



CSU34F10 用户手册

带 12 位 ADC 的 8 位 RISC FLASH MCU

REV 1.0.2

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	2015-12-28
REV 1.0.1	1、修改笔误	2016-1-26
REV 1.0.2	1、BSR 寄存器低 2 位名称修改，避免歧义 2、EEPROM 操作章节笔误修改	2016-5-14

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	8
2 标准功能	10
2.1 CPU 核	10
2.1.1 存储器	12
2.1.2 状态寄存器	16
2.1.3 SFR	17
2.2 时钟系统	20
2.2.1 概述	20
2.2.2 时钟框图	20
2.2.3 寄存器	21
2.2.4 内部高速 RC 时钟	22
2.2.5 内部低速 WDT 时钟	22
2.2.6 外部高速晶振时钟	22
2.2.7 外部低速晶振时钟	22
2.2.8 外部 RC 振荡器	22
2.2.9 外部时钟源	22
2.3 复位系统	23
2.3.1 上电复位	24
2.3.2 看门狗复位	24
2.3.3 低电压复位	24
2.3.4 外部硬件复位	26
2.4 中断	27
2.4.1 中断使能寄存器	29
2.4.2 中断标志寄存器	31
2.4.3 外部中断 0	33
2.4.4 外部中断 1	33
2.4.5 AD 中断溢出	35
2.4.6 定时器 0 溢出中断	35
2.4.7 定时/计数器 2 溢出中断	35
2.4.8 定时/计数器 3 溢出中断	35
2.4.9 定时/计数器 4 溢出中断	35
2.4.10 SPI 中断	35
2.4.11 COG0 互补输出发生器中断	35
2.4.12 COG1 互补输出发生器中断	35
2.4.13 COG2 互补输出发生器中断	35
2.4.14 UART 发送中断	35
2.4.15 UART 接收中断	36
2.4.16 LVD 检测中断	36
2.4.17 PUSH 和 POP 处理	36
2.5 定时器 0	37

2.6	I/O PORT	39
2.6.1	PT1 口	39
2.6.2	PT3 口	43
2.6.3	PT5 口	45
2.6.4	驱动能力配置	47
2.6.5	输入逻辑电平电压配置	47
3	增强功能	48
3.1	防误操作	48
3.2	HALT 和 SLEEP 模式	49
3.3	看门狗(WDT)	53
3.4	定时/计数器 2	55
3.4.1	寄存器说明	55
3.4.2	12 位可编程定时器	58
3.4.3	蜂鸣器	59
3.4.4	PWM	59
3.5	定时/计数器 3	60
3.5.1	寄存器说明	60
3.5.2	12 位可编程定时器	63
3.5.3	蜂鸣器	64
3.5.4	PWM	64
3.6	定时/计数器 4	65
3.6.1	寄存器描述	65
3.6.2	16 位可编程定时器	67
3.6.3	蜂鸣器	68
3.6.4	PWM	69
3.7	串行通信接口(UART)	70
3.7.1	工作方式	70
3.7.2	寄存器说明	72
3.7.3	波特率	74
3.8	SPI 接口	75
3.8.1	SPI 接口介绍	75
3.8.2	SPI 寄存器说明	78
3.8.3	SPI 操作说明:	80
3.9	模数转换器(ADC)	82
3.9.1	寄存器说明	82
3.9.2	转换时间	86
3.9.3	数字比较器	88
3.9.4	模拟比较器触发 ADC 启动转换	90
3.10	比较器/运算放大器	91
3.10.1	比较器 0/运算放大器 0	91
3.10.2	寄存器说明	91
3.10.3	比较器参考电压	96
3.10.4	比较器失调电压校正	96
3.10.5	比较器中断	96
3.11	互补输出发生器(COG)	97
3.11.1	COG0	99
3.11.2	COG1	101
3.11.3	COG2	104
3.12	数据查表和 EEPROM 操作	107
3.12.1	数据查表	107

3.12.2	EEPROM 操作	108
3.13	烧录模块	110
3.14	代码选项(SECTOR3)	111
4	MCU 指令集.....	114
5	电气特性	130
5.1	极限值	130
5.2	直流特性 (VDD = 5V, T _A = 25°C, 如无其他说明则都是此条件)	130
5.3	ADC 特性 (VDD = 5V, T _A = 25°C, 如无其他说明则都是此条件)	132
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)	133
5.5	32MHz IHRC 时钟频率特性	134
5.6	16MHz IHRC 时钟频率特性	134
5.7	8MHz IHRC 时钟频率特性	135
5.8	4MHz IHRC 时钟频率特性	135
5.9	2MHz IHRC 时钟频率特性	136
5.10	32KHz WDT 时钟频率特性	136
5.11	1.8V 掉电复位温度特性	136
5.12	2.0V 内部参考电压温度特性	137
5.13	3.0V 内部参考电压温度特性	137
5.14	4.0V 内部参考电压温度特性	137
6	封装图	139
6.1	SOP-16PIN	139
6.2	SOP-20PIN	140
6.3	DIP-20PIN	141
6.4	SSOP-20PIN	141
7	单片机产品命名规则	143
7.1	产品型号说明	143
7.2	命名举例说明	144
7.3	产品印字说明	144

1 产品概述

1.1 功能描述

CSU34F10 是带 13 路 12 位 SAR ADC 的 8 位 RISC MCU，内置 4K×16 位 FLASH 程序存储器。

1.2 主要特性

高性能的 RISC CPU

- 8 位单片机 MCU
- 4K×16 位 FLASH 程序存储器
- 488 字节数据存储器 (SRAM)
- 带 256 bytes E2PROM 功能
- 43 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(32/16/8/4/2MHz)和 ILRC(32KHz)振荡器，IHRC 精度为±1%@5V，25°C，ILRC 精度为±10%@5V，25°C
- 外部 32768Hz 晶振 (RTC) 或 2MHz~16MHz 晶振或 ERC 8M@5V DC

外设特性

- 18 位双向 I/O 口
- 3 路蜂鸣器输出
- 3 路 PWM 输出，每路 PWM 都从各自两个 IO 选择输出
- 3 个互补输出发生器(COG)
- UART 通信
- 一个 SPI (带 Master 和 Slave)
- 12 个内部中断，2 个外部中断
- 8 个具有唤醒功能的输入口
- 13+4 路 12bitADC
 - 内部 1.4V、2.0V、3.0V、4.0V、VDD、外部输入六种参考电压选择
 - 带数字比较器
- 1 个内置模拟比较器和 1 个运算放大器
- 提供 1 个在 1.4V，2.0V，3.0V 和 4.0V 之间可选的参考电压，可配置通过 IO 输出，精度为±1%@5V，25°C (不用 LDO 输出，只用于 ADC 校准)

- 输入电压范围 2.0 V~5.5V
- 低电压检测 (LVD) 引脚，内部提供 2.0V、2.4V、3.0V、3.6V、4.2V 电压检测
- 8 个开漏输出 (PT1.1,PT1.4,PT1.5, PT5.0~PT5.4)
- 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
- COM0~COM3 双边可配置驱动能力
驱动能力可控，最大 40mA

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 (1ms, 10ms, 49ms, 98ms)
- 内带 1.8V、2.0V、2.4V、3.0V 和 3.6V 低电压复位 (LVR)
- 定时器 0
 - 8 位可编程预分频的 8 位的定时计数器
- 定时/计数器 2
 - 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
- 定时/计数器 4
 - 可编程预分频的 16 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
 - 2.0V~5.5V@fcpu=8MHz (指令周期)
- 工作温度范围
 - -40~85°C

低功耗特性

- MCU 工作电流
 - 正常模式 1.5mA@4MHz(工作电压 5V)
 - 超低功耗模式下的电流小于 1μA

封装

- SOP16
- SOP20/DIP20/SSOP20

型号	ROM	RAM	堆栈	定时器	12 位 PWM	16 位 PWM	互补 PWM	互补输出发生器	IO	UART	SPI	ADC (CH*Bit)	模拟比较器	大驱动	封装
----	-----	-----	----	-----	----------	----------	--------	---------	----	------	-----	--------------	-------	-----	----

CSU34F10	4K*16	488	8	4	2	1	1	3	18	1	1	13*12	1	√	SOP20/DIP20/SOP16/SSOP20
----------	-------	-----	---	---	---	---	---	---	----	---	---	-------	---	---	--------------------------

1.3 PIN 配置

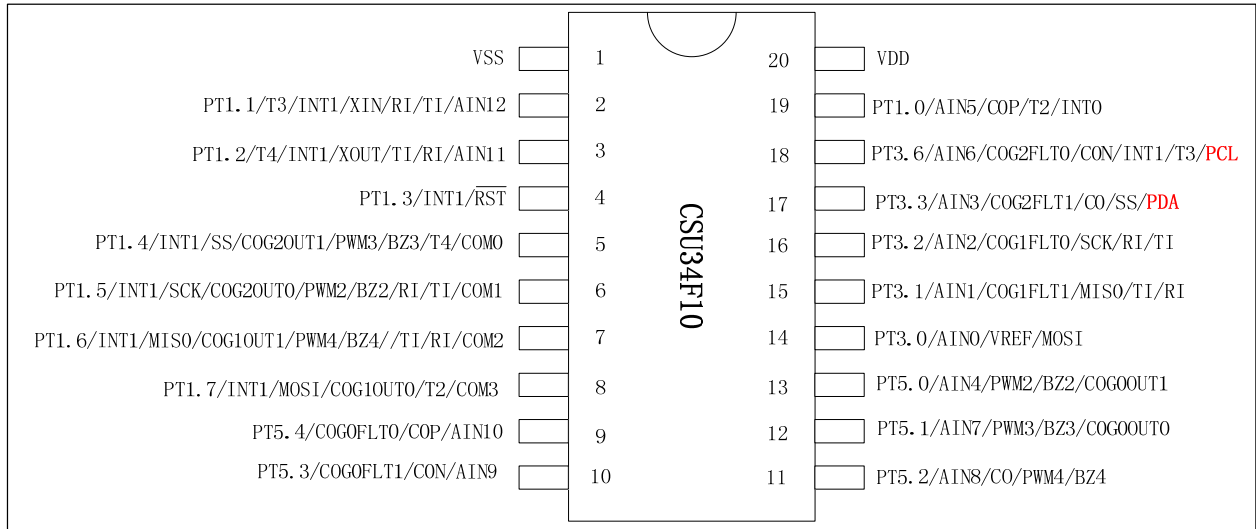


图1 通用 20pin 配置图

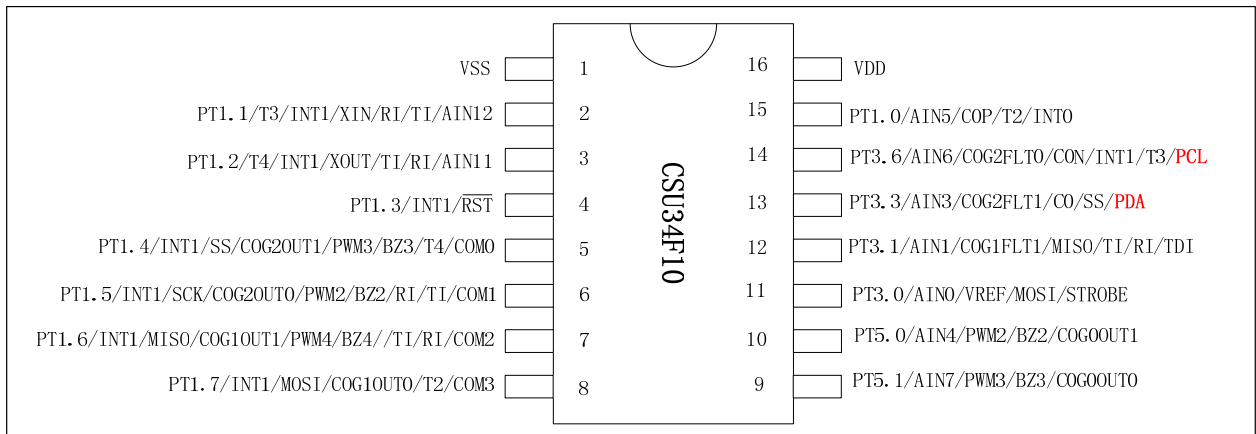


图2 通用 16pin 配置图

表 1 引脚说明表

管脚名称	输入/ 输出	20PIN 管脚序 号	16PIN 管脚序 号	描述
VSS	P	1	1	地
PT1.1/T3/INT1/X IN/RI/TI/ AIN12	I/O	2	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；UART 接收口；UART 发送口；ADC 输入 12
PT1.2/T4/INT1/X OUT/TI/RI /AIN11	I/O	3	3	IO；定时/计数器 4 外部输入；外部中断 1 输入，具有 唤醒功能；外置晶振输出；UART 发送口；UART 接 收口；具有开漏输出功能；ADC 输入 11
PT1.3/INT1/ RST	I/O	4	4	IO；外部中断 1 输入，具有唤醒功能；复位输入；
PT1.4/INT1/SS/C OG2OUT1/PWM 3/BZ3/T4/COM0	I/O	5	5	IO，具有开漏输出功能；外部中断 1 输入，具有唤醒功 能；SPI 从模式选择输入；COG2OUT1 输出；PWM3 输出；蜂鸣器 3 输出；定时/计数器 4 外部输入； COM0 输出(驱动能力可配置为 40/20mA)；

PT1.5/INT1/SCK/COG2OUT0/PWM2/BZ2/RI/TI/COM1	I/O	6	6	IO, 具有开漏输出功能; 外部中断 1 输入, 具有唤醒功能; SPI 模式的同步串口时钟的输入/输出口; COG2OUT1 输出; PWM2 输出; 蜂鸣器 2 输出; UART 接收口; UART 发送口; COG0FLT1 输入; COM1 输出(驱动能力可配置为 40/20mA);
PT1.6/INT1/MISO/COG1OUT1/PWM4/BZ4//TI/RI/COM2	I/O	7	7	IO; 外部中断 1 输入, 具有唤醒功能; SPI 模式的主模式数据输入口或从模式数据输出; COG1OUT1 输出; PWM4 输出; 蜂鸣器 4 输出; UART 发送口; UART 接收口; COM2 输出(驱动能力可配置为 40/20mA);
PT1.7/INT1/MOSI/COG1OUT0/T2/COM3	I/O	8	8	IO; 外部中断 1 输入, 具有唤醒功能; SPI 模式的主模式数据输出口或从模式数据输入; COG1OUT0 输出; 定时/计数器 2 外部输入; COM3 输出(驱动能力可配置为 40/20mA);
PT5.4/COG0FLT0/C0P/AIN10	I/O	9		IO, 具有开漏输出功能; COG0FLT0 输入; 比较器 0 的输入端/运放的正端输入; ADC 输入 10;
PT5.3/COG0FLT1/C0N/AIN9	I/O	10		IO, 具有开漏输出功能; COG0FLT1 输入; 比较器 0 的输入端/运放的负端输入; ADC 输入 9;
PT5.2/AIN8/CO/PWM4/BZ4	I/O	11		IO, 具有开漏输出功能; ADC 输入 8; 比较器 0 的输出端/运放的输出; PWM4 输出; 蜂鸣器 4 输出;
PT5.1/AIN7/PWM3/BZ3/C0G0OUT0	I/O	12	9	IO, 具有开漏输出功能; ADC 输入 7; PWM3 输出; 蜂鸣器 3 输出; C0G0OUT0 输出;
PT5.0/AIN4/PWM2/BZ2/COG0OUT1	I/O	13	10	IO, 具有开漏输出功能; ADC 输入 4; PWM2 输出; 蜂鸣器 2 输出; C0G0OUT1 输出;
PT3.0/AIN0/VREF/MOSI	I/O	14	11	IO; ADC 输入 0; ADC 参考电压输入/内部参考电压输出; SPI 模式的主模式数据输出口或从模式数据输入;
PT3.1/AIN1/COG1FLT1/MISO/TI/RI	I/O	15	12	IO; ADC 输入 1; COG1FLT1 输入; SPI 模式的主模式数据输入口或从模式数据输出; UART 发送口; UART 接收口;
PT3.2/AIN2/COG1FLT0/SCK/RI/TI	I/O	16		IO; ADC 输入 2; COG1FLT0 输入; SPI 模式的同步串口时钟的输入/输出口; UART 接收口; UART 发送口;
PT3.3/AIN3/COG2FLT1/C0/SS/PDA	I/O	17	13	IO; ADC 输入 3; COG2FLT1 输入; 比较器 0 的输出端/运放的输出; SPI 从模式选择输入; 烧录数据线;
PT3.6/AIN6/COG2FLT0/C0N/INT1/T3/PCL	I/O	18	14	IO; ADC 输入 6; COG2FLT0 输入; 比较器 0/运放 0 负端输入; INT1 输入, 具有唤醒功能; 定时/计数器 3 外部输入; 烧录时钟线;
PT1.0/AIN5/C0P/T2/INT0	I/O	19	15	I/O; ADC 输入 5; 比较器 0/运放 0 正端输入; 定时/计数器 2 外部输入; INT0 输入, 具有唤醒功能;
VDD	P	20	16	电源

2 标准功能

2.1 CPU核

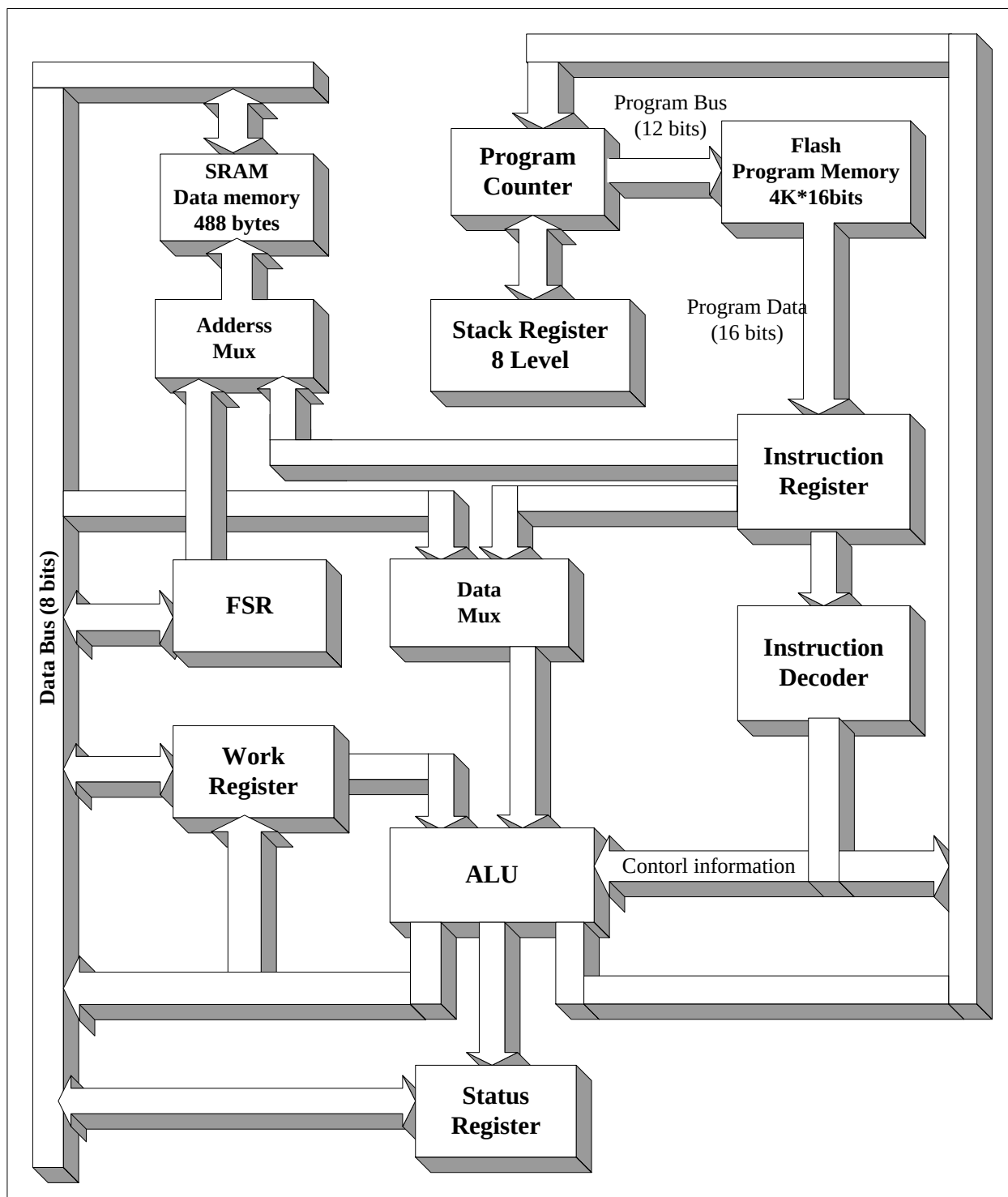


图3 CSU34F10 CPU 核的功能模块图

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（12bits），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	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。 CSU34F10 的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。 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。 立即数（8bits）：CPU 通过 ALU 利用此数据对工作寄存器进行操作。 控制信息：它记录着 ALU 的操作信息。
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	CSU34F10 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	CSU34F10 内带 4K×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 4K 的指令。程序存储器的地址总线是 12bits，数据总线是 16bits。
数据存储器	CSU34F10 内带 488bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 9bits，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 CSU34F10 中，该程序存储器是 4K*16 bit 的程序 FLASH；E2PROM 空间大小为 128*16 bit。对于程序员来说，该存储器可读可写。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。

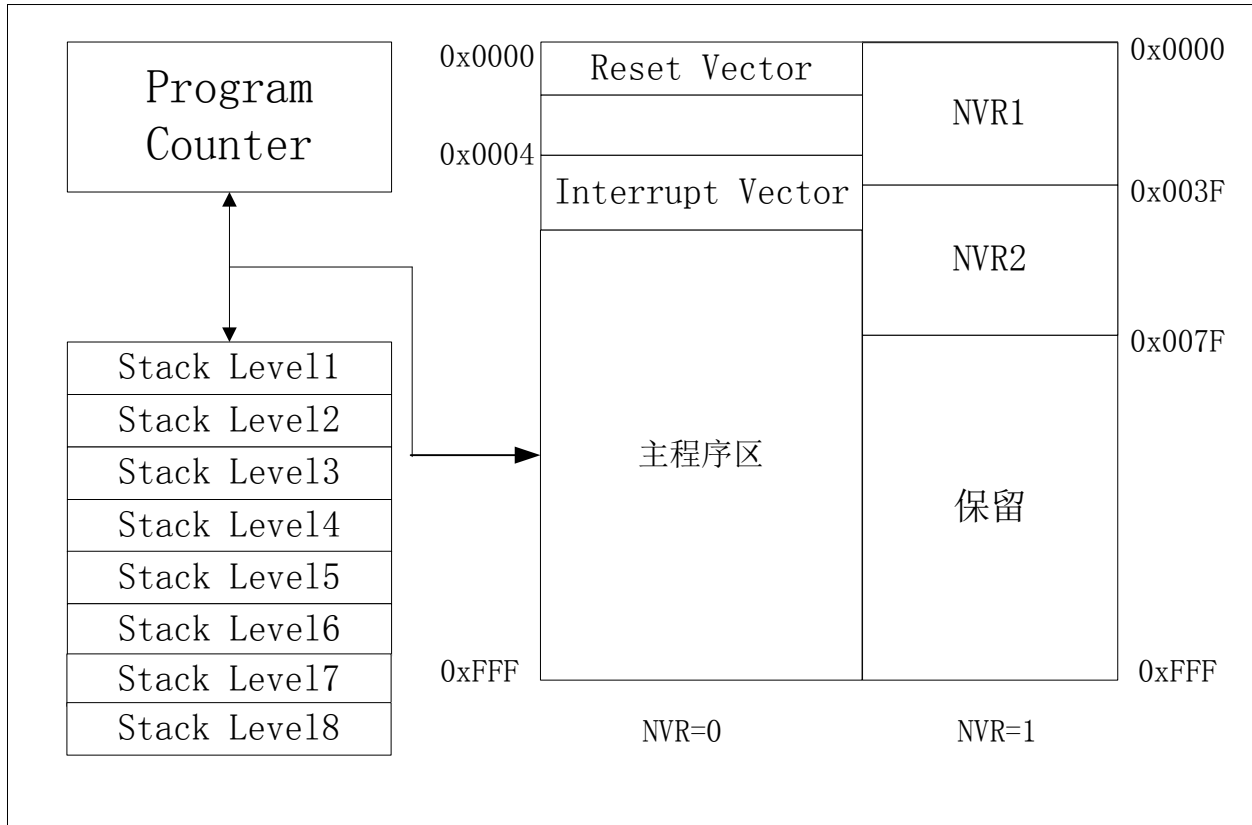


图4 程序存储器

(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 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。

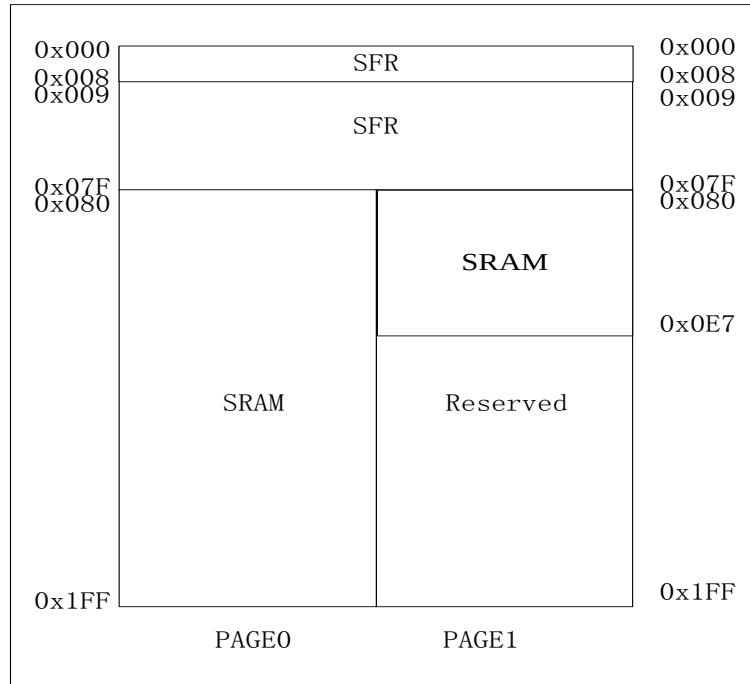


表 3 数据存储器地址分配

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 PAGE0	0x80	0x1FF
通用数据存储器 PAGE1	0x80	0xE7

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。其访问方式见下图。

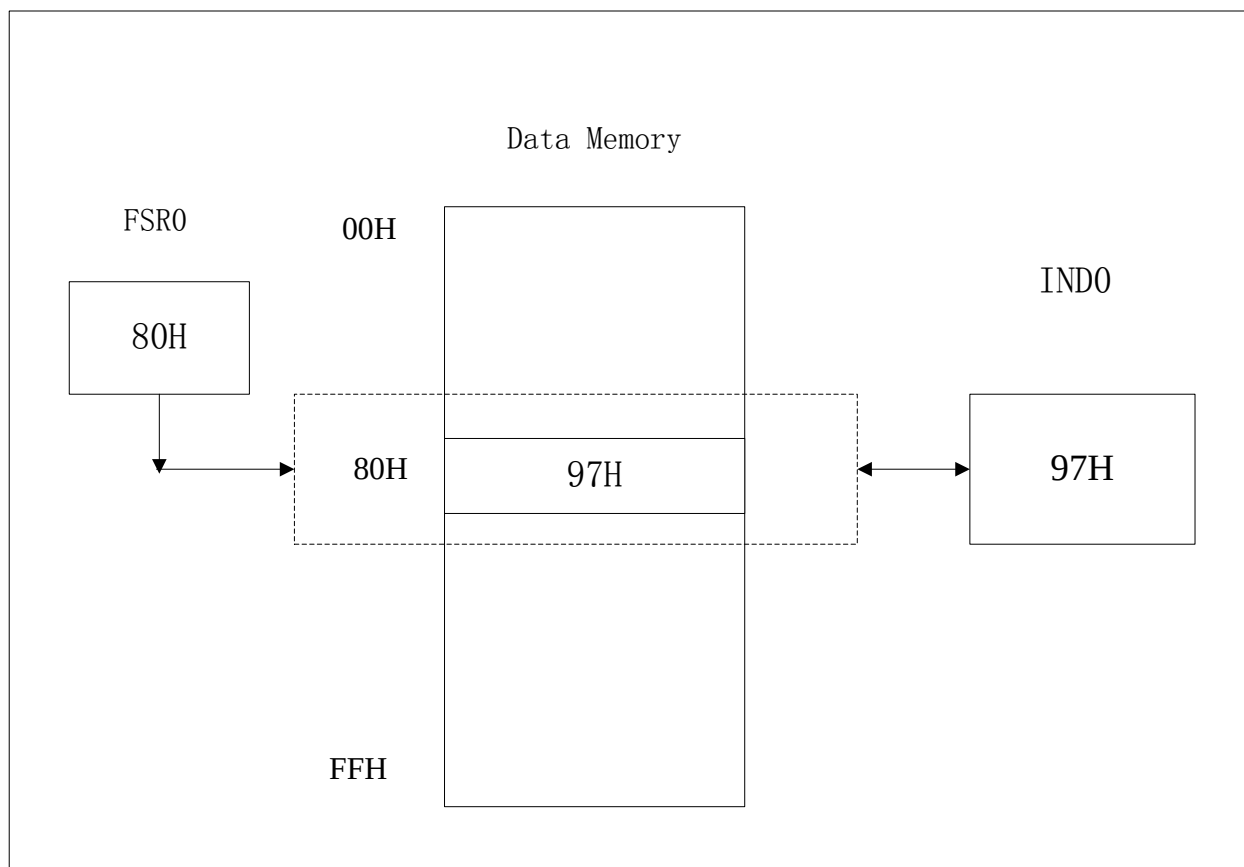


图5 间接地址访问

BSR 寄存器（地址为 08h）

特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
BSR	IRP0	IRP1					PGC1	PGC0
	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

Bit 6 IRP1: IND1 间接页寻址位

1 = 间接寻址 IND1 时，访问后 256byte 地址

0 = 间接寻址 IND1 时，访问前 256byte 地址

Bit 1 PGC1: PAGE 页选择

1 = 通过 IND1 访问时，访问 PAGE1，即后 512byte 地址

0 = 通过 IND1 访问时，访问 PAGE0，即前 512byte 地址

Bit 0 PGC0: PAGE 页选择

1 = 通过 IND0 访问时，访问 PAGE1，即后 512byte 地址

0 = 通过 IND0 访问时，访问 PAGE0，即前 512byte 地址

不同 PAGE 设置时 SFR 和 SRAM 寻址映射表

PAGE 位	SFR 读写	SRAM 直接寻址	SRAM 间接寻址
PGC1=0, PGC0=0	直接寻址和间接寻址 SFR 均无影响	直接寻址前 512 bytes (080H~1FFH)	IND0、IND1 均寻址前 512 bytes (080H~1FFH)
PGC1=1, PGC0=0	直接寻址和间接寻址	直接寻址前 512 bytes	IND0 均寻址前 512 bytes

	SFR 均无影响	(080H~1FFH)	(080H~1FFH) IND1 均寻址后 512 bytes (200H~267H)
PGC1=0, PGC0=1	直接寻址和间接寻址 SFR 均无影响	直接寻址后 512 bytes (200H~267H)	IND0 均寻址后 512 bytes (200H~267H) IND1 均寻址前 512 bytes (080H~1FFH)
PGC1=1, PGC0=1	直接寻址和间接寻址 SFR 均无影响	直接寻址后 512 bytes (200H~267H)	IND0 均寻址后 512 bytes (200H~267H) IND1 均寻址后 512 bytes (080H~1FFH)

注意：访问 200H~267H 的地址时，不能直接使用地址 200H~267H，因为我们直接寻址时最大地址为 9 位，最多只能访问到 1FFH，间接寻址地址寄存器也只有 8 位，因此不能直接使用地址 200H~267H。当我们要通过直接寻址访问 200H~267H 时，如下所示进行操作

- 设置 PGC0=1
- 直接寻址 80H 到 E7H 地址即可以访问 200H 到 267H 地址

当通过间接寻址访问 200H~267H 时，我们以通过 IND0 为例说明如何操作

- 设置 PGC0=1
- 向 FSR0 写入地址 80H~E7H，访问 IND0 即可实现对 200H~267H 操作

当对第 2 个 PAGE 操作完成后，请及时将 PGC0 或 PGC1 清零，否则可能造成 SRAM 访问混乱。

2.1.2 状态寄存器

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

状态寄存器（地址为 04h）

特性	R-0	R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0
STATUS	LVD36	LVD24	LVD42	PD	T0	DC	C	Z
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 LVD36: 3.6V LVD 工作电压标志位

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 5 LVD42 : 4.2V LVD 工作电压标志位

1: 表明芯片工作电压低于 4.2V

0: 表明芯片工作电压高于 4.2V

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 CLRWDI 指令之后

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

1: 看门狗定时溢出生

0: 上电复位后或硬件复位或 CLRWDI 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

SYSCFG1 寄存器（地址为 7dh）

特性	R-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
SYSCFG1	LVD20	LVD30						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 LVD20 : 2.0V LVD 工作电压标志位

1: 表明芯片工作电压低于 2.0V

0: 表明芯片工作电压高于 2.0V

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	IND0	以 FSR0 中内容作为地址的数据存储器中 PAGE0 的数据								xxxxxxxx
01h	IND1	以 FSR1 中内容作为地址的数据存储器中 PAGE1 的数据								xxxxxxxx
02h	FSR0	间接数据存储器的地址指针 0								00000000
03h	FSR1	间接数据存储器的地址指针 1								00000000
04h	STATUS	LVD36	LVD24	LVD42	PD	TO	DC	C	Z	xxx00000
05h	WORK	工作寄存器								00000000
06h	INTF		TM2IF		TMOIF	SRADIF		E1IF	E0IF	u0u0u0u0
07h	INTE	GIE	TM2IE		TMOIE	SRADIE		E1IE	E0IE	00u00u00
08h	BSR	IRP0	IRP1					PGC1	PGC0	00uuuu00
0Ah	EADRH					PAR[11:8]				uuuu0000
0Bh	EADRL	PAR[7:0]								00000000
0Ch	EDATH	EDATH[7:0]								00000000
0Dh	WDTCON	WDTEN			WDT_DLY		WTS[2:0]			0uu0u000
0Eh	WDTIN	WDTIN[7:0]								11111111
0Fh	TMOCON	TOEN	TORATE[2:0]				TORSTB	TOSEL[1:0]		0000u100
10h	TMOIN	TMOIN[7:0]								11111111
11h	TMOCNT	TMOCNT[7:0]								00000000
12h	PT1_TYPE	PT1_TYPE[7:0]								11111111
13h	PT3_TYPE		PT3_TYPE[6]			PT3_TYPE[3:0]				11uu1111
14h	PT5_TYPE				PT5_TYPE[4:0]					uuu11111
16h	MCK	CST	CST_IN	CST_WDT	EO_SLP	CPUCLK_SEL[1:0]		CLKSEL[1:0]		00100000
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[7:0]								00000000
22h	PT1PU	PT1PU[7:0]								00001000
23h	PT1CON0	PT110D	PT1W1[3:0]				E1M	E0M[1:0]		00000000
28h	PT3		PT3[6]			PT3[3:0]				xxxxxxxx
29h	PT3EN		PT3EN[6]			PT3EN[3:0]				u0uu0000
2ah	PT3PU		PT3PU[6]			PT3PU[3:0]				u0uu0000
2bh	PT3CON		PT3CON[6]			PT3CON[3:0]				u0uu0000
2ch	IODRV0					COMEN3	COMEN2	COMEN1	COMEN0	uuuu0000
2dh	PTVTH						PT5VTH	PT3VTH	PT1VTH	uuuuu000
30h	PT5				PT5[4:0]					uuuxxxxx
31h	PT5EN				PT5EN[4:0]					uuu00000
32h	PT5PU				PT5PU[4:0]					uuu00000
33h	PT5CON0				PT540D	PT530D	PT520D	PT510D	PT500D	uuu00000
34h	PT5CON1				PT5CON1[4:0]					uuu00000
35h	PT1PD	PT1PD[7:4]								0000uuuu
37h	PT5PD						PT5PD[2:0]			uuuuu000

38h	PT1CON1			PT150D	PT140D	PT1W2[3:0]				uu000000
39h	LVRCON					LVD_EN	LVR_FLT[2:0]			uuuu1000
3ch	INTF2		LVDIF	TM4IF	TM3IF					0u00uuuu
3dh	INTE2		LVD42IE	TM4IE	TM3IE	LVD36IE	LVD30IE	LVD24IE	LVD20IE	00000000
3eh	INTF3	CMP0IF	COG2IF	COG1IF	COG0IF		SPIIF	URTIF	URRIF	0000u000
3fh	INTE3		COG2IE	COG1IE	COG0IE		SPIIE	URTIE	URRIE	0000u000
40h	TM4CON	T4EN	T4RATE[2:0]			T4CKS	T4RSTB			00000100
41h	TM4INL	TM4IN[7:0]								11111111
42h	TM4INH	TM4IN[15:8]								11111111
43h	TM4CNTL	TM4CNT[7:0]								00000000
44h	TM4CNTH	TM4CNT[15:8]								00000000
45h	TM4RL	TM4R[7:0]								00000000
46h	TM4RH	TM4R[15:8]								00000000
47h	TM2INH					TM2IN[11:8]				uuuu1111
48h	TM2CNTH					TM2CNT[11:8]				uuuu0000
49h	TM2RH					TM2R[11:8]				uuuu0000
4ah	TM3INH					TM3IN[11:8]				uuuu1111
4bh	TM3CNTH					TM3CNT[11:8]				uuuu0000
4ch	TM3RH					TM3R[11:8]				uuuu0000
4dh	SSLPSAVE	SSLPSAVE[7:0]								00000000
4eh	LPOWER					FLT_EN	SUPSLEEP	DPSTB	LDOON	uuuu0000
4fh	SRADCON3		SRADSMOD[1:0]		SOURCE_SEL[1:0]		SRADSDLY[2: 0]			u00000000
50h	SRADCON0	SOURCE_MOD[1:0]		SRADACKS[1:0]			CMP0_R	SRADCKS[1:0]		00000000
51h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]		00000000
52h	SRADCON2	CHS[4:0]						REF_SEL[1:0]		00000u00
54h	SRADL	SRAD[7:0]								00000000
55h	SRADH					SRAD[11:8]				uuuu0000
56h	SROFTL	SROFT[7:0]								00000000
57h	SROFTH					SROFT[11:8]				uuuu0000
58h	COG0CON0	GOEN			G0POL1	G0POL0		G0CS[1:0]		00000u00
59h	COG0CON1	G0FSIM	G0RSIM	G0FS[2:0]			G0RS[2:0]			00000000
5ah	COG0ASD	GOASDE	GOARSEN	G0ASDL1	G0ASDL0	G0ASDSC3	G0ASDSC2	G0ASDSC1	G0ASDSC0	00000000
5bh	COG0DB	G0DBF[3:0]				G0DBR[3:0]				00000000
5ch	COG0BLK	G0BLKF[3:0]				G0BLKR[3:0]				00000000
5dh	COG0PH					G0PH[3:0]				uuuu0000
5eh	SEC	SECURITY[7:0]								00000000
5fh	ERASE_CON	ISPOF						NVR	ERASE_MO DE	0uuuuu00
60h	COG1CON0	G1EN			G1POL1	G1POL0		G1CS[1:0]		00000u00
61h	COG1CON1	G1FSIM	G1RSIM	G1FS[2:0]			G1RS[2:0]			00000000
62h	COG1ASD	G1ASDE	G1ARSEN	G1ASDL1	G1ASDL0	G1ASDSC3	G1ASDSC2	G1ASDSC1	G1ASDSC0	00000000
63h	COG1DB	G1DBF[3:0]				G1DBR[3:0]				00000000
64h	COG1BLK	G1BLKF[3:0]				G1BLKR[3:0]				00000000
65h	COG1PH					G1PH[3:0]				uuuu0000
66h	COG2CON0	G2EN			G2POL1	G2POL0		G2CS[1:0]		00000u00
67h	COG2CON1	G2FSIM	G2RSIM	G2FS[2:0]			G2RS[2:0]			00000000
68h	COG2ASD	G2ASDE	G2ARSEN	G2ASDL1	G2ASDL0	G2ASDSC3	G2ASDSC2	G2ASDSC1	G2ASDSC0	00000000
69h	COG2DB	G2DBF[3:0]				G2DBR[3:0]				00000000
6ah	CMPCON0	CMP0EN	COS0[3:0]					CMPIOS	CMPOUT	00000u00
6bh	CMPCON1		CMPISC[1:0]		CMPOFST_SEL[4:0]					u0010000
6ch	COG2BLK	G2BLKF[3:0]				G2BLKR[3:0]				00000000
6dh	COG2PH					G2PH[3:0]				uuuu0000
70h	SPICFG	SPIEN	MSTEN	CKPHA	CKPOL	MULMST	WIREMOD	SPIBSY	SPIRST	0000000u
71h	SPICN	SLVSEL	WCOL	MODCOL	SPIIOS					0000uuuu
72h	SPICKR	SCR[7:0]								00000011
73h	SPIDAT	SPIDAT[7:0]								00000000

74h	REFCFG		REFCFG[1:0]							00000000
77h	SCON1	SM0	SM1	SM2	REN	TB8	RB8	URTIOS	URRIOS	00000000
78h	SCON2	SMOD	TCLKS[1:0]		RCLKS[1:0]		URIOS[1:0]		UARTEN	00000000
79h	SBUF	SBUF[7:0]								00000000
7ah	SYSCFG	PT16PWM[1:0]		PT1CON[2:0]			T4CKS1	PT52PWM[1:0]		00000000
7dh	SYSCFG1	LVD20	LVD30	T3CKS1	PT14PWM[1:0]		T2CKS1	PT15PWM[1:0]		xx000000
7eh	TMCON		T4SEL[1]	T3SEL[1]	T2SEL[1]	T4SEL[0]	T3SEL[0]	T2SEL[0]		0000000u

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 32/16/8/4/2MHz 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。以上时钟都可以做为系统时钟源 Fosc。Fcpu 是 CPU 时钟频率。

普通模式（高速时钟）： $F_{cpu}=F_{osc}/N$ ， $N=2、4、8、16$

低速模式（低速时钟）： $F_{cpu}=F_{osc}/N$ ， $N=2、4、8、16$

2.2.2 时钟框图

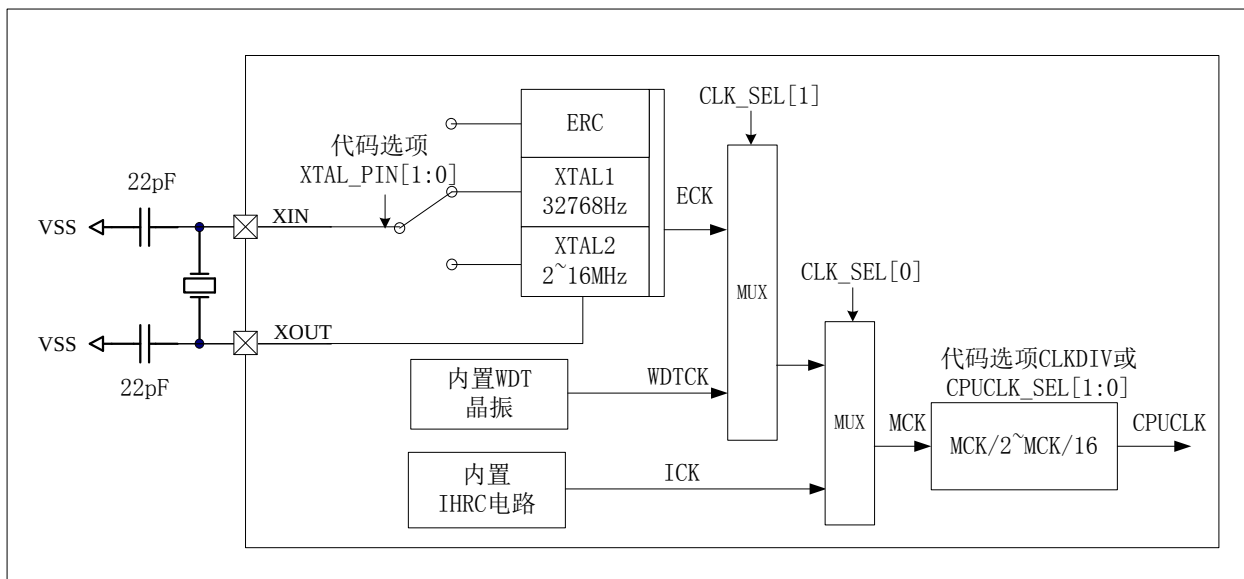


图6 CSU34F10 振荡器状态框图 A

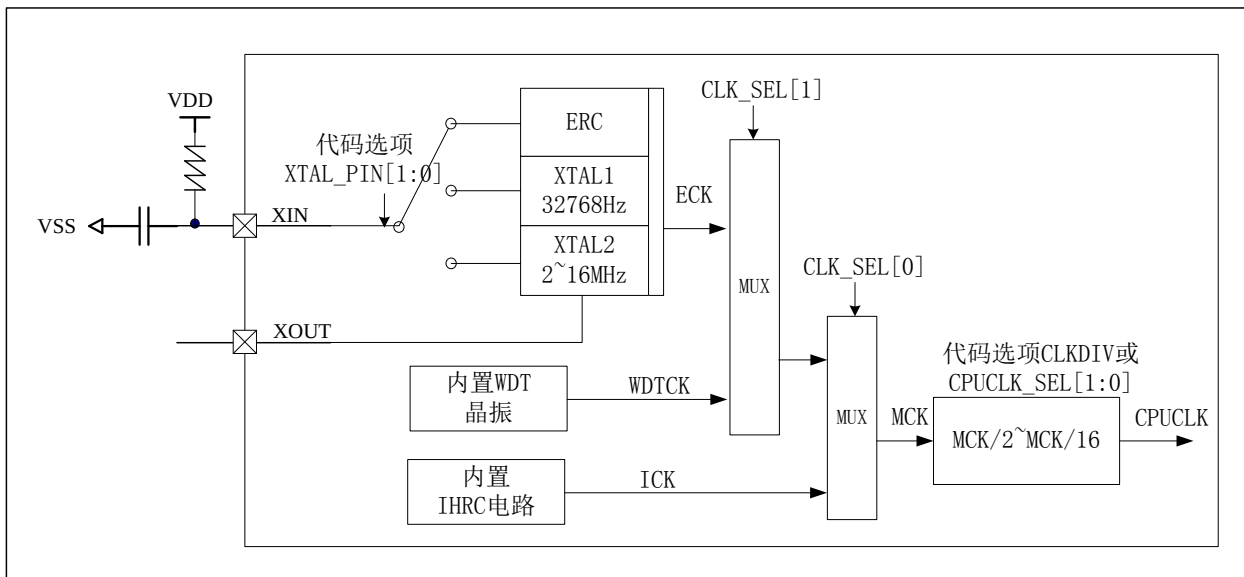


图7 CSU34F10 振荡器状态框图 B

2.2.3 寄存器

MCK 为写保护寄存器，要对 MCK 进行写操作，需先向 SEC 寄存器连续写入：C9h/3Ah/D3h，解锁写操作在 12 个指令周期内有效。12 个指令周期后写 MCK 寄存器操作无效。

表 5 CSU34F10 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><th>CPUCLK_SEL[1:0]</th><th>指令周期</th></tr><tr><td>00</td><td>指令周期=2 个时钟周期</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> 由于不支持 16Mhz 指令周期，故使用 32Mhz 内部晶振时，不能选择指令周期=2 个时钟周期。	CPUCLK_SEL[1:0]	指令周期	00	指令周期=2 个时钟周期	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	指令周期=2 个时钟周期											
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><th>CLK_SEL[1:0]</th><th>CPU 时钟</th></tr><tr><td>00</td><td>内部高速晶振时钟作为 CPU 时钟</td></tr><tr><td>01</td><td>外部晶振系统时钟作为 CPU 时钟</td></tr><tr><td>10</td><td>内部高速晶振时钟作为 CPU 时钟</td></tr><tr><td>11</td><td>内部 WDT 晶振时钟作为 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[1:0]	CPU 时钟	00	内部高速晶振时钟作为 CPU 时钟	01	外部晶振系统时钟作为 CPU 时钟	10	内部高速晶振时钟作为 CPU 时钟	11	内部 WDT 晶振时钟作为 CPU 时钟
CLK_SEL[1:0]	CPU 时钟											
00	内部高速晶振时钟作为 CPU 时钟											
01	外部晶振系统时钟作为 CPU 时钟											
10	内部高速晶振时钟作为 CPU 时钟											
11	内部 WDT 晶振时钟作为 CPU 时钟											

设置内部高速晶振时钟、外部晶振系统时钟或内部 WDT 晶振时钟作为 CPU 时钟时 SFR 配置

CST	CST_IN	CST_WDT	CLKSEL[1:0]	CPU 时钟
X	0	X	00 或 10	内部高速晶振时钟作为 CPU 时钟

0 且 XTAL_PIN[1:0]配置为外部时钟	X	X	01	外部晶振系统时钟作为 CPU 时钟
X	X	0	11	内部 WDT 晶振时钟作为 CPU 时钟

X 表示无关项，即该项为 0 或 1 都不影响。

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下顺序执行

2.2.4 内部高速RC时钟

内部高速 RC 时钟（32/16/8/4/2MHz），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

CSU34F10 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）

上电复位、硬件复位和 WDT 复位后会延时一段时间，然后芯片才进入正常工作状态，延时时间可通过代码选项配置（500us，2ms，49ms，98ms）。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。

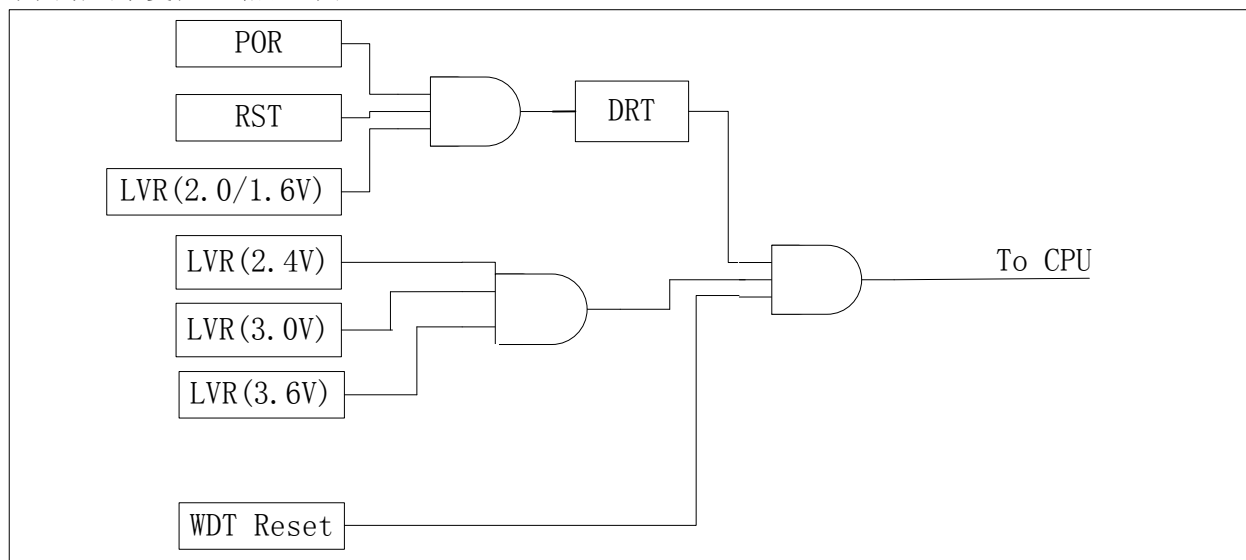


图8 复位电路原理图

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。

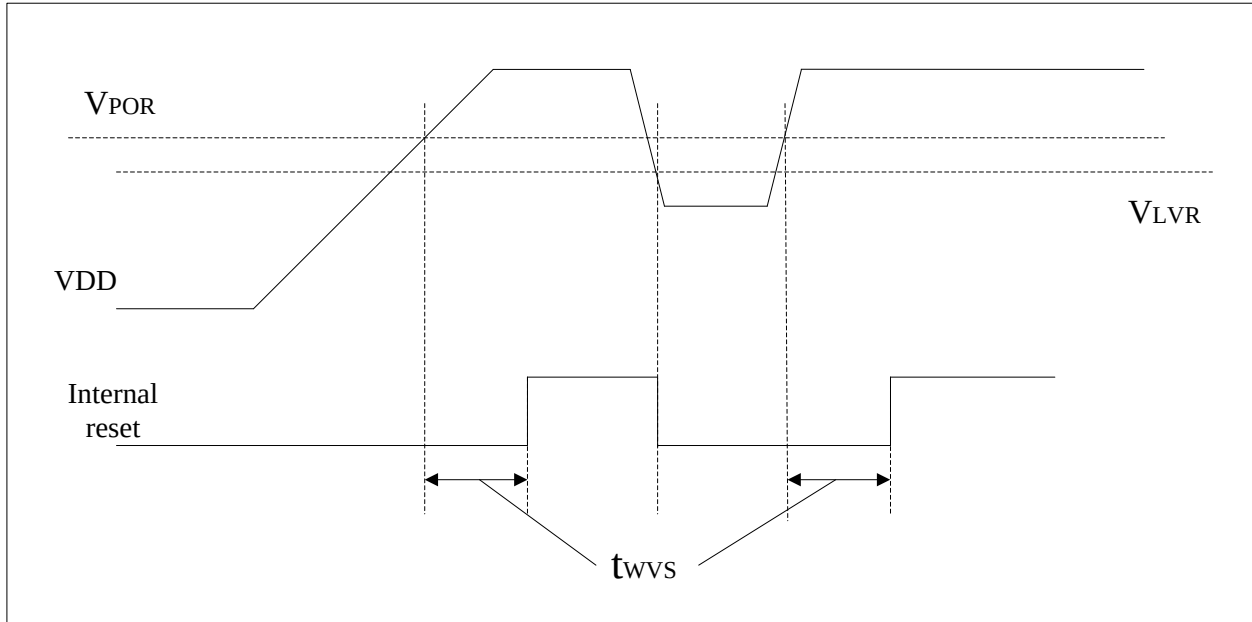


图9 上电复位电路示例及上电过程

参数	最小值	典型值	最大值
VPOR	1.6V	1.8V	2.0V
VLVR	1.6V	1.8V	2.0V
t _{wvs} (测试条件: VDD=5V, T=25℃)	450us	500us	550us

VPOR: 上电复位

VLVR: 低电压复位

t_{wvs}: 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，要实现正常上电，用户系统的上电速度要大于 0.07V/mS。

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 低电压复位

表 8 低电压复位寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3ch	INTF2		LVDIF							u000uuuu
3dh	INTE2		LVD42IE			LVD36IE	LVD30IE	LVD24IE	LVD20IE	u0000000
39h	LVRCON					LVD_EN	LVR_FLT[2: 0]			uuuu1000

表 9 INTF2 寄存器各位功能表

位地址	标识符	功能
6	LVDIF	低电压检测中断使能位 1 = 发生低电压中断，必须软件清 0 0 = 没发生低电压检测中断

表 10 INTE2 寄存器各位功能表

位地址	标识符	功能
6	LVD42IE	LVD42 中断使能位 1：使能 LVD 4.2V 中断 0：不使能 LVD4.2V 中断
3	LVD36IE	LVD36 中断使能位 1：使能 LVD 3.6V 中断 0：不使能 LVD3.6V 中断
2	LVD30IE	LVD30 中断使能位 1：使能 LVD 3.0V 中断 0：不使能 LVD3.0V 中断
1	LVD24IE	LVD24 中断使能位 1：使能 LVD 2.4V 中断 0：不使能 LVD2.4V 中断
0	LVD20IE	LVD20 中断使能位 1：使能 LVD 2.0V 中断 0：不使能 LVD2.0V 中断

表 11 LVRCON 寄存器各位功能表

位地址	标识符	功能
3	LVD_EN	低电压复位使能： 1'b1：使能低电压复位 1'b0：不使能低电压复位
2: 0	LVD_FTL[2:0]	低电压滤波选择：（本例以 16MHz 内部晶振为例，其他时钟滤波时间依次类推） 3'b000：低电压滤波时间 20us 3'b001：低电压滤波时间 40us 3'b010：低电压滤波时间 60us 3'b011：低电压滤波时间 80us 其他：不进行低电压滤波 1.8V 掉电复位无滤波，以上配置对 1.8V 上掉电无效

低电压复位通过设置代码选项中的 LVD_SEL[2: 0]来实现低电压复位方案。当外部电压低于低电压复位点，则系统进入低电压复位状态。

通过 LVD_FLT[2: 0]可对低电压滤波进行配置。选择不同的滤波方案。

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。

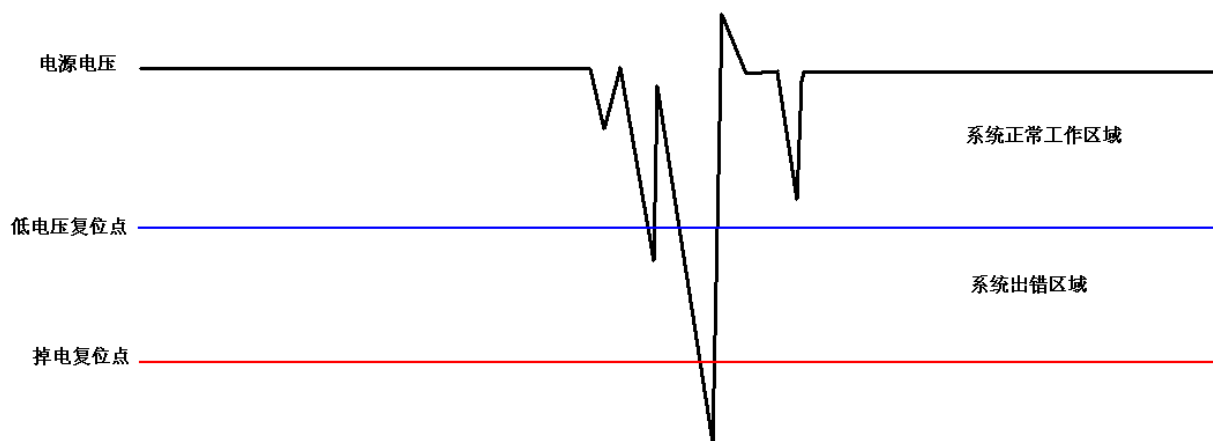


图10 系统掉电复位示意图

电压跌落可能会进入系统出错区域。进入系统出错区域，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。例如：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。

为避免进入系统出错区域，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，可使能外部硬件复位功能。外部硬件复位引脚为施密特触发结构，低电平有效。硬件复位引脚为高电平时，系统正常工作；硬件复位引脚为低电平时，系统复位。

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

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

2.4 中断

CSU34F10 有 14 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 14 个中断源(LVD 检测所有档位共用一个中断源入口)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。

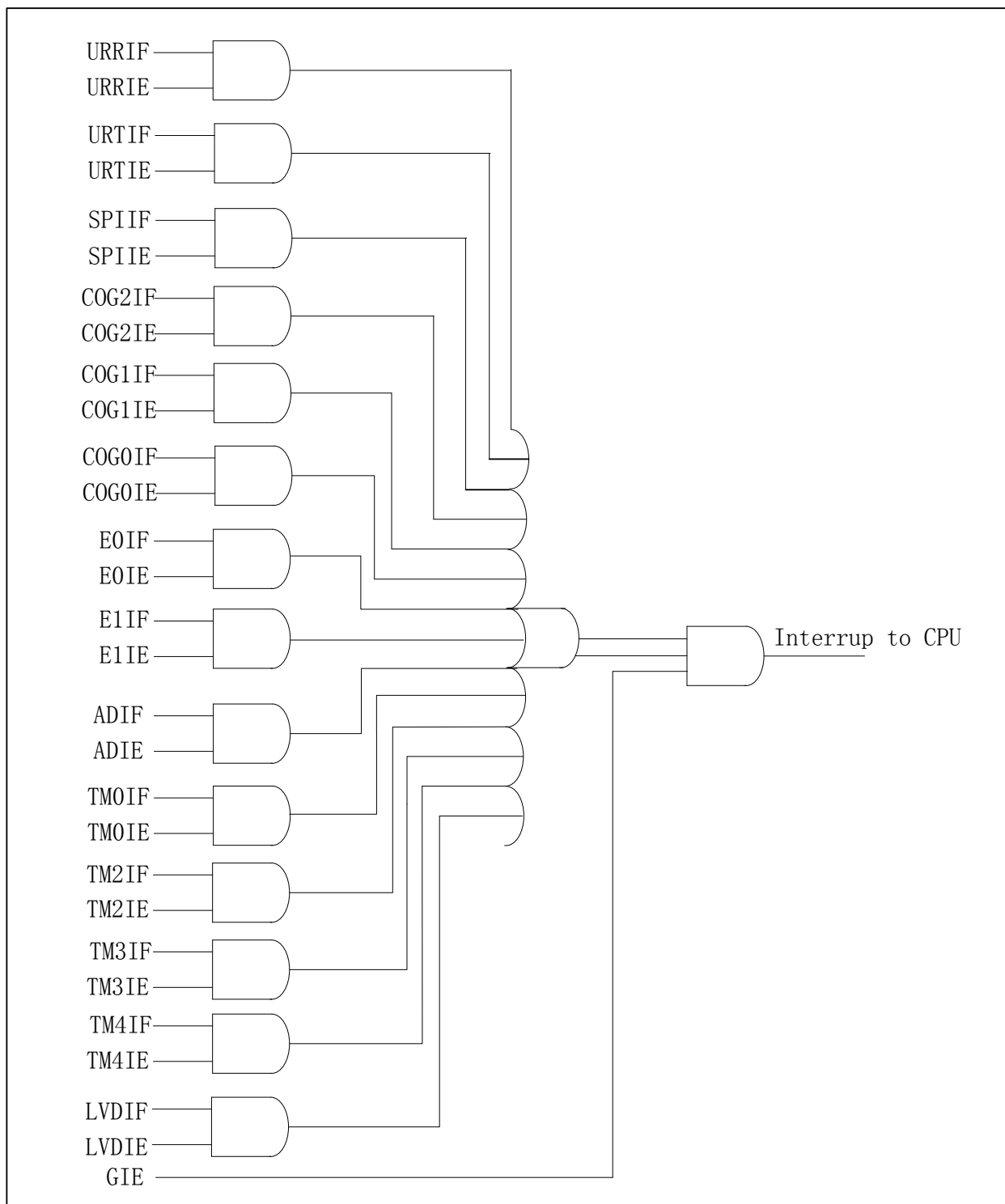


图11 中断逻辑

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	R/W-0	R/W-0	R/W-0	R/W-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	R/W-0	R/W-0	U-0	U-0	U-0	U-0
INTE2		LVD42IE	TM4IE	TM3IE	LVD36IE	LVD30IE	LVD24IE	LVD20IE
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 LVD42IE: 低电压检测 LVD42 使能标志

1 = 使能低电压检测 LVD42 中断

0 = 不使能低电压检测 LVD42 中断

Bit 5 TM4IE: 12-Bit 定时/计数器 4 中断使能标志

1 = 使能定时/计数器 4 中断

0 = 不使能定时/计数器 4 中断

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

Bit 3 LVD36IE: 低电压检测 LVD36 使能标志

1 = 使能低电压检测 LVD36 中断

0 = 不使能低电压检测 LVD36 中断

Bit 2 LVD30IE: 低电压检测 LVD30 使能标志

1 = 使能低电压检测 LVD30 中断

0 = 不使能低电压检测 LVD30 中断

Bit 1 LVD24IE: 低电压检测 LVD24 使能标志

1 = 使能低电压检测 LVD24 中断

0 = 不使能低电压检测 LVD24 中断

Bit 0 LVD20IE: 低电压检测 LVD20 使能标志

1 = 使能低电压检测 LVD20 中断

0 = 不使能低电压检测 LVD20 中断

INTE3 寄存器（地址为 3fh）

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
INTE3		COG2IE	COG1IE	COG0IE		SPIIE	URTIE	URRIE
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 COG2IE: 互补输出发生器 2 中断使能标志

1 = 使能互补输出发生器 2 中断

0 = 不使能互补输出发生器 2 中断

Bit 5 COG1IE: 互补输出发生器 1 中断使能标志

1 = 使能互补输出发生器 1 中断

0 = 不使能互补输出发生器 1 中断

Bit 4 COG0IE: 互补输出发生器 0 中断使能标志

1 = 使能互补输出发生器 0 中断

0 = 不使能互补输出发生器 0 中断

Bit 2 SPIIE: SPI 中断使能标志

1 = 使能 SPI 中断

0 = 不使能 SPI 中断

Bit 1 URTIE: UART 发送中断使能标志

1 = 使能 UART 发送中断

0 = 不使能 UART 发送中断

Bit 0 URRIE: UART 接收中断使能标志

1 = 使能 UART 接收中断

0 = 不使能 UART 接收中断

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	R/W -0	R/W -0	U-0	U-0	U-0	U-0
INTF2		LVDIF	TM4IF	TM3IF				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 LVDIF: 低电压检测中断标志，软件清零，硬件置高

1 = 发生低电压中断，必须软件清 0

0 = 没发生低电压检测中断

Bit 5 TM4IF: 12-Bit 定时/计数器 4 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 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	R/W -0	R/W -0	R/W -0	U-0	U-0	R/W -0	R/W -0
INTF3	CMP0IF	COG2IF	COG1IF	COG0IF		SPIIF	URTIF	URRIF
	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，无法进入中断函数，只能查询标志位

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

Bit 6 COG2IF: 互补输出发生器 2 中断标志，软件清零，硬件置高

1 = 发生中断，必须软件清 0

0 = 没发生中断

Bit 5 COG1IF: 互补输出发生器 1 中断标志，软件清零，硬件置高

1 = 发生中断，必须软件清 0

- 0 = 没发生中断
- Bit 4 COG0IF: 互补输出发生器 0 中断标志, 软件清零, 硬件置高
1 = 发生中断, 必须软件清 0
0 = 没发生中断
- Bit 2 SPIIF: SPI 中断标志, 软件清零, 硬件置高
1 = 发生 SPI 中断, 必须软件清 0
0 = 没发生 SPI 中断
- Bit 1 URTIF: UART 发送中断标志, 软件清零, 硬件置高
1 = 发生 UART 发送中断, 必须软件清 0
0 = 没发生 UART 发送中断
- Bit 0 URRIF: UART 接收中断标志, 软件清零, 硬件置高
1 = 发生 UART 接收中断, 必须软件清 0
0 = 没发生 UART 接收中断

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 PT3.0 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、PT1.7 和 PT3.6 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
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

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[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 3 PT1W2[3]:PT3.6 外部中断 1 使能
0 = 禁止 PT3.6 外部中断 1
1 = 使能 PT3.6 外部中断 1
- 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 TM0IE 为定时器 0 中断的使能位，INTF 寄存器中的 TM0IF 为中断标志位，软件清 0。当定时器 0 溢出时，不管 TM0IE 是否使能，TM0IF 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 定时/计数器 4 溢出中断

INTE2 寄存器中的 TM4IE 为定时/计数器 4 中断的使能位，INTF2 寄存器中的 TM4IF 为中断标志位，软件清 0。当定时/计数器 4 溢出时，不管 TM4IE 是否使能，TM4IF 都会硬件置 1。

2.4.10 SPI中断

INTE3 寄存器中的 SPIIE 为 SPI 中断的使能位，INTF3 寄存器中的 SPIIF 为中断标志位，软件清 0。当 SPI 完成一个字节数据发送或接收时，不管 SPIIE 是否使能，SPIIF 都会硬件置 1。

2.4.11 COG0 互补输出发生器中断

INTE3 寄存器中的 COG0IE 为互补输出发生器的使能位，INTF3 寄存器中的 COG0IF 为中断标志位，软件清 0。当互补输出发生器自动关闭发生时，不管 COG0IE 是否使能，COG0IF 都会硬件置 1。

2.4.12 COG1 互补输出发生器中断

INTE3 寄存器中的 COG1IE 为互补输出发生器的使能位，INTF3 寄存器中的 COG1IF 为中断标志位，软件清 0。当互补输出发生器自动关闭发生时，不管 COG1IE 是否使能，COG1IF 都会硬件置 1。

2.4.13 COG2 互补输出发生器中断

INTE3 寄存器中的 COG2IE 为互补输出发生器的使能位，INTF3 寄存器中的 COG2IF 为中断标志位，软件清 0。当互补输出发生器自动关闭发生时，不管 COG2IE 是否使能，COG2IF 都会硬件置 1。

2.4.14 UART发送中断

INTE3 寄存器中的 URTIE 为 UART 发送中断的使能位，INTF3 寄存器中的 URTIF 为中断标志位，软件清 0。当 UART 完成一个字节数据发送时，不管 URTIE 是否使能，URTIF 都会硬件置 1。

2.4.15 UART接收中断

INTE3 寄存器中的 URRIE 为 UART 接收中断的使能位，INTF3 寄存器中的 URRIF 为中断标志位，软件清 0。当 UART 完成一个字节数据接收时，不管 URRIE 是否使能，URRIF 都会硬件置 1。

2.4.16 LVD检测中断

INTE2 寄存器中的 LVD42IE/LVD36IE/LVD30IE/LVD24IE/LVD20IE 为 LVD 检测各档位的使能位，STATUS 寄存器中的 LVD36/LVD24/LVD42 及 SYSCFG1 寄存器中的 LVD20/LVD30 为中断标志位，软件清 0。当一个中断标志位置 1，且对应使能位使能时，产生一个 LVD 中断，中断标志位硬件置 1，软件清 0。

2.4.17 PUSH和POP处理

CSU34F10 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,elif    ;判断外部中断 1 标志
    goto ex1_int
    btfsc intf,tm0if    ;判断定时器 0 中断标志
    goto tm0_int
    btfsc intf,tm2if    ;判断定时/计数器 2 中断标志
    goto tm2_int
    btfsc intf,tm3if    ;判断定时/计数器 3 中断标志
    goto tm3_int
    ...
ex0_int:
    bcf intf, elif    ;清除 elif
    ...
    pop
    retfie
    ...
```

2.5 定时器 0

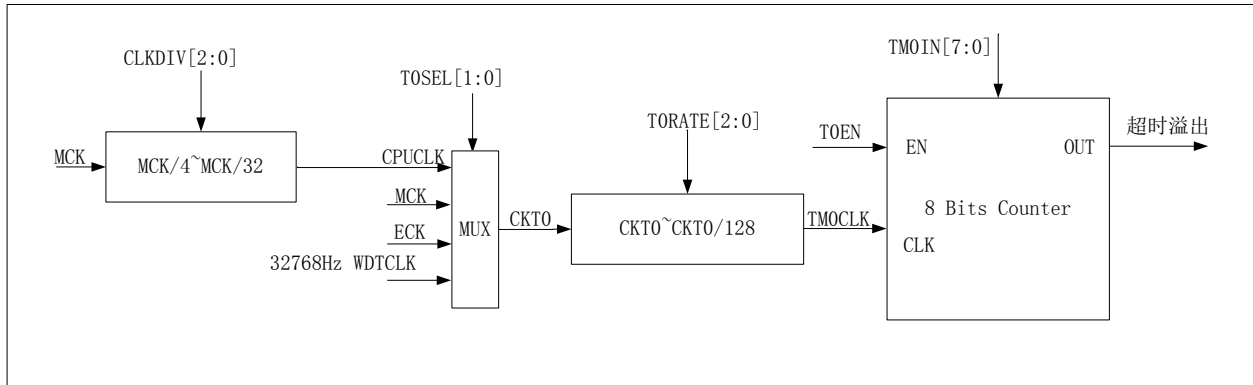


图12 定时器 0 功能框图

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 TM0CLK 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 12 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	T0EN	T0RATE[2:0]				T0RSTB	T0SEL[1:0]		0000u100
10H	TM0IN	TM0IN[7:0]								11111111
11H	TM0CNT	TM0CNT[7:0]								00000000

表 13 TM0CON 寄存器各位功能表

位地址	标识符	功能																		
7	T0EN	定时器 0 使能位 1: 使能定时器 0 0: 禁止定时器 0																		
6:4	T0RATE[2:0]	定时器 0 时钟选择 <table><tr><td>T0RATE [2:0]</td><td>TM0CLK</td></tr><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></table>	T0RATE [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
T0RATE [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	T0RSTB	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，T0RSTB 会自动置 1																		
1:0	T0SEL[1:0]	时钟源选择 <table><tr><td>T0SEL[1:0]</td><td>定时器 0 时钟源</td></tr><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 32768 WDT 时钟， 仅当内部 WDT 晶振打开时有效</td></tr></table>	T0SEL[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 32768 WDT 时钟， 仅当内部 WDT 晶振打开时有效								
T0SEL[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 32768 WDT 时钟， 仅当内部 WDT 晶振打开时有效																			

表 14 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 15 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: T0RSTB, 复位定时器 0 模块的计数器。
- 5) 设置寄存器标志位: TM0EN, 使能定时器 0 模块的 8 bits 计数器。
- 6) 当定时超时发生时, 程序计数器会跳转到 004H。

定时器 0 溢出时间计算方法:

定时器 0 溢出时间= (TM0IN+1) /TM0CLK.

2.6 I/O PORT

表 16 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	PT1CON0	PT11OD	PT1W1[3:0]				E1M	E0M[1:0]		00000000	
28h	PT3		PT3[6]			PT3[3:0]				xxxxxxxx	
29h	PT3EN		PT3EN[6]			PT3EN[3:0]				u0uu0000	
2ah	PT3PU		PT3PU[6]			PT3PU[3:0]				u0uu0000	
2bh	PT3CON		PT3CON[6]			PT3CON[3:0]				u0uu0000	
2ch	IODRV0				COMEN3		COMEN2	COMEN1	COMEN0	uuuu0000	
30h	PT5				PT5[4:0]					xxxxxxxx	
31h	PT5EN				PT5EN[4:0]					uuu00000	
32h	PT5PU				PT5PU[4:0]					uuu00000	
33h	PT5CON0				PT54OD	PT53OD	PT52OD	PT51OD	PT50OD	uuu00000	
34h	PT5CON1				PT5CON1[4:0]					uuu00000	
35h	PT1PD	PT1PD[7:4]								0000uuuu	
37h	PT5PD					PT5PD[2:0]				uuuuu000	
38h	PT1CON1			PT15OD	PT14OD	PT1W2[3:0]				u0000000	
12h	PT1TYPE	PT1_TYPE[7:0]								11111111	
13h	PT3TYPE		PT3_TYPE[6]				PT3_TYPE[3:0]				11uu1111
14h	PT5TYPE				PT5_TYPE[4:0]					uuu11111	
7ah	SYSCFG			PT1CON[2:0]						uu000uuu	

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。CSU34F10 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
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/W-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, 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 1 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

PT1PD 寄存器（地址为 35h）

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
PT1PD	PT1PD[7:4]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 PT1PD[7:4]: GPIO1 口下拉电阻使能标志

PT1PD[7] = GPIO1 bit 7 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT1PD[6] = GPIO1 bit 6 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT1PD[5] = GPIO1 bit 5 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT1PD[4] = GPIO1 bit 4 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

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	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

PT1_TYPE 寄存器（地址为 12h）

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
PT1_TYPE	PT1_TYPE[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1_TYPE[7:0]: GPIO1 口施密特输入使能标志

- PT1_TYPE[7] = GPIO1 bit 7 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT1_TYPE[6] = GPIO1 bit 6 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT1_TYPE[5] = GPIO1 bit 5 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT1_TYPE[4] = GPIO1 bit 4 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT1_TYPE[3] = GPIO1 bit 3 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT1_TYPE[2] = GPIO1 bit 2 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT1_TYPE[1] = GPIO1 bit 1 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT1_TYPE[0] = GPIO1 bit 0 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

PT1CON1 寄存器（地址为 38h）

特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1CON1			PT15OD	PT14OD	PT1W2[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 5 PT15OD: PT1.5 漏极开路使能位
0 = 禁止 PT1.5 漏极开路
1 = 使能 PT1.5 漏极开路
- Bit 4 PT14OD: PT1.4 漏极开路使能位
0 = 禁止 PT1.4 漏极开路
1 = 使能 PT1.4 漏极开路
- Bit 3 PT1W2[3]:PT3.6 外部中断 1 使能
0 = 禁止 PT3.6 外部中断 1

- 1 = 使能 PT3.6 外部中断 1
- 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

SYSCFG 寄存器（地址为 7ah）

特性	U-0	U-0	R/W -0	R/W -0	R/W-0	U-0	U-0	U-0
SYSCFG			PT1CON[2:0]					
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit5-3 SYSCFG[5:3]: GPIO1 口模拟/数字端口使能标志

PT1CON[2] = GPIO1bit 2 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT1CON[1] = GPIO1bit 1 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT1CON[0] = GPIO1bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

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

2.6.2 PT3 口

PT3 寄存器（地址为 28h）

特性	U-X	R/W-X	U-X	U-X	R/W-X	R/W-X	R/W-X	R/W-X
PT3		PT3[6]					PT3[3:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3[7:0]: GPIO3 口数据标志位

PT3[6] = GPIO3 bit 6 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	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3EN		PT3EN[6]					PT3EN[3:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3EN[7:0]: GPIO 3 口输入/输出控制标志

PT3EN[6] = GPIO3 bit 6 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	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3PU		PT3PU[6]					PT3PU[3:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-0 PT3PU[7:0]: GPIO3 口上拉电阻使能标志

PT3PU[6] = GPIO3 bit 6 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

PT3CON 寄存器（地址为 2bh）

特性	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3CON		PT3CON[6]					PT3CON[3:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-0 PT3CON[7:0]: GPIO3 口模拟/数字端口使能标志

PT3CON[6] = GPIO3bit 6 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

PT3_TYPE 寄存器（地址为 13h）

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
PT3_TYPE		PT3_TYPE[6]			PT3_TYPE[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 PT3_TYPE[6]: GPIO3 口施密特输入使能标志

PT3_TYPE[6] = GPIO3 bit 6 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

Bit 3-0 PT3_TYPE[3:0]: GPIO3 口施密特输入使能标志

PT3_TYPE[3] = GPIO3 bit 3 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

PT3_TYPE[2] = GPIO3 bit 2 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

PT3_TYPE[1] = GPIO3 bit 1 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

PT3_TYPE[0] = GPIO3 bit 0 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

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

2.6.3 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

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

Bit7-0 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

PT5PD 寄存器（地址为 37h）

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
PT5PD						PT5PD[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 2-0 PT5PD[2:0]: GPIO5 口下拉电阻使能标志

PT5PD[2] = GPIO5 bit 2 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT5PD[1] = GPIO5 bit 1 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT5PD[0] = GPIO5 bit 0 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT5CON0 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
PT5CON0				PT54OD	PT53OD	PT52OD	PT51OD	PT50OD
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit4 PT54OD: PT5.4 漏极开路使能
0 = 禁止 PT5.4 漏极开路
1 = 使能 PT5.4 漏极开路
- Bit3 PT53OD: PT5.3 漏极开路使能
0 = 禁止 PT5.3 漏极开路
1 = 使能 PT5.3 漏极开路
- Bit2 PT52OD: PT5.2 漏极开路使能
0 = 禁止 PT5.2 漏极开路
1 = 使能 PT5.2 漏极开路
- Bit1 PT51OD: PT5.1 漏极开路使能
0 = 禁止 PT5.1 漏极开路
1 = 使能 PT5.1 漏极开路
- Bit0 PT50OD: PT5.0 漏极开路使能
0 = 禁止 PT5.0 漏极开路
1 = 使能 PT5.0 漏极开路

PT5_TYPE 寄存器（地址为 14h）

特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5_TYPE				PT5_TYPE[4:0]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 4-0 PT5_TYPE[4:0]: GPIO5 口施密特输入使能标志
PT5_TYPE[4] = GPIO5 bit 4 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT5_TYPE[3] = GPIO5 bit 3 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT5_TYPE[2] = GPIO5 bit 2 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT5_TYPE[1] = GPIO5 bit 1 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
PT5_TYPE[0] = GPIO5 bit 0 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

PT5CON1 寄存器（地址为 34h）

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
PT5CON1				PT5CON1[4:0]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit4-0 PT5CON[4: 0]: GPIO5 口模拟/数字端口使能标志
PT5CON[4] = GPIO5bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口
PT5CON[3] = GPIO5bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口
PT5CON[2] = GPIO5bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口
PT5CON[1] = GPIO5bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口
PT5CON[0] = GPIO5bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

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

2.6.4 驱动能力配置

COM0~COM3 驱动能力可配置，驱动能力有 2 档可选，分别是 40/20mA。其他 IO 脚的驱动能力为默认驱动能力 20mA。

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
2ch	IODRV0					COMEN3	COMEN2	COMEN1	COMEN0	00000000

表 17 IODRV0 寄存器各位功能表

位地址	标识符	功能
3	COMEN3	COM3 (PT1.7) 驱动能力配置
		COMEN3 驱动能力配置
		1 40mA
		0 20mA
2	COMEN2	COM2 (PT1.6) 驱动能力配置
		COMEN1 驱动能力配置
		1 40mA
		0 20mA
1	COMEN1	COM1 (PT1.5) 驱动能力配置
		COMEN1 驱动能力配置
		1 40mA
		0 20mA
0	COMEN0	COM0 (PT1.4) 驱动能力配置
		COMEN0 驱动能力配置
		1 40mA
		0 20mA

2.6.5 输入逻辑电平电压配置

表 18 PTVTH 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
2dh	PTVTH						PT5VTH	PT3VTH	PT1VTH	uuuuu000

Bit 2-0 GPIO 输入逻辑电平电压配置

PT5VTH = GPIO5 输入逻辑电平电压配置

1'b0: 选择 VIH1/VIL1。

1'b1: 选择 VIH2/VIL2

PT3VTH = GPIO3 输入逻辑电平电压配置

1'b0: 选择 VIH1/VIL1。

1'b1: 选择 VIH2/VIL2

PT1VTH = GPIO1 输入逻辑电平电压配置

1'b0: 选择 VIH1/VIL1。

1'b1: 选择 VIH2/VIL2

VIH1/VIL1 和 VIH2/VIL2 的描述见[直流特性 \(VDD = 5V, TA = 25°C, 如无其他说明则都是此条件\)](#)

3 增强功能

3.1 防误操作

CSU34F10 为防误操作影响，在对 MCK/LPOWER SFR 进行写操作时，需先向 SEC 寄存器连续写入 C9h/3Ah/D3h，解锁写操作在 12 个指令周期内有效。12 个指令周期后再对 MCK/LPOWER 寄存器写操作则无效。

表 19 ISP 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电 复位值
5eh	SEC	SECURITY[7:0]								00000000

3.2 Halt和Sleep模式

CSU34F10 支持多种低功耗工作模式。为了使其处于待机状态，可以让 CPU 停止工作使 CSU34F10 进行停止或睡眠模式，减低功耗。具体描述如下：

表 20 低功耗寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
4dh	SSLPSAVE	SSLPSAVE[7:0]								00000000
4eh	LPOWER					FLT_EN	SUPSLEEP	DPSTB	LDOON	uuuu0000

表 21 SSLPSAVE 寄存器各位功能表

位地址	标识符	功能
7:0	SSLPSAVE[7: 0]	超低功耗模式保存寄存器。 进入超低功耗模式前，将信息保存进 SSLPSAVE[7: 0]；超低功耗模式唤醒后，SSLPSAVE[7: 0]数据不会丢失。

表 22 LPOWER 寄存器各位功能表

位地址	标识符	功能
3	FLT_EN	低功耗模式下，外部中断滤波使能信号 1'b0: 不对外部中断信号滤波 1'b1: 对外部中断进行 1us 滤波
2	SUPSLEEP	超低功耗模式选择位： 1'b0: 不使能超低功耗 sleep 模式 1'b1: 使能超低功耗 sleep 模式
1	DPSTB	sleep 模式下，FLASH 功耗模式选择 1'b0: 不使能 Flash 低功耗模式 1'b1: 使能 Flash 低功耗模式
0	LDOON	sleep 模式下 LDO 关闭选择： 1'b0: sleep 模式下不关闭 LDO 1'b1: sleep 模式下关闭 LDO

LPOWER 为写保护寄存器，要对 LPOWER 进行写操作，需先向 SEC 寄存器连续写入：

C9h/3Ah/D3h，解锁写操作在 12 个指令周期内有效。12 个指令周期后写 LPOWER 寄存器操作无效。

表 23 sleep 模式列表

模式	LPOWER[2:0]配置	芯片状态	说明
低功耗模式 0	3'b000	LDO 不关闭；时钟停止；逻辑功能不工作；Flash 低功耗模式不使能。	只能通过外部中断唤醒。sleep 前状态可以保持
低功耗模式 1	3'b010	LDO 不关闭；时钟停止；逻辑功能不工作；Flash 处于低功耗模式。	只能通过外部中断唤醒。sleep 前状态可以保持
低功耗模式 2	3'b0x1	LDO 关闭；时钟停止；逻辑功能不工作；Flash 处于低功耗模式。	只能通过外部中断唤醒，且外部中断唤醒方式可选（详见 IO PORT 一节描述）。唤醒需 100us；sleep 前状态可以保持。
超低功耗模式	3'b1xx	处于 PowerDown 状态，	进入超低功耗模式需保持外部中断为高电平，超低功耗模式只能通过外部中断唤醒，外部中断唤醒只需将中断信号拉低并维持 10us，即可唤醒芯片。唤醒到正常工作所等待时间与上电复位时间相同；仅可保存 sleep 前 SSLPSAVE[7:0]寄存器信息,其他信息丢失。

halt 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。

sleep 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。

可配置 DPSTB 和 LDOON 进入不同的 sleep 模式，详见上表。

LPOWER 寄存器被配置为 3'b1xx 后再执行 sleep 指令进入超低功耗模式，超低功耗模式下，功耗接近于 0。进入超低功耗前，需将使能的外部中断置高。当外部中断拉低，并且维持 10us 后，芯片可唤醒。

注：

芯片如果处于 sleep 状态，这时候降低电压，配置 1.8V/2.0V/2.4/3.0/3.6V 低电压复位不会起作用，低于 1.8V 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[7:0]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  
movlw c9h ;向 sec 寄存器连续写入 c9h/3ah/d3h，对 MCK/  
movwf sec ; TRIM_EX/SRADTRIM/LPOWER 寄存器保护解锁  
movlw 3ah  
movwf sec  
movlw d3h  
movwf sec  
movlw 03h ;选择低功耗模式 2  
movwf lpower ;  
sleep ;进入睡眠模式  
nop ; 保证 CPU 重启后程序能正常工作  
...
```

3.3 看门狗(WDT)

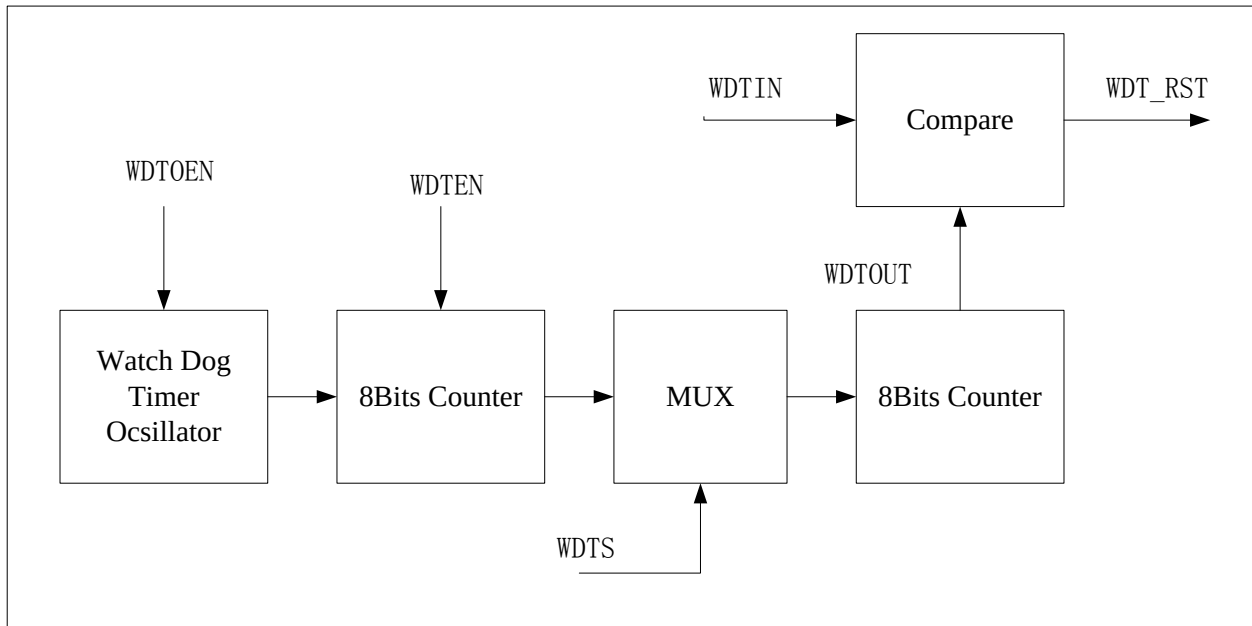


图13 看门狗定时器功能框图

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

当用户把 CST_WDT 清 0 时，则内部的看门狗定时器振荡器（32KHz）将会启动，产生的时钟被送到“8 bits 计数器 1”。当用户置位 WDTEN 时，“8 bits 计数器 1”开始计数，“8 bits 计数器 1”的输出是内部信号 WDTA[7:0]，被发送到一个受寄存器标志位 WDTS[2:0]控制的多路选择器，选择器的输出作为“8 bits 计数器 2”的时钟输入。当“8 bits 计数器 2”计数值与 WDTIN 数值相等时溢出，溢出时它会发送 WDT_RST 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 24 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			WDT_DLY		WDTS[2:0]			0uu0u000
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
6:5	保留	保留
4	WDT_DLY	看门狗复位延时使能位 0：看门狗复位后不延时 1：看门狗复位后延时 8ms
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	

注：WDT_DLY 延时 8ms 时系统一直处于看门狗复位状态，8ms 后解除；且 8ms 为使用内部 16M

$$\frac{16M}{f} * 8$$

晶振时的计时时间，若系统时钟使用其他时钟，则计时时间为 $\frac{16M}{f} * 8$ ms，其中 f 为系统使用的时钟频率，如选择外部 1M 时钟时，则计时时间为 128ms。

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：

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

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]	2000ms
001	WDTA [1]	1000ms
010	WDTA [2]	500ms
011	WDTA [3]	250ms
100	WDTA [4]	125ms
101	WDTA [5]	62.5ms
110	WDTA [6]	31.25ms
111	WDTA [7]	15.625ms

3.4 定时/计数器 2

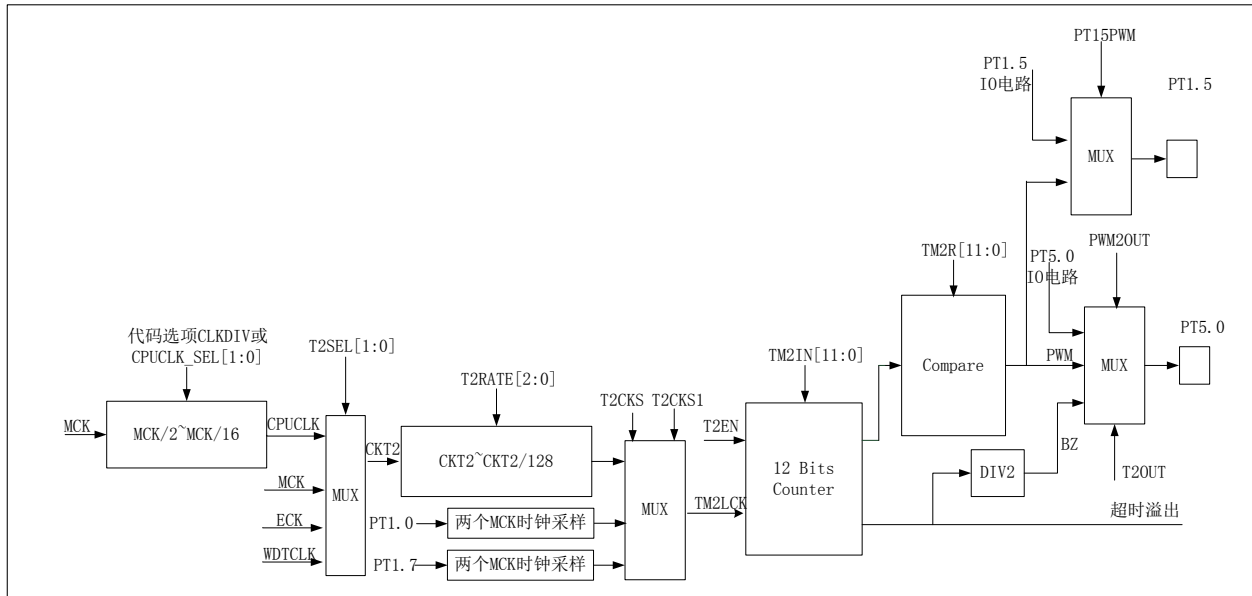


图14 定时/计数器 2 模块的功能框图

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；

3.4.1 寄存器说明

表 25 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
47h	TM2INH					TM2IN[11:8]				uuuu1111
48h	TM2CNTH					TM2CNT[11:8]				uuuu0000
49h	TM2RH					TM2R[11:8]				uuuu0000
7dh 1	SYSCFG						T2CKS1	PT15PWM[1:0]		xx000000
7eh	TMCON				T2SEL[1]			T2SEL[0]		00000000

表 26 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><tr><td>T2RATE [2:0]</td><td>TM2CLK</td></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> 可通过 T2SEL [1]选择定时器 2 时钟	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																			
3	T2CKS	定时/计数器 2 时钟源选择位 与寄存器 SYSCFG1 的 T2CKS1 位共同作用，选择定时/计数器 2 时钟源 <table><tr><td>T2CKS1</td><td>T2CKS</td><td>定时/计数器 2 时钟源</td></tr><tr><td>0</td><td>0</td><td>CKT2 的分频时钟</td></tr><tr><td>0</td><td>1</td><td>外部计数输入 PT1.0 作为时钟</td></tr><tr><td>1</td><td>0</td><td>外部计数输入 PT1.7 作为时钟</td></tr><tr><td>1</td><td>1</td><td>禁止</td></tr></table>	T2CKS1	T2CKS	定时/计数器 2 时钟源	0	0	CKT2 的分频时钟	0	1	外部计数输入 PT1.0 作为时钟	1	0	外部计数输入 PT1.7 作为时钟	1	1	禁止			
T2CKS1	T2CKS	定时/计数器 2 时钟源																		
0	0	CKT2 的分频时钟																		
0	1	外部计数输入 PT1.0 作为时钟																		
1	0	外部计数输入 PT1.7 作为时钟																		
1	1	禁止																		
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><tr><td>T2OUT</td><td>PWM2OUT</td><td>PT5.0 输出控制，仅当 PT5.0 配置为输出有效</td></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	<table><tr><td>0</td><td>1</td><td>蜂鸣器输出</td></tr><tr><td>1</td><td>0</td><td>PWM2 输出</td></tr><tr><td>1</td><td>1</td><td>PWM2 取反输出</td></tr></table>	0	1	蜂鸣器输出	1	0	PWM2 输出	1	1	PWM2 取反输出									
0	1	蜂鸣器输出																		
1	0	PWM2 输出																		
1	1	PWM2 取反输出																		

表 27 TM2IN 寄存器各位功能表

位地址	标识符	功能
7 : 0	TM2IN[7:0]	定时/计数器溢出值低 8 位

表 28 TM2INH 寄存器各位功能表

位地址	标识符	功能
3 : 0	TM2IN[11:8]	定时/计数器溢出值高 4 位

表 29 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 计数寄存器低 8 位，只读

表 30 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 计数寄存器高 4 位，只读

表 31 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低 8 位

表 32 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高 4 位

表 33 SYSCFG1 寄存器各位功能表

位地址	标识符	功能	
2	T2CKS1	定时/计数器 2 时钟源选择位 详见寄存器 TM2CON 的 T2CKS 描述	
1 : 0	PT15PWM[1:0]	PT1.5 口输出控制	
		PT15PWM[1:0]	PT1.5 输出控制，仅当 PT1.5 配置为输出有效
		2'b00	IO 输出
		2'b01	蜂鸣器输出
		2'b10	PWM2 输出
		2'b11	PWM2 取反输出

表 34 TMCON 寄存器各位功能表

位地址	标识符	功能	
4	T2SEL[1]	定时器 2 时钟选择	
1	T2SEL[0]	T2SEL[1:0]	CKT2 时钟选择
		00	CPUCLK
		01	MCK
		10	RTC
		11	WDT CLK(32K)

3.4.2 12 位可编程定时器

示范程序

以 16M 内部晶振，指令周期=4 个时钟周期为例，选择 CPU 时钟作为 ckt2 时钟源，并对时钟源进行 4 分频，产生 100us 的时间中断

```
...  
movlw 00100000b ;设置 tm2clk 为时钟源的 4 分频，fcup/4  
movwf tm2con ;定时/计数器 2 复位, I/O 口输出，选择 ckt2 作为时钟源  
bcf syscfg1, 2 ;(时钟源由 tm2con 的 t2cks 及 syscfg1 的 t2cks1 共同决定)  
bcf syscfg1, 1 ;I/O 口输出, 蜂鸣器、pwm2 不输出  
bcf syscfg1, 0 ;(输出由 syscfg1 的 pt15pwm[1:0] 决定)  
bcf tmcon, 1 ;ckt2 时钟选择，选择 cpu 时钟作为 ckt2 fcup=16M/4  
bcf tmcon, 4 ;(ckt2 由 tmcon 的 t2sel[1] 及 t2sel[0] 共同决定)  
movlw 99 ;设置定时时间为 (99+1)/(16M/4/4)=100us  
movwf tm2in ;  
clrf tm2inh ;上电复位值为 (uuuu1111), 不用时需清零  
movlw 11000000b ;使能全局中断及定时器 2 中断  
movwf inte ;  
bsf tm2con, 7 ;使能定时器 2  
...
```

3.4.3 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：

定时器 2 溢出时间 = $(TM2IN + 1) / TM2CLK$. (TM2IN 不为 0)

蜂鸣器周期计算方法：

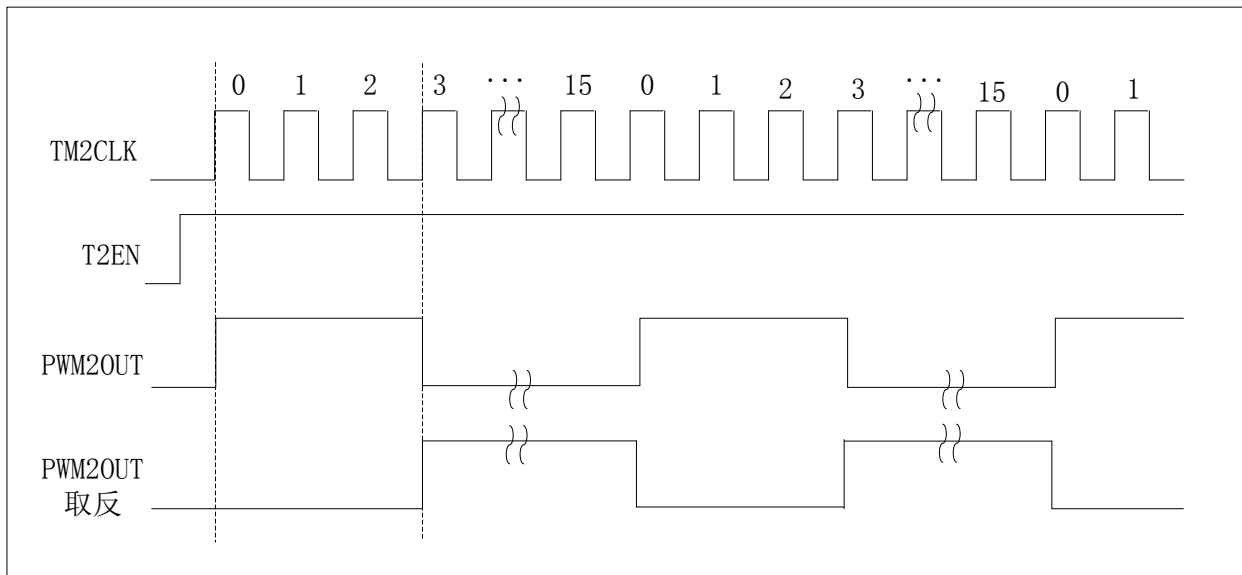
蜂鸣器周期 = $(TM2IN + 1) * 2 / TM2CLK$. (TM2IN 不为 0)

3.4.4 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：



3.5 定时/计数器 3

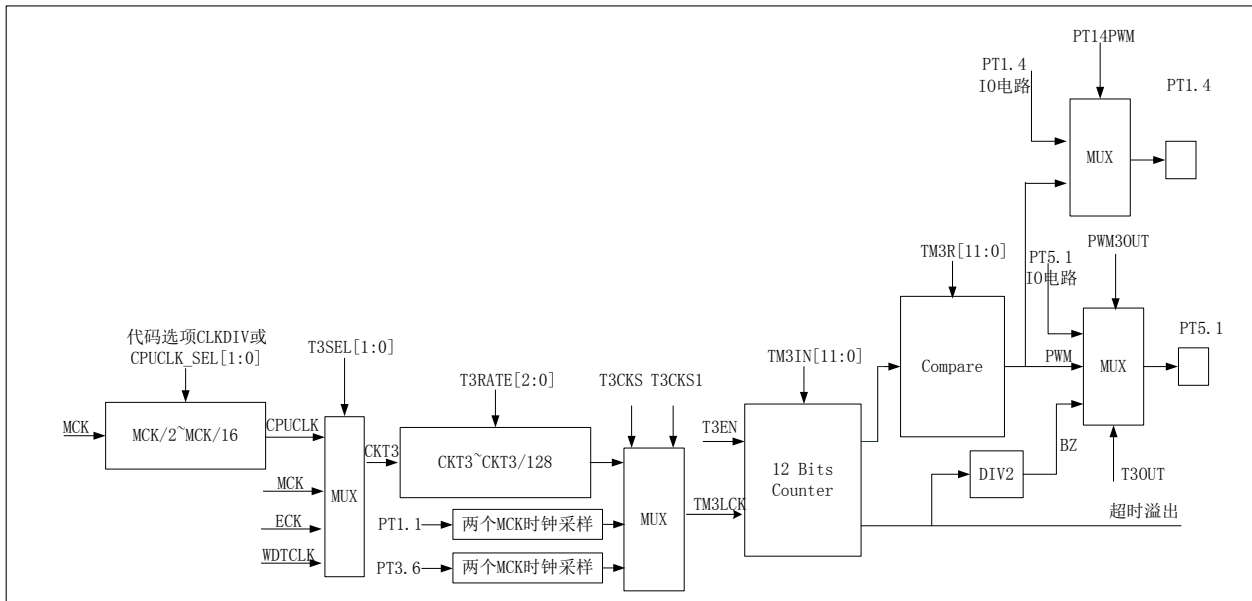


图15 定时/计数器 3 模块的功能框图

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 00h 递增到 TM3IN。用户需要设置 TM3IN（定时器模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，BZ 输出信号发生跳变。

主要功能：

- 1) 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；

3.5.1 寄存器说明

表 35 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
4ah	TM3INH					TM3IN[11:8]				uuuu1111
4bh	TM3CNTH					TM3CNT[11:8]				uuuu0000
4ch	TM3RH					TM3R[11:8]				uuuu0000
7dh	SYSCFG1			T3CKS1	PT14PWM					xx000000
7eh	TMCON			T3SEL[1]			T3SEL[0]			00000000

表 36 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><tr><th>T3RATE [2:0]</th><th>TM3CLK</th></tr><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></table> 可通过 T3SEL[1:0]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 与寄存器 SYSCFG1 的 T3CKS1 位共同作用，选择定时/计数器 2 时钟源 <table><tr><th>T3CKS1</th><th>T3CKS</th><th>定时/计数器 3 时钟源</th></tr><tr><td>0</td><td>0</td><td>CKT3 的分频时钟</td></tr><tr><td>0</td><td>1</td><td>外部计数输入 PT1.1 作为时钟</td></tr><tr><td>1</td><td>0</td><td>外部计数输入 PT3.6 作为时钟</td></tr><tr><td>1</td><td>1</td><td>禁止</td></tr></table>	T3CKS1	T3CKS	定时/计数器 3 时钟源	0	0	CKT3 的分频时钟	0	1	外部计数输入 PT1.1 作为时钟	1	0	外部计数输入 PT3.6 作为时钟	1	1	禁止			
T3CKS1	T3CKS	定时/计数器 3 时钟源																		
0	0	CKT3 的分频时钟																		
0	1	外部计数输入 PT1.1 作为时钟																		
1	0	外部计数输入 PT3.6 作为时钟																		
1	1	禁止																		
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><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><tr><td>0</td><td>0</td><td>IO 输出</td></tr><tr><td>0</td><td>1</td><td>蜂鸣器输出</td></tr><tr><td>1</td><td>0</td><td>PWM3 输出</td></tr><tr><td>1</td><td>1</td><td>PWM3 取反输出</td></tr></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	蜂鸣器输出	1	0	PWM3 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	蜂鸣器输出																		
1	0	PWM3 输出																		
1	1	PWM3 取反输出																		
0	PWM3OUT																			

表 37 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 38 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 39 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 40 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 41 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 42 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 43 SYSCFG1 寄存器各位功能表

位地址	标识符	功能	
5	T3CKS1	定时/计数器 3 时钟源选择位 详见寄存器 TM3CON 的 T3CKS 描述	
4 : 3	PT14PWM[1:0]	PT1.4 口输出控制	
		PT14PWM[1:0]	PT1.4 输出控制，仅当 PT1.4 配置为输出有效
		2'b00	IO 输出
		2'b01	蜂鸣器输出
		2'b10	PWM3 输出
		2'b11	PWM3 取反输出

表 44 TMCON 寄存器各位功能表

位地址	标识符	功能	
5	T3SEL[1]	定时器 3 时钟选择	
2	T3SEL[0]	T3SEL[1:0]	CKT3 时钟选择
		00	CPUCLK
		01	MCK
		10	RTC
		11	WDT CLK(32K)

3.5.2 12 位可编程定时器

示范程序

以 16M 内部晶振，指令周期=4 个时钟周期为例，选择 mck 时钟作为 ckt3 时钟源，并对时钟源进行 16 分频，产生 100us 的时间中断

```
...  
movlw 01000000b ;设置 tm2clk 为时钟源的 16 分频，fmck/16  
movwf tm3con ;定时/计数器 3 复位, I/O 口输出, 选择 ckt3 作为时钟源  
bcf syscfg1, 5 ;(时钟源由 tm3con 的 t3cks 及 syscfg1 的 t3cks1 共同决定)  
bcf syscfg1, 4 ;I/O 口输出, 蜂鸣器、pwm3 不输出  
bcf syscfg1, 3 ;(输出由 syscfg1 的 pt14pwm[1:0] 决定)  
bcf tmcon, 5 ;ckt3 时钟选择, 选择 mck 时钟作为时钟源 fmck=16M  
bsf tmcon, 2 ;(ckt3 由 tmcon 的 t3sel[1] 及 t3sel[0] 共同决定)  
movlw 99 ;设置定时时间为 (99+1)/(16M/16)=100us  
movwf tm3in ;  
clrf tm3inh ;上电复位值为 (uuuu1111), 不用时需清零  
bsf inte, 7 ;使能全局中断  
bsf inte2, 4 ;定时器 3 中断  
bsf tm3con, 7 ;使能定时器 3  
...
```

3.5.3 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：

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

蜂鸣器周期计算方法：

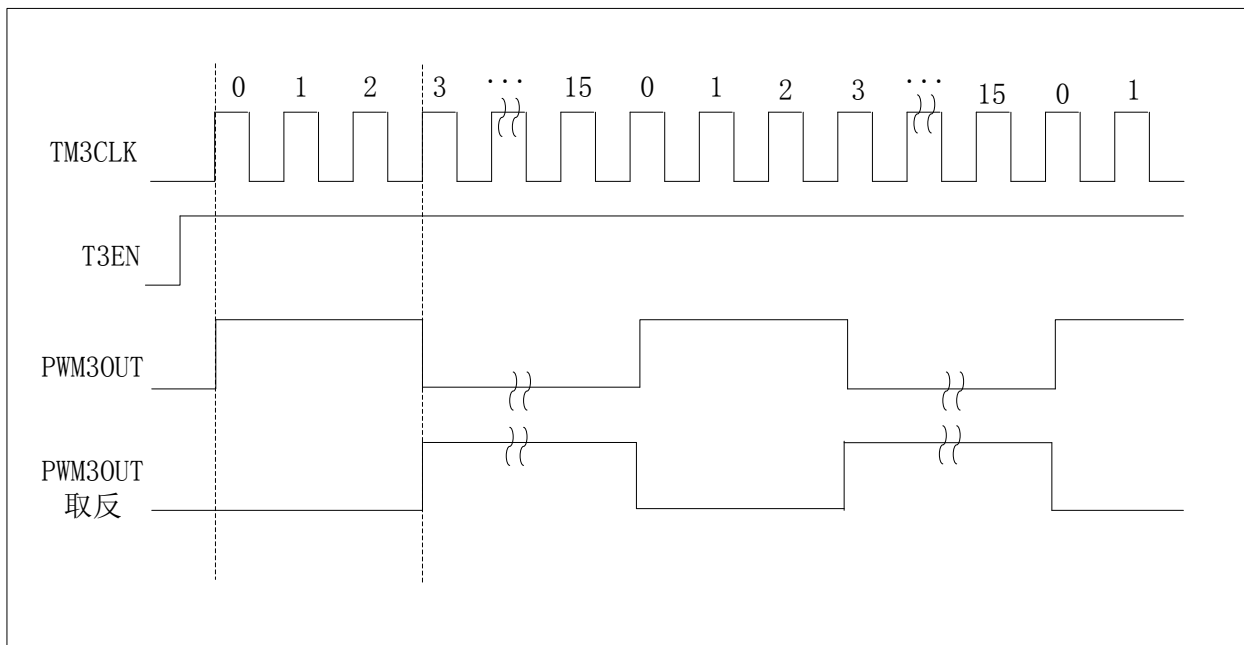
蜂鸣器周期 = $(TM3IN+1) * 2 / TM3CLK$. (TM3IN 不为 0)

3.5.4 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：



3.6 定时/计数器 4

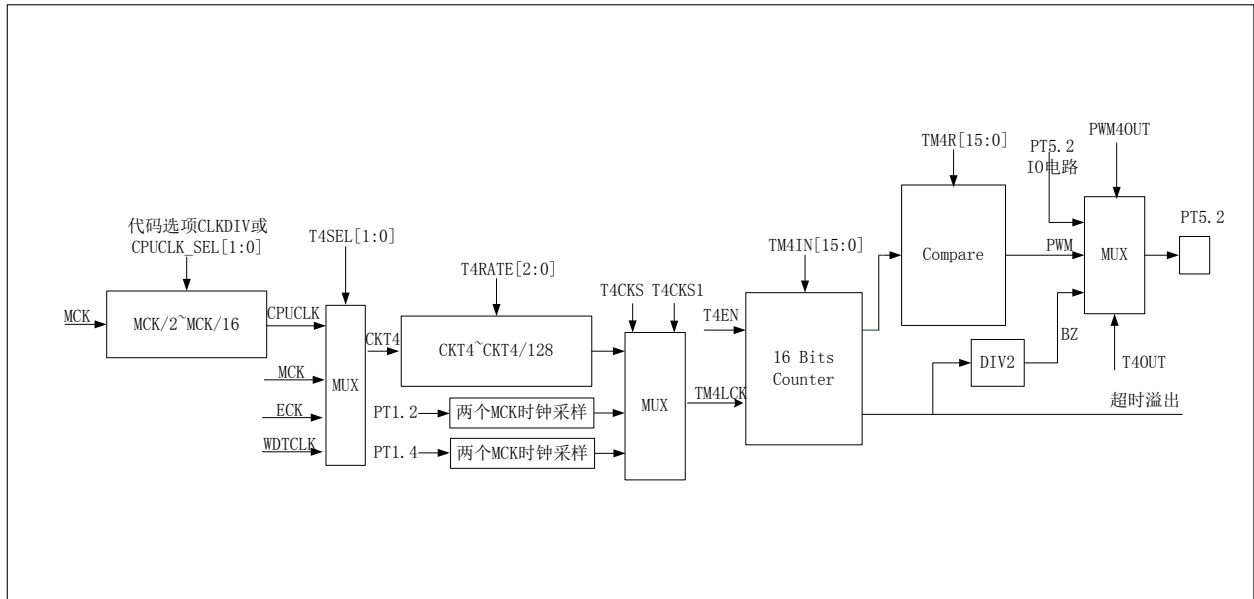


图16 定时/计数器 4 模块的功能框图

定时/计数器 4 模块的输入是 MCK 或者 CPUCLK。当用户设置了定时/计数器 4 模块的使能标志，16 bits 计数器将启动，从 00h 递增至 TM4IN。用户需要设置 TM4IN（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) 16 位可编程定时器；
- 2) 外部事件计数；
- 3) 蜂鸣器输出；
- 4) PWM 输出；

3.6.1 寄存器描述

表 45 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			TM4IF						uu00uuuu
3dh	INTE2			TM4IE						uu00uuuu
40h	TM4CON	T4EN	T4RATE[2:0]			T4CKS	T4RSTB			00000100
41h	TM4INL	TM4IN[7:0]								11111111
42h	TM4INH	TM4IN[15:8]								11111111
43h	TM4CNTL	TM4CNT[7:0]								00000000
44h	TM4CNTH	TM4CNT[15:8]								00000000
45h	TM4RL	TM4R[7:0]								00000000
46h	TM4RH	TM4R[15:8]								00000000
7ah	SYSCFG	PT16PWM[1:0]					T4CKS1	PT52PWM[1:0]		00xxx000
7eh	TMCON		T4SEL[1]			T4SEL[0]				00000000

表 46 TM4CON 寄存器各位功能表

位地址	标识符	功能																		
7	T4EN	定时/计数器 4 使能位 1: 使能定时器 4 0: 禁止定时器 4																		
6:4	T4RATE[2:0]	定时/计数器 4 时钟选择 <table><tr><td>T4RATE [2:0]</td><td>TM4CLK</td></tr><tr><td>000</td><td>CKT4</td></tr><tr><td>001</td><td>CKT4 /2</td></tr><tr><td>010</td><td>CKT4 /4</td></tr><tr><td>011</td><td>CKT4 /8</td></tr><tr><td>100</td><td>CKT4 /16</td></tr><tr><td>101</td><td>CKT4 /32</td></tr><tr><td>110</td><td>CKT4 /64</td></tr><tr><td>111</td><td>CKT4 /128</td></tr></table>	T4RATE [2:0]	TM4CLK	000	CKT4	001	CKT4 /2	010	CKT4 /4	011	CKT4 /8	100	CKT4 /16	101	CKT4 /32	110	CKT4 /64	111	CKT4 /128
T4RATE [2:0]	TM4CLK																			
000	CKT4																			
001	CKT4 /2																			
010	CKT4 /4																			
011	CKT4 /8																			
100	CKT4 /16																			
101	CKT4 /32																			
110	CKT4 /64																			
111	CKT4 /128																			
3	T4CKS	定时/计数器 4 时钟源选择位 与寄存器 SYSCFG 的 T4CKS1 位共同作用，选择定时/计数器 4 时钟源 <table><tr><td>T4CKS1</td><td>T4CKS</td><td>定时/计数器 4 时钟源</td></tr><tr><td>0</td><td>0</td><td>CKT4 的分频时钟</td></tr><tr><td>0</td><td>1</td><td>外部计数输入 PT1.2 作为时钟</td></tr><tr><td>1</td><td>0</td><td>外部计数输入 PT1.4 作为时钟</td></tr><tr><td>1</td><td>1</td><td>禁止</td></tr></table>	T4CKS1	T4CKS	定时/计数器 4 时钟源	0	0	CKT4 的分频时钟	0	1	外部计数输入 PT1.2 作为时钟	1	0	外部计数输入 PT1.4 作为时钟	1	1	禁止			
T4CKS1	T4CKS	定时/计数器 4 时钟源																		
0	0	CKT4 的分频时钟																		
0	1	外部计数输入 PT1.2 作为时钟																		
1	0	外部计数输入 PT1.4 作为时钟																		
1	1	禁止																		
2	T4RSTB	定时/计数器 4 复位 1: 禁止定时/计数器 4 复位 0: 使能定时/计数器 4 复位 当将该位为 0 时，定时器 4 复位后，T4RSBT 会自动置 1																		

表 47 TM4IN 寄存器各位功能表

位地址	标识符	功能
15:0	TM4IN[15:0]	定时/计数器溢出值

表 48 TM4CNT 寄存器各位功能表

位地址	标识符	功能
15:0	TM4CNT[15:0]	定时/计数器 4 计数寄存器，只读

表 49 TM4R 寄存器各位功能表

位地址	标识符	功能
15:0	TM4R[15:0]	定时/计数器 4 的 PWM 高电平占空比控制寄存器

表 50 TMCON 寄存器各位功能表

位地址	标识符	功能	
6	T4SEL[1]	定时器 4 时钟选择	
3	T4SEL[0]	T4SEL[1:0]	CKT4 时钟选择
		00	CPUCLK
		01	MCK
		10	RTC

		11	WDT CLK(32K)
--	--	----	--------------

表 51 SYSCFG 寄存器各位功能表

位地址	标识符	功能
7: 6	PT16PWM[1:0]	PT1.6 口输出控制
		PT1.6PWM[1:0] PT1.6 输出控制，仅当 PT1.6 配置为输出有效
		2'b00 IO 输出
		2'b01 蜂鸣器输出
		2'b10 PWM4 输出
		2'b11 PWM4 取反输出
2	T4CKS1	定时/计数器 4 时钟源选择位 详见寄存器 TM4CON 的 T4CKS 描述
1: 0	PT52PWM[1:0]	PT5.2 口输出控制
		PT52PWM[1:0] PT5.2 输出控制，仅当 PT5.2 配置为输出有效
		2'b00 IO 输出
		2'b01 蜂鸣器输出
		2'b10 PWM4 输出
		2'b11 PWM4 取反输出

3.6.2 16 位可编程定时器

示范程序

以 16M 内部晶振，指令周期=4 个时钟周期为例，选择 CPU 时钟作为 ckt4 时钟源，不对时钟源进行分频，产生 100us 的时间中断

...	
movlw 00000000b	;设置 tm2clk 为时钟源不分频，fcup
movwf tm4con	;定时/计数器 4 复位, IO 口输出, 选择 ckt4 作为时钟源
clrf syscfg	; (时钟源由 tm4con 的 t4cks 及 syscfg 的 t4cks1 共同决定)
	; IO 口输出, 蜂鸣器、pwm3 不输出
bcf tmcon, 6	; (输出由 syscfg 的 pt16pwm[1:0]、pt52pwm[1:0] 决定)
bcf tmcon, 3	; ckt4 时钟选择, 选择 cpu 时钟作为时钟源 fcpu=16M/4
movlw 8fh	; (ckt3 由 tmcon 的 t3sel[1] 及 t3sel[0] 共同决定)
movwf tm4inl	; 设置定时时间为 (399+1)/(16M/4)=100us
movlw 1	; 399 = 18fh
movwf tm4inh	; 上电复位值为 (11111111), 不用时需清零
bsf inte, 7	; 使能全局中断
bsf inte2, 5	; 定时器 4 中断
bsf tm4con, 7	; 使能定时器 4
...	

3.6.3 蜂鸣器

操作：

- 1) 设置 TM4CLK，为定时器模块选择输入。
- 2) 设置 TM4IN，选择定时器溢出值。
- 3) 设置寄存器标志位：TM4IE 与 GIE，使能定时器中断。
- 4) 清零寄存器标志位：T4RSTB，复位定时器模块的计数器。
- 5) 设置寄存器标志位：T4EN，使能定时器模块的 16 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 TM4IF 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：

定时器 2 溢出时间 = $(TM4IN + 1) / TM4CLK$. (TM4IN 不为 0)

蜂鸣器周期计算方法：

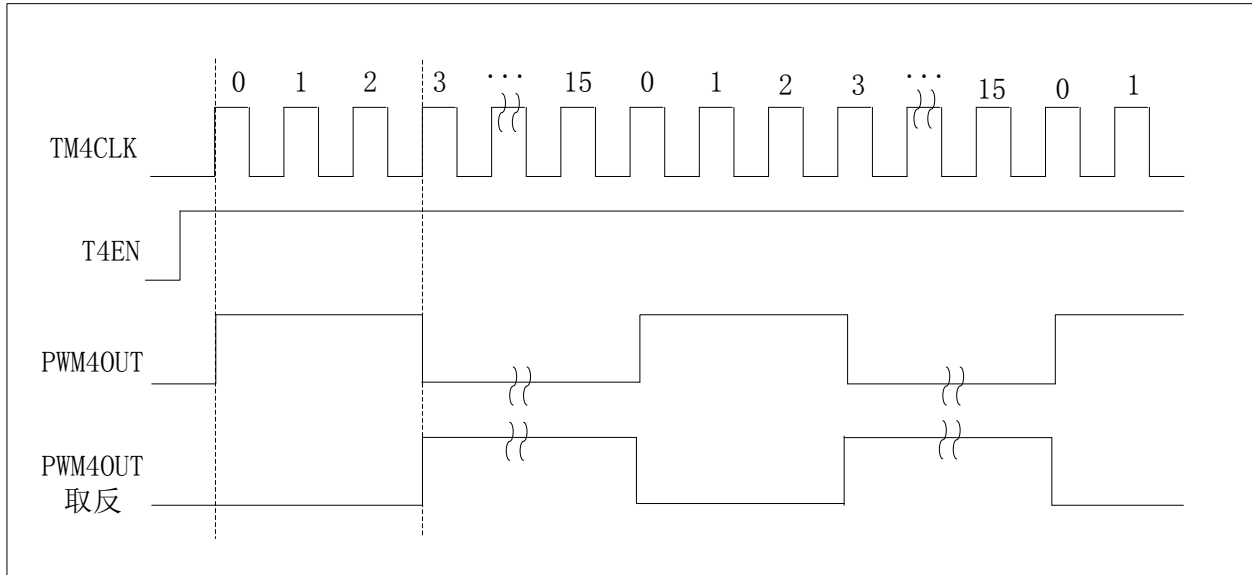
蜂鸣器周期 = $(TM4IN + 1) * 2 / TM4CLK$. (TM4IN 不为 0)

3.6.4 PWM

操作：

- 1) 设置 TM4CLK，为定时/计数器 4 模块选择输入。
- 2) 设置 TM4IN 来配置 PWM4 的周期。
- 3) 设置 TM4R 来配置 PWM4 的高电平的脉宽。
- 4) 使能 PWM4OUT 输出，配置 PT1.6 为输出端口，之后把 T4EN 置 1 启动定时器。
- 5) PWM4 从 PT1.6 输出。

周期为 TM4IN+1，高电平脉宽为 TM4R。如 TM4IN=0x0F，TM4R=0x03 的 PWM4 波形输出如下：



3.7 串行通信接口(UART)

CSU34F10 主要提供一个可编程全双工串行通信接口。该接口能同时进行数据的发送和接收，也可以作为一个同步移位寄存器使用。

3.7.1 工作方式

主要提供四种工作模式：

表 52 串口通信工作模式

SM0	SM1	模式	类型	波特率	帧长度	起始位	停止位	第 9 位
0	0	0	同步	fcpuclk/6	8bits	无	无	无
0	1	1	异步	定时器 2/3/4 的溢出率/16 或者 32	10bits	1	1	无
				SMOD 波特率				
				0 定时器溢出率/32				
1	0	2	异步	定时器 2/3/4 的溢出率/16 或者 32	11bits	1	1	0,1
				SMOD 波特率				
				0 fcpuclk/32				
1	1	3	异步	定时器 2/3/4 的溢出率/16 或者 32	11bits	1	1	0,1
				SMOD 波特率				
				0 定时器溢出率/32				

模式 0:

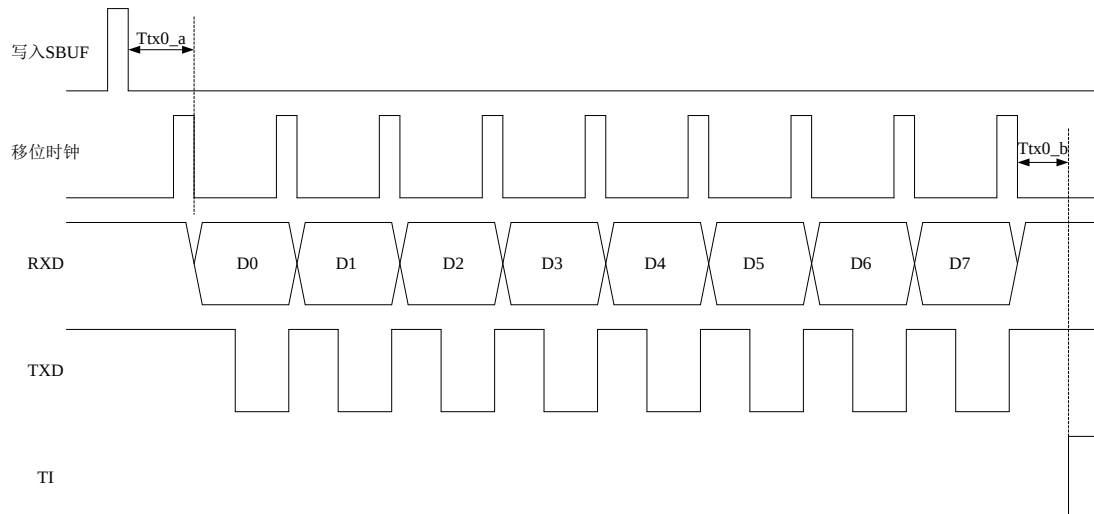


图17 UART 模式 0 发送数据波形

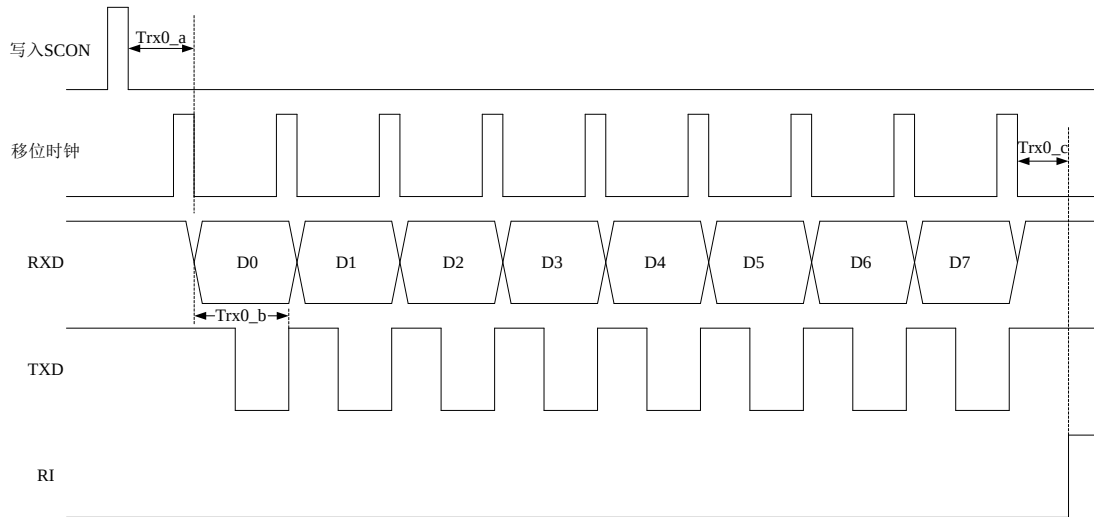


图18 UART 模式 0 接收数据波形

模式 1:

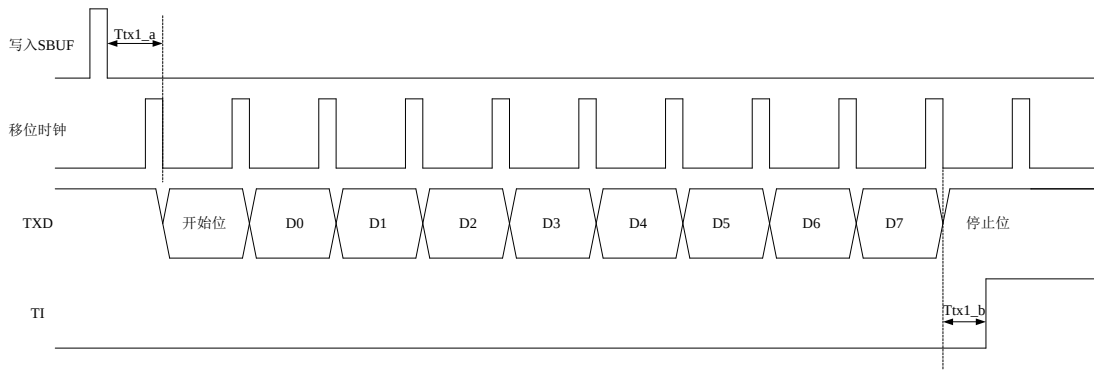


图19 UART 模式 1 发送数据波形

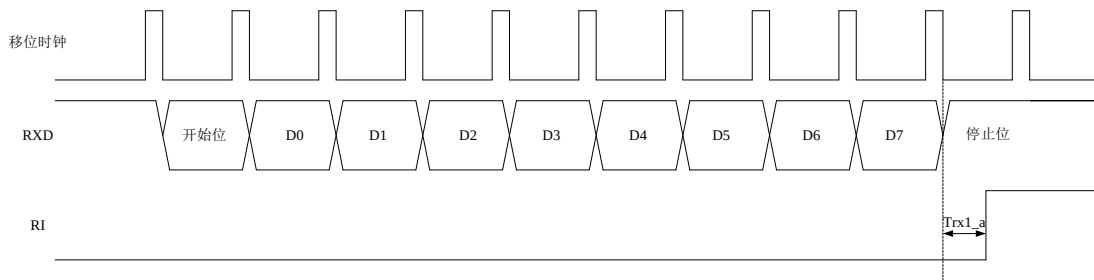


图20 UART 模式 1 接收数据波形

模式 2:

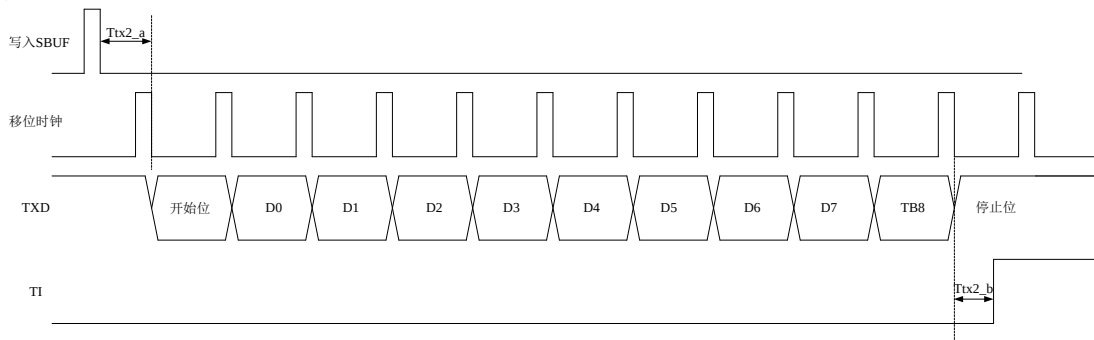


图21 UART 模式 2 发送数据波形

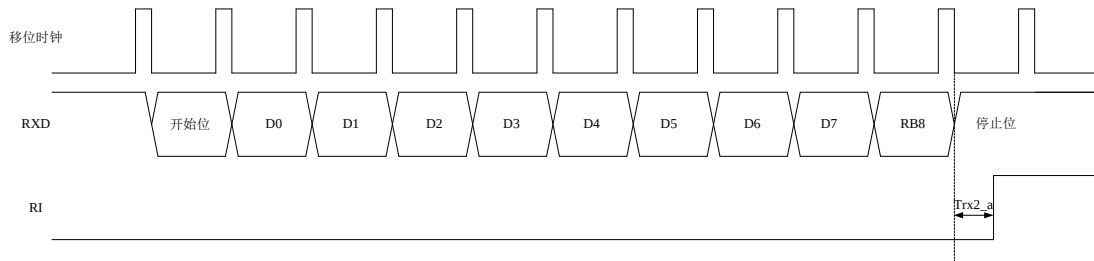


图22 UART 模式 2 接收数据波形

模式 3:

模式 3 的操作、数据结构同模式 2，它们的不同在于波特率的生成。模式 3 的数据传输时序同模式 2 相同，只是移位时钟的时钟源不同。进入模式 3 状态，需将 SCON1 寄存器的 SM0 标志位置 1，同时将 SM1 标志位置 1。

3.7.2 寄存器说明

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
3Eh	INTF3							URTIF	URRIF	0000u000
3Fh	INTE3							URTIE	URRIE	0000u000
77h	SCON1	SM0	SM1	SM2	REN	TB8	RB8	URTIOS	URRIOS	00000000
78h	SCON2	SMOD	TCLKS[1:0]		RCLKS[1:0]		URIOS[1:0]		UARTEN	00000000
79h	SBUF	SBUF[7:0]								00000000

表 53 SCON1 寄存器各位功能表

位地址	标识符	功能
7:6	SM0、SM1	串口通信工作方式选择寄存器 参见 3.73 章节 波特率
5	SM2	保留
4	REN	接收控制选择 1: 允许接收 0: 禁止接收
3	TB8	发送数据第 9 位
2	RB8	接收数据第 9 位
1	URTIOS	UART 接收发送口功能选择
		URIOS[1:0] URTIOS URRIOS UART 发送口
		00 0 0 PT1.5 做 RI 口，PT1.6 做 TI 口
		0 1 1 PT1.6 既做 RI 口，也做 TI 口，分时复用；PT1.5 做普通 IO 口
0	URRIOS	01 1 0 PT1.5 既做 RI 口，也做 TI 口，分时复用；PT1.6 做普通 IO 口
		1 1 1 PT1.5 做 TI 口，PT1.6 做 RI 口
		01 0 0 PT1.1 做 RI 口，PT1.2 做 TI 口
		0 1 1 PT1.2 既做 RI 口，也做 TI 口，分时复用；PT1.1 做普通 IO 口
0	URRIOS	1 0 1 PT1.1 既做 RI 口，也做 TI 口，分时复用；PT1.2 做普通 IO 口
		1 1 1 PT1.1 做 TI 口，PT1.2 做 RI 口

		10	0	0	PT3.2 做 RI 口, PT3.1 做 TI 口
			0	1	PT3.1 既做 RI 口, 也做 TI 口, 分时复用; PT3.2 做普通 IO 口
			1	0	PT3.2 既做 RI 口, 也做 TI 口, 分时复用; PT3.1 做普通 IO 口
			1	1	PT3.2 做 TI 口, PT3.1 做 RI 口

表 54 SCON2 寄存器各位功能表

位地址	标识符	功能
7	SMOD	波特率选择寄存器 参见 3.73 章节 波特率
6:5	TCLKS[1:0]	发送时钟选择位
		TCLKS[1:0] 发送时钟波
		00 选择定时器 4 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
		01 选择定时器 2 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
		10 选择定时器 3 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
		11 选择定时器 4 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
4:3	RCLKS[1:0]	接收时钟选择位
		TCLKS[1:0] 接收时钟波
		00 选择定时器 4 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
		01 选择定时器 2 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
		10 选择定时器 3 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
		11 选择定时器 4 溢出作为时钟, SMOD=1, 波特率为溢出率/16, SMOD=0, 波特率为溢出率/32
2:1	URIOS[1:0]	UART IO 组选择, 描述见上面
0	UARTEN	UART 模块使能 0: 不使能 UART 模块 1: 使能 UART 模块 UART 模块使能优先级比 SPI 模块使能低,

注: 当一个 IO 口分时复用为发送和接收脚时, 需通过软件配置 IO 口输入输出方向控制位来控制接收和发送。例如, PT1.5 做接收口时, 需将 PT1EN[5]置为 0, PT1.5 做发送口时, 需将 PT1EN[5]置为 1。

3.7.3 波特率

模式 1 和模式 3 的波特率公式为：

$$\text{波特率} = (2^{\text{SMOD}} / 32) * F_{\text{timer}} / (\text{TMXIN} + 1)$$

例如：选择模式 1，选用定时器 2 溢出率作为 UART 时钟，所以波特率为

$$\text{波特率} = (2^{\text{SMOD}} / 32) * F_{\text{timer}} / (\text{TM2IN} + 1)$$

表 55 波特率

波特率 (K)	Fsys=16MHz (SMOD=0)				
	实际波特率 (K)	偏差(%)	时钟 选择	定时器配置值 (TM4IN)	定时器预分频(16Mhz 时钟) {T4SEL[1:0],T4RATE[2:0]}
1.2	1.2005	0.042	MCK	1A0	01_000
2.4	2.3981	-0.08	MCK	0D0	01_000
4.8	4.807	0.16	MCK	068	01_000
9.6	9.6154	0.16	MCK	034	01_000
19.2	19.2308	0.16	MCK	01A	01_000

3.8 SPI接口

3.8.1 SPI接口介绍

CSU34F10 支持 SPI 接口，包括 master 模式和 slave 模式。

SPI 接口允许 MCU 与其他 SPI 接口设备进行全双工、同步、串行通信，它有两种模式：主模式和从模式。SPI 通信一般使用 4 线通信，即 SCK，MOSI，MISO， $\overline{SS}$ 这 4 根信号线，有时为了节省一个接口，也使用 3 线通信，即只包括 SCK，MOSI，MISO 这 3 根信号线。两个 SPI 设备通信时，一个作主机，一个作从机，SPI 接口间通信均由主机发起，主机发送时钟（SCK）和从机选择信号（ $\overline{SS}$ ）控制数据交换，主机和从机可同时发送和接收数据。

SPI 接口信号描述

MOSI：主机输出，从机输入

当 SPI 做为主机时该信号为主机数据输出，做为从机时为从机数据输入

MISO：主机输入，从机输出

当 SPI 做为主机时该信号为主机数据输入，做为从机时为从机数据输出

SCK：串行时钟

当 SPI 做为主机时该信号为串行时钟输出，做为从机时为串行时钟输入

$\overline{SS}$ ：从机选择

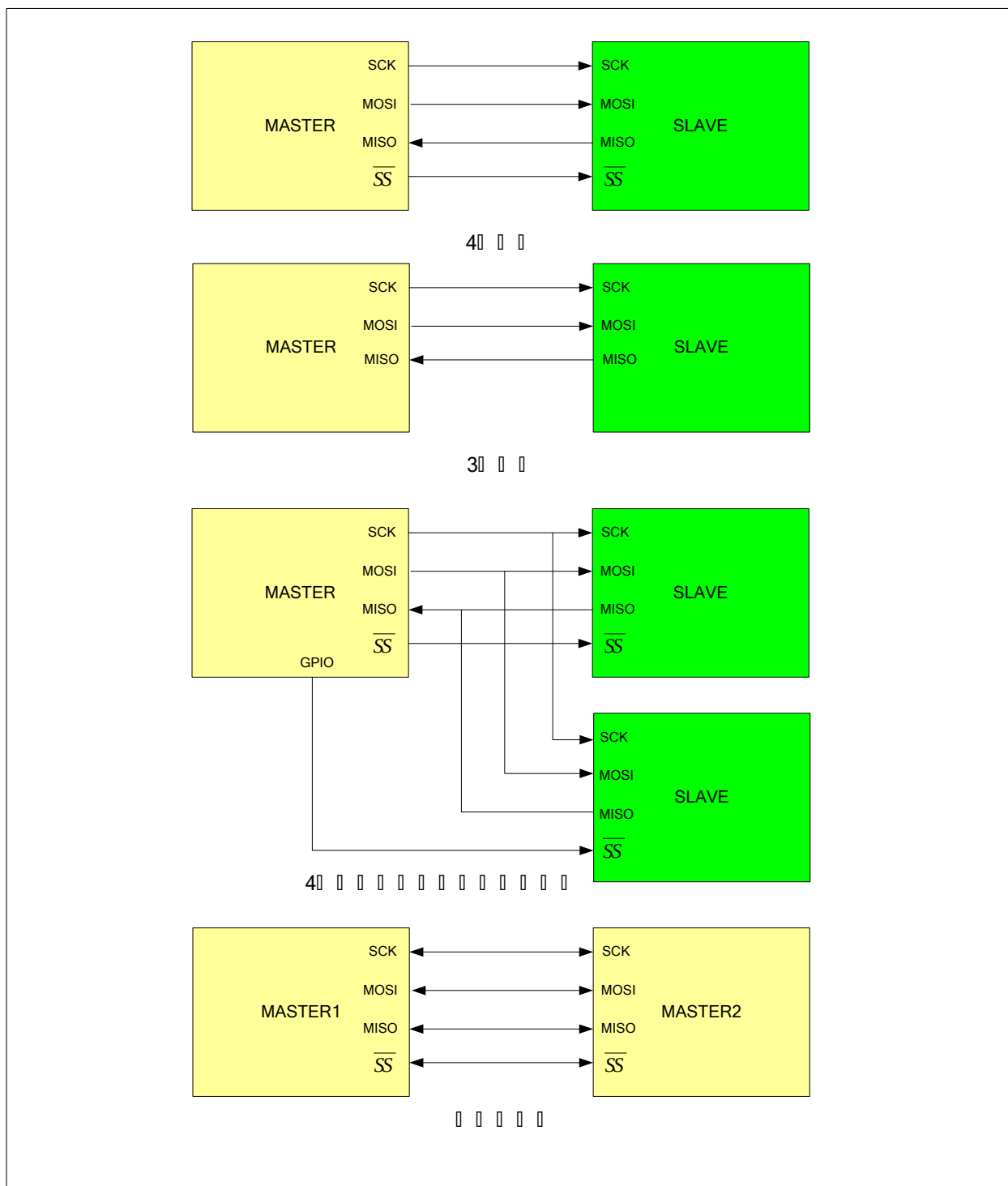
当 SPI 做为主机时该信号为从机选择输出，做为从机时为从机选择输入

SPI 串行时钟支持 4 种方式，即两种相位和两种极性组合

相位：0 表示在 SCK 奇数边沿（1,3,5,7……）采样数据，1 表示在 SCK 偶数边沿（2,4,6,8……）采样数据

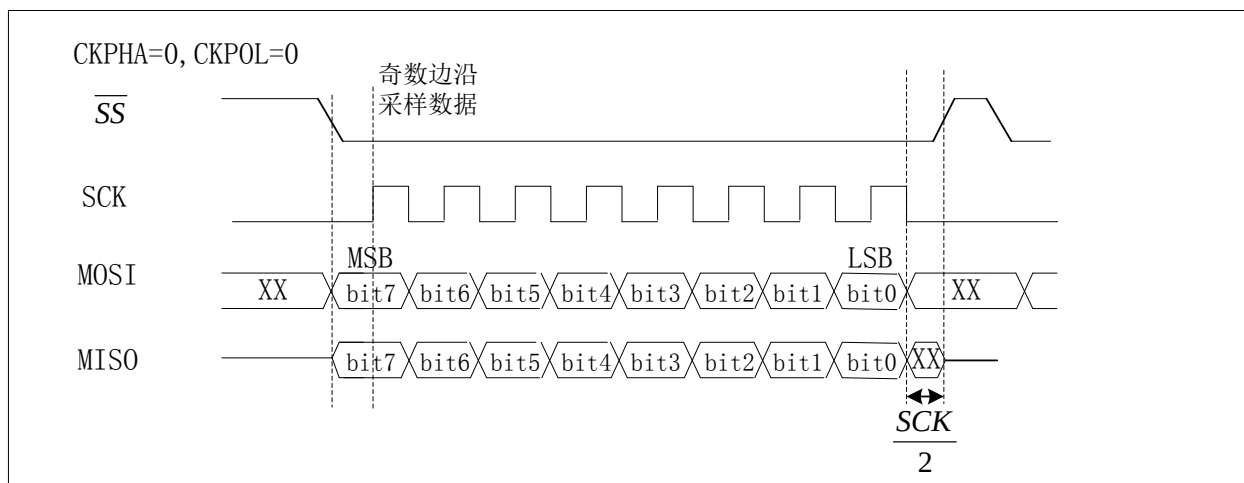
极性：0 表示 SCK 在总线空闲时总是低电平，1 表示 SCK 在总线空闲时为高电平

3.8.1.1 SPI连接方式

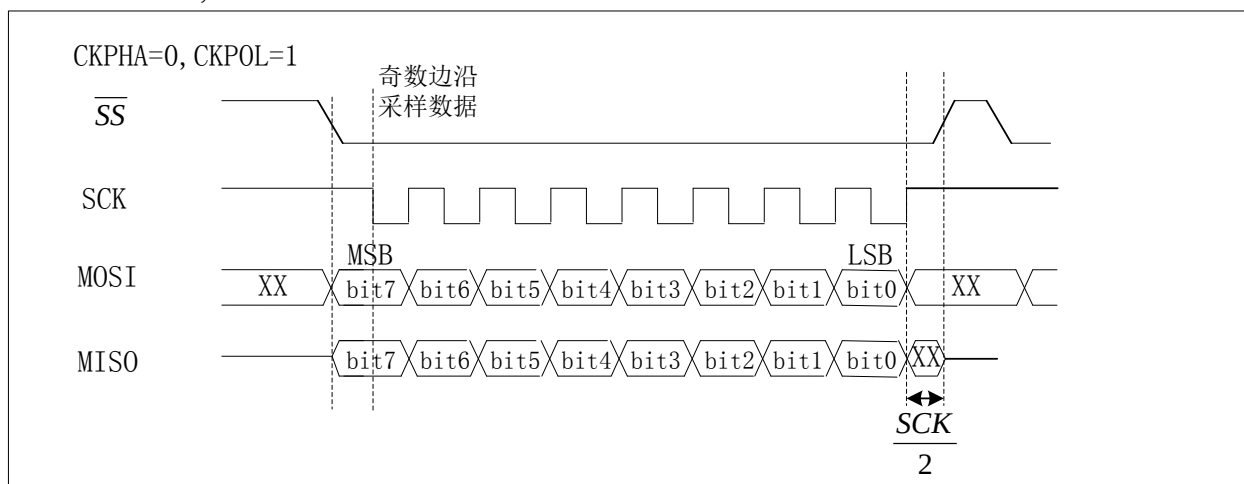


3.8.1.2 SPI时序图

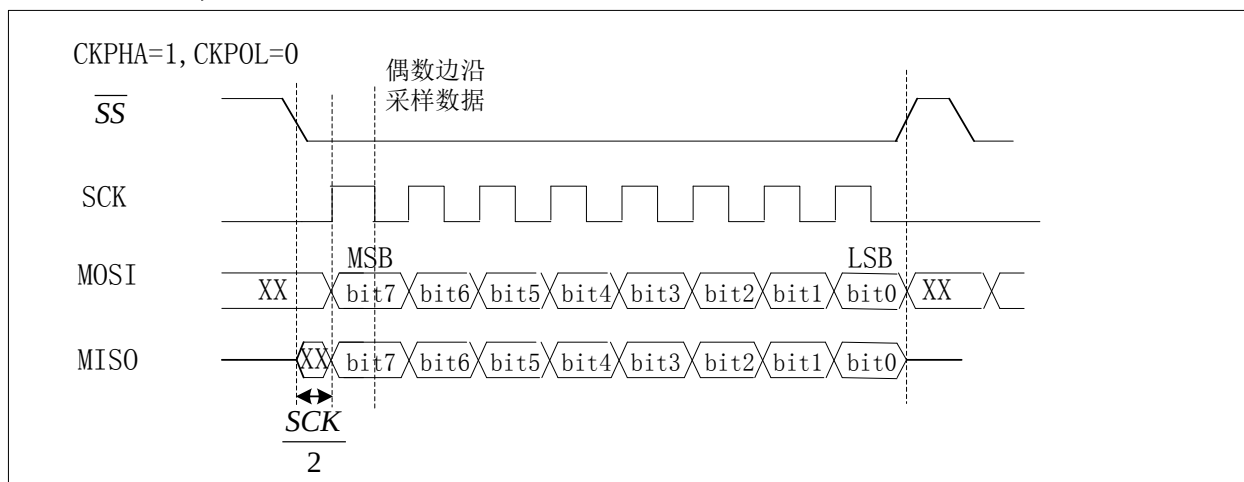
(1) CKPHA=0,CKPOL=0



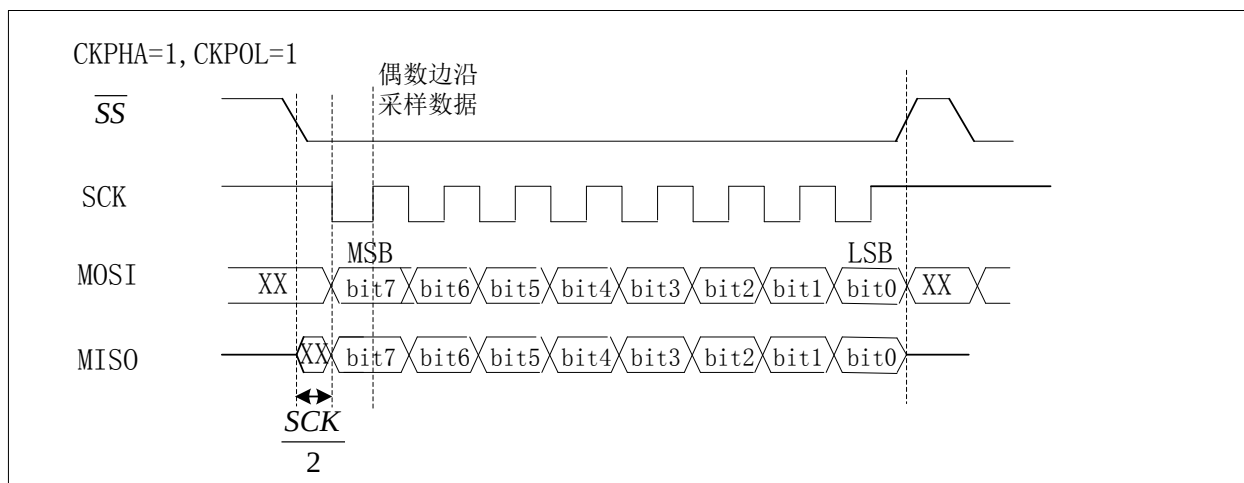
(2) CKPHA=0,CKPOL=1



(3) CKPHA=1,CKPOL=0



(4) CKPHA=0,CKPOL=1



3.8.2 SPI寄存器说明

表 56 SPI 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3eh	INTF3						SPIIF			0000u000
3fh	INTE3						SPIIE			0000u000
70h	SPICFG	SPIEN	MSTEN	CKPHA	CKPOL	MULMST	WIREMOD	SPIBSY	SPIRST	0000000u
71h	SPICN	SLVSEL	WCOL	MODCOL	SPIIOS					0000uuuu
72h	SPICKR	SCR[7:0]								00000011
73h	SPIDAT	SPIDAT[7:0]								00000000

表 57 INTF3 寄存器各位功能表

位地址	标识符	功能
2	SPIIF	SPI 中断标志位 1：发生 SPI 中断，必须软件清 0 0：没发生 SPI 中断

表 58 INTE3 寄存器各位功能表

位地址	标识符	功能
2	SPIIE	SPI 中断使能 1：使能 SPI 中断 0：关闭 SPI 中断

表 59 SPICFG 寄存器各位功能表

位地址	标识符	功能
7	SPIEN	SPI 使能 1：SPI 使能 0：SPI 关闭
6	MSTEN	模式选择 1：主模式

		0：从模式 多主机模式下当 SLVSEL (SPICN[7]) 为 1 时，MSTEN 自动清零
5	CKPHA	SPI 时钟相位控制 1：在 SCK 偶数边沿采样数据 0：在 SCK 奇数边沿采样数据
4	CKPOL	SPI 时钟极性控制 1：时钟 SCK 在总线空闲时是高电平 0：时钟 SCK 在总线空闲时是低电平
3	MULMST	多主机模式使能 1：使能多主机模式 0：禁止多主机模式 多主机模式时必须选择 4 线通信，即 WIREMOD 必须为 0
2	WIREMOD	3 线或 4 线 SPI 通信选择 0：SPI 为 4 线通信模式，即包括 SCK, MOSI, MISO, $\overline{SS}$ 1：SPI 为 3 线通信模式，即包括 SCK, MOSI, MISO
1	SPIBSY	SPI 总线通信指示 1：指示 SPI 总线正在使用 0：指示 SPI 总线闲置
0	SPIRST	SPI 总线复位 1：使能 SPI 复位 0：禁止 SPI 复位 写 1 时复位 SPI 计数寄存器，停止当前通信。复位完成后该位自动清零

表 60 SPICN 寄存器各位功能表

位地址	标识符	功能
7	SLVSEL	从模式标志（只读） 1： $\overline{SS}$ 输入为 0，从模式下 SPI 被选中 0： $\overline{SS}$ 输入为 1，从模式下 SPI 未被选中
6	WCOL	写冲突标志（软件清 0） 1：当仍有数据在发送时试图向发送数据寄存器中写数据，将会产生写冲突 0：没有写冲突
5	MODCOL	模式失效标志（软件清 0） 1：多主模式时产生了主模式冲突（ $\overline{SS}$ 为低，MSTEN: 1） 0：多主模式时未产生主模式冲突
4	SPIIOS	SPI IO 口选择信号 1：选择 PT3.0, PT3.1, PT3.2, PT3.3 作为 SPI 通信口 0：选择 PT1.4, PT1.5, PT1.6, PT1.7 作为 SPI 通信口， 对于同一组接口，SPI 模块使能优先级比 UART 模块使能高，当 SPIEN=1 时，PT1.4, PT1.5, PT1.6, PT1.7 作为 SPI 通信口；当 SPIEN=0 和 UARTEN=1，PT1.4, PT1.5 作为通信口；当 SPIEN=0 和 UARTEN=0，PT1.4, PT1.5, PT1.6, PT1.7 作为 IO 口。如果 UART 通信口选择 PT5.2 和 PT5.3 就可同时使用。
3:0	RESERVE	保留

多主机模式下，当 SPI 为主模式，若此时 $\overline{SS}$ 输入低电平，SLVSEL 位将置 1，这时将会产生模式失效，MSTEN 将自动清 0，SPI 变为从机，开始接收数据。接收数据完成时，MSTEN 不会自动置 1，需软件置 1

表 61 SPICKR 寄存器各位功能表

位地址	标识符	功能
7: 0	SCR [7:0]	SPI 时钟速率 SCR 决定了 SPI 通信中 SCK 的时钟频率。 $f_{sck} = \frac{CPUCLK}{SCR[7:0]+1}$ 其中， $3 \leq SCR[7:0] \leq 255$

表 62 SPIDAT 寄存器各位功能表

位地址	标识符	功能
7: 0	SPIDAT [7:0]	SPI 发送和接收的数据

3.8.3 SPI操作说明：

1、主模式

- 1) 设置 SPICKR，指定 SPI 时钟速率。
- 2) 设置 SPICFG，对 SPI 进行配置：选择主模式，相位，极性，选择 3 线或者 4 线模式。
- 3) 使能 SPI 中断：SPIIE 与 GIE。
- 4) 使能 SPI；
- 5) 向 SPI 数据寄存器 SPIDAT 中写入数据，SPI 模块就会自动开始发送和接收数据，发送或接收完成时将产生中断。响应中断并清中断标志位，若需要接收数据则读取 SPIDAT 寄存器中的数据，如果 SPIDAT 中的数据没有及时读取，将会被主机写入的数据覆盖；读数完成后继续写入数据，开始下一个数据传输。
如果 SPI 正在进行数据传输，此时向数据寄存器写入数据会产生写冲突，WCOL 位置 1，因此在写入数据后最好检查一下写冲突位，如果产生写冲突，则表示该数据未能发送出去，软件清写冲突位 WCOL，等待此次通信完成后再重新写入。
- 6) 所有数据全部传输完成时，取消对从机的选择，关闭中断使能和 SPI 使能。

主机收发数据程序示例：

```
MOVLW 08H
MOVWF SPICKR ;SPI 通信波特率设置
MOVLW 50H
MOVWF SPICFG ;SPI 设置，主模式，相位为 0，极性为 1，不使能多主模式，4 线模式
BCF INTF3, 2 ;清 SPI 中断标志位
BSF INTE3, 2 ;使能 SPI 中断
BSF INTE, GIE ;打开中断总使能
BSF SPICFG, 7 ;使能 SPI
MOVLW a5h
MOVWF SPIDAT ;发送数据 a5h，开始一次通信
..... ;SPI 中断处理
MOVFW SPIDAT ;SPI 接收数据处理
```

2、从模式

- 1) 设置 SPICFG，对 SPI 进行配置：选择从模式，相位，极性，选择 3 线或者 4 线模式。
- 2) 使能 SPI 中断：SPIIE 与 GIE。
- 3) 使能 SPI。
- 4) 若有数据要发送，则写入数据到 SPIDAT 寄存器，检查 WCOL，即写冲突位，如果产生写冲突，则写入数据无效，需等下次通信重新发送，清 WCOL，等待下次通信；若没有写冲突，则发送成功。若没有数据需要发送，则等待 SPI 接收中断，产生 SPI 接收中断时读取数据寄存器数据，接收数据。若还有数据要发送时，将数据写入 SPIDAT 寄存器中，继续以上操作，直到数据传输完成。若 SLVSEL 位为 1 则表示从机被选中，可以开始与主机通信。
- 5) 数据传输完成时，主机将取消对从机的选择，关闭 SPI 中断使能和 SPI 使能。

从机收发数据程序示例：

```

MOVLW 10H
MOVWF SPICFG    ;SPI 设置，从模式，相位为 0，极性为 1，不使能多主模式，4 线模式
BCF  INTF3, 2    ;清 SPI 中断标志位
BSF  INTE3, 2    ;使能 SPI 中断
BSF  INTE, GIE   ;打开中断总使能
BSF  SPICFG, 7   ;使能 SPI
MOVLW a5h
MOVWF SPIDAT     ;将 a5h 写入数据寄存器，等待主机开始一次通信（若无数据发送则可不
                  ;写数据到 SPIDAT, 直接等待 SPI 中断）
.....          ;SPI 中断处理
MOVFW SPIDAT     ;SPI 接收数据处理

```

3、多主机模式

- 1) 设置 SPICKR，指定 SPI 时钟速率。
- 2) 设置 SPICFG，对 SPI 进行配置：使能多主模式，相位，极性，选择 4 线模式。
- 3) 使能 SPI 中断：SPIIE 与 GIE。
- 4) 使能 SPI，使能主模式；
- 5) 向 SPI 数据寄存器 SPIDAT 中写入数据，SPI 模块就会自动开始发送和接收数据，发送或接收完成时将产生中断。响应中断并清中断标志位，若需要接收数据则读取 SPIDAT 寄存器中的数据，如果 SPIDAT 中的数据没有及时读取，将会被主机写入的数据覆盖；读数完成后继续写入数据，开始下一个数据传输。
向 SPI 数据寄存器写入数据后，检查模式失效位 MODCOL，如果该位为 1，则表示产生了模式失效，此时 SPI 为从模式，再检查写冲突标志位，若产生写冲突，则数据未发送出去；若没有产生写冲突，则数据在从模式下发送出去了。若没有产生模式失效，则表示数据在主模式下发送出去了。
- 6) 所有数据全部传输完成时，取消对从机的选择，关闭中断使能和 SPI 使能。

在一次通信中，配置好 SPI 相位，极性，时钟速率等后就不能改变。且主机与从机的相关配置要相同。若要与多个从机进行通信，可使用 IO 口模拟 $\overline{SS}$ 信号（即对 IO 进行拉低拉高操作，选择从机时拉低，取消选择时拉高）。多主机模式时， $\overline{SS}$ 口（PT1.4）上需要接上拉电阻，在空闲时 $\overline{SS}$ 能自动回到高电平。

3.9 模数转换器(ADC)

CSU34F10 模数转换模块共用 13 条外部通道 (AIN0~AIN12) 和 4 条特殊通道(内部 1/8VDD; 内部参考电压; GND; 运算放大器的输出 C0 作为 AD 的输入), 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(AIN0~AIN12), 然后把 SRADEN 置 1 使能 ADC, 之后把 SRADS 置 1, 启动 AD 转换。转换结束后, 系统自动将 SRADS 清 0, 并将转换结果存入寄存器 SRADL 和 SRADH 中。

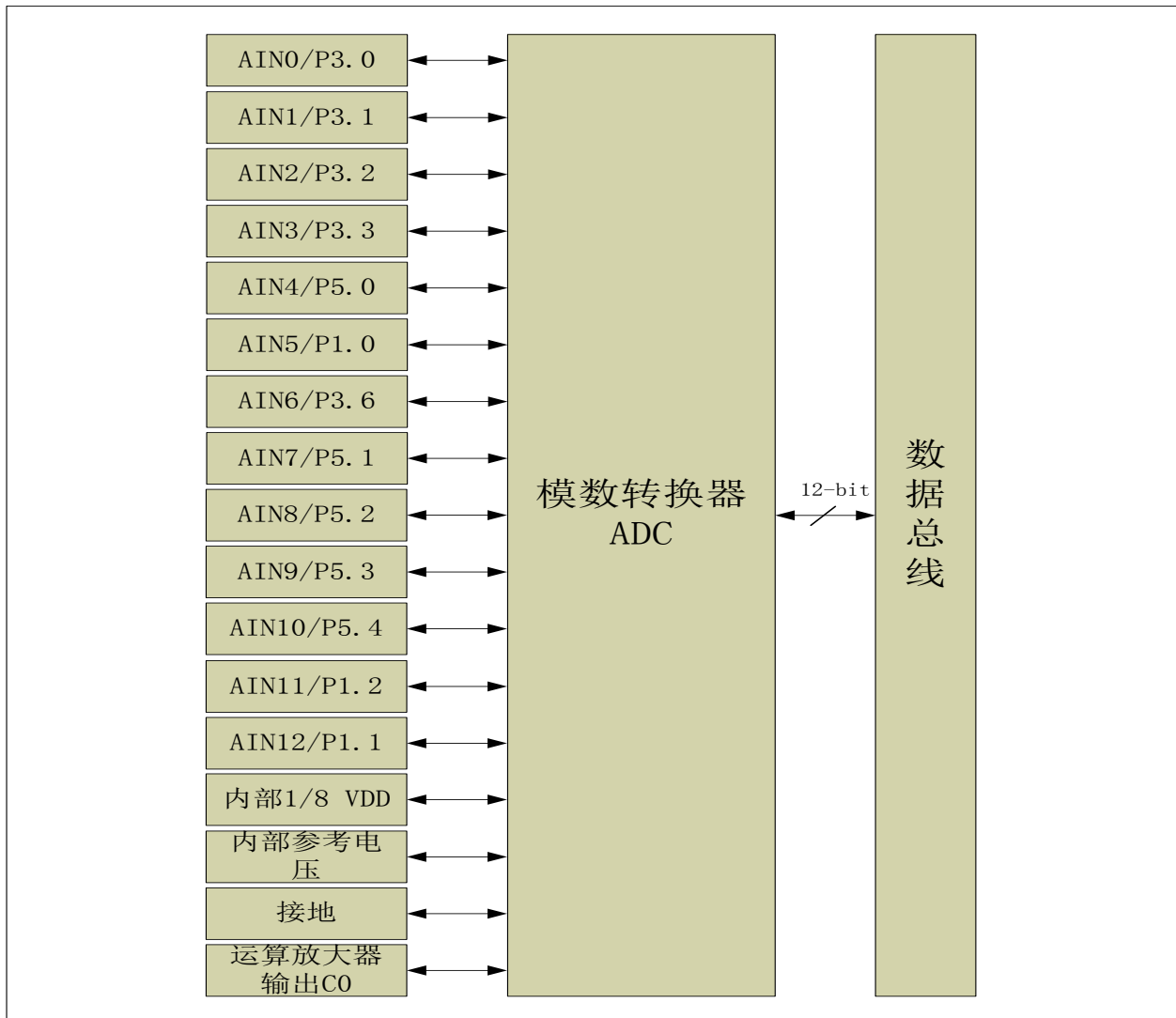


图23 模数转换器 ADC 功能框图

3.9.1 寄存器说明

表 63 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
4fh	SRADCON3		SRADSMOD[1:0]		SOURCE_SEL[1:0]		SRADSDLY[2:0]			uuu00000
50h	SRADCON0	SOURCE_MOD[1:0]		SRADACKS[1:0]			CMP0_R	SRADCKS[1:0]		uu00uu00
51h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]		00000000
52h	SRADCON2	CHS[4:0]						REF_SEL[1:0]		0000uuuu

54h	SRADL	SRAD[7:0]							00000000
55h	SRADH					SRAD[11:8]		uuuu0000	
56h	SROFTL	SROFT[7:0]							00000000
57h	SROFTH					SROFT[11:8]		uuuu0000	
74h	REFCFG		REFCFG[1:0]					00000000	

表 64 SRADCON0 寄存器各位功能表

位地址	标识符	功能	
7:6	SOURCE_MOD[1:0]	转换触发源触发模式选择:	
		SOURCE_MOD[1:0]	转换触发源触发模式
		2'b00	上升沿触发
		2'b01	下降沿触发
		2'b10	高电平触发
		2'b11	低电平触发
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 65 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）	
1:0	VREFS[1:0]	ADC 参考电源选择	
		VREFS[1:0]	AD 参考电压
		00	VDD
		01	PT3.0 外部参考电源输入
		10	内部参考电压
		11	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 66 SRADCON2 寄存器各位功能表

位地址	标识符	功能
7: 3	CHS[4:0]	ADC 输入通道选择位
		CHS[4:0]
		输入通道
		00000 AIN0 输入
		00010 AIN1 输入
		00100 AIN2 输入
		00110 AIN3 输入
		01000 AIN4 输入
		01010 AIN5 输入
		01100 AIN6 输入
		01110 AIN7 输入
		10000 内部 1/8VDD
		10010 内部参考电压
		10100 内部接地
		10110 运算放大器的输出 C0 作为 AD 的输入 (内部需要做滤波处理)
		00001 AIN8 输入
		00011 AIN9 输入
		00101 AIN10 输入
		00111 AIN11 输入
		01001 AIN12 输入
		11xxx 内短
		其他 无效
1: 0	REF_SEL[1:0]	内部参考电压 VREF 选择
		REF_SEL[1:0]
		内部参考电压
		00 1.4V
		01 2.0V
		10 3.0V
		11 4.0V

表 67 SRADCON3 寄存器各位功能表

位地址	标识符	功能
6: 5	SRADSMOD[1:0]	SRADS 启动位/状态位控制模式:
		SRADSMOD[1:0]
		SRADS 控制模式
		2'b00 通过直接配置 SRADS 寄存器开始 ADC 转换
		2'b01 通过 PWM 输出控制 ADC 开始转换
4:3	SOURCE_SEL[1:0]	2'b10 通过定时器溢出控制 ADC 开始转换
		2'b11 通过比较器输出控制 ADC 开始转换
		定时器/PWM 转换触发源选择: (SRADSMOD[1:0]选择 PWM 时, SOURCE_SEL[1:0]不能选择 2'b11)
		SOURCE_SEL[1:0]
		转换触发源
2: 0	SRADSDLY[2:0]	2'b00 TIMER2/PWM2
		2'b01 TIMER3/PWM3
		2'b10 TIMER4/PWM4
		2'b11 TIMER0
		SRADS 生效延时时间(CPUCLK)。 (当 SRADSMOD[1:0]! =2'b00 有效)
		SRADSDLY[2:0]
		延时时间
		3'b000 无延时
		3'b001 延时 1 cycle 置 SRADS 为 1'b1
		3'b010 延时 2 cycle 置 SRADS 为 1'b1
		3'b011 延时 3 cycle 置 SRADS 为 1'b1
		3'b100 延时 4 cycle 置 SRADS 为 1'b1
		3'b101 延时 5 cycle 置 SRADS 为 1'b1
		3'b110 延时 6 cycle 置 SRADS 为 1'b1
		3'b111 延时 7 cycle 置 SRADS 为 1'b1

表 68 REFCFG 寄存器各位功能表

位地址	标识符	功能
6	REFCFG[1]	PT31 输入作为参考电压选择: 1'b1: 使能 PT31 输入作为参考电压。 1'b0: 不使能 PT31 输入作为参考电压
5	REFCFG[0]	内部参考电压输出使能位。 1'b1: 使能 REF_SEL[1: 0]选择内部参考电压从 PT30 输出 1'b0: 不使能 REF_SEL[1: 0]选择内部参考电压从 PT30 输出
		REFCFG[0]
		VREFS[1:0]
		PT30
		1'b1 XX 输出 REF_SEL[1: 0] 所选内部参考电压
		1'b0 00 VDD
		01 PT3.0 外部参考电源输入
		10 内部参考电压
		11 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 69 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 70 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 71 SROFTL 寄存器各位功能表

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

表 72 SROFTH 寄存器各位功能表

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

表 73 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

3.9.2 转换时间

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

表 74 转换时间说明表⁽¹⁾

CLKDIV ⁽²⁾	CALIF	SRADCKS	SRADACKS	AD 转换时间 ⁽³⁾
00	0	10	00	$1 / ((16\text{MHz} / 2) / 4) \times (12 + 0 + 16) = 14\mu\text{s}$
			01	$1 / ((16\text{MHz} / 2) / 4) \times (12 + 0 + 8) = 10\mu\text{s}$

		11	00	$1 / ((16\text{MHz} / 2) / 8) \times (12 + 0 + 16) = 28\mu\text{s}$
			01	$1 / ((16\text{MHz} / 2) / 8) \times (12 + 0 + 8) = 20\mu\text{s}$
			10	$1 / ((16\text{MHz} / 2) / 8) \times (12 + 0 + 4) = 16\mu\text{s}$
	1	10	00	$1 / ((16\text{MHz} / 2) / 4) \times (12 + 1 + 16) = 14.5\mu\text{s}$
			01	$1 / ((16\text{MHz} / 2) / 4) \times (12 + 1 + 8) = 10.5\mu\text{s}$
		11	00	$1 / ((16\text{MHz} / 2) / 8) \times (12 + 1 + 16) = 29\mu\text{s}$
			01	$1 / ((16\text{MHz} / 2) / 8) \times (12 + 1 + 8) = 21\mu\text{s}$
			10	$1 / ((16\text{MHz} / 2) / 8) \times (12 + 1 + 4) = 17\mu\text{s}$
01	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	$1 / ((16\text{MHz} / 4) / 8) \times (12 + 1 + 2) = 30\mu\text{s}$
10	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	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 16) = 116\mu\text{s}$

11	0		01	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 8) = 84\mu\text{s}$
			10	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 4) = 68\mu\text{s}$
			11	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 2) = 60\mu\text{s}$
		00	00	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 0 + 16) = 28\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 0 + 8) = 20\mu\text{s}$
			10	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 0 + 4) = 16\mu\text{s}$
		01	00	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 16) = 56\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 8) = 40\mu\text{s}$
			10	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 4) = 32\mu\text{s}$
		10	11	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 0 + 2) = 28\mu\text{s}$
			00	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 16) = 112\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 8) = 80\mu\text{s}$
			10	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 4) = 64\mu\text{s}$
		11	11	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 0 + 2) = 48\mu\text{s}$
			00	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 16) = 224\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 8) = 160\mu\text{s}$
		11	10	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 4) = 128\mu\text{s}$
			11	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 0 + 2) = 112\mu\text{s}$
	1	00	00	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 1 + 16) = 29\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 1 + 8) = 21\mu\text{s}$
			10	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 1 + 4) = 17\mu\text{s}$
		01	00	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 16) = 58\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 8) = 42\mu\text{s}$
			10	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 4) = 34\mu\text{s}$
		10	11	$1 / ((16\text{MHz} / 16) / 2) \times (12 + 1 + 2) = 15\mu\text{s}$
			00	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 16) = 116\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 8) = 84\mu\text{s}$
			10	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 4) = 68\mu\text{s}$
		11	11	$1 / ((16\text{MHz} / 16) / 4) \times (12 + 1 + 2) = 60\mu\text{s}$
			00	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 16) = 232\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 8) = 168\mu\text{s}$
		11	10	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 4) = 136\mu\text{s}$
			11	$1 / ((16\text{MHz} / 16) / 8) \times (12 + 1 + 2) = 120\mu\text{s}$

(1) fosc=16MHz

(2) 代码选项

(3) AD 转换时间随 fosc 频率的改变而改变。

3.9.3 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：

- 通过 ADC 通道选择控制位 chs[4:0]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。
- 通过 ADC 通道选择控制位 chs[4:0]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 srofth
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
...
```

3.9.4 模拟比较器触发ADC启动转换

ADC 转换方式除了可以通过 SFR 直接配置启动外，还可以通过其他事件触发启动，如 PWM、比较器、定时器溢出等。

下面以比较器结果触发 ADC 转换为例说明其他事件触发 ADC 转换的方法，步骤如下：

- 1、配置比较器，使能比较器
- 2、使能 ADC 模块，配置 ADC 通道及参考电压等其他 ADC 常规配置
- 3、配置 SRADCON3 的 SRADSMOD[1:0]为 2'b11，通过比较器输出控制 ADC 开始转换
- 4、配置 SRADCON0 的 SOURCE_MOD[1:0]，选择触发边沿；配置 SRADCON3 的 SRADSDLY[2:0]，选择触发延迟时间。
- 5、等待比较器事件触发 ADC 转换
- 6、ADC 转换完成后，将产生 ADC 中断

3.10 比较器/运算放大器

3.10.1 比较器 0/运算放大器 0

CSU34F10 有一个模拟比较器/运算放大器，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，C0 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 CMPCON0 寄存器的 COS0[3:0]位。比较器只能查询中断标志位，无法进入中断。

3.10.2 寄存器说明

表 75 比较器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3eh	INTF3	CMP0IF								0000uuuu
6ah	CMPCON0	CMP0EN	COS0[3:0]					CMPIOS	CMPOUT ₀	00000u00
6bh	CMPCON1		CMPISC[1:0]		CMPOFST_SEL[4:0]					u0010000

表 76 CMPCON0 寄存器

位地址	标识符	功能																											
7	CMP0EN	比较器/运算放大器使能位 1: 使能 0: 禁止																											
6:3	COS0[3:0]	比较器/运算放大器选择 <table border="1"> <tr> <th>CMPIOS</th><th>COS0[3:0]</th><th>功能描述</th></tr> <tr> <td></td><td>4'b0000</td><td>比较器/运算放大器未使用，PT5.2、PT5.3、PT5.4 做为普通 IO</td></tr> <tr> <td></td><td>4'b 0001</td><td>用作运算放大器，PT5.2 做为运算放大器输出口，接运放正端接地，PT5.3 接运放负端，PT5.4 做为 IO</td></tr> <tr> <td></td><td>4'b 0010</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></td><td>4'b 0011</td><td>用作比较器，PT5.4 接比较器负端，比较器正端接 2VDD/5，PT5.2，PT5.3 做为普通 IO</td></tr> <tr> <td></td><td>4'b 0100</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></td><td>4'b 0101</td><td>用作比较器，PT5.4 接比较器负端，比较器正端接 VDD/2，PT5.2，PT5.3 做为普通 IO</td></tr> <tr> <td></td><td>4'b 0110</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></td><td>4'b 0111</td><td>用作比较器，PT5.3 接比较器负端，比较器正端接 VDD/2，PT5.2，PT5.4 做为普通 IO</td></tr> </table>	CMPIOS	COS0[3:0]	功能描述		4'b0000	比较器/运算放大器未使用，PT5.2、PT5.3、PT5.4 做为普通 IO		4'b 0001	用作运算放大器，PT5.2 做为运算放大器输出口，接运放正端接地，PT5.3 接运放负端，PT5.4 做为 IO		4'b 0010	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		4'b 0011	用作比较器，PT5.4 接比较器负端，比较器正端接 2VDD/5，PT5.2，PT5.3 做为普通 IO		4'b 0100	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		4'b 0101	用作比较器，PT5.4 接比较器负端，比较器正端接 VDD/2，PT5.2，PT5.3 做为普通 IO		4'b 0110	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		4'b 0111	用作比较器，PT5.3 接比较器负端，比较器正端接 VDD/2，PT5.2，PT5.4 做为普通 IO
CMPIOS	COS0[3:0]	功能描述																											
	4'b0000	比较器/运算放大器未使用，PT5.2、PT5.3、PT5.4 做为普通 IO																											
	4'b 0001	用作运算放大器，PT5.2 做为运算放大器输出口，接运放正端接地，PT5.3 接运放负端，PT5.4 做为 IO																											
	4'b 0010	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																											
	4'b 0011	用作比较器，PT5.4 接比较器负端，比较器正端接 2VDD/5，PT5.2，PT5.3 做为普通 IO																											
	4'b 0100	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																											
	4'b 0101	用作比较器，PT5.4 接比较器负端，比较器正端接 VDD/2，PT5.2，PT5.3 做为普通 IO																											
	4'b 0110	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																											
	4'b 0111	用作比较器，PT5.3 接比较器负端，比较器正端接 VDD/2，PT5.2，PT5.4 做为普通 IO																											

			4'b 1000	用作运算放大器，PT5.2 做为运算放大器输出，PT5.4 接运放正端，PT5.3 接运放负端，
			4'b 1001	用作运算放大器，PT5.2 做为运算放大器输出，接运放正端接 VDD/2，PT5.3 接运放负端，PT5.4 做为 IO
			4'b 1010	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
			4'b 1011	用作比较器，PT5.4 接比较器负端，比较器正端接 2VDD/5，PT5.2 做为比较器输出，PT5.3 做为普通 IO
			4'b 1100	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
			4'b 1101	用作比较器，PT5.4 接比较器负端，比较器正端接 VDD/2，PT5.2 做为比较器输出，PT5.3 做为普通 IO
			4'b 1110	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，PT5.4 做为普通 IO
			4'b 1111	用作比较器，PT5.3 接比较器负端，比较器正端接 VDD/2，PT5.2 做为比较器输出，PT5.4 做为普通 IO
		1'b1	4'b0000	比较器/运算放大器未使用，PT3.3、PT3.6、PT1.0 做为普通 IO
			4'b 0001	用作运算放大器，PT3.3 做为运算放大器输出，接运放正端接地，PT3.6 接运放负端，PT1.0 做为 IO
			4'b 0010	用作比较器，PT3.6 接比较器负端，PT1.0 接比较器正端，PT3.3 做为普通 IO
			4'b 0011	用作比较器，PT1.0 接比较器负端，比较器正端接 2VDD/5，PT3.3，PT3.6 做为普通 IO
			4'b 0100	用作比较器，PT3.6 接比较器正端，PT1.0 接比较器负端，PT3.3 做为普通 IO
			4'b 0101	用作比较器，PT1.0 接比较器负端，比较器正端接 VDD/2，PT3.3，PT3.6 做为普通 IO
			4'b 0110	用作比较器，PT3.6 接比较器正端，VREF/2 接比较器负端，PT1.0、PT3.3 做为普通 IO
			4'b 0111	用作比较器，PT3.6 接比较器负端，比较器正端接 VDD/2，PT3.3，PT1.0 做为普通 IO
			4'b 1000	用作运算放大器，PT3.3 做为运算放大器输出，PT1.0 接运放正端，PT3.6 接运放负端，

			4'b 1001	用作运算放大器，PT3.3 做为运算放大器输出，接运放正端接 VDD/2，PT3.6 接运放负端，PT1.0 做为 IO
			4'b 1010	用作比较器，PT3.6 接比较器负端，PT1.0 接比较器正端，PT3.3 做为比较器输出
			4'b 1011	用作比较器，PT1.0 接比较器负端，比较器正端接 2VDD/5，PT3.3 做为比较器输出，PT3.6 做为普通 IO
			4'b 1100	用作比较器，PT3.6 接比较器正端，PT1.0 接比较器负端，PT3.3 做为比较器输出
			4'b 1101	用作比较器，PT1.0 接比较器负端，比较器正端接 VDD/2，PT3.3 做为比较器输出，PT3.6 做为普通 IO
			4'b 1110	用作比较器，PT3.6 接比较器正端，VREF/2 接比较器负端，PT3.3 做为比较器输出，PT1.0 做为普通 IO
			4'b 1111	用作比较器，PT3.6 接比较器负端，比较器正端接 VDD/2，PT3.3 做为比较器输出，PT1.0 做为普通 IO
1	CMPIOS	比较器输出 IO 组选择，介绍见上面		
0	CMP0OUT	比较器的比较结果		

表 77 CMPCON1 寄存器

位地址	标识符	功能	
6: 5	CMPISC[1:0]	比较器/运算放大器内短	
		CMPISC[1:0]	内短说明
		2'b00	禁止比较器/运算放大器内部短接
		2'b01	比较器/运算放大器内短接地
		2'b10	比较器/运算放大器内短接输入正端
		2'b11	比较器/运算放大器内短接输入负端
4:0	CMPOFST_SEL[4:0]	比较器/运算放大器失调电压控制选择	
		CMPOFT_SEL[4:0]	失调电压
		5'b0_1111	使失调电压往负方向偏移 7.5mV
		5'b0_1110	使失调电压往负方向偏移 7mV
		...	...
		5'b0_0000	默认失调电压
		5'b1_0000	默认失调电压
		5'b1_0001	使失调电压往正方向偏 0.5mV
		...	...
		5'b1_1110	使失调电压往正方向偏移 7mV
		5'b1_1111	使失调电压往正方向偏移 7.5mV

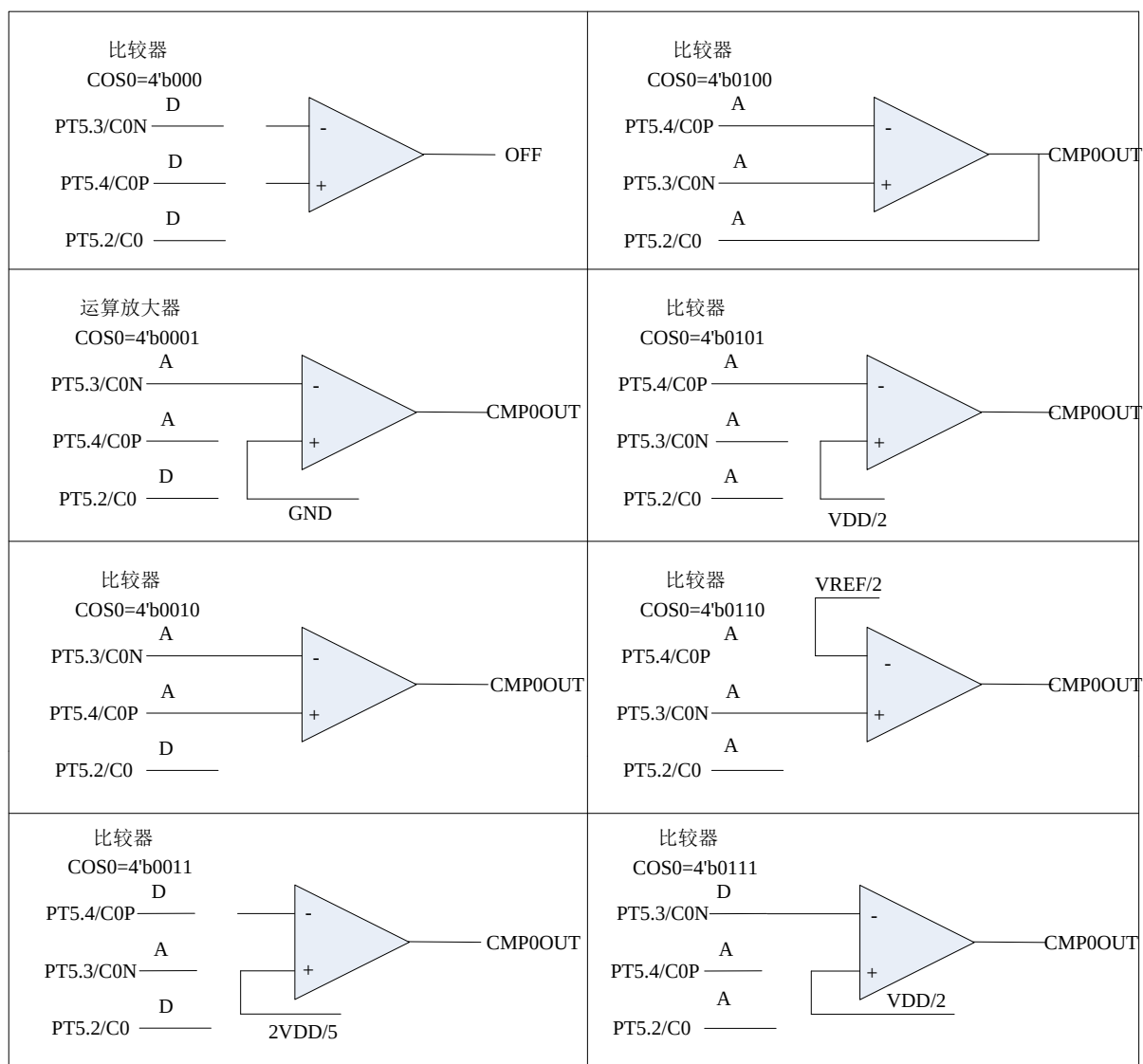
表 78 比较器/运算放大器的寄存器各位功能表

PT5.2	COS0[3]	COS0[2]	COS0[1]	COS0[0]	cmp0en	pt5en[2]	rst_n_pad	pt5od[2]
x(GPIO)	0	0	x	x	x	x	x	x
Z	0	1	0	0	0	x	x	x
Ao	0	1	0	0	1	x	x	x
Z	0	1	0	1	x	0	x	x
Z	0	1	0	1	x	x	0	x
0	0	1	0	1	0	1	1	x
C0 (1)	0	1	0	1	1	1	1	1
C0	0	1	0	1	1	1	1	0
Z	0	1	1	0	x	0	x	x
Z	0	1	1	0	x	x	0	x
0		1	1	0	0	1	1	x
C0 (1)	0	1	1	0	1	1	1	1
C0	0	1	1	0	1	1	1	0
Z	0	1	1	1	x	0	x	x
Z	0	1	1	1	x	x	0	x
0	0	1	1	1	0	1	1	x
C0 (1)	0	1	1	1	1	1	1	1
C0	0	1	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；C0 表示比较器输出；Z 表示高阻

图24 比较器及运算放大器的使用示意图



3.10.3 比较器参考电压

用户可以使用内部参考电压 VREF 作为比较器的一个输入，参考电压值可以为 VDD、内部 1.4v 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。

3.10.4 比较器失调电压校正

不同芯片由于离散性的原因，失调电压可能有正有负。

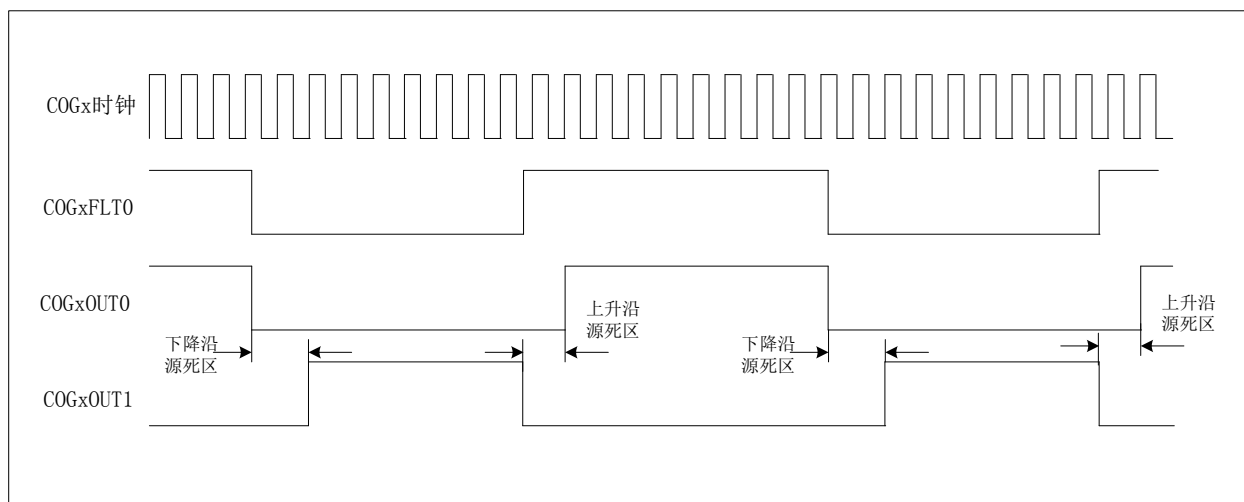
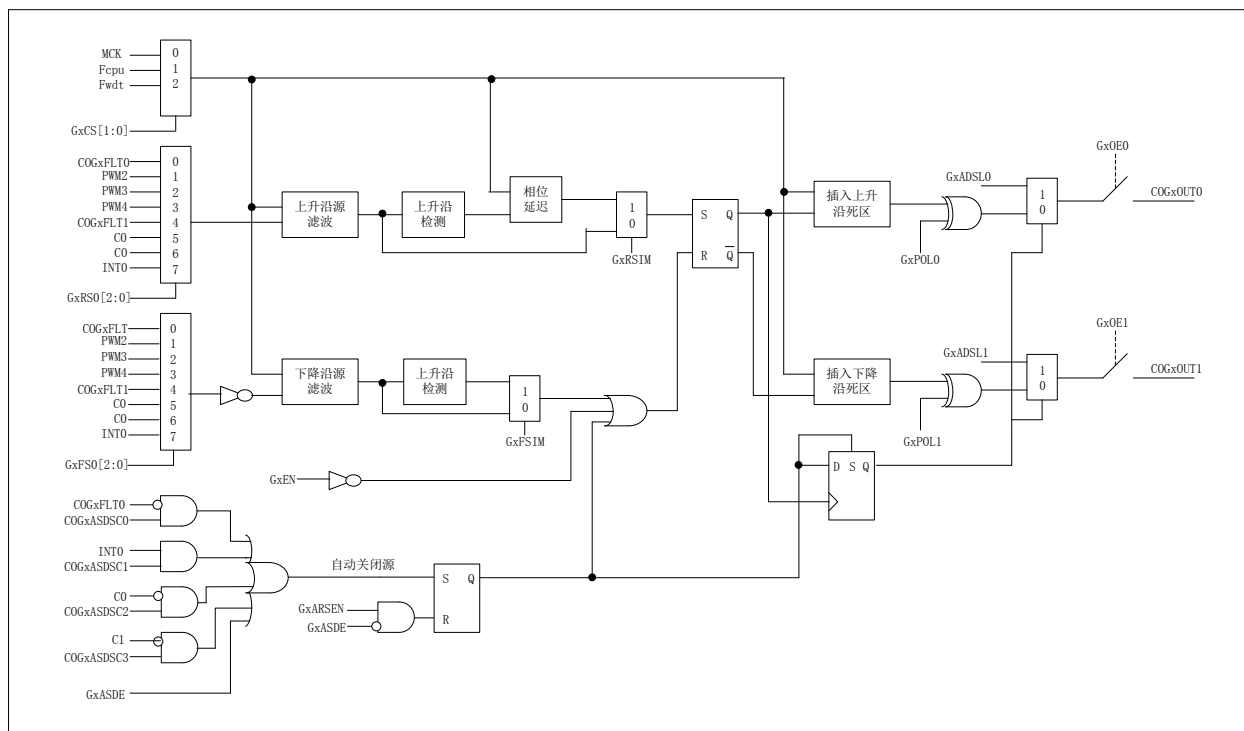
校正失调电压的方法及步骤：

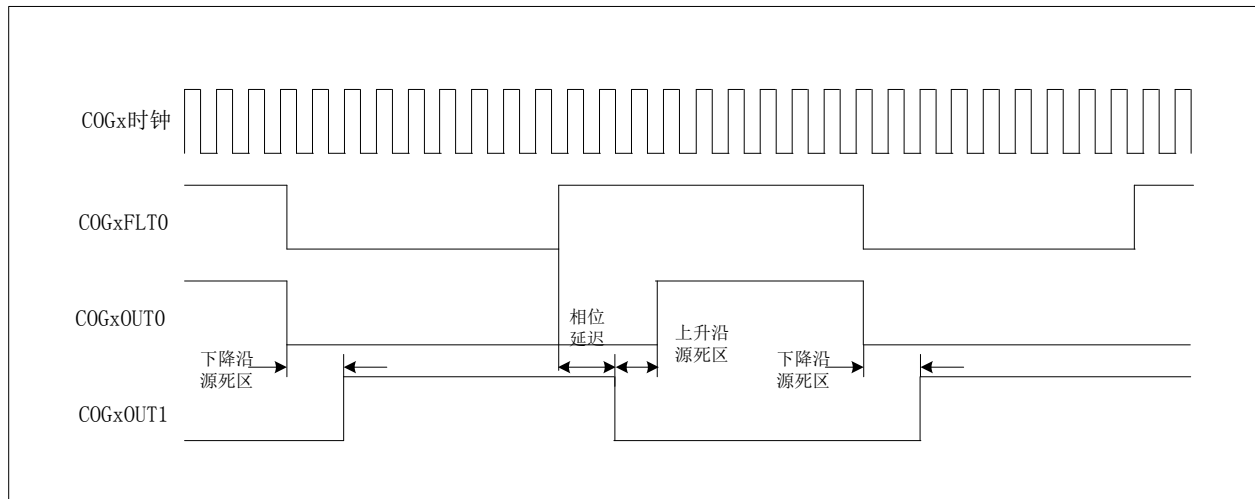
1. 比较器使能后，通过将 CMPCON1 寄存器中 CMPISC 位置 1 使比较器输入内部短接。
2. 配置 CMPCON1 寄存器 CMPOFST_SEL[4:0]值为 5'b0,查看 COMPOUT 值。
3. 若 CMPOUT 值从 0 变到 1，则记录此时 COMPOUT 值，并跳到步骤 5。
4. 若 CMPOUT 值保持为 0，则将 CMPOFST_SEL[4:0]值+5'b1，继续配置到 CMPOFST_SEL[4:0]，查看 COMPOUT 值。并跳转执行步骤 3。
5. 配置 CMPOFST_SEL[4:0]值为 5'b1_1111,查看 COMPOUT 值。
6. 若 CMPOUT 值从 1 变到 0，则记录此时 COMPOUT 值，并跳到步骤 8。
7. 若 CMPOUT 值保持为 1，则将 CMPOFST_SEL[4:0]值-5'b1，继续配置到 CMPOFST_SEL[4:0]，查看 COMPOUT 值。并跳转执行步骤 6。
8. 求第 3 步和第 6 步所记录的 COMOFST_SEL[4:0]平均值。
9. 将平均值配置到 COMOFST_SEL[4:0]寄存器，即完成了失调电压校正。

3.10.5 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。

3.11 互补输出发生器(COG)





3.11.1 COG0

3.11.1.1 寄存器说明

表 79 COG0 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电 复位值
3eh	INTF3				COG0IF					u00000000
3fh	INTE3				COG0IE					u00000000
58h	COG0CON0	GOEN			G0POL1	G0POL0		G0CS[1:0]		00000u00
59h	COG0CON1	G0FSIM	G0RSIM	G0FS[2:0]			G0RS[2:0]			00000000
5ah	COG0ASD	GOASDE	GOARSEN	GOASDL1	GOASDL0	G0ASDSC3	G0ASDSC2	G0ASDSC1	G0ASDSC0	00000000
5bh	COG0DB	G0DBF[3:0]				G0DBR[3:0]				00000000
5ch	COG0BLK	G0BLKF[3:0]				G0BLKR[3:0]				00000000
5dh	COG0PH					G0PH[3:0]				00000000

表 80 COG0CON0 寄存器各位功能表

位地址	标识符	功能	
7	G0EN	COG0 使能位 1: 使能 COG0 0: 禁止 COG0 使能 COG0 后, COG0OUT1、COG0OUT0 输出就有效了	
6: 5	RESERVE	保留	
4	G0POL1	COG0OUT1 输出极性控制位 1: 输出为取反极性 0: 输出为正常极性	
3	G0POL0	COG0OUT0 输出极性控制位 1: 输出为取反极性 0: 输出为正常极性	
1:0	G0CS[1:0]	COG0 时钟源选择位	
		G0CS[1:0]	COG0 时钟源
		00	MCK
		01	CPUCLK
		10	WDTCLK
		11	MCK

表 81 COG0CON1 寄存器各位功能表

位地址	标识符	功能	
7	G0FSIM	COG0 下降沿源输入模式位 1: 输入边沿敏感 0: 输入电平敏感	
6	G0RSIM	COG0 上升沿源输入模式位 1: 输入边沿敏感 0: 输入电平敏感	
5:3	G0FS[2:0]	COG0 下降沿源选择位	
		G0FS[2:0]	COG0 下降沿源
		000	COG0FLT0
		001	PWM2
		010	PWM3
		011	PWM4
		100	COG0FLT1
		101	C0
		110	C0
111	INT0		
2:0	G0RS[2:0]	COG0 上升沿源选择位	
		G0RS[2:0]	COG0 上升沿源
		000	COG0FLT0
		001	PWM2
		010	PWM3
		011	PWM4
		100	COG0FLT1
		101	C0
		110	C0
111	INT0		

表 82 COG0ASD 寄存器各位功能表

位地址	标识符	功能
7	G0ASDE	G0ASDE 自动关闭时间状态位 1: COG0 在自动关闭状态 0: COG0 不在自动关闭状态 发生自动关闭状态时, COG0IF 硬件置 1。
6	G0ARSEN	G0ASDE 自动重新开始使能位 1: COG0 自动重新开始使能 0: COG0 自动重新开始禁止
5	G0ASDL1	COG0OUT1 在自动关闭后的逻辑电平控制位 1: COG0OUT1 自动关闭后置 1 0: COG0OUT1 自动关闭后清 0
4	G0ASDL0	COG0OUT0 在自动关闭后的逻辑电平控制位 1: COG0OUT0 自动关闭后置 1 0: COG0OUT0 自动关闭后清 0
3	G0ASDSC3	COG0 自动关闭源使能位 3 1: 当 C0 引脚为低电平, COG0 自动关闭 0: C0 引脚不影响关闭
2	G0ASDSC2	COG0 自动关闭源使能位 2

		1: 当 COG0FLT1 引脚为低电平, COG0 自动关闭 0: COG0FLT1 引脚不影响关闭
1	G0ASDSC1	COG0 自动关闭源使能位 1 1: 当 PWM2 引脚为高电平, COG0 自动关闭 0: PWM2 引脚不影响关闭
0	G0ASDSC0	COG0 自动关闭源使能位 0 1: 当 COG0FLT0 引脚为低电平, COG0 自动关闭 0: COG0FLT0 引脚不影响关闭

表 83 COG0DB 寄存器各位功能表

位地址	标识符	功能
7:4	G0DBF[3:0]	COG0 下降沿死区计数值位 在输入下降沿事件后延迟 COG0 时钟的个数
3:0	G0DBR[3:0]	COG0 上升沿死区计数值位 在输入上升沿事件后延迟 COG0 时钟的个数

表 84 COG0BLK 寄存器各位功能表

位地址	标识符	功能
7:4	G0DBLK[3:0]	COG0 下降沿输入消隐控制位 在 COG0 时钟的个数内对输入进行滤波
3:0	G0DBLK[3:0]	COG0 上升沿输入消隐控制位 在 COG0 时钟的个数内对输入进行滤波

表 85 COG0PH 寄存器各位功能表

位地址	标识符	功能
3:0	G0PH[3:0]	COG0 上升沿事件相位延迟控制位 在 COG0 时钟的个数进行上升沿事件延迟

3.11.2 COG1

3.11.2.1 寄存器说明

表 86 COG1 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电 复位值
3eh	INTF3			COG1IF						0000u000
3fh	INTE3			COG1IE						0000u000
60h	COG1CON0	G1EN			G1POL1	G1POL0		G1CS[1:0]		00000u00
61h	COG1CON1	G1FSIM	G1RSIM	G1FS[2:0]			G1RS[2:0]			00000000
62h	COG1ASD	G1ASDE	G1ARSEN	G1ASDL1	G1ASDL0	G1ASDSHLT	G1ASDSC2	G1ASDSC1	G1ASDSFFLT	00000000
63h	COG1DB	G1DBF[3:0]				G1DBR[3:0]				00000000
64h	COG1BLK	G1BLKF[3:0]				G1BLKR[3:0]				00000000
65h	COG1PH					G1PH[3:0]				00000000

表 87 COG1CON0 寄存器各位功能表

位地址	标识符	功能	
7	G1EN	COG1 使能位 1: 使能 COG1 0: 禁止 COG1 使能 COG1 后, COG1OUT1、COG1OUT0 输出就有效了	
6: 5	RESERVE	保留	
4	G1POL1	COG1OUT1 输出极性控制位 1: 输出为取反极性 0: 输出为正常极性	
3	G1POL0	COG1OUT0 输出极性控制位 1: 输出为取反极性 0: 输出为正常极性	
1:0	G1CS[1:0]	COG1 时钟源选择位	
		G1CS[1:0]	COG1 时钟源
		00	MCK
		01	CPUCLK
		10	WDTCLK
		11	MCK

表 88 COG1CON1 寄存器各位功能表

位地址	标识符	功能	
7	G1FSIM	COG1 下降沿源输入模式位 1: 输入边沿敏感 0: 输入电平敏感	
6	G1RSIM	COG1 上升沿源输入模式位 1: 输入边沿敏感 0: 输入电平敏感	
5:3	G1FS[2:0]	COG1 下降沿源选择位	
		G1FS[2:0]	COG1 下降沿源
		000	COG1FLT0
		001	PWM2
		010	PWM3
		011	PWM4
		100	COG1FLT1
		101	C0
		110	C0
2:0	G1RS[2:0]	COG1 上升沿源选择位	
		G1RS[2:0]	COG1 上升沿源
		000	COG1FLT0
		001	PWM2
		010	PWM3
		011	PWM4
		100	COG1FLT1
		101	C0
		110	C0

		111	INT0	
--	--	-----	------	--

表 89 COG1ASD 寄存器各位功能表

位地址	标识符	功能
7	G1ASDE	G1ASDE 自动关闭时间状态位 1: COG1 在自动关闭状态 0: COG1 不在自动关闭状态 发生自动关闭状态时, COG1IF 硬件置 1。
6	G1ARSEN	G1ASDE 自动重新开始使能位 1: COG1 自动重新开始使能 0: COG1 自动重新开始禁止
5	G1ASDL1	COG1OUT1 在自动关闭后的逻辑电平控制位 1: COG1OUT1 自动关闭后置 1 0: COG1OUT1 自动关闭后清 0
4	G1ASDL0	COG1OUT0 在自动关闭后的逻辑电平控制位 1: COG1OUT0 自动关闭后置 1 0: COG1OUT0 自动关闭后清 0
3	G1ASDSC3	COG1 自动关闭源使能位 3 1: 当 C0 引脚为低电平, COG1 自动关闭 0: C0 引脚不影响关闭
2	G1ASDSC2	COG1 自动关闭源使能位 2 1: 当 COG1FLT1 引脚为低电平, COG1 自动关闭 0: COG1FLT1 引脚不影响关闭
1	G1ASDSC1	COG1 自动关闭源使能位 1 1: 当 PWM2 引脚为高电平, COG1 自动关闭 0: PWM2 引脚不影响关闭
0	G1ASDSC0	COG1 自动关闭源使能位 0 1: 当 COG1FLT0 引脚为低电平, COG1 自动关闭 0: COG1FLT0 引脚不影响关闭

表 90 COG1DB 寄存器各位功能表

位地址	标识符	功能
7:4	G1DBF[3:0]	COG1 下降沿死区计数值位 在输入下降沿事件后延迟 COG1 时钟的个数
3:0	G1DBR[3:0]	COG1 上升沿死区计数值位 在输入上升沿事件后延迟 COG1 时钟的个数

表 91 COG1BLK 寄存器各位功能表

位地址	标识符	功能
7:4	G1DBLK[3:0]	COG1 下降沿输入消隐控制位 在 COG1 时钟的个数内对输入进行滤波
3:0	G1DBLK[3:0]	COG1 上升沿输入消隐控制位 在 COG1 时钟的个数内对输入进行滤波

表 92 COG1PH 寄存器各位功能表

位地址	标识符	功能
3:0	G1PH[3:0]	COG1 上升沿事件相位延迟控制位 在 COG1 时钟的个数进行上升沿事件延迟

3.11.3 COG2

3.11.3.1 寄存器说明

表 93 COG2 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电 复位值
3eh	INTF3		COG2IF							00000000
3fh	INTE3		COG2IE							00000000
66h	COG2CON0	G2EN			G2POL1	G2POL0		G2CS[1:0]		00000u00
67h	COG2CON1	G2FSIM	G2RSIM	G2FS[2:0]			G2RS[2:0]			00000000
68h	COG2ASD	G2ASDE	G2ARSEN	G2ASDL1	G2ASDL0	G2ASDSHLT	G2ASDSC2	G2ASDSC1	G2ASDSFLT	00000000
69h	COG2DB	G2DBF[3:0]				G2DBR[3:0]				00000000
6ch	COG2BLK	G2BLKF[3:0]				G2BLKR[3:0]				00000000
6dh	COG2PH					G2PH[3:0]				00000000

表 94 COG2CON0 寄存器各位功能表

位地址	标识符	功能
7	G2EN	COG2 使能位 1: 使能 COG2 0: 禁止 COG2 使能 COG1 后, COG1OUT1、COG1OUT0 输出就有效了
6:5	RESERVE	保留
4	G2POL1	COG2OUT1 输出极性控制位 1: 输出为取反极性 0: 输出为正常极性
3	G2POL0	COG2OUT0 输出极性控制位 1: 输出为取反极性 0: 输出为正常极性
1:0	G2CS[1:0]	COG2 时钟源选择位
		G2CS[1:0] COG2 时钟源
		00 MCK
		01 CPUCLK
		10 WDTCLK
		11 MCK

表 95 COG2CON1 寄存器各位功能表

位地址	标识符	功能
7	G2FSIM	COG2 下降沿源输入模式位 1: 输入边沿敏感 0: 输入电平敏感

6	G2RSIM	COG2 上升沿源输入模式位 1: 输入边沿敏感 0: 输入电平敏感	
5:3	G2FS[2:0]	COG2 下降沿源选择位	
		G2FS[2:0]	COG2 下降沿源
		000	COG2FLT0
		001	PWM2
		010	PWM3
		011	PWM4
		100	COG2FLT1
		101	C0
		110	C0
		111	INT0
2:0	G2RS[2:0]	COG2 上升沿源选择位	
		G2RS[2:0]	COG2 上升沿源
		000	COG2FLT0
		001	PWM2
		010	PWM3
		011	PWM4
		100	COG2FLT1
		101	C0
		110	C0
		111	INT0

表 96 COG2ASD 寄存器各位功能表

位地址	标识符	功能
7	G2ASDE	G2ASDE 自动关闭时间状态位 1: COG2 在自动关闭状态 0: COG2 不在自动关闭状态 发生自动关闭状态时, COG2IF 硬件置 1。
6	G2ARSEN	G2ASDE 自动重新开始使能位 1: COG2 自动重新开始使能 0: COG2 自动重新开始禁止
5	G2ASDL1	COG2OUT1 在自动关闭后的逻辑电平控制位 1: COG2OUT1 自动关闭后置 1 0: COG2OUT1 自动关闭后清 0
4	G2ASDL0	COG2OUT0 在自动关闭后的逻辑电平控制位 1: COG2OUT0 自动关闭后置 1 0: COG2OUT0 自动关闭后清 0
3	G2ASDSC3	COG2 自动关闭源使能位 3 1: 当 C0 引脚为低电平, COG2 自动关闭 0: C0 引脚不影响关闭
2	G2ASDSC2	COG2 自动关闭源使能位 2 1: 当 COG2FLT1 引脚为低电平, COG2 自动关闭 0: COG2FLT1 引脚不影响关闭
1	G2ASDSC1	COG2 自动关闭源使能位 1 1: 当 PWM2 引脚为高电平, COG2 自动关闭 0: PWM2 引脚不影响关闭
0	G2ASDSC0	COG2 自动关闭源使能位 0

		1: 当 COG2FLT0 引脚为低电平, COG2 自动关闭 0: COG2FLT0 引脚不影响关闭
--	--	--

表 97 COG2DB 寄存器各位功能表

位地址	标识符	功能
7:4	G2DBF[3:0]	COG2 下降沿死区计数值位 在输入下降沿事件后延迟 COG2 时钟的个数
3:0	G2DBR[3:0]	COG2 上升沿死区计数值位 在输入上升沿事件后延迟 COG2 时钟的个数

表 98 COG2BLK 寄存器各位功能表

位地址	标识符	功能
7:4	G2DBLK[3:0]	COG2 下降沿输入消隐控制位 在 COG2 时钟的个数内对输入进行滤波
3:0	G2DBLK[3:0]	COG2 上升沿输入消隐控制位 在 COG2 时钟的个数内对输入进行滤波

表 99 COG2PH 寄存器各位功能表

位地址	标识符	功能
3:0	G2PH[3:0]	COG2 上升沿事件相位延迟控制位 在 COG2 时钟的个数进行上升沿事件延迟

3.12 数据查表和EEPROM操作

3.12.1 数据查表

通过 MOV_P 指令可以实现对于用户程序存储器内的数据读取，用户程序存储器的地址范围为 0000H~FFFH。当内部高速晶振选择 2MHz，指令周期选择 125KHz 时，无法进行写操作和擦除操作，因为无法满足烧录时序要求。

表 100 数据查表寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
05h	WORK	工作寄存器								00000000
0Ah	EADRH					PAR[11:8]				uuuu0000
0Bh	EADRL	PAR[7:0]								00000000
0Ch	EDATH	EDATH[7:0]								00000000

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。

读或写操作都是基于一个字（16 bits）的。

执行 TBLP 操作前，必须先解锁写保护，解锁写保护需向 SEC 寄存器连续写入 C9h/3Ah/D3h，解锁写操作在 12 个指令周期内有效。12 个指令周期后执行 TBLP 操作无效。

执行读操作时，在地址寄存器输入相应的值，之后执行 MOV_P 指令，便可在相应的 FLASH 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 0AH
movwf EADRH ;给高字节地址赋值
movlw 00H
movwf EADRL ;给低字节地址赋值
movp      ;执行读操作
nop
...
```

注意：

- 只能对没有读保护区域执行 MOV_P 指令进行数据查表。对于已经读保护的区域使用 MOV_P 读出数据为全 0。
- 用户模式下的 MOV_P 读保护通过代码选项 SECTOR2 中 USR_R_SEC[7:0] 寄存器控制，当 USR_R_SEC 中某一位为 1'b0，则用 MOV_P 指令对该位所对存储区域进行数据查表结果为 16'h0；当 USR_R_SEC 中某一位为 1'b1，则 MOV_P 指令可以读出存储区域所存储的数据。
- USR_R_SEC[7:0] 与读保护地址对应关系见下表。

表 101 USR_R_SEC 与读保护地址空间关系列表

USR_R_SEC[x]	程序区读保护地址
x=0	NVR1(128byte)
x=1	NVR2(128byte)
x=4	000H~3FFH

x=5	400H~7FFH
x=6	800H~BFFH
x=7	C00H~FFFH

3.12.2 EEPROM操作

CSU34F10 中的 EEPROM 分为 2 个 NVR 区域，即 2 个 NVR 区域（SECTOR1~SECTOR2），每个 NVR 分别对应 128byte 地址空间；通过寄存器 EADRL 的 PAR[6]控制选中 SECTOR,其中 PAR[6]为 1'b0/1'b1 分别对应于选中 NVR SECTOR1/SECTOR2。使用 WDT 时钟或者外部低速时钟作为系统时钟时，无法进行 TBLP 操作，当指令周期小于 250KHz 时，也无法进行 TBLP 操作。

表 102 ISP 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06h	INTF									u00000000
07h	INTE									00000000
5eh	SEC	SECURITY[7:0]								00000000
5fh	ERASE_CON	ISPOF						NVR	ERASE_MODE	0uuuuu00

表 103 ERASE_CON 寄存器各位功能表

位地址	标识符	功能
7	ISPOF	读/写/擦除操作失败标志位。 1'b0: 读/写/擦除操作成功。 1'b1: 读/写/擦除操作失败 硬件置位，软件清零
1	NVR	4K16 主程序区与 EEPROM 区选择 1'b0 : 4K16 主程序区 1'b1 : EEPROM 区
0	ERASE_MODE	擦除模式 1'b0: 512byte ERASE 1'b1: 128byte ERASE EEPROM 区只能 128 byte 擦除，主程序只能 512 byte 擦除。

NVR 用于选择对 4K16 主程序区还是 EEPROM 区进行操作。主程序区与 EEPROM 区都支持读/写/擦除操作。写与擦除操作通过 TBLP 指令完成，TBLP FFH 代表进行擦除操作；TBLP XH（X 不等于 FFH）代表进行写操作。

3.12.2.1 烧录模式下NVR操作

在烧录模式下支持主程序区与 EEPROM 区的读/写/擦除操作。

3.12.2.2 用户模式下NVR操作

用户模式下支持 EEPROM 的读/写/擦除操作。

MOVP 查表方式读取程序区方式在前面已经描述，此处不再赘述。

程序区与 EEPROM 的擦除操作通过 TBLP FFH 指令完成；写操作通过 TBLP XH（X 不等于 FFH）指令完成。并且 EEPROM 写/擦除操作也有保护位 USR_W_SEC[7:0]。当写/擦除保护位为 1'b1 时，对应地址空间可正常进行写/擦除操作；当写/擦除保护位为 1'b0 时，对应地址空间禁止写/擦除操作。USR_W_SEC[7:0]与写/擦除保护地址对应关系见下表。

表 104 USR_W_SEC 与写/擦除保护地址空间列表

USR_W_SEC[x]	程序区读保护地址
x=0	NVR1(128byte)
x=1	NVR2(128byte)
x=4	000H~3FFH
x=5	400H~7FFH
x=6	800H~BFFH
x=7	C00H~FFFH

进行写/擦除操作时，还需先输入三个序列号，然后在 16 个指令周期内执行 TBLP 指令，操作才能成功。否则，写/擦除操作不成功。

对 EEPROM 区的写操作步骤如下：

1. 配置 NVR 寄存器为 1'b1。
2. 配置{EADRH, EADRL}写操作地址。且通过地址的 bit8~bit6 确定写操作的 NVR SECTOR。
3. 依次对 SEC 寄存器写入 C9H,3AH,D3H。
4. 配置{EDATH, WORK}写操作数据。
5. 执行指令 TBLP XH(X 不等于 FF)操作（必须在对 SEC 寄存器写入 C9H,3AH,D3H 后 12 个指令周期内）。
6. 如果 USR_W_SEC[7:0]对应 bit 位为 1'b1，则可以写成功；如果 USR_W_SEC[7:0]对应 bit 位为 1'b0，则写失败。

对程序区的写操作步骤与上述步骤一致，区别在于将 NVR 寄存器为 1'b0。

对 EEPROM 区的擦除操作步骤如下：

1. 配置 NVR 寄存器为 1'b1。
2. 配置{EADRH, EADRL}写操作地址。且通过地址的 bit8~bit6 确定写操作的 NVR SECTOR。
3. 配置 ERASE_MODE 寄存器选择擦除模式。
4. 依次对 SEC 寄存器写入 C9H,3AH,D3H。
5. 10ms 内指令 TBLP FFH 操作。
6. 如果 USR_W_SEC[7:0]对应 bit 位为 1'b1，则{EADRH, EADRL}地址对应 PAGE 允许被擦除；如果 USR_W_SEC[7:0]对应 bit 位为 1'b0，则{EADRH, EADRL}地址对应 PAGE 禁止被擦除。

步骤 1~3 无先后顺序。对程序的写操作步骤与上述步骤一致，区别在于将 NVR 寄存器为 1'b0。

注意：1.对于 EEPROM 的擦除操作只能通过 Sector erase 方式。

2.进行擦除操作时，建议提前关闭 WDT 复位功能。

3.擦除操作时如有中断发生，则会延时到擦除操作完成后响应中断。

3.13 烧录模块

烧写器的接口：

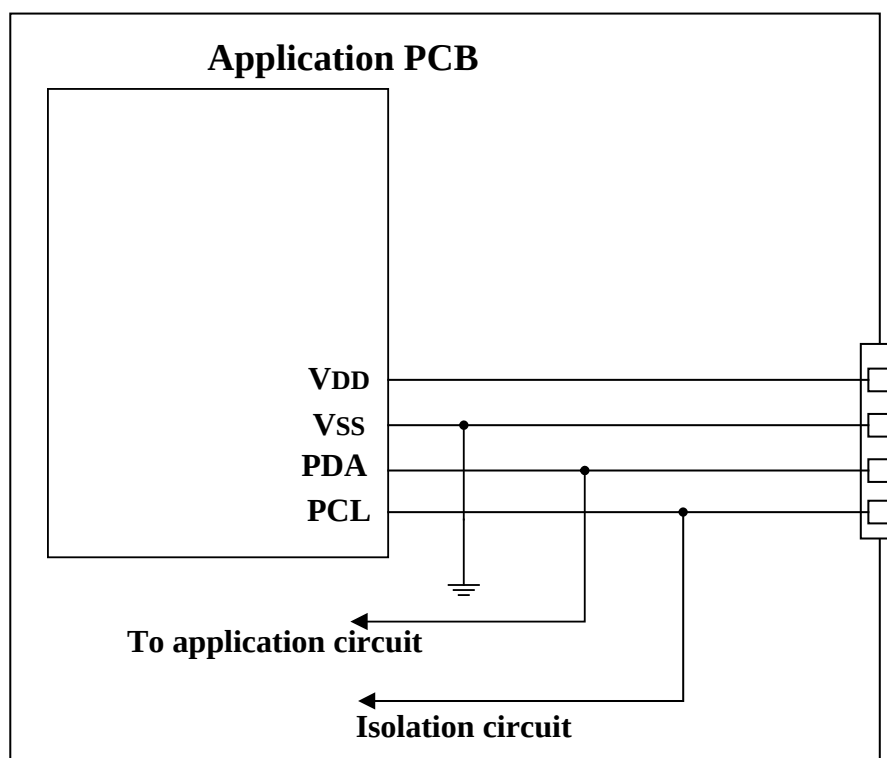


图25 烧写器接口图

表 105 烧录接口说明

端口名称	型式	说明
VDD	输入	电源正端
VSS	输入	电源负端
PDA	输入/输出	PT3[3]端口，数据信号
PCL	输入	PT3[6]端口，时钟信号

3.14 代码选项

1) OPTIION

标识符	功能
ICK_SEL[2:0]	内部晶振频率选择
	内部晶振频率
	16M
	8M
	4M
	2M
	32M
RST_DLY[1: 0]	上电复位延时时间选择位
	延时时间
	98ms
	49ms
	10ms
	1ms
INS_OP	指令周期配置选项 0: 指令周期只可通过代码选项配置 1: 指令周期可动态配置
WDTAEN	WDT 自动使能位。 0: WDT 使能自动使能功能。 1: WDT 不使能自动使能功能。 WDT 自动使能位置 1 后，上电复位完成后将自动打开 WDT 计数器，当计数到 FFH 时，WDT 溢出并复位芯片。
CLKDIV[1:0]	指令周期选择
	指令周期
	指令周期=2 个时钟周期
	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
LVD_SEL[2:0]	LVD 配置
	功能
	VDD 低于 1.8V 复位，LVD 复位检测系统关闭
	VDD 低于 1.8V 复位，LVD 复位系统； SYSCFG1 的 LVD20 作为 2.0V 的低电压检测器 STATUS 的 LVD24 作为 2.4V 的低电压检测器 SYSCFG1 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 STATUS 的 LVD42 作为 4.2V 的低电压检测器
	VDD 低于 2.0V 复位，LVD 复位系统； STATUS 的 LVD24 作为 2.4V 的低电压检测器 SYSCFG1 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 STATUS 的 LVD42 作为 4.2V 的低电压检测器
	VDD 低于 2.4V 复位，LVD 复位系统； SYSCFG1 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 STATUS 的 LVD42 作为 4.2V 的低电压检测器
	VDD 低于 3.0V 复位，LVD 复位系统；

	STATUS 的 LVD36 作为 3.6V 的低电压检测器 STATUS 的 LVD42 作为 4.2V 的低电压检测器	
	VDD 低于 3.6V 复位，LVD 复位系统； STATUS 的 LVD42 作为 4.2V 的低电压检测器	
RESET_PIN	复位引脚选择 1: PT1.3 作为复位引脚 0: PT1.3 作为普通 IO 口	
XTAL_PIN[1:0]	晶振引脚选择	
	晶振引脚	
	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	

2) PROG_R_SEC

标识符	功能
PROG_R_SEC[7:4]	对 4K16 程序区的烧录模式读保护位，每位对应 1K16 page 空间。 1'b1: 禁止该 page 读保护 1'b0: 使能该 page 读保护
PROG_R_SEC [1:0]	对应 SECTOR1/SECTOR2 区域烧录模式读保护位，每位对应一个 SECTOR。 1'b1: 禁止该 sector 读保护 1'b0: 使能该 sector 读保护

3) USR_R_SEC

标识符	功能
USR_R_SEC [7:4]	对 4K16 程序区的用户正常模式 MOVN 指令读保护位，每位对应 1K16 page 空间。 1'b1: 禁止该 page 读保护 1'b0: 使能该 page 读保护
USR_R_SEC [1:0]	对应 SECTOR1/SECTOR2 区的用户正常模式 MOVN 指令读保护位，每位对应一个 SECTOR。 1'b1: 禁止该 sector 读保护 1'b0: 使能该 sector 读保护

4) USR_W_SEC

标识符	功能
USR_W_SEC [7:4]	对 4K16 程序区的擦除/写保护位，每位对应 1K16 page 空间。 1'b1: 禁止该 page 擦除/写保护 1'b0: 使能该 page 擦除/写保护
USR_W_SEC [1:0]	对应开放给客户的两个 128byte 的 SECTOR 区域擦除/写保护位，每位对应一个 SECTOR。 1'b1: 禁止该 sector 擦除/写保护

	1'b0: 使能该 sector 擦除/写保护
--	-------------------------

4 MCU指令集

表 106 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	~
MOVP	Read eprom	3	~
MOVWF f	$[f] \leftarrow [W]$	1	~
NOP	No operation	1	~
POP	Pop W and Status	2	~
PUSH	Push W and Status	2	~
RETFIE	Pop PC and GIE =1	2	~
RETLW k	RETURN and W=k	2	~
RETURN	POP PC	2	~
RLF f,d	$[Destination < n+1 >] \leftarrow [f < n >]$	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	$[Destination] \leftarrow [f] - [W]$	1	C,DC,Z
SUBWFC f,d	$[Destination] \leftarrow [f] - [W] - 1 + C$	1	C,DC,Z
SWAPF f,d	swap f	1	~
TBLP ,f	erase f or write f		~
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 ~167h)

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 107 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<=167H 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<=167H 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<=167H 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<=167H 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<=167H 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<=167H 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<=167H 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<=FFFH
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<=167H
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<=167H 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 25H + 39H ----- 5EH + 06H ----- 64H

16

DECF	f 减 1
指令格式	DECF f, d 0<=f<=167H 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<=167H 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<=FFFH
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<=167H 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<=167H 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 (FLAG)=(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<=167H 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<=167H
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<=167H
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 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

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

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<=167H 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<=167H 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<=167H 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<=167H 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<=167H 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 ,f	将 FLASH 的{ EADRH ,EADRL}的地址所在的页擦除
指令格式	TBLP FFH
操作	擦除 FLASH({EADRH ,EADRL})所在的页
标志位	无
描述	FLASH({EADRH ,EADRL})所在的页存储单元变成 FFH
周期	与工作时钟和 ETM 的值有关 T=ETM*512*T(mcu_clk)
例子 1 TBLP ffh	在指令执行之前 flash memory: 17H= 00H {EADRH ,EADRL} = 17H 在指令执行之后 flash memory:

	17H = ffH
例子 2 TBLP f0h	在指令执行之前 flash memory: 17H= ffH {EADRH ,EADRL} = 17H {EDATH,WORK} = 5AH 在指令执行之后 flash memory: 17H = 5AH

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<=167H 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
极限温度	-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	5	5.5	V
		-40 °C ~+85 °C	2	5	5.5	V
FXT	外部晶振频率				16	MHz
EXT_CK	外部灌入时钟	25 °C, 代码选项 XTAL_PIN 选 ERC			16	MHz
		25 °C, 代码选项 XTAL_PIN 选外部低速晶振			50	KHz
		25 °C, 代码选项 XTAL_PIN 选外部高速晶振			4	MHz
Vpor	系统电源电压上升速率		0.07			V/ms
VIH1	数字输入高电平	PT1, PT3, PT5	0.7VDD			V
	复位输入高电平 (PT1.3 口)		0.8VDD			
VIL1	数字输入低电平	PT1, PT3, PT5			0.3VDD	V
	复位输入低电平 (PT1.3 口)				0.2VDD	
VIH2	数字输入高电平	PT1, PT3, PT5(普通 IO, VDD=5V)	2.0			V
	数字输入高电平	PT1, PT3, PT5(普通 IO, VDD=3V)	2.0			
	复位输入高电平 (PT1.3 口)		2.0			
	OSCI	PT1.1, PT1.2(做时钟口时)	2.5			
VIL2	数字输入低电平	PT1, PT3, PT5(普通 IO, VDD=5V)			0.8	V
	数字输入低电平	PT1, PT3, PT5(普通 IO, VDD=3V)			0.8	
	复位输入低				0.8	

	电平 (PT1.3口)					
	OSCI	PT1.1, PT1.2(做时钟口时)			1.0	
VIH3 (施密特输入关闭)	数字输入高电平	PT1, PT3, PT5	0.5VDD			V
VIL3 (施密特输入关闭)	数字输入低电平	PT1, PT3, PT5			0.5VDD	
I _{PU}	上拉电流	PT1, PT3, PT5; V _{in} = 0;		100		uA
I _{PD}	下拉电流	PT1, PT5; V _{in} = 0;		100		uA
I _{OH}	高电平输出电流	VOH=0.9VDD; VDD=5V (非大驱动口)		20		mA
		VOH=0.9VDD; VDD=3V (非大驱动口)		7		mA
		VOH=0.9VDD; VDD=5V (大驱动口, 根据配置输出相应驱动能力)		18		mA
		VOH=0.9VDD; VDD=3V (大驱动口, 根据配置输出相应驱动能力)		7		mA
I _{OL}	低电平输出电流	VOL=0.1VDD; VDD=5V (非大驱动口)		37		mA
		VOL=0.1VDD; VDD=3V (非大驱动口)		18		mA
		VOH=0.1VDD; VDD=5V (大驱动口, 根据配置输出相应驱动能力)		65		mA
		VOH=0.1VDD; VDD=3V (大驱动口, 根据配置输出相应驱动能力)		29		mA
LVD	复位电压/低电压检测电压	1.8V 上电/掉电复位点; -40~85 度		1.8		V
		2.0V 上电/掉电复位点; -40~85 度		2.2		
		2.4V 上电/掉电复位点; -40~85 度		2.6		
		3.0V 上电/掉电复位点; -40~85 度		3.1		
		3.6V 上电/掉电复位点; -40~85 度		3.9		
		4.2V 低电压检测点; -40~85 度		4.4		
I _{HRC}	内置 RC 时钟 32MHz	25°C, 5V	-1%	32.00	1%	MHz
		-40°C~85°C, 2.5V~5.5V	-5%	32.00	5%	
	内置 RC 时钟 16MHz	25°C, 5V	-1%	16.00	1%	MHz
		-40°C~85°C, 2.5V~5.5V	-4%	16.00	4%	
	内置 RC 时钟 8MHz	25°C, 5V	-1%	8.00	1%	MHz
		-40°C~85°C, 2.5V~5.5V	-4%	8.00	4%	
	内置 RC 时钟 4MHz	25°C, 5V	-1%	4.00	1%	MHz
		-40°C~85°C, 2.5V~5.5V	-4%	4.00	4%	
	内置 RC 时钟 2MHz	25°C, 5V	-1%	2.00	1%	MHz
		-40°C~85°C, 2.5V~5.5V	-4%	2.00	4%	
WDT	内置看门狗时钟	25°C, 5V	-10%	32.00	10%	KHz
		-40°C~85°C, 2.5V~5.5V	-20%	32.00	20%	KHz
T _{int0,1}	中断触发脉宽	25°C, 5V	T _{cpu}			ns
I _{DD1}	低功耗模式	低功耗模式 0、1 VDD=3V		540		uA

	电流	低功耗模式 2 VDD=3V, 关闭 WDT		1.2		uA
		低功耗模式 2 VDD=3V, 打开 WDT		2.5		uA
		低功耗模式 0、1 VDD=5V		600		uA
		低功耗模式 2 VDD=5V, 关闭 WDT		1.2		uA
		低功耗模式 2 VDD=5V, 打开 WDT		4.5		uA
		超低功耗模式		0.1		uA
IDD2	工作电流 IHRC=16MHz	内部振荡器关闭 (fcpu=fosc/4) fosc = 32768Hz,3V		0.6		mA
		内部振荡器关闭 (fcpu=fosc/4) fosc = 32768Hz,5V		0.7		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/2		1.5		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/4		1.2		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/8		1.0		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/16		0.95		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/2		1.65		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/4		1.35		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		1.1		mA
IDD5	复位或低电压复位电流	掉电复位或低电压复位			1	uA

5.3 ADC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	ADC 工作电压范围	25 °C	2.0	5	5.5	V
		-40 °C ~+85 °C	2.0	5	5.5	V
AIN0~ AIN13 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	3.5	10		uS

INL	积分非线性	SRADACKS[1:0]=01; SRADCKS[1:0]=01;				LSB
No missing code	无失码	VREFS[1:0]=01, 外部参考电压				Bits
		VREFS[1:0]=00, VDD 做为参考电压				Bits
		VREFS[1:0]=10, 内部参考电压				Bits
IVREF	内部参考电压 1.4V	REF_SEL[1:0]=00	-1%	1.40	1%	V
	内部参考电压 2.0V	REF_SEL[1:0]=01	-1%	2.00	1%	V
	内部参考电压 3.0V	REF_SEL[1:0]=10	-1%	3.00	1%	V
	内部参考电压 4.0V	REF_SEL[1:0]=11	-1%	4.00	1%	V
IVREF temp drift	内部参考电压温漂			50		ppm
Offset	ADC 失调电压					mV

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)

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	2.0	5	5.5	V
		-40 °C ~+85 °C	2.0	5	5.5	V
Comp Vos drift	比较器/运算放大器的失调电压温漂			5 *		uV
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		2.0V		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	Isourse (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 32MHz 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。

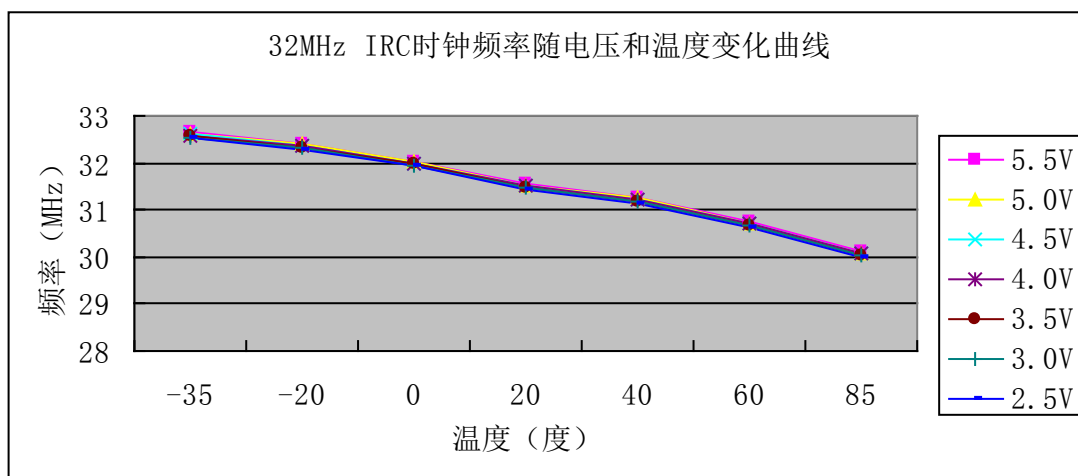


图26 RC 时钟频率的电压和温度特性

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。

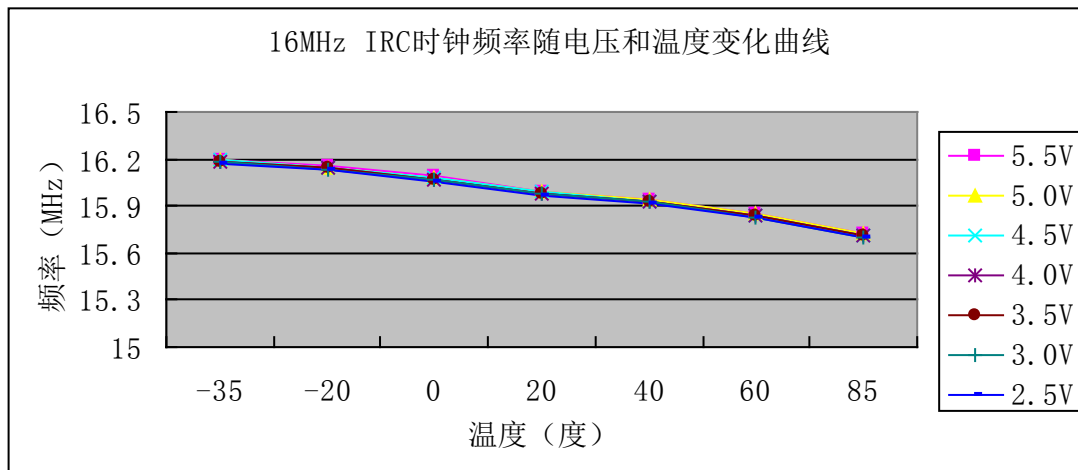


图27 RC 时钟频率的电压和温度特性

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。

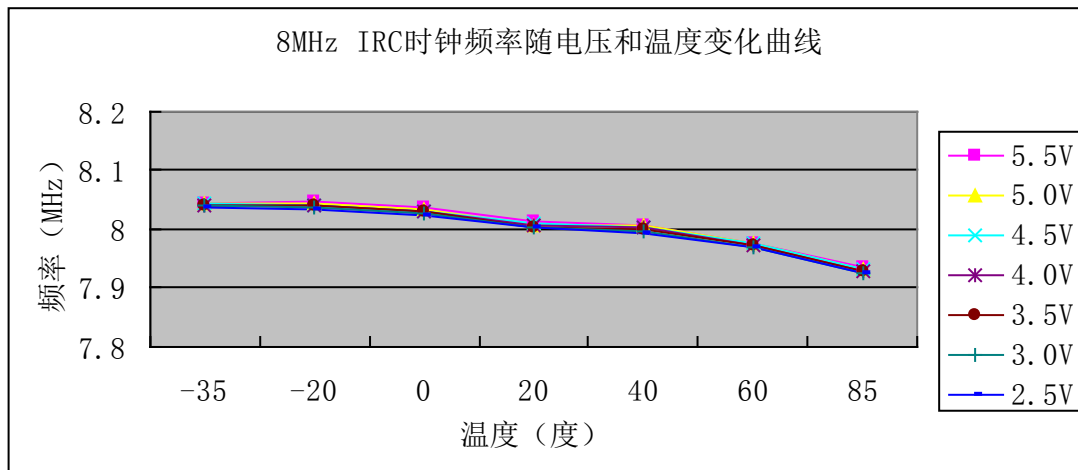


图28 RC 时钟频率的电压和温度特性

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。

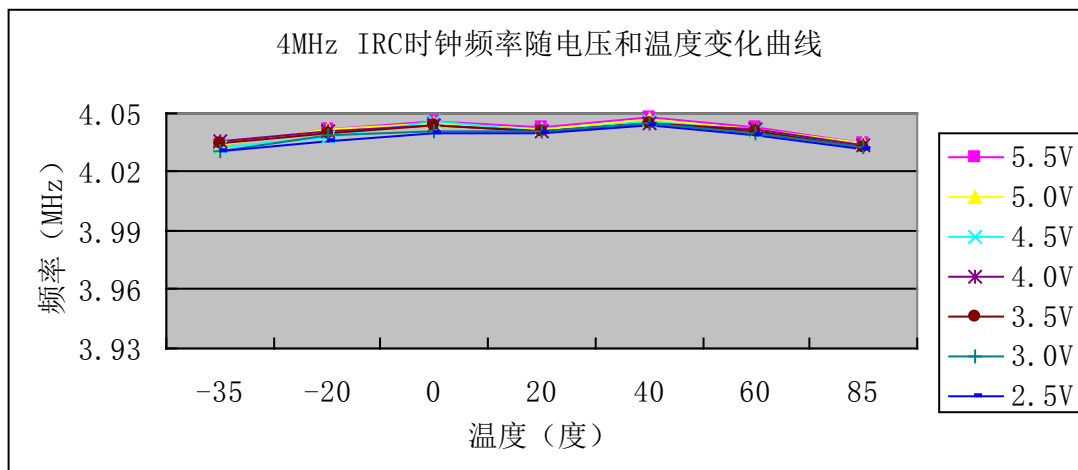


图29 RC 时钟频率的电压和温度特性

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。

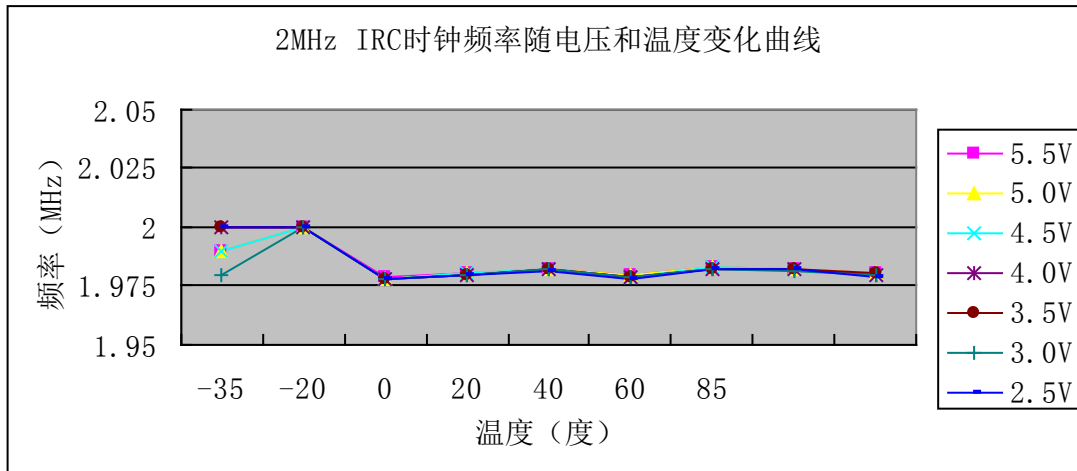


图30 RC 时钟频率的电压和温度特性

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。

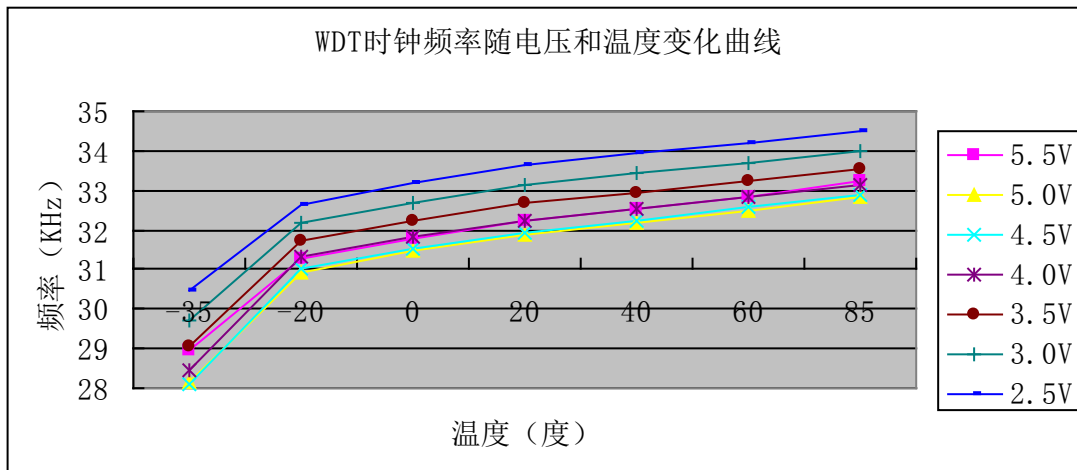


图31 WDT 频率的电压和温度特性

5.11 1.8V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。

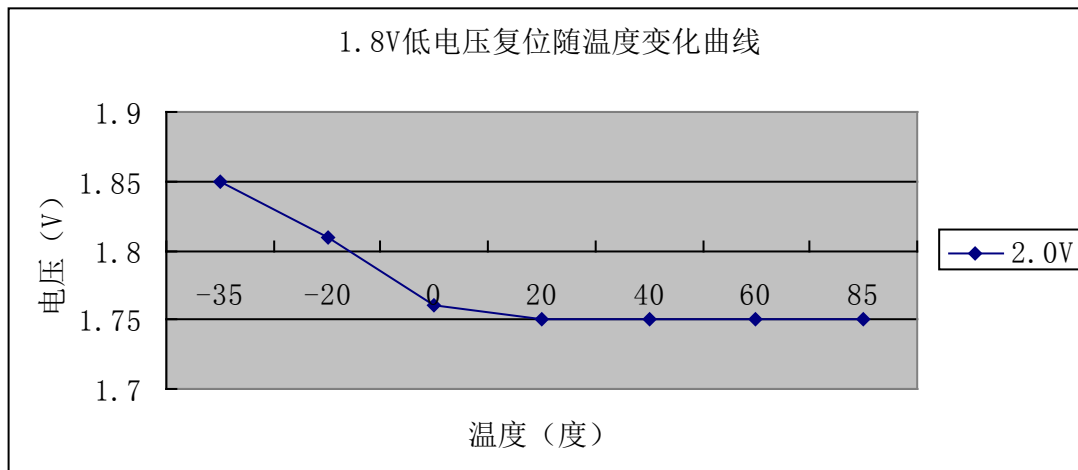


图32 2.0V 掉电复位温度特性

5.12 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。

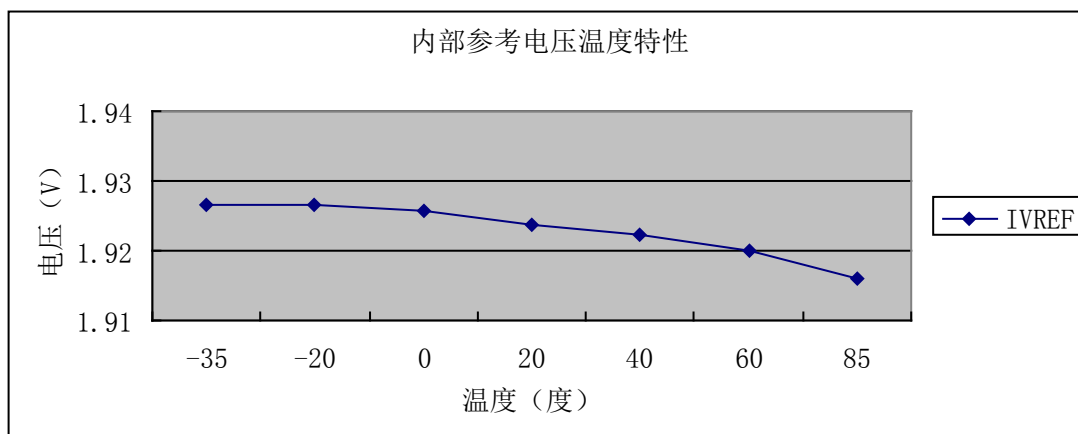


图33 内置参考电压温度特性

5.13 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。

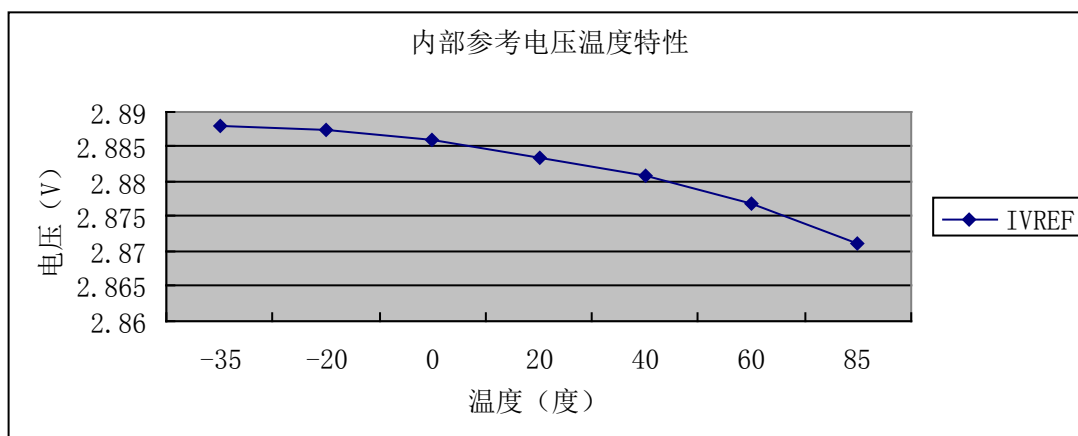


图34 内置参考电压温度特性

5.14 4.0V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。

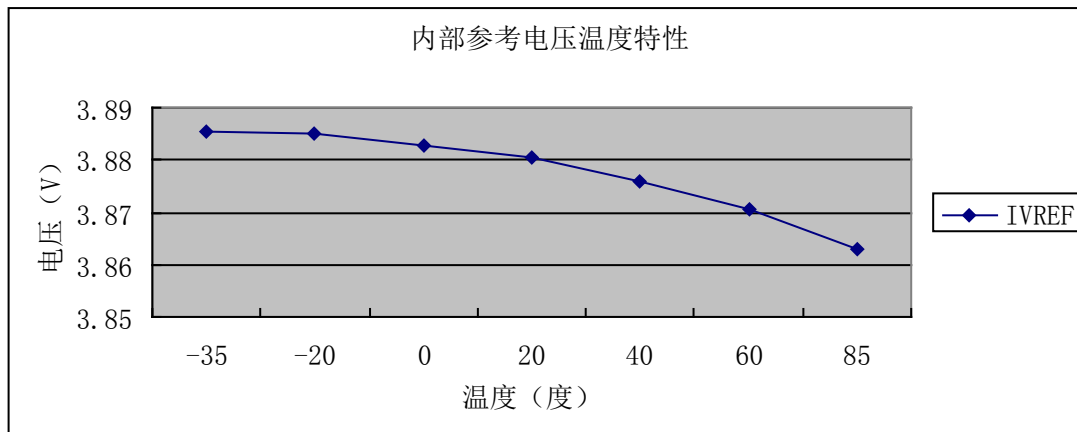
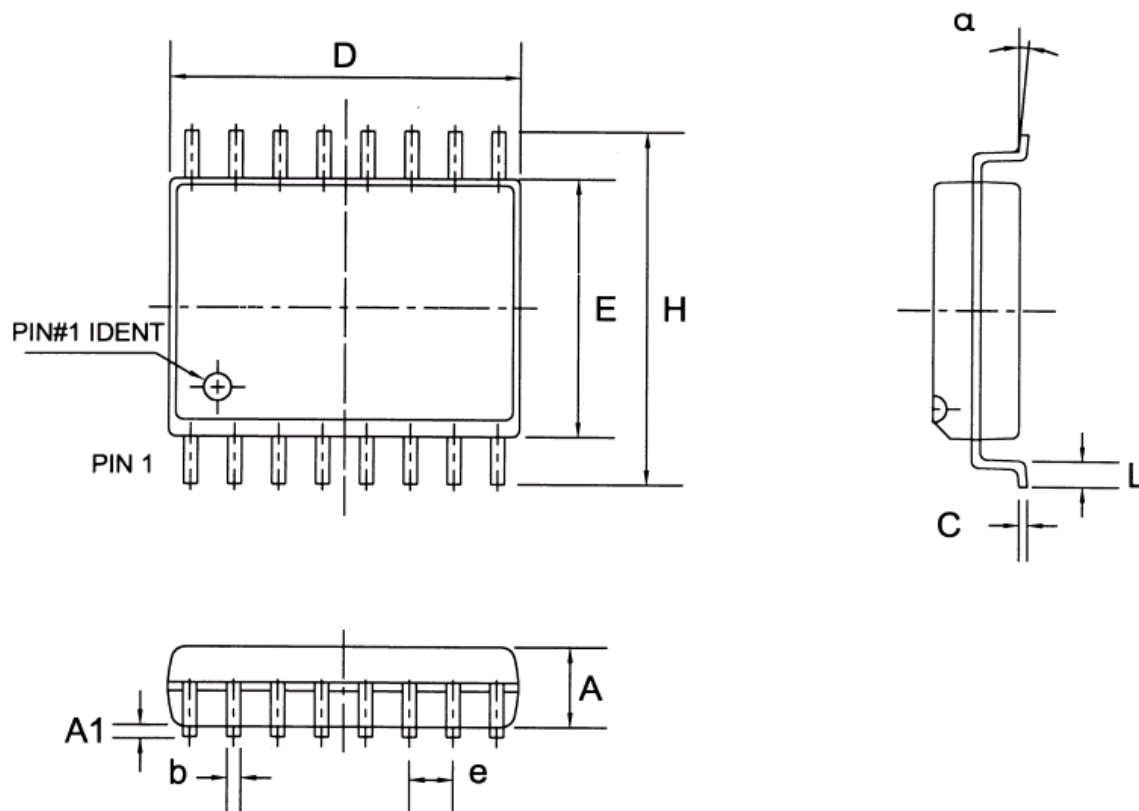


图35 内置参考电压温度特性

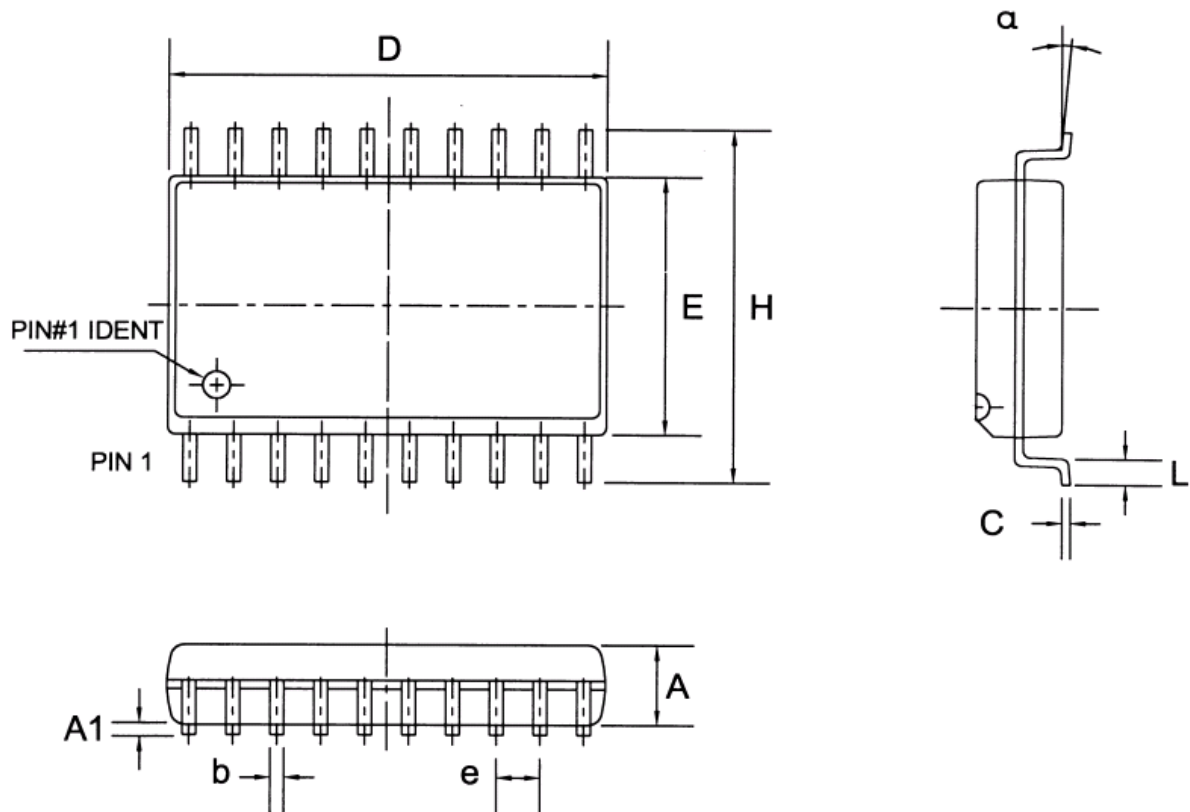
6 封装图

6.1 SOP-16pin



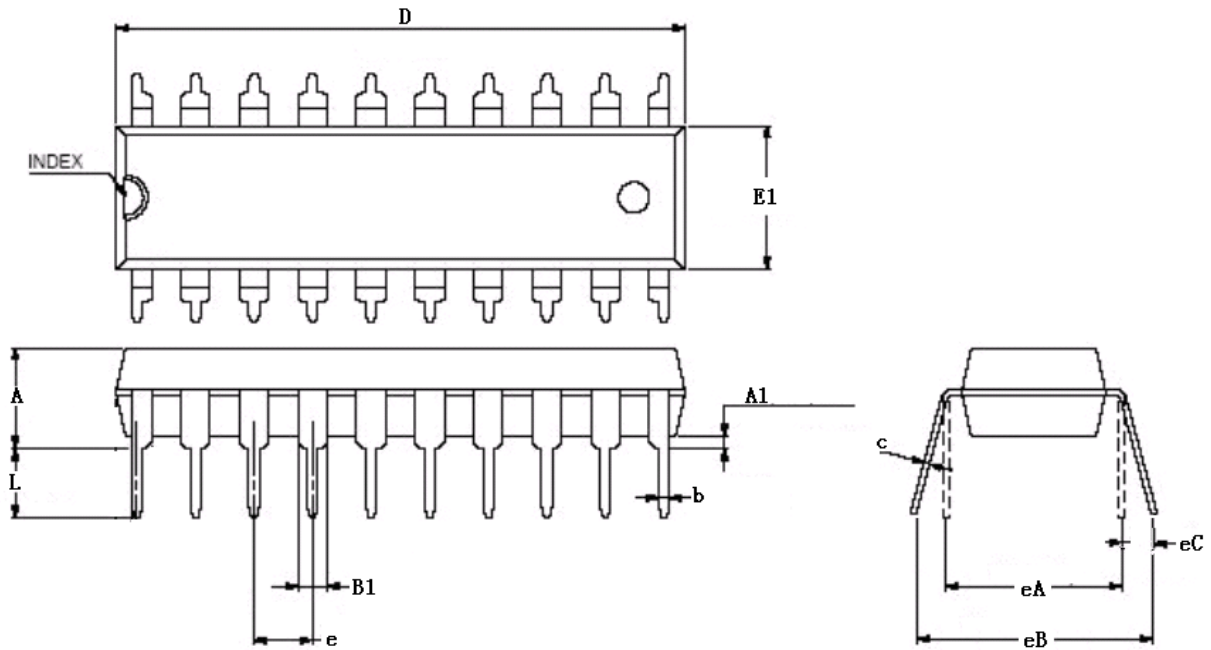
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	1.3	1.40	1.5
A1	0.05	-	0.225
b	0.39	-	0.48
C	0.21	-	0.26
D	9.70	9.90	10.10
E	3.70	3.90	4.10
e	1.27BSC		
α	0°	-	8°
H	5.8	6.0	6.2
L	0.5	-	0.8

6.2 SOP-20pin



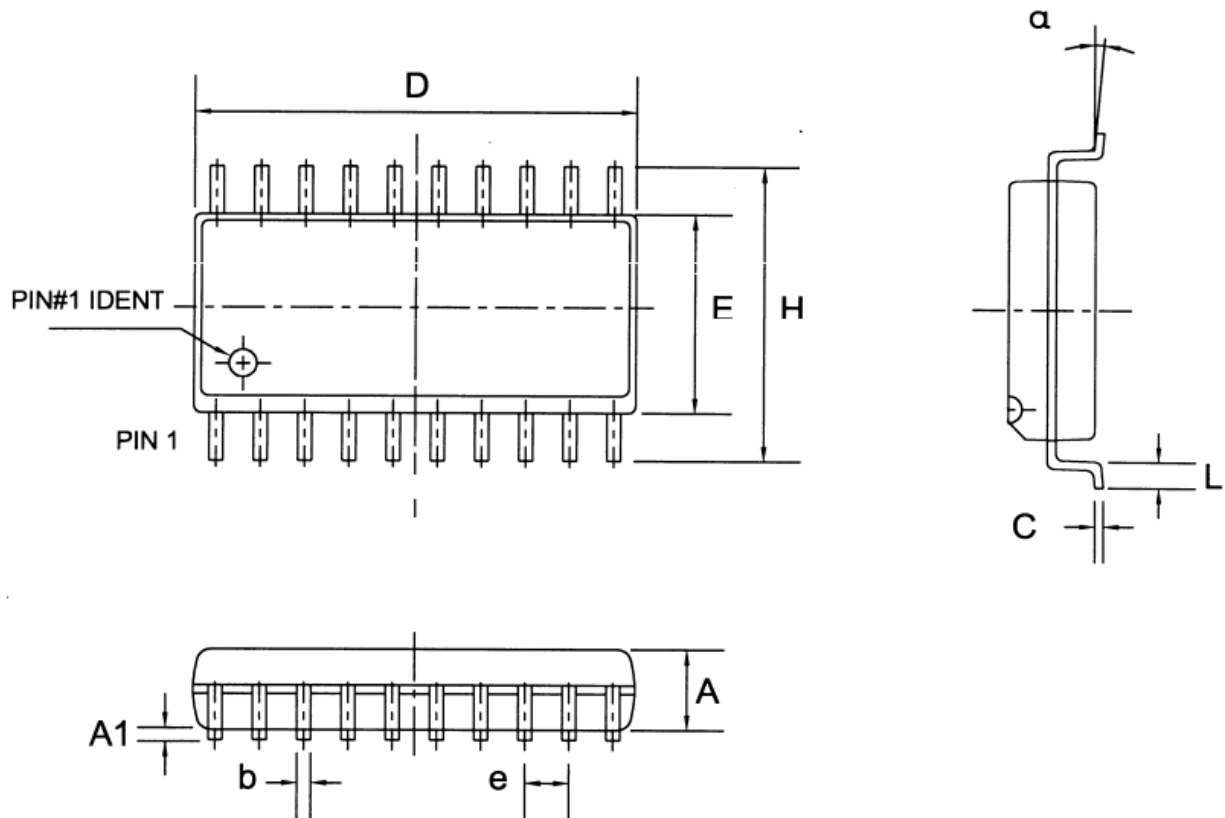
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	2.25	2.30	2.35
A1	0.1	-	0.3
b	0.35	-	0.44
C	0.26	-	0.31
D	12.6	12.8	13.0
E	7.3	7.5	7.7
e	1.27BSC		
α	0°	-	8°
H	10.1	10.3	10.5
L	0.7	-	1.0

6.3 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	-	-

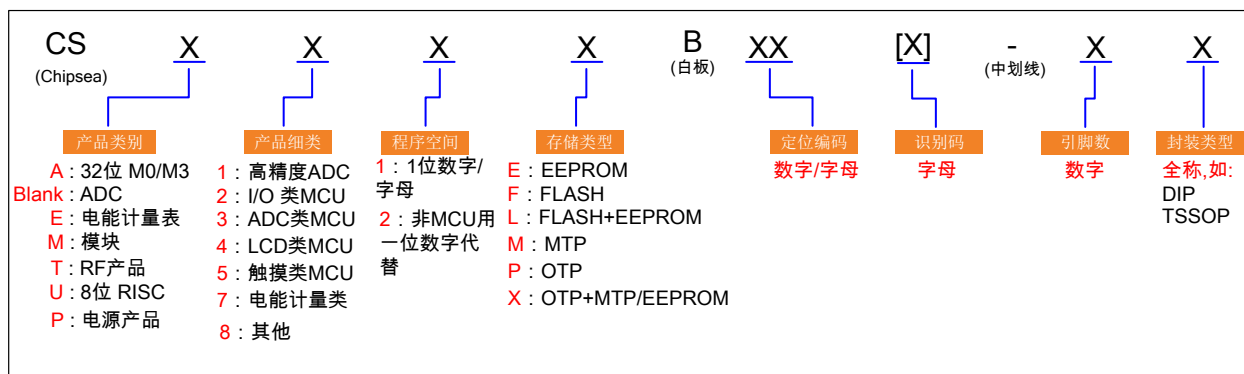
6.4 SSOP-20pin



SYMBOLS	MIN	NOR	MAX
	(mm)		
A	1.30	1.40	1.50
A1	0.10	-	0.25
b	0.23	-	0.33
C	0.21	-	0.26
D	8.45	8.65	8.85
E	3.70	3.90	4.10
e	0.635BSC		
a	0°	-	8°
H	5.80	6.00	6.20
L	0.50	-	0.80

7 单片机产品命名规则

7.1 产品型号说明

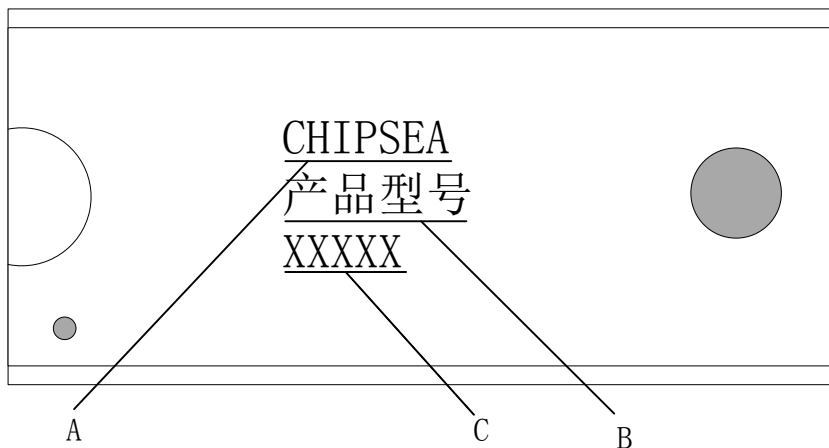


标示符	封装类型
BD	Bonding
DI	DIP
SD	SDIP
SO	SOP
SS	SSOP
TS	TSSOP
QF	QFP
LQ	LQFP
TQ	TQFP
QN	QFN

7.2 命名举例说明

名称	内核	ROM 类型	ROM 大小	功能 分类	产品定 位型号	芯片 版本	封装 形式	引脚数
CSU34F10-SSOP20	8 位 Risc MCU	Flash	4K*16	ADC	10	第 1 版	SSOP	20

7.3 产品印字说明



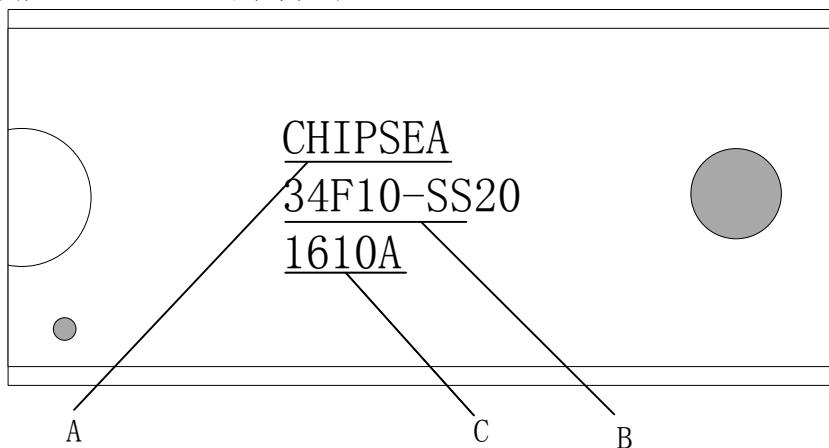
芯片正面印字一般有 3 行：

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， CSU34F10 的印字如下：



注： CSU34F10-SSOP20 的产品型号印字为 34F10-SS20。