, 然后插入一条 NOP 指令组成一个两个周期的指令。              |
| 周期                                        | 无跳转则为 1 个指令周期, 否则 2 个指令周期                                                                                                     |
| 例子<br>Node INCFSZ FLAG, 1<br>OP1:<br>OP2: | 在指令执行之前<br>PC=address(Node)<br>在指令执行之后<br>(FLAFG)=(FLAG)+1<br>If(FLAG)=0<br>PC=address(OP2)<br>If(FLAG)!=0<br>PC=address(OP1) |

22

|                 |                            |
|-----------------|----------------------------|
| IORLW           | 工作寄存器与立即数或                 |
| 指令格式            | IORLW K 0<=K<=FFH          |
| 操作              | (W)<←(W) K                 |
| 标志位             | Z                          |
| 描述              | 立即数与工作寄存器的内容或。结果保存到工作寄存器中。 |
| 周期              | 1                          |
| 例子<br>IORLW 85H | 在指令执行之前<br>W=69H           |

|  |                  |
|--|------------------|
|  | 在指令执行之后<br>W=EDH |
|--|------------------|

23

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| IORWF           | f 与工作寄存器或                                                   |
| 指令格式            | IORWF f, d 0<=f<=DFH d=0,1                                  |
| 操作              | (目的地址)<←(W) (f)                                             |
| 标志位             | Z                                                           |
| 描述              | f 和工作寄存器或<br>当 d 为 0 时, 结果保存到工作寄存器中<br>当 d 为 1 时, 结果保存到 f 中 |
| 周期              | 1                                                           |
| 例子<br>IORWF f,1 | 在指令执行前<br>W=88H f=23H<br>在指令执行后<br>W=88H f=ABH              |

24

|               |                                                  |
|---------------|--------------------------------------------------|
| MOVFW         | 传送到工作寄存器                                         |
| 指令格式          | MOVFW f 0<=f<=DFH                                |
| 操作            | (W)<←(f)                                         |
| 标志位           | 无                                                |
| 描述            | 将数据从 f 传送到工作寄存器                                  |
| 周期            | 1                                                |
| 例子<br>MOVFW f | 在指令执行之前<br>W=88H f=23H<br>在指令执行之后<br>W=23H f=23H |

25

|                 |                                      |
|-----------------|--------------------------------------|
| MOVLW           | 将立即数传送到工作寄存器中                        |
| 指令格式            | MOVLW K 0<=K<=FFH                    |
| 操作              | (W)<←K                               |
| 标志位             | 无                                    |
| 描述              | 将 8bit 的立即数传送到工作寄存器中                 |
| 周期              | 1                                    |
| 例子<br>MOVLW 23H | 在指令执行之前<br>W=88H<br>在指令执行之后<br>W=23H |



26

|                        |                                                                                               |
|------------------------|-----------------------------------------------------------------------------------------------|
| MOV <sub>P</sub>       | 读 EPROM 数据                                                                                    |
| 指令格式                   | MOV <sub>P</sub>                                                                              |
| 操作                     | 把 EPROM 数据读到 EDATH/WORK 中                                                                     |
| 标志位                    | 无                                                                                             |
| 描述                     | 把地址为 EADRH/EADRL 的 EPROM 数据读到 EDATH/WORK 中                                                    |
| 周期                     | 2                                                                                             |
| 例子<br>MOV <sub>P</sub> | 在指令执行之前<br>EADRH=04H, EADRL=00H<br>地址为 0400H 的 EPROM 数据位 1234H<br>在指令执行之后<br>EDATH=12H, W=34H |

27

|               |                                                  |
|---------------|--------------------------------------------------|
| MOVWF         | 将工作寄存器的值传送到 f 中                                  |
| 指令格式          | MOVWF f 0<=f<=DFH                                |
| 操作            | (f)<←(W)                                         |
| 标志位           | 无                                                |
| 描述            | 将工作寄存器的值传送到 f 中                                  |
| 周期            | 1                                                |
| 例子<br>MOVWF f | 在指令执行之前<br>W=88H f=23H<br>在指令执行之后<br>W=88H f=88H |

28

|      |     |
|------|-----|
| NOP  | 无操作 |
| 指令格式 | NOP |
| 操作   | 无操作 |
| 标志位  | 无   |
| 描述   | 无操作 |
| 周期   | 1   |

29

|      |                                                                                 |
|------|---------------------------------------------------------------------------------|
| PUSH | 把 work 和 status 寄存器入栈保护                                                         |
| 指令格式 | PUSH                                                                            |
| 操作   | (top stack)<←work/status                                                        |
| 标志位  | 无                                                                               |
| 描述   | 把 work 和 status 寄存器的值做入栈处理，支持 6 级堆栈，不同于 PC 堆栈；其中状态寄存器不包括 LVD36, LVD24, PD 和 TO。 |
| 周期   | 2                                                                               |

30

|      |                                                                                          |
|------|------------------------------------------------------------------------------------------|
| POP  | 把 work 和 status 寄存器出栈处理                                                                  |
| 指令格式 | POP                                                                                      |
| 操作   | (Top Stack) $\Rightarrow$ work/status<br>Pop Stack                                       |
| 标志位  | 无                                                                                        |
| 描述   | 把当前栈顶的值做出栈处理，分别更新 work 和 status 寄存器，支持 6 级堆栈，不同于 PC 堆栈；其中状态寄存器不包括 LVD36, LVD24, PD 和 TO。 |
| 周期   | 2                                                                                        |

31

|        |                                                                  |
|--------|------------------------------------------------------------------|
| RETFIE | 从中断返回                                                            |
| 指令格式   | RETFIE                                                           |
| 操作     | (Top Stack) $\Rightarrow$ PC<br>Pop Stack<br>1 $\Rightarrow$ GIE |
| 标志位    | 无                                                                |
| 描述     | PC 从堆栈顶部得到，然后出栈，设置全局中断使能位为 1                                     |
| 周期     | 2                                                                |

32

|       |                                                                 |
|-------|-----------------------------------------------------------------|
| RETLW | 返回，并将立即数送到工作寄存器中                                                |
| 指令格式  | RETLW K 0 $\leq$ K $\leq$ FFH                                   |
| 操作    | (W) $\leftarrow$ K<br>(Top Stack) $\Rightarrow$ PC<br>Pop Stack |
| 标志位   | 无                                                               |
| 描述    | 将 8bit 的立即数送到工作寄存器中，PC 值从栈顶得到，然后出栈                              |
| 周期    | 2                                                               |

33

|        |                                           |
|--------|-------------------------------------------|
| RETURN | 从子程序返回                                    |
| 指令格式   | RETURN                                    |
| 操作     | (Top Stack) $\Rightarrow$ PC<br>Pop Stack |
| 标志位    | 无                                         |
| 描述     | PC 值从栈顶得到，然后出栈                            |
| 周期     | 2                                         |

34

|                |                                                          |
|----------------|----------------------------------------------------------|
| RLF            | 带进位左移                                                    |
| 指令格式           | RLF f, d 0<=f<=DFH d=0,1                                 |
| 操作             | (目标地址[n+1])<←(f[n])<br>(目标地址[0])<←C<br>C<←(f[7])         |
| 标志位            | C, Z                                                     |
| 描述             | F 带进位左移一位<br>如果 d 为 0, 结果保存到工作寄存器<br>如果 d 为 1, 结果保存到 f 中 |
| 周期             | 1                                                        |
| 例子<br>RLF f, 1 | 在指令执行之前<br>C=0 W=88H f=E6H<br>在指令执行之后<br>C=1 W=88H f=CCH |

35

|                |                                                          |
|----------------|----------------------------------------------------------|
| RRF            | 带进位右移                                                    |
| 指令格式           | RRF f, d 0<=f<=DFH d=0,1                                 |
| 操作             | (目标地址[n-1])<←(f[n])<br>(目标地址[7])<←C<br>C<←(f[7])         |
| 标志位            | C                                                        |
| 描述             | F 带进位右移一位<br>如果 d 为 0, 结果保存到工作寄存器<br>如果 d 为 1, 结果保存到 f 中 |
| 周期             | 1                                                        |
| 例子<br>RRF f, 0 | 在指令执行之前<br>C=0 W=88H f=95H<br>在指令执行之后<br>C=1 W=4AH f=95H |

36

|       |                        |
|-------|------------------------|
| SLEEP | 晶振停止                   |
| 指令格式  | SLEEP                  |
| 操作    | CPU 晶振停止               |
| 标志位   | PD                     |
| 描述    | CPU 晶振停止。CPU 通过外部中断源重启 |
| 周期    | 1                      |

37

|                   |                                                              |
|-------------------|--------------------------------------------------------------|
| SUBLW             | 立即数减工作寄存器的值                                                  |
| 指令格式              | SUBLW K 0<=K<=FFH                                            |
| 操作                | (W)<-K-(W)                                                   |
| 标志位               | C, DC, Z                                                     |
| 描述                | 8bit 的立即数减去工作寄存器的值，结果保存到工作寄存器中                               |
| 周期                | 1                                                            |
| 例子<br>SUBLW 02H   | 在指令执行之前<br>W=01H<br>在指令执行之后<br>W=01H C=1(代表没有借位) Z=0(代表结果非零) |
| 例子 2<br>SUBLW 02H | 在指令执行之前<br>W=02H<br>在指令执行之后<br>W=00H C=1(代表没有借位) Z=1(代表结果为零) |
| 例子 2<br>SUBLW 02H | 在指令执行之前<br>W=03H<br>在指令执行之后<br>W=FFH C=0(代表有借位) Z=0(代表结果非零)  |

38

|                    |                                                             |
|--------------------|-------------------------------------------------------------|
| SUBWF              | f 的值减工作寄存器的值                                                |
| 指令格式               | SUBWF f, d 0<=f<=DFH d=0,1                                  |
| 操作                 | (目标地址)<-(f)-(W)                                             |
| 标志位                | C, DC, Z                                                    |
| 描述                 | f 的值减去工作寄存器的值。<br>如果 d 为 0，结果保存到工作寄存器<br>如果 d 为 1，结果保存到 f 中 |
| 周期                 | 1                                                           |
| 例子<br>SUBWF f, 1   | 在指令执行之前<br>f=33H W=01H<br>在指令执行之后<br>f=32H C=1 Z=0          |
| 例子 2<br>SUBWF f, 1 | 在指令执行之前<br>f=01H W=01H<br>在指令执行之后<br>f=00H C=1 Z=1          |
| 例子 3<br>SUBWF f, 1 | 在指令执行之前<br>f=04H W=05H<br>在指令执行之后<br>f=FFH C=0 Z=0          |

39

|                     |                                                              |
|---------------------|--------------------------------------------------------------|
| SUBWFC              | 带借位的减法                                                       |
| 指令格式                | SUBWFC f, d 0<=f<=DFH d=0,1                                  |
| 操作                  | (目标地址)<←(f)-(W)-1+C                                          |
| 标志位                 | C, DC, Z                                                     |
| 描述                  | f 的值减去工作寄存器的值<br>如果 d 为 0, 结果保存到工作寄存器<br>如果 d 为 1, 结果保存到 f 中 |
| 周期                  | 1                                                            |
| 例子<br>SUBWFC f, 1   | 在指令执行之前<br>W=01H f=33H C=1<br>在指令执行之后<br>f=32H C=1 Z=0       |
| 例子 2<br>SUBWFC f, 1 | 在指令执行之前<br>W=01H f=02H C=0<br>在指令执行之后<br>f=00H C=1 Z=1       |
| 例子 3<br>SUBWFC f, 1 | 在指令执行之前<br>W=05H f=04H C=0<br>在指令执行之后<br>f=FEH C=0 Z=0       |

40

|                 |                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------|
| SWAPF           | 交换寄存器的值                                                                                           |
| 指令格式            | SWAPF f, d 0<=f<=DFH d=0,1                                                                        |
| 操作              | (des[3:0])<←f[7:4]<br>(des[7:4])<←f[3:0]                                                          |
| 标志位             | 无                                                                                                 |
| 描述              | 把 f 寄存器的高 4 位数据给目标寄存器的低 4 位;<br>把 f 寄存器的低位数据给目标寄存器的高 4 位<br>d 为 1 时, f 寄存器为目标寄存器; 否则, w 寄存器为目标寄存器 |
| 周期              | 1                                                                                                 |
| 例子<br>SWAPF f,1 | 在指令执行之前<br>f=ACH<br>在指令执行之后<br>f=CAH                                                              |

41

|              |                                                              |
|--------------|--------------------------------------------------------------|
| TBLP         | 写 EPROM                                                      |
| 指令格式         | TBLP k (k 取 0)                                               |
| 操作           | 写 EPROM                                                      |
| 标志位          | 无                                                            |
| 描述           | 把 EDATH/WORK 的数据写到地址为 EADRH/EADRL 的 E2PROM 里                 |
| 周期           | 约为 3ms 时间                                                    |
| 例子<br>TBLP 0 | 在指令执行之前<br>EDATH=BAH, W=ACH, EADRH=04H, EADRL=00H<br>在指令执行之后 |

|                                |
|--------------------------------|
| 把 BAACH 写到地址为 0400H 的 E2PROM 里 |
|--------------------------------|

42

|                 |                                      |
|-----------------|--------------------------------------|
| XORLW           | 工作寄存器的值与立即数异或                        |
| 指令格式            | XORLW K 0<=K<=FFH                    |
| 操作              | $(W) \leftarrow (W) \wedge K$        |
| 标志位             | Z                                    |
| 描述              | 8bit 的立即数与工作寄存器的值异或，结果保存在工作寄存器中      |
| 周期              | 1                                    |
| 例子<br>XORLW 5FH | 在指令执行之前<br>W=ACH<br>在指令执行之后<br>W=F3H |

43

|                  |                                                                 |
|------------------|-----------------------------------------------------------------|
| XORWF            | f 的值与工作寄存器的值异或                                                  |
| 指令格式             | XORWF f, d 0<=f<=DFH d=0,1                                      |
| 操作               | (目标地址) $\leftarrow (W) \wedge (f)$                              |
| 标志位              | Z                                                               |
| 描述               | F 的值与工作寄存器的值异或，<br>当 d 为 0 时，结果保存到工作寄存器中<br>当 d 为 1 时，结果保存到 f 中 |
| 周期               | 1                                                               |
| 例子<br>XORWF f, 1 | 在指令执行之前<br>W=ACH f=5FH<br>在指令执行之后<br>f=F3H                      |

## 5 电气特性

### 5.1 极限值

| 参数       | 范围           | 单位 |
|----------|--------------|----|
| 电源 VDD   | 2.3~6.0      | V  |
| 引脚输入电压   | -0.3~VDD+0.3 | V  |
| 工作温度     | -40~+85      | °C |
| 存贮温度     | -55~+150     | °C |
| 焊接温度, 时间 | 220°C, 10 秒  |    |

### 5.2 直流特性 (VDD = 5V, T<sub>A</sub> = 25°C, 如无其他说明则都是此条件)

| 符号     | 参数           | 测试条件                         | 最小值     | 典型值  | 最大值    | 单位   |
|--------|--------------|------------------------------|---------|------|--------|------|
| VDD    | 工作电压         | 25 °C                        | 2.3     | 5    | 5.5    | V    |
|        |              | -40 °C ~+85 °C               | 2.5     | 5    | 5.5    | V    |
| FXT    | 外部晶振频率       |                              |         |      | 16     | MHz  |
| EXT_CK | 外部灌入时钟       | 25 °C, 代码选项 XTAL_PIN 选 ERC   |         |      | 16     | MHz  |
|        |              | 25 °C, 代码选项 XTAL_PIN 选外部低速晶振 |         |      | 50     | KHz  |
|        |              | 25 °C, 代码选项 XTAL_PIN 选外部高速晶振 |         |      | 4      | MHz  |
| Vpor   | 系统电源电压上升速率   |                              | 0.15    |      |        | V/ms |
| Tcpu   | 指令周期         | VDD: 2.3V~5.5V               | 2000    |      |        | ns   |
|        |              | VDD: 2.4V~5.5V               | 500     |      |        |      |
|        |              | VDD: 3.6V~5.5V               | 250     |      |        |      |
| VIH    | 数字输入高电平      | PT1, PT3, PT5                | 0.75VDD |      |        | V    |
|        | 复位输入高电平      |                              | 0.8VDD  |      |        |      |
| VIL    | 数字输入低电平      | PT1, PT3, PT5                |         |      | 0.3VDD | V    |
|        | 复位输入低电平      |                              |         |      | 0.2VDD |      |
| IPU    | 上拉电流         | PT1,PT3,PT5; Vin = 0;        |         | 50   |        | uA   |
| IOH    | 高电平输出电流      | VOH=0.9VDD; VDD=5V           |         | 9.5  |        | mA   |
|        |              | VOH=0.9VDD; VDD=3V           |         | 4.5  |        | mA   |
| IOL    | 低电平输出电流      | VOL=0.1VDD; VDD=5V           |         | 13.5 |        | mA   |
|        |              | VOL=0.1VDD; VDD=3V           |         | 6.8  |        | mA   |
| LVD    | 复位电压/低电压检测电压 | 2.0V 上电/掉电复位点; 25 度          | 1.9     | 2.0  | 2.3    | V    |
|        |              | 2.0V 上电/掉电复位点; -40~85 度      | 1.7     | 2.0  | 2.5    |      |
|        |              | 2.4V 上电/掉电复位点; 25 度          | 2.3     | 2.4  | 2.7    |      |
|        |              | 2.4V 上电/掉电复位点; -40~85 度      | 2.0     | 2.4  | 3.0    |      |



|         |            |                                               |          |          |       |       |    |
|---------|------------|-----------------------------------------------|----------|----------|-------|-------|----|
|         |            | 3.6V 上电/掉电复位点; 25 度                           | 3.3      | 3.6      | 3.9   |       |    |
|         |            | 3.6V 上电/掉电复位点; -40~85 度                       | 3.0      | 3.6      | 4.5   |       |    |
| IRC     | 内置 RC 时钟   | 25°C, 5V                                      | 15.68    | 16.0     | 16.32 | MHz   |    |
|         |            | -40°C~85°C, 2.3V~5.5V                         | 15.2     | 16.0     | 16.8  |       |    |
| WDT     | 内置看门狗时钟    | 25°C, 5V                                      | 29       | 32       | 35    | KHz   |    |
|         |            | -40°C~85°C, 2.3V~5.5V                         | 26       | 32       | 38    |       |    |
| Tint0,1 | 中断触发脉宽     | 25°C, 5V                                      | Tcpu     |          |       | ns    |    |
| IDD1    | sleep 模式电流 | VDD=3V, 关掉 WDT                                |          | 0.6      |       | uA    |    |
|         |            | VDD=3V, 打开 WDT                                |          | 2.5      |       | uA    |    |
|         |            | VDD=5V, 关掉 WDT                                |          | 0.8      |       | uA    |    |
|         |            | VDD=5V, 打开 WDT                                |          | 3.5      |       | uA    |    |
| IDD2    | 工作电流       | 内部振荡器关闭<br>(fcpu=fosc/4)<br>fosc = 32768Hz,3V |          | 8        |       | uA    |    |
|         |            | 内部振荡器关闭<br>(fcpu=fosc/4)<br>fosc = 32768Hz,5V |          | 13       |       |       |    |
| IDD3    | 工作电流       | 内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/4          |          | 2.5      |       | mA    |    |
|         |            | 内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/8          |          | 1.5      |       |       |    |
|         |            | 内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/16         |          | 0.85     |       |       |    |
|         |            | 内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/32         |          | 0.6      |       |       |    |
|         |            | 内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/4          |          | 4.0      |       |       |    |
|         |            | 内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/8          |          | 2.23     |       |       |    |
|         |            | 内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/16         |          | 1.39     |       |       |    |
|         |            | 内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/32         |          | 0.92     |       |       |    |
| IDD4    | 工作电流       | ERC 模式, VDD=5V, fcpu=fosc/4                   |          |          |       |       | mA |
|         |            | C=20P                                         | R=1K Ω   | F=8.8MHZ |       | 5.05  |    |
|         |            |                                               | R=3.3K Ω | F=7.2MHZ |       | 3.0   |    |
|         |            |                                               | R=10K Ω  | F=4.3MHZ |       | 1.6   |    |
|         |            |                                               | R=100K Ω | F=700KHZ |       | 0.24  |    |
|         |            |                                               | R=300K Ω | F=260KHZ |       | 0.09  |    |
|         |            | C=100P                                        | R=1K Ω   | F=9.3MHZ |       | 5.3   |    |
|         |            |                                               | R=3.3K Ω | F=7.7MHZ |       | 3.3   |    |
|         |            |                                               | R=10K Ω  | F=3.3MHZ |       | 1.3   |    |
|         |            |                                               | R=100K Ω | F=420KHZ |       | 0.17  |    |
|         |            |                                               | R=300K Ω | F=160KHZ |       | 0.066 |    |
|         |            | C=470P                                        | R=1K Ω   | F=9.3MHZ |       | 5.2   |    |

|  |  |                             |                 |          |  |       |  |  |  |
|--|--|-----------------------------|-----------------|----------|--|-------|--|--|--|
|  |  |                             | R=3.3K $\Omega$ | F=8.5MHZ |  | 3.4   |  |  |  |
|  |  |                             | R=10K $\Omega$  | F=3.3MHZ |  | 1.3   |  |  |  |
|  |  |                             | R=100K $\Omega$ | F=410KHZ |  | 0.16  |  |  |  |
|  |  |                             | R=300K $\Omega$ | F=160KHZ |  | 0.065 |  |  |  |
|  |  | ERC 模式, VDD=3V, fcpu=fosc/4 |                 |          |  |       |  |  |  |
|  |  | C=20P                       | R=1K $\Omega$   | F=7.1MHZ |  | 2.6   |  |  |  |
|  |  |                             | R=3.3K $\Omega$ | F=5.4MHZ |  | 1.5   |  |  |  |
|  |  |                             | R=10K $\Omega$  | F=2.8MHZ |  | 0.7   |  |  |  |
|  |  |                             | R=100K $\Omega$ | F=400KHZ |  | 0.098 |  |  |  |
|  |  |                             | R=300K $\Omega$ | F=150KHZ |  | 0.039 |  |  |  |
|  |  | C=100P                      | R=1K $\Omega$   | F=7.4MHZ |  | 2.67  |  |  |  |
|  |  |                             | R=3.3K $\Omega$ | F=4.6MHZ |  | 1.35  |  |  |  |
|  |  |                             | R=10K $\Omega$  | F=1.9MHZ |  | 0.55  |  |  |  |
|  |  |                             | R=100K $\Omega$ | F=250KHZ |  | 0.068 |  |  |  |
|  |  |                             | R=300K $\Omega$ | F=96KHZ  |  | 0.029 |  |  |  |
|  |  | C=470P                      | R=1K $\Omega$   | F=7.5MHZ |  | 2.7   |  |  |  |
|  |  |                             | R=3.3K $\Omega$ | F=4.6MHZ |  | 1.4   |  |  |  |
|  |  |                             | R=10K $\Omega$  | F=1.8MHZ |  | 0.5   |  |  |  |
|  |  |                             | R=100K $\Omega$ | F=250KHZ |  | 0.07  |  |  |  |
|  |  |                             | R=300K $\Omega$ | F=95KHZ  |  | 0.029 |  |  |  |

### 5.3 ADC特性 (VDD = 5V, T<sub>A</sub> = 25°C, 如无其他说明则都是此条件)

| 符号                        | 参数         | 测试条件                                  | 最小值 | 典型值  | 最大值  | 单位   |
|---------------------------|------------|---------------------------------------|-----|------|------|------|
| VDD                       | ADC 工作电压范围 | 25 °C                                 | 2.3 | 5    | 5.5  | V    |
|                           |            | -40 °C ~+85 °C                        | 2.5 | 5    | 5.5  | V    |
| AIN0~ AIN5 input voltage  | 模拟输入范围     | VREF 受寄存器 VREFS[1:0]控制                | 0   |      | VREF | V    |
| Vref input range          | 外部参考电压输入范围 | VREFS[1:0]=01                         | 0   |      | VDD  | V    |
| ADC current consumption   | ADC 功耗     | VDD=5V(VDD 作为参考电压)                    |     | 0.43 |      | mA   |
|                           |            | VDD=3V(VDD 作为参考电压)                    |     | 0.40 |      | mA   |
| ADC Conversion Cycle Time | ADC 转换周期   |                                       | 3.5 | 10   |      | uS   |
| INL                       | 积分非线性      | SRADACKS[1:0]=01;<br>SRADCKS[1:0]=01; |     | ±4   | ±8   | LSB  |
| No missing code           | 无失码        | VREFS[1:0]=01, 外部参考电压                 | 10  | 11   | 12   | Bits |
|                           |            | VREFS[1:0]=00, VDD 做为参考电压             | 9   | 10   | 11   | Bits |
|                           |            | VREFS[1:0]=10, 内部参考电压                 | 7   | 8    | 9    | Bits |
| IVREF                     | 内部参考电压     |                                       |     | 1.4  |      | V    |
| IVREF temp drift          | 内部参考电压温漂   |                                       |     | 50   |      | ppm  |
| Offset                    | ADC 失调电压   |                                       |     | 6    |      | mV   |

5.4 RC时钟频率特性

下图位实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

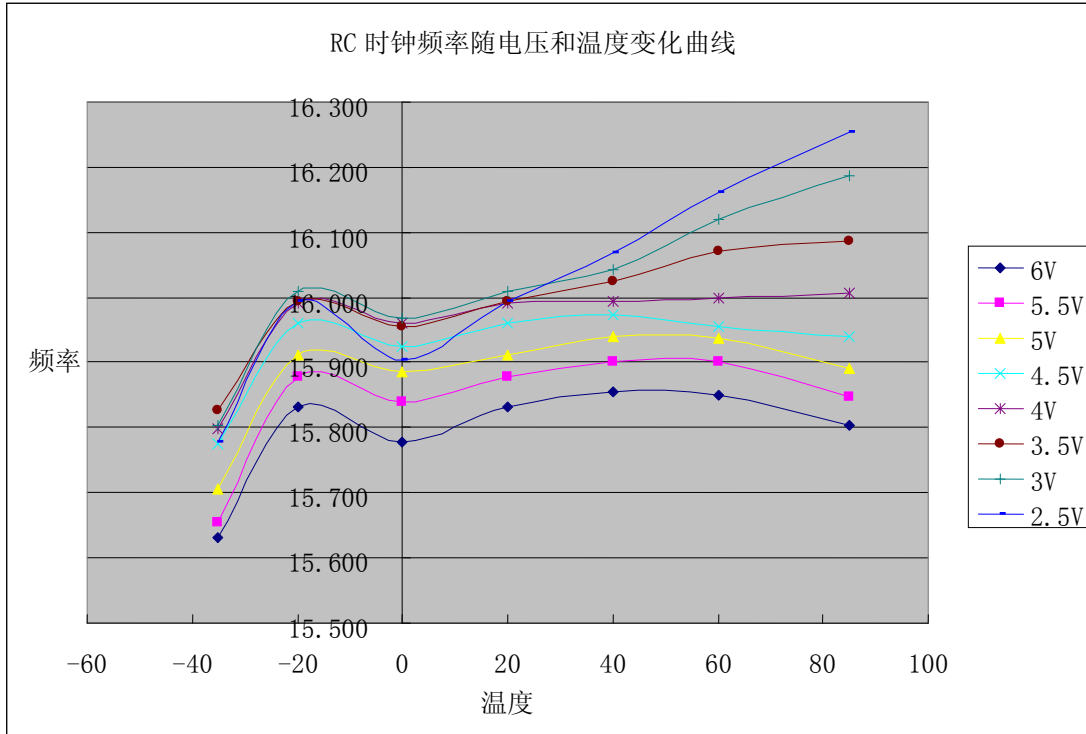


图17 RC 时钟频率的电压和温度特性

5.5 WDT时钟频率特性

下图位实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

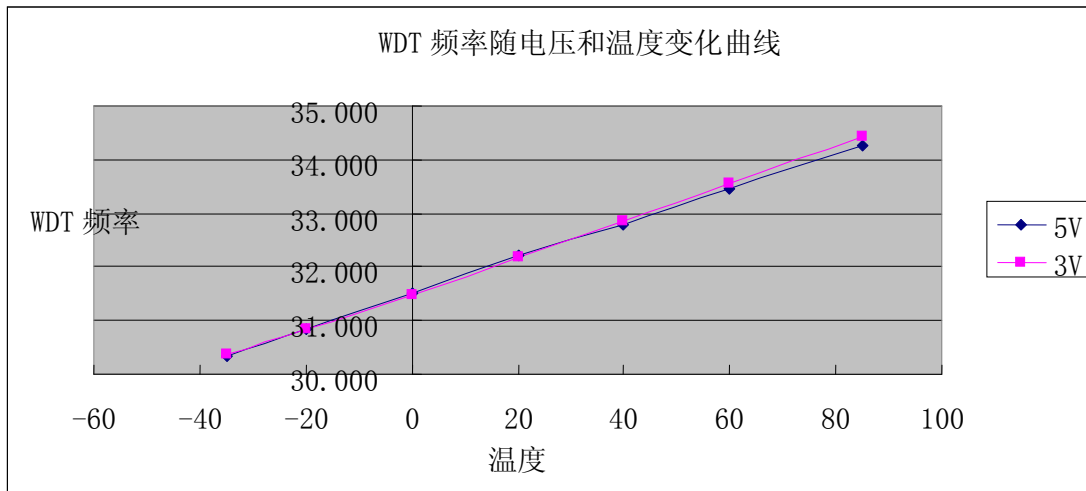


图18 WDT 频率的电压和温度特性

5.6 ERC频率特性

下表是实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

表 38 不同阻容的 ERC 频率

| 电容 (pF) | 电阻 (Kohm) | 电压 (V) | 晶振频率 (kHz) | 电流 (uA) |
|---------|-----------|--------|------------|---------|
| 47      | 1         | 5      | 9756.098   | 5630    |
|         |           | 3      | 8048.290   | 2951    |
|         | 10        | 5      | 3911.980   | 1585    |
|         |           | 3      | 2341.920   | 651     |
|         | 100       | 5      | 550.964    | 216.7   |
|         |           | 3      | 315.956    | 88.2    |
|         | 1000      | 5      | 71.685     | 36.15   |
|         |           | 3      | 41.885     | 18.49   |
| 10000   | 5         | 10.152 | 13.84      |         |
|         | 3         | 5.479  | 9.8        |         |
| 100     | 1         | 5      | 9935.420   | 5678    |
|         |           | 3      | 8106.191   | 2966    |
|         | 10        | 5      | 3629.764   | 1495    |
|         |           | 3      | 2063.983   | 593     |
|         | 100       | 5      | 477.042    | 194.22  |
|         |           | 3      | 282.785    | 81.07   |
|         | 1000      | 5      | 68.493     | 34.93   |
|         |           | 3      | 39.370     | 17.77   |
|         | 10000     | 5      | 10.139     | 13.69   |
|         |           | 3      | 5.281      | 9.6     |

下图是实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

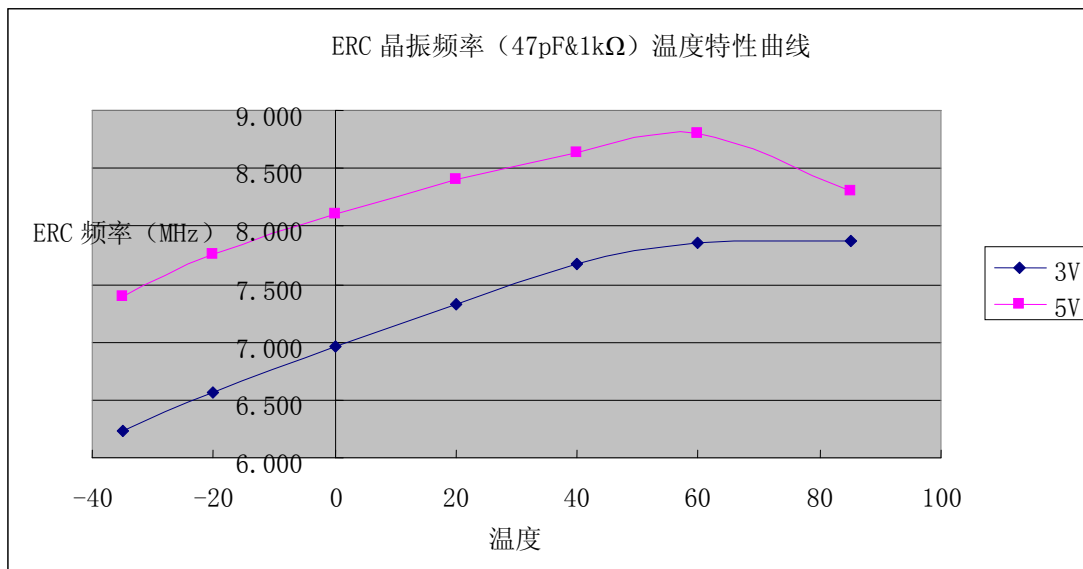
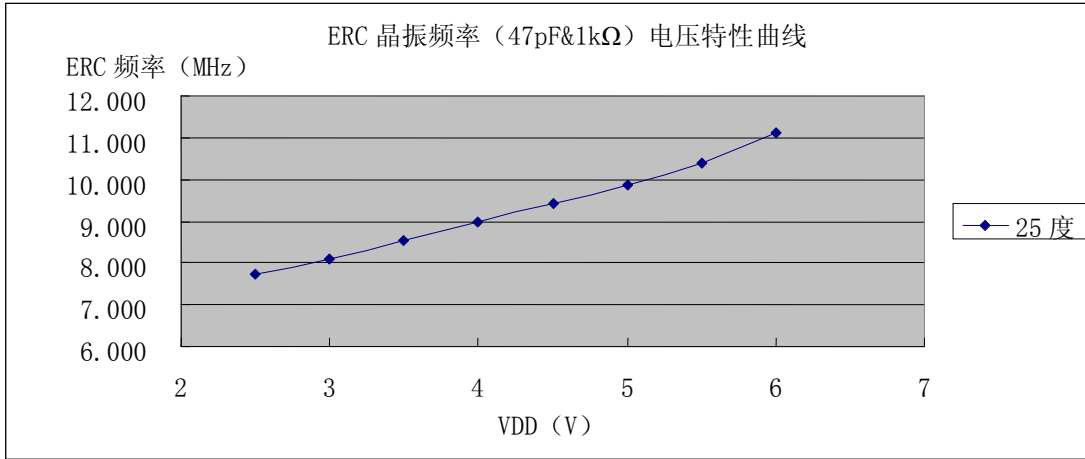


图19 ERC@47pF&1kΩ温度特性



### 5.7 2.0V掉电复位温度特性

下图位实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

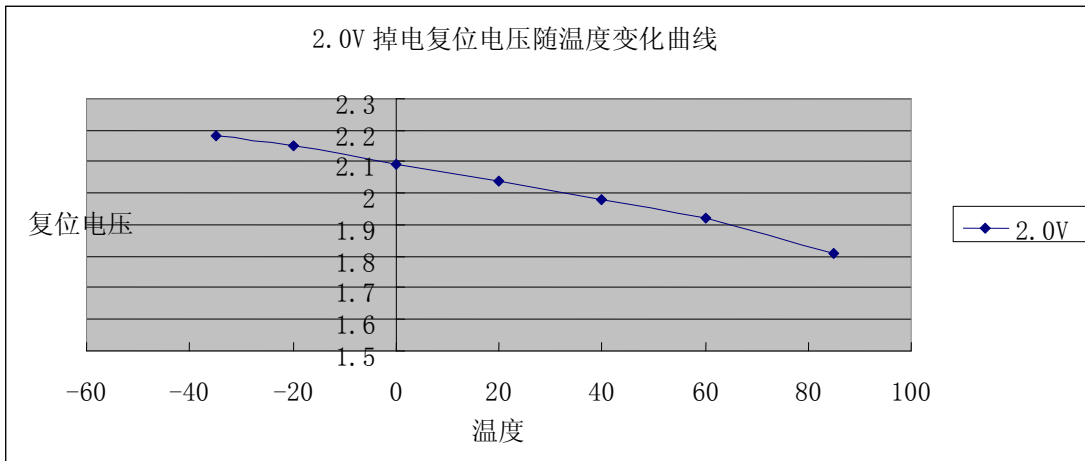


图20 2.0V 掉电复位温度特性

### 5.8 2.4V低电压复位温度特性

下图位实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

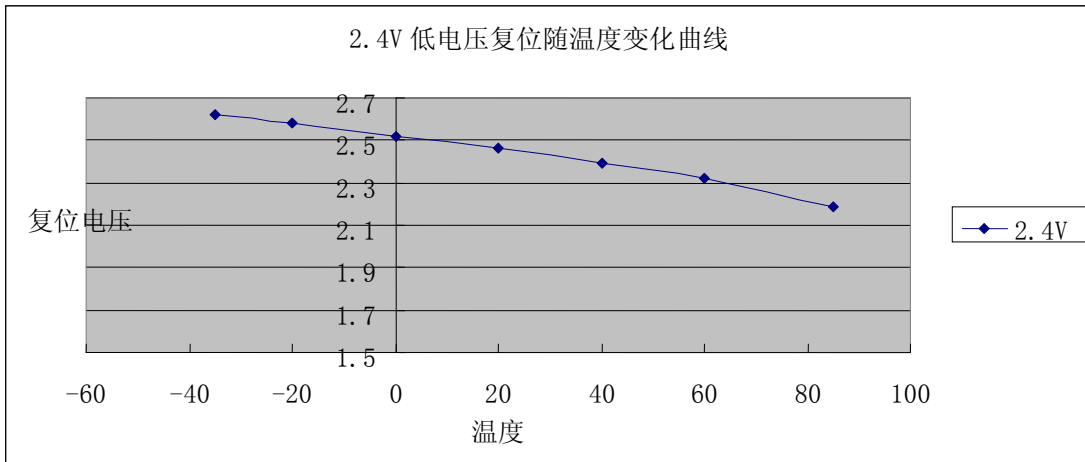


图21 2.4V 低电压复位温度特性

5.9 3.6V低电压复位温度特性

下图位实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

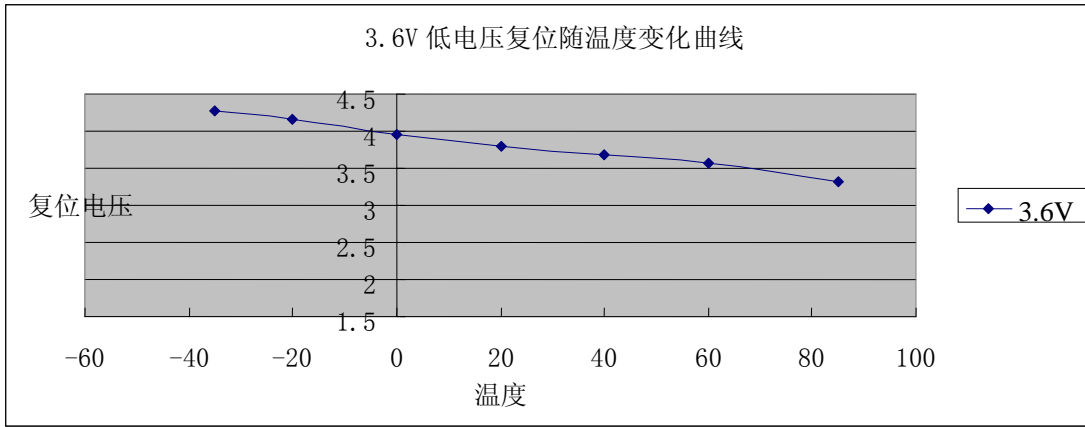


图22 3.6V 低电压复位温度特性

5.10 1.4V内部参考电压温度特性

下图位实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

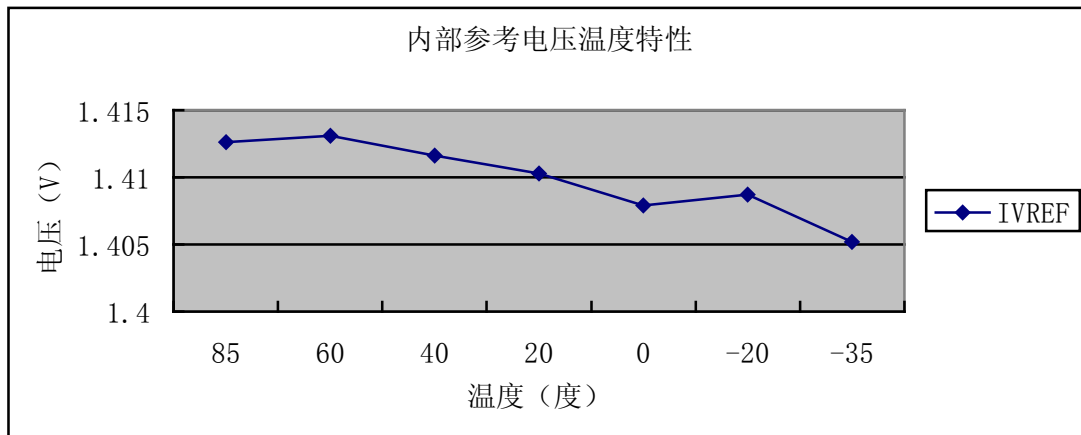
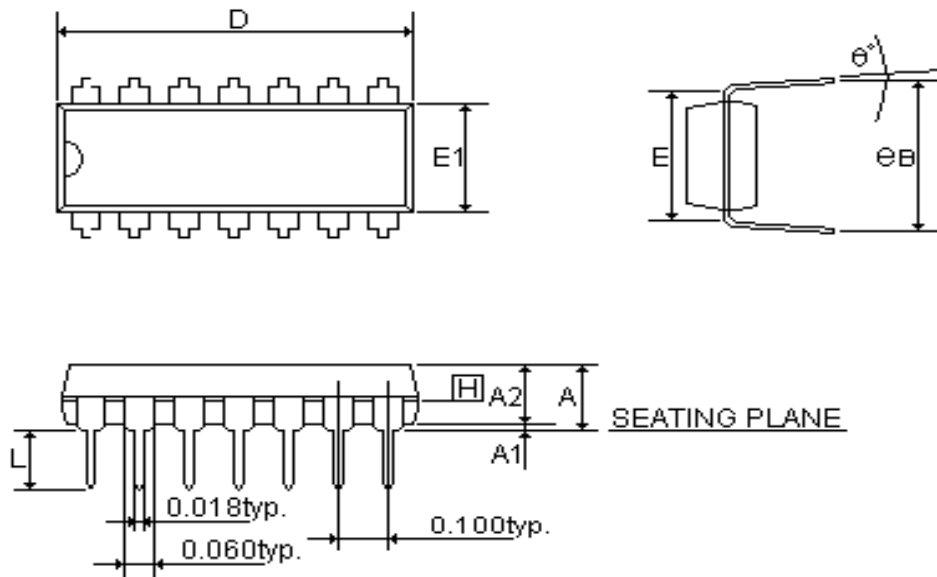


图23 内置参考电压温度特性

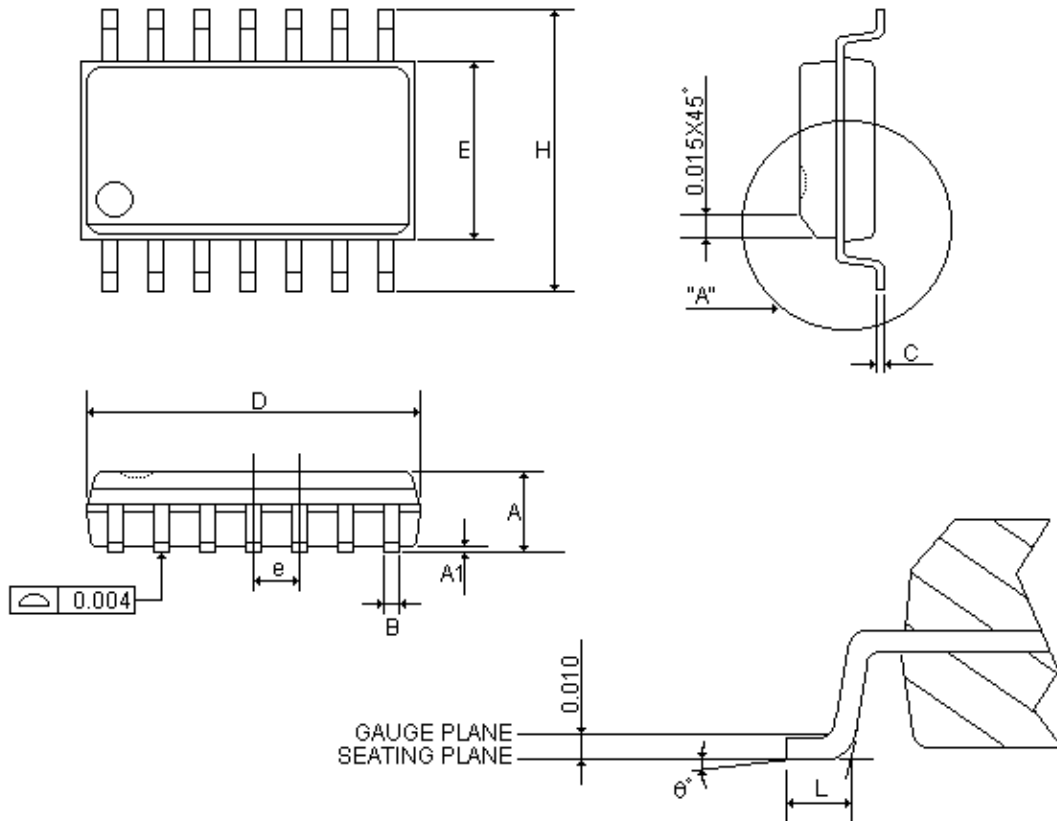
6 封装图

6.1 DIP-14pin



| SYMBOLS        | MIN    | NOR   | MAX   | MIN    | NOR   | MAX    |
|----------------|--------|-------|-------|--------|-------|--------|
|                | (inch) |       |       | (mm)   |       |        |
| A              | -      | -     | 0.210 | -      | -     | 5.334  |
| A1             | 0.015  | -     | -     | 0.381  | -     | -      |
| A2             | 0.125  | 0.130 | 0.135 | 3.175  | 3.302 | 3.429  |
| D              | 0.735  | 0.075 | 0.775 | 18.669 | 1.905 | 19.685 |
| E              | 0.300  |       |       | 7.62   |       |        |
| E1             | 0.245  | 0.250 | 0.255 | 6.223  | 6.35  | 6.477  |
| L              | 0.115  | 0.130 | 0.150 | 2.921  | 3.302 | 3.810  |
| e B            | 0.335  | 0.355 | 0.375 | 8.509  | 9.017 | 9.525  |
| $\theta^\circ$ | 0°     | 7°    | 15°   | 0°     | 7°    | 15°    |

6.2 SOP-14pin



| SYMBOLS        | MIN    | NOR   | MAX    | MIN    | NOR    | MAX    |
|----------------|--------|-------|--------|--------|--------|--------|
|                | (inch) |       |        | (mm)   |        |        |
| A              | 0.058  | 0.064 | 0.068  | 1.4732 | 1.6256 | 1.7272 |
| A1             | 0.004  | -     | 0.010  | 0.1016 | -      | 0.254  |
| B              | 0.013  | 0.016 | 0.020  | 0.3302 | 0.4064 | 0.508  |
| C              | 0.0075 | 0.008 | 0.0098 | 0.1905 | 0.2032 | 0.2490 |
| D              | 0.336  | 0.341 | 0.344  | 8.5344 | 8.6614 | 8.7376 |
| E              | 0.150  | 0.154 | 0.157  | 3.81   | 3.9116 | 3.9878 |
| e              | -      | 0.050 | -      | -      | 1.27   | -      |
| H              | 0.228  | 0.236 | 0.244  | 5.7912 | 5.9944 | 6.1976 |
| L              | 0.015  | 0.025 | 0.050  | 0.381  | 0.635  | 1.27   |
| $\theta^\circ$ | 0°     | -     | 8°     | 0°     | -      | 8°     |