



---

## CSU38M20 系列芯片用户手册

基于 MTP 的带 12bit ADC 的 8 位 RISC MCU

REV 1.1.0

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

## 版本历史

历史版本.	修改内容	版本日期
1.0.0	起草	2016-8-4
1.0.1	1、修改定时器蜂鸣器及 PWM 控制寄存器描述	2016-11-10
1.0.2	1、SOP20 脚位修改, GND 与 VDD 位置交换	2016-12-5
1.0.3	1、增加 TSSOP20 封装	2016-12-21
1.0.4	1、定时器 2、3、4 章节描述修改 2、增加 METCH[6]做内部参考电压输出控制 3、增加 SOP16、SSOP24 封装 4、增加 MTP 及 EEPROM 读写特性	2017-1-12
1.0.5	1、修改笔误	2017-4-1
1.0.6	1、增加 EEPROM 及 MTP 写操作电流, P191 2、I2C 时钟预分频公式更新, P120 3、I2C 中断寄存器 TXRX_STS 寄存器清零方式修改, P55、P121、P122 4、定时器描述更新, 修改对改变定时器溢出值的说明, 不能在短时间内连续两次修改定时器溢出值。P78、P86、P95 5、增加 QFN24 封装, P13、P22、P206	2017-6-15
1.0.7	1、定时器互补 PWM 框图修改, P84、P92、P103 2、笔误修改, P53 3、定时器 2、3、4 框图修改 P78、P86、P95	2017-7-10
1.0.8	1、QFN24-PIN 配置增加衬底说明 P12、P13	2018-1-17
1.0.9	1、修改比较器模块 CMPCON0 寄存器的 bit0 位描述, P141	2019-7-29
1.1.0	1、增加 QFN28 封装, P14、P22、P206	2019-8-2

## 目 录

版本历史 .....	2
目 录 .....	3
<b>1 产品概述 .....</b>	<b>7</b>
1.1 功能描述 .....	7
1.2 主要特性 .....	7
1.3 选型表 .....	9
1.4 PIN 配置 .....	10
1.4.1 SOP28-PIN 配置 .....	10
1.4.2 SSOP28-PIN 配置 .....	10
1.4.3 SOP24-PIN 配置 .....	10
1.4.4 SSOP24-PIN 配置 .....	11
1.4.5 SOP20-PIN 配置 .....	11
1.4.6 DIP20-PIN 配置 .....	11
1.4.7 TSSOP20-PIN 配置 .....	12
1.4.8 SOP16-PIN 配置 .....	12
1.4.9 QFN24 PIN 配置 .....	12
1.4.10 QFN28PIN 配置 .....	14
1.4.11 引脚说明 .....	14
<b>2 标准功能 .....</b>	<b>27</b>
2.1 CPU 核 .....	27
2.1.1 存储器 .....	29
2.1.2 状态寄存器 .....	33
2.1.3 SFR .....	34
2.2 时钟系统 .....	38
2.2.1 概述 .....	38
2.2.2 时钟框图 .....	38
2.2.3 时钟符号说明 .....	38
2.2.4 寄存器 .....	40
2.2.5 内部高速 RC 时钟 .....	42
2.2.6 内部低速 WDT 时钟 .....	42
2.2.7 外部高速晶振时钟 .....	42
2.2.8 外部低速晶振时钟 .....	42
2.3 复位系统 .....	43
2.3.1 上电复位 .....	44
2.3.2 看门狗复位 .....	44
2.3.3 低电压复位 .....	45
2.3.4 外部硬件复位 .....	46
2.4 PC 跳转 .....	47
2.4.1 PC 跳转操作 .....	47
2.4.2 PC 读操作 .....	47
2.5 中断 .....	49
2.5.1 中断使能寄存器 .....	51
2.5.2 中断标志寄存器 .....	53
2.5.3 外部中断 0 .....	55
2.5.4 外部中断 1 .....	55

2.5.5	外部中断 2 .....	55
2.5.6	AD 中断溢出 .....	56
2.5.7	定时器 0 溢出中断 .....	56
2.5.8	定时/计数器 2 溢出中断 .....	56
2.5.9	定时/计数器 3 溢出中断 .....	56
2.5.10	定时/计数器 4 溢出中断 .....	56
2.5.11	比较器 0 中断 .....	56
2.5.12	SPI 中断 .....	56
2.5.13	I2C 中断 .....	56
2.5.14	MUL 中断 .....	56
2.5.15	COG0 互补输出发生器中断 .....	56
2.5.16	PWM 暂停中断 .....	57
2.5.17	PWM 暂停失效中断 .....	57
2.5.18	UART 发送中断 .....	57
2.5.19	UART 接收中断 .....	57
2.5.20	LVD 检测中断 .....	57
2.5.21	PUSH 和 POP 处理 .....	57
2.6	定时器 0 .....	59
2.6.1	寄存器说明 .....	59
2.7	I/O PORT .....	61
2.7.1	PT1 口 .....	61
2.7.2	PT2 口 .....	64
2.7.3	PT3 口 .....	65
2.7.4	PT5 口 .....	67
2.7.5	IO 口外部中断使能 .....	69
2.7.6	驱动能力配置 .....	71
2.7.7	输入逻辑电平电压配置 .....	72
<b>3</b>	<b>增强功能 .....</b>	<b>73</b>
3.1	防误操作 .....	73
3.2	HALT 和 SLEEP 模式 .....	73
3.3	看门狗(WDT) .....	75
3.4	PWM 暂停控制器 .....	77
3.4.1	寄存器说明 .....	77
3.4.2	操作 .....	78
3.5	定时/计数器 2 .....	79
3.5.1	寄存器说明 .....	79
3.5.2	蜂鸣器 .....	83
3.5.3	PWM .....	84
3.5.4	互补式 PWM 输出 .....	84
3.6	定时/计数器 3 .....	87
3.6.1	寄存器说明 .....	87
3.6.2	蜂鸣器 .....	92
3.6.3	PWM .....	92
3.6.4	互补式 PWM 输出 .....	93
3.6.5	小数分频 .....	94
3.7	定时/计数器 4 .....	96
3.7.1	寄存器说明 .....	96
3.7.2	蜂鸣器 .....	103
3.7.3	PWM .....	103
3.7.4	互补式 PWM 输出 .....	104

3.7.5	捕获 .....	105
3.8	串行通信接口(UART).....	107
3.8.1	工作方式 .....	107
3.8.2	寄存器说明 .....	109
3.8.3	波特率 .....	111
3.9	SPI 接口.....	112
3.9.1	SPI 接口介绍.....	112
3.9.2	SPI 寄存器说明.....	115
3.9.3	SPI 操作说明: .....	117
3.10	两线串行接口(I2C).....	119
3.10.1	I2C MASTER 模块 .....	119
3.10.2	寄存器说明 .....	119
3.10.3	I2C 中断 .....	123
3.10.4	I2CM 的操作模式.....	123
3.11	模数转换器(ADC) .....	128
3.11.1	全差分输入 AD 转换结果.....	129
3.11.2	寄存器说明 .....	130
3.11.3	转换时间 .....	138
3.11.4	AD 失调电压校正.....	140
3.11.5	数字比较器 .....	141
3.12	比较器/运算放大器 .....	142
3.12.1	比较器 0/运算放大器 0.....	142
3.12.2	比较器参考电压 .....	144
3.12.3	比较器失调电压校正 .....	144
3.12.4	比较器中断 .....	145
3.13	互补输出发生器(COG).....	145
3.13.1	互补输出发生器框图 .....	145
3.13.2	COG0.....	146
3.14	5*21 LCD .....	150
3.14.1	寄存器说明 .....	150
3.14.2	硬件 LCD 驱动.....	156
3.14.3	软件 LCD 实现方法.....	160
3.15	乘法器 .....	161
3.15.1	寄存器说明 .....	161
3.16	CRC16.....	161
3.16.1	寄存器说明 .....	162
3.17	ISP .....	163
3.17.1	寄存器说明 .....	163
3.17.2	MTP 及 EEPROM 写操作 .....	166
3.17.3	MTP 及 EEPROM 读操作 .....	166
3.17.4	MTP 或 EEPROM 区 CRC 校验.....	167
3.18	烧录模块 .....	168
3.19	代码选项 .....	169
<b>4</b>	<b>MCU 指令集.....</b>	<b>172</b>
<b>5</b>	<b>电气特性.....</b>	<b>188</b>
5.1	极限值 .....	188
5.2	直流特性 (VDD = 5V, T <sub>A</sub> = 25 °C, 如无其他说明则都是此条件) .....	188
5.3	ADC 特性 (VDD = 5V, T <sub>A</sub> = 25 °C, 如无其他说明则都是此条件) .....	190
5.4	比较器的直流特性 .....	191

5.5	MTP 及 EEPROM 读写特性 .....	192
5.6	32MHz IRC 时钟频率特性.....	192
5.7	16MHz IRC 时钟频率特性.....	193
5.8	8MHz IRC 时钟频率特性.....	193
5.9	4MHz IRC 时钟频率特性.....	194
5.10	2MHz IRC 时钟频率特性.....	194
5.11	WDT 时钟频率特性 .....	194
5.12	2.0V 掉电复位温度特性.....	195
5.13	2.4V 低电压复位温度特性.....	195
5.14	3.0V 低电压复位温度特性.....	196
5.15	3.6V 低电压复位温度特性.....	196
5.16	1.4V 内部参考电压温度特性.....	196
5.17	2V 内部参考电压温度特性.....	197
5.18	3V 内部参考电压温度特性.....	197
5.19	4V 内部参考电压温度特性.....	197
<b>6</b>	<b>封装图.....</b>	<b>199</b>
6.1	SOP-28 PIN .....	199
6.2	SSOP-28 PIN .....	200
6.3	SOP-24 PIN .....	201
6.4	SSOP-24 PIN .....	202
6.5	SOP-20 PIN .....	202
6.6	DIP-20 PIN .....	204
6.7	TSSOP-20 PIN.....	205
6.8	SOP-16 PIN .....	205
6.9	QFN-24 PIN (4×4×0.75-0.4) .....	206
6.10	QFN-28 PIN (4×4×0.75-0.4) .....	207
<b>7</b>	<b>单片机产品命名规则.....</b>	<b>209</b>
7.1	产品型号说明 .....	209
7.2	命名举例说明 .....	210
7.3	产品印字说明 .....	210

## 1 产品概述

### 1.1 功能描述

CSU38M20 是一个带 12bitADC 的 8 位 RISC MCU，内置 8K×16 位 MTP 程序存储器。

### 1.2 主要特性

#### 高性能的 RISC CPU

- 8 位单片机 MCU
- 内置 8K×16 位程序存储器 MTP
- 1K 字节数据存储器 (SRAM)
- 带 256 字节 E2PROM 功能 (10000 次)
- 只有 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

- 32/16/8/4/2MHz 和 32KHz 的 IRC，精度为 ±1% (25°C, 5V)
- 外部 32768Hz 晶振 (RTC) 或 4MHz~16MHz 晶振

#### 外设特性

- 26 路双向 I/O 口，带上下拉。
- 3 路蜂鸣器输出，3 路 PWM 输出
- 3 组互补式 PWM 输出，输出驱动高达 56mA
- 14 个内部中断，3 个外部中断 (INT0/INT1/INT2)
- 26 个具有唤醒功能的输入口
- 26+5 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
  - 支持差分输入
  - 带数字比较器
  - 支持运算放大器输出做 AD 输入
- 1 个内置模拟比较器/运算放大器(可调零，支持内部放大功能)
- 提供 1 个在 1.4V、2.0V、3.0V 和 4.0V 之间可选的参考电压，精度为 ±1% (25°C, 5V)
- 低电压检测 (LVD) 引脚，内部提供 2.4V、2.5V、3.0V、3.6V、以及片外电压检测，默认开数字滤波 150us，具备中断/复位功能
- 8 个双边大驱动输出，PT3.2/PT3.3，PT1.5/PT1.6/PT1.7，PT5.5/PT5.6/PT5.7 可配

置为大驱动输出，高达 56mA

- 输入逻辑电平电压可配置，施密特可配置
- LCD 驱动，1/3、1/4、1/5duty,1/2bias 任意口 COM/SEG 可配；1/4、1/5、1/6duty,1/3bias 固定 6 个 COM 口
- UART 通信
- 一个 SPI (带 Master 和 Slave)
- I2C (仅带 Master)
- 乘法器 8\*8

#### 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 2.0V、2.4V、2.5V、3.0V、3.6V 和外部低电压复位 (LVR)
- 定时器 0
  - 8 位可编程预分频定时/计数器
- 定时/计数器 2
  - 12 位可编程预分频定时/计数器
- 定时/计数器 3
  - 12 位可编程预分频定时/计数器
  - 支持小数分频
- 增强型定时/计数器 4 支持波形捕获 (PWM)
  - 16 位可编程预分频定时/计数器
- 扩展型看门狗定时器 (32768Hz WDT)
  - 可编程的时间范围

#### CMOS 技术

- 工作电压范围
  - 2.4V~5.5V@fcpu=4MHz (指令周期)
  - 2.2V~5.5V@fcpu=2MHz (指令周期)
- 工作温度范围
  - -40~85 °C (工业级)

## 低功耗特性

- MCU 工作电流
  - 正常模式 3mA@fcpu=4MHz (工作电压 3V)
  - 正常模式 20uA@32KHz(工作电压 3V)
  - **Halt 模式 1.2mA(工作电压 3V)**
  - 休眠模式下的电流小于 1μA

## 封装

SOP28 、 SOP24 、 SOP20、 SSOP28、 SSOP24、  
TSSOP20、 SOP16、 DIP20、 QFN24



1.3 选型表

型号	ROM	EEPROM	RAM	堆栈	定时器	PWM	互补PWM	I/O	唤醒口	UART	SPI	I2C	LCD 驱动	ADC (CH*Bit)	模拟比较器	大驱动	外部晶振
CSU38M20-SOP28/SSOP28	8K*16	256*8	1K	8	4	3	3组	26	26	1	1	1	5*21 1/3 Bias	26*12	1	√	√
CSU38M20-SSOP24/SOP24	8K*16	256*8	1K	8	4	3	3组	22	22	1	1	1	5*17 1/3 Bias	22*12	1	√	√
CSU38M20-SOP20/DIP20	8K*16	256*8	1K	8	4	3	3组	18	18	1	1	1	5*13 1/3 Bias	18*12	--	√	√
CSU38M20-TSSOP20	8K*16	256*8	1K	8	4	3	2组	18	18	1	1	1	5*13 1/3 Bias	18*12	--	√	√
CSU38M20-SOP16	8K*16	256*8	1K	8	4	3	2组	14	14	1	1	1	5*9 1/3 Bias	14*12	--	√	√
CSU38M20-QFN24	8K*16	256*8	1K	8	4	3	3组	22	22	1	1	1	5*17 1/3 Bias	22*12	--	√	√

### 1.4 PIN 配置

#### 1.4.1 SOP28-PIN 配置

对于 SOP28 脚封装，PIN 排列如下图

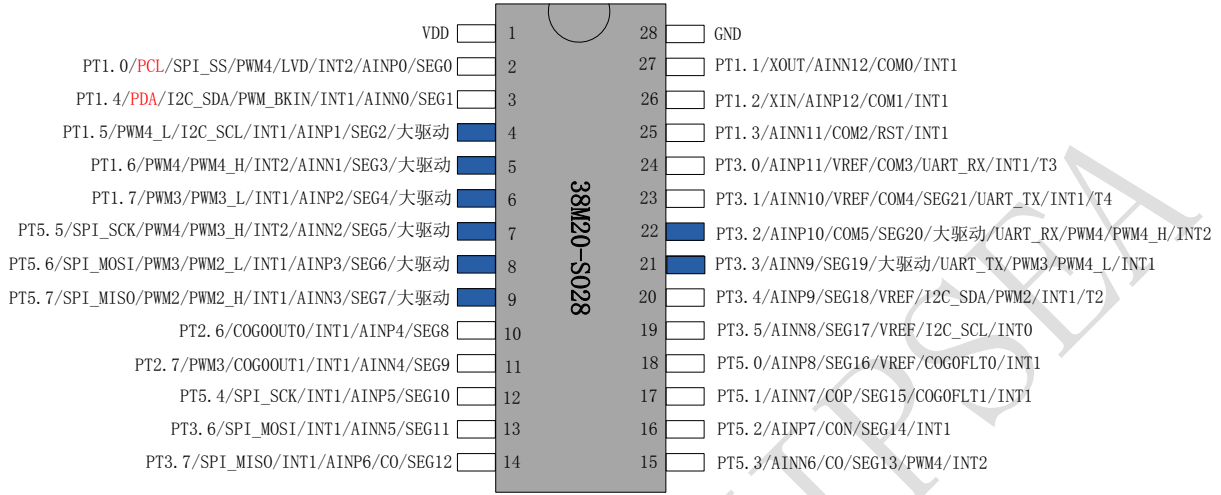


图1 SOP28-pin 配置图

#### 1.4.2 SSOP28-PIN 配置

CSU38M20 的 SSOP28 封装与 SOP28 的 PIN 排列有些不同，PT5.2 口与 PT3.7 口位置交换，PIN 图如下

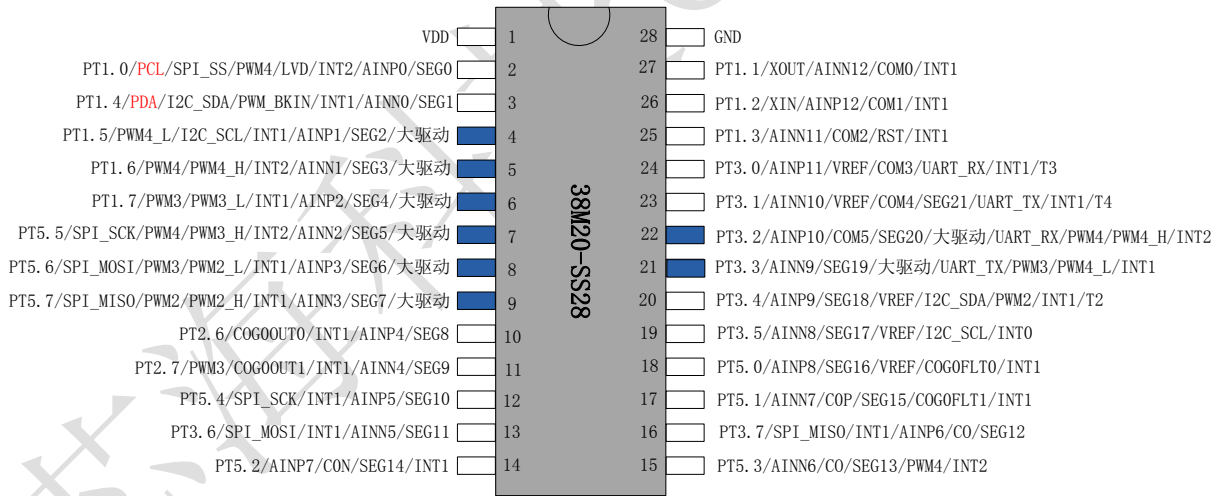


图2 SSOP28-pin 配置图

#### 1.4.3 SOP24-PIN 配置

对于 SOP24 脚封装，PIN 排列如下图

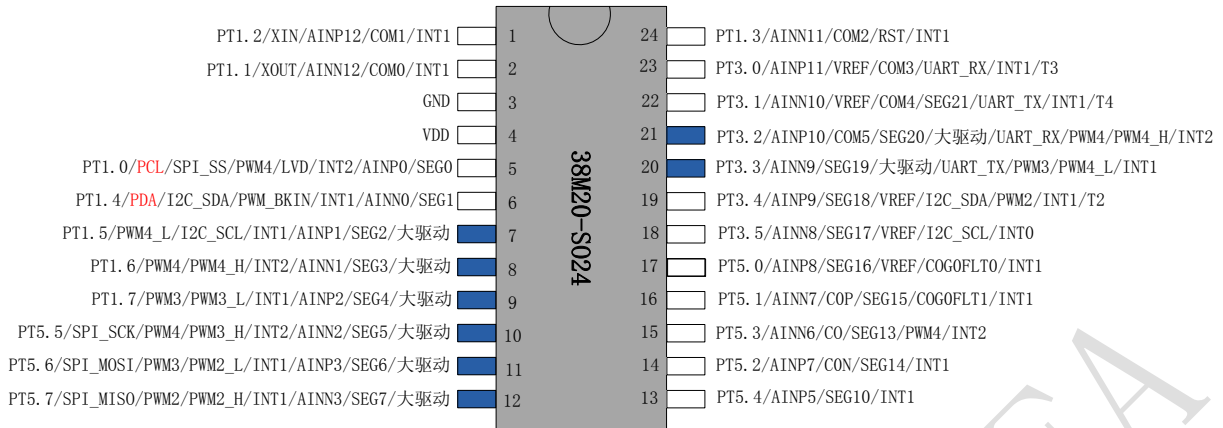


图3 SOP24-pin 配置图

### 1.4.4 SSOP24-PIN 配置

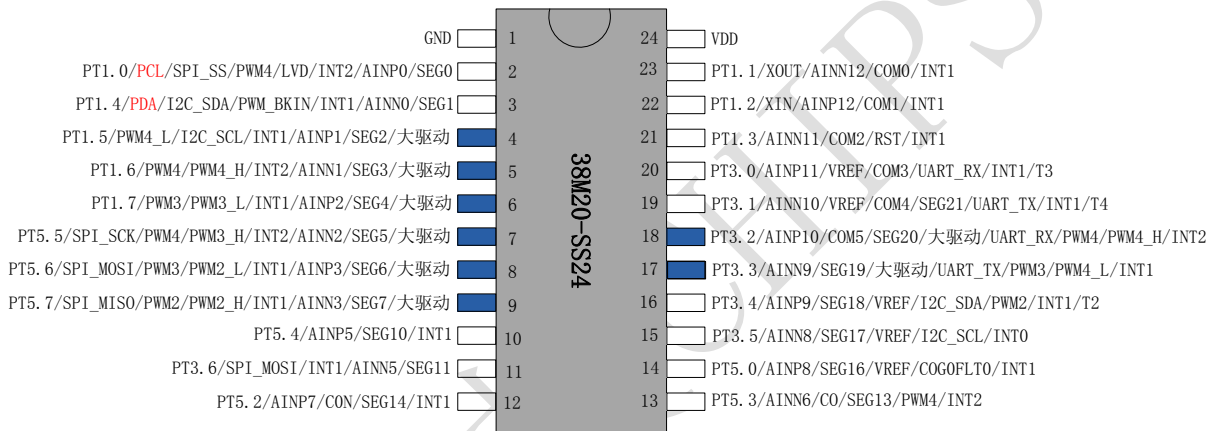


图4 SSOP24-pin 配置图

### 1.4.5 SOP20-PIN 配置

对于 SOP20 脚封装，PIN 排列如下图

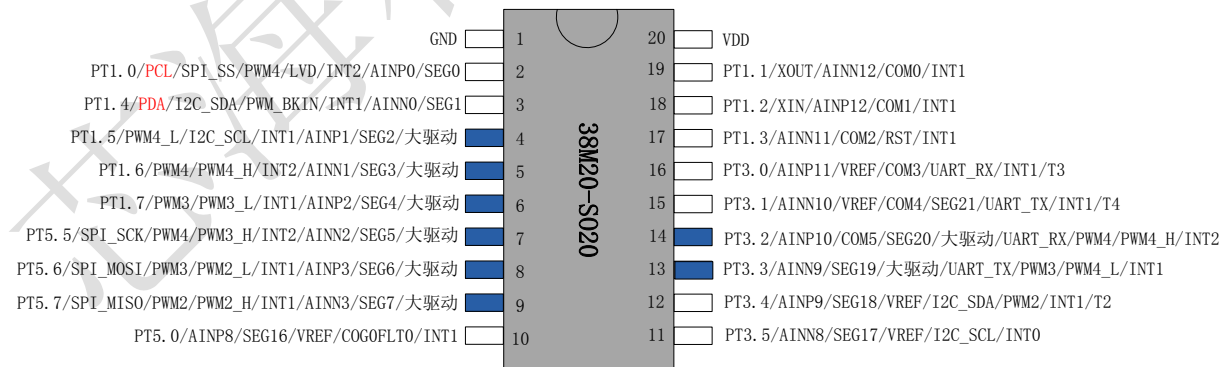


图5 SOP20-pin 配置图

### 1.4.6 DIP20-PIN 配置

对于 DIP20 脚封装，PIN 排列如下图

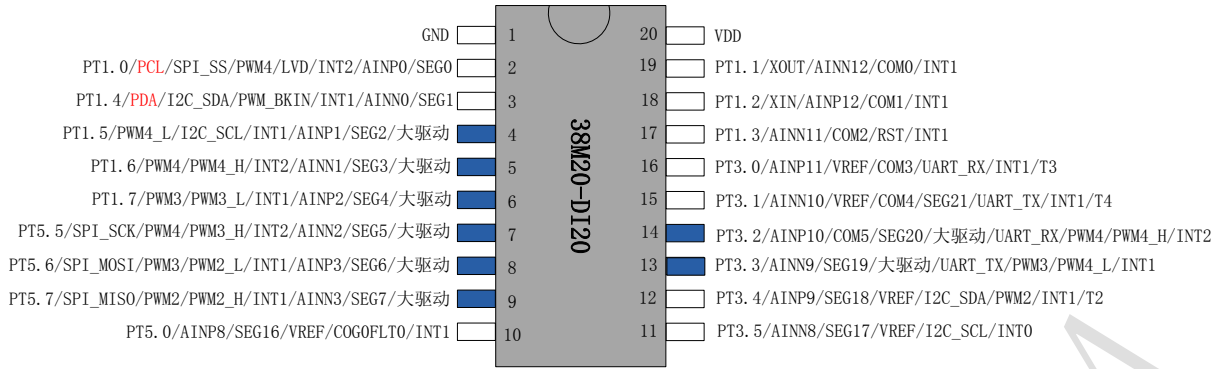


图6 DIP20-pin 配置图

### 1.4.7 TSSOP20-PIN 配置

对于 TSSOP20 脚封装，PIN 排列如下图

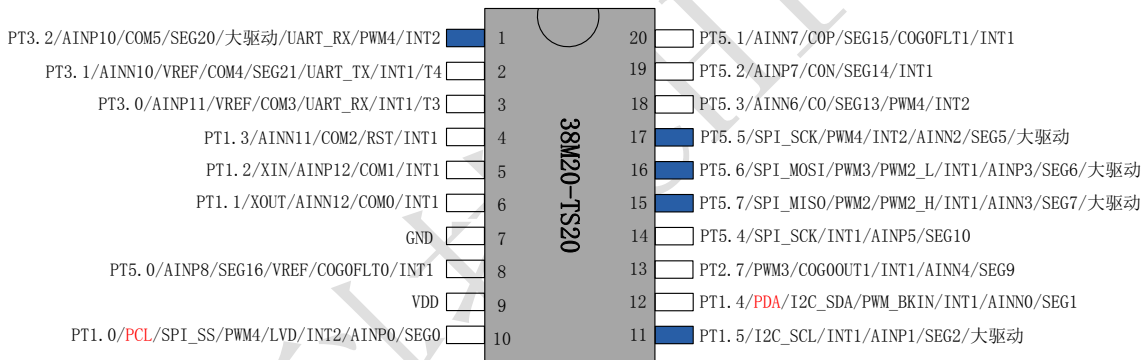


图7 TSSOP20-pin 配置图

### 1.4.8 SOP16-PIN 配置

对于 SOP16 脚封装，PIN 排列如下图

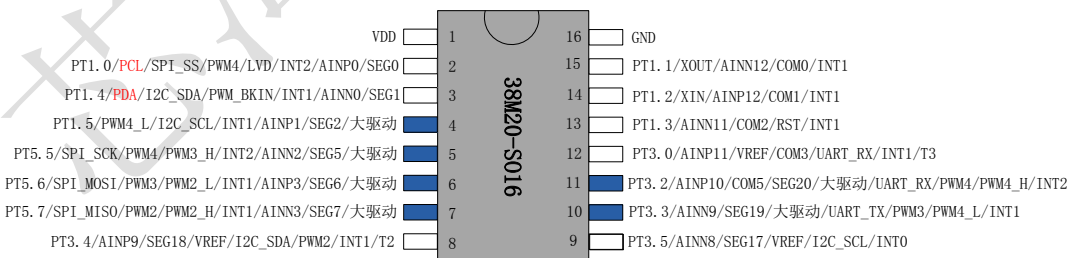


图8 SOP16-pin 配置图

### 1.4.9 QFN24 PIN 配置

该封装底部中间散热焊盘并非 GND，而是 PT1.3 口，使用时需要特别注意。

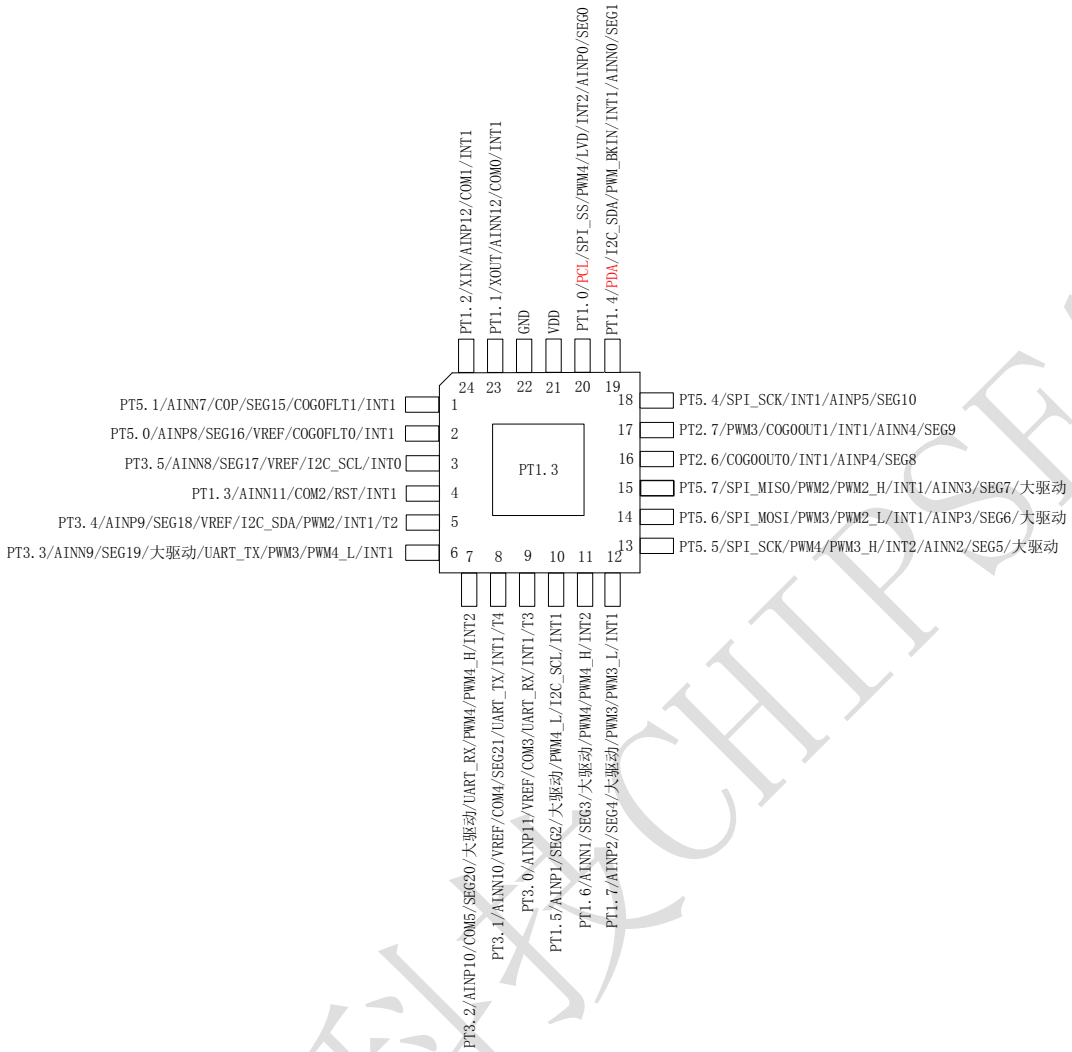


图9 QFN24-pin 配置图



图10 引脚说明表

SOP28 管脚 序号	SSOP28 管脚 序号	SSOP24 管脚 序号	SOP24 管脚序 号	DIP/S OP20 管脚序 号	SOP16 管脚 序号	TSS OP20 管脚 序号	复用功能	类型	
1	1	24	4	20	1	9	VDD	P	电源
2	2	2	5	2	2	10	PT1.0	I/O	IO口;
							PCL	I	烧录时钟口;
							SPI_SS	I/O	SPI从机选择;
							PWM4	O	PWM4输出;
							INT2	I	外部中断2;
							LVD	I	低电压检测电压点输入;
							AINP0	AI	ADC通道0的P端输入;
							SEG0	AO	LCD SEG0;
3	3	3	6	3	3	12	PT1.4	I/O	IO;
							PDA	I/O	烧录数据口;
							I2C_SDA	I/O	I2C数据口;
							PWM_BKIN	I	PWM暂停控制信号输入;
							INT1	I	外部中断1输入,具有唤醒功能;
							AINN0	AI	ADC通道0的N端输入;
							SEG1	AO	LCD SEG1;
4	4	4	7	4	4	11	PT1.5	I/O	IO;
							PWM4_L	O	定时器互补输出 PWM4L;
							I2C_SCL	O	I2C时钟口;
							INT1	I	外部中断1输入,具有唤醒功能;

								AINP1	AI	ADC 通道 1 的 P 端输入;
								SEG2	AO	LCD SEG2;
								大驱动	O	大驱动输出口;
								PT1.6	I/O	IO;
								PWM4	O	PWM4 输出;
								PWM4_H	O	PWM4H 输出;
5	5	5	8	5				AINN1	I	ADC 通道 1 的 N 端输入;
								INT2	AI	外部中断 2 输入, 具有唤醒功能;
								SEG3	AO	LCD SEG3;
								大驱动	O	大驱动输出口;
								PT1.7	I/O	IO;
								PWM3	O	PWM3 输出;
								PWM3_L	O	PWM3L 输出;
6	6	6	9	6				INT1	I	外部中断 1 输入, 具有唤醒功能;
								AINP2	AI	ADC 通道 2 的 P 端输入;
								SEG4	AO	LCD SEG4;
								大驱动	O	大驱动输出口;
								PT5.5	I/O	IO;
								SPI_SCK	I/O	SPI 通信时钟口;
								PWM4	O	PWM4 输出;
7	7	7	10	7	5	17		PWM3_H	O	PWM3H 输出;
								INT2	I	外部中断 2 输入, 具有唤醒功能;
								AINN2	AI	ADC 通道 2 的 N 端输入;



							SEG5	AO	LCD SEG5;
							大驱动	O	大驱动输出 口;
8	8	8	11	8	6	16	PT5.6	I/O	IO;
							SPI_MOSI	I/O	SPI 的主机 输出从机输 入口;
							PWM3	O	PWM3 输 出;
							PWM2_L	O	PWM2L 输 出;
							AINP3	AI	ADC 通道 3 的 P 端输 入;
							INT1	I	外部中断 1 输入, 具有 唤醒功能;
							SEG6	AO	LCD SEG6;
							大驱动	O	大驱动输出 口;
9	9	9	12	9	7	15	PT5.7	I/O	IO;
							SPI_MISO	I/O	SPI 的主机 输入从机输 出口;
							PWM2	O	PWM2 输 出;
							PWM2_H	O	PWM2H 输 出;
							INT1	I	外部中断 1 输入, 具有 唤醒功能;
							AINN3	AI	ADC 通道 3 的 N 端输 入;
							SEG7	AO	LCD SEG7;
							大驱动	O	大驱动输出 口;
10	10						PT2.6	I/O	IO;
							COG0OUT0	O	互补输出发 生器输出;
							INT1	I	外部中断 1 输入, 具有 唤醒功能;
							AINP4	AI	ADC 通道 4 的 P 端输 入;
							SEG8	AO	LCD

									SEG8;
11	11					13	PT2.7	I/O	IO;
							PWM3	O	PWM3 输出;
							COG0OUT1	O	互补输出发生器输出;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							AINN4	AI	ADC 通道 4 的 N 端输入;
							SEG9	AO	LCD SEG9;
12	12	10	13			14	PT5.4	I/O	IO;
							SPI_SCK	I/O	SPI 通信时钟口;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							AINP5	AI	ADC 通道 5 的 P 端输入;
							SEG10	AO	LCD SEG10;
13	13	11					PT3.6	I/O	IO;
							SPI_MOSI	I/O	SPI 的主机输出从机输入口;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							AINN5	AI	ADC 通道 5 的 N 端输入;
							SEG11	AO	LCD SEG11;
14	16						PT3.7	I/O	IO;
							SPI_MISO	I/O	SPI 的主机输入从机输出口;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							AINP6	AI	ADC 通道 6 的 P 端输入;
							CO	O/A O	比较器/运放输出;

							SEG12	AO	LCD SEG12;
15	15	13	15		18		PT5.3	I/O	IO;
							PWM4	O	PWM4 输出;
							INT2	I	外部中断 2 输入, 具有唤醒功能;
							AINN6	AI	ADC 通道 6 的 N 端输入;
							CO	O/A O	比较器/运放输出;
							SEG13	AO	LCD SEG13;
16	14	12	14		19		PT5.2	I/O	IO;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							AINP7	AI	ADC 通道 7 的 P 端输入;
							CON	AI	比较器负端输入;
17	17	16			20		PT5.1	I/O	IO;
							COG0FLT0	I	互补输出发生器外部触发输入 0;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							AINN7	AI	ADC 通道 7 的 N 端输入;
							COP	AI	比较器正端输入;
							SEG15	AO	LCD SEG15;
18	18	14	17	10	8		PT5.0	I/O	IO;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							COG0FLT1	I	互补输出发生器外部触发输入 1;
							AINP8	AI	ADC 通道 8 的 P 端输

									入;
							SEG16	AO	LCD SEG16;
							VREF	AI	外部参考电压输入;
19	19	15	18	11	9		PT3.5	I/O	IO;
							I2C_SCL	I/O	I2C 时钟口;
							INT0	I	外部中断 0 输入, 具有唤醒功能;
							AINN8	AI	ADC 通道 8 的 N 端输入;
							SEG17	AO	LCD SEG17;
							VREF	AI	外部参考电压输入;
20	20	16	19	12	8		PT3.4	I/O	IO;
							I2C_SDA	I/O	I2C 数据口;
							PWM2	O	PWM2 输出;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							T2	I	定时/计数器 2 外部计数输入;
							AINP9	AI	ADC 通道 9 的 P 端输入;
							SEG18	AO	LCD SEG18;
							VREF	AI	外部参考电压输入;
21	21	17	20	13	10		PT3.3	I/O	IO;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							UART_TX	O	UART 发送口;
							PWM3	O	PWM3 输出;
							PWM4_L	O	PWM4L 输出;
							AINN9	AI	ADC 通道 9 的 N 端输入;

							SEG19	AO	LCD SEG19;
							大驱动	O	大驱动输出 口;
							PT3.2	I/O	IO;
							INT2	I	外部中断 2 输入, 具有 唤醒功能;
							UART_RX	I	UART 接收 口;
							PWM4	O	PWM4 输 出;
22	22	18	21	14	11	1	PWM4_H	O	PWM4H 输 出;
							AINP10	AI	ADC 通道 10 的 P 端 输入;
							COM5	AO	LCD COM5;
							SEG20	AO	LCD SEG20;
							大驱动	O	大驱动;
							PT3.1	I/O	IO;
							UART_TX	O	UART 发送 口;
							INT1	I	外部中断 1 输入, 具有 唤醒功能;
							T4	I	定时/计数 器 4 外部计 数输入;
23	23	19	22	15		2	AINN10	AI	ADC 通道 10 的 N 端 输入;
							VREF	AI	外部参考电 压输入;
							COM4	AO	LCD COM4;
							SEG21	AO	LCD SEG21;

24	24	20	23	16	12	3	PT3.0	I/O	IO;
							UART_RX	I	UART 接收口;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							T3	I	定时/计数器 3 外部计数输入;
							AINP11	AI	ADC 通道 11 的 P 端输入;
							VREF	AI	外部参考电压输入;
							COM3	AO	LCD COM3;
25	25	21	24	17	13	4	PT1.3	I/O	IO;
							RST	I	外部硬件复位输入;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							AINN11	AI	ADC 通道 11 的 N 端输入;
							COM2	AO	LCD COM2;
26	26	22	1	18	14	5	PT1.2	I/O	IO;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							XIN	I	外置晶振输入;
							AINP12	AI	ADC 通道 12 的 P 端输入;
							COM1	AO	LCD COM1;
27	27	23	2	19	15	6	PT1.1	I/O	IO;
							INT1	I	外部中断 1 输入, 具有唤醒功能;
							XOUT	O	外置晶振输出;
							AINN12	AI	ADC 通道 12 的 N 端输入;
							COM0	AO	LCD COM0;

28	28	0	3	1	16	7	GND	P	地
----	----	---	---	---	----	---	-----	---	---

## 1.4.11.2 QFN24 引脚说明表

图11 QFN24 引脚说明表

QFN24 管脚序号	复用功能	类型	
1	PT5.1	I/O	IO;
	COG0FLT0	I	互补输出发生器外部触发输入 0;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINN7	AI	ADC 通道 7 的 N 端输入;
	COP	AI	比较器正端输入;
	SEG15	AO	LCD SEG15;
2	PT5.0	I/O	IO;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	COG0FLT1	I	互补输出发生器外部触发输入 1;
	AINP8	AI	ADC 通道 8 的 P 端输入;
	SEG16	AO	LCD SEG16;
	VREF	AI	外部参考电压输入;
3	PT3.5	I/O	IO;
	I2C_SCL	I/O	I2C 时钟口;
	INT0	I	外部中断 0 输入, 具有唤醒功能;
	AINN8	AI	ADC 通道 8 的 N 端输入;
	SEG17	AO	LCD SEG17;
	VREF	AI	外部参考电压输入;
4	PT1.3	I/O	IO;
	RST	I	外部硬件复位输入;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINN11	AI	ADC 通道 11 的 N 端输入;
	COM2	AO	LCD COM2;
5	PT3.4	I/O	IO;
	I2C_SDA	I/O	I2C 数据口;
	PWM2	O	PWM2 输出;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	T2	I	定时/计数器 2 外部计数输入;
	AINP9	AI	ADC 通道 9 的 P 端输入;
	SEG18	AO	LCD SEG18;
	VREF	AI	外部参考电压输入;
6	PT3.3	I/O	IO;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	UART_TX	O	UART 发送口;
	PWM3	O	PWM3 输出;
	PWM4_L	O	PWM4L 输出;
	AINN9	AI	ADC 通道 9 的 N 端输入;
	SEG19	AO	LCD SEG19;
	大驱动	O	大驱动输出口;

7	PT3.2	I/O	IO;
	INT2	I	外部中断 2 输入, 具有唤醒功能;
	UART_RX	I	UART 接收口;
	PWM4	O	PWM4 输出;
	PWM4_H	O	PWM4H 输出;
	AINP10	AI	ADC 通道 10 的 P 端输入;
	COM5	AO	LCD COM5;
	SEG20	AO	LCD SEG20;
	大驱动	O	大驱动;
8	PT3.1	I/O	IO;
	UART_TX	O	UART 发送口;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	T4	I	定时/计数器 4 外部计数输入;
	AINN10	AI	ADC 通道 10 的 N 端输入;
	VREF	AI	外部参考电压输入;
	COM4	AO	LCD COM4;
	SEG21	AO	LCD SEG21;
9	PT3.0	I/O	IO;
	UART_RX	I	UART 接收口;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	T3	I	定时/计数器 3 外部计数输入;
	AINP11	AI	ADC 通道 11 的 P 端输入;
	VREF	AI	外部参考电压输入;
	COM3	AO	LCD COM3;
10	PT1.5	I/O	IO;
	PWM4_L	O	定时器互补输出 PWM4L;
	I2C_SCL	O	I2C 时钟口;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINP1	AI	ADC 通道 1 的 P 端输入;
	SEG2	AO	LCD SEG2;
	大驱动	O	大驱动输出出口;
11	PT1.6	I/O	IO;
	PWM4	O	PWM4 输出;
	PWM4_H	O	PWM4H 输出;
	AINN1	I	ADC 通道 1 的 N 端输入;
	INT2	AI	外部中断 2 输入, 具有唤醒功能;
	SEG3	AO	LCD SEG3;
	大驱动	O	大驱动输出出口;
12	PT1.7	I/O	IO;
	PWM3	O	PWM3 输出;
	PWM3_L	O	PWM3L 输出;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINP2	AI	ADC 通道 2 的 P 端输入;
	SEG4	AO	LCD SEG4;
	大驱动	O	大驱动输出出口;
13	PT5.5	I/O	IO;
	SPI_SCK	I/O	SPI 通信时钟口;
	PWM4	O	PWM4 输出;



	PWM3_H	O	PWM3H 输出;
	INT2	I	外部中断 2 输入, 具有唤醒功能;
	AINN2	AI	ADC 通道 2 的 N 端输入;
	SEG5	AO	LCD SEG5;
	大驱动	O	大驱动输出出口;
14	PT5.6	I/O	IO;
	SPI_MOSI	I/O	SPI 的主机输出从机输入口;
	PWM3	O	PWM3 输出;
	PWM2_L	O	PWM2L 输出;
	AINP3	AI	ADC 通道 3 的 P 端输入;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	SEG6	AO	LCD SEG6;
	大驱动	O	大驱动输出出口;
15	PT5.7	I/O	IO;
	SPI_MISO	I/O	SPI 的主机输入从机输出出口;
	PWM2	O	PWM2 输出;
	PWM2_H	O	PWM2H 输出;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINN3	AI	ADC 通道 3 的 N 端输入;
	SEG7	AO	LCD SEG7;
	大驱动	O	大驱动输出出口;
16	PT2.6	I/O	IO;
	COG0OUT0	O	互补输出发生器输出;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINP4	AI	ADC 通道 4 的 P 端输入;
	SEG8	AO	LCD SEG8;
17	PT2.7	I/O	IO;
	PWM3	O	PWM3 输出;
	COG0OUT1	O	互补输出发生器输出;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINN4	AI	ADC 通道 4 的 N 端输入;
	SEG9	AO	LCD SEG9;
18	PT5.4	I/O	IO;
	SPI_SCK	I/O	SPI 通信时钟口;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINP5	AI	ADC 通道 5 的 P 端输入;
	SEG10	AO	LCD SEG10;
19	PT1.4	I/O	IO;
	PDA	I/O	烧录数据口;
	I2C_SDA	I/O	I2C 数据口;
	PWM_BKIN	I	PWM 暂停控制信号输入;
	INT1	I	外部中断 1 输入, 具有唤醒功能;
	AINN0	AI	ADC 通道 0 的 N 端输入;
	SEG1	AO	LCD SEG1;
20	PT1.0	I/O	IO 口;
	PCL	I	烧录时钟口;
	SPI_SS	I/O	SPI 从机选择;
	PWM4	O	PWM4 输出;

	INT2	I	外部中断 2;
	LVD	I	低电压检测电压点输入;
	AINP0	AI	ADC 通道 0 的 P 端输入;
	SEG0	AO	LCD SEG0;
21	VDD	P	电源
22	GND	P	地
	PT1.1	I/O	IO;
23	INT1	I	外部中断 1 输入, 具有唤醒功能;
	XOUT	O	外置晶振输出;
	AINN12	AI	ADC 通道 12 的 N 端输入;
	COM0	AO	LCD COM0;
	PT1.2	I/O	IO;
24	INT1	I	外部中断 1 输入, 具有唤醒功能;
	XIN	I	外置晶振输入;
	AINP12	AI	ADC 通道 12 的 P 端输入;
	COM1	AO	LCD COM1;

2 标准功能

2.1 CPU 核

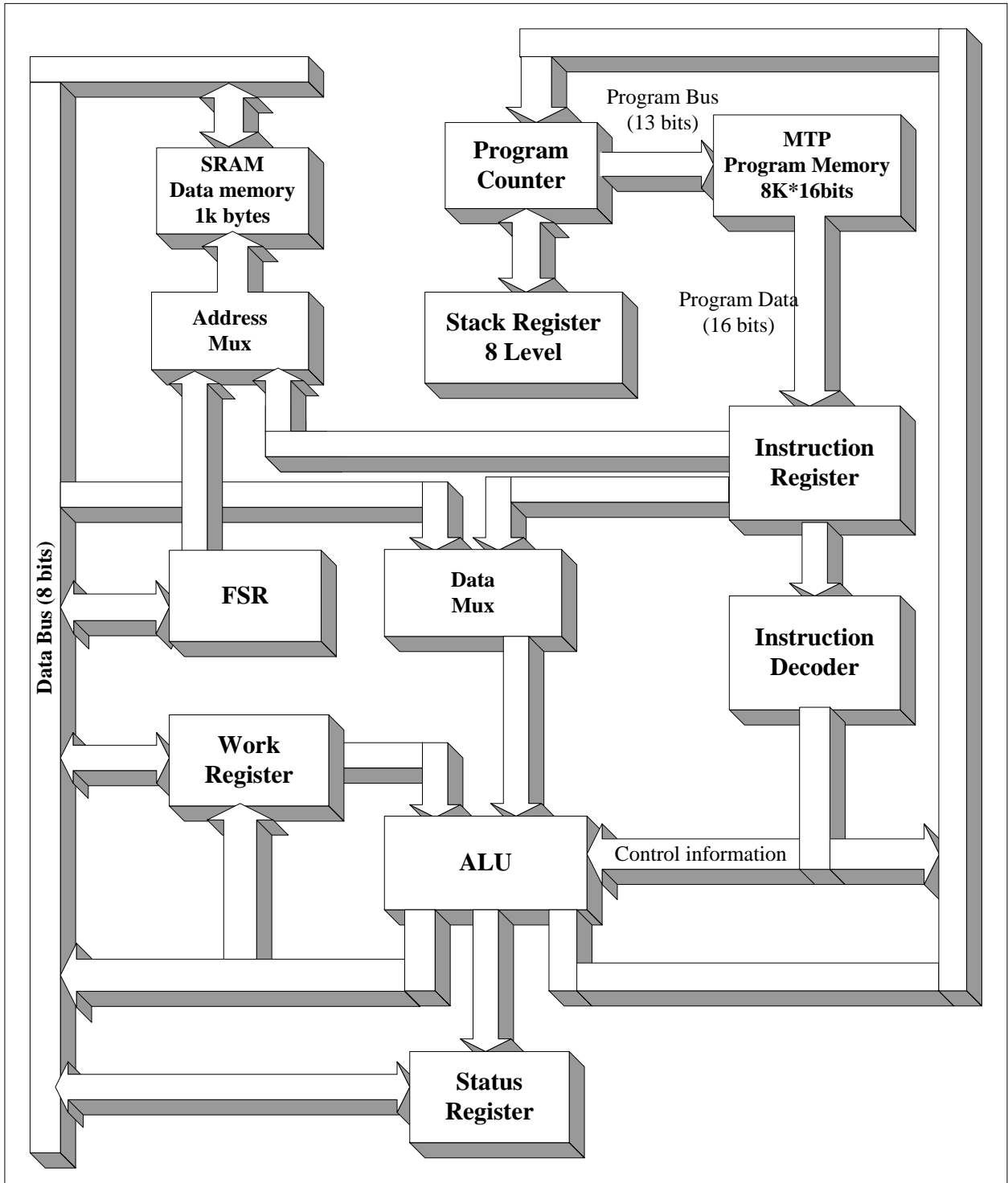


图12 CSU38M20 CPU 核的功能模块图

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 1 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（13bits），指令指针推送到程序存储器，然后自动加 1 以进行下一次周期。
堆栈寄存器	堆栈寄存器是用来记录程序返回的指令指针。当程序调用函数，程序计数器会将指令指针推送到堆栈寄存器。在函数执行结束之后，堆栈寄存器会将指令指针送回到程序计数器以继续原来的程序处理。
指令寄存器	<p>程序计数器将指令指针（程序存储器地址）推送到程序存储器，程序存储器将程序存储器的数据（16bits）推送到指令寄存器。</p> <p>CSU38M20 的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。</p> <p>直接地址（8bits）：数据存储器的地址。CPU 能利用此地址来对数据存储器进行操作。</p> <p>立即数（8bits）：CPU 通过 ALU 利用此数据对工作寄存器进行操作。</p> <p>控制信息：它记录着 ALU 的操作信息。</p>
指令译码器	指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。
算术逻辑单元	算术逻辑单元不仅能完成 8 位二进制的加，减，加 1，减 1 等算术计算，还能对 8 位变量进行逻辑的与，或，异或，循环移位，求补，清零等逻辑运算。
工作寄存器	工作寄存器是用来缓存数据存储器中某些存储地址的数据。
状态寄存器	当 CPU 利用 ALU 处理寄存器数据时，如下的状态会随着如下顺序变化：PD，TO，DC，C 及 Z。
文件选择寄存器	CSU38M20 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	CSU38M20 内带 8K×16 位的 MTP 作为程序存储器。由于指令的操作码（OPCODE）是 16bits，用户最多能编程 8K 的指令。程序存储器的地址总线是 13bits，数据总线是 16bits。
数据存储器	CSU38M20 内带 1K bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 10bits，数据总线是 8bits。

2.1.1 存储器

2.1.1.1 程序存储器

程序存储器主要用于指令的存储，在 CSU38M20 中，该程序存储器是 8K\*16bit 的程序 MTP。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。

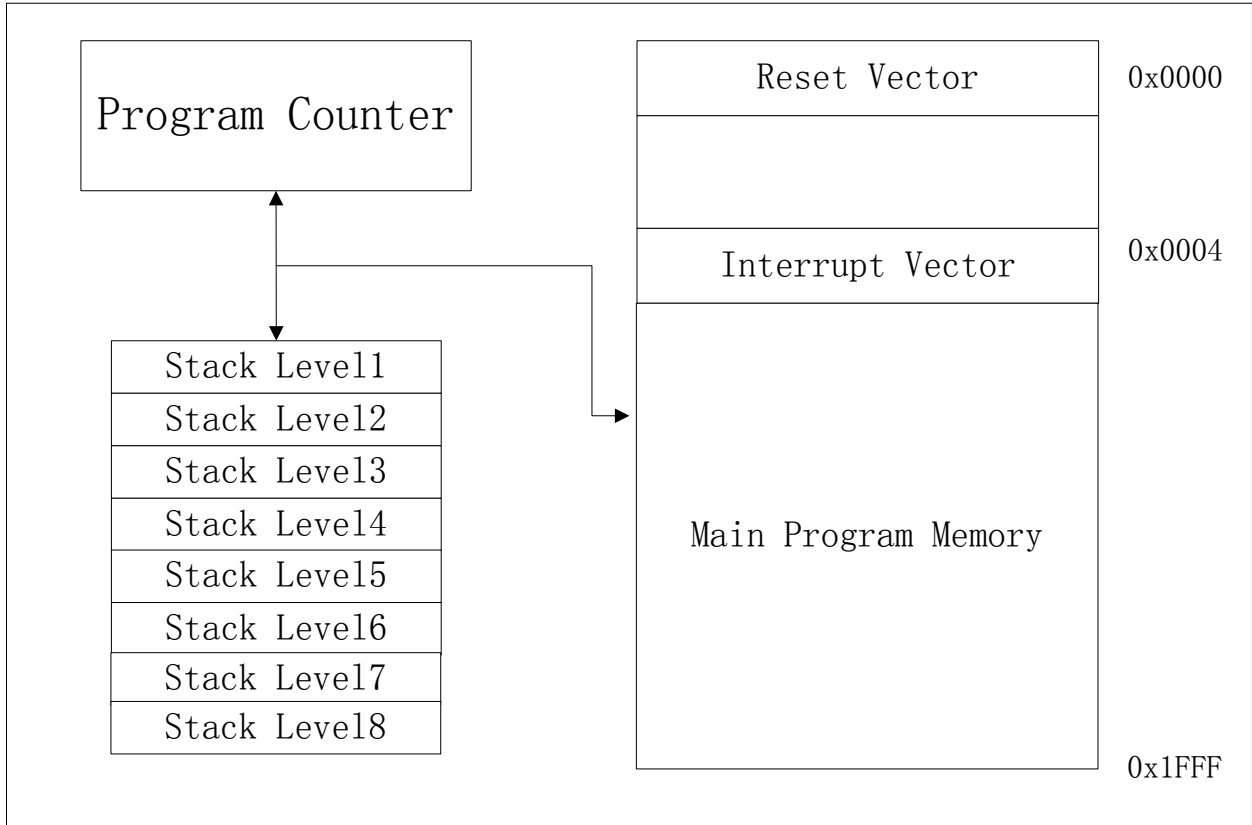


图13 程序存储器

2.1.1.2 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。

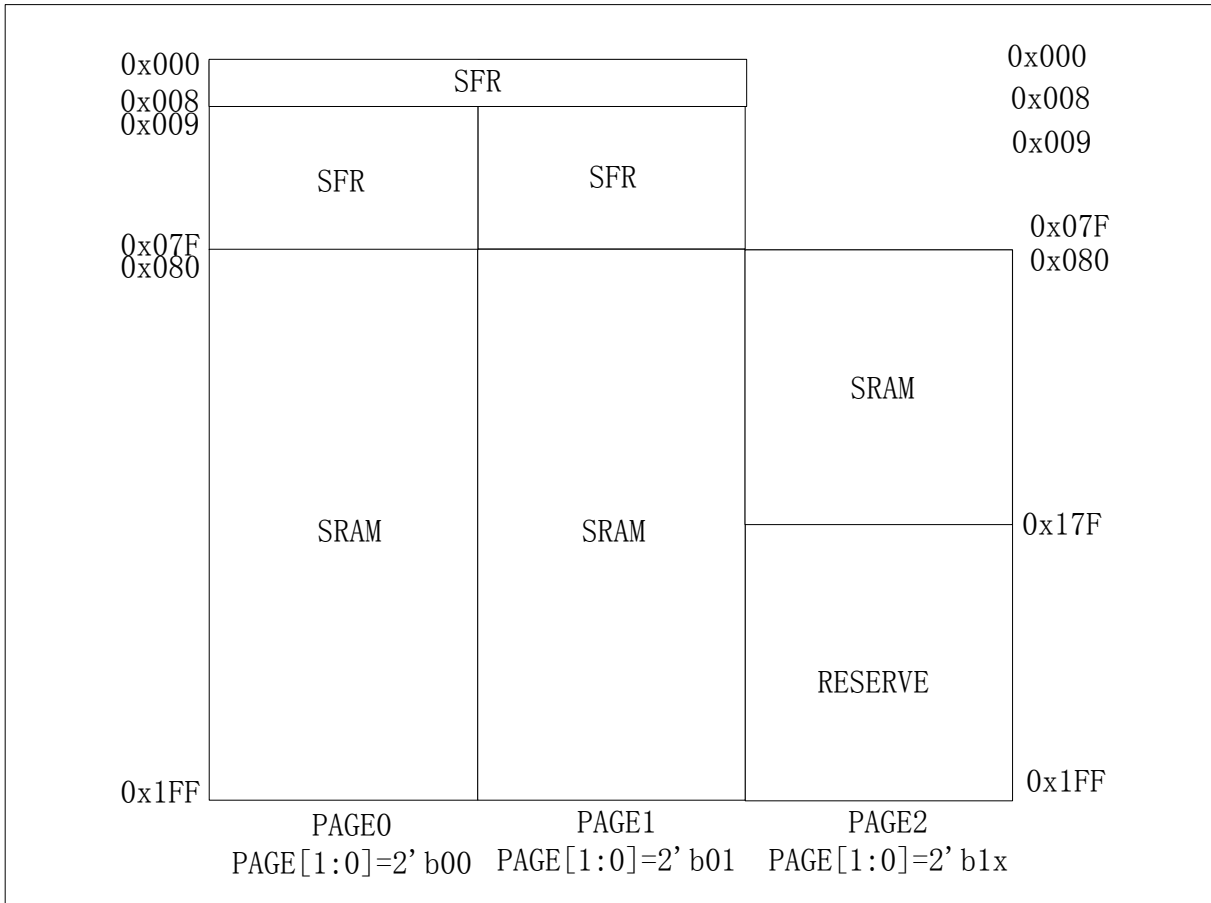


表 2 数据存储器地址分配

数据存储器	起始地址	结束地址
系统特殊功能寄存器	0x00	0x08
外设特殊功能寄存器 PAGE0	0x09	0x7F
外设特殊功能寄存器 PAGE1	0x09	0x7F
通用数据存储器 PAGE0	0x80	0x1FF
通用数据存储器 PAGE1	0x80	0x1FF
通用数据存储器 PAGE2	0x80	0x17F

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

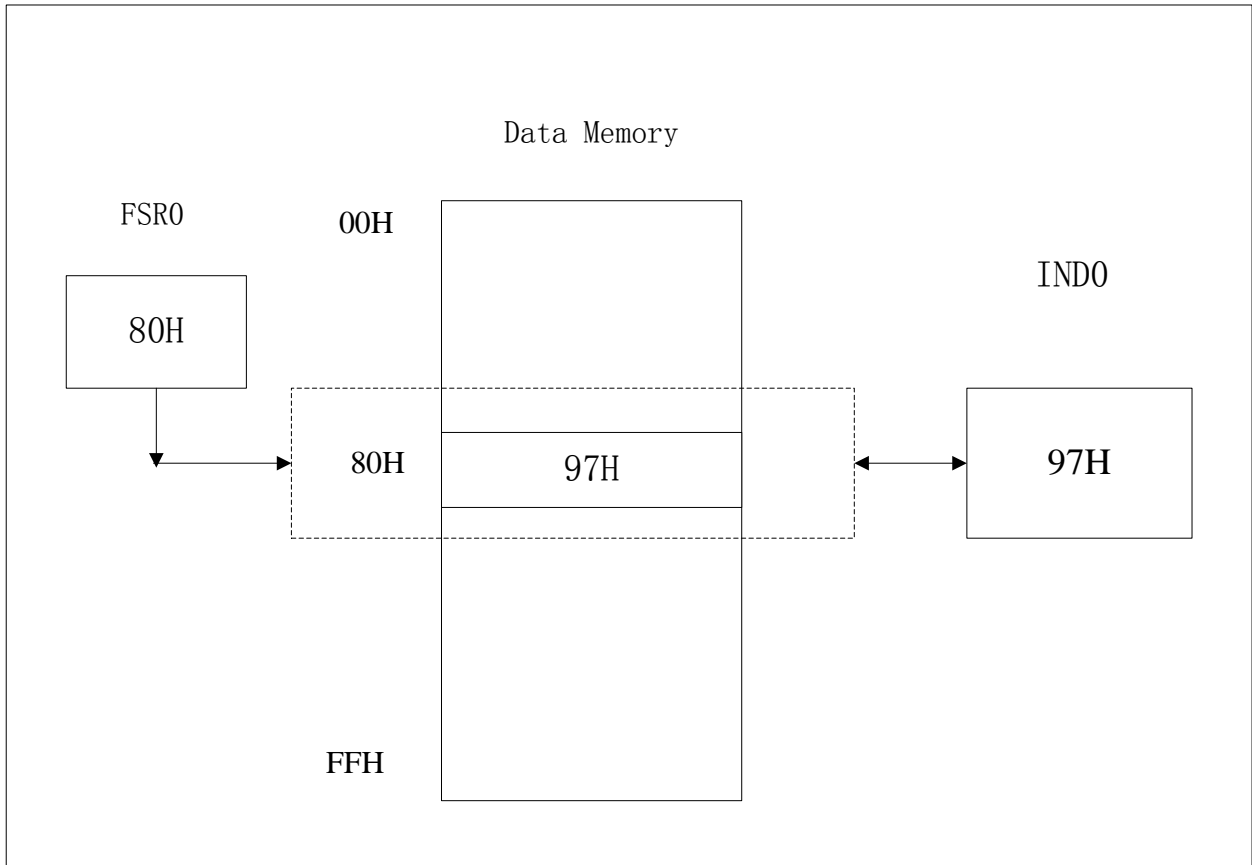


图14 间接地址访问

**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					Page[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

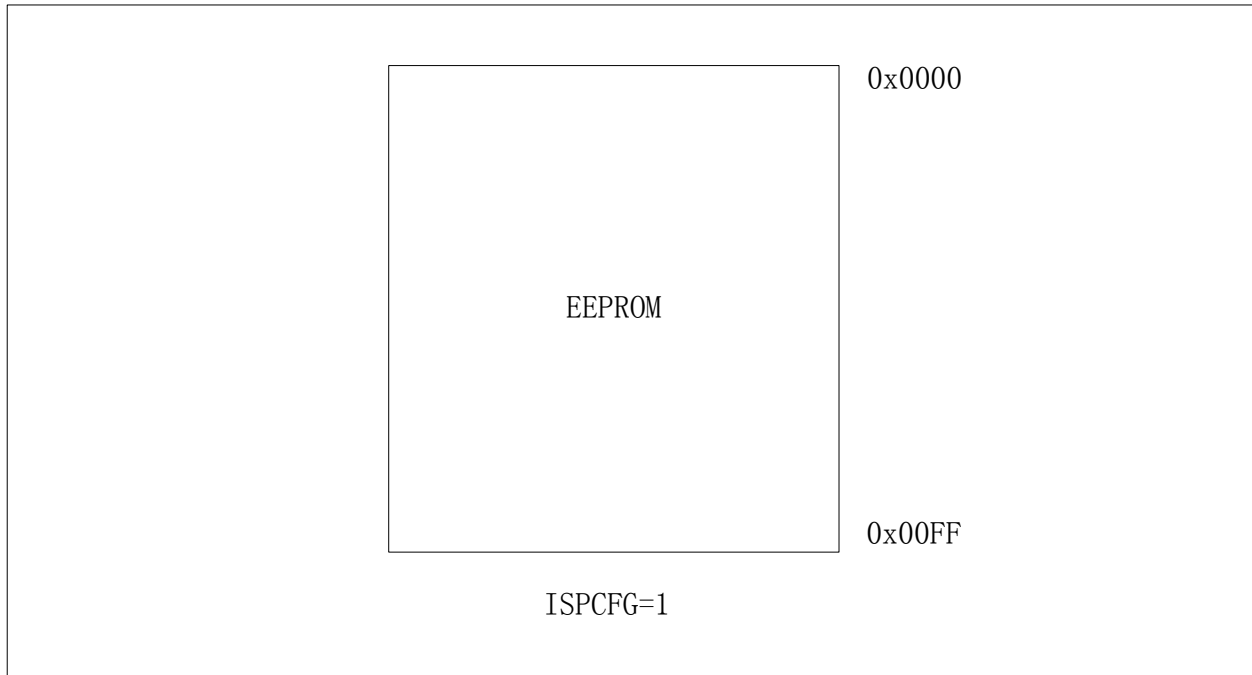
Bit 7 IRP0: IND0 间接页寻址位  
 1 = 间接寻址 IND0 时，访问后 256byte 地址  
 0 = 间接寻址 IND0 时，访问前 256byte 地址

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 1:0 PAGE[1:0]: PAGE 页选择  
 00 : 直接寻址和间接寻址访问 PAGE0  
 01 : 直接寻址和间接寻址访问 PAGE1  
 1x : 直接寻址和间接寻址访问 PAGE2

**2.1.1.3 EERPOM 数据存储**

CSU38M20 内置 256 字节 EEPROM 存储器，擦写次数为 10000 次，写操作电压范围为 2.5V~5.5V。程序无法直接访问 EEPROM，只能通过 MOVPE 及 TBLPE 指令对 EEPROM 进行读写操作，要对 EEPROM 进行操作，必须将 ISPCON 寄存器的 ISPCFG 位置 1，EEPROM 存储器地址为 000h~0FFh。对 EEPROM 进行写操作必须解锁写保护，详细说明见章节 [ISP](#)。





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	LVD			PD	T0	DC	C	Z
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7 LVD: LVD 工作电压标志位
  - 1: 表明芯片工作电压低于用户配置的低电压检测电压
  - 0: 表明芯片工作电压高于用户配置的低电压检测电压
- Bit 4 PD: 掉电标志位。通过对此位写 0 清零，sleep 后置此位
  - 1: 执行 SLEEP 指令后
  - 0: 上电复位后或硬件复位或 CLRWDT 指令之后
- Bit 3 T0: 看门狗定时溢出标志。通过对此位写 0 清零，看门狗定时溢出设置此位
  - 1: 看门狗定时溢出发生
  - 0: 上电复位后或硬件复位或 CLRWDT 指令后或 SLEEP 指令后
- Bit 2 DC: 半字节进位标志/借位标志
  - 用于借位时，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

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

2.1.3 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 3 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
PAGE0										
00h	IND0	以 FSR0 中内容作为地址的数据存储器中的数据								xxxxxxxx
01h	IND1	以 FSR1 中内容作为地址的数据存储器中的数据								xxxxxxxx
02h	FSR0	间接数据存储器的地址指针 0								00000000
03h	FSR1	间接数据存储器的地址指针 1								00000000
04h	STATUS	LVD			PD	TO	DC	C	Z	xxx00000
05h	WORK	工作寄存器								00000000
06h	INTF		TM2IF		TMOIF	SRADIF	E2IF	E1IF	E0IF	u0u00000
07h	INTE	GIE	TM2IE		TMOIE	SRADIE	E2IE	E1IE	E0IE	00u00000
08h	BSR	IRP0	IRP1					PAGE[1:0]		00uuuu00
09h	ISPCON	CHKRSLT							ISPCFG	uuuuuuu0
0Ah	EADRH						PAR[12:8]			uuu00000
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	PCLOADL				PCLOAD[7:0]					00000000
13h	PCLOADH						PCLOAD[12:8]			uuu00000
14h	PCWRPROT				PCWRPROT[7:0]					00000000
15h	XTALCON	XTAL_GM								0uuuuuuu
16h	MCK	CST	CST_IN	CST_WDT	EO_SLP	CPUCLK_SEL[1:0]		CLKSEL[1:0]		10001100
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	TM2INH						TM2IN[11:8]			uuuu1111
1ch	TM2CNTH						TM2CNT[11:8]			uuuu0000
1dh	TM2RH						TM2R[11:8]			uuuu0000
1eh	TM2CON1	PWM2IS				T2SEL[1:0]		PT34PWM[1:0]		0uuu0000
1Fh	INTCFG					E2M	E1M	EOM[1:0]		uuuu0000
20h	PT1				PT1[7:0]					xxxxxxxx
21h	PT1EN				PT1EN[7:0]					00000000
22h	PT1PU				PT1PU[7:0]					00000000
23h	PT1PD				PT1PD[7:0]					00000000
24h	PT1INT				PT1INT[7:0]					00000000
25h	PT2INT	PT2INT[7:6]								00uuuuuu
26h	PT3INT				PT3INT[7:0]					00000000
27h	PT5INT				PT5INT[7:0]					00000000
28h	PT1CON				PT1CON[7:0]					00000000
29h	PT1TYPE				PT1TYPE[7:0]					00000000
2ah	PT2	PT2[7:6]								xxxxxxxx

2bh	PT2EN	PT2EN[7:6]								00uuuuuu	
2ch	PT2PU	PT2PU[7:6]								00uuuuuu	
2dh	PT2PD	PT2PD[7:6]								00uuuuuu	
2eh	PT2TYPE	PT2TYPE[7:6]								00uuuuuu	
2fh	PT2CON	PT2CON[7:6]								00uuuuuu	
30h	PT3	PT3[7:0]								xxxxxxxx	
31h	PT3EN	PT3EN[7:0]								00000000	
32h	PT3PU	PT3PU[7:0]								00000000	
33h	PT3PD	PT3PD[7:0]								00000000	
34h	PT3CON	PT3CON[7:0]								00000000	
35h	PT3TYPE	PT3TYPE[7:0]								00000000	
36h	PT5	PT5[7:0]								xxxxxxxx	
37h	PT5EN	PT5EN[7:0]								00000000	
38h	PT5PU	PT5PU[7:0]								00000000	
39h	PT5PD	PT5PD[7:0]								00000000	
3ah	PT5TYPE	PT5TYPE[7:0]								00000000	
3bh	PT5CON	PT5CON[7:0]								00000000	
3ch	INTF2	LVDIF	TM4IF	TM3IF	CAPIF	BKIF	BKNAIF			u000000u	
3dh	INTE2	LVDIE	TM4IE	TM3IE	CAPIE	BKIE	BKNAIE			u000000u	
3eh	INTF3	CMPOIF	MULIF	COGOIF	I2CIF	SPIIF	URTIF	URRIF		00000000	
3fh	INTE3	CMPOIE	MULIE	COGOIE	I2CIE	SPIIE	URTIE	URRIE		00000000	
40h	TM3CON	T3EN	T3RATE[2:0]		T3CKS	T3RSTB	T3OUT	PWM3OUT		00000100	
41h	TM3IN	TM3IN[7:0]								11111111	
42h	TM3CNT	TM3CNT[7:0]								00000000	
43h	TM3R	TM3R[7:0]								00000000	
44h	TM3INH	TM3IN[11:8]								uuuu1111	
45h	TM3CNTH	TM3CNT[11:8]								uuuu0000	
46h	TM3RH	TM3R[11:8]								uuuu0000	
47h	TM3CON1	PT17PWM[1:0]		PT27PWM[1:0]		T3SEL[1:0]		PT33PWM[1:0]		00000000	
48h	TM3CON2	SIGN	TM3PNT_DUTY[6:0]								00000000
49h	TM3CON3	PNT_EN			PWM3IS	DT3CNT2[2:0]				0uuu0000	
4ah	DT3CON0	DT3_EN	DT3CK[1:0]		P3L_IS	P3H_IS	DT3CNT1[2:0]			00000000	
4bh	DT3CON1		P3LOEN	P3LINV	P3HOEN	P3HINV	PWM3DL1[2:0]			u0000000	
4ch	DT2CON0	DT2_EN	DT2CK[1:0]		P2L_IS	P2H_IS	DT2CNT1[2:0]			00000000	
4dh	DT2CON1		P2LOEN	P2LINV	P2HOEN	P2HINV	PWM2DL1[2:0]			u0000000	
4eh	DT2CON2						DT2CNT2[2:0]			uuuuuu000	
4fh	ISPCON2	TBLP_CLK_DIV[7:0]								00000000	
50h	TM4CON	T4EN	T4RATE[2:0]		T4CKS	T4RSTB	T4OUT	PWM4OUT		00000100	
51h	TM4IN	TM4IN[7:0]								11111111	
52h	TM4INH	TM4INH[15:8]								11111111	
53h	TM4CNT	TM4CNT[7:0]								00000000	
54h	TM4CNTH	TM4CNTH[15:8]								00000000	
55h	TM4R	TM4R[7:0]								00000000	
56h	TM4RH	TM4RH[15:8]								00000000	
57h	TM4CON1	PT16PWM[1:0]		PT32PWM[1:0]		PT53PWM[1:0]		PT55PWM[1:0]		00000000	
58h	TM4CON2	PWM4IS	CLK_INV4	T4SEL[1:0]						0u0u00uu	
59h	DT4CON0	DT4_EN	DT4CK[1:0]		P4L_IS	P4H_IS	DT4CNT1[2:0]			00000000	
5ah	DT4CON1	PWM4PO	P4LOEN	P4LINV	P4HOEN	P4HINV	PWM4DL1[2:0]			00000000	
5bh	DT4CON2						DT4CNT2[2:0]			00000000	
5ch	CAPCON	CAPEN	CAPPIN[2:0]			CAPCFG	CAPFLT[1:0]			0000u000	
5dh	CAPCON2						CAPMODE[1:0]			uuuuuu00	
5eh	TM4CAPL	TM4CAP[7:0]								00000000	
5fh	TM4CAPH	TM4CAP[15:8]								00000000	
60h	SRADCON0	SOURCE_MOD[1:0]		SRADACKS[1:0]		DMOF	CMPO_R	SRADCKS[1:0]		00000000	
61h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]		00000000	



62h	SRADCON2	CHS[4:0]				AUTOS	REF_SEL[1:0]			00000000
63h	SRADCON3	SRADSMOD[2:0]		SOURCE_SEL[1:0]		SRADSDLY[2:0]			00000000	
64h	SRADCON4	AINSEL[2:0]		IO_REF_EN[4:0]				00000000		
66h	SRADL	SRAD[7:0]							00000000	
67h	SRADH	SRAD[11:8]							uuuu0000	
68h	SROFTL	SROFT[7:0]							00000000	
69h	SROFTH	SROFT[11:8]							uuuu0000	
6ah	CMPCON0	CMPOEN	CMP_PSEL[2:0]			CMP_NSEL[2:0]		CMPOUT0	00000000	
6bh	CMPCON1	CMP_MODE	CMP_IO_SE L	CMP_IO_EN	CMPOFST_SEL[4:0]				00010000	
6ch	PTVTH				PT5VTH	PT3VTH	PT2VTH	PT1VTH	uuuu0000	
6dh	WRPRT	REGWRPROT[7:0]							00000000	
6eh	PWMBKR	PWMOEB	ATUOE	BKIPH	BKIEN	BKMOD[2:0]			0000u000	
70h	IODRV	BGDRV7	BGDRV6	BGDRV5	BGDRV4	BGDRV3	BGDRV2	BGDRV1	BGDRV0	00000000
71h	COGOCON0	GOEN	GOOE1	GOOE0	GOPOL1	GOPOL0	GOCS[1:0]		00000u00	
72h	COGOCON1	GOFSIM	GORSIM	GOFSS[2:0]		GORS[2:0]			00000000	
73h	COGOASD	GOASDE	GOARSEN	GOASDL1	GOASDL0	GOASDSC3	GOASDSC2	GOASDSC1	GOASDSC0	00000000
74h	COGODB	GODBF[3:0]			GODBR[3:0]			00000000		
75h	COGOBLK	GOBLKF[3:0]			GOBLKR[3:0]			00000000		
76h	COGOPH				GOPH[3:0]			uuuu0000		
77h	LVRCON	LVD_SEL[2:0]		LV DEN	LVDCHK	LVR_FLT[2:0]			10010000	
78h	SCON1	SM0	SM1	SM2	REN	TB8	RB8	URTIOS	URRIOS	00000000
79h	SCON2	SMOD	TCLKS[1:0]		RCLKS[1:0]		URIOS	UARTEN		000000u0
7ah	SBUF	SBUF[7:0]							00000000	
7dh	METCH	METCH[7:0]							00000000	
PAGE1										
1ah	CRCDL	CRC_DATA[7:0]							00000000	
1bh	CRCINITL	CRC_INIT[7:0]							11111111	
1ch	CRCINITH	CRC_INIT[15:8]							11111111	
1dh	ISPCRC	CRC16EN				CRC_LEN[3:0]			0uuu0000	
1eh	ISPCRCDL	ISP_CRC_DATA[7:0]							11111111	
1fh	ISPCRCDH	ISP_CRC_DATA[15:8]							11111111	
20h	CTRL	i2c_en	i2c_ios	WAIT	STO	SOFTTRST	STA	TXEXPRG		000u0000
21h	RXDATA	RXDATA[7:0]							00000000	
22h	TXDATA	TXDATA[7:0]							00000000	
23h	PRESC	PRESC[7:0]							00000000	
24h	TXRX_STS				TX_DATA_NACK	SA_NACK	TX_UNFLW	RX_OVFLW	NORMAL_END	uuuu0000
25h	DATA_STS			TX_EMPTY	TX_FULL	RX_EMPTY	RX_FULL		uu1u01u0	
26h	TXRX_IE				ETX_DATA_NACK	ESA_NACK	ETX_UNFLW	ERX_OVFLW	ENORMAL_END	uuuu0000
28h	CALLADDRO	RN	CALLADDRO[6:0]						00000000	
2ah	RSTDISBL	RSTDISB[7:0]							xxxxxxxx	
2bh	RSTDISBH	RSTDISB[15:8]							xxxxxxxx	
30h	LCDCFG1	LCDEN	LCDWS	LCDDUTY[1:0]		LCDCKSEL[1:0]		LCDCKS[1:0]		00000000
31h	LCDCFG2	LCDREF[1:0]					LCDSCKS[3:0]			00000000
32h	LCDCOM	SEG_EN[7:0]							00000000	
33h	LCDCOM1	SEG_EN[15:8]							00000000	
34h	LCDCOM2	SEG21_COM4_EN[1:0]		SEG20_COM5_EN[1:0]		SEG_EN[19:16]			00000000	
35h	LCDCOM3								COM_EN[3:0]	uuuu0000
36h	LCDSEG0								SEG0[5:0]	uu000000
37h	LCDSEG1								SEG1[5:0]	uu000000
38h	LCDSEG2								SEG2[5:0]	uu000000
39h	LCDSEG3								SEG3[5:0]	uu000000
3ah	LCDSEG4								SEG4[5:0]	uu000000

3bh	LCDSEG5									SEG5[5:0]	uu000000
3ch	LCDSEG6									SEG6[5:0]	uu000000
3dh	LCDSEG7									SEG7[5:0]	uu000000
3eh	LCDSEG8									SEG8[5:0]	uu000000
3fh	LCDSEG9									SEG9[5:0]	uu000000
40h	LCDSEG10									SEG10[5:0]	uu000000
41h	LCDSEG11									SEG11[5:0]	uu000000
42h	LCDSEG12									SEG12[5:0]	uu000000
43h	LCDSEG13									SEG13[5:0]	uu000000
44h	LCDSEG14									SEG14[5:0]	uu000000
45h	LCDSEG15									SEG15[5:0]	uu000000
46h	LCDSEG16									SEG16[5:0]	uu000000
47h	LCDSEG17									SEG17[5:0]	uu000000
48h	LCDSEG18									SEG18[5:0]	uu000000
49h	LCDSEG19									SEG19[5:0]	uu000000
4Ah	LCDSEG20									SEG20[5:0]	uu000000
4Bh	LCDSEG21									SEG21[5:0]	uu000000
50h	MULCTL	MULBG									0uuuuuuu
51h	MULDA0L	MUL_DATA0[7:0]									00000000
52h	MULDA1L	MUL_DATA 1[7:0]									00000000
6ch	SPICFG	SPIEN	MSTEN	CKPHA	CKPOL	MULMST	WIREMOD	SPIBSY	SPIRST		00000000
6dh	SPICN	SLVSEL	WCOL	MODCOL	SPIIOS						0000uuuu
6eh	SPICKR	SCR[7:0]									00000011
6fh	SPIDAT	SPIDAT[7:0]									00000000

注：进行读操作时，无效位读出为 0

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

## 2.2 时钟系统

### 2.2.1 概述

芯片的时钟系统包括内置 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。**选择 4MHz 指令周期时，必须开启 2.4V 或以上低电压复位，否则在低于 2.4V 时会出现程序跑飞问题。**

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

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

### 2.2.2 时钟框图

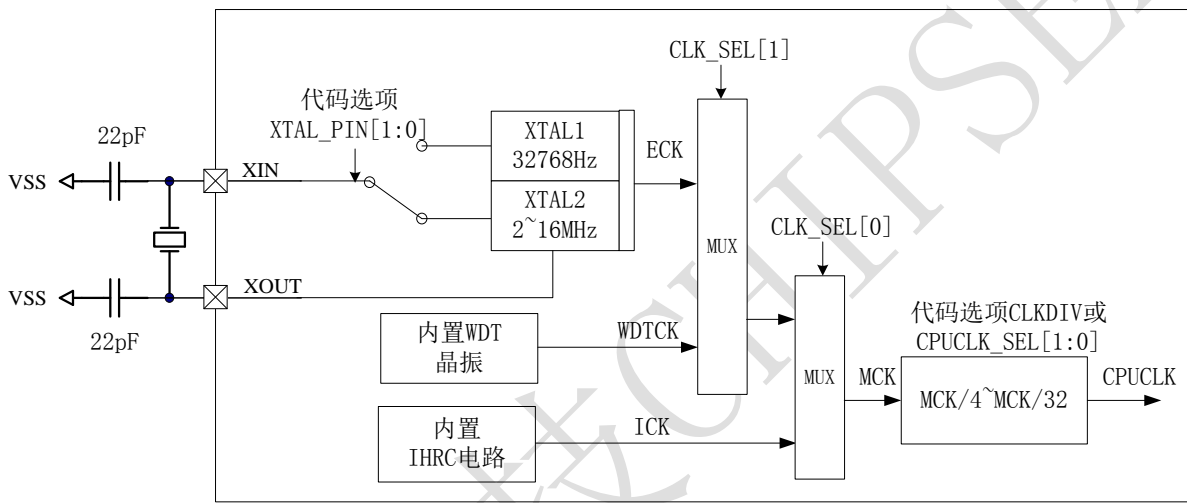


图15 CSU38M20 振荡器状态框图 A

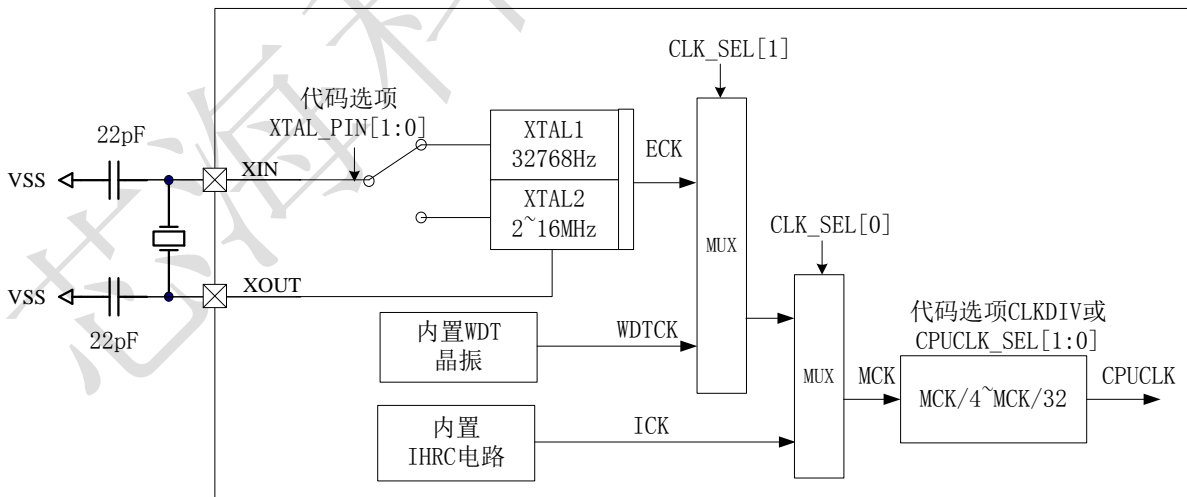


图16 CSU38M20 振荡器状态框图 B

### 2.2.3 时钟符号说明

符号	说明
ICK	内置高速时钟
ECK	外部高速时钟或者外部 32768Hz 时钟
WDTCK	内置低速 32KHz WDT 时钟
MCK	使用 CLKSEL[1:0]寄存器选择后的时钟源，未进行任何分频
CPUCLK	MCK 经过分频后作为 CPU 主时钟，可以进行 4/8/16/32 分频，分频值由代码选项或者 CPUCLK_SEL[1:0]决定。
Fosc	使用 CLKSEL[1:0]寄存器选择后的时钟源频率，未进行任何分频
Fcpu	CPUCLK 频率，与指令周期相对应

2.2.4 寄存器

MCK 为写保护寄存器，必须解锁写保护才能对该寄存器进行写操作。解锁写保护需对 WRPRT 寄存器连续写入 C9H, 3AH, D3H, 解锁后 16 个指令周期内都能对 MCK 进行写操作，16 个指令周期后解锁自动失效，不能再进行写操作。

表 4 CSU38M20 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
15h	XTALCON	XTAL_GM								0uuuuuuu
16h	MCK	CST	CST_IN	CST_WDT	EO_SLP	CPUCLK_SEL[1:0]	CLKSEL[1:0]			10001100

表 5 XTALCON 寄存器（地址为 15h）各位功能表

特性	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
XTALCON	XTAL_GM							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	XTAL_GM	外置高速晶振跨导控制位 1: 使能外部高速晶振跨导 0: 禁止外部高速晶振跨导 仅当外部高速晶振使能时有效

表 6 MCK 寄存器（地址为 16h）各位功能表

特性	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
MCK	CST	CST_IN	CST_WDT	EO_SLP	CPUCLK_SEL[1:0]	CLKSEL[1:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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"> <thead> <tr> <th>CPUCLK_SEL[1:0]</th> <th>指令周期</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>指令周期=4 个时钟周期</td> </tr> <tr> <td>01</td> <td>指令周期=8 个时钟周期</td> </tr> <tr> <td>10</td> <td>指令周期=16 个时钟周期</td> </tr> <tr> <td>11</td> <td>指令周期=32 个时钟周期</td> </tr> </tbody> </table>	CPUCLK_SEL[1:0]	指令周期	00	指令周期=4 个时钟周期	01	指令周期=8 个时钟周期	10	指令周期=16 个时钟周期	11	指令周期=32 个时钟周期
CPUCLK_SEL[1:0]	指令周期											
00	指令周期=4 个时钟周期											
01	指令周期=8 个时钟周期											
10	指令周期=16 个时钟周期											
11	指令周期=32 个时钟周期											





		由于不支持 8Mhz 指令周期，故使用 32Mhz 内部晶振时，不能选择指令周期=4 个时钟周期。												
1:0	CLKSEL[1:0]	<p>时钟源选择位</p> <table border="1"> <tr> <td>CLK_SEL[1]</td> <td>时钟源 2 选择</td> </tr> <tr> <td>0</td> <td>外部晶振系统时钟</td> </tr> <tr> <td>1</td> <td>内部 WDT 晶振时钟</td> </tr> </table> <table border="1"> <tr> <td>CLK_SEL[0]</td> <td>CPU 时钟</td> </tr> <tr> <td>0</td> <td>使用时钟源 1(内部高速晶振时钟)作为 CPU 时钟</td> </tr> <tr> <td>1</td> <td>使用时钟源 2 作为 CPU 时钟</td> </tr> </table> <p>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)</p> <p>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</p>	CLK_SEL[1]	时钟源 2 选择	0	外部晶振系统时钟	1	内部 WDT 晶振时钟	CLK_SEL[0]	CPU 时钟	0	使用时钟源 1(内部高速晶振时钟)作为 CPU 时钟	1	使用时钟源 2 作为 CPU 时钟
CLK_SEL[1]	时钟源 2 选择													
0	外部晶振系统时钟													
1	内部 WDT 晶振时钟													
CLK_SEL[0]	CPU 时钟													
0	使用时钟源 1(内部高速晶振时钟)作为 CPU 时钟													
1	使用时钟源 2 作为 CPU 时钟													

系统时钟源切换示意图，其中，不能直接从内置 WDT 晶振到外部晶振，也不能直接从外部晶振切换到内置 WDT 晶振。

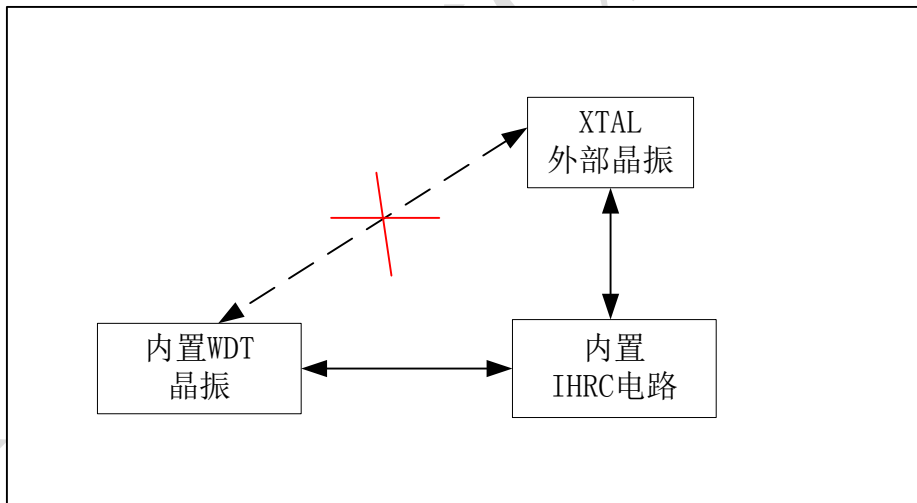


图17 系统时钟源切换示意图

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

### 2.2.5 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。

### 2.2.6 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。

### 2.2.7 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。

### 2.2.8 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。

2.3 复位系统

CSU38M20 有以下方式复位：

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

下图给出了复位电路原理图。

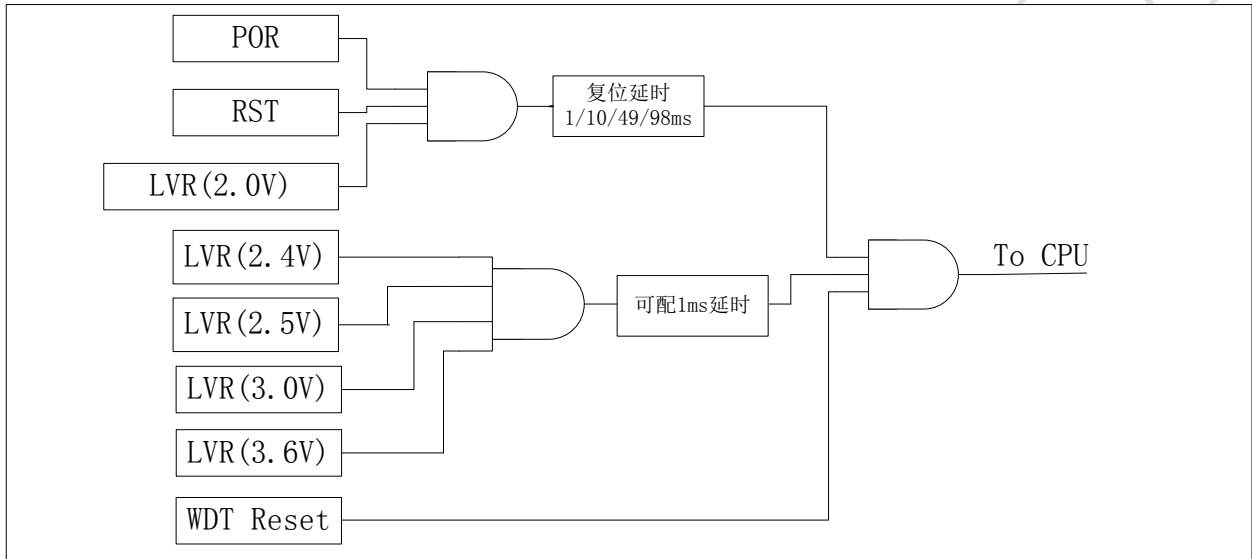


图18 复位电路原理图

上电复位、硬件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（1ms，10ms，49ms，98ms）。LVR 低电压复位 2.4V、2.5V、3.0V、3.6V 可配置 1ms 延时或者不延时，WDT 复位不延时，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

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。

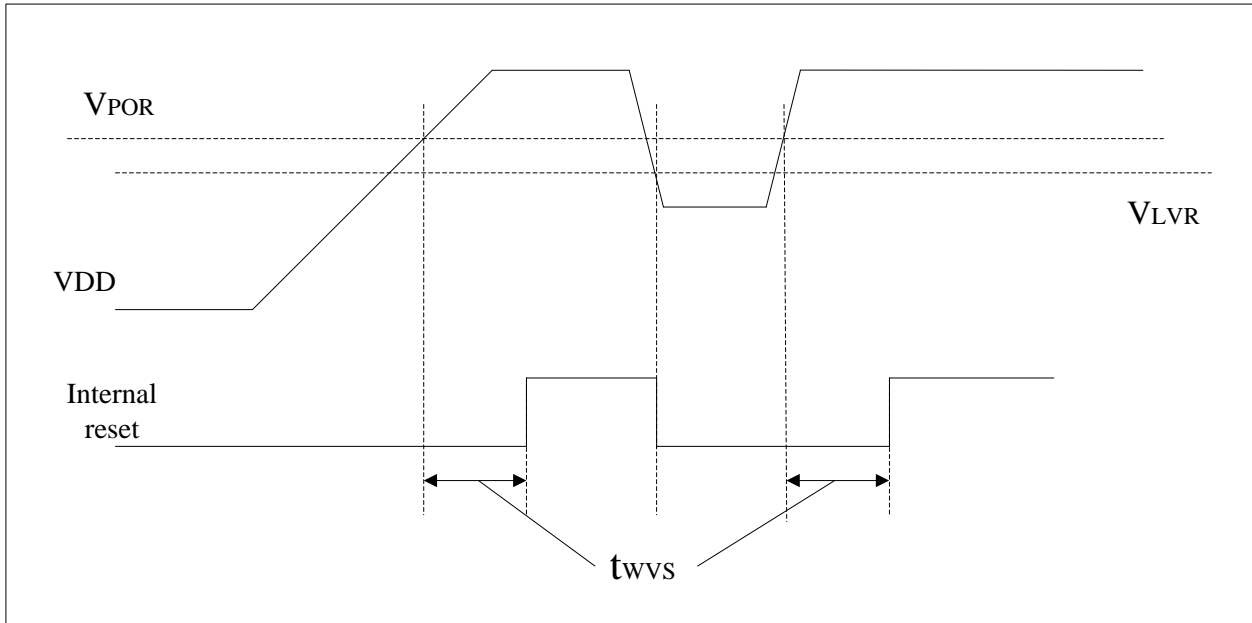


图19 上电复位电路示例及上电过程

参数	最小值	典型值	最大值
VPOR	1.7V	1.8V	1.9V
VLVR	1.7V	1.8V	1.9V
twvs (测试条件: VDD=5V, T=25°C)	450us	500us	550us

VPOR: 上电复位

VLVR: 低电压复位

twvs: 等待电压稳定时间

### 2.3.1 上电复位

系统上电呈现逐渐上升的曲线形式，需要一定时间才能达到正常的工作电压（对于不同的指令周期所需工作电压是不同的，指令周期越快相应所需的工作电压就越高，见 5.2 直流特性）。要求用户系统的上电速度要大于 0.07V/mS，尤其是要注意指令周期是 4MHz 时，因为他要求的工作电压最高。

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
PAGE1										
2ah	RSTDISBL	RSTDISB[7:0]								xxxxxxxx
2bh	RSTDISBH	RSTDISB[15:8]								xxxxxxxx

写保护寄存器，向 RSTDISBH、RSTDISBL 写入 35h、ACh 即可禁止上电复位

### 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
3dh	INTE2		LVDIE							u0000000
77h	LVRCON	LVD_SEL[2: 0]			LVDEN	LVDCHK	LVR_FLT[2: 0]			10010000

表 9 INTE2 寄存器各位功能表

位地址	标识符	功能
6	LVDIE	LVD 中断使能位 1：使能 LVD 中断 0：不使能 LVD 中断

表 10 LVRCON 寄存器（地址为 77h）各位功能表

特性	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
LVRCON	LVD_SEL[2: 0]			LVDEN	LVDCHK	LVD_FTL[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7: 5	LVD_SEL[2: 0]	LVD 配置
		LVD_SEL[2:0] 功能
		000 VDD 低于 2.0V 复位
		001 2.4V 低电压检测
		010 2.5V 低电压检测
		011 3.0V 低电压检测
		100 3.6V 低电压检测
		101 110 111 PT10 口输入电压检测，
4	LVDEN	LVD 使能 1'b0: 不使能 LVD 1'b1: 使能 LVD
3	LVDCHK	低电压检测功能选择： 1'b1: 低电压检测作为中断 1'b0: 低电压检测作为复位
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 3'b100: 低电压滤波时间 150us

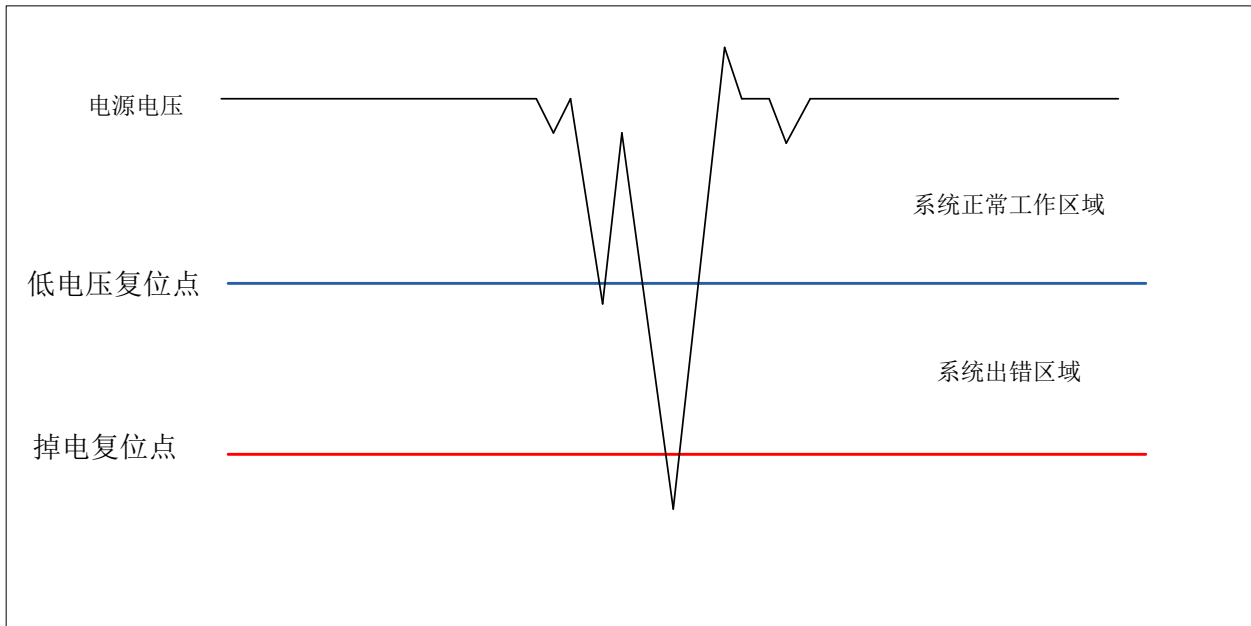


其他： 不进行低电压滤波
--------------

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。



电压跌落可能会进入系统死区。进入系统死区，即电源电压不能满足系统的最小工作电压要求。系统掉电复位示意图如上图所示。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。

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

不同指令周期的系统出错区域不同，取决于指令周期工作电压范围，见 5.2。掉电复位性能的改善可以通过如下几点实现：

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

#### 2.3.4 外部硬件复位

外部复位由代码选项 RESET\_PIN 控制，见 3.10。通过设置该代码选项，可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 PC 跳转

表 11 CSU38M20 PC 跳转寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
12h	PCLOADL	PCLOAD[7:0]								00000000
13h	PCLOADH				PCLOAD[12:8]					uuu00000
14h	PCWRPROT	PCWRPROT[7:0]								00000000

表 12 PCLOADL 寄存器（地址为 12H）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PCLOADL	PCLOAD[7:0]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

位地址	标识符	功能
7 : 0	PCLOAD[7:0]	PC 值低 8 位。

表 13 PCLOADH 寄存器（地址为 13H）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PCLOADH	PCLOAD[12:8]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

位地址	标识符	功能
7 : 0	PCLOAD[12:8]	PC 值高 5 位。

表 14 PCWRPROT 寄存器（地址为 14H）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PCWRPROT	PCWRPROT[7:0]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

位地址	标识符	功能
7 : 0	PCWRPROT[7:0]	PC 写保护寄存器，当需要对 PCLOADL、PCLOADH 寄存器进行写操作时，需要先对 PCWRPROT 寄存器连续写入 C9H, 3AH, D3H 进行解锁。

## 2.4.1 PC 跳转操作

PC 跳转寄存器为写保护寄存器，必须向 PCWRPROT 写入 C9H, 3AH, D3H 解锁写保护，才能对 PCLOADL、PCLOADH 进行写操作，并且不会失效，只有当用户再次写入其他值时，该寄存器才会再次被写保护。每次改变 PC 时，必须先写 PCLOADH，再写 PCLOADL，写完 PCLOADL，PC 将跳转到 PCLOADH、PCLOADL 给定的值。

## 2.4.2 PC 读操作

当要获取当前 PC 值时，可以通过读取 PCLOADL 读取 PC 值低 8 位，同时，PC 高位（高 5 位）将会被锁存在 PCLOADH 的读 BUFF 中，之后再读取 PCLOADH 即可获取完整的 13 位 PC 值，可以理解为 PCLOADH 与 PCLOADL 联动。对 PCLOADL，PCLOADH 进行读操作，读回的并不是向 PCLOADL 和 PCLOADH 写入的值，读写使用不同的 BUFF 进行数据缓存，因此读出数据与写入数据并无关联。



## 2.5 中断

CSU38M20 有 17 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 17 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 halt 停止模式。

外部中断、比较器中断、使用内部低速 WDT 时钟作为计时时钟的定时器中断可以唤醒 SLEEP 模式。

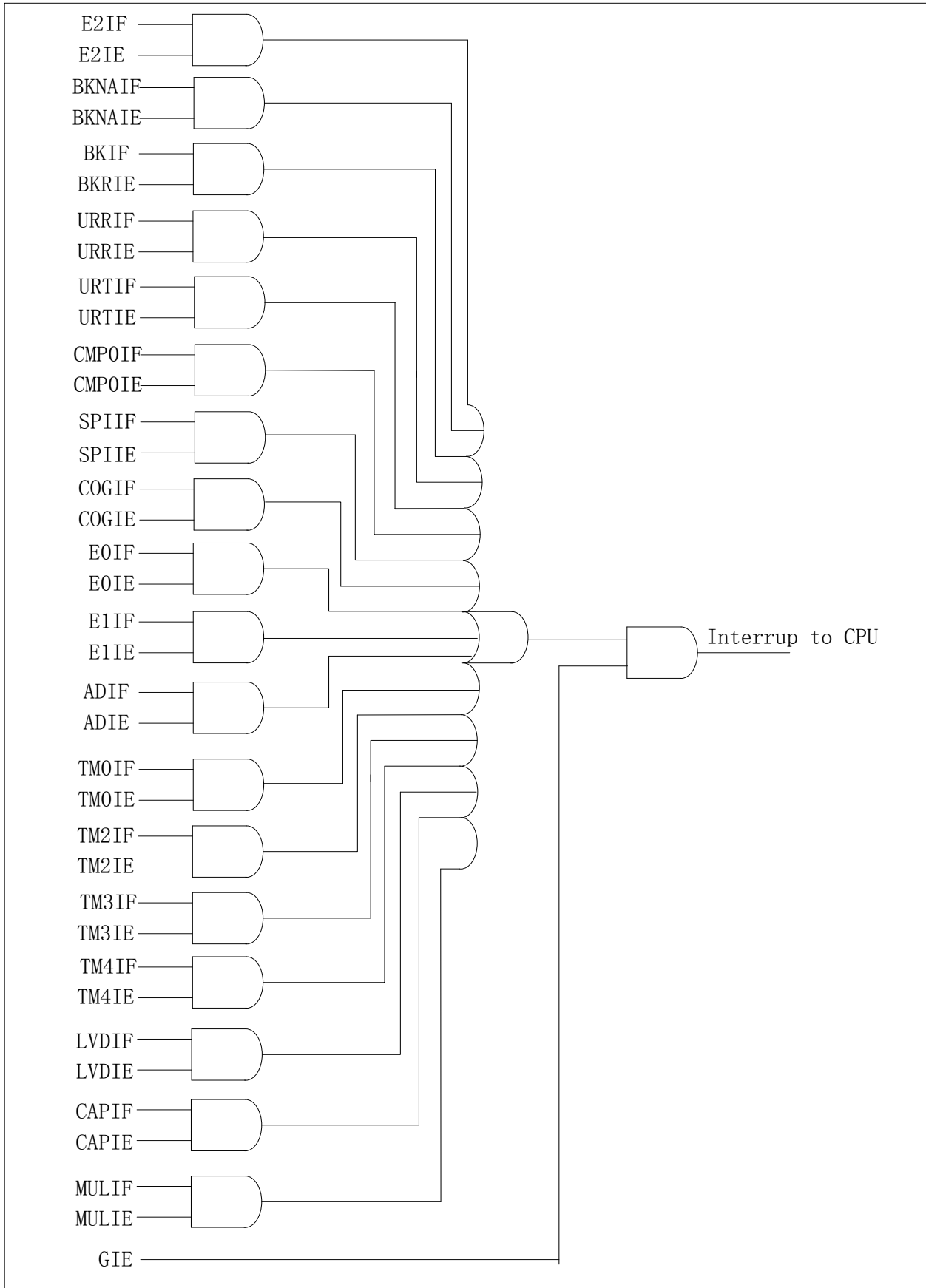


图21 中断逻辑

## 2.5.1 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	E2IE	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 2 E2IE: 外部中断 2 使能标志

1 = 使能外部中断 2

0 = 不使能外部中断 2

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		LVDIE	TM4IE	TM3IE	CAPIE	BKIE	BKNAIE	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 LVDIE: 低电压检测中断使能标志

1 = 使能低电压检测中断

0 = 不使能低电压检测中断

Bit 5 TM4IE: 16-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 CAPIE: 定时器 4 捕捉中断使能标志

1 = 使能定时器 4 捕捉中断

0 = 不使能定时器 4 捕捉中断

Bit 2 BKIE: PWM 暂停中断使能标志

1 = 使能 PWM 暂停中断

0 = 不使能 PWM 暂停中断

Bit 1 BKNAIE: PWM 暂停失效中断使能标志



1 = 使能 PWM 暂停失效中断  
0 = 不使能 PWM 暂停失效中断

**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	CMP0IE	MULIE		COG0IE	I2CIE	SPIIE	URTIE	URRIE
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7 **CMP0IE**: 比较器 0 中断使能标志  
1 = 使能比较器 0 中断  
0 = 不使能比较器 0 中断
- Bit 6 **MULIE**: 乘法器中断使能标志  
1 = 使能乘法器中断  
0 = 不使能乘法器中断
- Bit 4 **COG0IE**: 互补输出发生器中断使能标志  
1 = 使能互补输出发生器中断  
0 = 不使能互补输出发生器中断
- Bit 3 **I2CIE**: I2C 中断使能标志  
1 = 使能 I2C 中断  
0 = 不使能 I2C 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

2.5.2 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	R/W -0	R/W -0	R/W -0
INTF		TM2IF		TM0IF	SRADIF	E2IF	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 2 E2IF: 外部中断 2 中断标志，软件清零，硬件置高  
1 = 外部中断 2 发生中断，必须软件清 0  
0 = 外部中断 2 没发生中断
- 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	CAPIF	BKIF	BKNAIF	
	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 = 发生低电压检测，必须软件清 0  
0 = 没发生低电压检测中断
- Bit 5 TM4IF: 16-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
- Bit 3 CAPIF: 定时器 4 捕捉中断标志位，软件清零，硬件置高  
1 = 发生定时器 4 捕捉中断  
0 = 没有定时器 4 捕捉中断
- Bit 2 BKIF: PWM 暂停中断标志位  
1 = 发生 PWM 暂停事件  
0 = 未发生 PWM 暂停事件
- Bit 1 BKNAIF: PWM 暂停失效中断使能标志  
1 = PWM\_BKIN 从有效翻转到失效状态  
0 = PWM\_BKIN 未从有效翻转到失效状态



**INTF3 寄存器（地址为 3eh）**

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
INTF3	CMP0IF	MULIF		COG0IF	I2CIF	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  
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 **MULIF**: 乘法器中断标志，软件清零，硬件置高  
1 = 发生乘法器中断，必须软件清 0  
0 = 没发乘法器生中断
- Bit 4 **COG0IF**: 互补输出发生器中断标志，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 3 **I2CIF**: I2C 中断标志，软件清零，硬件置高  
1 = 发生 I2C 中断，必须软件清 0  
0 = 没发生 I2C 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 = 不定位



### 2.5.3 外部中断 0

PT3.5 为外部中断 0 的输入端。触发方式由 INTCFG 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 PT3.5 被触发，中断标志位 E0IF 就会置 1。

#### INTCFG 寄存器（地址为 1fh）

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
INTCFG					E2M	E1M	E0M[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 3 E2M: 外部中断 2 触发模式  
1 = 外部中断 2 为下降沿触发  
0 = 外部中断 2 在状态改变时触发
- 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

### 2.5.4 外部中断 1

PT1.1、PT1.2、PT1.3、PT1.4、PT1.5、PT1.7、PT2.6、PT2.7、PT3.0、PT3.1、PT3.3、PT3.4、PT3.6、PT5.0、PT5.1、PT5.2、PT5.4、PT5.6 和 PT5.7 都可作为外部中断 1 的输入端。触发方式由 INTCFG 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。

寄存器详见 IO PORT 一节。

### 2.5.5 外部中断 2

PT1.0、PT1.6、PT3.2、PT5.3 和 PT5.5 都可作为外部中断 2 的输入端。触发方式由 INTCFG 寄存器中的 E2M 寄存器决定。INTE 寄存器中的 E2IE 为外部中断 2 的使能位，INTF 寄存器中的 E2IF 为中断标志位，硬件置 1，软件清 0。不管 E2IE 是否使能，只要外部中断 2 被触发，中断标志位 E2IF 就会置 1。

寄存器详见 IO PORT 一节。

#### 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



### 2.5.6 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。

### 2.5.7 定时器 0 溢出中断

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

### 2.5.8 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。

### 2.5.9 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。

### 2.5.10 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。

### 2.5.11 比较器 0 中断

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

### 2.5.12 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。

### 2.5.13 I2C 中断

INTE3 寄存器中的 I2CIE 为 I2C 中断的使能位，INTF3 寄存器中的 I2CIF 为中断标志位，软件清 0。当 I2C 完成一个字节数据发送或接收时，不管 I2CIE 是否使能，I2CIF 都会硬件置 1。当 I2C 中断被响应后，除了清除 I2CIF 标志位外，还需要对 TXRX\_STS 寄存器进行一次读操作和一次写操作来清除 I2C 状态标志位，如果不做该操作，系统会一直响应 I2C 中断。

### 2.5.14 MUL 中断

INTE3 寄存器中的 MULIE 为 MUL 中断的使能位，INTF3 寄存器中的 MULIF 为中断标志位，软件清 0。当 MUL 完成一次乘法运算时，不管 MULIE 是否使能，MULIF 都会硬件置 1。

### 2.5.15 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。



### 2.5.16 PWM 暂停中断

INTE2 寄存器中的 BKIE 为 PWM 暂停中断的使能位，INTF2 寄存器中的 BKIF 为中断标志位，软件清 0。当 PWM\_BKIN 从无效状态翻转到有效状态时，不管 BKIE 是否使能，BKIF 都会硬件置 1。

### 2.5.17 PWM 暂停失效中断

INTE2 寄存器中的 BKNAIE 为 PWM 暂停失效中断的使能位，INTF2 寄存器中的 BKNAIF 为中断标志位，软件清 0。当 PWM\_BKIN 从有效状态翻转到无效状态时，不管 BKNAIE 是否使能，BKNAIF 都会硬件置 1。

### 2.5.18 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。

### 2.5.19 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。

### 2.5.20 LVD 检测中断

INTE2 寄存器中的 LVDIE 为 LVD 检测的使能位，STATUS 寄存器中的 LVD 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。

### 2.5.21 PUSH 和 POP 处理

CSU38M20 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
...
```

2.6 定时器 0

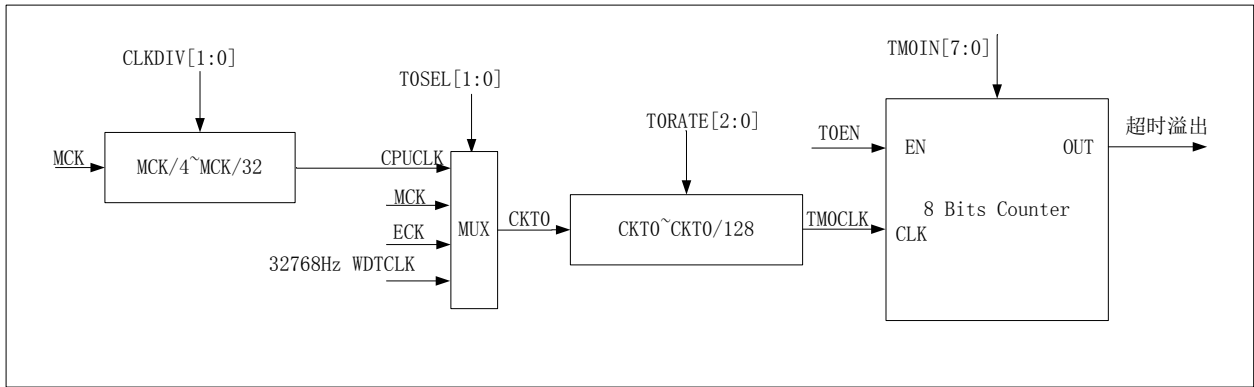


图22 定时器 0 功能框图

定时器 0 模块的输入为 MCK 或者 CPUCLK。在定时器 0 模块集成了一个分频器，分频的时钟 TMOCLK 作为 8 bits 计数器的输入时钟。当用户设置了定时器 0 模块的使能标志，8 bits 计数器将启动，将会从 000H 递增到 TMOIN。用户需要设置 TMOIN（定时器 0 模块中断信号选择器）以选择定时超时中断信号。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。

2.6.1 寄存器说明

表 15 定时器 0 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF				TM0IF					u0u00u00
07H	INTE	GIE			TM0IE					00u00u00
0FH	TM0CON	TOEN	TORATE[2:0]				TORSTB	TOSEL[1:0]		0000u100
10H	TM0IN	TM0IN[7:0]								11111111
11H	TM0CNT	TM0CNT[7:0]								00000000

表 16 TM0CON 寄存器（地址为 0FH）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
TM0CON	TOEN	TORATE[2:0]				TORSTB	TOSEL[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能														
7	TOEN	定时器 0 使能位 1: 使能定时器 0 0: 禁止定时器 0														
6:4	TORATE[2:0]	定时器 0 时钟选择 <table border="1" style="width: 100%;"> <thead> <tr> <th>TORATE [2:0]</th> <th>TMOCLK</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>CKT0</td> </tr> <tr> <td>001</td> <td>CKT0 /2</td> </tr> <tr> <td>010</td> <td>CKT0 /4</td> </tr> <tr> <td>011</td> <td>CKT0 /8</td> </tr> <tr> <td>100</td> <td>CKT0 /16</td> </tr> <tr> <td>101</td> <td>CKT0 /32</td> </tr> </tbody> </table>	TORATE [2:0]	TMOCLK	000	CKT0	001	CKT0 /2	010	CKT0 /4	011	CKT0 /8	100	CKT0 /16	101	CKT0 /32
TORATE [2:0]	TMOCLK															
000	CKT0															
001	CKT0 /2															
010	CKT0 /4															
011	CKT0 /8															
100	CKT0 /16															
101	CKT0 /32															

		110	CKT0 /64										
		111	CKT0 /128										
2	TORSTB	定时器 0 复位 1: 禁止定时器 0 复位 0: 使能定时器 0 复位 当将该位为 0 时, 定时器 0 复位后, TORSTB 会自动置 1											
1:0	TOSEL[1:0]	时钟源选择 <table border="1"> <tr> <td>TOSEL[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>		TOSEL[1:0]	定时器 0 时钟源	00	CPUCLK	01	MCK	10	外部 32768Hz 晶振时钟, 仅当外部接 32768Hz 晶振, 且晶振打开时有效	11	内部 32768 WDT 时钟, 仅当内部 WDT 晶振打开时有效
TOSEL[1:0]	定时器 0 时钟源												
00	CPUCLK												
01	MCK												
10	外部 32768Hz 晶振时钟, 仅当外部接 32768Hz 晶振, 且晶振打开时有效												
11	内部 32768 WDT 时钟, 仅当内部 WDT 晶振打开时有效												

表 17 TM0IN 寄存器 (地址为 10H) 各位功能表

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
TM0IN	TM0IN[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 18 TM0CNT 寄存器 (地址为 11H) 各位功能表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM0CNT	TM0CNT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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

操作:

- 1) 设置 TM0CLK, 为定时器 0 模块选择输入。
- 2) 设置 TM0IN, 选择定时器 0 溢出值。(溢出值: 1~255)
- 3) 设置寄存器标志位: TM0IE 与 GIE, 使能定时器 0 中断。
- 4) 清零寄存器标志位: TORSTB, 复位定时器 0 模块的计数器。
- 5) 设置寄存器标志位: 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:

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

2.7 I/O PORT

表 19 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	PT1PD	PT1PD[7:0]								00000000	
24h	PT1INT	PT1INT[7:0]								00000000	
25h	PT2INT	PT2INT[7:6]									00uuuuuu
26h	PT3INT	PT3INT[7:0]								00000000	
27h	PT5INT	PT5INT[7:0]								00000000	
28h	PT1CON	PT1CON[7:0]								00000000	
29h	PT1TYPE	PT1TYPE[7:0]								00000000	
2ah	PT2	PT2[7:0]								xxxxxxxx	
2bh	PT2EN	PT2EN[7:6]									00uuuuuu
2ch	PT2PU	PT2PU[7:6]									00uuuuuu
2dh	PT2PD	PT2PD[7:6]									00uuuuuu
2eh	PT2TYPE	PT2TYPE[7:6]									00uuuuuu
2fh	PT2CON	PT2CON[7:6]									00uuuuuu
30h	PT3	PT3[7:0]								xxxxxxxx	
31h	PT3EN	PT3EN[7:0]								00000000	
32h	PT3PU	PT3PU[7:0]								00000000	
33h	PT3PD	PT3PD[7:0]								00000000	
34h	PT3CON	PT3CON[7:0]								00000000	
35h	PT3TYPE	PT3TYPE[7:0]								00000000	
36h	PT5	PT5[7:0]								xxxxxxxx	
37h	PT5EN	PT5EN[7:0]								00000000	
38h	PT5PU	PT5PU[7:0]								00000000	
39h	PT5PD	PT5PD[7:0]								00000000	
3ah	PT5TYPE	PT5TYPE[7:0]								00000000	
3bh	PT5CON	PT5CON[7:0]								00000000	
70h	IODRV	BGDRV7	BGDRV6	BGDRV5	BGDRV4	BGDRV3	BGDRV2	BGDRV1	BGDRV0	00000000	

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。CSU38M20 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。

2.7.1 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 0 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

**PT1PU 寄存器 (地址为 22h)**

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1PU	PT1PU[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PU[7:0]: GPIO1 口上拉电阻使能标志

PT1PU[7] = GPIO1 bit 7 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT1PU[6] = GPIO1 bit 6 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT1PU[5] = GPIO1 bit 5 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT1PU[4] = GPIO1 bit 4 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT1PU[3] = GPIO1 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT1PU[2] = GPIO1 bit 2 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT1PU[1] = GPIO1 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT1PU[0] = GPIO1 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

**PT1PD 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
PT1PD	PT1PD[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PD[7:0]: 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  
 PT1PD[3] = GPIO1 bit 3 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT1PD[2] = GPIO1 bit 2 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT1PD[1] = GPIO1 bit 1 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT1PD[0] = GPIO1 bit 0 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT1CON 寄存器（地址为 28h）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1CON	PT1CON[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-0 PT1CON[7:0]: GPIO1 口模拟/数字端口使能标志

PT1CON[7] = GPIO1bit 7 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT1CON[6] = GPIO1bit 6 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT1CON[5] = GPIO1bit 5 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT1CON[4] = GPIO1bit 4 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT1CON[3] = GPIO1bit 3 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

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

PT1TYPE 寄存器（地址为 29h）

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
PT1TYPE	PT1TYPE[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1TYPE[7:0]: GPIO1 口施密特输入使能标志

PT1TYPE[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

PT1TYPE[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

PT1TYPE[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

PT1TYPE[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

PT1TYPE[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

PT1TYPE[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

PT1TYPE[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

PT1TYPE[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

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

## 2.7.2 PT2 口

PT2 寄存器（地址为 2Ah）

特性	R/W-X	R/W-X	U-0	U-0	U-0	U-0	U-0	U-0
PT2	PT2[7:6]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-6 PT2[7:6]: GPIO2 口数据标志位  
 PT2[7] = GPIO2 bit 7 的数据标志位  
 PT2[6] = GPIO2 bit 6 的数据标志位

PT2EN 寄存器（地址为 2Bh）

特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PT2EN	PT2EN[7]	PT2EN[6]						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 PT2EN[7:6]: GPIO 2 口输入/输出控制标志  
 PT2EN[7] = GPIO2 bit 7 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口  
 PT2EN[6] = GPIO2 bit 6 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT2PU 寄存器（地址为 2Ch）

特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PT2PU	PT2PU[7:6]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 PT2PU[7:6]: GPIO2 口上拉电阻使能标志  
 PT2PU[7] = GPIO2 bit 7 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT2PU[6] = GPIO2 bit 6 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

PT2PD 寄存器（地址为 2Dh）

特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PT2PD	PT2PD[7:6]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 PT2PD[7:6]: GPIO2 口下拉电阻使能标志  
 PT2PD[7] = GPIO2 bit 7 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT2PD[6] = GPIO2 bit 6 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

PT2TYPE 寄存器（地址为 2Eh）

特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PT2TYPE	PT2TYPE[7:6]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 PT2TYPE[7:6]: GPIO2 口施密特输入使能标志  
 PT2TYPE[7] = GPIO2 bit 7 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入  
 PT2TYPE[6] = GPIO2 bit 6 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入

PT2CON 寄存器（地址为 2Fh）

特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PT2CON	PT2CON[7:6]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0





Bit7-6 PT2CON [7:6]: GPIO2 口模拟/数字端口使能标志

PT1CON[7] = GPIO1bit 7 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

PT1CON[6] = GPIO1bit 6 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

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

### 2.7.3 PT3 口

#### PT3 寄存器 (地址为 30h)

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
PT3	PT3[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3[7:0]: GPIO3 口数据标志位

PT3[7] = GPIO3 bit 7 的数据标志位

PT3[6] = GPIO3 bit 6 的数据标志位

PT3[5] = GPIO3 bit 5 的数据标志位

PT3[4] = GPIO3 bit 4 的数据标志位

PT3[3] = GPIO3 bit 3 的数据标志位

PT3[2] = GPIO3 bit 2 的数据标志位

PT3[1] = GPIO3 bit 1 的数据标志位

PT3[0] = GPIO3 bit 0 的数据标志位

#### PT3EN 寄存器 (地址为 31h)

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
PT3EN	PT3EN[7:0]							
	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[7] = GPIO3 bit 7 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

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[5] = GPIO3 bit 5 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT3EN[4] = GPIO3 bit 4 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT3EN[3] = GPIO3 bit 3 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT3EN[2] = GPIO3 bit 2 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT3EN[1] = GPIO3 bit 1 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT3EN[0] = GPIO3 bit 0 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

#### PT3PU 寄存器 (地址为 32h)

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
PT3PU	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[7] = GPIO3 bit 7 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

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[5] = GPIO3 bit 5 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT3PU[4] = GPIO3 bit 4 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT3PU[3] = GPIO3 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT3PU[2] = GPIO3 bit 2 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT3PU[1] = GPIO3 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT3PU[0] = GPIO3 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

**PT3PD 寄存器 (地址为 33h)**

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
PT3PD	PT3PD[3:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-0 PT3PD[7:0]: GPIO3 口上拉电阻使能标志

PT3PD[7] = GPIO3 bit 7 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT3PD[6] = GPIO3 bit 6 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT3PD[5] = GPIO3 bit 5 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT3PD[4] = GPIO3 bit 4 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT3PD[3] = GPIO3 bit 3 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT3PD[2] = GPIO3 bit 2 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT3PD[1] = GPIO3 bit 1 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻  
 PT3PD[0] = GPIO3 bit 0 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻

**PT3CON 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
PT3CON	PT3CON[7:0]							
	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[7] = GPIO3bit 7 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口  
 PT3CON[6] = 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[5] = GPIO3bit 5 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口  
 PT3CON[4] = GPIO3bit 4 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口  
 PT3CON[3] = GPIO3bit 3 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口  
 PT3CON[2] = GPIO3bit 2 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口  
 PT3CON[1] = GPIO3bit 1 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口  
 PT3CON[0] = GPIO3bit 0 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口

**PT3TYPE 寄存器 (地址为 35h)**

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
PT3TYPE	PT3TYPE[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3TYPE[7:0]: GPIO3 口施密特输入使能标志

PT3TYPE[7] = GPIO3 bit 7 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入  
 PT3TYPE[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  
 PT3TYPE[5] = GPIO3 bit 5 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入  
 PT3TYPE[4] = GPIO3 bit 4 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入  
 PT3TYPE[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  
 PT3TYPE[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  
 PT3TYPE[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  
 PT3TYPE[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

**2.7.4 PT5 口**

**PT5 寄存器 (地址为 36h)**

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
PT5	PT5[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT5[7:0]: GPIO5 口数据标志位  
 PT5[7] = GPIO5 bit 7 的数据标志位  
 PT5[6] = GPIO5 bit 6 的数据标志位  
 PT5[5] = GPIO5 bit 5 的数据标志位  
 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 37h)**

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
PT5EN	PT5EN[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT5EN[7:0]: GPIO5 口输入/输出控制标志  
 PT5EN[7] = GPIO5 bit 7 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口  
 PT5EN[6] = GPIO5 bit 6 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口  
 PT5EN[5] = GPIO5 bit 5 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口  
 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 38h)**

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
PT5PU	PT5PU[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-0 PT5PU[7:0]: GPIO5 口上拉电阻使能标志  
 PT5PU[7] = GPIO5 bit 7 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT5PU[6] = GPIO5 bit 6 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 PT5PU[5] = GPIO5 bit 5 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻  
 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 39h)**

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
PT5PD	PT5PD[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT5PD[7:0]: GPIO5 口下拉电阻使能标志

- PT5PD[7] = GPIO5 bit 7 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻
- PT5PD[6] = GPIO5 bit 6 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻
- PT5PD[5] = GPIO5 bit 5 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻
- PT5PD[4] = GPIO5 bit 4 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻
- PT5PD[3] = GPIO5 bit 3 控制标志位; 0 = 断开下拉电阻, 1 = 使用下拉电阻
- 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

**PT5TYPE 寄存器 (地址为 3Ah)**

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
PT5TYPE	PT5TYPE[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT5TYPE[7:0]: GPIO5 口施密特输入使能标志

- PT5TYPE[7] = GPIO5 bit 7 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
- PT5TYPE[6] = GPIO5 bit 6 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
- PT5TYPE[5] = GPIO5 bit 5 控制标志位; 0 = 关闭施密特输入, 1 = 打开施密特输入
- PT5TYPE[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
- PT5TYPE[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
- PT5TYPE[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
- PT5TYPE[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
- PT5TYPE[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

**PT5CON 寄存器 (地址为 3Bh)**

特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5CON	PT5CON[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-0 PT5CON[7: 0]: GPIO5 口模拟/数字端口使能标志

- PT5CON[7] = GPIO5bit 7 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口
- PT5CON[6] = GPIO5bit 6 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口
- PT5CON[5] = GPIO5bit 5 的 I/O 控制标志位; 0 = 定义为数字口, 1 = 定义为模拟口
- PT5CON[4] = GPIO5bit 4 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 3 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 2 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 1 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

**2.7.5 IO 口外部中断使能**

**PT1INT 寄存器 (地址为 24h)**

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
PT1INT	PT1INT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7 PT1INT[7]: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 6 PT1INT[6]:PT1.6 外部中断 2 使能  
0 = 禁止 PT1.6 外部中断 2  
1 = 使能 PT1.6 外部中断 2
- Bit 5 PT1INT[5]: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 4 PT1INT[4]: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 3 PT1INT[3]: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 2 PT1INT[2]: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
- Bit 1 PT1INT[1]: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 0 PT1INT[0]:PT1.0 外部中断 2 使能  
0 = 禁止 PT1.0 外部中断 2  
1 = 使能 PT1.0 外部中断 2

**PT2INT 寄存器 (地址为 25h)**

特性	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PT2INT	PT2INT[7:6]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7 PT2INT[7]:PT2.7 外部中断 1 使能  
0 = 禁止 PT2.7 外部中断 1  
1 = 使能 PT2.7 外部中断 1
- Bit 3 PT2INT[6]:PT2.6 外部中断 1 使能  
0 = 禁止 PT2.6 外部中断 1  
1 = 使能 PT2.6 外部中断 1

**PT3INT 寄存器 (地址为 26h)**

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



PT3INT	PT3INT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Bit 7	PT3INT [7]:PT3.7 外部中断 1 使能 0 = 禁止 PT3.7 外部中断 1 1 = 使能 PT3.7 外部中断 1							
Bit 6	PT3INT [6]: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 5	PT3INT [5]:PT3.5 外部中断 0 使能 0 = 禁止 PT3.5 外部中断 0 1 = 使能 PT3.5 外部中断 0							
Bit 4	PT3INT [4]:PT3.4 外部中断 1 使能 0 = 禁止 PT3.4 外部中断 1 1 = 使能 PT3.4 外部中断 1							
Bit 3	PT3INT [3]:PT3.3 外部中断 1 使能 0 = 禁止 PT3.3 外部中断 1 1 = 使能 PT3.3 外部中断 1							
Bit 2	PT3INT [2]:PT3.2 外部中断 2 使能 0 = 禁止 PT3.2 外部中断 2 1 = 使能 PT3.2 外部中断 2							
Bit 1	PT3INT [1]:PT3.1 外部中断 1 使能 0 = 禁止 PT3.1 外部中断 1 1 = 使能 PT3.1 外部中断 1							
Bit 0	PT3INT [0]:PT3.0 外部中断 1 使能 0 = 禁止 PT3.0 外部中断 1 1 = 使能 PT3.0 外部中断 1							

**PT5INT 寄存器（地址为 27h）**

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
PT5INT	PT5INT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Bit 7	PT5INT [7]:PT5.7 外部中断 1 使能 0 = 禁止 PT5.7 外部中断 1 1 = 使能 PT5.7 外部中断 1							
Bit 6	PT5INT [6]:PT5.6 外部中断 1 使能 0 = 禁止 PT5.6 外部中断 1 1 = 使能 PT5.6 外部中断 1							
Bit 5	PT5INT [5]:PT5.5 外部中断 2 使能 0 = 禁止 PT5.5 外部中断 2 1 = 使能 PT5.5 外部中断 2							
Bit 4	PT5INT [4]:PT5.4 外部中断 1 使能 0 = 禁止 PT5.4 外部中断 1 1 = 使能 PT5.4 外部中断 1							
Bit 3	PT5INT [3]:PT5.3 外部中断 2 使能 0 = 禁止 PT5.3 外部中断 2 1 = 使能 PT5.3 外部中断 2							
Bit 2	PT5INT [2]:PT5.2 外部中断 1 使能 0 = 禁止 PT5.2 外部中断 1 1 = 使能 PT5.2 外部中断 1							
Bit 1	PT5INT [1]:PT5.1 外部中断 1 使能							





- 0 = 禁止 PT5.1 外部中断 1  
 1 = 使能 PT5.1 外部中断 1
- Bit 0 PT5INT [0]:PT5.0 外部中断 1 使能  
 0 = 禁止 PT5.0 外部中断 1  
 1 = 使能 PT5.0 外部中断 1

### 2.7.6 驱动能力配置

CSU38M20 有 8 个 IO 口驱动能力可配置，驱动能力有 2 档可选，分别是 56/20mA，可配置驱动能力的 IO 口有 PT3.3、PT3.2、PT5.7、PT5.6、PT5.5、PT1.7、PT1.6、PT1.5。其他 IO 口的驱动能力默认为 20mA。

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
70h	IODRV	BGDRV7	BGDRV6	BGDRV5	BGDRV4	BGDRV3	BGDRV2	BGDRV1	BGDRV0	00000000

#### IODRV 寄存器（地址为 70h）

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	R/W-0
IODRV	BGDRV7	BGDRV6	BGDRV5	BGDRV4	BGDRV3	BGDRV2	BGDRV1	BGDRV0	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

位地址	标识符	功能
7	BGDRV7	PT3.3 驱动能力配置
		BGDRV7 驱动能力配置
		1 56mA 0 20mA
6	BGDRV6	PT3.2 驱动能力配置
		BGDRV6 驱动能力配置
		1 56mA 0 20mA
5	BGDRV5	PT5.7 驱动能力配置
		BGDRV5 驱动能力配置
		1 56mA 0 20mA
4	BGDRV4	PT5.6 驱动能力配置
		BGDRV4 驱动能力配置
		1 56mA 0 20mA
3	BGDRV3	PT5.5 驱动能力配置
		BGDRV3 驱动能力配置
		1 56mA 0 20mA
2	BGDRV2	PT1.7 驱动能力配置
		BGDRV1 驱动能力配置
		1 56mA 0 20mA
1	BGDRV1	PT1.6 驱动能力配置
		BGDRV1 驱动能力配置
		1 56mA 0 20mA

0	BGDRV0	PT1.5 驱动能力配置	
		BGDRV0	驱动能力配置
		1	56mA
		0	20mA

### 2.7.7 输入逻辑电平电压配置

表 20 PT1VTH 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
6Ch	PTVTH					PT5VTH	PT3VTH	PT2VTH	PT1VTH	uuuu0000

#### PTVTH 寄存器（地址为 6Ch）

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PTVTH					PT5VTH	PT3VTH	PT2VTH	PT1VTH
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3-0 PTVTH[3:0]: 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

PT2VTH = GPIO2 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



### 3 增强功能

#### 3.1 防误操作

CSU38M20 为防误操作影响，在对 MCK、TM3CON3、ISPCRC、PWMBKR 进行写操作以及执行 TBLP 指令时，需先向 WRPRT 寄存器连续写入 C9h/3Ah/D3h 进行解锁，执行 TBLP 指令也需要先进行解锁。

表 21 ISP 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
6dh	WRPRT	REGWRPROT [7:0]								00000000

#### 3.2 Halt 和 Sleep 模式

为了使其处于待机状态，可以让 CPU 停止工作使 CSU38M20 进行停止或睡眠模式，减低功耗。具体描述如下：

##### 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，需要做以下操作：

- 1、 关闭 ADC 模块，比较器/运算放大器，LCD 模块
- 2、 对于输入电平不固定的输入口，打开上拉或下拉电阻
- 3、 固定输入低电平或者高电平的 IO 口，配置为输入口并关闭上下拉电阻

注意：当在 SLEEP 前配置为 PWM 输出或者蜂鸣器输出，进入 SLEEP 前关闭定时器使能，PWM 输出口仍然会输出固定电平，若 SLEEP 模式下 IO 电平要由 PTx(x=1, 2, 3, 5)寄存器决定，则必须将 TMnCON (n=2、3、4)寄存器的低 2 位清零。

##### 注：

芯片如果处于 sleep 状态，这时候降低电压，配置 2.4/3.0/3.6V 低电压复位不会起作用，低于 2.0V 掉电复位点才会复位。如果 sleep 唤醒后，此时还处于低电压复位点以下，则会立即复位。

## Halt 示范程序:

```

...
movlw 01h
movwf pt1up ;断开 pt1 除 bit0(pt1[0])外的其他接口的上拉电阻
movlw feh
movwf pt1en ;pt1 口除 bit0(pt1[0])做输入口外, 其他接口作为输出口(pt1.3 除外)
clrf pt1 ;将 pt1[4:1]输出为低
clrf pt3up ;断开 pt3 上拉电阻
clrf pt3en ;pt3 口用作输入口
clrf pt3con ;pt3 口用作数字口
clrf pt3 ;将 pt3 输出为低
clrf pt5up ;断开 pt5 上拉电阻
clrf pt5en ;pt5 口用作输入口
clrf pt5 ;将 pt5 输出为低
clrf intf ;清除中断标志位
movlw 81h
movwf inte ;使能外部中断 0
halt ;进入停止模式
nop ;保证 CPU 重启后程序能正常工作
...

```

## Sleep 示范程序:

```

...
movlw 01h
movwf pt1up ;断开 pt1 除 bit0(pt1[0])外的其他接口的上拉电阻
movlw feh
movwf pt1en ;pt1 口除 bit0(pt1[0])做输入口外, 其他接口作为输出口(pt1.3 除外)
clrf pt1 ;将 pt1[4:1]输出为低
clrf pt3up ;断开 pt3 上拉电阻
clrf pt3en ;pt3 口用作输入口
clrf pt3con ;pt3 口用作数字口
clrf pt3 ;将 pt3 输出为低
clrf pt5up ;断开 pt5 上拉电阻
clrf pt5en ;pt5 口用作输入口
clrf pt5 ;将 pt5 输出为低
clrf intf ;清除中断标志位
movlw 81h
movwf inte ;使能外部中断 0
sleep ;进入睡眠模式
nop ;保证 CPU 重启后程序能正常工作
...

```

3.3 看门狗(WDT)

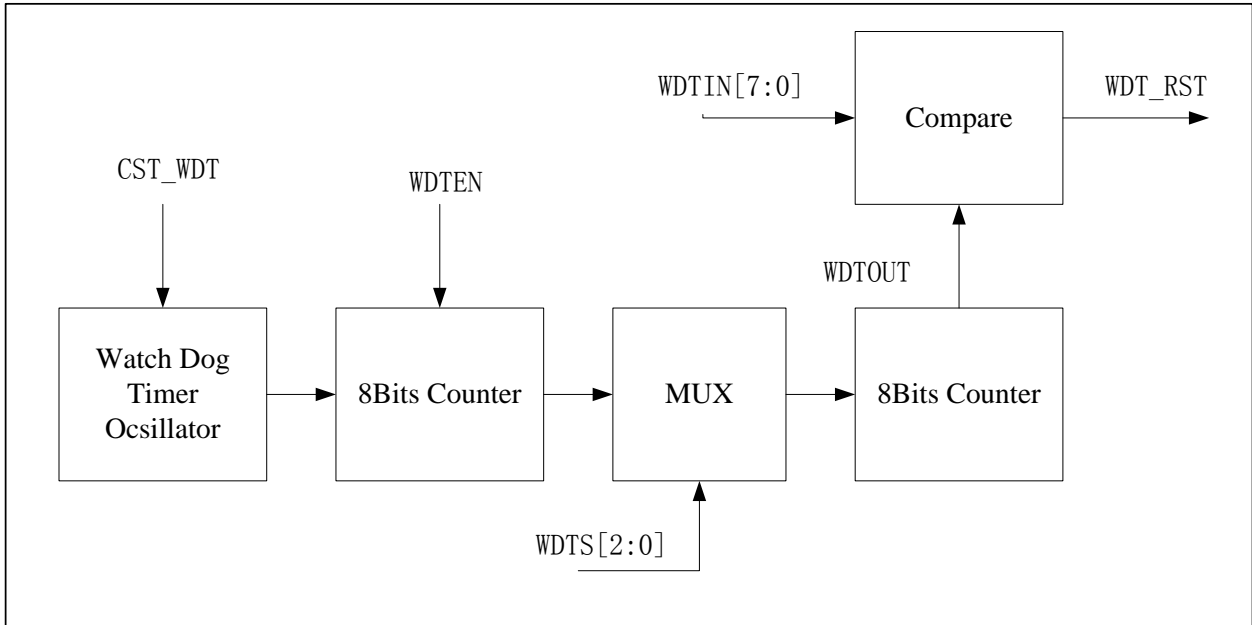


图23 看门狗定时器功能框图

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 WDT\_S[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 22 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		WDT_S[2:0]			0uu0u000
0Eh	WDTIN	WDT_IN[7:0]								11111111

表 23 WDTCON 寄存器 (地址为 0Dh) 各位功能表

特性	R/W-0	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
WDTCON	WDTEN			WDT_DLY		WDT_S[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 16ms

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		

表 24 WDTIN 寄存器（地址为 0EH）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WDTIN	WDT_IN[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7: 0	WDT_IN[7:0]	WDT 计数输入，默认值为 FFh

注：WDT\_DLY 延时 8ms 时系统一直处于看门狗复位状态，16ms 后解除；且 16ms 为使用内部 16M 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  $\frac{16M}{f} * 8 \text{ ms}$ ，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[2:0]，选择 WDT 时钟频率。
2. 设置 WDTIN，选择不同的溢出时间值
2. 置位寄存器标志位：WDTEN，使能 WDT。
3. 把 CST\_WDT 清 0，打开 WDT 的晶振。
4. 在程序中执行 CLRWDT 指令复位 WDT。

WDT 溢出时间计算公式：

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

WDTS[2:0]范围为 0~7，WDTIN[7:0]范围为 0~255。

WDTS[2:0]	计数器时钟	时间（当 WDTIN==FFH）
000	WDTCLK /256	2000ms
001	WDTCLK /128	1000ms
010	WDTCLK /64	500ms
011	WDTCLK /32	250ms
100	WDTCLK /16	125ms
101	WDTCLK /8	62.5ms
110	WDTCLK /4	31.25ms
111	WDTCLK /2	15.625ms



### 3.4 PWM 暂停控制器

PWM 暂停控制器主要用来暂停 PWM 输出。PWMBKR 为写保护寄存器，必须解锁写保护才能对该寄存器进行写操作。解锁写保护需对 WRPRT 寄存器连续写入 C9H, 3AH, D3H, 解锁后 16 个指令周期内都能对 PWMBKR 进行写操作，16 个指令周期后解锁自动失效，不能再进行写操作。

主要功能：

- 1) 同时暂停所有 PWM 输出。
- 2) 暂停部分 PWM 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
3ch	INTF2						BKIF	BKNAIF		u000000u
3dh	INTE2						BKIE	BKNAIE		u000000u
6eh	PWMBKR	PWMOEB	ATUOE	BKIPH	BKIEN		BKMOD[2:0]			0000u000

表 26 PWMBKR 寄存器（地址为 6EH）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PWMBKR	PWMOEB	ATUOE	BKIPH	BKIEN		BKMOD [2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7	PWMOEB	PWM 输出禁止位 1: 禁止 BKMOD [2:0]选择的 PWM 输出 0: 使能 PWM 输出 当 PWM_BKIN 输入有效时，PWMOEB 被硬件异步置 1，根据 ATUOE 的配置，该位可以被软件清零或硬件清零。 PWM 输出停止后，输出被强制拉到空闲状态，互补 PWM 输出也是如此。										
6	ATUOE	PWM 自动输出使能位 0: PWMOEB 只能软件清零 1: PWMOEB 可以软件清零或者被一个新的 PWM_BKIN 输入自动清零（如 PWM_BKIN 输入无效时）										
5	BKIPH	PWM_BKIN 输入有效极性控制位 0: PWM_BKIN 输入低电平有效 1: PWM_BKIN 输入高电平有效										
4	BKIEN	PWM_BKIN 输入使能位 0: 禁止 PWM_BKIN 1: 使能 PWM_BKIN										
2:0	BKMOD [2:0]	停止模式选择： <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>MOD[2:0]</th> <th>停止模式选择</th> </tr> </thead> <tbody> <tr> <td>3'b000</td> <td>停止所有 PWM 输出</td> </tr> <tr> <td>3'b001</td> <td>停止 PWM2 输出</td> </tr> <tr> <td>3'b010</td> <td>停止 PWM3 输出</td> </tr> <tr> <td>3'b100</td> <td>停止 PWM4 输出</td> </tr> </tbody> </table>	MOD[2:0]	停止模式选择	3'b000	停止所有 PWM 输出	3'b001	停止 PWM2 输出	3'b010	停止 PWM3 输出	3'b100	停止 PWM4 输出
MOD[2:0]	停止模式选择											
3'b000	停止所有 PWM 输出											
3'b001	停止 PWM2 输出											
3'b010	停止 PWM3 输出											
3'b100	停止 PWM4 输出											

		3'b011	停止 PWM2、PWM3 输出
		3'b101	停止 PWM2、PWM4 输出
		3'b110	停止 PWM3、PWM4 输出
		3'b111	停止所有 PWM 输出

注：停止 PWM2 则会停止 PWM2 及其对应的互补 PWM 输出，对 PWM3、PWM4 也是同样。PWM 输出停止后，输出被强制拉到空闲状态。

注：PWMBKR 为写保护寄存器，必须解锁写保护才能对该寄存器进行写操作。解锁写保护需对 WRPRT 寄存器连续写入 C9H, 3AH, D3H，解锁后 16 个指令周期内都能对 MCK 进行写操作，16 个指令周期后解锁自动失效，不能再进行写操作。

### 3.4.2 操作

操作：

- 1) 设置停止 PWM 触发源，
- 2) 设置停止模式 BKMOD
- 3) 设置寄存器标志位：PWMBKIE 与 GIE，使能定时器中断。
- 4) 使能 PWM\_BKIN 输入。

3.5 定时/计数器 2

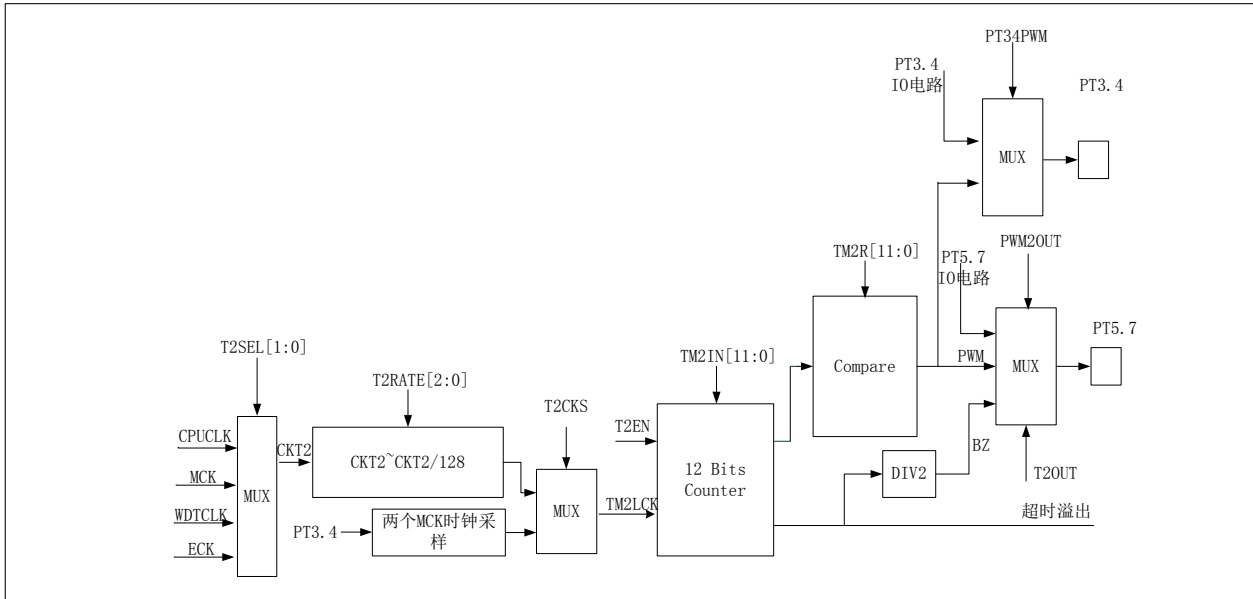


图24 定时/计数器 2 模块的功能框图

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。

当设置定时器溢出时间及占空比时，必须先配置高位，再配置低位。例如，当需要将定时器 2 溢出值配置为 12'h0137，那么需要先配置 TM2INH 为 01h，然后配置 TM2IN 为 37h。对 TM2R 和 TM2RH 也必须同样操作。同时，不能在小于 5 个定时器时钟周期内，连续 2 次对 TM2INH/TM2IN 或者 TM2RH/TM2R 进行赋值，否则会导致 2 次赋值都失败。

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；

3.5.1 寄存器说明

表 27 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	T2RST B	T2OUT	PWM2OUT		00000100
18h	TM2IN	TM2IN[7:0]								11111111
19h	TM2CNT	TM2CNT[7:0]								00000000
1ah	TM2R	TM2R[7:0]								00000000
1bh	TM2INH						TM2IN[11:8]			uuuu1111
1ch	TM2CNTH						TM2CNT[11:8]			uuuu0000
1dh	TM2RH						TM2R[11:8]			uuuu0000



1eh	TM2CON1	PWM2IS				T2SEL[1:0]	PT34PWM[1:0]	0uuu0000
4ch	DT2CON0	DT2_EN	DT2CK[1:0]		P2L_IS	P2H_IS	DT2CNT1[2:0]	00000000
4dh	DT2CON1		P2LOEN	P2LINV	P2HOEN	P2HINV	PWM2DL1[2:0]	u0000000
4eh	DT2CON2						DT2CNT2[2:0]	uuuuu000

表 28 TM2CON 寄存器（地址为 17H）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能																		
7	T2EN	定时/计数器 2 使能位 1: 使能定时器 2 0: 禁止定时器 2																		
6:4	T2RATE[2:0]	定时/计数器 2 时钟分频 <table border="1"> <tr> <th>T2RATE [2:0]</th> <th>TM2CLK</th> </tr> <tr> <td>000</td> <td>CKT2</td> </tr> <tr> <td>001</td> <td>CKT2 /2</td> </tr> <tr> <td>010</td> <td>CKT2 /4</td> </tr> <tr> <td>011</td> <td>CKT2 /8</td> </tr> <tr> <td>100</td> <td>CKT2 /16</td> </tr> <tr> <td>101</td> <td>CKT2 /32</td> </tr> <tr> <td>110</td> <td>CKT2 /64</td> </tr> <tr> <td>111</td> <td>CKT2 /128</td> </tr> </table> 可通过 T2SEL [1:0]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 1: 外部计数时钟输入作为时钟 (PT3.4) 0: CKT2 的分频时钟																		
2	T2RSTB	定时/计数器 2 复位 1: 禁止定时/计数器 2 复位 0: 使能定时/计数器 2 复位 当将该位为 0 时, 定时器 2 复位后, T2RSTB 会自动置 1																		
1	T2OUT	PT5.7 口输出控制 <table border="1"> <tr> <th>T2OUT</th> <th>PWM2OUT</th> <th>PT5.7 输出控制, 仅当 PT5.7 配置为输出有效</th> </tr> <tr> <td>0</td> <td>0</td> <td>普通 IO</td> </tr> <tr> <td>1</td> <td>0</td> <td>蜂鸣器输出</td> </tr> <tr> <td>0</td> <td>1</td> <td>PWM2 输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>PWM2 取反输出</td> </tr> </table>	T2OUT	PWM2OUT	PT5.7 输出控制, 仅当 PT5.7 配置为输出有效	0	0	普通 IO	1	0	蜂鸣器输出	0	1	PWM2 输出	1	1	PWM2 取反输出			
T2OUT	PWM2OUT	PT5.7 输出控制, 仅当 PT5.7 配置为输出有效																		
0	0	普通 IO																		
1	0	蜂鸣器输出																		
0	1	PWM2 输出																		
1	1	PWM2 取反输出																		
0	PWM2OUT	PT5.7 做 PWM 或者蜂鸣器输出时, PT5.7 必须配置为输出口																		

表 29 TM2IN 寄存器（地址为 18H）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM2IN	TM2IN[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0





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 30 TM2INH 寄存器（地址为 1BH）各位功能表

特性	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
TM2INH	TM2IN[11:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 31 TM2CNT 寄存器（地址为 19H）各位功能表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM2CNT	TM2CNT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 32 TM2CNTH 寄存器（地址为 1CH）各位功能表

特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
TM2CNTH	TM2CNT[11:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 33 TM2R 寄存器（地址为 1AH）各位功能表

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
TM2R	TM2R[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 34 TM2RH 寄存器（地址为 1DH）各位功能表

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM2RH	TM2R[11:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 35 TM2CON1 寄存器（地址为 1EH）各位功能表

特性	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM2CON1	PWM2IS				T2SEL[1:0]		PT34PWM[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7	PWM2IS	PWM2 空闲状态控制位（空闲状态是指定时器模块未使能时输出状态） 0: 空闲时输出低电平 1: 空闲时输出高电平										
3:2	T2SEL[1:0]	定时器 2 时钟选择 <table border="1"> <tr> <td>T2SEL[1:0]</td> <td>CKT2 时钟选择</td> </tr> <tr> <td>00</td> <td>CPUCLK</td> </tr> <tr> <td>01</td> <td>MCK</td> </tr> <tr> <td>10</td> <td>WDT CLK(32K)</td> </tr> <tr> <td>11</td> <td>ECK</td> </tr> </table>	T2SEL[1:0]	CKT2 时钟选择	00	CPUCLK	01	MCK	10	WDT CLK(32K)	11	ECK
T2SEL[1:0]	CKT2 时钟选择											
00	CPUCLK											
01	MCK											
10	WDT CLK(32K)											
11	ECK											
1:0	PT34PWM[1:0]	PT3.4 口输出控制 <table border="1"> <tr> <td>PT34PWM[1:0]</td> <td>PT3.4 输出控制，仅当 PT3.4 配置为输出有效</td> </tr> <tr> <td>2'b00</td> <td>普通 IO</td> </tr> <tr> <td>2'b01</td> <td>蜂鸣器输出</td> </tr> <tr> <td>2'b10</td> <td>PWM2 输出</td> </tr> <tr> <td>2'b11</td> <td>PWM2 取反输出</td> </tr> </table> PT3.4 做 PWM 或者蜂鸣器输出时，PT3.4 必须配置为输出口	PT34PWM[1:0]	PT3.4 输出控制，仅当 PT3.4 配置为输出有效	2'b00	普通 IO	2'b01	蜂鸣器输出	2'b10	PWM2 输出	2'b11	PWM2 取反输出
PT34PWM[1:0]	PT3.4 输出控制，仅当 PT3.4 配置为输出有效											
2'b00	普通 IO											
2'b01	蜂鸣器输出											
2'b10	PWM2 输出											
2'b11	PWM2 取反输出											

表 36 DT2CON0 寄存器（地址为 4CH）各位功能表

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
DT2CON0	DT2_EN	DT2CK[1:0]		P2L_IS	P2H_IS	DT2CNT1[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7	DT2_EN	定时器 2 互补式 PWM 输出使能位 0: 定时器 2 互补式输出禁止 1: 定时器 2 互补式 PWM 输出使能										
6:5	DT2CK[1:0]	定时器 2 死区时间时钟选择 <table border="1"> <tr> <td>DT2CK[1:0]</td> <td>DT2_CLK</td> </tr> <tr> <td>00</td> <td>MCK</td> </tr> <tr> <td>01</td> <td>MCK/2</td> </tr> <tr> <td>10</td> <td>MCK/4</td> </tr> <tr> <td>11</td> <td>MCK/8</td> </tr> </table>	DT2CK[1:0]	DT2_CLK	00	MCK	01	MCK/2	10	MCK/4	11	MCK/8
DT2CK[1:0]	DT2_CLK											
00	MCK											
01	MCK/2											
10	MCK/4											
11	MCK/8											
4	P2L_IS	互补输出 PWM2L 空闲状态控制位（空闲状态是指定时器模块未使能或互补 PWM 输出未使能时输出状态） 0: 空闲时输出低电平 1: 空闲时输出高电平										
3	P2H_IS	互补输出 PWM2H 空闲状态控制位（空闲状态是指定时器模块未使能或互补 PWM 输出未使能时输出状态） 0: 空闲时输出低电平 1: 空闲时输出高电平										

2:0	DT2CNT1[2:0]	PWM2H 死区时间选择 死区时间=DT2CNT1[2:0]*DT2_CLK
-----	--------------	---

表 37 DT2CON1 寄存器（地址为 4DH）各位功能表

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
DT2CON1		P2LOEN	P2LINV	P2HOEN	P2HINV	PWM2DL1[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
6	P2LOEN	PWM2L 输出控制 0: 普通 IO 1: PWM2_L 输出
5	P2LINV	PWM2L 相位控制 0: PWM2_L 正常相位输出 1: PWM2_L 取反输出
4	P2HOEN	PWM2H 输出控制 0: 普通 IO 1: PWM2_H 输出
3	P2HINV	PWM2_H 相位控制 0: PWM2_H 正常相位输出 1: PWM2_H 取反输出
2:0	PWM2DL1[2:0]	互补 PWM2 延时控制 延时时间为 TM2CLK*PWM2DL1[2:0]

表 38 DT2CON2 寄存器（地址为 4EH）各位功能表

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DT2CON1						DT2CNT2[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
2:0	DT2CNT2[2:0]	PWM2L 死区时间选择 死区时间=DT2CNT2[2:0]*DT2_CLK

### 3.5.2 蜂鸣器

操作：

- 5) 设置 TM2CLK，为定时器模块选择输入。
- 6) 设置 TM2IN，选择定时器溢出值。
- 7) 设置寄存器标志位：TM2IE 与 GIE，使能定时器中断。
- 8) 清零寄存器标志位：T2RSTB，复位定时器模块的计数器。
- 9) 设置寄存器标志位：T2EN，使能定时器模块的 12 bits 计数器。
- 10) 当定时超时发生时，BZ 输出信号发生跳变，可作为蜂鸣器输出；寄存器标志位 TM2IF 会自复位，程序计数器会复位为 04h。

定时器 2 溢出时间计算方法：

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

蜂鸣器周期计算方法：

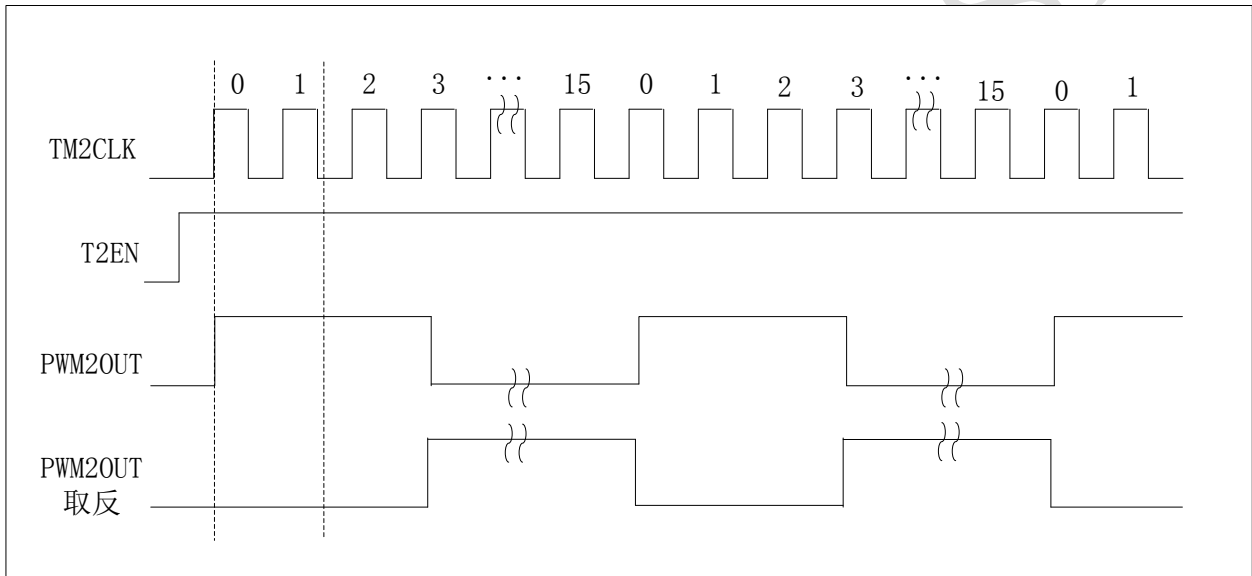
$$\text{蜂鸣器周期} = (\text{TM2IN} + 1) * 2 / \text{TM2CLK.} \quad (\text{TM2IN 不为 } 0)$$

### 3.5.3 PWM

操作：

- 1) 设置 TM2CLK，为定时/计数器 2 模块选择输入。
- 2) 设置 TM2IN 来配置 PWM2 的周期。
- 3) 设置 TM2R 来配置 PWM2 的高电平的脉宽。
- 4) 使能 PWM2OUT 输出，配置 PT5.7 为输出端口，之后把 T2EN 置 1 启动定时器。
- 5) PWM 从 PT5.7 输出。

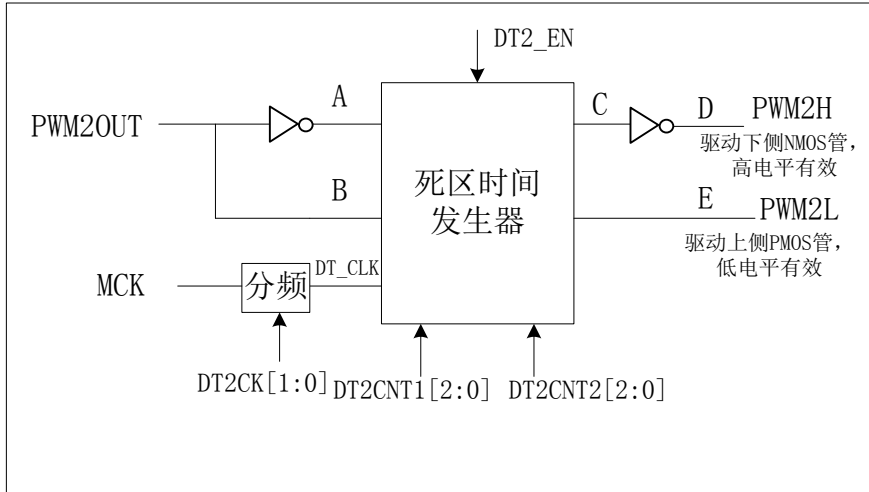
周期为 TM2IN+1，高电平脉宽为 TM2R。如 TM2IN=0x0F，TM2R=0x03 的 PWM2 波形输出如下：



### 3.5.4 互补式 PWM 输出

CSU38M20 提供源于定时器 2 的 PWM 输出的一对互补式输出，可用作 PWM 驱动信号。对于 PMOS 管上侧驱动，PWM 输出为低电平有效，而对于 NMOS 管下侧驱动，PWM 输出为高电平有效。当这对互补式输出同时用于驱动 PMOS 和 NMOS 时，死区时间发生器插入一死区时间以防止直流电流过大，该死区时间可通过 DT2CON0 寄存器的 DT2CK[1:0]和 DT2CNT[2:0]位来定义。在每个死区时间发生器输入信号的上升沿时插入一个死区时间。通过死区插入电路，输出信号最终发送至外部功率晶体管。

图25 互补式 PWM 输出方框图

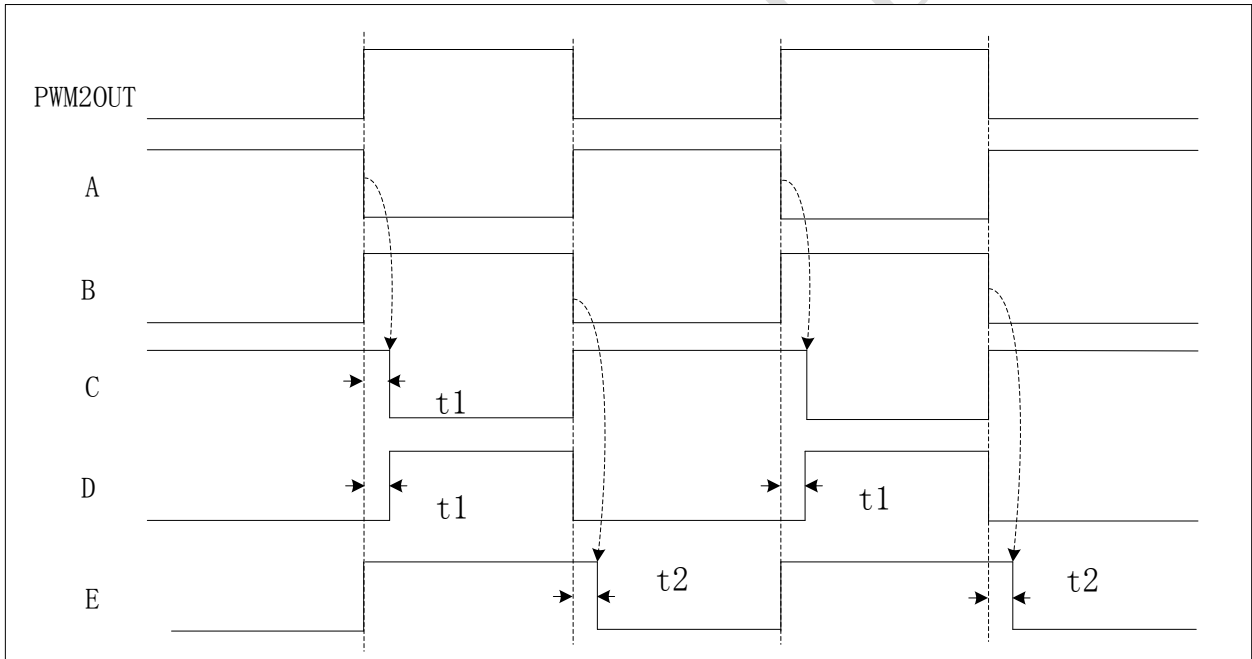


互补式 PWM 输出波形

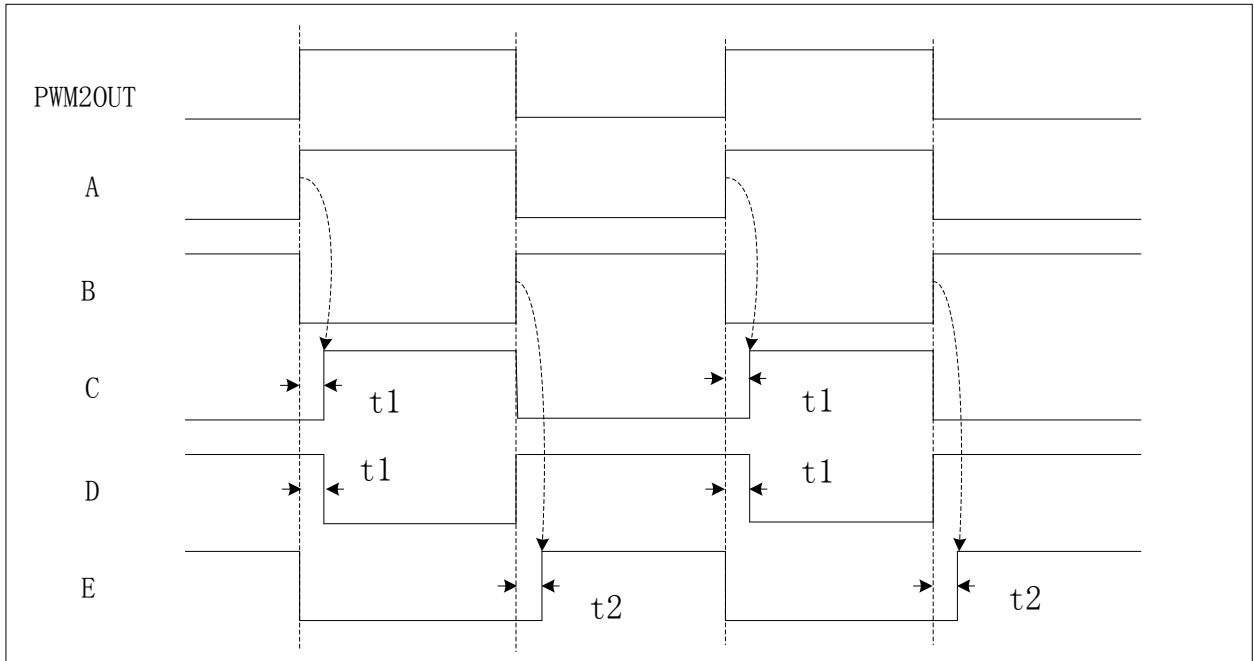
PWM2OUT 上升沿增加时间为  $t_1$  的死区时间，下降沿增加时间为  $t_2$  的死区时间。  
其中

$$t_1 = DT2CNT1[2:0] * DT2\_CLK$$

$$t_2 = DT2CNT2[2:0] * DT2\_CLK$$



PWM 输出取反后的互补 PWM 输出



3.6 定时/计数器 3

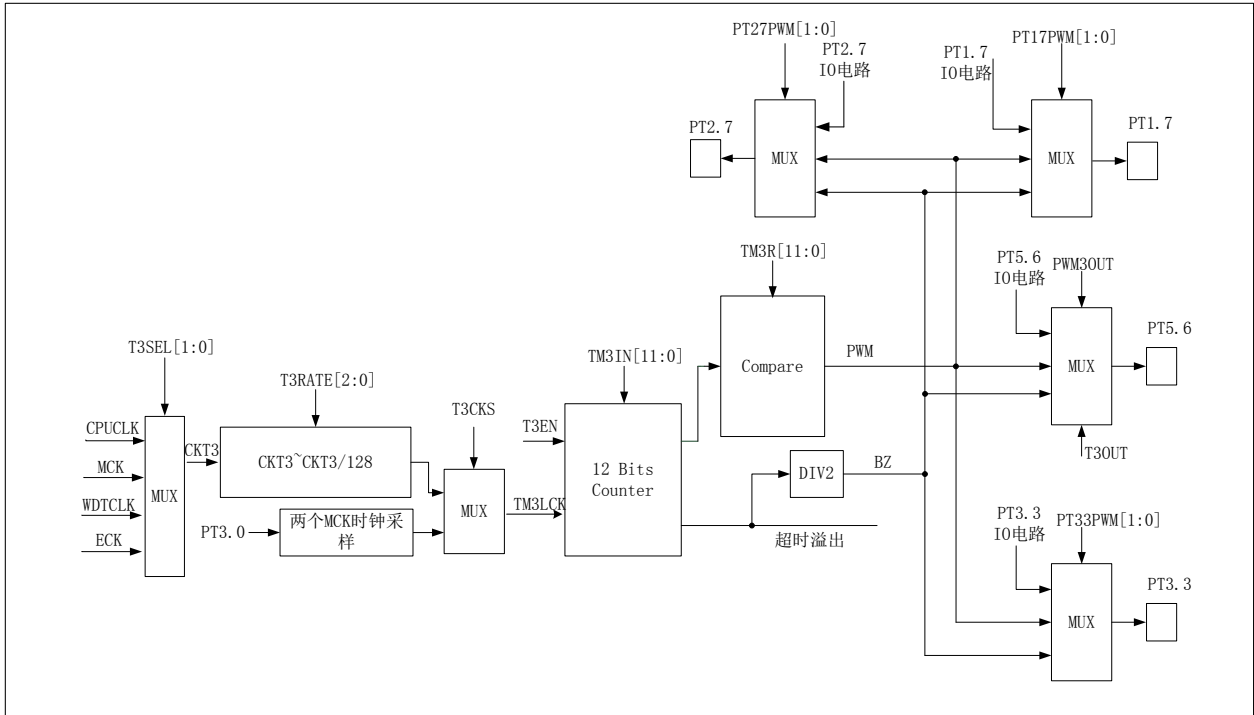


图26 定时/计数器 3 模块的功能框图

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。

当设置定时器溢出时间及占空比时，必须先配置高位，再配置低位。例如，当需要将定时器 3 溢出值配置为 12'h0137，那么需要先配置 TM3INH 为 01h，然后配置 TM3IN 为 37h。对 TM3R 和 TM3RH 也必须同样操作。同时，不能在小于 5 个定时器时钟周期内，连续 2 次对 TM3INH/TM3IN 或者 TM3RH/TM3R 进行赋值，否则会导致 2 次赋值都失败。

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;

3.6.1 寄存器说明

表 39 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					u0u00u00
3dh	INTE2				TM3IE					00u00u00
40h	TM3CON	T3EN	T3RATE[2:0]		T3CKS	T3RSTB	T3OUT	PWM3OUT		00000100
41h	TM3IN	TM3IN[7:0]								11111111
42h	TM3CNT	TM3CNT[7:0]								00000000
43h	TM3R	TM3R[7:0]								00000000



44h	TM3INH							TM3IN[11:8]	uuuu1111	
45h	TM3CNTH							TM3CNT[11:8]	uuuu0000	
46h	TM3RH							TM3R[11:8]	uuuu0000	
47h	TM3CON1	PT17PWM[1:0]	PT27PWM[1:0]		T3SEL[1:0]			PT33PWM[1:0]	00000000	
48h	TM3CON2	SIGN	TM3PNT_DUTY[6:0]							01000000
49h	TM3CON3	PNT_EN				PWM3IS		DT3CNT2[2:0]	0uuu0000	
4ah	DT3CON0	DT3_EN	DT3CK[1:0]	P3L_IS	P3H_IS			DT3CNT1[2:0]	00000000	
4bh	DT3CON1		P3LOEN	P3LINV	P3HOEN	P3HINV		PWM3DL1[2:0]	u0000000	

表 40 TM3CON 寄存器（地址为 40H）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
TM3CON	T3EN	T3RATE[2:0]			T3CKS	T3RSTB	T3OUT	PWM3OUT
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能																		
7	T3EN	定时/计数器 3 使能位 1: 使能定时器 3 0: 禁止定时器 3																		
6:4	T3RATE[2:0]	定时/计数器 3 时钟分频 <table border="1"> <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>CKT /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	CKT /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	CKT /128																			
3	T3CKS	定时/计数器 3 时钟源选择位 1: 外部计数时钟输入作为时钟 (PT3.0) 0: CKT3 的分频时钟																		
2	T3RSTB	定时/计数器 3 复位 1: 禁止定时/计数器 3 复位 0: 使能定时/计数器 3 复位 当将该位为 0 时, 定时器 3 复位后, T3RSTB 会自动置 1																		
1	T3OUT	PT5.6 口输出控制 <table border="1"> <tr> <th>T3OUT</th> <th>PWM3OUT</th> <th>PT5.6 输出控制, 仅当 PT5.6 配置为输出有效</th> </tr> <tr> <td>0</td> <td>0</td> <td>普通 IO</td> </tr> <tr> <td>1</td> <td>0</td> <td>蜂鸣器输出</td> </tr> </table>	T3OUT	PWM3OUT	PT5.6 输出控制, 仅当 PT5.6 配置为输出有效	0	0	普通 IO	1	0	蜂鸣器输出									
T3OUT	PWM3OUT	PT5.6 输出控制, 仅当 PT5.6 配置为输出有效																		
0	0	普通 IO																		
1	0	蜂鸣器输出																		
0	PWM3OUT	<table border="1"> <tr> <th>T3OUT</th> <th>PWM3OUT</th> <th>PT5.6 输出控制, 仅当 PT5.6 配置为输出有效</th> </tr> <tr> <td>0</td> <td>1</td> <td>PWM3 输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>PWM3 取反输出</td> </tr> </table> PT5.6 做 PWM 或者蜂鸣器输出时, PT5.6 必须配置为输出口	T3OUT	PWM3OUT	PT5.6 输出控制, 仅当 PT5.6 配置为输出有效	0	1	PWM3 输出	1	1	PWM3 取反输出									
T3OUT	PWM3OUT	PT5.6 输出控制, 仅当 PT5.6 配置为输出有效																		
0	1	PWM3 输出																		
1	1	PWM3 取反输出																		

表 41 TM3IN 寄存器（地址为 41H）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
----	-------	-------	-------	-------	-------	-------	-------	-------



TM3IN	TM3IN[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
7 : 0	TM3IN[7:0]	定时/计数器溢出值低 8 位						

表 42 TM3INH 寄存器（地址为 44H）各位功能表

特性	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
TM3INH					TM3IN[11:8]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
3 : 0	TM3IN[11:8]	定时/计数器溢出值高 4 位						

表 43 TM3CNT 寄存器（地址为 42H）各位功能表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM3CNT	TM3CNT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
7 : 0	TM3CNT[7:0]	定时/计数器 3 计数寄存器低 8 位，只读						

表 44 TM3CNTH 寄存器（地址为 45H）各位功能表

特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
TM3CNTH					TM3CNT[11:8]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
3 : 0	TM3CNT[11:8]	定时/计数器 3 计数寄存器高 4 位，只读						

表 45 TM3R 寄存器（地址为 43H）各位功能表

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
TM3R	TM3R[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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低 8 位						

表 46 TM3RH 寄存器（地址为 46H）各位功能表

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3RH					TM3R[11:8]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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高 4 位
-------	------------	---------------------------------

表 47 TM3CON1 寄存器（地址为 47H）各位功能表

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
TM3CON1	PT17PWM[1:0]		PT27PWM[1:0]		T3SEL[1:0]		PT33PWM[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7 : 6	PT17PWM[1:0]	PT1.7 口输出控制 <table border="1"> <tr> <td>PT17PWM[1:0]</td> <td>PT1.7 输出控制，仅当 PT1.7 配置为输出有效</td> </tr> <tr> <td>2'b00</td> <td>普通 IO</td> </tr> <tr> <td>2'b01</td> <td>蜂鸣器输出</td> </tr> <tr> <td>2'b10</td> <td>PWM3 输出</td> </tr> <tr> <td>2'b11</td> <td>PWM3 取反输出</td> </tr> </table> PT1.7 做 PWM 或者蜂鸣器输出时，PT1.7 必须配置为输出口	PT17PWM[1:0]	PT1.7 输出控制，仅当 PT1.7 配置为输出有效	2'b00	普通 IO	2'b01	蜂鸣器输出	2'b10	PWM3 输出	2'b11	PWM3 取反输出
PT17PWM[1:0]	PT1.7 输出控制，仅当 PT1.7 配置为输出有效											
2'b00	普通 IO											
2'b01	蜂鸣器输出											
2'b10	PWM3 输出											
2'b11	PWM3 取反输出											
5 : 4	PT27PWM[1:0]	PT2.7 口输出控制 <table border="1"> <tr> <td>PT27PWM[1:0]</td> <td>PT2.7 输出控制，仅当 PT2.7 配置为输出有效</td> </tr> <tr> <td>2'b00</td> <td>普通 IO</td> </tr> <tr> <td>2'b01</td> <td>蜂鸣器输出</td> </tr> <tr> <td>2'b10</td> <td>PWM3 输出</td> </tr> <tr> <td>2'b11</td> <td>PWM3 取反输出</td> </tr> </table> PT2.7 做 PWM 或者蜂鸣器输出时，PT2.7 必须配置为输出口	PT27PWM[1:0]	PT2.7 输出控制，仅当 PT2.7 配置为输出有效	2'b00	普通 IO	2'b01	蜂鸣器输出	2'b10	PWM3 输出	2'b11	PWM3 取反输出
PT27PWM[1:0]	PT2.7 输出控制，仅当 PT2.7 配置为输出有效											
2'b00	普通 IO											
2'b01	蜂鸣器输出											
2'b10	PWM3 输出											
2'b11	PWM3 取反输出											
3:2	T3SEL[1:0]	定时器 3 时钟选择 <table border="1"> <tr> <td>T3SEL[1:0]</td> <td>CKT3 时钟选择</td> </tr> <tr> <td>00</td> <td>CPUCLK</td> </tr> <tr> <td>01</td> <td>MCK</td> </tr> <tr> <td>10</td> <td>WDT CLK(32K)</td> </tr> <tr> <td>11</td> <td>ECK</td> </tr> </table>	T3SEL[1:0]	CKT3 时钟选择	00	CPUCLK	01	MCK	10	WDT CLK(32K)	11	ECK
T3SEL[1:0]	CKT3 时钟选择											
00	CPUCLK											
01	MCK											
10	WDT CLK(32K)											
11	ECK											
1 : 0	PT33PWM[1:0]	PT3.3 口输出控制 <table border="1"> <tr> <td>PT33PWM[1:0]</td> <td>PT3.3 输出控制，仅当 PT3.3 配置为输出有效</td> </tr> <tr> <td>2'b00</td> <td>普通 IO</td> </tr> <tr> <td>2'b01</td> <td>蜂鸣器输出</td> </tr> <tr> <td>2'b10</td> <td>PWM3 输出</td> </tr> <tr> <td>2'b11</td> <td>PWM3 取反输出</td> </tr> </table> PT3.3 做 PWM 或者蜂鸣器输出时，PT3.3 必须配置为输出口	PT33PWM[1:0]	PT3.3 输出控制，仅当 PT3.3 配置为输出有效	2'b00	普通 IO	2'b01	蜂鸣器输出	2'b10	PWM3 输出	2'b11	PWM3 取反输出
PT33PWM[1:0]	PT3.3 输出控制，仅当 PT3.3 配置为输出有效											
2'b00	普通 IO											
2'b01	蜂鸣器输出											
2'b10	PWM3 输出											
2'b11	PWM3 取反输出											

表 48 TM3CON2 寄存器（地址为 48H）各位功能表

特性	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3CON2	SIGN		TM3PNT_DUTY[6:0]					
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	SIGN	小数分频符号位，仅在使能小数分频后有效

		0: 表示正, 即小数分频时占空比 TM3R[11:0]到 TM3R[11:0]+1 之间 1: 表示负, 即小数分频时占空比 TM3R[11:0]-1 到 TM3R[11:0]之间
6:0	TM3PNT_DUTY[6:0]	定时器 3 的 PWM 小数分频配置, 上电复位值为 7'b100_0000 当 SIGN=0 时, 小数分频值为 TM3R[11:0]+ TM3PNT_DUTY[6:0]/128 当 SIGN=1 时, 小数分频值为 TM3R[11:0]- TM3PNT_DUTY[6:0]/128 以上计算公式仅在使能小数分频后有效。

表 49 TM3CON3 寄存器 (地址为 49H) 各位功能表

特性	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3CON3	PNT_EN				PWM3IS	DT3CNT2[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	PNT_EN	定时器 3 的 PWM 小数分频功能使能位 (写保护位) 0: 禁止定时器 3 的 PWM 小数分频 1: 使能定时器 3 的 PWM 小数分频 对寄存器 WRPRT 连续写入 C9H, 3AH, D3H, 即可解锁写保护。
3	PWM3IS	PWM3 空闲状态控制位 (空闲状态是指定时器模块未使能时输出状态) 0: 空闲时输出低电平 1: 空闲时输出高电平
2:0	DT3CNT2[2:0]	PWM3L 死区时间选择 死区时间=DT3CNT2[2:0]*DT3_CLK

表 50 DT3CON0 寄存器 (地址为 4AH) 各位功能表

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
DT3CON0	DT3_EN	DT3CK[1:0]		P3L_IS	P3H_IS	DT3CNT1[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7	DT3_EN	定时器 3 互补式 PWM 输出使能位 0: 定时器 3 互补式输出禁止 1: 定时器 3 互补式 PWM 输出使能										
6:5	DT3CK[1:0]	定时器 3 死区时间时钟选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>DT3CK[1:0]</td> <td>DT3_CLK</td> </tr> <tr> <td>00</td> <td>MCK</td> </tr> <tr> <td>01</td> <td>MCK/2</td> </tr> <tr> <td>10</td> <td>MCK/4</td> </tr> <tr> <td>11</td> <td>MCK/8</td> </tr> </table>	DT3CK[1:0]	DT3_CLK	00	MCK	01	MCK/2	10	MCK/4	11	MCK/8
DT3CK[1:0]	DT3_CLK											
00	MCK											
01	MCK/2											
10	MCK/4											
11	MCK/8											
4	P3L_IS	互补输出 PWM3L 空闲状态控制位 (空闲状态是指定时器模块未使能或互补 PWM 输出未使能时输出状态) 0: 空闲时输出低电平 1: 空闲时输出高电平										
3	P3H_IS	互补输出 PWM3H 空闲状态控制位 (空闲状态是指定时器模块未使能或互补 PWM 输出未使能时输出状态) 0: 空闲时输出低电平										

		1: 空闲时输出高电平
2:0	DT3CNT1[2:0]	PWM3H 死区时间选择 死区时间=DT3CNT1[2:0]*DT3_CLK

表 51 DT3CON1 寄存器（地址为 4BH）各位功能表

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
DT3CON1		P3LOEN	P3LINV	P3HOEN	P3HINV	PWM3DL1[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
6	P3LOEN	PWM3L 输出控制 0: 普通 IO 1: PWM3_L 输出
5	P3LINV	PWM3L 相位控制 0: PWM3_L 正常相位输出 1: PWM3_L 取反输出
4	P3HOEN	PWM3H 输出控制 0: 普通 IO 1: PWM3_H 输出
3	P3HINV	PWM3_H 相位控制 0: PWM3_H 正常相位输出 1: PWM3_H 取反输出
2:0	PWM3DL1[2:0]	互补 PWM3 延时控制 延时时间为 TM3CLK*PWM3DL1[2:0]

### 3.6.2 蜂鸣器

操作:

- 1) 设置 TM3CLK, 为定时器模块选择输入。
- 2) 设置 TM3IN, 选择定时器溢出值。
- 3) 设置寄存器标志位: TM3IE 与 GIE, 使能定时器中断。
- 4) 清零寄存器标志位: T3RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T3EN, 使能定时器模块的 12 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出; 寄存器标志位 TM3IF 会自复位, 程序计数器会复位为 04h。

定时器 3 溢出时间计算方法:

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

蜂鸣器周期计算方法:

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

### 3.6.3 PWM

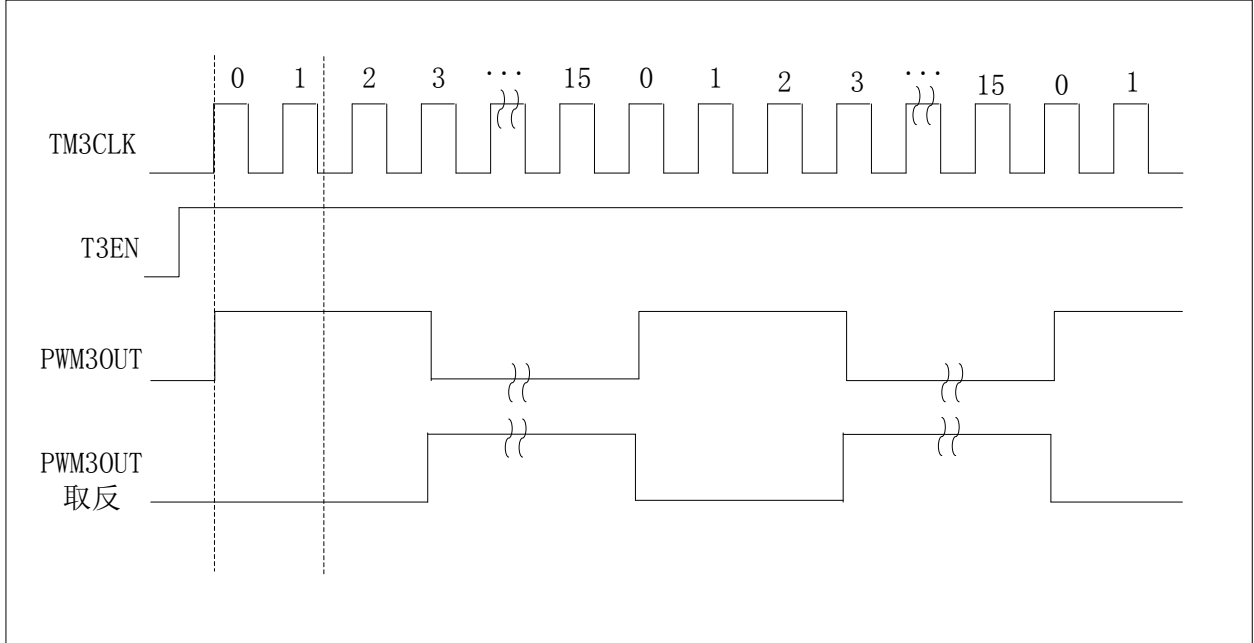
操作:

- 1) 设置 TM3CLK, 为定时/计数器 3 模块选择输入。



- 2) 设置 TM3IN 来配置 PWM3 的周期。
- 3) 设置 TM3R 来配置 PWM3 的高电平的脉宽。
- 4) 使能 PWM3OUT 输出，配置 PT5.6 为输出端口，之后把 T3EN 置 1 启动定时器。
- 5) PWM3 从 PT5.6 输出。

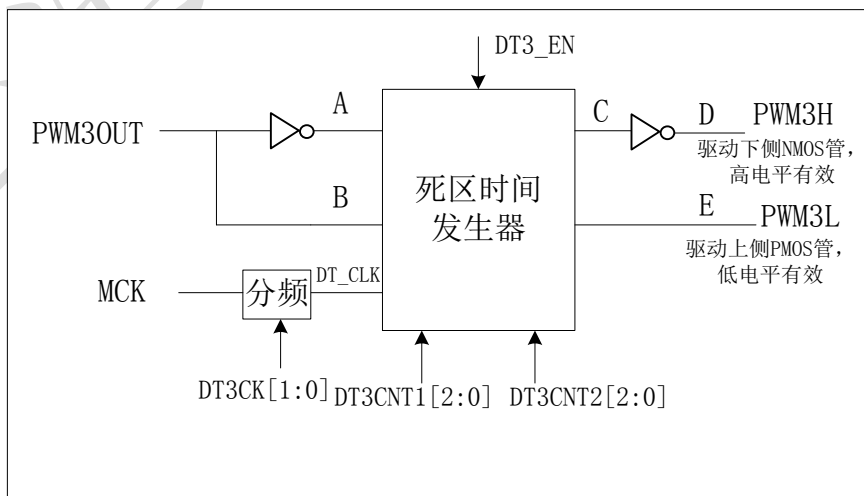
周期为  $TM3IN+1$ ，高电平脉宽为  $TM3R$ 。如  $TM3IN=0x0F$ ， $TM3R=0x03$  的 PWM3 波形输出如下：



### 3.6.4 互补式 PWM 输出

CSU38M20 提供源于定时器 3 的 PWM 输出的一对互补式输出，可用作 PWM 驱动信号。对于 PMOS 管上侧驱动，PWM 输出为低电平有效，而对于 NMOS 管下侧驱动，PWM 输出为高电平有效。当这对互补式输出同时用于驱动 PMOS 和 NMOS 时，死区时间发生器插入一死区时间以防止直流电流过大，该死区时间可通过 DT3CON0 寄存器的 DT3CK[1:0]和 DT3CNT[2:0]位来定义。在每个死区时间发生器输入信号的上升沿时插入一个死区时间。通过死区插入电路，输出信号最终发送至外部功率晶体管。

图27 互补式 PWM 输出方框图



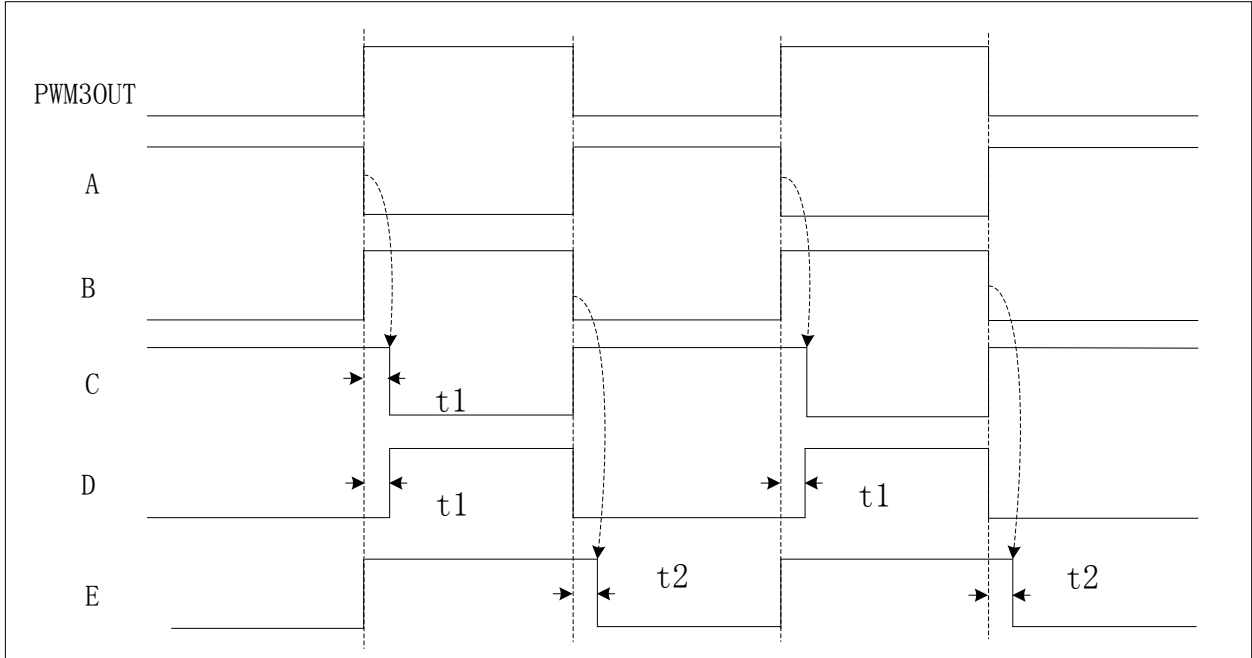
互补式 PWM 输出波形

PWM3OUT 上升沿增加时间为  $t_1$  的死区时间，下降沿增加时间为  $t_2$  的死区时间。

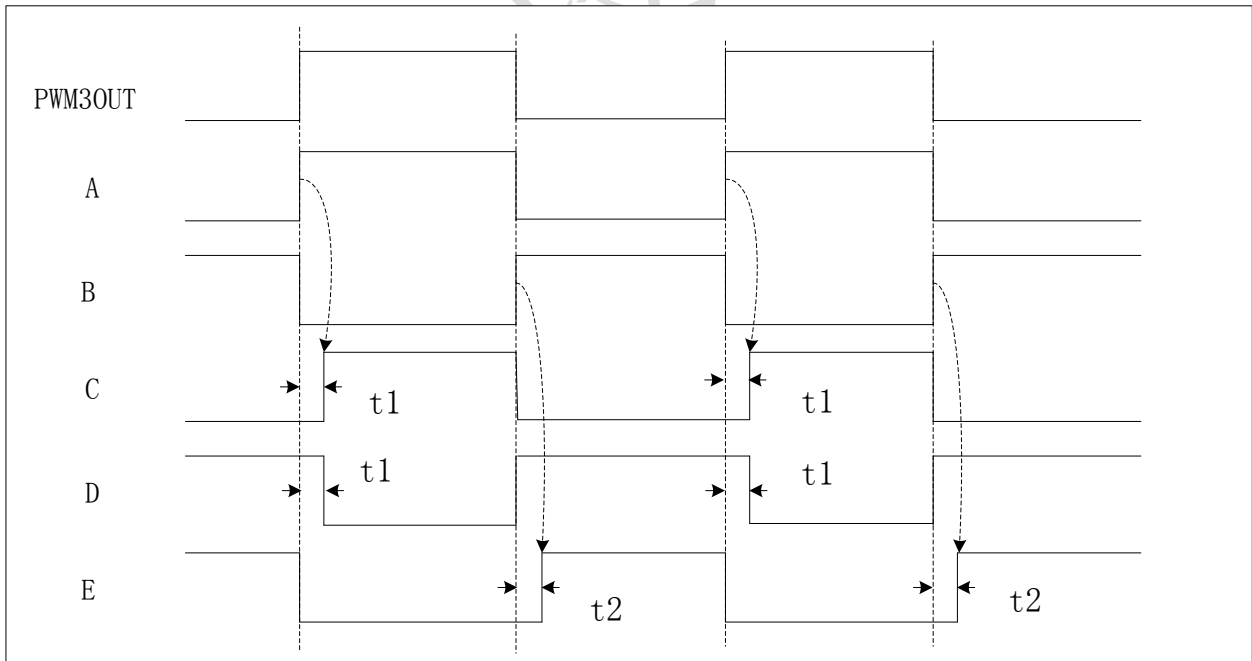
其中

$$t_1 = DT3CNT1[2:0] * DT3\_CLK$$

$$t_2 = DT3CNT2[2:0] * DT3\_CLK$$



PWM 输出取反后的互补 PWM 输出



### 3.6.5 小数分频

小数分频通过计数器在  $TM3R$  和  $TM3R \pm 1$  之间不断变化，通过  $TM3PNT\_DUTY[6:0]$  控制 2 种数值的比例，最终取平均得到小数分频。如下表



SIGN	TM3R	TM3PNT_DUTY[6:0]	小数分频值
0	10	0	10
	10	1	$10+1/128$
	10	2	$10+2/128$
	10	3	$10+3/128$
	·	·	·
	·	·	·
	·	·	·
1	10	127	$10+127/128$
	10	0	10
	10	1	$10-1/128$
	10	2	$10-2/128$
	10	3	$10-3/128$
	·	·	·
	·	·	·
·	·	·	
10	127	$10-127/128$	

3.7 定时/计数器 4

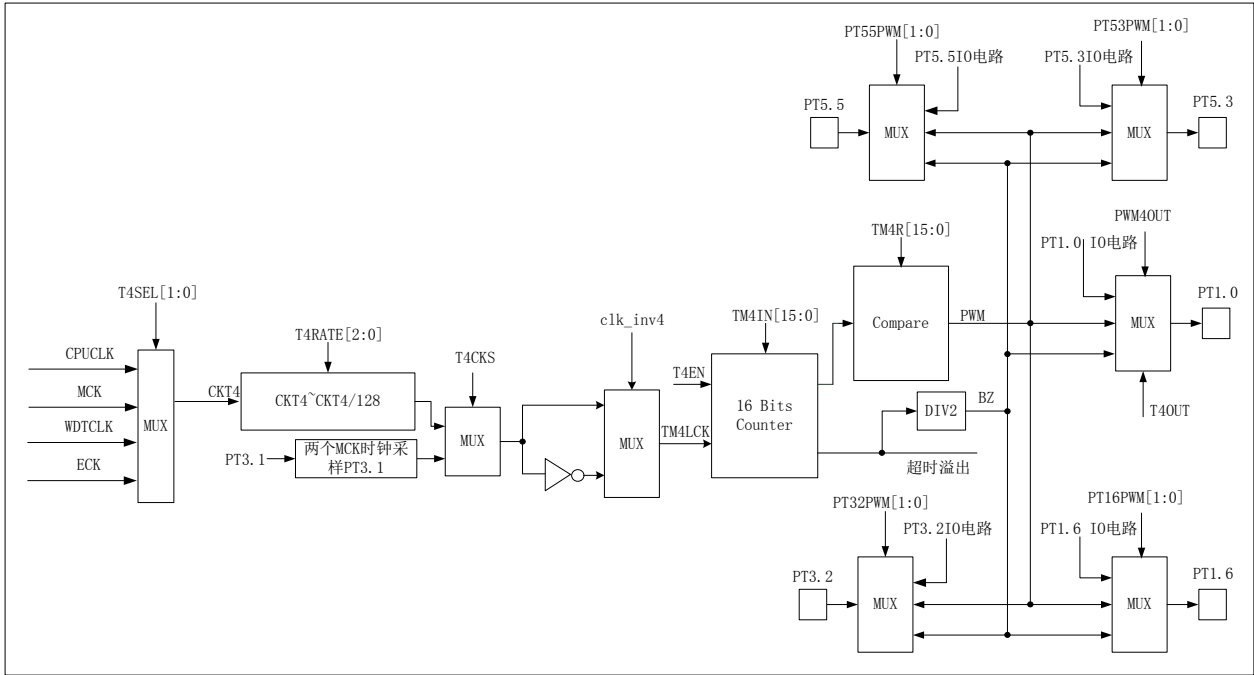


图28 定时/计数器 4 模块的功能框图

定时/计数器 4 模块的输入是 TM4CLK。当用户设置了定时/计数器 4 模块的使能标志，8 bits 计数器将启动，从 000h 递增到 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。

当设置定时器溢出时间及占空比时，必须先配置高位，再配置低位。例如，当需要将定时器 4 溢出值配置为 12'h0137，那么需要先配置 TM4INH 为 01h，然后配置 TM4IN 为 37h。对 TM4R 和 TM4RH 也必须同样操作。同时，不能在小于 5 个定时器时钟周期内，连续 2 次对 TM4INH/TM4IN 或者 TM4RH/TM4R 进行赋值，否则会导致 2 次赋值都失败。

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；

3.7.1 寄存器说明

表 52 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					uuu0uuuu
3dh	INTE2				TM4IE					uuu0uuuu
50h	TM4CON	T4EN	T4RATE[2:0]			T4CKS	T4RSTB	T4OUT	PWM4OUT	00000100
51h	TM4IN	TM4IN[7:0]								11111111
52h	TM4INH	TM4INH[15:8]								11111111
53h	TM4CNT	TM4CNT[7:0]								00000000
54h	TM4CNTH	TM4CNTH[15:8]								00000000
55h	TM4R	TM4R[7:0]								00000000





56h	TM4RH	TM4RH[15:8]						00000000
57h	TM4CON1	PT16PWM[1:0]	PT32PWM[1:0]		PT53PWM[1:0]		PT55PWM[1:0]	00000000
58h	TM4CON2	PWM4IS	CLK_INV4	T4SEL[1:0]			0u0u00uu	
59h	DT4CON0	DT4_EN	DT4CK[1:0]	P4L_IS	P4H_IS	DT4CNT1[2:0]		000uu000
5ah	DT4CON1	PWM4PO	P4LOEN	P4LINV	P4HOEN	P4HINV	PWM4DL1[2:0]	00000000
5bh	DT4CON2						DT4CNT2[2:0]	uuuuu000
5ch	CAPCON	CAPEN	CAPPIN[2:0]		CAPCFG	CAPFLT[1:0]		0000000u
5dh	CAPCON2						CAPMODE[1:0]	uuuuuu00
5eh	TM4CAPL	TM4CAP[7:0]						00000000
5fh	TM4CAPI	TM4CAP[15:8]						00000000

表 53 TM4CON 寄存器（地址为 50H）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
TM4CON	T4EN	T4RATE[2:0]			T4CKS	T4RSTB	T4OUT	PWM4OUT
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 时钟分频 <table border="1"> <tr> <th>T4RATE [2:0]</th> <th>TM4CLK</th> </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> 可通过 T4SEL [1:0]选择定时器 4 时钟	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 1: 外部计数时钟输入作为时钟（PT3.1） 0: CKT4 的分频时钟																		
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																		
1	T4OUT	PT1.0 口输出控制 <table border="1"> <tr> <th>T4OUT</th> <th>PWM4OUT</th> <th>PT1.0 输出控制，仅当 PT1.0 配置为输出有效</th> </tr> <tr> <td>0</td> <td>0</td> <td>普通 IO</td> </tr> <tr> <td>1</td> <td>0</td> <td>蜂鸣器输出</td> </tr> <tr> <td>0</td> <td>1</td> <td>PWM4 输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>PWM4 取反输出</td> </tr> </table>	T4OUT	PWM4OUT	PT1.0 输出控制，仅当 PT1.0 配置为输出有效	0	0	普通 IO	1	0	蜂鸣器输出	0	1	PWM4 输出	1	1	PWM4 取反输出			
T4OUT	PWM4OUT	PT1.0 输出控制，仅当 PT1.0 配置为输出有效																		
0	0	普通 IO																		
1	0	蜂鸣器输出																		
0	1	PWM4 输出																		
1	1	PWM4 取反输出																		
0	PWM4OUT	PT1.0 做 PWM 或者蜂鸣器输出时，PT1.0 必须配置为输出口																		

表 54 TM4IN 寄存器（地址为 51H）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
----	-------	-------	-------	-------	-------	-------	-------	-------

TM4IN	TM4IN[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7 : 0	TM4IN[7:0]	定时/计数器溢出值或捕获总计数值

表 55 TM4INH 寄存器（地址为 52H）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM4INH	TM4IN[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7 : 0	TM4IN[15:8]	定时/计数器溢出值或捕获总计数值

表 56 TM4CNT 寄存器（地址为 53H）各位功能表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM4CNT	TM4CNT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7 : 0	TM4CNT[7:0]	定时/计数器 4 计数寄存器，只读

表 57 TM4CNTH 寄存器（地址为 54H）各位功能表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM4CNTH	TM4CNT[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7 : 0	TM4CNT[15:8]	定时/计数器 4 计数寄存器，只读

表 58 TM4R 寄存器（地址为 55H）各位功能表

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
TM4R	TM4R[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7 : 0	TM4R[7:0]	定时/计数器 4 的 PWM 高电平占空比控制寄存器 或捕获高电平占空比

表 59 TM4RH 寄存器（地址为 56H）各位功能表

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
TM4RH	TM4R[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7 : 0	TM4R[15:8]	定时/计数器 4 的 PWM 高电平占空比控制寄存器或捕获高电平占空比

表 60 TM4CON1 寄存器（地址为 57H）各位功能表

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
TM4CON1	PT16PWM[1:0]		PT32PWM[1:0]		PT53PWM[1:0]		PT55PWM[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7 : 6	PT16PWM[1:0]	PT1.6 口输出控制
		PT16PWM[1:0] PT1.6 输出控制，仅当 PT1.6 配置为输出有效
		2'b00 普通 IO
		2'b01 蜂鸣器输出
		2'b10 PWM4 输出
		2'b11 PWM4 取反输出
PT1.6 做 PWM 或者蜂鸣器输出时，PT1.6 必须配置为输出口		
5 : 4	PT32PWM[1:0]	PT3.2 口输出控制
		PT32PWM[1:0] PT3.2 输出控制，仅当 PT3.2 配置为输出有效
		2'b00 普通 IO
		2'b01 蜂鸣器输出
		2'b10 PWM4 输出
		2'b11 PWM4 取反输出
PT3.2 做 PWM 或者蜂鸣器输出时，PT3.2 必须配置为输出口		
3:2	PT53PWM[1:0]	PT5.3 口输出控制
		PT53PWM[1:0] PT5.3 输出控制，仅当 PT5.3 配置为输出有效
		2'b00 普通 IO
		2'b01 蜂鸣器输出
		2'b10 PWM4 输出
		2'b11 PWM4 取反输出
PT5.3 做 PWM 或者蜂鸣器输出时，PT5.3 必须配置为输出口		
1 : 0	PT55PWM[1:0]	PT5.5 口输出控制
		PT55PWM[1:0] PT5.5 输出控制，仅当 PT5.5 配置为输出有效
		2'b00 普通 IO
		2'b01 蜂鸣器输出
		2'b10 PWM4 输出
		2'b11 PWM4 取反输出
PT5.5 做 PWM 或者蜂鸣器输出时，PT5.5 必须配置为输出口		

表 61 DT4CON0 寄存器（地址为 59H）各位功能表

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
DT4CON0	DT4_EN	DT4CK[1:0]		P4L_IS	P4H_IS	DT4CNT1[2:0]		



	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符		功能					
7	DT4_EN		定时器 4 互补式 PWM 输出使能位 0: 定时器 4 互补式输出禁止 1: 定时器 4 互补式 PWM 输出使能					
6:5	DT4CK[1:0]		定时器 4 死区时间时钟选择					
			DT4CK[1:0]	DT4_CLK				
			00	MCK				
			01	MCK/2				
			10	MCK/4				
			11	MCK/8				
4	P4L_IS		互补输出 PWM4L 空闲状态控制位（空闲状态是指定时器模块未使能或互补 PWM 输出未使能时输出状态） 0: 空闲时输出低电平 1: 空闲时输出高电平					
3	P4H_IS		互补输出 PWM4L 空闲状态控制位（空闲状态是指定时器模块未使能或互补 PWM 输出未使能时输出状态） 0: 空闲时输出低电平 1: 空闲时输出高电平					
2:0	DT4CNT1[2:0]		PWM4H 死区时间选择 死区时间=DT4CNT1[2:0]*DT4_CLK					

表 62 DT4CON1 寄存器（地址为 5AH）各位功能表

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
DT4CON1	PWM4PO	P4LOEN	P4LINV	P4HOEN	P4HINV	PWM4DL1[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	PWM4PO	互补 PWM4 输出引脚选择 0: 互补 PWM4 从 PT1.5, PT1.6 输出 1: 互补 PWM4 从 PT3.2, PT3.3 输出
6	P4LOEN	PWM4L 输出控制 0: 普通 IO 1: PWM4_L 输出
5	P4LINV	PWM4L 相位控制 0: PWM4_L 正常相位输出 1: PWM4_L 取反输出
4	P4HOEN	PWM4H 输出控制 0: 普通 IO 1: PWM4_H 输出
3	P4HINV	PWM4_H 相位控制 0: PWM4_H 正常相位输出 1: PWM4_H 取反输出
2:0	PWM4DL1[2:0]	互补 PWM4 延时控制 延时时间为 TM4CLK*PWM4DL1[2:0]

表 63 DT4CON2 寄存器（地址为 5BH）各位功能表

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DT4CON2						DT4CNT2[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
2:0	DT4CNT2[2:0]	PWM4L 死区时间选择 死区时间=DT4CNT2[2:0]*DT4_CLK

表 64 TM4CON2 寄存器（地址为 58H）各位功能表

特性	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0
TM4CON2			CLK_INV4		T4SEL[1:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
5	CLK_INV4	定时器 4 时钟取反 0: 定时器 4 时钟不取反 1: 定时器 4 时钟取反										
3: 2	T4SEL[1:0]	定时器 4 时钟选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>T4SEL[1:0]</th> <th>CKT4 时钟选择</th> </tr> <tr> <td>00</td> <td>CPUCLK</td> </tr> <tr> <td>01</td> <td>MCK</td> </tr> <tr> <td>10</td> <td>WDT CLK(32K)</td> </tr> <tr> <td>11</td> <td>ECK</td> </tr> </table>	T4SEL[1:0]	CKT4 时钟选择	00	CPUCLK	01	MCK	10	WDT CLK(32K)	11	ECK
T4SEL[1:0]	CKT4 时钟选择											
00	CPUCLK											
01	MCK											
10	WDT CLK(32K)											
11	ECK											

表 65 CAPCON 寄存器（地址为 5CH）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CAPCON	CAPEN		CAPPIN[2:0]			CAPCFG	CAPFLT[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能																
7	CAPEN	定时器 4 捕获功能使能 0: 禁止定时器 4 捕获功能 1: 使能定时器 4 捕获功能																
6:4	CAPPIN[2:0]	捕获输入脚选择位 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>CAPPIN[2:0]</th> <th>捕获输入脚</th> </tr> <tr> <td>000</td> <td>禁止捕获输入</td> </tr> <tr> <td>001</td> <td>使能捕获 PT10 输入</td> </tr> <tr> <td>010</td> <td>使能捕获 PT16 输入</td> </tr> <tr> <td>011</td> <td>使能捕获 PT32 输入</td> </tr> <tr> <td>100</td> <td>使能捕获 PT53 输入</td> </tr> <tr> <td>101</td> <td>使能捕获 PT55 输入</td> </tr> <tr> <td>11x</td> <td>禁止捕获输入</td> </tr> </table> 切换捕获脚时，必须先关闭捕获使能位，否则可能造成误触发	CAPPIN[2:0]	捕获输入脚	000	禁止捕获输入	001	使能捕获 PT10 输入	010	使能捕获 PT16 输入	011	使能捕获 PT32 输入	100	使能捕获 PT53 输入	101	使能捕获 PT55 输入	11x	禁止捕获输入
CAPPIN[2:0]	捕获输入脚																	
000	禁止捕获输入																	
001	使能捕获 PT10 输入																	
010	使能捕获 PT16 输入																	
011	使能捕获 PT32 输入																	
100	使能捕获 PT53 输入																	
101	使能捕获 PT55 输入																	
11x	禁止捕获输入																	

3	RESERVE	保留
2	CAPCFG	捕获计数器配置位，用于配置发生捕获事件时是否清除计数器计数值 0: 发生捕获事件时清除计数值 1: 发生捕获事件时不清除计数值
1: 0	CAPFLT[1:0]	捕获输入滤波选择
		CAPFLT[1:0] 捕获输入滤波选择
		00 不进行滤波
		01 2个 TM4 时钟周期滤波
		10 4个 TM4 时钟周期滤波
11 8个 TM4 时钟周期滤波		
		捕获输入滤波用于消除输入引脚的毛刺对捕获造成的影响，可以有效避免误触发。

表 66 CAPCON2 寄存器（地址为 5DH）各位功能表

特性	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
CAPCON2							CAPMODE[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7:2	RESERVE	保留
1:0	CAPMODE[1:0]	捕获模式选择
		CAPMODE[1:0] 捕获模式
		00 上升沿触发捕获
		01 下降沿触发捕获
		10 高电平触发捕获 设置为高电平触发捕获时，当 CAPCFG=0 时，计数器在上升沿清零，在下降沿触发捕获事件，记录当前计数值；当 CAPCFG=1 时，计数器在上升沿和下降沿都触发一次捕获事件，两次捕获事件记录的计数值的差值即为高电平宽度。
11 低电平触发捕获 设置为低电平触发捕获时，当 CAPCFG=0 时，计数器在下降沿清零，在上升沿触发捕获事件，记录当前计数值；当 CAPCFG=1 时，计数器在上升沿和下降沿都触发一次捕获事件，两次捕获事件记录的计数值的差值即为低电平宽度。		

表 67 TM4CAPL 寄存器（地址为 5EH）各位功能表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM4CAPL	TM4CAP[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7:0	TM4CAP[7:0]	定时器 4 捕获值低 8 位，只读

表 68 TM4CAPH 寄存器（地址为 5FH）各位功能表



特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM4CAPH	TM4CAP [15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7:0	TM4CAP [15:8]	定时器 4 捕获值高 8 位，只读

### 3.7.2 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 8 bits 计数器。
- 6) 当定时超时发生时，BZ 输出信号发生跳变，可作为蜂鸣器输出；寄存器标志位 TM4IF 会自复位，程序计数器会复位为 04h。

定时器 4 溢出时间计算方法:

$$\text{定时器 4 溢出时间} = (\text{TM4IN} + 1) / \text{TM4CLK.} \quad (\text{TM4IN 不为 0})$$

蜂鸣器周期计算方法:

$$\text{蜂鸣器周期} = (\text{TM4IN} + 1) * 2 / \text{TM4CLK.} \quad (\text{TM4IN 不为 0})$$

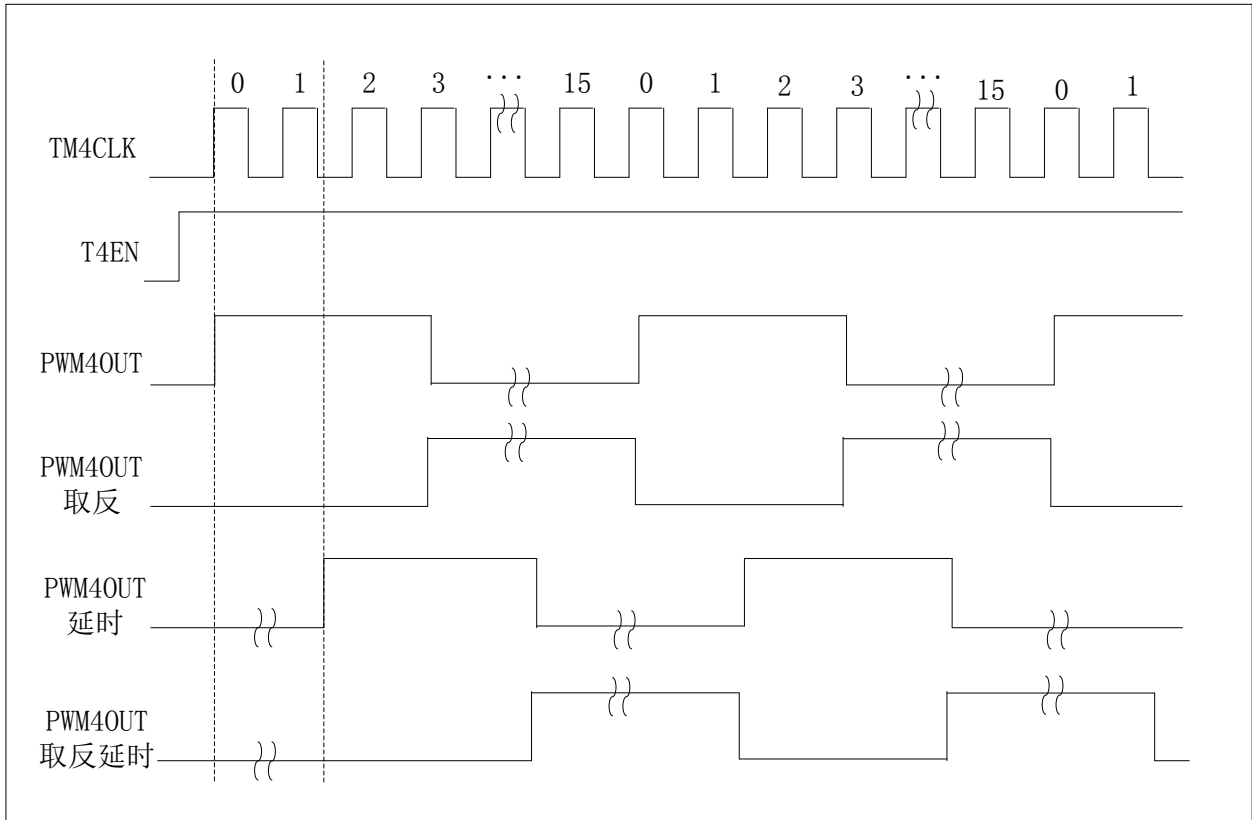
### 3.7.3 PWM

PWM 输出优先级

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 PT1.0 为输出端口，之后把 T4EN 置 1 启动定时器。
- 5) PWM4 从 PT1.0 输出。

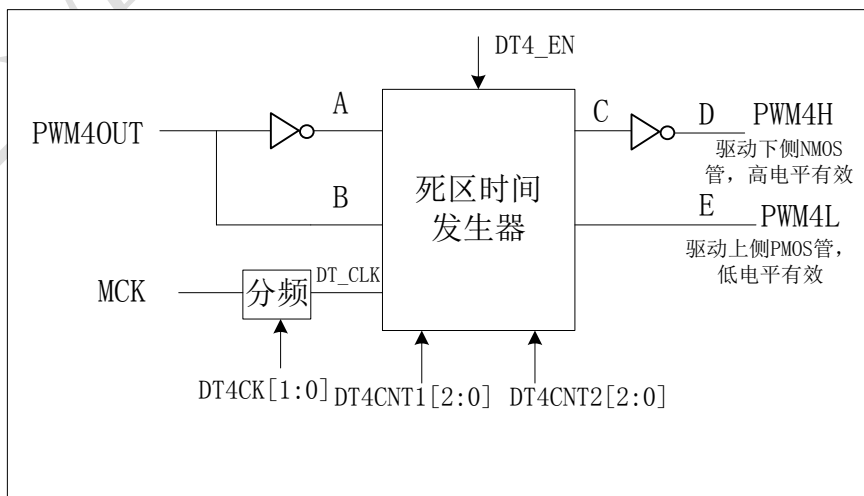
周期为 TM4IN+1，高电平脉宽为 TM4R。如 TM4IN=0x0F，TM4R=0x03 的 PWM4 波形输出如下:



### 3.7.4 互补式 PWM 输出

CSU38M20 提供源于定时器 4 的 PWM 输出的一对互补式输出，可用作 PWM 驱动信号。对于 PMOS 管上侧驱动，PWM 输出为低电平有效，而对于 NMOS 管下侧驱动，PWM 输出为高电平有效。当这对互补式输出同时用于驱动 PMOS 和 NMOS 时，死区时间发生器插入一死区时间以防止直流电流过大，该死区时间可通过 DT4CON0 寄存器的 DT4CK[1:0]和 DT4CNT[2:0]位来定义。在每个死区时间发生器输入信号的上升沿时插入一个死区时间。通过死区插入电路，输出信号最终发送至外部功率晶体管。通过 DT4CON1 寄存器的 PWM4PO 位，可以选择互补式 PWM 输出的位置。

图29 互补式 PWM 输出方框图





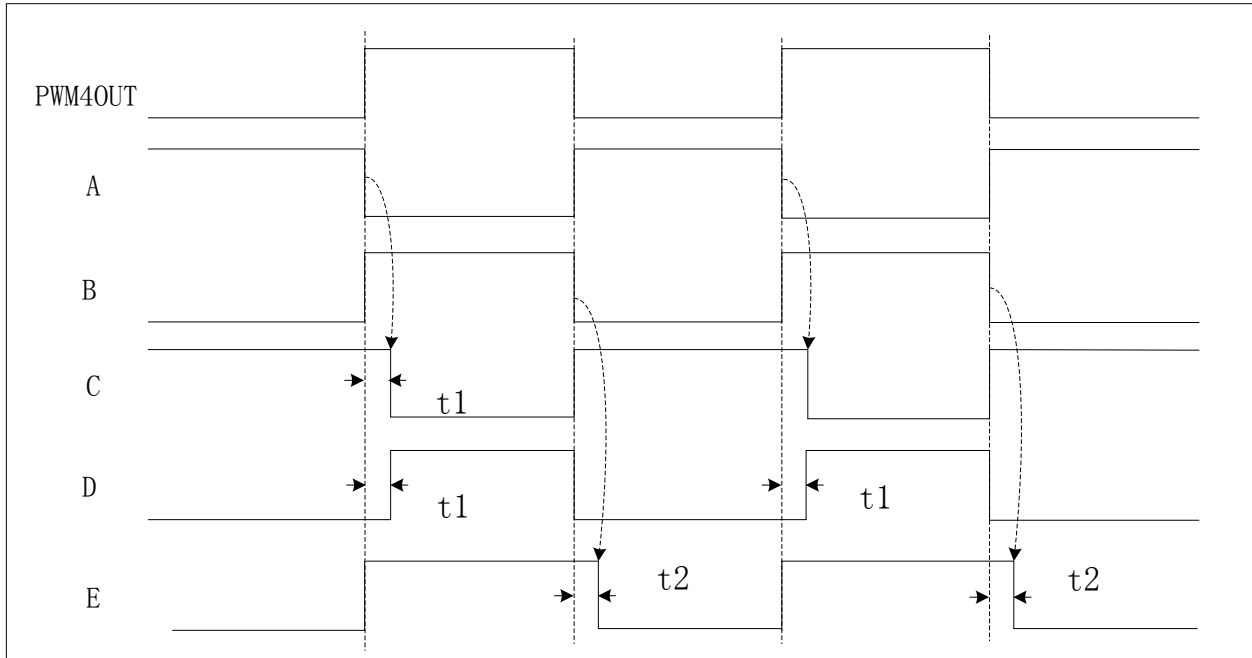
互补式 PWM 输出波形

PWM4OUT 上升沿增加时间为  $t_1$  的死区时间，下降沿增加时间为  $t_2$  的死区时间。

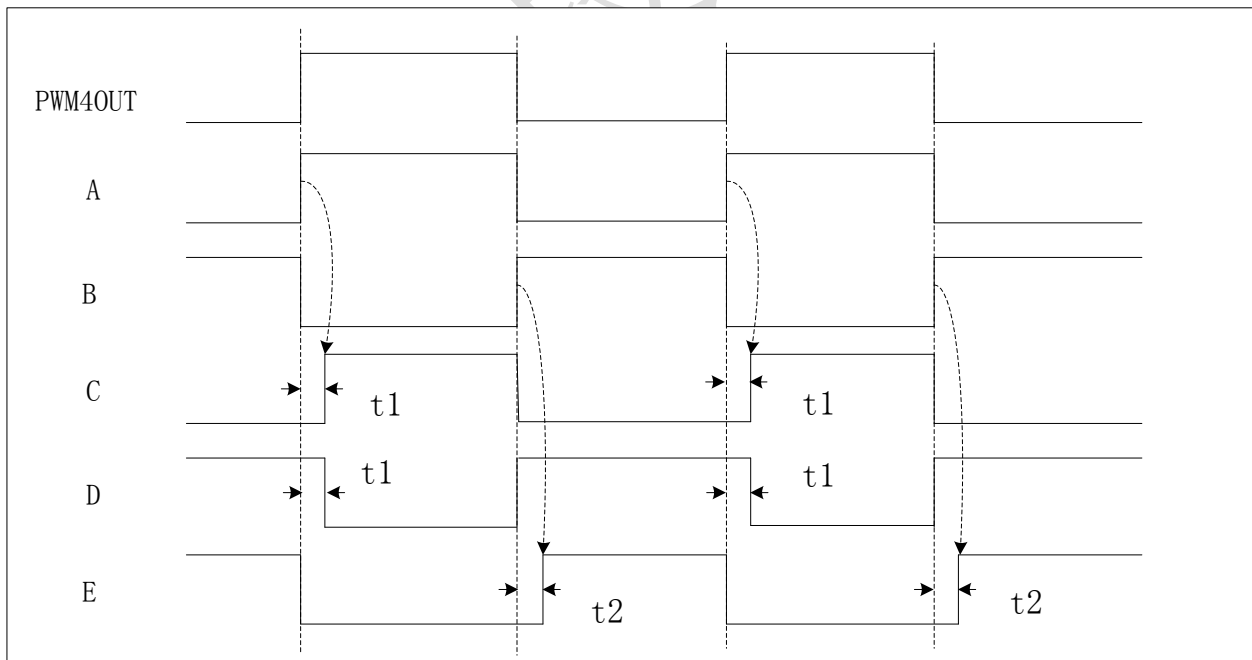
其中

$$t_1 = DT4CNT1[2:0] * DT4\_CLK$$

$$t_2 = DT4CNT2[2:0] * DT4\_CLK$$



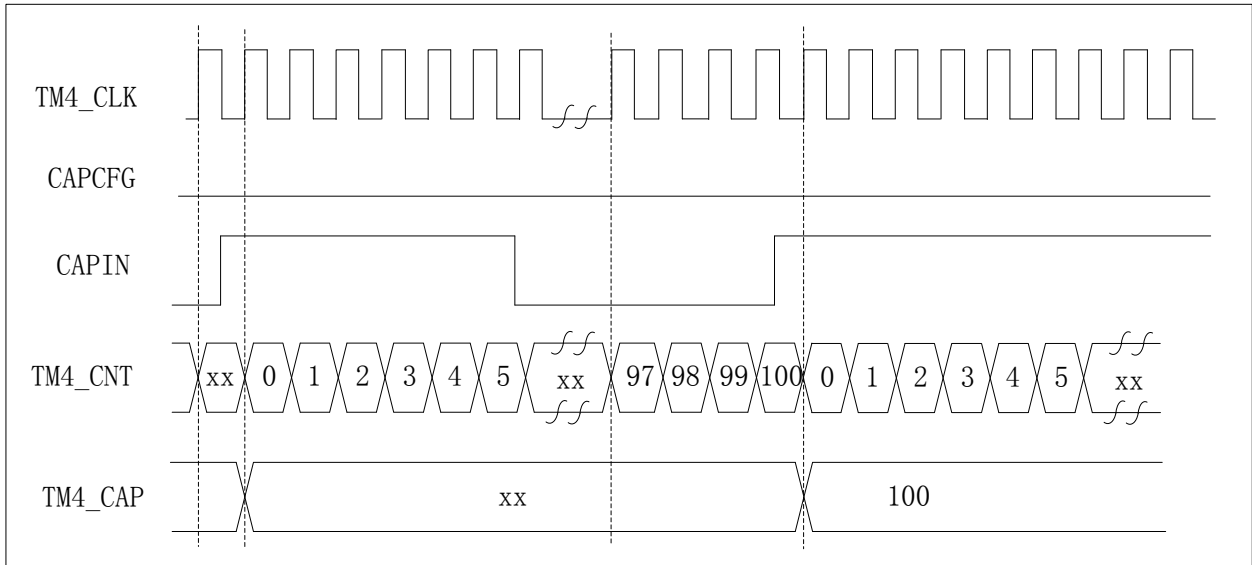
PWM 输出取反后的互补 PWM 输出



3.7.5 捕获

捕获功能可以用于测量外部信号周期或电平脉冲宽度。测周期波形如下





操作:

- 1) 设置 TM4CLK
- 2) 设置 CAPCON 来配置捕捉输入、发生捕获时是否清除计数器、捕捉输入滤波。
- 3) 设置 CAPCON 来配置捕捉模式。
- 4) 使能定时器 4 计数器，使能捕捉功能
- 5) 当产生捕获中断时，从 TM4CAPL、TM4CAPH 获取捕捉数值

### 3.8 串行通信接口(UART)

CSU38M20 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。

#### 3.8.1 工作方式

主要提供四种工作模式：

表 69 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	SMOD 波特率	11bits	1	1	0,1
				0 fcpuclk/32				
				1 fcpuclk/16				
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				
				1 定时器溢出率/16				

#### 模式 0:

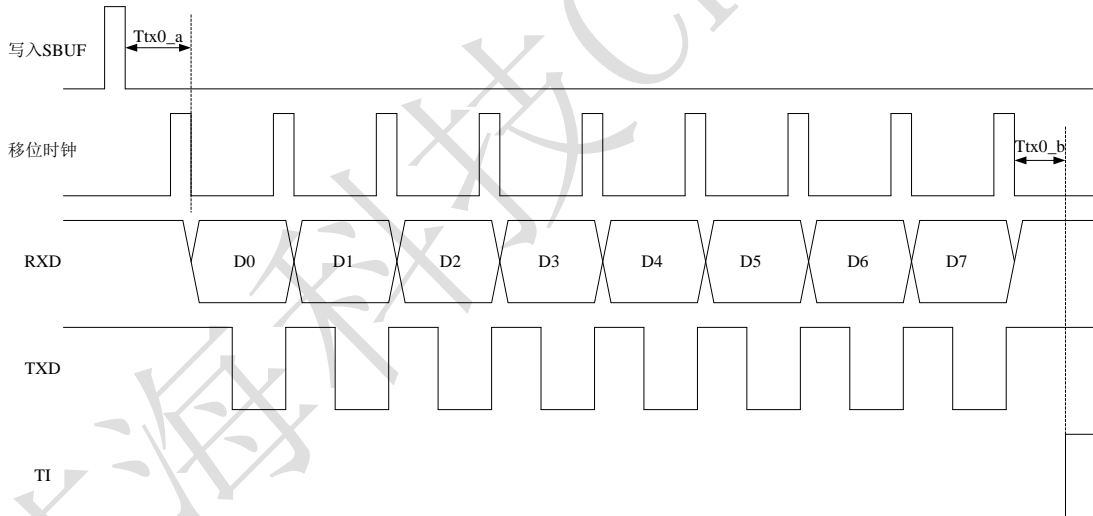


图30 UART 模式 0 发送数据波形

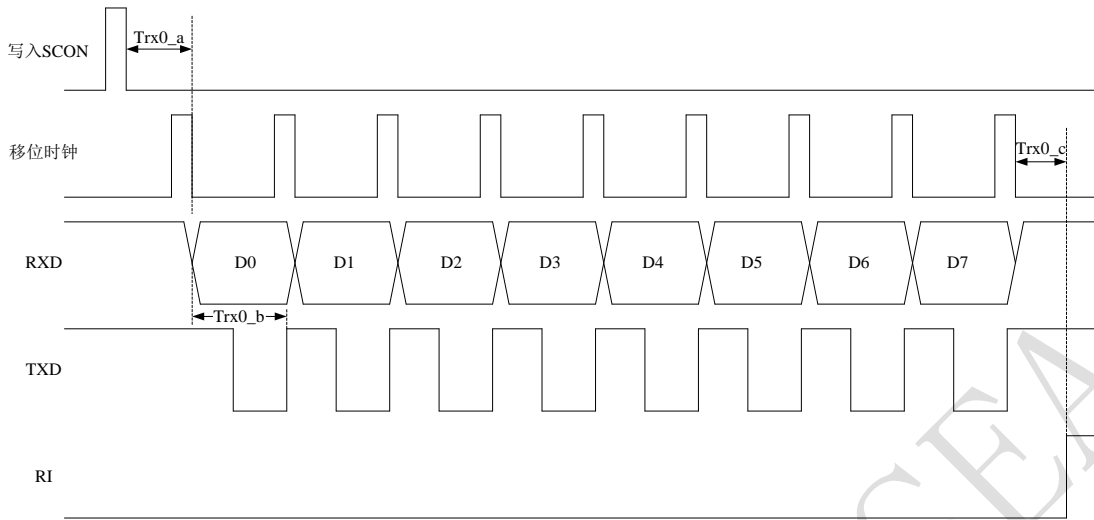


图31 UART 模式 0 接收数据波形

模式 1:

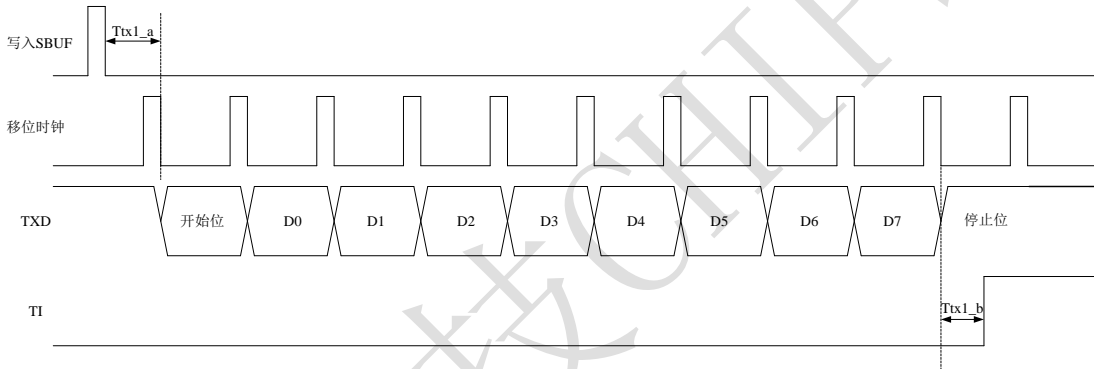


图32 UART 模式 1 发送数据波形

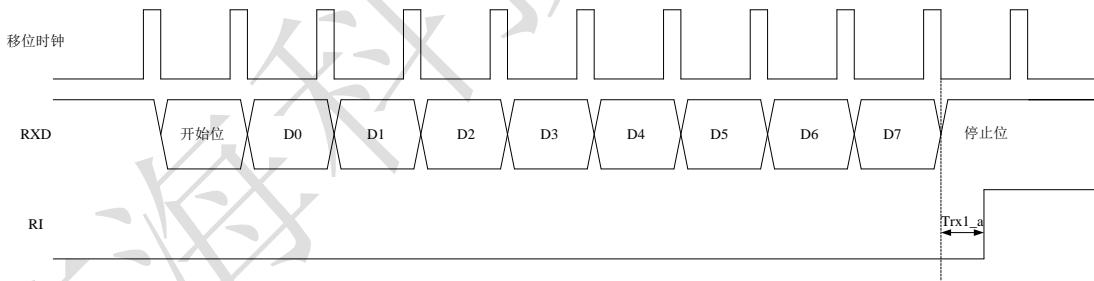


图33 UART 模式 1 接收数据波形

模式 2:

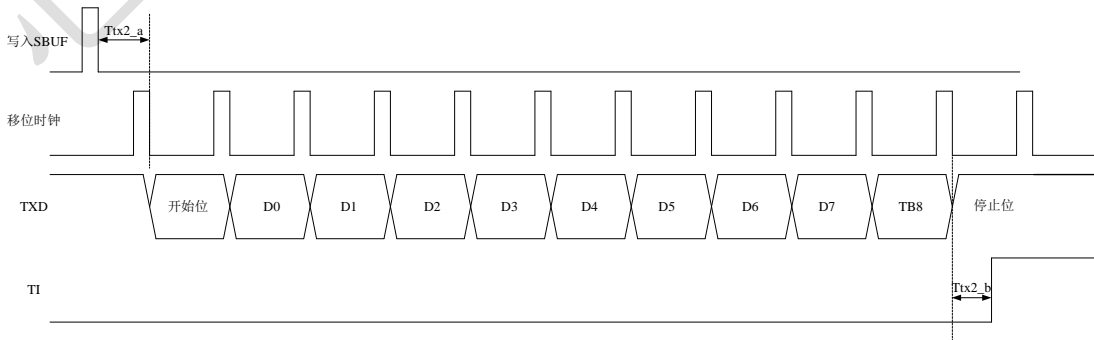


图34 UART 模式 2 发送数据波形

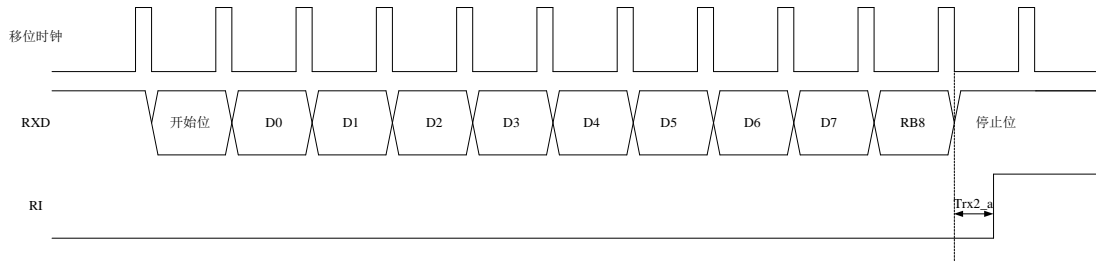


图35 UART 模式 2 接收数据波形

**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。

**3.8.2 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	00000000
3Fh	INTE3							URTIE	URRIE	
78h	SCON1	SM0	SM1	SM2	REN	TB8	RB8	URTIOS	URRIOS	00000000
79h	SCON2	SMOD	TCLKS[1:0]		RCLKS[1:0]		URIOS		UARTEN	000000u0
7ah	SBUF	SBUF[7:0]								00000000

表 70 SCON1 寄存器（地址为 78H）各位功能表

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	R/W-0
SCON1	SM0	SM1	SM2	REN	TB8	RB8	URTIOS	URRIOS	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

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 <a href="#">工作方式</a>																				
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																				
0	URRIOS	<table border="1"> <thead> <tr> <th>URIOS</th> <th>URTIOS</th> <th>URRIOS</th> <th>UART 发送口</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>PT3.0 做 RI 口，PT3.1 做 TI 口</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>PT3.1 既做 RI 口，也做 TI 口，分时复用；PT3.0 做普通 IO 口</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>PT3.0 既做 RI 口，也做 TI 口，分时复用；PT3.1 做普通 IO 口</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>PT3.0 做 TI 口，PT3.1 做 RI 口</td> </tr> </tbody> </table>	URIOS	URTIOS	URRIOS	UART 发送口	0	0	0	PT3.0 做 RI 口，PT3.1 做 TI 口	0	1	1	PT3.1 既做 RI 口，也做 TI 口，分时复用；PT3.0 做普通 IO 口	1	0	0	PT3.0 既做 RI 口，也做 TI 口，分时复用；PT3.1 做普通 IO 口	1	1	1	PT3.0 做 TI 口，PT3.1 做 RI 口
		URIOS	URTIOS	URRIOS	UART 发送口																	
		0	0	0	PT3.0 做 RI 口，PT3.1 做 TI 口																	
		0	1	1	PT3.1 既做 RI 口，也做 TI 口，分时复用；PT3.0 做普通 IO 口																	
1	0	0	PT3.0 既做 RI 口，也做 TI 口，分时复用；PT3.1 做普通 IO 口																			
1	1	1	PT3.0 做 TI 口，PT3.1 做 RI 口																			

	1	0	0	PT3.2 做 RI 口, PT3.3 做 TI 口
		0	1	PT3.3 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; PT3.3 做普通 IO 口
		1	1	PT3.2 做 TI 口, PT3.3 做 RI 口

注: 模式 0 下 URTIOS, URRIOS 必须配置为 2'b11。

表 71 SCON2 寄存器 (地址为 79H) 各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
SCON2	SMOD	TCLKS[1:0]		RCLKS[1:0]		URIOS		UARTEN
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	SMOD	波特率选择寄存器 参见 <a href="#">工作方式</a>
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
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
2	URIOS	UART IO 组选择
		0: 选择 PT3.0、PT3.1 做 UART 通信口 1: 选择 PT3.2、PT3.3 做 UART 通信口
1	RESERVE	保留



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，PT3.0 做接收口时，需将 PT3EN[0]置为 0，PT3.0 做发送口时，需将 PT3EN[0]置为 1。

表 72 SBUF 寄存器（地址为 7AH）各位功能表

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

位地址	标识符	功能
7:0	SBUF[7:0]	UART 发送或接收数据寄存器。 当进行发送时，只需向 SBUF 写入数据，即可启动 UART 进行数据发送。 当进行数据接收时，接收完成时数据将存储在 SBUF 中。当 SBUF 中的接收数据没有及时读取，UART 再次接收到数据时将覆盖上次接收的数据。

3.8.3 波特率

波特率 (K)	Fsys=16MHz (SMOD=1)				
	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,T4RATE}
1.2	1.2005	0.042	MCK	0340	01000
2.4	2.3981	-0.08	MCK	01A0	01000
4.8	4.807	0.16	MCK	00CF	01000
9.6	9.6154	0.16	MCK	0067	01000
19.2	19.2308	0.16	MCK	0033	01000
38.4	38.4615	0.16	MCK	0019	01000
51.2	50	-2.35	MCK	0013	01000
57.6	58.824	-2.1	MCK	0011	01000
102.4	100	-2.35	MCK	0009	01000
115.2	111.11	-3.55	MCK	0008	01000

### 3.9 SPI 接口

#### 3.9.1 SPI 接口介绍

CSU38M20 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

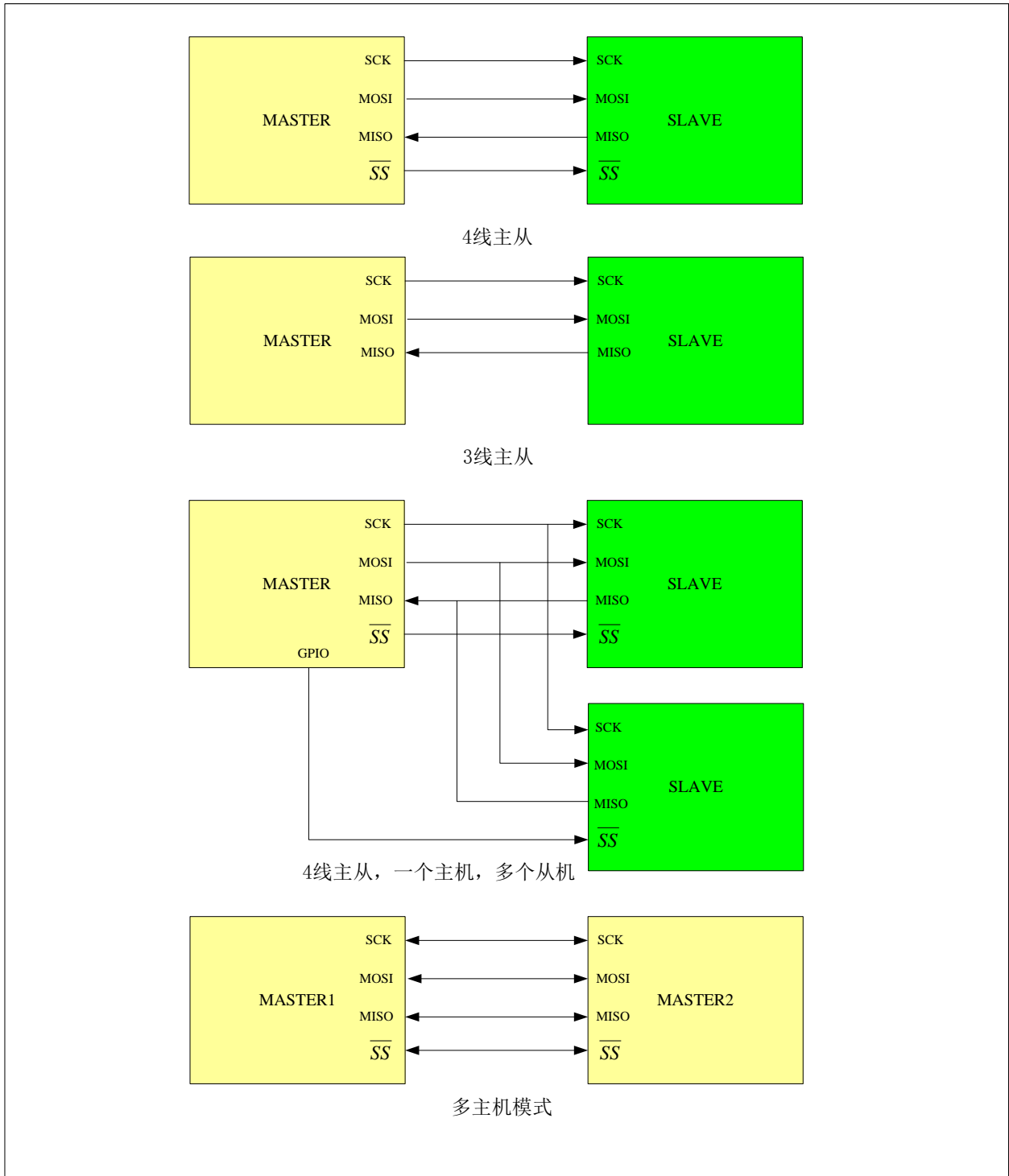
SPI 串行时钟支持 4 种方式，即两种相位和两种极性组合

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

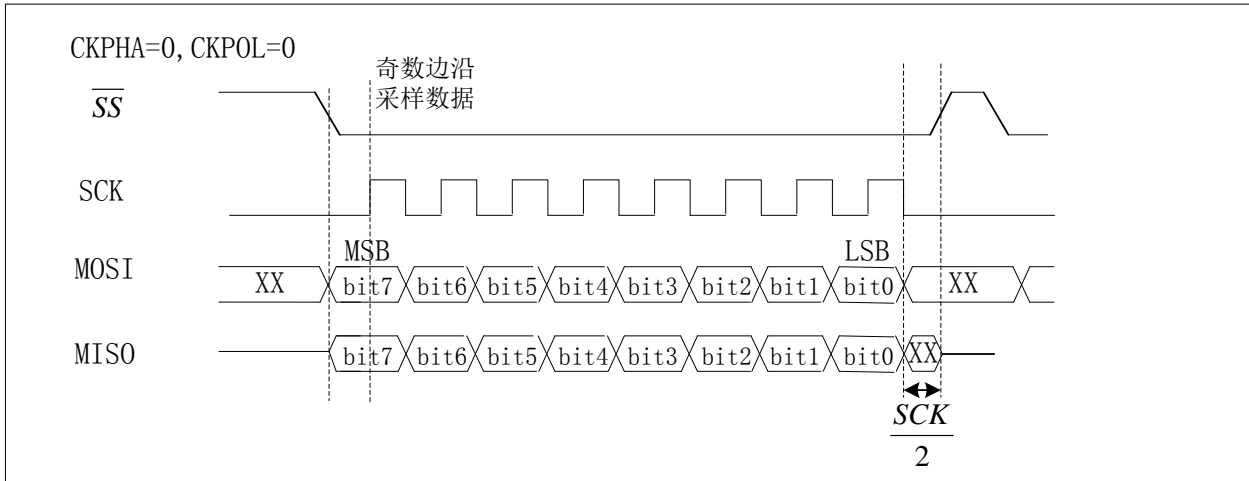
##### 3.9.1.1 SPI 连接方式



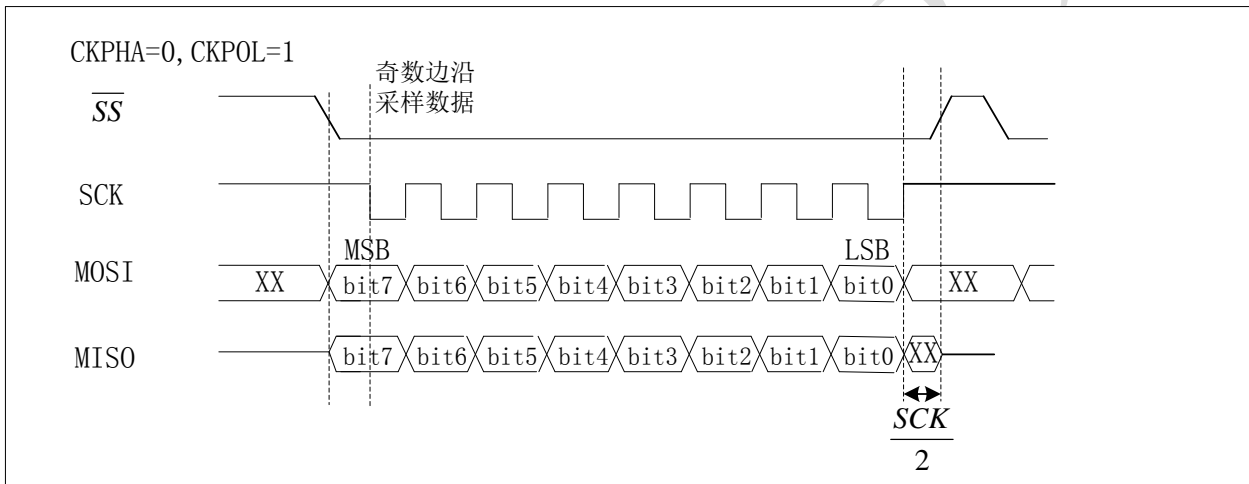


3.9.1.2 SPI 时序图

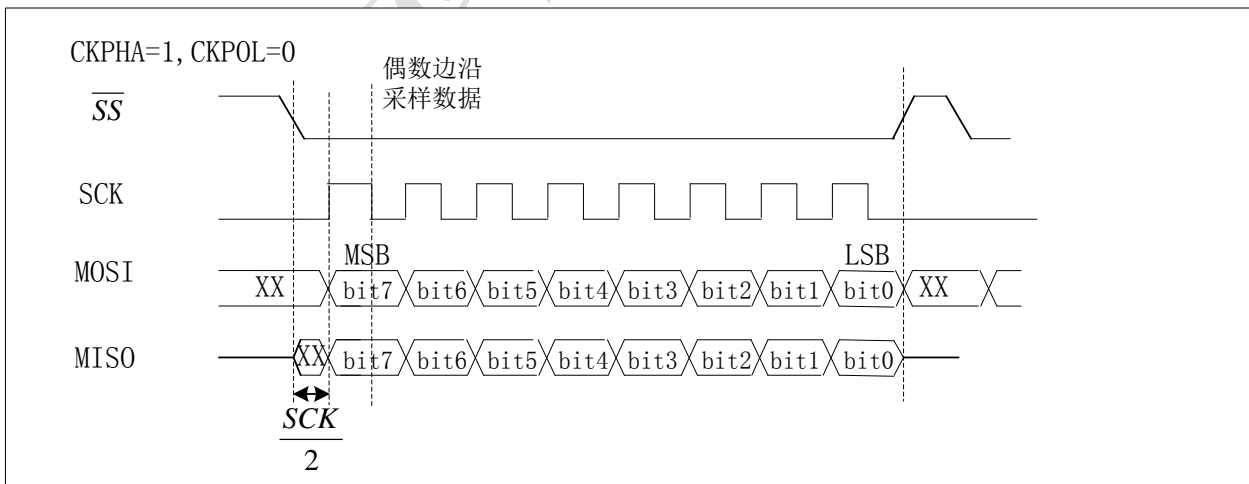
(1) CKPHA=0, CKPOL=0



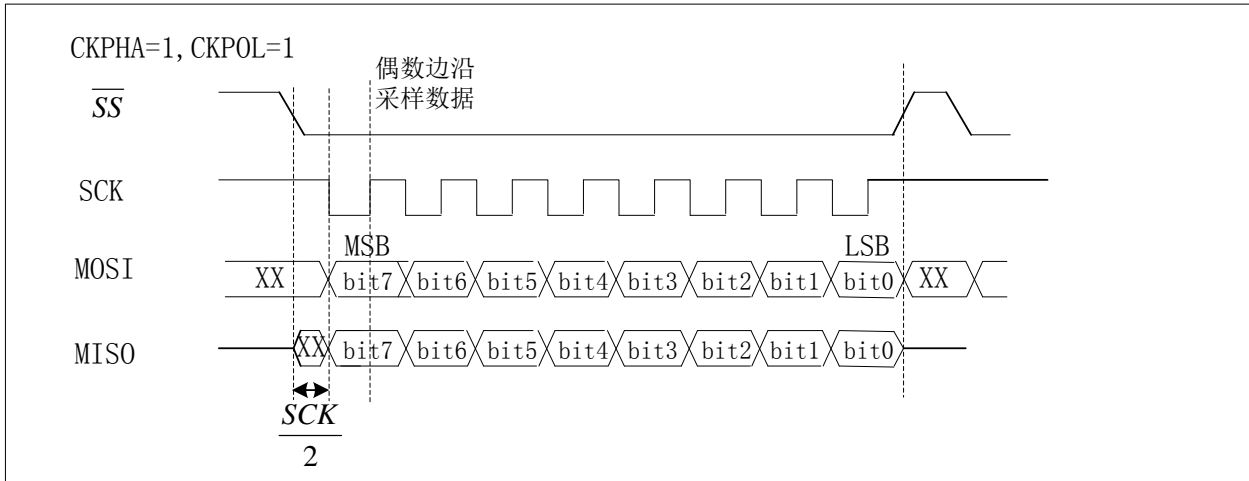
(2) CKPHA=0,CKPOL=1



(3) CKPHA=1,CKPOL=0



(4) CKPHA=1,CKPOL=1



3.9.2 SPI 寄存器说明

表 73 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
PAGE1										
6ch	SPICFG	SPIEN	MSTEN	CKPHA	CKPOL	MULMST	WIREMOD	SPIBSY	SPIRST	00000000
6dh	SPICN	SLVSEL	WCOL	MODCOL	SPIIOS					0000uuuu
6eh	SPICKR	SCR[7:0]								00000011
6fh	SPIDAT	SPIDAT[7:0]								00000000

表 74 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 75 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 76 SPICFG 寄存器（地址为 PAGE1 的 6CH）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
SPICFG	SPIEN	MSTEN	CKPHA	CKPOL	MULMST	WIREMOD	SPIBSY	SPIRST



	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
--	------	------	------	------	------	------	------	------

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 77 SPICN 寄存器（地址为 PAGE1 的 6DH）各位功能表

特性	R-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
SPICN	SLVSEL	WCOL	MODCOL	SPIIOS				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 0：选择 PT1.0, PT5.5, PT5.6, PT5.7 作为 SPI 通信口 1：选择 PT1.0, PT5.4, PT3.6, PT3.7 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
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 78 SPICKR 寄存器（地址为 PAGE1 的 6EH）各位功能表

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

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 79 SPIDAT 寄存器（地址为 PAGE1 的 6FH）各位功能表

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

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

### 3.9.3 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。

### 3.10 两线串行接口(I2C)

#### 3.10.1 I2C MASTER 模块

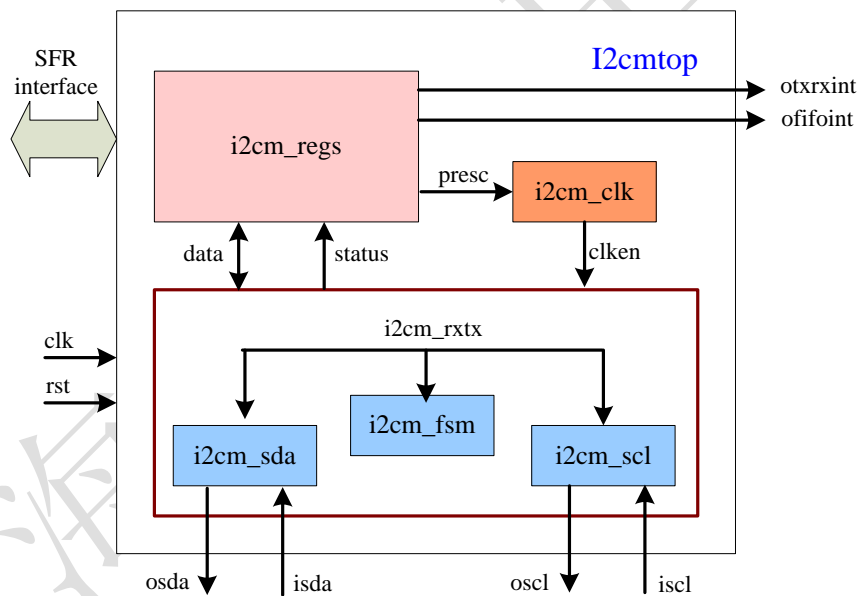


图36 I2CM 模块功能框图

#### 3.10.2 寄存器说明

表 80 I2CM 寄存器列表(PAGE1)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3eh	INTF3					I2CIF				0000u000
3fh	INTE3					I2CIE				0000u000
PAGE1										
20h	CTRL	I2C_EN	I2C_IOS	WAIT		STO	SOFTRST	STA	TXEXP	000u0000

									RG	
21h	RXDATA	RXDATA[7:0]								00000000
22h	TXDATA	TXDATA[7:0]								00000000
23h	PRESC	PRESC[7:0]								00000000
24h	TXRX_STS		TX_DATA_NACK	SA_NACK	TX_UNFLW	RX_OVFLW	NORMAL_END	uuu00000		
25h	DATA_STS		TX_EMPTY		TX_FULL	RX_EMPTY		RX_FULL	uu1u01u0	
26h	TXRX_IE		ETX_DATA_NACK	ESA_NACK	ETX_UNFLW	ERX_OVFLW	ENORMAL_END	uuu00000		
28h	CALLADDR0	RWN	CALLADDR0[6:0]						00000000	

表 81 INTF3 寄存器各位功能表

位地址	标识符	功能
2	I2CIF	I2C 中断标志位 1：发生 I2C 中断，必须软件清 0 0：没发生 I2C 中断

表 82 INTE3 寄存器各位功能表

位地址	标识符	功能
2	I2CIE	I2C 中断使能 1：使能 I2C 中断 0：关闭 I2C 中断

表 83 CTRL 控制器寄存器（地址为 PAGE1 的 20H）列表

特性	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R-0	R/W-0
CTRL	I2C_EN	I2C_IOS	I2CM_WAIT		I2CM_STO	I2CSOFT_RST	I2CM_STA	I2CM_TXEXPRTG
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	I2C_EN	I2C 模块使能位 0：禁止 I2C 通信 1：使能 I2C 通信
6	I2C_IOS	I2C 通信口选择 0：PT1.4、PT1.5 作为 I2C 通信口 1：PT3.4、PT3.5 作为 I2C 通信口
5	I2CM_WAIT	等待状态模式 1：当 RX 溢出时，在 SCL 上产生等待状态 0：当 RX 溢出时，发送“无应答”信号，停止传送
4	RESERVE	保留
3	I2CM_STO	产生 STOP 命令



		当该标志位为 1 时，当前字节正常结束，在应答周期会产生 STOP 信号；当开始传送或发现错误，控制器会清除该标志位。
2	I2CSOFTRST	软件复位 一旦 IDLE 状态到达，该标志位会自动清 0。
1	I2CM_STA	产生开始命令 当该标志位为 1 时，开始传送数据；当开始传送或发现错误，控制器会清除该标志位。
0	I2CM_TXEXPRG	发送或结束正在进行 当该标志位为 1 时，表明 I2C 帧正在 I2C 总线中传送，即 I2C 处于 BUSY 状态。

表 84 RXDATA 数据接收寄存器（地址为 PAGE1 的 21H）列表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RXDATA	I2CM_RXDATA[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
7:0	I2CM_RXDATA	只读，该寄存器用于被 CPU 读取接收的数据。						

表 85 TXDATA 数据发送寄存器（地址为 PAGE1 的 22H）列表

特性	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
TXDATA	I2CM_TXDATA[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
7:0	I2CM_TXDATA	只写，该寄存器用于被 CPU 写要发送的数据；只有发送缓冲器为空才能被写入。						

表 86 PRESC 时钟预分频寄存器（地址为 PAGE1 的 23H）列表

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
PRESC	I2CM_PRESC[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
7:0	I2CM_PRESC[7:0]	时钟预分频寄存器 $F_{scl} = \frac{F_{CPUCLK}}{5 * (PRESC + 1)}$						

表 87 TXRX\_STS 发送接收状态寄存器（地址为 PAGE1 的 24H）各位功能描述

特性	U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
TXRX_STS				I2CM_TX_DATA_NACK	I2CM_SA_NACK	I2CM_TX_UNF_L	I2CM_RX_OVF_LW	I2CM_NORMAL_END
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
4	I2CM_ TX_DATA_NACK	<b>发送数据无应答（只读）</b> 在发送过程中数据字节没有应答，会发送一个 STOP 信号
3	I2CM_ SA_NACK	<b>从机地址无应答（只读）</b> 从机地址没有应答，会发送一个 STOP 信号。
2	I2CM_ TX_UNFL	<b>发送下溢（只读）</b> 需要数据发送但还没准备好，会发送一个 STOP 信号。
1	I2CM_ RX_OVFLW	<b>接收溢出（只读）</b> 接收寄存器满而无法被写入新接收的数据会发送“无应答”信号和 STOP 信号。
0	I2CM_ NORMAL_END	<b>正常结束（只读）</b> 当正常结束发送 STOP 命令后置 1；当新的 I2C 访问开始会自动清 0。

当 I2C 中断被响应后，除了清除 I2CIF 标志位外，还需要对 TXRX\_STS 寄存器进行一次读操作和一次写操作来清除 I2C 状态标志位，如果不进行该操作，系统会一直响应 I2C 中断。对寄存器 TXRX\_STS 进行读写操作时，第 0 位~第 4 位会自动清 0。

表 88 DATA\_STS 数据状态寄存器（地址为 PAGE1 的 25H）列表

特性	U-0	U-0	R-1	U-0	R-0	R-1	U-0	R-0
DATA_S TS			I2CM_ TX_EMP TY		I2CM_ TX_FUL L	I2CM_ RX_EMP TY		I2CM_ RX_FUL L
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
5	I2CM_ TX_EMPTY	<b>发送缓冲器空标志（只读）</b> 1：发送缓冲器为空。CPU 对 TXDATA 寄存器进行写操作时会清除该标志位。 0：至少还有一个发送的数据字节是可用的。
3	I2CM_ TX_FULL	<b>发送缓冲器满标志（只读）</b> 1：发送缓冲器为满。TXRX 控制器要求新的数据时会清除该标志位。 0：发送缓冲器为空。
2	I2CM_ RX_EMPTY	<b>接收缓冲器空标志（只读）</b> 1：接收缓冲器为空。TXRX 控制器要求新的数据时会清除该标志位。 0：至少还有一个接收的数据是可用的。
0	I2CM_ RX_FULL	<b>接收缓冲器满标志（只读）</b> 1：接收缓冲器为满。CPU 对 RXDATA 寄存器进行写操作会清除该标志位。 0：接收缓冲器为空。

表 89 TXRX\_IE 发送接收中断使能寄存器（地址为 PAGE1 的 26H）列表

特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXRX_IE				I2CM_ ETX_DA TA_NAC	I2CM_ ESA_NA CK	I2CM_ ETX_UN FL	I2CM_ ERX_OV FLW	I2CM_ ENORM AL_END



				K				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
4	I2CM_ETX_DATA_NACK	使能发送数据无应答中断
3	I2CM_ESA_NACK	使能从机地址无应答中断
2	I2CM_ETX_UNFL	使能发送下溢中断
1	I2CM_ERX_OVFLW	使能接收溢出中断
0	I2CM_ENORMAL_END	使能正常结束中断

表 90 CALLADDR0 呼叫地址寄存器（地址为 PAGE1 的 28H）列表

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
CALLADDR0	I2CM_RWN	CALLADDR0[6:0]						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	I2CM_RWN	<b>I2C 发送的读写控制位</b> 0: 主机发送 1: 主机接收
6:0	CALLADDR0	<b>呼叫地址</b> 在 I2C 开始发送之前该寄存器必须被写。

### 3.10.3 I2C 中断

I2C 支持多种中断，详见 TXRX\_STS 寄存器。当 I2C 中断被响应后，除了清除 I2CIF 标志位外，还需要对 TXRX\_STS 寄存器进行一次读操作和一次写操作来清除 I2C 状态标志位，如果进行读写操作，系统会一直响应 I2C 中断。对寄存器 TXRX\_STS 进行读写操作时，第 0 位~第 4 位会自动清 0。

### 3.10.4 I2CM 的操作模式

#### 3.10.4.1 主机发送模式

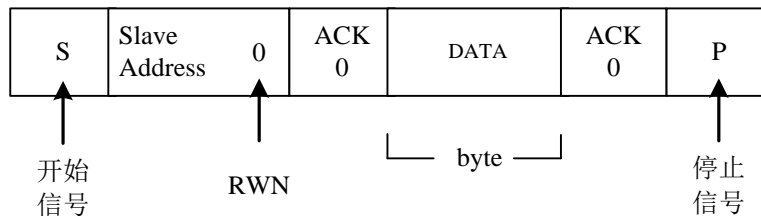


图37 I2CM 主机发送过程图

#### (1) 初始化

在开始传输之前，要把 slave 的地址写入 CALLADDR0 寄存器，且 RWN 必须为 0。（2）开始

设置 CTRL 寄存器的 I2CSTA 位为 1，I2C 控制器在总线上产生开始信号，随后 I2CSTA 将被清 0。开始信号产生之后，slave 地址将载入移位寄存器，按从低位到高位顺序在总线传输，8 位数据发送完之后，等待 slave 发送应答信号。

(3) 数据发送

接收到 slave 的应答信号之后，I2C 控制器把数据载入移位寄存器，开始 8 位数据的传输。如果还有数据传输，I2C 控制器发送请求先把接下来要发送的数据载入移位寄存器，然后继续传输。

(4) 停止或重复传输

最后一个字节的数据传输完成之后，I2C 控制等待 slave 的应答信号。在接收应答信号时，如果 I2CSTA 位为 1，I2C 控制器在总线上产生 Start 信号，开始一次新的传输，以便访问其它的 slave 器件或者改变数据方向；如果 I2CSTA 位为 0，CTRL 控制寄存器的 NORMAL\_END 置 1，I2C 控制器在总线上产生 Stop 信号，停止与 slave 通信。

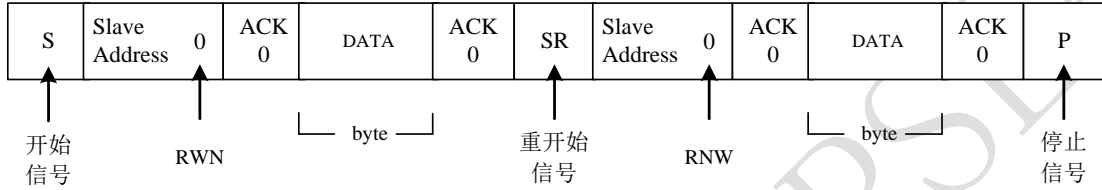


图38 I2CM 主机发送重复传输过程图

主机发送流程图如下：

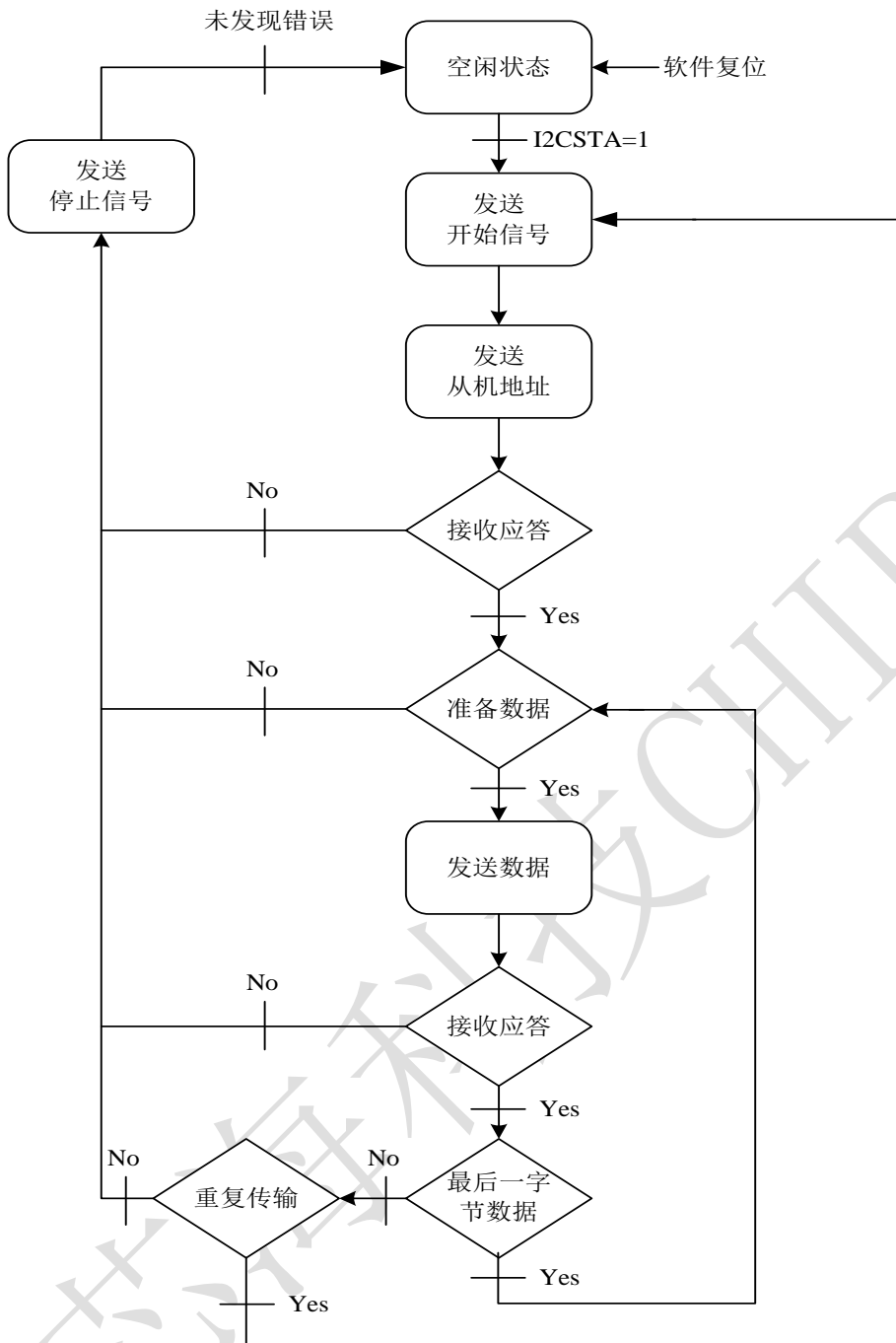


图39 I2CM 主机发送流程图

3.10.4.2 主机接收模式

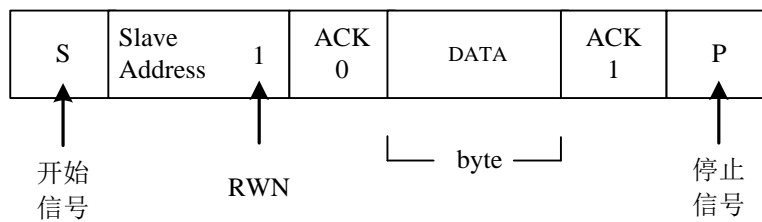


图40 I2CM 主机接收过程图

(1) 初始化

在开始传输之前，要把 slave 的地址写入 CALLADDR0 寄存器，且 RWN 必须为 1。

(2) 开始

设置 CTRL 寄存器的 I2CSTA 位为 1，I2C 控制器在总线上产生开始信号，随后 I2CSTA 将被清 0。开始信号产生之后，slave 地址将载入移位寄存器，按从低位到高位顺序在总线传输，8 位数据发送完之后，等待 slave 发送应答信号。

(3) 接收数据

接收到 slave 的应答信号之后，I2C 控制器把接收到的一字节数据存入数据接收寄存器，如果接收到的数据不是最后一字节的数据，I2C 控制器在 I2C 总线上发送“应答”信号，开始下一字节的数据接收；否则，I2C 控制器在 I2C 总线上发送“无应答”信号，slave 必须释放 I2C 总线，让 I2C 控制器产生 Stop 信号。

(4) 停止或重复传输

I2C 控制器在 I2C 总线上发送“无应答”信号后，如果 I2CSTA 位为 1，I2C 控制器在总线上产生 Start 信号，开始一次新的传输，以便访问其它的 slave 器件或者改变数据方向；如果 I2CSTA 位为 0，CTRL 控制寄存器的 NORMAL\_END 置 1，I2C 控制器在总线上产生 Stop 信号，停止与 slave 通信。

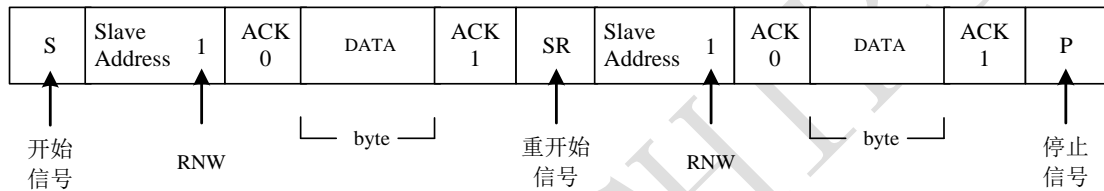


图41 I2CM 主机接收重复传输过程图

主机接收流程图如下：

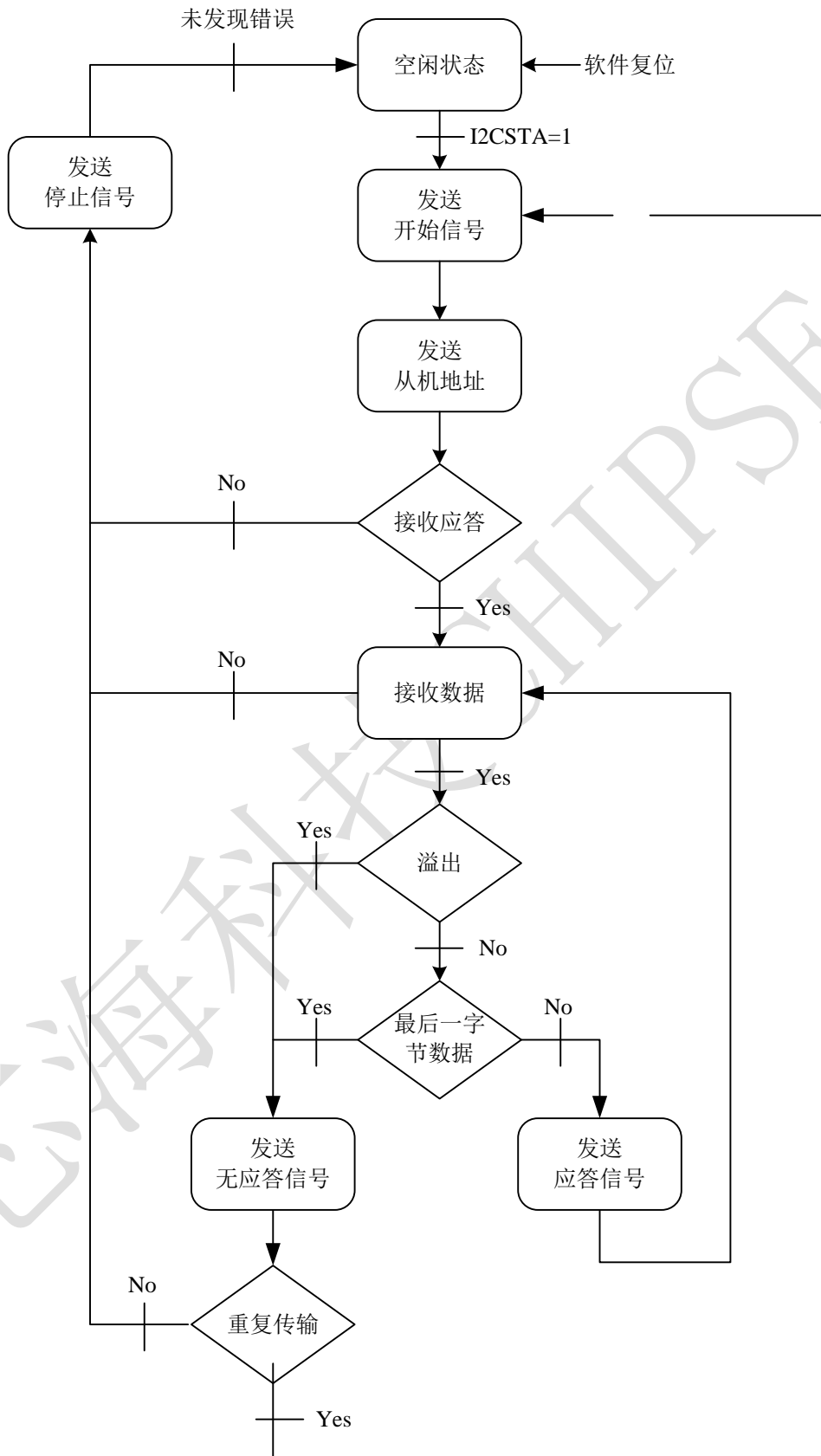


图42 I2CM 主机接收流程图

3.11 模数转换器(ADC)

CSU38M20 模数转换模块共用 26 条外部通道 (AINP0/AINN0~AINP12/AINN12) 和 5 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; 运放放大 16 倍的输出作为 AD 输入), 可以将模拟信号转换成 12 位数字信号, 其中最高位为符号位。进行 AD 转换时, 首先要选择输入通道(AINP0/AINN0~AINP12/AINN12)以及配置 ADC 输入选择, 然后把 SRADEN 置 1 使能 ADC, 之后把 SRADS 置 1, 启动 AD 转换。转换结束后, 系统自动将 ADS 清 0, 并将转换结果存入寄存器 SRADL 和 SRADH 中。

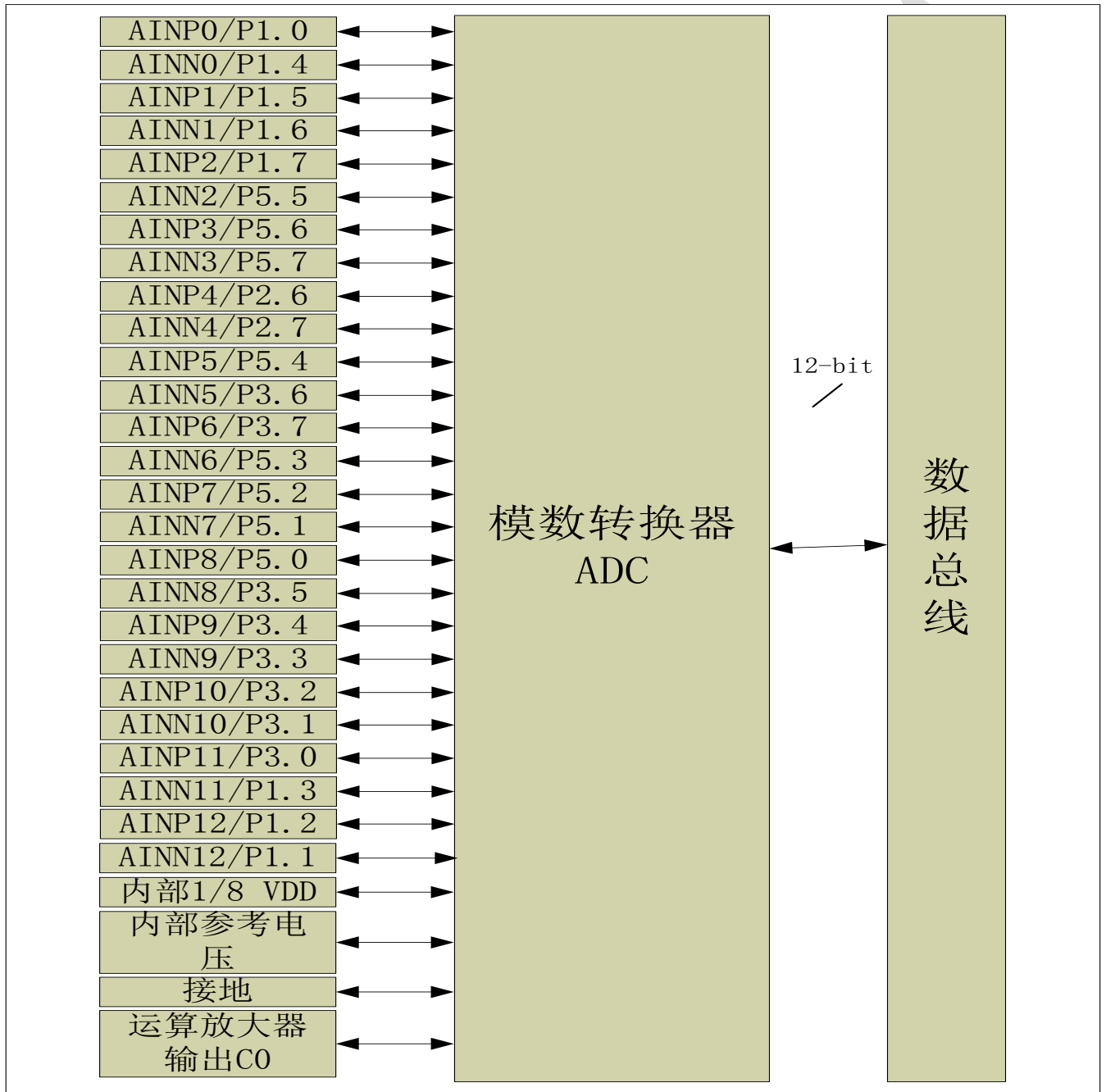


图43 模数转换器 ADC 功能框图



3.11.1 全差分输入 AD 转换结果

当 DMOF=0 时，AD 转换结果为有符号数，其中最高位为 0 表示负值，最高位为 1 表示正值。

当 DMOF=1 时，AD 转换结果为 2 的补码，最高位为符号位，最高位为 1 时表示负值，最高位为 0 时表示正值。

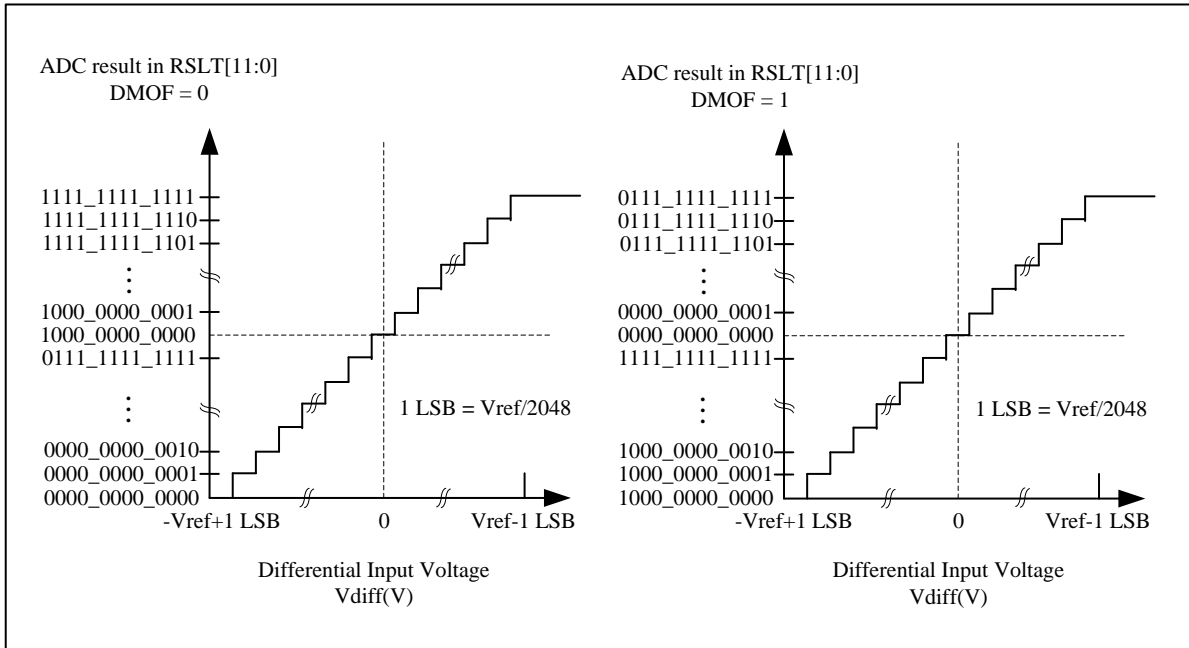


图44 全差分输入和单端输入负端接地时 AD 转换结果

当配置为单端输入，AINPn(n=0~12)，且负端接 VREF，即 AINSEL[2:0]配置为 101 时，AD 转换结果如下

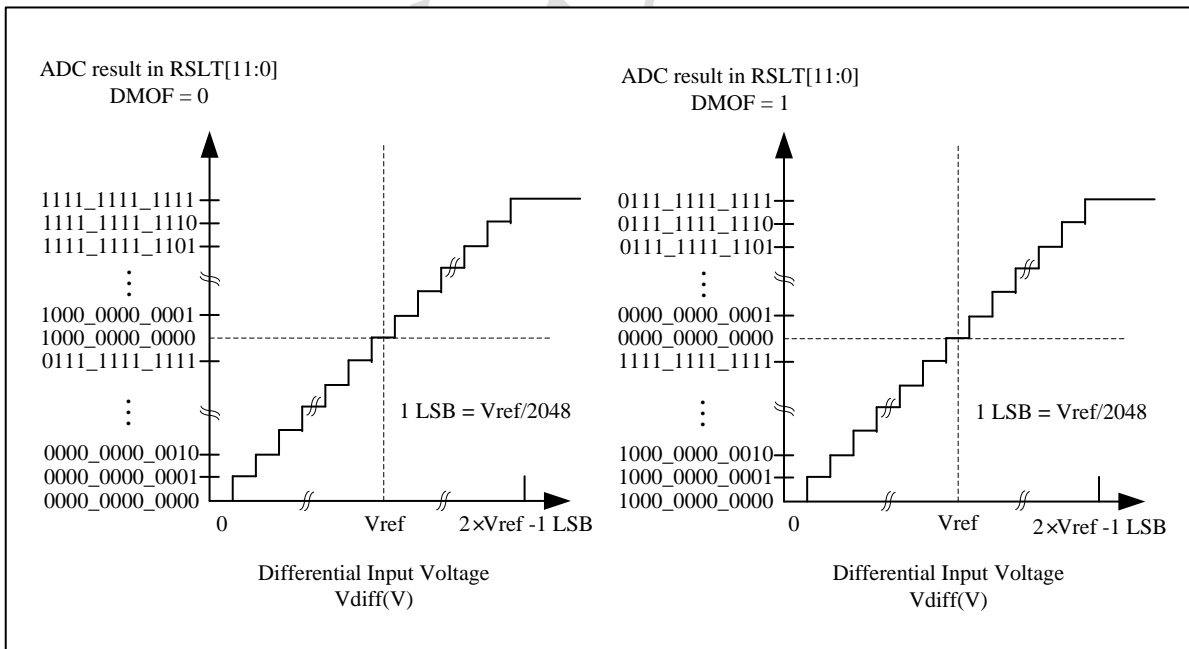


图45 AINPn 做正端输入，负端接 VREF 时的 AD 转换结果

当配置为单端输入，AINNn(n=0~12)，且正端接 VREF，即 AINSEL[2:0]配置为 110 时，AD 转换结果如下



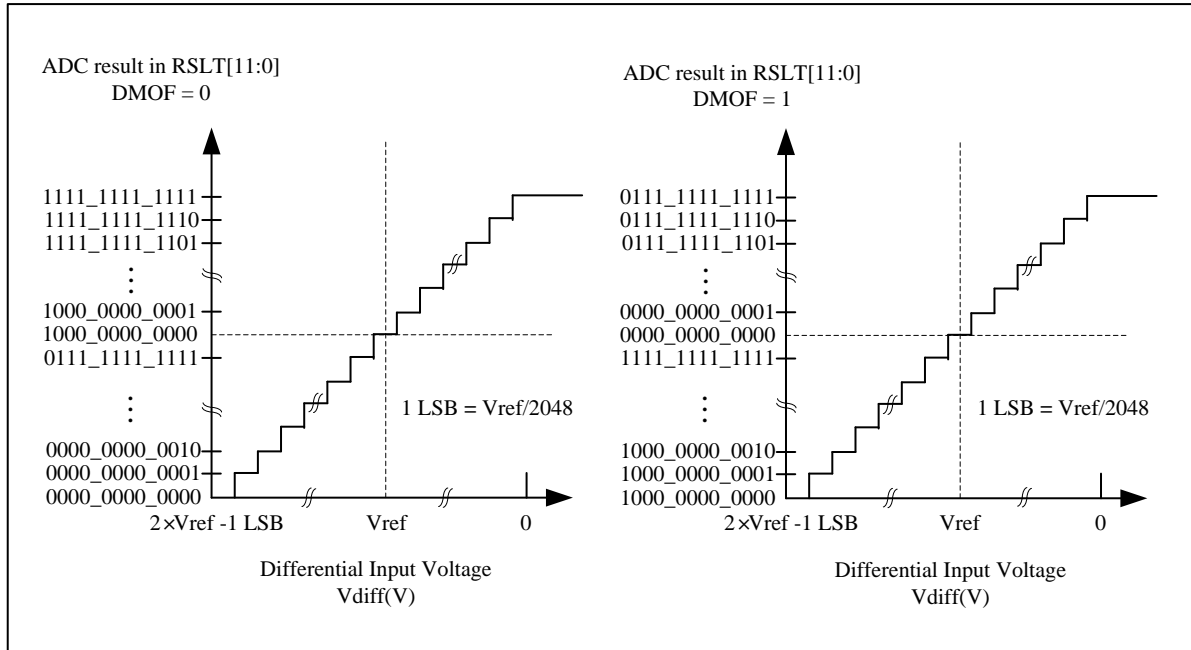


图46 AINNn 做负端输入，正端接 VREF 时的 AD 转换结果

3.11.2 寄存器说明

表 91 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				u0u0u00
07h	INTE	GIE				ADIE				00u00u00
60h	SRADCON0	SOURCE_MOD[1:0]		SRADACKS[1:0]		DMOF	CMP0_R	SRADCKS[1:0]		uu00uu00
61h	SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]		00000000
62h	SRADCON2	CHS[4:0]					AUTOS	REF_SEL[1:0]		00000000
63h	SRADCON3	SRADSMOD[2:0]			SOURCE_SEL[1:0]		SRADSDLY[2:0]			00000000
64h	SRADCON4	AINSEL[2:0]			IO_REF_EN[4:0]					00000000
66h	SRADL	SRAD[7:0]								00000000
67h	SRADH					SRAD[11:8]			uuuu0000	
68h	SROFTL	SROFT[7:0]								00000000
69h	SROFTH					SROFT[11:8]			uuuu0000	

表 92 SRADCON0 寄存器（地址为 60H）各位功能表

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
SRADCON0	SOURCE_MOD[1:0]		SRADACKS[1:0]		DMOF	CMP0_R	SRADCKS[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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： <table border="1" style="width: 100%;"> <tr> <td>SOURCE_MOD[1:0]</td> <td>转换触发源触发模式</td> </tr> <tr> <td>2'b00</td> <td>上升沿触发</td> </tr> </table>	SOURCE_MOD[1:0]	转换触发源触发模式	2'b00	上升沿触发
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 <table border="1"> <tr> <td>SRADACKS[1:0]</td> <td>ADC 输入信号获取时间</td> </tr> <tr> <td>00</td> <td>16 个 ADC 时钟</td> </tr> <tr> <td>01</td> <td>8 个 ADC 时钟</td> </tr> <tr> <td>10</td> <td>4 个 ADC 时钟</td> </tr> <tr> <td>11</td> <td>2 个 ADC 时钟</td> </tr> </table> <p>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。</p>		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
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												
3	DMOF	差分输入模式输出格式 1 = A/D 转换结果为 2 的补码 0 = A/D 转换结果为有符号数(直接二进制格式)											
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 <table border="1"> <tr> <td>SRADCKS[1:0]</td> <td>ADC 采样时钟</td> </tr> <tr> <td>00</td> <td>CPUCLK</td> </tr> <tr> <td>01</td> <td>CPUCLK/2</td> </tr> <tr> <td>10</td> <td>CPUCLK/4</td> </tr> <tr> <td>11</td> <td>CPUCLK/8</td> </tr> </table>		SRADCKS[1:0]	ADC 采样时钟	00	CPUCLK	01	CPUCLK/2	10	CPUCLK/4	11	CPUCLK/8
SRADCKS[1:0]	ADC 采样时钟												
00	CPUCLK												
01	CPUCLK/2												
10	CPUCLK/4												
11	CPUCLK/8												

表 93 SRADCON1 寄存器（地址为 61H）各位功能表

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
SRADCON1	SRADE N	SRAD S	OFTEN	CALIF	ENOV	OFFEX	VREFS[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 如果使能 8 次转换取平均值功能（AUTOS 置 1），则不能再查询该寄存器位判断转换是否完成，而需要判断 AUTOS 位。
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 <table border="1"> <tr> <td>VREFS[1:0]</td> <td>AD 参考电压</td> </tr> <tr> <td>00</td> <td>VDD</td> </tr> <tr> <td>01</td> <td>PT3.0 外部参考电源输入</td> </tr> <tr> <td>10</td> <td>内部参考电压</td> </tr> <tr> <td>11</td> <td>内部参考电压, PT3.0 可外接电容作为内置参考电压滤波使用, 以提高精度。</td> </tr> </table>	VREFS[1:0]	AD 参考电压	00	VDD	01	PT3.0 外部参考电源输入	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。
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 94 SRADCON2 寄存器 (地址为 62H) 各位功能表

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
SRADCON2	CHS[4:0]					AUTOS	REF_SEL[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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	AINP0 输入、AINN0 输入
		00001	AINP1 输入、AINN1 输入
		00010	AINP2 输入、AINN2 输入
		00011	AINP3 输入、AINN3 输入
		00100	AINP4 输入、AINN4 输入
		00101	AINP5 输入、AINN5 输入
		00110	AINP6 输入、AINN6 输入
		00111	AINP7 输入、AINN7 输入
		01000	AINP8 输入、AINN8 输入
		01001	AINP9 输入、AINN9 输入
		01010	AINP10 输入、AINN10 输入
		01011	AINP11 输入、AINN11 输入
		01100	AINP12 输入、AINN12 输入
		10000	内部 1/8VDD
		10001	内部参考电压
		10010	内部接地
10011	运算放大器的输出 C0 作为 AD 的输入 (内部需要做滤波处理)		
10100	运放放大 16 倍的输出作为 AD 输入		
	其他	无效	
2	AUTOS	ADC 自动转换使能位, 8 次转换完成后, 该位硬件清零	

		<p>0: 禁止 ADC 自动转换 1: 使能 ADC 自动转换 使能自动转换后, 将自动完成 8 次 ADC 转换, 并对 8 次转换结果取平均值, 最终结果保存在 OFTEN 选择的寄存器中。设置自动转换使能后, 只有当 8 次 AD 转换都完成时, 才会产生 ADC 中断, 而不会产生 8 次 AD 中断。通过判断 AUTOS 位可以查询 8 次转换是否完成, AUTOS 位软件置 1, 8 次转换完成后硬件清零。</p>										
1: 0	REF_SEL[1:0]	<p>内部参考电压 VREF 选择</p> <table border="1"> <tr> <th>REF_SEL[1:0]</th> <th>内部参考电压</th> </tr> <tr> <td>00</td> <td>1.4V</td> </tr> <tr> <td>01</td> <td>2.0V</td> </tr> <tr> <td>10</td> <td>3.0V</td> </tr> <tr> <td>11</td> <td>4.0V</td> </tr> </table>	REF_SEL[1:0]	内部参考电压	00	1.4V	01	2.0V	10	3.0V	11	4.0V
REF_SEL[1:0]	内部参考电压											
00	1.4V											
01	2.0V											
10	3.0V											
11	4.0V											

表 95 SRADCON3 寄存器 (地址为 63H) 各位功能表

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
SRADCON3	SRADSMOD[2:0]			SOURCE_SEL[1:0]		SRADSDLY[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能																		
7: 5	SRADSMOD[2:0]	<p>SRADS 启动位/状态位控制模式:</p> <table border="1"> <tr> <th>SRADSMOD[2:0]</th> <th>SRADS 控制模式</th> </tr> <tr> <td>3'b000</td> <td>通过直接配置 SRADS 寄存器开始 ADC 转换</td> </tr> <tr> <td>3'b001</td> <td>通过 PWM 输出控制 ADC 开始转换</td> </tr> <tr> <td>3'b010</td> <td>通过定时器溢出控制 ADC 开始转换</td> </tr> <tr> <td>3'b011</td> <td>通过比较器输出控制 ADC 开始转换</td> </tr> <tr> <td>3'b100</td> <td>通过外部中断 0 控制 ADC 开始转换</td> </tr> <tr> <td>3'b101</td> <td>通过外部中断 1 控制 ADC 开始转换</td> </tr> <tr> <td>3'b110</td> <td>通过外部中断 2 控制 ADC 开始转换</td> </tr> <tr> <td>3'b111</td> <td>保留</td> </tr> </table>	SRADSMOD[2:0]	SRADS 控制模式	3'b000	通过直接配置 SRADS 寄存器开始 ADC 转换	3'b001	通过 PWM 输出控制 ADC 开始转换	3'b010	通过定时器溢出控制 ADC 开始转换	3'b011	通过比较器输出控制 ADC 开始转换	3'b100	通过外部中断 0 控制 ADC 开始转换	3'b101	通过外部中断 1 控制 ADC 开始转换	3'b110	通过外部中断 2 控制 ADC 开始转换	3'b111	保留
SRADSMOD[2:0]	SRADS 控制模式																			
3'b000	通过直接配置 SRADS 寄存器开始 ADC 转换																			
3'b001	通过 PWM 输出控制 ADC 开始转换																			
3'b010	通过定时器溢出控制 ADC 开始转换																			
3'b011	通过比较器输出控制 ADC 开始转换																			
3'b100	通过外部中断 0 控制 ADC 开始转换																			
3'b101	通过外部中断 1 控制 ADC 开始转换																			
3'b110	通过外部中断 2 控制 ADC 开始转换																			
3'b111	保留																			
4:3	SOURCE_SEL[1:0]	<p>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)</p> <table border="1"> <tr> <th>SOURCE_SEL[1:0]</th> <th>转换触发源</th> </tr> <tr> <td>2'b00</td> <td>TIMER2/PWM2</td> </tr> <tr> <td>2'b01</td> <td>TIMER3/PWM3</td> </tr> <tr> <td>2'b10</td> <td>TIMER4/PWM4</td> </tr> <tr> <td>2'b11</td> <td>TIMER0</td> </tr> </table>	SOURCE_SEL[1:0]	转换触发源	2'b00	TIMER2/PWM2	2'b01	TIMER3/PWM3	2'b10	TIMER4/PWM4	2'b11	TIMER0								
SOURCE_SEL[1:0]	转换触发源																			
2'b00	TIMER2/PWM2																			
2'b01	TIMER3/PWM3																			
2'b10	TIMER4/PWM4																			
2'b11	TIMER0																			
2: 0	SRADSDLY[2:0]	<p>SRADS 生效延时时间(CPUCLK)。 (当 SRADSMOD[1:0] != 2'b00 有效)</p> <table border="1"> <tr> <th>SRADSDLY[2:0]</th> <th>延时时间</th> </tr> <tr> <td>3'b000</td> <td>无延时</td> </tr> <tr> <td>3'b001</td> <td>延时 1 cycle 置 SRADS 为 1'b1</td> </tr> </table>	SRADSDLY[2:0]	延时时间	3'b000	无延时	3'b001	延时 1 cycle 置 SRADS 为 1'b1												
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 96 SRADCON 4 寄存器（地址为 64H）各位功能表

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
SRADCON4	AINSEL[2:0]			IO_REF_EN[4:0]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能																								
7:5	AINSEL[2:0]	<p>SAR ADC 正负两端输入选择</p> <table border="1"> <tr> <td>AINSEL[2:0]</td> <td>SAR ADC 正负两端输入选择</td> </tr> <tr> <td>000</td> <td>采用差分输入通道 AINP 和 AINN</td> </tr> <tr> <td>001</td> <td>采用差分输入通道 AINP，且 ADC 负端接地</td> </tr> <tr> <td>010</td> <td>采用差分输入通道 AINN，且 ADC 正端接地</td> </tr> <tr> <td>011</td> <td>采用差分输入通道 AINP 和 AINN，正负两端交换</td> </tr> <tr> <td>100</td> <td>采用差分输入通道 AINP 和 AINN</td> </tr> <tr> <td>101</td> <td>采用差分输入通道 AINP，且 ADC 负端接 VREF</td> </tr> <tr> <td>110</td> <td>采用差分输入通道 AINN，且 ADC 正端接 VREF</td> </tr> <tr> <td>111</td> <td>采用差分输入通道 AINP 和 AINN，正负两端交换</td> </tr> </table> <p>当使用单端模式时，推荐使用 101 和 110 这两档，使用这两档测试结果更好。另外 AINP 和 AINN 输入中，AINP 口测试精度会更好，尽量选择 AINP 口做 ADC 输入。</p>	AINSEL[2:0]	SAR ADC 正负两端输入选择	000	采用差分输入通道 AINP 和 AINN	001	采用差分输入通道 AINP，且 ADC 负端接地	010	采用差分输入通道 AINN，且 ADC 正端接地	011	采用差分输入通道 AINP 和 AINN，正负两端交换	100	采用差分输入通道 AINP 和 AINN	101	采用差分输入通道 AINP，且 ADC 负端接 VREF	110	采用差分输入通道 AINN，且 ADC 正端接 VREF	111	采用差分输入通道 AINP 和 AINN，正负两端交换						
AINSEL[2:0]	SAR ADC 正负两端输入选择																									
000	采用差分输入通道 AINP 和 AINN																									
001	采用差分输入通道 AINP，且 ADC 负端接地																									
010	采用差分输入通道 AINN，且 ADC 正端接地																									
011	采用差分输入通道 AINP 和 AINN，正负两端交换																									
100	采用差分输入通道 AINP 和 AINN																									
101	采用差分输入通道 AINP，且 ADC 负端接 VREF																									
110	采用差分输入通道 AINN，且 ADC 正端接 VREF																									
111	采用差分输入通道 AINP 和 AINN，正负两端交换																									
4	IO_REF_EN[4]	<p>PT3.0 参考电压接口使能位 IO_REF_EN。                      1'b1: 使能 IO 口为参考电压接口                      1'b0: 不使能 IO 口为参考电压接口</p> <table border="1"> <tr> <td>IO_REF_EN[4]</td> <td>METCH[6]</td> <td>VREFS[1:0]</td> <td>PT30</td> </tr> <tr> <td>1'b0</td> <td>X</td> <td>XX</td> <td>其他 IO 口功能</td> </tr> <tr> <td rowspan="4">1'b1</td> <td rowspan="4">1</td> <td>00</td> <td>VDD 作为 ADC 参考，内部参考电压从 PT3.0 输出</td> </tr> <tr> <td>01</td> <td>内部参考电压从 PT3.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。</td> </tr> <tr> <td>10</td> <td><b>禁止</b></td> </tr> <tr> <td>11</td> <td>内部参考电压从 PT3.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。</td> </tr> <tr> <td rowspan="2">0</td> <td rowspan="2">0</td> <td>00</td> <td>VDD 作为 ADC 参考</td> </tr> <tr> <td>01</td> <td>PT3.0 外部电压作为 ADC 参考输入</td> </tr> </table>	IO_REF_EN[4]	METCH[6]	VREFS[1:0]	PT30	1'b0	X	XX	其他 IO 口功能	1'b1	1	00	VDD 作为 ADC 参考，内部参考电压从 PT3.0 输出	01	内部参考电压从 PT3.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。	10	<b>禁止</b>	11	内部参考电压从 PT3.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。	0	0	00	VDD 作为 ADC 参考	01	PT3.0 外部电压作为 ADC 参考输入
IO_REF_EN[4]	METCH[6]	VREFS[1:0]	PT30																							
1'b0	X	XX	其他 IO 口功能																							
1'b1	1	00	VDD 作为 ADC 参考，内部参考电压从 PT3.0 输出																							
		01	内部参考电压从 PT3.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。																							
		10	<b>禁止</b>																							
		11	内部参考电压从 PT3.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。																							
0	0	00	VDD 作为 ADC 参考																							
		01	PT3.0 外部电压作为 ADC 参考输入																							

				10	内部参考电压作为 ADC 参考
				11	内部参考电压从 PT3.0 输出, 再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用, 以提高精度。
3	IO_REF_EN[3]	PT3.1 参考电压接口使能位 IO_REF_EN。 1' b1: 使能 IO 口为参考电压接口 1' b0: 不使能 IO 口为参考电压接口			
		IO_REF_EN[3]	METCH[6]	VREFS[1:0]	PT31
		1'b0	X	XX	其他 IO 口功能
		1'b1	1	00	VDD 作为 ADC 参考, 内部参考电压从 PT3.1 输出
				01	内部参考电压从 PT3.1 输出, 再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用, 以提高精度。
				10	禁止
		0	0	11	内部参考电压从 PT3.1 输出, 再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用, 以提高精度。
				00	VDD 作为 ADC 参考
				01	PT3.1 外部电压作为 ADC 参考输入
				10	内部参考电压作为 ADC 参考
		11	11	11	内部参考电压从 PT3.1 输出, 再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用, 以提高精度。

2	IO_REF_EN[2]	PT3.4 参考电压接口使能位 IO_REF_EN。 1' b1: 使能 IO 口为参考电压接口 1' b0: 不使能 IO 口为参考电压接口			
		IO_REF_EN[2]	METCH[6]	VREFS[1:0]	PT34
		1'b0	X	XX	其他 IO 口功能
		1'b1	1	00	VDD 作为 ADC 参考, 内部参考电压从 PT3.4 输出
				01	内部参考电压从 PT3.4 输出, 再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用, 以提高精度。



				10	禁止
				11	内部参考电压从 PT3.4 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。
			0	00	VDD 作为 ADC 参考
				01	PT3.4 外部电压作为 ADC 参考输入
				10	内部参考电压作为 ADC 参考
				11	内部参考电压从 PT3.4 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。
		PT3.5 参考电压接口使能位 IO_REF_EN。 1' b1: 使能 IO 口为参考电压接口 1' b0: 不使能 IO 口为参考电压接口			
		IO_REF_EN[1]	METCH[6]	VREFS[1:0]	PT35
		1'b0	X	XX	其他 IO 口功能
		1'b1	1	00	VDD 作为 ADC 参考，内部参考电压从 PT3.5 输出
				01	内部参考电压从 PT3.5 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。
				10	禁止
				11	内部参考电压从 PT3.5 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。
			0	00	VDD 作为 ADC 参考
				01	PT3.5 外部电压作为 ADC 参考输入
				10	内部参考电压作为 ADC 参考
				11	内部参考电压从 PT3.5 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。
		PT5.0 参考电压接口使能位 IO_REF_EN。 1' b1: 使能 IO 口为参考电压接口 1' b0: 不使能 IO 口为参考电压接口			
0	IO_REF_EN[0]	IO_REF_EN[0]	METCH[6]	VREFS[1:0]	PT50



		1'b0	X	XX	其他 IO 口功能		
		1'b1	1	00	VDD 作为 ADC 参考，内部参考电压从 PT5.0 输出		
				01	内部参考电压从 PT5.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。		
				10	禁止		
				11	内部参考电压从 PT5.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。		
				0	0	00	VDD 作为 ADC 参考
						01	PT5.0 外部电压作为 ADC 参考输入
						10	内部参考电压作为 ADC 参考
						11	内部参考电压从 PT5.0 输出，再作为 ADC 参考输入。可外接电容作为内置参考电压滤波使用，以提高精度。

表 97 SRADL 寄存器（地址为 66H）各位功能表

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SRADL	SRAD[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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

表 98 SRADH 寄存器（地址为 67H）各位功能表

特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
SRADH					SRAD[11:8]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
3: 0	SRAD[11:8]	<p>ADC 数据的高 4 位，只可读。最高位为符号位。</p> <p>当 DMOF=0 时，AD 转换结果为有符号数，其中最高位为 0 表示负值，最高位为 1 表示正值。</p> <p>当 DMOF=1 时，AD 转换结果为 2 的补码，最高位为符号位，最高位为 1 时表示负数，取反加 1 即可得到 AD 值。</p>



表 99 SROFTL 寄存器（地址为 68H）各位功能表

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
SROFTL	SROFT[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
7: 0	SROFT[7:0]	校正值数据的低 8 位						

表 100 SROFTH 寄存器（地址为 69H）各位功能表

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SROFTH					SROFT[11:8]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符	功能						
3: 0	SROFT[11:8]	校正值数据的高 4 位						

表 101 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

3.11.3 转换时间

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

表 102 转换时间说明表<sup>(1)</sup>

CLKDIV <sup>(2)</sup>	CALIF	SRADCKS	SRADACKS	AD 转换时间 <sup>(3)</sup>
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	$1 / ( (16\text{MHz} / 2) / 8 ) \times (12 + 0 + 8) = 20\mu\text{s}$
	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	$1 / ((16\text{MHz} / 8) / 2) \times (12 + 0 + 4) = 16\mu\text{s}$
			11	$1 / ((16\text{MHz} / 8) / 4) \times (12 + 0 + 2) = 24\mu\text{s}$
		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	$1 / ((16\text{MHz} / 8) / 4) \times (12 + 0 + 2) = 24\mu\text{s}$
			00	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 0 + 16) = 112\mu\text{s}$
			01	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 0 + 8) = 80\mu\text{s}$
	11	10	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 0 + 4) = 64\mu\text{s}$	
		11	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 0 + 2) = 56\mu\text{s}$	
		00	00	$1 / ((16\text{MHz} / 8) / 1) \times (12 + 1 + 16) = 14.5\mu\text{s}$
			01	$1 / ((16\text{MHz} / 8) / 1) \times (12 + 1 + 8) = 10.5\mu\text{s}$
		01	00	$1 / ((16\text{MHz} / 8) / 2) \times (12 + 1 + 16) = 29\mu\text{s}$
			01	$1 / ((16\text{MHz} / 8) / 2) \times (12 + 1 + 8) = 21\mu\text{s}$
	10		$1 / ((16\text{MHz} / 8) / 2) \times (12 + 1 + 4) = 17\mu\text{s}$	
	10	00	$1 / ((16\text{MHz} / 8) / 4) \times (12 + 1 + 16) = 58\mu\text{s}$	
		01	$1 / ((16\text{MHz} / 8) / 4) \times (12 + 1 + 8) = 42\mu\text{s}$	
		10	$1 / ((16\text{MHz} / 8) / 4) \times (12 + 1 + 4) = 34\mu\text{s}$	
		11	$1 / ((16\text{MHz} / 8) / 4) \times (12 + 1 + 2) = 30\mu\text{s}$	
	11	00	$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 16) = 116\mu\text{s}$	
01		$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 8) = 84\mu\text{s}$		
10		$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 4) = 68\mu\text{s}$		
11		$1 / ((16\text{MHz} / 8) / 8) \times (12 + 1 + 2) = 60\mu\text{s}$		
11	0	00	00	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 0 + 16) = 28\mu\text{s}$
			01	$1 / ((16\text{MHz} / 16) / 1) \times (12 + 0 + 8) = 20\mu\text{s}$

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

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

### 3.11.4 AD 失调电压校正

不同芯片由于离散性的原因，AD 的失调电压可能有正有负。

校正失调电压的方法：

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

```

...
clrf sradcon1      ;VDD 为参考电压,often=0,calif=0;enov=0,offex=0,vrefs=00
movlw 10h
movwf sradcon2    ;chs[4:0]=00010, 选择通道 2
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 srادل
movwf adtmp1_1
movlw sradh
movwf adtmph_1
...
bsf sradcon1,2    ;offex=1
bsf sradcon1,6    ;srad=1, 开始转换
btfsc sradcon1,6 ;检测转换是否完成
goto $-1
movlw srادل
movwf adtmp1_2
movlw sradh
movwf adtmph_2
aver adtmph_1,adtmp1_1,adtmph_2,adtmp1_2 ;求两次 AD 值平均值, 并保存在
;adtmph_1,adtmp1_1
...

```

### 3.11.5 数字比较器

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

将比较的正端信号接到 ADC 的 AINP 通道，负端信号接到 ADC 的 AINN 通道，进行 ADC 转换后，得到转换结果。如果 DMOF 为 0，则转换结果的最高位取反即为符号位，可以直接通过该符号位判断正负。如果 DMOF 为 1，则转换结果的最高位即符号位。

3.12 比较器/运算放大器

3.12.1 比较器 0/运算放大器 0

CSU38M20 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。

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3eh	INTF3	CMPOIF								0000uuuu
3fh	INTE3	CMPOIE								0000uuuu
6ah	CMPCON0	CMPOEN	CMP_PSEL[2:0]			CMP_NSEL[2:0]			CMPOUT0	00000000
6bh	CMPCON1	CMP_MODE	CMP_IO_SEL	CMP_IO_EN	CMPOFST_SEL[4:0]					00010000

表 103 CMPCON0 寄存器（地址为 6AH）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0
CMPCON0	CMPOEN	CMP_PSEL[2:0]			CMP_NSEL[2:0]			CMPOUT
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能																		
7	CMPOEN	比较器/运算放大器使能位 1: 使能 0: 禁止																		
6:4	CMP_PSEL[2:0]	比较器/运算放大器正端输入端信号选择 <table border="1"> <thead> <tr> <th>CMP_PSEL[2:0]</th> <th>输入口</th> </tr> </thead> <tbody> <tr><td>3'b000</td><td>GND</td></tr> <tr><td>3'b 001</td><td>PT51</td></tr> <tr><td>3'b 010</td><td>PT52</td></tr> <tr><td>3'b 011</td><td>1/2VREF</td></tr> <tr><td>3'b 100</td><td>1/2VDD</td></tr> <tr><td>3'b 101</td><td>0.4VDD</td></tr> <tr><td>3'b 110</td><td>GND</td></tr> <tr><td>3'b 111</td><td>VDD</td></tr> </tbody> </table>	CMP_PSEL[2:0]	输入口	3'b000	GND	3'b 001	PT51	3'b 010	PT52	3'b 011	1/2VREF	3'b 100	1/2VDD	3'b 101	0.4VDD	3'b 110	GND	3'b 111	VDD
CMP_PSEL[2:0]	输入口																			
3'b000	GND																			
3'b 001	PT51																			
3'b 010	PT52																			
3'b 011	1/2VREF																			
3'b 100	1/2VDD																			
3'b 101	0.4VDD																			
3'b 110	GND																			
3'b 111	VDD																			
3:1	CMP_NSEL[2:0]	比较器/运算放大器负端输入端信号选择 <table border="1"> <thead> <tr> <th>CMP_NSEL[2:0]</th> <th>输入口</th> </tr> </thead> <tbody> <tr><td>3'b000</td><td>GND</td></tr> <tr><td>3'b 001</td><td>PT51</td></tr> <tr><td>3'b 010</td><td>PT52</td></tr> <tr><td>3'b 011</td><td>1/2VREF</td></tr> <tr><td>3'b 100</td><td>1/2VDD</td></tr> <tr><td>3'b 101</td><td>0.4VDD</td></tr> <tr><td>3'b 110</td><td>Feed Back Resister</td></tr> <tr><td>3'b 111</td><td>VDD</td></tr> </tbody> </table>	CMP_NSEL[2:0]	输入口	3'b000	GND	3'b 001	PT51	3'b 010	PT52	3'b 011	1/2VREF	3'b 100	1/2VDD	3'b 101	0.4VDD	3'b 110	Feed Back Resister	3'b 111	VDD
CMP_NSEL[2:0]	输入口																			
3'b000	GND																			
3'b 001	PT51																			
3'b 010	PT52																			
3'b 011	1/2VREF																			
3'b 100	1/2VDD																			
3'b 101	0.4VDD																			
3'b 110	Feed Back Resister																			
3'b 111	VDD																			
0	CMPOUT	比较器的比较结果，做运放时该位无效 1: C0P 电压大于 C0N																		



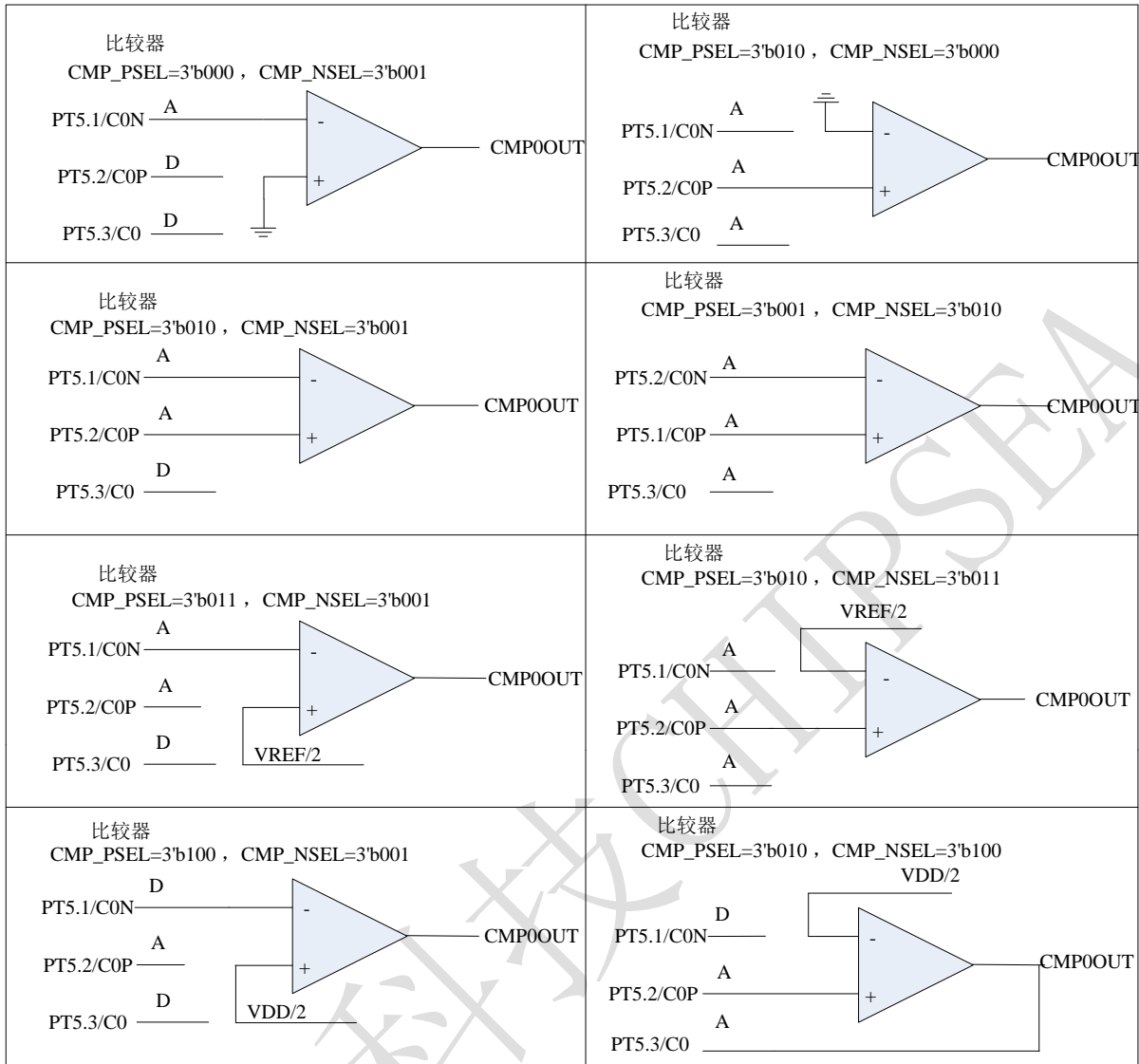
	0: COP 电压小于 CON
--	-----------------

表 104 CMPCON1 寄存器 (地址为 6BH)

特性	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
CMPCON1	CMP_MODE	CMP_IO_SEL	CMP_IO_EN	CMPOFST_SEL[4:0]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	CMP_MODE	比较器/运放模式选择位 1: 用作运放 0: 用作比较器
6	CMP_IO_SEL	比较器/运放输出端口选择 0: 从 PT53 输出 1: 从 PT37 输出
5	CMP_IO_EN	比较器结果从 IO 输出使能信号, <b>做运放时该位无效</b> 0: 比较器不从 IO 输出 1: 比较器从 IO 输出
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		

下图为各种情况下比较器及运算放大器的使用示意图



### 3.12.2 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、内部参考电压和输入的外部参考电压，通过 SRADCON1 寄存器的 VREFS[1:0]位进行选择。

### 3.12.3 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 CMPCON0 寄存器中 CMP\_PSEL[2:0]位和 CMP\_NSEL[2:0]置 3'b100，使比较器输入都接到 VDD/2。
2. 配置 CMPCON1 寄存器 CMPOFST\_SEL[4:0]值为 5'b0，查看 COMPOUT 值。
3. 若 COMPOUT 值从 0 变到 1，则记录此时 COMPOUT 值，并跳到步骤 5。
4. 若 COMPOUT 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 COMPOUT 值从 1 变到 0，则记录此时 COMPOUT 值，并跳到步骤 8。



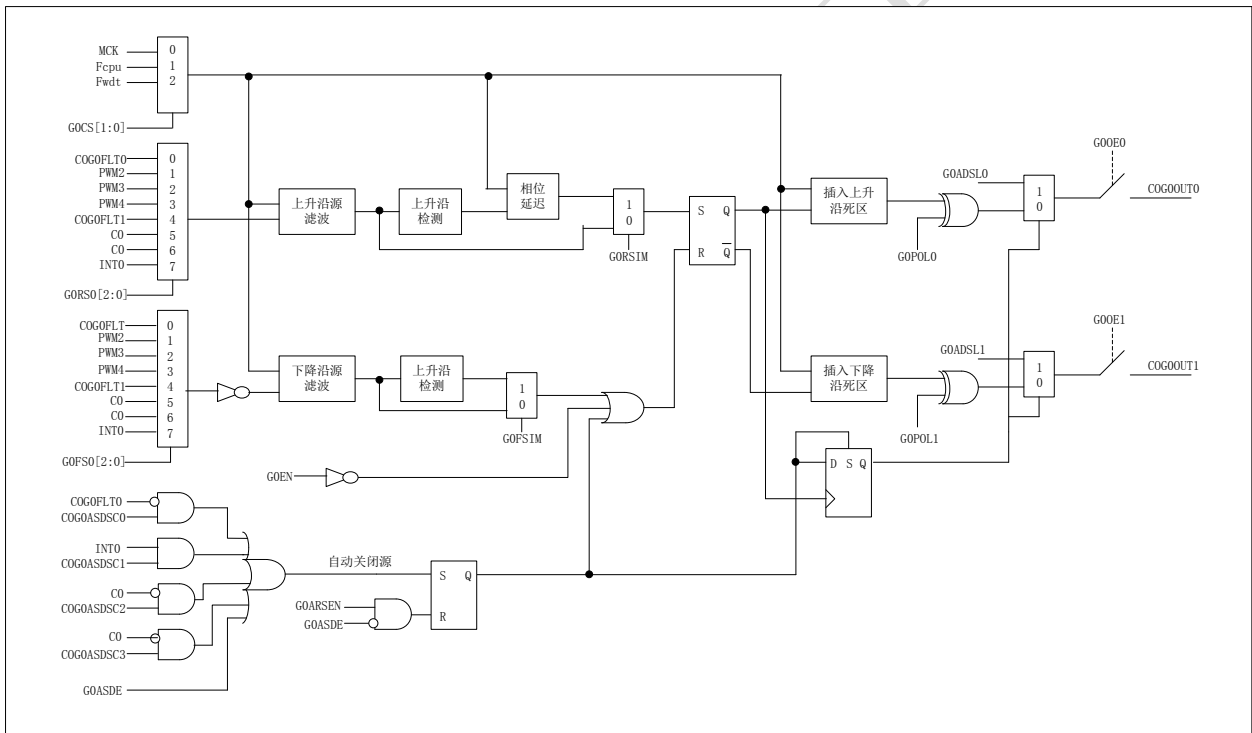
7. 若 CMPOUT 值保持为 1，则将 CMPOFST\_SEL[4:0]值-5'b1，继续配置到 CMPOFST\_SEL[4:0]，查看 COMPOUT 值。并跳转执行步骤 6。
8. 求第 3 步和第 6 步所记录的 COMOFST\_SEL[4:0]平均值。
9. 将平均值配置到 COMOFST\_SEL[4:0]寄存器，即完成了失调电压校正。

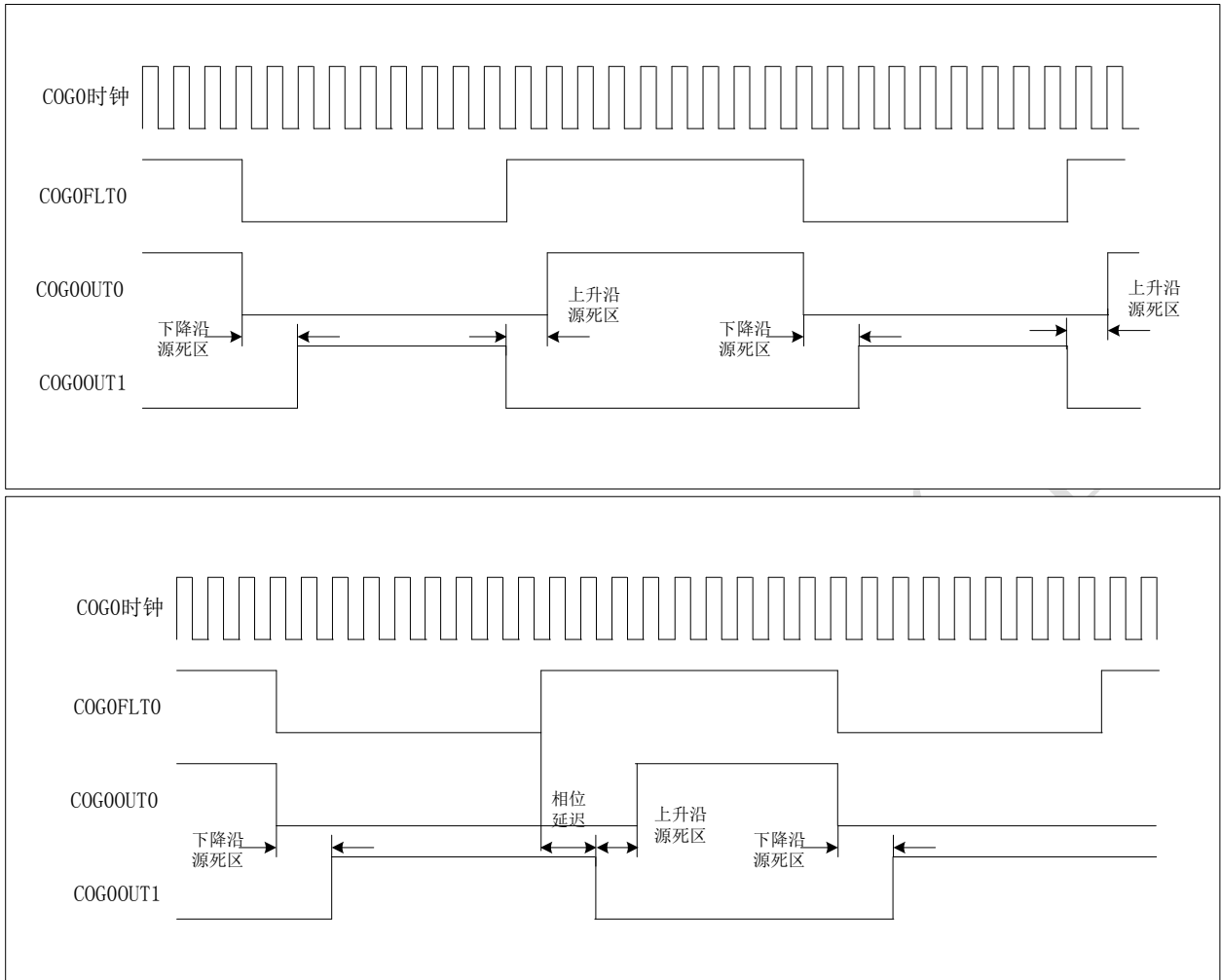
### 3.12.4 比较器中断

- 1、必须是比较器使能的情况下，比较器中断使能才能有效。
- 2、在比较器使能的情况下，比较器输出端的状态变化都会使比较器结果变化标志位置 1。如果将比较器中断使能，比较器输出端的状态变化会使芯片进入中断，同时置标志位。
- 3、比较器中断标志位硬件置位，需软件清零。

## 3.13 互补输出发生器(COG)

### 3.13.1 互补输出发生器框图





3.13.2 COG0

3.13.2.1 寄存器说明

表 105 COG0 寄存器列表(PAGE1)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3eh	INTF3				COG0IF					u0000000
3fh	INTE3				COG0IE					u0000000
71h	COG0CON0	GOEN	GOOE1	GOOE0	GOPOL1	GOPOL0		GOCS[1:0]		00000u00
72h	COG0CON1	G0FSIM	GORSIM	G0FS[2:0]			GORS[2:0]			00000000
73h	COG0ASD	GOASDE	GOARSEN	GOASDL1	GOASDL0	GOASDSC3	GOASDSC2	GOASDSC1	GOASDSC0	00000000
74h	COG0DB	G0DBF[3:0]				G0DBR[3:0]				00000000
75h	COG0BLK	G0BLKF[3:0]				G0BLKR[3:0]				00000000
76h	COG0PH					G0PH[3:0]				00000000

表 106 COG0CON0 寄存器（地址为 71H）各位功能表

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
COG0CON	GOEN	GOOE1	GOOE0	GOPOL1	GOPOL0		GOCS[1:0]	



0								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7	G0EN	COG0 使能位 1: 使能 COG0 0: 禁止 COG0										
6	G0OE1	COG0OUT1 输出使能位 1: 使能 COG0OUT1 输出 0: 禁止 COG0OUT1 输出										
5	G0OE0	COG0OUT0 输出使能位 1: 使能 COG0OUT0 输出 0: 禁止 COG0OUT0 输出										
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 <table border="1"> <tr> <td>G0CS[1:0]</td> <td>COG0 时钟源</td> </tr> <tr> <td>00</td> <td>MCK</td> </tr> <tr> <td>01</td> <td>CPUCLK</td> </tr> <tr> <td>10</td> <td>WDTCLK</td> </tr> <tr> <td>11</td> <td>MCK</td> </tr> </table>	G0CS[1:0]	COG0 时钟源	00	MCK	01	CPUCLK	10	WDTCLK	11	MCK
G0CS[1:0]	COG0 时钟源											
00	MCK											
01	CPUCLK											
10	WDTCLK											
11	MCK											

表 107 COG0CON1 寄存器（地址为 72H）各位功能表

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
COG0CON1	G0FSIM	G0RSIM	G0FS[2:0]			G0RS[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 <table border="1"> <tr> <td>G0FS[2:0]</td> <td>COG0 下降沿源</td> </tr> <tr> <td>000</td> <td>COG0FLT0</td> </tr> <tr> <td>001</td> <td>PWM2</td> </tr> <tr> <td>010</td> <td>PWM3</td> </tr> <tr> <td>011</td> <td>PWM4</td> </tr> <tr> <td>100</td> <td>COG0FLT1</td> </tr> <tr> <td>101</td> <td>C0</td> </tr> <tr> <td>110</td> <td>C0</td> </tr> </table>	G0FS[2:0]	COG0 下降沿源	000	COG0FLT0	001	PWM2	010	PWM3	011	PWM4	100	COG0FLT1	101	C0	110	C0
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 108 COG0ASD 寄存器（地址为 73H）各位功能表

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
COG0ASD	G0ASDE	G0ARSEN	G0ASDL1	G0ASDL0	G0ASDSC3	G0ASDSC2	G0ASDSC1	G0ASDSC0
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 109 COG0DB 寄存器（地址为 74H）各位功能表

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
COG0DB	G0DBF[3:0]				G0DBR[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 110 COG0BLK 寄存器（地址为 75H）各位功能表

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
COG0BLK	G0DBLK[3:0]				G0DBLK[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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 111 COG0PH 寄存器（地址为 76H）各位功能表

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
COG0PH	G0PH[3:0]				G0PH[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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

3.14 5\*21 LCD

CSU38M20 中内置一个 5\*21 LCD 驱动，1/5duty,1/2bias，但这需要配合软件使用。

3.14.1 寄存器说明

表 112 LCD 寄存器列表(PAGE1)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
7dh	METCH	METCH[7:0]								00000000
PAGE1										
30h	LCDCFG1	LCDEN	LCDWS	LCDDUTY[1:0]		LCDCKSEL[1:0]		LCDCKS[1:0]		00000000
31h	LCDCFG2	LCDREF[1:0]				LCDSCKS[3:0]			00uu0000	
32h	LCDCOM	SEG_EN[7:0]								00000000
33h	LCDCOM1	SEG_EN[15:8]								00000000
34h	LCDCOM2	SEG21_COM4_EN[1:0]		SEG20_COM5_EN[1:0]		SEG_EN[19:16]			00000000	
35h	LCDCOM3					COM_EN[3:0]			uuuu0000	
36h	LCDSEG0					SEG0[5:0]			uu000000	
37h	LCDSEG1					SEG1[5:0]			uu000000	
38h	LCDSEG2					SEG2[5:0]			uu000000	
39h	LCDSEG3					SEG3[5:0]			uu000000	
3ah	LCDSEG4					SEG4[5:0]			uu000000	
3bh	LCDSEG5					SEG5[5:0]			uu000000	
3ch	LCDSEG6					SEG6[5:0]			uu000000	
3dh	LCDSEG7					SEG7[5:0]			uu000000	
3eh	LCDSEG8					SEG8[5:0]			uu000000	
3fh	LCDSEG9					SEG9[5:0]			uu000000	
40h	LCDSEG10					SEG10[5:0]			uu000000	
41h	LCDSEG11					SEG11[5:0]			uu000000	
42h	LCDSEG12					SEG12[5:0]			uu000000	
43h	LCDSEG13					SEG13[5:0]			uu000000	
44h	LCDSEG14					SEG14[5:0]			uu000000	
45h	LCDSEG15					SEG15[5:0]			uu000000	
46h	LCDSEG16					SEG16[5:0]			uu000000	
47h	LCDSEG17					SEG17[5:0]			uu000000	
48h	LCDSEG18					SEG18[5:0]			uu000000	
49h	LCDSEG19					SEG19[5:0]			uu000000	
4Ah	LCDSEG20					SEG20[5:0]			uu000000	
4Bh	LCDSEG21					SEG21[5:0]			uu000000	

表 113 METCH 寄存器（地址为 7DH）各位功能表

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

位地址	标识符	功能
7	METCH[7]	LCD 驱动 1/2BIAS 使能位 0:禁止 1/2BIAS 1:使能 1/2BIAS 仅在 LCDEN 使能时有效，SEG_EN 或者 COM_EN 使能的 IO 口将输出

		1/2VDD 电压。
--	--	------------

表 114 LCDCFG1 寄存器（地址为 PAGE1 的 30H）各位功能表

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
LCDCFG1	LCDEN	LCDWS	LCDDUTY[1:0]		LCDCKSEL[1:0]		LCDCKS[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7	LCDEN	硬件 LCD 使能位 1: LCD 驱动使能, 时钟打开 0: LCD 驱动禁止, 时钟关闭										
6	LCDWS	LCD 驱动波形选择 0: A 波形 1: B 波形										
5: 4	LCDDUTY[1:0]	LCD Duty 选择位 <table border="1"> <tr> <th>LCDDUTY [1:0]</th> <th>LCD Duty</th> </tr> <tr> <td>00</td> <td>1/6 duty</td> </tr> <tr> <td>01</td> <td>1/5 duty</td> </tr> <tr> <td>1x</td> <td>1/4 duty</td> </tr> </table>	LCDDUTY [1:0]	LCD Duty	00	1/6 duty	01	1/5 duty	1x	1/4 duty		
LCDDUTY [1:0]	LCD Duty											
00	1/6 duty											
01	1/5 duty											
1x	1/4 duty											
3: 2	LCDCKSEL[1:0]	LCD 时钟源选择 <table border="1"> <tr> <th>LCDCKSEL[1:0]</th> <th>时钟</th> </tr> <tr> <td>00</td> <td>MCK/512</td> </tr> <tr> <td>01</td> <td>内部 32K WDT 时钟, 仅当内部 WDT 晶振打开时有效</td> </tr> <tr> <td>1X</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> </table>	LCDCKSEL[1:0]	时钟	00	MCK/512	01	内部 32K WDT 时钟, 仅当内部 WDT 晶振打开时有效	1X	外部 32768Hz 晶振时钟, 仅当外部接 32768Hz 晶振, 且晶振打 开时有效		
LCDCKSEL[1:0]	时钟											
00	MCK/512											
01	内部 32K WDT 时钟, 仅当内部 WDT 晶振打开时有效											
1X	外部 32768Hz 晶振时钟, 仅当外部接 32768Hz 晶振, 且晶振打 开时有效											
1:0	LCDCKS[1:0]	LCD 帧频选择 <table border="1"> <tr> <th>LCDCKS[1:0]</th> <th>LCD 的帧频</th> </tr> <tr> <td>11</td> <td>LCD 输入时钟频率的 1/32</td> </tr> <tr> <td>10</td> <td>LCD 输入时钟频率的 1/16</td> </tr> <tr> <td>01</td> <td>LCD 输入时钟频率的 1/8</td> </tr> <tr> <td>00</td> <td>LCD 输入时钟频率的 1/4</td> </tr> </table>	LCDCKS[1:0]	LCD 的帧频	11	LCD 输入时钟频率的 1/32	10	LCD 输入时钟频率的 1/16	01	LCD 输入时钟频率的 1/8	00	LCD 输入时钟频率的 1/4
LCDCKS[1:0]	LCD 的帧频											
11	LCD 输入时钟频率的 1/32											
10	LCD 输入时钟频率的 1/16											
01	LCD 输入时钟频率的 1/8											
00	LCD 输入时钟频率的 1/4											

表 115 LCDCFG2 寄存器（地址为 PAGE1 的 31H）各位功能表

特性	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDCFG2	LCDREF[1:0]				LCDSCK[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能										
7: 6	LCDREF[1:0]	LCD 分压电阻选择 <table border="1"> <tr> <th>LCDREF[1:0]</th> <th>LCD 分压电阻阻值</th> </tr> <tr> <td>00</td> <td>200Kohm</td> </tr> <tr> <td>01</td> <td>100Kohm</td> </tr> <tr> <td>10</td> <td>50Kohm</td> </tr> <tr> <td>11</td> <td>10Kohm</td> </tr> </table>	LCDREF[1:0]	LCD 分压电阻阻值	00	200Kohm	01	100Kohm	10	50Kohm	11	10Kohm
LCDREF[1:0]	LCD 分压电阻阻值											
00	200Kohm											
01	100Kohm											
10	50Kohm											
11	10Kohm											

3: 0	LCDSCK[3:0]	LCDSCK[3:0]	LCD 时钟分频
		0000	LCD SCK/32
		0001	LCD SCK/30
		0010	LCD SCK/28
		0011	LCD SCK/26
		0100	LCD SCK/24
		0101	LCD SCK/22
		0110	LCD SCK/20
		0111	LCD SCK/18
		1000	LCD SCK/16
		1001	LCD SCK/14
		1010	LCD SCK/12
		1011	LCD SCK/10
		1100	LCD SCK/8
		1101	LCD SCK/6
		1110	LCD SCK/4
1111	LCD SCK/2		

表 116 LCDCOM 寄存器（地址为 PAGE1 的 32H）各位功能表

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

位地址	标识符	功能
7	SEG_EN[7]	SEG7 使能控制位 1: PT5.7 做 SEG7 0: PT5.7 做普通 IO 口
6	SEG_EN[6]	SEG6 使能控制位 1: PT5.6 做 SEG6 0: PT5.6 做普通 IO 口
5	SEG_EN[5]	SEG5 使能控制位 1: PT5.5 做 SEG5 0: PT5.5 做普通 IO 口
4	SEG_EN[4]	SEG4 使能控制位 1: PT1.7 做 SEG4 0: PT1.7 做普通 IO 口
3	SEG_EN[3]	SEG3 使能控制位 1: PT1.6 做 SEG3 0: PT1.6 做普通 IO 口
2	SEG_EN[2]	SEG2 使能控制位 1: PT1.5 做 SEG2 0: PT1.5 做普通 IO 口
1	SEG_EN[1]	SEG1 使能控制位 1: PT1.4 做 SEG1 0: PT1.4 做普通 IO 口
0	SEG_EN[0]	SEG0 使能控制位 1: PT1.0 做 SEG0 0: PT1.0 做普通 IO 口



表 117 LCDCOM1 寄存器（地址为 PAGE1 的 33H）各位功能表

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
LCDCOM1	SEG_EN[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	SEG_EN[15]	SEG15 使能控制位 1: PT5.1 做 SEG15 0: PT5.1 做普通 IO 口
6	SEG_EN[14]	SEG14 使能控制位 1: PT5.2 做 SEG14 0: PT5.2 做普通 IO 口
5	SEG_EN[13]	SEG13 使能控制位 1: PT5.3 做 SEG13 0: PT5.3 做普通 IO 口
4	SEG_EN[12]	SEG12 使能控制位 1: PT3.7 做 SEG12 0: PT3.7 做普通 IO 口
3	SEG_EN[11]	SEG11 使能控制位 1: PT3.6 做 SEG11 0: PT3.6 做普通 IO 口
2	SEG_EN[10]	SEG10 使能控制位 1: PT5.4 做 SEG10 0: PT5.4 做普通 IO 口
1	SEG_EN[9]	SEG9 使能控制位 1: PT2.7 做 SEG9 0: PT2.7 做普通 IO 口
0	SEG_EN[8]	SEG8 使能控制位 1: PT2.6 做 SEG8 0: PT2.6 做普通 IO 口

表 118 LCDCOM2 寄存器（地址为 PAGE1 的 34H）各位功能表

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
LCDCOM2	SEG21_COM4_EN[1:0]	SEG20_COM5_EN[1:0]	SEG_EN[19:16]					
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能								
7:6	SEG21_COM4_EN[1:0]	SEG21 或 COM4 使能控制位 <table border="1"> <tr> <td>SEG21_COM4_EN[1:0]</td> <td>SEG21 或 COM4 使能控制位</td> </tr> <tr> <td>00</td> <td>PT3.1 做普通 IO 口</td> </tr> <tr> <td>01</td> <td>PT3.1 做 SEG21</td> </tr> <tr> <td>1x</td> <td>PT3.1 做 COM4</td> </tr> </table>	SEG21_COM4_EN[1:0]	SEG21 或 COM4 使能控制位	00	PT3.1 做普通 IO 口	01	PT3.1 做 SEG21	1x	PT3.1 做 COM4
SEG21_COM4_EN[1:0]	SEG21 或 COM4 使能控制位									
00	PT3.1 做普通 IO 口									
01	PT3.1 做 SEG21									
1x	PT3.1 做 COM4									
5:4	SEG20_COM5_EN[1:0]	SEG20 或 COM5 使能控制位 <table border="1"> <tr> <td>SEG20_COM5_EN[1:0]</td> <td>SEG20 或 COM5 使能控制位</td> </tr> </table>	SEG20_COM5_EN[1:0]	SEG20 或 COM5 使能控制位						
SEG20_COM5_EN[1:0]	SEG20 或 COM5 使能控制位									

		00	PT3.2 做普通 IO 口
		01	PT3.2 做 SEG20
		1x	PT3.2 做 COM5
3	SEG_EN[19]	SEG19 使能控制位 1: PT3.3 做 SEG19 0: PT3.3 做普通 IO 口	
2	SEG_EN[18]	SEG18 使能控制位 1: PT3.4 做 SEG18 0: PT3.4 做普通 IO 口	
1	SEG_EN[17]	SEG17 使能控制位 1: PT3.5 做 SEG17 0: PT3.5 做普通 IO 口	
0	SEG_EN[16]	SEG16 使能控制位 1: PT5.0 做 SEG16 0: PT5.0 做普通 IO 口	

表 119 LCDCOM3 寄存器（地址为 PAGE1 的 35H）各位功能表

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
LDCOM3					COM_EN[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
3	COM_EN[3]	COM3 使能控制位 1: PT3.0 做 COM3 0: PT3.0 做普通 IO 口
2	COM_EN[2]	COM2 使能控制位 1: PT1.3 做 COM2 0: PT1.3 做普通 IO 口
1	COM_EN[1]	COM1 使能控制位 1: PT1.2 做 COM1 0: PT1.2 做普通 IO 口
0	COM_EN[0]	COM0 使能控制位 1: PT1.1 做 COM0 0: PT1.1 做普通 IO 口

表 120 LCDSEGn (n=0~21) 寄存器（地址为 PAGE1 的 36H~4BH）各位功能表

特性	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDSEGn			SEGn [5:0]					
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7: 6	RESERVE	保留
5: 0	SEGn	SEGn (n=0~21) 口数据寄存器

LCD 显示存储映射



	7	6	5	4	3	2	1	0
			COM5	COM4	COM3	COM2	COM1	COM0
0			SEG0[5:0]					
1			SEG1[5:0]					
2			SEG2[5:0]					
3			SEG3[5:0]					
4			SEG4[5:0]					
5			SEG5[5:0]					
.			.					
.			.					
.			.					
			SEG18[5:0]					
19			SEG19[5:0]					
20			SEG20[5:0]					
21			SEG21[5:0]					

3.14.2 硬件 LCD 驱动

3.14.2.1 LCD 控制模式

LCD 驱动器有 3 种控制模式：1/4duty，1/5duty 和 1/6duty，设置寄存器 LCDDUTY[1:0]选择一种模式。

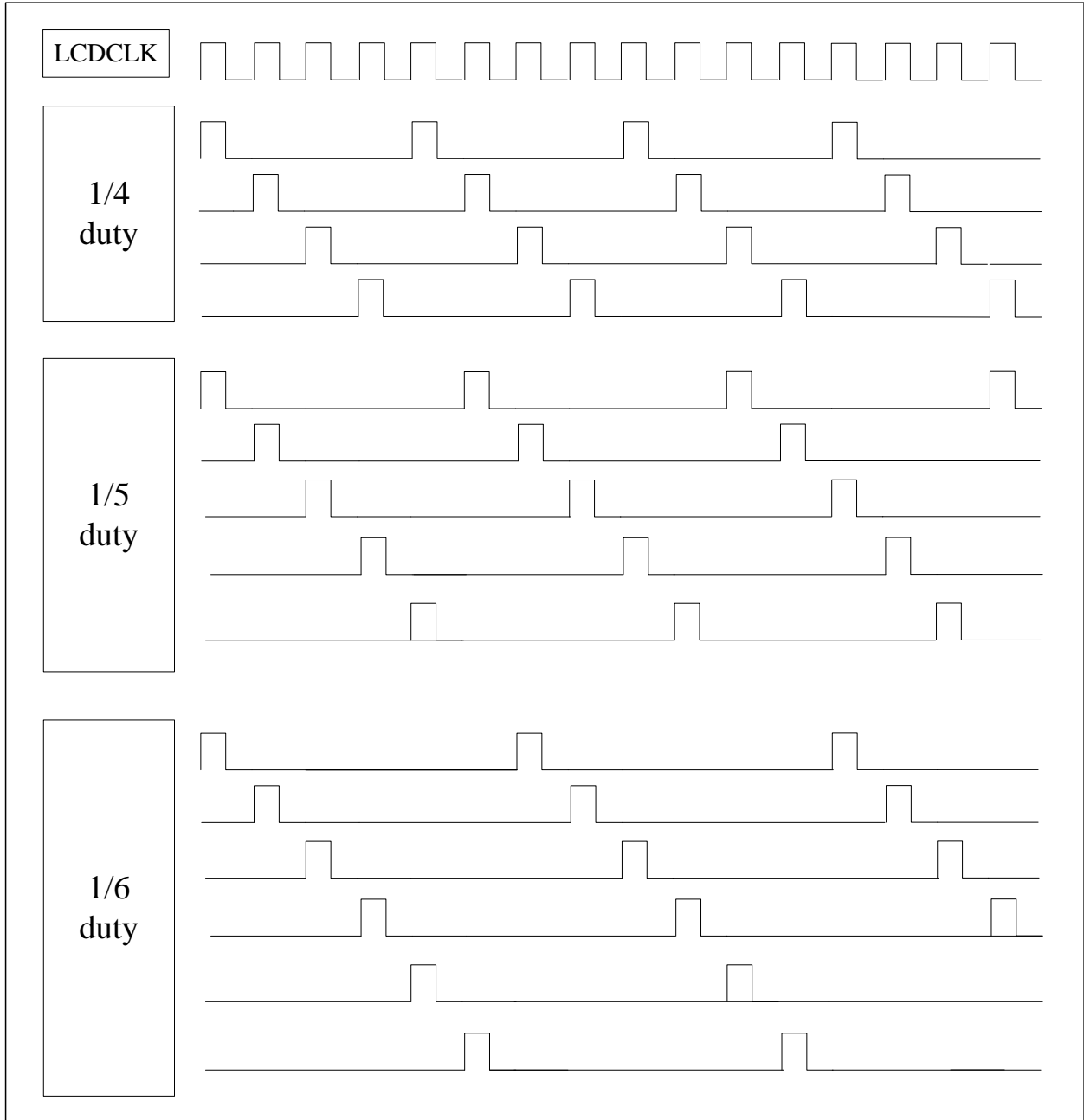


图47 LCD 的 duty 模式工作周期

3.14.2.2 LCD 帧频选择

通过设置寄存器标志 LCDSCK [3:0]对 LCD 时钟进行第一次分频。对 LCD 模块的输入时钟进行分频以获得 LCDCK。

表 121 LCDCLK 选择列表

LCDSCKS				LCDCLK(KHz)	
0	0	0	0	LCDCLK/32	1
0	0	0	1	LCDCLK/30	1.067
0	0	1	0	LCDCLK/28	1.143
0	0	1	1	LCDCLK/26	1.231
0	1	0	0	LCDCLK/24	1.333
0	1	0	1	LCDCLK/22	1.455
0	1	1	0	LCDCLK/20	1.6
0	1	1	1	LCDCLK/18	1.778
1	0	0	0	LCDCLK/16	2
1	0	0	1	LCDCLK/14	2.286
1	0	1	0	LCDCLK/12	2.667
1	0	1	1	LCDCLK/10	3.2
1	1	0	0	LCDCLK/8	4
1	1	0	1	LCDCLK/6	5.333
1	1	1	0	LCDCLK/4	8
1	1	1	1	LCDCLK/2	16

然后设置寄存器 LCDCKS[1:0]的值，LCD 模块的时钟再次进行分频，得到最终的 LCD 时钟。

表 122 LCD 帧频选择列表

LCDCKS[ 1:0]	LCD 帧频率 (LCDCK)
00	LCD 输入时钟频率/4(LCDCLK/4)
01	LCD 输入时钟频率/8(LCDCLK/8)
10	LCD 输入时钟频率/16(LCDCLK/16)
11	LCD 输入时钟频率/32(LCDCLK/32)

### 3.14.2.3 LCD 偏置电压

LCD 驱动器有 3 个偏置电压，V1、V2、V3，有 1 种电源模式：1/3bias。偏置电压的产生电路采用内部电阻分压，优点是可以节省 V2 与 V1 pin 的外部电容，缺点是在打开 LCD 时分压电路上存在静态功耗，静态功耗的大小与分压电阻的阻值有关，驱动能力较弱，分压电阻越大驱动能力越弱。

● 1/3bias 电源系统

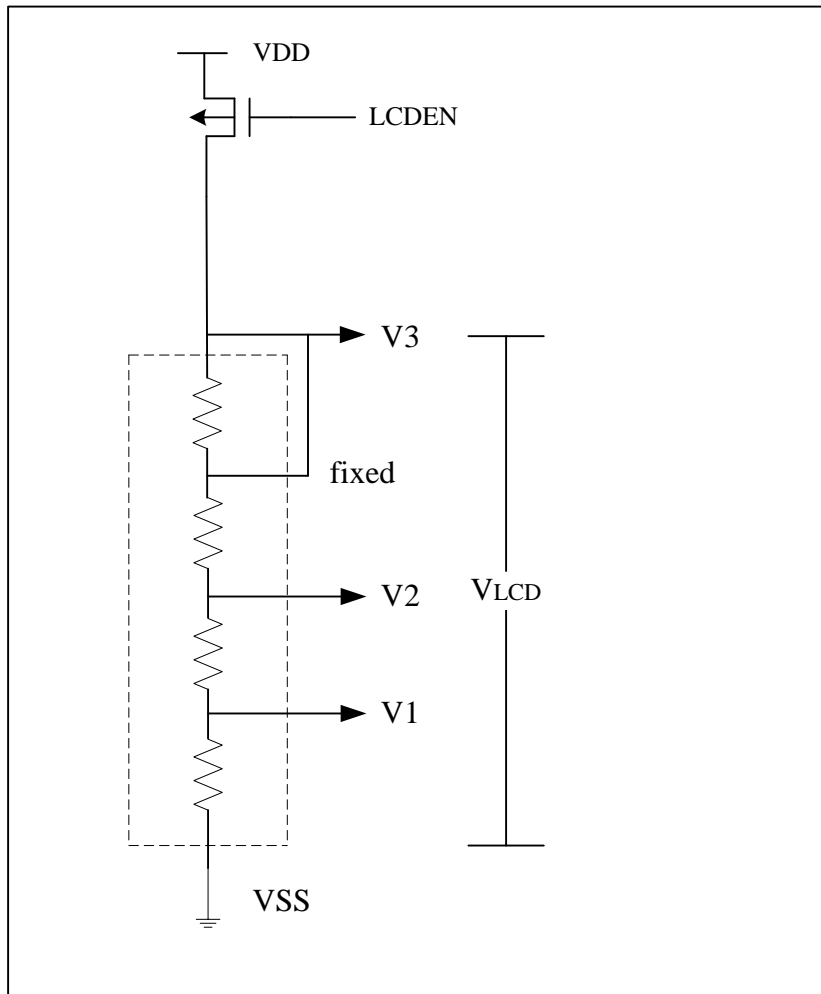


图48 LCD 的 1/3bias 电源系统电路连接图（电阻分压）

3.14.2.4 LCD 驱动波形

LCD 驱动波形如下  
1/4duty 1/3bias A 波形

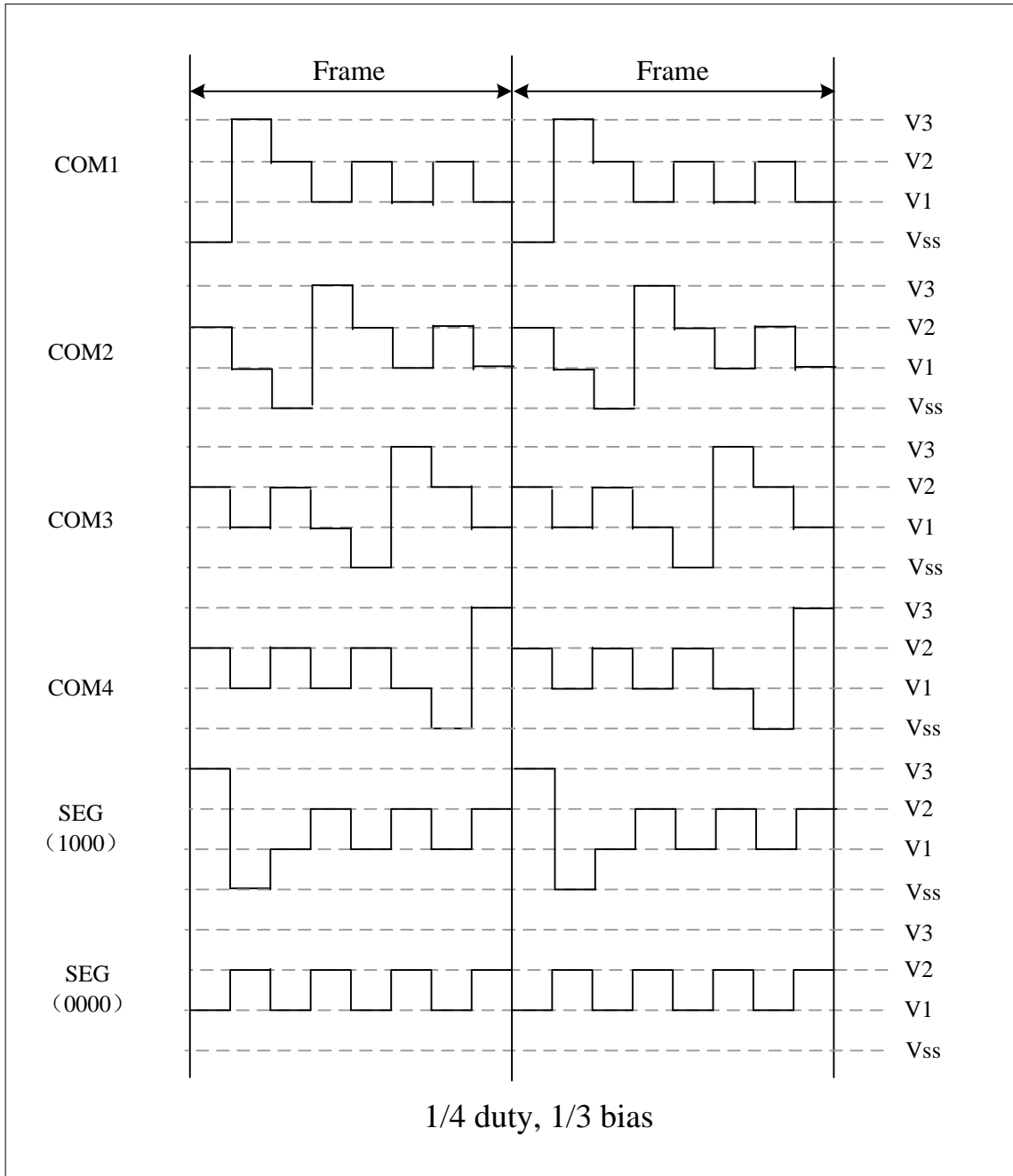


图 3-1 LCD 的 1/4duty 1/3bias 电源系统的时钟 (A 波形)

1/4duty 1/3bias B 波形

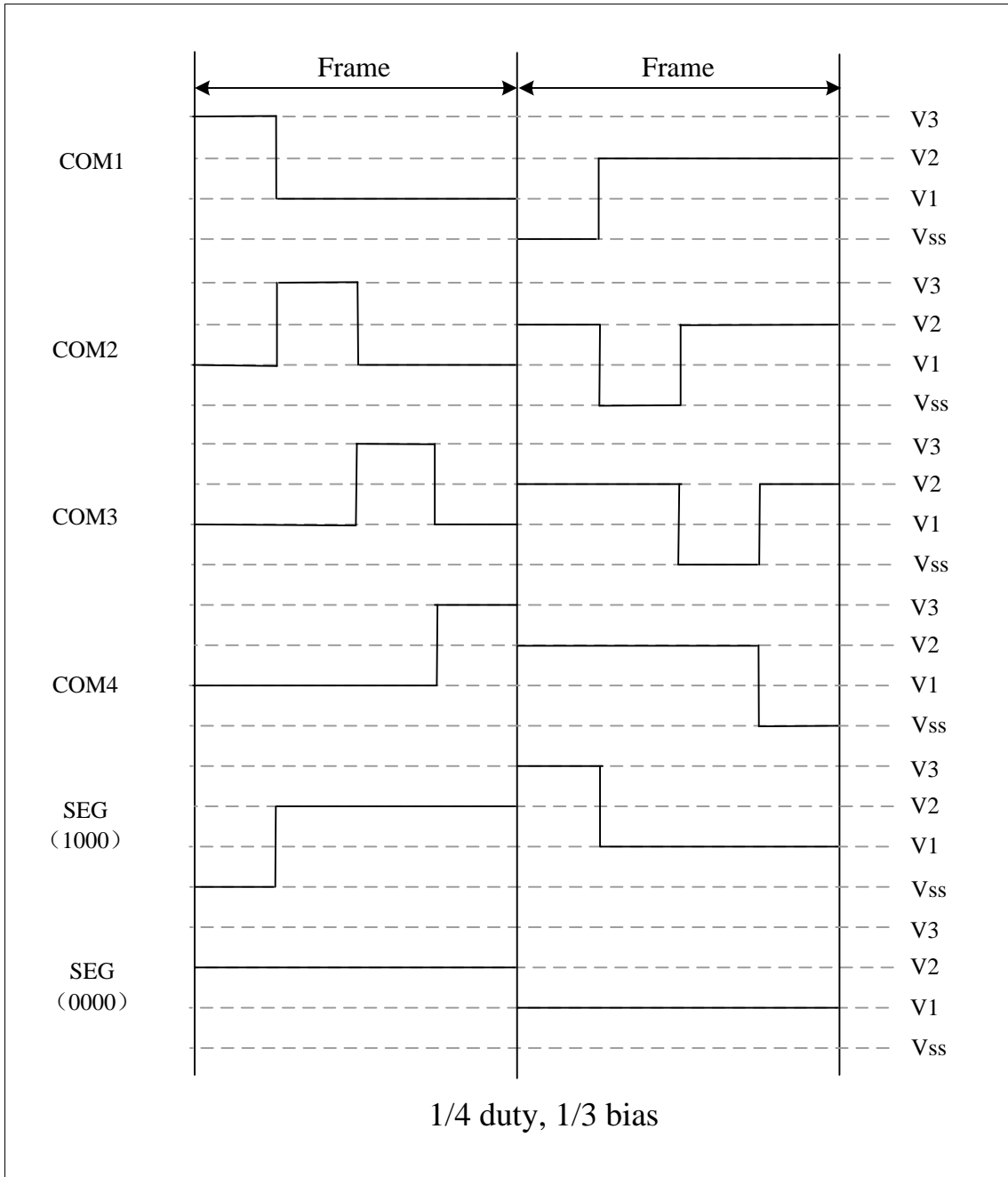


图49 LCD 的 1/4duty 1/3bias 驱动波形 (B 波形)

### 3.14.3 软件 LCD 实现方法

通过软件方式实现 1/2 BIAS LCD 驱动的方法如下：

当需要输出 1/2VDD 时，将 LCDEN 置 1，METCH[7]置 1，将需要输出 1/2VDD 的 IO 口的 COM\_EN 或 SEG\_EN 置 1，对应的 IO 口输出就是 1/2VDD。

而 0V 或 VDD 可以直接通过 IO 口功能实现，因此，只需要将对应 IO 口配置为普通输出口即可实现。



### 3.15 乘法器

CSU38M20 提供一个 8 位无符号乘法器，可以在 9 指令周期内完成乘法运算。

#### 3.15.1 寄存器说明

表 123 MUL 寄存器列表(PAGE1)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3ah	MULCTL	MULBG								0uuuuuuu
3bh	MULDA0L	MUL_DATA0[7:0]								00000000
3ch	MULDA1L	MUL_DATA1[7:0]								00000000

表 124 MULCTL 寄存器（地址为 PAGE1 的 3AH）各位功能表

特性	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
MULCTL	MULBG							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	MULBG	乘法运算开始控制位 0: 未开始乘法运算 1: 开始乘法运算，运算完成时清 0 乘法完成后，结果将保存到数据寄存器 MUL_DATA1 和 MUL_DATA0 中

表 125 MULDA0L 寄存器（地址为 PAGE1 的 3BH）各位功能表

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
MULDA0L	MUL_DATA0[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7: 0	MUL_DATA0[7:0]	被乘数低 8 位或运算结果 bit7 到 bit0

表 126 MULDA1L 寄存器（地址为 PAGE1 的 3CH）各位功能表

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
MULDA1L	MUL_DATA1[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7: 0	MUL_DATA1[7:0]	乘数低 8 位运算或运算结果 bit15 到 bit8

### 3.16 CRC16

CRC16 基于生成多项式  $G(X) = X^{16} + X^{12} + X^5 + 1$  (CRC-CCITT)

每次支持 8bit 校验，CRC 初值可以通过 CRCINITL、CRCINITH 进行配置，默认为 0xFFFF，用户可以编程修改为任意值，每次 CRC 校验后结果自动保存在 CRCINITL、CRCINITH 中，覆盖原有初值，做为下一次计算的初值。对 CRCDL 进行写操作一个时钟周期后，CRC 校验即可完成。

3.16.1 寄存器说明

表 127 MUL 寄存器列表(PAGE1)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
1ah	CRCDL	CRC_DATA[7:0]								00000000
1bh	CRCINITL	CRC_INIT[7:0]								11111111
1ch	CRCINITH	CRC_INIT[15:8]								11111111

表 128 CRCDL 寄存器（地址为 PAGE1 的 1AH）各位功能表

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	R/W-0
CRCDL	CRC_DATA[7:0]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

位地址	标识符	功能
7: 0	CRC_DATA[7:0]	CRC 数据输入寄存器，对该寄存器进行写操后即开始进行 CRC16 计算，初始值保存在 CRCINITL、CRCINITH 中，计算完成后结果也保存在 CRCINITL、CRCINITH 中。当对 16 位数据进行计算时，先计算低 8 位，再计算高 8 位

表 129 CRCINITL 寄存器（地址为 PAGE1 的 1BH）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CRCINITL	CRC_INIT[7:0]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

位地址	标识符	功能
7: 0	CRC_INIT[7:0]	CRC16 初值低 8 位或 CRC16 计算结果低 8 位

表 130 CRCINITH 寄存器（地址为 PAGE1 的 1CH）各位功能表

特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CRCINITH	CRC_INIT[15:8]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

位地址	标识符	功能
7: 0	CRC_INIT[15:8]	CRC16 初值高 8 位或 CRC16 计算结果高 8 位

3.17 ISP

通过 MOV<sub>P</sub> 和 TBLP 指令可以实现对于用户程序存储器和 EEPROM 内的数据读取和写入，用户程序存储器的地址范围为 000H~1FFFH。EEPROM 地址范围为 000H~0FFH，对 EEPROM 的操作以字节（8bit）为单位，EEPROM 大小为 256×8bit，对 EEPROM 进行读写时，高 8 位数据无效，仅低 8 位数据有效。对 EEPROM 进行写操作，必须保证 VDD 电压大于 2.5V，否则可能导致写失败。对 MTP 进行写操作时，必须保证 VDD 在 4.5V 到 5.5V 之间，否则可能导致写失败。使用 WDT 时钟或者外部 32768HZ 晶振时钟做系统时钟，不支持 TBLP。TBLP 为写保护操作，必须解锁写保护才能进行 TBLP 操作。解锁写保护需对 WRPRT 寄存器连续写入 C9H, 3AH, D3H，解锁后 16 个指令周期内都能进行 TBLP 操作，16 个指令周期后解锁自动失效，不能再进行 TBLP 操作，TBLP 一旦开始执行，即使解锁自动失效，TBLP 操作也不会中断。**执行 TBLP 指令时建议关闭全局中断使能。**

3.17.1 寄存器说明

表 131 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[12:8]					uuu00000
0Bh	EADRL	PAR[7:0]								00000000
0Ch	EDATH	EDATH[7:0]								00000000
09h	ISPCON	CHKRSLT							ISPCFG	uuuuuuu0
4fh	ISPCON2	TBLP_CLK_DIV[7:0]								00000000
6dh	WRPRT	REGWRPROT[7:0]								00000000
PAGE1										
1dh	ISPCRC	CRC16EN				CRC_LEN[3:0]				0uuu0000
1eh	ISPCRCDL	ISP_CRC_DATA[7:0]								11111111
1fh	ISPCRCDH	ISP_CRC_DATA[15:8]								11111111

表 132 EADRH 寄存器（地址为 0AH）各位功能表

特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EADRH				PAR[12:8]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7:5	RESERVE	保留
4:0	PAR[12:8]	MTP 地址高位

表 133 EADRL 寄存器（地址为 0BH）各位功能表

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

位地址	标识符	功能
7:0	PAR[7:0]	MTP 或 EEPROM 地址低 8 位

表 134 EDATH 寄存器（地址为 0CH）各位功能表



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

位地址	标识符	功能
7:0	EDATH[7:0]	读或写操作数据高 8 位，低位数据存储在 WORK 寄存器中。 进行读操作时，执行完 MOVP 后，可以从该寄存器读取数据高 8 位，从 WORK 寄存器读取数据低 8 位。 进行写操作时，在执行 TBLP 前，需将数据高 8 位写入该寄存器，数据低 8 位写入 WORK 寄存器。 因为 EEPROM 数据位宽为 8bit，因此对 EEPROM 进行读写操作数据均存储在 WORK 寄存器中。

表 135 ISPCON 寄存器（地址为 09H）各位功能表

特性	R-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
ISPCON	CHKRSLT							ISPCFG
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	CHKRSLT	TBLP 后读取写入值进行校验的结果，只读 0: 校验失败，读取数据与写入数据不一致 1: 校验成功，读取数据与写入数据一致 每次进行 TBLP 后会自动对写入地址进行一次读取，校验读取数据与写入数据是否一致，一致则该位置 1，否则清零。
6:1	RESERVE	保留
0	ISPCFG	数据读写操作配置位 0: 对 MTP 存储器进行操作 (8K×16bit) 1: 对 EEPROM 进行操作 (256×8bit)

表 136 ISPCON2 寄存器（地址为 4FH）各位功能表

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
ISPCON2	TBLP_CLK_DIV[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7:0	TBLP_CLK_DIV[7:0]	TBLP 操作时钟分频寄存器，对于不同晶振频率及指令周期，需要根据下面的配置调整该寄存器值。16MHz 内部晶振，4MHz 指令周期对应配置为 19h。 执行 TBLP 指令时，需要生成一个时钟用于进行 MTP 及 EEPROM 的写操作，该时钟周期典型值为 12.5us，要求在 10us~15us 之间。 时钟周期计算公式为 $TBLP\_CLK\_DIV[7:0] \times 2 \times Tcpuclk$ 若不配置该寄存器或时钟配置误差太大，可能导致写操作失败。

表 137 WRPRT 寄存器（地址为 6DH）各位功能表



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

位地址	标识符	功能
7:0	REGWRPROT[7:0]	写操作解锁寄存器，对该寄存器连续写入 C9H, 3AH, D3H, 即可解锁写保护。对部分关键寄存器进行写操作或者执行 TBLP 指令时，必须解锁写保护。

表 138 ISPCRC 寄存器（地址为 PAGE1 的 1DH）各位功能表

特性	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ISPCRC	CRC16EN				CRC_LEN[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7	CRC16EN	MTP 程序区或 EERPOM 区数据 CRC16 校验使能位（写保护） 0: 禁止 CRC 校验 1: 使能 CRC 校验 使能 CRC 校验后，执行 MOVP 指令启动 CRC 校验，CRC 校验时间随配置校验数据长度变化，可能会耗时几百微秒到几毫米
3:0	CRC_LEN[3:0]	CRC 校验长度选择，对 MTP 主程序区，校验长度单位为 512*16，对 EEPROM，校验长度单位为 128*8，通过 ISPCFG 进行选择 对 MTP 主程序区，校验长度计算公式 $(CRC\_LEN+1) \times 512$ MTP 主程序区为 8K，CRC_LEN 最大配置为 1111b 对 EEPROM 区，校验长度为 $(CRC\_LEN+1) \times 128$ EEPROM 区最大长度为 256 bytes，因此 CRC_LEN[3:0] 只有最低位有效，高位无效 CRC 校验起始地址由 EADRH、EADRL 配置，对 MTP，起始地址加校验长度不能超过 8K，对 EERPOM，起始地址加校验长度不能超过 256

表 139 ISPCRCDL 寄存器（地址为 PAGE1 的 1EH）各位功能表

特性	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1
ISPCRCDL	ISP_CRC_DATA[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能
7:0	ISP_CRC_DATA[7:0]	CRC 校验结果低 8 位（只读）

表 140 ISPCRCDH 寄存器（地址为 PAGE1 的 1FH）各位功能表

特性	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1
ISPCRCDH	ISP_CRC_DATA[15:8]							



	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位地址	标识符		功能					
15:8	ISP_CRC_DATA[15:8]		CRC 校验结果高 8 位（只读）					

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。

对 MTP 的读或写操作都是基于一个字（16 bits）的。  
对 EEPROM 的读或写操作都是基于一个字节（8 bits）的。

执行读操作时，在地址寄存器输入相应的值，之后执行 MOVP 指令，便可在相应的 MTP 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
...
    
```

要对 MTP 和 EEPROM 进行写操作，必须解锁写保护，依次对 REGWRPROT [7:0]寄存器写入 C9H, 3AH, D3H 即可解除写保护。解除写保护后，必须在 16 个指令周期内进行操作，16 个指令周期后写保护解锁自动失效。

### 3.17.2 MTP 及 EEPROM 写操作

当需要对 MTP 或 EEPROM 进行写操作时，代码选项中对应区间的写加密不能使能，如果使能写加密，则写操作无法完成，每次进行写操作后都会出现校验不过的现象。MTP 每 1K 空间对应一个写加密位，推荐用户只对程序区进行加密，而需要进行数据更新的地址空间不进行写加密。对 MTP 及 EEPROM 的写操作步骤如下：

- 如果是对 MTP 操作，将 ISPCON 寄存器的 ISPCFG 位清零，如果是对 EEPROM 进行操作，则将 ISPCON 寄存器的 ISPCFG 位置 1。
- 配置 {EADRH, EADRL} 写操作地址。当对 EEPROM 进行操作时，PAR[12:8]需置 0。
- 根据当前指令周期配置 ISPCON2，使得 MTP 时钟尽量接近 12.5us 的周期。
- 建议关闭全局中断使能 GIE
- 向 WRPRT 寄存器写入 C9H, 3AH, D3H，解锁写保护。
- 配置 {EDATH, WORK} 写操作数据。
- 执行指令 TBLP XH 操作（必须在对 WRPRT 寄存器写入 C9H,3AH,D3H 后 16 个指令周期内）。
- 写操作完成后，检查 ISPCON 寄存器的 CHKRSLT 位，当该位为 0 时，表示写操作校验失败，读出数据与写入数据不一致。当该位为 1 时，表示写操作成功。

### 3.17.3 MTP 及 EEPROM 读操作



当需要对 MTP 或 EEPROM 进行读操作时，代码选项中对应区间的读加密不能使能，如果使能读加密，则读操作无法完成，每次进行读操作后得到的数据都为全 0。MTP 每 1K 空间对应一个读加密位，推荐用户只对程序区进行加密，而需要进行数据读取的地址空间不进行读加密。对 MTP 及 EEPROM 的读操作步骤如下：

- 如果是对 MTP 操作，将 ISPCON 寄存器的 ISPCFG 位清零，如果是对 EEPROM 进行操作，则将 ISPCON 寄存器的 ISPCFG 位置 1。
- 配置 {EADRH, EADRL} 写操作地址。当对 EEPROM 进行操作时，PAR[12:8] 需置 0。
- 执行指令 MOVP 操作
- 读取完成

#### 3.17.4 MTP 或 EEPROM 区 CRC 校验

CSU38M20 提供对程序区或 EEPROM 区数据的 CRC16 校验。CRC 校验起始地址均为 000h，主程序区最小校验长度为 512\*16，EEPROM 区最小校验长度为 128\*8，通过 ISPCON 寄存器的 ISPCFG 位选择对 MTP 区（主程序区）或 EEPROM 区进行 CRC 校验操作。无论主程序区或 EEPROM 区是否加密，CRC 校验都可以进行。CRC 校验步骤如下：

- 通过 ISPCON 寄存器的 ISPCFG 位配置对主程序区或 EEPROM 区进行操作。
- 通过 ISPCRC 寄存器的 CRC\_LEN[3:0] 配置 CRC 校验长度，其中 EEPROM 区长度不能超过 256\*8。
- 将 ISPCRC 寄存器的 CRC16EN 置 1，使能 CRC 校验。
- 执行 MOVP 指令，启动 CRC 校验。
- 校验完成后，读取 ISPCRCDL、ISPCRCDH 获取 16 位 CRC 校验结果。

CRC16 校验算法请参考章节 [CRC16](#)



3.18 烧录模块

烧写器的接口:

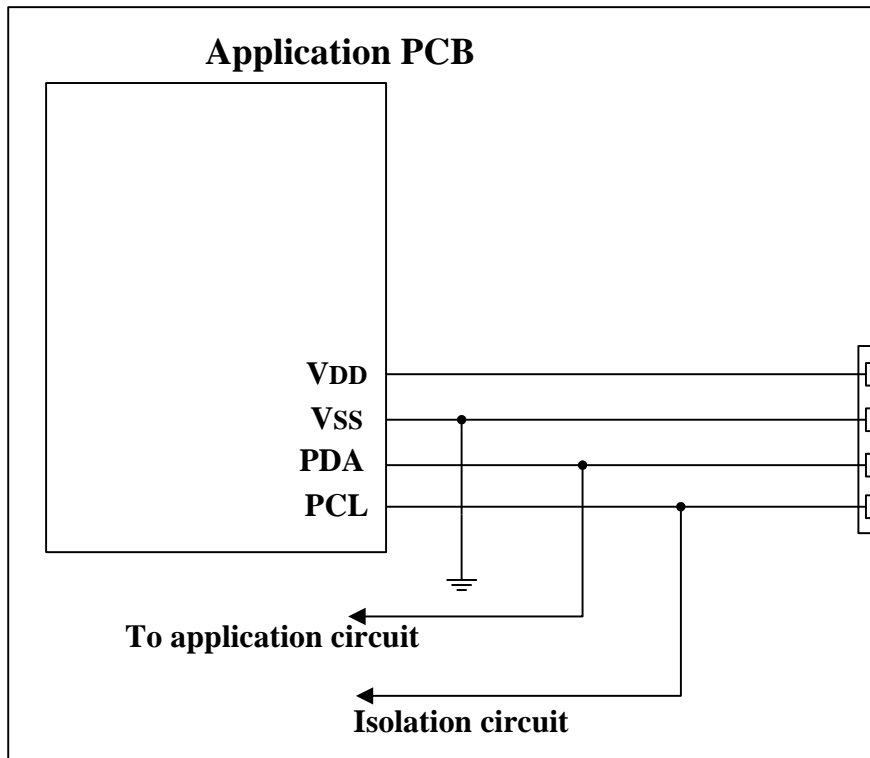


图50 烧写器接口图

表 141 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	PT1[4]端口, 数据信号
PCL	输入	PT1[0]端口, 时钟信号



3.19 代码选项

1) OPTION

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
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: 指令周期可通过程序配置 <b>通过程序配置指令周期时，使用 MCK 寄存器的 CPUCLK_SEL[1:0] 位进行配置。</b>
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。
CLKDIV[1:0]	指令周期选择
	指令周期
	指令周期=4 个时钟周期
	指令周期=8 个时钟周期
	指令周期=16 个时钟周期
	指令周期=32 个时钟周期
<b>选择 4MHz 指令周期时，必须开启 2.4V 或以上低电压复位</b>	
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 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
LVR_DLY	低电压复位延时配置 0: 低电压复位不进行延时 1: 低电压复位延时 1ms（在延时期间芯片处于复位状态）
LVR_OP	LVD 配置选项 0: LVD 只可通过代码选项配置 1: LVD 可通过程序配置 <b>通过程序配置 LVD 电压时，使用 LVRCON 寄存器的 LVD_SEL[2:0]位进行选择。LDV_SEL[2:0]默认值为 100，即默认为 3.6V 低电压</b>



	复位，所以如果系统工作电压小于 3.6V 的，不能选择 LVD 通过程序配置，否则会导致芯片一直处于复位状态。							
LVD_SEL[2: 0]	LVD 配置 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>功能</td></tr> <tr><td>VDD 低于 2.0V 复位</td></tr> <tr><td>2.4V 低电压检测</td></tr> <tr><td>2.5V 低电压检测</td></tr> <tr><td>3.0V 低电压检测</td></tr> <tr><td>3.6V 低电压检测</td></tr> <tr><td>PT10 口输入电压检测，</td></tr> </table>	功能	VDD 低于 2.0V 复位	2.4V 低电压检测	2.5V 低电压检测	3.0V 低电压检测	3.6V 低电压检测	PT10 口输入电压检测，
功能								
VDD 低于 2.0V 复位								
2.4V 低电压检测								
2.5V 低电压检测								
3.0V 低电压检测								
3.6V 低电压检测								
PT10 口输入电压检测，								
SECURITY	代码加密位 0: 代码加密 1: 代码不加密							

2) MTP 及 EEPROM 进行 MOVP 及 TBLP 加密位

标识符	功能
USR_R_SEC[8:0]	EEPROM 及 MTP 进行 MOVP 加密位，加密后不能进行读取
	第 1K*16bit 空间 (0000H~03FFH) 加密位，1 表示加密，0 表示不加密
	第 2K*16bit 空间 (0400H~07FFH) 加密位，1 表示加密，0 表示不加密
	第 3K*16bit 空间 (0800H~0BFFH) 加密位，1 表示加密，0 表示不加密
	第 4K*16bit 空间 (0C00H~0FFFH) 加密位，1 表示加密，0 表示不加密
	第 5K*16bit 空间 (1000H~13FFH) 加密位，1 表示加密，0 表示不加密
	第 6K*16bit 空间 (1400H~17FFH) 加密位，1 表示加密，0 表示不加密
	第 7K*16bit 空间 (1800H~1BFFH) 加密位，1 表示加密，0 表示不加密
	第 8K*16bit 空间 (1C00H~1FFFH) 加密位，1 表示加密，0 表示不加密
EEPROM 空间读加密位，1 表示加密，0 表示不加密	

标识符	功能
USR_W_SEC[8:0]	EEPROM 及 MTP 进行 TBLP 加密位，加密后不能进行写操作
	第 1K*16bit 空间 (0000H~03FFH) 加密位，1 表示加密，0 表示不加密
	第 2K*16bit 空间 (0400H~07FFH) 加密位，1 表示加密，0 表示不加密
	第 4K*16bit 空间 (0C00H~0FFFH) 加密位，1 表示加密，0 表示不加密



	第 5K*16bit 空间 (1000H~13FFH) 加密位, 1 表示加密, 0 表示不加密	
	第 6K*16bit 空间 (1400H~17FFH) 加密位, 1 表示加密, 0 表示不加密	
	第 7K*16bit 空间 (1800H~1BFFH) 加密位, 1 表示加密, 0 表示不加密	
	第 8K*16bit 空间 (1C00H~1FFFH) 加密位, 1 表示加密, 0 表示不加密	
	EEPROM 空间读加密位, 1 表示加密, 0 表示不加密	

## 4 MCU 指令集

表 142 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 eeprom	3	~
MOVWF f	$[f] \leftarrow [W]$	1	~
NOP	No operation	1	~
POP	Pop Work,BSR and Status	2	~
PUSH	Push Work,BSR and Status	2	~
RETFIE	Pop PC and GIE = 1	2	~
RETLW k	RETURN and $W = k$	2	~
RETURN	POP PC	2	~
RLF f,d	$[Destination<n+1>] \leftarrow [f<n>]$	1	C,Z
RRF f,d	$[Destination<n-1>] \leftarrow [f<n>]$	1	C,Z
SLEEP	STOP OSC	1	PD
SUBLW k	$[W] \leftarrow k - [W]$	1	C,DC,Z
SUBWF f,d	$[Destinnation] \leftarrow [f] - [W]$	1	C,DC,Z
SUBWFC f,d	$[Destinnation] \leftarrow [f] - [W] - 1 + C$	1	C,DC,Z
SWAPF f,d	swap f	1	~
TBLP k	Write memory	-	~
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~1FFh)

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 143 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 $(W) < 7FH$ $(PC) \leftarrow (PC) + 1 + (W) - 100H$ 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<=1FFH d=0,1
操作	[目标地址]<←(f)+(W)
标志位	C, CD, Z
描述	将 f 的内容和工作寄存器的内容加到一起。 如果 d 是 0, 结果保存到工作寄存器中。 如果 d 是 1, 结果保存到 f 中。
周期	1
例子 1 ADDWF f 0	指令执行之前: f=C2H W=17H 在指令执行之后 f=C2H W=D9H
例子 2 ADDWF f 1	指令执行之前 f=C2H W=17H 指令执行之后 f=D9H W=17H

4

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

5

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

6

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

7

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

8

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

9

BTFSC	如果 bit 测试为 0 则跳转
指令格式	BTFSC f, b 0<=f<=1FFH 0<=b<=7
操作	Skip if (f[b])=0
标志位	无
描述	如果 f 的 bit 位是 0，下一条取到的指令将被丢到，然后执行一条空指令组成一个两周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期
例子 NODE 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<=1FFH 0<=b<=7
操作	Skip if (f[b])=1
标志位	无
描述	如果 f 的 bit 位是 1，下一条取到的指令将被丢到，然后执行一条空指令组成一个两周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期
例子 NODE 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<=1FFH
操作	(f)←0
标志位	Z
描述	将 f 的内容清零
周期	1
例子 CLRF WORK	在指令执行之前 WORK=5AH 在指令执行之后 WORK=00H

\*注。当 clrf 清除 status 寄存器时，标志位 Z 不会置高

13

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

14

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

15

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

16

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

17

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

18

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

21

INCFSZ	f 加 1, 如果结果为 0 跳转
指令格式	INCFSZ f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<-(f)+1 如果结果为 0 就跳转
标志位	无
描述	f 的内容加 1。 如果 d 为 0, 结果保存到工作寄存器中。 如果 d 为 1, 结果保存到 f 中 如果结果为 0, 下一条已经取到的指令将被丢掉, 然后插入一条 NOP 指令组成一个两个周期的指令。
周期	无跳转则为 1 个指令周期, 否则 2 个指令周期
例子 Node INCFSZ FLAG, 1 OP1: OP2:	在指令执行之前 PC=address(Node) 在指令执行之后 (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<=1FFH d=0,1
操作	(目的地址)<←(W) (f)
标志位	Z
描述	f 和工作寄存器或 当 d 为 0 时, 结果保存到工作寄存器中 当 d 为 1 时, 结果保存到 f 中
周期	1
例子 IORWF f,1	在指令执行前 W=88H f=23H 在指令执行后 W=88H f=ABH

24

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

25

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

26

MOVVP	读 EPROM 数据
指令格式	MOVVP
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 MOVVP	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<=1FFH
操作	(f)<←(W)
标志位	无
描述	将工作寄存器的值传送到 f 中
周期	1
例子 MOVWF f	在指令执行之前 W=88H f=23H 在指令执行之后 W=88H f=88H

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) $\Rightarrow$ 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) $\Rightarrow$ PC Pop Stack 1 $\Rightarrow$ GIE
标志位	无
描述	PC 从堆栈顶部得到，然后出栈，设置全局中断使能位为 1
周期	2

32

RETLW	返回，并将立即数送到工作寄存器中
指令格式	RETLW K 0 $\leq$ K $\leq$ FFH
操作	(W) $\leftarrow$ K (Top Stack) $\Rightarrow$ 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) $\Rightarrow$ 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<=1FFH 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<=1FFH d=0,1
操作	(目标地址[n-1])<-(f[n]) (目标地址[7])<-C C<-(f[0])
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 通过中断唤醒
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<=1FFH d=0,1
操作	(目标地址)<-(f)-(W)
标志位	C, DC, Z
描述	f 的值减去工作寄存器的值。 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 SUBWF f, 1	在指令执行之前 f=33H W=01H 在指令执行之后 f=32H C=1 Z=0
例子 2 SUBWF f, 1	在指令执行之前 f=01H W=01H 在指令执行之后 f=00H C=1 Z=1
例子 3 SUBWF f, 1	在指令执行之前 f=04H W=05H 在指令执行之后 f=FFH C=0 Z=0

39

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

40

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

41

TBLP	写 Memory
指令格式	TBLP k (k 取 0) 8 8
操作	写 Memory
标志位	无
描述	把 EDATH/WORK 的数据写到存储器地址 EADRH/EADRL
周期	约为 300us 时间
例子 TBLP 0	在指令执行之前 EDATH=bah, W=Ach, EADRH=04h, EADRL=00h 在指令执行之后 把 baach 写到存储器地址 0400h



42

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

## 5 电气特性

### 5.1 极限值

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

### 5.2 直流特性 (VDD = 5V, T<sub>A</sub> = 25℃, 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℃	2.2	5	5.5	V
		-40℃ ~+85℃	2.3	5	5.5	V
FXT	外部晶振频率				16	MHz
EXT_CK	外部灌入时钟	25℃, 代码选项 XTAL_PIN 选外部低速晶振			50	KHz
		25℃, 代码选项 XTAL_PIN 选外部高速晶振			4	MHz
Vpor (上电复位 延时 1ms)	系统电源电压上升速率			3.3		V/ms
Vpor (上电复位 延时 10ms)	系统电源电压上升速率			0.33		V/ms
Vpor (上电复位 延时 49ms)	系统电源电压上升速率			0.067		V/ms
Vpor (上电复位 延时 98ms)	系统电源电压上升速率			0.033		V/ms
Tcpu	指令周期	VDD: 2.4V~5.5V	250			ns
		VDD: 2.2V~5.5V	500			ns
VIH1 (VTH_SEL=0, 打开施密特输入)	数字输入高电平	PT1, PT3, PT5	0.7VDD			V
VIL1 (VTH_SEL=0, 打开施密特输入)	数字输入低电平	PT1, PT3, PT5			0.3VDD	V
VIH2 (VTH_SEL=1, 打开施密特输入)	数字输入高电平	PT1, PT3, PT5(普通 IO)	2.0			V
	OSCI	PT1.1, PT1.2(做时钟口时)	2.5			
VIL2	数字输入低	PT1, PT3, PT5			0.8	V

(VTH_SEL=1, 打开施密特输入)	电平					
	OSCI	PT1.1, PT1.2(做时钟口时)			1.0	
VIH3 (关闭施密特输入)	数字输入高电平	PT1, PT3, PT5		VDD/2		V
VIL3 (关闭施密特输入)	数字输入低电平	PT1, PT3, PT5		VDD/2		V
IPU	上拉电流	PT1, PT3, PT5; Vin = 0; VDD=5V;		100		uA
IPD	下拉电流	PT1, PT3, PT5; Vin = VDD; VDD=5V;		100		uA
IOH	高电平输出电流	VOH=0.9VDD; VDD=5V		17		mA
		VOH=0.9VDD; VDD=3V		8		mA
		VOH=0.9VDD; VDD=5V (可配置大驱动口)		53		mA
		VOH=0.9VDD; VDD=3V (可配置大驱动口)		23		mA
IOL	低电平输出电流	VOL=0.1VDD; VDD=5V		40		mA
		VOL=0.1VDD; VDD=3V		18		mA
		VOL=0.1VDD; VDD=5V (可配置大驱动口)		80		mA
		VOL=0.1VDD; VDD=3V (可配置大驱动口)		40		mA
LVD	复位电压/低电压检测电压	2.0V 上电/掉电复位点; -40~85 度	1.6	1.8	2.0	V
		2.4V 上电/掉电复位点; -40~85 度	2.2	2.4	2.6	
		2.5V 上电/掉电复位点; -40~85 度	2.3	2.5	2.7	
		3.0V 上电/掉电复位点; -40~85 度	2.8	3.0	3.2	
		3.6V 上电/掉电复位点; -40~85 度	3.4	3.6	3.8	
IRC	内置 RC 时钟 32MHz	25°C, 5V	-1%	32.0	+1%	MHz
		-40°C~85°C, 2.3V~5.5V	-3%	32.0	+3%	
	内置 RC 时钟 16MHz	25°C, 5V	-1%	16.0	+1%	MHz
		-40°C~85°C, 2.3V~5.5V	-3%	16.0	+3%	
	内置 RC 时钟 8MHz	25°C, 5V	-1%	8.0	+1%	MHz
		-40°C~85°C, 2.3V~5.5V	-3%	8.0	+3%	
	内置 RC 时钟 4MHz	25°C, 5V	-1%	4.0	+1%	MHz
		-40°C~85°C, 2.3V~5.5V	-3%	4.0	+3%	
	内置 RC 时钟 2MHz	25°C, 5V	-1%	2.0	+1%	MHz
		-40°C~85°C, 2.3V~5.5V	-3%	2.0	+3%	
WDT	内置看门狗时钟	25°C, 5V	-5%	32768	+5%	Hz
		-40°C~85°C, 2.3V~5.5V	-20%	32768	+20%	Hz
Tint0,1	中断触发脉宽	25°C, 5V	Tcpu			ns
IDD1	sleep 模式电流	VDD=3V, 关掉 WDT		0.8		uA
		VDD=3V, 打开 WDT		2.5		uA
		VDD=5V, 关掉 WDT		1.0		uA
		VDD=5V, 打开 WDT		4		uA
IDD2	Halt 模式	VDD=3V		1200		uA

IDD3	工作电流 LIRC=32768 Hz	内部振荡器关闭 (fcpu=fosc/4) fosc = 32768Hz, VDD=3V		19		uA
		内部振荡器关闭 (fcpu=fosc/4) fosc = 32768Hz, VDD=5V		22		uA
	工作电流 IRC=16MHz	内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/4		3.2		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/8		2		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/16		1.5		mA
		内部晶振模式, F=16MHz, VDD=3V, fcpu=fosc/32		1.2		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/4		4.0		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/8		2.6		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/16		2.0		mA
		内部晶振模式, F=16MHz, VDD=5V, fcpu=fosc/32		1.5		mA
	工作电流 IRC=32MHz	内部晶振模式, F=32MHz, VDD=5V, fcpu=fosc/4		7.5		mA
	工作电流 IRC=8MHz	内部晶振模式, F=8MHz, VDD=5V, fcpu=fosc/4		2.2		mA
		内部晶振模式, F=8MHz, VDD=3V, fcpu=fosc/4		1.6		mA
	工作电流 IRC=4MHz	内部晶振模式, F=4MHz, VDD=5V, fcpu=fosc/4		1.35		mA
		内部晶振模式, F=4MHz, VDD=3V, fcpu=fosc/4		1.1		mA
	工作电流 IRC=2MHz	内部晶振模式, F=2MHz, VDD=5V, fcpu=fosc/4		0.9		mA
		内部晶振模式, F=2MHz, VDD=3V, fcpu=fosc/4		0.7		mA

5.3 ADC 特性 (VDD = 5V, TA = 25 °C, 如无其他说明则都是此条件)

符号	参数	测试条件	最小值	典型值	最大值	单位
VDD	ADC 工作电 压范围	25 °C	2.3	5	5.5	V
		-40 °C ~+85 °C	2.5	5	5.5	V
AIN0~ AIN5 input voltage	模拟输入范 围	VREF 受寄存器 VREFS[1:0]控制	0		VREF	V
Vref input range	外部参考电 压输入范围	VREFS[1:0]=01	0		VDD	V
ADC current consumption	ADC 功耗	VDD=5V(VDD 作为参考电压)		0.85		mA
		VDD=3V(VDD 作为参考电压)		0.65		mA

ADC Conversion Cycle Time	ADC 转换周期	注意模拟信号的输出阻抗对于ADC 转换周期的限制，典型值10uS 要求输出阻抗不超过 10K		10		uS
INL	积分非线性	SRADACKS[1:0]=01; SRADCKS[1:0]=01;		±4	±8	LSB
No missing code	无失码	VREFS[1:0]=01, 外部参考电压				Bits
		VREFS[1:0]=00, VDD 做为参考电压				Bits
		VREFS[1:0]=10, 内部参考电压	8	9	10	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%	2.0	+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			4		mV

#### 5.4 比较器的直流特性

符号	参数	测试条件	最小值	典型值	最大值	单位
VDD	工作电压	25 °C	2.3	5	5.5	V
		-40 °C ~+85 °C	2.5	5	5.5	V
Comp Vos	比较器/运算放大器的失调电压 (校准后)				±1	mV
Comp Vos drift	比较器/运算放大器的失调电压温漂			5*		uV/°C
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	VDD=5V		550		uA
Iopa	运算放大器工作电流	VDD=5V		700		uA
cmp amplify multiple	运算放大器内置放大倍数			16	16.5	
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			75		dB
CMRR	比较器/运算放大器共模抑制比			60*		dB
Comp/opa vdd range	比较器/运算放大器工作电压范围		2.3V		5.5	V
Comp Reponse	比较器响应时间			1 <sup>(1)</sup>		uS



CMP LSB	比较器最小分辨率			1		mV
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	Isource (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 MTP 及 EEPROM 读写特性

EEPROM 特性如下表

参数	测试条件	最小值	典型值	最大值	单位
写操作电压范围	-40°C~85°C	2.5		5.5	V
读操作电压范围	-40°C~85°C	2.3	5	5.5	V
写操作时间	5V, 25°C		700		us
	3V, 25°C		800		us
	2.5V, 25°C		3		ms
写操作电流	5V, 25°C		30		mA
写操作次数	5V, 25°C		10000		次

MTP 作为数据存储器件时, 特性如下

参数	测试条件	最小值	典型值	最大值	单位
写操作电压范围	-40°C~85°C	4.5		5.5	V
读操作电压范围	-40°C~85°C	2.3	5	5.5	V
写操作时间	5V, 25°C		700		us
写操作电流	5V, 25°C		30		mA
写操作次数	5V, 25°C		1000		次

## 5.6 32MHz IRC 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。



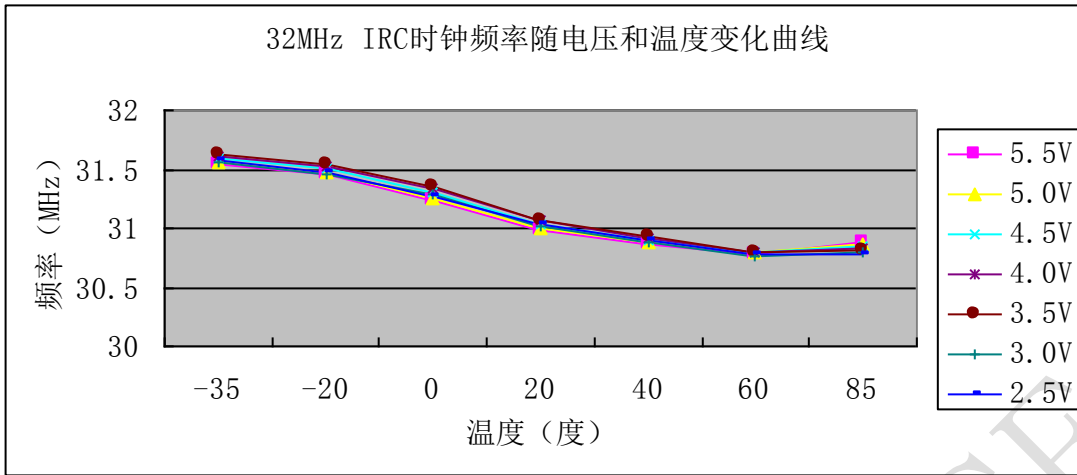


图51 RC时钟频率的电压和温度特性

### 5.7 16MHz IRC 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。

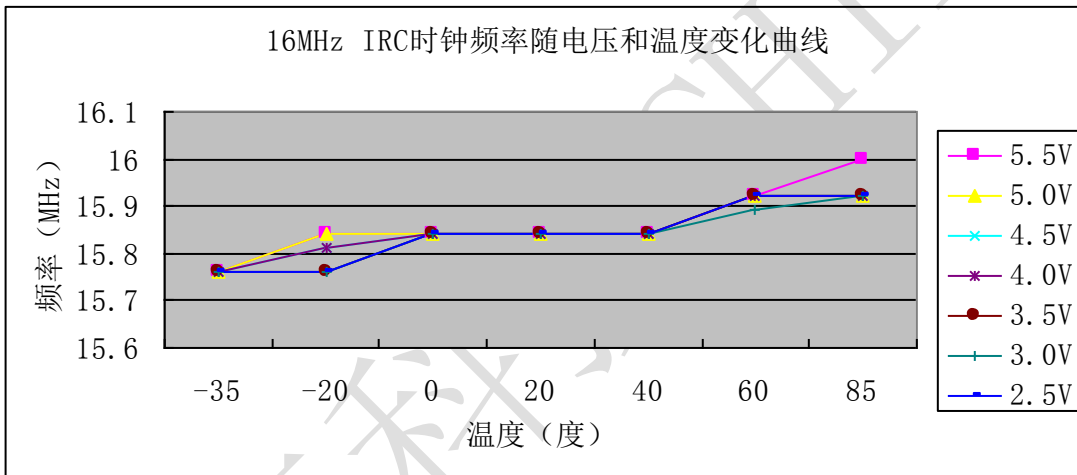


图52 RC时钟频率的电压和温度特性

### 5.8 8MHz IRC 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。

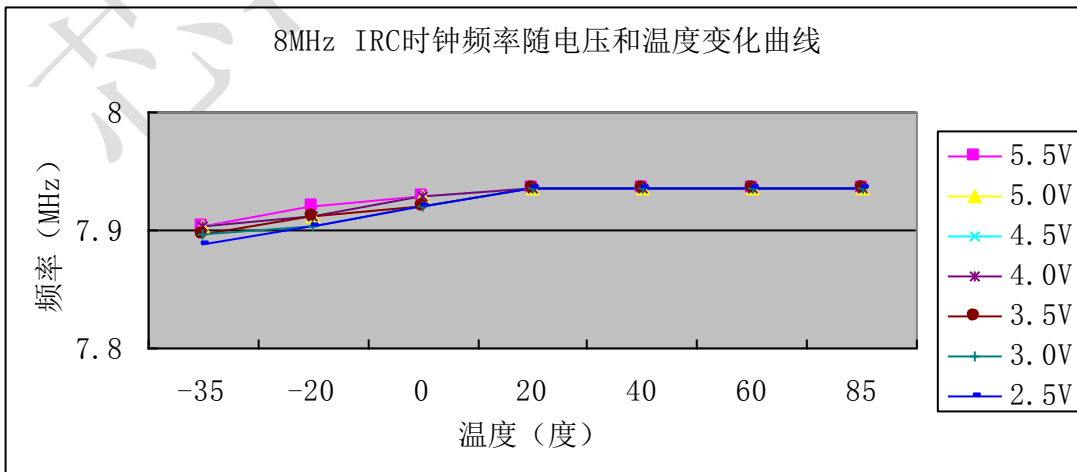


图53 RC时钟频率的电压和温度特性

5.9 4MHz IRC 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。

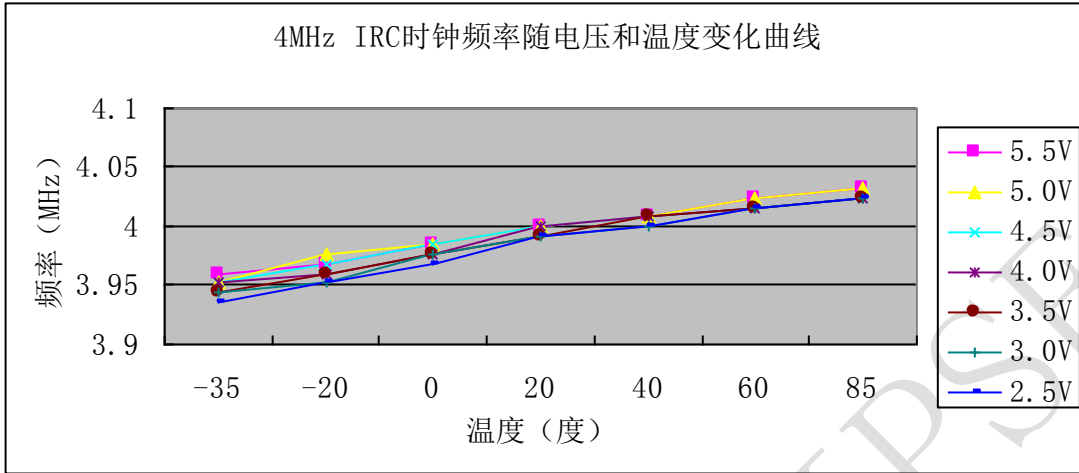


图54 RC时钟频率的电压和温度特性

5.10 2MHz IRC 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。

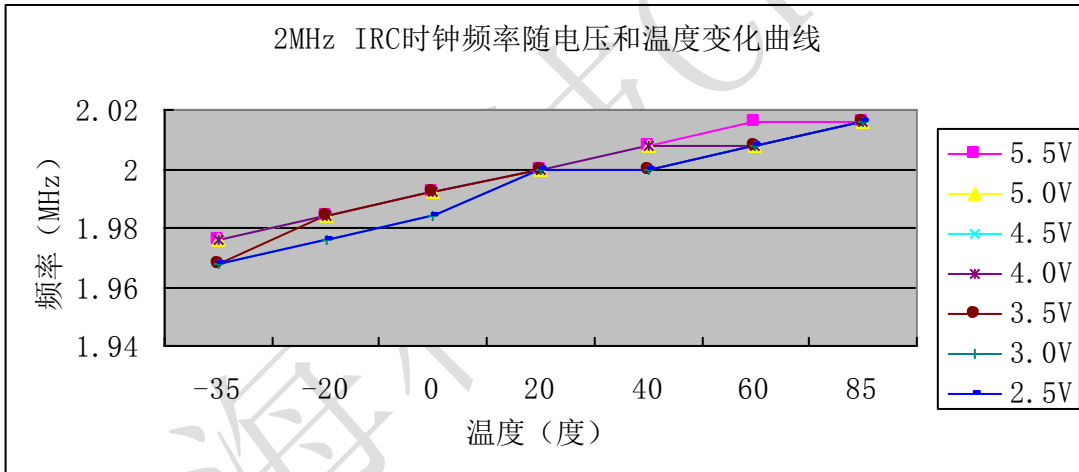


图55 RC时钟频率的电压和温度特性

5.11 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。

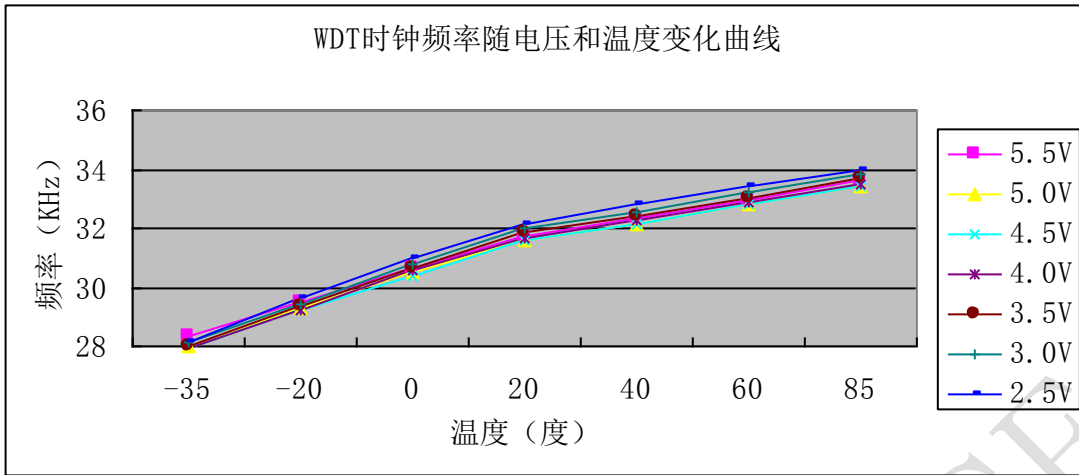


图56 WDT 频率的电压和温度特性

### 5.12 2.0V 掉电复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

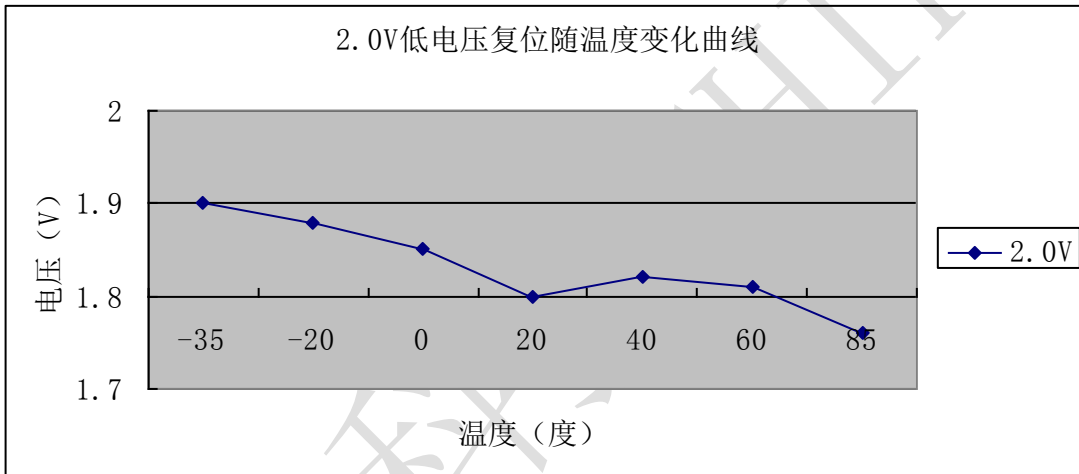


图57 2.0V 掉电复位温度特性

### 5.13 2.4V 低电压复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

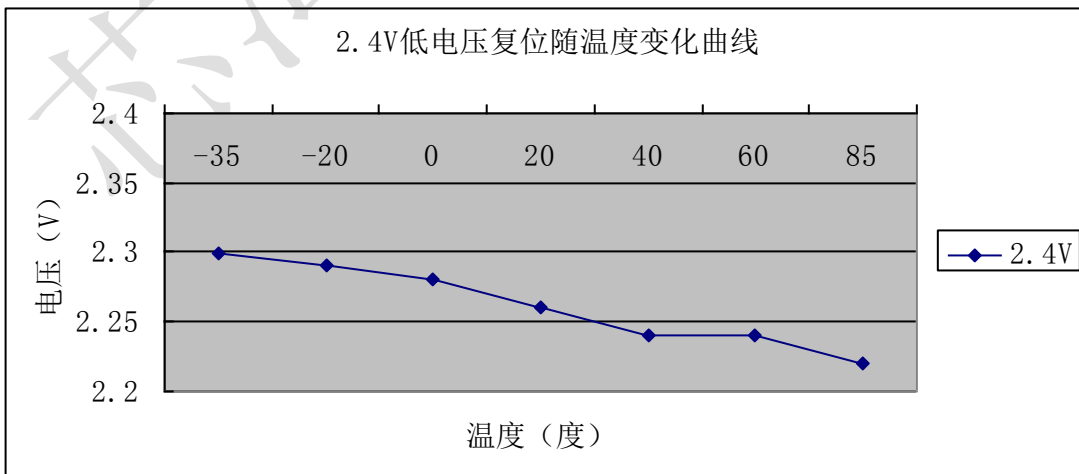
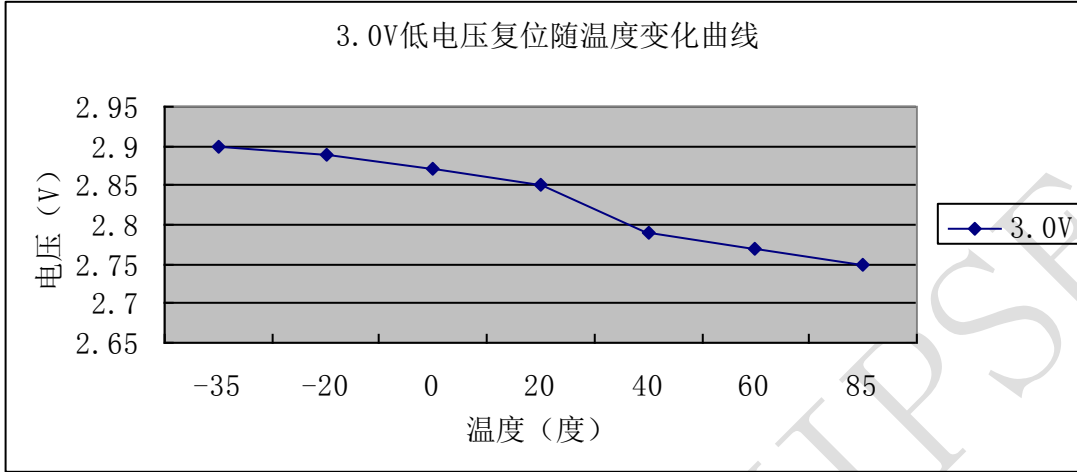


图58 2.4V 低电压复位温度特性

5.14 3.0V 低电压复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。



5.15 3.6V 低电压复位温度特性

下图为实际芯片的测试数据，不同芯片会略微有所差异，仅供参考。

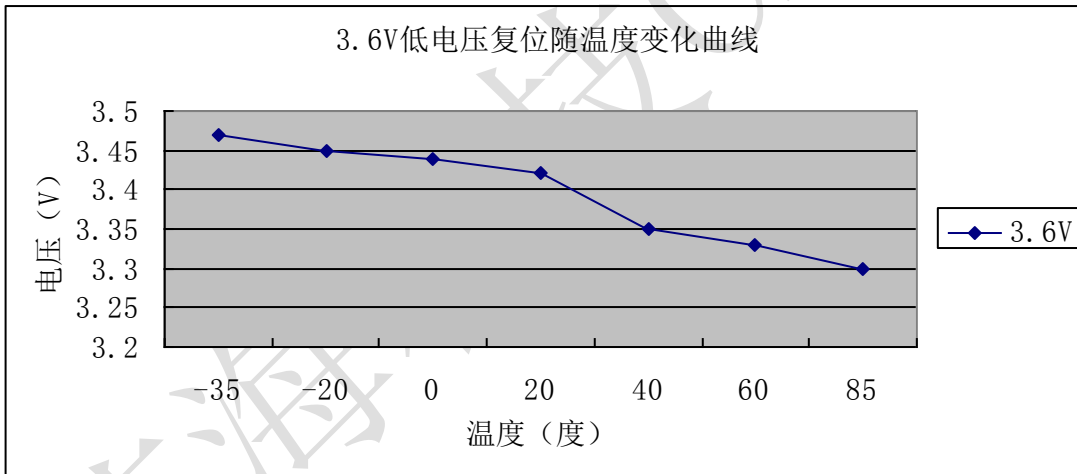


图59 3.6V 低电压复位温度特性

5.16 1.4V 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。

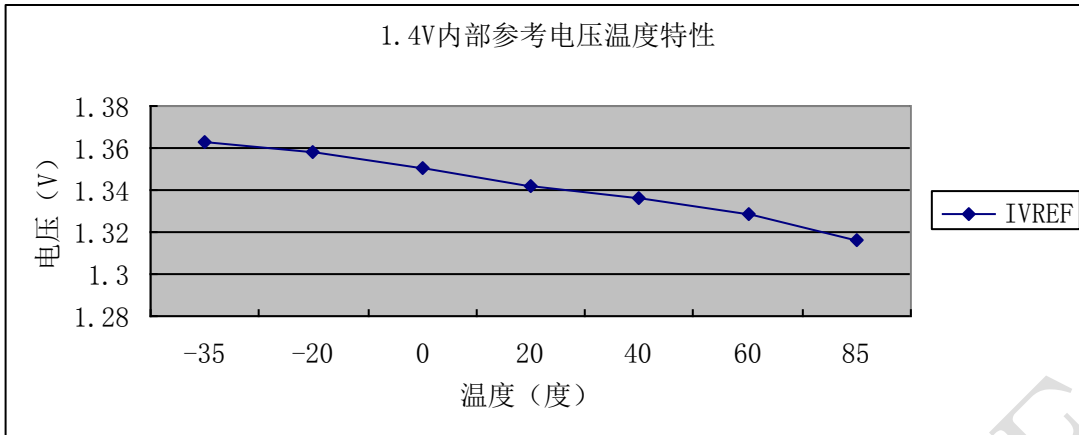


图60 内置参考电压温度特性

### 5.17 2V 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。

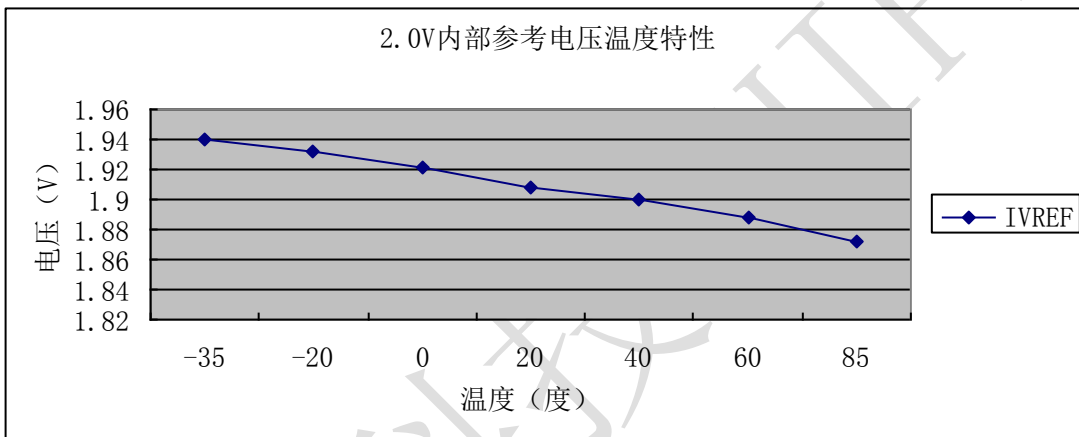


图61 内置参考电压温度特性

### 5.18 3V 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。

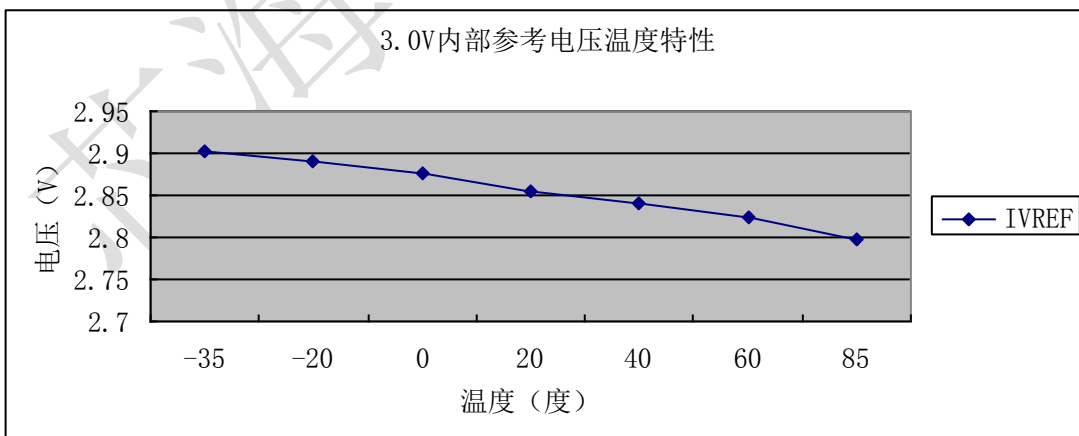


图62 内置参考电压温度特性

### 5.19 4V 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。

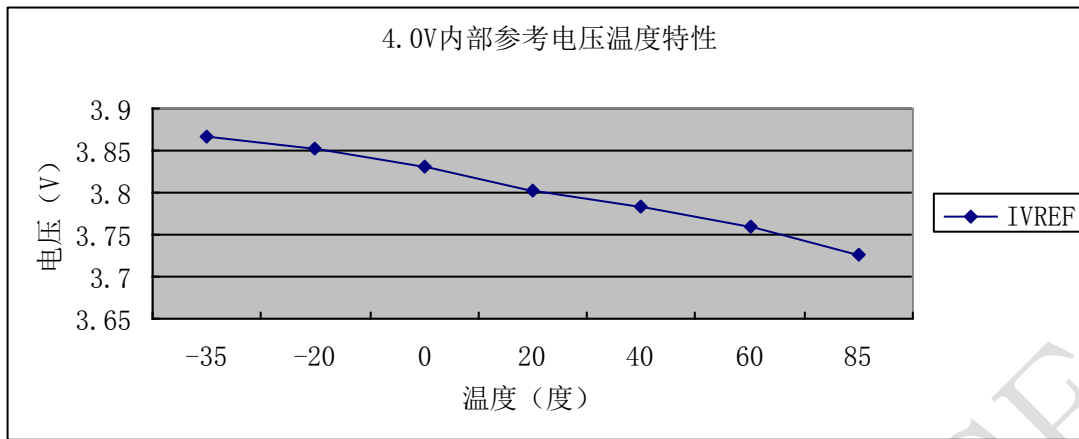
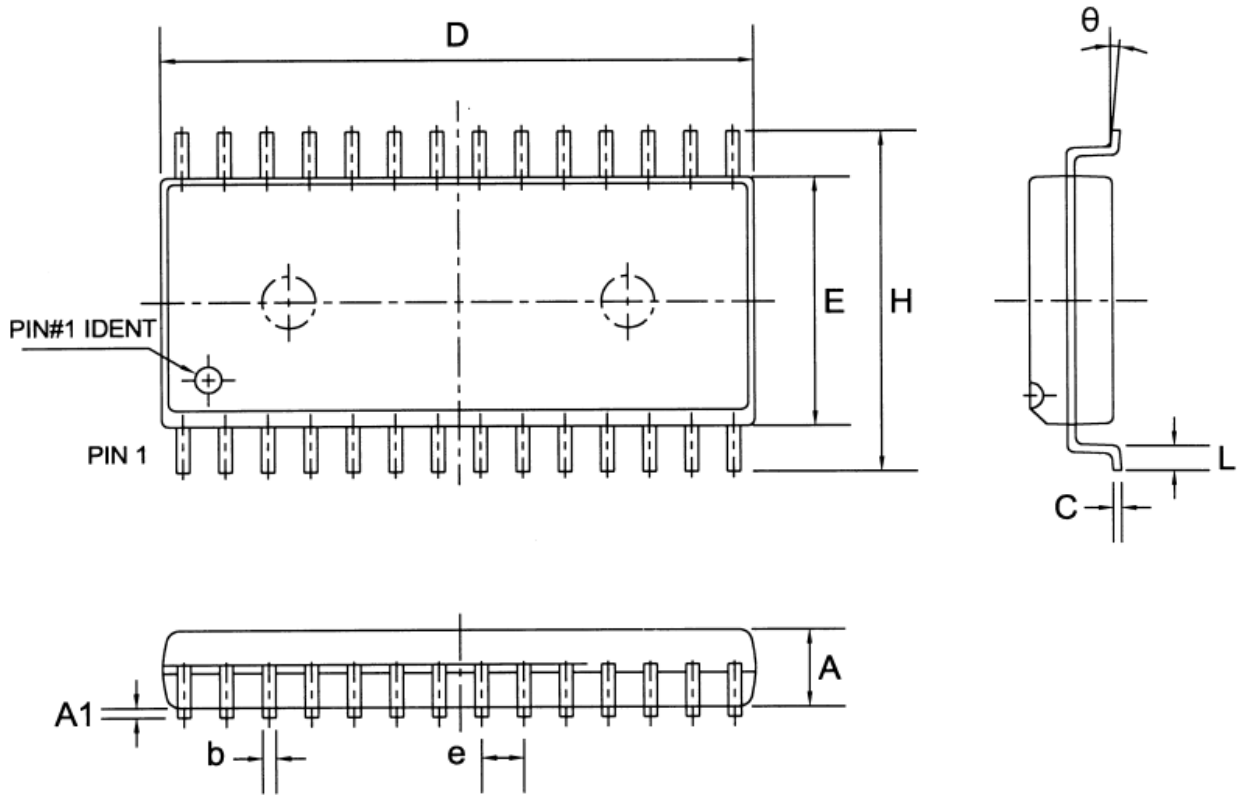


图63 内置参考电压温度特性

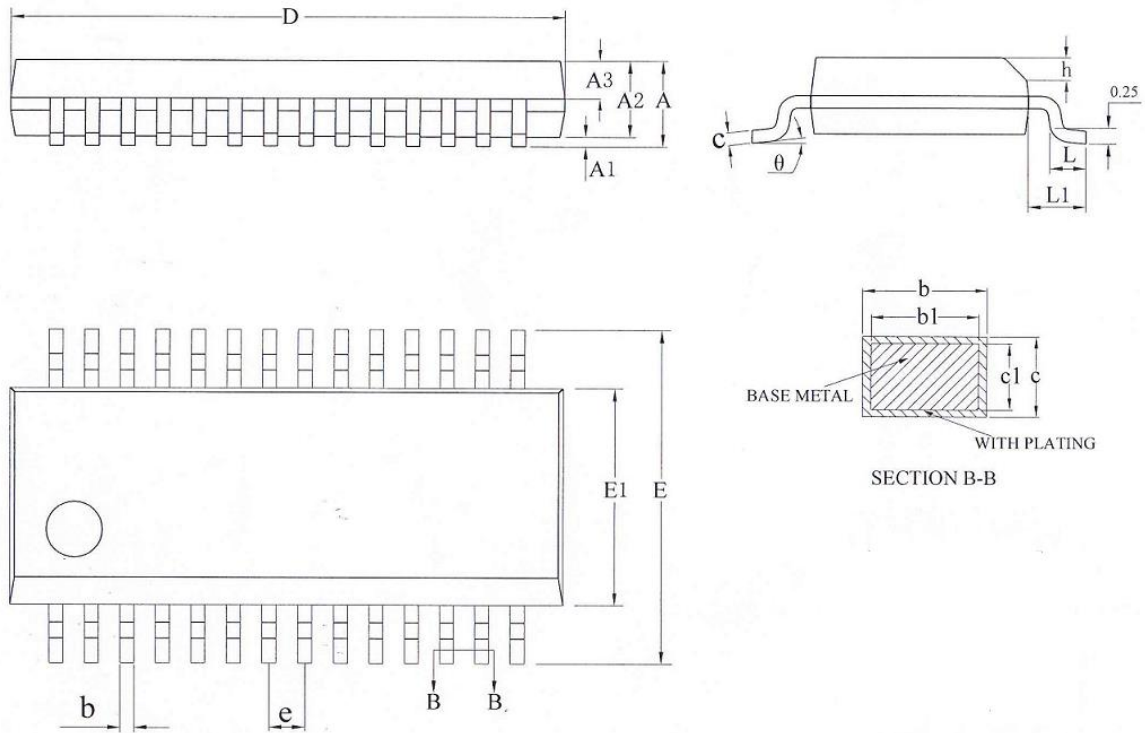
6 封装图

6.1 SOP-28 PIN



SYMBOLS	MIN	NOR	MAX
	(mm)		
A	-	-	2.65
A1	0.10	-	0.30
b	0.39	-	0.48
C	0.25	-	0.31
D	17.80	18.00	18.20
E	7.30	7.50	7.70
e	1.27BSC		
θ	0°	-	8°
H	10.10	10.30	10.50
L	0.70	-	1.00

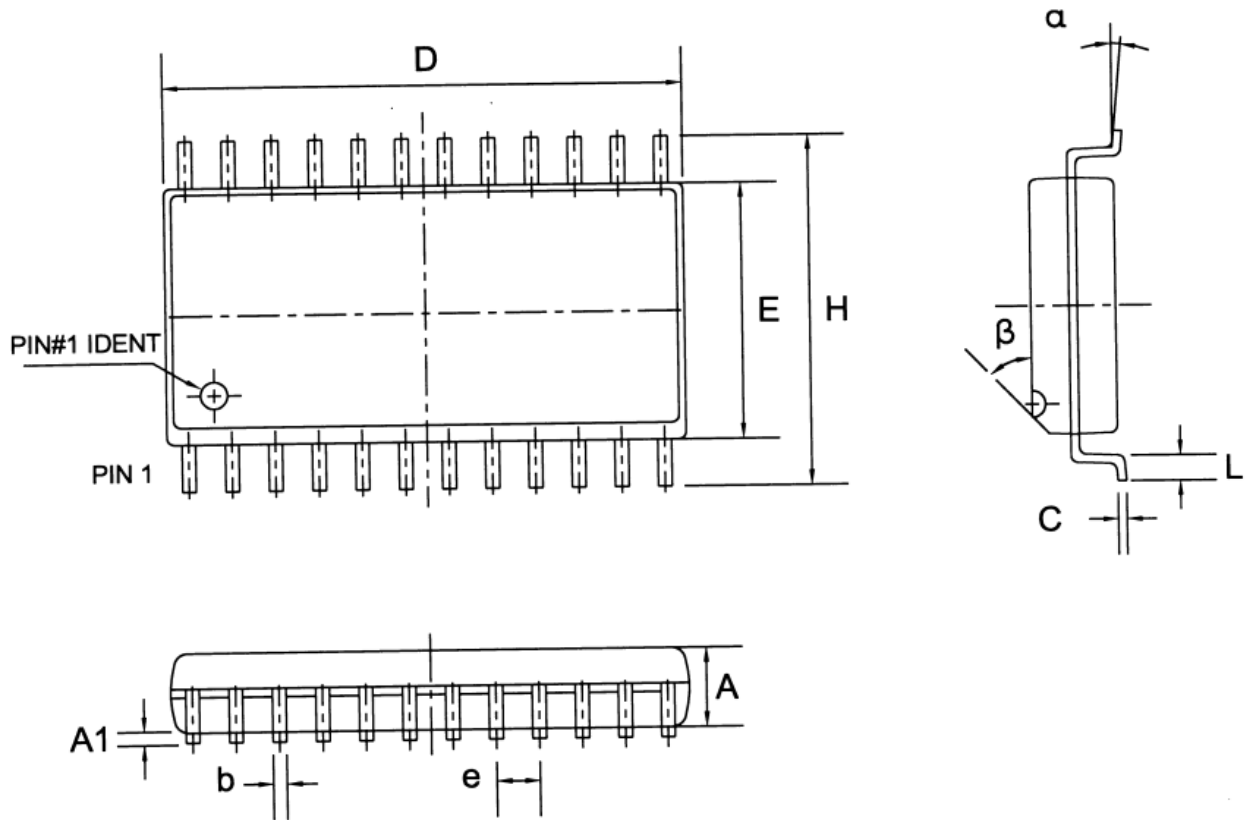
6.2 SSOP-28 PIN



SYMBOLS	MIN	NOR	MAX
	(mm)		
A	-	-	1.75
A1	0.05	-	0.225
b	0.23	-	0.31
C	0.20	-	0.24
D	9.80	9.90	10.00
E1	3.80	3.90	4.00
E	5.80	6.00	6.20
e	0.635BSC		
$\theta$	0°	-	8°
L	0.50	-	0.80

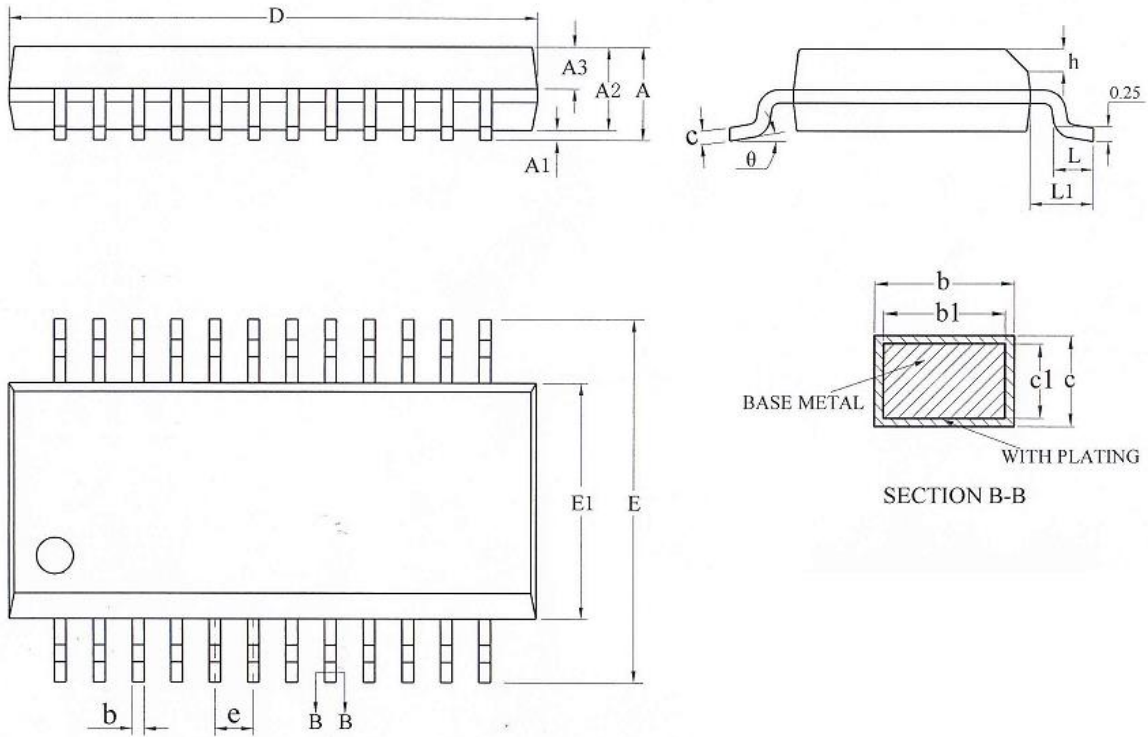


6.3 SOP-24 PIN



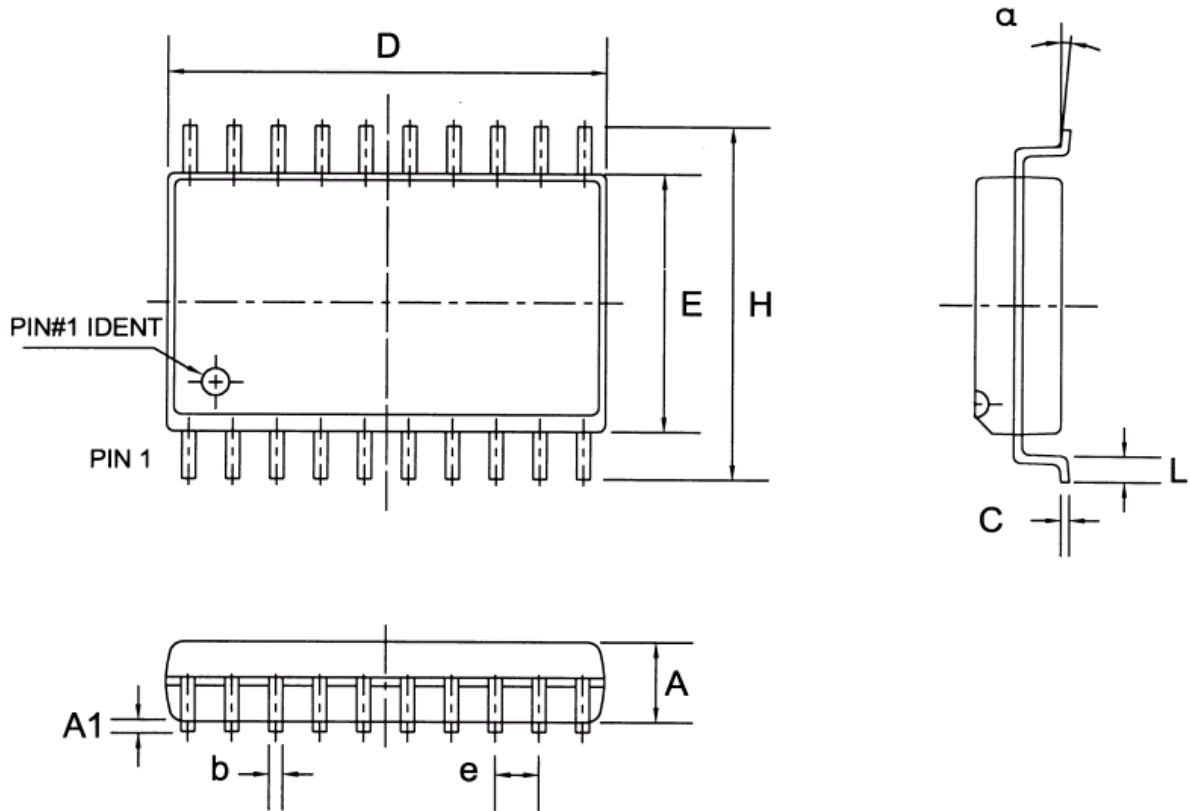
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	-	-	2.70
A1	0.10	0.20	0.30
b	0.39	-	0.48
C	0.25	-	0.31
D	15.2	15.4	15.6
E	7.30	7.50	7.70
e	1.27BSC		
$\alpha$	0°	-	8°
$\beta$	-	45°	-
H	10.1	10.3	10.5
L	0.70	0.85	1.00

6.4 SSOP-24 PIN



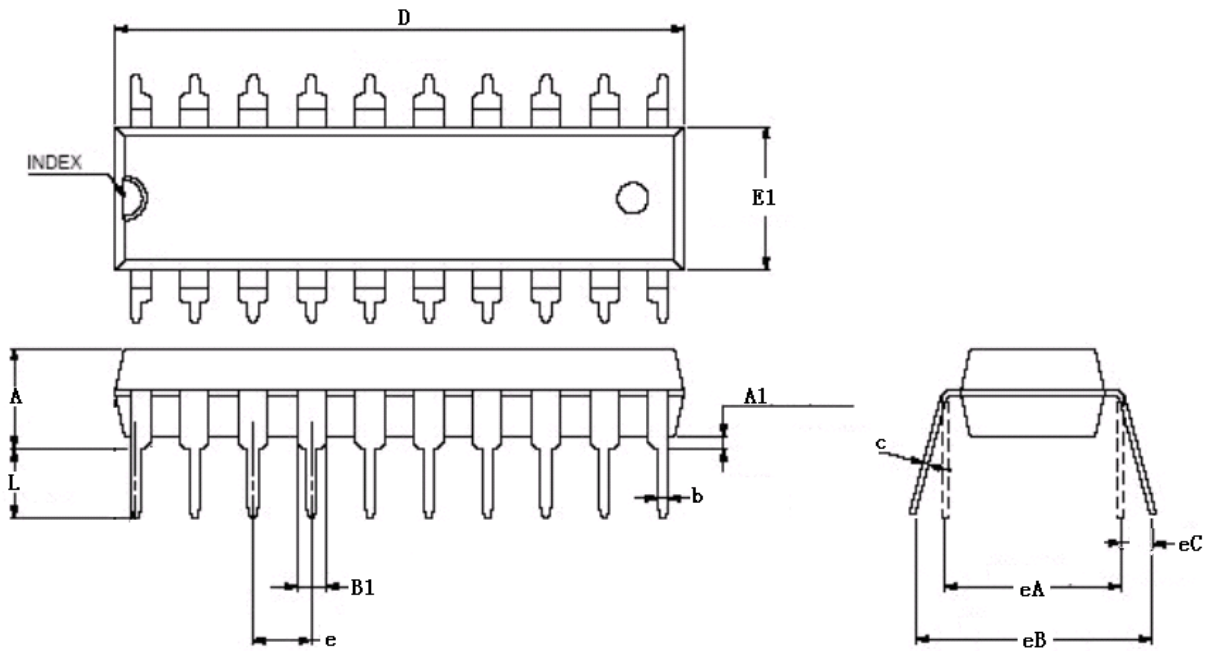
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	-	-	1.75
A1	0.10	0.15	0.25
b	0.23	-	0.31
C	0.20	-	0.24
D	8.55	8.65	8.75
E1	3.80	3.90	4.00
E	5.80	6.00	6.20
e	0.635BSC		
$\theta$	0°	-	8°
L	0.50	-	0.80

6.5 SOP-20 PIN



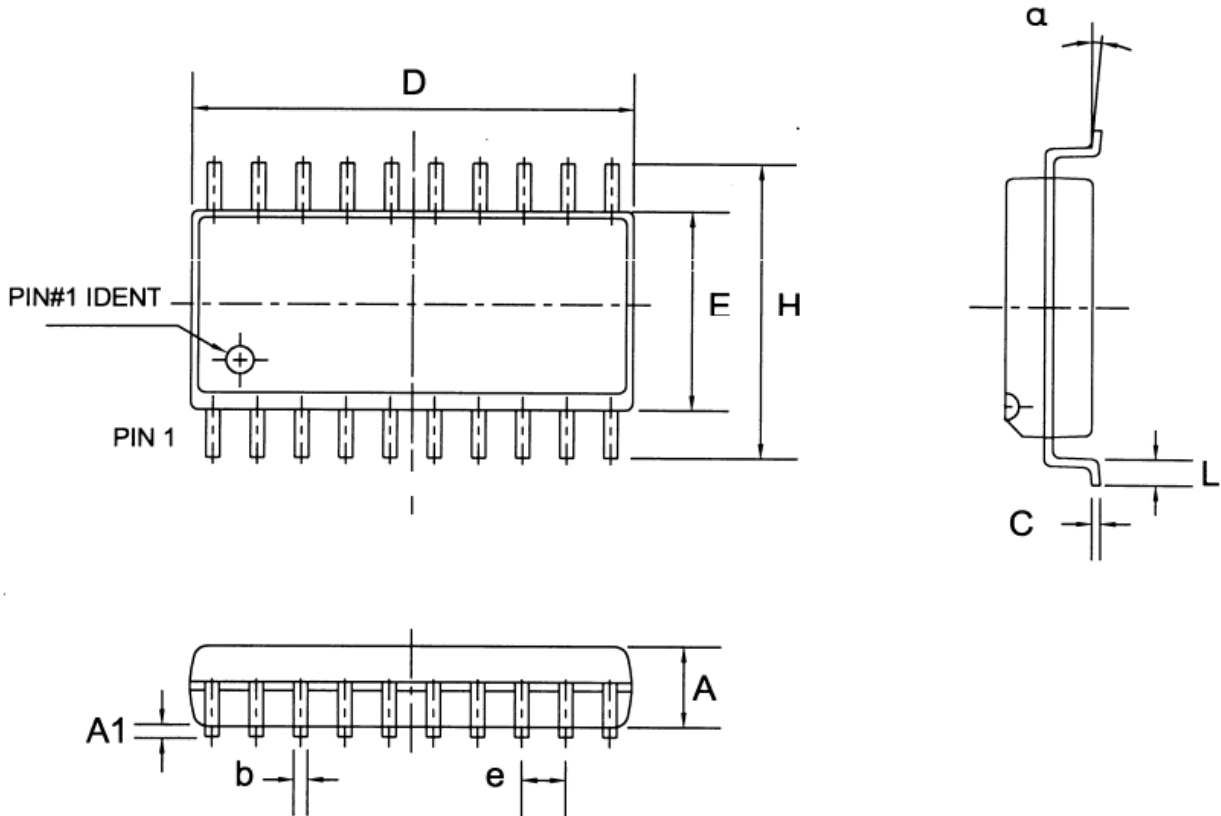
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	2.25	2.30	2.35
A1	0.1	-	0.3
b	0.35	-	0.44
C	0.26	-	0.31
D	12.6	12.8	13.0
E	7.3	7.5	7.7
e	1.27BSC		
$\alpha$	0°	-	8°
H	10.1	10.3	10.5
L	0.7	-	1.0

6.6 DIP-20 PIN



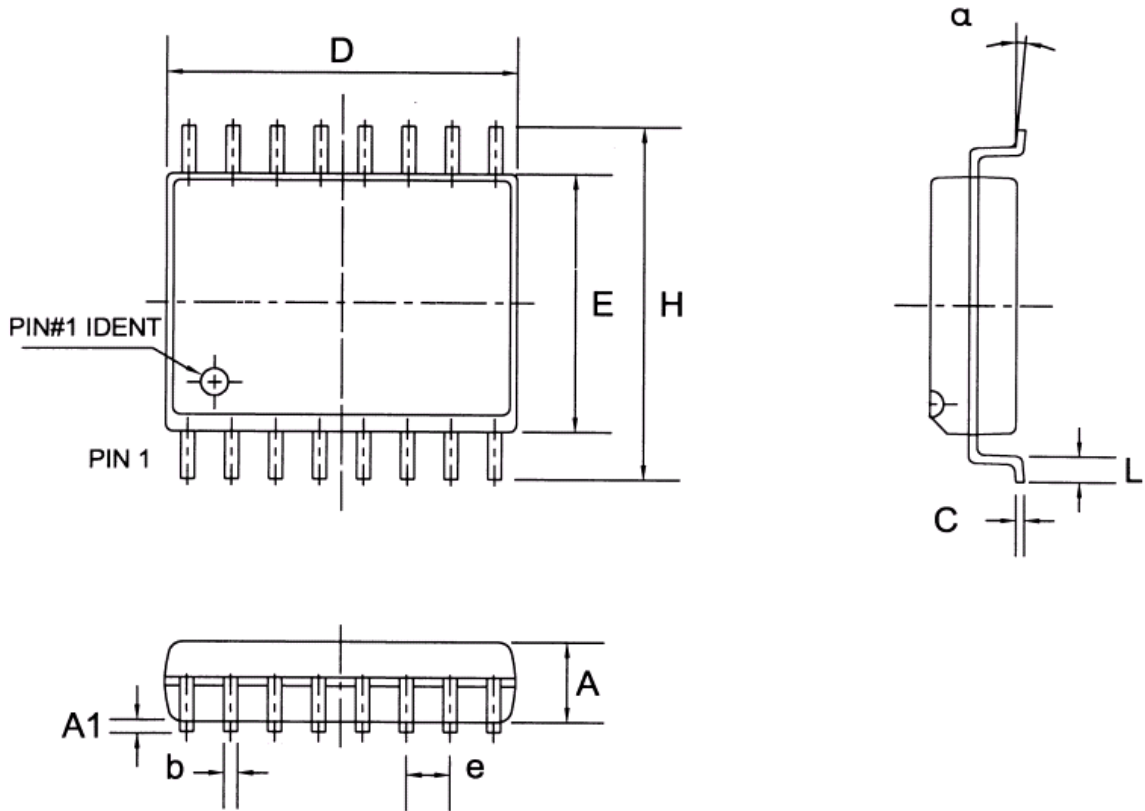
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	3.6	3.8	4.0
A1	0.51	-	-
b	0.44	-	0.53
B1	1.52BSC		
c	0.25	-	0.31
D	26.03	26.23	26.43
E1	6.35	6.55	6.75
e	2.54BSC		
eA	7.62BSC		
eB	7.62	-	9.30
eC	0	-	0.84
L	3.0	-	-

6.7 TSSOP-20 PIN



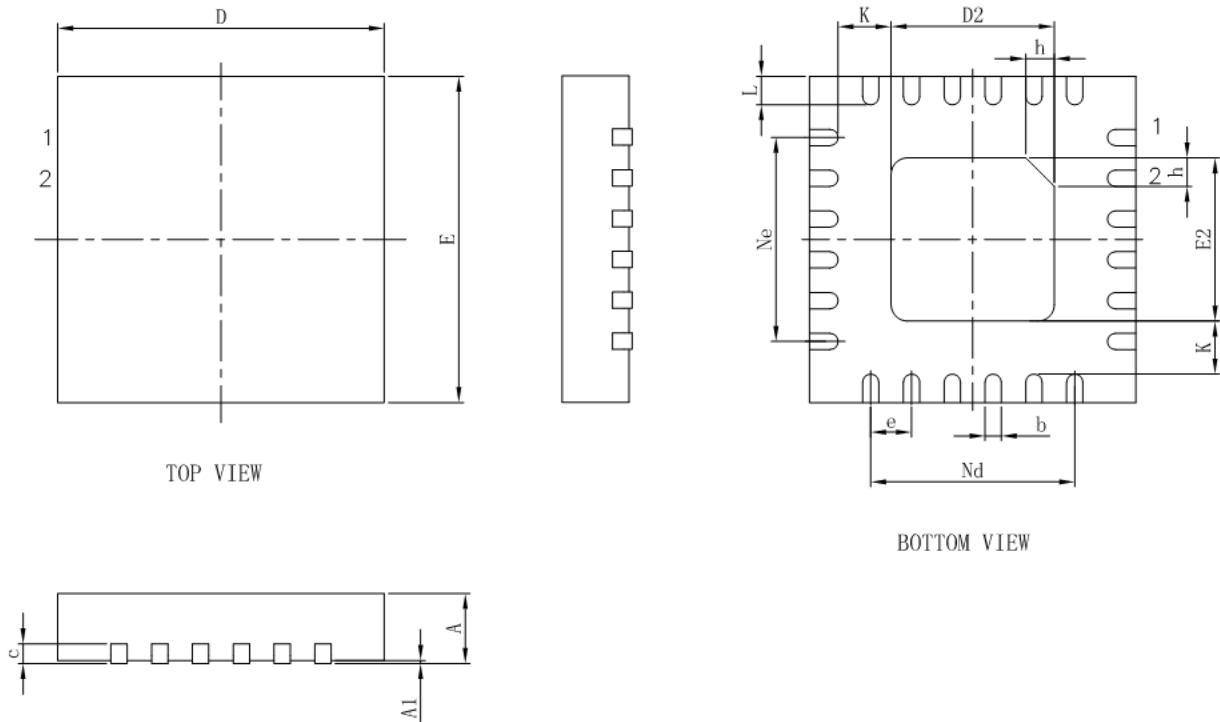
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	0.80	1.00	1.05
A1	0.05	-	0.15
b	0.19	-	0.30
C	0.09	-	0.20
D	6.40	6.50	6.60
E	4.30	4.40	4.50
e	0.65BSC		
$\alpha$	0°	-	8°
H	6.20	6.40	6.60
L	0.45	0.60	0.75

6.8 SOP-16 PIN



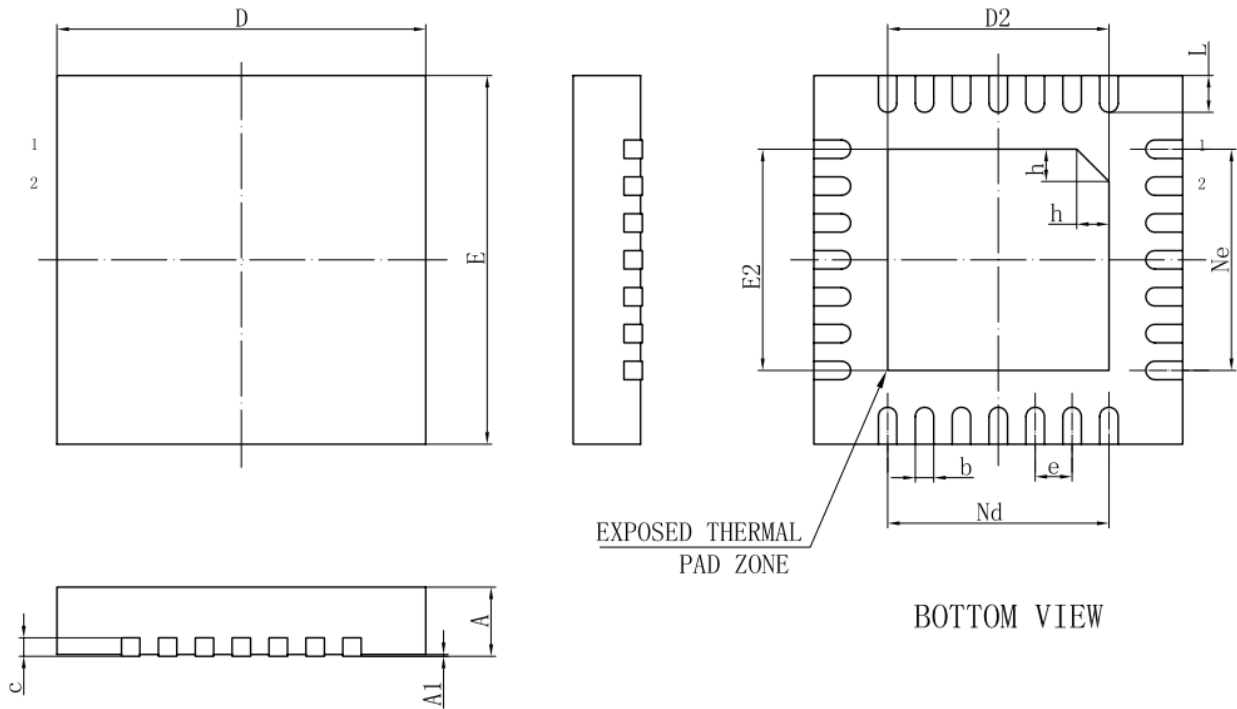
SYMBOLS	MIN	NOR	MAX
	(mm)		
A	1.3	1.40	1.5
A1	0.05	-	0.225
b	0.39	-	0.48
C	0.21	-	0.26
D	9.70	9.90	10.10
E	3.70	3.90	4.10
e	1.27BSC		
$\alpha$	0°	-	8°
H	5.8	6.0	6.2
L	0.5	-	0.8

6.9 QFN-24 PIN (4×4×0.75-0.5)



SYMBOLS	MIN	NOR	MAX
	(mm)		
A	0.70	0.75	0.80
A1	-	0.02	0.05
b	0.18	0.25	0.30
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	1.90	2.00	2.10
e	0.50BSC		
Ne	2.50BSC		
Nd	2.50BSC		
E	3.90	4.00	4.10
E2	1.90	2.00	2.10
L	0.30	0.40	0.50
K	0.20	-	-
h	0.30	0.35	0.40

6.10 QFN-28 PIN (4×4×0.75-0.4)

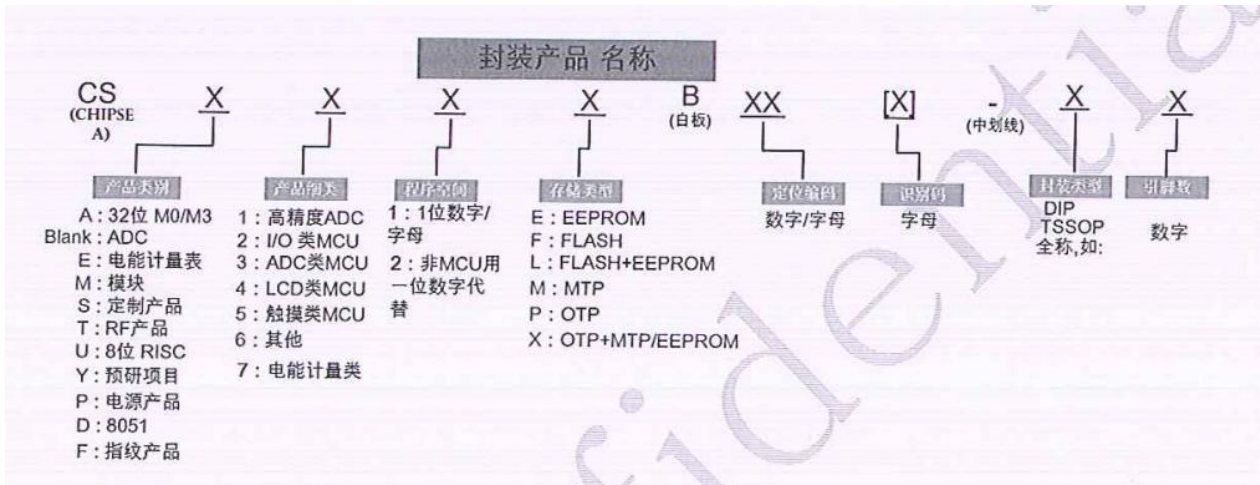


SYMBOLS	MIN	NOR	MAX
	(mm)		
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.30	2.40	2.50
e	0.40BSC		
Nd	2.40BSC		
E	3.90	4.00	4.10
E2	2.30	2.40	2.50
Ne	2.40BSC		
L	0.35	0.40	0.45
h	0.30	0.35	0.40



## 7 单片机产品命名规则

### 7.1 产品型号说明

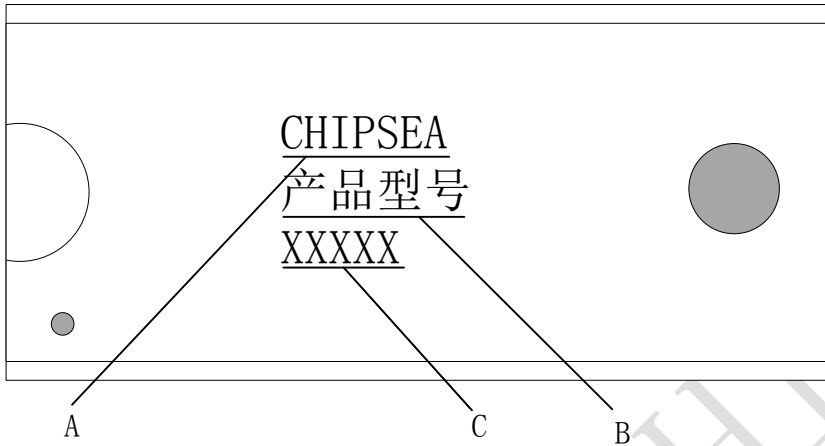


标示符	封装类型
BD	Bonding
DI	DIP
SD	SDIP
SO	SOP
SS	SSOP
TS	TSSOP
QF	QFP
LQ	LQFP
TQ	TQFP
QN	QFN

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
CSU38M20-SOP28	8 位 Risc MCU	MTP	8K*16	ADC	20	第 1 版	SOP	28

7.3 产品印字说明



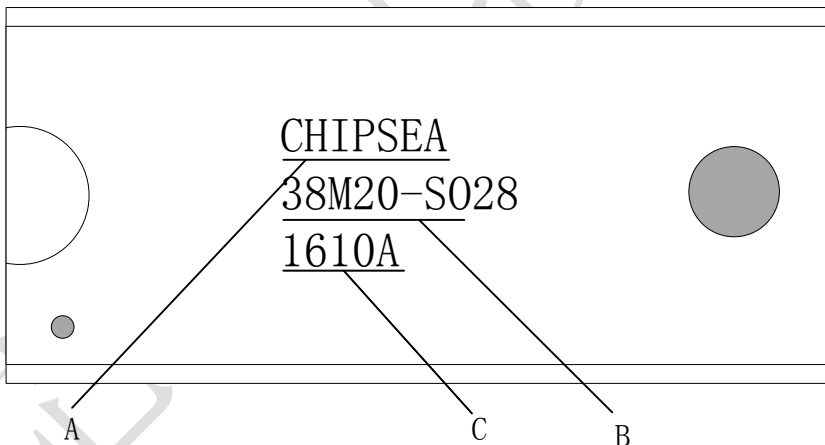
芯片正面印字一般有 3 行：

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，CSU38M20 的印字如下：



注：CSU38M20-SOP28 的产品型号印字为 38M20-SO28。