



芯海科技
CHIPSEA

CSU18M88

用户手册

基于 MTP 的带 24Bits ADC 的 8 位 RISC SOC
REV 3.4

芯海科技（深圳）股份有限公司

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

电 话：+(86 755)86169257 传 真：+(86 755)86169057

网 站：www.chipsea.com 邮 编：518067

微信号：芯海科技



版本历史

历史版本	修改内容	版本日期
REV 1.0	初始版本	2017-05-05
REV 2.0	(1) 删除“时钟自带 LDO 稳压，可以 Bypass”。 (2) 删除 I2C 的 400Khz 高速功能描述。CST_EN 寄存器位必须配置为 0 (3) 增加 RTCCON 备注“注：小时制切换时需要软件更新小时寄存器 RTCHOUR”。 (4) 修改 VS 的管脚描述。 (5) “11 = 24 Bits ADC 输入端连接到 AIN2 和 AIN3, AIN2 为 Vin+, AIN3 为 Vin-; ” 改为“11, 不可选”。 “2 路输入全差分” 改为“1 路输入全差分”。 18M88 的 AIN2 和 AIN3 输入通道不开放。 (6) DAW 指令描述“W=25+39 =64=5EH”改为“W=25+39=5EH” (7) LCDSCLK 统一改成 LCDSCK (8) 删除 6FH 寄存器 (9) 补充对 AIENB2[6][7]对 PT1.0、PT1.1 的控制说明 (10) ISleep3 增加测试条件“LCDREF[2:0]=101, VLCD=3V”。 (11) 更新了 LQFP48L 的封装外形图	2017-10-19
REV 2.1	其中 F4H, PAGE=0 的 174H,PAGE=1 的 174H,PAGE=2 的 174H PAGE=0 的 1F4H,PAGE=1 的 1F4H,PAGE=2 的 1F4H FBH, PAGE=0 的 17BH,PAGE=1 的 17BH,PAGE=2 的 17BH; PAGE=0 的 1FBH,PAGE=1 的 1FBH,PAGE=2 的 1FBH; PAGE=0 的 169H,PAGE=1 的 169H,PAGE=2 的 169H E9H, PAGE=0 的 1E9H,PAGE=1 的 1E9H,PAGE=2 的 1E9H 这些 RAM 地址暂不开放。	2017-11-01
REV 3.0	1. 更新 I2C 2. 加入 I2C START 中断，此中断不能和 TM3 共存	2018-01-08
REV 3.1	1、更改电源极限值由 2.4——3.6 改为-0.3——6	2018-03-14
REV 3.2	1、VS 最小值由 2.35V 改为 2.4V 2、增加 BGID 的配置说明 103 页	2018-04-09
REV 3.3	1. 关闭了部分 PGA 配置。只保留 PGA=1/10/20/80/160 /320。寄存器位 GAIN2 改成 S_GAIN。 2. 修改了 bgid 的描述。“bgid=1, 只有 VS=3.0V、3.3V 可以配置这个档位，仅限 PGA=320。”	2018-06-21
REV 3.4	1、寄存器 Metch[5]=1 改为 0 101 页 2、直流特性中，增加电源 DVDD, AVDD 纹波 155 页 3、增加 24Bits ADC 的特性中的噪声测量条件 156 页 4、修改 24Bits ADC 的特性中参考电压温度系数 157 页 5、修改笔误及排版格式	2018-11-07

目 录

版本历史	2
目 录	3
1 产品概述	6
1.1 功能描述.....	6
1.2 主要特性.....	6
1.3 产品型号、封装和 PIN 配置.....	7
2 标准功能	31
2.1 CPU 核.....	31
2.1.1 存储器.....	32
2.1.2 状态寄存器.....	35
2.1.3 中断寄存器.....	36
2.2 SFR.....	40
2.3 时钟系统.....	43
2.3.1 石英/陶瓷振荡器.....	43
2.3.2 内部振荡器.....	43
2.3.3 CPU 指令周期.....	43
2.3.4 TM0CLK (定时器 0 模块输入时钟).....	44
2.3.5 TM1CLK (定时器 1 模块输入时钟).....	46
2.3.6 TM2CLK (定时器 2 模块输入时钟).....	46
2.3.7 TM3CLK (定时器 3 模块输入时钟).....	47
2.3.8 UART0CLK.....	48
2.3.9 UART1CLK.....	49
2.3.10 蜂鸣器时钟.....	50
2.3.11 SPICLK.....	51
2.3.12 LCDCLK.....	51
2.3.13 I2CCLK.....	51
2.4 复位系统.....	53
2.5 中断.....	54
2.6 PC 跳转.....	56
2.7 定时器 0.....	57
2.8 I/O PORT.....	59
2.8.1 PT1 寄存器 (地址为 1DH).....	60
2.8.2 PT2 寄存器 (地址为 21H).....	64
2.8.3 PT3 寄存器 (地址为 24H).....	67
2.8.4 PT4 寄存器 (地址为 27H).....	68
2.8.5 PT5 寄存器 (地址为 2AH).....	69
2.8.6 PTCON 寄存器 (地址为 2DH).....	70
3 增强功能	74

3.1	Halt 和 Sleep 模式.....	74
3.2	看门狗(WDT).....	75
3.3	电源系统.....	76
3.3.1	Regulator.....	76
3.3.2	低电压比较器.....	77
3.4	定时/计数器 1.....	79
3.4.1	寄存器说明.....	79
3.4.2	定时器 1 使用说明.....	80
3.4.3	蜂鸣器 1 使用说明.....	81
3.4.4	PWM1 使用说明.....	81
3.5	定时/计数器 2.....	82
3.5.1	寄存器说明.....	82
3.5.2	定时器 2 使用说明.....	83
3.5.3	蜂鸣器 2 使用说明.....	84
3.5.4	PWM2 使用说明.....	84
3.6	定时/计数器 3.....	85
3.6.1	寄存器说明.....	85
3.6.2	定时器 3 使用说明.....	87
3.6.3	蜂鸣器 3 使用说明.....	87
3.6.4	PWM3 使用说明.....	87
3.7	串行通信接口 0/1.....	89
3.7.1	波特率配置.....	89
3.7.2	自动波特率.....	89
3.7.3	发送流程.....	90
3.7.4	接收流程.....	90
3.7.5	寄存器说明.....	91
3.8	I2C 从机.....	95
3.8.1	I2C 数据传输起始和终止条件.....	95
3.8.2	7 位地址寻址.....	96
3.8.3	数据传输时序.....	97
3.8.4	寄存器说明.....	99
3.8.5	I2C 使用说明.....	101
3.9	24 Bits ADC 模块.....	104
3.9.1	24 Bits ADC 寄存器说明.....	104
3.10	10 Bits ADC 模块.....	108
3.10.1	10 Bits ADC 寄存器说明.....	108
3.10.2	10 Bits ADC 的操作步骤.....	109
3.11	SPI 接口介绍.....	110
3.11.1	SPI 连接方式.....	110
3.11.2	SPI 时序图.....	111
3.11.3	SPI 寄存器说明.....	113
3.11.4	SPI 操作说明.....	115
3.12	LCD Driver.....	117
3.12.1	LCD 控制模式.....	117
3.12.2	LCD 帧频选择.....	117
3.12.3	LCD 偏置电压.....	119
3.12.4	LCD 驱动波形.....	121
3.12.5	LCD 寄存器说明.....	130
3.12.6	LCD 的操作步骤.....	133
3.13	RTC.....	135

3.13.1	功能描述	135
3.13.2	寄存器说明	135
3.13.3	RTC 的操作步骤.....	138
3.14	数据查表	139
3.15	烧录模块	141
4	MCU 指令集.....	142
5	电气特性.....	155
5.1	极限值	155
5.2	直流特性	155
5.3	24 Bits ADC 的特性	156
5.4	10 Bits SARADC 的特性.....	157
6	封装图外型尺寸.....	159

1 产品概述

1.1 功能描述

CSU18M88 是一颗带 24 Bits ADC 的 SOC，内置 8k×16 位 MTP 程序存储器、128 字节 EEPROM 和 896 字节数据存储器。

1.2 主要特性

高性能的 RISC CPU

8 位 MCU

内置 8k×16Bits 的 MTP 程序存储器(烧录次数 1000 次)和 128Bytes EEPROM(烧录次数 10000 次)

896 字节数据存储器 (SRAM)：需要翻页 43 条指令、8 级存储堆栈

指令周期为：ICK/4、ICK/8、ICK/16、ICK/32

振荡器

8MHz/10.5MHz、4MHz/5.2MHz 振荡器，精度为±1%（常温 25℃，3.3V）

3kHz WDT 振荡器，精度为±10%。

可外接 32768Hz 晶振作为 RTC 时钟，支持 RTC 功能

外设特性

37 个双向 I/O 口(PT1:8 个、PT2:8 个、PT3:8 个、PT4:8 个、PT5:5 个，其中 PT3、PT4 复用作 LCD 口)

10 个内部中断：24 Bits ADC、10Bits ADC、UART0 接收/发送、UART1 接收/发送、TIMER0、TIMER1、TIMER2、TIMER3、SPI 中断、RTC 中断

2 个外部中断：

INT0：PT1.1 或 PT1.2 或 PT1.4 或 PT1.7 或 PT2.1 或 PT2.2 或 PT2.4 或 PT2.7

INT1：PT1.0 或 PT1.3 或 PT1.5 或 PT1.6 或 PT2.0 或 PT2.3 或 PT2.5 或 PT2.6

16 个具有唤醒功能的输入口：PT1.0~PT1.7、PT2.0~PT2.7

1 路蜂鸣器：由 PT2.1 输出，电流能力 6mA

1 路输入全差分 24 Bits Sigma-Delta 型 ADC，1~840 倍多种增益选择，10Hz~5KHz 多种 DataRate 选择

内置温度传感器

内置 8 路单端输入的 10 位 ADC，PT1.0、

PT1.1、PT2.0~PT2.3、PT2.6、PT2.7、

具有 RTC 功能，可以显示年、月、日、星期、小时、分、秒信息，无闹钟功能

2 路 UART：PT1.2/1.3 或 PT2.2/2.3、PT1.6/1.7 或 PT2.6/2.7

波特率 115200/57600/19200/9600

1 路 SPI(带 Master 和 Slave：PT1.2、1.3、1.4、1.5)，最高通信速率 500kHz

内置比较器：用来做电源电压比较和外部电压检测

支持 4*20、6*18 点阵 LCD

专用微控制器的特性

上电复位 (POR)

上电复位延迟定时器 (典型值 39ms)

低电压复位 (LVR，典型值偏差 50 mV)

定时器 0

-可编程预分频的 8 位的定时器

定时/计数器 1、2、3

-可编程预分频的 8 位的分频器

看门狗定时器 (3K WDT)

CMOS 技术

电压工作范围

— DVDD: 2.4V~3.6V; AVDD: 2.4V~3.6V

低功耗特性

MCU 工作电流

— 正常模式(工作电压 3V，27℃)

数字部分 0.46mA@1MHzMCU 时钟(不开外设)

模拟部分典型值 0.45mA@一般模式

模拟部分 0.75mA@性能模式

— 休眠模式下的电流典型值小于 0.8μA

— 休眠+WDT 模式下的电流典型值小于 2μA

封装：SOP16、SOP20、SOP24、LQFP48

1.3 产品型号、封装和 PIN 配置

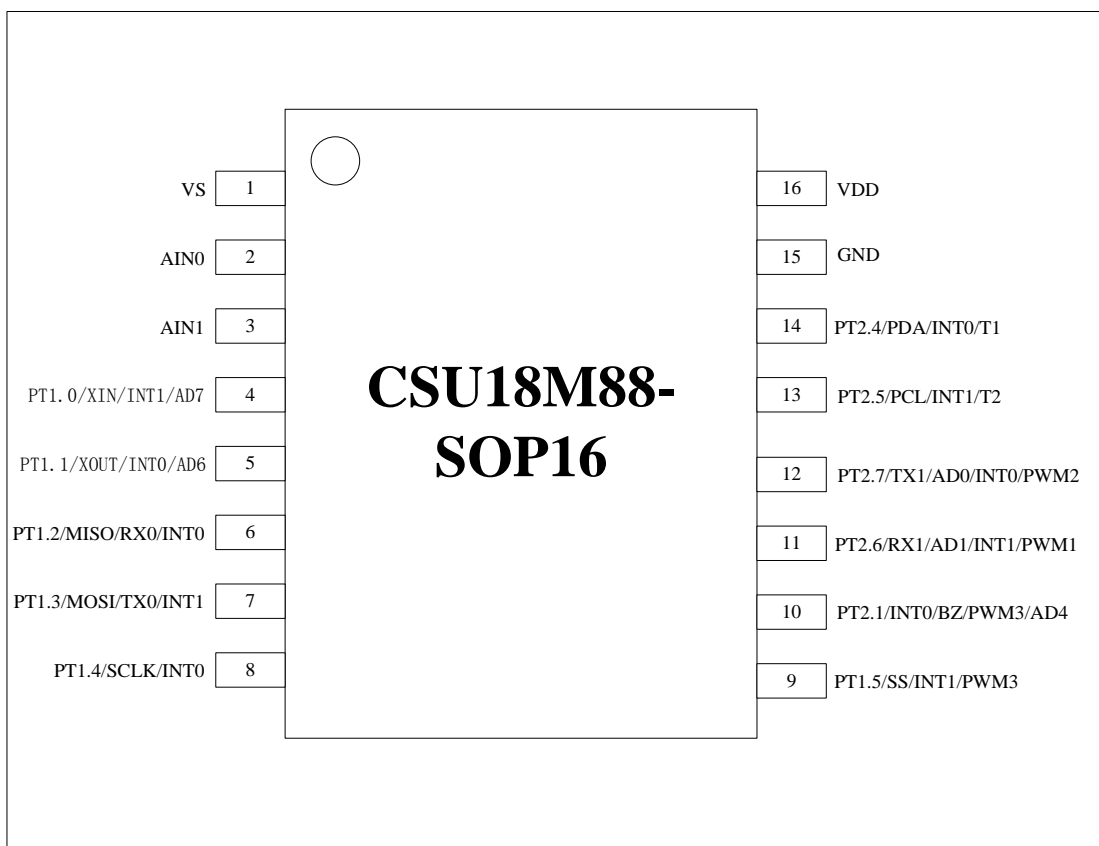


图 1 CSU18M88 SOP16 封装 pin 脚图

表 1 引脚说明表

管脚名称	输入 / 输出	管脚序号	描述										
VS	I	1	芯片的 LDO 输出，外接稳压电容，并做为 ADC 的参考电压输入。										
AIN0~1	I	2~3	24 Bits ADC 模拟差分输入端，寄存器位 SINL[1:0]控制差分输入： =00：输入端连接到 AIN0 和 AIN1:AIN0 为 Vin+、AIN1 为 Vin- =01：内短 =10：24 Bits ADC 输入端连接到温度传感器										
PT1.0/XIN/INT1/AD7	I/O	4	I/O：32768Hz 晶振输入；外部中断 1；ADC 模拟输入通道 7 <table border="1" style="width: 100%;"> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> <tr> <td>AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0</td> <td>IO(默认值)</td> </tr> <tr> <td>AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1</td> <td>外部中断 1 输入</td> </tr> <tr> <td>AIENB3=0</td> <td>32768Hz 时钟输入</td> </tr> <tr> <td>AIENB3=1 且 AIENB2[7]=0</td> <td>ADC 模拟输入</td> </tr> </table>	控制信号	IO 功能描述	AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0	IO(默认值)	AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1	外部中断 1 输入	AIENB3=0	32768Hz 时钟输入	AIENB3=1 且 AIENB2[7]=0	ADC 模拟输入
控制信号	IO 功能描述												
AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0	IO(默认值)												
AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1	外部中断 1 输入												
AIENB3=0	32768Hz 时钟输入												
AIENB3=1 且 AIENB2[7]=0	ADC 模拟输入												

PT1.1/XOUT/INT0/AD6	I/O	5	I/O: 32768Hz 晶振输出; 外部中断 0; ADC 模拟输入通道 6					
			控制信号		IO 功能描述			
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=0		IO(默认值)			
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=1		外部中断 0 输入			
			AIENB3=0		32768Hz 晶振输出			
AIENB3=1 且 AIENB2[6]=0		ADC 模拟输入						
PT1.2/MISO/RX0/INT0	I/O	6	I/O: SPI 主模式数据输入或 SPI 从模式数据输出; 串口 0 输入; 外部中断 0;					
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW0[1]	IO 功能描述
			0	X	0	X	0	IO (默认值)
			0	X	1	1	0	
			0	X	0	X	1	外部中断 0 输入
			0	X	1	1	1	
			1	1	X	X	X	SPI 主模式数据输入
1	0	X	X	X	SPI 从模式数据输出			
0	X	1	0	X	串口 0 输入			
PT1.3/MOSI/TX0/INT1	I/O	7	I/O: SPI 主模式数据输出或 SPI 从模式数据输入; 串口 0 输出; 外部中断 1;					
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW1[1]	IO 功能描述
			0	X	0	X	0	IO (默认值)
			0	X	1	1	0	
			0	X	0	X	1	外部中断 1 输入
			0	X	1	1	1	
			1	1	X	X	X	SPI 主模式数据输出
1	0	X	X	X	SPI 从模式数据输入			
0	X	1	0	X	串口 0 输出			
PT1.4 /SCLK/INT0	I/O	8	I/O: SPI 模式串口时钟输入/输出; 外部中断 0; IO 电压需要根据实际选择 1.8V/3.3V 通信, 通过 PT14_VDD 来控制, PT14_VDD=1 时, PT1.4 为开漏输出, 施密特触发器打开					
			控制信号		IO 功能描述			
			SPIEN=0, PTW0[2]=0		IO (默认值)			
			SPIEN=0, PTW0[2]=1		外部中断 0 输入			
			SPIEN=1, MSTEN=1		SPI 主模式串 时钟输出			
SPIEN=1, MSTEN=0		SPI 从模式串 时钟输入						

PT1.5/SS/INT1/PWM3	I/O	9	I/O; SPI 从模式片选输入口; 外部中断 1; PWM3 输出,IO 电压需要根据实际选择 1.8V/3.3V, PT15_VDD 来控制					<table border="1"> <thead> <tr> <th>SPI EN</th> <th>MST EN</th> <th>WIREM OD</th> <th>T3O UT</th> <th>PWM3O UT</th> <th>PWM3_ SEL</th> <th>PTW1 [2]</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>0</td> <td>X</td> <td>0</td> <td rowspan="2">IO (默认值)</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>0</td> <td>X</td> <td>1</td> <td rowspan="2">外部中断 1 输入</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>1</td> <td>0</td> <td>X</td> <td>蜂鸣器 3 输出</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>1</td> <td>0</td> <td>0</td> <td>X</td> <td>PWM3 输出</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>1</td> <td>1</td> <td>0</td> <td>X</td> <td>PWM3 取反输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 主模式片选输出口</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>0</td> <td>IO</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>1</td> <td>外部中断 1 输入</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 从模式片选输入口</td> </tr> </tbody> </table>		SPI EN	MST EN	WIREM OD	T3O UT	PWM3O UT	PWM3_ SEL	PTW1 [2]	IO 功能描述	0	X	X	0	0	X	0	IO (默认值)	0	X	X	X	X	1	0	0	X	X	0	0	X	1	外部中断 1 输入	0	X	X	X	X	1	1	0	X	X	0	1	0	X	蜂鸣器 3 输出	0	X	X	1	0	0	X	PWM3 输出	0	X	X	1	1	0	X	PWM3 取反输出	1	1	0	X	X	X	X	SPI 主模式片选输出口	1	1	1	X	X	X	0	IO	1	1	1	X	X	X	1	外部中断 1 输入	1	0	X	X	X	X	X	SPI 从模式片选输入口
			SPI EN	MST EN	WIREM OD	T3O UT	PWM3O UT	PWM3_ SEL	PTW1 [2]	IO 功能描述																																																																																													
			0	X	X	0	0	X	0	IO (默认值)																																																																																													
			0	X	X	X	X	1	0																																																																																														
			0	X	X	0	0	X	1	外部中断 1 输入																																																																																													
			0	X	X	X	X	1	1																																																																																														
			0	X	X	0	1	0	X	蜂鸣器 3 输出																																																																																													
			0	X	X	1	0	0	X	PWM3 输出																																																																																													
			0	X	X	1	1	0	X	PWM3 取反输出																																																																																													
			1	1	0	X	X	X	X	SPI 主模式片选输出口																																																																																													
			1	1	1	X	X	X	0	IO																																																																																													
			1	1	1	X	X	X	1	外部中断 1 输入																																																																																													
1	0	X	X	X	X	X	SPI 从模式片选输入口																																																																																																
PT2.1/INT0/BZ/PWM3/AD4	I/O	10	I/O ; 外部中断 0; 蜂鸣器输出; PWM3 输出; ADC 模拟输入通道 4					<table border="1"> <thead> <tr> <th>AIENB2[4]</th> <th>BZE N</th> <th>T3OU T</th> <th>PWM3OU T</th> <th>PWM3_ SE L</th> <th>PTW0[4]</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>X</td> <td>0</td> <td rowspan="2">IO (默认值)</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>X</td> <td>1</td> <td rowspan="2">外部中断 0 输入</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>ADC 模拟输入</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>X</td> <td>蜂鸣器 3 输出</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>X</td> <td>PWM3 输出</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>X</td> <td>PWM3 取反输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>蜂鸣器输出</td> </tr> </tbody> </table>		AIENB2[4]	BZE N	T3OU T	PWM3OU T	PWM3_ SE L	PTW0[4]	IO 功能描述	1	0	0	0	X	0	IO (默认值)	1	0	X	X	0	0	1	0	0	0	X	1	外部中断 0 输入	1	0	X	X	0	1	0	X	X	X	X	X	ADC 模拟输入	1	0	0	1	1	X	蜂鸣器 3 输出	1	0	1	0	1	X	PWM3 输出	1	0	1	1	1	X	PWM3 取反输出	1	1	X	X	X	X	蜂鸣器输出																										
			AIENB2[4]	BZE N	T3OU T	PWM3OU T	PWM3_ SE L	PTW0[4]	IO 功能描述																																																																																														
			1	0	0	0	X	0	IO (默认值)																																																																																														
			1	0	X	X	0	0																																																																																															
			1	0	0	0	X	1	外部中断 0 输入																																																																																														
			1	0	X	X	0	1																																																																																															
			0	X	X	X	X	X	ADC 模拟输入																																																																																														
			1	0	0	1	1	X	蜂鸣器 3 输出																																																																																														
			1	0	1	0	1	X	PWM3 输出																																																																																														
			1	0	1	1	1	X	PWM3 取反输出																																																																																														
1	1	X	X	X	X	蜂鸣器输出																																																																																																	
PT2.6/RX1/AD1/INT1/PWM1	I/O	11	I/O; 串口 1 输入; ADC 模拟输入通道 1; 外部中断 1; PWM1 输出					<table border="1"> <thead> <tr> <th>UART1 EN</th> <th>UART1_ S EL</th> <th>AIENB2 [1]</th> <th>T1O UT</th> <th>PWM1O UT</th> <th>PWM1_ S EL</th> <th>PTW1[7]</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>1</td> <td>0</td> <td>0</td> <td>X</td> <td>0</td> <td rowspan="2">IO (默认值)</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>X</td> <td>X</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>0</td> <td>0</td> <td>X</td> <td>1</td> <td rowspan="2">外部中断 1 输入</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>X</td> <td>X</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>X</td> <td>蜂鸣器 1 输出</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>X</td> <td>PWM1 输出</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>X</td> <td>PWM1 取反输出</td> </tr> <tr> <td>X</td> <td>X</td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>ADC 模拟输入</td> </tr> </tbody> </table>		UART1 EN	UART1_ S EL	AIENB2 [1]	T1O UT	PWM1O UT	PWM1_ S EL	PTW1[7]	IO 功能描述	0	X	1	0	0	X	0	IO (默认值)	0	X	1	X	X	0	0	0	X	1	0	0	X	1	外部中断 1 输入	0	X	1	X	X	0	1	0	X	1	0	1	1	X	蜂鸣器 1 输出	0	X	1	1	0	1	X	PWM1 输出	0	X	1	1	1	1	X	PWM1 取反输出	X	X	0	X	X	X	X	ADC 模拟输入																								
			UART1 EN	UART1_ S EL	AIENB2 [1]	T1O UT	PWM1O UT	PWM1_ S EL	PTW1[7]	IO 功能描述																																																																																													
			0	X	1	0	0	X	0	IO (默认值)																																																																																													
			0	X	1	X	X	0	0																																																																																														
			0	X	1	0	0	X	1	外部中断 1 输入																																																																																													
			0	X	1	X	X	0	1																																																																																														
			0	X	1	0	1	1	X	蜂鸣器 1 输出																																																																																													
			0	X	1	1	0	1	X	PWM1 输出																																																																																													
			0	X	1	1	1	1	X	PWM1 取反输出																																																																																													
X	X	0	X	X	X	X	ADC 模拟输入																																																																																																

			1	1	1	X	X	X	X	串口 1 输入
			1	0	1	0	0	X	0	IO
			1	0	1	X	X	0	0	
			1	0	1	0	0	X	1	外部中断 1 输入
			1	0	1	X	X	0	1	
PT2.7/TX1/AD0/INT0/PWM2	I/O	12	I/O: 串口 1 输出; ADC 模拟输入通道 0; 外部中断 0; PWM2 输出							
			UART1 EN	UART1_S EL	AIENB2 [0]	T2O UT	PWM2O UT	PWM2_S EL	PTW0[7]	IO 功能描述
			0	X	1	0	0	X	0	IO (默认值)
			0	X	1	X	X	0	0	外部中断 0 输入
			0	X	1	0	0	X	1	
			0	X	1	X	X	0	1	蜂鸣器 2 输出
			0	X	1	0	1	1	X	PWM2 输出
			0	X	1	1	0	1	X	PWM2 取反输出
			X	X	0	X	X	X	X	ADC 模拟输入
			1	1	1	X	X	X	X	串口 1 输出
			1	0	1	0	0	X	0	IO
			1	0	1	X	X	0	0	
			1	0	1	0	0	X	1	外部中断 0 输入
			1	0	1	X	X	0	1	
PT2.5/PCL/ INT1/T2	I/O	13	I/O: 烧录的时钟口; 外部中断 1; 定时器 2 时钟输入							
			PTW1[6]	I/O 功能描述						
			0	I/O 和烧录数据口 (默认值)						
			1	外部中断 1 输入						
PT2.4/PDA/ INT0/T1	I/O	14	I/O: 烧录的数据口; 外部中断 0; 定时器 1 时钟输入							
			PTW0[6]	I/O 功能描述						
			0	I/O 和烧录数据口 (默认值)						
			1	外部中断 0 输入						
GND	P	15	模拟地, 外部连接 AGND							
VDD	P	16	模拟电源							

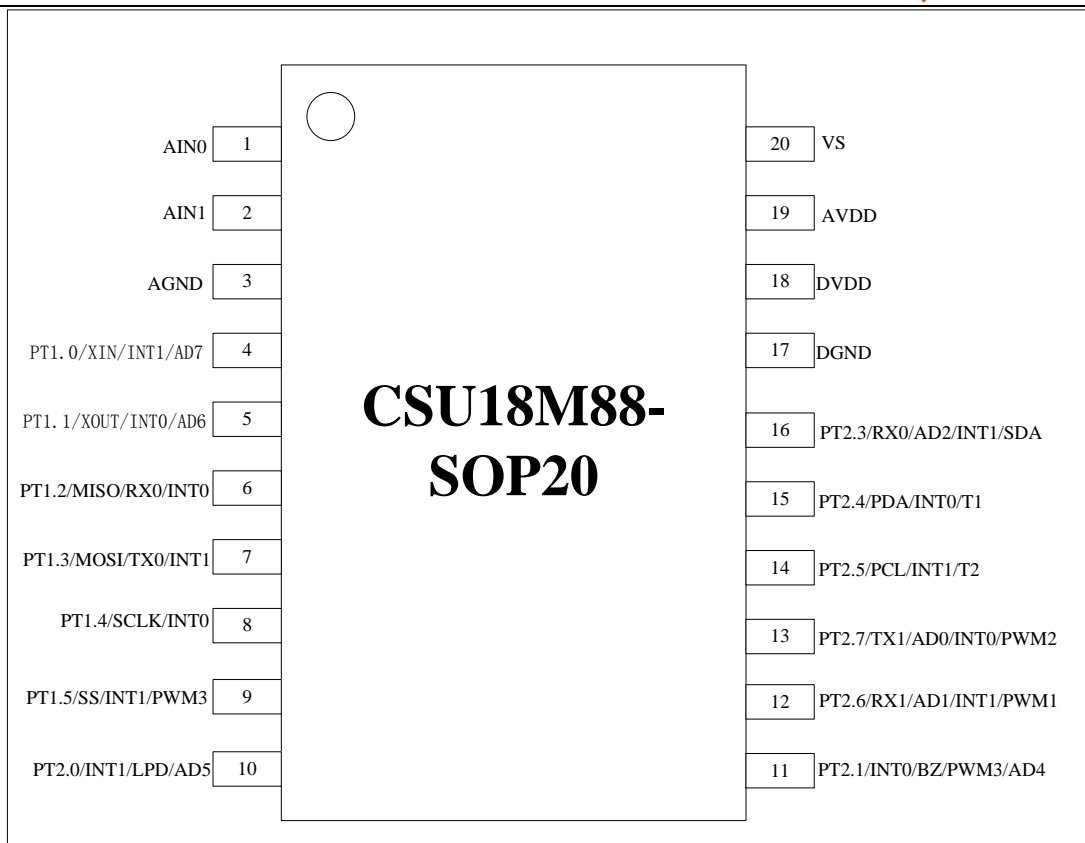


图 2 CSU18M88 SOP20 封装 pin 脚图

表 2 引脚说明表

管脚名称	输入 / 输出	管脚序号	描述	
AIN0~1	I	1~2	24 Bits ADC 模拟差分输入端，寄存器位 SINL[1:0]控制差分输入： =00：输入端连接到 AIN0 和 AIN1:AIN0 为 Vin + 、AIN1 为 Vin - =01：内短 =10：24 Bits ADC 输入端连接到温度传感器	
AGND	P	3	模拟地，外部连接 AGND	
PT1.0/XIN/INT1/AD7	I/O	4	I/O: 32768Hz 晶振输入；外部中断 1；ADC 模拟输入通道 7	
			控制信号	IO 功能描述
			AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0	IO(默认值)
			AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1	外部中断 1 输入
			AIENB3=0	32768Hz 时钟输入
AIENB3=1 且 AIENB2[7]=0	ADC 模拟输入			
PT1.1/XOUT/INT0/AD6	I/O	5	I/O: 32768Hz 晶振输出；外部中断 0；ADC 模拟输入通道 6	
			控制信号	IO 功能描述
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=0	IO(默认值)
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=1	外部中断 0 输入
			AIENB3=0	32768Hz 晶振输出
AIENB3=1 且 AIENB2[6]=0	ADC 模拟输入			

PT1.2/MISO/RX0/INT0	I/O	6	I/O: SPI 主模式数据输入/出口或 SPI 从模式数据输入/出口; 串口 0 输入; 外部中断 0;				<table border="1"> <thead> <tr> <th>SPIEN</th> <th>MSTEN</th> <th>UART0EN</th> <th>UART0_SEL</th> <th>PTW0[1]</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>0</td> <td>X</td> <td>0</td> <td rowspan="2">IO (默认值)</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>X</td> <td>0</td> <td>X</td> <td>1</td> <td rowspan="2">外部中断 0 输入</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 主模式数据输入/出口</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 从模式数据输入/出口</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>0</td> <td>X</td> <td>串口 0 输入</td> </tr> </tbody> </table>		SPIEN	MSTEN	UART0EN	UART0_SEL	PTW0[1]	IO 功能描述	0	X	0	X	0	IO (默认值)	0	X	1	1	0	0	X	0	X	1	外部中断 0 输入	0	X	1	1	1	1	1	X	X	X	SPI 主模式数据输入/出口	1	0	X	X	X	SPI 从模式数据输入/出口	0	X	1	0	X	串口 0 输入																																																
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW0[1]	IO 功能描述																																																																																														
			0	X	0	X	0	IO (默认值)																																																																																														
			0	X	1	1	0																																																																																															
			0	X	0	X	1	外部中断 0 输入																																																																																														
			0	X	1	1	1																																																																																															
			1	1	X	X	X	SPI 主模式数据输入/出口																																																																																														
			1	0	X	X	X	SPI 从模式数据输入/出口																																																																																														
0	X	1	0	X	串口 0 输入																																																																																																	
PT1.3/MOSI/TX0/INT1	I/O	7	I/O: SPI 主模式数据输入/出口或 SPI 从模式数据输入/出口; 串口 0 输出; 外部中断 1;				<table border="1"> <thead> <tr> <th>SPIEN</th> <th>MSTEN</th> <th>UART0EN</th> <th>UART0_SEL</th> <th>PTW1[1]</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>0</td> <td>X</td> <td>0</td> <td rowspan="2">IO (默认值)</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>X</td> <td>0</td> <td>X</td> <td>1</td> <td rowspan="2">外部中断 1 输入</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 主模式数据输入/出口</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 从模式数据输入/出口</td> </tr> <tr> <td>0</td> <td>X</td> <td>1</td> <td>0</td> <td>X</td> <td>串口 0 输出</td> </tr> </tbody> </table>		SPIEN	MSTEN	UART0EN	UART0_SEL	PTW1[1]	IO 功能描述	0	X	0	X	0	IO (默认值)	0	X	1	1	0	0	X	0	X	1	外部中断 1 输入	0	X	1	1	1	1	1	X	X	X	SPI 主模式数据输入/出口	1	0	X	X	X	SPI 从模式数据输入/出口	0	X	1	0	X	串口 0 输出																																																
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW1[1]	IO 功能描述																																																																																														
			0	X	0	X	0	IO (默认值)																																																																																														
			0	X	1	1	0																																																																																															
			0	X	0	X	1	外部中断 1 输入																																																																																														
			0	X	1	1	1																																																																																															
			1	1	X	X	X	SPI 主模式数据输入/出口																																																																																														
			1	0	X	X	X	SPI 从模式数据输入/出口																																																																																														
0	X	1	0	X	串口 0 输出																																																																																																	
PT1.4 /SCLK/INT0	I/O	8	I/O: SPI 模式串口时钟输入/输出; 外部中断 0; IO 电压需要根据实际选择 1.8V/3.3V 通信, 通过 PT14_VDD 来控制, PT14_VDD=1 时, PT1.4 为开漏输出, 施密特触发器打开				<table border="1"> <thead> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>SPIEN=0, PTW0[2]=0</td> <td>IO (默认值)</td> </tr> <tr> <td>SPIEN=0, PTW0[2]=1</td> <td>外部中断 0 输入</td> </tr> <tr> <td>SPIEN=1, MSTEN=1</td> <td>SPI 主模式串 时钟输入/出口</td> </tr> <tr> <td>SPIEN=1, MSTEN=0</td> <td>SPI 从模式串 时钟输入/出口</td> </tr> </tbody> </table>		控制信号	IO 功能描述	SPIEN=0, PTW0[2]=0	IO (默认值)	SPIEN=0, PTW0[2]=1	外部中断 0 输入	SPIEN=1, MSTEN=1	SPI 主模式串 时钟输入/出口	SPIEN=1, MSTEN=0	SPI 从模式串 时钟输入/出口																																																																																				
			控制信号	IO 功能描述																																																																																																		
			SPIEN=0, PTW0[2]=0	IO (默认值)																																																																																																		
			SPIEN=0, PTW0[2]=1	外部中断 0 输入																																																																																																		
			SPIEN=1, MSTEN=1	SPI 主模式串 时钟输入/出口																																																																																																		
SPIEN=1, MSTEN=0	SPI 从模式串 时钟输入/出口																																																																																																					
PT1.5/SS/INT1/PWM3	I/O	9	I/O: SPI 从模式片选输入/出口; 外部中断 1; PWM3 输出, IO 电压需要根据实际选择 1.8V/3.3V, PT15_VDD 来控制				<table border="1"> <thead> <tr> <th>SPI EN</th> <th>MST EN</th> <th>WIREM OD</th> <th>T3O UT</th> <th>PWM3O UT</th> <th>PWM3_ SEL</th> <th>PTW1 [2]</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>0</td> <td>X</td> <td>0</td> <td rowspan="2">IO (默认值)</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>0</td> <td>X</td> <td>1</td> <td rowspan="2">外部中断 1 输入</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>0</td> <td>1</td> <td>0</td> <td>X</td> <td>蜂鸣器 3 输出</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>1</td> <td>0</td> <td>0</td> <td>X</td> <td>PWM3 输出</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>1</td> <td>1</td> <td>0</td> <td>X</td> <td>PWM3 取反输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 主模式片选输入/出口</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>0</td> <td>IO</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>X</td> <td>X</td> <td>X</td> <td>1</td> <td>外部中断 1 输入</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> <td>SPI 从模式片选输入/出口</td> </tr> </tbody> </table>		SPI EN	MST EN	WIREM OD	T3O UT	PWM3O UT	PWM3_ SEL	PTW1 [2]	IO 功能描述	0	X	X	0	0	X	0	IO (默认值)	0	X	X	X	X	1	0	0	X	X	0	0	X	1	外部中断 1 输入	0	X	X	X	X	1	1	0	X	X	0	1	0	X	蜂鸣器 3 输出	0	X	X	1	0	0	X	PWM3 输出	0	X	X	1	1	0	X	PWM3 取反输出	1	1	0	X	X	X	X	SPI 主模式片选输入/出口	1	1	1	X	X	X	0	IO	1	1	1	X	X	X	1	外部中断 1 输入	1	0	X	X	X	X	X	SPI 从模式片选输入/出口
			SPI EN	MST EN	WIREM OD	T3O UT	PWM3O UT	PWM3_ SEL	PTW1 [2]	IO 功能描述																																																																																												
			0	X	X	0	0	X	0	IO (默认值)																																																																																												
			0	X	X	X	X	1	0																																																																																													
			0	X	X	0	0	X	1	外部中断 1 输入																																																																																												
			0	X	X	X	X	1	1																																																																																													
			0	X	X	0	1	0	X	蜂鸣器 3 输出																																																																																												
			0	X	X	1	0	0	X	PWM3 输出																																																																																												
			0	X	X	1	1	0	X	PWM3 取反输出																																																																																												
			1	1	0	X	X	X	X	SPI 主模式片选输入/出口																																																																																												
			1	1	1	X	X	X	0	IO																																																																																												
1	1	1	X	X	X	1	外部中断 1 输入																																																																																															
1	0	X	X	X	X	X	SPI 从模式片选输入/出口																																																																																															

PT2.0/INT1/LPD/AD5	I/O	10	I/O: 外部中断 1; 模拟低电压检测输入; ADC 模拟输入通道 5							
			AIENB1	AIENB2[5]	PTW1[4]	I/O 功能描述				
			1	1	0	I/O				
			1	1	1	外部中断 1 输入				
			X	0	X	ADC 模拟输入				
			0	X	X	模拟低电压检测输入(默认值)				
PT2.1/INT0/BZ/PWM3/AD4	I/O	11	I/O : 外部中断 0; 蜂鸣器输出; PWM3 输出; ADC 模拟输入通道 4							
			AIENB2[4]	BZEN	T3OUT	PWM3OUT	PWM3_SEL	PTW0[4]	IO 功能描述	
			1	0	0	0	X	0	IO (默认值)	
			1	0	X	X	0	0		
			1	0	0	0	X	1	外部中断 0 输入	
			1	0	X	X	0	1		
			0	X	X	X	X	X	ADC 模拟输入	
			1	0	0	1	1	X	蜂鸣器 3 输出	
			1	0	1	0	1	X	PWM3 输出	
			1	0	1	1	1	X	PWM3 取反输出	
1	1	X	X	X	X	蜂鸣器输出				
PT2.6/RX1/AD1/INT1/PWM1	I/O	12	I/O: 串口 1 输入; ADC 模拟输入通道 1; 外部中断 1; PWM1 输出							
			UART1_EN	UART1_SEL	AIENB2[1]	TIO_UT	PWM1OUT	PWM1_SEL	PTW1[7]	IO 功能描述
			0	X	1	0	0	X	0	IO (默认值)
			0	X	1	X	X	0	0	
			0	X	1	0	0	X	1	外部中断 1 输入
			0	X	1	X	X	0	1	
			0	X	1	0	1	1	X	蜂鸣器 1 输出
			0	X	1	1	0	1	X	PWM1 输出
			0	X	1	1	1	1	X	PWM1 取反输出
			X	X	0	X	X	X	X	ADC 模拟输入
			1	1	1	X	X	X	X	串口 1 输入
			1	0	1	0	0	X	0	IO
			1	0	1	X	X	0	0	
1	0	1	0	0	X	1	外部中断 1 输入			
1	0	1	X	X	0	1				

PT2.7/TX1/AD0/INT0/PWM2	I/O	13	I/O; 串口 1 输出; ADC 模拟输入通道 0; 外部中断 0; PWM2 输出							
			UART1_EN	UART1_SEL	AIENB2[0]	T2OUT	PWM2OUT	PWM2_SEL	PTW0[7]	IO 功能描述
			0	X	1	0	0	X	0	IO (默认值)
			0	X	1	X	X	0	0	外部中断 0 输入
			0	X	1	0	0	X	1	
			0	X	1	X	X	0	1	蜂鸣器 2 输出
			0	X	1	0	1	1	X	PWM2 输出
			0	X	1	1	0	1	X	PWM2 取反输出
			0	X	1	1	1	1	X	ADC 模拟输入
			X	X	0	X	X	X	X	串口 1 输出
			1	1	1	X	X	X	X	IO
			1	0	1	0	0	X	0	
			1	0	1	X	X	0	0	外部中断 0 输入
1	0	1	0	0	X	1				
1	0	1	X	X	0	1				
PT2.5/PCL/INT1/T2	I/O	14	I/O; 烧录的时钟口; 外部中断 1; 定时器 2 时钟输入							
			PTW1[6]	I/O 功能描述						
			0	I/O 和烧录数据口 (默认值)						
			1	外部中断 1 输入						
PT2.4/PDA/INT0/T1	I/O	15	I/O; 烧录的数据口; 外部中断 0; 定时器 1 时钟输入							
			PTW0[6]	I/O 功能描述						
			0	I/O 和烧录数据口 (默认值)						
			1	外部中断 0 输入						
PT2.3/RX0/AD2/INT1/SDA	I/O	31	I/O; 串口 0 输入; ADC 模拟输入通道 2; 外部中断 1;							
			UART0EN	UART0_SEL	AIENB2[2]	PTW1[5]	I2C_EN	IO 功能描述		
			0	X	1	0	0	IO (默认值)		
			0	X	1	1	0	外部中断 1 输入		
			X	X	0	X	X	ADC 模拟输入		
			1	1	1	X	0	串口 0 输入		
			1	0	1	0	0	IO		
1	0	1	1	0	外部中断 1 输入					
X	X	1	X	1	I2C 数据 SDA 脚					
DGND	P	17	数字地							
DVDD	P	18	数字电源							
AVDD	P	19	模拟电源							
VS	I	20	参考输入							

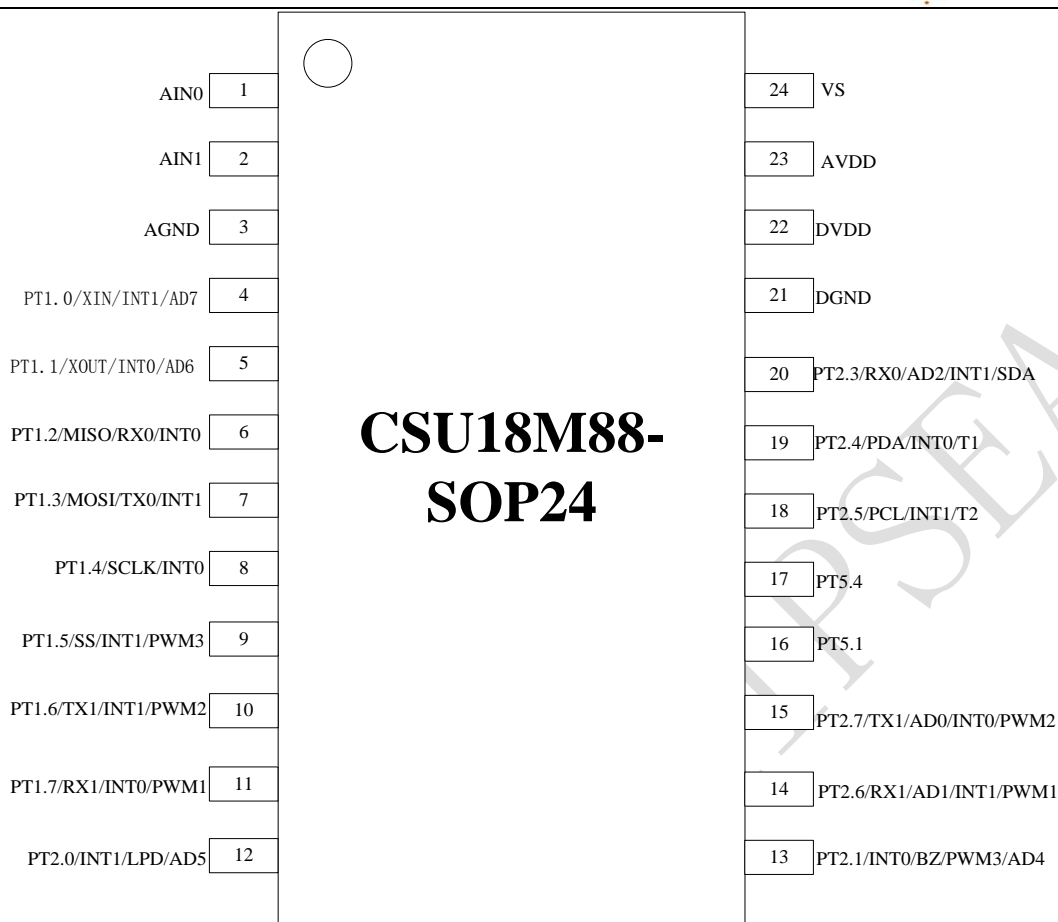


图 3 CSU18M88 SOP24 封装 pin 脚图

表 3 引脚说明表

管脚名称	输入 / 输出	管脚序号	描述										
AIN0~1	I	1~2	24 Bits ADC 模拟差分输入端，寄存器位 SINL[1:0] 控制差分输入： =00：输入端连接到 AIN0 和 AIN1；AIN0 为 Vin+、AIN1 为 Vin- =01：内短 =10：24 Bits ADC 输入端连接到温度传感器										
AGND	P	3	模拟地，外部连接 AGND										
PT1.0/XIN/INT1/AD7	I/O	4	I/O；32768Hz 晶振输入；外部中断 1；ADC 模拟输入通道 7 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0</td> <td>IO(默认值)</td> </tr> <tr> <td>AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1</td> <td>外部中断 1 输入</td> </tr> <tr> <td>AIENB3=0</td> <td>32768Hz 时钟输入</td> </tr> <tr> <td>AIENB3=1 且 AIENB2[7]=0</td> <td>ADC 模拟输入</td> </tr> </tbody> </table>	控制信号	IO 功能描述	AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0	IO(默认值)	AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1	外部中断 1 输入	AIENB3=0	32768Hz 时钟输入	AIENB3=1 且 AIENB2[7]=0	ADC 模拟输入
控制信号	IO 功能描述												
AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0	IO(默认值)												
AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1	外部中断 1 输入												
AIENB3=0	32768Hz 时钟输入												
AIENB3=1 且 AIENB2[7]=0	ADC 模拟输入												

PT1.1/XOUT/INT0/A D6	I/O	5	I/O; 32768Hz 晶振输出; 外部中断 0; ADC 模拟输入通道 6							
			控制信号			IO 功能描述				
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=0			IO(默认值)				
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=1			外部中断 0 输入				
			AIENB3=0			32768Hz 晶振输出				
AIENB3=1 且 AIENB2[6]=0			ADC 模拟输入							
PT1.2/MISO/RX0/INT0	I/O	6	I/O; SPI 主模式数据输入或 SPI 从模式数据输出; 串口 0 输入; 外部中断 0;							
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW0[1]	IO 功能描述		
			0	X	0	X	0	IO (默认值)		
			0	X	1	1	0			
			0	X	0	X	1	外部中断 0 输入		
			0	X	1	1	1			
			1	1	X	X	X	SPI 主模式数据输入		
			1	0	X	X	X	SPI 从模式数据输出		
0	X	1	0	X	串口 0 输入					
PT1.3/MOSI/TX0/INT1	I/O	7	I/O; SPI 主模式数据输出或 SPI 从模式数据输入; 串口 0 输出; 外部中断 1;							
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW1[1]	IO 功能描述		
			0	X	0	X	0	IO (默认值)		
			0	X	1	1	0			
			0	X	0	X	1	外部中断 1 输入		
			0	X	1	1	1			
			1	1	X	X	X	SPI 主模式数据输出		
			1	0	X	X	X	SPI 从模式数据输入		
0	X	1	0	X	串口 0 输出					
PT1.4/SCLK/INT0	I/O	8	I/O; SPI 模式串口时钟输入/输出; 外部中断 0; ,IO 电压需要根据实际选择 1.8V/3.3V 通信, 通过 PT14_VDD 来控制, PT14_VDD=1 时, PT1.4 为开漏输出, 施密特触发器打开							
			控制信号			IO 功能描述				
			SPIEN=0, PTW0[2]=0			IO (默认值)				
			SPIEN=0, PTW0[2]=1			外部中断 0 输入				
			SPIEN=1, MSTEN=1			SPI 主模式串 时钟输出				
SPIEN=1, MSTEN=0			SPI 从模式串 时钟输入							
PT1.5/SS/INT1/PWM3	I/O	9	I/O; SPI 从模式片选输入; 外部中断 1; PWM3 输出,IO 电压需要根据实际选择 1.8V/3.3V, PT15_VDD 来控制							
			SPI EN	MST EN	WIREFM OD	T3O UT	PWM3O UT	PWM3 SEL	PTW1 [2]	IO 功能描述
			0	X	X	0	0	X	0	IO (默认值)
			0	X	X	X	X	1	0	
			0	X	X	0	0	X	1	外部中断 1 输入
			0	X	X	X	X	1	1	
			0	X	X	0	1	0	X	蜂鸣器 3 输出
			0	X	X	1	0	0	X	PWM3 输出
			0	X	X	1	1	0	X	PWM3 取反输出
			1	1	0	X	X	X	X	SPI 主模式片选 输出
			1	1	1	X	X	X	0	IO
			1	1	1	X	X	X	1	外部中断 1 输入
1	0	X	X	X	X	X	SPI 从模式片选 输入			

PT1.6/TX1/INT1/PWM2	I/O	10	I/O: 串口 1 输出; 外部中断 1; PWM2 输出						
			UART1EN	UART1_SEL	T2OUT	PWM2OUT	PWM2_SEL	PTW1[3]	IO 功能描述
			0	X	0	0	X	0	IO (默认值)
			0	X	X	X	1	0	
			0	X	0	0	X	1	外部中断 1 输入
			0	X	X	X	1	1	
			0	X	0	1	0	X	蜂鸣器 2 输出
			0	X	1	0	0	X	PWM2 输出
			0	X	1	1	0	X	PWM2 取反输出
			1	0	X	X	X	X	串口 1 输出
1	1	X	X	X	0	IO			
1	1	X	X	X	1	外部中断 1 输入			
PT1.7/RX1/INT0/PWM1	I/O	11	I/O: 串口 1 输入; 外部中断 0; PWM1 输出						
			UART1EN	UART1_SEL	T1OUT	PWM1OUT	PWM1_SEL	PTW0[3]	IO 功能描述
			0	X	0	0	X	0	IO (默认值)
			0	X	X	X	1	0	
			0	X	0	0	X	1	外部中断 0 输入
			0	X	X	X	1	1	
			0	X	0	1	0	X	蜂鸣器 1 输出
			0	X	1	0	0	X	PWM1 输出
			0	X	1	1	0	X	PWM1 取反输出
			1	0	X	X	X	X	串口 1 输入
1	1	X	X	X	0	IO			
1	1	X	X	X	1	外部中断 0 输入			
PT2.0/INT1/LPD/AD5	I/O	12	I/O: 外部中断 1; 模拟低电压检测输入; ADC 模拟输入通道 5						
			AIENB1	AIENB2[5]	PTW1[4]	I/O 功能描述			
			1	1	0	I/O			
			1	1	1	外部中断 1 输入			
X	0	X	ADC 模拟输入						
0	X	X	模拟低电压检测输入(默认值)						

PT2.1/INT0/BZ/PWM3/AD4	I/O	13	I/O ; 外部中断 0; 蜂鸣器输出; PWM3 输出; ADC 模拟输入通道 4						IO 功能描述	
			AIENB2[4]	BZEN	T3OUT	PWM3OUT	PWM3_SEL	PTW0[4]	IO (默认值)	
			1	0	0	0	X	0		
			1	0	X	X	0	0	外部中断 0 输入	
			1	0	0	0	X	1		
			1	0	X	X	0	1	ADC 模拟输入	
			0	X	X	X	X	X		
			1	0	0	1	1	X	蜂鸣器 3 输出	
			1	0	1	0	1	X	PWM3 输出	
			1	0	1	1	1	X	PWM3 取反输出	
1	1	X	X	X	X	蜂鸣器输出				
PT2.6/RX1/AD1/INT1/PWM1	I/O	14	I/O; 串口 1 输入; ADC 模拟输入通道 1; 外部中断 1; PWM1 输出						IO 功能描述	
			UART1EN	UART1_SEL	AIENB2[1]	T1OUT	PWM1OUT	PWM1_SEL	PTW1[7]	IO (默认值)
			0	X	1	0	0	X	0	
			0	X	1	X	X	0	0	外部中断 1 输入
			0	X	1	0	0	X	1	
			0	X	1	X	X	0	1	蜂鸣器 1 输出
			0	X	1	0	1	1	X	PWM1 输出
			0	X	1	1	0	1	X	PWM1 取反输出
			0	X	1	1	1	1	X	ADC 模拟输入
			X	X	0	X	X	X	X	串口 1 输入
			1	1	1	X	X	X	X	IO
			1	0	1	0	0	X	0	
			1	0	1	X	X	0	0	外部中断 1 输入
			1	0	1	0	0	X	1	
1	0	1	X	X	0	1				

PT2.7/TX1/AD0/INT0/PWM2	I/O	15	I/O: 串口 1 输出; ADC 模拟输入通道 0; 外部中断 0; PWM2 输出						I/O 功能描述	
			UART1EN	UART1_SEL	AIENB2[0]	T2OUT	PWM2OUT	PWM2_SEL	PTW0[7]	IO (默认值)
			0	X	1	0	0	X	0	外部中断 0 输入
			0	X	1	X	X	0	0	
			0	X	1	0	0	X	1	蜂鸣器 2 输出
			0	X	1	X	X	0	1	
			0	X	1	0	1	1	X	PWM2 输出
			0	X	1	1	1	1	X	PWM2 取反输出
			X	X	0	X	X	X	X	ADC 模拟输入
			1	1	1	X	X	X	X	串口 1 输出
			1	0	1	0	0	X	0	IO
			1	0	1	X	X	0	0	
			1	0	1	0	0	X	1	
1	0	1	X	X	0	1	外部中断 0 输入			
PT5.1	I/O	16	I/O							
PT5.4	I/O	17	I/O							
PT2.5/PCL/INT1/T2	I/O	18	I/O: 烧录的时钟口; 外部中断 1; 定时器 2 时钟输入							
			PTW1[6]	I/O 功能描述						
			0	I/O 和烧录数据口 (默认值)						
1	外部中断 1 输入									
PT2.4/PDA/INT0/T1	I/O	19	I/O: 烧录的数据口; 外部中断 0; 定时器 1 时钟输入							
			PTW0[6]	I/O 功能描述						
			0	I/O 和烧录数据口 (默认值)						
1	外部中断 0 输入									
PT2.3/RX0/AD2/INT1/SDA	I/O	20	I/O: 串口 0 输入; ADC 模拟输入通道 2; 外部中断 1;							
			UART0EN	UART0_SEL	AIENB2[2]	PTW1[5]	I2C_EN	I/O 功能描述		
			0	X	1	0	0	IO (默认值)		
			0	X	1	1	0	外部中断 1 输入		
			X	X	0	X	X	ADC 模拟输入		
			1	1	1	X	0	串口 0 输入		
			1	0	1	0	0	IO		
1	0	1	1	0	外部中断 1 输入					
X	X	1	X	1	I2C 数据 SDA 脚					
DGND	P	21	数字地							
DVDD	P	22	数字电源							
AVDD	P	23	模拟电源							
VS	I	24	参考输入							

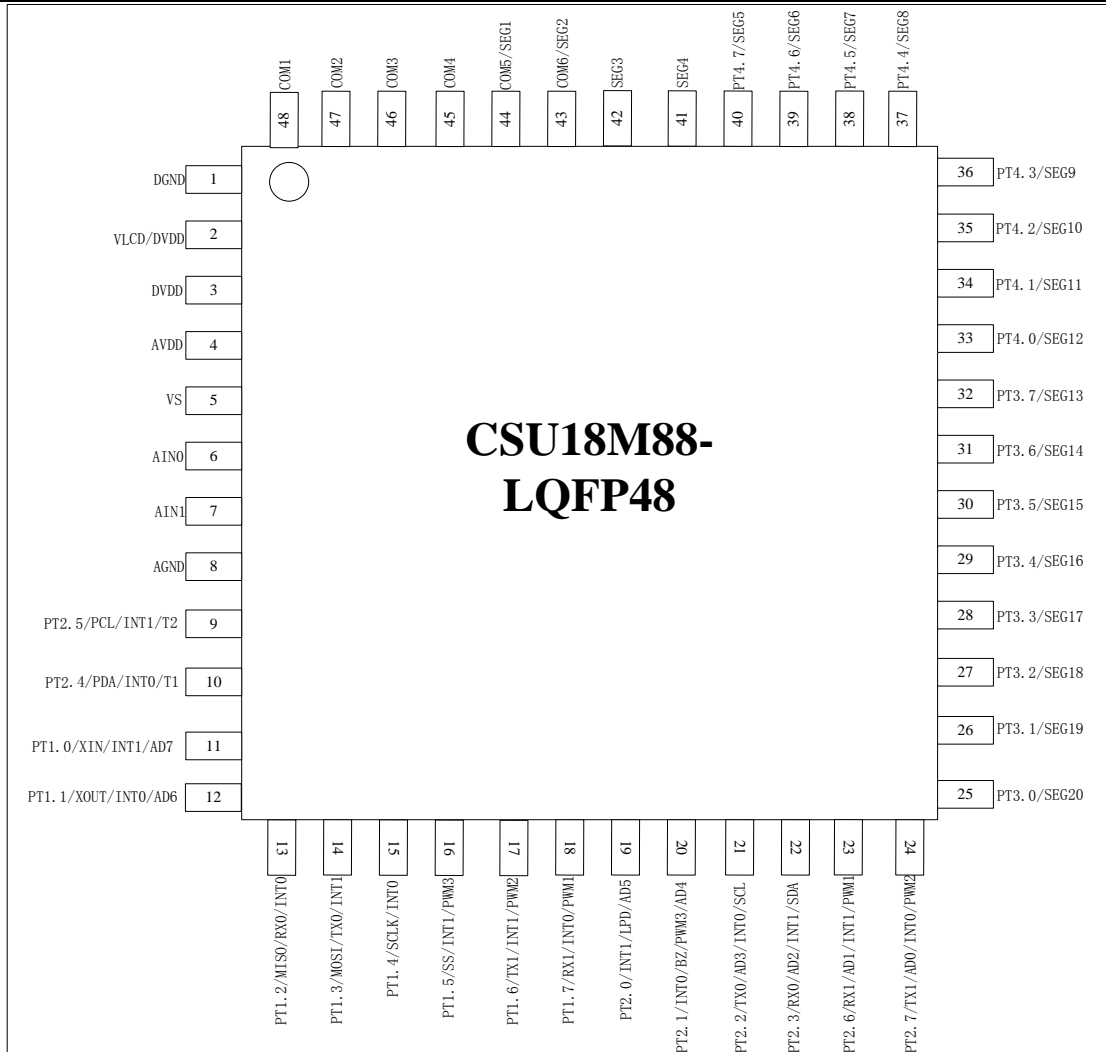


图 4 CSU18M88 LQFP48 封装 pin 脚图

表 4 引脚说明表

管脚名称	输入 / 输出	管脚序号	描述
DGND	P	1	数字地
VLCD/DVDD	P	2	当使用外部电源作为 LCD 模块电源时，外部接 LCD 电源，接 1uF 滤波电容； 当使用内部电荷泵作 LCD 模块电源时，需外接 1uF 电容； 当 PT3 和 PT4 需要作为通用 IO 时，引脚接 DVDD；
DVDD	P	3	数字电源
AVDD	P	4	模拟电源
VS	I	5	参考输入

AIN0~1	I	6~7	24 Bits ADC 模拟差分输入端，寄存器位 SINL[1:0]控制差分输入： =00：输入端连接到 AIN0 和 AIN1:AIN0 为 Vin+、AIN1 为 Vin- =01：内短 =10：24 Bits ADC 输入端连接到温度传感器	
AGND	P	8	模拟地，外部连接 AGND	
PT2.5/PCL/INT1/T2	I/O	9	I/O；烧录的时钟口；外部中断 1；定时器 2 时钟输入	
			PTW1[6]	I/O 功能描述
			0	I/O 和烧录数据口（默认值）
1	外部中断 1 输入			
PT2.4/PDA/INT0/T1	I/O	10	I/O；烧录的数据口；外部中断 0；定时器 1 时钟输入	
			PTW0[6]	I/O 功能描述
			0	I/O 和烧录数据口（默认值）
1	外部中断 0 输入			
PT1.0/XIN/INT1/AD7	I/O	19	I/O；32768Hz 晶振输入；外部中断 1；ADC 模拟输入通道 7	
			控制信号	IO 功能描述
			AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=0	IO(默认值)
			AIENB2[7]=1 且 AIENB3=1 且 PTW1[0]=1	外部中断 1 输入
			AIENB3=0	32768Hz 时钟输入
AIENB3=1 且 AIENB2[7]=0	ADC 模拟输入			
PT1.1/XOUT/INT0/AD6	I/O	20	I/O；32768Hz 晶振输出；外部中断 0；ADC 模拟输入通道 6	
			控制信号	IO 功能描述
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=0	IO(默认值)
			AIENB2[6]=1 且 AIENB3=1 且 PTW0[0]=1	外部中断 0 输入
			AIENB3=0	32768Hz 晶振输出
AIENB3=1 且 AIENB2[6]=0	ADC 模拟输入			

PT1.2/MISO/RX0/INT0	I/O	13	I/O: SPI 主模式数据输入/出口或 SPI 从模式数据输入/出口; 串口 0 输入; 外部中断 0;					
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW0[1]	IO 功能描述
			0	X	0	X	0	IO (默认值)
			0	X	1	1	0	
			0	X	0	X	1	外部中断 0 输入
			0	X	1	1	1	
			1	1	X	X	X	SPI 主模式数据输入
			1	0	X	X	X	SPI 从模式数据输出
0	X	1	0	X	串口 0 输入			
PT1.3/MOSI/TX0/INT1	I/O	14	I/O: SPI 主模式数据输出/出口或 SPI 从模式数据输入/出口; 串口 0 输出; 外部中断 1;					
			SPIEN	MSTEN	UART0EN	UART0_SEL	PTW1[1]	IO 功能描述
			0	X	0	X	0	IO (默认值)
			0	X	1	1	0	
			0	X	0	X	1	外部中断 1 输入
			0	X	1	1	1	
			1	1	X	X	X	SPI 主模式数据输出
			1	0	X	X	X	SPI 从模式数据输入
0	X	1	0	X	串口 0 输出			
PT1.4 /SCLK/INT0	I/O	15	I/O: SPI 模式串口时钟输入/输出; 外部中断 0; ,IO 电压需要根据实际选择 1.8V/3.3V 通信, 通过 PT14_VDD 来控制, PT14_VDD=1 时, PT1.4 为开漏输出, 施密特触发器打开					
			控制信号		IO 功能描述			
			SPIEN=0, PTW0[2]=0		IO (默认值)			
			SPIEN=0, PTW0[2]=1		外部中断 0 输入			
			SPIEN=1, MSTEN=1		SPI 主模式串 时钟输出			
SPIEN=1, MSTEN=0		SPI 从模式串 时钟输入						

PT1.5/SS/INT1/PWM3	I/O	16	I/O; SPI 从模式片选输入口; 外部中断 1; PWM3 输出,IO 电压需要根据实际选择 1.8V/3.3V, PT15_VDD 来控制					IO 功能描述			
			SPI EN	MST EN	WIREM OD	T3O UT	PWM3O UT		PWM3_ SEL	PTW1 [2]	
			0	X	X	0	0		X	0	IO (默认值)
			0	X	X	X	X		1	0	
			0	X	X	0	0		X	1	外部中断 1 输入
			0	X	X	X	X		1	1	
			0	X	X	0	1		0	X	蜂鸣器 3 输出
			0	X	X	1	0		0	X	PWM3 输出
			0	X	X	1	1		0	X	PWM3 取反输出
			1	1	0	X	X		X	X	SPI 主模式片选 输出口
			1	1	1	X	X		X	0	IO
			1	1	1	X	X		X	1	外部中断 1 输入
1	0	X	X	X	X	X	SPI 从模式片选 输入口				
PT1.6/TX1/INT1/PWM2	I/O	17	I/O; 串口 1 输出; 外部中断 1; PWM2 输出					IO 功能描述			
			UARTIE N	UART1_SE L	T2O UT	PWM2O UT	PWM2_SE L		PTW1[3]		
			0	X	0	0	X		0	IO (默认值)	
			0	X	X	X	1		0		
			0	X	0	0	X		1	外部 中断 1 输入	
			0	X	X	X	1		1		
			0	X	0	1	0		0	X	蜂 鸣 器 2 输 出
			0	X	1	0	0		0	X	PWM 2 输出
			0	X	1	1	0		0	X	PWM 2 取反 输出
			1	0	X	X	X		X	X	串口 1 输出
			1	1	X	X	X		X	0	IO
			1	1	X	X	X		X	1	外部 中断 1 输入

PT1.7/RX1/INT0/PWM1	I/O	18	I/O: 串口 1 输入; 外部中断 0; PWM1 输出					IO 功能描述		
			UART1EN	UART1_SEL	T1IOUT	PWM1IOUT	PWM1_SEL		PTW0[3]	
			0	X	0	0	X		0	IO (默认值)
			0	X	X	X	1		0	
			0	X	0	0	X		1	外部中断 0 输入
			0	X	X	X	1		1	
			0	X	0	1	0		X	蜂鸣器 1 输出
			0	X	1	0	0		X	PWM1 输出
			0	X	1	1	0		X	PWM1 取反输出
			1	0	X	X	X		X	串口 1 输入
			1	1	X	X	X		0	IO
1	1	X	X	X	1	外部中断 0 输入				
PT2.0/INT1/LPD/AD5	I/O	19	I/O: 外部中断 1; 模拟低电压检测输入; ADC 模拟输入通道 5							
			AIENB1	AIENB2[5]	PTW1[4]	I/O 功能描述				
			1	1	0	I/O				
			1	1	1	外部中断 1 输入				
			X	0	X	ADC 模拟输入				
0	X	X	模拟低电压检测输入(默认值)							

PT2.1/INT0/BZ/PWM3/AD4	I/O	20	I/O ; 外部中断 0; 蜂鸣器输出; PWM3 输出; ADC 模拟输入通道 4					IO 功能描述	
			AIENB2[4]	BZEN	T3OUT	PWM3OUT	PWM3_SEL	PTW0[4]	IO (默认值)
			1	0	0	0	X	0	
			1	0	X	X	0	0	外部中断 0 输入
			1	0	0	0	X	1	
			1	0	X	X	0	1	ADC 模拟输入
			0	X	X	X	X	X	
			1	0	0	1	1	X	蜂鸣器 3 输出
			1	0	1	0	1	X	PWM3 输出
			1	0	1	1	1	X	PWM3 取反输出
1	1	X	X	X	X	蜂鸣器输出			
PT2.2/TX0/AD3/INT0/SCL	I/O	21	I/O; 串口 0 输出; ADC 模拟输入通道 3; 外部中断 0;					IO 功能描述	
			UART0EN	UART0_SEL	AIENB2[3]	PTW0[5]	I2C_EN	IO (默认值)	
			0	X	1	0	0	外部中断 0 输入	
			0	X	1	1	0	ADC 模拟输入	
			X	X	0	X	X	串口 0 输出	
			1	1	1	X	0	IO	
			1	0	1	0	0	外部中断 0 输入	
			1	0	1	1	0	I2C 时钟 SCL 脚	
PT2.3/RX0/AD2/INT1/SDA	I/O	22	I/O; 串口 0 输入; ADC 模拟输入通道 2; 外部中断 1;					IO 功能描述	
			UART0EN	UART0_SEL	AIENB2[2]	PTW1[5]	I2C_EN	IO (默认值)	
			0	X	1	0	0	外部中断 1 输入	
			0	X	1	1	0	ADC 模拟输入	
			X	X	0	X	X	串口 0 输入	
			1	1	1	X	0	IO	
			1	0	1	0	0	外部中断 1 输入	
			1	0	1	1	0	I2C 数据 SDA 脚	

I/O	I/O	23	I/O: 串口 1 输入; ADC 模拟输入通道 1; 外部中断 1; PWM1 输出					IO 功 能 描 述		
			UART1E N	UART1_SE L	AIENB2[1]	T1OU T	PWM1OU T		PWM1_SE L	PTW1[7]
PT2.6/RX1/AD1/INT 1/PWM1	I/O	23	0	X	1	0	0	X	0	IO(默 认值)
			0	X	1	X	X	0	0	
			0	X	1	0	0	X	1	外 部 中 断 输 入
			0	X	1	X	X	0	1	
			0	X	1	0	1	1	X	蜂 鸣 器 1 输 出
			0	X	1	1	0	1	X	PWM 1 输 出
			0	X	1	1	1	1	X	PWM 1 取 反 输 出
			X	X	0	X	X	X	X	ADC 模 拟 输 入
			1	1	1	X	X	X	X	串 口 1 输 入
			1	0	1	0	0	X	0	IO
			1	0	1	X	X	0	0	
			1	0	1	0	0	X	1	
			1	0	1	X	X	0	1	外 部 中 断 输 入

PT2.7/TX1/AD0/INT0/PWM2	I/O	24	I/O: 串口 1 输出; ADC 模拟输入通道 0; 外部中断 0; PWM2 输出							
			UART1 EN	UART1_SEL	AIENB2 [0]	T2OUT	PWM2OUT	PWM2_SEL	PTW0[7]	IO 功能描述
			0	X	1	0	0	X	0	IO (默认值)
			0	X	1	X	X	0	0	
			0	X	1	0	0	X	1	外部中断 0 输入
			0	X	1	X	X	0	1	
			0	X	1	0	1	1	X	蜂鸣器 2 输出
			0	X	1	1	0	1	X	PWM2 输出
			0	X	1	1	1	1	X	PWM2 取反输出
			X	X	0	X	X	X	X	ADC 模拟输入
			1	1	1	X	X	X	X	串口 1 输出
			1	0	1	0	0	X	0	IO
			1	0	1	X	X	0	0	
			1	0	1	0	0	X	1	外部中断 0 输入
1	0	1	X	X	0	1				
PT3.0/SEG20	I/O	25	I/O: LCD Segment							
			SEGCON1[0]	IO 功能描述						
			=1	开启 LCD SEG 功能						
			=0	IO (默认值)						
PT3.1/SEG19	I/O	26	I/O: LCD Segment							
			SEGCON1[1]	IO 功能描述						
			=1	开启 LCD SEG 功能						
			=0	IO (默认值)						
PT3.2/SEG18	I/O	27	I/O: LCD Segment							
			SEGCON1[2]	IO 功能描述						
			=1	开启 LCD SEG 功能						
			=0	IO (默认值)						

PT3.3/SEG17	I/O	28	I/O: LCD Segment	
			SEGCON1[3]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT3.4/SEG16	I/O	29	I/O: LCD Segment	
			SEGCON1[4]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT3.5/SEG15	I/O	30	I/O: LCD Segment	
			SEGCON1[5]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT3.6/SEG14	I/O	31	I/O: LCD Segment	
			SEGCON1[6]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT3.7/SEG13	I/O	32	I/O: LCD Segment	
			SEGCON1[7]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT4.0/SEG12	I/O	33	I/O: LCD Segment	
			SEGCON0[0]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT4.1/SEG11	I/O	34	I/O: LCD Segment	
			SEGCON0[1]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)

PT4.2/SEG10	I/O	35	I/O: LCD Segment	
			SEGCON0[2]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT4.3/SEG9	I/O	36	I/O: LCD Segment	
			SEGCON0[3]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT4.4/SEG8	I/O	37	I/O: LCD Segment	
			SEGCON0[4]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT4.5/SEG7	I/O	38	I/O: LCD Segment	
			SEGCON0[5]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT4.6/SEG6	I/O	39	I/O: LCD Segment	
			SEGCON0[6]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
PT4.7/SEG5	I/O	40	I/O: LCD Segment	
			SEGCON0[7]	IO 功能描述
			=1	开启 LCD SEG 功能
			=0	IO (默认值)
SEG4	O	41	LCD Segment	
SEG3	O	42	LCD Segment	
COM6/SEG2	O	43	LCD Com; LCD Segment	
COM5/SEG1	O	44	LCD Com; LCD Segment	
COM4	O	45	LCD Com	
COM3	O	46	LCD Com	
COM2	O	47	LCD Com	

COM1	O	48	LCD Com
------	---	----	---------

封装 功能特性	SOP16	SOP20	SOP24	LQFP48
XTAL	√	√	√	√
BZ	√	√	√	√
INT0	6	6	7	8
INT1	5	7	8	8
PWM1	√	√	√	√
PWM2	√	√	√	√
PWM3	√	√	√	√
T1	√	√	√	√
T2	√	√	√	√
LPD 低电压检测		√	√	√
UART0	√	√	√	√
UART1	√	√	√	√
I2C				√
SPI	√	√	√	√
LCD Driver				√
10 Bit ADC	√	√	√	√
24 Bit ADC	√	√	√	√
RTC	√	√	√	√
烧录的时钟口	√	√	√	√
烧录的数据口	√	√	√	√
I/O	11	13	17	32

2 标准功能

2.1 CPU 核

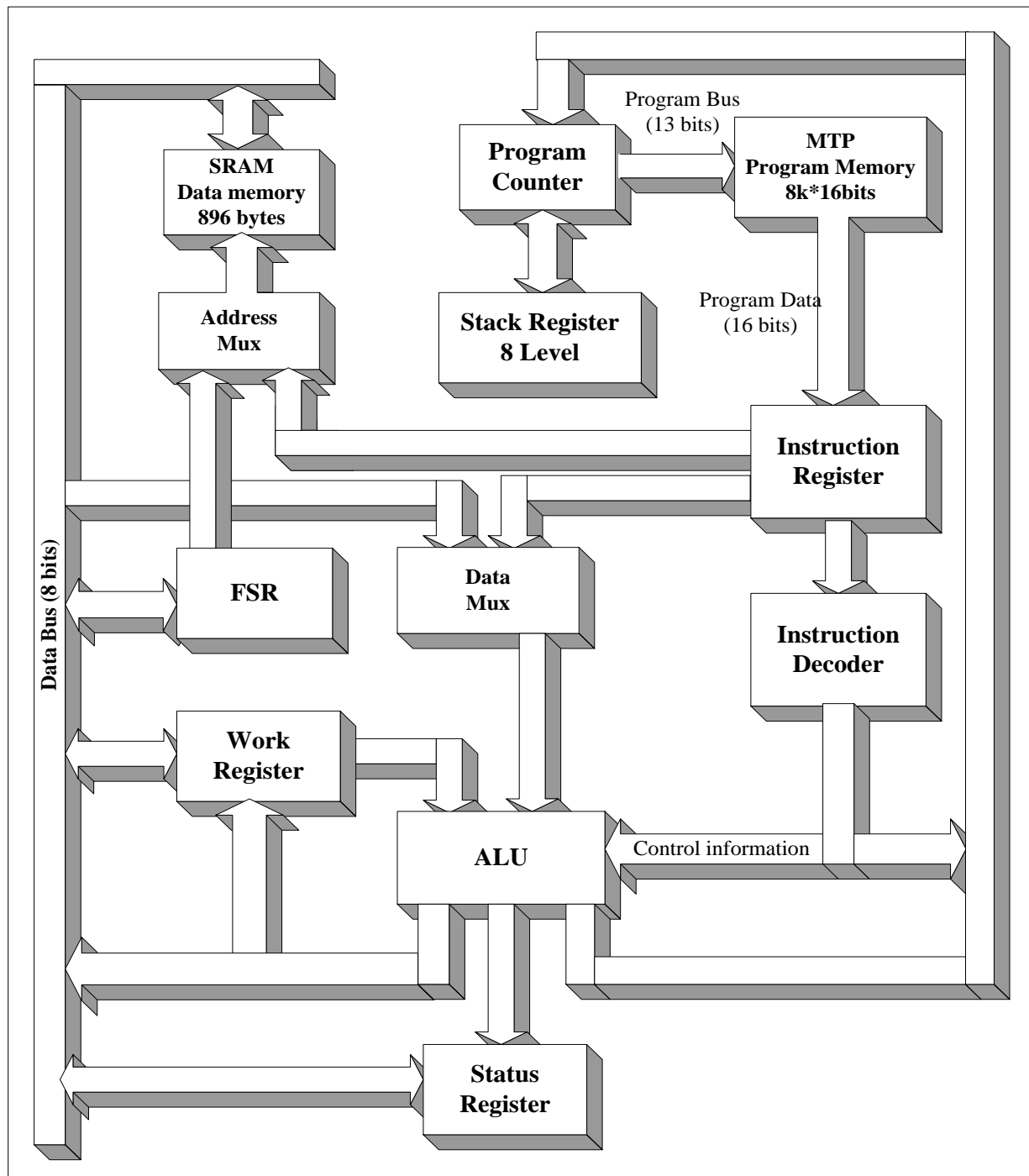


图 5 CPU 核的功能模块图

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

表 5 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。
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程序计数器以继续原来的程序处理。
指令寄存器	<p>程序计数器将指令指针（程序存储器地址）推送到程序存储器，程序存储器将程序存储器的数据（16bits）及指令推送到指令寄存器。</p> <p>芯片的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。</p> <p>CPU 能将立即数推送到工作寄存器，或者进行某些处理后，根据控制信息，将立即数存储到直接地址所指向的数据存储器寄存器中。</p> <p>直接地址（9bits）</p> <p>数据存储器的地址。CPU 能利用此地址来对数据存储器进行操作。</p> <p>直接数据（8bits）</p> <p>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</p> <p>它记录着 ALU 的操作信息。</p>
指令译码器	指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。
算术逻辑单元	算术逻辑单元不仅能完成 8 位二进制的加，减，加 1，减 1 等算术计算，还能对 8 位变量进行逻辑的与，或，异或，循环移位，求补，清零等逻辑运算。
工作寄存器	工作寄存器是用来缓存数据存储器中某些存储地址的数据。
状态寄存器	当 CPU 利用 ALU 处理寄存器数据时，如下的状态会随着如下顺序变化：PD, TO, DC, C 及 Z。
文件选择寄存器	在芯片的指令集中，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	芯片内带 8k×16 位的 MTP 作为程序存储器。由于指令的操作码（OPCODE）是 16bits，用户最多只能编程 8192×16 的指令。程序存储器的地址总线是 13bits，数据总线是 16bits。
EEPROM 存储器	芯片内带 128×8 位的 EEPROM，EEPROM 会占用 MTP 的地址线。
数据存储器	芯片内带 896 bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 9bits，数据总线是 8bits。

2.1.1 存储器

(1) 程序存储器

芯片的程序存储器是 8192×16bit 的 MTP(地址范围是 0000H-1FFFH)。系统的 reset 地址实际为 0x0000，中断入口地址实际为 0x0004。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。

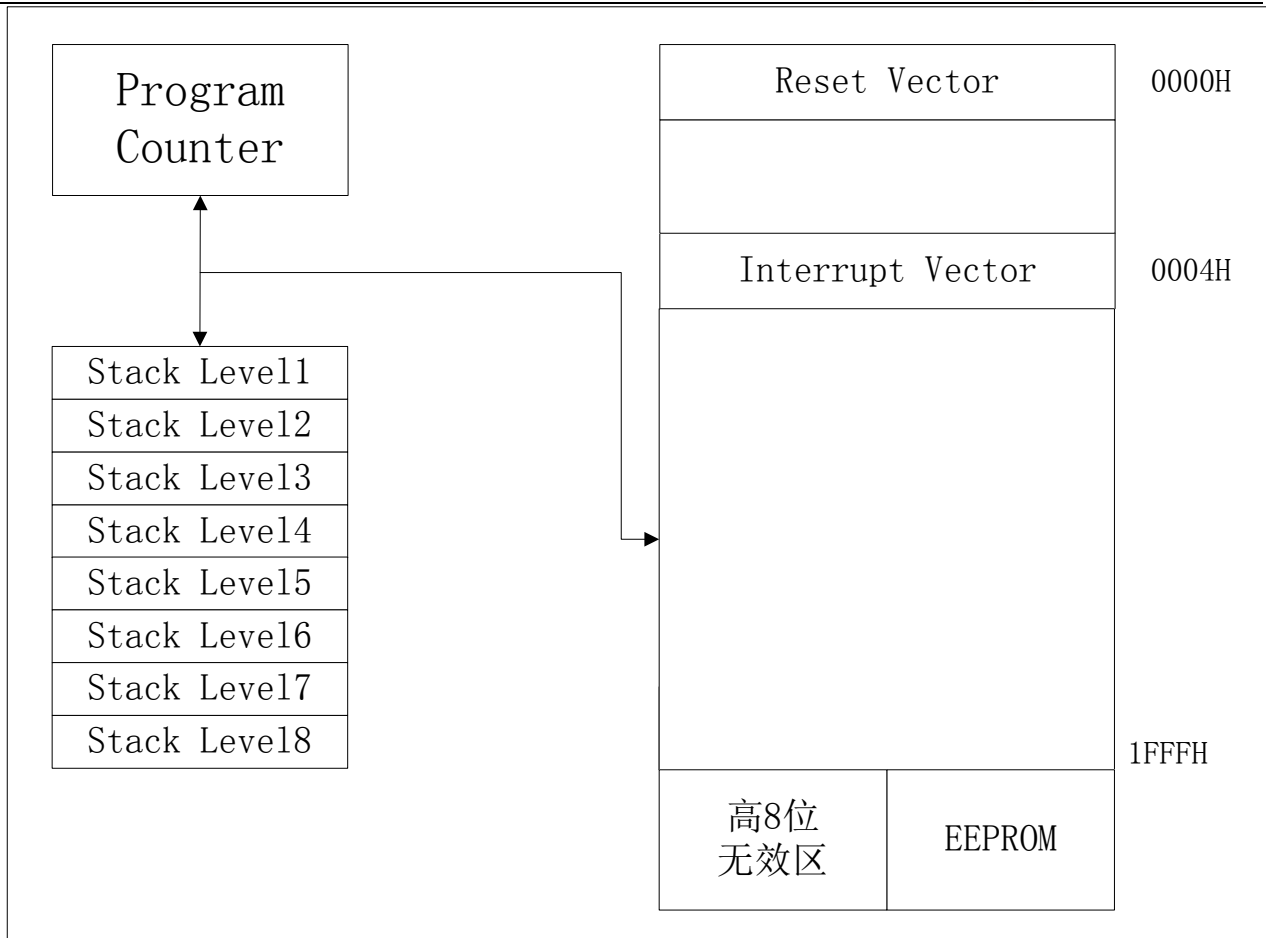


图 6 程序存储器

(2) EEPROM 存储器

EEPROM 主要用于程序非易性数据的存储,EEPROM 大小是 128Bytes,地址范围是 2000H-2007FH。

(3) 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 0x000 至 0x008 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 0x009 至 0x07F 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。

注意:数据存储器的空间是 896Bytes。

其中 F4H, PAGE=0 的 174H,PAGE=1 的 174H,PAGE=2 的 174H

PAGE=0 的 1F4H,PAGE=1 的 1F4H,PAGE=2 的 1F4H

FBH, PAGE=0 的 17BH,PAGE=1 的 17BH,PAGE=2 的 17BH;

PAGE=0 的 1FBH,PAGE=1 的 1FBH,PAGE=2 的 1FBH;

PAGE=0 的 169H,PAGE=1 的 169H,PAGE=2 的 169H

E9H, PAGE=0 的 1E9H,PAGE=1 的 1E9H,PAGE=2 的 1E9H 这些 RAM 地址暂不开放。

注意:SRAM 有 128Byte 地址不受 page 控制。

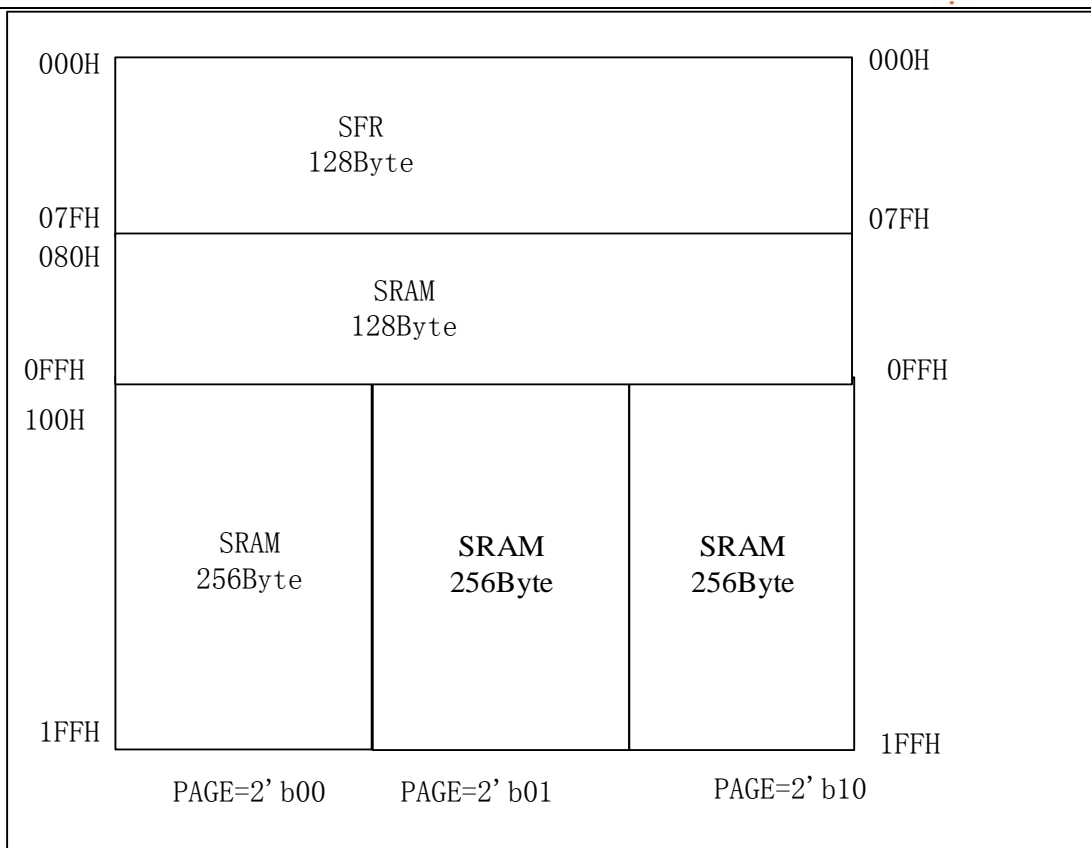


图 7 数据存储器的

表 6 数据存储地址分配

数据存储	起始地址	结束地址
系统特殊功能寄存器	000H	008H
外设特殊功能寄存器	009H	07FH
通用数据存储 (PAGE=2'b00)	080H	1FFH
通用数据存储 (PAGE=2'b01)	080H	1FFH
通用数据存储 (PAGE=2'b10)	080H	1FFH

Bank 选择寄存器 (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	-	-	-	-	PAGE1	PAGE0
-	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]: 页选择位

在间接寻址或者直接寻址的页控制信号, SFR 地址和 RAM 前面 128 个地址不受 Page 控制。

通过 IND、IRP0 和 FSR0 或 IND、IRP1 和 FSR1 寄存器可以对数据存储器和特殊功能寄存器进行间接访问。当从间接地址寄存器(IND0/IND1)读入数据时，MCU 实际上是以 FSR0/FSR1 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/IND1)写入数据时，MCU 实际上是以 FSR0/FSR1 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 8 间接地址访问。

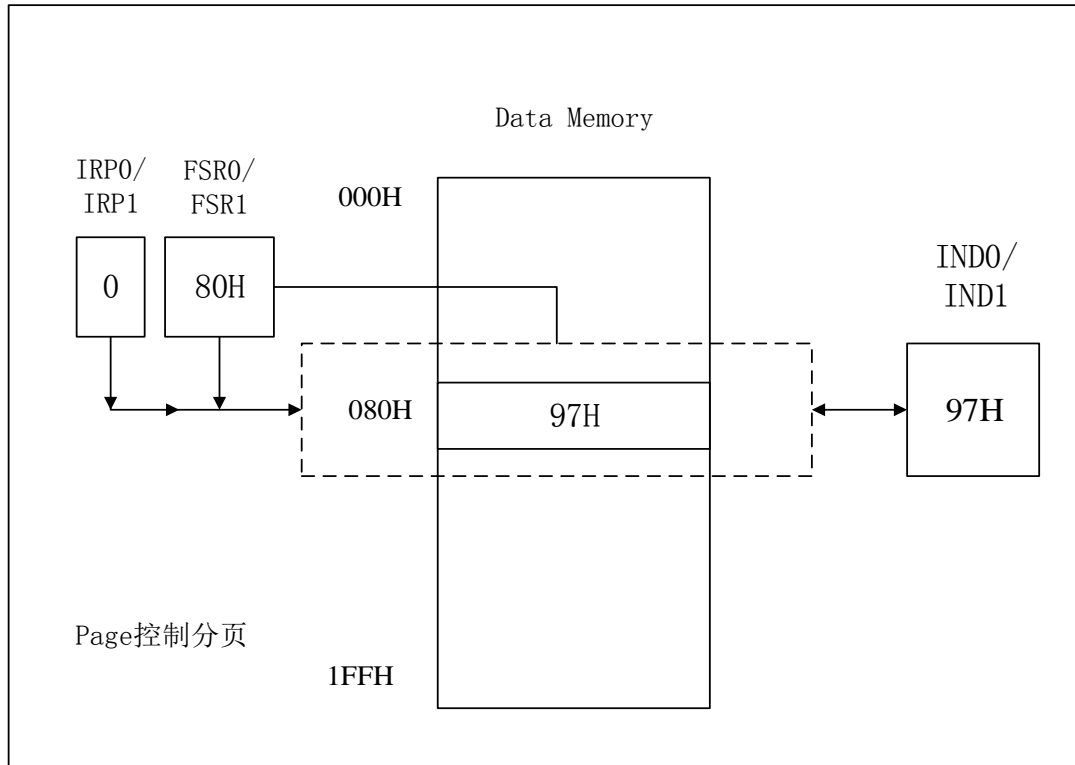


图 8 间接地址访问

2.1.2 状态寄存器

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

状态寄存器（地址为 04H）

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

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

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

1 = 看门狗定时溢出发生

0 = 上电复位后

Bit 2 DC: 半字节进位标志/借位标志，用于 ADDWF (C) 及 SUBWF (C)

用于借位时，极性相反

1 = 结果的第 4 位出现进位溢出

0 = 结果的第 4 位不出现进位溢出

Bit 1 C: 进位标志/借位标志

用于借位时，极性相反

- 1 = 结果的最高位 (MSB) 出现进位溢出
- 0 = 结果的最高位 (MSB) 不出现进位溢出

Bit 0 Z: 零标志

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

注: DC、C、Z 寄存器位写入的数据可能会被运算结果冲刷掉。

特性 (Property) :

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

2.1.3 中断寄存器

中断系统的入口地址为 0x0004，各个中断之间没有优先级，靠程序控制各个中断的优先级。只要有中断标志位，就会有中断响应，响应中断之后需要软件将中断标志位清除，否则会不断响应中断。

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

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF	-	TM2IF	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	u0000000
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	00000000
32H	INTF2	UR1_RHIF	UR1_RNIF	UR1_THIF	UR1_TEIF	UR0_RHIF	UR0_RNIF	UR0_THIF	UR0_TEIF	00000000
33H	INTE2	UR1_RHIE	UR1_RNIE	UR1_THIE	UR1_TEIE	UR0_RHIE	UR0_RNIE	UR0_THIE	UR0_TEIE	00000000
5AH	INTF3	I2C_TIF	I2C_RIF	UR1ERR_IE	UR1WK_IF	UR0WK_IF	TM3IF	RTCIF	SPIIF	00000000
5BH	INTE3	I2C_TIE	I2C_RIE	UR0ERR_IE	UR1WK_IE	UR0WK_IE	TM3IE	RTCIE	SPIIE	00000000
0EH	WDTCON	WDTEN	ROOT_EN	AIENB3	I2C_DIV[1:0]		WDTS[2:0]			00100000
14H	ROOT	PROT[7:0]								00000000

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	R/W-0
INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7GIE: 全局中断使能标志，此 BIT 在 WDT 溢出会被硬件清除。

- 1 = 使能所有非屏蔽中断
- 0 = 不使能所有中断

Bit 6 TM2IE: 8-Bit 定时器 2 中断使能标志

- 1 = 使能定时器 1 中断
- 0 = 不使能定时器 1 中断

Bit 5 TM1IE: 8-Bit 定时器 1 中断使能标志

- 1 = 使能定时器 1 中断
- 0 = 不使能定时器 1 中断

Bit 4 TM0IE: 8-Bit 定时器 0 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 AD2IE: 10-bit AD 中断使能标志

- 1 = 使能 10-bit AD 中断
- 0 = 不使能 10-bit AD 中断

Bit 2 ADIE: Sigma Delta AD 中断使能标志

- 1 = 使能 Sigma Delta AD 中断
- 0 = 不使能 Sigma Delta AD 中断

Bit 1 E1IE: 外部中断 1 使能标志

- 1 = 使能外部中断 1

0 = 不使能外部中断 1

Bit 0 E0IE: 外部中断 0 使能标志

1 = 使能外部中断 0

0 = 不使能外部中断 0

INTF 寄存器 (地址为 06H)

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
INTF	-	TM2IF	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 TM2IF: 8-Bit 定时器 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 5 TM1IF: 8-Bit 定时器 1 中断标志, 软件清零, 硬件置高

1 = 发生定时中断, 必须软件清 0

0 = 没发生定时中断

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

1 = 发生定时中断, 必须软件清 0

0 = 没发生定时中断

Bit 3 AD2IF: 10-bit AD 中断标志, 软件清零, 硬件置高

1 = 发生 10-bit AD 中断, 必须软件清 0

0 = 没发生 10-bit AD 中断

Bit 2 ADIF: Sigma Delta AD 中断标志, 软件清零, 硬件置高

1 = 发生 Sigma Delta AD 中断, 必须软件清 0

0 = 没发生 Sigma Delta AD 中断

Bit 1 E1IF: 外部中断 1 中断标志, 软件清零, 硬件置高

1 = 外部中断 1 发生中断, 必须软件清 0

0 = 外部中断 1 没发生中断

Bit 0 E0IF: 外部中断 0 中断标志, 软件清零, 硬件置高

1 = 外部中断 0 发生中断, 必须软件清 0

0 = 外部中断 0 没发生中断

INTE2 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
INTE2	UR1_RHIE	UR1_RNIE	UR1_THIE	UR0_TEIE	UR0_RHIE	UR0_RNIE	UR0_THIE	UR0_TEIE
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 UR1_RHIE: 串口 1 接收 FIFO 半满中断使能(当 ROOT_EN =1 时, 才能被改写)

1 = 使能串口 1 接收 FIFO 半满中断

0 = 不使能串口 1 接收 FIFO 半满中断

Bit 6 UR1_RNIE: 串口 1 接收 FIFO 非空中断使能(当 ROOT_EN =1 时, 才能被改写)

1 = 使能串口 1 接收 FIFO 非空中断

0 = 不使能串口 1 接收 FIFO 非空中断

Bit 5 UR1_THIE: 串口 1 发送 FIFO 半空中断使能(当 ROOT_EN =1 时, 才能被改写)

1 = 使能串口 1 发送 FIFO 为半空的中断

0 = 不使能串口 1 发送 FIFO 为半空的中断

Bit 4 UR1_TEIE: 串口 1 发送 FIFO 空中断使能(当 ROOT_EN =1 时, 才能被改写)

1 = 使能串口 1 发送 FIFO 为空中断

0 = 不使能串口 1 发送 FIFO 为空中断

Bit 3 UR0_RHIE: 串口 0 接收 FIFO 半满中断使能

1 = 使能串口 0 接收 FIFO 半满中断

0 = 不使能串口 0 接收 FIFO 半满中断

Bit 2 UR0_RNIE: 串口 0 接收 FIFO 非空中断使能

- 1 = 使能串口 0 接收 FIFO 非空中断
- 0 = 不使能串口 0 接收 FIFO 非空中断

Bit 1 UR0_THIF: 串口发送 FIFO 半空中断使能

- 1 = 使能串口 0 发送 FIFO 为半空的中断
- 0 = 不使能串口 0 发送 FIFO 为半空的中断

Bit 0 UR0_TEIE: 串口 0 发送 FIFO 空中断使能

- 1 = 使能串口 0 发送 FIFO 为空中断
- 0 = 不使能串口 0 发送 FIFO 为空中断

INTF2 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
INTF2	UR1_RHIF	UR1_RNIF	UR1_THIF	UR0_TEIF	UR0_RHIF	UR0_RNIF	UR0_THIF	UR0_TEIF
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 UR1_RHIE: 串口 1 接收 FIFO 半满中断发生标志

- 1 = 发生串口 1 接收 FIFO 半满中断
- 0 = 没发生串口 1 接收 FIFO 半满中断

Bit 6 UR1_RNIF: 串口 1 接收 FIFO 非空中断发生标志

- 1 = 发生串口 1 接收 FIFO 非空中断
- 0 = 没发生串口 1 接收 FIFO 非空中断

Bit 5 UR1_THIF: 串口 1 发送 FIFO 半空中断发生标志

- 1 = 发生串口 1 发送 FIFO 为半空的中断
- 0 = 没发生串口 1 发送 FIFO 为半空的中断

Bit 4 UR1_TEIF: 串口 1 发送 FIFO 空中断发生标志

- 1 = 发生串口 1 发送 FIFO 为空中断
- 0 = 没发生串口 1 发送 FIFO 为空中断

Bit 3 UR0_RHIF: 串口 0 接收 FIFO 半满中断发生标志

- 1 = 发生串口 0 接收 FIFO 半满中断
- 0 = 没发生串口 0 接收 FIFO 半满中断

Bit 2 UR0_RNIF: 串口 0 接收 FIFO 非空中断发生标志

- 1 = 发生串口 0 接收 FIFO 非空中断
- 0 = 没发生串口 0 接收 FIFO 非空中断

Bit 1 UR0_THIF: 串口 0 发送 FIFO 半空中断发生标志

- 1 = 发生串口 0 发送 FIFO 为半空的中断
- 0 = 没发生串口 0 发送 FIFO 为半空的中断

Bit 0 UR0_TEIF: 串口 0 发送 FIFO 空中断发生标志

- 1 = 发生串口 0 发送 FIFO 为空中断
- 0 = 没发生串口 0 发送 FIFO 为空中断

INTE3 寄存器 (地址为 5BH)

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
INTE3	I2C_TIE	I2C_RIE	UR0ERR_IE	UR1WK_IE	UR0WK_IE	TM3IE/ I2C_ST_IE	RTCIE	SPIIE
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

BIT7 I2C_TIE: I2C 从机发送中断使能

- 1 : 使能 I2C 从机发送中断
- 0 : 关闭 I2C 从机发送中断

BIT6 I2C_RIE: I2C 从机接收中断使能

- 1 : 使能 I2C 从机接收中断
- 0 : 关闭 I2C 从机接收中断

Bit5 UR0ERR_IE: UART0 错误中断使能, UART0 错误包括 RXFIFO 溢出和接收停止位错误, 清此中断需要对相应的溢出标志或者接收停止位错误标志清零。

- 1 = 使能 UART 0 错误中断使能
- 0 = 不使能 UART0 错误中断使能

Bit 4 UR1WK_IE: UART1 唤醒中断使能标志, 该中断具有唤醒 Sleep 功能(只有在 ROOT_EN=1, 才能改写)

- 1 = 使能 UART 1 唤醒中断
- 0 = 不使能 UART 1 唤醒中断

Bit 3 UR0WK_IE: UART 0 唤醒中断使能标志, 该中断具有唤醒 Sleep 功能

- 1 = 使能 UART 0 唤醒中断
- 0 = 不使能 UART 0 唤醒中断

Bit 2 TM3IE/I2C_ST_IE: 8-Bit 定时器 3 中断使能 或者 I2C_START 中断使能
METCH[5]=0, 为定时器 3 中断使能,
METCH[5]=1, 为 I2C_START 中断使能。

- 1 = 使能定时器 3 或者 I2C_START 中断
- 0 = 不使能定时器 3 或者 I2C_START 中断

Bit 1 RTCIE: RTC 中断使能标志

- 1 = 使能 RTC 中断
- 0 = 不使能 RTC 中断

Bit 0 SPIIE: SPI 中断使能标志

- 1 = 使能 SPI 中断
- 0 = 不使能 SPI 中断

INTF3 寄存器 (地址为 5AH)

特性	R-0	R-0	RW-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTF3	I2C_TIF	I2C_RIF	UR1ERR_IE	UR1WK_IF	UR0WK_IF	TM3IF/ I2C_ST_IF	RTCIF	SPIIF
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

BIT7 I2C_TIF 从机发送中断标志位
1=I2C 从机请求 MCU 写入发送数据中断, 必须软件清 0
0=没发生 I2C 从机请求 MCU 写入发送数据中断

BIT6 I2C_RIF I2C 从机接收中断标志位
1 =I2C 从机完成 1-Byte 数据接收中断, 必须软件清 0
0 = 没发生 I2C 从机接收中断

Bit5 UR1ERR_IE: UART1 错误中断使能, UART1 错误包括 RXFIFO 溢出和接收停止位错误, 清此中断需要对相应的溢出标志或者接收停止位错误标志清零。

- 1 = 使能 UART 1 错误中断使能
- 0 = 不使能 UART1 错误中断使能

Bit4 UR1WK_IF: UART 1 唤醒中断标志, 该中断具有唤醒 Sleep 功能

- 1 = 发生 UART 1 唤醒中断, 必须软件清 0
- 0 = 没发生 UART 1 唤醒中断

Bit3 UR0WK_IF: UART 0 唤醒中断标志, 该中断具有唤醒 Sleep 功能

- 1 = 发生 UART 0 唤醒中断, 必须软件清 0
- 0 = 没发生 UART 0 唤醒中断

Bit2 TM3IF/I2C_ST_IF: 8-Bit 定时器 3 中断标志 或者 I2C_START 中断标志
METCH[5]=0, 为定时器 3 中断标志,
METCH[5]=1, 为 I2C_START 中断标志。

- 1 = 发生定时器 3 或者 I2C_START 中断, 必须软件清 0
- 0 = 没有发生定时器 3 或者 I2C_START 中断

- Bit1 RTCIF: RTC 中断标志
1 = 发生 RTC 中断, 必须软件清 0
0 = 没有发生 RTC 中断
- Bit0 SPIIF: SPI 中断标志
1 = 发生 SPI 中断, 必须软件清 0
0 = 没有发生 SPI 中断

ROOT 寄存器 (地址为 14H)

特性	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ROOT	PROT[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

如果需要改变 ROOT_EN 的值, RPROT 按照顺序写入 35H, C8H, 17H 解锁写保护, 才能对 ROOT_EN 进行写操作进行改变 ROOT_EN 的值, 改写 ROOT_EN 的指令需要接着序列后面。

当按照顺序写入 35H, C8H, 17H 解锁 ROOT_EN 的写保护, 解锁后对 ROOT 寄存器再写入其他值后, ROOT_EN 被锁住, 不能被改变。

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

2.2 SFR

特殊功能寄存器 (SFR) 包含系统专用寄存器和辅助专用寄存器

表 7 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
00H	IND0	以 FSR0 中内容作为地址的数据存储器中的数据								xxxxxxx	
01H	IND1	以 FSR1 中内容作为地址的数据存储器中的数据								xxxxxxx	
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	-	-	-	PD	TO	DC	C	Z	uuu00xxx	
05H	WORK	工作寄存器								00000000	
06H	INTF	-	TM2IF	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	u0000000	
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	00000000	
08H	BSR	IRP0	IRP1	-	-	-	-	PAGE[1:0]		00uuuu00	
09H	MCK	ICK_IDO_BY	M3_CK	M2_CK	M1_CK	ECK_EN	WDT_CLK_EN	ICK_EN	CLK_SEL	00100010	
0AH	EADRH	PROG_BUSY	READ_CHECK	PARH[5:0]							00000000
0BH	EADRL	PARL[7:0]								00000000	
0CH	EDAT	EDAT[7:0]								11111111	
0DH	EOPEN	EOPEN[7:0]								00000000	
0EH	WDTCON	WDTEN	ROOT_EN	AIENB3	I2C_DIV[1:0]		WDTS[2:0]			00100000	
0FH	WDTIN	WDTIN[7:0]								00000000	
10H	ADOH	ADO [23:16]								00000000	
11H	ADOL	ADO[15:8]								00000000	
12H	PCLOADL	PCLOAD[7:0]								00000000	
13H	PCLOADH	-	-	-	PCLOAD[12:8]					uuu00000	
14H	ROOT	PROT[7:0]								00000000	
15H	PCK	UARTIDIV			UART0DIV			S_beep[1:0]		00000000	
16H	ADCFG	ADSC	GAIN1[1:0]		S_GAIN[1:0]		R_GAIN	CFG_C[1:0]		00000000	
17H	ANACFG	LDOEN	LDOS[1:0]		BGR_ENB	BGID	SINL[1:0]		ADEN	00000000	
18H	TEMPC	TEMPC[7:0]								00000000	
19H	AD2OH	AD2EN	AD2START	AD3_SL[3:0]			AD2O[9:8]			00000000	
1AH	AD2OL	AD2O[7:0]								00000000	
1BH	LVDCON	LVDEN	AD2_REF	AIENB1	SILB[2:0]		LBOUT	LB_RST_CON		001000x0	
1CH	ADOLL	ADO[7:0]								00000000	
1DH	PT1	PT1[7:0]								xxxxxxx	
1EH	PT1EN	PT1EN[7:0]								00000000	
1FH	PT1PU	PT1UP[7:0]								00000000	
20H	AIENB	AIENB2[7:0]								11111111	
21H	PT2	PT2[7:0]								xxxxxxx	
22H	PT2EN	PT2EN[7:0]								00000000	
23H	PT2PU	PT2PU[7:0]								00000000	
24H	PT3	PT3[7:0]								xxxxxxx	
25H	PT3EN	PT3EN[7:0]								00000000	
26H	PT3PU	PT3PU[7:0]								00000000	
27H	PT4	PT4[7:0]								xxxxxxx	
28H	PT4EN	PT4EN[7:0]								00000000	
29H	PT4PU	PT4PU[7:0]								00000000	
2AH	PT5	-	-	-	PT5[4:0]					000xxxx	
2BH	PT5EN	SMT5_EN	SMT2_EN	SMT1_EN	PT5EN[4:0]					11100000	
2CH	PT5PU	PT15_VDD	PT14_VDD	I2C_VDD	PT5PU[4:0]					00000000	
2DH	PTCON	BZEN	PWM3_SEL	PWM2_SEL	PWM1_SEL	E1M[1:0]		E0M[1:0]		00000000	
2EH	PTINT0	PTW0[7:0]								00000000	
2FH	PTINT1	PTW1[7:0]								00000000	
30H	SEGCON0	SEGCON0[7:0]								00000000	
31H	SEGCON1	SEGCON1[7:0]								00000000	
32H	INTF2	UR1_RHIF	UR1_RNIF	UR1_THIF	UR0_TEIF	UR0_RHIF	UR0_RNIF	UR0_THIF	UR0_TEIF	00000000	
33H	INTE2	UR1_RHIE	UR1_RNIE	UR1_THIE	UR0_TIEI	UR0_RHIE	UR0_RNIE	UR0_THIE	UR0_TIEI	00000000	
34H	TM0CON	TOEN	TORATE[2:0]			TORSTB		TOSEL		0000u100	
35H	TM0IN	TM0IN[7:0]								00000000	
36H	TM0CNT	TM0CNT[7:0]								00000000	
37H	TM1CON	T1EN	T1RATE[2:0]			T1CKS	T1RSTB	T1OUT	PWM1OUT	00000100	
38H	TM1IN	TM1IN[7:0]								00000100	
39H	TM1CNT	TM1CNT[7:0]								00000000	
3AH	TM1R	TM1R[7:0]								00000000	
3BH	TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT	00000100	
3CH	TM2IN	TM2IN[7:0]								00000000	
3DH	TM2CNT	TM2CNT[7:0]								00000000	
3EH	TM2R	TM2R[7:0]								00000000	
3FH	TM3CON	T3EN	T3RATE[2:0]			T3CKS	T3RSTB	T3OUT	PWM3OUT	00000100	
40H	TM3IN	TM3IN[7:0]								00000000	
41H	TM3CNT	TM3CNT[7:0]								00000000	
42H	TM3R	TM3R[7:0]								00000000	

43H	LCD1	-	-	-	-	SEG1[3:0]				uuuu0000	
44H	LCD2	-	-	-	-	SEG2[3:0]				uuuu0000	
45H	LCD3	-	-	-	-	SEG3[5:0]				uu000000	
46H	LCD4	-	-	-	-	SEG4[5:0]				uu000000	
47H	LCD5	-	-	-	-	SEG5[5:0]				uu000000	
48H	LCD6	-	-	-	-	SEG6[5:0]				uu000000	
49H	LCD7	-	-	-	-	SEG7[5:0]				uu000000	
4AH	LCD8	-	-	-	-	SEG8[5:0]				uu000000	
4BH	LCD9	-	-	-	-	SEG9[5:0]				uu000000	
4CH	LCD10	-	-	-	-	SEG10[5:0]				uu000000	
4DH	LCD11	-	-	-	-	SEG11[5:0]				uu000000	
4EH	LCD12	-	-	-	-	SEG12[5:0]				uu000000	
4FH	LCD13	-	-	-	-	SEG13[5:0]				uu000000	
50H	LCD14	-	-	-	-	SEG14[5:0]				uu000000	
51H	LCD15	-	-	-	-	SEG15[5:0]				uu000000	
52H	LCD16	-	-	-	-	SEG16[5:0]				uu000000	
53H	LCD17	-	-	-	-	SEG17[5:0]				uu000000	
54H	LCD18	-	-	-	-	SEG18[5:0]				uu000000	
55H	LCD19	-	-	-	-	SEG19[5:0]				uu000000	
56H	LCD20	-	-	-	-	SEG20[5:0]				uu000000	
57H	LCDCN1	-	LCDFC[1:0]		WDT_LCD	CSE_LCD	-	VLCDX[1:0]		u0000u00	
58H	LCDCN2	LDCSCK[3:0]			LCDFR		LCDREF[2:0]			00000000	
59H	LCDENR	LDCCK[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL	11000000	
5AH	INTF3	I2C_TIF	I2C_RIF	UR1ERR_I E	UR1WK_IF	UR0WK_IF	TM3IF/ I2C_ST_IF	RTCIF	SPIIF	00000000	
5BH	INTE3	I2C_TIE	I2C_RIE	UR0ERR_IE	UR1WK_IE	UR0WK_IE	TM3IE/ I2C_ST_IE	RTCIE	SPIIE	00000000	
5CH	SPICFG	SPIEN	MSTEN	CKPHA	CKPOL	MULMST	WIREMOD	SPIBSY	SPIRST	00000000	
5DH	SPICN	SLVSEL	WCOL	MODCOL	-	-	-	-	-	000uuuuu	
5EH	SPICKR	SCR[7:0]									00000011
5FH	SPIDAT	SPIDAT[7:0]									00000000
60H	RTCCON	LIR	-	24HR/12H R	RTCEN	INTEGER[3:0]				0u000111	
61H	RTCAER	AER[7:0]									00000000
62H	RTCYEAR	10YEAR(0~9)				1YEAR(0~9)				00010101	
63H	RTCMON	-			10MON(0~1)	1MON(0~9)				uuu10001	
64H	RTCDAY	-	-	10DAY(0~3)	1DAT(0~9)				uu000001		
65H	RTCHOUR	-	AM_PM	10HOUR10(0~2)	1HOUR(0~9)				u0000001		
66H	RTCMIN	-	-	10MIN(0~5)	1MIN(0~9)				uu000000		
67H	RTCSEC	-	-	10SEC(0~5)	1SEC(0~9)				u0000000		
68H	I2CCON	I2C_EN	AWK_EN	CTS_EN	ACK_EN	I2CSTUS[3:0]				0000uuuu	
69H	I2CDAT	I2CDAT[7:0]									00000000
6AH	FRACTION	METCH[5:4]			FRACTION[5:0]					00000000	
6BH	ADCON	METCH[3:0]				ADM[3:0]				00000000	
6CH	TEST	TEST[7:0]									00000000
6DH	WDT_TRIM	TMOD[1:0]		ICK_SEL[1:0]		-				00xxuuuu	
70H	UR0_CR1	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN	TX_EN	UART_SEL	UART_EN	00000000	
71H	UR0_BRR0	BRR0									00000000
72H	UR0_BRR1	-	-	-	-	BRR1				00000000	
73H	UR0_TX_REG	TX_REG									00000000
74H	UR0_RX_REG	RX_REG									00000000
75H	UR0_CR2	-	RXF_WATER_LEVEL			TXF_WATER_LEVEL				00110011	
76H	UR0_ST	-	-	-	-	RX_BUSY	TX_BUSY	OV_ERR	STOP_ERR	00000000	
77H	UR1_CR1	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN	TX_EN	UART_SEL	UART_EN	00000000	
78H	UR1_BRR0	BRR0									00000000
79H	UR1_BRR1	-	-	-	-	BRR1				00000000	
7AH	UR1_TX_REG	TX_REG									00000000
7BH	UR1_RX_REG	RX_REG									00000000
7CH	UR1_CR2	-	RXF_WATER_LEVEL			TXF_WATER_LEVEL				00110011	
7DH	UR1_ST	-	-	-	-	RX_BUSY	TX_BUSY	OV_ERR	STOP_ERR	00000000	
7EH	RTCDWR	-				DWR[2:0] (0~6)				00000000	
7FH	I2CADR	I2CADR[6:0]						GC_EN		01010100	

注：进行读操作时，无效位读出为 0

特性 (Property) :

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

2.3 时钟系统

2.3.1 石英/陶瓷振荡器

石英/陶瓷振荡器由 XIN/XOUT 口驱动，此振荡器只有低速模式 可以连接 32768Hz 的晶振。

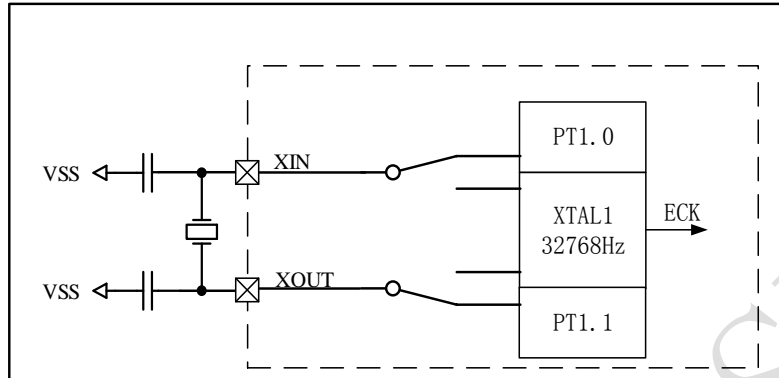


图 9 外部振荡器 1

2.3.2 内部振荡器

芯片内置 8M(10.5M/5.25M/4M)HZ 时钟，上电默认内置 8M 时钟使能打开，默认一直工作；在 sleep 模式下，内置 8M 时钟自动停止工作，减小功耗。

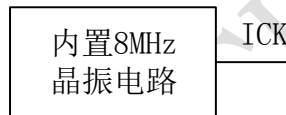


图 10 ICK 振荡器状态框图

芯片内置 3K WDT 时钟，在烧录模式和正常模式下，3K 时钟都受 wden 的控制，wdten=1 时才使能 3k WDT 时钟，wdten 默认值为 0。

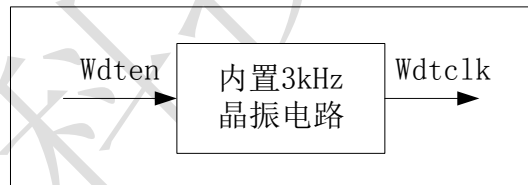


图 11 Wdt 振荡器状态框图

2.3.3 CPU 指令周期

表 8 CPU 指令周期寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
09H	MCCK	ICK_IDO_BY	M3_CK	M2_CK	M1_CK	ECK_EN	WDT_CLK_EN	ICK_EN	CLK_SEL	00100010

用户可以通过 MCK 寄存器中的 M3_CK、M2_CK、M1_CK 寄存器位来选择指令周期 (CPUCLK)。

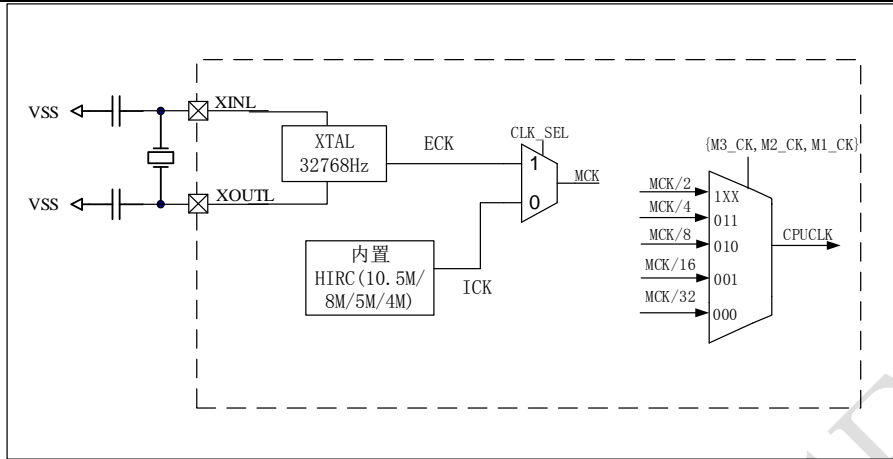


图 12 CPU 指令周期时钟示意图

表 9 指令周期选择列表

M3_CK	M2_CK	M1_CK	指令周期(KHz)
0	0	0	MCK/32
0	0	1	MCK/16
0	1	0	MCK/8(默认值)
0	1	1	MCK/4
1	X	X	不可用

位地址	标识符	功能	
7	ICK_IDO_BY	ick_ldo_bypass 功能控制 1: 关闭 ICK 模块的 LDO 0: 打开 ICK 模块的 LDO (默认)	
6	M3_CK	指令周期的分频控制, 见表 9	
5	M2_CK		
4	M1_CK		
3	ECK_EN	外部晶振 1 (32768Hz) 启动开关 1: 外部晶振 1 打开 0: 外部晶振 1 关闭 (默认)	
2	WDT_CLK_EN	内部 WDT 时钟启动开关 1: WDT_CLK 启动 0: WDT_CLK 关闭 (默认)	
1	ICK_EN	内部高速时钟启动开关 1: 内部时钟启动 (默认打开) 0: 内部时钟关闭	
0	CLK_SEL	时钟源选择位	
		CLK_SEL	CPU 时钟
		0	内部时钟作为系统时钟
1	外部晶振作为系统时钟		

2.3.4 TM0CLK (定时器 0 模块输入时钟)

TM0CLK 用于定时器 0 模块。Timer 的时钟源来自于 CPUCLK 或 WDT, 时钟源通过 T0SEL 进行选择。时钟源经过分频之后产生 TM0CLK 用做定时器 0 模块的时钟。

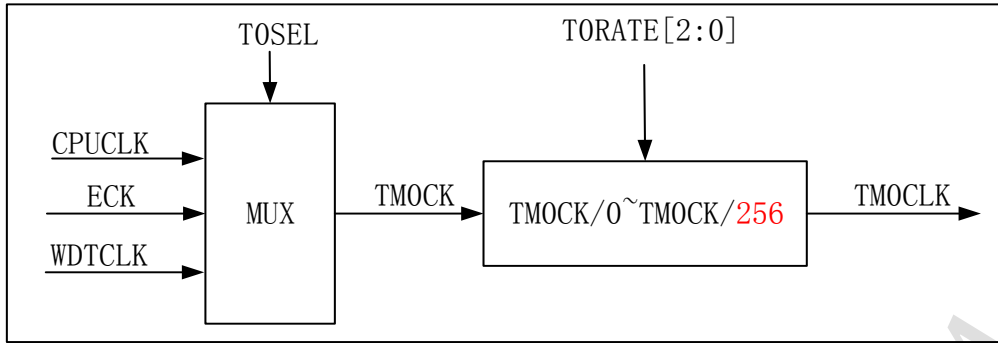


图 13 定时器 0 时钟分频示意图

表 10 定时器 0 控制寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
34H	TM0CON	TOEN	TORATE[2:0]		-	TORSTB	-	TOSEL	0000u100	

表 11 TM0CON 寄存器各位功能表

位地址	标识符	功能																											
7	TOEN	定时器 0 使能位 1: 使能定时器 0 0: 禁止定时器 0																											
6:4	TORATE[2:0]	定时器 0 时钟选择 <table border="1"> <thead> <tr> <th>TORATE [2:0]</th> <th>TM0CLK=CPUCLK</th> <th>TM0CLK=WDT/ECK</th> </tr> </thead> <tbody> <tr><td>000</td><td>TM0CK</td><td>TM0CK</td></tr> <tr><td>001</td><td>TM0CK /2</td><td>TM0CK /4</td></tr> <tr><td>010</td><td>TM0CK /4</td><td>TM0CK /8</td></tr> <tr><td>011</td><td>TM0CK /8</td><td>TM0CK /16</td></tr> <tr><td>100</td><td>TM0CK /16</td><td>TM0CK /32</td></tr> <tr><td>101</td><td>TM0CK /32</td><td>TM0CK /64</td></tr> <tr><td>110</td><td>TM0CK /64</td><td>TM0CK /128</td></tr> <tr><td>111</td><td>TM0CK /128</td><td>TM0CK /256</td></tr> </tbody> </table>	TORATE [2:0]	TM0CLK=CPUCLK	TM0CLK=WDT/ECK	000	TM0CK	TM0CK	001	TM0CK /2	TM0CK /4	010	TM0CK /4	TM0CK /8	011	TM0CK /8	TM0CK /16	100	TM0CK /16	TM0CK /32	101	TM0CK /32	TM0CK /64	110	TM0CK /64	TM0CK /128	111	TM0CK /128	TM0CK /256
TORATE [2:0]	TM0CLK=CPUCLK	TM0CLK=WDT/ECK																											
000	TM0CK	TM0CK																											
001	TM0CK /2	TM0CK /4																											
010	TM0CK /4	TM0CK /8																											
011	TM0CK /8	TM0CK /16																											
100	TM0CK /16	TM0CK /32																											
101	TM0CK /32	TM0CK /64																											
110	TM0CK /64	TM0CK /128																											
111	TM0CK /128	TM0CK /256																											
3	NC	保留位																											
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	时钟源选择 <table border="1"> <thead> <tr> <th>TOSEL[1:0]</th> <th>定时器 0 时钟源(TM0CK)</th> </tr> </thead> <tbody> <tr><td>00</td><td>CPUCLK</td></tr> <tr><td>01</td><td>CPUCLK</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 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效</td></tr> </tbody> </table>	TOSEL[1:0]	定时器 0 时钟源(TM0CK)	00	CPUCLK	01	CPUCLK	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 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效																	
TOSEL[1:0]	定时器 0 时钟源(TM0CK)																												
00	CPUCLK																												
01	CPUCLK																												
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 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效																												

2.3.5 TM1CLK (定时器 1 模块输入时钟)

TM1CLK 用于定时器 1 模块。Timer1 的时钟源来自于 CPUCLK。时钟源经过分频之后产生 TM1CLK 用作定时器 1 模块的时钟。

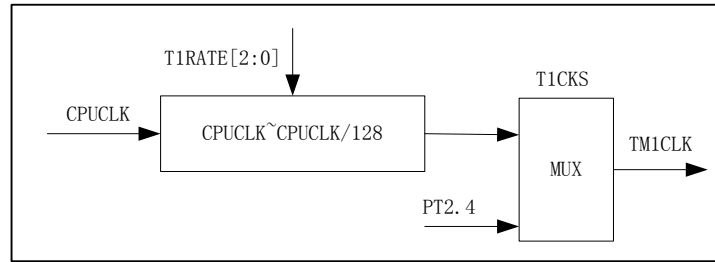


图 14 定时器 1 时钟分频示意图

表 12 定时器 1 控制寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
37H	TM1CON	T1EN	TIRATE[2:0]			TICKS	TIRSTB	T1OUT	PWM1OUT	00000100

表 13 TM1CON 寄存器各位功能表

位地址	标识符	功能																		
7	T1EN	定时/计数器 1 使能位 1: 使能定时器/计数器 1 0: 禁止定时器/计数器 1																		
6:4	TIRATE[2:0]	定时/计数器 1 时钟分频 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>TIRATE [2:0]</th> <th>TM1CLK</th> </tr> </thead> <tbody> <tr><td>000</td><td>CPUCLK</td></tr> <tr><td>001</td><td>CPUCLK/2</td></tr> <tr><td>010</td><td>CPUCLK/4</td></tr> <tr><td>011</td><td>CPUCLK/8</td></tr> <tr><td>100</td><td>CPUCLK/16</td></tr> <tr><td>101</td><td>CPUCLK/32</td></tr> <tr><td>110</td><td>CPUCLK/64</td></tr> <tr><td>111</td><td>CPUCLK/128</td></tr> </tbody> </table>	TIRATE [2:0]	TM1CLK	000	CPUCLK	001	CPUCLK/2	010	CPUCLK/4	011	CPUCLK/8	100	CPUCLK/16	101	CPUCLK/32	110	CPUCLK/64	111	CPUCLK/128
TIRATE [2:0]	TM1CLK																			
000	CPUCLK																			
001	CPUCLK/2																			
010	CPUCLK/4																			
011	CPUCLK/8																			
100	CPUCLK/16																			
101	CPUCLK/32																			
110	CPUCLK/64																			
111	CPUCLK/128																			
3	TICKS	定时/计数器 1 时钟源选择位 1: PT2.4 作为时钟 0: CPUCLK 的分频时钟																		
2	TIRSTB	定时/计数器 1 复位 1: 禁止定时/计数器 1 复位 0: 使能定时/计数器 1 复位 当将该位为 0 时, 定时器 1 复位后, TIRSTB 会自动置 1																		
1	T1OUT	PT1.7/PT2.6 口输出控制 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>T1OUT</th> <th>PWM1OUT</th> <th>PT1.7/PT2.6 输出控制</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>IO 输出</td></tr> </tbody> </table>	T1OUT	PWM1OUT	PT1.7/PT2.6 输出控制	0	0	IO 输出												
T1OUT	PWM1OUT	PT1.7/PT2.6 输出控制																		
0	0	IO 输出																		
0	PWM1OUT	<table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr><td>0</td><td>1</td><td>蜂鸣器输出</td></tr> <tr><td>1</td><td>0</td><td>PWM1 输出</td></tr> <tr><td>1</td><td>1</td><td>PWM1 取反输出</td></tr> </tbody> </table>	0	1	蜂鸣器输出	1	0	PWM1 输出	1	1	PWM1 取反输出									
0	1	蜂鸣器输出																		
1	0	PWM1 输出																		
1	1	PWM1 取反输出																		

2.3.6 TM2CLK (定时器 2 模块输入时钟)

TM1CLK 用于定时器 2 模块。Timer2 的时钟源来自于 CPUCLK。时钟源经过分频之后产生 TM2CLK 用做定时器 2 模块的时钟。

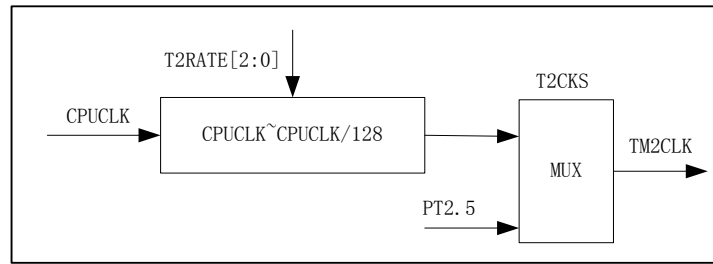


图 15 定时器 2 时钟分频示意图

表 14 定时器 2 控制寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3BH	TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT	00000100

表 15 TM2CON 寄存器各位功能表

位地址	标识符	功能
7	T2EN	定时/计数器 2 使能位 1: 使能定时器/计数器 2 0: 禁止定时器/计数器 2
6:4	T2RATE[2:0]	定时/计数器 1 时钟分频
		T2RATE [2:0] TM2CLK
		000 CPUCLK
		001 CPUCLK/2
		010 CPUCLK/4
		011 CPUCLK/8
		100 CPUCLK/16
		101 CPUCLK/32
10 CPUCLK/64		
111 CPUCLK/128		
3	T2CKS	定时/计数器 2 时钟源选择位 1: PT2.5 作为时钟 0: CPUCLK 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	PT1.6/PT2.7 输出控制
		T2OUT PWM2OUT PT1.6/PT2.7 输出控制
0	PWM2OUT	0 0 IO 输出
		0 1 IO 输出
		1 0 PWM2 输出
		1 1 PWM2 取反输出

2.3.7 TM3CLK (定时器 3 模块输入时钟)

TM3CLK 用于定时器 3 模块。Timer 的时钟源来自于 CPUCLK 或 WDT, 时钟源通过 T3SEL 进行选择。时钟源经过分频之后产生 TM3CLK 用做定时器 3 模块的时钟。

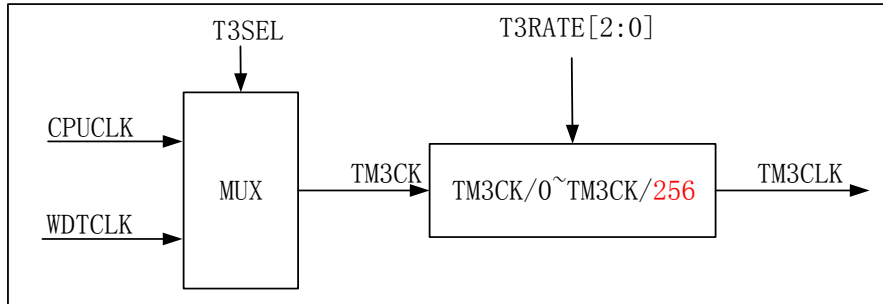


图 16 定时器 0 时钟分频示意图

表 16 定时器 0 控制寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
3FH	TM3CON	T3EN	T3RATE[2:0]			T3SEL	T3RSTB	T3OUT	PWM3OUT	00000100

表 17 TM3CON 寄存器各位功能表

位地址	标识符	功能																											
7	T3EN	定时器 3 使能位 1: 使能定时器 3 0: 禁止定时器 3																											
6:4	T3RATE[2:0]	定时器 3 时钟选择 <table border="1"> <thead> <tr> <th>T3RATE [2:0]</th> <th>TM3CLK=CPUCLK</th> <th>TM3CLK=WDT</th> </tr> </thead> <tbody> <tr><td>000</td><td>TM3CK</td><td>TM3CK</td></tr> <tr><td>001</td><td>TM3CK /2</td><td>TM3CK /4</td></tr> <tr><td>010</td><td>TM3CK /4</td><td>TM3CK /8</td></tr> <tr><td>011</td><td>TM3CK /8</td><td>TM3CK /16</td></tr> <tr><td>100</td><td>TM3CK /16</td><td>TM3CK /32</td></tr> <tr><td>101</td><td>TM3CK /32</td><td>TM3CK /64</td></tr> <tr><td>110</td><td>TM3CK /64</td><td>TM3CK /128</td></tr> <tr><td>111</td><td>TM3CK /128</td><td>TM3CK /256</td></tr> </tbody> </table>	T3RATE [2:0]	TM3CLK=CPUCLK	TM3CLK=WDT	000	TM3CK	TM3CK	001	TM3CK /2	TM3CK /4	010	TM3CK /4	TM3CK /8	011	TM3CK /8	TM3CK /16	100	TM3CK /16	TM3CK /32	101	TM3CK /32	TM3CK /64	110	TM3CK /64	TM3CK /128	111	TM3CK /128	TM3CK /256
T3RATE [2:0]	TM3CLK=CPUCLK	TM3CLK=WDT																											
000	TM3CK	TM3CK																											
001	TM3CK /2	TM3CK /4																											
010	TM3CK /4	TM3CK /8																											
011	TM3CK /8	TM3CK /16																											
100	TM3CK /16	TM3CK /32																											
101	TM3CK /32	TM3CK /64																											
110	TM3CK /64	TM3CK /128																											
111	TM3CK /128	TM3CK /256																											
3	T3SEL	时钟源选择 <table border="1"> <thead> <tr> <th>T3SEL</th> <th>定时器 3 时钟源(TM3CK)</th> </tr> </thead> <tbody> <tr><td>0</td><td>CPUCLK</td></tr> <tr><td>1</td><td>内部 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效</td></tr> </tbody> </table>	T3SEL	定时器 3 时钟源(TM3CK)	0	CPUCLK	1	内部 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效																					
T3SEL	定时器 3 时钟源(TM3CK)																												
0	CPUCLK																												
1	内部 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效																												
2	T3RSTB	定时器 3 复位 1: 禁止定时器 3 复位 0: 使能定时器 3 复位 当将该位为 3 时，定时器 3 复位后，T3RSTB 会自动置 1																											
1	T3OUT	PT1.5/PT2.1 口输出控制																											
0	PWM3OUT	<table border="1"> <thead> <tr> <th>T3OUT</th> <th>PWM3OUT</th> <th>PT1.5/PT2.1 输出控制</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>IO 输出</td></tr> <tr><td>0</td><td>1</td><td>蜂鸣器输出</td></tr> <tr><td>1</td><td>0</td><td>PWM3 输出</td></tr> <tr><td>1</td><td>1</td><td>PWM3 取反输出</td></tr> </tbody> </table>	T3OUT	PWM3OUT	PT1.5/PT2.1 输出控制	0	0	IO 输出	0	1	蜂鸣器输出	1	0	PWM3 输出	1	1	PWM3 取反输出												
T3OUT	PWM3OUT	PT1.5/PT2.1 输出控制																											
0	0	IO 输出																											
0	1	蜂鸣器输出																											
1	0	PWM3 输出																											
1	1	PWM3 取反输出																											

2.3.8 UART0CLK

UART0CLK 用于 UART0 模块。UART0CLK 的时钟源来自于内部时钟，分频系数为 2~64。

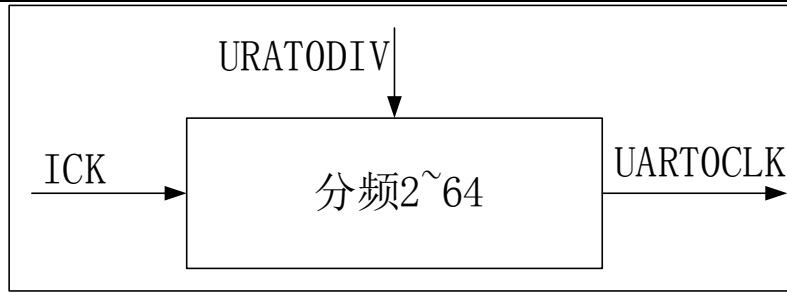


图 17 UART 时钟分频示意图

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
15H	PCK	UART1DIV			UART0DIV			S_beep[1:0]		00000000
14H	ROOT	PROT[7:0]								00000000

PCK 寄存器

位地址	标识符	功能
7:5	UART1DIV	UART1CLK 时钟分频选择，只有当 ROOT_EN =1 时，才能被改写
		UART1DIV[2:0] ICK=8MHz 情况下
		000 ICK /2
		001 ICK /4
		010 ICK /8
		011 ICK /16
		100 ICK /32
		101 ICK /64
		110 ICK /64
4:2	UART0DIV	UART0CLK 时钟分频选择
		UART0DIV[2:0] ICK =8MHz 情况下
		000 ICK /2
		001 ICK /4
		010 ICK /8
		011 ICK /16
		100 ICK /32
		101 ICK /64
		110 ICK /64
111 ICK /64		
1:0	-	-

2.3.9 UART1CLK

UART1CLK 用于 UART1 模块。UART1CLK 的时钟源来自于内部时钟，分频系数为 2~64。

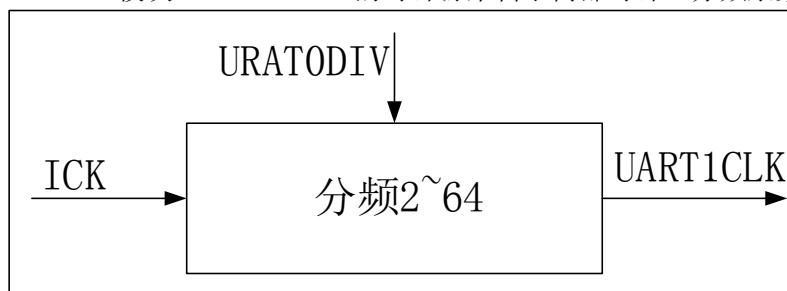


图 18 UART 时钟分频示意图

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
15H	PCK	UART1DIV			UART0DIV			S_bEEP[1:0]		00000000	
14H	ROOT	PROT[7:0]									00000000

PCK 寄存器

位地址	标识符	功能
7:5	UART1DIV	UART1CLK 时钟分频选择, 只有当 ROOT_EN=1 时, 才能被改写
		UART1DIV[2:0] ICK=8MHz 情况下
		000 ICK/2
		001 ICK/4
		010 ICK/8
		011 ICK/16
		100 ICK/32
		101 ICK/64
4:2	UART0DIV	UART0CLK 时钟分频选择
		UART0DIV[2:0] ICK=8MHz 情况下
		000 ICK/2
		001 ICK/4
		010 ICK/8
		011 ICK/16
		100 ICK/32
		101 ICK/64
110 ICK/64		
111 ICK/64		
1:0	-	-

2.3.10 蜂鸣器时钟

表 18 蜂鸣器时钟寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit 3	Bit 2	Bit 1	Bit0	上电复位值
15H	PCK	UART1DIV			UART0DIV			S_BEEP[1:0]		u0000000u
2DH	PTCON	BZEN	PWM3_SEL	PWM2_SEL	PWM1_SEL	E1M[1:0]		E0M[1:0]		00000000

芯片有一个蜂鸣器时钟用于蜂鸣器源, 通过 BZEN 寄存器位使能蜂鸣器功能。用户通过设置 S_BEEP 寄存器标志位来改变蜂鸣时钟, 设置如下图。

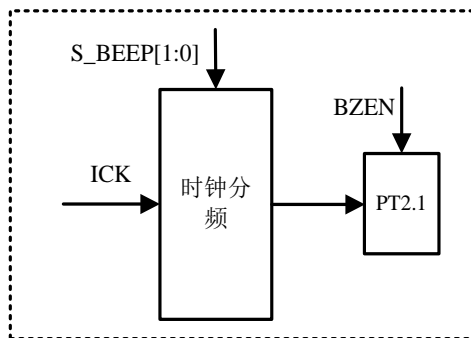


图 19 蜂鸣器时钟选择示意图

表 19 蜂鸣器时钟选择列表

ick_sel[0]	S_BEEP		时钟源(KHz)	BEEP CLOCK(KHz)	
0	0	0	ICK=8000/4000	ICK/512	16/8
0	0	1	ICK=8000/4000	ICK/1024	8/4
0	1	0	ICK=8000/4000	ICK/2048	4/2
0	1	1	ICK=8000/4000	ICK/4096	2/1
1	0	0	ICK=10500/5250	ICK/640	16.4/8.2
1	0	1	ICK=10500/5250	ICK/1280	8.2/4.1
1	1	0	ICK=10500/5250	ICK/2560	4.1/2.05
1	1	1	ICK=10500/5250	ICK/5120	2.05/1.02

ick_sel[1:0]频率选择位

11=10.5MHz

10=8MHz

01=5.25MHz(10.5MHz/2)

00=4MHz(默认值)

2.3.11 SPICLK

芯片支持 SPI 接口, 包括 master 模式和 slave 模式。SPI 通信速率最快为 MCUCLK 频率的三分之一, SPI 速率参见 SPI 章节的寄存器说明: SCR[7:0]。

2.3.12 LCDCLK

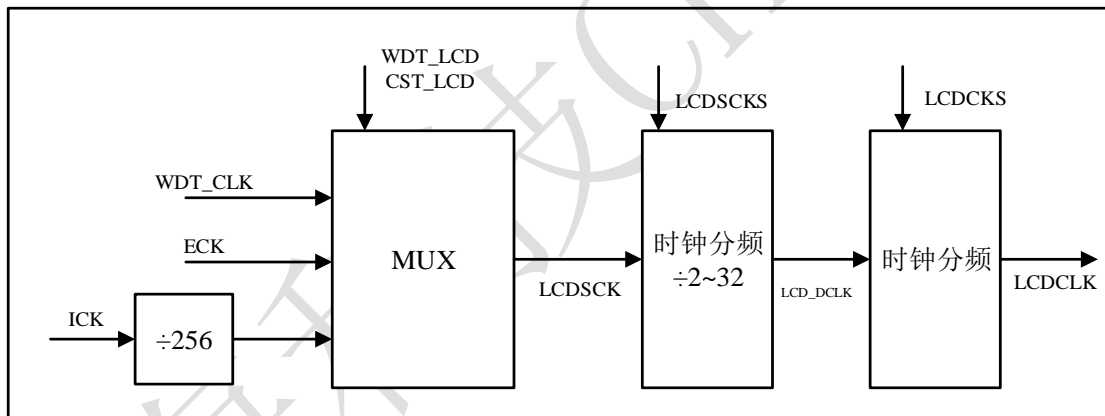


图 20 LCD 帧频时钟选择

参见 3.12.2 LCD 帧频选择

2.3.13 I2CCLK

I2CCLK 用于 I2C 模块。I2C 的时钟源来自于内部时钟, 分频系数为 2~16。

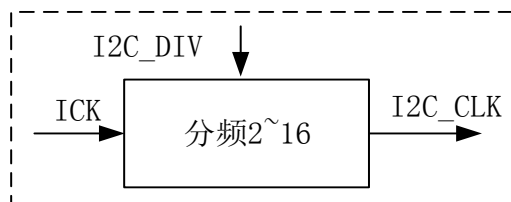


图 21 I2C 时钟分频示意图

地	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复
---	----	------	------	------	------	------	------	------	------	-----

址									位值
0EH	WDTCON	WDTEN	ROOT_EN	AIENB3	I2C_DIV[1:0]		WDTS[2:0]		00100000

WDTCON 寄存器

位地址	标识符	功能	
-	-	-	
-	-	-	
4:3	I2C_DIV	I2C_CLK 时钟分频选择	
		I2C_DIV[1:0]	ICK
		00	ICK /2
		01	ICK /4
		10	ICK /8
		11	ICK /16
-	-	-	

2.4 复位系统

芯片包括以下几种复位方式：

- 1) 上电复位
- 2) 掉电复位
- 3) 低电压检测复位
- 4) 看门狗复位

当上电复位、低电压复位、低电压检测复位发生时，所有的系统寄存器恢复默认状态，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 0000H 处重新开始运行。

当 watch dog 复位发生时，系统寄存器仍然保持原来的值不变，但是此时 PC 指针复位。复位结束后，系统从向量 0000H 处重新开始运行。

注：低电压复位是对低电压比较器产生的信号 LBOUT 进行判断产生。

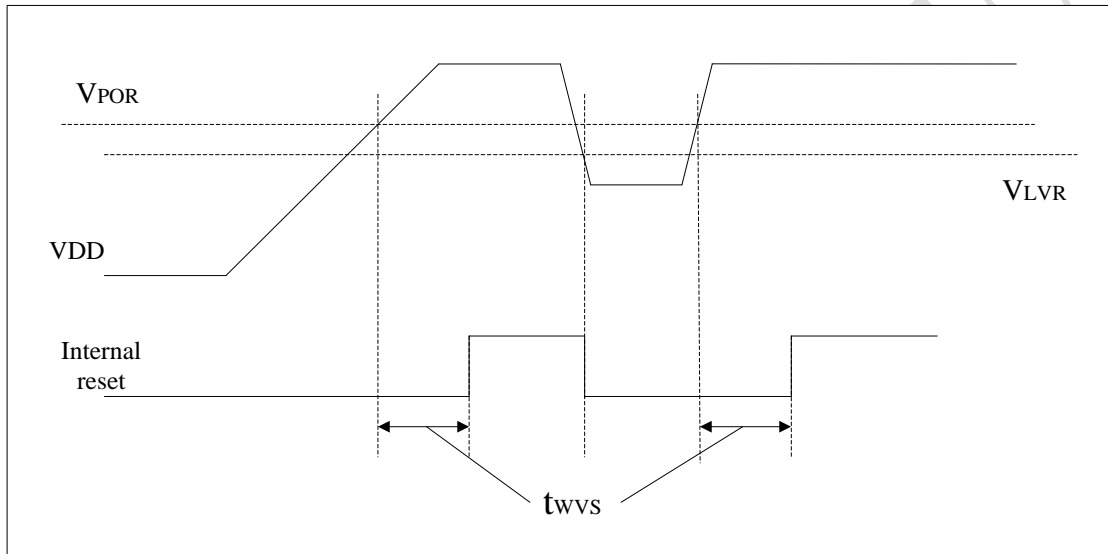


图 22 上电复位电路示例及上电过程

参数	最小值	典型值	最大值
VPOR	-	2.2V	-
VLVR	-	2.0V	-
tWVS	-	39ms	-

VPOR：上电复位

VLVR：低电压复位

tWVS：等待电压稳定时间

2.5 中断

芯片有 14 个中断源：

1) 外部中断 0

PT1.1 或 PT1.2 或 PT1.4 或 PT1.7 或 PT2.1 或 PT2.2 或 PT2.4 或 PT2.7 为外部中断 0 的输入端。触发方式由 PT1MR 寄存器中的 E0M[1:0]寄存器决定。INTE 寄存器中的 E0IE 为外部中断 0 的使能位，INTF 寄存器中的 E0IF 为中断标志位，软件清 0。可唤醒 sleep 或 halt 模式。

PT1MR 寄存器（地址为 2DH）

特性	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PTCON	BZEN	PWM3_SEL	PWM2_SEL	PWM1_SEL	E1M[1:0]		E0M[1:0]	
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3-2 E1M[1:0]: 外部中断 1 触发模式
 11 = 外部中断 1 在状态改变时触发
 10 = 外部中断 1 在状态改变时触发
 01 = 外部中断 1 为上升沿触发
 00 = 外部中断 1 为下降沿触发

Bit 1-0 E0M[1:0]: 外部中断 0 触发模式
 11 = 外部中断 0 在状态改变时触发
 10 = 外部中断 0 在状态改变时触发
 01 = 外部中断 0 为上升沿触发
 00 = 外部中断 0 为下降沿触发

2) 外部中断 1

PT1.0 或 PT1.3 或 PT1.5 或 PT1.6 或 PT2.0 或 PT2.3 或 PT2.5 或 PT2.6 都可作为外部中断 1 的输入端。触发方式由 PT1MR 寄存器中的 E1M[1:0]寄存器决定。INTE 寄存器中的 E1IE 为外部中断 1 的使能位，INTF 寄存器中的 E1IF 为中断标志位，软件清 0。

- 3) SDAD 中断溢出
- 4) 10Bits ADC 中断溢出
- 5) 定时器 0 溢出中断
- 6) 定时器 1 溢出中断
- 7) 定时器 2 溢出中断
- 8) 定时器 3 溢出中断
- 9) UART0 发送中断
- 10) UART0 接收中断
- 11) UART1 发送中断
- 12) UART1 接收中断
- 13) SPI 中断
- 14) RTC 中断

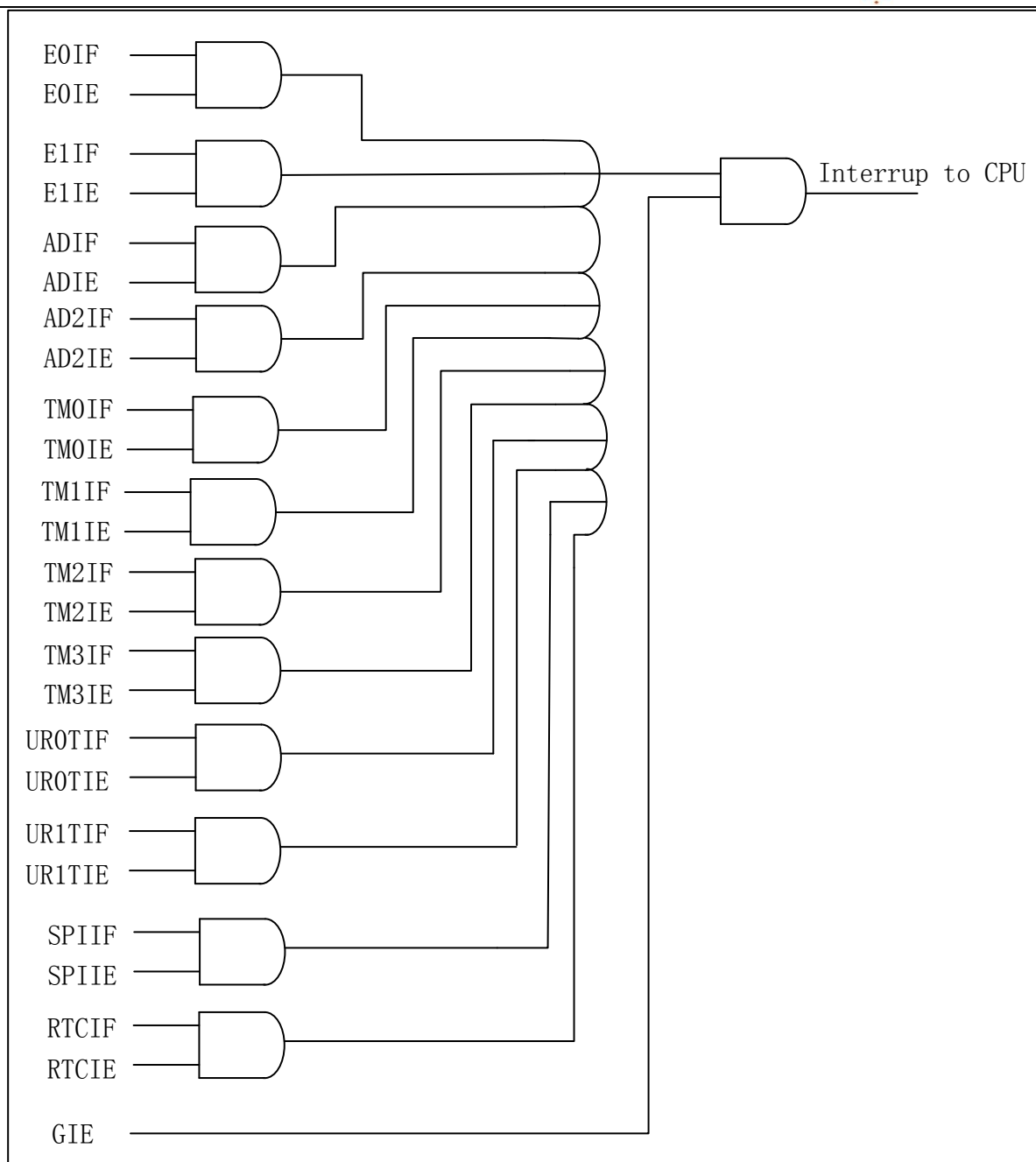


图 23 中断逻辑

芯片有 14 个中断源，但只有 1 个中断入口地址 0x0004。与中断相关的 SFR：中断使能控制寄存器 INTE 和中断标志位寄存器 INTF。这 14 个中断源都各自有一个中断使能，和一个总使能位 GIE，并且它们的标志位硬件置位，软件清 0。

当响应中断时，会把当前的 PC 值入栈保护，并把 PC 置为 0x0004，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。

所有的外部中断、内部中断和 WDT 溢出都可唤醒 sleep 睡眠模式和 halt 停止模式。

2.6 PC 跳转

表 20 PC 跳转寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
12H	PCLOADL	PCLOAD[7:0]								00000000
13H	PCLOADH	-	-	-	PCLOAD[12:8]					uuu00000
14H	ROOT	PROT[7:0]								00000000

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。

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。

2.7 定时器 0

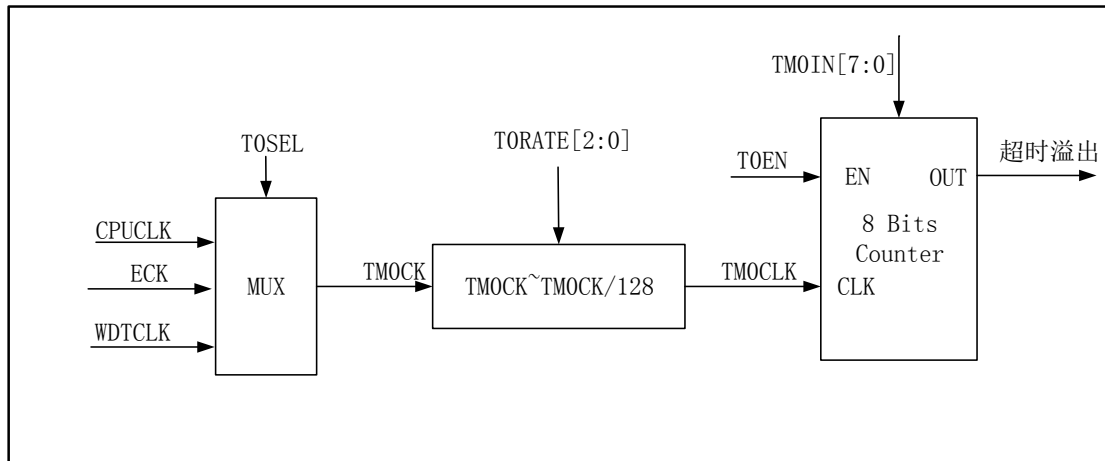


图 24 定时器 0 功能框图

定时器 0 的输入为 CPUCLK、WDTCLK 或者 ECK。在定时器 0 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 00H 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 0004H 以执行中断服务程序。

表 21 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	-	TM2IF	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	u0000000
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	00000000
34H	TM0CON	TOEN	TORATE[2:0]			-	TORSTB	TOSEL		0000u100
35H	TMOIN	TMOIN[7:0]								00000000
36H	TMOCNT	TMOCNT[7:0]								00000000

表 22 TM0CON 寄存器各位功能表

位地址	标识符	功能																											
7	TOEN	定时器 0 使能位 1: 使能定时器 0 0: 禁止定时器 0																											
6:4	TORATE[2:0]	定时器 0 时钟选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>TORATE [2:0]</th> <th>TM0CLK=CPUCLK</th> <th>TM0CLK=WDT/ECK</th> </tr> </thead> <tbody> <tr><td>000</td><td>TM0CK</td><td>TM0CK</td></tr> <tr><td>001</td><td>TM0CK /2</td><td>TM0CK /4</td></tr> <tr><td>010</td><td>TM0CK /4</td><td>TM0CK /8</td></tr> <tr><td>011</td><td>TM0CK /8</td><td>TM0CK /16</td></tr> <tr><td>100</td><td>TM0CK /16</td><td>TM0CK /32</td></tr> <tr><td>101</td><td>TM0CK /32</td><td>TM0CK /64</td></tr> <tr><td>110</td><td>TM0CK /64</td><td>TM0CK /128</td></tr> <tr><td>111</td><td>TM0CK /128</td><td>TM0CK /256</td></tr> </tbody> </table>	TORATE [2:0]	TM0CLK=CPUCLK	TM0CLK=WDT/ECK	000	TM0CK	TM0CK	001	TM0CK /2	TM0CK /4	010	TM0CK /4	TM0CK /8	011	TM0CK /8	TM0CK /16	100	TM0CK /16	TM0CK /32	101	TM0CK /32	TM0CK /64	110	TM0CK /64	TM0CK /128	111	TM0CK /128	TM0CK /256
TORATE [2:0]	TM0CLK=CPUCLK	TM0CLK=WDT/ECK																											
000	TM0CK	TM0CK																											
001	TM0CK /2	TM0CK /4																											
010	TM0CK /4	TM0CK /8																											
011	TM0CK /8	TM0CK /16																											
100	TM0CK /16	TM0CK /32																											
101	TM0CK /32	TM0CK /64																											
110	TM0CK /64	TM0CK /128																											
111	TM0CK /128	TM0CK /256																											
3	NC	保留位																											
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	时钟源选择																											

		T0SEL[1:0]	定时器 0 时钟源(TM0CK)
		00	CPUCLK
		01	CPUCLK
		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 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效

表 23 TM0IN 寄存器各位功能表

位地址	标识符	功能
7 : 0	TM0IN[7:0]	定时器 0 溢出值

表 24 TM0CNT 寄存器各位功能表

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

操作：

设置 TM0CLK，为定时器 0 选择输入。

设置 TM0IN，选择定时器 0 溢出值。

设置寄存器标志位：TM0IE 与 GIE，使能定时器 0 中断。

设置寄存器标志位：TM0EN，使能定时器 0 的 8 bits 计数器。

清零寄存器标志位：TORSTB，复位定时器 0 的计数器（复位的持续时间为 1.5*TM0CK，注意不是 TM0CLK）。

当定时超时发生时，寄存器标志位 TM0IF 会自复位，程序计数器会复位为 0004H。

定时器 0 溢出时间计算方法：

定时器 0 溢出时间= (TM0IN+1) /TM0CLK

2.8 I/O PORT

芯片包括 32 个双向 IO 口。所有 IO 口都可以配置上拉。

其他 IO 特性：

IO 其他功能复用见相关功能模块描述。

表 25 I/O 口寄存器表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
1DH	PT1	PT1[7:0]								xxxxxxxx
1EH	PT1EN	PT1EN[7:0]								00000000
1FH	PT1PU	PT1UP[7:0]								00000000
20H	AIENB	AIENB2[7:0]								11111111
21H	PT2	PT2[7:0]								xxxxxxxx
22H	PT2EN	PT2EN[7:0]								00000000
23H	PT2PU	PT2PU[7:0]								00000000
24H	PT3	PT3[7:0]								xxxxxxxx
25H	PT3EN	PT3EN[7:0]								00000000
26H	PT3PU	PT3PU[7:0]								00000000
27H	PT4	PT4[7:0]								xxxxxxxx
28H	PT4EN	PT4EN[7:0]								00000000
29H	PT4PU	PT4PU[7:0]								00000000
2AH	PT5	-	-	-	PT5[4:0]					000xxxxx
2BH	PT5EN	SMT5_EN	SMT2_EN	SMT1_EN	PT5EN[4:0]					11100000
2CH	PT5PU	PT15_VDD	PT14_VDD	I2C_VDD	PT5PU[4:0]					00000000
2DH	PTCON	BZEN	PWM3_SEL	PWM2_SEL	PWM1_SEL	E1M[1:0]	E0M[1:0]			00000000
2EH	PTINT0	PTW0[7:0]								00000000
2FH	PTINT1	PTW1[7:0]								00000000
30H	SEGCON0	SEGCON0[7:0]								00000000
31H	SEGCON1	SEGCON1[7:0]								00000000
0EH	WDTCN	WDTEN	ROOT_EN	AIENB3	I2C_DIV[1:0]		WDTS[2:0]			00100000
1BH	LVDCN	LVDEN	AD2_REF	AIENB1	SILB[2:0]			LBOUT	LB_RST_CON	001000x0
6BH	ADCON	METCH[3:0]				ADM[3:0]				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。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。

2.8.1 PT1 寄存器（地址为 1DH）

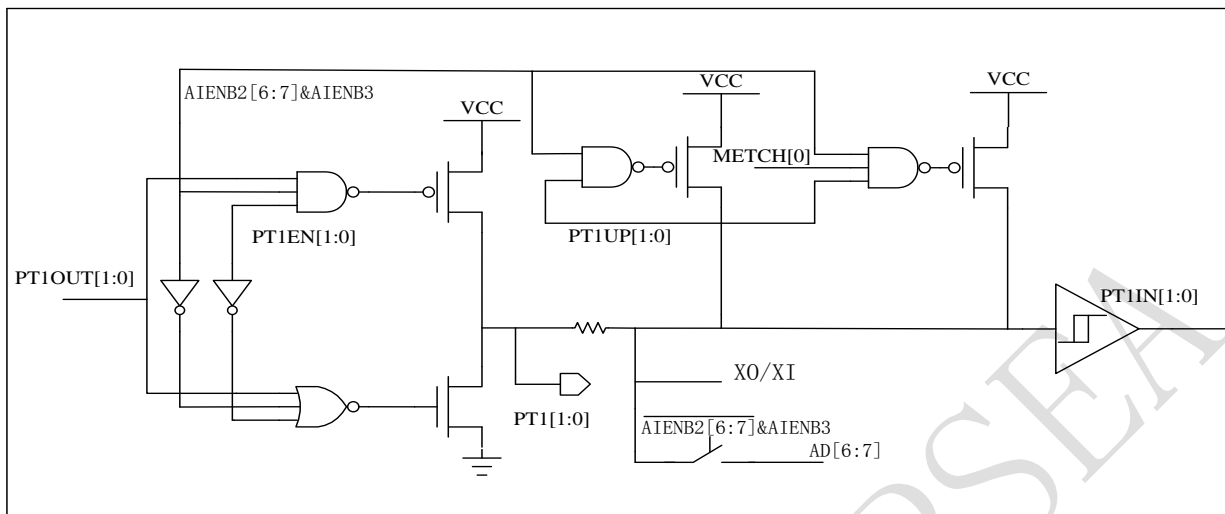


图 25 PT1[1:0] 功能框图

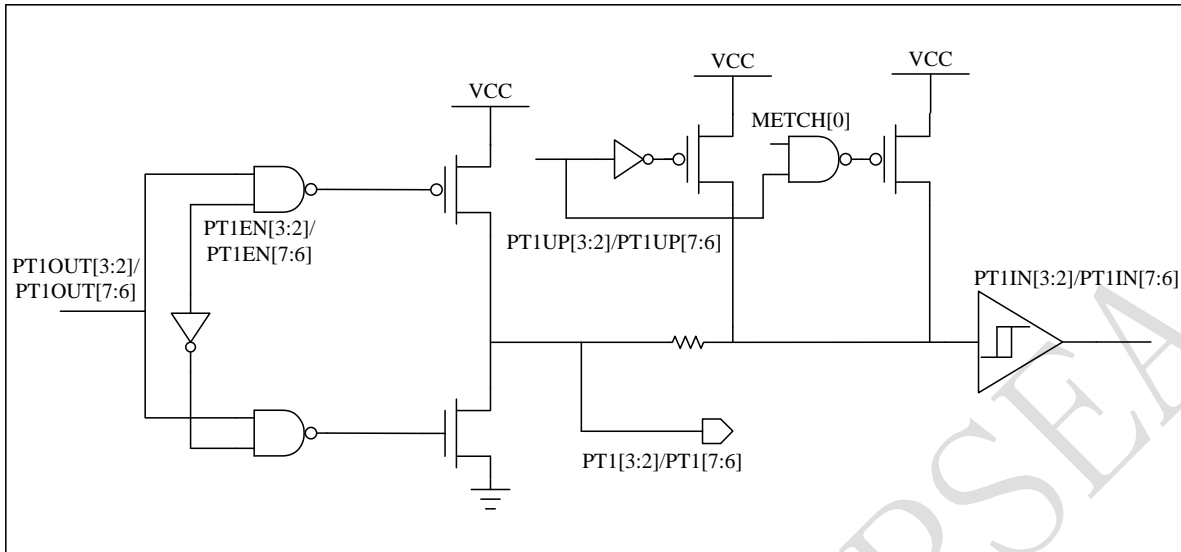


图 26 PT1[3:2]/ PT1[7:6] 功能框图

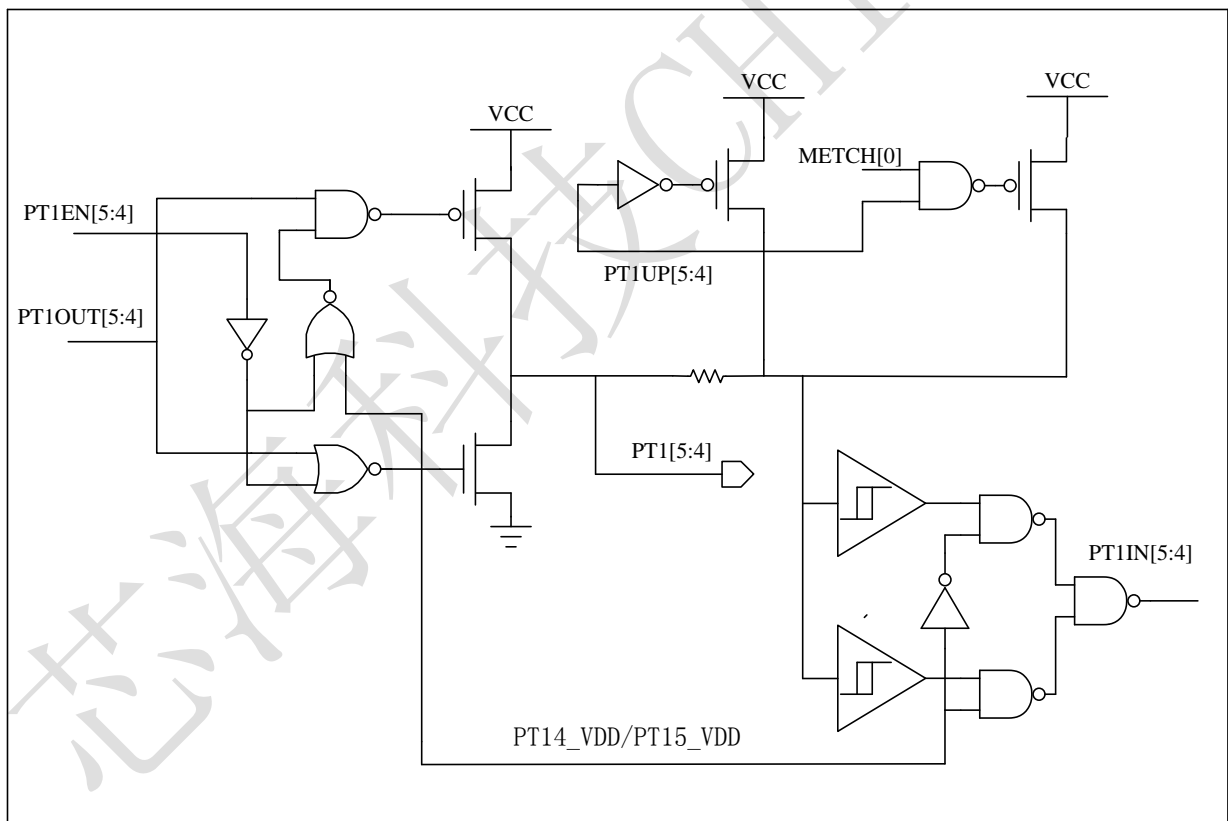


图 27 PT1[5:4] 功能框图

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 1EH)

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, 默认为输入口

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 1FH)

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
PT1UP	PT1UP[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, 默认断开上拉电阻

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

上拉电阻选择 10K 欧或者 30K 欧, 由 METCH[0] 控制, METCH[0]=0 上拉选择 30K 欧, METCH[0]=1 上拉选择 10K 欧

AIENB 寄存器 (地址为 20H)

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
AIENB	AIENB2[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 AIENB2[7]: PT1.0 数模接口定义, 默认为数字口

0 = 并且 AIENB3=1 时为 ADC 模拟输入
 1 = 并且 AIENB3=1 为数字接口

Bit 6 AIENB2[6]: PT1.1 数模接口定义, 默认为数字口

0 = 并且 AIENB3=1 时为 ADC 模拟输入
 1 = 并且 AIENB3=1 为数字接口

Bit 5 AIENB2[5]: PT2.0 数模接口定义, 默认为数字口

- 0 = PT2.0 定义为模拟接口
- 1 = PT2.0 定义为数字接口
- Bit 4 AIENB2[4]: PT2.1 数模接口定义, 默认为数字口
 - 0 = PT2.1 定义为模拟接口
 - 1 = PT2.1 定义为数字接口
- Bit 3 AIENB2[3]: PT2.2 数模接口定义, 默认为数字口
 - 0 = PT2.2 定义为模拟接口
 - 1 = PT2.2 定义为数字接口
- Bit 2 AIENB2[2]: PT2.3 数模接口定义, 默认为数字口
 - 0 = PT2.3 定义为模拟接口
 - 1 = PT2.3 定义为数字接口
- Bit 1 AIENB2[1]: PT2.6 数模接口定义, 默认为数字口
 - 0 = PT2.6 定义为模拟接口
 - 1 = PT2.6 定义为数字接口

当串口 1 选择 PT2.6 作为串口 1 的 RX; PT2.7 作为串口 1 的 TX 串口时候 (UART1_SEL=1), 只有当 ROOT_EN=1, 才能改变此项。

- Bit 0 AIENB2[0]: PT2.7 数模接口定义, 默认为数字口
 - 0 = PT2.7 定义为模拟接口
 - 1 = PT2.7 定义为数字接口

当串口 1 选择 PT2.6 作为串口 1 的 RX; PT2.7 作为串口 1 的 TX 串口时候 (UART1_SEL=1), 只有当 ROOT_EN=1, 才能改变此项。

串口通信 1 UART1_SEL 接口选择:

- 0: PT1.7 作为串口 1 的 RX; PT1.6 作为串口 1 的 TX
- 1: PT2.6 作为串口 1 的 RX; PT2.7 作为串口 1 的 TX

WDTCN (地址为 0EH)

特性	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WDTCN	WDTEN	ROOT_EN	AIENB3	I2C_DIV[1:0]		WDTS[2:0]		
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 5 AIENB3: PT1.0/1.1 数模接口定义, 默认为数字口
 - 0 = PT1.0/1.1 定义为模拟接口, 外部晶振输入
 - 1 = PT1.0/1.1 定义为数字接口。设置 PT1.0 为数字口, 需要设置 AIENB2[7]=1; 设置 PT1.1 为数字口, 需要设置 AIENB2[6]=1

LVDCN (地址为 1BH)

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
LVDCN	LVDEN	AD2_REF	AIENB1	SILB[2:0]			LBOUT	LB_RST_CON
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 5 AIENB1: PT2.0 数模接口定义, 默认为数字口
 - 0 = PT2.0 定义为模拟接口, 低电压检测输入口
 - 1 = PT20 定义为数字接口

2.8.2 PT2 寄存器（地址为 21H）

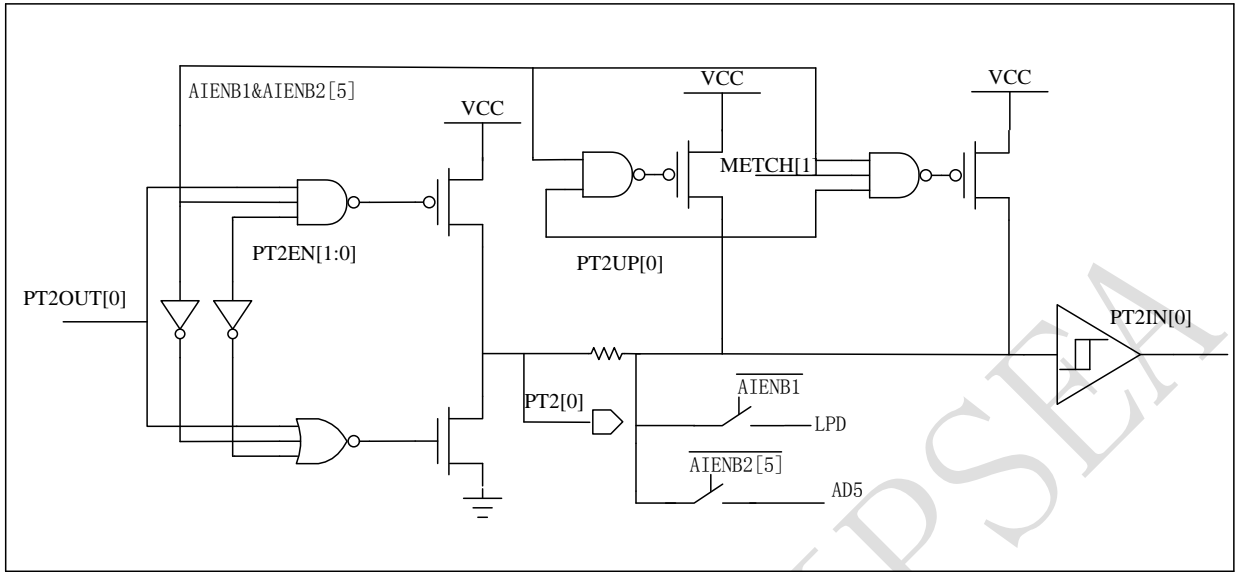


图 28 PT2[0] 功能框图

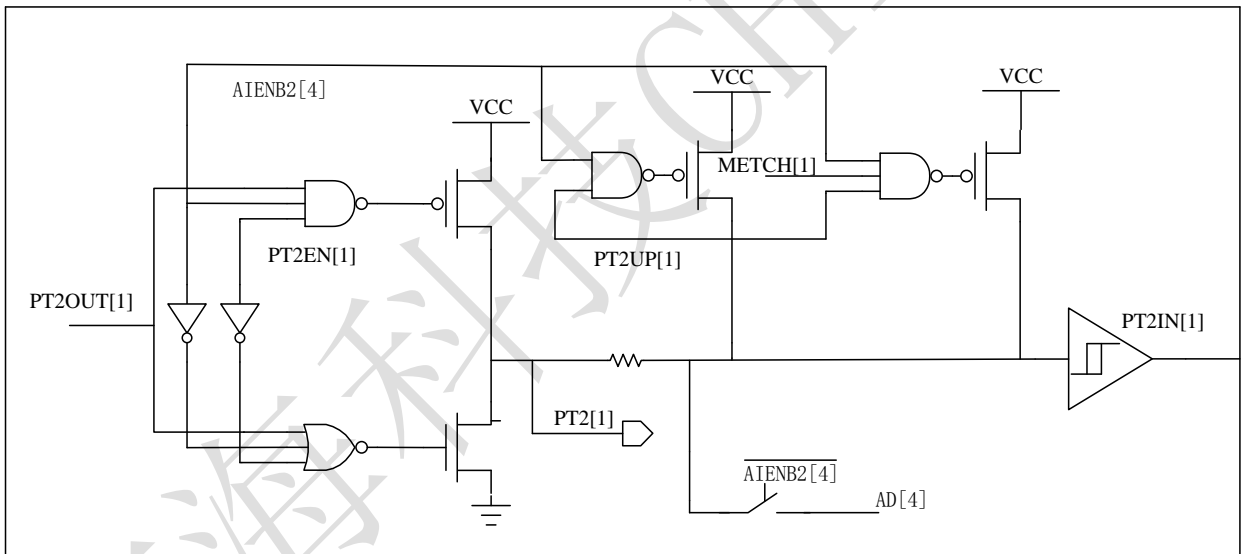


图 29 PT2[1] 功能框图

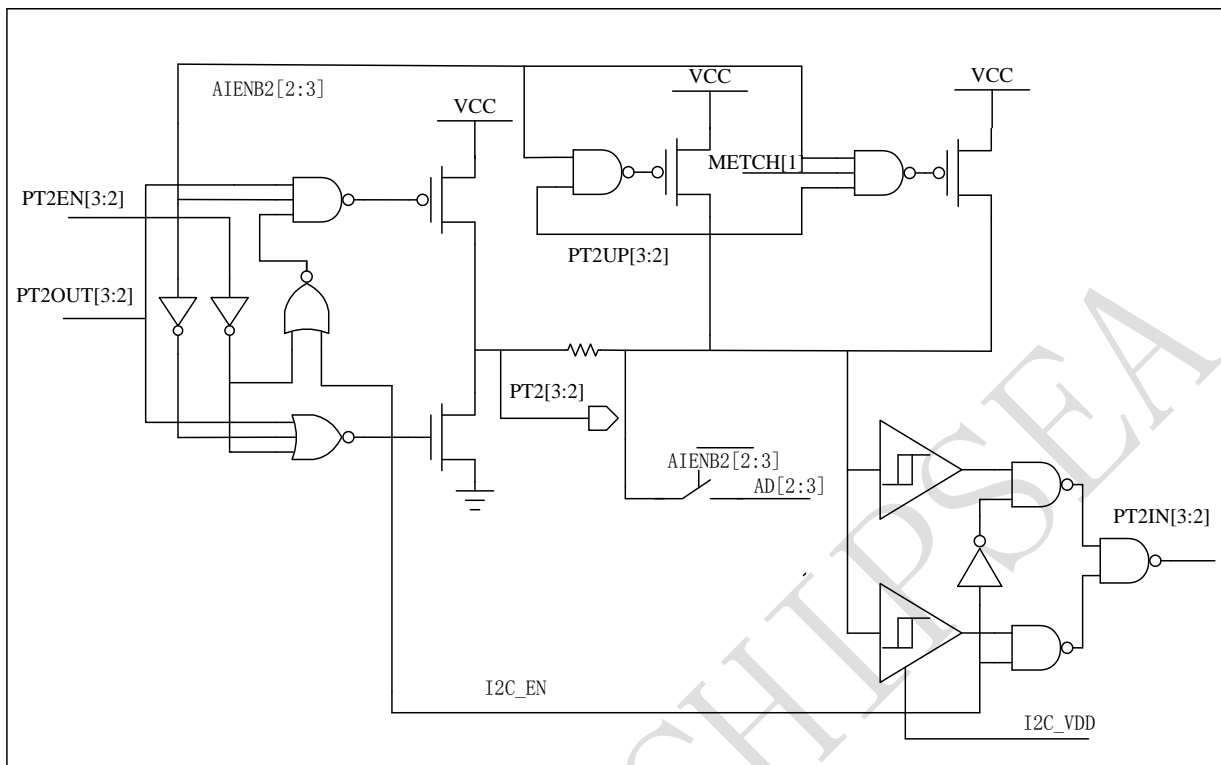


图 30 PT2[3:2] 功能框图

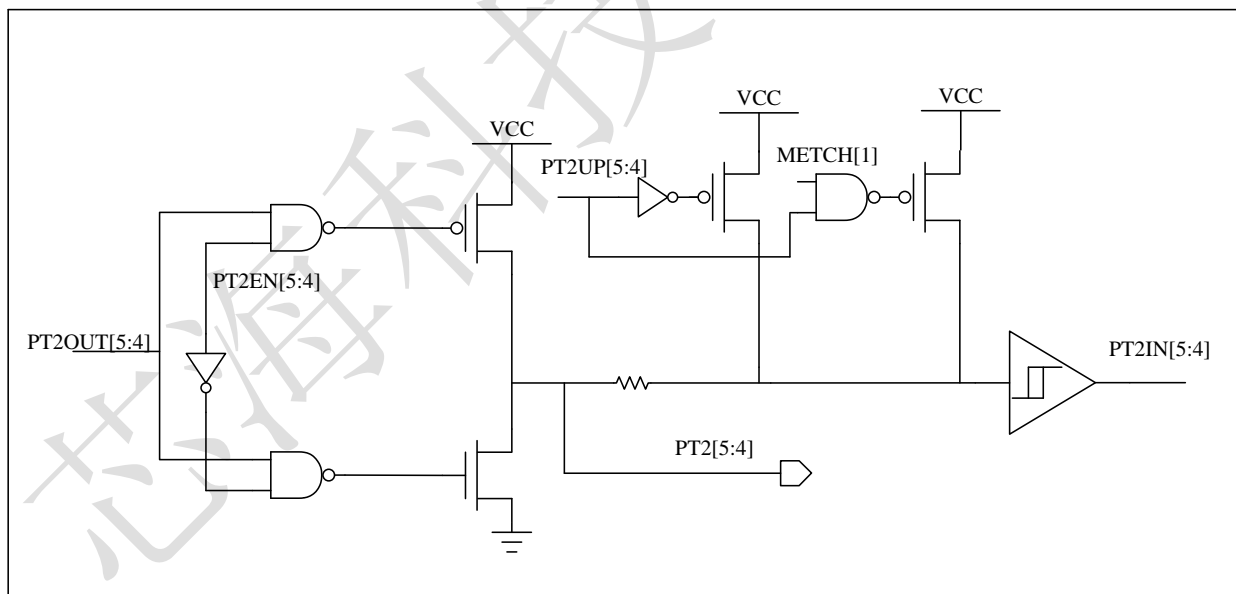


图 31 PT2[5:4] 功能框图

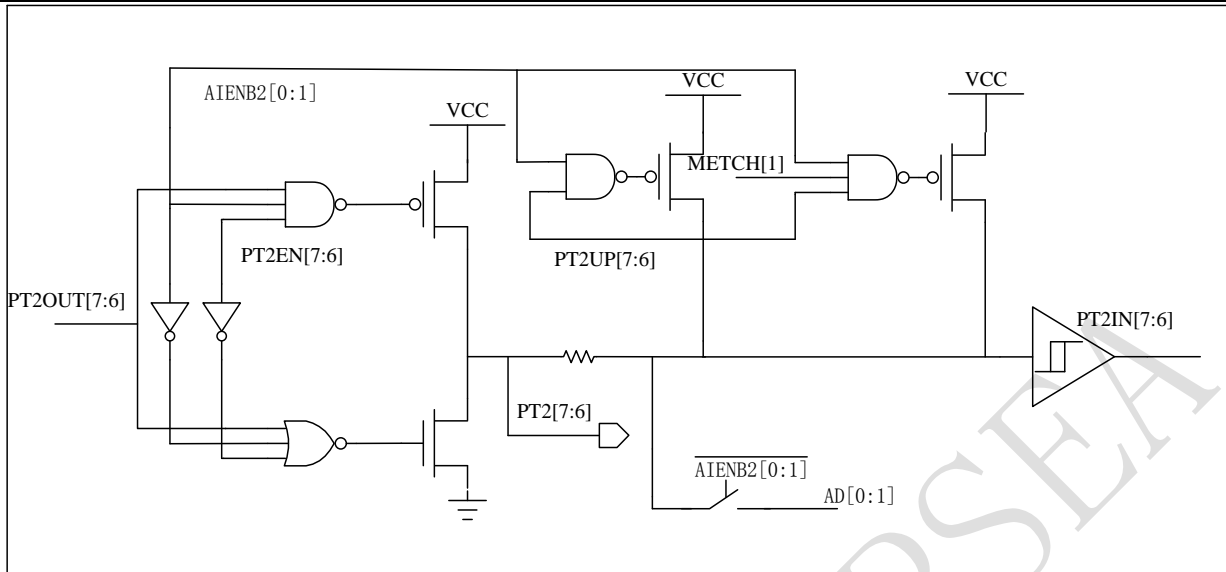


图 32 PT2[7:6] 功能框图

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
PT2	PT2[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7:0 PT2[7:0]: 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
- PT2[5] = GPIO2 bit 5 的数据标志位
- PT2[4] = GPIO2 bit 4 的数据标志位
- PT2[3] = GPIO2 bit 3 的数据标志位
- PT2[2] = GPIO2 bit 2 的数据标志位
- PT2[1] = GPIO2 bit 1 的数据标志位
- PT2[0] = GPIO2 bit 0 的数据标志位

PT2EN 寄存器 (地址为 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
PT2EN	PT2EN[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7:0 PT2EN[7:0]: GPIO2 口输入/输出控制标志, 默认为输入口

- 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
- PT2EN[5] = GPIO2 bit 5 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[4] = GPIO2 bit 4 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[3] = GPIO2 bit 3 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[2] = GPIO2 bit 2 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[1] = GPIO2 bit 1 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
- PT2EN[0] = GPIO2 bit 0 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT2PU 寄存器 (地址为 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
PT2PU	PT2PU[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7:0 PT2PU[7:0]: GPIO2 口上拉电阻使能标志, 默认断开上拉电阻
 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
 PT2PU[5] = GPIO2 bit 5 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT2PU[4] = GPIO2 bit 4 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT2PU[3] = GPIO2 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT2PU[2] = GPIO2 bit 2 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT2PU[1] = GPIO2 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT2PU[0] = GPIO2 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

上拉电阻选择 10K 欧或者 30K 欧, 由 METCH[1] 控制, METCH[1]=0 上拉选择 30K 欧, METCH[1]=1 上拉选择 10K 欧姆。

2.8.3 PT3 寄存器 (地址为 24H)

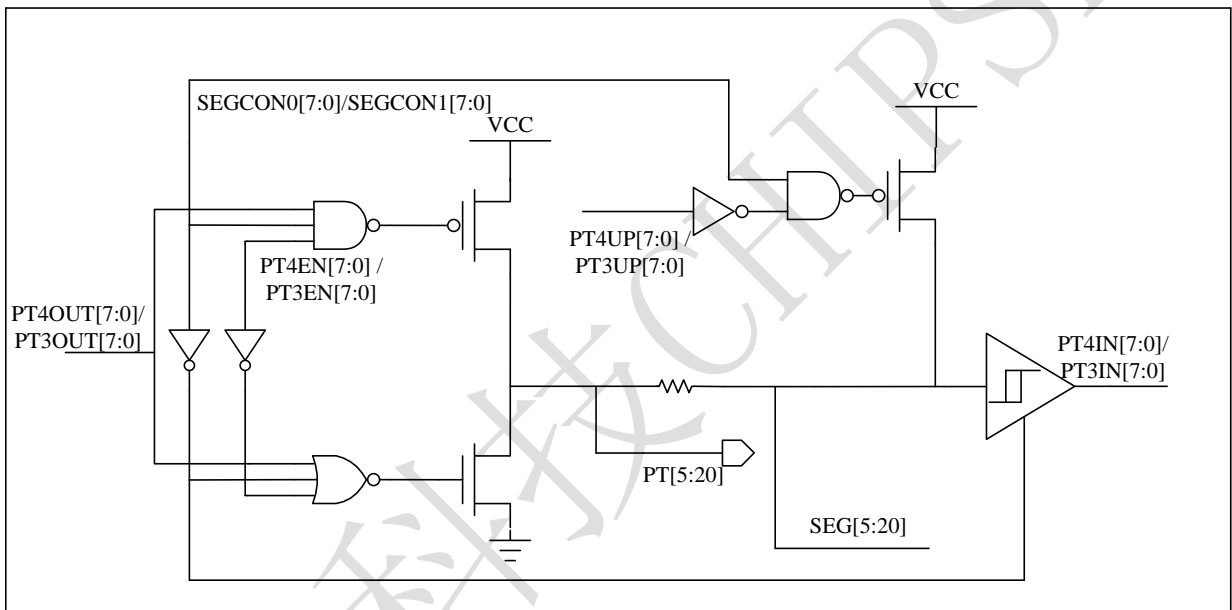


图 33 PT3[7:0]/ PT4[7:0]功能框图

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 25H)

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]: GPIO3 口输入/输出控制标志, 默认为输入口

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 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
PT3PU	PT3PU[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7:0 PT3PU[7:0]: GPIO3 口上拉电阻使能标志, 默认断开上拉电阻

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

2.8.4 PT4 寄存器 (地址为 27H)

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
PT4	PT4[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7:0 PT4[7:0]: GPIO4 口数据标志位

PT4[7] = GPIO4 bit 7 的数据标志位

PT4[6] = GPIO4 bit 6 的数据标志位

PT4[5] = GPIO4 bit 5 的数据标志位

PT4[4] = GPIO4 bit 4 的数据标志位

PT4[3] = GPIO4 bit 3 的数据标志位

PT4[2] = GPIO4 bit 2 的数据标志位

PT4[1] = GPIO4 bit 1 的数据标志位

PT4[0] = GPIO4 bit 0 的数据标志位

PT4EN 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
PT4EN	PT4EN[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7:0 PT4EN[7:0]: GPIO4 口输入/输出控制标志, 默认为输入口

PT4EN[7] = GPIO4 bit 7 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT4EN[6] = GPIO4 bit 6 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT4EN[5] = GPIO4 bit 5 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT4EN[4] = GPIO4 bit 4 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
 PT4EN[3] = GPIO4 bit 3 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
 PT4EN[2] = GPIO4 bit 2 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
 PT4EN[1] = GPIO4 bit 1 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口
 PT4EN[0] = GPIO4 bit 0 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT4PU 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
PT4PU	PT4PU[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7:0 PT4PU[7:0]: GPIO4 口上拉电阻使能标志, 默认断开上拉电阻
 PT4PU[7] = GPIO4 bit 7 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT4PU[6] = GPIO4 bit 6 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT4PU[5] = GPIO4 bit 5 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT4PU[4] = GPIO4 bit 4 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT4PU[3] = GPIO4 bit 3 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT4PU[2] = GPIO4 bit 2 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT4PU[1] = GPIO4 bit 1 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻
 PT4PU[0] = GPIO4 bit 0 控制标志位; 0 = 断开上拉电阻, 1 = 使用上拉电阻

2.8.5 PT5 寄存器 (地址为 2AH)

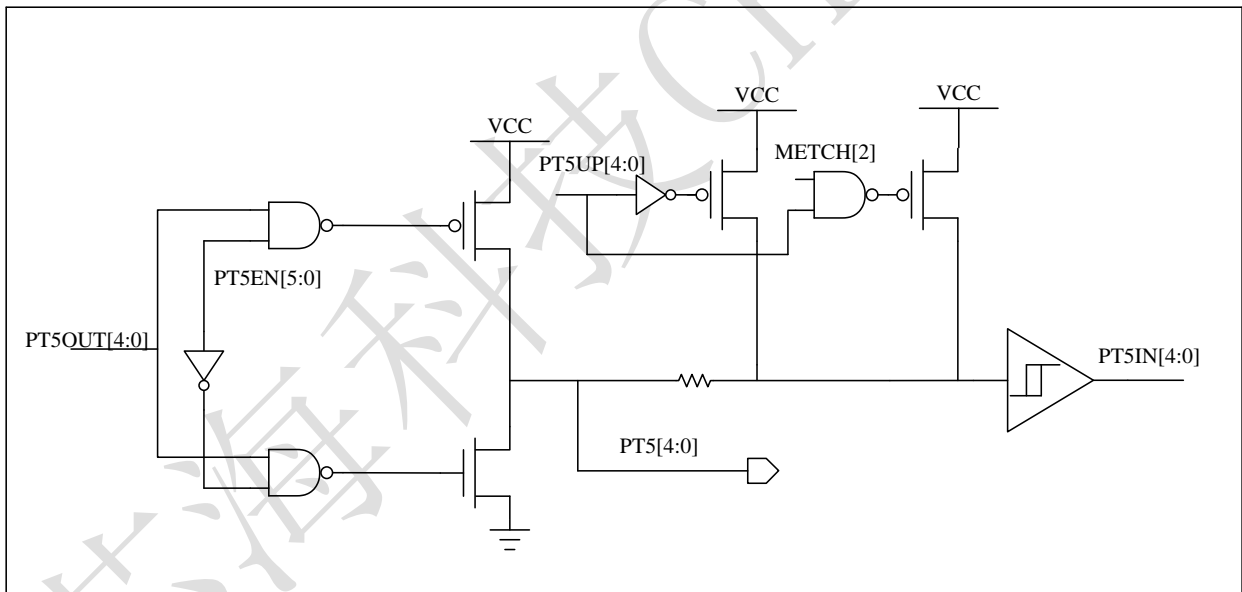


图 34 PT5[4:0]功能框图

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

Bit 4:0 PT5[4:0]: GPIO5 口数据标志位
 PT5[4] = GPIO5 bit 4 的数据标志位
 PT5[3] = GPIO5 bit 3 的数据标志位
 PT5[2] = GPIO5 bit 2 的数据标志位
 PT5[1] = GPIO5 bit 1 的数据标志位
 PT5[0] = GPIO5 bit 0 的数据标志位

PT5EN 寄存器（地址为 2BH）

特性	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT5EN	SMT5_EN	SMT2_EN	SMT1_EN	PT5EN[4:0]				
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 SMT5_EN, PT5.0~PT5.4 输入施密特使能控制信号

0 = PT5.0~PT5.4 输入施密特使能关闭

1 = PT5.0~PT5.4 输入施密特使能打开, 默认打开

Bit6 SMT2_EN, PT2.0~PT2.7 输入施密特使能控制信号

0 = PT2.0~PT2.7 输入施密特使能关闭

1 = PT2.0~PT2.7 输入施密特使能打开, 默认打开

Bit 5 SMT1_EN, PT1.0~PT1.7 输入施密特使能控制信号

0 = PT1.0~PT1.7 输入施密特使能关闭

1 = PT1.0~PT1.7 输入施密特使能打开, 默认打开

Bit4:0 PT5EN[4:0]: GPIO5 口输入/输出控制标志, 默认为输入口

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 2CH）

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	PT15_VDD	PT14_VDD	I2C_VDD	PT5PU[4:0]				
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7: PT15_VDD: PT1.5 与 1.8V 芯片通信时, 需要设置 PT15_VDD=1

Bit6: PT14_VDD: PT1.4 与 1.8V 芯片通信时, 需要设置 PT14_VDD=1

Bit5: I2C_VDD: 当外部 I2C 工作电压为 1.8V 通信时, 需要设置 I2C_VDD=1。

Bit 4:0 PT5PU[4:0]: GPIO5 口上拉电阻使能标志, 默认断开上拉电阻

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

上拉电阻选择 10K 欧或者 30K 欧, 由 METCH[2] 控制, METCH[2]=0 上拉选择 30K 欧, METCH[2]=1 上拉选择 10K 欧

2.8.6 PTCON 寄存器（地址为 2DH）

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
PTCON	BZEN	PWM3_SEL	PWM2_SEL	PWM1_SEL	E1M[1:0]		E0M[1:0]	
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 BZEN: 蜂鸣器使能标志, 默认不使能

1 = 使能蜂鸣器功能, PT2.1 为蜂鸣器输出接口

0 = 不使能蜂鸣器功能, PT2.1 定义普通用途 I/O 口

Bit 6 PWM3_SEL PT1.5/PT2.1 口输出选择信号

1: Timer3 的输出选择 PT2.1

0: Timer3 输出选择 PT1.5

Bit 5 PWM2_SEL PT1.6/PT2.5 口输出选择信号

1: Timer2 的输出选择 PT2.5

0: Timer2 的输出选择 PT1.6

- Bit 4 PWM1_SEL PT1.7/PT2.4 口输出选择信号
1: Timer1 的输出选择 PT2.4
0: Timer1 的输出选择 PT1.7
- Bit 3-2 E1M[1:0]: GPIO2 口 bit 1 中断触发模式
11 = 外部中断 1 在状态改变时触发
10 = 外部中断 1 在状态改变时触发
01 = 外部中断 1 为上升沿触发
00 = 外部中断 1 为下降沿触发
- Bit 1-0 E0M[1:0]: GPIO2 口 bit 0 中断触发模式
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

PTINT0 寄存器 (地址为 2EH)

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
PTINT	PTW0[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

此寄存器 PTW0[7:0] 只能 1 个 Bit 为 1, 如果存在两个 Bit 以上为 1, 那么外部中断 0 选择高位为 1 对应的 PAD 位置。

- Bit 7 PTW0[7]: PT2.7 外部中断 0 使能, 默认为 0
0 = 禁止 PT2.7 外部中断 0
1 = 使能 PT2.7 外部中断 0
- Bit 6 PTW0[6]: PT2.4 外部中断 0 使能, 默认为 0
0 = 禁止 PT2.4 外部中断 0
1 = 使能 PT2.4 外部中断 0
- Bit 5 PTW0[5]: PT2.2 外部中断 0 使能, 默认为 0
0 = 禁止 PT2.2 外部中断 0
1 = 使能 PT2.2 外部中断 0
- Bit 4 PTW0[4]: PT2.1 外部中断 0 使能, 默认为 0
0 = 禁止 PT2.1 外部中断 0
1 = 使能 PT2.1 外部中断 0
- Bit 3 PTW0[3]: PT1.7 外部中断 0 使能, 默认为 0
0 = 禁止 PT1.7 外部中断 0
1 = 使能 PT1.7 外部中断 0
- Bit 2 PTW0[2]: PT1.4 外部中断 0 使能, 默认为 0
0 = 禁止 PT1.4 外部中断 0
1 = 使能 PT1.4 外部中断 0
- Bit 1 PTW0[1]: PT1.2 外部中断 0 使能, 默认为 0
0 = 禁止 PT1.2 外部中断 0
1 = 使能 PT1.2 外部中断 0
- Bit 0 PTW0[0]: PT1.1 外部中断 0 使能, 默认为 0
0 = 禁止 PT1.1 外部中断 0
1 = 使能 PT1.1 外部中断 0

PTINT1 寄存器 (地址为 2FH)

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
PTINT	PTW1[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

此寄存器 PTW1[7:0] 只能 1 个 Bit 为 1，如果存在两个 Bit 以上为 1，那么外部中断 1 选择高位为 1 对应的 PAD 位置。

- Bit 7 PTW1[7]: PT2.6 外部中断 1 使能，默认为 0
 0 = 禁止 PT2.6 外部中断 1
 1 = 使能 PT2.6 外部中断 1
- Bit 6 PTW1[6]: PT2.5 外部中断 1 使能，默认为 0
 0 = 禁止 PT2.5 外部中断 1
 1 = 使能 PT2.5 外部中断 1
- Bit 5 PTW1[5]: PT2.3 外部中断 1 使能，默认为 0
 0 = 禁止 PT2.3 外部中断 1
 1 = 使能 PT2.3 外部中断 1
- Bit 4 PTW1[4]: PT2.0 外部中断 1 使能，默认为 0
 0 = 禁止 PT2.0 外部中断 1
 1 = 使能 PT2.0 外部中断 1
- Bit 3 PTW1[3]: PT1.6 外部中断 1 使能，默认为 0
 0 = 禁止 PT1.6 外部中断 1
 1 = 使能 PT1.6 外部中断 1
- Bit 2 PTW1[2]: PT1.5 外部中断 1 使能，默认为 0
 0 = 禁止 PT1.5 外部中断 1
 1 = 使能 PT1.5 外部中断 1
- Bit 1 PTW1[1]: PT1.3 外部中断 1 使能，默认为 0
 0 = 禁止 PT1.3 外部中断 1
 1 = 使能 PT1.3 外部中断 1
- Bit 0 PTW1[0]: PT1.0 外部中断 1 使能，默认为 0
 0 = 禁止 PT1.0 外部中断 1
 1 = 使能 PT1.0 外部中断 1

SEGCON0 寄存器（地址为 30H）

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
SEGCON0	SEGCON0[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Bit 7	SEGCON0 [7]: PT4.7 LCD SEG 功能使能 0 = 禁止 LCD SEG 功能（默认值） 1 = 使能 LCD SEG 功能							
Bit 6	SEGCON0 [6]: PT4.6 LCD SEG 功能使能 0 = 禁止 LCD SEG 功能（默认值） 1 = 使能 LCD SEG 功能							
Bit 5	SEGCON0 [5]: PT4.25 LCD SEG 功能使能 0 = 禁止 LCD SEG 功能（默认值） 1 = 使能 LCD SEG 功能							
Bit 4	SEGCON0 [4]: PT4.4 LCD SEG 功能使能 0 = 禁止 LCD SEG 功能（默认值） 1 = 使能 LCD SEG 功能							
Bit 3	SEGCON0 [3]: PT4.3 LCD SEG 功能使能 0 = 禁止 LCD SEG 功能（默认值） 1 = 使能 LCD SEG 功能							
Bit 2	SEGCON0 [2]: PT4.2 LCD SEG 功能使能 0 = 禁止 LCD SEG 功能（默认值） 1 = 使能 LCD SEG 功能							
Bit 1	SEGCON0 [1]: PT4.1 LCD SEG 功能使能							

- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 0 SEGCON0 [0]: PT4.0 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能

SEGCON1 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
SEGCON1	SEGCON1[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7 SEGCON1 [7]: PT3.7 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 6 SEGCON1 [6]: PT3.6 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 5 SEGCON1 [5]: PT3.5 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 4 SEGCON1 [4]: PT3.4 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 3 SEGCON1 [3]: PT3.3 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 2 SEGCON1 [2]: PT3.2 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 1 SEGCON1 [1]: PT3.1 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能
- Bit 0 SEGCON1 [0]: PT3.0 LCD SEG 功能使能
- 0 = 禁止 LCD SEG 功能 (默认值)
- 1 = 使能 LCD SEG 功能

3 增强功能

3.1 Halt 和 Sleep 模式

为了使芯片处于待机状态，可以让 CPU 停止工作使芯片进行停止或进入睡眠模式，减小功耗。这两种模式描述如下：

停止模式

CPU 执行停止指令(Halt)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。在进入 sleep 时，必须手动关闭所有功能模块的使能信号，才能保证功耗达到 1.5uA 以下。

停止模式下可以唤醒的方式有：

- 1) 外部中断 0
- 2) 外部中断 1
- 3) SDAD 中断溢出
- 4) 10Bits ADC 中断溢出
- 5) 定时器 0 溢出中断
- 6) 定时器 1 溢出中断
- 7) 定时器 2 溢出中断
- 8) 定时器 3 溢出中断
- 9) UART0 发送中断
- 10) UART0 接收中断
- 11) UART1 发送中断
- 12) UART1 接收中断
- 13) SPI 中断
- 14) RTC 中断

睡眠模式

CPU 执行睡眠指令(Sleep)后，内置 8M 振荡器停止工作(WDT 时钟不受影响)直到出现一个外部中断指令唤醒 CPU。为了避免由中断返回(Interrupt Return)引起的程序错误，建议睡眠指令之后加一 NOP 指令以保证程序的正常运行。在睡眠模式下的功耗大约有 1uA。

为了保证 CPU 在睡眠模式下的功耗最小，在执行睡眠指令之前，需要把 IO 口的上拉电阻断开，并且保证所有的 I/O 口是接到 DVDD 或 DGND 电平。(修改)

睡眠模式下可以唤醒的方式有：

- 1) 外部中断 0
- 2) 外部中断 1
- 3) 定时器中断 0 (使用 WDT/ECK 时钟时)
- 4) 定时器中断 3 (使用 WDT 时钟时)
- 5) RTC 中断
- 6) UART 0 唤醒中断(进入 Sleep 后当 UART 接口接收到低电平唤醒)
- 7) UART 1 唤醒中断(进入 Sleep 后当 UART 接口接收到低电平唤醒)
- 8) WDT 溢出
- 9) I2C_Slave 广播唤醒或者匹配地址才唤醒

3.2 看门狗(WDT)

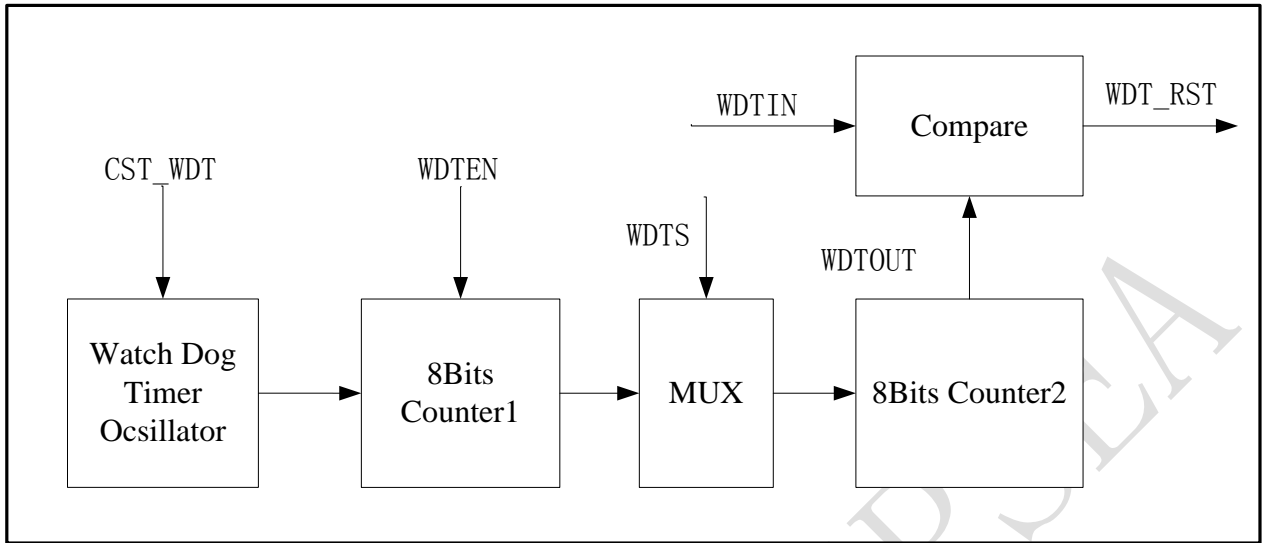


图 35 看门狗定时器功能框图

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 PC 指针 复位, 复位 CPU 核和 GIE 使能位, 其他 SFR 寄存器不会被复位。

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 WDT_IN 数值相等时溢出, 溢出时它会发送 WDTOUT 信号复位 CPU 及置位 TO 标志位。用户可以使用指令 CLRWDT 复位 WDT。

表 26 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	-	-	-	PD	TO	DC	C	Z	uuu00xxx
0EH	WDTCON	WDTEN	ROOT_EN	AIENB3	I2C_DIV[1:0]		WDT_S[2:0]			00100000
0FH	WDTIN	WDT_IN[7:0]								00000000

操作:

1. 设置 WDT_S[2:0], 选择 WDT 时钟频率。
2. 设置 WDT_IN, 选择不同的溢出时间值
2. 置位寄存器标志位: WDTEN, 使能 WDT。
4. 在程序中执行 CLRWDT 指令复位 WDT。

WDT_S[2:0]	计数器时钟	时间 (当 WDTIN==BBH)
000	WDTA[7] : WDT_CLK/256	16s
001	WDTA[6] : WDT_CLK/128	8s
010	WDTA[5] : WDT_CLK/64	4s
011	WDTA[4] : WDT_CLK/32	2s
100	WDTA[3] : WDT_CLK/16	1s
101	WDTA[2] : WDT_CLK/8	0.5s
110	WDTA[1] : WDT_CLK/4	0.25s
111	WDTA[0] : WDT_CLK/2	0.125s

3.3 电源系统

3.3.1 Regulator

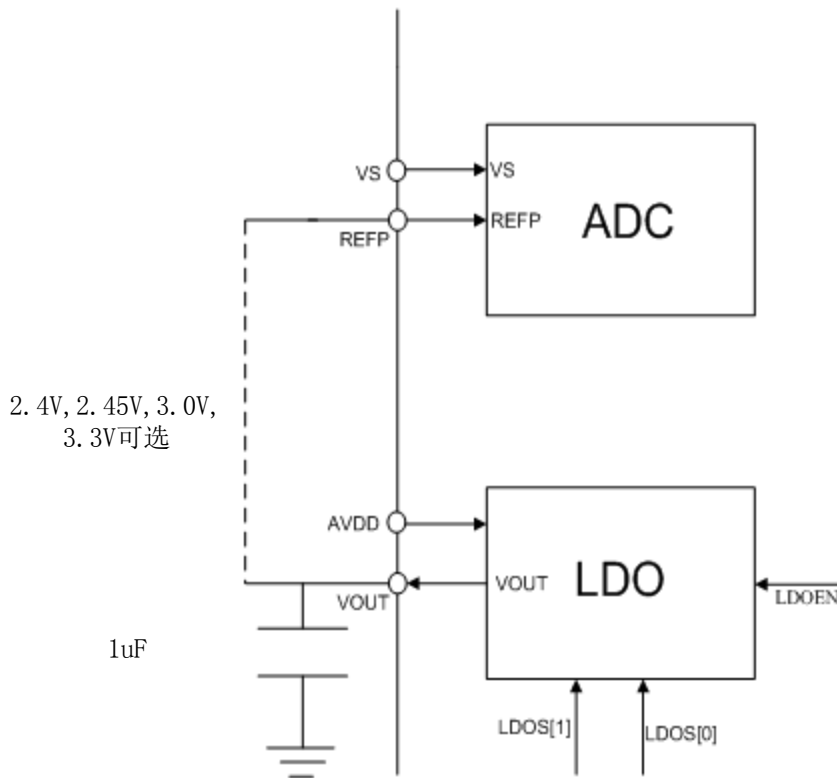


图 36 稳压电路

如图 36 所示，用于产生 VS 作为传感器和 24 Bits ADC 的参考电压，通过选择 LDOS 可以使输出 2.4V，2.45V，3.0V，3.3V 可选。LDOEN 作为 LDO 的使能信号。LDO 的控制寄存器标志是 LDOEN 与 LDOS。输出电压是 VS。

表 27 稳压电路寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
17H	ANACFG G	LDOEN N	LDOS[1:0]		BGR_EN B	BGID	SINL[1:0]		ADEN	00000000

ANACFG 寄存器（地址=17H）

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
ANACFG	LDOEN	LDOS[1:0]		BGR_ENB	BGID	SINL[1:0]		ADEN
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 LDOEN: LDO 使能控制

LDOEN=0 LDO 关闭

LDOEN=1 LDO 打开

Bit6~5 LDOS[1:0]: VS 电压值选择

LDOS[1:0] 00 VS=3.3V

LDOS[1:0] 01 VS=3.0V

LDOS[1:0] 10 VS=2.45V

LDOS[1:0] 11 VS=2.4V

Bit4 BGR_ENB: Bandgap 使能控制（只读，Bandgap 给芯片提供零温漂电压、电流）

BGR_ENB=0 Bandgap 打开

- BGR_ENB=1 Bandgap 关闭
- Bit3 BGID: ADC 模式选择
 - BGID=0 一般模式
 - BGID=1 性能模式
- Bit 2-1 ADC 通道选择
 - SINL[1:0] :
 - 00 = 24 Bits ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 V_{in+} , AIN1 为 V_{in-} ;
 - 01 = 内短;
 - 10 = 24 Bits ADC 输入端连接到 TEMP;
 - 11 = 24 Bits ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 V_{in-} , AIN1 为 V_{in+} ;
- Bit 0 ADEN: 24 Bits ADC 使能标志
 - 1 = 24 Bits ADC 使能
 - 0 = 24 Bits ADC 不使能

3.3.2 低电压比较器

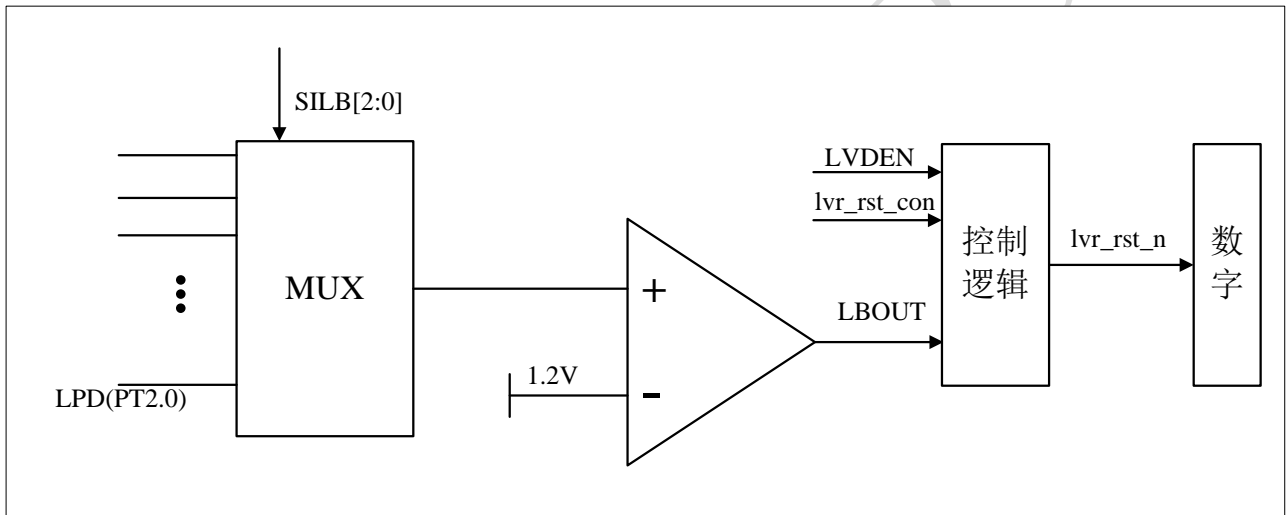


图 37 低电压比较功能模块框图

低电压比较器用于 DVDD 的低电压检测。芯片集成一个可产生 $1/2DVDD$ 及 $1/3DVDD$ 的分压器。多路选择器用于选择不同的分压连接到低电压比较器的输入端。多路选择器的输出与 1.2V 进行比较，它的控制寄存器标志是 SILB[2:0]及 LVDEN，比较器的输出是 LBOUT，LBOUT 为只读，在 LVDEN=1 打开 LB_RST_CON，会对 LBOUT 进行判断，如果 LBOUT=0 的时间超过 100us，芯片产生复位。请看图 37。

表 28 低电压比较器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit 4	Bit 3	Bit 2	Bit1	Bit0	上电复位值
1B H	LVDCO N	LVDE N	AD2_RE F	AIENB 1	SILB[2:0]			LBOU T	LB_RST_CO N	001000x 0

操作:

设置寄存器标志位 LVDEN，使能低电压比较器。

比较器输出是 LBOUT。

注：在 sleep 模式下 LVD 模块不工作，另外 ICK_EN=0。

表 29 低电压比较器检测电压的选择列表

SILB[2:0]	检测电压	满足条件	则
000	DVDD	DVDD>2.4V	LBOUT=1
001	DVDD	DVDD>2.5V	LBOUT=1
010	DVDD	DVDD>2.6V	LBOUT=1
011	DVDD	DVDD>2.7V	LBOUT=1
100	DVDD	DVDD>2.8V	LBOUT=1
101	DVDD	DVDD>3.0V	LBOUT=1
110	LPD	LPD>1.2V	LBOUT=1
111	DVDD	DVDD>3.6V	LBOUT=1

LVDCON 寄存器 (地址=1BH)

特性	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R-x	R/W-0
NETE	LVDCON	AD2_REF	AIENB1	SILB[2:0]			LBOUT	LB_RST_CON
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

位地址	标识符	功能																																				
7	LVDCON	低电压检测使能信号 <table border="1"> <tr> <td>LVDCON</td> <td>-</td> </tr> <tr> <td>0</td> <td>低电压检测不使能</td> </tr> <tr> <td>1</td> <td>低电压检测使能</td> </tr> </table>	LVDCON	-	0	低电压检测不使能	1	低电压检测使能																														
LVDCON	-																																					
0	低电压检测不使能																																					
1	低电压检测使能																																					
4:2	SILB[2:0]	低电压比较器检测电压的选择 <table border="1"> <thead> <tr> <th>SILB[2:0]</th> <th>检测电压</th> <th>满足条件</th> <th>则</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>DVDD</td> <td>DVDD>2.4V</td> <td>LBOUT=1</td> </tr> <tr> <td>001</td> <td>DVDD</td> <td>DVDD>2.5V</td> <td>LBOUT=1</td> </tr> <tr> <td>010</td> <td>DVDD</td> <td>DVDD>2.6V</td> <td>LBOUT=1</td> </tr> <tr> <td>011</td> <td>DVDD</td> <td>DVDD>2.7V</td> <td>LBOUT=1</td> </tr> <tr> <td>100</td> <td>DVDD</td> <td>DVDD>2.8V</td> <td>LBOUT=1</td> </tr> <tr> <td>101</td> <td>DVDD</td> <td>DVDD>3.0V</td> <td>LBOUT=1</td> </tr> <tr> <td>110</td> <td>LPD</td> <td>LPD>1.2V</td> <td>LBOUT=1</td> </tr> <tr> <td>111</td> <td>DVDD</td> <td>DVDD>3.6V</td> <td>LBOUT=1</td> </tr> </tbody> </table>	SILB[2:0]	检测电压	满足条件	则	000	DVDD	DVDD>2.4V	LBOUT=1	001	DVDD	DVDD>2.5V	LBOUT=1	010	DVDD	DVDD>2.6V	LBOUT=1	011	DVDD	DVDD>2.7V	LBOUT=1	100	DVDD	DVDD>2.8V	LBOUT=1	101	DVDD	DVDD>3.0V	LBOUT=1	110	LPD	LPD>1.2V	LBOUT=1	111	DVDD	DVDD>3.6V	LBOUT=1
SILB[2:0]	检测电压	满足条件	则																																			
000	DVDD	DVDD>2.4V	LBOUT=1																																			
001	DVDD	DVDD>2.5V	LBOUT=1																																			
010	DVDD	DVDD>2.6V	LBOUT=1																																			
011	DVDD	DVDD>2.7V	LBOUT=1																																			
100	DVDD	DVDD>2.8V	LBOUT=1																																			
101	DVDD	DVDD>3.0V	LBOUT=1																																			
110	LPD	LPD>1.2V	LBOUT=1																																			
111	DVDD	DVDD>3.6V	LBOUT=1																																			
1	LBOUT	低电压检测标志，只读。 参见 SILB[2:0]说明																																				
0	LB_RST_CON	低电压复位使能控制信号，仅在 LVDCON=1 时该控制位才有意义 0: 时低电压复位使能关闭 1: 时低电压复位使能开启；在 LVDCON=1 且 LBOUT=0 的时间超过 100us 时，芯片内部产生低电压复位信号，复位整个数字电路。																																				

3.4 定时/计数器 1

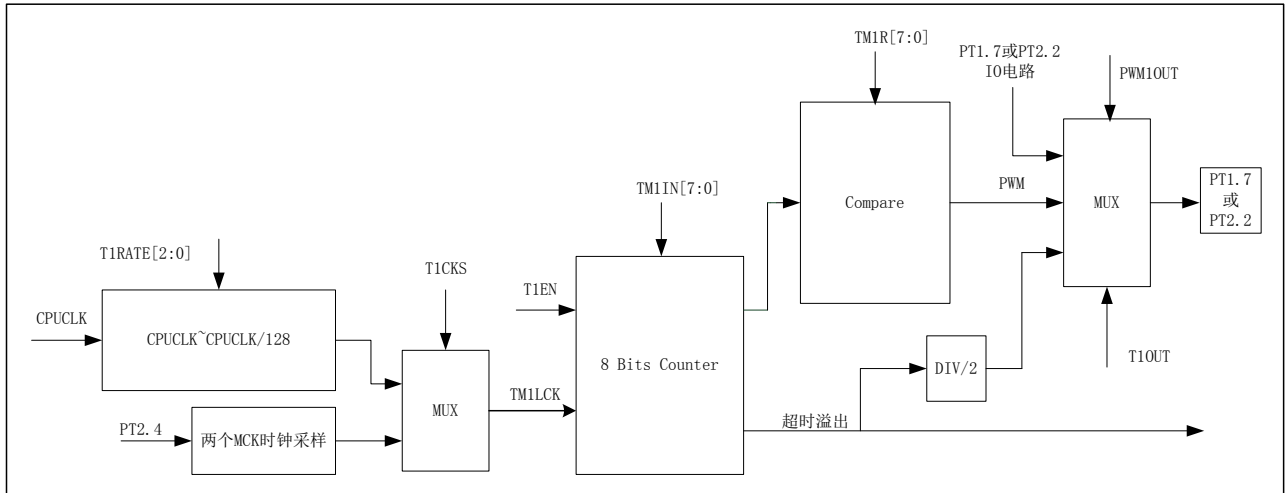


图 38 定时/计数器 1 模块的功能框图

定时/计数器 1 模块的输入是 TM1CLK。当用户设置了定时/计数器 1 模块的使能标志，8 bits 计数器将启动，从 00h 递增到 TM1IN。用户需要设置 TM1IN（定时器模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，BZ 输出信号发生跳变。

主要功能：

- 1) 8 位可编程定时器；
- 2) 外部事件计数；
- 3) 蜂鸣器输出；
- 4) PWM1 输出；

3.4.1 寄存器说明

表 30 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	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	00000000
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	00000000
37H	TM1CON	T1EN	T1RATE[2:0]			TICKS	T1RSTB	T1OUT	PWM1OUT	00000100
38H	TM1IN	TM1IN[7:0]								00000100
39H	TM1CNT	TM1CNT[7:0]								00000000
3AH	TM1R	TM1R[7:0]								00000000
2DH	PTCON	BZEN	PWM3_SEL	PWM2_SEL	PWM1_SEL	E1M[1:0]		E0M[1:0]		00000000 0

表 31 TM1CON 寄存器各位功能表

位地址	标识符	功能
7	T1EN	定时/计数器 1 使能位 1: 使能定时器/计数器 1 0: 禁止定时器/计数器 1
6:4	T1RATE[2:0]	定时/计数器 1 时钟分频
		T1RATE [2:0] TM1CLK
		000 CPUCLK 001 CPUCLK/2

		010	011	100	101	110	111	CPUCLK/4 CPUCLK/8 CPUCLK/16 CPUCLK/32 CPUCLK/64 CPUCLK/128
3	T1CKS	定时/计数器 1 时钟源选择位 1: PT2.4 作为时钟 0: CPUCLK 的分频时钟						
2	T1RSTB	定时/计数器 1 复位 1: 禁止定时/计数器 1 复位 0: 使能定时/计数器 1 复位 当将该位为 0 时, 定时器 1 复位后, T1RSTB 会自动置 1						
1	T1OUT	PT1.7/PT2.6 口输出控制						
		T1OUT	PWM1OUT	PT1.7/PT2.6 输出控制				
		0	0	IO 输出				
		0	1	蜂鸣器输出				
0	PWM1OUT	1	0	PWM1 输出				
		1	1	PWM1 取反输出				

表 32 PTCON 寄存器各位功能表

位地址	标识符	功能
4	PWM1_SEL	PT1.7/PT2.6 口输出选择信号 1: Timer1 的输出选择 PT2.6 0: Timer1 的输出选择 PT1.7

表 33 TM1IN 寄存器各位功能表

位地址	标识符	功能
7 : 0	TM1IN[7:0]	定时/计数器溢出值

表 34 TM1CNT 寄存器各位功能表

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

表 35 TM1R 寄存器各位功能表

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

3.4.2 定时器 1 使用说明

操作:

- 1) 设置 TM1CLK, 为定时器 1 模块选择输入。
- 2) 设置 TM1IN, 选择定时器 1 溢出值。
- 3) 设置寄存器标志位: TM1IE 与 GIE, 使能定时器中断。

- 4) 清零寄存器标志位: T1RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T1EN, 使能定时器模块的计数器。
- 6) 当定时超时发生时, 寄存器标志位 TM1IF 会自复位, 程序计数器会复位为 0004H。

定时器 1 溢出时间计算方法:

$$\text{定时器 1 溢出时间} = (\text{TM1IN} + 1) / \text{TM1CLK.} \quad (\text{TM1IN 不为 } 0)$$

3.4.3 蜂鸣器 1 使用说明

操作:

- 1) 把 PT1.7 或 PT2.2 配置为输出口。
- 2) 设置 TM1CLK, 为定时器模块选择输入。
- 3) 设置 TM1IN, 选择定时器溢出值。
- 4) 清零寄存器标志位: T1RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T1EN, 使能定时器模块的 8 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出。

蜂鸣器周期计算方法:

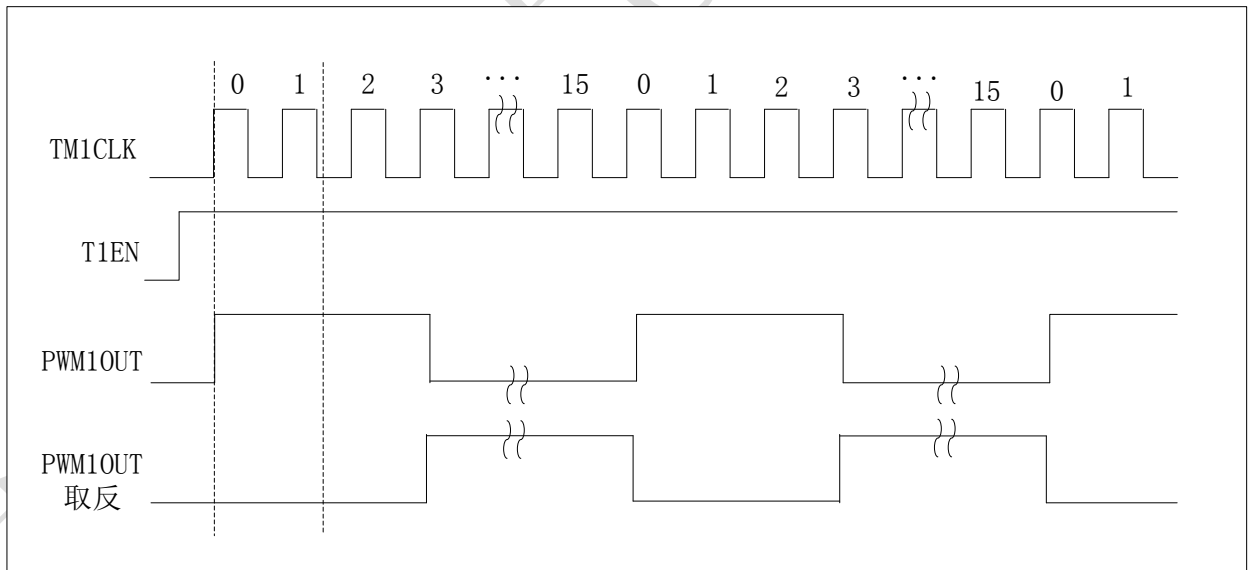
$$\text{蜂鸣器周期} = (\text{TM1IN} + 1) * 2 / \text{TM1CLK.} \quad (\text{TM1IN 不为 } 0)$$

3.4.4 PWM1 使用说明

操作:

- 1) 设置 TM1CLK, 为定时/计数器 1 模块选择输入。
- 2) 设置 TM1IN 来配置 PWM1 的周期。
- 3) 设置 TM1R 来配置 PWM1 的高电平的脉宽。
- 4) 使能 PWM1OUT 输出, 配置 PT1.7/PT2.2 口为输出端口, 之后把 T1EN 置 1 启动定时器。
- 5) PWM1 从 PT1.7/PT2.2 口输出。

周期为 TM1IN+1, 高电平脉宽为 TM1R。如 TM1IN=0x0F, TM1R=0x03 的 PWM1 波形输出如下:



3.5 定时/计数器 2

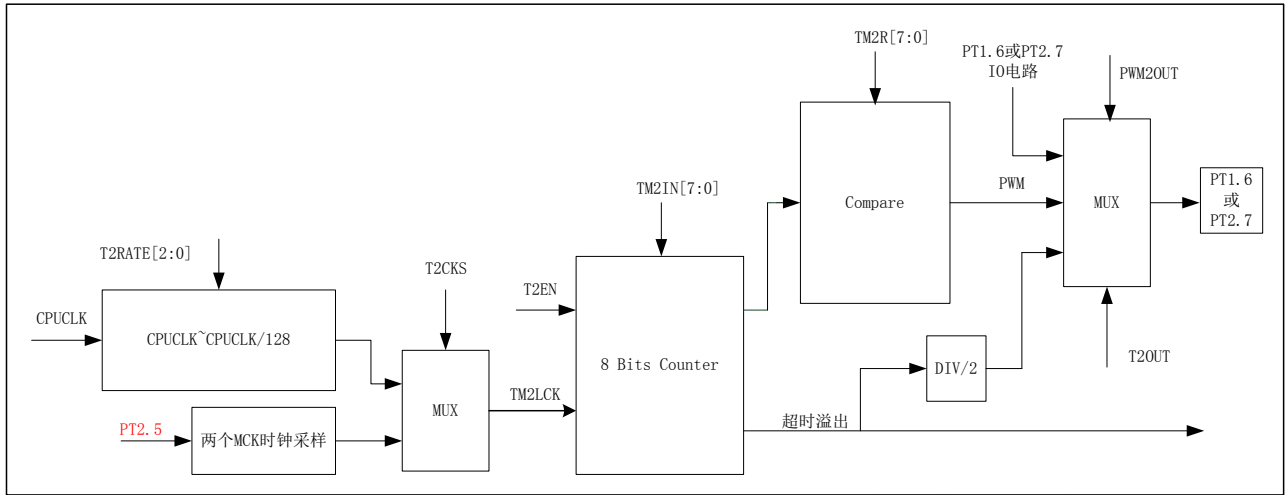


图 39 定时/计数器 2 模块的功能框图

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，8 bits 计数器将启动，从 00h 递增到 TM2IN。用户需要设置 TM2IN（定时器模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，BZ 输出信号发生跳变。

主要功能：

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

3.5.1 寄存器说明

表 36 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	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	u0000000
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	00000000
3BH	TM2CON	T2EN	T2RATE[2:0]			T2CKS	T2RSTB	T2OUT	PWM2OUT	00000100
3CH	TM2IN	TM2IN[7:0]								00000000
3DH	TM2CNT	TM2CNT[7:0]								00000000
3EH	TM2R	TM2R[7:0]								00000000
2DH	PTCON	BZEN	PWM3_SEL	PWM2_SEL	PWM1_SEL	E1M[1:0]		E0M[1:0]		00000000

表 37 TM2CON 寄存器各位功能表

位地址	标识符	功能
7	T2EN	定时/计数器 2 使能位 1: 使能定时器/计数器 2 0: 禁止定时器/计数器 2

6:4	T2RATE[2:0]	定时/计数器 2 时钟分频		
		T2RATE [2:0]	TM2CLK	
		000	CPUCLK	
		001	CPUCLK/2	
		010	CPUCLK/4	
		011	CPUCLK/8	
		100	CPUCLK/16	
		101	CPUCLK/32	
		110	CPUCLK/64	
		111	CPUCLK/128	
3	T2CKS	定时/计数器 2 时钟源选择位 1: PT2.5 作为时钟 0: CPUCLK 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	PT1.6/PT2.7 口输出控制		
		T2OUT	PWM2OUT	PT1.6/PT2.7 输出控制
0	PWM2OUT	0	0	IO 输出
		0	1	蜂鸣器输出
		1	0	PWM2 输出
		1	1	PWM2 取反输出

表 38 PTCON 寄存器各位功能表

位地址	标识符	功能
5	PWM2_SEL	PT1.6/PT2.7 口口输出选择信号 1: Timer2 的输出选择 PT2.7 0: Timer2 的输出选择 PT1.6

表 39 TM2IN 寄存器各位功能表

位地址	标识符	功能
7 : 0	TM2IN[7:0]	定时/计数器溢出值

表 40 TM2CNT 寄存器各位功能表

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

表 41 TM2R 寄存器各位功能表

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

3.5.2 定时器 2 使用说明

操作:

- 1) 设置 TM2CLK, 为定时器 2 模块选择输入。
- 2) 设置 TM2IN, 选择定时器 2 溢出值。
- 3) 设置寄存器标志位: TM2IE 与 GIE, 使能定时器中断。
- 4) 清零寄存器标志位: T2RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T2EN, 使能定时器模块的计数器。
- 6) 当定时超时发生时, 寄存器标志位 TM2IF 会自复位, 程序计数器会复位为 0004H。

定时器 2 溢出时间计算方法:

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

3.5.3 蜂鸣器 2 使用说明

操作:

- 1) 把 PT1.6/PT2.7 配置为输出口。
- 2) 设置 TM2CLK, 为定时器模块选择输入。
- 3) 设置 TM2IN, 选择定时器溢出值。
- 4) 清零寄存器标志位: T2RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T2EN, 使能定时器模块的 8 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出。

蜂鸣器周期计算方法:

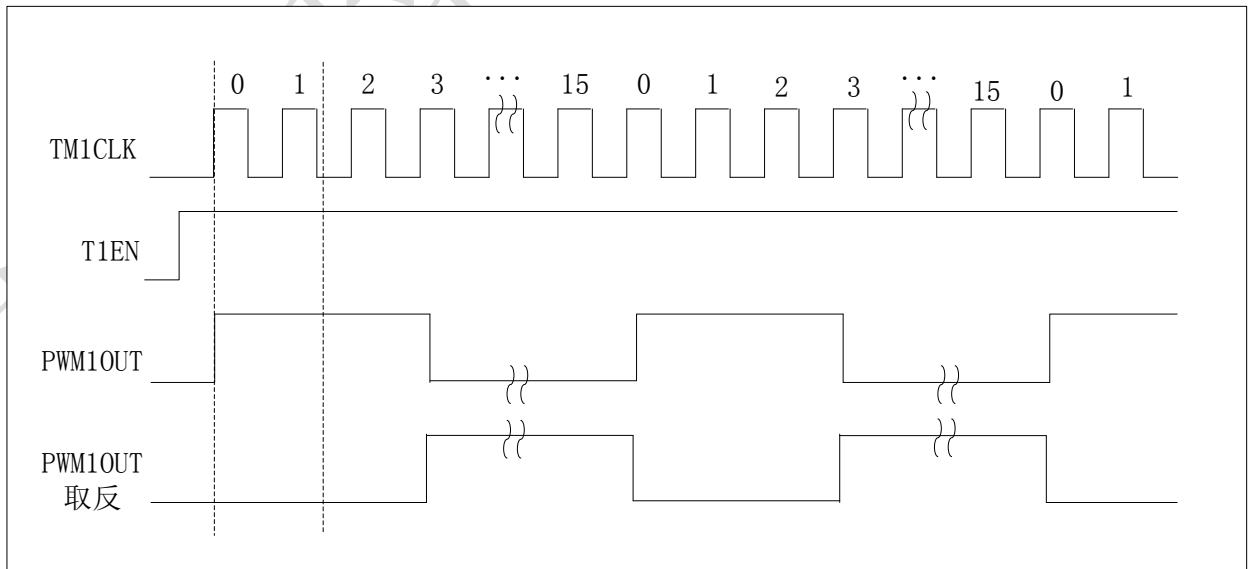
$$\text{蜂鸣器周期} = (\text{TM2IN} + 1) * 2 / \text{TM2CLK.} \quad (\text{TM2IN 不为 } 0)$$

3.5.4 PWM2 使用说明

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 PT1.6/PT2.7 为输出端口, 之后把 T2EN 置 2 启动定时器。
- 5) PWM2 从 PT1.6/PT2.7 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:



3.6 定时/计数器 3

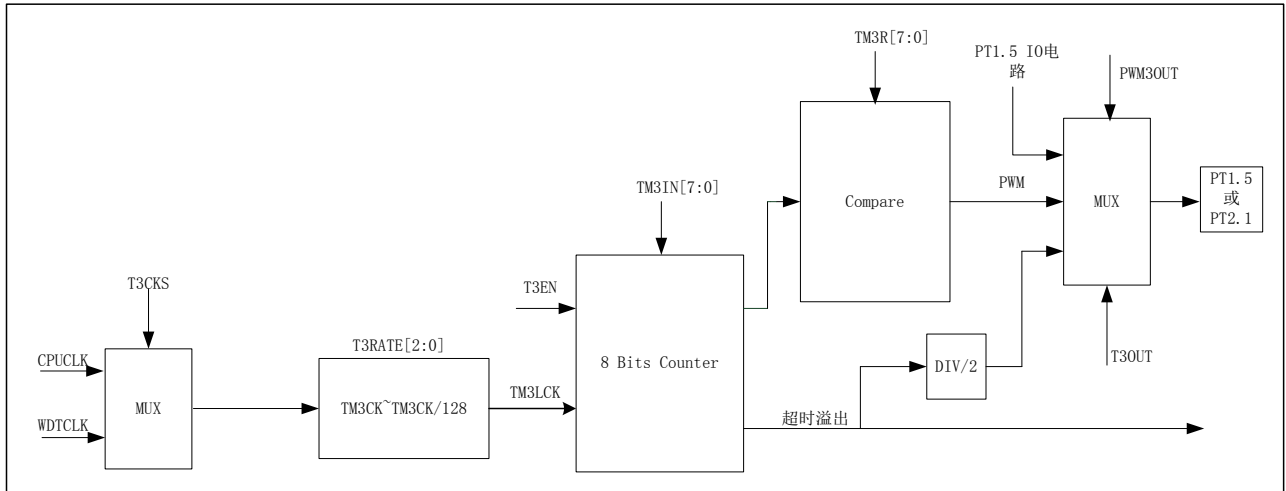


图 40 定时/计数器 3 模块的功能框图

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，8 bits 计数器将启动，从 00h 递进到 TM3IN。用户需要设置 TM3IN（定时器模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，BZ 输出信号发生跳变。

主要功能：

- 1) 8 位可编程定时器；
- 2) 外部事件计数；
- 3) 蜂鸣器输出；
- 4) PWM3 输出；
- 5)

3.6.1 寄存器说明

表 42 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
5BH	INTE ₃	I2C_TIE	I2C_RIE	UR0ERR_IE	UR1WK_IE	UR0WK_IE	TM3IE	RTCIE	SPIIE	0000000
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	0000000
3FH	TM3CON	T3EN	T3RATE[2:0]			T3CKS	T3RSTB	T3OUT	PWM3OUT	0000010
40H	TM3IN	TM3IN[7:0]								0000000
41H	TM3CNT	TM3CNT[7:0]								0000000
42H	TM3R	TM3R[7:0]								0000000
2DH	PTCON	BZEN	PWM3SEL	PWM2SEL	PWM1SEL	E1M[1:0]		E0M[1:0]		0000000

表 43 TM3CON 寄存器各位功能表

位地址	标识符	功能
-----	-----	----

7	T3EN	定时器 3 使能位 1: 使能定时器 3 0: 禁止定时器 3																											
6:4	T3RATE[2:0]	定时器 3 时钟选择 <table border="1"> <tr> <td>T3RATE [2:0]</td> <td>TM3CLK=CPUCLK</td> <td>TM3CLK=WDT</td> </tr> <tr> <td>000</td> <td>TM3CK</td> <td>TM3CK</td> </tr> <tr> <td>001</td> <td>TM3CK /2</td> <td>TM3CK /4</td> </tr> <tr> <td>010</td> <td>TM3CK /4</td> <td>TM3CK /8</td> </tr> <tr> <td>011</td> <td>TM3CK /8</td> <td>TM3CK /16</td> </tr> <tr> <td>100</td> <td>TM3CK /16</td> <td>TM3CK /32</td> </tr> <tr> <td>101</td> <td>TM3CK /32</td> <td>TM3CK /64</td> </tr> <tr> <td>110</td> <td>TM3CK /64</td> <td>TM3CK /128</td> </tr> <tr> <td>111</td> <td>TM3CK /128</td> <td>TM3CK /256</td> </tr> </table>	T3RATE [2:0]	TM3CLK=CPUCLK	TM3CLK=WDT	000	TM3CK	TM3CK	001	TM3CK /2	TM3CK /4	010	TM3CK /4	TM3CK /8	011	TM3CK /8	TM3CK /16	100	TM3CK /16	TM3CK /32	101	TM3CK /32	TM3CK /64	110	TM3CK /64	TM3CK /128	111	TM3CK /128	TM3CK /256
T3RATE [2:0]	TM3CLK=CPUCLK	TM3CLK=WDT																											
000	TM3CK	TM3CK																											
001	TM3CK /2	TM3CK /4																											
010	TM3CK /4	TM3CK /8																											
011	TM3CK /8	TM3CK /16																											
100	TM3CK /16	TM3CK /32																											
101	TM3CK /32	TM3CK /64																											
110	TM3CK /64	TM3CK /128																											
111	TM3CK /128	TM3CK /256																											
3	T3SEL	时钟源选择 <table border="1"> <tr> <td>T3SEL</td> <td>定时器 3 时钟源(TM3CK)</td> </tr> <tr> <td>0</td> <td>CPUCLK</td> </tr> <tr> <td>1</td> <td>内部 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效</td> </tr> </table>	T3SEL	定时器 3 时钟源(TM3CK)	0	CPUCLK	1	内部 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效																					
T3SEL	定时器 3 时钟源(TM3CK)																												
0	CPUCLK																												
1	内部 3K WDT 时钟， 仅当内部 WDT 时钟打开时有效																												
2	T3RSTB	定时器 3 复位 1: 禁止定时器 3 复位 0: 使能定时器 3 复位 当将该位为 3 时，定时器 3 复位后，T3RSTB 会自动置 1																											
1	T3OUT	PT1.5/PT2.1 口输出控制																											
0	T3OUT	<table border="1"> <tr> <td>T3OUT</td> <td>PWM3OUT</td> <td>PT1.5/PT2.1 输出控制</td> </tr> <tr> <td>0</td> <td>0</td> <td>IO 输出</td> </tr> <tr> <td>0</td> <td>1</td> <td>蜂鸣器输出</td> </tr> <tr> <td>1</td> <td>0</td> <td>PWM3 输出</td> </tr> <tr> <td>1</td> <td>1</td> <td>PWM3 取反输出</td> </tr> </table>	T3OUT	PWM3OUT	PT1.5/PT2.1 输出控制	0	0	IO 输出	0	1	蜂鸣器输出	1	0	PWM3 输出	1	1	PWM3 取反输出												
T3OUT	PWM3OUT	PT1.5/PT2.1 输出控制																											
0	0	IO 输出																											
0	1	蜂鸣器输出																											
1	0	PWM3 输出																											
1	1	PWM3 取反输出																											

表 44 PTCN 寄存器各位功能表

位地址	标识符	功能
6	PWM3_SEL	PT1.5/PT2.1 口输出选择信号 1: Timer3 的输出选择 PT2.1 0: Timer3 输出选择 PT1.5

表 45 TM3IN 寄存器各位功能表

位地址	标识符	功能
7 : 0	TM3IN[7:0]	定时/计数器溢出值

表 46 TM3CNT 寄存器各位功能表

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

表 47 TM3R 寄存器各位功能表

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

3.6.2 定时器 3 使用说明

操作:

- 1) 设置 TM3CLK, 为定时器 3 模块选择输入。
- 2) 设置 TM3IN, 选择定时器 3 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, 使能定时器模块的计数器。
- 6) 当定时超时发生时, 寄存器标志位 TM3IF 会自复位, 程序计数器会复位为 0004H。

定时器 3 溢出时间计算方法:

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

3.6.3 蜂鸣器 3 使用说明

操作:

- 1) 把 PT1.5/PT2.1 配置为输出口。
- 2) 设置 TM3CLK, 为定时器模块选择输入。
- 3) 设置 TM3IN, 选择定时器溢出值。
- 4) 清零寄存器标志位: T3RSTB, 复位定时器模块的计数器。
- 5) 设置寄存器标志位: T3EN, 使能定时器模块的 8 bits 计数器。
- 6) 当定时超时发生时, BZ 输出信号发生跳变, 可作为蜂鸣器输出。

蜂鸣器周期计算方法:

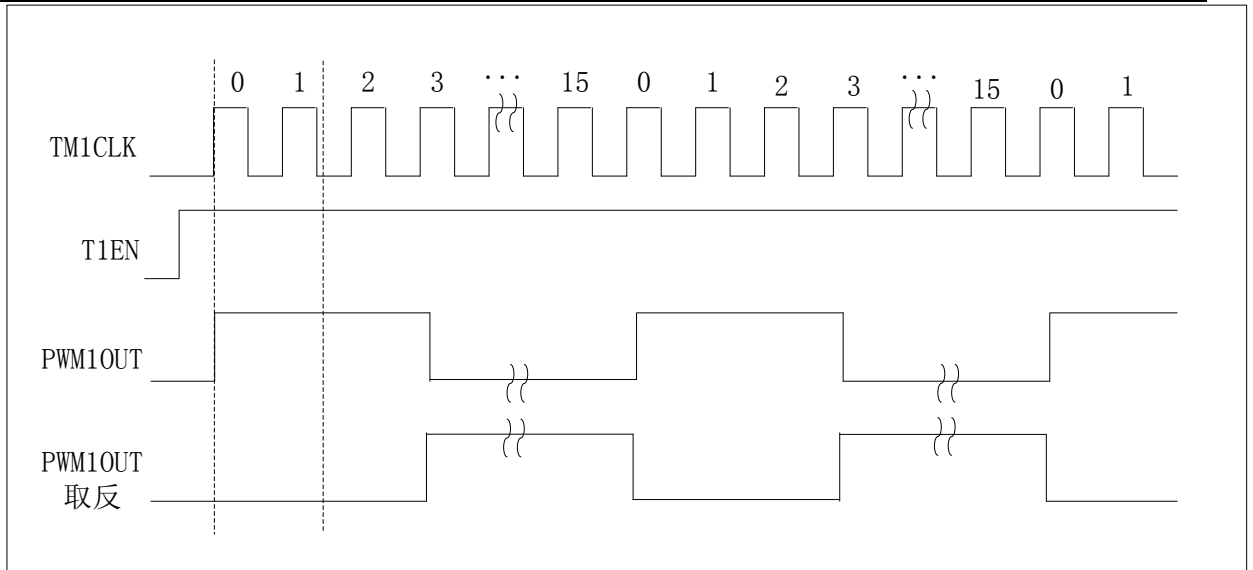
蜂鸣器周期 = (TM3IN+1) * 2 / TM3CLK. (TM3IN 不为 0)

3.6.4 PWM3 使用说明

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 PT1.5/PT2.1 为输出端口, 之后把 T3EN 置 3 启动定时器。
- 5) PWM3 从 PT1.5/PT2.1 输出。

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



3.7 串行通信接口 0/1

芯片提供两个可编程全双工异步串行通信接口。

支持同时数据的发送和接收；

波特率可配，最高速率能够支持 115200；

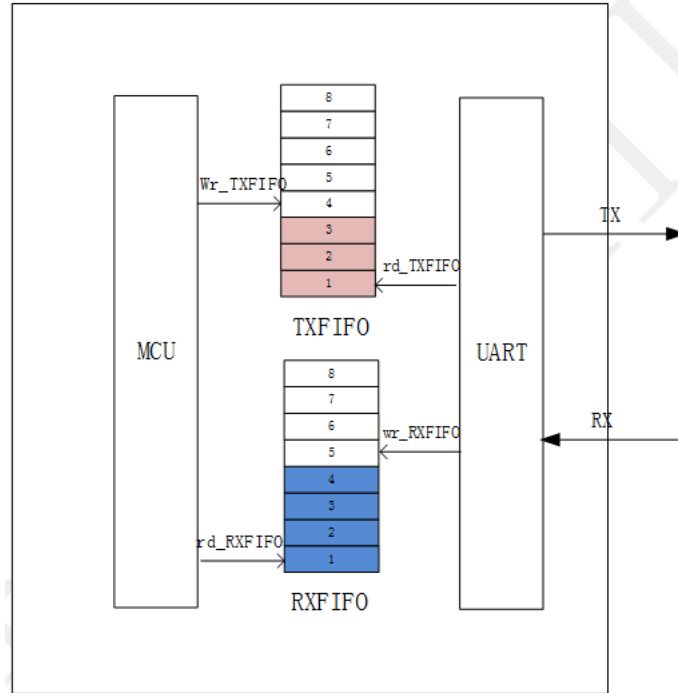
支持自动波特率；

接收和发送都有 8 级的 FIFO，RX_FIFO(8*9Bit)，TX_FIFO(8*9Bit)

接收中断支持非空中断和接收流水线中断

发送中断支持发送空中断和发送流水线中断

支持 8/9 位数据发送/接收。



3.7.1 波特率配置

UART0/1 的波特率配置公式，
$$BR = \frac{UART0/1_CLK}{BRR0 + \frac{BRR1}{10}}$$

其中 $UART0/1_CLK = \frac{ICK}{2^{UART0/1_DIV+1}}$ ，UART0/1_DIV 的范围为 0~5；BRR1 的范围为 0~9

注：UART0/1_CLK 表示 UART0_CLK 或者 UART1_CLK，UART0/1_DIV 同理。

例如：ICK=8MHz，若要以 115200bps 波特率通讯，采用 UART_CLK0/1=4MHz，进行第一步的计算，计算出 BRR0=34, BRR1=7；如果 BRR0 的结果小于 100，可以分频系数保持不变。如果 BRR0 计算的结果大于 100，那么分频系数 UART_DIV0/1 加 1。分频系数越大，功耗就越低。

3.7.2 自动波特率

自动波特率的流程

- (1) 配置 AUTO_BR =1；
- (2) 为了提高自动波特率的精度，配置 UART0/1_DIV =0。（如果已经知道波特率的大概范围，可以预先配置 UART0/1_DIV 来降低 UART 功耗）
- (3) 配置 UART_EN =1；使能 UART0/1 串口；
- (4) UART 上位机发送特定的数据，此特定的数据为 55；

- (5) 在接收完成特定的数据，硬件清 AUTO_BR；波特率配置自动完成。
注意本次接收不会引发接收完成标志和接收中断。

3.7.3 发送流程

发送的 TXFIFO 深度为 8，可以缓存 8 个字节的数据。
发送流程为：

- (1) 设置 BRR0/BRR1，选择合适的波特率；
- (2) 配置 UART_EN =1；使能 UART0/1 串口；
- (3) 配置中断使能：发送水线中断
- (4) 如果需要发送第 9 位，将 TX9_EN 置 1
- (5) 接着配置 TX_EN，此操作会使硬件自动置 UR0/1_TEIF 和 UR0/1_THIF，如果中断使能，会进入发送中断。
- (6) 再发送中断：

发送空中断：MCU 可以连续写最多 8 次 TX_REG(如果需要发送第 9Bit 先写 TX9D 再写 TX_REG)寄存器，UART 开始发送数据，当 TXFIFO 被读空后，报空中断，可以不停的对 TX_REG 写入数据，写入的次数为 8。

发送水线中断：可以不停的对 TX_REG 写入数据，写入的次数为设定水线值，UART 开始发送数据，当 TXFIFO 被读到低于设定的水线，报发送水线中断，可以不停对 TX_REG 写入数据，写入的次数为水线值。

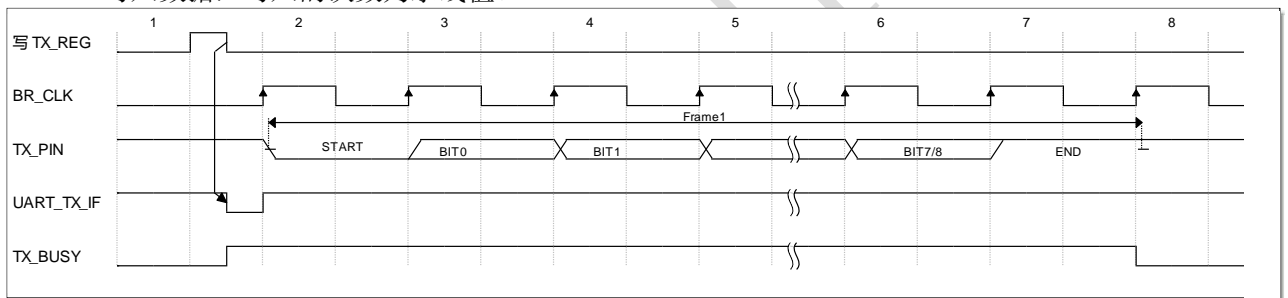


图 41 发送时序图

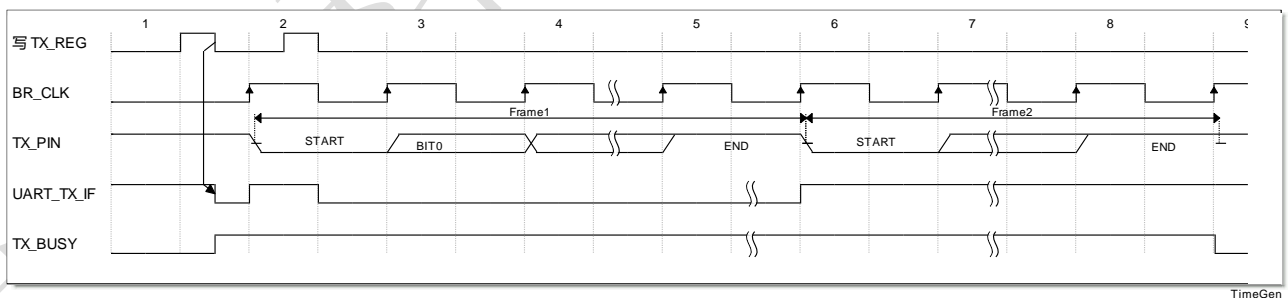


图 42 连续发送时序图（空中断）

3.7.4 接收流程

- (1) 设置 BRR0/BRR1，选择合适的波特率；
- (2) 配置 UART_EN =1；使能 UART0 串口；
- (3) 打开非空中断或者接收水线中断，或者两者全部打开
- (4) 根据需求，配置中断使能 GIE, UR0/1_RNIE, UR0/1_RHIE
- (5) 如果需要接收第 9 位，将 RX9_EN 置 1
- (6) 接着配置 RX_EN=1，

- (7) 当接收完成，硬件置位 UR0/1_RNIF, UR0/1_RHIF, 会进入接收中断。
- (8) 当发生接收中断时：
 - 如果是非空中断，MCU 可以读取 1 次 RX_REG 寄存器来获得字节数据(如果有第 9Bit 数据必须先读 RX_REG 才读 RX9D)
 - 如果是接收水线中断，MCU 可以连续读取配置的水线值次 RX_REG 寄存器来获得字节数据(如果有第 9Bit 数据必须先读 RX_REG 才读 RX9D)
- (9) 当 RXFIFO 溢出时会发生错误，通过使能 RX_EN =0, 可以清除错误标志

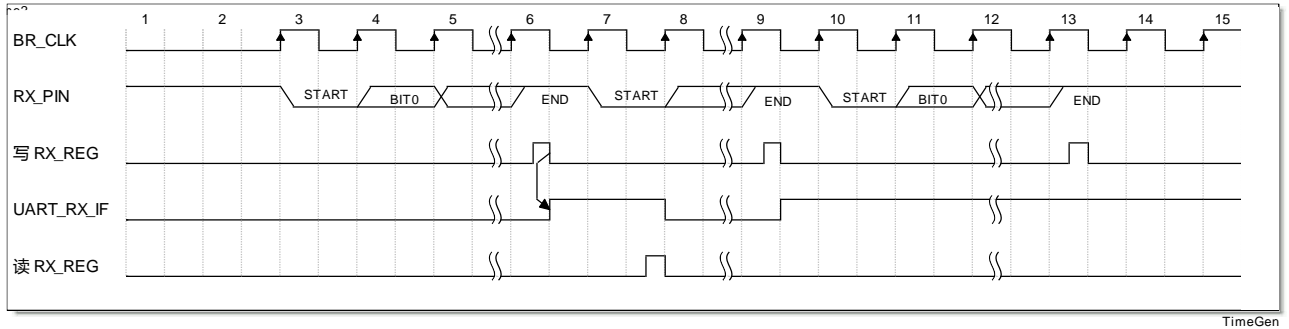


图 43 接收时序图

3.7.5 寄存器说明

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
32H	INTF2	UR1_RHIF	UR1_RNIF	UR1_THIF	UR1_TEIF	UR0_RHIF	UR0_RNIF	UR0_THIF	UR0_TEIF	00000000
33H	INTE2	UR1_RHIE	UR1_RNIE	UR1_THIE	UR1_TIEIE	UR0_RHIE	UR0_RNIE	UR0_THIE	UR0_TIEIE	00000000
5AH	INTF3	I2C_TIF	I2C_RIF	UR1ERR_IE	UR1WK_IF	UR0WK_IF	TM3IF	RTCIF	SPIIF	00000000
5BH	INTE3	I2C_TIE	I2C_RIE	UR0ERR_IE	UR1WK_IE	UR0WK_IE	TM3IE	RTCIE	SPIIE	00000000
15H	PCK	UART1DIV			UART0DIV			S_bEEP[1:0]		00000000
70H/ 77H	UR0_CR1/ UR1_CR1	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN	TX_EN	UART0_SEL/ UART1_SEL	UART0_EN/ UART1_EN	00000000
71H/ 78H	UR0_BRR0/ UR1_BRR1	BRR0								00000000
72H/ 79H	UR0_BRR1/ UR1_BRR1	AUTO_BR	-	-	-	BRR1				00000000
73H/ 7AH	UR0_TX_REG/ UR1_TX_REG	TX_REG								00000000
74H/ 7BH	UR0_RX_REG/ UR1_RX_REG	RX_REG								00000000

75H/ 7CH	UR0_CR2/ UR1_CR2	-	RXF_WATER_LEVEL			-	TXF_WATER_LEVEL			00110011
76H/ 7DH	UR0_ST/ UR1_ST/	-	-	-	-	RX_BUS Y	TX_BU SY	OV_ER R	STOP_E RR	00000000

UR0_CR1/UR1_CR1 寄存器

位地址	标识符	功能
7	TX9D	发送数据第 9 位 1: 发送第 9 位数据为 1 0: 发送第 9 位数据为 0
6	RX9D	接收数据第 9 位, 只读, 不可写 1: 接收第 9 位数据为 1 0: 接收第 9 位数据为 0
5	TX9_EN	发送数据第 9 位使能 1: 使能发送第 9 位 0: 关闭发送第 9 位
4	RX9_EN	接收数据第 9 位使能 1: 使能接收第 9 位 0: 关闭接收第 9 位
3	RX_EN	接收控制选择 1: 允许接收 0: 禁止接收
2	TX_EN	发送使能 1: 发送使能打开 0: 发送使能关闭
1	UART0_SEL/ UART1_SEL	串口通信 0 UART0_SEL 接口选择: 0: PT1.2 作为串口 0 的 RX; PT1.3 作为串口 0 的 TX 1: PT2.3 作为串口 0 的 RX; PT2.2 作为串口 0 的 TX 串口通信 1 UART1_SEL 接口选择: 0: PT1.7 作为串口 1 的 RX; PT1.6 作为串口 1 的 TX 1: PT2.6 作为串口 1 的 RX; PT2.7 作为串口 1 的 TX (只有当 ROOT_EN=1, 才能改变此项)。
0	UART0_EN/ UART1_EN	串口 0/串口 1 使能控制 1: 打开 UART 0: 关闭 UART

UR0_BRR0/UR1_BRR0 寄存器

位地址	标识符	功能
7: 0	BRR0	波特率设置寄存器 0, 整数部分

UR0_BRR1/UR1_BRR1 寄存器

位地址	标识符	功能
7	AUTO_BR	自动波特率使能: 1'b0: 自动波特率功能未打开, 不需要软件写 0。 1'b1: 打开自动波特率功能, 要求外部输入数据为 55H, 硬件清零。 注意 此 Bit 不可读。
6:4	RFU	-
3: 0	BRR1	波特率设置寄存器 1, 小数部分

UART 的波特率配置公式，
$$BR = \frac{UART_CLK}{BRR0 + \frac{BRR1}{10}}$$

UR0_TX_REG/UR1_TX_REG 寄存器

位地址	标识符	功能
7:0	TX_REG	串口发送数据寄存器

UR0_RX_REG/UR1_RX_REG 寄存器

位地址	标识符	功能
7:0	RX_REG	串口接收数据寄存器

注意：对 RX_REG 连续读的时候，需要插入其他 Sfr 操作，比如说读出结果后写入 RAM 操作，写入 RAM 的后 7 位地址不能和 RX_REG 地址相同。

UR0_CR2/UR1_CR2 寄存器

位地址	标识符	功能
7	-	-
6:4	TXF_WATER_LEVEL	TXFIFO 的数据量小于设定值后，报发送水线中断 3'h7 当 TXIFO 数据量小于等于 6 时，报发送水线中断 3'h6 当 TXIFO 数据量小于等于 5 时，报发送水线中断 3'h1: 当 TXIFO 数据量小于等于 1 时，报发送水线中断 3'h0: 当 TXIFO 数据量等于 0 时，报发送水线
3	RFU	-
2:0	RXF_WATER_LEVEL	RXFIFO 的数据量大于设定值后，报接收水线中断 3'h7: 当接收的数据存满时，报接收水线中断 3'h6: 当接收的数据必须大于 6 时，报接收水线中断 3'h1: 当接收的数据必须大于 1 时，报接收水线中断 3'h0: 当接收的数据必须大于 0 时，报接收水线中断, 相当于非空中断

UR0_ST/UR1_ST 寄存器状态寄存器

位地址	标识符	功能
7:6	-	-
5	-	-
4	-	-
3	RX_BUSY	接收 Busy 指示信号 0: 表示接收端未进行接收 1: 表示接收端正在接收中。 注：当进入 Sleep 后，可以采用 UART 唤醒中断进行唤醒。CPU 被唤醒后，可以查询此 Bit。如果 RX_BUSY=0，表示此次唤醒可能是由于干扰造成。
2	TX_BUSY	发送寄存器 TX_REG 的数据是否全部串行发送完毕 0: 已经全部发送完毕 1: 还未全部发送完毕
1	RXOV_ERR	接收 RXFIFO 溢出错误标志，可以触发错误中断，硬件置 1，软件清 0 0: 未发现溢出错误 1: 发现溢出错误，

0	STOP_ERR	接收停止位错误标志，可以触发错误中断，硬件置 1，软件清 0 0: 接收时未发现停止位错误 1: 接收时发现停止位错误
---	----------	---

INTE3 中断使能寄存器寄存器

Bit6 UR1ERR_IE: UART1 错误中断使能，UART1 错误包括 RXFIFO 溢出和接收停止位错误，清此中断需要对相应的溢出标志或者接收停止位错误标志清零。

- 1 = 使能 UART 1 错误中断使能
- 0 = 不使能 UART1 错误中断使能

Bit5 UR0ERR_IE: UART0 错误中断使能，UART0 错误包括 RXFIFO 溢出和接收停止位错误，清此中断需要对相应的溢出标志或者接收停止位错误标志清零。

- 1 = 使能 UART 0 错误中断使能
- 0 = 不使能 UART0 错误中断使能

Bit 4 UR1WK_IE: UART1 唤醒中断使能标志，该中断具有唤醒 Sleep 功能(只有在 ROOT_EN=1, 才能改写)

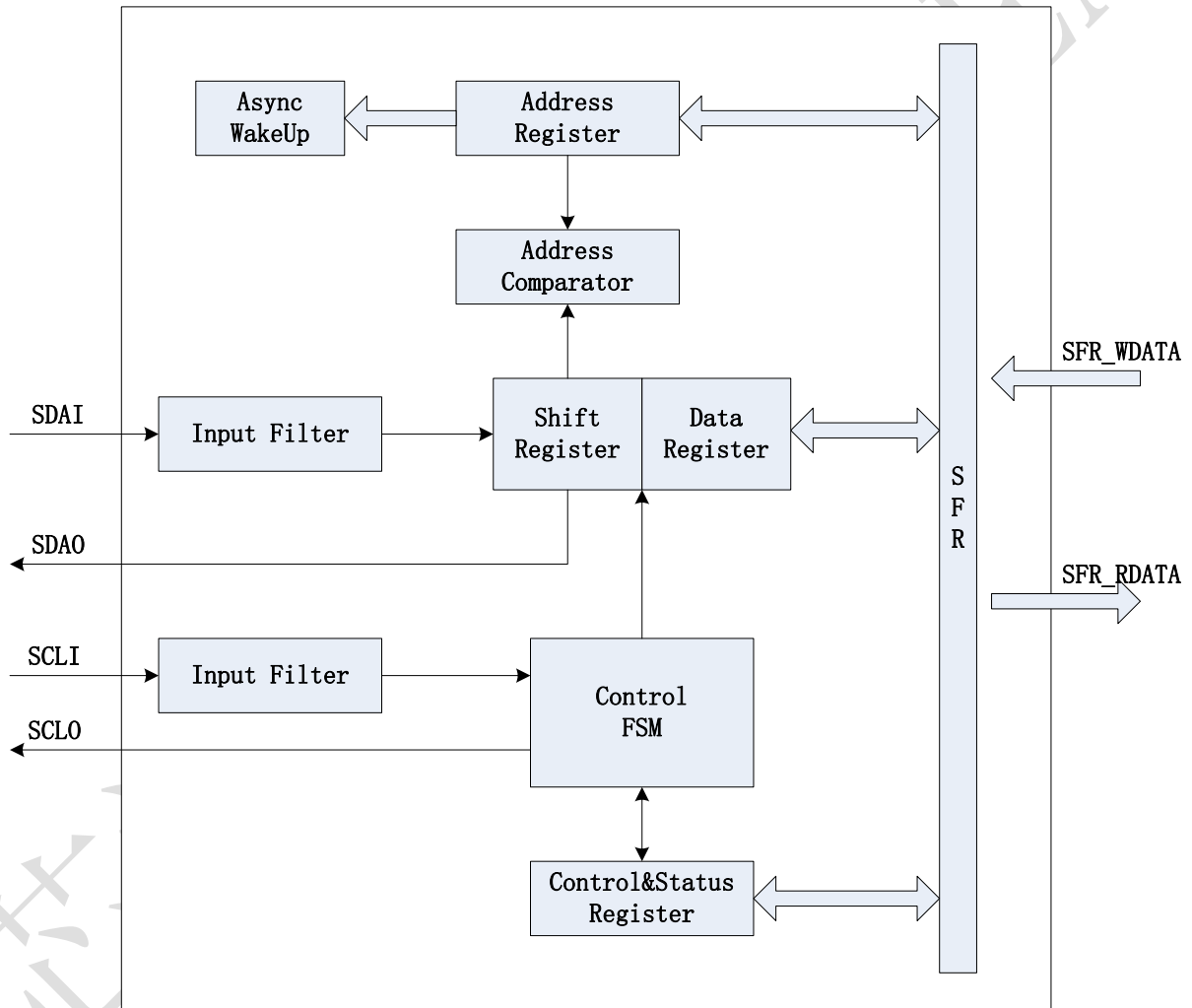
- 1 = 使能 UART 1 唤醒中断
- 0 = 不使能 UART 1 唤醒中断

Bit 3 UR0WK_IE: UART 0 唤醒中断使能标志，该中断具有唤醒 Sleep 功能

- 1 = 使能 UART 0 唤醒中断
- 0 = 不使能 UART 0 唤醒中断

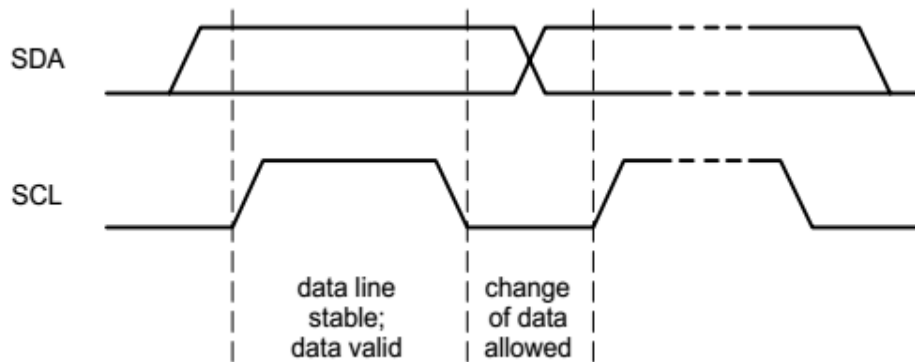
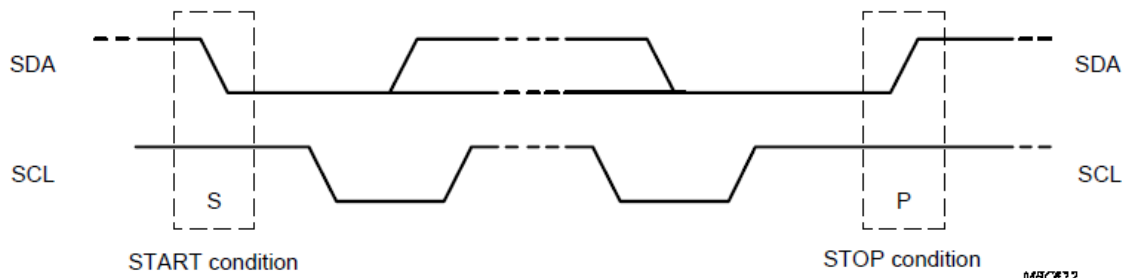
3.8 I2C 从机

CSU18MX88 集成一个 I2C 从机模块，可支持 8 位双向的数据收发。
 符合 I2C 总线规范，可以支持标准模式（100k bit/s）和快速模式（400k bit/s）数据传输
 支持 7 位地址寻址，默认从机地址为 0x2A，从机地址寄存器可配
 支持广播地址呼叫
 支持时钟低电平延长
 支持 SCL/SDA 输入滤波
 支持从机接收模式和从机发送模式；
 接收响应状态（ACK or NO ACK）软件可配
 支持 Sleep 模式异步唤醒，支持 Halt 模式唤醒



3.8.1 I2C 数据传输起始和终止条件

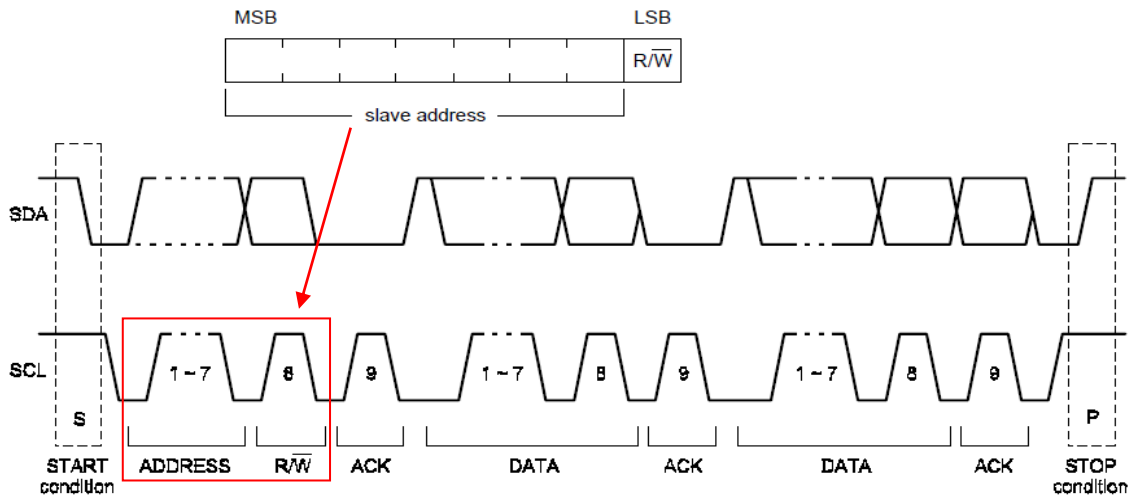
当 SCL 线为高电平时，SDA 线由高电平向低电平切换，定义为 I2C 传输起始 START。
 当 SCL 线为高电平时，SDA 线由低电平向高电平切换，定义为 I2C 传输终止 STOP。
 在正常的数据传输过程中，数据只允许在 SCL 为低时候切换，SCL 为高时传输数据保持不变。



3.8.2 7 位地址寻址

I2C 数据传输遵循下图所示格式。在起始条件 (S) 后，I2C 主机发送从机地址。这个地址共有 7 位，紧接着的第 8 位为数据方向位 (R/W)，该位为 0 表示主机向从机发送数据，该位为 1 表示主机请求从从机读取数据。从机接收到主机发送的 7 位寻址地址以后，会跟从机本身的地址进行比较，如果地址相同，从机向主机发送相应信号 (ACK)。如果主机发送的地址跟从机本身的地址不同，从机则不响应该寻址。

当主机发送的从机地址为 0 时，主机通过广播呼叫来寻址 I2C 总线上的每一个器件，支持广播呼叫功能的器件如果需要跟主机交互数据，需要发送响应 (ACK)。



3.8.3 数据传输时序

主机和从机之间的数据传输遵从高位先传低位后传原则。

主机发送数据，从机接收数据时序如下图 44。

主机接收数据，从机发送数据时序如下图 45。

复合传输格式，传输改变方向的时候，起始条件和从机地址都会被重复发送，但 R/W 位取反。复合传输时序如下图 46。

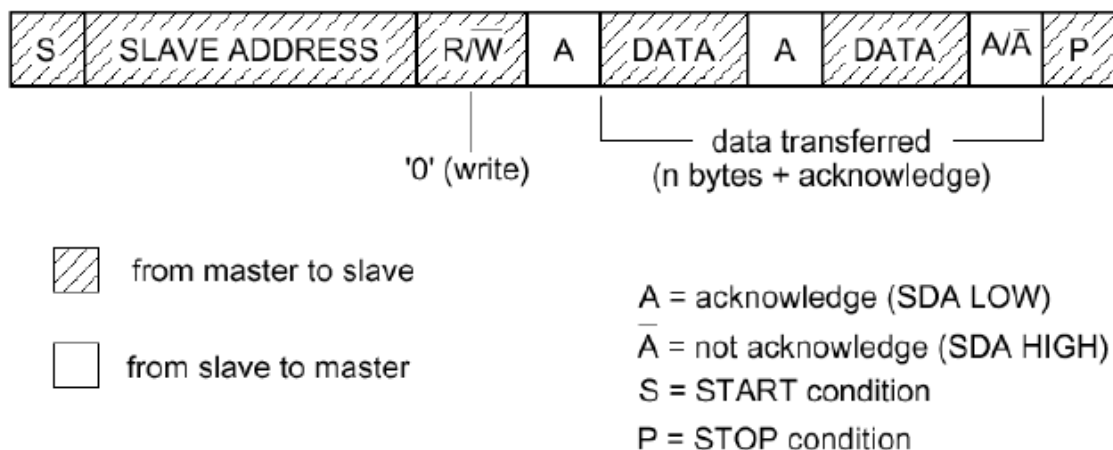


图 44 主机发送从机接收时序图

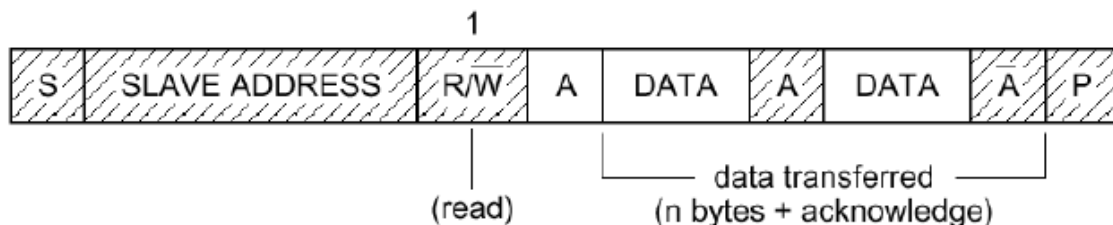


图 45 主机接收从机发送时序图

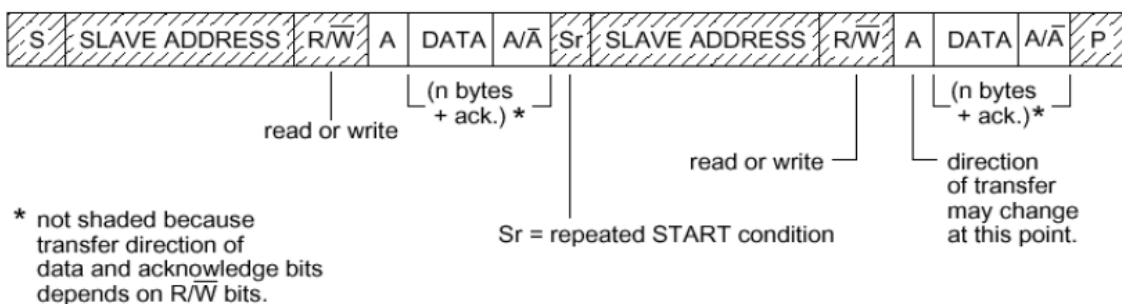


图 46 复合传输时序图

3.8.4 寄存器说明

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
5A	INTF3	I2C_TIF	I2C_RIF	UR1E RR_E	UR1W K_IF	UR0WK _IF	TM3IF / I2C_S T_IF	RTCIF	SPIIF	00000000
5B	INTE3	I2C_TIE	I2C_RIE	UR0E RR_IE	UR1W K_IE	UR0WK _IE	TM3IE / I2C_S T_IE	RTCIE	SPIIE	00000000
68	I2CCON	I2C_EN	AWK_EN	CST_EN	ACK_EN	I2CSTUS[3:0]				00000000
69	I2CDAT	-	I2CDAT[7:0]							00000000
7F	I2CADR	I2CADR[6:0]							GC_EN	01001100

表 48 INTF3 寄存器各位功能表

位地址	标识符	功能
6	I2C_RIF	I2C 从机接收中断标志位 当 I2C_EN=1 且 AWK_EN=0 时： 1：I2C 从机完成 1-Byte 数据并且发送 ACK 后产生接收中断，必须软件清 0 0：没发生 I2C 从机接收中断 当 I2C_EN=0 且 AWK_EN=1 时： 1：在 Sleep 模式下 I2C 从机完成地址匹配产生中断，唤醒系统，系统唤醒后需要软件配置 AWK_EN=0 清除该中断（当前地址匹配会被 I2C 从机忽略，如需数据传输需要 I2C 主机重新寻址从机） 0：没发生 I2C 从机地址匹配中断
3	TM3IF/I2C_ST_IF	当 METCH[5]=0 时为 TM3IF 中断标志位 当 METCH[5]=1 时为 START 中断标志位

表 49 INTE3 寄存器各位功能表

位地址	标识符	功能
6	I2C_RIE	I2C 从机接收中断使能 1：使能 I2C 从机接收中断 0：关闭 I2C 从机接收中断
3	TM3IE/I2C_ST_IE	当 METCH[5]=0 时为 TM3IF 中断使能位 当 METCH[5]=1 时为 I2C START 中断使能位

表 50 INTF3 寄存器各位功能表

位地址	标识符	功能
7	I2C_TIF	I2C 从机发送中断标志位 1：I2C 从机请求 MCU 写入发送数据中断，只能通过往 I2CDAT 写入数据清 0，默认值为 1 0：没发生 I2C 从机请求 MCU 写入发送数据中断

表 51 INTE3 寄存器各位功能表

位地址	标识符	功能
7	I2C_TIE	I2C 从机发送中断使能 1 : 使能 I2C 从机发送中断 0 : 关闭 I2C 从机发送中断

I2CCON 寄存器

位地址	标识符	功能
7	I2C_EN	I2C 从机使能 1: 使能 I2C 从机 (需要关闭异步唤醒使能 AWK_EN=0) 0: 关闭 I2C 从机
6	AWK_EN	I2C 异步唤醒使能 1: 在系统处于 Sleep 模式, 如果 I2C 从机接收到主机发送的地址跟本机的地址相同 (如果 GC_EN=1 则可以相应广播地址), 则通过 I2C_RIF 给出中断, 唤醒系统 (需要关闭 I2C 从机使能 I2C_EN=0) 0: 关闭异步唤醒
5	CST_EN	I2C 从机时钟低电平延长使能 1: 使能 I2C 从机时钟低电平延长 0: 关闭 I2C 从机时钟低电平延长
4	ACK_EN	I2C 从机响应使能 1: 使能 I2C 从机响应 0: 关闭 I2C 从机响应
3: 0	I2CSTUS	I2C 从机状态寄存器, 只读 4'h0 : I2C_IDLE 4'h1 : I2C_STDET 4'h2 : I2C_ADDR 4'h3 : I2C_ADDRACK 4'h4 : I2C_RX 4'h5 : I2C_RXACK 4'h6 : I2C_RXNACK 4'h7 : I2C_TX 4'h8 : I2C_TXACK 4'h9 : I2C_CLKSTR 4'hF : I2C_ERR

I2CDAT 寄存器

位地址	标识符	功能
7: 0	I2CDAT	I2C 数据寄存器 当 I2C 工作在从机接收模式, 完成 1-Byte 数据接收后可以从该寄存器读出接收的数据 当 I2C 工作在从机发送模式, 写入该寄存器的数据将被发送给主机

I2CADR 寄存器

位地址	标识符	功能
7: 1	I2CADR	I2C 从机地址, 默认为 0x26
0	GC_EN	I2C 从机广播地址响应使能 1: 使能 I2C 从机广播地址响应 (是否响应还要看 ACK_EN 的状态) 0: 关闭 I2C 从机广播地址响应

3.8.5 I2C 使用说明

I2C Slave 的初始化:

- 1) 如果需要支持 I2C 发送数据提前写入功能，需要配置 FRACTION 寄存器的 Metch[5]=1；如果不需要支持 I2C 发送数据提前写入功能，配置 FRACTION 寄存器的 Metch[5]=0。
- 2) 根据 I2C 的工作频率，需要配置 I2C 从机的时钟分频系数 I2C_DIV[1:0]，具体配置参照下面的表格。

I2C 通信速率 (K bit/s)	ICK	指令周期	I2C_DIV[1:0]
400	8MHz 或 10.5MHz	ICK /2 或 ICK /4	0
100	8MHz 或 10.5MHz	ICK /2 或 ICK /4	0 或 1
100	4MHz 或 5.25MHz	ICK /2	0

- 3) 配置 I2C 从机的地址 I2CADR[7:1]，默认为 0x26。
- 4) 使能 I2C 从机，关闭异步唤醒功能，I2C_EN=1，AWK_EN=0，根据需求使能低时钟延长，CST_EN=1，使能从机响应，ACK_EN=1。
- 5) I2C 从机自动检测 I2C 主机发送的起始状态 START 或 RESTART，并把接收到的地址跟本机地址进行比较，如果地址匹配则发送响应给主机，准备收发数据。

I2C Slave 的接收

当 I2CDAT 数据接收到数据后，会置 I2C_RIF 标志位，如果中断使能 I2C_RIE 打开，会报中断。对 I2CDAT 读取数据，会清除中断标志 I2C_RIF。
注：当 I2C 数据溢出后，会发送 NACK。

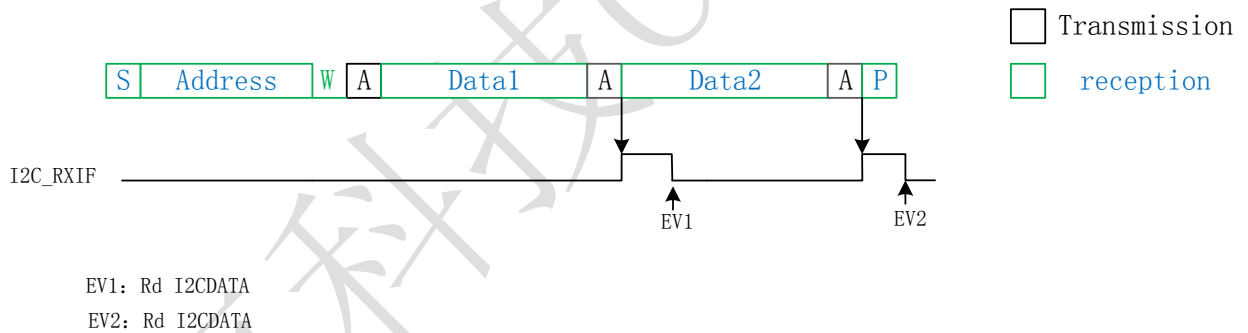


图 47 从机连续接收时序图

I2C Slave 的发送:

不需要支持 I2C 从机发送提前写入功能的流程如下:

配置 FRACTION 寄存器的 Metch[5]=0。

当 I2CDAT 为空时，I2C_TIF 的标志位 1，上电默认值也为 1。如果中断使能 I2C_TIE 打开，报中断。硬件读取 I2CDAT 和 I2CSlave 接收到数据，都会清除中断标志 I2C_TIF。

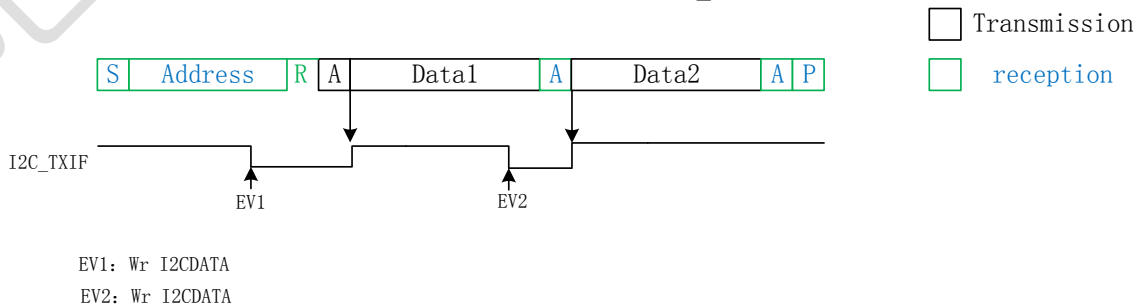


图 48 从机连续发送时序图

I2C Slave 发送中 CLOCK STRETCH:

当 I2CDAT 为空时, I2C_TIF 标志位为 1, 此时地址完成匹配并且 R/W=1, 需要发送数据, 如果 CST_EN=1, I2C Slave 进入 CLOCK STRETCH。

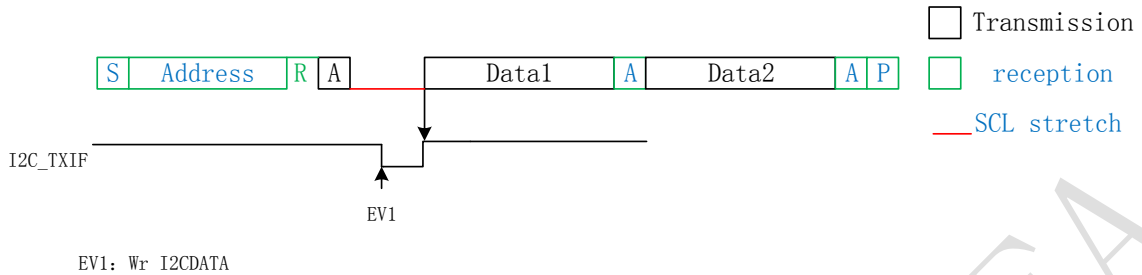


图 49 从机发送 SCL stretch 时序图

当时序上需要 I2C 从机发送数据, 这时 I2CDAT 为空且没有打开时钟延长功能 CST_EN=0, 如果 I2C 从机地址匹配, 则回复 ACK 并跳出发送周期。如果从机地址不匹配, 则直接回复 NACK。

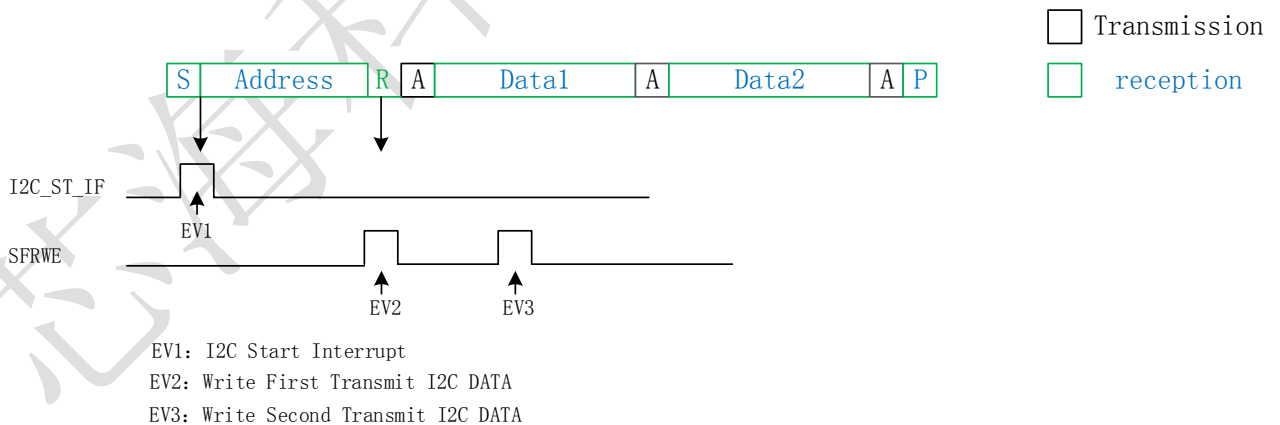
支持 I2C 从机发送提前写入功能的流程如下:

配置 FRACTION 寄存器的 Match[5]=1。

这时原来 TIM3IF 中断变为 I2C 从机接收到 START 信号中断, 使能该中断, 配置 TM3IE/I2C_ST_IE=1。

系统提前预知当前传输是 I2C 主机来读该芯片中的 I2C 从机, 在系统收到到 I2C Start 中断时, 提前写入需要发送的数据, 在需要从机发送数据时, 发送数据已经准备好, 在不开时钟延长功能或者主机不支持时钟延长功能的情况下, 通过提前写入数据保证不会出现发送数据来不及准备的传输错误。

注: 由于从 START 信号到数据发送只有 8 个 I2C 总线时钟周期, 如果 I2C 总线跑在 400KHz, 系统中断中需要处理的内容过多, 打开提前写入功能还是会出现系统来不及写入发送数据, 导致传输出错的情况。



当 I2CDAT 为空时, I2C_TIF 的标志位 1, 上电默认值也为 1. 如果中断使能 I2C_TIE 打开, 报中断。硬件读取 I2CDAT 和 I2CSlave 接收到数据, 都会清除中断标志 I2C_TIF。

I2C Slave 低功耗模式下唤醒

在系统进入 Sleep 模式之前，配置 I2C_EN=0， AWK_EN=1， CST_EN=0， ACK_EN=0， 系统进入睡眠模式后，当主机发送的地址与从机相同（如果 GC_EN=1 则可以相应广播地址），I2C 从机产生 I2C_RIF 中断，唤醒系统。系统唤醒后，在正常工作模式需要把异步唤醒功能关闭 AWK_EN=0。为了简化应用，不建议把 I2C 从机异步唤醒功能和 I2C 从机正常数据收发功能同时使用。

3.9 24 Bits ADC 模块

芯片包含一个 24 位的 sigma delta 型的模数转换器（ADC）。该 ADC 的基准由内部的基准电压 VS 提供，但是也可以由外部提供基准电压（此时需要关闭内部的基准电压，将外部的基准电压接入 REFP）。同时还提供内部温度检测以及内短测试功能。

首先打开 VS (LDOEN)，然后配置好 24 Bits ADC 的各种配置参数，打开全局中断使能以及 24 Bits ADC 中断使能，然后将 ADEN 打开，在刚开始工作时，数字滤波器需要一段建立时间，对于三阶数字滤波器，需要三个数据转换周期的建立时间。

3.9.1 24 Bits ADC 寄存器说明

表 52 24 Bits ADC 功能模块相关寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF	-	TM2IF	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	u000000 0
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	000000 0
10H	ADOH	ADO[23:16]								000000 0
11H	ADOL	ADO[15:8]								000000 0
1CH	ADOLL	ADO[7:0]								000000 0
6BH	ADCON	METCH[3:0]				ADM[3:0]				000000 0
16H	ADCFG	ADSC	GAIN1[1:0]	S_GAIN[1:0]		R_GAIN	CFG_C[1:0]			000000 0
17H	ANACFG	LDOEN	LDOS[1:0]	BGR_EN	BGID	SINL[1:0]		ADEN	000000 0	
18H	TEMPC	TEMPC[7:0]								000000 0

ADOH 寄存器（地址为 10H）

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOH	ADO[23:16]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

ADOL 寄存器（地址为 11H）

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

ADOLL 寄存器（地址为 1CH）

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

Bit 23-0 ADO[23:0]: 24 Bits ADC 数字输出

ADO[23] = 24 Bits ADC 数字输出符号位。0 = 输出为正；1 = 输出为负。

ADO[22] = 24 Bits ADC 数字输出数据 bit23

~

ADO[0] = 24 Bits ADC 数字输出数据 bit0

ADCON 寄存器 (地址为 6BH)

特性	RW-0	RW-0	RW-0	RW-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCON	METCH[3:0]				ADM[3:0]			
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit3-0 ADM[3:0]:24 Bits ADC 降采样速率选择寄存器

表 53 24 Bits ADC 输出速率选择列表

ADM[3:0]	第一次数据稳定时间	DataRate(Hz)
0000	256 *T _{CK_SMP}	CK_SMP/64
0001	512 *T _{CK_SMP}	CK_SMP/128
0010	1024*T _{CK_SMP}	CK_SMP/256
0011	2048*T _{CK_SMP}	CK_SMP/512
0100	4096*T _{CK_SMP}	CK_SMP/1024
0101	8192*T _{CK_SMP}	CK_SMP/2048
0110	16384*T _{CK_SMP}	CK_SMP/4096
0111	16384*T _{CK_SMP}	CK_SMP/8192
1000	20495*T _{CK_SMP}	CK_SMP/16384
1001	24583*T _{CK_SMP}	CK_SMP/16384
1010	40979*T _{CK_SMP}	CK_SMP/32768
1011	49167*T _{CK_SMP}	CK_SMP/32768
1100	81970*T _{CK_SMP}	CK_SMP/65536
1101	24576*T _{CK_SMP}	CK_SMP/8192
1110	49152*T _{CK_SMP}	CK_SMP/16384
1111	98304*T _{CK_SMP}	CK_SMP/32768

ADCFG 寄存器 (地址为 16H)

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
ADCFG	ADSC	GAIN1[1:0]		S_GAIN[1:0]		R_GAIN	CFG_C[1:0]	
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 1:0 CFG_C[1:0]: 配置成 00 不变

ADC 的采样时钟频率选择 CK_SMP:

ADSC	GAIN1[1:0]	CK_SMP
0	00	ICK/8
0	01	ICK/32
0	10	ICK/48
0	11	ICK/64
1	00	ICK/4
1	01	ICK/16
1	10	ICK/24
1	11	ICK/32

ADC 的 PGA 选择:

Gain	GAIN1<1:0>	R_GAIN	S_GAIN <1:0>	bgid 配置
1	00	0	00	0
10	00	0	01	0
20	00	0	10	0
80	01	0	01	0
160	11	0	01	0
320	11	0	10	1

对于 GAIN=320 时, VS 电压值只能选择 VS=3.3V 或 VS=3V, 即 LDOS[1:0]只能配置成 00 或 01。

ANACFG 寄存器 (地址为 17H)

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
-	LDOEN	LDOS[1:0]		BGR_ENB	BGID	SINL[1:0]		ADEN
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 LDOEN: LDO 电源使能信号
 LDOEN=1: LDO 电源使能
 LDOEN=0: LDO 电源不使能

Bit 6:5 LDOS[1:0]: VS 电压值选择
 LDOS[1:0] 00 VS=3.3V
 LDOS[1:0] 01 VS=3.0V
 LDOS[1:0] 10 VS=2.45V
 LDOS[1:0] 11 VS=2.4V

Bit 4 BGR_ENB: bandgap 使能
 BGR_ENB =0 打开 bandgap
 BGR_ENB =1 关闭 bandgap

Bit 3 BGID: ADC 模式选择。
 bgid=0, 当 VS=2.4V、2.45V、3.0V、3.3V 可以配置这个档位, PGA=1, 10, 20, 80, 160。
 bgid=1, 只有 VS=3.0V、3.3V 可以配置这个档位, 仅限 PGA=320。

Bit 2:1 ADC 通道选择
 SINL[1:0] :
 00 = 24 Bits ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 Vin + , AIN1 为 Vin -;
 01 = 内短;
 10 = 24 Bits ADC 输入端连接到 TEMP;
 11 = 不可选。

Bit 0 ADEN: 24 Bits ADC 使能标志
 1 = 24 Bits ADC 使能
 0 = 24 Bits ADC 不使能

TEMPC 寄存器 (地址为 18H)

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
TEMPC	TEMPC [7:4]				TEMPC[3:0]			
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

TEMPC[7:4]: 温度补偿选择。

TEMPC[7:5]	TEMPC[4]	增益温度特性补偿(ppm)@2.4V
011	0	-130
010	0	-90
001	0	-45
000	0	0
100	0	0
101	0	45
110	0	90
111	0	130

011	1	-13
010	1	-9
001	1	-4
000	1	0
100	1	0
101	1	4
110	1	9
111	1	13

Tempc[3:0]: 配置 0000

3.10 10 Bits ADC 模块

芯片内置一个 10 Bits 的 ADC，输出数据位宽为 10 位，信号由 PT1.0/PT1.1/PT2.0/PT2.2 等口输入。

3.10.1 10 Bits ADC 寄存器说明

表 54 10 Bits ADC 功能模块相关寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF	-	TM2IF	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	u0000000
07H	INTE	GIE	TM2IE	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	00000000
19H	AD2OH	AD2EN	AD2START	AD2_SL[3:0]			AD2O[9:8]			00000000
1AH	AD2OL	AD2O[7:0]								00000000
1BH	LVDCON	LVDEN	AD2_REF	AIENB1	SILB[2:0]		LBOUT	LB_RST_CON		001000x0

AD2OH 寄存器（地址为 19H）

特性	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	R-0	R-0
AD2OH	AD2EN	AD2_START	AD2_SL[3:0]				AD2O[9:8]		
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit7 AD2EN: 10 Bits ADC 使能信号: AD2EN 从 0 变 1 之后数据开始转换, 转换完成之后产生中断标志。

=0, 10 Bits ADC 不使能;
=1, 10 Bits ADC 使能;

Bit6 AD2_START: 10 Bits ADC 的转换开始信号, 软件置 1, 当转化完成后硬件清 0。

=0, 10 Bits ADC 转换完成, 软件写 0 无效, 硬件自动清零。
=1, 开始进行 10 Bits ADC, 软件写 1 有效;

Bit5:2 AD2_SL[3:0]: 10 Bits ADC 通道选择。

=0000, PT2.7;
=0001, PT2.6;
=0010, PT2.3;
=0011, PT2.2;
=0100, PT2.1;
=0101, PT2.0;
=0110, PT1.1;
=0111, PT1.0;
=1000, VS;
=1001, DVDD/2;
=其他, DVDD/2;

Bit1:0 AD2O[9:8]: 10 Bits ADC 数据的高两位。

LVDCON 寄存器（地址为 1BH）

特性	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	R-0	R-0
LVDCON	LVDEN	AD2_REF	AIENB1	SILB[2:0]		LBOUT	LB_RST_CON	
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit6 AD2_REF: 10 BitsADC 参考电压选择。

=0, DVDD;
=1, LDO 输出电压;

AD2OL 寄存器（地址为 1AH）

特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
AD2OL	AD2O[7:0]							
-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7:0 AD2OL[7:0]: 10 Bits ADC 数据的低 8 位。

3.10.2 10 Bits ADC 的操作步骤

1. 使能相应的中断
2. 置位 AD2EN
3. 置位 AD2START
4. 等待中断，读取 AD2O[9:0]。
5. 中断响应之后，清除中断
6. 关闭 AD2EN

注:每次数据转换必须在 AD2EN=1 的前提下，把 AD2START 从 0 变 1。

3.11 SPI 接口介绍

芯片支持 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 SCLK, 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 SCLK, 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（SCLK）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

SCLK: 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

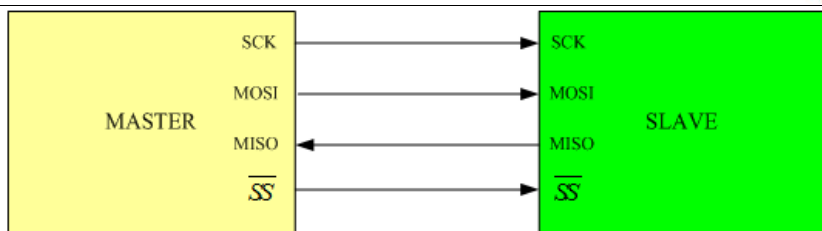
当 SPI 做为主机时该信号为从机选择输出，做为从机时为从机选择输入

SPI 串行时钟支持 4 种方式，即两种相位和两种极性组合

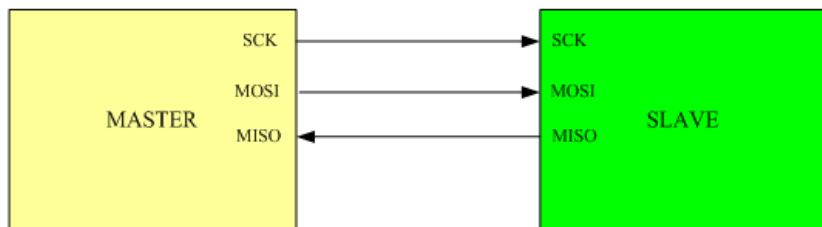
相位：0 表示在 SCLK 奇数边沿（1,3,5,7.....）采样数据，1 表示在 SCLK 偶数边沿（2,4,6,8.....）采样数据

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

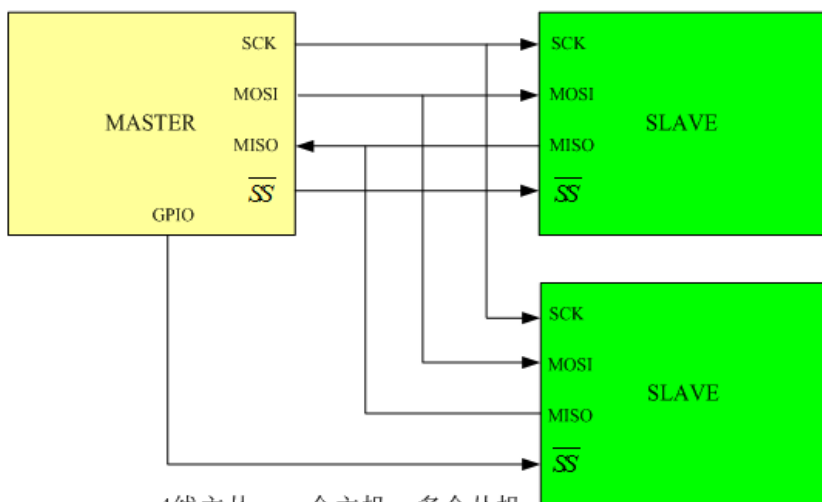
3.11.1 SPI 连接方式



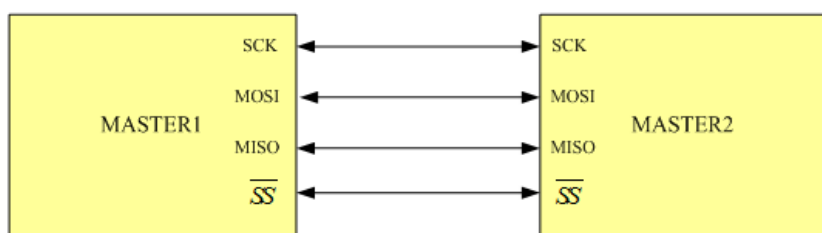
4线主从



3线主从



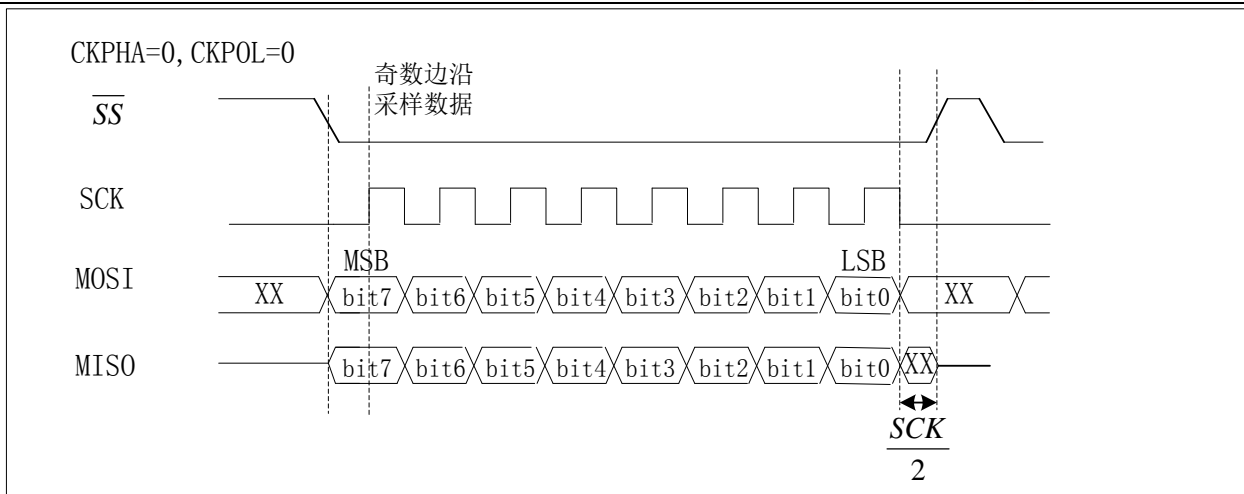
4线主从，一个主机，多个从机



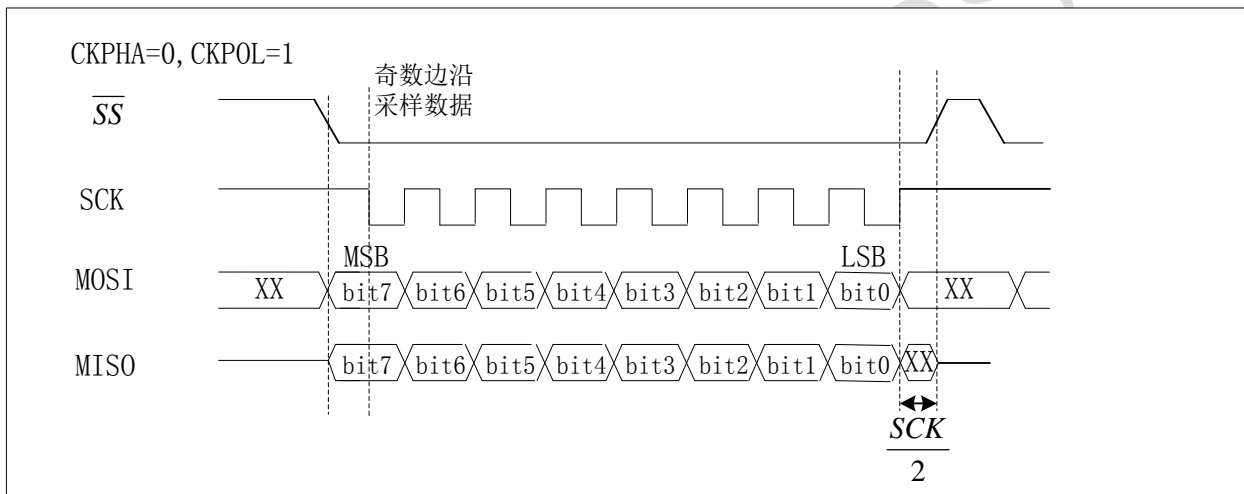
多主机模式

3.11.2 SPI 时序图

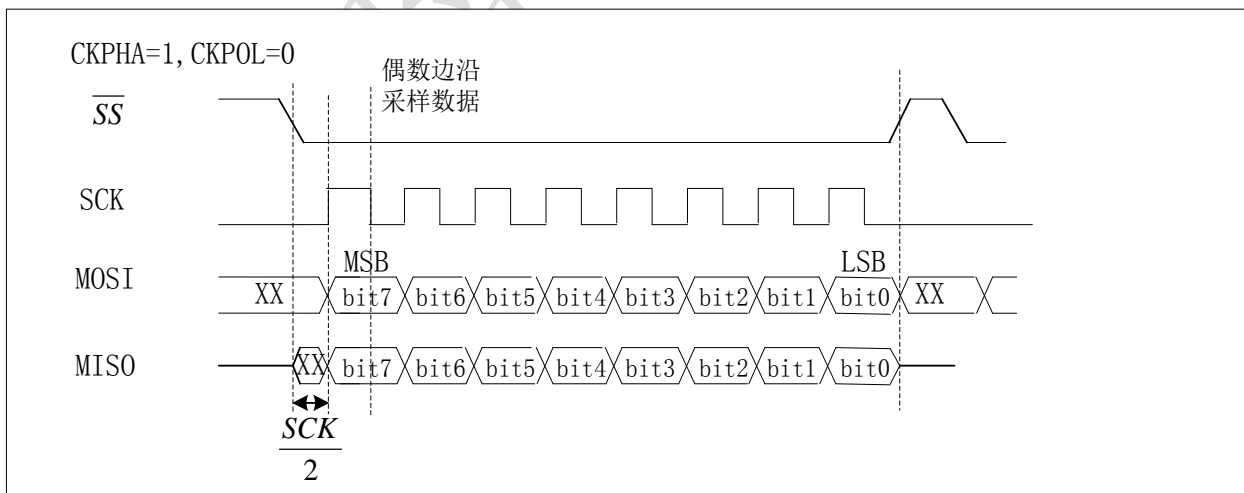
(1) CKPHA=0,CKPOL=0



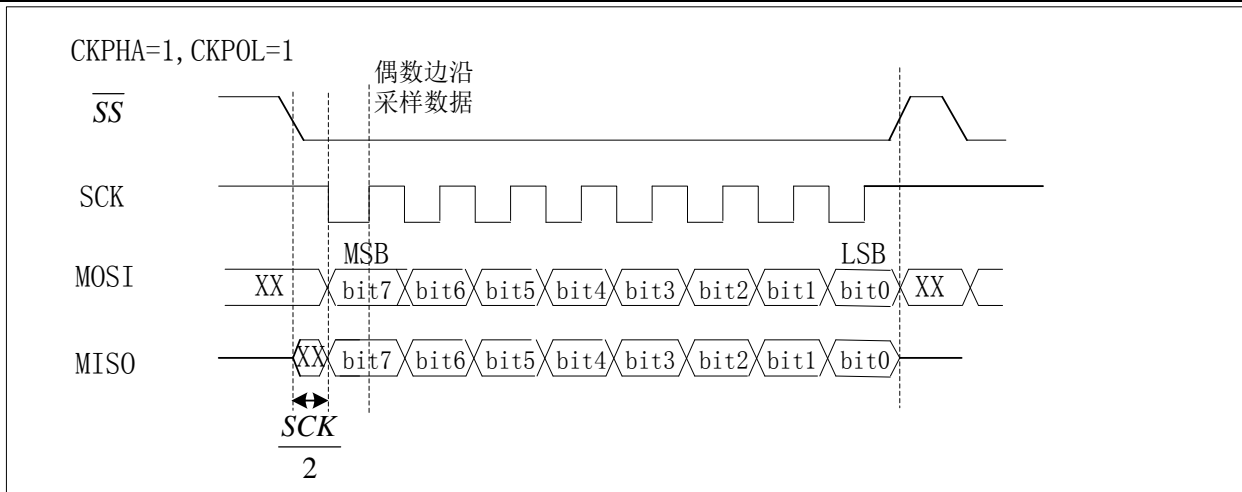
(2) CKPHA=0,CKPOL=1



(3) CKPHA=1,CKPOL=0



(4) CKPHA=0,CKPOL=1



3.11.3 SPI 寄存器说明

表 55 SPI 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
5AH	INTF3	I2C_TIF	I2C_RIF	UR1ERR_I E	UR1WK_I F	UR0WK_IF	TM3IF	RTCIF	SPIIF	00000000
5BH	INTE3	I2C_TIE	I2C_RIE	UR0ERR_I E	UR1WK_I E	UR0WK_IE	TM3IE	RTCIE	SPIIE	00000000
5CH	SPICFG	SPIEN	MSTEN	CKPHA	CKPOL	MULMST	WIREMOD	SPIBSY	SPIRST	00000000
5DH	SPICN	SLVSEL	WCOL	MODCOL	-	-	-	-	-	000uuuuu
5EH	SPICKR	SCR[7:0]								00000011
5FH	SPIDAT	SPIDAT[7:0]								00000000

表 56 INTF3 寄存器各位功能表

位地址	标识符	功能
0	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 57 INTE3 寄存器各位功能表

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

表 58 SPICFG 寄存器各位功能表

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

		1 : 主模式 0 : 从模式 多主机模式下当 SLVSEL (SPICN[7]) 为 1 时, MSTEN 自动清零
5	CKPHA	SPI 时钟相位控制 1 : 在 SCLK 偶数边沿采样数据 0 : 在 SCLK 奇数边沿采样数据
4	CKPOL	SPI 时钟极性控制 1 : 时钟 SCLK 在总线空闲时是高电平 0 : 时钟 SCLK 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 SCLK, MOSI, MISO, \overline{SS} 1 : SPI 为 3 线通信模式, 即包括 SCLK, 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 59 SPICN 寄存器各位功能表

位地址	标识符	功能
7	SLVSEL	从模式标志 (只读) 1 : \overline{SS} 输入为 0, 从模式下 SPI 被选中 0 : \overline{SS} 输入为 1, 从模式下 SPI 未被选中
6	WCOL	写冲突标志 (软件清 0) 1 : 当仍有数据在发送时试图向发送数据寄存器中写数据, 将会产生写冲突 0 : 没有写冲突
5	MODCOL	模式失效标志 (软件清 0) 1 : 多主模式时产生了主模式冲突 (\overline{SS} 为低, MSTEN: 1) 0 : 多主模式时未产生主模式冲突
4:0	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 60 SPICKR 寄存器各位功能表

位地址	标识符	功能
7: 0	SCR [7:0]	SPI 时钟速率

		<p>SCR 决定了 SPI 通信中 SCLK 的时钟频率。</p> $f_{sck} = \frac{CPUCLK}{SCR[7:0]+1}$ <p>其中, $3 \leq SCR[7:0] \leq 255$</p>
--	--	---

表 61 SPIDAT 寄存器各位功能表

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

3.11.4 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 INTF2,4 ;清 SPI 中断标志位
BSF INTE2,4 ;使能 SPI 中断
BSF INTE,GIE ;打开中断总使能
    
```

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 INTF2,4 ;清 SPI 中断标志位
BSF INTE2,4 ;使能 SPI 中断
BSF INTE,GIE ;打开中断总使能
    
```

```
BSF    SPICFG,7 ;使能 SPI
MOVLW  a5h
```

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 (PT4.5) 上需要接上拉电阻, 在空闲时 \overline{SS} 能自动回到高电平。

3.12 LCD Driver

LCD 驱动器有 20 个 segment 输出 (SEG1-SEG20) 和 6 个 common 输出 (COM1-COM6)

3.12.1 LCD 控制模式

LCD 驱动器有 4 种控制模式: 1/2duty, 1/3duty, 1/4duty 及 1/6duty, 设置寄存器标志 LCD_DUTY[1:0] 选择一种模式。

表 62 LCD 的 duty 选择列表

LCD_DUTY[1:0]	控制模式	SEG1-20							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00	1/6duty	-	-	COM6	COM5	COM4	COM3	COM2	COM1
01	1/2duty	-	-	-	-	-	-	COM2	COM1
10	1/3duty	-	-	-	-	-	COM3	COM2	COM1
11	1/4duty	-	-	-	-	COM4	COM3	COM2	COM1

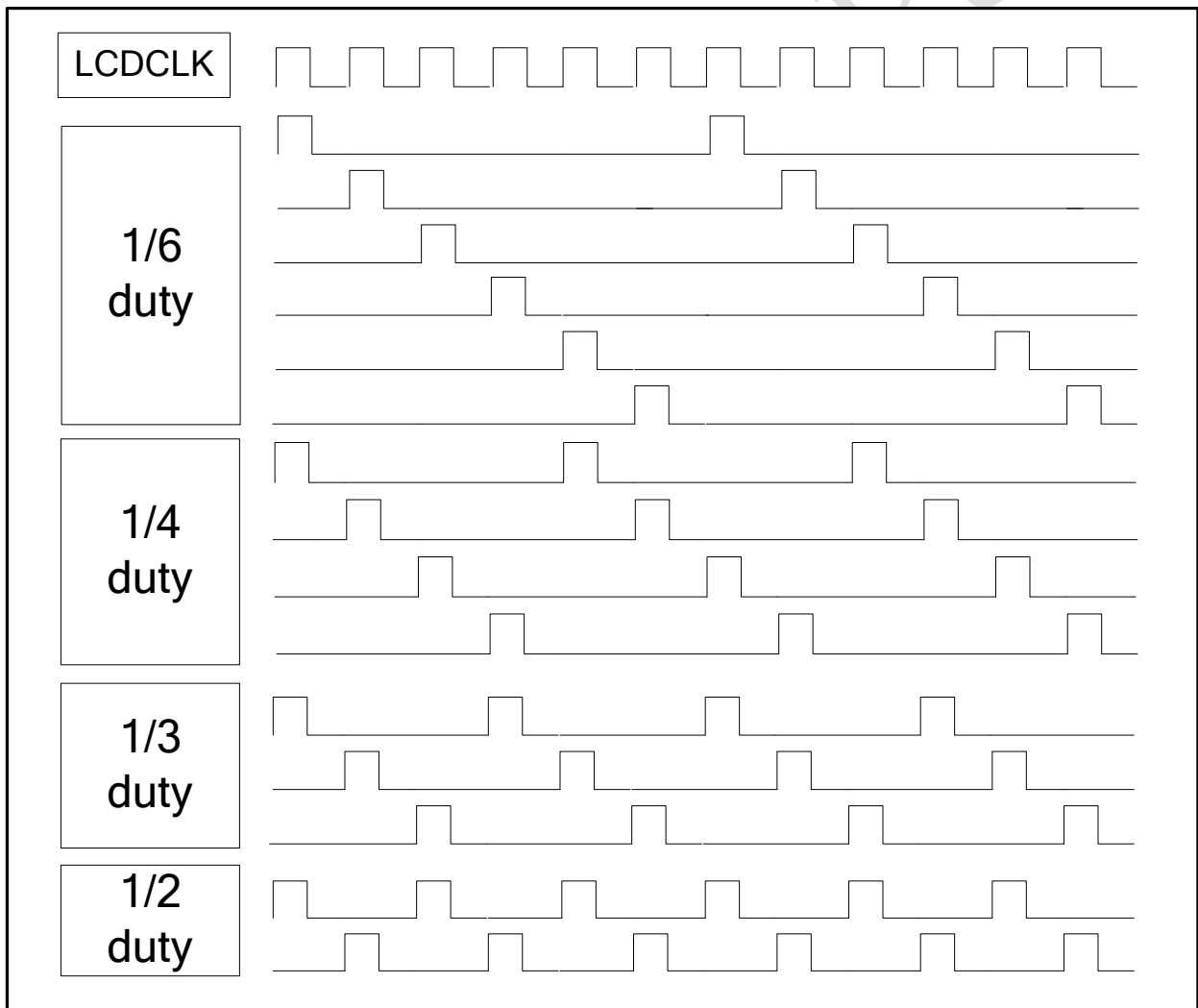


图 50 LCD 的 duty 模式工作周期

3.12.2 LCD 帧频选择

LCD 的帧频率可以通过设置寄存器标志 LCDCKS[1:0]确定。对 LCD 模块的输入时钟进行分频以获得 LCDCLK。

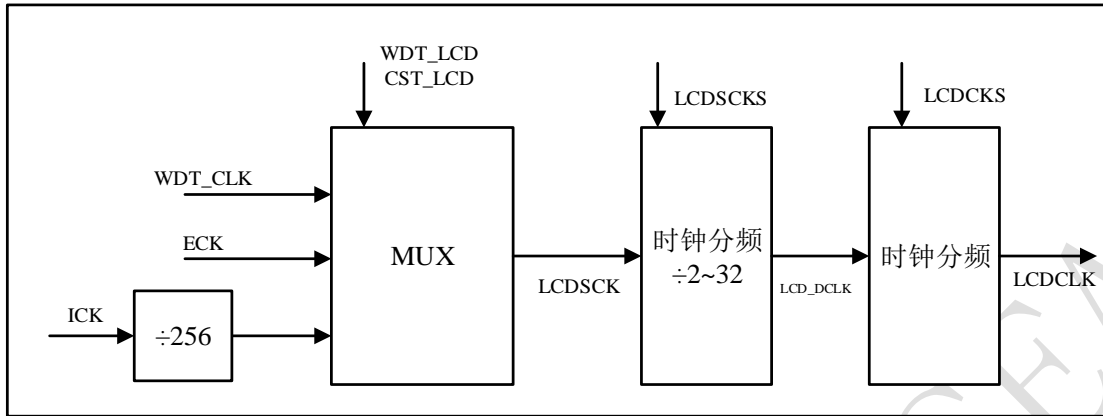


图 51 LCD 帧频时钟选择

表 63 LCDSCK 选择列表

WDT_LCD	CSE_LCD	CLK_SEL	时钟源(Hz)		LCDSCK(KHz)	
1	X	X	WDT	3K	WDT	3
0	1	X	ECK	32K	ECK	32
0	0	0	ICK	10.5M/8M 或 5.2M/4M	ICK/256	41/31 或 20/15
0	0	1	ECK	32	ECK	32

表 64 LCD_DCLK 选择列表

WDT_LCD	LCDSCKS				LCD_DCLK	LCD_DCLK (Hz)		
	@WDT	@ECK	@ICK					
0	1	1	1	1	LCDSCK/32	93.7	1000	1281/976 或 640/488
0	1	1	1	0	LCDSCK/30	100	1066	1367/1041 或 683/520
0	1	1	0	1	LCDSCK/28	107.1	1142	1464/1116 或 732/558
0	1	1	0	0	LCDSCK/26	115.3	1230	1577/1201 或 788/600
0	1	0	1	1	LCDSCK/24	125	1333	1708/1302 或 854/651
0	1	0	1	0	LCDSCK/22	136.3	1454	1864/1420 或 932/710
0	1	0	0	1	LCDSCK/20	150	1600	2050/1562 或 1025/781
0	1	0	0	0	LCDSCK/18	166.6	1777	2278/1736 或 1139/866
0	0	1	1	1	LCDSCK/16	187.5	2000	2563/1953 或 1281/976
0	0	1	1	0	LCDSCK/14	214	2285	2929/2232 或 1464/1116
0	0	1	0	1	LCDSCK/12	250	2666	3417/2604 或 1708/1302
0	0	1	0	0	LCDSCK/10	300	3200	4101/3125 或 2050/1562
0	0	0	1	1	LCDSCK/8	375	4000	5126/3906 或 2563/1953
0	0	0	1	0	LCDSCK/6	500	5333	6835/5208 或 3417/2604
0	0	0	0	1	LCDSCK/4	750	8000	10253/7812 或 5126/3906
0	0	0	0	0	LCDSCK/2	1500	16000	20507/15625 或 10253/7812

表 65 LCDCLK 选择列表

LCDCKS[1:0]	LCD 时钟 (LCDCLK)
00	LCD_DCLK

01	LCD_DCLK/2
10	LCD_DCLK/4
11	LCD_DCLK/8

LCDCLK 的频率计算公式： $f_{LCDCLK} = f_{LCDCLK} / (2 * (LCDCLKS + 1) * 2^{LCDCLKS})$

表 66 LCD 帧频选择列表

LCD_DUTY[1:0]	控制模式	LCD 帧频
00	1/6duty	$(f_{LCDCLK})/6$
01	1/2duty	$(f_{LCDCLK})/2$
10	1/3duty	$(f_{LCDCLK})/3$
11	1/4duty	$(f_{LCDCLK})/4$

3.12.3 LCD 偏置电压

LCD 驱动器有 3 个偏置电压，V1、V2 及 V3，有 2 种电源模式：1/3bias、1/2bias。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。

采用电阻分压的方式产生偏置电压

1/3bias 电源系统

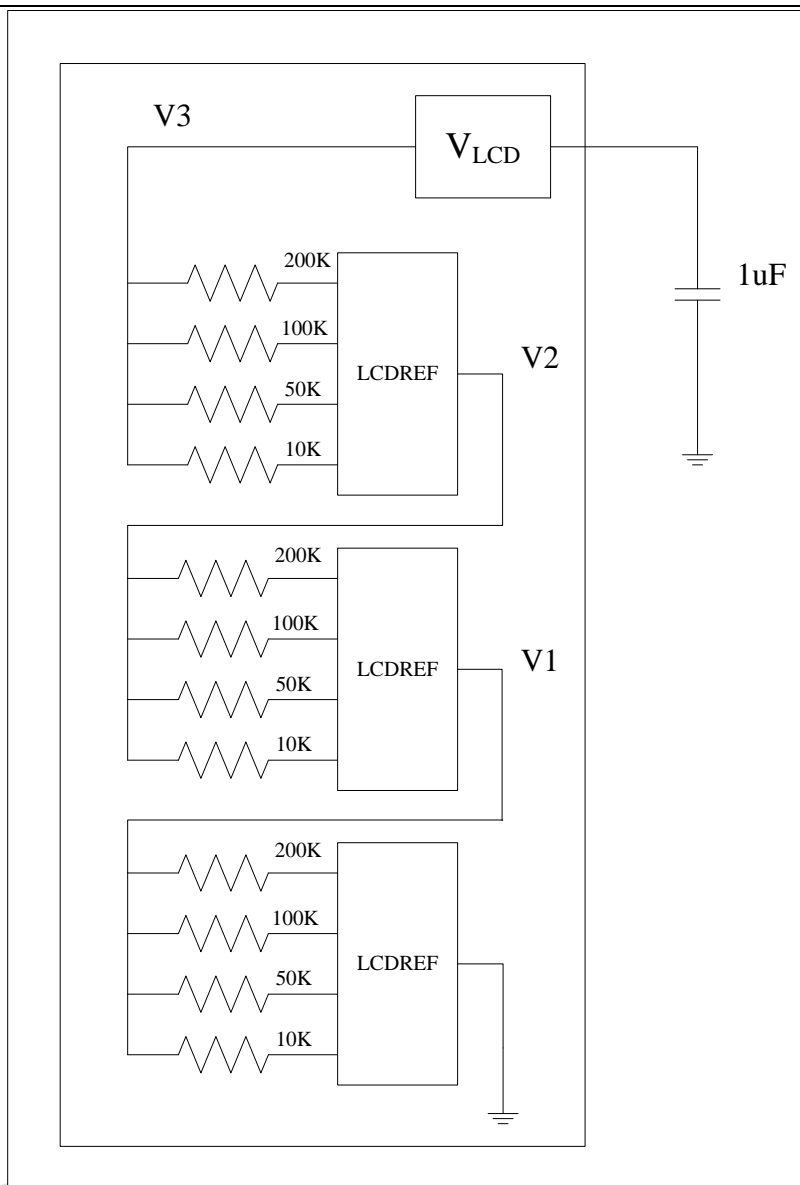


图 52 LCD 的 1/3bias 电源系统电路连接图（电阻分压）

1/2bias 电源系统

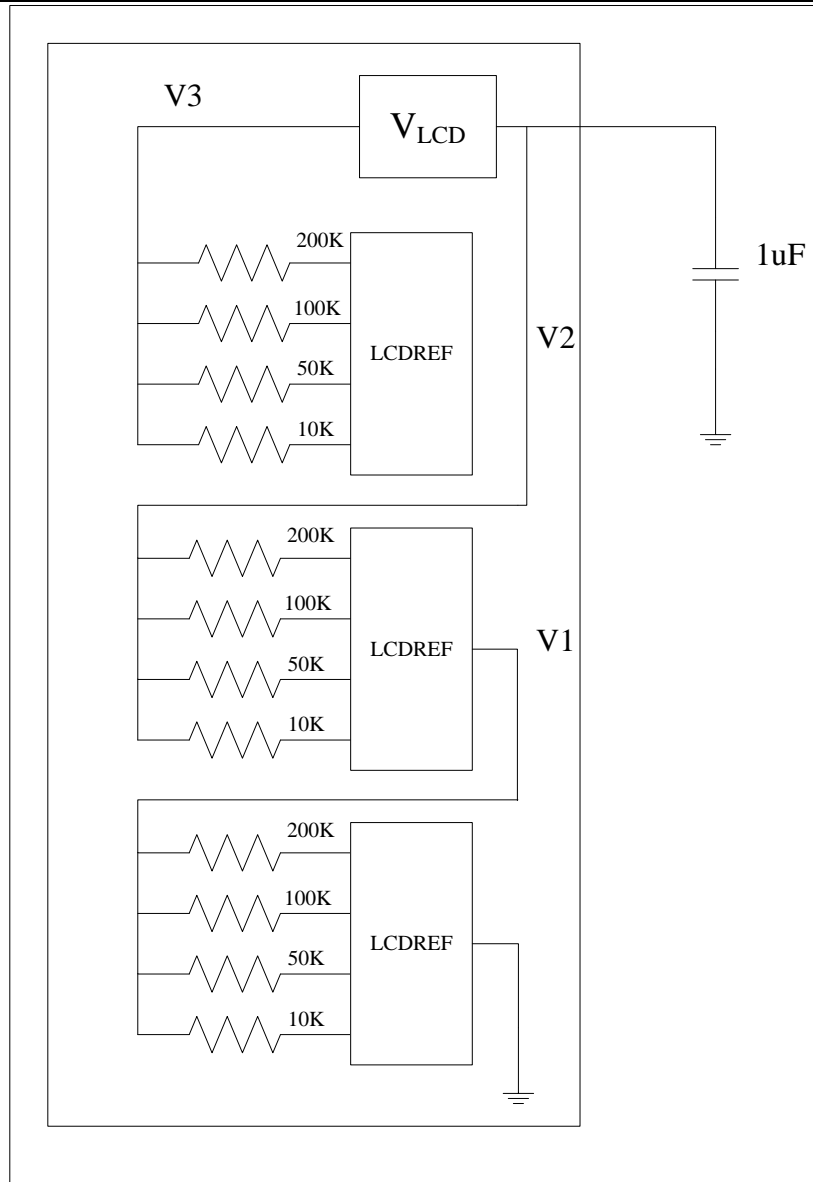


图 53 LCD 的 1/2bias 电源系统电路连接图（电阻分压）

LCD 电源可以采用内部的电荷泵来提供，也可以采用内部的电源来提供，此时需要关闭内部的电荷泵。

3.12.4 LCD 驱动波形

LCD 驱动波形分为 A 和 B 两种波形，通过寄存器 LCDWS 来选择，其中 B 波形对于大尺寸的显示效果更好。

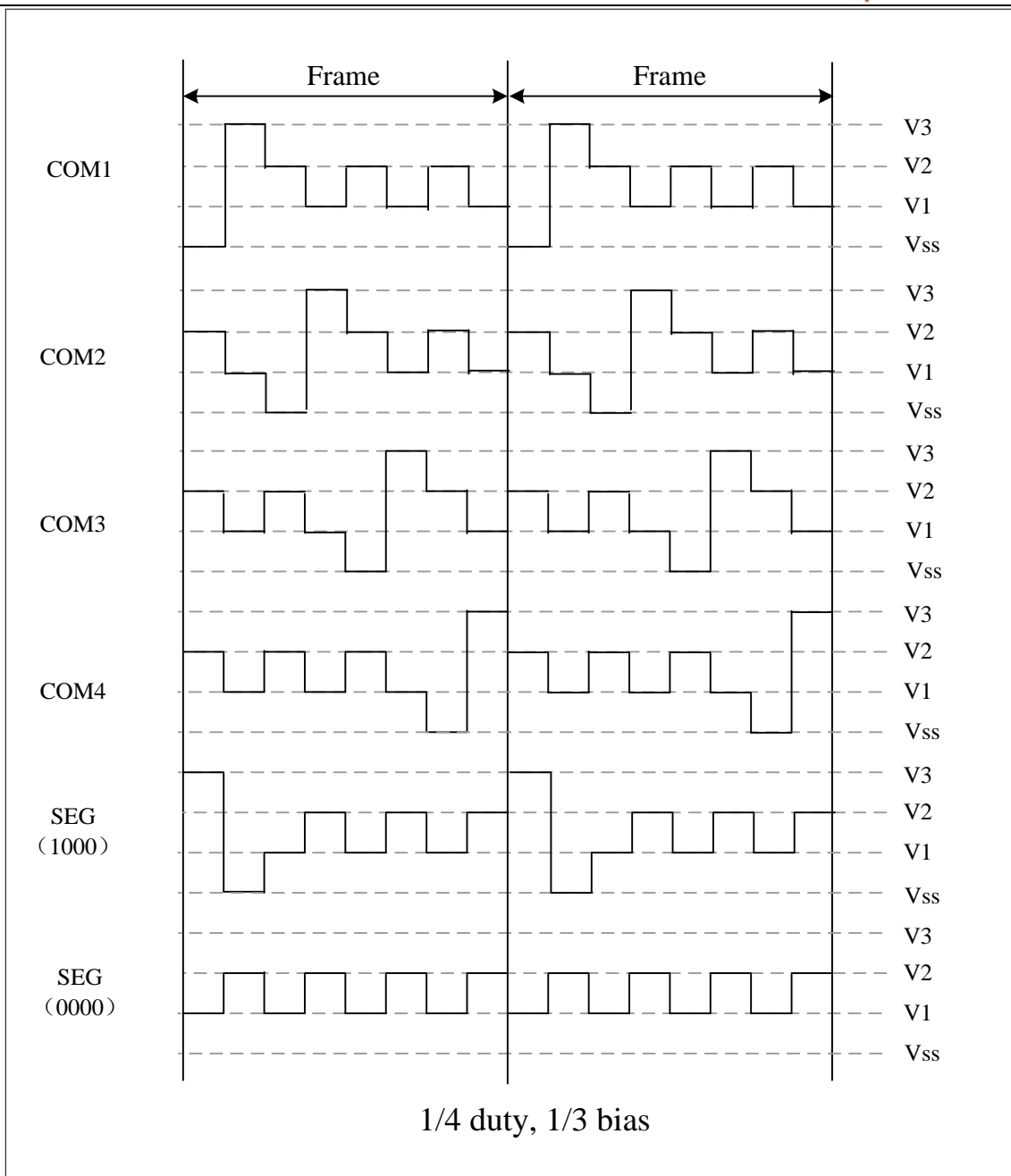


图 54 LCD 的 1/4duty 1/3bias 电源系统的时钟 (A 波形)

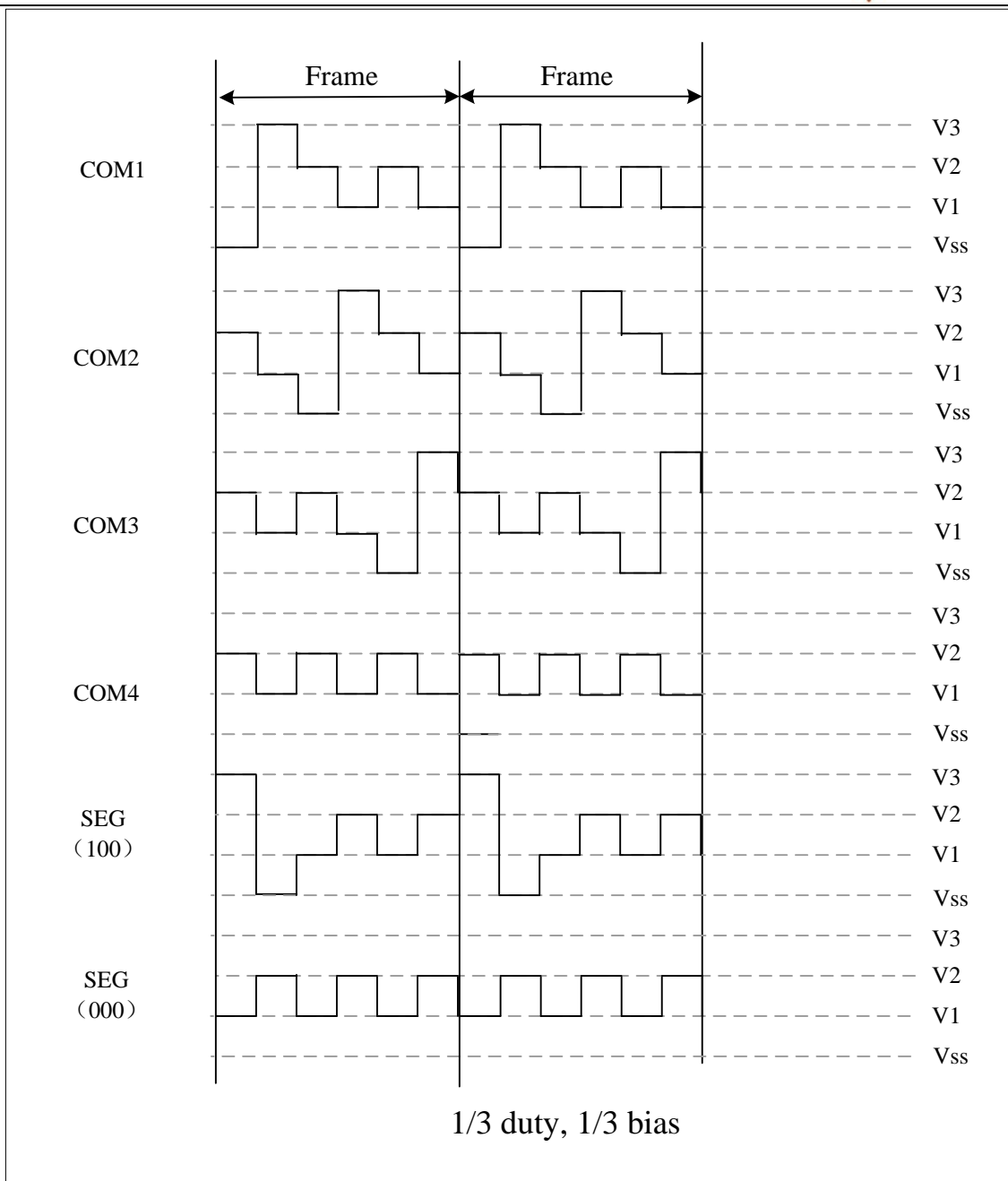


图 55 LCD 的 1/3duty 1/3bias 电源系统的时钟 (A 波形)

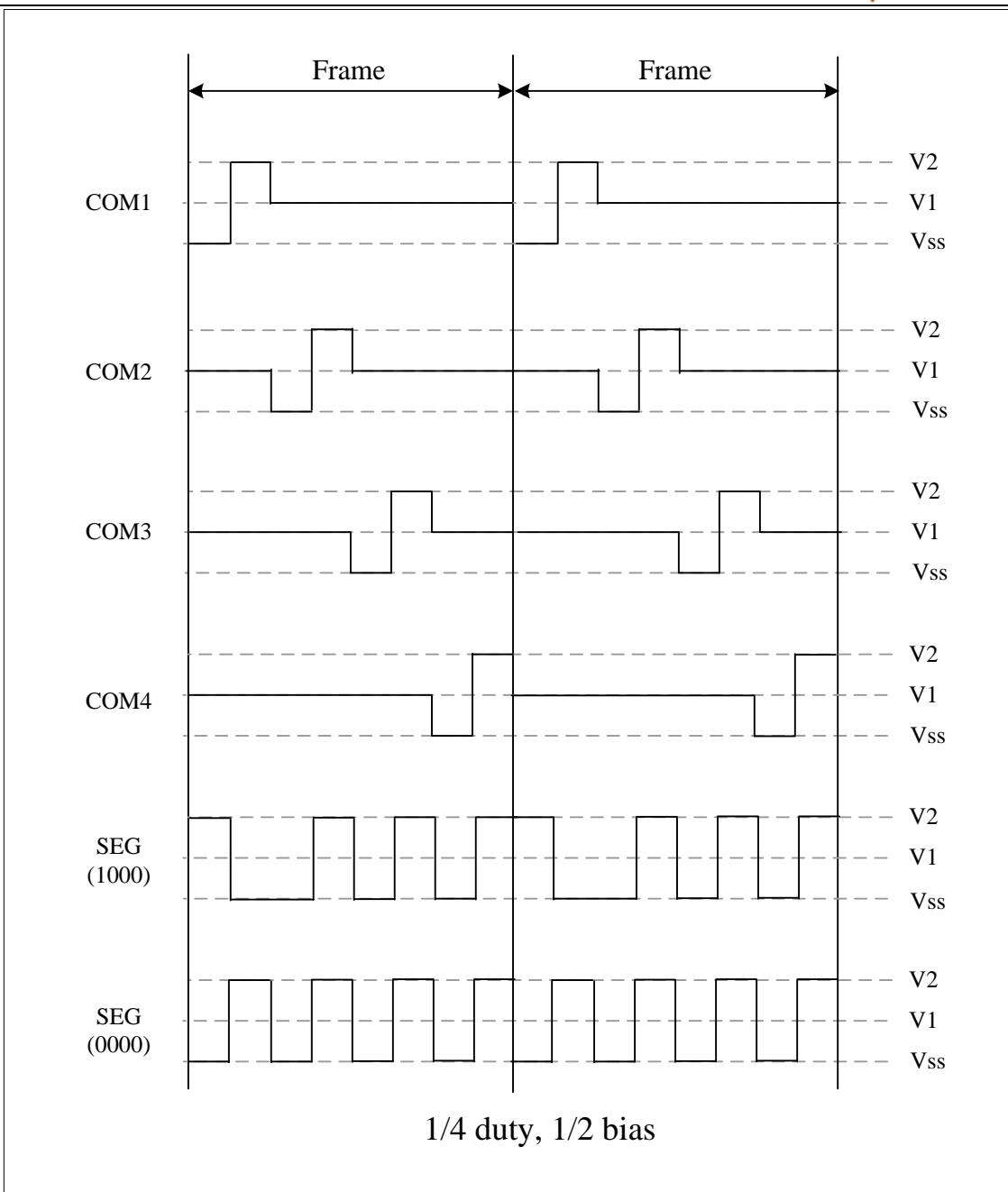


图 56 LCD 的 1/4duty 1/2bias 电源系统的时钟 (A 波形)

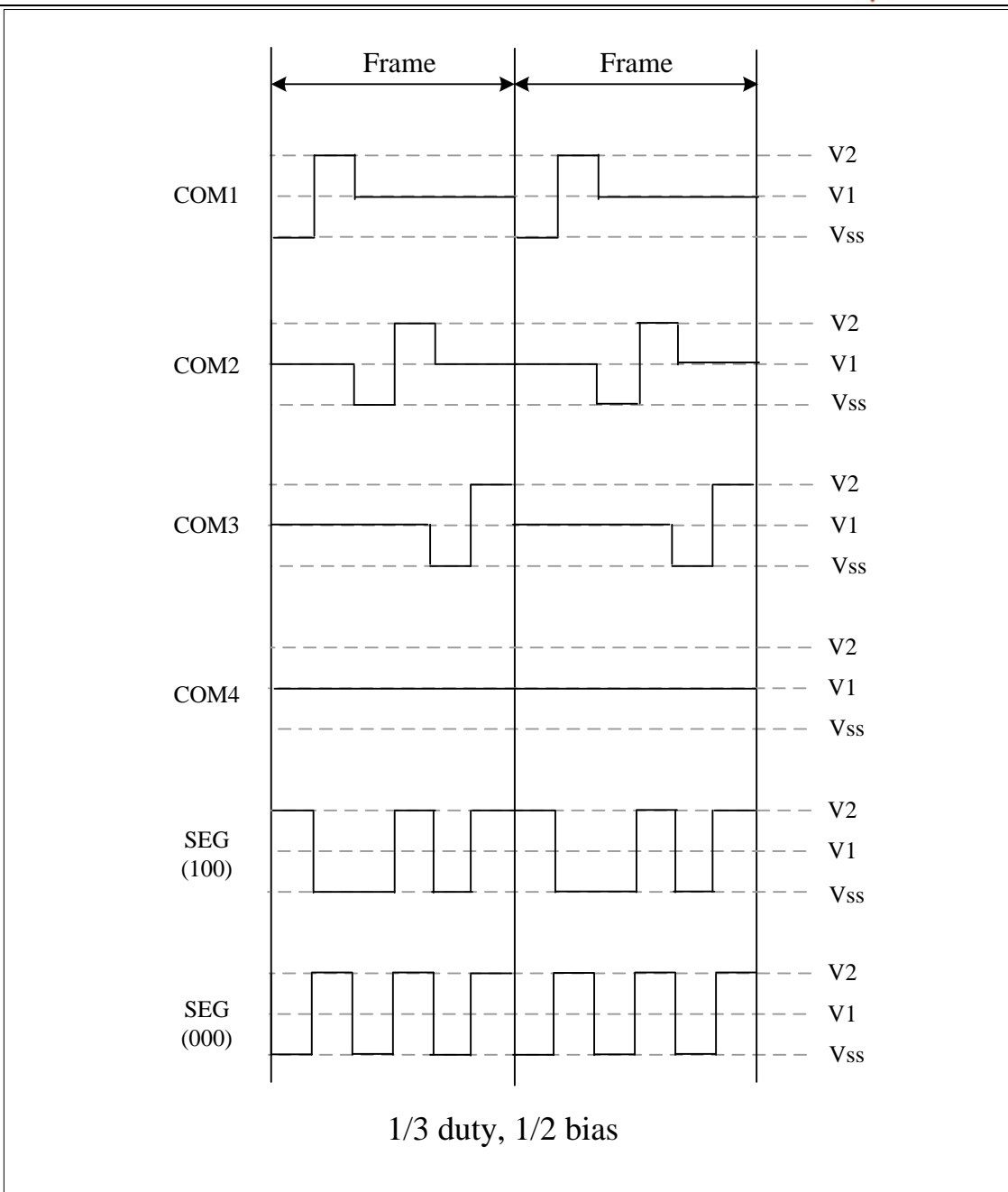


图 57 LCD 的 1/3duty 1/2bias 电源系统的时钟 (A 波形)

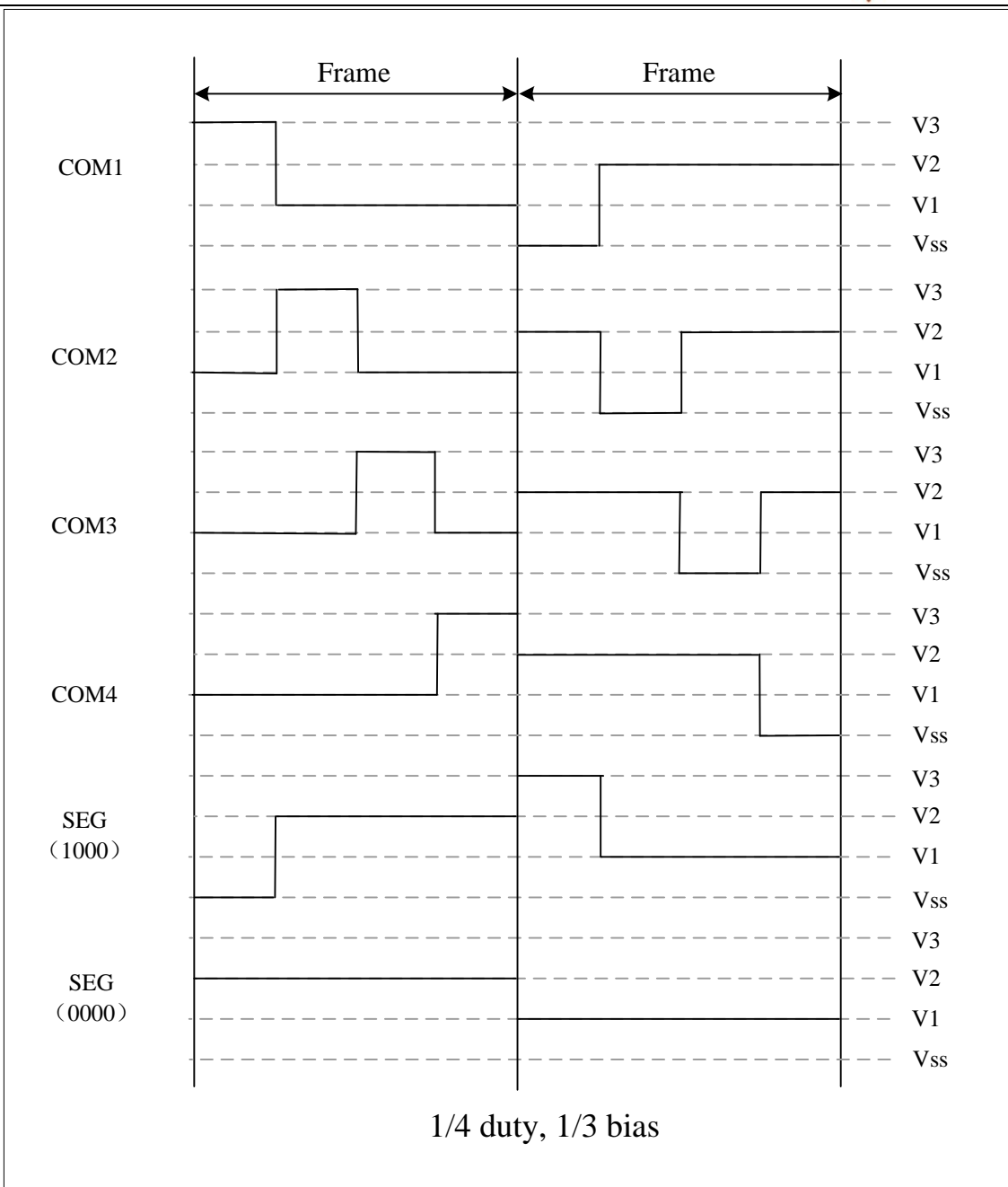


图 58 LCD 的 1/4duty 1/3bias 电源系统的时钟 (B 波形)

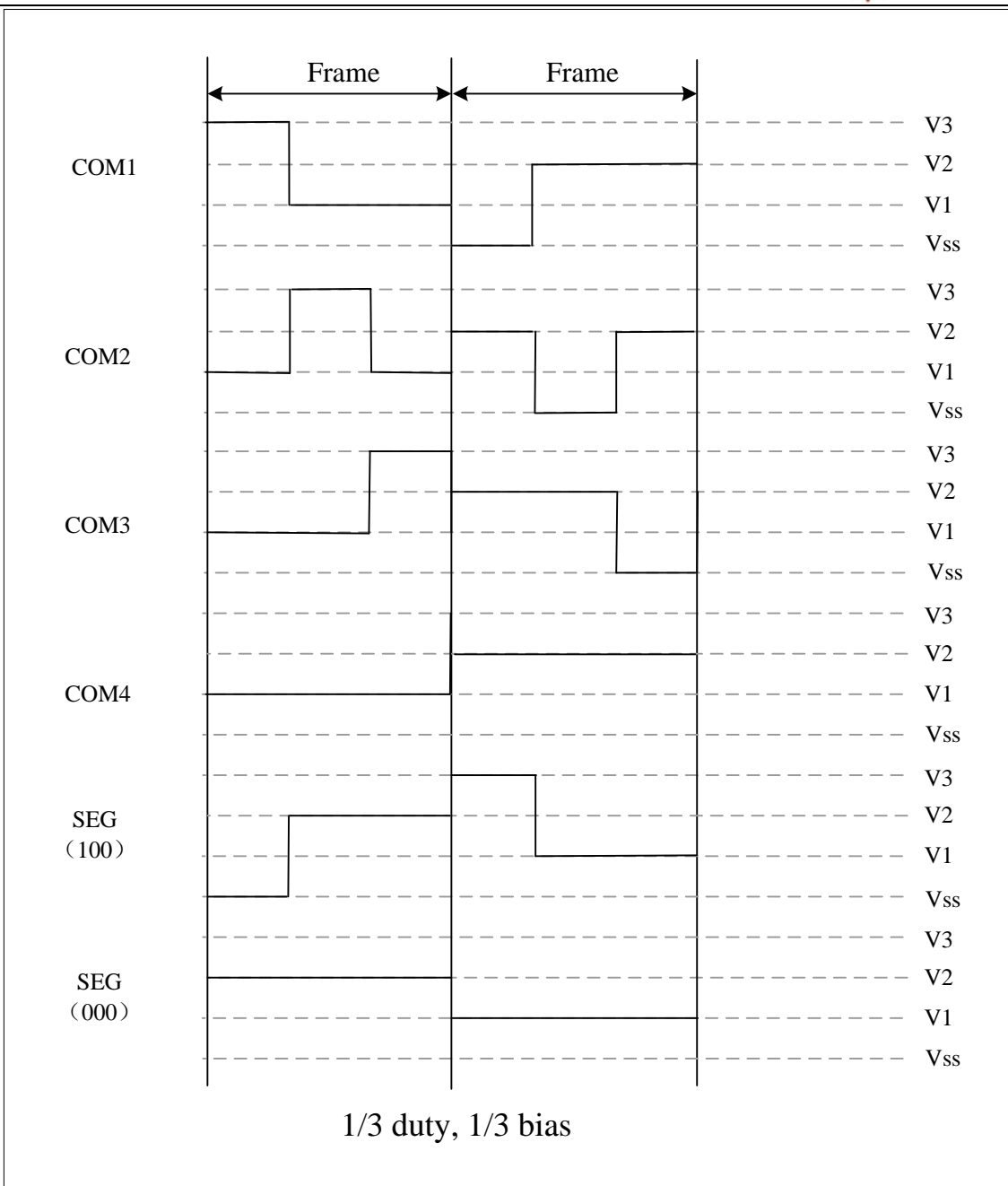


图 59 LCD 的 1/3duty 1/3bias 电源系统的时钟 (B 波形)

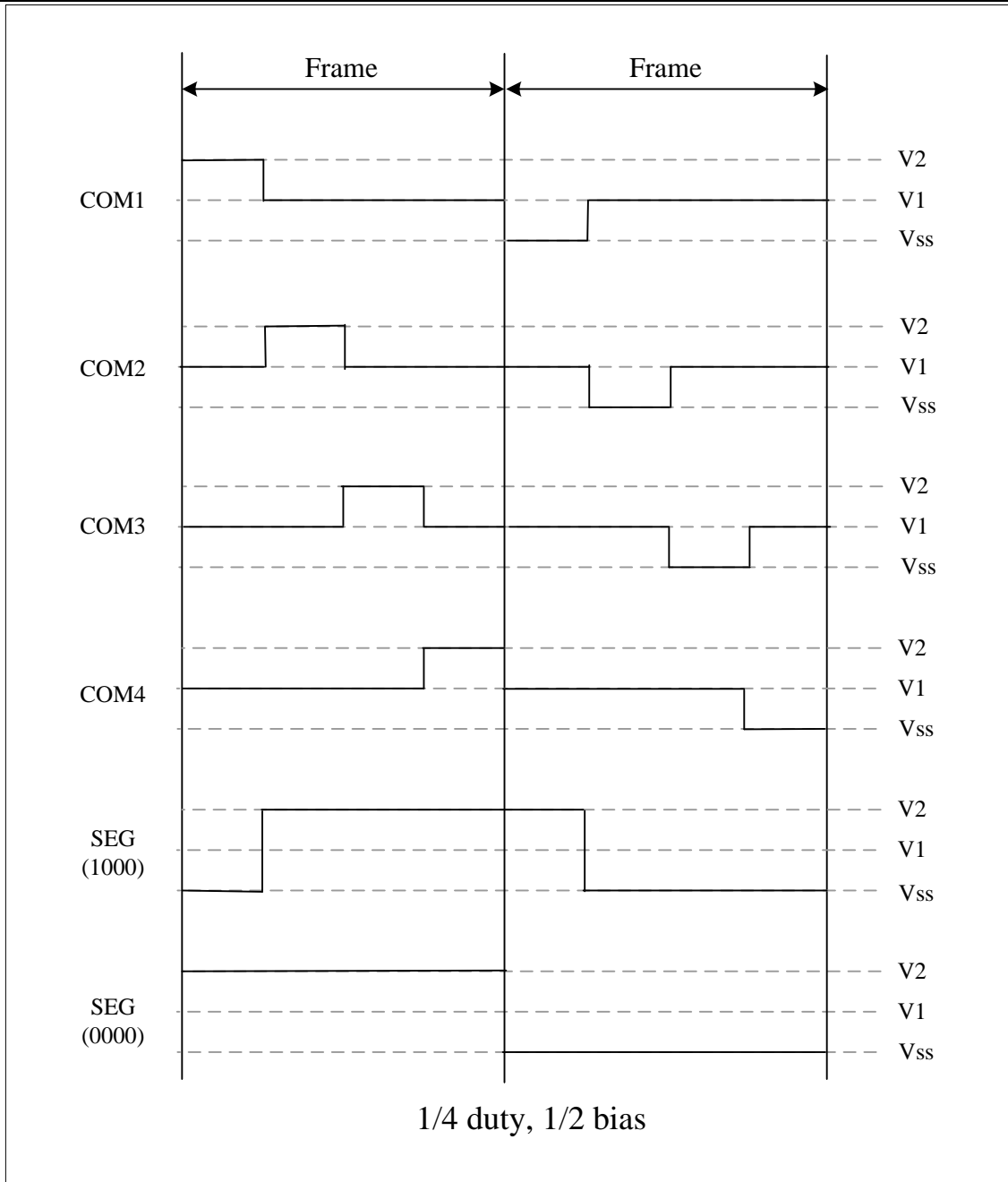


图 60 LCD 的 1/4duty 1/2bias 电源系统的时钟 (B 波形)

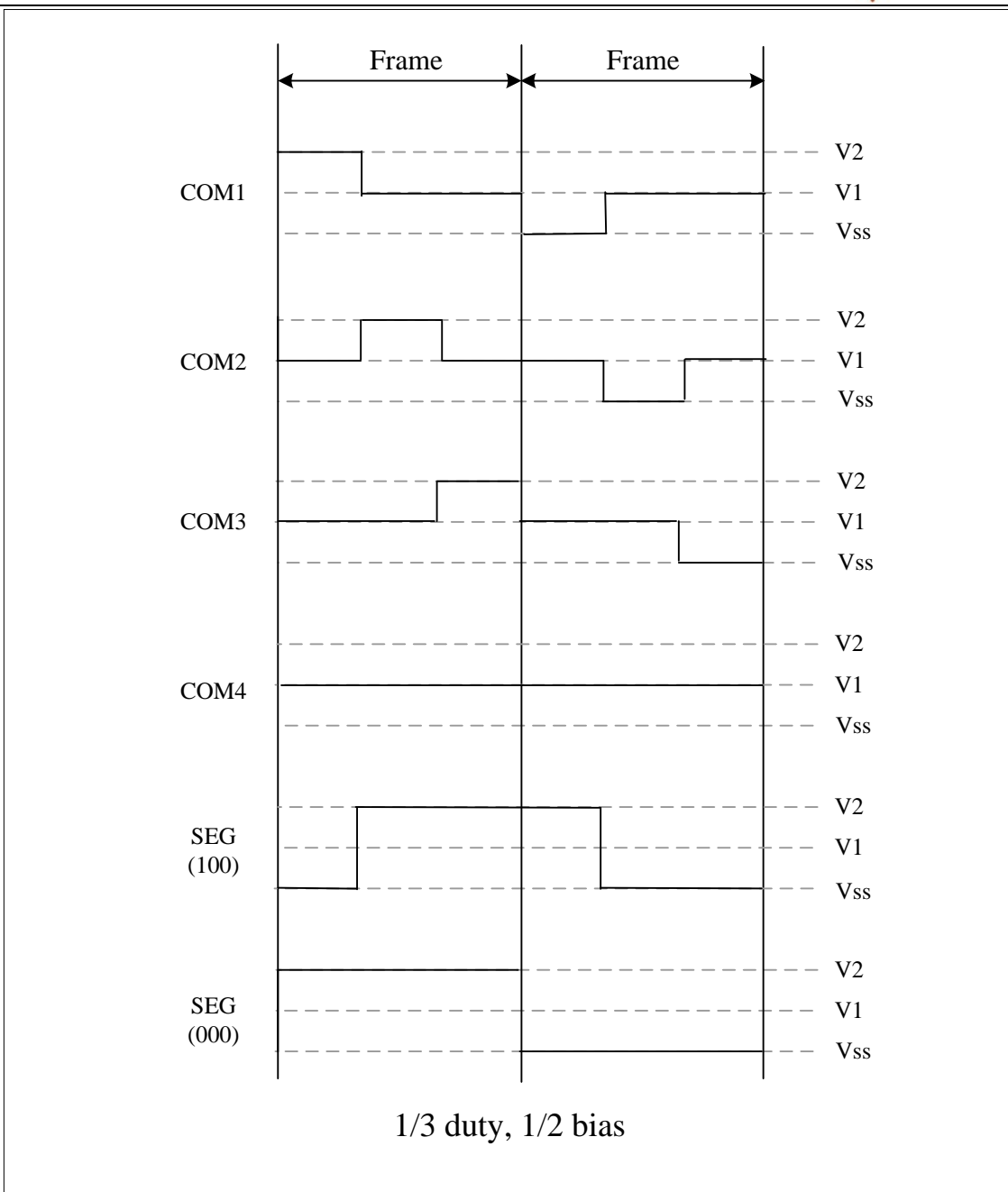


图 61 LCD 的 1/3duty 1/2bias 电源系统的时钟 (B 波形)

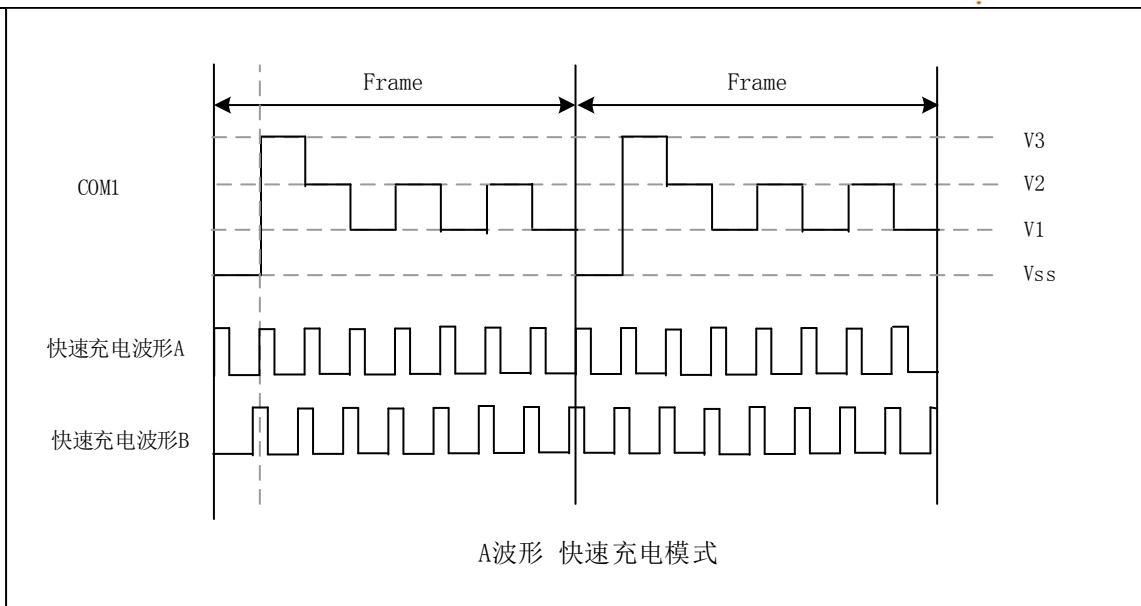


图 62 A 波形快速充电模式波形

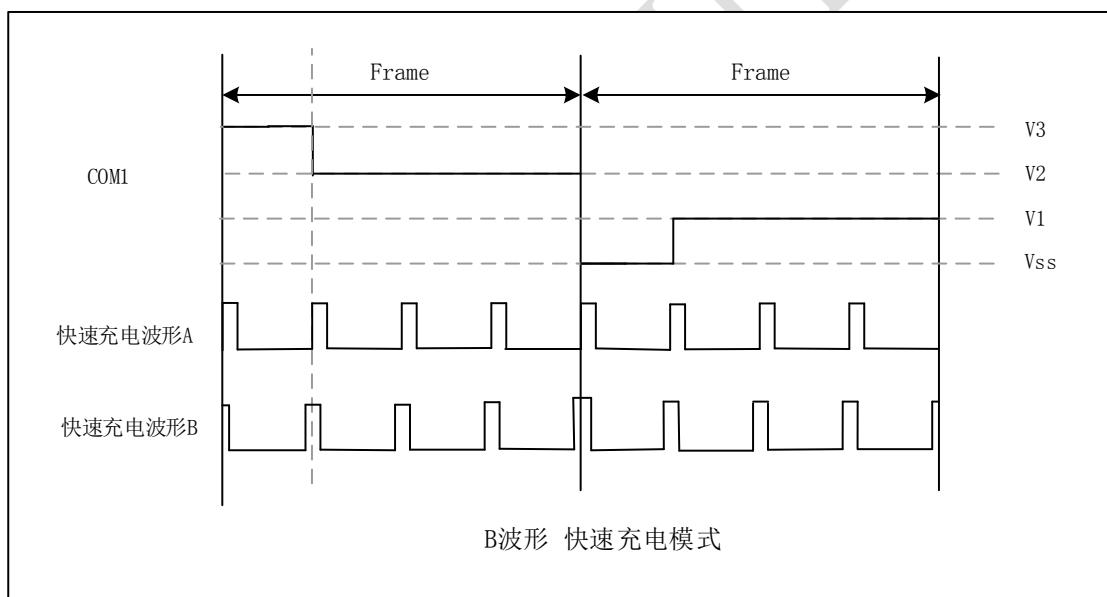


图 63 B 波形快速充电模式波形

3.12.5 LCD 寄存器说明

LCD 驱动器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
43H	LCD1	-				SEG1[3:0]				uuuu000 0
44H	LCD2	-				SEG2[3:0]				uuuu000 0

45H	LCD3	-	SEG3[5:0]				uu00000	
46H	LCD4	-	SEG4[5:0]				uu00000	
47H	LCD5	-	SEG5[5:0]				uu00000	
48H	LCD6	-	SEG6[5:0]				uu00000	
49H	LCD7	-	SEG7[5:0]				uu00000	
4AH	LCD8	-	SEG8[5:0]				uu00000	
4BH	LCD9	-	SEG9[5:0]				uu00000	
4CH	LCD10	-	SEG10[5:0]				uu00000	
4DH	LCD11	-	SEG11[5:0]				uu00000	
4EH	LCD12	-	SEG12[5:0]				uu00000	
4FH	LCD13	-	SEG13[5:0]				uu00000	
50H	LCD14	-	SEG14[5:0]				uu00000	
51H	LCD15	-	SEG15[5:0]				uu00000	
52H	LCD16	-	SEG16[5:0]				uu00000	
53H	LCD17	-	SEG17[5:0]				uu00000	
54H	LCD18	-	SEG18[5:0]				uu00000	
55H	LCD19	-	SEG19[5:0]				uu00000	
56H	LCD20	-	SEG20[5:0]				uu00000	
57H	LCDCN1	-	LCDFC[1:0]	WDT_LCD	CSE_LCD	-	VLCDX[1:0]	u0000u0
58H	LCDCN2	LCDSCK[3:0]			LCDFR	LCDREF[2:0]		0000000
59H	LCDENR	LCDCK[1:0]	LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]	ENPMPL	1100000
30H	SEGCON0	SEGCON0[7:0]						0000000
31H	SEGCON1	SEGCON1[7:0]						0000000

LCD1 寄存器

位地址	标识符	功能	
3:0	SEG1	LCD 驱动器控制信号	
		SEG1	-
		SEG1[3]	COM4
		SEG1[2]	COM3
		SEG1[1]	COM2

		SEG1[0]	COM1
--	--	---------	------

LCD3 寄存器

位地址	标识符	功能	
5:0	SEG3	LCD 驱动器控制信号	
		SEG3	-
		SEG3[5]	COM6
		SEG3[4]	COM5
		SEG3[3]	COM4
		SEG3[2]	COM3
		SEG3[1]	COM2
		SEG3[0]	COM1

LCDCN1 寄存器

位地址	标识符	功能	
6:5	LCDFC[1:0]	快速充电模式脉宽选择	
		LCDFC[1:0]	脉宽
		00	1/2com 周期
		01	1/4 com 周期
		10	1/8 com 周期
4	WDT_LCD	选择看门狗作为时钟源 参见 3.12.2LCD 帧频选择	
3	CSE_LCD	选择外部低速晶振作为时钟源 参见 3.12.2LCD 帧频选择	
2	-	保留	
1:0	VLCDX[1:0]	VLCD 输出电压选择	
		VLCDX[1:0]	VLCD 输出电压
		00	2.6V
		01	2.8V
		10	3.0V
		11	3.2V
备注：选用内部电荷泵做电源，需外接 1uF 电容			

LCDCN2 寄存器

位地址	标识符	功能	
7:4	LCDSCKS[3:0]	LCD 帧频选择 参见 3.12.2LCD 帧频选择	
3	LCDFR	LCD 充电模式波形选择(波形参考图 62 A 波形快速充电模式波形和图 63 B 波形快速充电模式波形)	
		LCDFR	-
		0	快速充电模式 A 波形
		1	快速充电模式 B 波形
2:0	LCDREF[2:0]	LCD 分压电阻选择	
		LCDREF[2:0]	-
		000	100Kohm
		001	50Kohm
		010	20Kohm
		011	10Kohm
		100	快速充电模式，20Kohm 100Kohm 电阻切换

		101	快速充电模式, 20Kohm 200Kohm 电阻切换
		110	快速充电模式, 10Kohm 100Kohm 电阻切换
		111	快速充电模式, 10Kohm 200Kohm 电阻切换

LCDENR 寄存器

位地址	标识符	功能	
7:6	LCDCKS[1:0]	LCD 帧频选择	
		LCDCKS[1:0]	LCD 的帧频
		11	LCD 输入时钟频率的 1/16
		10	LCD 输入时钟频率的 1/8
		01	LCD 输入时钟频率的 1/4
		00	LCD 输入时钟频率的 1/2
5	LCDEN	LCD 驱动器使能标志	
		LCDEN	-
		0	LCD 驱动器不使能。 LCD 的时钟被停止
		1	LCD 驱动器使能。LCD 的时钟被启动
4	LCDWS	LCD 充电模式波形选择	
		LCDWS	LCD 波形
		0	A 波形
		1	B 波形
3	LEVEL	LCD 驱动器的偏置电压选择器	
		LEVEL	LCD 驱动器的偏置电压
		0	1/3bias
		1	1/2bias
2:1	LCD_DUTY[1:0]	LCD 驱动器控制模式 (SEG duty 周期)	
		LCD_DUTY[1:0]	LCD 驱动器控制模式
		00	1/6duty 周期模式
		01	1/2duty 周期模式
		10	1/3duty 周期模式
		11	1/4duty 周期模式
0	ENPMPL	LCD charge pump 使能标志位	
		ENPMPL	-
		0	LCD charge pump 关闭
		1	LCD charge pump 打开

3.12.6 LCD 的操作步骤

- 1) 将段接口连接到 LCD 面板。
- 2) 设置 SEGCON0/ SEGCON1 为 0xFF
- 3) 设置 ENPMPL 使能 LCD 电荷泵;
- 4) 设置寄存器标志 LEVEL 选择 LCD 驱动器电源系统。(0 = 1/3bias, 1 = 1/2bias)
- 5) 置寄存器标志 LCDCKS[1:0]选择 LCD 的时钟频率 (参见 LCD 帧频选择)
- 6) 设置寄存器标志 LCD_DUTY[1:0], 选择控制模式。(SEG duty 周期)
- 7) 置位 LCDEN 以使能 LCD 驱动器

表 67 LCD 的 duty 控制模式选择列表

LCD_DUTY[1:0]	控制模式
00	1/6
01	1/2
10	1/3
11	1/4

3.13 RTC

实时时钟(RTC)单元提供给用户实时时间以及日历信息。RTC的时钟源由外部 32768Hz 晶振提供。RTC 数据信息由 BCD 码格式进行表示。该 RTC 提供时间计数器(秒,分,时)和日历计数器(日,月,年),供用户用来查看时间。该 RTC 具有如下功能:

- 12-小时或 24-小时模式可选择
- 闰年自动补偿
- 星期计数器
- 所有时间和日期信息由 BCD 码表示
- 中断功能, 0.5s 中断一次

3.13.1 功能描述

用户必须注意 RTC 模块不检查载入的数据是否越界, 不会对数据的合理性进行检查。

寄存器 RTCAER 作为保护 RTC 寄存器写密码。RTCAER 位 7~0 必须被设置为 0x96 来解除访问限制。可以通过寄存器位 24HR/12HR 选择是 24 小时格式还是 12 小时格式。

RTC 单元通过星期寄存器(RTCDWR)来提供星期信息。定义的值为由 0 至 6, 用于表示周日至周六。

RTC 允许软件对时钟输入进行数字补偿, 时钟输入的频率必须在 32776Hz 到 32761Hz 范围内。下列为对时钟输入进行补偿的示例。

例 1:

RTC 频率测量值:32773.65Hz(>32768Hz)
 整数部分:32773=>0x8005
 Integer=0x05-0x01+0x08=0x0c
 分数部分:0.65x60=39=>0x27
 Fraction=0x27

例 2

RTC 频率测量值:32765.27Hz(\cong 32768Hz)
 整数部分:32765=>0x7ffd
 Integer=0x0d-0x01-0x08=0x04
 分数部分:0.27x60=16.2=>0x10
 Fraction=0x10

3.13.2 寄存器说明

表 68 RTC 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
5AH	INT F3	I2C_ TIF	I2C_RIF	UR1ER R_IE	UR1W K_IF	UR0WK _IF	TM3I F	RTCI F	SPII F	u0000000
5BH	INT E3	I2C_ TIE	I2C_RIE	UR0ER R_IE	UR1W K_IE	UR0WK _IE	TM3I E	RTCI E	SPII E	u0000000
60H	RTCC ON	LIR	-	4HR/12 HR	RTCE N	INTEGER[3:0]			0u000111	
61H	RTC AER	AER[7:0]							00000000	
62H	RTC YEA R	10YEAR(0~9)			1YEAR(0~9)			00010101		
63H	RTC MON	-		10MO N(0~1)	1MON(0~9)			uuu10001		
64H	RTC DAY	-		10DAY(0~3)		1DAT(0~9)			uu000001	
65H	RTC	-	AM_P	10HOUR10(0~2)		1HOUR(0~9)			u0000001	

	HOUR		M				
66H	RTC MIN	-	10MIN(0~5)		1MIN(0~9)		u0000000
67H	RTCEC	-	10SEC(0~5)		1SEC(0~9)		u0000000
6AH	FRACTION	METCH[5:4]		FRACTION[5:0]			0000000
7EH	RTCDWR	-	-	-	-	-	DWR[2:0] (0~6) 0000000

INTF3 寄存器

位地址	标识符	功能
1	RTCIF	RTC 中断标志位 1 : 发生 RTC 中断, 必须软件清 0 0 : 没发生 RTC 中断

INTE3 寄存器

位地址	标识符	功能
1	RTCIE	RTC 中断使能 1 : 使能 RTC 中断 0 : 关闭 RTC 中断

RTCCON 寄存器

位地址	标识符	功能																																				
7	LIR	闰年指示寄存器(只读) 1=该年为闰年 0=该年非闰年																																				
6	-	RFU																																				
5	4HR/12HR	24-小时/12-小时模式选择 用于表示 TLR 和 TAR 为 24-小时或 12-小时模式 1=选择 24-小时制 0=选择 12-小时制, 带 AM 和 PM 指示 注: 小时制切换时需要软件更新小时寄存器 RTCHOUR																																				
4	RTCEN	RTC 使能控制 1=RTC 功能使能 0=RTC 功能禁止																																				
3:0	INTEGER[3:0]	RTC 时钟补偿整数部分, 每秒钟补偿一次, 默认值 0111。0x96 写保护 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>检测到的值的整数</th> <th>INTEGER[3:0]</th> <th>检测到的值的整数</th> <th>INTEGER[3:0]</th> </tr> </thead> <tbody> <tr><td>32776</td><td>1111</td><td>32768</td><td>0111</td></tr> <tr><td>32775</td><td>1110</td><td>32767</td><td>0110</td></tr> <tr><td>32774</td><td>1101</td><td>32766</td><td>0101</td></tr> <tr><td>32773</td><td>1100</td><td>32765</td><td>0100</td></tr> <tr><td>32772</td><td>1011</td><td>32764</td><td>0011</td></tr> <tr><td>32771</td><td>1010</td><td>32763</td><td>0010</td></tr> <tr><td>32770</td><td>1001</td><td>32762</td><td>0001</td></tr> <tr><td>32769</td><td>1000</td><td>32761</td><td>0000</td></tr> </tbody> </table>	检测到的值的整数	INTEGER[3:0]	检测到的值的整数	INTEGER[3:0]	32776	1111	32768	0111	32775	1110	32767	0110	32774	1101	32766	0101	32773	1100	32765	0100	32772	1011	32764	0011	32771	1010	32763	0010	32770	1001	32762	0001	32769	1000	32761	0000
检测到的值的整数	INTEGER[3:0]	检测到的值的整数	INTEGER[3:0]																																			
32776	1111	32768	0111																																			
32775	1110	32767	0110																																			
32774	1101	32766	0101																																			
32773	1100	32765	0100																																			
32772	1011	32764	0011																																			
32771	1010	32763	0010																																			
32770	1001	32762	0001																																			
32769	1000	32761	0000																																			

RTCAER 寄存器

位地址	标识符	功能
-----	-----	----

7:0	AER[7:0]	RTC 寄存器写使能密码 0x96=使能 RTC 写操作 Others=禁用 RTC 写操作 注：只对 RTCYEAR、RTCMONTH、RTCDATE、RTCHOUR、RTCMIN、RTCSEC 和 RTCDWR 有效
-----	----------	---

RTCYEAR 寄存器:

位地址	标识符	功能
7:4	10YEAR	10YearCalendarDigit(0~9)
3:0	1YEAR	1YearCalendarDigit(0~9)

RTCMON 寄存器:

位地址	标识符	功能
4	10MON	10MonthCalendarDigit(0~1)
3:0	1MON	1MonthCalendarDigit(0~9)

RTCDAY 寄存器:

位地址	标识符	功能
5:4	10DAY	10DayCalendarDigit(0~3)
3:0	1DAY	1DayCalendarDigit(0~9)

RTCHOUR 寄存器

位地址	标识符	功能
6	AM_PM	AM 和 PM 指示 =0 表示 AM, =1 时表示 PM
5:4	10HR	10HourTimeDigit(0~2)
3:0	1HR	1HourTimeDigit(0~9)

如果是 24 小时制, RTCHOUR[5:4]变化是 0~2; RTCHOUR[3:0]变化是 0~9;

如果是 12 小时制, RTCHOUR[5:4] 变化是 0~1; RTCHOUR[3:0]变化是 0~9; ;

RTCMIN 寄存器

位地址	标识符	功能
6:4	10MIN	10MinTimeDigit(0~5)
3:	1MIN	1MinTimeDigit(0~9)

RTCSEC 寄存器

位地址	标识符	功能
6:4	10SEC	10SecTimeDigit(0~5)
3:0	1SEC	1SecTimeDigit(0~9)

RTCDWR 寄存器

位地址	标识符	功能												
2:0	DWR[2:0]	星期寄存器 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>DWR</th> <th>星期</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>星期日</td> </tr> <tr> <td>1</td> <td>星期一</td> </tr> <tr> <td>2</td> <td>星期二</td> </tr> <tr> <td>3</td> <td>星期三</td> </tr> <tr> <td>4</td> <td>星期四</td> </tr> </tbody> </table>	DWR	星期	0	星期日	1	星期一	2	星期二	3	星期三	4	星期四
DWR	星期													
0	星期日													
1	星期一													
2	星期二													
3	星期三													
4	星期四													

		5	星期五	
		6	星期六	

注:TLR 为 BCD 码格式的计数器, RTC 不会对载入值进行检测.
括号内列出的为可接受的值.

FRACTION 寄存器

Bit	描述	
7:6	-	保留
5:0	FRACTION[5:0]	RTC 时钟频率补偿小数部分, 每 60 秒补偿一次 公式=(检测到的小数部分)x60 注: FRACTION 的值必须按照 16 进制格式表示。

3.13.3 RTC 的操作步骤

1. 将 PT1.0/1.1 设为模拟 IO
2. RTCAER 写入 96H
3. 设置 RTC 相关寄存器
4. 设置时间格式
5. 设置 RTCEN 为高, 使能 RTC

3.14 数据查表

电路要求：在线烧录时，使用芯片自带的升压电路。

地址要求：芯片通过 PARH[5:0]和 PARL[7:0]寄存器来选择地址，寻址空间是 0000H~207FH，包含程序存储器(地址范围 0000H~1FFFH)和非易性数据存储器(地址范围 2000H~207FH)；

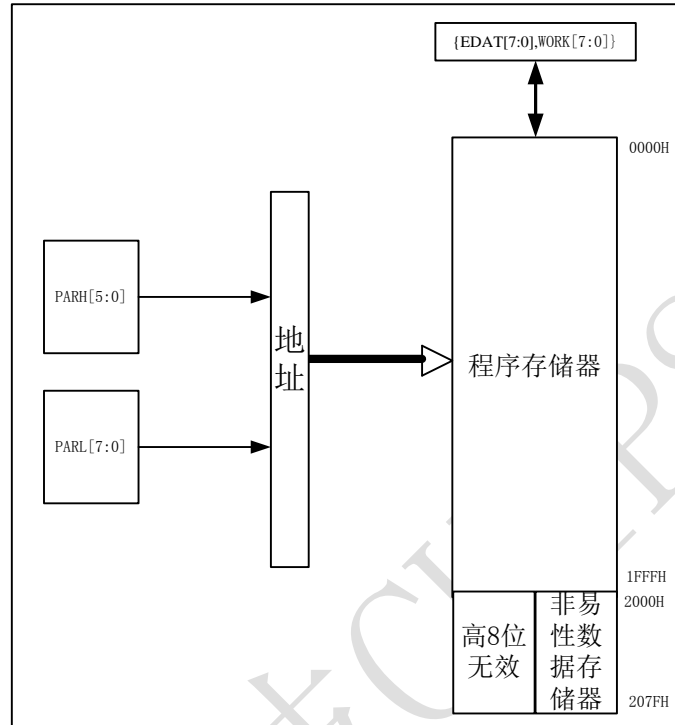


图 64 在线烧录地址寻址示意图

表 69 在线烧录寄存器列表

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	PROG_BUSY	READ_CHECK	PARH[5:0]						00000000
0BH	EADRL	PARL[7:0]								00000000
0CH	EDAT	EDAT[7:0]								11111111
0DH	EOPEN	EOPEN[7:0]								00000000

EOPEN：在线烧录保护寄存器，在对 MTP 烧录时必须对该地址写入 AAH、55H、A5H；对 EEPROM 烧录时必须对该地址写入 96H、69H、5AH；对其他地址寄存器进行写操作时，EOPEN 寄存器会被清零。（除了 work 寄存器）

EADRH /EADRL：提供 MTP 在线烧录或者在线读 MTP 的地址。

EDAT：提供 MTP 在线烧录时的烧录数据或者在线读 MTP 时的读出的数据高 8 位。

Work：提供 MTP 在线烧录时的烧录数据或者在线读 MTP 时的读出的数据低 8 位。地址范围 2000H~207FH 时，只有 Work 寄存器有效。

READ_CHECK：当 MTP 在线烧录完毕后，硬件会自动对该地址的数据读出后比较，如果发生错误，此 Bit 为 1，如果正确此 Bit 为 0。

PROG_BUSY：进入 MTP 在线烧录 可以打开 WDT 计时，如果出现烧录异常一直没有等到 Busy 信号拉低，WDT 溢出会终止烧录，在 WDT 复位 PC 后可以读取此 Bit，如果此 Bit 还是为 1，表示 WDT 复位时 BUSY 还处于高。

操作方式：

在线烧录时：

1. 使用 CLRWDT 指令复位 WDT；
2. 将烧录地址写入 EADRH, EADRL 寄存器；
3. 高八位数据写入 EDAT 寄存器；
4. 对 EOPEN 写入相应的数据；
5. 将烧录数据低八位数据写入 work 寄存器；注意此步骤不能提前。
6. 用在线烧录指令（TBLP）烧录；

注：在线烧录时，要打开 WDT，但在对 EOPEN 操作之前先使用 CLRWDT 指令复位 WDT。

在线读 MTP 数据时：

1. 将读 MTP 地址写入 EADRH, EADRL 寄存器；
2. 用在线读 MTP 指令（MOVP）读出 MTP 数据，执行该指令后，读出存放在 EDAT、work 寄存器；

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 WORK 寄存器中。执行一次读操作需要 3 个指令周期。

注：在使用 TBLP 指令时，不能开中断。

3.15 烧录模块

烧写器的接口，能读/写 SFR 寄存器：

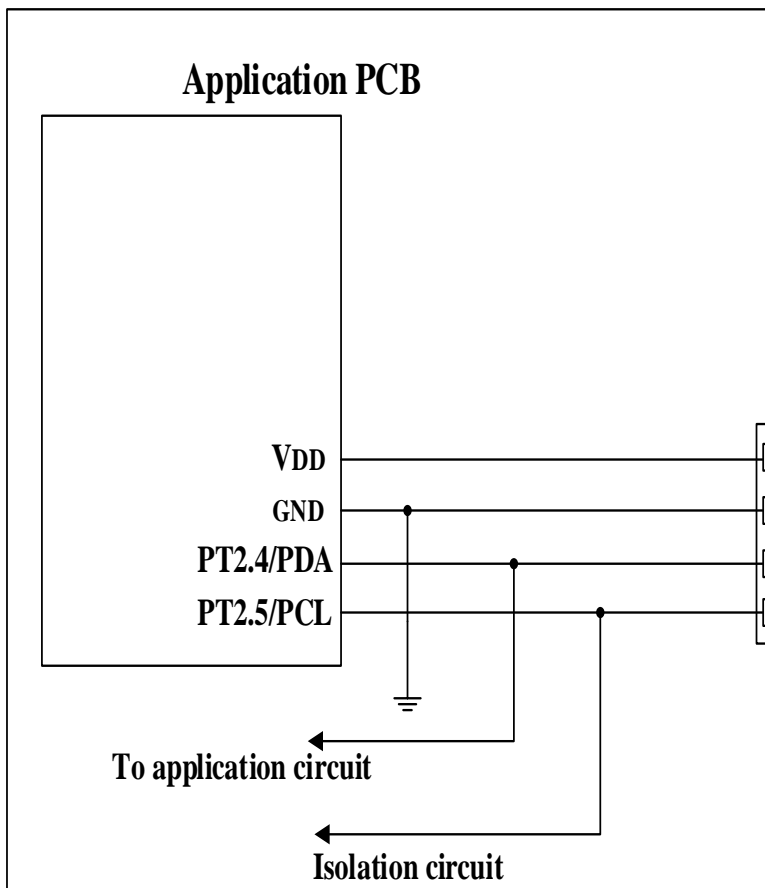


图 65 烧写器接口图

表 70 接口说明

端口名称	型式	说明	备注
VDD	输入	电源正端	-
GND	输入	电源负端	-
PDA	输入/输出	PT2.4 端口，数据信号	-
PCL	输入	PT2.5 端口，时钟信号	-

在烧录模式进行写操作时，可以通过 PDA 输出口的状态，判断写操作是否完成。PDA=1 时，写操作还在进行中；PDA=0 时，写操作完成，可以进行下一次的写操作。

4 MCU 指令集

表 71 MCU 指令集

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

参数说明:

f: 数据存储器地址(000H ~ 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

MCU 指令集描述

1

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

2

ADDPCW	将 W 的内容加到 PC 中
指令格式	ADDPCW
操作	(PC)<—(PC)+1+(W) 当(W)<=7FH (PC)<—(PC)+1+(W)-100H 其余
标志位	没有
描述	将地址 PC+1+W 加载到 PC 中
周期	2
例子 1 ADDPCW	在指令执行之前: W=7FH , PC=0212H 指令执行之后: PC=0292H
例子 2 ADDPCW	在指令执行之前: W=80H , PC=0212H 指令执行之后: PC=0193H
例子 3 ADDPCW	在指令执行之前: W=FEH , PC=0212H 指令执行之后: PC=0211H

3

ADDWF	加工作寄存器到 f
指令格式	ADDWF f,d 0<=f<=1FFH d=0,1
操作	[目标地址]<—(f)+(W)
标志位	C, CD, Z

描述	将 f 的内容和工作寄存器的内容加到一起。 如果 d 是 0，结果保存到工作寄存器中。 如果 d 是 1，结果保存到 f 中。
周期	1
例子 1 ADDWF f 0	指令执行之前： f=C2H W=17H 在指令执行之后 f=C2H W=D9H
例子 2 ADDWF f 1	指令执行之前 f=C2H W=17H 指令执行之后 f=D9H W=17H

4

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

5

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

6

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

ANDWF f, 1	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	在程序执行以前 PC=address(NODE)

OP1: OP2:	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<=1FFFH
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 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	在指令执行之前

COMF f, 1	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=25； ADDLW 39 DAW	在 DAW 指令执行之前 W=25+39 =5EH 在指令执行之后 W=64H

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

25

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

26

MOVP	读查表区数据
-------------	--------

指令格式	MOVP
操作	把 EPROM 数据读到 WORK 中
标志位	无
描述	把地址为 EADRH/EADRL 的查表区数据读到 WORK 中
周期	2
例子 MOVP	在指令执行之前 EADRH=04H, EADRL=00H 地址为 0400H 的查表区数据位 34H 在指令执行之后 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 4 级堆栈, 不同于 PC 堆栈; 其中状态寄存器不包括 LVD36, LVD24, PD 和 TO。
周期	2

30

POP	把 work 和 status 寄存器出栈处理
指令格式	POP
操作	(Top Stack)=work/status Pop Stack
标志位	无
描述	把当前栈顶的值做出栈处理, 分别更新 work 和 status 寄存器, 支持 4 级堆栈, 不同于 PC 堆栈; 其中状态寄存器不包括 LVD36, LVD24, PD 和 TO。
周期	2

31

RETFIE	从中断返回
指令格式	RETFIE
操作	(Top Stack)=PC

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

32

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

33

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

34

RLF	带进位左移
指令格式	RLF f, d 0<=f<=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 时钟停止
标志位	PD
描述	CPU 时钟停止。CPU 通过外部中断源重启
周期	1

37

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

38

SUBWF	f 的值减工作寄存器的值
指令格式	SUBWF f, d 0<=f<=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	将 MTP 的{EADRH,EADRL}的地址写入{WORK}中的数据
指令格式	TBLP k
操作	MTP({EADRH,EADRL})<←({WORK})
标志位	无
描述	将 MTP 的 {EADRH,EADRL} 的地址写入 {WORK} 中的数据
周期	与工作时钟有关
例子 TBLP 100	在指令执行之前 MTP memory: 17H= FFH EADRH =01H EADRL = 17H WORK = 05H 在指令执行之后 MTP memory: 117H = 05H

42

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

43

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

5 电气特性

5.1 极限值

表 72 芯片极限值

参数	范围	单位
电源 DVDD, AVDD	-0.3~6	V
引脚输入电压	-0.3~DVDD+0.3 -0.3~AVDD+0.3	V
工作温度	-40~+85	°C
存贮温度	-55~+150	°C
焊接温度, 时间	220°C, 10 秒	-

5.2 直流特性

(DVDD, AVDD= 3V, T_A = 25°C, 如无其他说明则都是此条件)

符号	参数	测试条件	最小值	典型值	最大值	单位
VDD	工作电源	DVDD/AVDD/VLCD	2.4	3	3.6	V
DVDD/AVDD D ripple	电源 DVDD, AVDD 纹波	-	-100	-	100	mV
IDD1	数字电源电流 (含 HIRC8MHz)	指令周期 = 2MHz	-	0.66	-	mA
IDD2	数字电源电流 (含 HIRC8MHz)	指令周期 = 1MHz	-	0.38	-	mA
IDD3	数字电源电流 (含 HIRC 4MHz)	指令周期 = 500KHz	-	0.18	-	mA
IDD4	模拟电源 (ADC 一般模式)	-	-	0.5	-	mA
IDD5	模拟电源 (ADC 性能模式)	-	-	0.8	-	mA
ISleep1 ⁽¹⁾	睡眠模式下电源电 流	睡眠指令	-	0.55	-	uA
ISleep2 ⁽¹⁾	睡眠模式下开 WDT 电源电流	睡眠指令	-	1.4	-	uA
ISleep3	睡眠模式下开 LCD 开外置晶振开 RTC 电源电流	睡眠指令;外部输入 VLCD (LCDREF[2:0]=101, VLCD=3V)	-	20	-	uA
VIH	数字输入高电平	PT1(开 smit)、PT2(开 smit)、 PT3、PT4、PT5(开 smit)	0.7*VD D	-	-	V
VIL	数字输入低电平	PT1(开 smit)、PT2(开 smit)、 PT3、PT4、PT5(开 smit)	-	-	0.3*VD D	V
VIH	数字输入高电平	PT1(关闭 smit)、PT2(关闭 smit)	0.5*VD D	-	-	V
VIL	数字输入低电平	PT1(关闭 smit)、PT2(关闭 smit)	-	-	0.5*VD D	V
IPU	上拉电流	PT1、PT2、PT3、PT4、PT5	-	140	-	uA
IOH	高电平输出电流	VOH=0.9*VDD	-	4	-	mA

		(除 PT2.1、PT2.6、PT2.7)				
IOL	低电平输出电流	$VOL=0.1*VDD$ (除 PT2.1、PT2.6、PT2.7)	-	-3	-	mA
IOH	高电平输出电流	$VOH=0.9*VDD$ 仅 PT2.1	-	6	-	mA
IOL	低电平输出电流	$VOL=0.1*VDD$ 仅 PT2.1	-	-7	-	mA
IOH	高电平输出电流	$VOH=0.9*VDD$ 仅 PT2.6、PT2.7	-	10	-	mA
IOL	低电平输出电流	$VOL=0.1*VDD$ 仅 PT2.6、PT2.7	-	-12	-	mA
IREF	VS 稳压器输出电流	$VS=2.4V$	-	60	-	mA
VLREF	用于低电压检测的内部参考电压	-	-	1.2	-	V
TCLREF	用于低电压检测的内部参考电压温度系数	$TB_{AB} = -40\sim 80^{\circ}C$	-	50	-	ppm/ $^{\circ}C$
LDO PSRR	LDO 的直流电源电压抑制比	-	-	-60	-	dB
VLBAT	低电池检测电压	$SILB[2:0]=000$	-	2.4	-	V
		$SILB[2:0]=001$	-	2.5	-	
		$SILB[2:0]=010$	-	2.6	-	
		$SILB[2:0]=011$	-	2.7	-	
		$SILB[2:0]=100$	-	2.8	-	
		$SILB[2:0]=101$	-	3.0	-	
		$SILB[2:0]=110$	-	1.2	-	
	$SILB[2:0]=111$	-	3.6	-		
	低电压检测电压误差	-	-	± 50	-	mV
FRC ⁽²⁾	内置 RC 振荡器	-	7.92	8	8.08	MHz
FWDT ⁽¹⁾	内置看门狗时钟	-	2.7	3	3.3	KHz

- 1、样片测试保证，不在量产进行测试
- 2、设计保证，不在量产进行测试

5.3 24 Bits ADC 的特性

表 73 Sigma Delta ADC 性能指标

($V_{ref} = 3.3V$, $T_A = 25^{\circ}C$, 如无其他说明则都是此条件)

参数	条件	最小值	典型值	最大值	单位	
模拟输入	共模输入电压	-	GND	$V_{REF}/2$	V	
	满幅输入电压 (A_{IN+})-(A _{IN-})	-	-	$\pm V_{REF}/PGA$	V	
	差分输入阻抗	-	-	8/PGA	MΩ	
系统性能	分辨率	-	-	24 (带符号位)	Bits	
	输入噪声 (rms)	增益=1 @10sps	-	1.3	-	uv
		增益=8 @10sps	-	240	-	nV
		增益=20 @10sps	-	200	-	nV
		增益=80 @10sps	-	80	-	nV
		增益=160 @10sps	-	58	-	nV
增益=252 @10sps	-	58	-	nV		

		增益=320 @10sps	-	50	-	nV
		增益=400 @10sps	-	58	-	nV
		增益=800 @10sps	-	50	-	nV
	积分线性度	增益=80/160/320	-	±0.003	-	% of 0.7*FS
	失调误差	增益=80/160/320	-	40	-	uV
	失调误差漂移	增益=80/160/320	-	-0.085	-	uV/°C
	增益误差 ⁽²⁾	增益=80/160/320	-	-	-8	%
	增益温漂 ⁽²⁾	无补偿, 增益=80/160/320 @tempc<7:4>=0000/1000	-	±10	-	ppm/°C
	增益温漂补偿 ⁽²⁾	增益=80/160/320 @tempc<7:4>=0010	-	-45	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=1010	-	45	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=0100	-	-90	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=1100	-	90	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=0110	-	-130	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=1110	-	130	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=0011	-	-4	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=1011	-	4	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=0101	-	-9	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=1101	-	9	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=0111	-	-13	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=1111	-	13	-	ppm/°C
		增益=80/160/320 @tempc<7:4>=0000/1000	-	0	-	ppm/°C
参考电压		VS	LDOS[1:0]= 11	-	2.4	-
	LDOS[1:0]= 10		-	2.45	-	
	LDOS[1:0]= 01		-	3.0	-	
	LDOS[1:0]= 00		-	3.3	-	
	参考电压误差	-	-	±100	-	mV
	参考电压温度系数 ⁽²⁾	-	-100	-	100	ppm/°C

- 1、样片测试保证，不在量产进行测试
- 2、设计保证，不在量产进行测试

5.4 10 Bits SARADC 的特性

表 74 SAR ADC 性能指标
(DVDD= 3.3V, T_A = 25°C, 如无其他说明则都是此条件)

参数		条件	最小值	典型值	最大值	单位
系统性能	输入电压范围	AD0~AD5	0	-	DVDD	V
	分辨率	-	-	10	-	Bits
	INL	-	-	±2	-	LSB
	DNL	-	-	±1	-	LSB
	输入失调电压	-	-	7	-	LSB
	DataRate	-	-	100K	-	SPS
	I _{dd}	-	-	60	-	uA
参考电压	VS	LDOS[1:0]= 11, AD2_REF=1	2.25	2.4	2.45	V
		LDOS[1:0]= 10, AD2_REF=1	2.35	2.45	2.55	
		LDOS[1:0]= 01, AD2_REF=1	2.9	3.0	3.1	
		LDOS[1:0]= 00, AD2_REF=1	3.2	3.3	3.4	
	DVDD	AD2_REF=0	2.4	-	3.6	

6 封装图外型尺寸

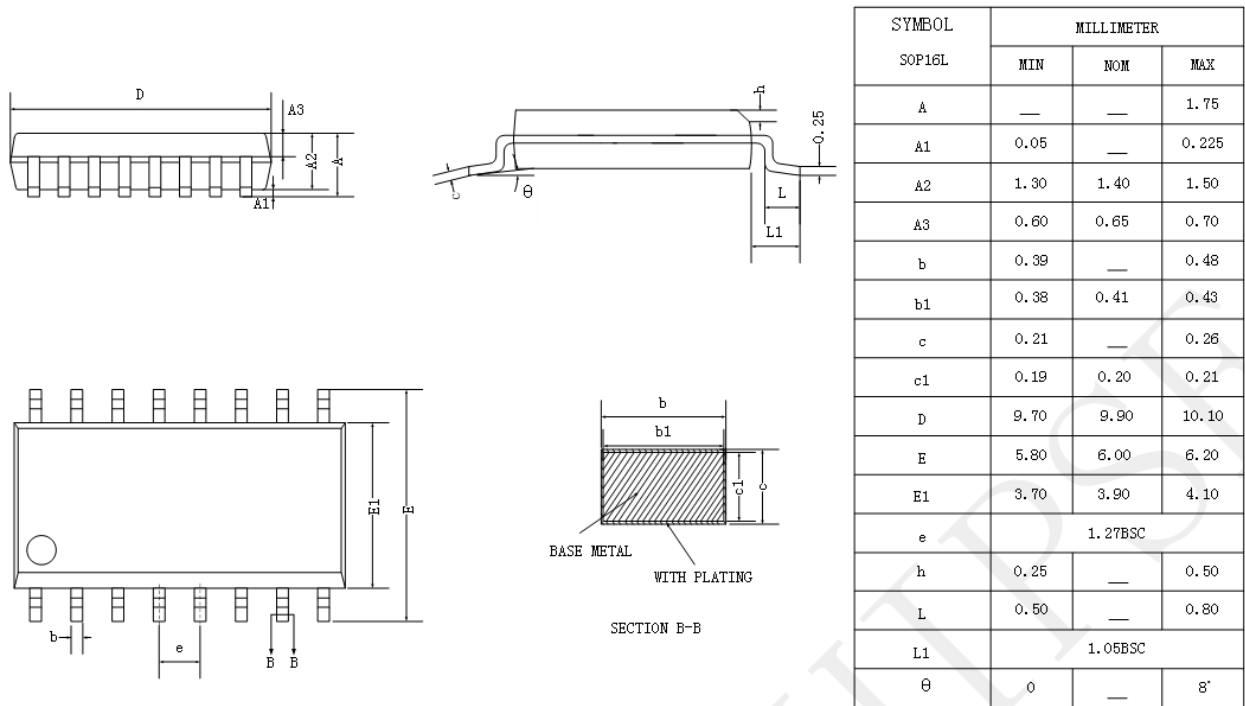


图 6-1 CSU18M88 SOP16 封装外型尺寸说明

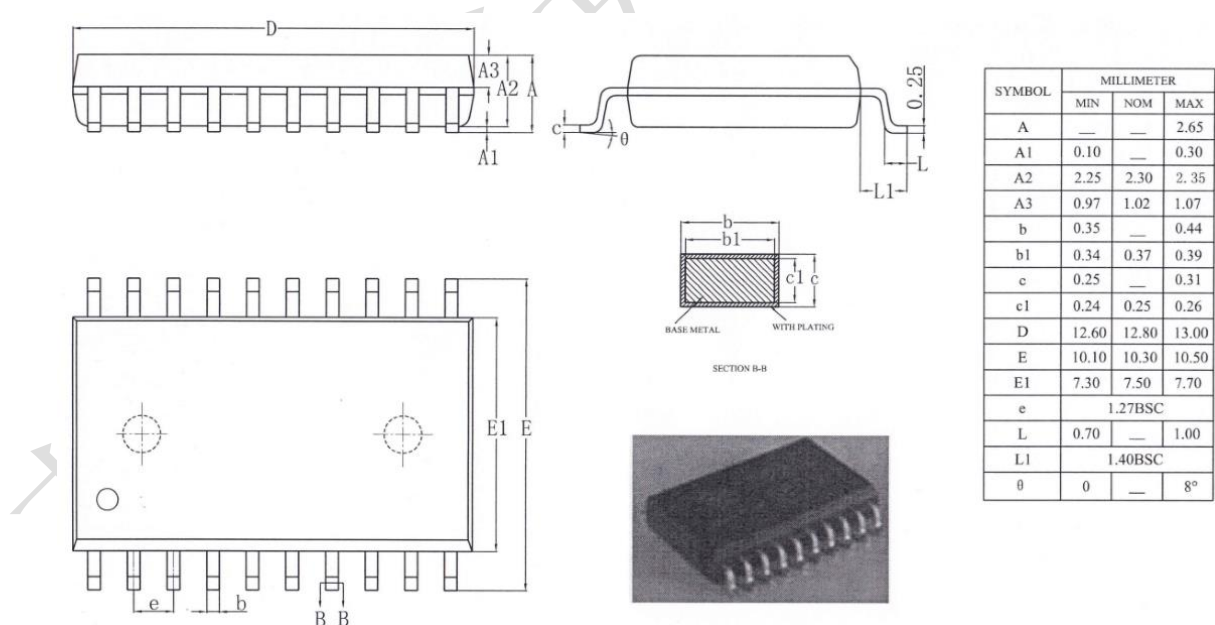
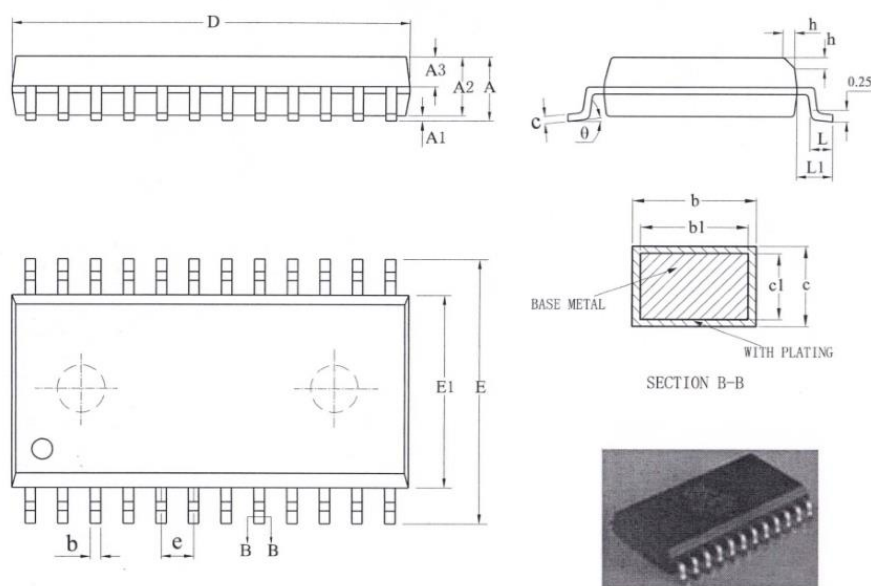
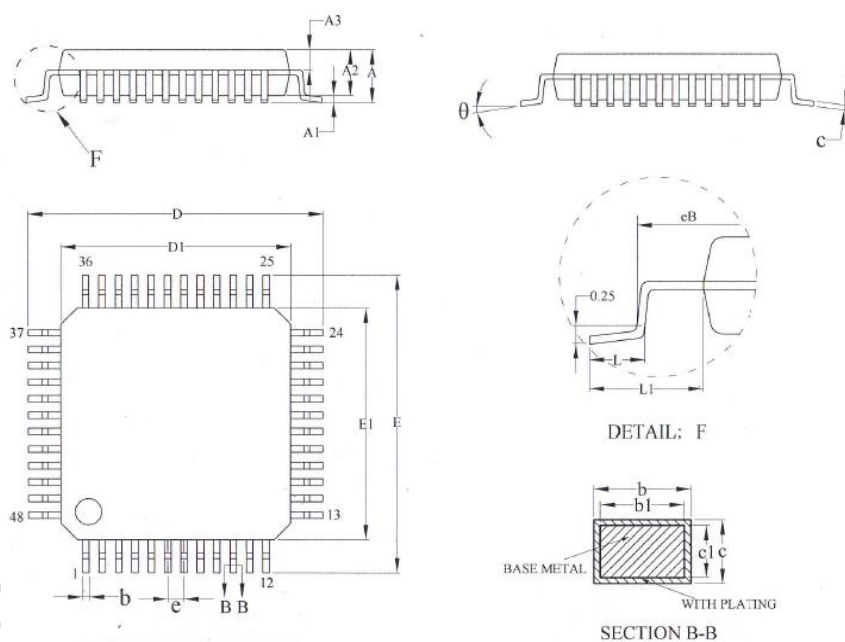


图 66 CSU18M88 SOP20 封装外型尺寸说明



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	2.36	2.54	2.64
A1	0.10	0.20	0.30
A2	2.26	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	—	0.48
b1	0.38	0.41	0.43
c	0.25	—	0.31
c1	0.24	0.25	0.26
D	15.20	15.40	15.60
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40BSC		
h	0.25	—	0.75
θ	0	—	8°

图 67 CSU18M88 SOP24 封装外型尺寸说明



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	—	0.26
b1	0.17	0.20	0.23
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.50BSC		
L	0.40	—	0.65
L1	1.00REF		
θ	0	—	7°

图 68 CSU18M88 LQFP48 封装外型尺寸说明