



CSU8RP3235/CSU8RP3236 用户手册

带 12-bit ADC 的 8 位 RISC OTP MCU

REV 1.2

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

微 信 号：芯海科技

微信二维码：



版本历史

历史版本.	修改内容	版本日期
REV 1.0	初始版本	2013-12-1
REV 1.1	增加 PT5.0 和 PT5.1 大驱动描述	2015-3-17
REV 1.2	在 PIN 配置中增加 PT1.3 的使用注意说明	2015-3-17

目 录

版本历史	2
目 录	3
1 产品概述	6
1.1 功能描述	6
1.2 主要特性	6
1.3 PIN 配置	7
2 标准功能	9
2.1 CPU 核	9
2.1.1 存储器	11
2.1.2 状态寄存器	13
2.1.3 SFR	14
2.2 时钟系统	16
2.2.1 概述	16
2.2.2 时钟框图	16
2.2.3 寄存器	17
2.2.4 内部高速 RC 时钟	18
2.2.5 内部低速 WDT 时钟	18
2.2.6 外部高速晶振时钟	18
2.2.7 外部低速晶振时钟	18
2.2.8 外部 RC 振荡器	18
2.2.9 外部时钟源	18
2.3 复位系统	19
2.3.1 上电复位	20
2.3.2 看门狗复位	20
2.3.3 掉电复位	20
2.3.4 外部硬件复位	21
2.4 中断	22
2.4.1 中断使能寄存器	23
2.4.2 中断标志寄存器	24
2.4.3 外部中断 0	25
2.4.4 外部中断 1	25
2.4.5 AD 中断溢出	27
2.4.6 定时器 0 溢出中断	27
2.4.7 定时/计数器 2 溢出中断	27
2.4.8 定时/计数器 3 溢出中断	27
2.4.9 比较器 0 中断	27
2.4.10 PUSH 和 POP 处理	27
2.5 定时器 0	28
2.6 I/O PORT	30
2.6.1 PT1 口	30
2.6.2 PT3 口	33
2.6.3 PT5 口	34
3 增强功能	36
3.1 HALT 和 SLEEP 模式	36
3.2 看门狗(WDT)	38

3.3	定时/计数器 2	40
3.3.1	寄存器说明	40
3.3.2	蜂鸣器	42
3.3.3	PWM.....	42
3.4	定时/计数器 3.....	44
3.4.1	寄存器说明	44
3.4.2	蜂鸣器	46
3.4.3	PWM.....	46
3.5	模数转换器 (ADC)	48
3.5.1	寄存器说明	48
3.5.2	转换时间	52
3.5.3	使用内部参考电压的校准方法	54
3.5.4	AD 失调电压校正.....	56
3.5.5	数字比较器	56
3.5.6	内部测量 VDD 的电压.....	58
3.6	比较器/运算放大器	60
3.6.1	比较器/运算放大器	60
3.6.2	比较器参考电压	62
3.6.3	比较器中断	62
3.7	4*5 LCD	63
3.8	输入逻辑电平电压配置	64
3.9	烧录模块	65
3.10	代码选项	66
4	MCU 指令集.....	67
5	电气特性	82
5.1	极限值	82
5.2	直流特性 (VDD = 5V, T _A = 25°C, 如无其他说明则都是此条件)	82
5.3	ADC 特性 (VDD = 5V, T _A = 25°C, 如无其他说明则都是此条件)	84
5.4	比较器/运算放大器的直流特性 (VDD = 5V, T _A = 25°C, 如无其他说明则都是此条件)	84
5.5	LCD 直流特性 (VDD = 5V, T _A = 25°C, 如无其他说明则都是此条件)	85
5.6	16MHz IHRC 时钟频率特性	86
5.7	8MHz IHRC 时钟频率特性	86
5.8	4MHz IHRC 时钟频率特性	86
5.9	2MHz IHRC 时钟频率特性	87
5.10	32KHz WDT 时钟频率特性	87
5.11	2.0V 掉电复位温度特性	88
5.12	2.4V 低电压复位温度特性	88
5.13	3.0V 低电压复位温度特性	88
5.14	3.6V 低电压复位温度特性	89
5.15	2.0V 内部参考电压温度特性	89
5.16	3.0V 内部参考电压温度特性	89
5.17	4.0V 内部参考电压温度特性	90
6	封装图	91
6.1	SOP-16PIN	91
6.2	DIP-16PIN	92
6.3	SOP-20PIN	93
6.4	DIP-20PIN	94
7	单片机产品命名规则	95

7.1	产品型号说明	95
7.2	命名举例说明	96
7.3	产品印字说明	96

1 产品概述

1.1 功能描述

CSU8RP3235/CSU8RP3236 是带 12-bit ADC 的 8 位 RISC MCU，内置 2K×16 位 OTP 程序存储器。

1.2 主要特性

高性能的 RISC CPU

- 8 位单片机 MCU
- 2K×16 位 OTP 程序存储器
- 256 字节数据存储器 (SRAM)
- 41 条单字指令
- 8 级 PC 存储堆栈
- 8 级 PUSH 和 POP 存储堆栈

振荡器

- 内带 IHRC(16/8/4/2MHz)和 ILRC(32KHz)振荡器，IHRC 精度为±1%@5V，25°C，ILRC 精度为±2%@5V，25°C
- 外部 32768Hz 晶振 (RTC) 或 2MHz~16MHz 晶振或 ERC 8M@5V DC，ERC 4M@3V DC

外设特性

- 17 位双向 I/O 口,1 位输入口
- 2 路蜂鸣器输出，2 路 PWM 输出
- 5 个内部中断，2 个外部中断
- 8 个具有唤醒功能的输入口
- 5 路 12bitADC
 - 内部 2.0V、3.0V、4.0V、VDD、外部输入 5 种参考电压选择
 - 带数字比较器
- 1 个内置模拟比较器/运算放大器
- 提供 1 个在 2.0V，3.0V 和 4.0V 之间可选的参考电压，2.0V 和 3.0V 精度为±1%@5V，25°C
- 低电压检测 (LVD) 引脚，内部提供 2.4V、3.0V、3.6V 电压检测
- 6 个开漏输出 (P1.1、P5.0~P5.4；外部上拉电压不高于 VDD+0.3V)
- PT5.0 和 PT5.1 可配置为大驱动输出，驱动

电流高达 40mA

- 所有 IO 可配置上拉电阻，同时 PT5.0 和 PT1.4 还支持下拉电阻配置。
- 输入逻辑电平电压可配置，在 5V 工作电压下可支持与 3V 器件通信
- 4*5 LCD 驱动，1/4duty,1/2bias

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 (98ms)
- 内带 2.0V、2.4V、3.0V 和 3.6V 低电压复位 (LVR)
- 定时器 0
 - 可编程预分频的 8 位的定时计数器
- 定时/计数器 2
 - 可编程预分频的 12 位的分频器
- 定时/计数器 3
 - 可编程预分频的 12 位的分频器
- 扩展型看门狗定时器 (32KHz WDT)
 - 可编程的时间范围

CMOS 技术

- 工作电压范围
 - 2.2V~5.5V@fcpu=4MHz (指令周期)
- 工作温度范围
 - -40~85°C (工业级)

低功耗特性

- MCU 工作电流
 - 正常模式 1mA@4MHz (工作电压 3V)
 - 正常模式 10uA@32KHz (工作电压 3V)
 - 休眠模式下的电流小于 1μA

封装

- SOP16/DIP16
- SOP20/DIP20

应用范围

- 移动电源
- 小家电

型号	ROM	RAM	堆栈 IO	Wake-up	8/12 位定时器	12 位 PWM/Buzzer	ADC (Ch*Bit)	比较器 /运放	LCD	封装
CSU8RP3235	2K*16	256	8 14	4	1/2	2	5*12	1/1	/	SOP16/DIP16
CSU8RP3236	2K*16	256	8 18	8	1/2	2	5*12	1/1	4*5	SOP20/DIP20

1.3 PIN 配置

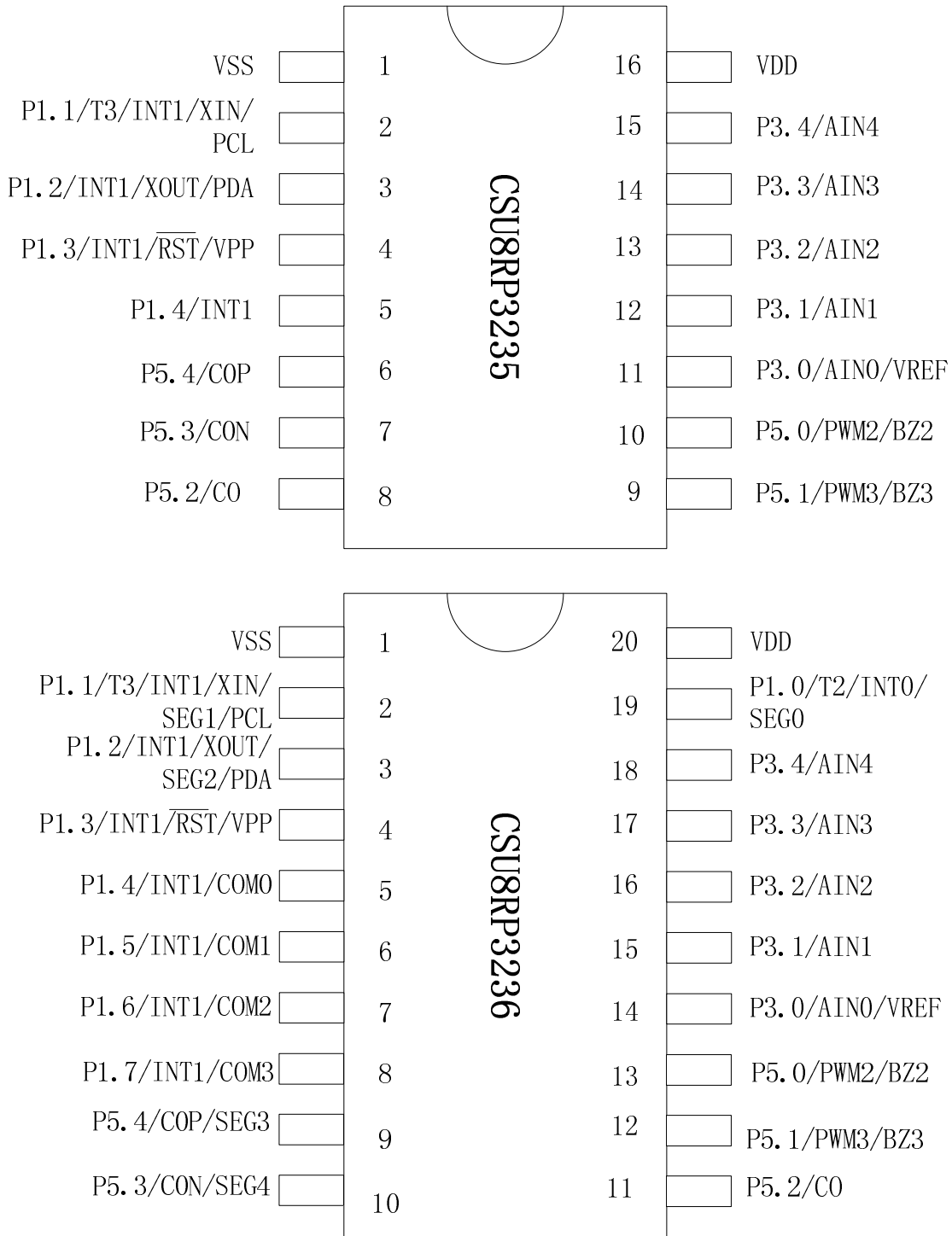


图1 配置图

表 1 引脚说明表

管脚名称 (CSU8RP3236)	输入/输出	管脚序号	描述
VSS	P	1	地
P1.1/T3/INT1/XIN/SEG1/PCL	I/O	2	IO, 具有开漏输出功能; 定时/计数器 3 外部输入; 外部中断 1 输入, 具有唤醒功能; 外置晶振输入; LCD

			驱动 SEG1 输出；烧录时钟线
P1.2/INT1/XOUT/SEG2/PDA	I/O	3	IO；外部中断 1 输入，具有唤醒功能；外置晶振输出；LCD 驱动 SEG2 输出；烧录数据线
P1.3/INT1/ \overline{RST} /VPP	I	4	普通输入口(默认开上拉)；外部中断 1 输入，具有唤醒功能；复位输入；烧录电压
P1.4/INT1/COM0	I/O	5	IO；外部中断 1 输入，具有唤醒功能；LCD 驱动 COM0 输出；
P1.5/INT1/COM1	I/O	6	IO；外部中断 1 输入，具有唤醒功能；LCD 驱动 COM1 输出；
P1.6/INT1/COM2	I/O	7	IO；外部中断 1 输入，具有唤醒功能；LCD 驱动 COM2 输出；
P1.7/INT1/COM3	I/O	8	IO；外部中断 1 输入，具有唤醒功能；LCD 驱动 COM3 输出；
P5.4/C0P/SEG3	I/O	9	IO，具有开漏输出功能；比较器的输入端/运放的正端输入；LCD 驱动 SEG3 输出；
P5.3/C0N/SEG4	I/O	10	IO，具有开漏输出功能；比较器的输入端/运放的负端输入；LCD 驱动 SEG4 输出；
P5.2/CO	I/O	11	IO，具有开漏输出功能；比较器/运放的输出
P5.1/PWM3/BZ3	I/O	12	IO(默认开上拉)，具有开漏输出功能；PWM3 输出；蜂鸣器输出
P5.0/PWM2/BZ2	I/O	13	IO(默认开下拉)，具有开漏输出功能；PWM2 输出；蜂鸣器输出
P3.0/AIN0/VREF	I/O	14	IO；ADC 输入 0；ADC 参考电压输入
P3.1/AIN1	I/O	15	IO；ADC 输入 1
P3.2/AIN2	I/O	16	IO；ADC 输入 2
P3.3/AIN3	I/O	17	IO；ADC 输入 3
P3.4/AIN4	I/O	18	IO；ADC 输入 4
P1.0/T2/INT0/SEG0	I/O	19	IO；定时/计数器 2 外部输入；外部中断 0 输入，具有唤醒功能；LCD 驱动 SEG0 输出；
VDD	P	20	电源

注：PT1.3 输入信号的上升沿时间小于 10us 时，对电源有干扰，建议在使用 AD 测量的情况下该输入口作为电平检测或按键中断口。

2 标准功能

2.1 CPU 核

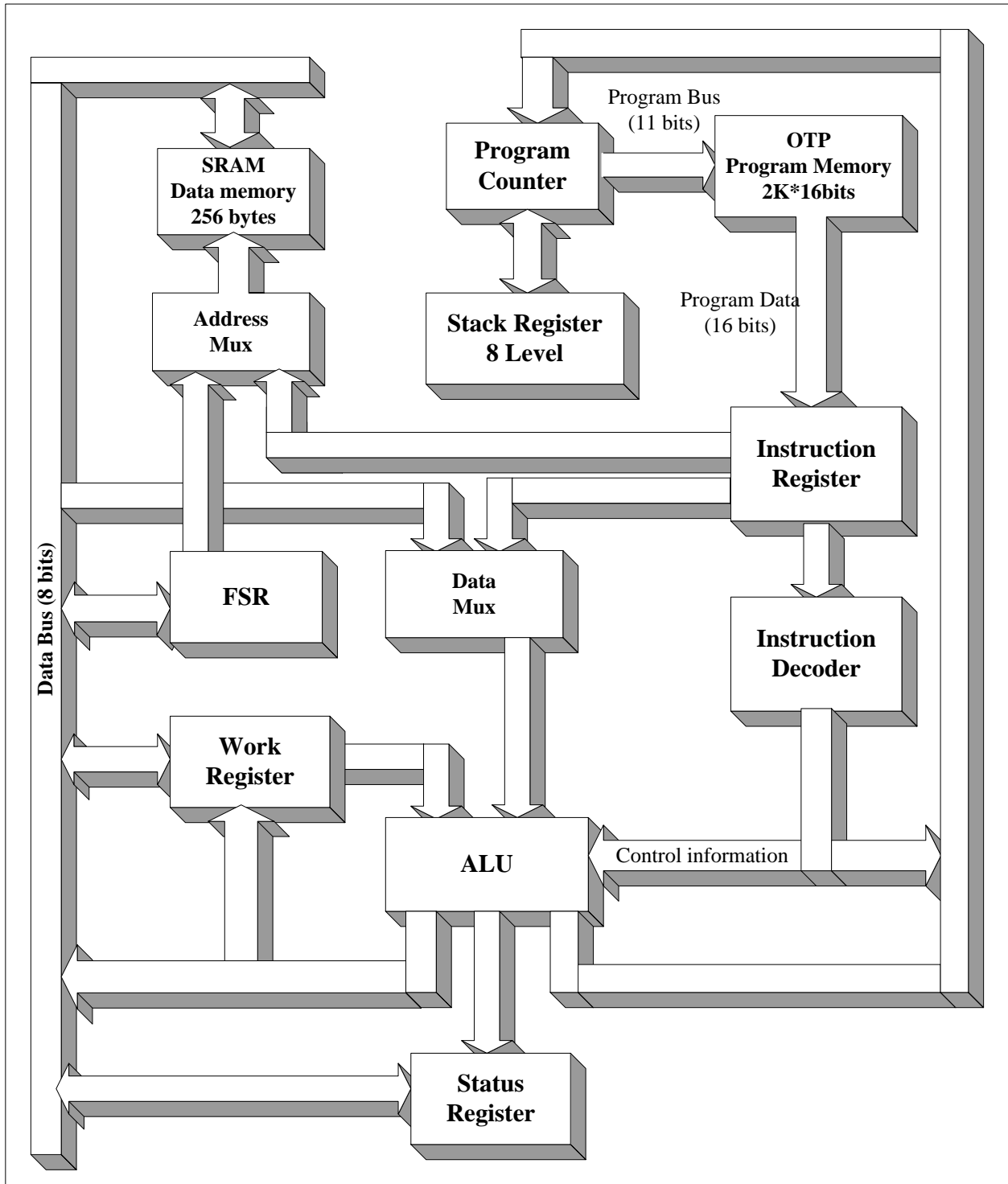


图2 CSU8RP3235/CSU8RP3236 CPU 核的功能模块图

据 CPU 核的功能模块图，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（11bits），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。
堆栈寄存器	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）推送到指令寄存器。</p> <p>CSU8RP3235/CSU8RP3236 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>直接地址（8bits）：数据存储器的地址。CPU 能利用此地址来对数据存储器进行操作。</p> <p>立即数（8bits）：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：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：PD，TO，DC，C 及 Z。
文件选择寄存器	在 CSU8RP3235/CSU8RP3236 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	CSU8RP3235/CSU8RP3236 内带 2K×16 位的 OTP 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 2K 的指令。程序存储器的地址总线是 11bits，数据总线是 16bits。
数据存储器	CSU8RP3235/CSU8RP3236 内带 256bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 8bits，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 CSU8RP3235/CSU8RP3236 中，该程序存储器是 2K*16bit 的程序 OTP，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 0x000，中断入口地址为 0x004，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。

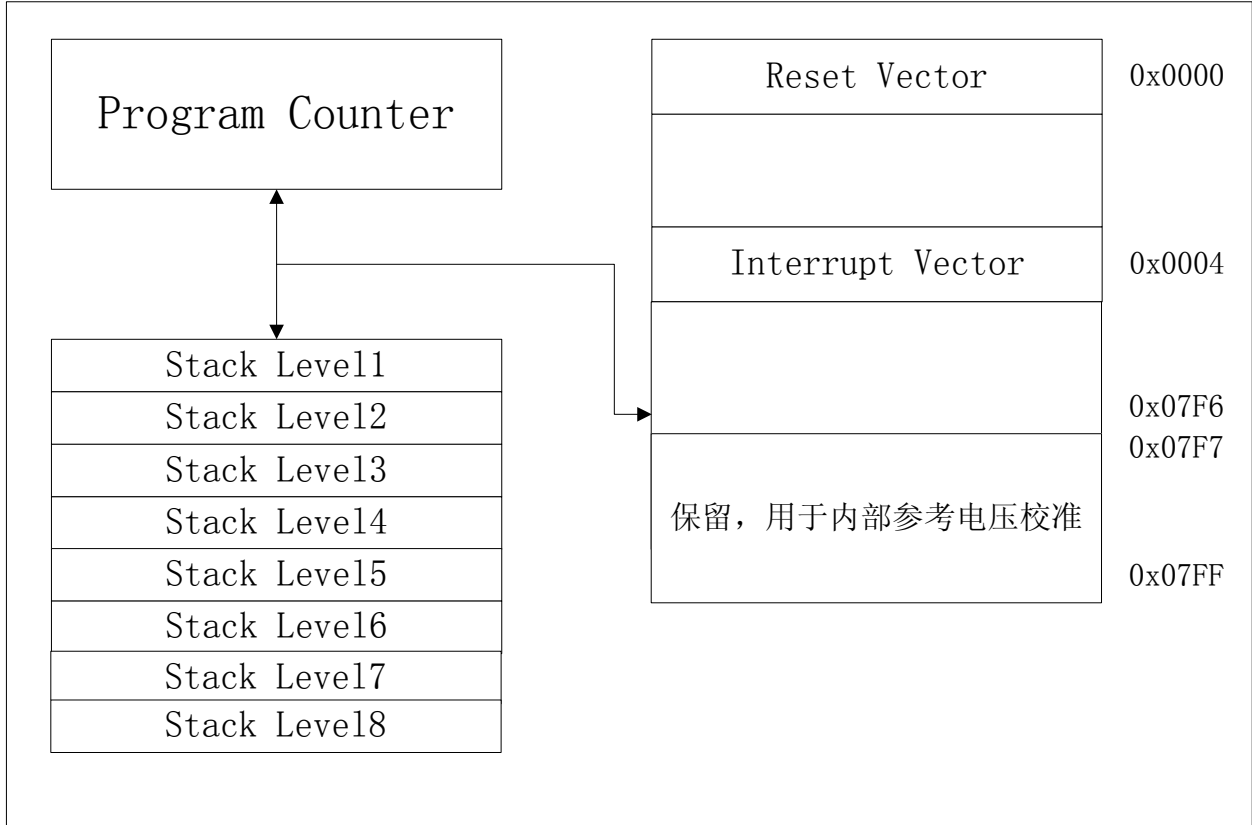


图3 程序存储器

(2) 数据存储器的

数据存储器的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 7FH 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	0x00	0x07
外设特殊功能寄存器	0x08	0x7F
通用数据存储器	0x80	0x17F

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。

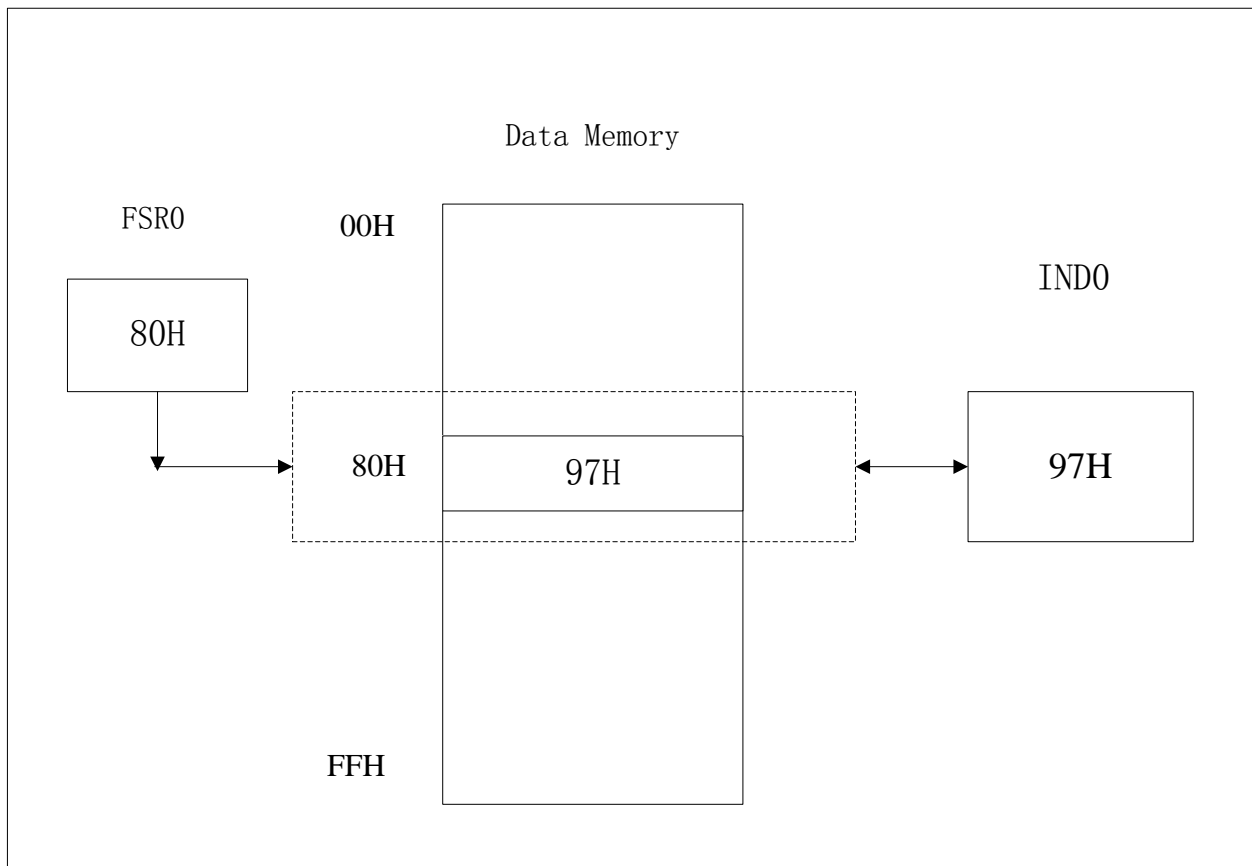


图4 间接地址访问

BSR 寄存器（地址为 08h）

特性	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
BSR	IRP0							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 IRP0: IND0 间接页寻址位

1 = 间接寻址 IND0 时，访问后 256byte 地址

0 = 间接寻址 IND0 时，访问前 256byte 地址

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。TO 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	TO	DC	C	Z
METCH2		LVD30						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 LVD36: 3.6V LVD 工作电压标志位

1: 表明芯片工作电压低于 3.6V

0: 表明芯片工作电压高于 3.6V

Bit 6 LVD24: 2.4V LVD 工作电压标志位

1: 表明芯片工作电压低于 2.4V

0: 表明芯片工作电压高于 2.4V

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 TO: 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

METCH2 寄存器（地址为 7dh）

特性	U-0	R-0	R/W-0	U-0	U-0	U-0	U-0	U-0
METCH2		LVD30						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 LVD30: 3.0V LVD 工作电压标志位

1: 表明芯片工作电压低于 3.0V

0: 表明芯片工作电压高于 3.0V

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	INDO	以 FSR0 中内容作为地址的数据存储器中的数据								xxxxxxxx
02h	FSR0	间接数据存储器地址指针 0								00000000
04h	STATUS	LVD36	LVD24		PD	TO	DC	C	Z	xxu00000
05h	WORK	工作寄存器								00000000
06h	INTF		TM2IF		TMOIF	SRADIF		E1IF	E0IF	u0u00u00
07h	INTE	GIE	TM2IE		TMOIE	SRADIE		E1IE	E0IE	00u00u00
08h	BSR	IRPO								0uuuuuuu
0Ah	EADRH						PAR[11:8]			uuuu0000
0Bh	EADRL				PAR[7:0]					00000000
0Ch	EDATH				EDATH[7:0]					00000000
0Dh	WDTCON	WDTEN					WTS[2:0]			0uuuu000
0Eh	WDTIN				WDTIN[7:0]					11111111
0Fh	TMOCON	TOEN	TORATE[2:0]				TORSTB	TOSEL[1:0]		0000u100
10h	TM0IN				TM0IN[7:0]					11111111
11h	TMOCNT				TMOCNT[7:0]					00000000
16h	MCK	CST	CST_IN	CST_WDT	EO_SLP	CPUCLK_SEL[1:0]		CLKSEL[1:0]		00101100
17h	TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT	00000100
18h	TM2IN				TM2IN[7:0]					11111111
19h	TM2CNT				TM2CNT[7:0]					00000000
1ah	TM2R				TM2R[7:0]					00000000
1bh	TM3CON	T3EN	T3RATE[2:0]			T3CKS	T3RSTB	T3OUT	PWM3OUT	00000100
1ch	TM3IN				TM3IN[7:0]					11111111
1dh	TM3CNT				TM3CNT[7:0]					00000000
1eh	TM3R				TM3R[7:0]					00000000
20h	PT1				PT1[7:0]					xxxxxxxx
21h	PT1EN		PT1EN[7:4]				PT1EN[2:0]			0000u000
22h	PT1PU		PT1PU[7:0]							00001000
23h	PT1CON0	PT110D	PT1W1[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[7:0]							00000000
30h	PT5					PT5[4:0]				uuuxxxxx
31h	PT5EN					PT5EN[4:0]				uuu00000
32h	PT5PU					PT5PU[4:0]				uuu00000
33h	PT5CON					PT5OD[4:0]				uuu00000
38h	PT1CON1					PT1W2[2:0]				uuuuu000
3ch	INTF2				TM3IF					uuu0uuuu
3dh	INTE2				TM3IE					uuu0uuuu
3eh	INTF3	CMPOIF								0uuuuuuu
3fh	INTE3	CMPOIE								0uuuuuuu
40h	TM2INH					TM2IN[11:8]				uuuu1111
41h	TM2CNTH					TM2CNT[11:8]				uuuu0000
42h	TM2RH					TM2R[11:8]				uuuu0000
43h	TM3INH					TM3IN[11:8]				uuuu1111
44h	TM3CNTH					TM3CNT[11:8]				uuuu0000

45h	TM3RH							TM3R[11:8]	uuuu0000	
50h	SRADCON0	SROFT_SEL[1:0]		SRADACKS[1:0]			CMPO_R	SRADCKS[1:0]	0000u000	
51h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]	00000000	
52h	SRADCON2			CHS[3:0]				REF_SEL[1:0]	0000uu00	
54h	SRADL							SRAD[7:0]	00000000	
55h	SRADH							SRAD[11:8]	uuuu0000	
56h	SROFTL							SROFT[7:0]	00000000	
57h	SROFTH							SROFT[11:8]	uuuu0000	
5ah	TRIM_REF							TRIM_REF[7:0]	10000000	
6ah	CMPCON0	CMPEN		COS[2:0]				CMPOUT	0000uuu0	
6bh	CMPCON1						CMPOFT_SEL[1:0]		uuuuuu00u	
7dh	METCH2		LVD30						uxuuuuuu	
7eh	METCH	VTHSEL	PT50PD			PT14PD	T3SEL	T2SEL	VREF_OEN	01uu0000

注：进行读操作时，无效位读出为 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

2.2 时钟系统

2.2.1 概述

芯片的时钟系统包括内置 16/8/4/2MHz 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。Fcpu 是 CPU 时钟频率。

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

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

2.2.2 时钟框图

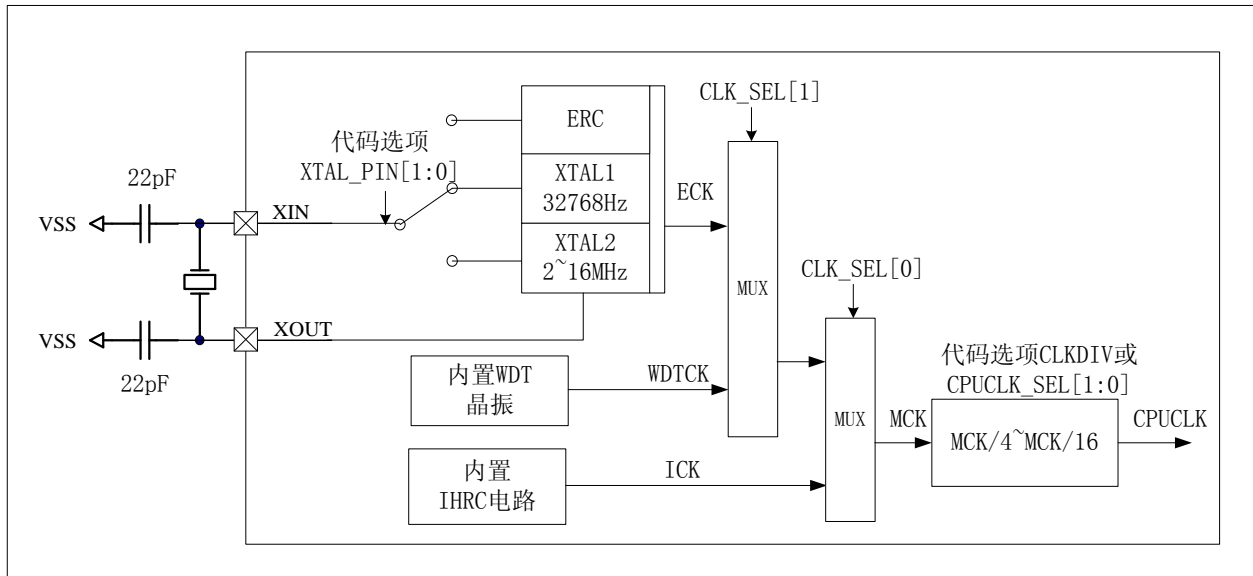


图5 CSU8RP3235/CSU8RP3236 振荡器状态框图 A

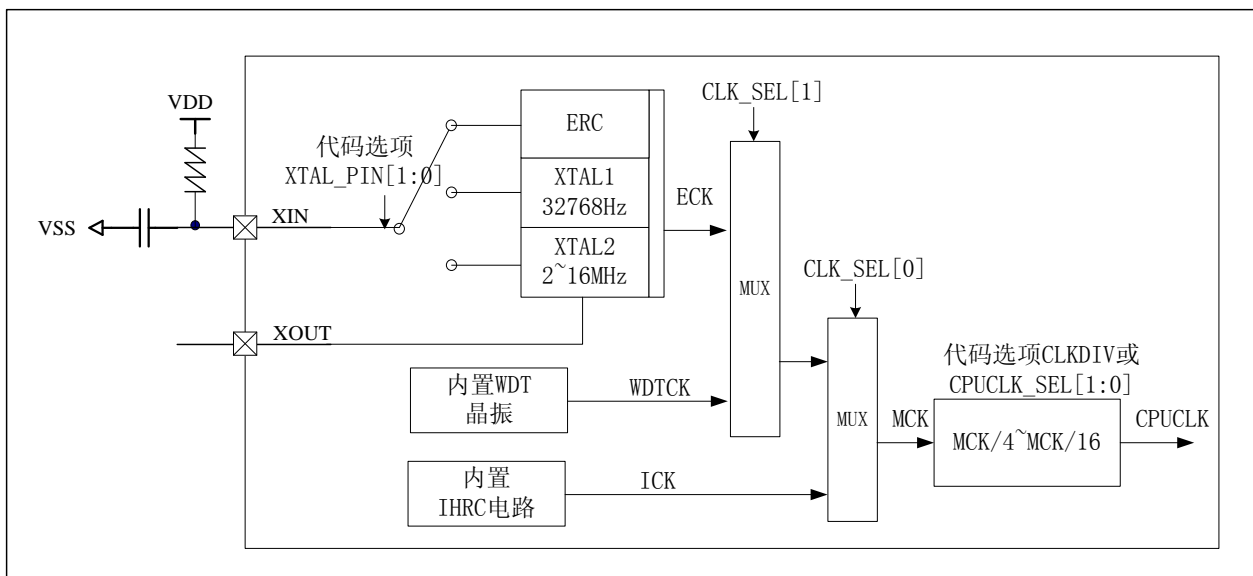


图6 CSU8RP3235/CSU8RP3236 振荡器状态框图 B

2.2.3 寄存器

表 5 CSU8RP3235/CSU8RP3236 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	CPUCLK_SEL[1:0]		CLKSEL[1:0]		00101100

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 1: 外部晶振关闭 0: 外部晶振打开												
6	CST_IN	内部晶振启动开关 1: 内部晶振关闭 0: 内部晶振打开												
5	CST_WDT	内部 WDT 晶振启动开关 1: 内部 WDT 晶振关闭 0: 内部 WDT 晶振打开												
4	EO_SLP	外部低速晶振控制位 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 0: sleep 模式下关闭外部晶振												
3:2	CPUCLK_SEL[1:0]	指令周期配置 <table border="1"> <tr> <td>CPUCLK_SEL[1:0]</td> <td>指令周期</td> </tr> <tr> <td>00</td> <td>保留，进制配置为 00</td> </tr> <tr> <td>01</td> <td>指令周期=4 个时钟周期</td> </tr> <tr> <td>10</td> <td>指令周期=8 个时钟周期</td> </tr> <tr> <td>11</td> <td>指令周期=16 个时钟周期</td> </tr> </table> 代码选项 INS_OP 设置为支持动态指令周期配置，则 CPUCLK_SEL[1:0]有效，可通过 CPUCLK_SEL[1:0]实现动态指令周期配置。	CPUCLK_SEL[1:0]	指令周期	00	保留，进制配置为 00	01	指令周期=4 个时钟周期	10	指令周期=8 个时钟周期	11	指令周期=16 个时钟周期		
CPUCLK_SEL[1:0]	指令周期													
00	保留，进制配置为 00													
01	指令周期=4 个时钟周期													
10	指令周期=8 个时钟周期													
11	指令周期=16 个时钟周期													
1:0	CLKSEL[1:0]	时钟源选择位 <table border="1"> <tr> <td>CLK_SEL[1]</td> <td>时钟源 2 选择</td> </tr> <tr> <td>0</td> <td>外部晶振系统时钟</td> </tr> <tr> <td>1</td> <td>内部 WDT 晶振时钟</td> </tr> </table> <table border="1"> <tr> <td>CLK_SEL[0]</td> <td>CPU 时钟</td> </tr> <tr> <td>0</td> <td>使用时钟源 1(内部高速晶振时钟)作为 CPU 时钟</td> </tr> <tr> <td>1</td> <td>使用时钟源 2 作为 CPU 时钟</td> </tr> </table> 进行时钟源切换时，不能直接从外部晶振切换到内部 WDT 晶振或者从内部 WDT 晶振切换到外部晶振，即当 CLK_SEL[1:0]=2'b01 时，不能直接将它改写为 2'b 10 或 2'b 11，同样 CLK_SEL[1:0]=2'b11 或 2'b 10 时，不能直接将它改写为 2'b 01，否则可能导致切换不成功或系统工作不正常。 (CLK_SEL[0]=1 时，CLK_SEL[1]不可写) 如果要在外部晶振和内部 WDT 晶振间切换，建议先切换到内部高速晶振，再从内部高速晶振切换到外部晶振或内部 WDT 晶振	CLK_SEL[1]	时钟源 2 选择	0	外部晶振系统时钟	1	内部 WDT 晶振时钟	CLK_SEL[0]	CPU 时钟	0	使用时钟源 1(内部高速晶振时钟)作为 CPU 时钟	1	使用时钟源 2 作为 CPU 时钟
CLK_SEL[1]	时钟源 2 选择													
0	外部晶振系统时钟													
1	内部 WDT 晶振时钟													
CLK_SEL[0]	CPU 时钟													
0	使用时钟源 1(内部高速晶振时钟)作为 CPU 时钟													
1	使用时钟源 2 作为 CPU 时钟													

对 MCK 寄存器进行写操作时，建议使用 bcf 或 bsf 指令。

```

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
bsf mck, 6      ;关闭内部晶振
...

```

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上顺序执行

2.2.4 内部高速 RC 时钟

内部高速 RC 时钟（16/8/4/2MHz），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 PT1.1、PT1.2 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 时钟可作为 WDT 和定时器 0 使用，也可作为系统主时钟。

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，PT1.1、PT1.2 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，PT1.1、PT1.2 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，PT1.1 口做为 RC 输入引脚，PT1.2 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 PT1.1 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

CSU8RP3235/CSU8RP3236 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。

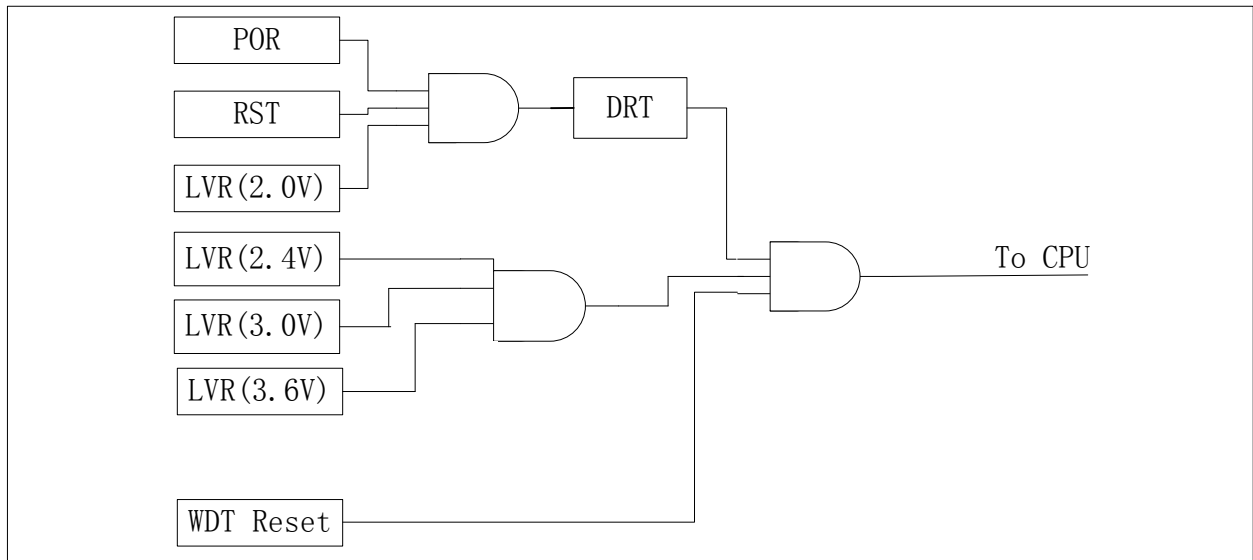


图7 复位电路原理图

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。

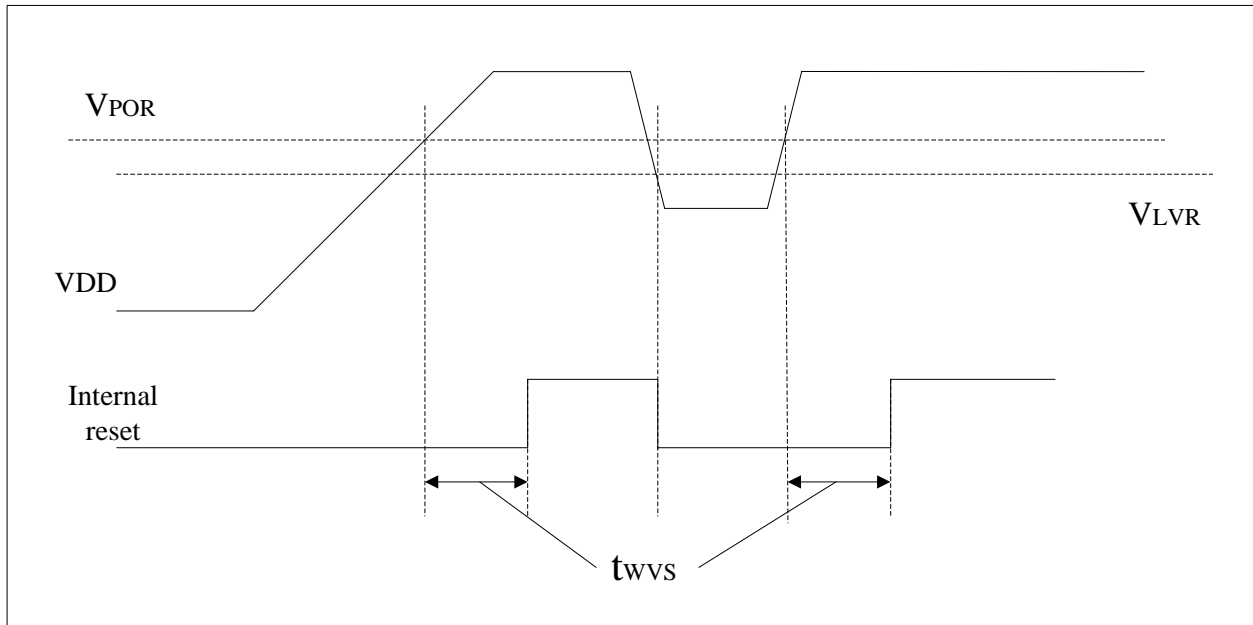


图8 上电复位电路示例及上电过程

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
twvs (测试条件: VDD=5V, T=25°C)	78.4ms	98ms	117.6ms

VPOR: 上电复位

VLVR: 低电压复位

twvs: 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 0.07V/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。

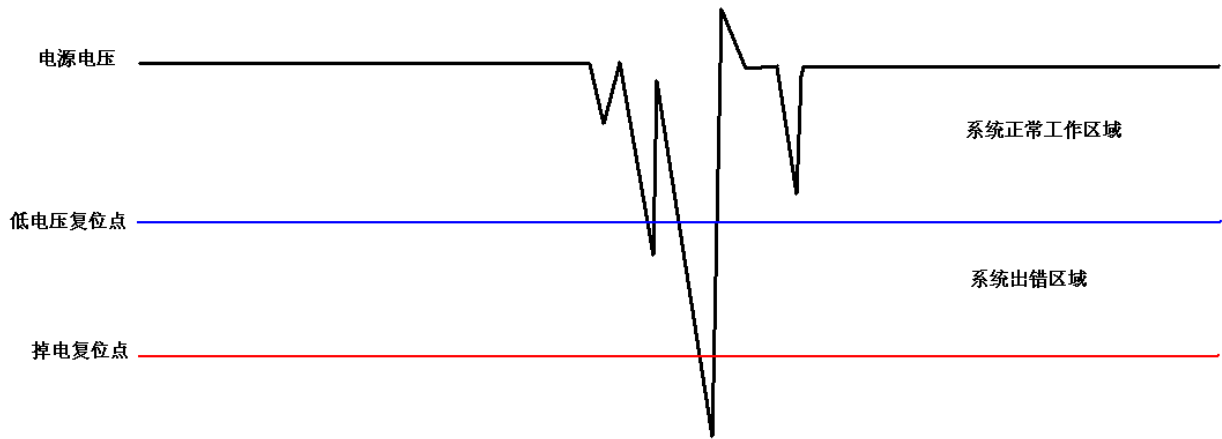


图9 系统掉电复位示意图

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](#)。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 3.10](#)。通过设置该代码选项，可启用外部硬件复位功能。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。

在芯片代码选项启用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

CSU8RP3235/CSU8RP3236 有 7 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、INTE2、INTE3 和中断标志位寄存器 INTF、INTF2、INTF3。这 8 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。标志位在没有打开中断使能时，状态改变时也会置 1

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。

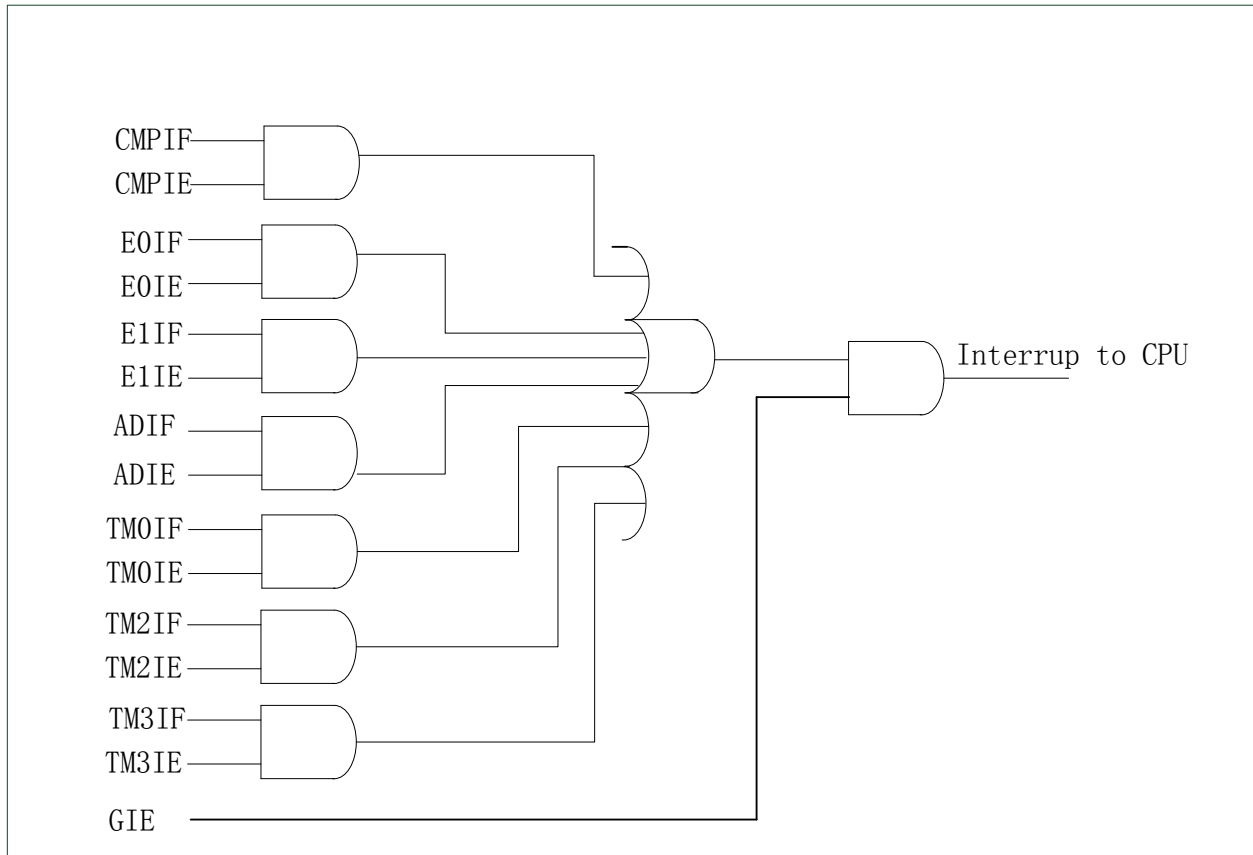


图10 中断逻辑

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: 12-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: 12-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

INTE3 寄存器（地址为 3fh）

特性	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
INTE2	CMP0IF							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 CMP0IE: 比较器 0 中断使能标志

- 1 = 使能比较器 0 中断
- 0 = 不使能比较器 0 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。中断标志位，就算其中断使能不为 1 时，也可能硬件置位。

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: 12-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: 12-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

INTF3 寄存器（地址为 3eh）

特性	R/W -0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
INTF2	CMP0IF							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 CMP0IF: 比较器 0 中断标志，软件清零，硬件置高
1 = 发生比较器 0 中断，必须软件清 0
0 = 没发生比较器 0 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 PT1CON0 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。不管 E0IE 是否使能，只要 PT1.0 被触发，中断标志位 E0IF 就会置 1。

PT1CON0 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
PT1CON0						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、PT1.2、PT1.3、PT1.4、PT1.5、PT1.6 和 PT1.7 都可作为外部中断 1 的输入端。触发方式由 PT1CON0 寄存器中的 E1M 寄存器决定。INTE 寄存器中的 E1IE 为外部中断 1 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。不管 E1IE 是否使能，只要外部中断 1 被触发，中断标志位 E1IF 就会置 1。

PT1CON0 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
PT1CON0		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，外部中断 1 不会被触发，中断标志位 E1IF 不会被置 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，外部中断 1 不会被触发，中断标志位 E1IF 不会被置 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，外部中断 1 不会被触发，中断标志位 E1IF 不会被置 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，外部中断 1 不会被触发，中断标志位 E1IF 不会被置 1
 1 = 使能 PT1.1 外部中断 1

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

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
PT1CON1						PT1W2[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 2 PT1W2[2]:PT1.7 外部中断 1 使能
 0 = 禁止 PT1.7 外部中断 1
 1 = 使能 PT1.7 外部中断 1
- Bit 1 PT1W2[1]:PT1.6 外部中断 1 使能
 0 = 禁止 PT1.6 外部中断 1
 1 = 使能 PT1.6 外部中断 1
- Bit 0 PT1W2[0]:PT1.2 外部中断 1 使能
 0 = 禁止 PT1.2 外部中断 1
 1 = 使能 PT1.2 外部中断 1

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

2.4.5 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，不管 SRADIE 是否使能，SRADIF 都会硬件置 1。

2.4.6 定时器 0 溢出中断

INTE 寄存器中的 TMOIE 为定时器 0 中断的使能位，INTF 寄存器中的 TMOIF 为中断标志位，软件清 0。当定时器 0 溢出时，不管 TMOIE 是否使能，TMOIF 都会硬件置 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，不管 TM2IE 是否使能，TM2IF 都会硬件置 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，不管 TM3IE 是否使能，TM3IF 都会硬件置 1。

2.4.9 比较器 0 中断

INTE3 寄存器中的 CMP0IE 为比较器中断的使能位，INTF3 寄存器中的 CMP0IF 为中断标志位，软件清 0。当比较器 0 结果改变时，不管 CMP0IE 是否使能，CMP0IF 都会硬件置 1。

2.4.10 PUSH 和 POP 处理

CSU8RP3235/CSU8RP3236 有 8 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,e1if ;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

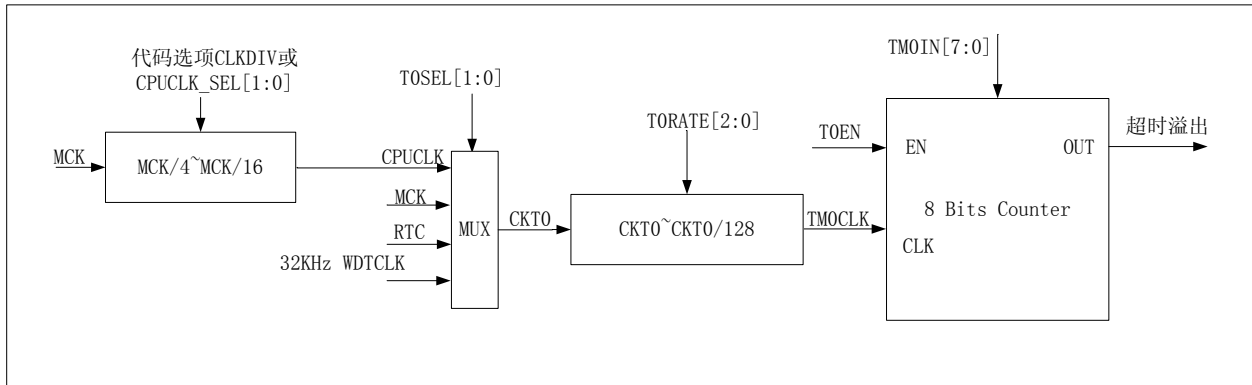


图11 定时器 0 功能框图

定时器 0 模块的输入为 MCK 或者 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）以选择定时器超时时间。当定时超时发生时，中断标志位会硬件置 1，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					u0u00u00
07H	INTE	GIE			TM0IE					00u00u00
0FH	TM0CON	TOEN	TORATE[2:0]				TORSTB	TOSEL[1:0]		0000u100
10H	TMOIN	TMOIN[7:0]								11111111
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 1: 使能定时器 0 0: 禁止定时器 0																		
6:4	TORATE[2:0]	定时器 0 时钟选择 <table border="1"> <thead> <tr> <th>TORATE [2:0]</th> <th>TM0CLK</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>CKT0</td> </tr> <tr> <td>001</td> <td>CKT0 /2</td> </tr> <tr> <td>010</td> <td>CKT0 /4</td> </tr> <tr> <td>011</td> <td>CKT0 /8</td> </tr> <tr> <td>100</td> <td>CKT0 /16</td> </tr> <tr> <td>101</td> <td>CKT0 /32</td> </tr> <tr> <td>110</td> <td>CKT0 /64</td> </tr> <tr> <td>111</td> <td>CKT0 /128</td> </tr> </tbody> </table>	TORATE [2:0]	TM0CLK	000	CKT0	001	CKT0 /2	010	CKT0 /4	011	CKT0 /8	100	CKT0 /16	101	CKT0 /32	110	CKT0 /64	111	CKT0 /128
TORATE [2:0]	TM0CLK																			
000	CKT0																			
001	CKT0 /2																			
010	CKT0 /4																			
011	CKT0 /8																			
100	CKT0 /16																			
101	CKT0 /32																			
110	CKT0 /64																			
111	CKT0 /128																			
2	TORSTB	定时器 0 复位 1: 禁止定时器 0 复位 0: 使能定时器 0 复位 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 <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>CPUCLK</td> </tr> <tr> <td>01</td> <td>MCK</td> </tr> <tr> <td>10</td> <td>外部 32768Hz 晶振时钟, 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, 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	CPUCLK	01	MCK	10	外部 32768Hz 晶振时钟, 仅当外部接 32768Hz 晶振, 且晶振打开时有效	11	内部 32K WDT 时钟, 仅当内部 WDT 晶振打开时有效								
TOSEL[1:0]	定时器 0 时钟源																			
00	CPUCLK																			
01	MCK																			
10	外部 32768Hz 晶振时钟, 仅当外部接 32768Hz 晶振, 且晶振打开时有效																			
11	内部 32K WDT 时钟, 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: TM0IE 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: TOEN, 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:

$$\text{定时器 0 溢出时间} = (\text{TM0IN} + 1) / \text{TM0CLK}.$$

2.6 I/O PORT

CSU8RP3235/CSU8RP3236 包括 17 个双向 IO 口，1 个输入口 PT1.3。所有 IO 口都可以配置上拉，PT5.0、PT1.4 还可以配置下拉，其中 PT5.0 默认打开下拉，PT1.3、PT5.1 默认打开上拉。

其他 IO 特性：

- 6 个开漏输出（P1.1、P5.0~P5.4）
- PT3.0~PT3.4 可配置为模拟口。
- PT1.0 为外部中断 0，PT1.1~PT1.7 可配置为外部中断 1 输入，触发方式可配置

IO 其他功能复用见相关功能模块描述

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[7:0]								xxxxxxxx
21h	PT1EN	PT1EN[7:4]					PT1EN[2:0]			0000u000
22h	PT1PU	PT1PU[7:0]								00001000
23h	PT1CON0	PT11OD	PT1W1[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[7:0]								00000000
30h	PT5					PT5[4:0]			uuuxxxxx	
31h	PT5EN					PT5EN[4:0]			uuu00000	
32h	PT5PU					PT5PU[4:0]			uuu00010	
33h	PT5CON					PT5OD[4:0]			uuu00000	
38h	PT1CON1						PT1W2[2:0]		uuuuu000	
7eh	METCH		PT50PD			PT14PD				01uu0000

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。CSU8RP3235/CSU8RP3236 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	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT1	PT1[7:0]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7-0 PT1[7:0]: GPIO1 口数据标志

PT1[7] = GPIO1 bit 7 数据标志位

PT1[6] = GPIO1 bit 6 数据标志位

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[2] = GPIO1 bit 2 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

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

PT1EN 寄存器（地址为 21h）

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PT1EN	PT1EN[7:4]					PT1EN[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1EN[7:0]: GPIO1 口输入/输出控制标志

PT1EN[7] = GPIO1 bit 7 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT1EN[6] = GPIO1 bit 6 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

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[2] = GPIO1 bit 2 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

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

PT1PU 寄存器（地址为 22h）

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
PT1PU	PT1PU[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PU[7:0]: GPIO1 口上拉电阻使能标志

PT1PU[7] = GPIO1 bit 7 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT1PU[6] = GPIO1 bit 6 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

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[2] = GPIO1 bit 2 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

METCH 寄存器（地址为 7Eh）

特性	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
METCH					PT14PD			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3 PT14PD: GPIO1 bit4 口下拉电阻使能标志（默认打开下拉）

0 = 断开下拉电阻, 1 = 使用下拉电阻

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

PT1CON0 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	
PT1CON0	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

注：做开漏输出时，外部上拉电压不高于 VDD+0.3V。

PT1CON1 寄存器（地址为 38h）

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
PT1CON1						PT1W2[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 2 PT1W2[2]:PT1.7 外部中断 1 使能
0 = 禁止 PT1.7 外部中断 1
1 = 使能 PT1.7 外部中断 1
- Bit 1 PT1W2[1]:PT1.6 外部中断 1 使能
0 = 禁止 PT1.6 外部中断 1
1 = 使能 PT1.6 外部中断 1
- Bit 0 PT1W2[0]:PT1.2 外部中断 1 使能
0 = 禁止 PT1.2 外部中断 1
1 = 使能 PT1.2 外部中断 1

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 = 不确定位

PT3CON 寄存器（地址为 2bh）

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
PT3CON	PT3CON[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit4-0 PT3CON[4:0]: GPIO1/GPIO3 口模拟/数字端口使能标志

PT3CON[7] = GPIO1bit 7 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT3CON[6] = GPIO1bit 6 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT3CON[5] = GPIO1bit 5 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

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

注: PT3.0 设为参考电压输出时, 直接作为模拟口输出电压, 优先级高于 PT3CON[0]设置。

2.6.3 PT5 口
PT5 寄存器（地址为 30h）

特性	U-0	U-0	U-0	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT5	PT5[4:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 4-0 PT5[4:0]: GPIO5 口数据标志位

PT5[4] = GPIO5 bit 4 的数据标志位

PT5[3] = GPIO5 bit 3 的数据标志位

PT5[2] = GPIO5 bit 2 的数据标志位

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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5EN	PT5EN[4:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 4-0 PT5EN[4:0]: GPIO5 口输入/输出控制标志

PT5EN[4] = GPIO5 bit 4 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT5EN[3] = GPIO5 bit 3 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT5EN[2] = GPIO5 bit 2 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

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

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

PT5PU 寄存器（地址为 32h）

特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5PU				PT5PU[4:0]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 4-0 PT5PU[4:0]: GPIO5 口上拉电阻使能标志

- PT5PU[4] = GPIO5 bit 4 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT5PU[3] = GPIO5 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT5PU[2] = GPIO5 bit 2 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- 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

PT5CON 寄存器（地址为 33h）

特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5CON				PT5OD[4:0]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 4-0 PT5CON[4:0]: GPIO5 口控制标志

- PT5CON[4] = GPIO5 bit 4 控制标志位; 0 = 禁止开漏输出, 1 = 使能开漏输出
- PT5CON[3] = GPIO5 bit 3 控制标志位; 0 = 禁止开漏输出, 1 = 使能开漏输出
- PT5CON[2] = GPIO5 bit 2 控制标志位; 0 = 禁止开漏输出, 1 = 使能开漏输出
- 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

注: 做开漏输出时, 外部上拉电压不高于 VDD+0.3V。

METCH 寄存器（地址为 7Eh）

特性	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
METCH		PT50PD			PT14PD			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 6 PT50PD: GPIO5 bit0 口下拉电阻使能标志（默认打开下拉）
0 = 断开下拉电阻, 1 = 使用下拉电阻
- Bit 3 PT14PD: GPIO1 bit4 口下拉电阻使能标志（默认打开下拉）
0 = 断开下拉电阻, 1 = 使用下拉电阻

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

3 增强功能

3.1 Halt 和 Sleep 模式

CSU8RP3235/CSU8RP3236 支持低功耗工作模式。为了使其处于待机状态，可以让 CPU 停止工作使 CSU8RP3235/CSU8RP3236 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 2.4/3.0/3.6V 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 pt1en ;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 pt1en ;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)

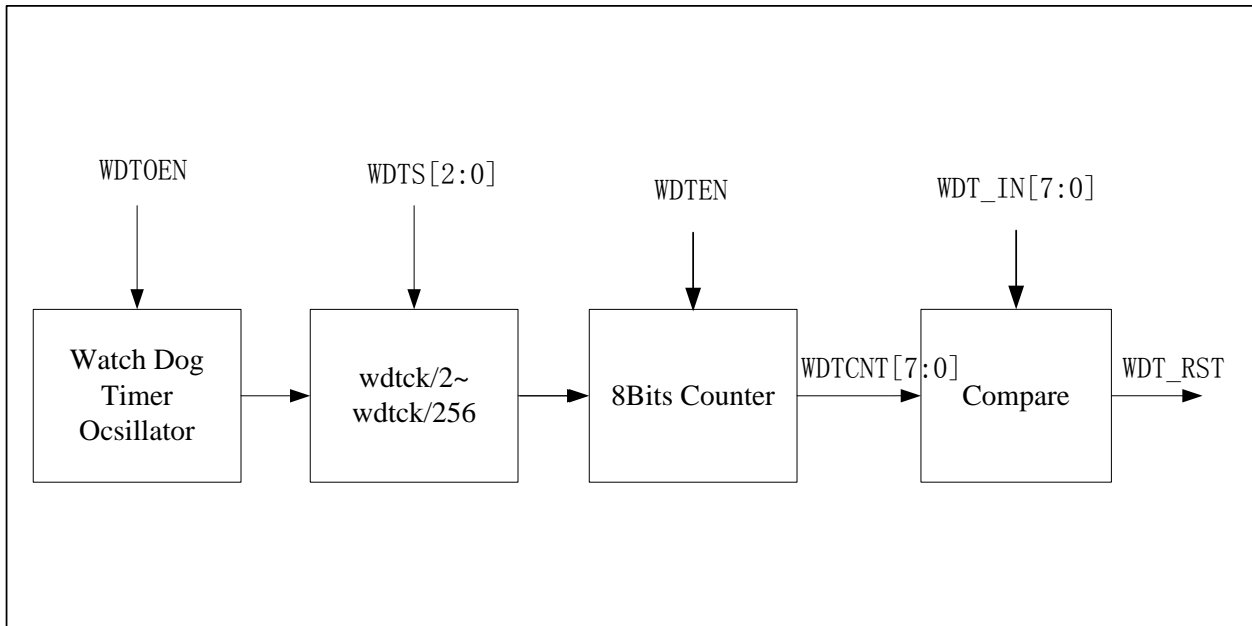


图12 看门狗定时器功能框图

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, 产生的时钟会进行 2 到 256 分频。用户通过寄存器标志位 **WDTS[2:0]** 选择看门狗模块的计数时钟。当用户使能 **WDTEN** 时, 8 位计数器开始计数, 当 8 bits 计数器计数值与 **WDTIN** 数值相等时溢出, 溢出时它会发送 **WDTOVERFLOW** 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				xxu00000	
0DH	WDTCON	WDTEN					WDTS[2:0]			0uuuu000	
0Eh	WDTIN	WDT_IN[7:0]									11111111

位地址	标识符	功能	
7	WDTEN	看门狗使能位, 高电平有效	
6:3	保留	保留	
2: 0	WDTS[2:0]	WDT 计数时钟选择	
		WDTS [2:0]	WDT 计数时钟
		000	WDTCLK /256
		001	WDTCLK /128
		010	WDTCLK /64
		011	WDTCLK /32
		100	WDTCLK /16
		101	WDTCLK /8
		110	WDTCLK /4
111	WDTCLK /2		

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：

$$\text{溢出时间} = \frac{2^{(8-\text{WDTS}[2:0])}}{32768} * (\text{WDTIN}[7:0] + 1)$$

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

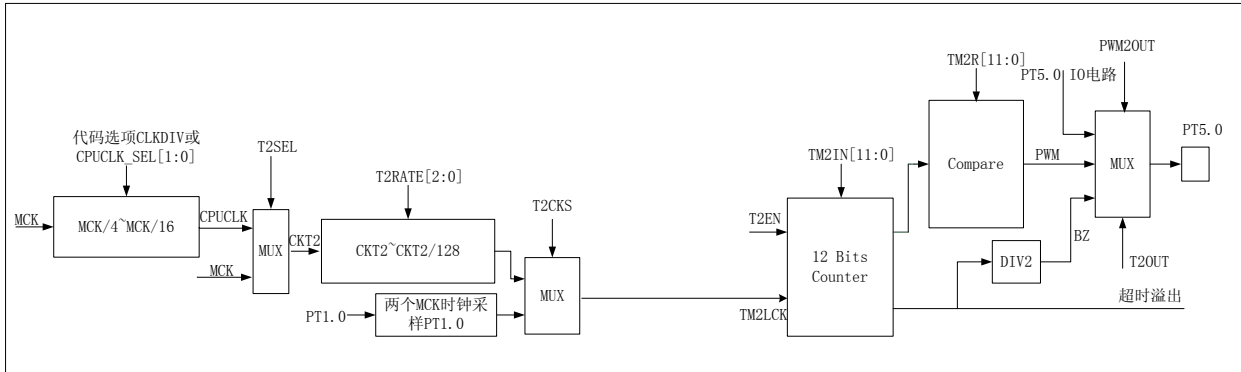


图13 定时/计数器 2 模块的功能框图

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，12 bits 计数器将启动，从 000h 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) 12 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 寄存器说明

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		00000100	
18h	TM2IN	TM2IN[7:0]									11111111
19h	TM2CNT	TM2CNT[7:0]									00000000
1ah	TM2R	TM2R[7:0]									00000000
40h	TM2INH					TM2IN[11:8]				uuuu1111	
41h	TM2CNTH					TM2CNT[11:8]				uuuu0000	
42h	TM2RH					TM2R[11:8]				uuuu0000	
7eh	METCH							T2SEL		01uu0000	

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 1: 使能定时器 2 0: 禁止定时器 2																		
6:4	T2RATE[2:0]	定时/计数器 2 时钟分频																		
		<table border="1"> <tr> <th>T2RATE [2:0]</th> <th>TM2CLK</th> </tr> <tr> <td>000</td> <td>CKT2</td> </tr> <tr> <td>001</td> <td>CKT2 /2</td> </tr> <tr> <td>010</td> <td>CKT2 /4</td> </tr> <tr> <td>011</td> <td>CKT2 /8</td> </tr> <tr> <td>100</td> <td>CKT2 /16</td> </tr> <tr> <td>101</td> <td>CKT2 /32</td> </tr> <tr> <td>110</td> <td>CKT2 /64</td> </tr> <tr> <td>111</td> <td>CKT2 /128</td> </tr> </table>	T2RATE [2:0]	TM2CLK	000	CKT2	001	CKT2 /2	010	CKT2 /4	011	CKT2 /8	100	CKT2 /16	101	CKT2 /32	110	CKT2 /64	111	CKT2 /128
		T2RATE [2:0]	TM2CLK																	
		000	CKT2																	
		001	CKT2 /2																	
		010	CKT2 /4																	
		011	CKT2 /8																	
		100	CKT2 /16																	
		101	CKT2 /32																	
		110	CKT2 /64																	
111	CKT2 /128																			
可通过 METCH[1]选择定时器 2 时钟																				
3	T2CKS	定时/计数器 2 时钟源选择位 1: PT1.0 作为时钟 0: CPUCLK 或 MCK 的分频时钟																		
2	T2RSTB	定时/计数器 2 复位 1: 禁止定时/计数器 2 复位 0: 使能定时/计数器 2 复位 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																		
		<table border="1"> <tr> <th>T2OUT</th> <th>PWM2OUT</th> <th>PT5.0 输出控制, 仅当 PT5.0 配置为输出有效</th> </tr> <tr> <td>0</td> <td>0</td> <td>IO 输出</td> </tr> </table>	T2OUT	PWM2OUT	PT5.0 输出控制, 仅当 PT5.0 配置为输出有效	0	0	IO 输出												
T2OUT	PWM2OUT	PT5.0 输出控制, 仅当 PT5.0 配置为输出有效																		
0	0	IO 输出																		
0	PWM2OUT	0	PWM2 输出																	
		1	蜂鸣器输出																	
		1	PWM2 取反输出																	

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 TM2INH 寄存器各位功能表

位地址	标识符	功能
3: 0	TM2IN[11:8]	定时/计数器溢出值

表 18 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 19 TM2CNTH 寄存器各位功能表

位地址	标识符	功能
3: 0	TM2CNT[11:8]	定时/计数器 2 计数寄存器, 只读

表 20 TM2R 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

表 21 TM2RH 寄存器各位功能表

位地址	标识符	功能
3 : 0	TM2R[11:8]	定时/计数器 2 的 PWM 高电平占空比控制寄存器

表 22 METCH 寄存器各位功能表

位地址	标识符	功能
1	T2SEL	定时器 2 时钟选择 0: CPUCLK 1: MCK

3.3.2 蜂鸣器

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 12 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; 寄存器标志位 TM2IF 会自复位, 程序计数器会复位为 04h。

定时器 2 溢出时间计算方法:

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

蜂鸣器周期计算方法:

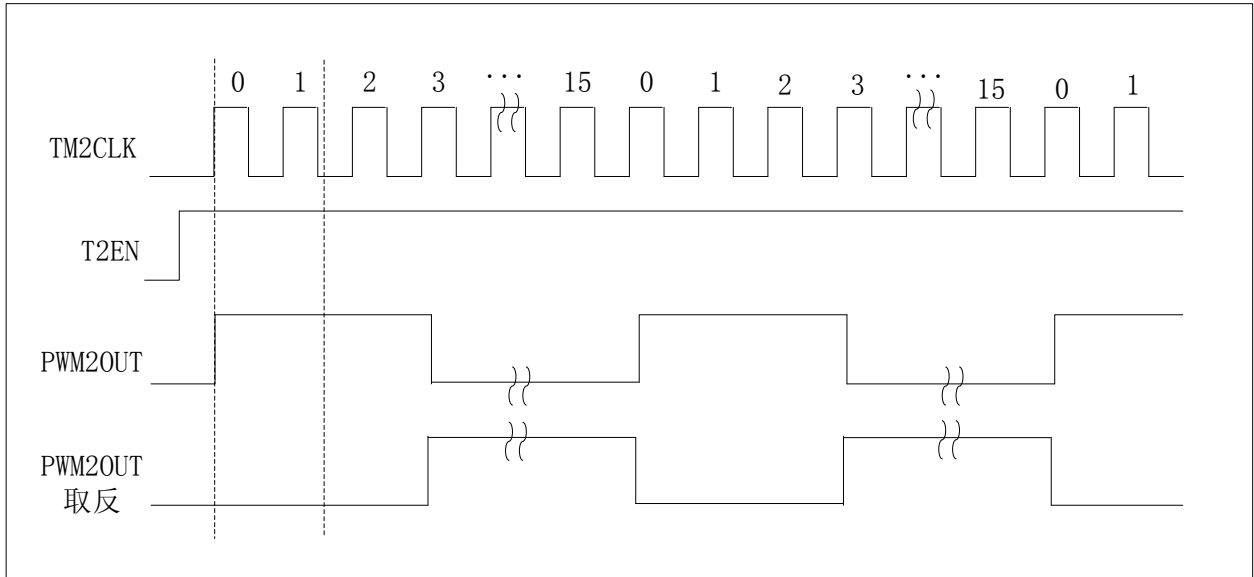
$$\text{蜂鸣器周期} = (\text{TM2IN} + 1) * 2 / \text{TM2CLK}. \quad (\text{TM2IN 不为 } 0)$$

3.3.3 PWM

操作:

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

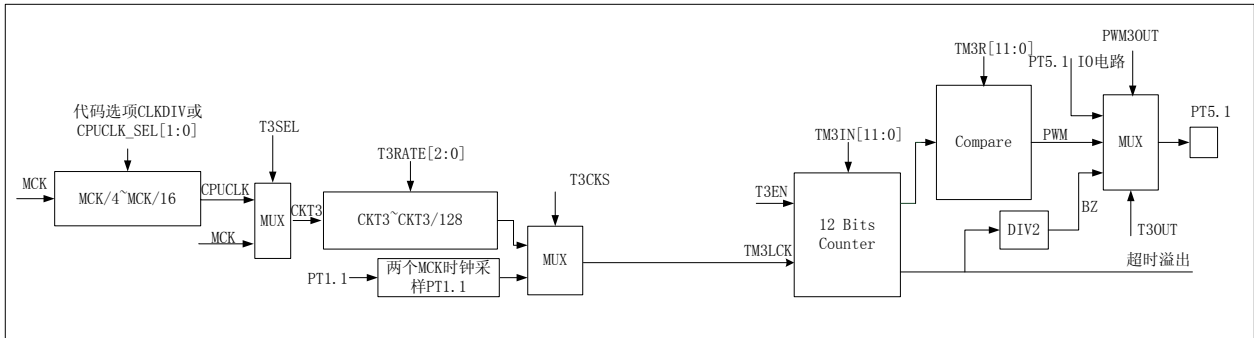


图14 定时/计数器 1 模块的功能框图

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，12 bits 计数器将启动，从 000h 递增至到 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) 12 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 寄存器说明

表 23 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		00000100
1ch	TM3IN	TM3IN[7:0]								11111111
1dh	TM3CNT	TM3CNT[7:0]								00000000
1eh	TM3R	TM3R[7:0]								00000000
43h	TM3INH					TM3IN[11:8]				uuuu1111
44h	TM3CNTH					TM3CNT[11:8]				uuuu0000
45h	TM3RH					TM3R[11:8]				uuuu0000
7eh	METCH						T3SEL			01uu0000

表 24 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 1: 使能定时器 3 0: 禁止定时器 3																		
6:4	T3RATE[2:0]	定时/计数器 3 时钟分频 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>T3RATE [2:0]</th> <th>TM3CLK</th> </tr> </thead> <tbody> <tr><td>000</td><td>CKT3</td></tr> <tr><td>001</td><td>CKT3 /2</td></tr> <tr><td>010</td><td>CKT3 /4</td></tr> <tr><td>011</td><td>CKT3 /8</td></tr> <tr><td>100</td><td>CKT3 /16</td></tr> <tr><td>101</td><td>CKT3 /32</td></tr> <tr><td>110</td><td>CKT3 /64</td></tr> <tr><td>111</td><td>CKT3 /128</td></tr> </tbody> </table> 可通过 METCH[2]选择定时器 3 时钟	T3RATE [2:0]	TM3CLK	000	CKT3	001	CKT3 /2	010	CKT3 /4	011	CKT3 /8	100	CKT3 /16	101	CKT3 /32	110	CKT3 /64	111	CKT3 /128
T3RATE [2:0]	TM3CLK																			
000	CKT3																			
001	CKT3 /2																			
010	CKT3 /4																			
011	CKT3 /8																			
100	CKT3 /16																			
101	CKT3 /32																			
110	CKT3 /64																			
111	CKT3 /128																			
3	T3CKS	定时/计数器 3 时钟源选择位 1: PT1.1 作为时钟 0: CPUCLK 或 MCK 的分频时钟																		
2	T3RSTB	定时/计数器 3 复位 1: 禁止定时/计数器 3 复位 0: 使能定时/计数器 3 复位 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 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>T3OUT</th> <th>PWM3OUT</th> <th>PT5.1 输出控制, 仅当 PT5.1 配置为输出有效</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>IO 输出</td></tr> <tr><td>0</td><td>1</td><td>PWM3 输出</td></tr> <tr><td>1</td><td>0</td><td>蜂鸣器输出</td></tr> <tr><td>1</td><td>1</td><td>PWM3 取反输出</td></tr> </tbody> </table>	T3OUT	PWM3OUT	PT5.1 输出控制, 仅当 PT5.1 配置为输出有效	0	0	IO 输出	0	1	PWM3 输出	1	0	蜂鸣器输出	1	1	PWM3 取反输出			
T3OUT	PWM3OUT	PT5.1 输出控制, 仅当 PT5.1 配置为输出有效																		
0	0	IO 输出																		
0	1	PWM3 输出																		
1	0	蜂鸣器输出																		
1	1	PWM3 取反输出																		
0	PWM3OUT	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>T3OUT</th> <th>PWM3OUT</th> <th>PT5.1 输出控制, 仅当 PT5.1 配置为输出有效</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>IO 输出</td></tr> <tr><td>0</td><td>1</td><td>PWM3 输出</td></tr> <tr><td>1</td><td>0</td><td>蜂鸣器输出</td></tr> <tr><td>1</td><td>1</td><td>PWM3 取反输出</td></tr> </tbody> </table>	T3OUT	PWM3OUT	PT5.1 输出控制, 仅当 PT5.1 配置为输出有效	0	0	IO 输出	0	1	PWM3 输出	1	0	蜂鸣器输出	1	1	PWM3 取反输出			
T3OUT	PWM3OUT	PT5.1 输出控制, 仅当 PT5.1 配置为输出有效																		
0	0	IO 输出																		
0	1	PWM3 输出																		
1	0	蜂鸣器输出																		
1	1	PWM3 取反输出																		

表 25 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 26 TM3INH 寄存器各位功能表

位地址	标识符	功能
3 : 0	TM3IN[11:8]	定时/计数器溢出值

表 27 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 28 TM3CNTH 寄存器各位功能表

位地址	标识符	功能
3 : 0	TM3CNT[11:8]	定时/计数器 3 计数寄存器, 只读

表 29 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

表 30 TM3RH 寄存器各位功能表

位地址	标识符	功能
3 : 0	TM3R[11:8]	定时/计数器 3 的 PWM 高电平占空比控制寄存器

表 31 METCH 寄存器各位功能表

位地址	标识符	功能
2	T3SEL	定时器 3 时钟选择 0: CPUCLK 1: MCK

3.4.2 蜂鸣器

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 12 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; 寄存器标志位 TM3IF 会自复位, 程序计数器会复位为 04h。

定时器 3 溢出时间计算方法:

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

蜂鸣器周期计算方法:

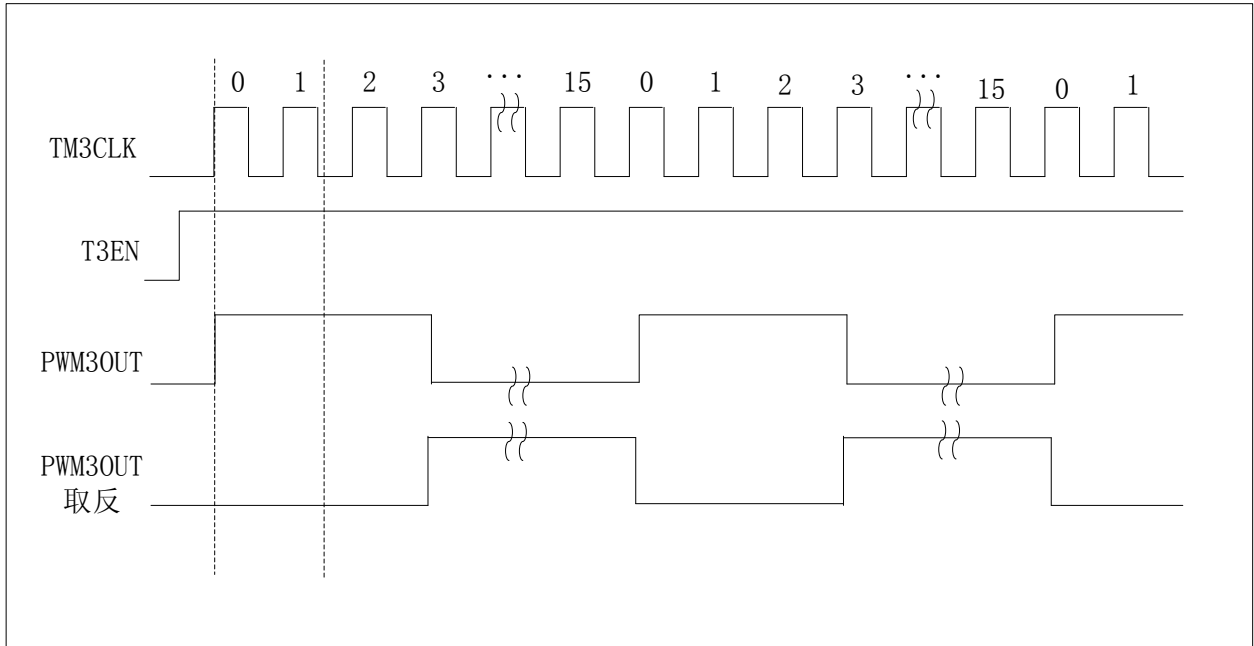
$$\text{蜂鸣器周期} = (\text{TM3IN} + 1) * 2 / \text{TM3CLK. (TM3IN 不为 0)}$$

3.4.3 PWM

操作:

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

CSU8RP3235/CSU8RP3236 模数转换模块共用 5 条外部通道 (AIN0~AIN4) 和 5 条特殊通道 (AIN8: 内部 1/8VDD; AIN9: 内部参考电压; AIN10: GND; AIN11: 运算放大器 0 输出作输入; , 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, 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 ADS 清 0, 并将转换结果存入寄存器 SRADL 和 SRADH 中。

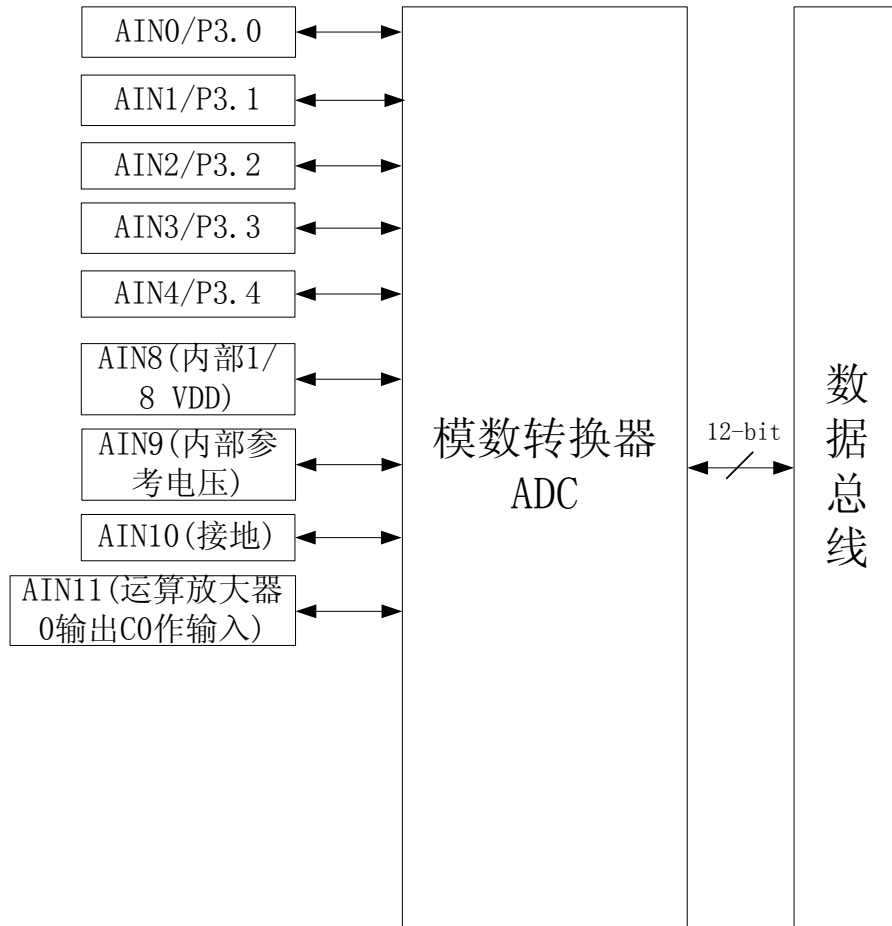


图15 模数转换器 ADC 功能框图

3.5.1 寄存器说明

表 32 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	SROFT_SEL[1:0]		SRADACKS[1:0]			CMP0_R	SRADCKS[1:0]		0000u000	
51h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]		00000000	
52h	SRADCON2	CHS[3:0]							REF_SEL[1:0]		0000uu00
54h	SRADL	SRAD[7:0]								00000000	
55h	SRADH	SRAD[11:8]								uuuu0000	
56h	SROFTL	SROFT[7:0]								00000000	
57h	SROFTH	SROFT[11:8]								uuuu0000	
5ah	TRIM_REF	TRIM_REF[7:0]								10000000	

7eh	METCH							VREF_O EN	00000000
-----	-------	--	--	--	--	--	--	--------------	----------

表 33 SRADCON0 寄存器各位功能表

位地址	标识符	功能
7:6	SROFT_SEL[1:0]	AD 失调电压选择
		SROFT_SEL[1:0] 失调电压校正方式选择
		00 默认失调电压
		01 使失调电压往负方向偏 3mV
		10 使失调电压往正方向偏 3mV
		11 使失调电压往正方向偏 6mV
配置 SROFT_SEL[1:0]=10 或 SROFT_SEL[1:0]=11, 会变为正方向的失调电压, 可以在初始化选择通道 AIN10 测出正的失调电压, 并放入 SROFTL, SROFTH 寄存器中, 且把 CALIF, ENOV 置 1, 这样得到的 AD 值已抵消了失调电压。		
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
		11 2 个 ADC 时钟
不同的外部信号输入阻抗对应要求不同的信号获取时间。外部信号输入阻抗越大要求的获取时间越长。对应外部信号 10K 的输入阻抗, 为了保证 AD 精度, 要使 ADC 输入信号的获取时间大于或等于 4us。		
2	CMP0_R	运算放大器 0 到 ADC 输入电阻选择位 0: 不接电阻 1: 接 1K 电阻
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 34 SRADCON1 寄存器各位功能表

位地址	标识符	功能
7	SRADEN	ADC 使能位 1: 使能 0: 禁止
6	SRADS	ADC 启动位/状态控制位 1: 开始, 转换过程中 0: 停止, 转换结束 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 1: 转换结果放在 SROFT 寄存器中 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) 1: 使能校正, 即 AD 转换的结果是减去了 SROFT 失调电压值 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) 1: 使能, 上溢或下溢直接是减去后的结果 0: 禁止, 下溢为 000h, 上溢为 fffh															
2	OFFEX	OFFSET 交换 1: 比较器两端信号交换 0: 比较器两端信号不交换 (正端为信号, 负端为参考电压) 仅对 SROFT_SEL[1:0]=00 时有效, 当 SROFT_SEL[1:0]不为 00 时, OFFEX 需配置为 0。															
1:0	VREFS[1:0]	ADC 参考电源选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2">VREFS[1:0]</th> <th>AD 参考电压</th> </tr> <tr> <td>00</td> <td></td> <td>VDD</td> </tr> <tr> <td>01</td> <td></td> <td>PT3.0 外部参考电源输入(由于所有 IO 做输入时, 最大允许电流 I_{max} 为 50mA, 所以外接 LDO 电压时, 需用电阻限流)</td> </tr> <tr> <td>10</td> <td></td> <td>内部参考电压</td> </tr> <tr> <td>11</td> <td></td> <td>PT3.0 输出内部参考电压, PT3.0 可外接电容作为内置参考电压滤波使用, 以提高精度。</td> </tr> </table>	VREFS[1:0]		AD 参考电压	00		VDD	01		PT3.0 外部参考电源输入(由于所有 IO 做输入时, 最大允许电流 I_{max} 为 50mA, 所以外接 LDO 电压时, 需用电阻限流)	10		内部参考电压	11		PT3.0 输出内部参考电压, PT3.0 可外接电容作为内置参考电压滤波使用, 以提高精度。
VREFS[1:0]		AD 参考电压															
00		VDD															
01		PT3.0 外部参考电源输入(由于所有 IO 做输入时, 最大允许电流 I_{max} 为 50mA, 所以外接 LDO 电压时, 需用电阻限流)															
10		内部参考电压															
11		PT3.0 输出内部参考电压, 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 35 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
		0000	AIN0 输入
		0001	AIN1 输入
		0010	AIN2 输入
		0011	AIN3 输入
		0100	AIN4 输入
		0101	保留
		0110	保留
		0111	保留
		1000	AIN8 输入, 内部 1/8VDD
		1001	AIN9 输入, 内部参考电压
		1010	AIN10 输入, 内部接地
		1011	AIN11 输入, 运算放大器 0 的输出 C0 作为 AD 的输入
		1100	保留
其它	内短		
3: 2	保留	需配置为 0	
1: 0	REF_SEL[1:0]	内部参考电压选择	
		REF_SEL[1:0]	内部参考电压
		00	保留
		01	2.0V
		10	3.0V
		11	4.0V

表 36 METCH 寄存器各位功能表

位地址	标识符	功能
0	VREF_OEN	SAR_ADC 内部参考电压输出使能 0: 不输出内部参考电压 1: 输出内部参考电压 注: 当 VREF_OEN=0 时, 把 VREFS[1:0]配置为 2'b11, 也可以通过 PT3.0 输出内部参考电压。

表 37 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 38 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 39 SROFTL 寄存器各位功能表

位地址	标识符	功能
7: 0	SROFT[7:0]	校正数据值的低 8 位

表 40 SROFTH 寄存器各位功能表

位地址	标识符	功能
3: 0	SROFT[11:8]	校正数据值的高 4 位

表 41 TRIM_REF 寄存器各位功能表

位地址	标识符	功能
7: 0	TRIM_REF[7:0]	内部参考电压 TRIMMING 值

TRIM_REF 寄存器的相关用法参照下文 3.5.3 节。

表 42 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 位 AD 转换时间 = (1/ADC 时钟频率) × (12+CALIF+ADC 输入信号获取时间)

表 43 转换时间说明表⁽¹⁾

指令周期	CALIF	SRADCKS	SRADACKS	AD 转换时间
4MHz	0	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		$1 / ((16\text{MHz} / 4) / 8) \times (12 + 0 + 2) = 28\mu\text{s}$	
	1	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	$1 / ((16\text{MHz} / 4) / 8) \times (12 + 1 + 2) = 30\mu\text{s}$			
2MHz	0	00	00	$1 / ((16\text{MHz} / 8) / 1) \times (12 + 0 + 16) = 14\mu\text{s}$
			01	$1 / ((16\text{MHz} / 8) / 1) \times (12 + 0 + 8) = 10\mu\text{s}$
		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		$1 / ((16\text{MHz} / 8) / 8) \times (12 + 0 + 2) = 56\mu\text{s}$	
	1	00	00	$1 / ((16\text{MHz} / 8) / 1) \times (12 + 1 + 16) = 14.5\mu\text{s}$
			01	$1 / ((16\text{MHz} / 8) / 1) \times (12 + 1 + 8) = 10.5\mu\text{s}$
		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	$1 / \left((16\text{MHz} / 8) / 8 \right) \times (12 + 1 + 4) = 68\mu\text{s}$	
			11	$1 / \left((16\text{MHz} / 8) / 8 \right) \times (12 + 1 + 2) = 60\mu\text{s}$	
1MHz	0	00	00	$1 / \left((16\text{MHz} / 16) / 1 \right) \times (12 + 0 + 16) = 28\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 16) / 1 \right) \times (12 + 0 + 8) = 20\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 16) / 1 \right) \times (12 + 0 + 4) = 16\mu\text{s}$	
		01	00	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 0 + 16) = 56\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 0 + 8) = 40\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 0 + 4) = 32\mu\text{s}$	
			11	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 0 + 2) = 28\mu\text{s}$	
		10	00	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 0 + 16) = 112\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 0 + 8) = 80\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 0 + 4) = 64\mu\text{s}$	
			11	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 0 + 2) = 48\mu\text{s}$	
		11	00	$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 0 + 16) = 224\mu\text{s}$	
	01		$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 0 + 8) = 160\mu\text{s}$		
	10		$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 0 + 4) = 128\mu\text{s}$		
	11		$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 0 + 2) = 112\mu\text{s}$		
	1	00	00	$1 / \left((16\text{MHz} / 16) / 1 \right) \times (12 + 1 + 16) = 29\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 16) / 1 \right) \times (12 + 1 + 8) = 21\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 16) / 1 \right) \times (12 + 1 + 4) = 17\mu\text{s}$	
		01	00	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 1 + 16) = 58\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 1 + 8) = 42\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 1 + 4) = 34\mu\text{s}$	
			11	$1 / \left((16\text{MHz} / 16) / 2 \right) \times (12 + 1 + 2) = 15\mu\text{s}$	
		10	00	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 1 + 16) = 116\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 1 + 8) = 84\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 1 + 4) = 68\mu\text{s}$	
			11	$1 / \left((16\text{MHz} / 16) / 4 \right) \times (12 + 1 + 2) = 60\mu\text{s}$	
		11	00	$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 1 + 16) = 232\mu\text{s}$	
	01		$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 1 + 8) = 168\mu\text{s}$		
	10		$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 1 + 4) = 136\mu\text{s}$		
	11		$1 / \left((16\text{MHz} / 16) / 8 \right) \times (12 + 1 + 2) = 120\mu\text{s}$		
	500KHz	0	00	00	$1 / \left((16\text{MHz} / 32) / 1 \right) \times (12 + 0 + 16) = 56\mu\text{s}$
				01	$1 / \left((16\text{MHz} / 32) / 1 \right) \times (12 + 0 + 8) = 40\mu\text{s}$
10				$1 / \left((16\text{MHz} / 32) / 1 \right) \times (12 + 0 + 4) = 32\mu\text{s}$	
11				$1 / \left((16\text{MHz} / 32) / 1 \right) \times (12 + 0 + 2) = 28\mu\text{s}$	
01			00	$1 / \left((16\text{MHz} / 32) / 2 \right) \times (12 + 0 + 16) = 112\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 32) / 2 \right) \times (12 + 0 + 8) = 80\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 32) / 2 \right) \times (12 + 0 + 4) = 64\mu\text{s}$	
			11	$1 / \left((16\text{MHz} / 32) / 2 \right) \times (12 + 0 + 2) = 56\mu\text{s}$	
10			00	$1 / \left((16\text{MHz} / 32) / 4 \right) \times (12 + 0 + 16) = 224\mu\text{s}$	
			01	$1 / \left((16\text{MHz} / 32) / 4 \right) \times (12 + 0 + 8) = 160\mu\text{s}$	
			10	$1 / \left((16\text{MHz} / 32) / 4 \right) \times (12 + 0 + 4) = 128\mu\text{s}$	
			11	$1 / \left((16\text{MHz} / 32) / 4 \right) \times (12 + 0 + 2) = 96\mu\text{s}$	
11		00	$1 / \left((16\text{MHz} / 32) / 8 \right) \times (12 + 0 + 16) = 448\mu\text{s}$		
		01	$1 / \left((16\text{MHz} / 32) / 8 \right) \times (12 + 0 + 8) = 320\mu\text{s}$		
		10	$1 / \left((16\text{MHz} / 32) / 8 \right) \times (12 + 0 + 4) = 256\mu\text{s}$		
		11	$1 / \left((16\text{MHz} / 32) / 8 \right) \times (12 + 0 + 2) = 224\mu\text{s}$		
1	00	00	$1 / \left((16\text{MHz} / 32) / 1 \right) \times (12 + 1 + 16) = 58\mu\text{s}$		

		01	$1 / ((16\text{MHz} / 32) / 1) \times (12 + 1 + 8) = 42\mu\text{s}$	
		10	$1 / ((16\text{MHz} / 32) / 1) \times (12 + 1 + 4) = 34\mu\text{s}$	
		11	$1 / ((16\text{MHz} / 32) / 1) \times (12 + 1 + 2) = 30\mu\text{s}$	
	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	$1 / ((16\text{MHz} / 32) / 2) \times (12 + 1 + 4) = 68\mu\text{s}$	
		10	11	$1 / ((16\text{MHz} / 32) / 2) \times (12 + 1 + 2) = 60\mu\text{s}$
			00	$1 / ((16\text{MHz} / 32) / 4) \times (12 + 1 + 16) = 232\mu\text{s}$
			01	$1 / ((16\text{MHz} / 32) / 4) \times (12 + 1 + 8) = 168\mu\text{s}$
			10	$1 / ((16\text{MHz} / 32) / 4) \times (12 + 1 + 4) = 136\mu\text{s}$
		11	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	$1 / ((16\text{MHz} / 32) / 8) \times (12 + 1 + 2) = 240\mu\text{s}$

3.5.3 使用内部参考电压的校准方法

内部参考电压作参考时, 由于参考电压本身的具有离散性, 所以需要对内部参考电压进行校正。在烧录芯片的时候, 已经在程序区(地址为 0x7F7~0x7FF)烧录了参考电压的校准程序段, 所以在上电的时候, 需要把这个校准系数读出来, 并赋值给 TRIM_REF 寄存器。

读取校准系数用法:

```
movlw xxh
call 07f7h
movwf TRIM_REF
```

给 W 工作寄存器赋值 (xxh=02h/01h/00h 分别表示读取/2.0V/3.0V/4.0V 的校准系数), 使用 call 07f7h 调用校准程序段, 之后对应的内部参考电压的校准系数会存在 W 工作寄存器。最后把 W 工作寄存器值赋给 TRIM_REF 寄存器, 则完成内部参考电压的校准。如果需要用到 2 个或 2 个以上内部参考电压, 只需把这些内部参考电压的校准系数各自保存起来, 在使用的过程中只需把这个系数赋值给 TRIM_REF 寄存器, 即对相应的参考电压进行校准。可参照以下例程:

(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
;*****
AD_ROUT: ;上电先从程序区中读出内部参考电压的校准系数
    movlw 02H
    call 07f7H
    movwf AD20_TEMP ;把内部参考电压2.0V的校准系数放在AD20_TEMP中

    movlw 01H
    call 07f7H
```

```

movwf    AD30_TEMP        ; 把内部参考电压3.0V的校准系数放在AD30_TEMP中

movlw    00H
call     07f7H
movwf    AD40_TEMP        ; 把内部参考电压4.0V的校准系数放在AD40_TEMP中
return

AD_init:                    ;CPUCLK = 16/4 =4MHZ
movlw    11000001B        ;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
movlw    00000001B        ;00H           AIN0输入, 2.0V
movwf    SRADCON2
return

...

ADC30_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    38h, SRADCON2    ;转换第3通道的信号
bsf     SRADCON1, ADEN
movff    AD30_TEMP, TRIM_REF ;内部参考电压3.0V校准
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
...

ADC20_CONVERT:
movlf    00000100B        ;把其它的模拟口设为数字口, 打开当前的模拟口
movwf    PT3CON
movlf    28h, SRADCON2    ;转换第2通道的信号
bsf     SRADCON1, ADEN
movff    AD20_TEMP, TRIM_REF ;内部参考电压2.0V校准
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
...
    
```

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。

方法一：

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_40us
    ...
    bsf sradcon1,6    ;srad=1, 开始转换
    btfsc sradcon1,6 ;检测转换是否完成
    goto $-1
    movlw srادل
    movwf adtmpl_1
    movlw sradh
    movwf adtmph_1
    ...
    bsf sradcon1,2    ;offex=1
    bsf sradcon1,6    ;srad=1, 开始转换
    btfsc sradcon1,6 ;检测转换是否完成
    goto $-1
    movlw srادل
    movwf adtmpl_2
    movlw sradh
    movwf adtmph_2
    aver adtmph_1,adtmpl_1,adtmph_2,adtmpl_2 ;求两次 AD 值平均值, 并保存在
                                           ;adtmph_1,adtmpl_1
    ...

```

方法二：

- 1) 配置 SROFT_SEL[1:0]=2'b11 或 SROFT_SEL[1:0]=2'b10，把失调电压变为正方向的失调电压。
- 2) 初始化时，选择 AIN10 通道(即 AD 输入内部接地)，AD 转换得到的结果为失调电压的值，并把这个值保存在 SROFTL，SROFTH 寄存器中。
- 3) 在测量信号时，把 CALIF，ENOV 都置 1，之后 AD 转换的结果自动减去之前的失调电压值，并保存在 SRADL 和 SRADH 寄存器中。

注：初始化计算得到的失调电压是相对于当前的 ADC 的参考电压，之后只有在相同的参考电压下，才可以进行减法校正。如果有使用过程中有多个不同的参考电压切换（参考电压都是恒定的），在初始化时，可以保存多个失调电压值，之后各自减去失调电压。

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 sroftth
    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 8，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 88h
    movwf sradcon2    ;chs[3:0]=0101,选择通道8,1/8VDD
    bsf  sradcon1,7    ;使能ADC模块
    call delay_10us
    bsf  sradcon1,6    ;srad=1,开始转换
    btfs c sradcon1,6 ;检测转换是否完成
    goto $-1
    movlw srادل
    movwf adtmp1
    movlw sradh
    movwf adtmph
    bcf  status,c
    rlf  adtmp1
    rlf  adtmph        ;AD值乘以2
    rlf  adtmp1
    rlf  adtmph        ;AD值乘以4
    rlf  adtmp1
    rlf  adtmph        ;AD值乘以8,小数点在adtmph的bit3和bit4之间
    ...
```

3.6 比较器/运算放大器

CSU8RP3235/CSU8RP3236 有一个模拟比较器/运算放大器，但是同一个核心电路，不能同时使用，只能分时复用。由 CMP_IO 配置模拟比较器/运算放大器的管脚。CMP_IO=0 时，做模拟比较器/运算放大器 0；否则，做模拟比较器/运算放大器 1。

3.6.1 比较器/运算放大器

模拟比较器/运算放大器 0，带两个模拟输入端 C0P 和 C0N，也可以使用内部参考电压 VREF/2 作为比较器的一个输入，CO 脚可做为比较器/运算放大器的输出。

如果在输入输出之间外间一个反馈电阻，则比较器可做运算放大器使用。做运算放大器使用时，需配置 CMPCON 寄存器的 COS[2:0]位为 3'b100。

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3eh	INTF3	CMPIF								0uuuuuuu
3fh	INTE3	CMPIE								0uuuuuuu
6ah	CMPCON0	COMPEN	COS[2:0]						CMPOUT	0000uuu0
6bh	CMPCON1						CMPOFT_SEL[1:0]			uuuuu00u

表 44 CMPCON 寄存器各位功能表

位地址	标识符	功能																		
7	COMPEN	比较器/运算放大器使能位 1: 使能 0: 禁止																		
6:4	COS[2:0]	比较器/运算放大器选择 <table border="1"> <thead> <tr> <th>COS[2:0]</th> <th>功能描述</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>比较器/运算放大器位使用，PT5.2、PT5.3、PT5.4 做为普通 IO</td> </tr> <tr> <td>001</td> <td>用作比较器，PT5.3 接比较器负端，PT5.4 接比较器正端，PT5.2 做为普通 IO</td> </tr> <tr> <td>010</td> <td>用作比较器，PT5.3 接比较器正端，PT5.4 接比较器负端，PT5.2 做为普通 IO</td> </tr> <tr> <td>011</td> <td>用作比较器，PT5.3 接比较器正端，VREF/2 接比较器负端，PT5.4、PT5.2 做为普通 IO</td> </tr> <tr> <td>100</td> <td>用作运算放大器，PT5.2 做为运算放大器输出口，PT5.4 接运放正端，PT5.3 接运放负端，</td> </tr> <tr> <td>101</td> <td>用作比较器，PT5.3 接比较器负端，PT5.4 接比较器正端，PT5.2 做为比较器输出</td> </tr> <tr> <td>110</td> <td>用作比较器，PT5.3 接比较器正端，PT5.4 接比较器负端，PT5.2 做为比较器输出</td> </tr> <tr> <td>111</td> <td>用作比较器，PT5.3 接比较器正端，VREF/2 接比较器负端，PT5.2 做为比较器输出</td> </tr> </tbody> </table>	COS[2:0]	功能描述	000	比较器/运算放大器位使用，PT5.2、PT5.3、PT5.4 做为普通 IO	001	用作比较器，PT5.3 接比较器负端，PT5.4 接比较器正端，PT5.2 做为普通 IO	010	用作比较器，PT5.3 接比较器正端，PT5.4 接比较器负端，PT5.2 做为普通 IO	011	用作比较器，PT5.3 接比较器正端，VREF/2 接比较器负端，PT5.4、PT5.2 做为普通 IO	100	用作运算放大器，PT5.2 做为运算放大器输出口，PT5.4 接运放正端，PT5.3 接运放负端，	101	用作比较器，PT5.3 接比较器负端，PT5.4 接比较器正端，PT5.2 做为比较器输出	110	用作比较器，PT5.3 接比较器正端，PT5.4 接比较器负端，PT5.2 做为比较器输出	111	用作比较器，PT5.3 接比较器正端，VREF/2 接比较器负端，PT5.2 做为比较器输出
COS[2:0]	功能描述																			
000	比较器/运算放大器位使用，PT5.2、PT5.3、PT5.4 做为普通 IO																			
001	用作比较器，PT5.3 接比较器负端，PT5.4 接比较器正端，PT5.2 做为普通 IO																			
010	用作比较器，PT5.3 接比较器正端，PT5.4 接比较器负端，PT5.2 做为普通 IO																			
011	用作比较器，PT5.3 接比较器正端，VREF/2 接比较器负端，PT5.4、PT5.2 做为普通 IO																			
100	用作运算放大器，PT5.2 做为运算放大器输出口，PT5.4 接运放正端，PT5.3 接运放负端，																			
101	用作比较器，PT5.3 接比较器负端，PT5.4 接比较器正端，PT5.2 做为比较器输出																			
110	用作比较器，PT5.3 接比较器正端，PT5.4 接比较器负端，PT5.2 做为比较器输出																			
111	用作比较器，PT5.3 接比较器正端，VREF/2 接比较器负端，PT5.2 做为比较器输出																			
0	CMPOUT	比较器的比较结果																		

表 45 METCH 寄存器各位功能表

位地址	标识符	功能				
2:1	CMPOFT_SEL[1:0]	比较器/运算放大器失调电压选择 <table border="1"> <thead> <tr> <th>CMPOFT_SEL[1:0]</th> <th>失调电压</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>默认失调电压</td> </tr> </tbody> </table>	CMPOFT_SEL[1:0]	失调电压	00	默认失调电压
CMPOFT_SEL[1:0]	失调电压					
00	默认失调电压					

		01	使失调电压往负方向偏 5mV
		10	使失调电压往正方向偏 5mV
		11	使失调电压往正方向偏 10mV

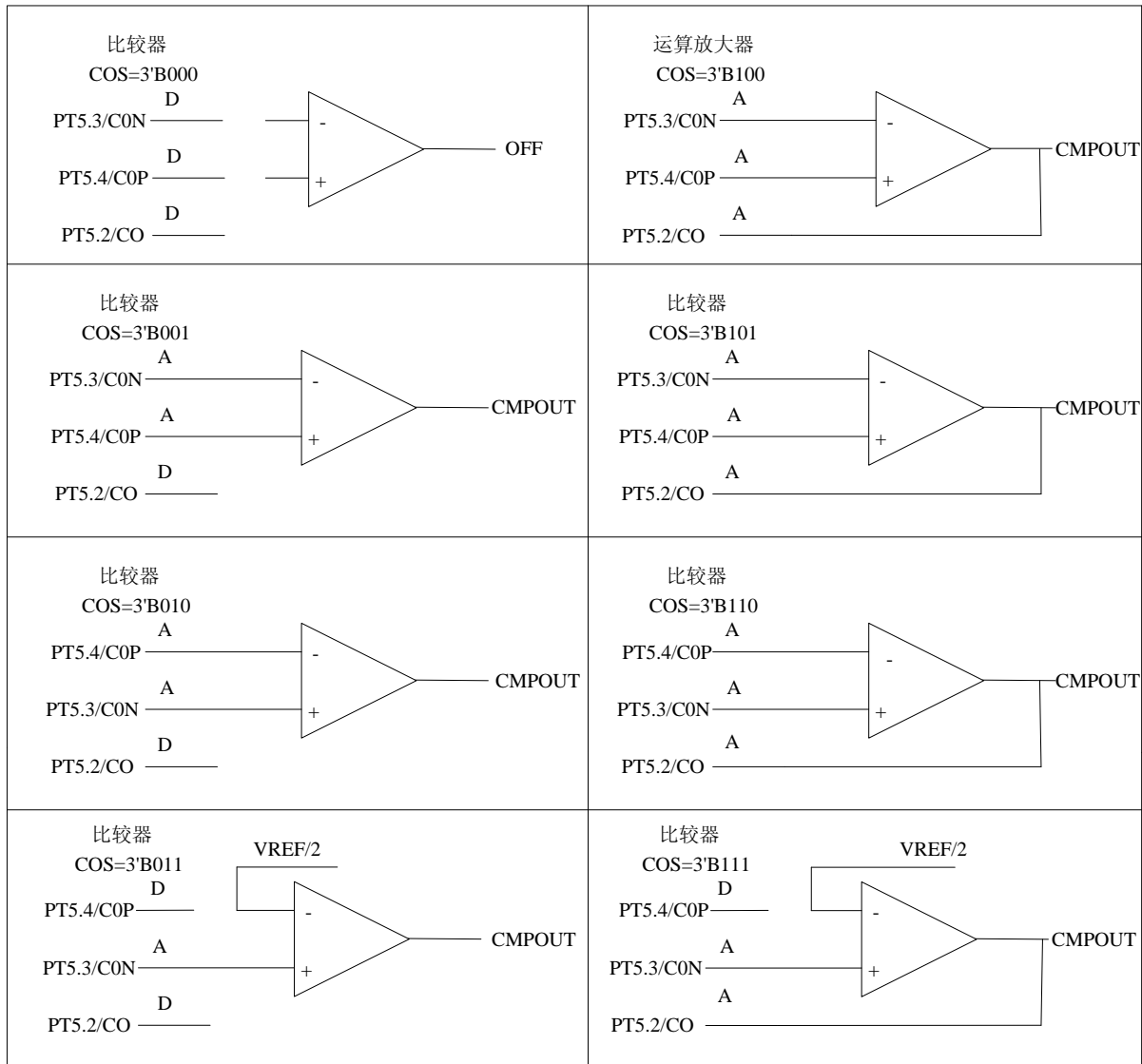
表 46 比较器/运算放大器的寄存器各位功能表

PT5.2/PT1.7	COS[2]	COS[1]	COS[0]	cmp_en	pt5en[2]/pt1en[7]	pt5od[2]
x(GPIO)	0	x	x	x	x	x
Z	1	0	0	0	x	x
Ao	1	0	0	1	x	x
Z	1	0	1	x	0	x
Z	1	0	1	x	x	x
0	1	0	1	0	1	x
Co (1)	1	0	1	1	1	1
Co	1	0	1	1	1	0
Z	1	1	0	x	0	x
Z	1	1	0	x	x	x
0	1	1	0	0	1	x
Co (1)	1	1	0	1	1	1
Co	1	1	0	1	1	0
Z	1	1	1	x	0	x
Z	1	1	1	x	x	x
0	1	1	1	0	1	x
Co (1)	1	1	1	1	1	1
Co	1	1	1	1	1	0

注：（1）此处的比较器输出是开漏输出，需要外部上拉电阻。

Ao 表示运算放大器模拟输出；Co 表示比较器输出；Z 表示高阻

下图为各种情况下比较器及运算放大器的使用示意图



3.6.2 比较器参考电压

用户可以使用内部参考电压 V_{REF} 作为比较器的一个输入，参考电压值可以为 V_{DD} 、内部 2.0/3.0/4.0V 参考电压和 P3.0 输入的外部参考电压，通过 SRADCON2 寄存器的 VREFS[1:0] 位进行选择。

3.6.3 比较器中断

- 1、必须是比较器使能的情况下，比较器中断使能才能有效。
- 2、在比较器使能的情况下，比较器输出端的状态变化都会使比较器结果变化标志位置 1。如果将比较器中断使能，比较器输出端的状态变化会使芯片进入中断，同时置标志位。

3.7 4*5 LCD

CSU8RP3235/CSU8RP3236 中内置一个 4*5 LCD 驱动，1/4duty,1/2bias，但这需要配合软件使用。使用 LCD 的时候，需把 SRADEN 置 1。

当使能 LCD 时，在 $VDD \geq 3V$ 的情况下，PT1.4(COM0)，PT1.5(COM1)，PT1.6(COM2)，PT1.7(COM3)，PT1.0(SEG0)，PT1.1(SEG1)，PT1.2(SEG2)，PT5.4(SEG3)和 PT5.3(SEG4)这 9 个 IO 口输出高电平时，IO 电压为 3V；输出低电压时，IO 口电压为 0V。另外 PT1.4(COM0)，PT1.5(COM1)，PT1.6(COM2)和 PT1.7(COM3)做输入时电压为 1.5V。这就提供了 3V LCD 所需的电压，只需通过程序对这些 IO 的控制就可以进行 LCD 控制。

3.8 输入逻辑电平电压配置

表 47 METCH 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
7eh	METCH	VTHSEL								01uu0000

表 48 METCH 寄存器各位功能表

位地址	标识符	功能						
7	VTHSEL	输入逻辑电平电压控制信号						
		VTHSEL	输入逻辑电平					
		0	符号	参数	最小值	典型值	最大值	单位
		VIH1	数字输入高电平	0.75VDD			V	
			复位输入高电平	0.8VDD			V	
		VIL1	数字输入低电平			0.3VDD	V	
			复位输入低电平			0.2VDD	V	
		1	符号	参数	最小值	典型值	最大值	单位
		VIH2	数字输入高电平	2.0			V	
			复位输入高电平	2.0			V	
			OSCI(PT1.4,PT1.5)	2.5			V	
		VIL2	数字输入低电平			0.8	V	
			复位输入低电平			0.8	V	
			OSCI(PT1.4,PT1.5)			1.0	V	

3.9 烧录模块

烧写器的接口：

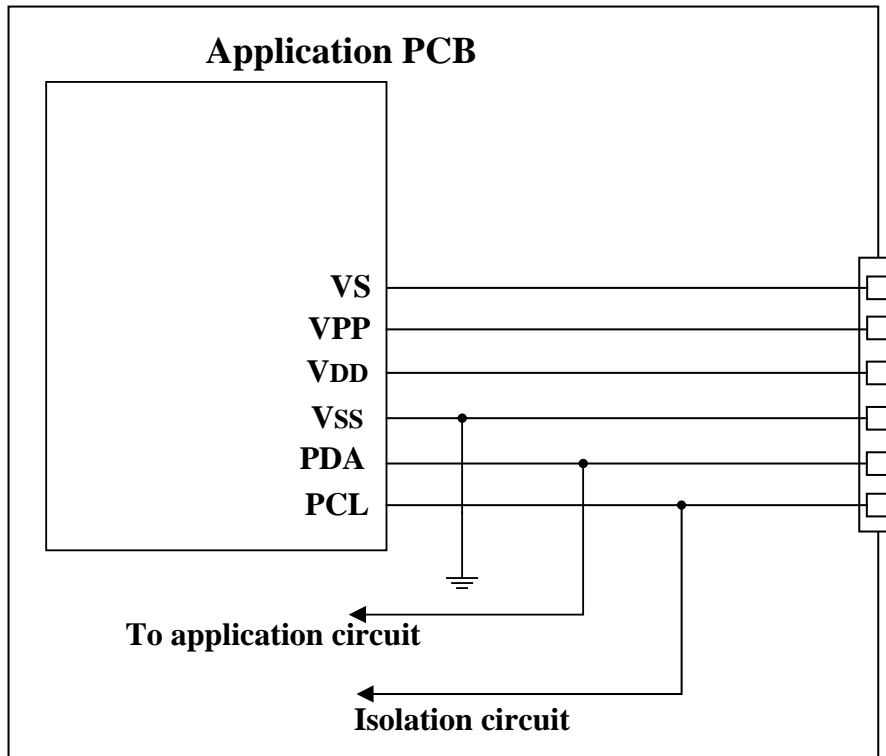


图16 烧写器接口图

表 49 烧录接口说明

端口名称	型式	说明
VPP	输入	PT1[3]端口，烧录电源
VDD	输入	电源正端
VSS	输入	电源负端
PDA	输入/输出	PT1[2]端口，数据信号 ^[1]
PCL	输入	PT1[1]端口，时钟信号 ^[2]
VS	输入	PT3[0] AD 校正信号线

3.10 代码选项

标识符	功能
ICK_SEL	内部晶振频率选择
	2M
	4M
	8M 16M
BDRI_OPO	PT5.0, PT5.1 驱动能力选择位
	20mA@5V 40mA@5V
INS_OP	指令周期配置选项
	指令周期不支持动态配置, 指令周期由代码选项 CLKDIV 决定 指令周期支持动态配置, 指令周期由 MCK 寄存器中的 CPUCLK_SEL[1:0]决定
LCD_OP	LCD 使能位
	使能 LCD 禁止 LCD
CLKDIV	指令周期选择
	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
LVD_SEL	LVD 配置
	VDD 低于 2.0V, LVD 复位系统
	VDD 低于 2.0V, LVD 复位系统; STATUS 的 LVD24 作为 2.4V 的低电压检测器 STATUS 的 LVD30 作为 3.0V 的低电压检测器 STATUS 的 LVD36 作为 3.6V 的低电压检测器
	VDD 低于 2.4V, LVD 复位系统; STATUS 的 LVD30 作为 3.0V 的低电压检测器 STATUS 的 LVD36 作为 3.6V 的低电压检测器
	VDD 低于 3.0V, LVD 复位系统; STATUS 的 LVD36 作为 3.6V 的低电压检测器 VDD 低于 3.6V, LVD 复位系统;
RESET_PIN	复位引脚选择
	PT1.3 作为复位引脚 PT1.3 作为普通输入口
XTAL_PIN	晶振引脚选择
	PT1.1 和 PT1.2 作为普通 IO 口
	PT1.1 作为外部 RC 时钟输入; 或者 PT1.1 通过外部时钟源直接灌入时钟; PT1.2 还是普通 IO 口;
	PT1.1 和 PT1.2 接外部晶振为 32768Hz PT1.1 和 PT1.2 接外部晶振 2M~16MHz; 还可通过 PT1.1 灌时钟, PT1.2 悬空
SECURITY	代码保密位
	使能代码加密 禁止代码加密

4 MCU 指令集

表 50 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	$[f < b >] \leftarrow 0$	1	~
BSF f,b	$[f < b >] \leftarrow 1$	1	~
BTFSC f,b	Jump if $[f < b >] = 0$	1/2	~
BTFSS f,b	Jump if $[f < b >] = 1$	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	~
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	$[Destination < n+1 >] \leftarrow [f < n >]$	1	C,Z
RRF f,d	$[Destination < n-1 >] \leftarrow [f < n >]$	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	~
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 ~17Fh)

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 51 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 ADDLW 08H	在指令执行之前: W=08H 在指令执行之后: 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 (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 ADDPCW	在指令执行之前: W=7FH, PC=0212H 指令执行之后: PC=0292H
例子 2 ADDPCW	在指令执行之前: W=80H, PC=0212H 指令执行之后: PC=0193H
例子 3 ADDPCW	在指令执行之前: W=FEH, PC=0212H 指令执行之后: PC=0211H

3

ADDWF	加工作寄存器到 f
指令格式	ADDWF f,d 0<=f<=1FFH 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。 如果 d 是 0, 结果保存到工作寄存器中。 如果 d 是 1, 结果保存到 f 中。
周期	1
例子 1 ADDWF f 0	指令执行之前: f=C2H W=17H 在指令执行之后 f=C2H W=D9H
例子 2 ADDWF f 1	指令执行之前 f=C2H W=17H 指令执行之后 f=D9H W=17H

4

ADDWFC	将 W f 和进位位相加
指令格式	ADDWFC f, d 0<=f<=1FFH 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 当 d 为 0 时结果保存到工作寄存器 当 d 为 1 时结果保存到 f 中
周期	1
例子 ADDWFC f, 1	指令执行之前 C=1 f=02H W=4DH 指令执行之后 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 ANDLW 5FH	在指令执行之前 W=A3H 在指令执行之后 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<=1FFH 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 如果 d 为 0 结果保存到工作寄存器中 如果 d 为 1 结果保存到 f 中
周期	1
例子 1 ANDWF f, 0	在指令执行之前 W=0FH f=88H 在指令执行之后 W=08H f=88H
例子 2 ANDWF f, 1	在指令执行之前 W=0FH f=88H 在指令执行之后 W=0FH f=08H

7

BCF	清除 f 的某一位
指令格式	BCF f, b 0<=f<=1FFH 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 BCF FLAG 2	指令执行之前： FLAG=8DH 指令执行之后： FLAG=89H

8

BSF	F 的 b 位置 1
指令格式	BSF f, b 0<=f<=1FFH 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 BSF FLAG 2	在指令执行之前 FLAG=89H 在指令执行之后 FLAG=8DH

9

BTFS	如果 bit 测试为 0 则跳转
指令格式	BTFS f, b 0<=f<=1FFH 0<=b<=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 NODE BTFS FLAG 2 OP1: OP2:	在程序执行以前 PC=address(NODE) 指令执行之后 If(FLAG[2])=0 PC=address(OP2) If(FLAG[2])=1 PC=address(OP1)

10

BTFS	如果 bit 测试为 1，则跳转
指令格式	BTFS f, b 0<=f<=1FFH 0<=b<=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 NODE BTFS FLAG 2 OP1: OP2:	在程序执行以前 PC=address(NODE) 指令执行之后 If(FLAG[2])=0 PC=address(OP1) If(FLAG[2])=1 PC=address(OP2)

11

CALL	子程序调用
指令格式	CALL K 0<=K<=7FFH
操作	(top stack)<—PC+1 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<=1FFH
操作	(f)←0
标志位	Z
描述	将 f 的内容清零
周期	1
例子 CLRF WORK	在指令执行之前 WORK=5AH 在指令执行之后 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 CLRWDT	指令执行之后 WDT=0

14

COMF	f 取反
指令格式	COMF f, d 0<=f<=1FFH d=0,1
操作	(目的地址)←NOT(f)
标志位	Z
描述	将 f 的内容取反， 当 d 为 0 时，结果保存到工作寄存器中， 当 d 为 1 时，结果保存到 f 中。
周期	1
例子 COMF f, 0	在指令执行之前 W=88H, f=23H 在指令执行之后 W=DCH, f=23H
例子 2 COMF f, 1	在指令执行之前 W=88H, f=23H 在指令执行之后 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。 如果低半字节的值大于 9 或 DC 为 1 时，低半字节加 6； 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 若 W=25H; ADDLW 39H DAW	在 DAW 指令执行之前 W=25H+39H =64=5EH 在指令执行之后 W= (64) BCD <pre> 25H + 39H ----- 5EH + 06H ----- 64H </pre>

16

DECF	f 减 1
指令格式	DECF f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<←(f)-1
标志位	Z
描述	F 的内容减 1 当 d 为 0 时，结果保存到工作寄存器中 当 d 为 1 时，结果保存到 f 中。
周期	1
例子 DECF f, 0	在指令执行之前 W=88H f=23H 在指令执行之后 W=22H f=23H
例子 2 DECF f, 1	在指令执行之前 W=88H f=23H 在指令执行之后 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<=1FFH 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。 如果 d 为 0, 结果保存到工作寄存器中。 如果 d 为 1, 结果保存到 f 中 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 Node DECFSZ FLAG, 1 OP1: OP2:	在指令执行之前 PC=address(Node) 在指令执行之后 (FLAFG)=(FLAG)-1 If(FLAG)=0 PC=address(OP2) If(FLAG)!=0 PC=address(OP1)

18

GOTO	无条件跳转
指令格式	GOTO K 0<=K<=7FFH
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<=1FFH d=0,1
操作	(目的地址)<-(f)+1
标志位	Z
描述	f 加 1 如果 d 为 0, 结果保存到工作寄存器中 如果 d 为 1, 结果保存到 f 中。
周期	1
例子 INCF f, 0	在指令执行之前 W=88H f=23H 在指令执行之后 W=24H f=23H
例子 2 INCF f, 1	在指令执行之前 W=88H f=23H 在指令执行之后 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<=1FFH 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。 如果 d 为 0, 结果保存到工作寄存器中。 如果 d 为 1, 结果保存到 f 中 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 Node INCFSZ FLAG, 1 OP1: OP2:	在指令执行之前 PC=address(Node) 在指令执行之后 (FLAFG)=(FLAG)+1 If(FLAG)=0 PC=address(OP2) If(FLAG)!=0 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 IORLW 85H	在指令执行之前 W=69H 在指令执行之后 W=EDH

23

IORWF	f 与工作寄存器或
指令格式	IORWF f, d 0<=f<=1FFH 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 当 d 为 0 时, 结果保存到工作寄存器中 当 d 为 1 时, 结果保存到 f 中
周期	1
例子 IORWF f,1	在指令执行前 W=88H f=23H 在指令执行后 W=88H f=ABH

24

MOVFW	传送到工作寄存器
指令格式	MOVFW f 0<=f<=1FFH
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 MOVFW f	在指令执行之前 W=88H f=23H 在指令执行之后 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 MOVLW 23H	在指令执行之前 W=88H 在指令执行之后 W=23H

26

MOVWF	将工作寄存器的值传送到 f 中
指令格式	MOVWF f 0<=f<=1FFH
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 MOVWF f	在指令执行之前 W=88H f=23H 在指令执行之后 W=88H f=88H

27

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

28

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 8 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

29

POP	把 work 和 status 寄存器出栈处理
指令格式	POP
操作	(Top Stack)⇒work/status 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 8 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

RETFIE	从中断返回
指令格式	RETFIE
操作	(Top Stack)⇒PC Pop Stack 1⇒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

31

RETLW	返回，并将立即数送到工作寄存器中
指令格式	RETLW K 0≤K≤FFH
操作	(W)←K (Top Stack)⇒PC 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

32

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

33

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

34

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

35

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

36

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 SUBLW 02H	在指令执行之前 W=01H 在指令执行之后 W=01H C=1(代表没有借位) Z=0(代表结果非零)
例子 2 SUBLW 02H	在指令执行之前 W=02H 在指令执行之后 W=00H C=1(代表没有借位) Z=1(代表结果为零)
例子 2 SUBLW 02H	在指令执行之前 W=03H 在指令执行之后 W=FFH C=0(代表有借位) Z=0(代表结果非零)

37

SUBWF	f 的值减工作寄存器的值
指令格式	SUBWF f, d 0<=f<=1FFH 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。 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 SUBWF f, 1	在指令执行之前 f=33H W=01H 在指令执行之后 f=32H C=1 Z=0
例子 2 SUBWF f, 1	在指令执行之前 f=01H W=01H 在指令执行之后 f=00H C=1 Z=1
例子 3 SUBWF f, 1	在指令执行之前 f=04H W=05H 在指令执行之后 f=FFH C=0 Z=0

38

SUBWFC	带借位的减法
指令格式	SUBWFC f, d 0<=f<=1FFH 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 如果 d 为 0, 结果保存到工作寄存器 如果 d 为 1, 结果保存到 f 中
周期	1
例子 SUBWFC f, 1	在指令执行之前 W=01H f=33H C=1 在指令执行之后 f=32H C=1 Z=0
例子 2 SUBWFC f, 1	在指令执行之前 W=01H f=02H C=0 在指令执行之后 f=00H C=1 Z=1
例子 3 SUBWFC f, 1	在指令执行之前 W=05H f=04H C=0 在指令执行之后 f=FEH C=0 Z=0

39

SWAPF	交换寄存器的值
指令格式	SWAPF f, d 0<=f<=1FFH d=0,1
操作	(des[3:0])<←f[7:4] (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; 把 f 寄存器的低位数据给目标寄存器的高 4 位 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 SWAPF f,1	在指令执行之前 f=ACH 在指令执行之后 f=CAH

40

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)<←(W)^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 XORLW 5FH	在指令执行之前 W=ACH 在指令执行之后 W=F3H

41

XORWF	f 的值与工作寄存器的值异或
--------------	----------------

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

5 电气特性

5.1 极限值

参数	范围	单位
电源 VDD	-0.3~6.0	V
引脚输入电压	-0.3~VDD+0.3	V
工作温度	-40~+125	°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_A = 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	2.2	5	5.5	V
		-40 °C ~+85 °C	2.3	5	5.5	V
FXT	外部晶振频率				16	MHz
I _{max}	IO 口做输入时允许最大电流				50	mA
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
V _{por}	系统电源电压上升速率		0.07			V/ms
T _{cpu}	指令周期	VDD: 2.2V~5.5V	250			ns
VIH1 (VTH_SEL=0)	数字输入高电平	PT1, PT3, PT5	0.75VDD			V
	复位输入高电平		0.8VDD			
VIL1 (VTH_SEL=0)	数字输入低电平	PT1, PT3, PT5			0.3VDD	V
	复位输入低电平				0.2VDD	
VIH2 (VTH_SEL=1)	数字输入高电平	PT1, PT3, PT5(普通 IO)	2.0			V
	复位输入高电平		2.0			
	OSCI	PT1.1, PT1.2(做时钟口时)	2.5			
VIL2 (VTH_SEL=1)	数字输入低电平	PT1, PT3, PT5			0.8	V
	复位输入低电平				0.8	
	OSCI	PT1.1, PT1.2(做时钟口时)			1.0	
IPU	上拉电流	PT1, PT3, PT5; Vin = 0;		35		uA

IOH	高电平输出电流	VOH=0.9VDD; VDD=5V		18		mA
		VOH=0.9VDD; VDD=3V		7.5		mA
		VOH=0.9VDD; VDD=5V PT5.0 和 PT5.1 配置大驱动		37		mA
		VOH=0.9VDD; VDD=3V PT5.0 和 PT5.1 配置大驱动		17		mA
IOL	低电平输出电流	VOL=0.1VDD; VDD=5V		20		mA
		VOL=0.1VDD; VDD=3V		9		mA
		VOL=0.1VDD; VDD=5V PT5.0 和 PT5.1 配置大驱动		52		mA
		VOL=0.1VDD; VDD=3V PT5.0 和 PT5.1 配置大驱动		25		mA
LVD	复位电压/低电压检测电压	2.0V 上电/掉电复位点; -40~85 度	1.9	2.1	2.3	V
		2.4V 上电/掉电复位点; -40~85 度	2.4	2.5	2.7	
		3.0V 上电/掉电复位点; -40~85 度	3.0	3.2	3.4	
		3.6V 上电/掉电复位点; -40~85 度	3.7	4.2	4.4	
IHRC	内置 RC 时钟 16MHz	25°C, 5V	15.84	16.00	16.16	MHz
		-40°C~85°C, 2.3V~5.5V	15.20	16.00	16.80	
	内置 RC 时钟 8MHz	25°C, 5V	7.92	8.00	8.08	MHz
		-40°C~85°C, 2.3V~5.5V	7.60	8.00	8.40	
	内置 RC 时钟 4MHz	25°C, 5V	3.96	4.00	4.04	MHz
		-40°C~85°C, 2.3V~5.5V	3.80	4.00	4.20	
	内置 RC 时钟 2MHz	25°C, 5V	1.98	2.00	2.02	MHz
		-40°C~85°C, 2.3V~5.5V	1.90	2.00	2.10	
WDT	内置看门狗时钟	25°C, 5V	31.36	32.00	32.64	KHz
		-40°C~85°C, 2.3V~5.5V	30.04	32.00	33.60	KHz
Tint0,1	中断触发脉宽	25°C, 5V	Tcpu			ns
IDD1	sleep 模式电流	VDD=3V, 关掉 WDT		0.3		uA
		VDD=3V, 打开 WDT		2.4		uA
		VDD=5V, 关掉 WDT		0.6		uA
		VDD=5V, 打开 WDT		3.2		uA
IDD2	工作电流 IHRC=16MHz	内部振荡器关闭 (fcpu=fosc/4) fosc = 32768Hz,3V		10		uA
		内部振荡器关闭 (fcpu=fosc/4) fosc = 32768Hz,5V		16.5		uA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/4		1.0		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/8		0.7		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/16		0.5		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/4		1.8		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/8		1.2		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/16		0.9		mA

IDD4	工作电流	ERC 模式, VDD=5V, fcpu=fosc/4					mA
		C=47P	R=1KΩ	F=11.1MHZ		8.80	
			R=10KΩ	F=4.1MHZ		4.93	
			R=100KΩ	F=468KHZ		5.03	
			R=1MΩ	F=61KHZ		4.44	
		ERC 模式, VDD=3V, fcpu=fosc/4					
		C=47P	R=1KΩ	F=8.8MHZ		4.62	
			R=10KΩ	F=1.9MHZ		2.65	
			R=100KΩ	F=249KHZ		2.48	
			R=1MΩ	F=30KHZ		2.05	

5.3 ADC 特性 (VDD = 5V, TA = 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)		0.45		mA
		VDD=3V(VDD 作为参考电压)		0.4		mA
ADC Conversion Cycle Time	ADC 转换周期	注意模拟信号的输出阻抗对于 ADC 转换周期的限制, 典型值 10uS 要求输出阻抗不超过 10K		10		uS
INL	积分非线性	SRADACKS[1:0]=01; 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 2.0V	REF_SEL[1:0]=01		2.00		V
	内部参考电压 3.0V	REF_SEL[1:0]=10		3.00		V
	内部参考电压 4.0V	REF_SEL[1:0]=11		4.00		V
IVREF temp drift	内部参考电压温漂			50		ppm
Offset	ADC 失调电压	sroft_sel[1:0]=00		-3		mV
		sroft_sel[1:0]=01		-6		
		sroft_sel[1:0]=10		0.5		
		sroft_sel[1:0]=11		2		mV

5.4 比较器/运算放大器的直流特性 (VDD = 5V, TA = 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
Comp Vos drift	比较器/运算放大器			5*		uV

	的失调电压温漂					
Comp IVR	比较器输入电压范围	VDD=5V, GND=0V	0		5	V
COMP OVS	比较器/运算放大器输出电压范围	VDD=5V, GND=0V, Rload=10KΩ	0		0.3	V
			4.7		5	V
Icomp	比较器工作电流			150		uA
Iopa	运算放大器工作电流			350		uA
OPA amplify multiple	运算放大器内置放大倍数			16		
Large signal voltage gain	运算放大器大信号增益	Rload=2KΩ		40*		V/mV
PSRR	比较器/运算放大器电源电压抑制比			60*		dB
CMRR	比较器/运算放大器共模抑制比			60*		dB
Comp/opa vdd range	比较器/运算放大器工作电压范围		2.3V		5.5	V
Comp Reponse	比较器响应时间			1 ⁽¹⁾		uS
CMP LSB	比较器最小分辨率			1		mV
CMP Vos	比较器失调电压	CMPOFT_SEL[1:0]=00		2		mV
		CMPOFT_SEL[1:0]=01		-3		
		CMPOFT_SEL[1:0]=10		8		
		CMPOFT_SEL[1:0]=11		19		
Comp VCM	比较器、运算放大器共模电压输入范围		0		5	V
OPA output current	运算放大器输出电流	Isorce (v+=1, v-=0, VDD=5V, CO=2V)	20*			mA
		Isink (v+=0, v-=1, VDD=5V, CO=2V)	20*			mA
		Isink (v+=0, v-=1, VDD=5V, CO=200mV)	3*			mA
OPA THD	运放的谐波失真 Vsin=10KHz	VDD=5V, Vopp=5V		60*		dB
		VDD=5V, Vopp=4V		80*		dB
		VDD=5V, Vopp=3V		95*		dB
		VDD=2.3, Vopp=2.3V		55*		dB
		VDD=2.3V, Vopp=1V		75*		dB
Max CAP	允许直接接到运算放大器输出端的最大到地的负载电容				100	pF

(1) 响应时间测量是在比较器一端接(VDD-1.5)/2, 另外一端从 VSS 跳变到 VDD-1.5。

* 表示理论设计值, 未经过实际测试。

5.5 LCD 直流特性 (VDD = 5V, TA = 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	3.0	5	5.5	V
		-40 °C ~+85 °C	3.0	5	5.5	V
Vlcom	COM V1 电压	3V~5V, 25 °C (输出低电平)		0		V

V2com	COM V2 电压	3V~5V, 25 °C (输入不接上拉)		1.50		V
V3com	COM V3 电压	3V~5V, 25 °C (输出高电平)		3.00		V
V1seg	SEG V1 电压	3V~5V, 25 °C (输出低电平)		0		V
V3com	SEG V3 电压	3V~5V, 25 °C (输出高电平)		3.00		V

5.6 16MHz IHRC 时钟频率特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

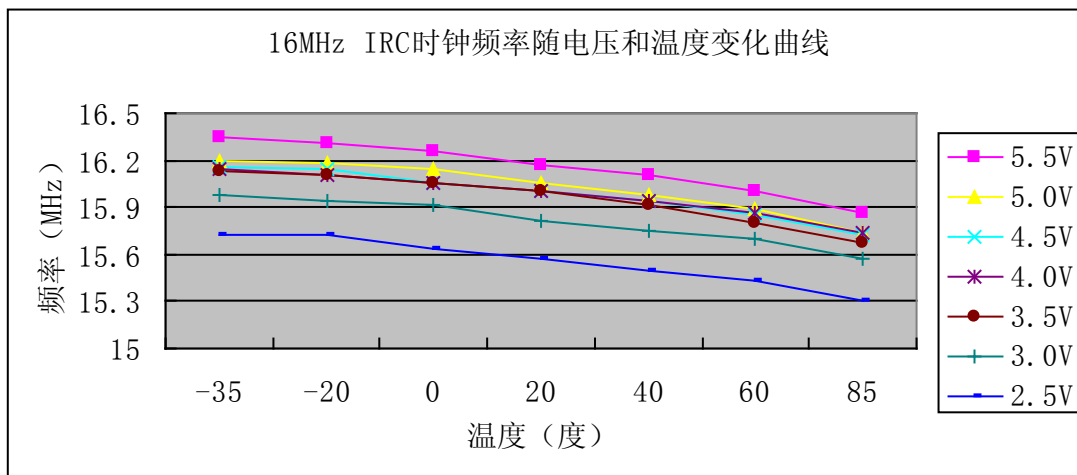


图17 RC 时钟频率的电压和温度特性

5.7 8MHz IHRC 时钟频率特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

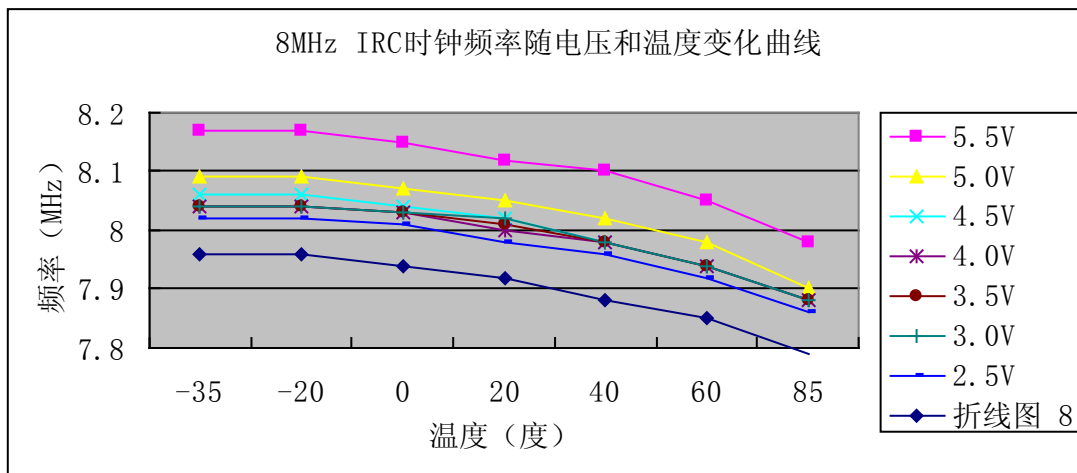


图18 RC 时钟频率的电压和温度特性

5.8 4MHz IHRC 时钟频率特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

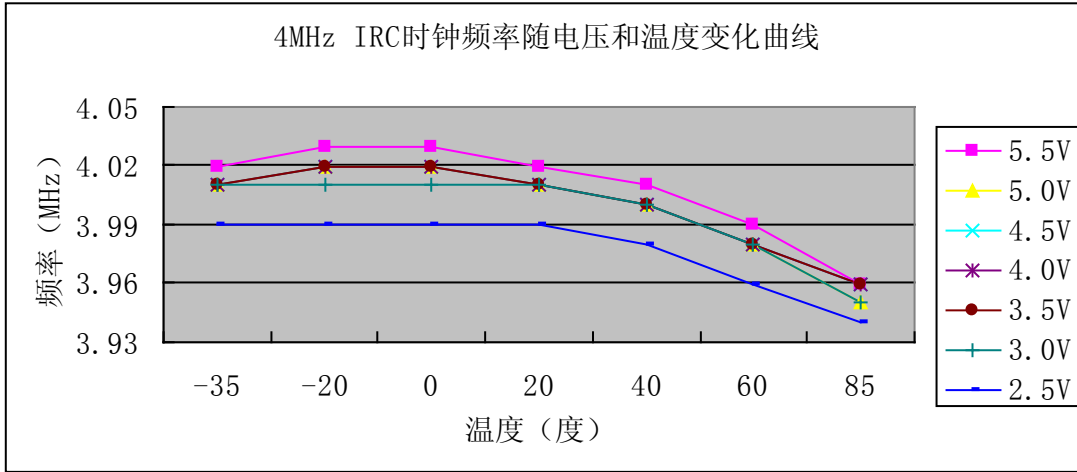


图19 RC时钟频率的电压和温度特性

5.9 2MHz IHRC 时钟频率特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

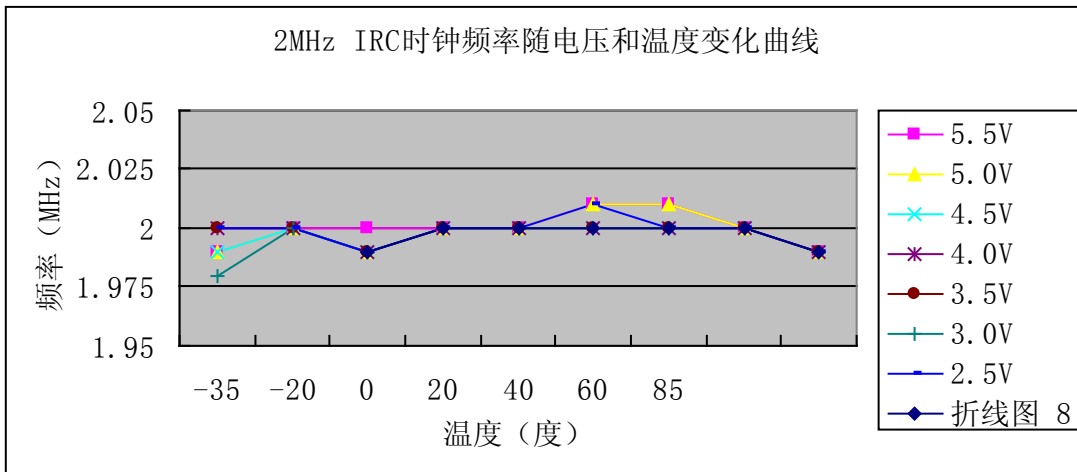


图20 RC时钟频率的电压和温度特性

5.10 32KHz WDT 时钟频率特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

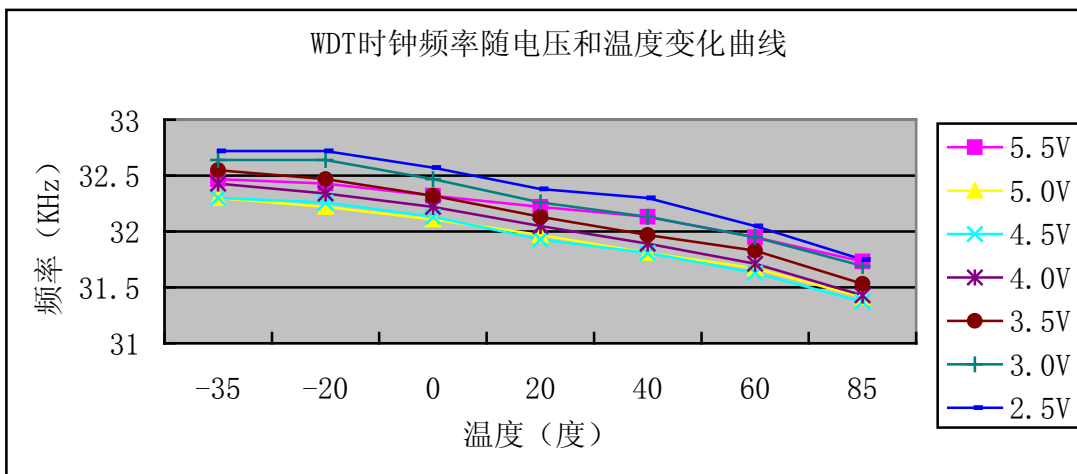


图21 WDT 频率的电压和温度特性

5.11 2.0V 掉电复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

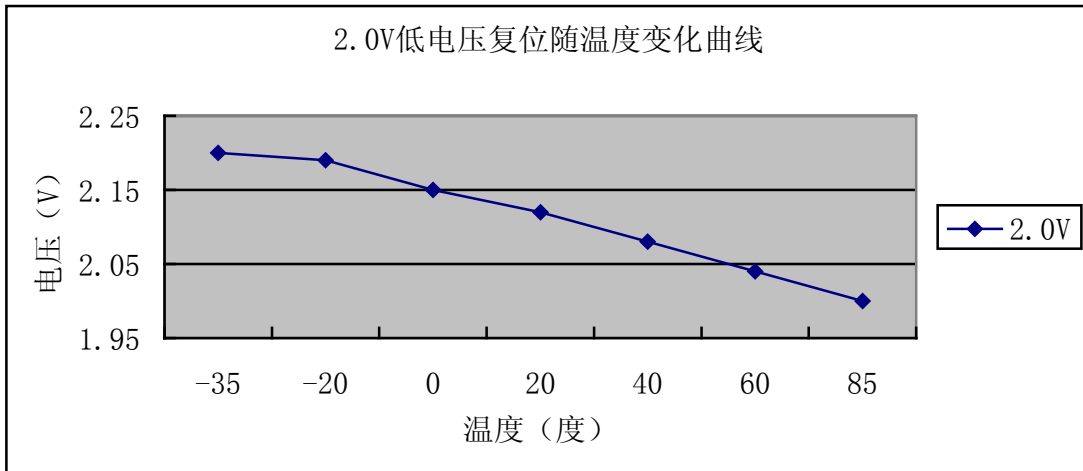
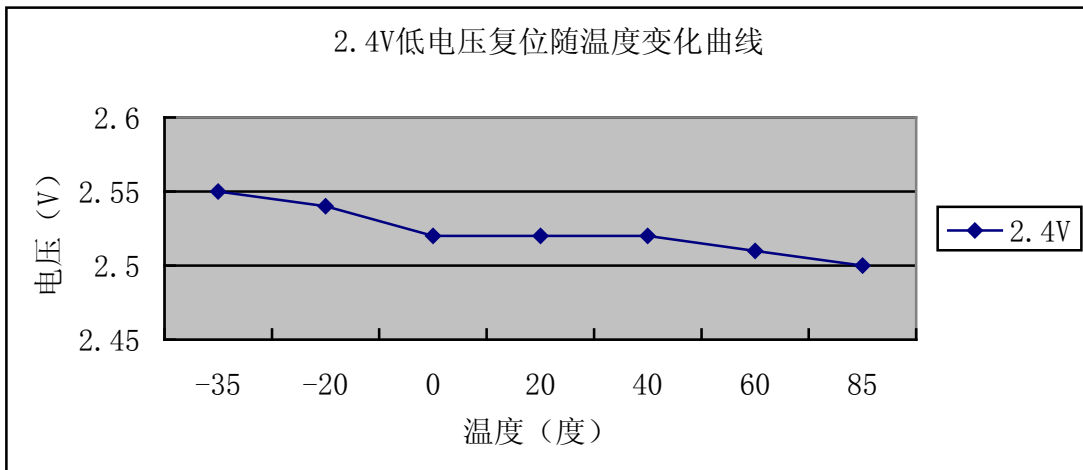


图22 2.0V 掉电复位温度特性

5.12 2.4V 低电压复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。



5.13 3.0V 低电压复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

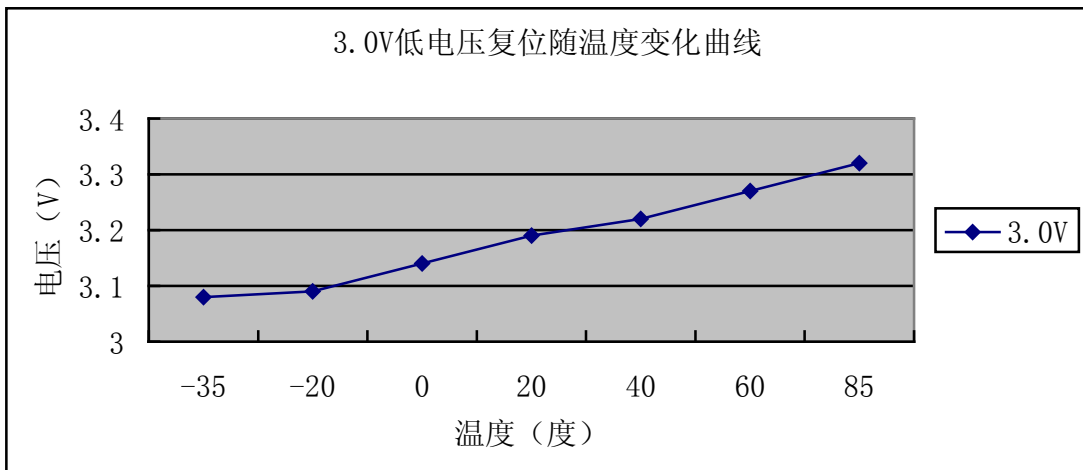


图23 3.0V 低电压复位温度特性

图24

5.14 3.6V 低电压复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

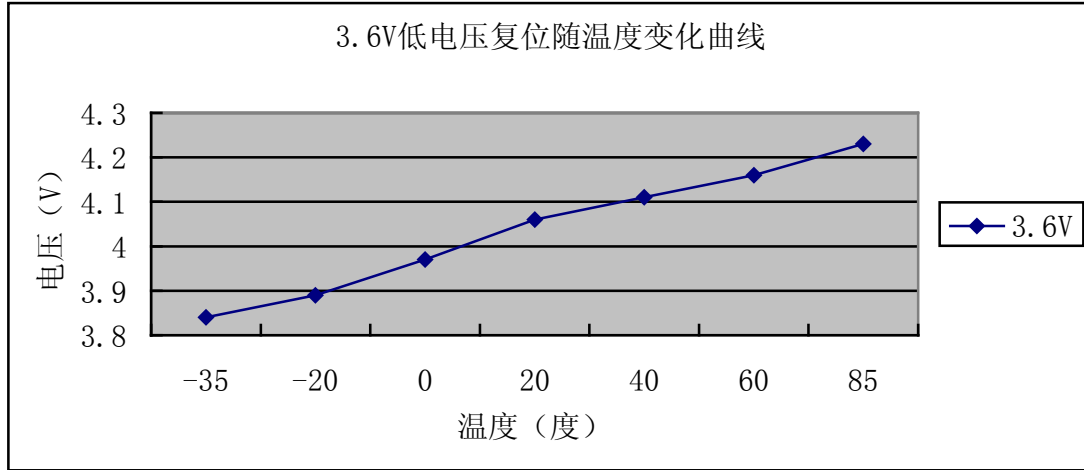


图25 3.6V 低电压复位温度特性

5.15 2.0V 内部参考电压温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

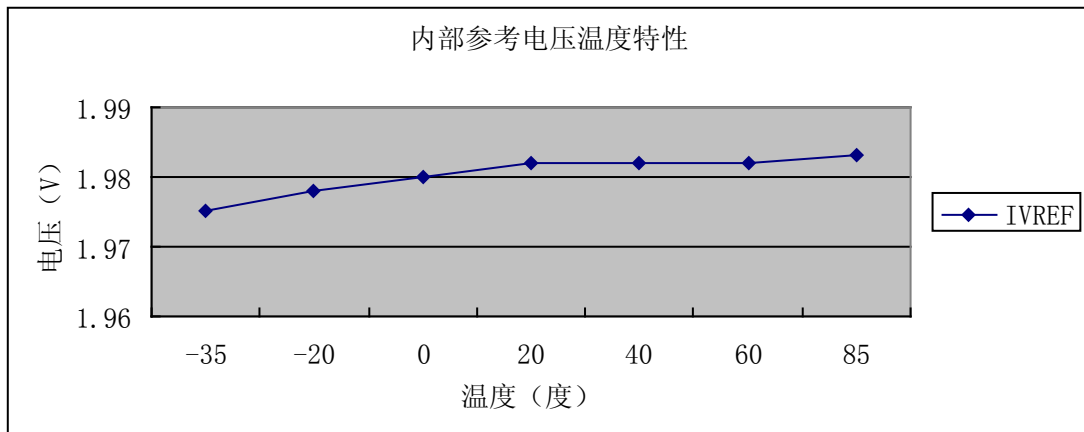


图26 内置参考电压温度特性

5.16 3.0V 内部参考电压温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

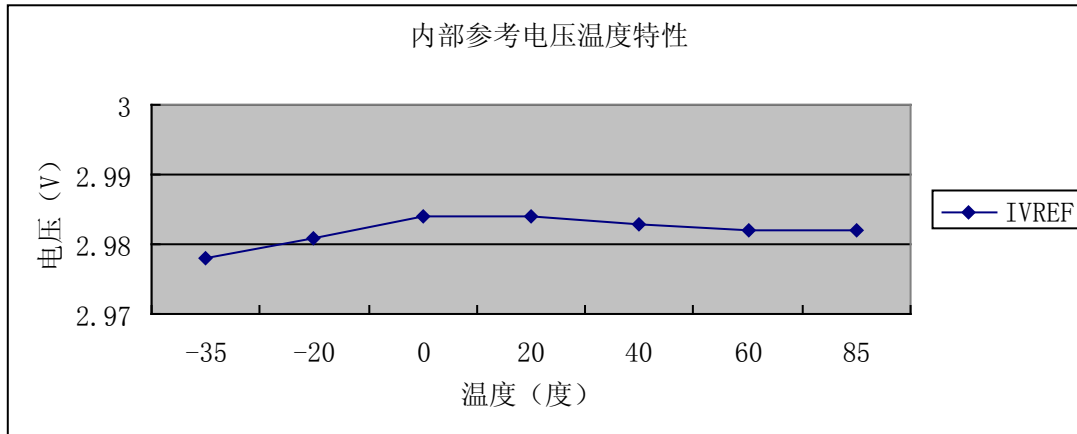


图27 内置参考电压温度特性

5.17 4.0V 内部参考电压温度特性

下图为实际芯片的测试数据（对精度要求高的场合，不建议使用），不同芯片会略微有所差异，仅供参考。

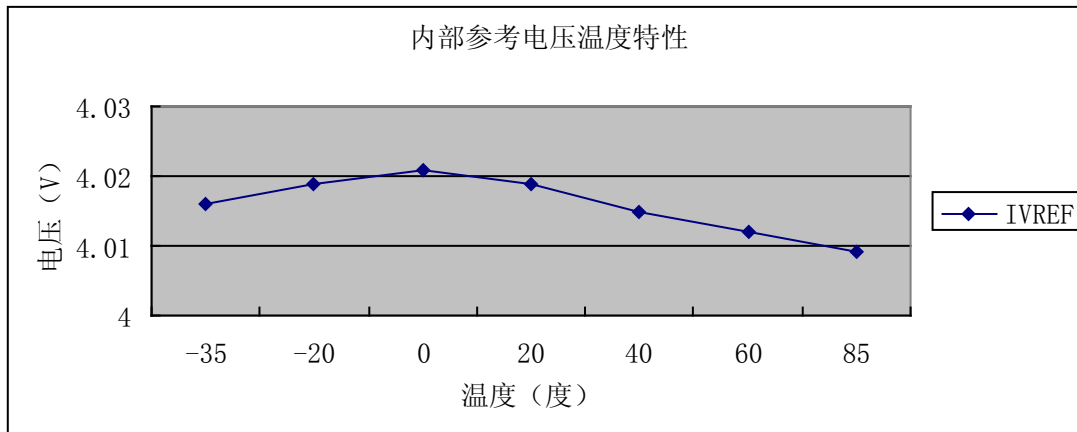
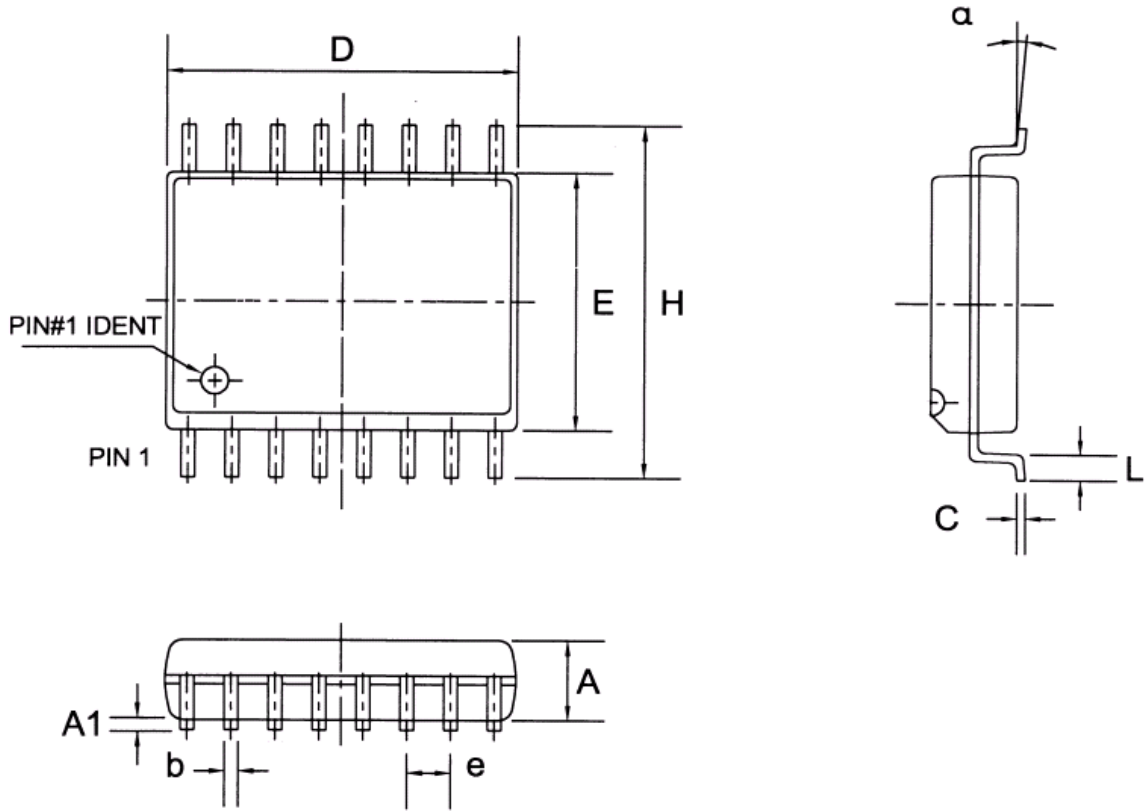


图28 内置参考电压温度特性

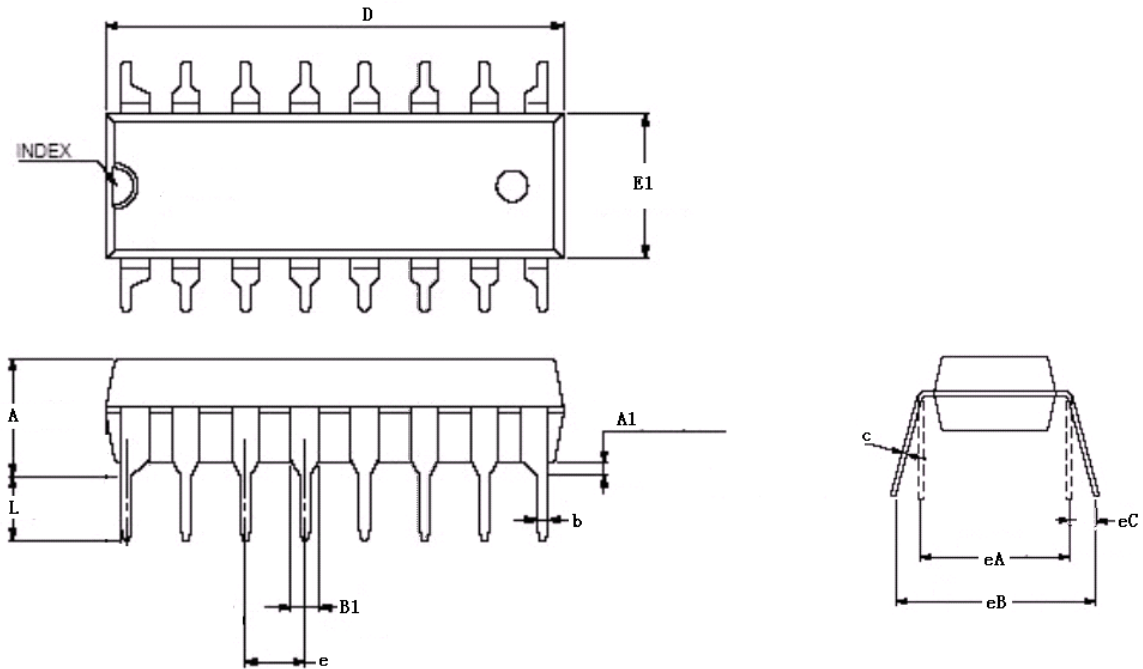
6 封装图

6.1 SOP-16pin



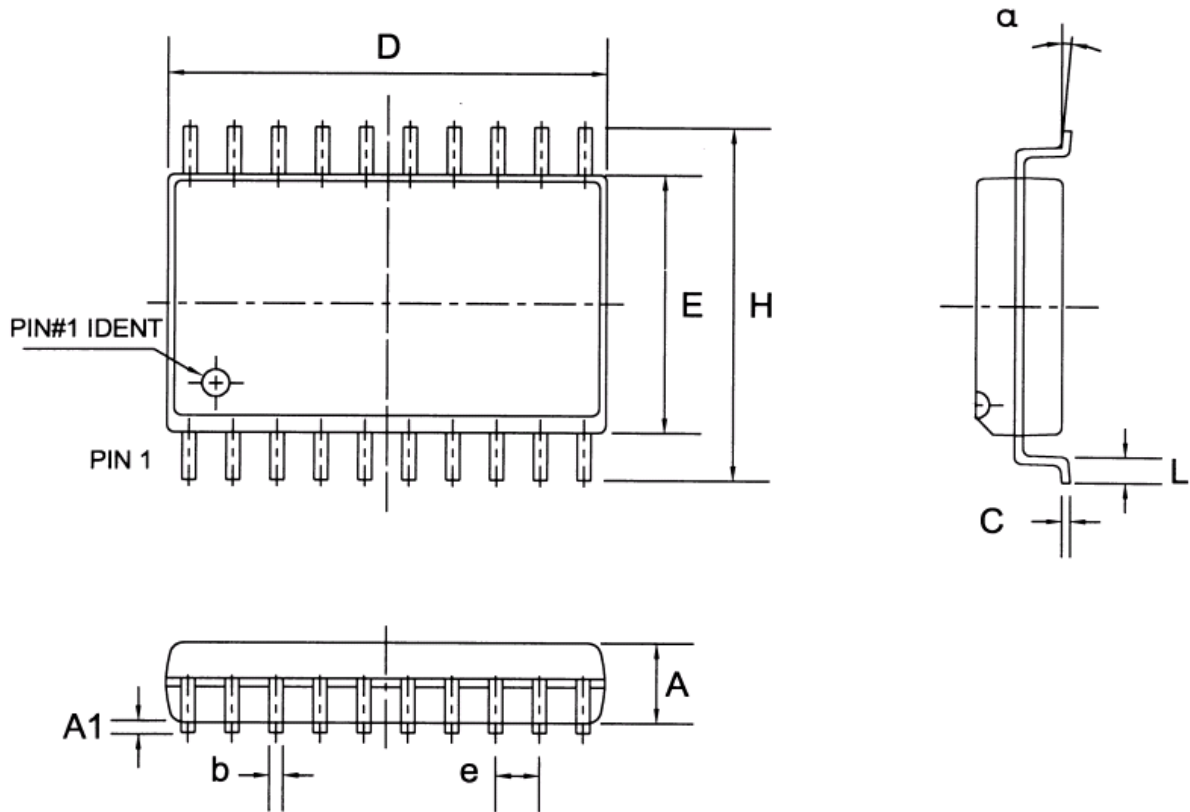
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	1.3	1.40	1.5
A1	0.05	-	0.225
b	0.39	-	0.48
C	0.21	-	0.26
D	9.70	9.90	10.10
E	3.70	3.90	4.10
e	1.27BSC		
α	0°	-	8°
H	5.8	6.0	6.2
L	0.5	-	0.8

6.2 DIP-16pin



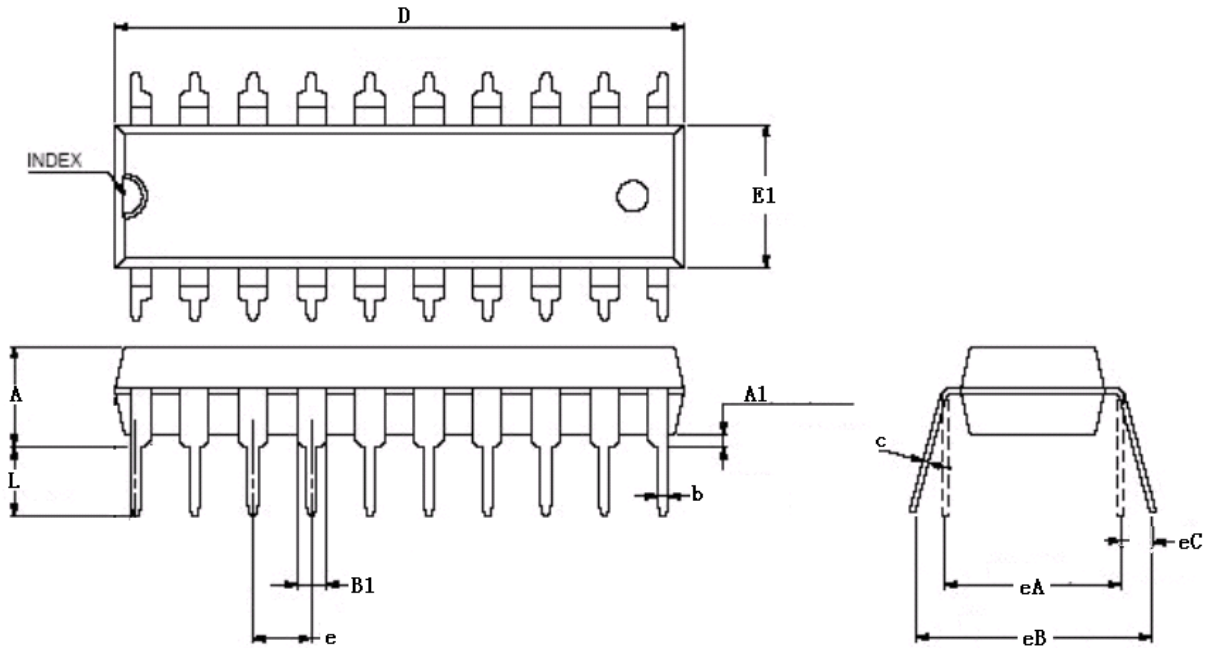
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	3.6	3.8	4.0
A1	0.51	-	-
b	0.44	-	0.53
B1	1.52BSC		
c	0.25	-	0.31
D	18.90	19.10	19.30
E1	6.15	6.35	6.55
e	2.54BSC		
eA	7.62BSC		
eB	7.62	-	9.50
eC	0	-	0.94
L	3.0	-	-

6.3 SOP-20pin



SYMBOLS	MIN	NOR	MAX
	(mm)		
A	2.25	2.30	2.35
A1	0.1	-	0.3
b	0.35	-	0.44
C	0.26	-	0.31
D	12.6	12.8	13.0
E	7.3	7.5	7.7
e	1.27BSC		
α	0°	-	8°
H	10.1	10.3	10.5
L	0.7	-	1.0

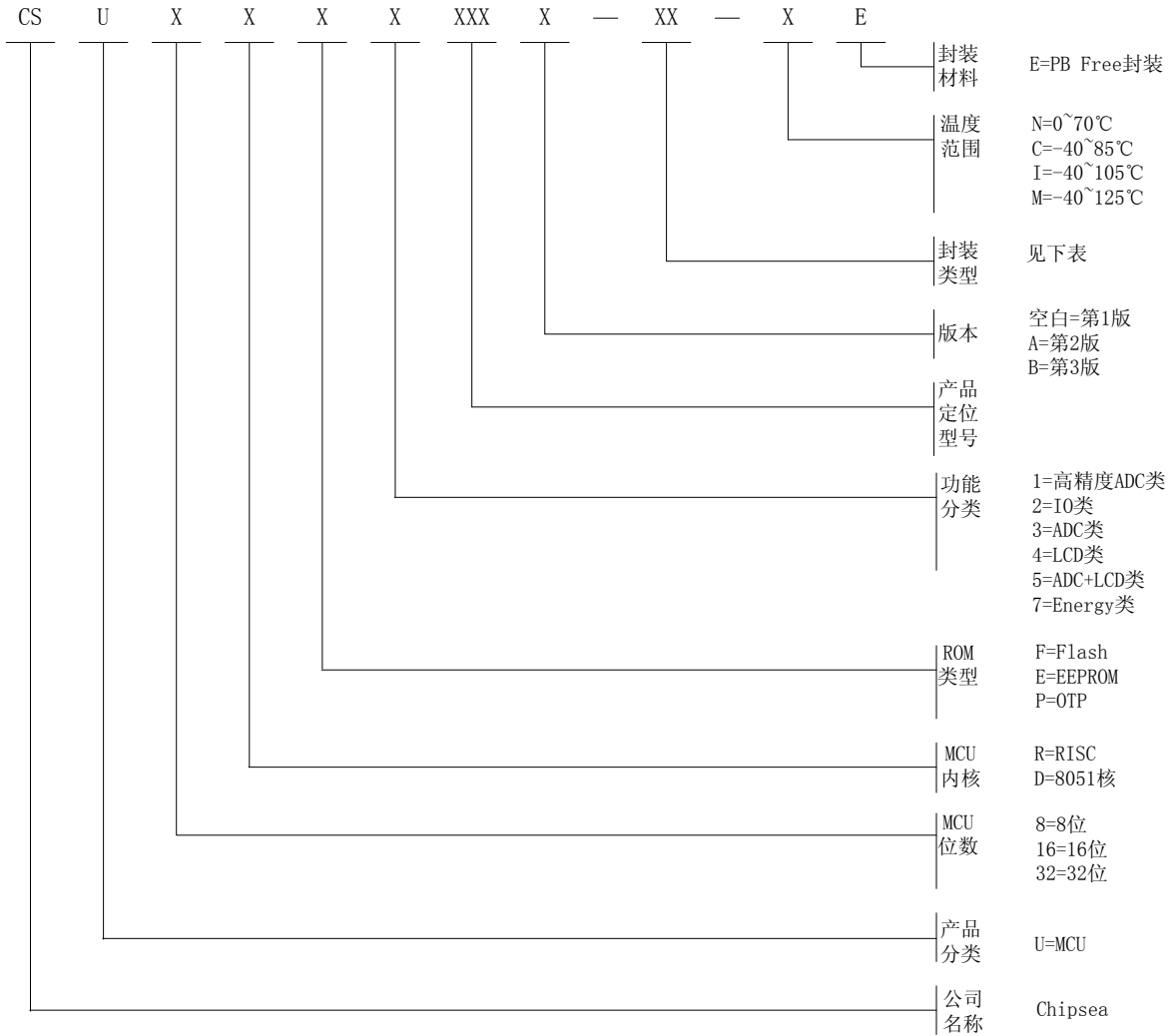
6.4 DIP-20pin



SYMBOLS	MIN	NOR	MAX
	(mm)		
A	3.6	3.8	4.0
A1	0.51	-	-
b	0.44	-	0.53
B1	1.52BSC		
c	0.25	-	0.31
D	26.03	26.23	26.43
E1	6.35	6.55	6.75
e	2.54BSC		
eA	7.62BSC		
eB	7.62	-	9.30
eC	0	-	0.84
L	3.0	-	-

7 单片机产品命名规则

7.1 产品型号说明

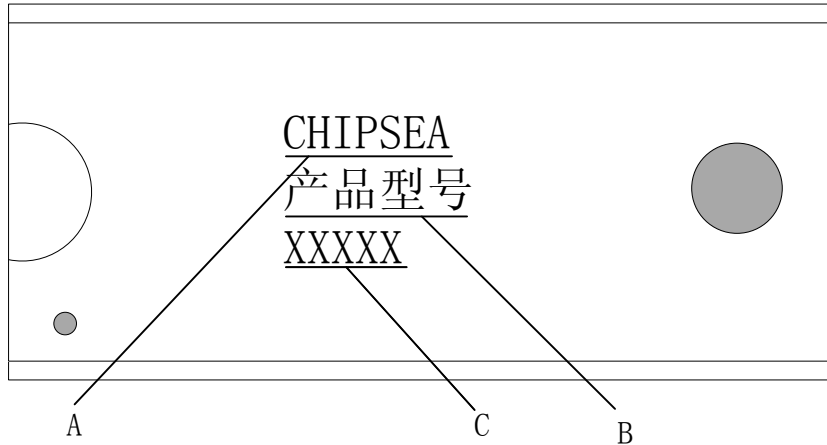


标示符	封装类型
BD	Bonding
DI	DIP
SD	SDIP
SO	SOP
SS	SSOP
TS	TSSOP
QF	QFP
LQ	LQFP
TQ	TQFP
QN	QFN

7.2 命名举例说明

名称	内核	ROM类型	功能分类	产品定位型号	芯片版本	封装形式	工作温度范围	封装材料
CSU8RP3236-SO-CE	8 位 Risc MCU	OTP	ADC	426	第 1 版	SOP	-40~85 °C	无铅封装(PB-Free 封装)

7.3 产品印字说明



芯片正面印字一般有 3 行：

第一行为公司名称，为 **CHIPSEA**。

第二行为产品型号。对于一些小尺寸封装，会对产品型号进行缩减。

第三行为日期码。从左端起算，前两位为公历年号后两位；第三第四位为本年度日历周数，不足两位时左端补 0；最后一位为产品随机号。

例如，CSU8RP3236 印字如下：

