



CSU8RF2111/CSU8RF2112 用户手册

8 位 RISC FLASH MCU

Rev.1.0.1

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

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 FLASH
- 96 字节数据存储器 (SRAM)
- 56 字节的 E2PROM, 用于数据存储
- 只有 43 条单字指令
- 4 级存储堆栈
- 支持在线烧录

振荡器

- 内带 16MHz 振荡器
- 外部 32768Hz 晶振 (RTC) 或 4MHz~16MHz 晶振

外设特性

- 11 位双向 I/O 口, 1 位输入口
- 1 路蜂鸣器输出
- 1 路 PWM 输出
- 2 个内部中断, 1 个外部中断
- 5 个具有唤醒功能的输入口
- 低电压检测 (LVD) 引脚, 内部提供 2.4V、3.6V 电压比较
- 四个开漏输出口

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 1
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 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

- 8/14-pin SOP DIP

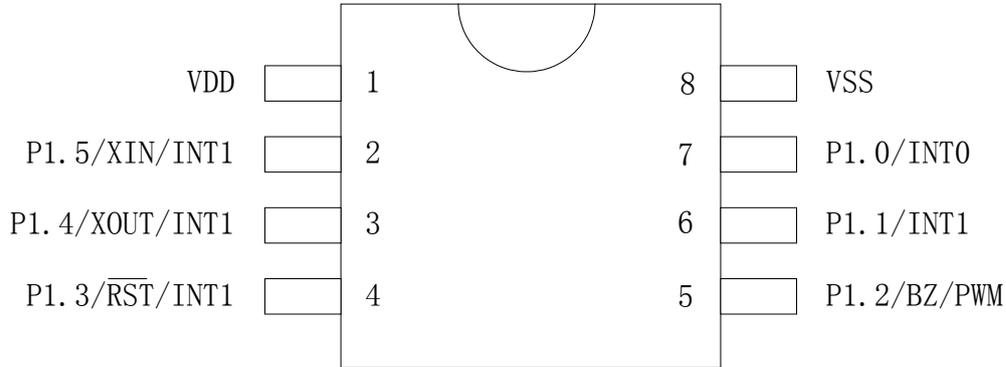
历史修改记录

历史版本.	修改内容	版本日期
Rev. 1.0	换新 LOGO 初版本	2012-12-24
Rev. 1.0.1	修改 CPU 功能模块图	2013-4-11

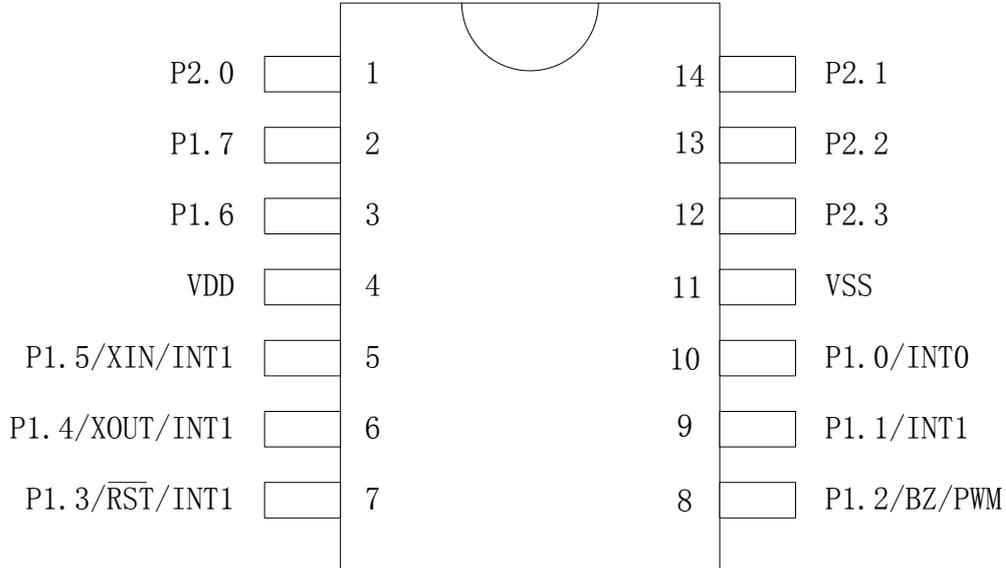
目 录

历史修改记录.....	3
目 录.....	4
1 PIN配置.....	5
2 标准功能.....	7
2.1 CPU核.....	7
2.1.1 存储器.....	9
2.1.2 状态寄存器.....	11
2.1.3 INTE及INTF中断寄存器.....	12
2.2 SFR.....	14
2.3 时钟系统.....	15
2.4 复位系统.....	17
2.5 中断.....	19
2.6 定时器 0.....	21
2.7 定时/计数器 1.....	23
2.7.1 PWM.....	25
2.8 I/O PORT.....	26
3 增强功能.....	30
3.1 HALT和SLEEP模式.....	30
3.2 看门狗(WDT).....	32
3.3 数据E2PROM.....	34
3.4 烧录模块.....	35
3.5 代码选项.....	36
4 MCU指令集.....	37
5 电气特性.....	53
5.1 极限值.....	53
5.2 直流特性 (VDD = 5V, T _A = 25°C, 如无其他说明则都是此条件).....	53
5.3 电气特性曲线图.....	54
6 封装图.....	55
6.1 DIP-14PIN.....	55
6.2 SOP-14PIN.....	56

1 PIN 配置



CSU8RF2111



CSU8RF2112

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
PT2.0	I/O	1	IO
PT1.7	I/O	2	IO
PT1.6	I/O	3	IO
VDD	P	4	电源
PT1.5/XIN/INT1	I/O	5	IO; 外置晶振输入; 外部中断 1 输入, 具有唤醒功能
PT1.4/XOUT/INT1	I/O	6	IO; 外置晶振输出; 外部中断 1 输入, 具有唤醒功能
PT1.3/ \overline{RST} /INT1	I	7	普通输入口; 复位输入; 外部中断 1 输入, 具有唤醒功能
PT1.2/BZ/PWM	I/O	8	IO; 蜂鸣器输出; PWM 输出
PT1.1/INT1	I/O	9	IO; 外部中断 1 输入, 具有唤醒功能
PT1.0/INT0	I/O	10	IO; 外部中断 0 输入, 还具有唤醒功能
VSS	P	11	地
PT2.3	I/O	12	IO
PT2.2	I/O	13	IO
PT2.1	I/O	14	IO

2 标准功能

2.1 CPU核

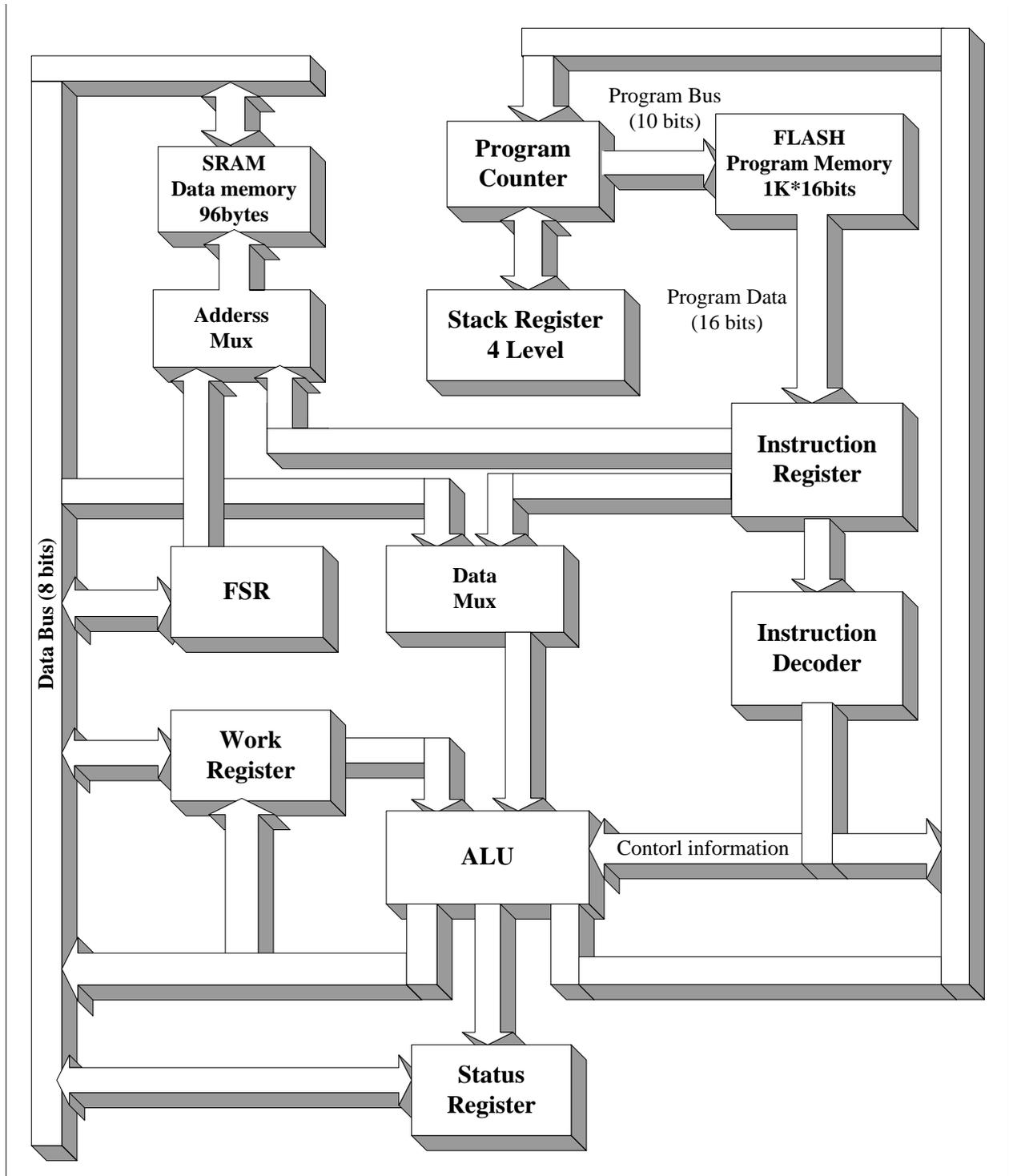


图2 CSU8RF2111/CSU8RF2112 CPU核的功能模块图

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>CSU8RF2111/CSU8RF2112 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） 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>直接数据（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 CSU8RF2111/CSU8RF2112 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	CSU8RF2111/CSU8RF2112 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	CSU8RF2111/CSU8RF2112 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 CSU8RF2111/CSU8RF2112 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~41BH。

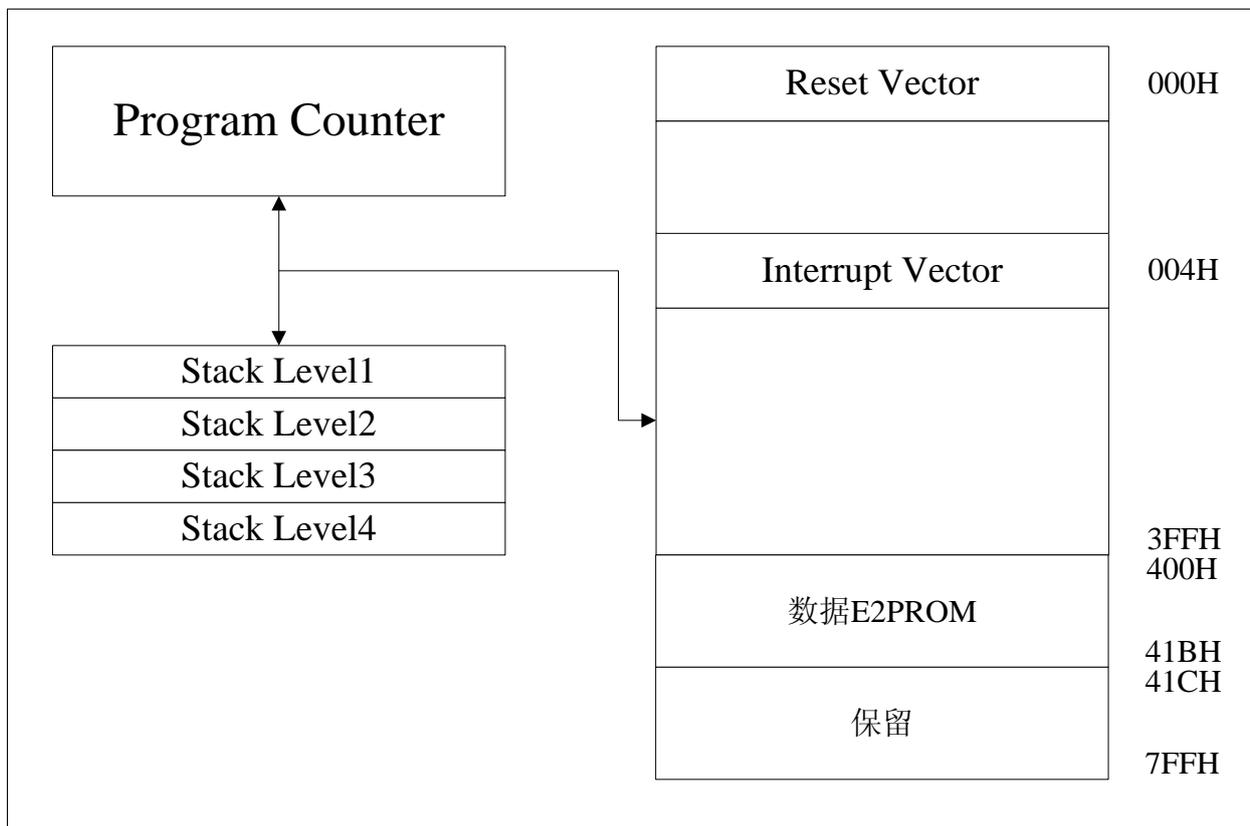


图3 程序存储器

(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 27H 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	27H
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。

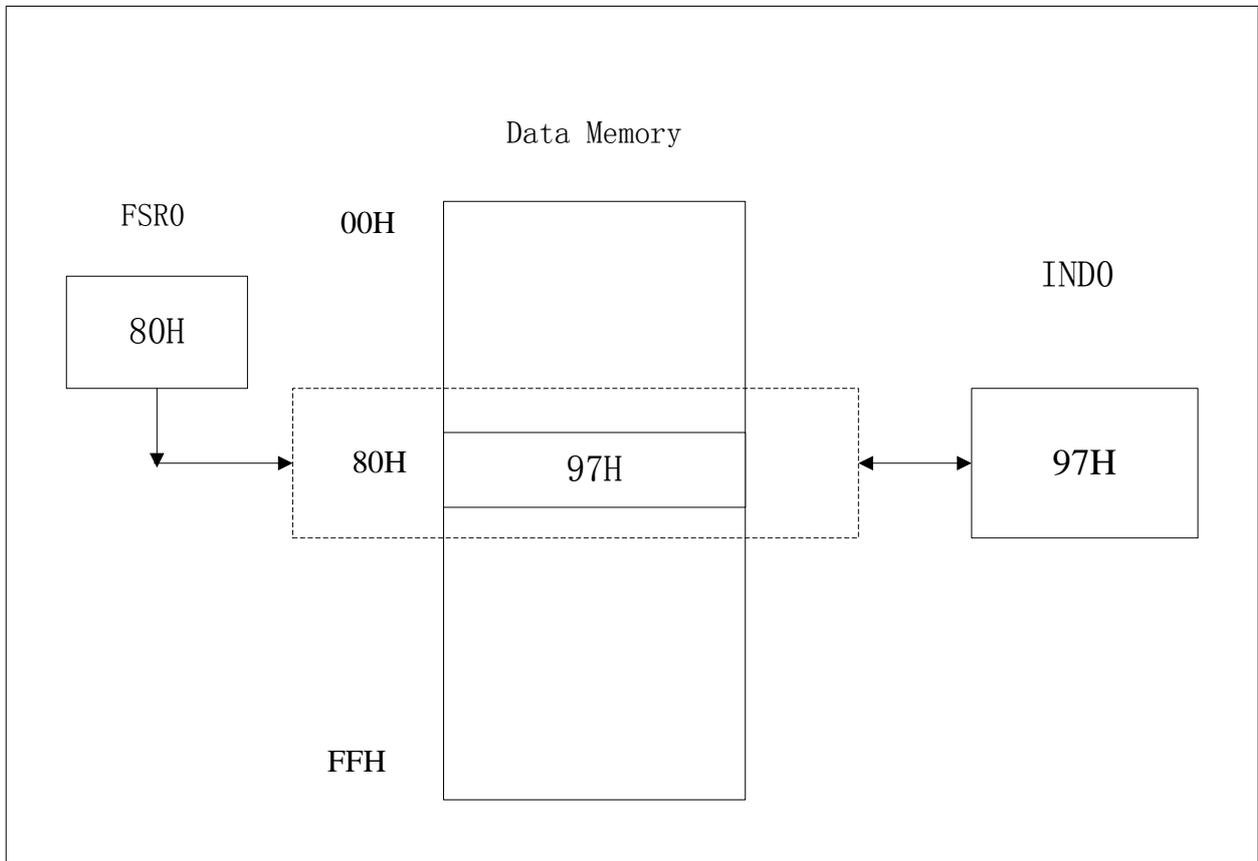


图4 间接地址访问

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。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
	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'b10 和 2'b11 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 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 = 看门狗定时溢出发生
- 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 ADDWF (C) 及 SUBWF (C)

- 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 INTE 及 INTF 中断寄存器

中断系统的入口地址为 004H，各个中断之间没有优先级，靠程序控制各个中断的优先级。只要有中断标志位，就会有中断响应，响应中断之后需要软件将中断标志位清除，否则会不断响应中断。

INTE 及 INTF 寄存器是可读、可写的，包括使能位及标志位，用于中断器件。

INTE 寄存器（地址为 07H）

特性	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
INTE	GIE		TM1IE	TM0IE			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 5 TM1IE: 8-Bit 定时/计数器 1 中断使能标志
1 = 使能定时/计数器 1 中断
0 = 不使能定时/计数器 1 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 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

INTF 寄存器（地址为 06H）

特性	U-0	U-0	R/W-0	R/W-0	U-0	U-0	W-0	R/W-0
INTF			TM1IF	TM0IF			E1IF	E0IF
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 5 TM1IF: 8-Bit 定时/计数器 1 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 1 E1IF: 外部中断 1 中断标志，中断标志位不可读，软件清零。
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

注：E1IF 只可清 0，进入中断时，如果 E1IE 为 1，则需对 E1IF 清 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 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			TM1IF	TM0IF			E1IF	E0IF	uu00uu00
07H	INTE	GIE		TM1IE	TM0IE			E1IE	E0IE	0u00uu00
0AH	EADRH						EDAR[10:8]			uuuuu0xxx
0BH	EADRL	EDAR[7:0]								xxxxxxxx
0CH	EDATH	EDATH[7:0]								xxxxxxxx
0DH	WDTCON	WDTEN					WDTS[2:0]			0uuuu0xxx
0EH	WDTIN	WDTIN[7:0]								xxxxxxxx
0FH	TM0CON	T0EN	T0RATE[2:0]				T0RSTB	T0SEL[1:0]		0xxxu1xx
10H	TM0IN	TM0IN[7:0]								xxxxxxxx
11H	TM0CNT	TM0CNT[7:0]								00000000
12H	TM1CON	T1EN	T1RATE[2:0]			T1CKS	T1RSTB	T1OUT	PWMOUT	0xxxx1xx
13H	TM1IN	TM1IN[7:0]								xxxxxxxx
14H	TM1CNT	TM1CNT[7:0]								00000000
15H	TM1R	TM1R[7:0]								xxxxxxxx
16H	MCK	CST	CST_IN	CST_WDT	EO_SLP				CLKSEL	0010uuu0
20H	PT1	PT1[7:0]								xxxxxxxx
21H	PT1EN	PT1EN[7:0]								00000000
22H	PT1PU	PT1PU[7:0]								00000000
23H	PT1CON	PT11OD	PT1W[3:0]				E1M	E0M[1:0]		00000000
24H	PT2	PT2[3:0]								uuuu0xxx
25H	PT2EN	PT2EN[3:0]								uuuu0000
26H	PT2PU	PT2PU[3:0]								uuuu0000
27H	PT2CON						PT22OD	PT21OD	PT20OD	uuuuu000

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

2.3 时钟系统

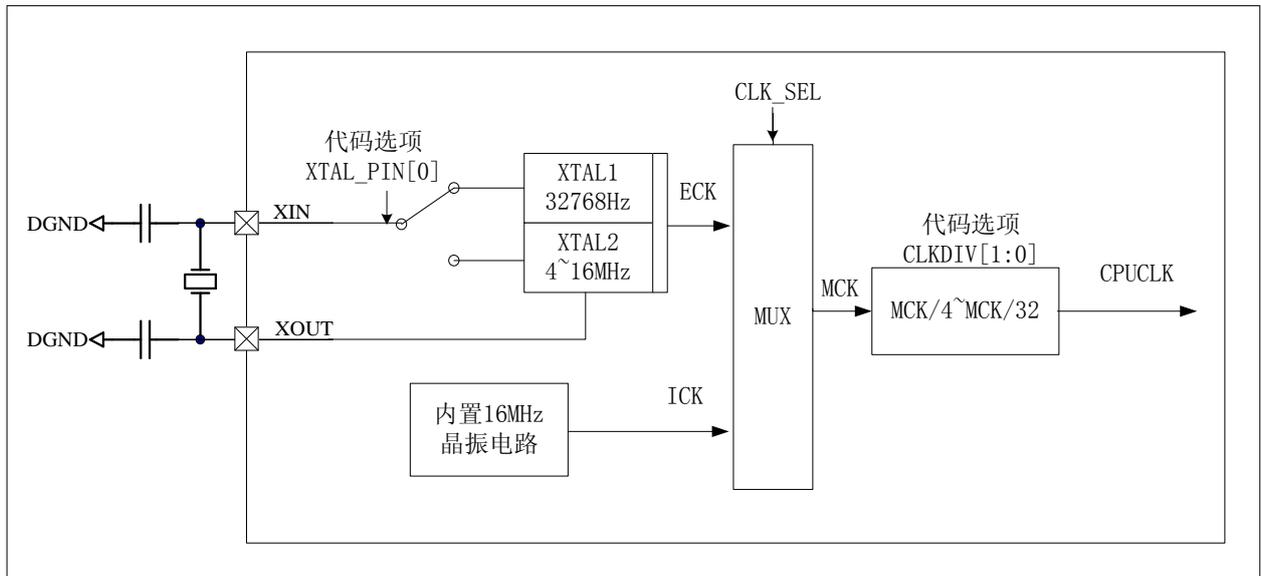


图5 CSU8RF2111/CSU8RF2112 振荡器状态框图

表 5 CSU8RF2111/CSU8RF2112 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

CSU8RF2111/CSU8RF2112 有两个时钟源。一个是内部集成的时钟，16MHz 的时钟供 CPU 工作,另一个是外部时钟。

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						
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_20ms ;延时 20ms
bsf mck, 0      ;切换到外部晶振
nop
nop
bsf mck, 6      ;关闭内部晶振
...
    
```

2.4 复位系统

CSU8RF2111/CSU8RF2112 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）

下表给出了各种复位情况下的 TO，PD 标志位。

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。

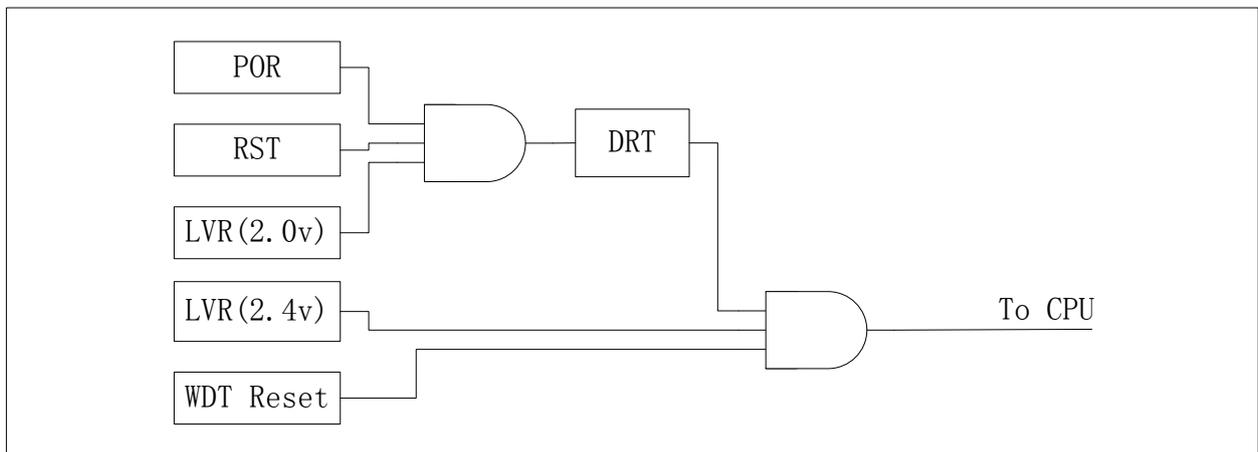


图6 复位电路原理图

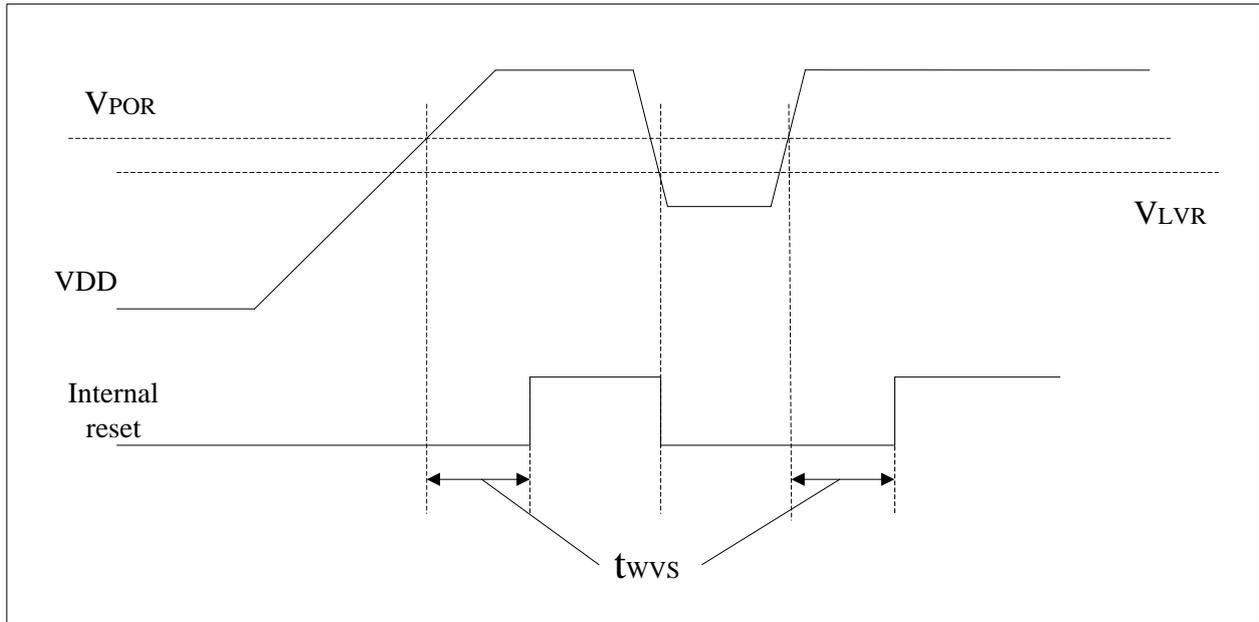


图7 上电复位电路示例及上电过程

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 _{wvs} (测试条件: VDD=5V, T=25℃)	31ms	40ms	48ms

VPOR: 上电复位

VLVR: 低电压复位

t_{wvs}: 等待电压稳定时间

2.5 中断

CSU8RF2111/CSU8RF2112 有 4 个中断源：

1) 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，软件清 0。可唤醒 sleep 或 halt 模式。

PT1CON 寄存器（地址为 23H）

特性	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) 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 1 的使能位，INTF 寄存器中的 E1IF 为中断标志位，软件清 0。外部中断 1 只作唤醒 sleep 或 halt 模式所用；正常工作时，外部中断 1 不会使 CPU 进入中断。

3) 定时器 0 溢出中断

4) 定时/计数器 1 溢出中断

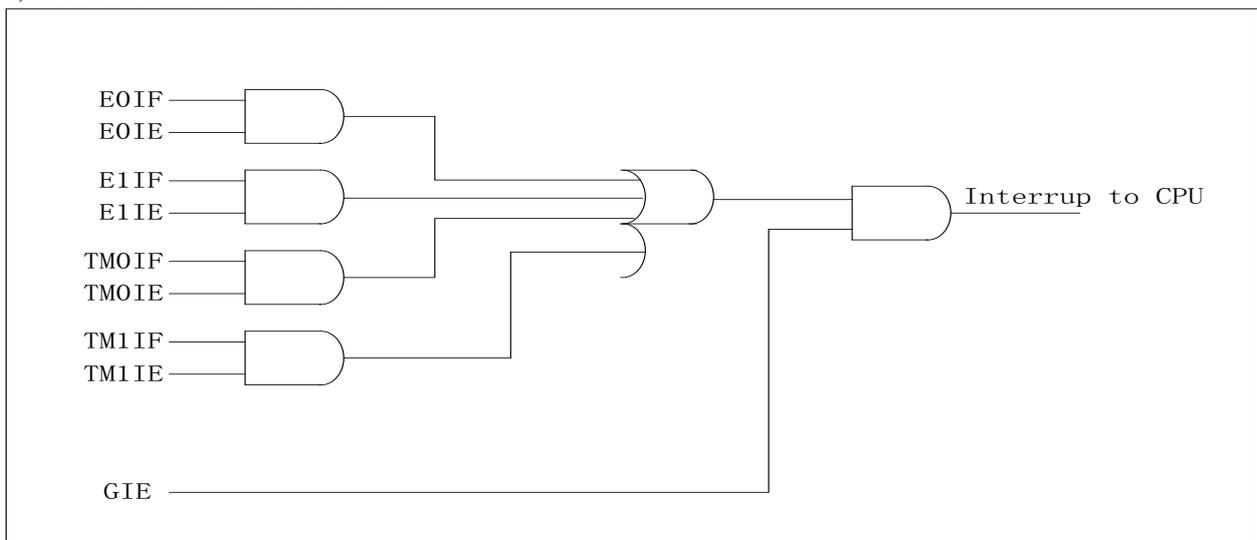


图8 中断逻辑

CSU8RF2111/CSU8RF2112 有 4 个中断源，但只有 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，执行 RETFIE 指令返回到之前的程序，硬件自动将 GIE 置 1。

所有的外部中断、RTC 或 32K WDT 作为时钟源的定时器 0 溢出中断都可唤醒 sleep 睡眠模式和 halt 停止模式。

中断服务程序：

```
...
org 004H
goto int_server
...
int_server:
    btfsc intf, e0if    ;判断外部中断 0 标志
    goto ex0_int
    btfsc intf, tm0if   ;判断定时器 0 中断标志
    goto tm0_int
    btfsc intf, tmlif   ;判断定时/计数器 1 中断标志
    goto tml_int
    goto ex1_int        ;外部中断 1 标志不可读，故排除不是其他中断后，才认为是该中断
    ...
ex1_int:
    bcf intf, elif      ;清除 elif
    ...
```

2.6 定时器 0

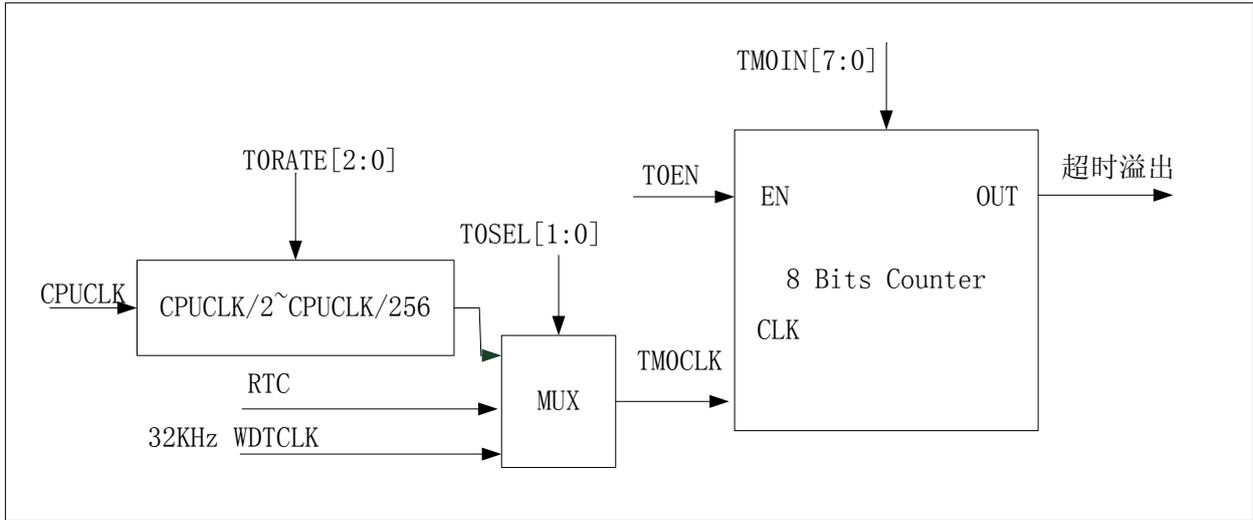


图9 定时器 0 功能框图

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 TM0IN。用户需要设置 TM0IN（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	TOSEL[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 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>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 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>TM0CLK</td> </tr> <tr> <td>01</td> <td>TM0CLK</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	TM0CLK	01	TM0CLK	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	TM0CLK																			
01	TM0CLK																			
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: 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, 寄存器标志位 TMOIF 会自置位, 程序计数器会复位为 04H。

定时器 0 溢出时间计算方法:

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

2.7 定时/计数器 1

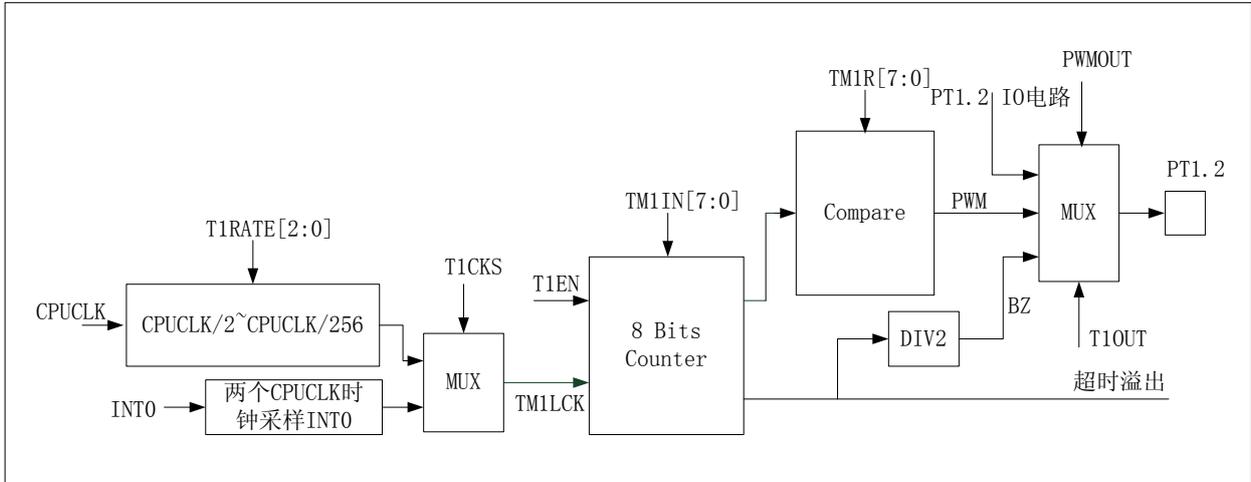


图10 定时/计数器 1 模块的功能框图

定时/计数器 1 模块的输入是 TM1CLK。当用户设置了定时/计数器 1 模块的使能标志，8 bits 计数器将启动，从 00H 递增到 TM1IN。用户需要设置 TM1IN（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；

表 12 定时/计数器 1 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF			TM1IF						uu00uu00
07H	INTE	GIE		TM1IE						0u00uu00
12H	TM1CON	T1EN	T1RATE[2:0]		T1CKS	T1RSTB	T1OUT	PWMOUT		0xxxx1xx
13H	TM1IN	TM1IN[7:0]								xxxxxxxx
14H	TM1CNT	TM1CNT[7:0]								00000000
15H	TM1R	TM1R[7:0]								xxxxxxxx

表 13 TMICON 寄存器各位功能表

位地址	标识符	功能																		
7	T1EN	定时/计数器 1 使能位 1: 使能定时器 1 0: 禁止定时器 1																		
6:4	T1RATE[2:0]	定时/计数器 1 时钟选择 <table border="1"> <thead> <tr> <th>T1RATE [2:0]</th> <th>TM1CLK</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>	T1RATE [2:0]	TM1CLK	000	CPUCLK /2	001	CPUCLK /4	010	CPUCLK /8	011	CPUCLK /16	100	CPUCLK /32	101	CPUCLK /64	110	CPUCLK /128	111	CPUCLK /256
T1RATE [2:0]	TM1CLK																			
000	CPUCLK /2																			
001	CPUCLK /4																			
010	CPUCLK /8																			
011	CPUCLK /16																			
100	CPUCLK /32																			
101	CPUCLK /64																			
110	CPUCLK /128																			
111	CPUCLK /256																			
3	T1CKS	定时/计数器 1 时钟源选择位 1: INT0 作为时钟 0: CPUCLK 的分频时钟																		
2	T1RSTB	定时/计数器 1 复位 1: 禁止定时/计数器 1 复位 0: 使能定时/计数器 1 复位 当将该位为 0 时, 定时器 1 复位后, T1RSTB 会自动置 1																		
1	T1OUT	PT1.2 口输出控制 <table border="1"> <thead> <tr> <th>PWMOUT</th> <th>T1OUT</th> <th>PT1.2 输出控制, 仅当 PT1.2 配置为输出有效</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>IO 输出</td> </tr> </tbody> </table>	PWMOUT	T1OUT	PT1.2 输出控制, 仅当 PT1.2 配置为输出有效	0	0	IO 输出												
PWMOUT	T1OUT	PT1.2 输出控制, 仅当 PT1.2 配置为输出有效																		
0	0	IO 输出																		
0	PWMOUT	<table border="1"> <tbody> <tr> <td>0</td> <td>1</td> <td>蜂鸣器输出</td> </tr> <tr> <td>1</td> <td>0</td> <td>PWM 输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>PWM 输出</td> </tr> </tbody> </table>	0	1	蜂鸣器输出	1	0	PWM 输出	1	1	PWM 输出									
0	1	蜂鸣器输出																		
1	0	PWM 输出																		
1	1	PWM 输出																		

表 14 TM1IN 寄存器各位功能表

位地址	标识符	功能
7: 0	TM1IN[7:0]	定时/计数器溢出值 (溢出值: 1~255)

表 15 TM1CNT 寄存器各位功能表

位地址	标识符	功能
7: 0	TM1CNT[7:0]	定时/计数器 1 计数寄存器, 只读

表 16 TM1R 寄存器各位功能表

位地址	标识符	功能
7: 0	TM1R[7:0]	定时/计数器 1 的 PWM 高电平占空比控制寄存器

操作：

- 1) 设置 TM1CLK，为定时/计数器 1 模块选择输入。
- 2) 设置 TM1IN，选择定时/计数器 1 溢出值。（溢出值：1~255）
- 3) 设置寄存器标志位：TM1IE 与 GIE，使能定时/计数器 1 中断。
- 4) 清零寄存器标志位：T1RSTB，复位定时/计数器 1 模块的计数器。
- 5) 设置寄存器标志位：T1EN，使能定时/计数器 1 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；寄存器标志位 TM1IF 会自复位，程序计数器会复位为 004H。

定时器 1 溢出时间计算方法：

$$\text{定时器 1 溢出时间} = (\text{TM1IN} + 1) / \text{TM1CLK}.$$

蜂鸣器周期计算方法：

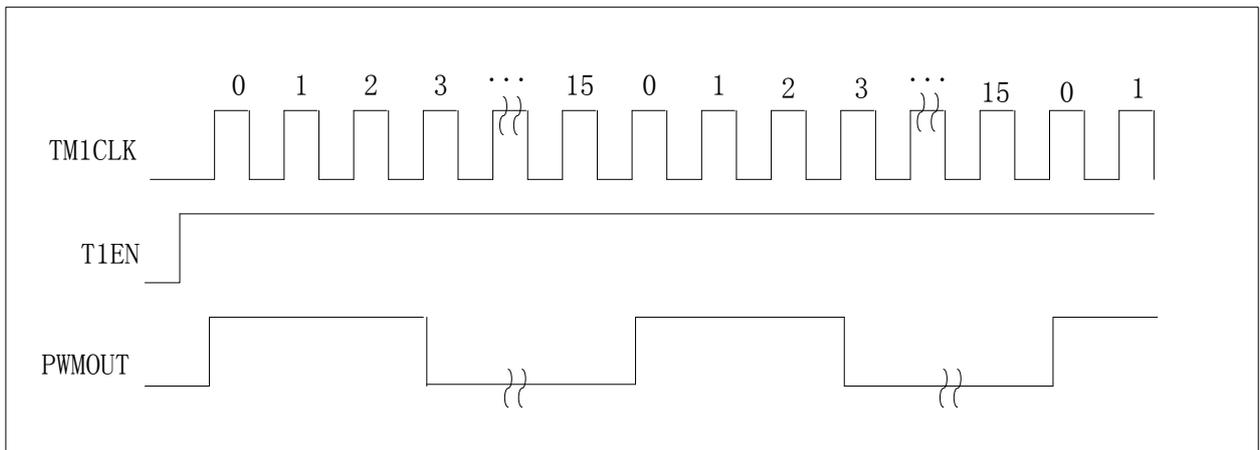
$$\text{蜂鸣器周期} = (\text{TM1IN} + 1) * 2 / \text{TM1CLK}.$$

2.7.1 PWM

操作：

- 1) 设置 TM1CLK，为定时/计数器 1 模块选择输入。
- 2) 设置 TM1IN 来配置 PWM 的周期。
- 3) 设置 TM1R 来配置 PWM 的高电平的脉宽。
- 4) 使能 PWMOUT 输出，配置 PT1.2 为输出端口，之后把 T1EN 置 1 启动定时器。
- 5) PWM 从 PT1.2 输出。

周期为 TM1IN+1，高电平脉宽为 TM1R。如 TM1IN=0FH，TM1R=03H 的 PWM 波形输出如下：



2.8 I/O PORT

表 17 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[7:0]								00000000
22H	PT1PU	PT1PU[7:0]								00000000
23H	PT1CON	PT1IOD	PT1W[3:0]				E1M	E0M[1:0]		00000000
24H	PT2	PT2[3:0]								uuuuuxxxx
25H	PT2EN	PT2EN[3:0]								uuuu0000
26H	PT2PU	PT2PU[3:0]								uuuu0000
27H	PT2CON					PT22OD		PT21OD	PT20OD	uuuuu000

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。CSU8RF2111/CSU8RF2112 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。

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
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

PT1EN 寄存器（地址为 21H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0
PT1EN	PT1EN[7:0]							
	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[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[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

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	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

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	PT11OD	PT1W[3:0]				E1M	E0M[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7 PT11OD: 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

PT2 寄存器 (地址为 24H)

特性	U-0	U-0	U-0	U-0	R/W-X	R/W-X	R/W-X	R/W-X
PT2					PT2[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7-0 PT2[3:0]: GPIO2 口数据标志位
- PT2[3] = GPIO2 bit 3 的数据标志位
- PT2[2] = GPIO2 bit 2 的数据标志位
- PT2[1] = GPIO2 bit 1 的数据标志位
- PT2[0] = GPIO2 bit 0 的数据标志位

PT2EN 寄存器 (地址为 25H)

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2EN					PT2EN[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7-0 PT2EN[3:0]: GPIO 2 口输入/输出控制标志
- PT2EN[3] = GPIO2 bit 3 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[2] = GPIO2 bit 2 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[1] = GPIO2 bit 1 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[0] = GPIO2 bit 0 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT2PU 寄存器 (地址为 26H)

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2PU					PT2PU[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7-0 PT2PU[3:0]: GPIO2 口上拉电阻使能标志
- PT2PU[3] = GPIO2 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT2PU[2] = GPIO2 bit 2 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT2PU[1] = GPIO2 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
- PT2PU[0] = GPIO2 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻



PT2CON 寄存器（地址为 27H）

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
PT2CON						PT22OD	PT21OD	PT20OD
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 2 PT22OD: PT2.2 漏极开路使能位

0 = 禁止 PT2.2 漏极开路

1 = 使能 PT2.2 漏极开路

Bit 2 PT21OD: PT2.1 漏极开路使能位

0 = 禁止 PT2.1 漏极开路

1 = 使能 PT2.1 漏极开路

Bit 2 PT20OD: PT2.0 漏极开路使能位

0 = 禁止 PT2.0 漏极开路

1 = 使能 PT2.0 漏极开路

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



3 增强功能

3.1 Halt和Sleep模式

CSU8RF2111/CSU8RF2112 支持低功耗工作模式。为了使 CSU8RF2111/CSU8RF2112 处于待机状态，可以让 CPU 停止工作使 CSU8RF2112 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 I/O 口是接到 VDD 或 VSS 电平。

注：

所有的中断都可唤醒 sleep 睡眠模式和 halt 停止模式。

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
clrf pt1 ;将 PT1 输出为低
clrf pt2up ;断开 pt2 上拉电阻
clrf pt2en ;PT2 口用作输入口
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
clrf pt1 ;将 PT1 输出为低
clrf pt2up ;断开 PT2 上拉电阻
clrf pt2en ;PT2 口用作输入口
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)

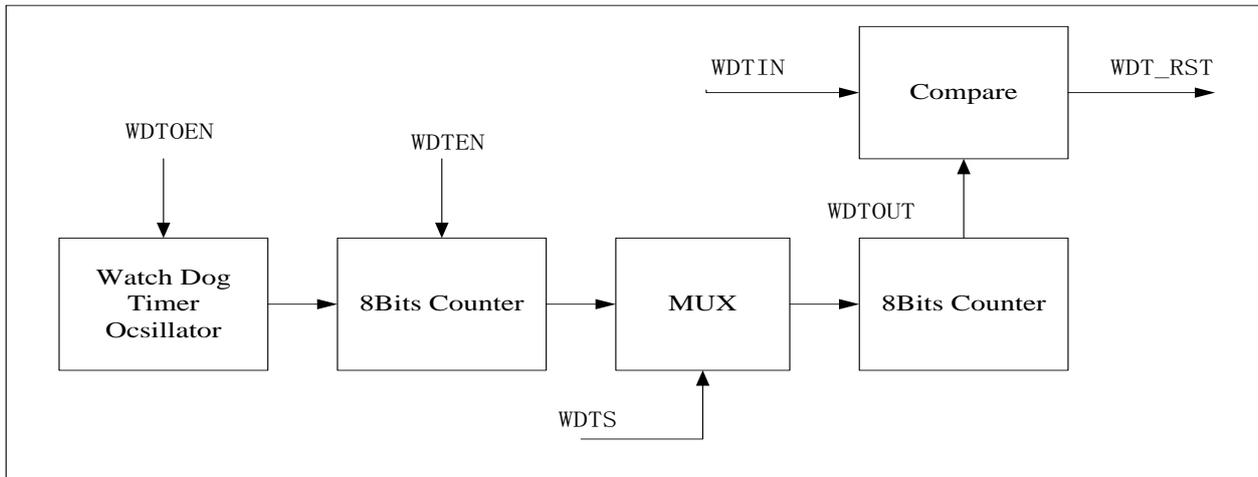


图11 看门狗定时器功能框图

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 WDT_IN 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 18 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	WDT_IN[7:0]								xxxxxxxx

WDTCON 寄存器（地址为 0DH）

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



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。

3.3 数据E2PROM

CSU8RF2111/CSU8RF2112 具有 56 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~41BH。

表 19 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
...
    
```

3.4 烧录模块

烧写器的接口:

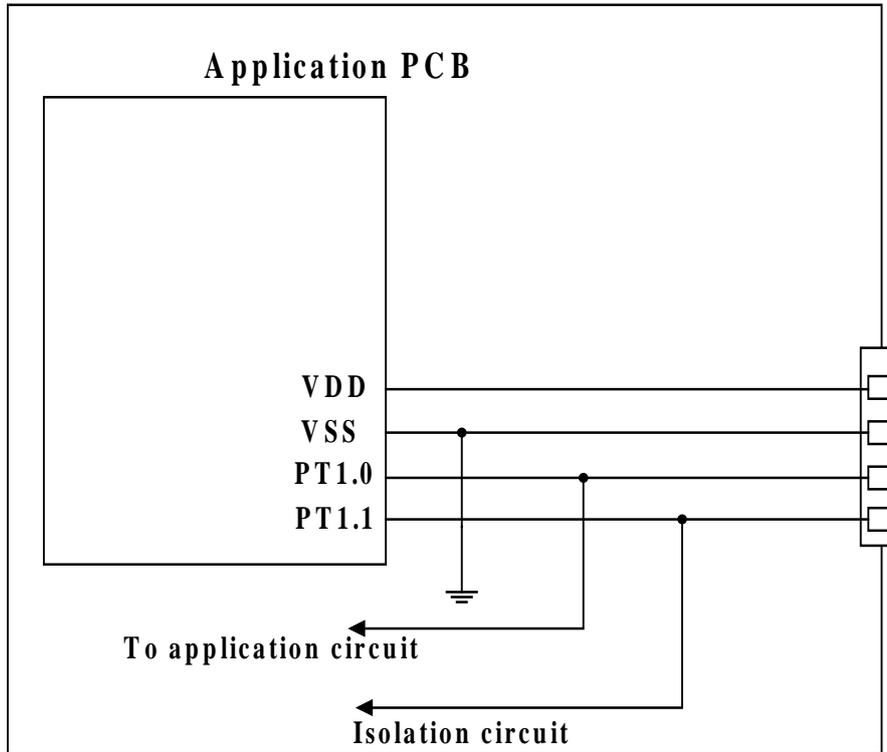


图12 烧写器接口图

表 20 烧录接口说明

端口名称	说明
VDD	电源正端
VSS	电源负端
DATA	PT1.0 端口
CLK	PT1.1 端口

3.5 代码选项

标识符	功能					
CLKDIV	指令周期选择 <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 <table border="1"> <tr> <td>功能</td> </tr> <tr> <td>VDD 低于 2.0V, LVD 复位系统</td> </tr> <tr> <td>VDD 低于 2.0V, LVD 复位系统; STATUS 的 LVD24 作为 2.4V 的低电压检测器</td> </tr> <tr> <td>VDD 低于 2.4V, LVD 复位系统; STATUS 的 LVD36 作为 3.6V 的低电压检测器</td> </tr> <tr> <td>VDD 低于 2.0V, LVD 复位系统; STATUS 的 LVD36 作为 3.6V 的低电压检测器</td> </tr> </table>	功能	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	VDD 低于 2.4V, LVD 复位系统; STATUS 的 LVD36 作为 3.6V 的低电压检测器	VDD 低于 2.0V, LVD 复位系统; STATUS 的 LVD36 作为 3.6V 的低电压检测器
功能						
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						
VDD 低于 2.4V, LVD 复位系统; STATUS 的 LVD36 作为 3.6V 的低电压检测器						
VDD 低于 2.0V, LVD 复位系统; STATUS 的 LVD36 作为 3.6V 的低电压检测器						
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 <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 作为外部时钟输入端口, PT1.4 作为普通 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 4M~16MHz</td> </tr> </table>	晶振引脚	PT1.4 和 PT1.5 作为普通 IO 口	PT1.5 作为外部时钟输入端口, PT1.4 作为普通 IO 口	PT1.4 和 PT1.5 接外部晶振为 32768Hz	PT1.4 和 PT1.5 接外部晶振 4M~16MHz
晶振引脚						
PT1.4 和 PT1.5 作为普通 IO 口						
PT1.5 作为外部时钟输入端口, PT1.4 作为普通 IO 口						
PT1.4 和 PT1.5 接外部晶振为 32768Hz						
PT1.4 和 PT1.5 接外部晶振 4M~16MHz						
SECURITY	代码保密位 禁止代码加密 使能代码加密					

4 MCU 指令集

表 21 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	$[\text{Destination}] \leftarrow [f] + [W]$	1	C,DC,Z
ADDWFC f,d	$[\text{Destination}] \leftarrow [f] + [W] + C$	1	C,DC,Z
ANDLW k	$[W] \leftarrow [W] \text{ AND } k$	1	Z
ANDWF f,d	$[\text{Destination}] \leftarrow [W] \text{ AND } [f]$	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	$[\text{Destination}] \leftarrow [f] - 1$	1	Z
DECFSZ f,d	$[\text{Destination}] \leftarrow [f] - 1$, jump if the result is zero	1/2	~
GOTO k	$PC \leftarrow k$	2	~
HALT	CPU Stop	1	~
INCF f,d	$[\text{Destination}] \leftarrow [f] + 1$	1	Z
INCFSZ f,d	$[\text{Destination}] \leftarrow [f] + 1$, jump if the result is zero	1/2	~
IORLW k	$[W] \leftarrow [W] \text{ OR } k$	1	Z
IORWF f,d	$[\text{Destination}] \leftarrow [W] \text{ OR } [f]$	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	$[\text{Destination} \langle n+1 \rangle] \leftarrow [f \langle n \rangle]$	1	C,Z
RRF f,d	$[\text{Destination} \langle n-1 \rangle] \leftarrow [f \langle n \rangle]$	1	C,Z
SLEEP	STOP OSC	1	PD
SUBLW k	$[W] \leftarrow k - [W]$	1	C,DC,Z
SUBWF f,d	$[\text{Destinnation}] \leftarrow [f] - [W]$	1	C,DC,Z
SUBWFC f,d	$[\text{Destinnation}] \leftarrow [f] - [W] - 1 + C$	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	$[\text{Destination}] \leftarrow [W] \text{ XOR } [f]$	1	Z

参数说明:

f: 数据存储器地址(00H ~FFH)

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 22 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	$(W) \leftarrow (W) + K$
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	$(PC) \leftarrow (PC) + 1 + (W)$ 当 $(W) \leq 7FH$ $(PC) \leftarrow (PC) + 1 + (W) - 100H$ 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<=FFH 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<=FFH 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<=FFH 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<=FFH 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<=FFH 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

BTFSC	如果 bit 测试为 0 则跳转
指令格式	BTFSC f, b 0<=f<=FFH 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 BTFSC 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

BTFSS	如果 bit 测试为 1, 则跳转
指令格式	BTFSS f, b 0<=f<=FFH 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 BTFSS 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<=3FFH
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<=FFH
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 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<=FFH 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<=FFH 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<=FFH 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 (FLAG)=(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<=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<=FFH 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<=FFH 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<=FFH 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<=FFH
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

MOVP	读 EPROM 数据
指令格式	MOVP
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 MOVP	在指令执行之前 EADRH=04H, EADRL=00H



	地址为 0400H 的 EPROM 数据位 1234H 在指令执行之后 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<=FFH
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

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 4 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)=>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 4 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)=>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

32

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

33

RETURN	从子程序返回
指令格式	RETURN
操作	(Top Stack)=>PC 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<=FFH d=0,1
操作	(目标地址[n+1])<←(f[n]) (目标地址[0])<←C C<←(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

35

RRF	带进位右移
指令格式	RRF f, d 0<=f<=FFH d=0,1
操作	(目标地址[n-1])<←(f[n]) (目标地址[7])<←C C<←(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
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 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)

38

SUBWF	f 的值减工作寄存器的值
指令格式	SUBWF f, d 0<=f<=FFH 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
--	---------------

39

SUBWFC	带借位的减法
指令格式	SUBWFC f, d 0<=f<=FFH 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

40

SWAPF	交换寄存器的值
指令格式	SWAPF f, d 0<=f<=FFH 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

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 TBLP 0	在指令执行之前 EDATH=BAH, W=ACH, EADRH=04H, EADRL=00H 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)←(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

43

XORWF	f 的值与工作寄存器的值异或
指令格式	XORWF f, d 0<=f<=FFH 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	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_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	2.3	5	5.5	V
		-40 °C ~+85 °C	2.4	5	5.5	V
FXT	外部晶振频率				16	MHz
SVDD	VDD 上电速度	*特性值不是测试结果	0.15			V/ms
Tcpu	指令周期	VDD: 2.3V~5.5V	2000			ns
		VDD: 2.7V~5.5V	500			
		VDD: 4.5V~5.5V	250			
IDD1	电源电流 1 (fcpu=fosc/16)	晶振=16MHz,3V		0.8		mA
		晶振=16MHz,5V		1.3		mA
IDD2	电源电流 2 (fcpu=fosc/4)	内部振荡器关闭 fosc = 32768Hz,3V		8		uA
		内部振荡器关闭 fosc = 32768Hz,5V		13		uA
IPO	睡眠模式下电源电流	睡眠指令(打开上电复位 电路), 3V		0.7		uA
		睡眠指令(打开上电复位 电路), 5V		0.8		uA
VIH	数字输入高电平	PT1, PT2	0.75VDD			V
	复位输入高电平		0.75VDD			
VIL	数字输入低电平	PT1, PT2			0.3VDD	V
	复位输入低电平				0.3VDD	
IPU	上拉电流	PT1,2 Vin = 0		30		uA
IOH	高电平输出电流	VOH=0.9VDD		12		mA
IOL	低电平输出电流	VOL=0.1VDD		12		mA
LVD	低电压检测电压	LVD_SEL = 2'b01	2.1	2.4	2.7	V
		LVD_SEL = 2'b1x	3.3	3.6	3.9	
FRC	内置 RC 振荡器	25°C, 5V	15.68	16.0	16.32	MHz
		-40°C~85°C 2.3V~5.5V		16.0		
FWDT	内置看门狗时钟	25°C, 5V	29	32	36	KHz
Tint0,1	中断触发脉宽	25°C, 5V	Tcpu			ns

5.3 电气特性曲线图

以下曲线图不是实际的测试结果，只是反映电气特性的随温度或电压变化的大致趋势，与具体芯片的实际值可能存在一定的偏差。

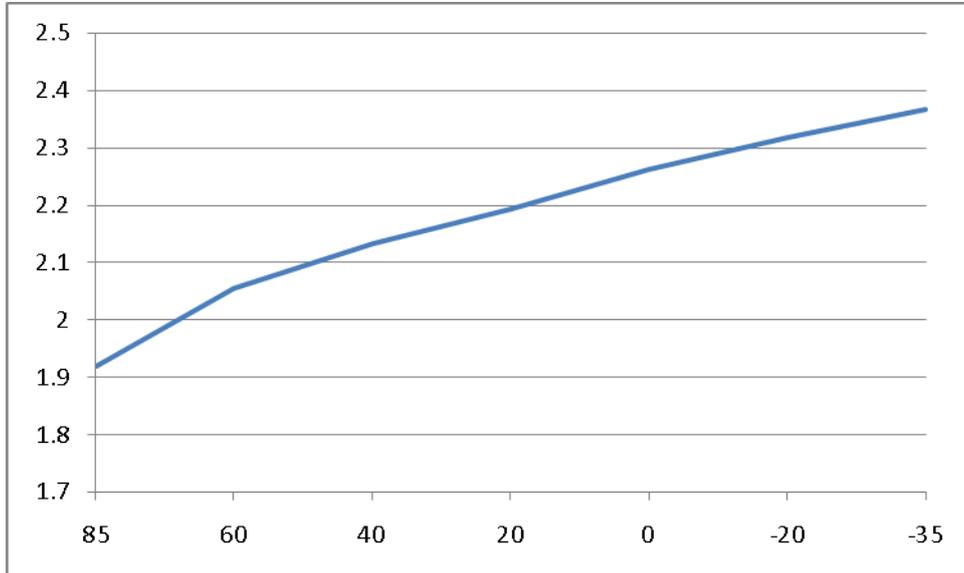


图 11 上电复位电压值随温度变化曲线

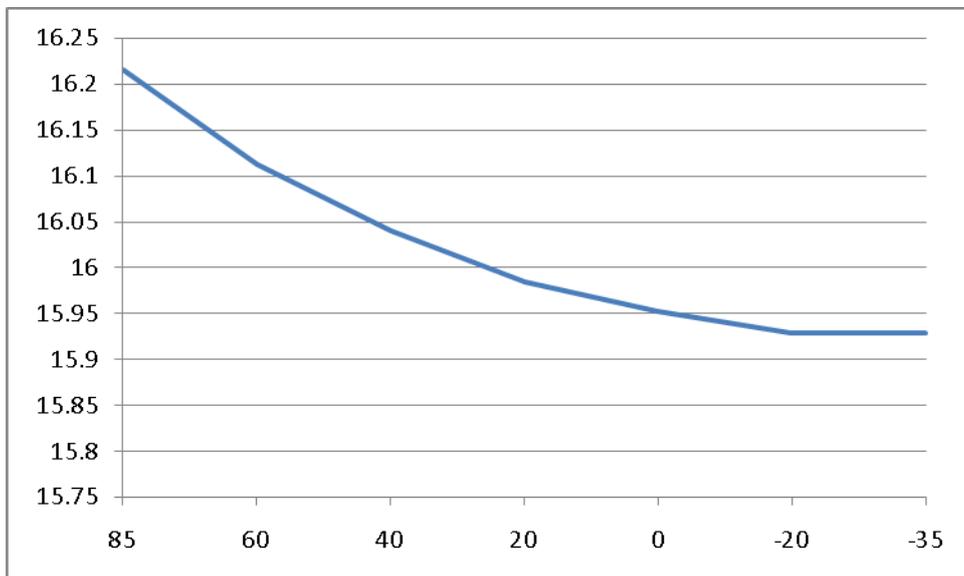
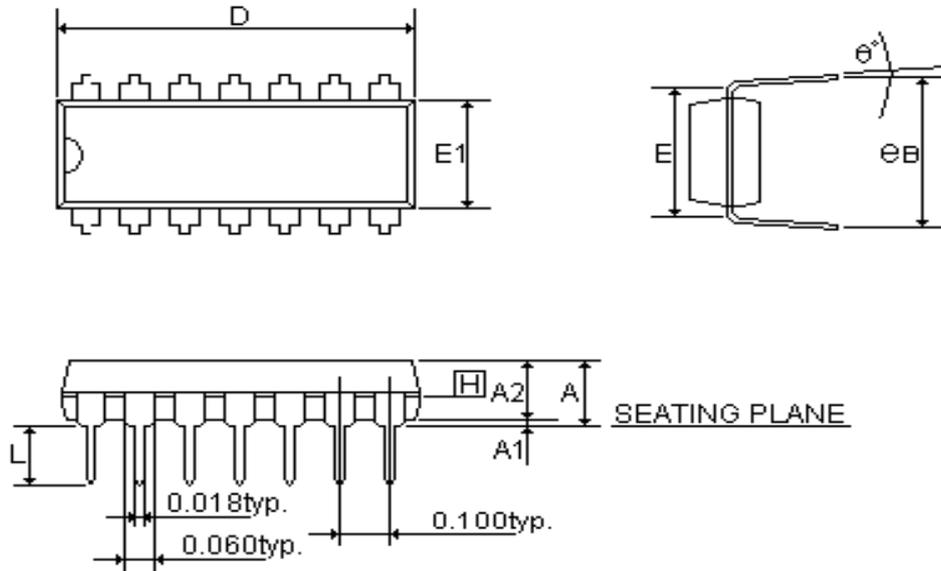


图 12 内部晶振频率随温度变化曲线

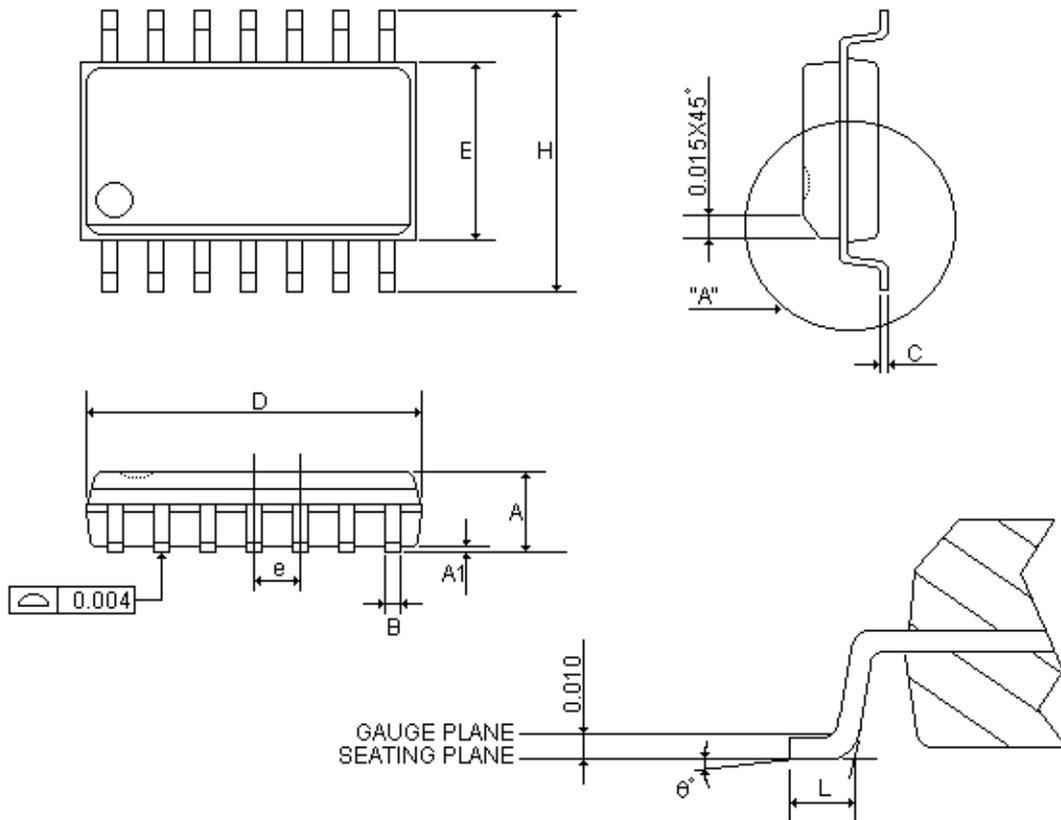
6 封装图

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
θ°	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
θ °	0°	-	8°	0°	-	8°