



CSU18MB86 系列用户手册

基于 MTP ROM 的 8 位 RISC MCU

REV. 1.3

通讯地址:深圳市南山区南海大道 1079 号花园城数码大厦 A 栋 9 楼

邮政编码:518067

公司电话:+(86 755)86169257

传 真:+(86 755)86169057

公司网站:www.chipsea.com

微信二维码:





CSU18MB86

8 位 MTP ROM 单片机产品介绍

高性能的 RISC CPU

- 8 位单片机 MCU
- 内置 8k×16Bits 的 MTP 程序存储器(烧录次数不低于 1000 次)和 128Bytes EEPROM(烧录次数不低于 10000 次)
- 488 字节数据存储器 (SRAM)：需要翻页
- 43 条指令、8 级存储堆栈
- 最快指令周期为：2MHz(默认值)、1MHz、500kHz、250kHz

振荡器

- 内置 14.7456MHz/16MHz 振荡器
- 内置 3kHz WDT 振荡器
- 可外接 32768Hz 晶振作为 RTC 时钟，由 PT4.0/PT4.1 口输入

专用微控制器的特性

- 上电复位 (POR)
- 上电复位延迟定时器 (39ms)
- 内带低电压复位 (LVR)
- 定时器 0
 - 可编程预分频的 8 位的定时器
- 定时/计数器 1
 - 可编程预分频的 8 位的分频器
- 看门狗定时器 (3K WDT)

外设特性

- 最多 15 个双向 I/O 口(PT1:4 个、PT2:8 个、PT4:3 个，其中 PT2 可以用作 LED 驱动口)
- 7 个内部中断：24bit-ADC、10bit-ADC、UART 接收/发送、TIMER0、TIMER1、RTC 中断
- 2 个外部中断：
 - INT0：PT1.0 或 PT1.2 或 PT2.0
 - INT1：PT1.1 或 PT1.3 或 PT2.1 或 PT4.1
- 7 个具有唤醒功能的输入口：PT1.0、PT1.1、PT1.2、PT1.3、PT2.0、PT2.1、

PT4.1

- 1 路蜂鸣器：由 PT1.2 输出，电流能力 6mA
- 1 路输入全差分 24bit Sigma-Delta 型 ADC，ADC 采样频率可选：250kHz、500kHz (参考 ICK=16MHz 计算值)
 - PGA 可选：1、16、64、128 倍，
 - 速率可选：30Hz、60Hz、120Hz、240Hz、480Hz、960Hz、1920Hz、3840Hz
- 内置温度传感器
- 内置 10 位 ADC，用来测量电池电压，信号由 PT2.2 口输入
- 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
- 低电压检测 (LVD) 引脚：PT1.3
- 1 路 UART：PT1.0/1.1，波特率 19200/9600
- 内置比较器：用来做电源电压比较和外部电压检测(即 LVD 功能)
- 内置 3.8V Charge pump(LED 模式下 PT2 口的驱动电压)，使用 charge pump 时需要接 20uF 的电容(两个 10uF 的)

功耗特性

- MCU 工作电流
 - 正常模式 1.2mA@16MHz(工作电压 3V，指令周期 2MHz)
 - 休眠模式下的电流小于 2μA

CMOS 技术

- 电压工作范围
 - DVDD 2.4V~3.6V
 - AVDD 2.4V~3.6V

封装

- SOP16L、SSOP20、SSOP24

应用场合

- 电子衡器
- 精密测量及控制系统

历史修改记录

时间	记录	版本号
2017-1-11	初稿完成	1.0
2017-6-2	1.增加 ICK_SEL 代码选项位，1 代表 14.7456MHz，0 代表 16MHz。用于客户选择 ICK 工作频率。	1.1
2017/8/2	修改表 1-1CSU18MB86-SSOP24 引脚说明表笔误	1.2
2017/10/18	(1) 修改笔误：20H 寄存器、波特率数值等。 (2) 增加 LED 相关寄存器的描述。 (3) 更新封装尺寸图。	1.3

目 录

历史修改记录.....	3
1 产品概述.....	6
1.1 主要特性.....	6
1.2 封装信息.....	6
2 标准功能.....	12
2.1 CPU 核.....	12
2.1.1 存储器.....	14
2.1.2 状态寄存器.....	18
2.1.3 中断寄存器.....	19
2.2 SFR.....	21
2.2.1 系统专用寄存器.....	21
2.2.2 辅助专用寄存器.....	21
2.3 时钟系统.....	23
2.3.1 振荡器.....	23
2.3.2 CPU 指令周期.....	23
2.3.3 TM0CLK (定时器 0 模块输入时钟).....	24
2.3.4 TM1CLK (定时器 1 模块输入时钟).....	25
2.3.5 UARTCLK.....	26
2.3.6 蜂鸣器时钟 (仅 CSU18MB86-SSOP24 支持).....	27
2.3.7 24bit-ADCCLK.....	28
2.3.8 LEDCLK.....	29
2.4 定时器 0.....	30
2.5 I/O PORT.....	32
2.5.1 数字 I/O 口、URAT 接口、外部中断输入与蜂鸣器输出: PT1[2:0].....	37
2.5.2 带模拟输入通道的数字 I/O 口与外部中断 1 输入: PT1[3].....	39
2.5.3 数字 I/O 口、外部中断输入与 LED 驱动: PT2[1:0].....	41
2.5.4 带模拟输入通道的数字 I/O 口与 LED 驱动: PT2[2].....	43
2.5.5 数字 I/O 口与 LED 驱动: PT2[7:3].....	45
2.5.6 带模拟输入通道的数字 I/O 口、外部中断 1 输入与 PWM 输出: PT4[1:0].....	47
2.5.7 数字 I/O 口: PT4[2].....	49
3 增强功能.....	51
3.1 电源系统.....	51
3.1.1 Regulator.....	51
3.1.2 低电压比较器.....	53
3.1.3 Charge pump.....	55
3.2 HALT 与 SLEEP 模式.....	57
3.3 复位系统.....	58
3.4 看门狗.....	59
3.5 定时/计数器 1.....	60
3.5.1 寄存器说明.....	60
3.5.2 定时器.....	61
3.5.3 蜂鸣器(仅 CSU18MB86-SSOP24 支持).....	62
3.5.4 PWM(仅 CSU18MB86-SSOP24 支持).....	62
3.6 24BIT-ADC 模块.....	63
3.6.1 24Bit ADC 寄存器说明.....	63

3.6.2	24Bit ADC 增益的温度特性调整	65
3.6.3	温度传感器	65
3.7	10Bit ADC 模块	67
3.7.1	10Bit ADC 寄存器说明	67
3.7.2	10BitADC 的操作步骤	67
3.8	LED DRIVER	68
3.8.1	寄存器说明	68
3.8.2	LED 的操作步骤	71
3.9	串行通信接口(CSU18MB86-SOP16 不支持)	72
3.9.1	工作方式	72
3.9.2	多机通信	78
3.9.3	寄存器说明	78
3.9.4	波特率	79
3.10	RTC(仅 CSU18MB86-SSOP24 支持)	80
3.10.1	功能描述	80
3.10.2	寄存器说明	80
3.10.3	RTC 的操作步骤	83
3.11	MTP 模块	84
3.12	MTP 在线烧录	85
4	MCU 指令集	87
5	电气特性	102
5.1	最大极限值	102
5.2	直流特性	102
5.3	24Bit ADC 的特性	103
6	封装	104
6.1	SOP16L 封装尺寸	104
6.2	SSOP20 封装尺寸	104
6.3	SSOP24 封装尺寸	105

1 产品概述

1.1 主要特性

CSU18MB86 系列芯片是一个 8 位 CMOS 单芯片 MCU，内置 8k×16 位 MTP 程序存储器、128 字节 EEPROM 和 488 字节数据存储器，带有 1 路全差分模拟信号输入的 24 位 ADC 和 1 路单端模拟信号输入的 10 位 ADC，带有 LED 驱动，最大可以直接驱动 56 个发光二极管。

1.2 封装信息

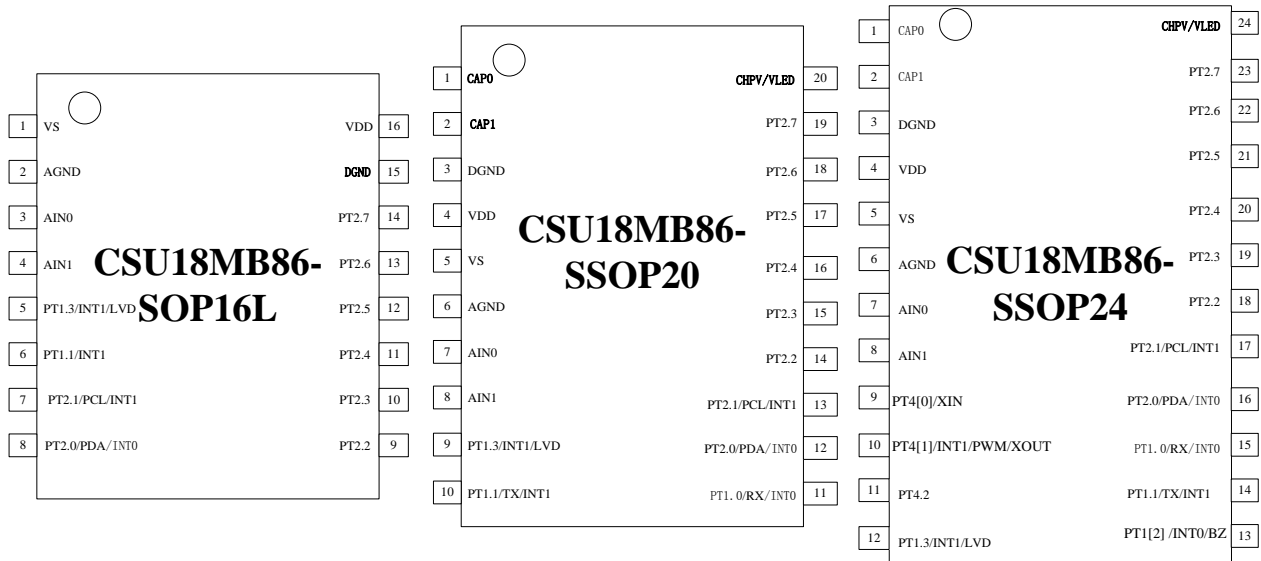


图 1-CSU18MB86 系列功能引脚图

表 1-1CSU18MB86-SOP16 引脚说明表

管脚名称	输入/输出	管脚序号	描述										
DGND	P	15	芯片数字地										
VDD	P	16	芯片电源，输入范围 2.4V~3.6V										
VS	O/I	1	内置 LDO 电压输出/24Bit ADC 参考电压输入 选择内置 LDO 电压输出时，输出电压四档可选：2.35V、2.45V、2.8V、3V，外接 1uF 电容 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LDOS[1:0]</th> <th>LDO 输出电压</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>3.0V</td> </tr> <tr> <td>01</td> <td>2.8V</td> </tr> <tr> <td>10</td> <td>2.45V</td> </tr> <tr> <td>11</td> <td>2.35V</td> </tr> </tbody> </table>	LDOS[1:0]	LDO 输出电压	00	3.0V	01	2.8V	10	2.45V	11	2.35V
LDOS[1:0]	LDO 输出电压												
00	3.0V												
01	2.8V												
10	2.45V												
11	2.35V												
AGND	P	2	芯片模拟地										
AIN0~1	I	3~4	24Bit ADC 模拟差分输入端，寄存器位 SINL[1:0]控制差分输入信号选择模式： =00：输入端连接到 AIN0 和 AIN1:AIN0 为 Vin+、AIN1 为 Vin- =01：内短 =10：24BitADC 输入端连接到内部温度传感器 =11：输入端连接到 AIN0 和 AIN1:AIN0 为 Vin-、AIN1 为 Vin+										

PT1[3]/INT1/LVD	I/O	5	I/O 或外部中断 1 输入或模拟低电压检测输入		
			控制信号	IO 功能描述	
			AIENB1=1 且 PTW1[1]=0	IO	
			AEINB1=1 且 PTW1[1]=1	外部中断 1 输入	
			AIENB1=0	自动开启低电压检测输入(默认值)	
PT1[1]/INT1/TX	I/O	6	IO 或外部中断 1 输入		
			控制信号	IO 功能描述	
			PTW1[0]=0	IO (默认值)	
			PTW1[0]=1	外部中断 1 输入	
PT2[1]/INT1/PCL/LED	I/O	7	IO 或外部中断 1 输入或 MTP 烧写的时钟或 LED 输出口, 默认为 MTP 烧写的时钟和数字 IO		
			注: PTW1[2]=0 时为 IO; PTW1[2]=1 时为外部中断 1 输入;		
			控制信号	IO 功能描述	
			PT2CON[1]=0 且 LEDEN=0	MTP 烧写的时钟和 IO (默认值) 或 INT1	
			PT2CON[1]=0 且 LEDEN=1	LED 输出口(电流能力可调)	
			PT2CON[1]=1 且 LEDEN=0	IO 或 INT1	
			PT2CON[1]=1 且 LEDEN=1	IO 或 INT1	
PT2[0]/INT0/PDA/LED	I/O	8	IO 或外部中断 0 输入或 MTP 烧写的的数据或 LED 输出口, 默认为 MTP 烧写的的数据和数字 IO		
			注: PTW0[2]=0 时为 IO; PTW0[2]=1 时为外部中断 0 输入;		
			控制信号	IO 功能描述	
			PT2CON[0]=0 且 LEDEN=0	MTP 烧写的的数据和 IO (默认值) 或 INTO	
			PT2CON[0]=0 且 LEDEN=1	LED 输出口(电流能力可调)	
			PT2CON[0]=1 且 LEDEN=0	IO 或 INTO	
			PT2CON[0]=1 且 LEDEN=1	IO 或 INTO	
PT2[2]/LED/AIN	I/O	9	IO 或 LED 输出口或 10-Bit ADC 信号输入(单端输入模式), 默认为数字 IO		
			控制信号	IO 功能描述	
			AIENB2=0	模拟口 做为 10Bit ADC 信号输入口	
			AIENB2=1	PT2CON[x]=0 且 LEDEN=0	IO (默认值)
				PT2CON[x]=0 且 LEDEN=1	LED 输出口(电流能力可调)
			PT2CON[x]=1 且 LEDEN=X	IO	
			PT2CON[x]=1 且 LEDEN=1	IO	
PT2[3]/LED ~2[7]/LED	I/O	10~14	IO 或 LED 输出口, 默认为数字 IO		
			注: 下表中的 x=3~7, 对应控制的是 PT2[3]~2[7]		
			控制信号	IO 功能描述	
			PT2CON[x]=0 且 LEDEN=0	IO (默认值)	
			PT2CON[x]=0 且 LEDEN=1	LED 输出口(电流能力可调)	
			PT2CON[x]=1 且 LEDEN=0	IO	
			PT2CON[x]=1 且 LEDEN=1	IO	

表 1-2CSU18MB86-SSOP20 引脚说明表

管脚名称	输入/输出	管脚序号	描述
CAPO/CAP1	AI	1、2	模拟电容输入口, 使用时跨接 1uF 电容
DGND	P	3	芯片数字地

VDD	P	4	芯片电源，输入范围 2.4V~3.6V										
VS	O/I	5	<p>内置 LDO 电压输出/24Bit ADC 参考电压输入 选择内置 LDO 电压输出时，输出电压四档可选：2.35V、2.45V、2.8V、3V，外接 1uF 电容</p> <table border="1"> <thead> <tr> <th>LDOS[1:0]</th> <th>LDO 输出电压</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>3.0V</td> </tr> <tr> <td>01</td> <td>2.8V</td> </tr> <tr> <td>10</td> <td>2.45V</td> </tr> <tr> <td>11</td> <td>2.35V</td> </tr> </tbody> </table>	LDOS[1:0]	LDO 输出电压	00	3.0V	01	2.8V	10	2.45V	11	2.35V
LDOS[1:0]	LDO 输出电压												
00	3.0V												
01	2.8V												
10	2.45V												
11	2.35V												
AGND	P	6	芯片模拟地										
AIN0~1	I	7~8	<p>24Bit ADC 模拟差分输入端，寄存器位 SINL[1:0]控制差分输入信号选择模式： =00：输入端连接到 AIN0 和 AIN1:AIN0 为 Vin+、AIN1 为 Vin- =01：内短 =10：24BitADC 输入端连接到内部温度传感器 =11：输入端连接到 AIN0 和 AIN1:AIN0 为 Vin-、AIN1 为 Vin+</p>										
PT1[3]/INT1/LVD	I/O	9	<p>I/O 或外部中断 1 输入或模拟低电压检测输入</p> <table border="1"> <thead> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>AIENB1=1 且 PTW1[1]=0</td> <td>IO</td> </tr> <tr> <td>AEINB1=1 且 PTW1[1]=1</td> <td>外部中断 1 输入</td> </tr> <tr> <td>AIENB1=0</td> <td>自动开启低电压检测输入(默认值)</td> </tr> </tbody> </table>	控制信号	IO 功能描述	AIENB1=1 且 PTW1[1]=0	IO	AEINB1=1 且 PTW1[1]=1	外部中断 1 输入	AIENB1=0	自动开启低电压检测输入(默认值)		
控制信号	IO 功能描述												
AIENB1=1 且 PTW1[1]=0	IO												
AEINB1=1 且 PTW1[1]=1	外部中断 1 输入												
AIENB1=0	自动开启低电压检测输入(默认值)												
PT1[1]/INT1/TX	I/O	10	<p>IO 或外部中断 1 输入或串口输出</p> <table border="1"> <thead> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>UARTEN=0 且 PTW1[0]=0</td> <td>IO (默认值)</td> </tr> <tr> <td>UARTEN=0 且 PTW1[0]=1</td> <td>外部中断 1 输入</td> </tr> <tr> <td>UARTEN=1</td> <td>自动开启串口输出</td> </tr> </tbody> </table>	控制信号	IO 功能描述	UARTEN=0 且 PTW1[0]=0	IO (默认值)	UARTEN=0 且 PTW1[0]=1	外部中断 1 输入	UARTEN=1	自动开启串口输出		
控制信号	IO 功能描述												
UARTEN=0 且 PTW1[0]=0	IO (默认值)												
UARTEN=0 且 PTW1[0]=1	外部中断 1 输入												
UARTEN=1	自动开启串口输出												
PT1[0]/INT0/RX	I/O	11	<p>I/O 或外部中断 0 输入或串口输入</p> <table border="1"> <thead> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>UARTEN=0 且 PTW0[0]=0</td> <td>IO (默认值)</td> </tr> <tr> <td>UARTEN=0 且 PTW0[0]=1</td> <td>外部中断 0 输入</td> </tr> <tr> <td>UARTEN=1</td> <td>自动开启串口输入</td> </tr> </tbody> </table>	控制信号	IO 功能描述	UARTEN=0 且 PTW0[0]=0	IO (默认值)	UARTEN=0 且 PTW0[0]=1	外部中断 0 输入	UARTEN=1	自动开启串口输入		
控制信号	IO 功能描述												
UARTEN=0 且 PTW0[0]=0	IO (默认值)												
UARTEN=0 且 PTW0[0]=1	外部中断 0 输入												
UARTEN=1	自动开启串口输入												
PT2[0]/INT0/PDA/LED	I/O	12	<p>IO 或外部中断 0 输入或 MTP 烧写的的数据或 LED 输出口，默认为 MTP 烧写的的数据和数字 IO 注：PTW0[2]=0 时为 IO；PTW0[2]=1 时为外部中断 0 输入；</p> <table border="1"> <thead> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>PT2CON[0]=0 且 LEDEN=0</td> <td>MTP 烧写的的数据和 IO (默认值) 或 INTO</td> </tr> <tr> <td>PT2CON[0]=0 且 LEDEN=1</td> <td>LED 输出口(电流能力可调)</td> </tr> <tr> <td>PT2CON[0]=1 且 LEDEN=0</td> <td>IO 或 INTO</td> </tr> <tr> <td>PT2CON[0]=1 且 LEDEN=1</td> <td>IO 或 INTO</td> </tr> </tbody> </table>	控制信号	IO 功能描述	PT2CON[0]=0 且 LEDEN=0	MTP 烧写的的数据和 IO (默认值) 或 INTO	PT2CON[0]=0 且 LEDEN=1	LED 输出口(电流能力可调)	PT2CON[0]=1 且 LEDEN=0	IO 或 INTO	PT2CON[0]=1 且 LEDEN=1	IO 或 INTO
控制信号	IO 功能描述												
PT2CON[0]=0 且 LEDEN=0	MTP 烧写的的数据和 IO (默认值) 或 INTO												
PT2CON[0]=0 且 LEDEN=1	LED 输出口(电流能力可调)												
PT2CON[0]=1 且 LEDEN=0	IO 或 INTO												
PT2CON[0]=1 且 LEDEN=1	IO 或 INTO												
PT2[1]/INT1/PCL/LED	I/O	13	<p>IO 或外部中断 1 输入或 MTP 烧写的的时钟或 LED 输出口，默认为 MTP 烧写的的时钟和数字 IO 注：PTW1[2]=0 时为 IO；PTW1[2]=1 时为外部中断 1 输入；</p> <table border="1"> <thead> <tr> <th>控制信号</th> <th>IO 功能描述</th> </tr> </thead> <tbody> <tr> <td>PT2CON[1]=0 且 LEDEN=0</td> <td>MTP 烧写的的时钟和 IO (默认值) 或 INT1</td> </tr> <tr> <td>PT2CON[1]=0 且 LEDEN=1</td> <td>LED 输出口(电流能力可调)</td> </tr> <tr> <td>PT2CON[1]=1 且 LEDEN=0</td> <td>IO 或 INT1</td> </tr> <tr> <td>PT2CON[1]=1 且 LEDEN=1</td> <td>IO 或 INT1</td> </tr> </tbody> </table>	控制信号	IO 功能描述	PT2CON[1]=0 且 LEDEN=0	MTP 烧写的的时钟和 IO (默认值) 或 INT1	PT2CON[1]=0 且 LEDEN=1	LED 输出口(电流能力可调)	PT2CON[1]=1 且 LEDEN=0	IO 或 INT1	PT2CON[1]=1 且 LEDEN=1	IO 或 INT1
控制信号	IO 功能描述												
PT2CON[1]=0 且 LEDEN=0	MTP 烧写的的时钟和 IO (默认值) 或 INT1												
PT2CON[1]=0 且 LEDEN=1	LED 输出口(电流能力可调)												
PT2CON[1]=1 且 LEDEN=0	IO 或 INT1												
PT2CON[1]=1 且 LEDEN=1	IO 或 INT1												

PT2[2]/LED/AIN	I/O	14	IO 或 LED 输出口或 10-Bit ADC 信号输入(单端输入模式), 默认为数字 IO		
			控制信号	IO 功能描述	
			AIENB2=0	模拟口 做为 10Bit ADC 信号输入口	
			AIENB2=1	PT2CON[x]=0 且 LEDEN=0	IO (默认值)
				PT2CON[x]=0 且 LEDEN=1	LED 输出口(电流能力可调)
PT2CON[x]=1 且 LEDEN=X	IO				
PT2CON[x]=1 且 LEDEN=1	IO				
PT2[3]/LED ~2[7]/LED	I/O	15~19	IO 或 LED 输出口, 默认为数字 IO 注: 下表中的 x=3~7, 对应控制的是 PT2[3]~2[7]		
			控制信号	IO 功能描述	
			PT2CON[x]=0 且 LEDEN=0	IO (默认值)	
			PT2CON[x]=0 且 LEDEN=1	LED 输出口(电流能力可调)	
			PT2CON[x]=1 且 LEDEN=0	IO	
PT2CON[x]=1 且 LEDEN=1	IO				
CHPV/VLED	P	20	ChargePump 电源输出口或 PT2 LED 电源驱动输入口		

表 1-3CSU18MB86-SSOP24 引脚说明表

管脚名称	输入/输出	管脚序号	描述	
CAP0/CAP1	AI	1~2	模拟电容输入口, 使用时跨接 1uF 电容	
DGND	P	3	芯片数字地	
VDD	P	4	芯片电源, 输入范围 2.4V~3.6V	
VS	O/I	5	内置 LDO 电压输出/24Bit ADC 参考电压输入 选择内置 LDO 电压输出时, 输出电压四档可选: 2.35V、2.45V、2.8V、3V, 外接 1uF 电容	
			LDOS[1:0]	LDO 输出电压
			00	3.0V
			01	2.8V
			10	2.45V
11	2.35V			
AGND	P	6	芯片模拟地	
AIN0~1	I	7~8	24Bit ADC 模拟差分输入端, 寄存器位 SINL[1:0]控制差分输入信号选择模式: =00: 输入端连接到 AIN0 和 AIN1:AIN0 为 Vin+、AIN1 为 Vin- =01: 内短 =10: 24BitADC 输入端连接到内部温度传感器 =11: 输入端连接到 AIN0 和 AIN1:AIN0 为 Vin-、AIN1 为 Vin+	
PT4[0]/XIN	I/O	9	IO 或 32768Hz 晶振输入	
			控制信号	IO 功能描述
			AIENB3=1	IO(默认值)
AIENB3=0	32768Hz 时钟输入			
PT4[1]/INT1/PWM/XOUT	I/O	10	IO 或外部中断 1 输入或 PWM 输出或 32768Hz 晶振输出	
			控制信号	IO 功能描述
			AIENB3=1 且 PTW1[3]=0	IO(默认值)
			AIENB3=1 且 PTW1[3]=1	外部中断 1 输入
			AIENB3=1 且 PWM1OUT=1	PWM1 输出(需配置 IO 方向)
			AIENB3=1、PWM1OUT=0 且 T1OUT=1	蜂鸣器 1 输出
AIENB3=0	32768Hz 晶振输出			

PT4[2]	I/O	11	IO		
PT1[3]/INT1/LVD	I/O	12	I/O 或外部中断 1 输入或模拟低电压检测输入		
			控制信号	IO 功能描述	
			AIENB1=1 且 PTW1[1]=0	IO	
			AEINB1=1 且 PTW1[1]=1	外部中断 1 输入	
			AIENB1=0	自动开启低电压检测输入 (默认值)	
PT1[2] /INT0/BZ	I/O	13	I/O 或外部中断 0 输入或蜂鸣器输出		
			控制信号	IO 功能描述	
			BZEN=0 且 PTW0[1]=0	IO (默认值)	
			BZEN=0 且 PTW0[1]=1	外部中断 0 输入	
			BZEN=1	蜂鸣器输出	
PT1[1] /INT1/TX	I/O	14	IO 或外部中断 1 输入或串口输出		
			控制信号	IO 功能描述	
			UARTEN=0 且 PTW1[0]=0	IO (默认值)	
			UARTEN=0 且 PTW1[0]=1	外部中断 1 输入	
			UARTEN=1	自动开启串口输出	
PT1[0]/INT0/RX	I/O	15	I/O 或外部中断 0 输入或串口输入		
			控制信号	IO 功能描述	
			UARTEN=0 且 PTW0[0]=0	IO (默认值)	
			UARTEN=0 且 PTW0[0]=1	外部中断 0 输入	
			UARTEN=1	自动开启串口输入	
PT2[0]/INT0/PDA/LED	I/O	16	IO 或外部中断 0 输入或 MTP 烧写的数字或 LED 输出口, 默认为 MTP 烧写的数字和数字 IO		
			注: PTW0[2]=0 时为 IO; PTW0[2]=1 时为外部中断 0 输入;		
			控制信号	IO 功能描述	
			PT2CON[0]=0 且 LEDEN=0	MTP 烧写的数字和 IO (默认值) 或 INT0	
			PT2CON[0]=0 且 LEDEN=1	LED 输出口(电流能力可调)	
			PT2CON[0]=1 且 LEDEN=0	IO 或 INT0	
			PT2CON[0]=1 且 LEDEN=1	IO 或 INT0	
PT2[1] /INT1/PCL/LED	I/O	17	IO 或外部中断 1 输入或 MTP 烧写的时钟或 LED 输出口, 默认为 MTP 烧写的时钟和数字 IO		
			注: PTW1[2]=0 时为 IO; PTW1[2]=1 时为外部中断 1 输入;		
			控制信号	IO 功能描述	
			PT2CON[1]=0 且 LEDEN=0	MTP 烧写的时钟和 IO (默认值) 或 INT1	
			PT2CON[1]=0 且 LEDEN=1	LED 输出口(电流能力可调)	
			PT2CON[1]=1 且 LEDEN=0	IO 或 INT1	
			PT2CON[1]=1 且 LEDEN=1	IO 或 INT1	
PT2[2]/LED/AIN	I/O	18	IO 或 LED 输出口或 10-Bit ADC 信号输入(单端输入模式), 默认为数字 IO		
			控制信号	IO 功能描述	
			AIENB2=0	模拟口 做为 10Bit ADC 信号输入口	
			AIENB2=1	PT2CON[x]=0 且 LEDEN=0	IO (默认值)
				PT2CON[x]=0 且 LEDEN=1	LED 输出口(电流能力可调)
PT2CON[x]=1 且 LEDEN=X	IO				
			PT2CON[x]=1 且 LEDEN=1	IO	

PT2[3]/LED ~2[7]/LED	I/O	19~23	IO 或 LED 输出口，默认为数字 IO	
			注：下表中的 x=3~7，对应控制的是 PT2[3]~2[7]	
			控制信号	IO 功能描述
			PT2CON[x]=0 且 LEDEN=0	IO（默认值）
			PT2CON[x]=0 且 LEDEN=1	LED 输出口 (电流能力可调)
PT2CON[x]=1 且 LEDEN=0	IO			
PT2CON[x]=1 且 LEDEN=1	IO			
CHPV/VLED	P	24	ChargePump 电源输出口或 PT2 LED 电源驱动输入口	

2 标准功能

2.1 CPU 核

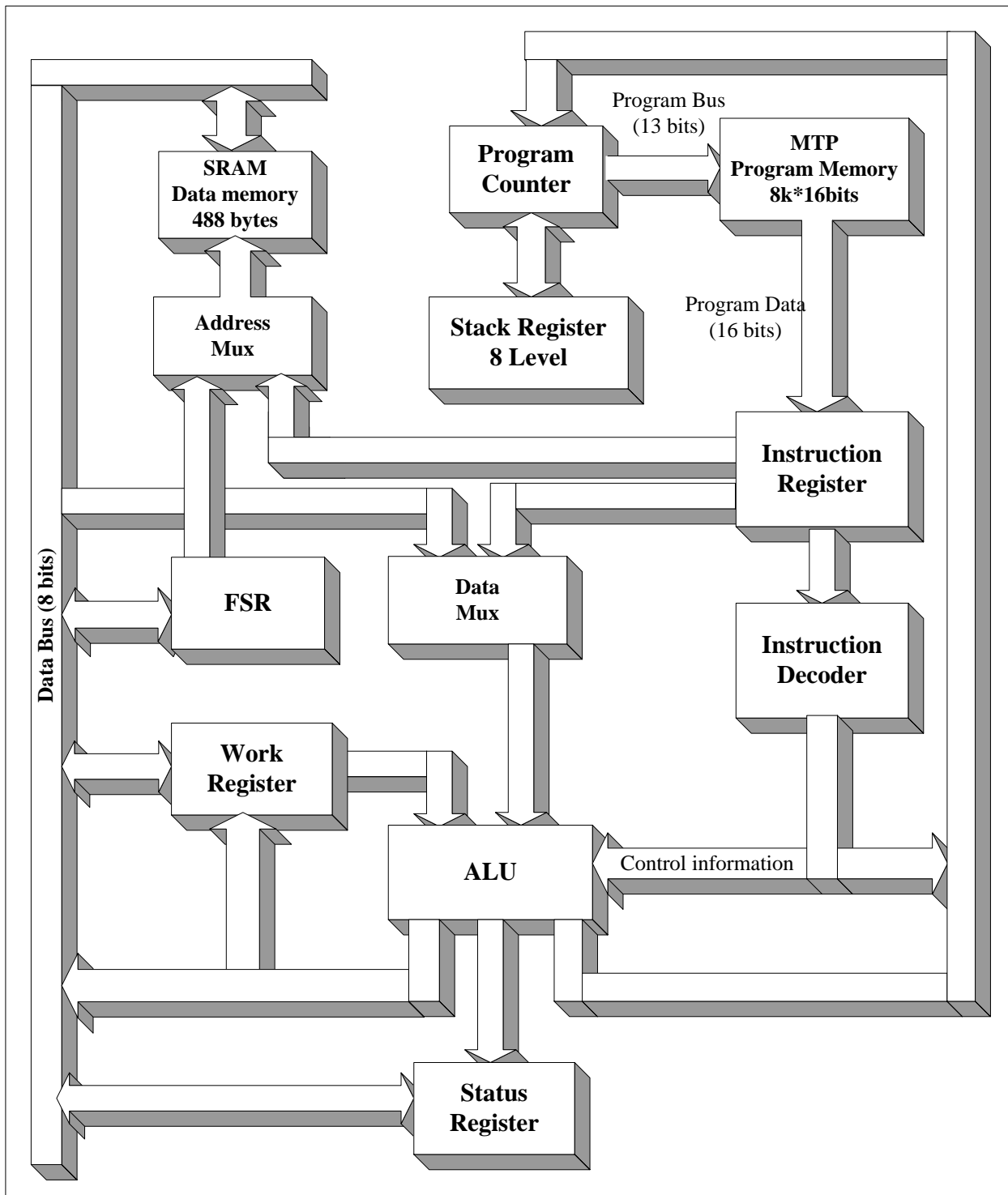


图 2-1 CSU18MB86 系列 CPU 核的功能模块图

从 CPU 核的功能模块图中，可以看到它主要包含 8 个主要寄存器及 3 个存储器单元。

表 2-1 MCU 架构说明

模块名称	描述
程序计数器	此寄存器在 CPU 的工作周期期间起到很重要的作用，它记录 CPU 每个周期处理程序存储器中指令的指针。在一个 CPU 周期中，程序计数器将程序存储器地址（13bits），指令指针推送到程序存储器，然后自动加 1 以进行下一次周期。
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>CSU18MB86 系列的指令是 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） 数据存储器地址。CPU 能利用此地址来对数据存储器进行操作。</p> <p>直接数据（8bits） CPU 通过 ALU 利用此数据对工作寄存器进行操作。</p> <p>控制信息 它记录着 ALU 的操作信息。</p>
指令译码器	指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。
算术逻辑单元	算术逻辑单元不仅能完成 8 位二进制的加，减，加 1，减 1 等算术计算，还能对 8 位变量进行逻辑的与，或，异或，循环移位，求补，清零等逻辑运算。
工作寄存器	工作寄存器是用来缓存数据存储器中某些存储地址的数据。
状态寄存器	当 CPU 利用 ALU 处理寄存器数据时，如下的状态会随着如下顺序变化：PD，TO，DC，C 及 Z。
文件选择寄存器	在 CSU18MB86 系列的指令集中，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	CSU18MB86 系列内带 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	CSU18MB86 系列内带 128×8 位的 EEPROM，EEPROM 会占用 MTP 的地址线。
数据存储器	CSU18MB86 系列内带 488 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

程序存储器主要用于指令的存储，在 CSU18MB86 系列中，该程序存储器是 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。

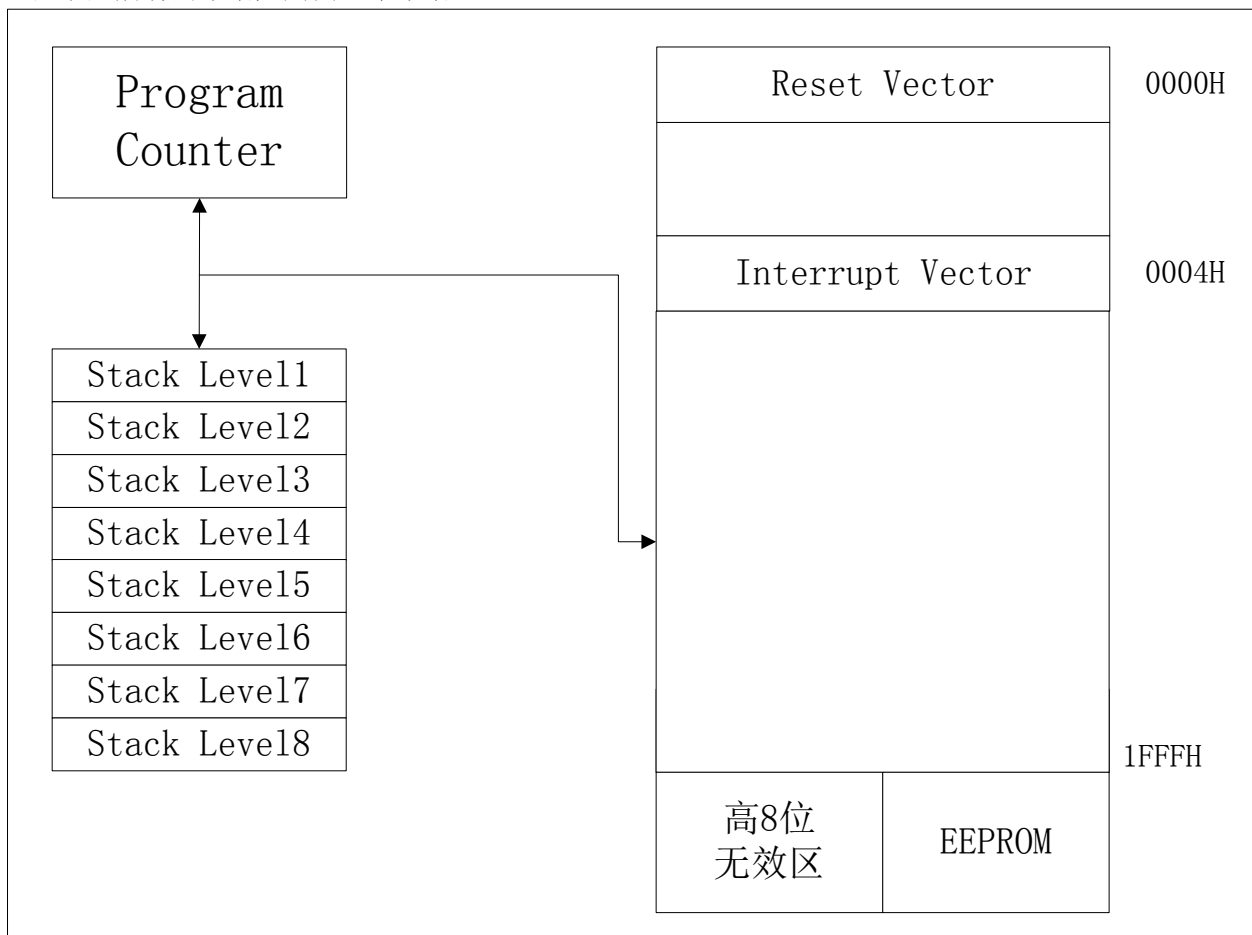


图 2-2 程序存储器

(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-207FH。

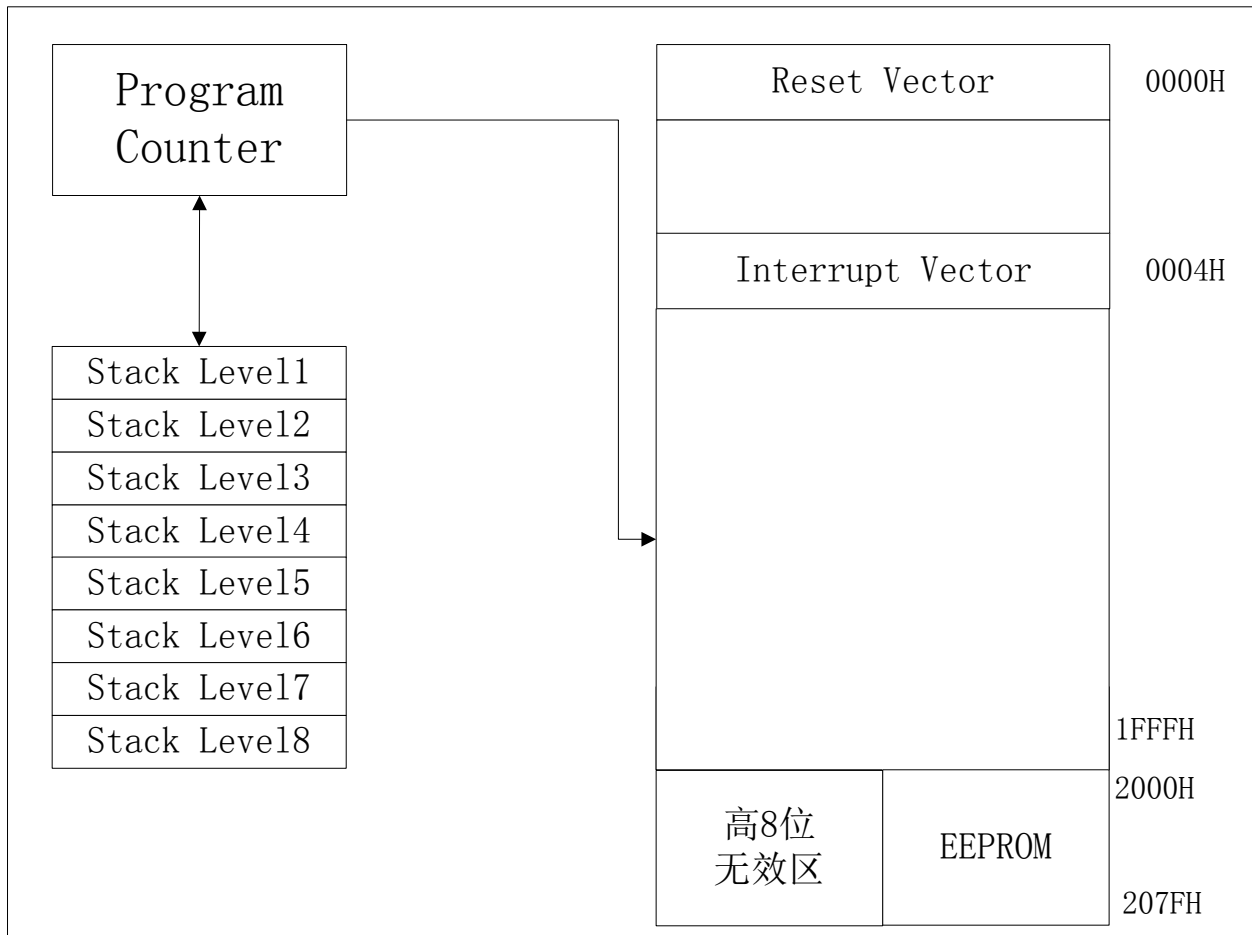


图 2-3 非易性数据存储器

(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。

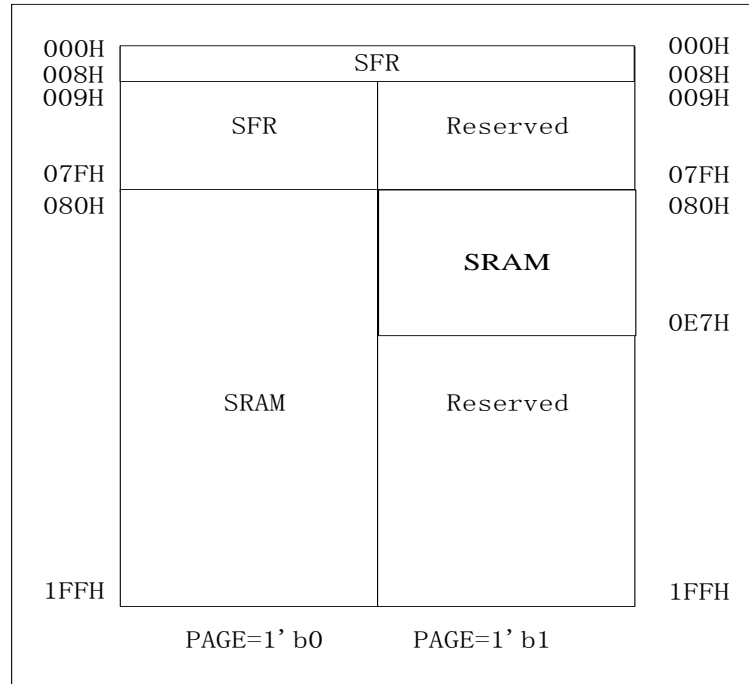


图 2-4 数据存储器

表 2-2 数据存储器地址分配

数据存储器	起始地址	结束地址
系统特殊功能寄存器	000H	008H
外设特殊功能寄存器	009H	07FH
通用数据存储器 (PAGE=1'b0)	080H	1FFH
通用数据存储器 (PAGE=1'b1)	080H	0E7H

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 PAGE1: 页选择位

- 1 = 间接寻址IND1时, 访问后512byte地址
- 0 = 间接寻址IND1时, 访问前512byte地址

Bit0 PAGE0: 页选择位

- 1 = 直接和间接寻址IND0时, 访问后512byte地址
- 0 = 直接和间接寻址IND0时, 访问前512byte地址

(注意第二页中只有系统特殊功能寄存器和128Byte的SRAM可以访问, 其余地址保留)

通过 IND0 及 PAGE0、FSR0 或 IND1 及 PAGE1、FSR1 寄存器可以对数据存储器以及特殊功能寄存器进行间接访问。当从间接地址寄存器(IND0/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 2-5 间接地址访问。

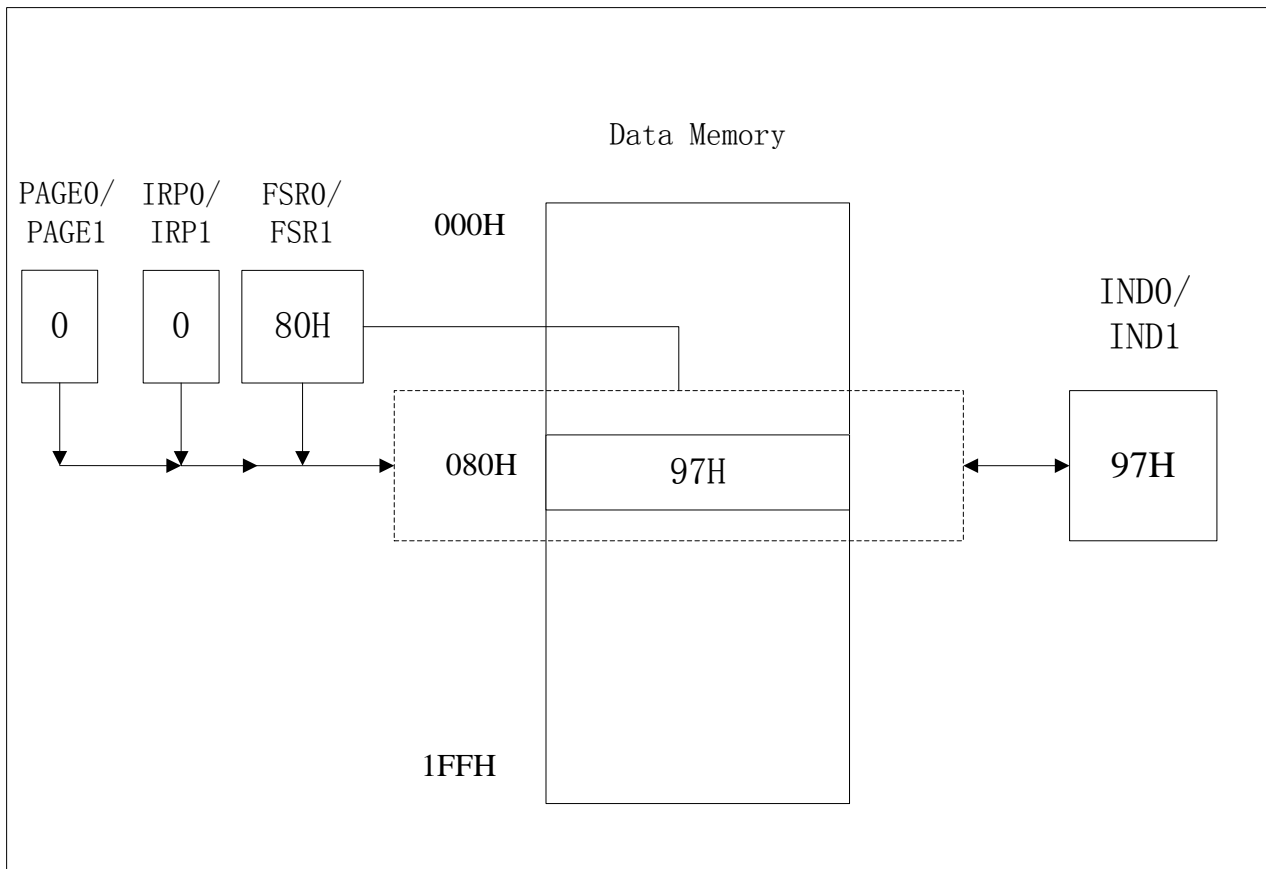


图 2-5 间接地址访问

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，用于中断器件。

INTE 寄存器（地址为 07H）

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

Bit 7 GIE: 全局中断使能标志
1 = 使能所有非屏蔽中断
0 = 不使能所有中断

Bit 5 TM1IE: 8-Bit 定时器 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

Bit3 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: 24-bit AD 中断使能标志
1 = 使能 AD 中断
0 = 不使能 AD 中断

Bit 1 E1IE: 外部中断 1 使能标志
1 = 使能外部中断 1
0 = 不使能外部中断 1

Bit 0 E0IE: 外部中断 0 使能标志
1 = 使能外部中断 0
0 = 不使能外部中断 0

INTF 寄存器（地址为 06H）

特性	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTF			TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 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 TMOIF: 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	U-0	U-0	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0
INTE2					RTCIE		URTIE	URRIE
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3 RTCIE: RTC 中断使能标志
1 = 使能 RTC 中断
0 = 不使能 RTC 中断

Bit 1 URTIE: 串口发送中断使能标志
1 = 使能串口发送中断
0 = 不使能串口发送中断

Bit 0 URRIE: 串口接收中断使能标志
1 = 使能串口接收中断
0 = 不使串口接收中断

INTF2 寄存器 (地址为 32H)

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

Bit 3 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

Bit 1 URTIF: 串口通信发送中断标志
1 = 发生串口发送中断，必须软件清 0
0 = 没有发生串口发送中断

Bit0 URRIF: 串口通信接收中断标志
1 = 发生串口接收中断，必须软件清 0
0 = 没有发生串口接收中断

特性 (Property) :

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



2.2 SFR

2.2.1 系统专用寄存器

系统专用寄存器用于完成 CPU 核的功能，由间接地址，间接地址指针，状态寄存器，工作寄存器，中断标志及中断控制寄存器。

表 2-3 系统寄存器表

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	-	-	TM1IF	TM0IF	AD2IF	AD1IF	E1IF	E0IF	uu000000
07H	INTE	GIE	-	TM1IE	TM0IE	AD2IE	AD1IE	E1IE	E0IE	0u000000
08H	BSR	IRP0	IRP1	-	-	-	-	PAGE1	PAGE0	00uuuu00

2.2.2 辅助专用寄存器

辅助专用寄存器是为辅助功能而设计，比如I/O口，定时器 0，定时器 1，24-bit ADC，10-bit ADC，信号的条件控制寄存器，UART，LED驱动。详细描述请看表 2-4 辅助专用寄存器列表及以下章节。

表 2-4 辅助专用寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
0AH	EADRH	-	-	PARH[5:0]						-	uu000000
0BH	EADRL	PARL[7:0]								00000000	
0CH	EDAT	EDAT[7:0]								11111111	
0DH	EOPEN	EOPEN[7:0]								00000000	
0EH	WDTCON	WDTEN	-	-	-	-	WDTS[2:0]			0uuuu000	
0FH	WDTIN	WDTIN[7:0]								10111011	
10H	ADOH	ADO [23:16]								00000000	
11H	ADOL	ADO[15:8]								00000000	
12H	ADOLL	ADO[7:0]								00000000	
13H	ADCON	-	-	-	-	ADSC	ADM[2:0]			uuuu0000	
14H	MCK	-	-	-	-	-	M2_CK	M1_CK	-	uuuuu11u	
15H	PCK	-	-	-	-	-	S_beep[1:0]		-	u000000u	
18H	NETA	SINL[1:0]		-	CM_SEL	-	-			00u0uuuu	
1AH	NETC	CHOPM[1:0]		-	-	ADG[1:0]		ADEN		0000000u	
1CH	NETE	LDOS[1:0]		-	-	SILB[2:0]		ENLB	LB_RST_CON	00u00000	
1DH	NETF	-	-	LDOEN	-	-	BGID[1:0]		-	uU0uu01u	
1FH	SVD	-								LBOU	uuuuuuux
20H	PT1	-	-	-	-	PT1[3:0]				uuuuxxxx	
21H	PT1EN	-	-	-	-	PT1EN[3:0]				uuuu0000	
22H	PT1PU	-	-	-	-	PT1PU[3:0]				uuuu0000	
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1	-			1uu10uuu	
24H	PT2	PT2[7:0]								xxxxxxx	
25H	PT2EN	PT2EN[7:0]								00000000	
26H	PT2PU	PT2PU[7:0]								00000000	
27H	PT4	-				PT4[2:0]				uuuuuuuu	
28H	PT4EN	-				PT4EN[2:0]				uuuuu000	
29H	PT4PU	-				PT4PU[2:0]				uuuuu000	
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]		0uuu0000	
2EH	PT2CON	PT2CON[7:0]								00000000	
2FH	PTINT	PTW1[3:0]				PTW0[2:0]				0000u000	
32H	INTF2	-	-	-	-	RTCIF	-	URTIF	URRIF	uuuu0u00	
33H	INTE2	-	-	-	-	RTCIE	-	URTIE	URRIE	uuuu0u00	
34H	TM0CON	TOEN	TORATE[2:0]			-		TORSTB	TOSEL	0000u1u0	
35H	TM0IN	TM0IN[7:0]								11111111	
36H	TM0CNT	TM0CNT[7:0]								00000000	
37H	TM1CON	T1EN	T1RATE[2:0]			T1CKS	T1RSTB	T1IOUT	PWM1IOUT	00000100	
38H	TM1IN	TM1IN[7:0]								11111111	
39H	TM1CNT	TM1CNT[7:0]								00000000	
3AH	TM1R	TM1R[7:0]								00000000	



3BH	LED1	LED1[7:0]						00000000	
3CH	LED2	LED2[7:0]						00000000	
3DH	LED3	LED3[7:0]						00000000	
3EH	LED4	LED4[7:0]						00000000	
3FH	LED5	LED5[7:0]						00000000	
40H	LED6	LED6[7:0]						00000000	
41H	LED7	LED7[7:0]						00000000	
42H	LEDCON1	LED_CURRENT[2:0]		-		-		LED_PMODE	00000100
44H	CHPCON	-	-	-	CHPV5	-	CHPCLKS[1:0]	CHPEN	uuu0u100
48H	AD2OH	AD2EN	-	-	-	-	AD2O[9:8]		0uuuuu00
49H	AD2OL	AD2O[7:0]						00000000	
59H	TEMPC	TEMPC [7:0]						00000000	
60H	RTCCON	LIR	-	24hr/12hr	-	RTCEN	-		0uuu0uu0
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	-	-	10HOUR10(0~2)		1HOUR(0~9)		uu000001	
66H	RTCMIN	-	10MIN(0~5)			1MIN(0~9)		u0000000	
67H	RTCSEC	-	10SEC(0~5)			1SEC(0~9)		u0000000	
68H	RTCDWR	-	-	-	-	-	DWR[2:0] (0~6)		uuuuuu000
69H	INTEGER	-	-	-	-	INTEGER[3:0]		uuuu0111	
6AH	FRACTION	FRACTION						uu000000	
7AH	SCON1	SM0	SM1	SM2	REN	TB8	RB8	UARTEN	00000u0
7BH	SCON2	SMOD	-	-	-	-	-	-	0uuuuuuu
7CH	SBUF	-						00000000	
7DH	LEDEN	LEDEN[7:0]						00000000	
7FH	LEDZCON	LEDZCON[7:0]						00000000	

2.3 时钟系统

2.3.1 振荡器

CSU18MB86 系列内置 14.7456MHz/16MHz 振荡器和 3KHz WDT 振荡器。

在正常工作时，内置 14.7456MHz/16MHz 振荡器一直工作；在执行 sleep 指令后，内置 14.7456MHz/16MHz 振荡器自动停止工作，减小功耗。

表 2-5 内部振荡器状态选择列表

睡眠指令 (sleep)	内部振荡器状态
1	Disable
0	Enable

CSU18MB86 系列内置 3KHz WDT 振荡器，受 wdten 的控制，wdten=1 时 3KHz WDT 振荡器工作，wdten 默认值为 0。

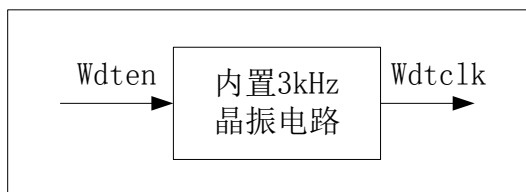


图 2-6 CSU18MB86 系列 Wdt 振荡器状态框图

2.3.2 CPU 指令周期

表 2-6 CSU18MB86 系列 CPU 指令周期寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
14H	MCK	-	-	-	-	-	M2_CK	M1_CK	-	uuuuu11u

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

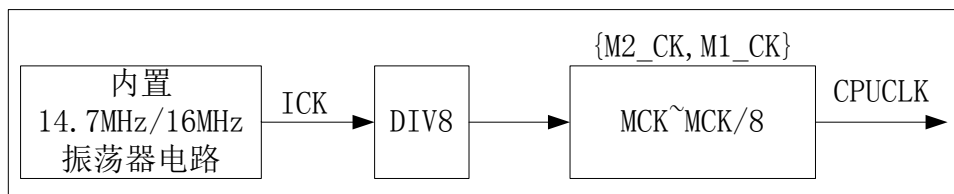


图 2-7 CPU 指令周期时钟示意图

表 2-7 指令周期选择列表

M2_CK	M1_CK	指令周期(KHz)
0	0	250
0	1	500
1	0	1000
1	1	2000(默认值)

备注：上表指令周期是参考 ICK=16MHz 来计算的，对于 ICK=14.7456MHz 可依比例计算获得。

2.3.3 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 用作定时器 0 模块的时钟。

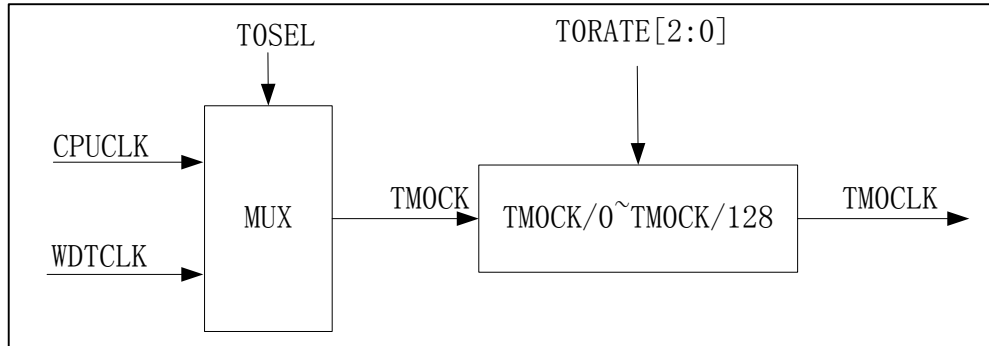


图 2-8 定时器 0 时钟分频示意图

表 2-8 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	-	T0SEL	0000u1u0

表 2-9 TM0CON 寄存器各位功能表

位地址	标识符	功能																		
7	TOEN	定时器 0 使能位 1: 使能定时器 0 0: 禁止定时器 0																		
6:4	TORATE[2:0]	定时器 0 时钟选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>TORATE [2:0]</th> <th>TM0CLK</th> </tr> </thead> <tbody> <tr><td>000</td><td>TM0CK</td></tr> <tr><td>001</td><td>TM0CK /2</td></tr> <tr><td>010</td><td>TM0CK /4</td></tr> <tr><td>011</td><td>TM0CK /8</td></tr> <tr><td>100</td><td>TM0CK /16</td></tr> <tr><td>101</td><td>TM0CK /32</td></tr> <tr><td>110</td><td>TM0CK /64</td></tr> <tr><td>111</td><td>TM0CK /128</td></tr> </tbody> </table>	TORATE [2:0]	TM0CLK	000	TM0CK	001	TM0CK /2	010	TM0CK /4	011	TM0CK /8	100	TM0CK /16	101	TM0CK /32	110	TM0CK /64	111	TM0CK /128
TORATE [2:0]	TM0CLK																			
000	TM0CK																			
001	TM0CK /2																			
010	TM0CK /4																			
011	TM0CK /8																			
100	TM0CK /16																			
101	TM0CK /32																			
110	TM0CK /64																			
111	TM0CK /128																			
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	NC	保留位																		
0	T0SEL	时钟源选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>T0SEL</th> <th>定时器 0 时钟源(TM0CK)</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 晶振打开时有效</td></tr> </tbody> </table>	T0SEL	定时器 0 时钟源(TM0CK)	0	CPUCLK	1	内部 3K WDT 时钟， 仅当内部 WDT 晶振打开时有效												
T0SEL	定时器 0 时钟源(TM0CK)																			
0	CPUCLK																			
1	内部 3K WDT 时钟， 仅当内部 WDT 晶振打开时有效																			

2.3.4 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。

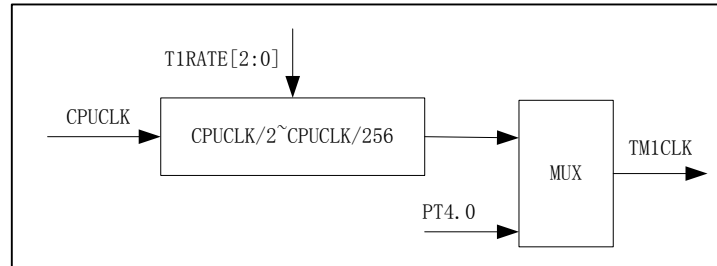


图 2-9 定时器 1 时钟分频示意图

表 2-10 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]			T1CKS	T1RSTB	T1IOUT	PWM1IOUT	00000100

表 2-11 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"> <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	T1CKS	定时/计数器 1 时钟源选择位，仅 CSU18MB86-SSOP24 支持 1: PT4.0 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	T1IOUT	PT4.1 口输出控制 <table border="1"> <thead> <tr> <th>T1IOUT</th> <th>PWM1IOUT</th> <th>PT4.1 输出控制，仅当 PT4.1 配置为输出有效</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>IO 输出</td></tr> <tr><td>0</td><td>1</td><td>IO 输出</td></tr> <tr><td>1</td><td>0</td><td>PWM2 输出</td></tr> <tr><td>1</td><td>1</td><td>PWM2 取反输出</td></tr> </tbody> </table>	T1IOUT	PWM1IOUT	PT4.1 输出控制，仅当 PT4.1 配置为输出有效	0	0	IO 输出	0	1	IO 输出	1	0	PWM2 输出	1	1	PWM2 取反输出			
T1IOUT	PWM1IOUT	PT4.1 输出控制，仅当 PT4.1 配置为输出有效																		
0	0	IO 输出																		
0	1	IO 输出																		
1	0	PWM2 输出																		
1	1	PWM2 取反输出																		
0	PWM1IOUT	<table border="1"> <thead> <tr> <th>T1IOUT</th> <th>PWM1IOUT</th> <th>PT4.1 输出控制，仅当 PT4.1 配置为输出有效</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>IO 输出</td></tr> <tr><td>0</td><td>1</td><td>IO 输出</td></tr> <tr><td>1</td><td>0</td><td>PWM2 输出</td></tr> <tr><td>1</td><td>1</td><td>PWM2 取反输出</td></tr> </tbody> </table>	T1IOUT	PWM1IOUT	PT4.1 输出控制，仅当 PT4.1 配置为输出有效	0	0	IO 输出	0	1	IO 输出	1	0	PWM2 输出	1	1	PWM2 取反输出			
T1IOUT	PWM1IOUT	PT4.1 输出控制，仅当 PT4.1 配置为输出有效																		
0	0	IO 输出																		
0	1	IO 输出																		
1	0	PWM2 输出																		
1	1	PWM2 取反输出																		

2.3.5 UARTCLK

UARTCLK 用于 UART 模块。UARTCLK 的时钟源来自于内部时钟，分频系数为 52 或 8。



图 2-10 UART 时钟分频示意图

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
7BH	SCON2	SMOD	URATCLKS							00uuuuuu

SCON2 寄存器

位地址	标识符	功能
7	SMOD	波特率选择寄存器
6	UARTCLKS	UARTCLK 时钟分频选择 =0: UARTCLK 为内置时钟 ICLK 的 52 分频; =1: UARTCLK 为内置时钟 ICLK 的 8 分频;

注：波特率选 57.6KHz/115.2KHz 时，UARTCLKS=1；波特率选择 9.6KHz/19.2KHz 时，UARTCLKS=0

2.3.6 蜂鸣器时钟（仅 CSU18MB86-SSOP24 支持）

表 2-12 蜂鸣器时钟寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
15h	PCK						S_BEEP[1:0]			u000000u
27H	PT1CON	BZEN				E1M[1:0]		E0M[1:0]		0uuu0000

CSU18MB86 系列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。

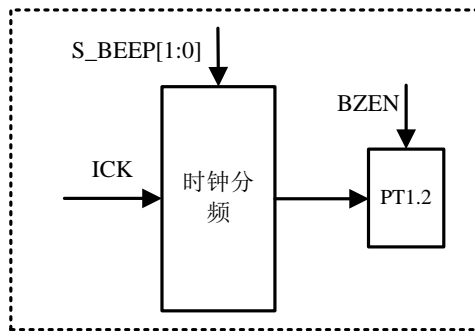


图 2-11 蜂鸣器时钟选择示意图

表 2-13 蜂鸣器时钟选择列表

S_BEEP		时钟源(KHz)		BEEP CLOCK(KHz)	
0	0	ICK	16000	ICK/1024	16
0	1	ICK	16000	ICK/2048	8
1	0	ICK	16000	ICK/4096	4
1	1	ICK	16000	ICK/8192	2

备注：上表蜂鸣器的时钟是参考 ICK=16MHz 来计算的，对于 ICK=14.7456MHz 可依比例计算获得。

2.3.7 24bit-ADCCLK

CSU18MB86 系列包含一个 24 位的 sigma delta 型的模数转换器（ADC），ADC 采样频率可选：500kHz 或 250kHz（参考 ICK=16MHz 计算值，对于 ICK=14.7456MHz 可依比例计算获得），由 ADSC 控制；增益放大倍数可选：1 倍、16 倍、64 倍、128 倍，由 ADG[1:0]控制。

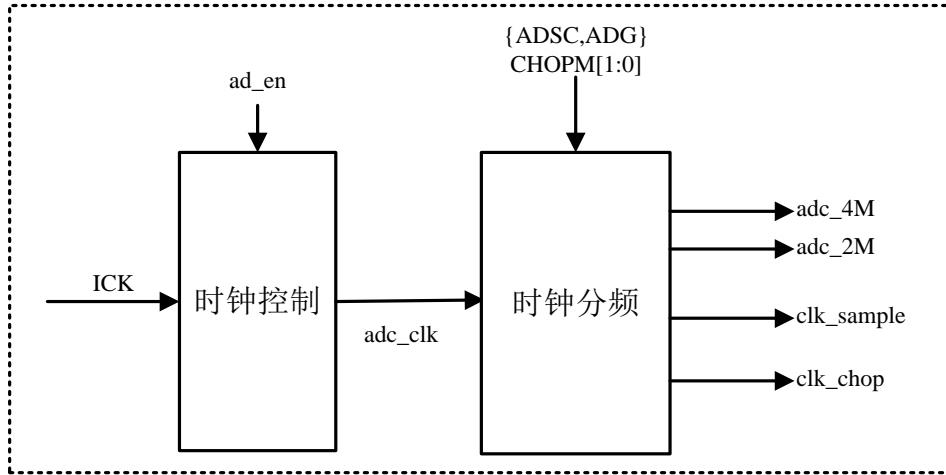


图 2-12 ADC 时钟选择

表 2-14 clk_sample 频率选择

序号	ADSC	ADG	采样频率 (clk_sample)	PGA
1	0	00	250K	1
2	0	01	250K	16
3	0	10	250K	64
4	0	11	250K	128
5	1	00	500k	1
6	1	01	500k	16
7	1	10	500k	64
8	1	11	500k	128

备注：上表 ADC 的采样频率是参考 ICK=16MHz 来计算的，对于 ICK=14.7456MHz 可依比例计算获得。

2.3.8 LEDCLK

LED 模块工作时钟通过内置时钟 ICK 分频得到，四种频率可选：500kHz、250kHz、125kHz(默认值)、62.5kHz，可以通过设置寄存器位 LEDCLKS[1:0]确定。

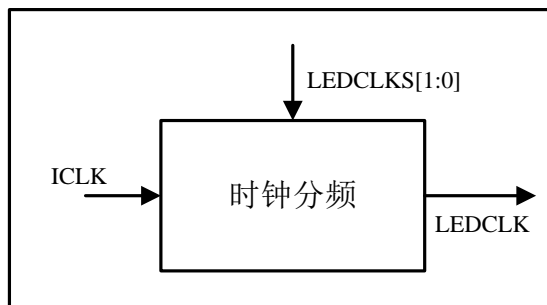


图 2-13 CSU18MB86 系列 LED 时钟产生示意图

表 2-15 LEDCLK 选择列表

时钟源(KHz)		LEDCLKS	LEDCLK(KHz)		扫描周期(Hz)
ICK	16000	=00	ICK/32	500	2232
ICK	16000	=01	ICK/64	250	1116
ICK	16000	=10	ICK/128	125	558
ICK	16000	=11	ICK/256	62.5	279

备注：上表 LEDCLKS 是参考 ICK=16MHz 来计算的，对于 ICK=14.7456MHz 可依比例计算获得。

2.4 定时器 0

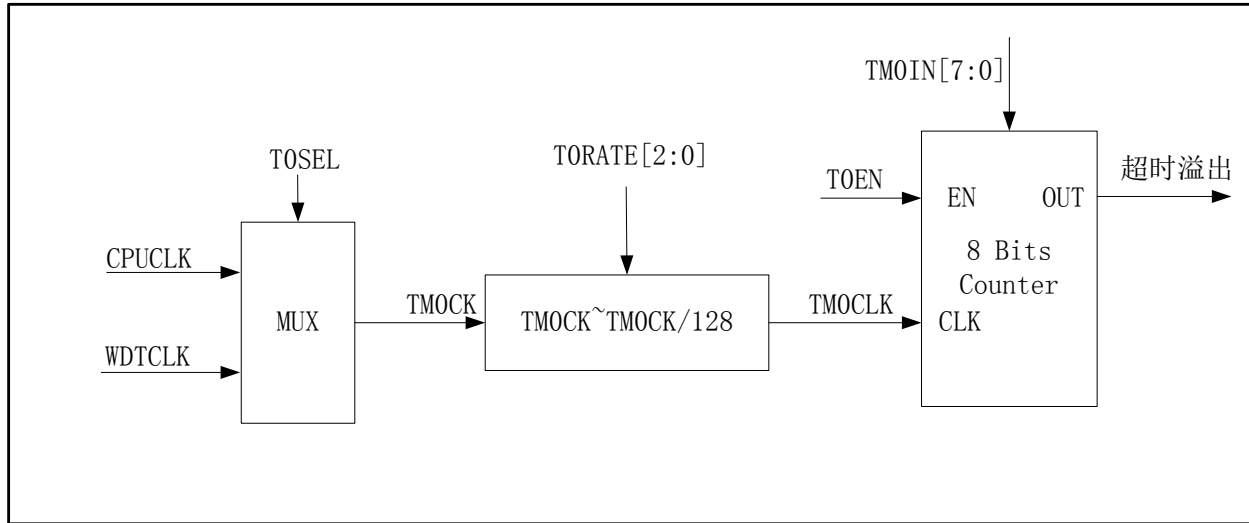


图 2-14 定时器 0 功能框图

定时器 0 的输入为 CPUCLK 或 WDTCLK。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 TM0IN。用户需要设置 TM0IN（定时器 0 模块中断信号选择器）以选择定时超时中断信号。当定时超时发生时，中断标志位会自设置，程序计数器会跳转到 0004H 以执行中断服务程序。

表 2-16 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	-	-	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	uu000000
07H	INTE	GIE	-	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	0u000000
34H	TM0CON	TOEN	TORATE[2:0]				TORSTB	TOSEL		0000u1u0
35H	TM0IN	TM0IN[7:0]								11111111
36H	TM0CNT	TM0CNT[7:0]								00000000

表 2-17 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%;"> <thead> <tr> <th>TORATE [2:0]</th> <th>TMOCLK</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>TM0CK</td> </tr> <tr> <td>001</td> <td>TM0CK /2</td> </tr> <tr> <td>010</td> <td>TM0CK /4</td> </tr> <tr> <td>011</td> <td>TM0CK /8</td> </tr> <tr> <td>100</td> <td>TM0CK /16</td> </tr> <tr> <td>101</td> <td>TM0CK /32</td> </tr> <tr> <td>110</td> <td>TM0CK /64</td> </tr> <tr> <td>111</td> <td>TM0CK /128</td> </tr> </tbody> </table>	TORATE [2:0]	TMOCLK	000	TM0CK	001	TM0CK /2	010	TM0CK /4	011	TM0CK /8	100	TM0CK /16	101	TM0CK /32	110	TM0CK /64	111	TM0CK /128
TORATE [2:0]	TMOCLK																			
000	TM0CK																			
001	TM0CK /2																			
010	TM0CK /4																			
011	TM0CK /8																			
100	TM0CK /16																			
101	TM0CK /32																			
110	TM0CK /64																			
111	TM0CK /128																			
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	NC	保留位	
0	TOSEL	时钟源选择	
		TOSEL	定时器 0 时钟源(TM0CK)
		0	CPUCLK
		1	内部 3K WDT 时钟, 仅当内部 WDT 晶振打开时有效

表 2-18 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 2-19 TMOCNT 寄存器各位功能表

位地址	标识符	功能
7 : 0	TMOCNT[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: TMOIE 与 GIE, 使能定时器 0 中断。

清零寄存器标志位: TORSTB, 复位定时器 0 的计数器。

设置寄存器标志位: TMOEN, 使能定时器 0 的 8 bits 计数器。

当定时超时发生时, 寄存器标志位 TMOIF 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.

2.5 I/O port

CSU18MB86 系列最多 15 个双向 IO 口。所有 IO 口都可以配置上拉。

其他 IO 特性:

- PT1.3 可配置为模拟口，做为 LVD 使用，模拟功能由 AIENB1 控制。
- PT2.2 可配置为模拟口，做为 10bit-ADC 输入信号使用，模拟功能由 AIENB2 控制。
- PT4.0/4.1 可配置为模拟口，做为外部 37628Hz 晶振口，模拟功能由 AIENB3 控制。
- PT1.0、PT1.2、PT2.0 为可配置为外部中断 0，PT1.1、PT1.3、PT2.1、PT4.1 可配置为外部中断 1 输入，外部中断触发方式可配置
- PT2.0~2.7 可以配置为大电流输出模式

IO 其他功能复用见相关功能模块描述

表 2-20 I/O 口寄存器表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
20H	PT1	-	-	-	-	PT1[3:0]			uuuuxxxx	
21H	PT1EN	-	-	-	-	PT1EN[3:0]			uuuu0000	
22H	PT1PU	-	-	-	-	PT1UP[3:0]			uuuu0000	
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1			1uu10uuu	
24H	PT2	PT2[7:0]							xxxxxxx	
25H	PT2EN	PT2EN[7:0]							00000000	
26H	PT2PU	PT2PU[7:0]							00000000	
27H	PT4	PT4[2:0]							uuuuuxxx	
28H	PT4EN	PT4EN[2:0]							uuuuu000	
29H	PT4PU	PT4PU[2:0]							uuuuu000	
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]	0uuu0000	
2EH	PT2CON	PT2CON[7:0]							00000000	
2FH	PTINT	PTW1[3:0]						PTW0[2:0]		0000u000

微控制器中的普通用途 I/O 口（GPIO）用于普通的用途的输入与输出功能。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。CSU18MB86 系列的部分 GPIO 可以被定义为其它的特殊功能。在本节，只说明 GPIO 的普通用途 I/O 口功能，特殊功能将会在接下来的章节中说明。

注意：所有读 IO 的操作均是对 PT 口的状态进行读取，而不是读 PT 寄存器的值

PT1 寄存器（地址为 20H）

特性	U-0	U-0	U-0	U-0	R/W-X	R/W-X	R/W-X	R/W-X
PT1	-	-	-	-	PT1[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3-0 PT1[3:0]: GPIO1 口数据标志

PT1[3] = GPIO1 bit 3 数据标志位

PT1[2] = GPIO1 bit 2 数据标志位

PT1[1] = GPIO1 bit 1 数据标志位

PT1[0] = GPIO1 bit 0 数据标志位

PT1EN 寄存器（地址为 21H）

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1EN	-	-	-	-	PT1EN[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1EN[3:0]: GPIO1 口输入/输出控制标志，默认为输入口

PT1EN[3] = GPIO1 bit 3 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口

PT1EN[2] = GPIO1 bit 2 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口

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

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

PT1PU 寄存器（地址为 22H）

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1UP	-	-	-	-	PT1UP[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PU[3:0]: GPIO1 口上拉电阻使能标志，默认断开上拉电阻

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

AENB 寄存器（地址为 23H）

特性	R/W-1	U-0	U-0	R/W-1	R/W-0	U-0	U-0	U-0
AIENB	AIENB3	-	-	AIENB2	AIENB1	-		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 AIENB3: PT4.0/4.1 数模接口定义，默认为数字口

0 = PT4.0/4.1 定义为模拟接口

1 = PT4.0/4.1 定义为数字接口

Bit 4 AIENB2: 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 3 AIENB1: PT1.3 数模接口定义，默认为模拟口

0 = PT1.3 定义为模拟接口

1 = PT1.3 定义为数字接口

PT2 寄存器（地址为 24H）

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 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
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 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
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

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[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 2-0 PT4[2:0]: GPIO4 口数据标志位

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	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
PT4EN	-					PT4EN[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 2-0 PT4EN[2:0]: GPIO4 口输入/输出控制标志, 默认为输入口

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	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
PT4PU	-					PT4PU[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 2-0 PT4PU[2:0]: GPIO4 口上拉电阻使能标志, 默认断开上拉电阻
 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

PT1CON 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
PT1CON	BZEN	-	-	-	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, PT1.2 为蜂鸣器输出接口
 0 = 不使能蜂鸣器功能, PT1.2 定义普通用途 I/O 口

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

PT2CON 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
PT2CON	PT2CON[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT2CON[7:0]: PT2 口模式控制, 与 LEDEN 配合使用
 当 PT2CON[x]=0 且 LEDEN=0 时, 对应的 PT2.x 口为 IO 口, 电流驱动能力为 3mA;
 当 PT2CON[x]=0 且 LEDEN=1 时, 对应的 PT2.x 口为 LED 输出口, 电流驱动能力由 LED_Current 控制;
 当 PT2CON[x]=1 且 LEDEN=0 时, 对应的 PT2.x 口为 IO 口, 电流驱动能力为 3mA;
 当 PT2CON[x]=1 且 LEDEN=1 时, 对应的 PT2.x 口为 IO 口, 电流驱动能力为 3mA;

PTINT 寄存器 (地址为 2FH)

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTINT	PTW1[3:0]				-	PTW0[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7 PTW1[3]: PT4.1 外部中断 1 使能, 默认为 0
0 = 禁止 PT4.1 外部中断 1
1 = 使能 PT4.1 外部中断 1
- Bit 6 PTW1[2]: PT2.1 外部中断 1 使能, 默认为 0
0 = 禁止 PT2.1 外部中断 1
1 = 使能 PT2.1 外部中断 1
- Bit 5 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 4 PTW1[0]: PT1.1 外部中断 1 使能, 默认为 0
0 = 禁止 PT1.1 外部中断 1
1 = 使能 PT1.1 外部中断 1
- Bit 2 PTW0[2]: PT2.0 外部中断 0 使能, 默认为 0
0 = 禁止 PT2.0 外部中断 0
1 = 使能 PT2.0 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]: PT1.0 外部中断 0 使能, 默认为 0
0 = 禁止 PT1.0 外部中断 0
1 = 使能 PT1.0 外部中断 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

2.5.1 数字 I/O 口、URAT 接口、外部中断输入与蜂鸣器输出：PT1[2:0]

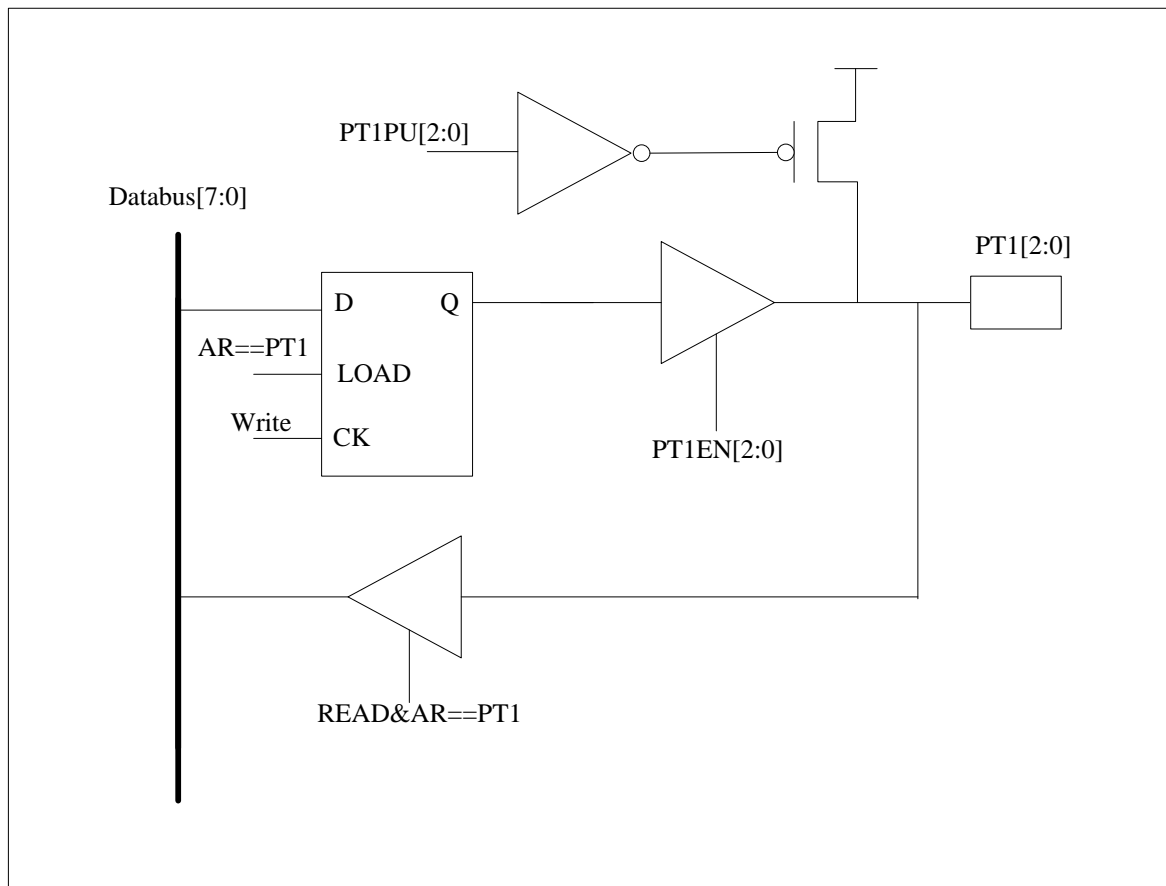


图 2-15 PT1[2:0] 功能框图

GPIO1 口（PT1[2:0]）功能框图如上图所示。GPIO 的主要功能是用于数据总线与接口之间的交换。通过控制寄存器标志 PT1EN[2:0] 以决定接口是输入或输出。输入与输出功能及相关的功能解释如下。

- 输入

GPIO1 接口 bit 2~bit 0（PT1[2:0]）可用于输入数字。当 PT1EN[n] 置为 0 时，PT1[2:0] 设置为数字输入。

PT 1[0] 和 PT 1[2] 可设置为外部中断 0 输入，PT 1[1] 可设置为外部中断 1 输入；PT 1[0] 也可设置为串口输入。

- 输出

CSU18MB86 系列通过内部 D 触发器输出数字信号。当程序通过 PT1 输出数据时，数据首先被发送到数据总线，当有写信号及 AR（CSU18MB86 系列内部器件地址指针）指向 PT1 时，然后 D 触发器会锁存数据从 PT1 口输出。

PT 1[1] 可设置为串口输出，PT 1[2] 可设置为蜂鸣器输出。

- 上拉电阻

CSU18MB86 系列在 PT1 口集成内部上拉电阻功能，上拉电阻大约为 60KΩ（上拉电流大约为 50uA。当程序要运行至睡眠模式之前，须禁止 PT1PU）。可通过控制寄存器标志 PT1PU[2:0] 决定是否连接上拉电阻。当接上拉电阻时，输入数据默认为高（即为 1）。

表 2-21 PT1 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
20H	PT1	-	-	-	-	PT1[3:0]				uuuuxxxx
21H	PT1EN	-	-	-	-	PT1EN[3:0]				uuuu0000
22H	PT1PU	-	-	-	-	PT1UP[3:0]				uuuu0000
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1				1uu10uuu
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]		0uuu0000
2EH	PT2CON	PT2CON[7:0]								00000000
2FH	PTINT	PTW1[3:0]				PTW0[2:0]				0000u000

读数据操作：

1. 清零寄存器标志位：PT1EN[n]（n 是用户要控制的 bit）。PT1 [n]被定义为输入接口。
2. 置位寄存器标志位：PT1PU[n]。PT1 [n]接口连接到一个内部上拉电阻。
3. 在信号从外部输入后，用户可以从 PT1[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT1EN[n]。PT1 [n]被定义为输出接口。
2. 置位相应的寄存器标志 PT1PU[n]。PT1 [n]连接到内部的上拉电阻。
3. 设置 PT1[n]作为数据输出，内部的 D 触发器将锁存数据直到 PT1[n]的数据改变。

外部中断操作（以下降沿触发为例子）

1. 清零寄存器标志位 PT1EN[n]。PT1[n]被定义为输入接口。
2. 置位相应的寄存器标志 PT1PU[n]。PT1[n]连接到内部的上拉电阻。
3. 置 E0M[1:0]为 00，定义 INT0 的中断触发模式为“下降沿触发”。
4. 置 E1M[1:0]为 00，定义 INT1 的中断触发模式为“下降沿触发”。
5. 置 PTW1[3:0]为 0001，定义 PT1.1 为 INT1 的中断源。
6. 置 PTW0[2:0]为 010，定义 PT1.2 为 INT0 的中断源。
7. 置 PTW0[2:0]为 001，定义 PT1.0 为 INT0 的中断源。

UART 操作（以下降沿触发为例子）

1. 置位 UARTEN，PT1.0 和 PT1.1 口就变为 UART 口。
注:CSU18MB86-SOP16 无 UART 功能，UARTEN 不能置为高。

PT1.2 蜂鸣器输出操作：

1. 置位寄存器标志位 PT1EN[2]。PT1[2]定义为输出接口。
2. 置位寄存器标志位 S_BEEP，设置蜂鸣器频率。
3. 置位寄存器标志位 BZEN。PT1[2]就作为蜂鸣器输出接口。
4. 将一个蜂鸣器与 PT1 bit2 口连接。蜂鸣器就可以正确工作。

注意操作：

1. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10KΩ），当 PT1PU[n]被置位时，可以增加输出的驱动电流。

2.5.2 带模拟输入通道的数字 I/O 口与外部中断 1 输入：PT1[3]

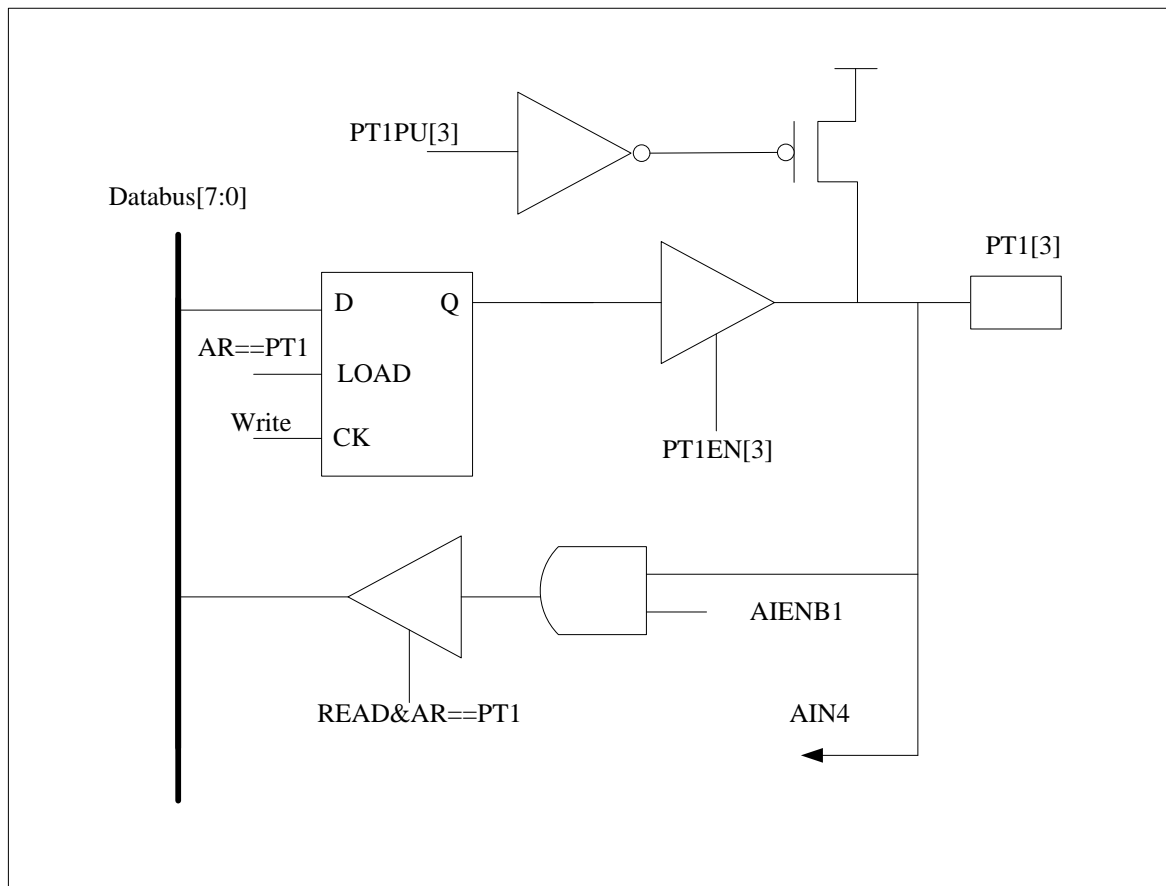


图 2-16 PT1[3] 功能框图

GPIO1 口 (PT1[3]) 功能框图如图 2-16 PT1[3] 功能框图所示。GPIO 的主要功能是用于数据总线与接口之间的交换。通过控制寄存器标志 PT1EN[4] 以决定接口是输入或输出。输入与输出功能及相关的功能解释如下。

- 输入

GPIO1 接口 bit 3 (PT1[3]) 可用于输入数字或模拟信号。用户应该控制寄存器标志 AIENB1 决定输入信号的类型。如果 AIENB1 被置位 (即为 1)，GPIO1 接口中的与门允许数字信号连接到数据总线，否则，输入信号被定义为模拟信号，模拟信号被发送到相应的功能模块

- 输出

CSU18MB86 系列通过内部 D 触发器输出数字信号。当程序通过 PT1 输出数据时，数据首先被发送到数据总线，当有写信号及 AR (CSU18MB86 系列内部器件地址指针) 指向 PT1 时，然后 D 触发器会锁存数据从 PT1 口输出。

- 上拉电阻

CSU18MB86 系列在 PT1 口集成内部上拉电阻功能，上拉电阻大约为 $60K\Omega$ (上拉电流大约为 $50\mu A$)。当程序要运行至睡眠模式之前，须禁止 PT1PU)。可通过控制寄存器标志 PT1PU[3] 决定是否连接上拉电阻。当接口接上拉电阻时，输入数据默认为高 (即为 1)。

表 2-22 PT1 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
20H	PT1	-	-	-	-	PT1[3:0]				uuuuxxxx
21H	PT1EN	-	-	-	-	PT1EN[3:0]				uuuu0000
22H	PT1PU	-	-	-	-	PT1UP[3:0]				uuuu0000
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1				1uu10uuu

读数据操作：

1. 清零寄存器标志位：PT1EN[n]（n 是用户要控制的 bit）。PT1 [n]被定义为输入接口。
2. 置位寄存器标志位：PT1PU[n]。PT1 [n]接口连接到一个内部上拉电阻。
3. 如果输入信号是数字信号，置位寄存器标志位：AIENB1。
4. 如果输入信号是模拟信号，清零寄存器标志位：AIENB1,同时将 PT1EN[3]置低（设置为数字输入），PT1UP[3]置低（没有上拉电阻）。需先使能 ENLB，模拟输入才能正常工作。
5. 在信号从外部输入后，用户可以从 PT1[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT1EN[n]。PT1 [n]被定义为输出接口。
2. 置位相应的寄存器标志 PT1PU[n]。PT1 [n]连接到内部的上拉电阻。
3. 设置 PT1[n]作为数据输出，内部的 D 触发器将锁存数据直到 PT1[n]的数据改变。

外部中断操作（以下降沿触发为例子）

1. 清零寄存器标志位 PT1EN[n]。PT1[n]被定义为输入接口。
2. 置位相应的寄存器标志 PT1PU[n]。PT1[n]连接到内部的上拉电阻。
3. 置 E1M[1:0]为 00，定义 INT1 的中断触发模式为“下降沿触发”。
4. 置 PTW1[3:0]为 0010，定义 PT1.3 为 INT1 的中断源。

注意操作：

2. 为了在睡眠模式下保持低工作电流，置位 AIENB1 使 PT1 悬空。
3. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10K Ω ），当 PT1PU[n]被置位时，可以增加输出的驱动电流，

2.5.3 数字 I/O 口、外部中断输入与 LED 驱动：PT2[1:0]

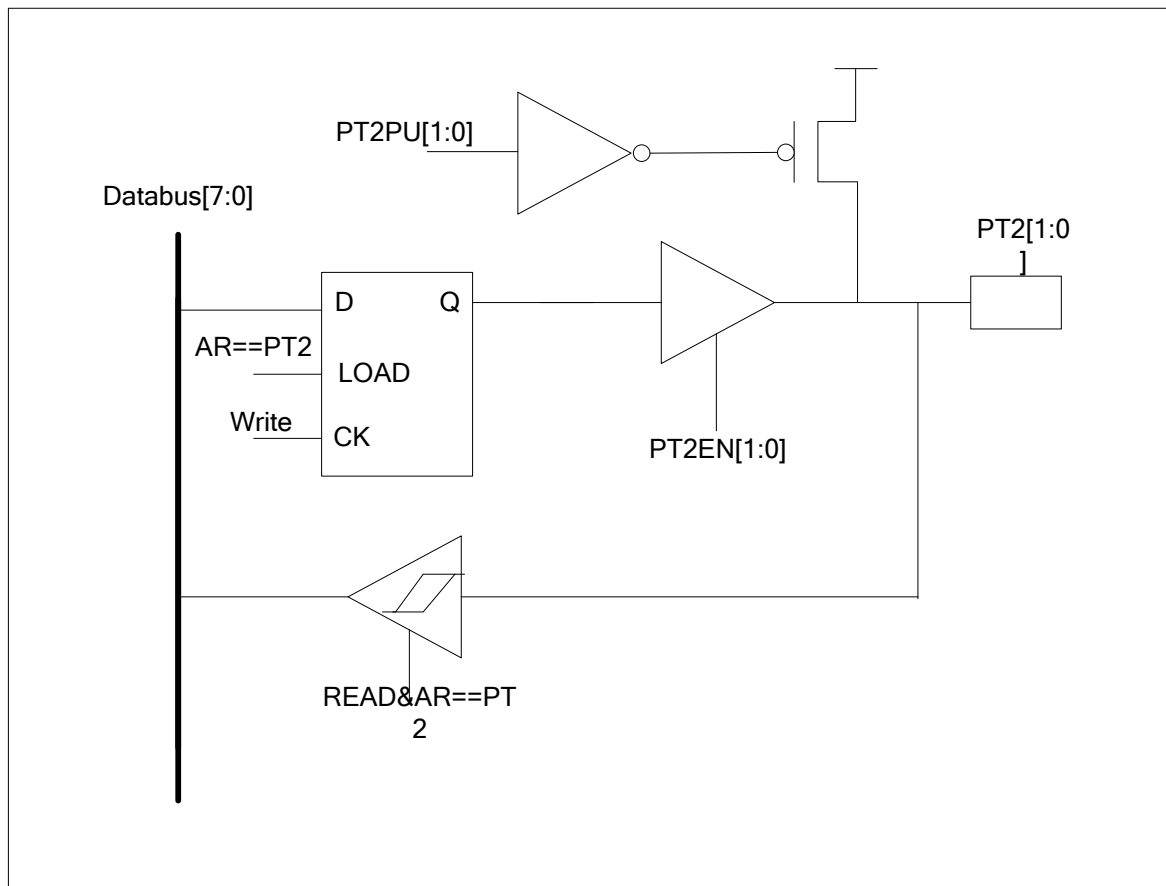


图 2-17 PT2[1:0]功能框图

GPIO2 口的 bit1~0 (PT2[1:0]) 功能框图如图 2-17 所示。此 GPIO 口的主要功能是用于数据在数据总线与端口之间的输入/输出。通过控制寄存器标志 PT2EN[1:0] 以决定接口是输入或输出。输入与输出功能及相关的功能解释如下：

- 输入

GPIO2 接口 bit 1~bit 0 (PT2[1:0]) 可用于输入数字。当 PT2EN[n] 置为 0 时，PT2[1:0] 设置为数字输入。这两个输入接口是施密特触发，上/下触发电平分别为 0.7VDD/0.3VDD。

PT 2[0] 可设置为外部中断 0 输入，PT 2[1] 可设置为外部中断 1 输入。

- 输出

CSU18MB86 系列通过内部 D 触发器输出数字信号。当程序通过 PT2 输出数据时，数据首先被发送到数据总线，当有写信号及 AR (CSU18MB86 系列内部器件地址指针) 指向 PT2 时，然后 D 触发器会锁存数据从 PT2 口输出。

当 PT2CON[n]=0 且 LEDEN=1 时，PT 2[n] 可作为 LED 驱动输出口。

- 上拉电阻

CSU18MB86 系列在 PT2 口集成内部上拉电阻功能，上拉电阻大约为 60KΩ (上拉电流大约为 50uA)。当程序要运行至睡眠模式之前，须禁止 PT2PU)。可通过控制寄存器标志 PT2PU[1:0] 决定是否连接上拉电阻。当接口接上拉电阻时，输入数据默认为高 (即为 1)。

表 2-23 PT2 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1				1uu10uuu	
24H	PT2	PT2[7:0]								xxxxxxxx	
25H	PT2EN	PT2EN[7:0]								00000000	
26H	PT2PU	PT2PU[7:0]								00000000	
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]		0uuu0000	
2EH	PT2CON	PT2CON[7:0]								00000000	
2FH	PTINT	PTW1[3:0]							PTW0[2:0]		0000u000

读数据操作：

1. 清零寄存器标志位：PT2EN[n]（n 是用户要控制的 bit）。PT2[n]被定义为输入接口。
2. 置位寄存器标志位：PT2PU[n]。PT2[n]接口连接到一个内部上拉电阻。
3. 在信号从外部输入后，用户可以从 PT2[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT2EN[n]。PT2[n]被定义为输出接口。
2. 置位相应的寄存器标志 PT2PU[n]。PT2[n]连接到内部的上拉电阻。
3. 设置 PT2[n]作为数据输出，内部的 D 触发器将锁存数据直到 PT2[n]的数据改变。

外部中断操作（以下降沿触发为例子）

1. 清零寄存器标志位 PT2EN[n]。PT2[n]被定义为输入接口。
2. 置位相应的寄存器标志 PT2PU[n]。PT2[n]连接到内部的上拉电阻。
3. 置 E0M[1:0]为 00，定义 INT0 的中断触发模式为“下降沿触发”。
4. 置 E1M[1:0]为 00，定义 INT1 的中断触发模式为“下降沿触发”。
5. 置 PTW1[3:0]为 0100，定义 PT2.1 为 INT1 的中断源。
6. 置 PTW0[2:0]为 100，定义 PT2.0 为 INT0 的中断源。

LED 驱动操作：

1. 置 PT2CON[n]为 0。
2. 置位 LEDEN。

注意操作：

1. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10K Ω ），当 PT2PU[n]被置位时，可以增加输出的驱动电流，

2.5.4 带模拟输入通道的数字 I/O 口与 LED 驱动: PT2[2]

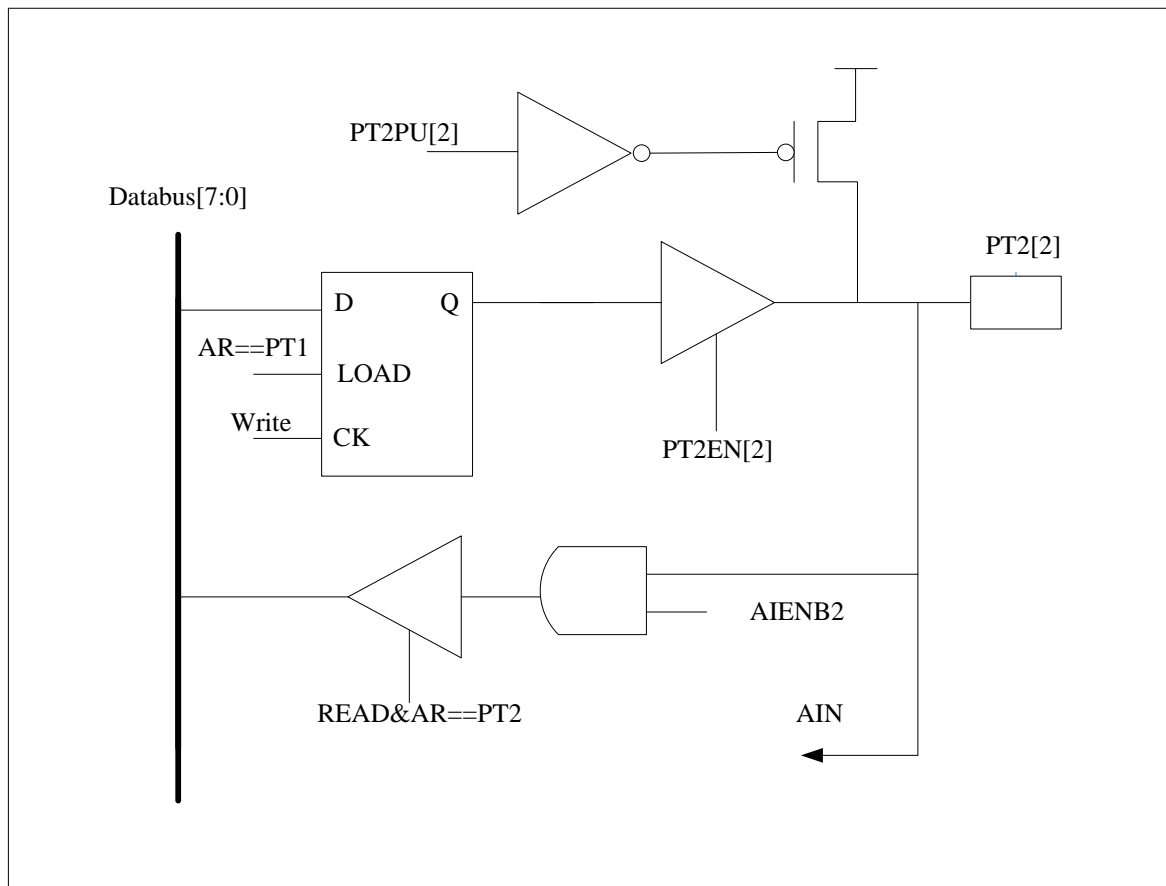


图 2-18 PT2[2] 功能框图

GPIO2 口 (PT2[2]) 功能框图如 2-18 PT2[2] 功能框图所示。GPIO 的主要功能是用于数据总线与接口之间的交换。通过控制寄存器标志 PT2EN[2] 以决定接口是输入或输出。输入与输出功能及相关的功能解释如下。

- 输入

GPIO2 接口 bit 2 (PT2[2]) 可用于输入数字或模拟信号。用户应该控制寄存器标志 AIENB1 决定输入信号的类型。如果 AIENB2 被置位 (即为 1), GPIO2 接口中的与门允许数字信号连接到数据总线, 否则, 输入信号被定义为模拟信号, 模拟信号被发送到相应的功能模块

- 输出

CSU18MB86 系列通过内部 D 触发器输出数字信号。当程序通过 PT2 输出数据时, 数据首先被发送到数据总线, 当有写信号及 AR (CSU18MB86 系列内部器件地址指针) 指向 PT2 时, 然后 D 触发器会锁存数据从 PT2 口输出。

- 上拉电阻

CSU18MB86 系列在 PT2 口集成内部上拉电阻功能, 上拉电阻大约为 $60K\Omega$ (上拉电流大约为 $50\mu A$)。当程序要运行至睡眠模式之前, 须禁止 PT2PU)。可通过控制寄存器标志 PT2PU[2] 决定是否连接上拉电阻。当接口接上拉电阻时, 输入数据默认为高 (即为 1)。

表 2-24 PT2 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1				1uu10uuu	
24H	PT2	PT2[7:0]								xxxxxxx	
25H	PT2EN	PT2EN[7:0]								00000000	
26H	PT2PU	PT2PU[7:0]								00000000	
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]		0uuu0000	
2EH	PT2CON	PT2CON[7:0]								00000000	
2FH	PTINT	PTW1[3:0]							PTW0[2:0]		0000u000

读数据操作：

1. 清零寄存器标志位：PT2EN[n]（n 是用户要控制的 bit）。PT2 [n]被定义为输入接口。
2. 置位寄存器标志位：PT2PU[n]。PT2 [n]接口连接到一个内部上拉电阻。
3. 如果输入信号是数字信号，置位寄存器标志位：AIENB2。
4. 如果输入信号是模拟信号，清零寄存器标志位：AIENB2,同时将 PT2EN[2]置低（设置为数字输入），PT2UP[2]置低（没有上拉电阻）。需先使能 AD2EN，模拟输入才能正常工作。
5. 在信号从外部输入后，用户可以从 PT2[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT2EN[n]。PT2[n]被定义为输出接口。
2. 置位相应的寄存器标志 PT2PU[n]。PT2[n]连接到内部的上拉电阻。
3. 设置 PT2[n]作为数据输出，内部的 D 触发器将锁存数据直到 PT2[n]的数据改变。

LED 驱动操作：

1. 置 PT2CON[2]为 0。
2. 置位 LEDEN。

注意操作：

1. 为了在睡眠模式下保持低工作电流，置位 AIENB2 使 PT2 悬空。
2. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10K Ω ），当 PT2PU[n]被置位时，可以增加输出的驱动电流，

2.5.5 数字 I/O 口与 LED 驱动: PT2[7:3]

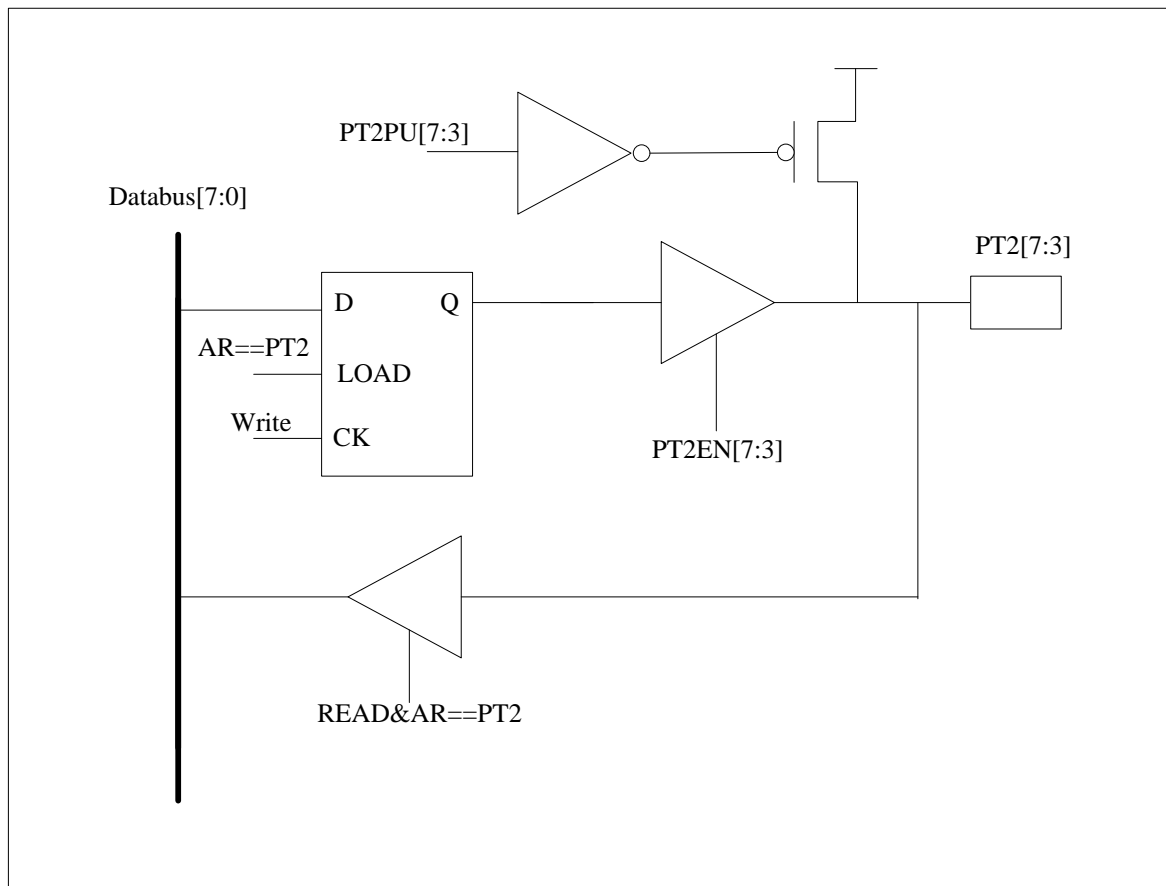


图 2-19 PT2[7:3]功能框图

GPIO2 口的 bit7~3 (PT2[7:3]) 功能框图如图 2-17 所示。此 GPIO 口的主要功能是用于数据在数据总线与端口之间的输入/输出。通过控制寄存器标志 PT2EN[7:3] 以决定接口是输入或输出。输入与输出功能及相关的功能解释如下:

- 输入

GPIO2 接口 bit7~bit3 (PT2[7:3]) 可用于输入数字。当 PT2EN[n] 置为 0 时, PT2[7:3] 设置为数字输入。

- 输出

CSU18MB86 系列通过内部 D 触发器输出数字信号。当程序通过 PT2 输出数据时, 数据首先被发送到数据总线, 当有写信号及 AR (CSU18MB86 系列内部器件地址指针) 指向 PT2 时, 然后 D 触发器会锁存数据从 PT2 口输出。

当 PT2CON[n]=0 且 LEDEN=1 时, PT 2[n] 可作为 LED 驱动输出口。

- 上拉电阻

CSU18MB86 系列在 PT2 口集成内部上拉电阻功能, 上拉电阻大约为 60KΩ (上拉电流大约为 50uA。当程序要运行至睡眠模式之前, 须禁止 PT2PU)。可通过控制寄存器标志 PT2PU[7:3] 决定是否连接上拉电阻。当接口接上拉电阻时, 输入数据默认为高 (即为 1)。

表 2-25 PT2 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1				1uu10uuu
24H	PT2	PT2[7:0]								xxxxxxx
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]		0uuu0000
2EH	PT2CON	PT2CON[7:0]								00000000
2FH	PTINT	PTW1[3:0]						PTW0[2:0]		0000u000

读数据操作：

1. 清零寄存器标志位：PT2EN[n]（n 是用户要控制的 bit）。PT2[n]被定义为输入接口。
2. 置位寄存器标志位：PT2PU[n]。PT2[n]接口连接到一个内部上拉电阻。
3. 在信号从外部输入后，用户可以从 PT2[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT2EN[n]。PT2[n]被定义为输出接口。
2. 置位相应的寄存器标志 PT2PU[n]。PT2[n]连接到内部的上拉电阻。
3. 设置 PT2[n]作为数据输出，内部的 D 触发器将锁存数据直到 PT2[n]的数据改变。

LED 驱动操作：

3. 置 PT2CON[n]为 0。
4. 置位 LEDEN。

注意操作：

2. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10K Ω ），当 PT2PU[n]被置位时，可以增加输出的驱动电流，

2.5.6 带模拟输入通道的数字 I/O 口、外部中断 1 输入与 PWM 输出：PT4[1:0]

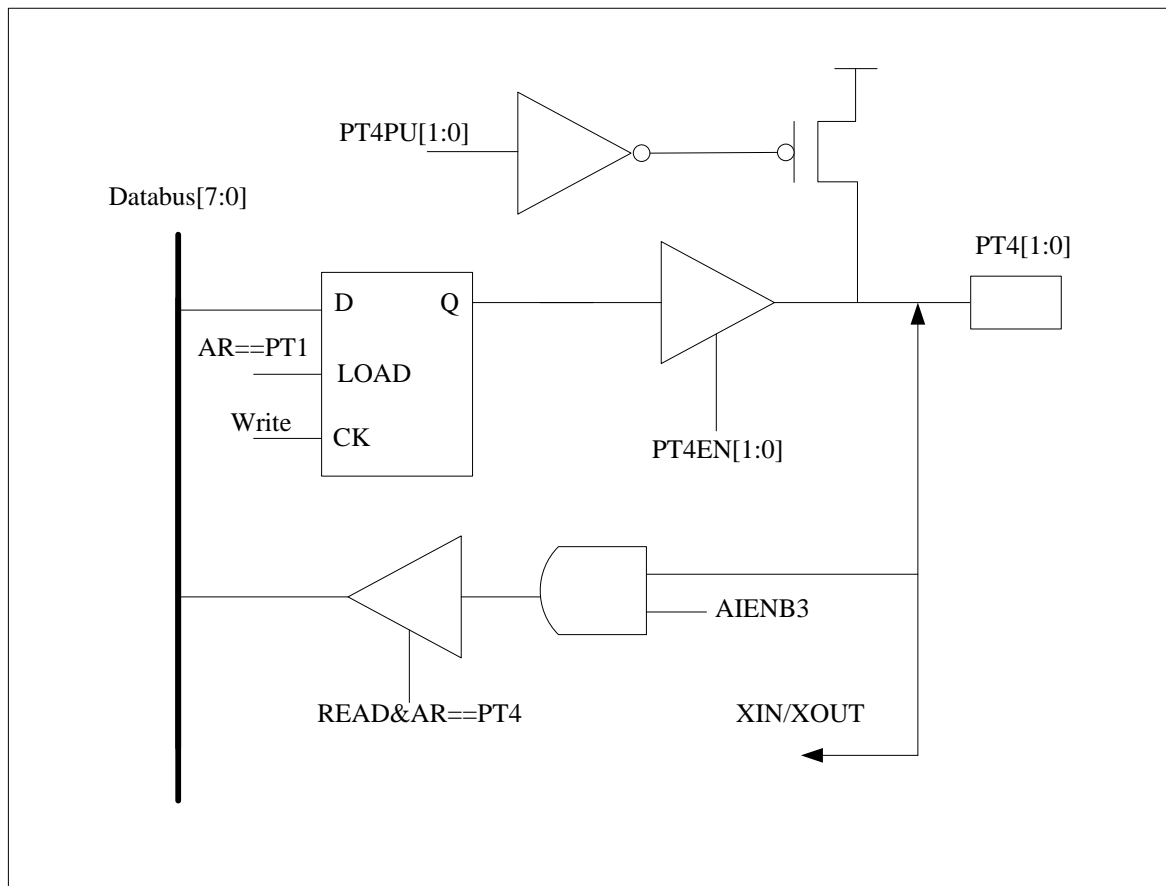


图 2-20 PT4[1:0] 功能框图

GPIO4 口（PT4[1:0]）功能框图如图 2-16 PT1[3] 功能框图所示。GPIO 的主要功能是用于数据总线与接口之间的交换。通过控制寄存器标志 PT4EN[1:0] 以决定接口是输入或输出。输入与输出功能及相关的功能解释如下。

- 输入

GPIO4 接口 bit 1~0（PT4[1:0]）可用于输入数字或模拟信号。用户应该控制寄存器标志 AIENB3 决定输入信号的类型。如果 AIENB3 被置位（即为 1），GPIO4 接口中的与门允许数字信号连接到数据总线，否则，输入信号被定义为模拟信号，模拟信号被发送到相应的功能模块。

当 AIENB3=0 时 PT4[0] 为模拟输入口；否则为数字接口。

- 输出

CSU18MB86 系列通过内部 D 触发器输出数字信号。当程序通过 PT4 输出数据时，数据首先被发送到数据总线，当有写信号及 AR（CSU18MB86 系列内部器件地址指针）指向 PT4 时，然后 D 触发器会锁存数据从 PT4 口输出。

当 AIENB3=0 时 PT4[1] 为模拟输出口；否则为数字接口。

- 上拉电阻

CSU18MB86 系列在 PT4 口集成内部上拉电阻功能，上拉电阻大约为 60KΩ（上拉电流大约为 50uA。当程序要运行至睡眠模式之前，须禁止 PT4PU）。可通过控制寄存器标志 PT4PU[1:0] 决定是否连接上拉电阻。当接口接上拉电阻时，输入数据默认为高（即为 1）。

表 2-26 PT4 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1				1uu10uuu
27H	PT4	PT4[2:0]								uuuuuxxx
28H	PT4EN	PT4EN[2:0]								uuuuu000
29H	PT4PU	PT4PU[2:0]								uuuuu000
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]		0uuu0000
2FH	PTINT	PTW1[3:0]						PTW0[2:0]		0000u000

读数据操作：

1. 清零寄存器标志位：PT4EN[n]（n 是用户要控制的 bit）。PT4 [n]被定义为输入接口。
2. 置位寄存器标志位：PT4PU[n]。PT4 [n]接口连接到一个内部上拉电阻。
3. 如果输入信号是数字信号，置位寄存器标志位：AIENB3。
4. 如果输入信号是模拟信号，清零寄存器标志位：AIENB3,同时将 PT4EN[1:0]置低（设置为数字输入），PT4UP[1:0]置低（没有上拉电阻）。
5. 在信号从外部输入后，用户可以从 PT4[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT4EN[n]。PT4 [n]被定义为输出接口。
2. 置位相应的寄存器标志 PT4PU[n]。PT4 [n]连接到内部的上拉电阻。
3. 设置 PT4[n]作为数据输出，内部的 D 触发器将锁存数据直到 PT1[n]的数据改变。

外部中断操作（以下降沿触发为例子）：

1. 清零寄存器标志位 PT4EN[n]。PT4[n]被定义为输入接口。
2. 置位相应的寄存器标志 PT4PU[n]。PT4[n]连接到内部的上拉电阻。
3. 置 E1M[1:0]为 00，定义 INT1 的中断触发模式为“下降沿触发”。
4. 置 PTW1[3:0]为 1000，定义 PT4.1 为 INT1 的中断源。

PWM 输出操作：

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 PT4.1 为输出端口，之后把 T1EN 置 1 启动定时器。
5. PWM1 从 PT4.1 输出。

注意操作：

1. 为了在睡眠模式下保持低工作电流，置位 AIENB3 使 PT4[1:0]悬空。
2. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10KΩ），当 PT1PU[n]被置位时，可以增加输出的驱动电流，

2.5.7 数字 I/O 口：PT4[2]

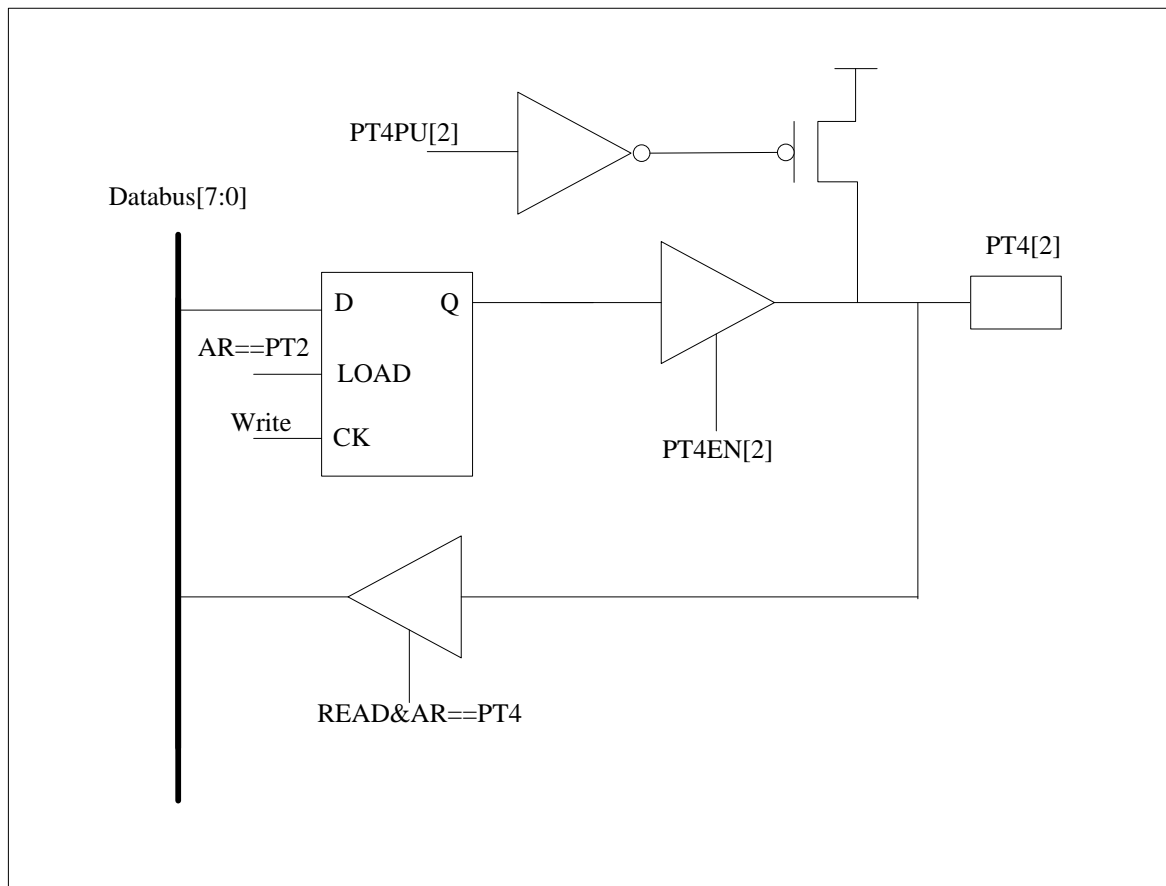


图 2-21 PT2[7:3]功能框图

GPIO4 口的 bit2 (PT4[2]) 功能框图如图 2-17 所示。此 GPIO 口的主要功能是用于数据在数据总线与端口之间的输入/输出。通过控制寄存器标志 PT4EN[2] 以决定接口是输入或输出。输入与输出功能及相关的功能解释如下：

- 输入

GPIO4 接口 bit2 (PT4[2]) 可用于输入数字。当 PT4EN[2] 置为 0 时，PT4[2] 设置为数字输入。

- 输出

CSU18MB86 系列通过内部 D 触发器输出数字信号。当程序通过 PT4 输出数据时，数据首先被发送到数据总线，当有写信号及 AR (CSU18MB86 系列内部器件地址指针) 指向 PT4 时，然后 D 触发器会锁存数据从 PT4 口输出。

- 上拉电阻

CSU18MB86 系列在 PT4 口集成内部上拉电阻功能，上拉电阻大约为 60K Ω (上拉电流大约为 50 μ A)。当程序要运行至睡眠模式之前，须禁止 PT4PU)。可通过控制寄存器标志 PT4PU[2] 决定是否连接上拉电阻。当接口接上拉电阻时，输入数据默认为高 (即为 1)。

表 2-27 PT2 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
23H	AIENB	AIENB3	-	-	AIENB2	AIENB1				1uu10uuu
27H	PT4	PT4[2:0]								uuuuuxxx
28H	PT4EN	PT4EN[2:0]								uuuuu000
29H	PT4PU	PT4PU[2:0]								uuuuu000
2DH	PT1CON	BZEN	-	-	-	E1M[1:0]		E0M[1:0]		0uuu0000
2FH	PTINT	PTW1[3:0]						PTW0[2:0]		0000u000

读数据操作：

1. 清零寄存器标志位：PT4EN[n]（n 是用户要控制的 bit）。PT4[n]被定义为输入接口。
2. 置位寄存器标志位：PT4PU[n]。PT4[n]接口连接到一个内部上拉电阻。
3. 在信号从外部输入后，用户可以从 PT4[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT4EN[n]。PT4[n]被定义为输出接口。
2. 置位相应的寄存器标志 PT4PU[n]。PT4[n]连接到内部的上拉电阻。
3. 设置 PT4[n]作为数据输出，内部的 D 触发器将锁存数据直到 PT4[n]的数据改变。

注意操作：

1. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10KΩ），当 PT4PU[n]被置位时，可以增加输出的驱动电流。

3 增强功能

3.1 电源系统

3.1.1 Regulator

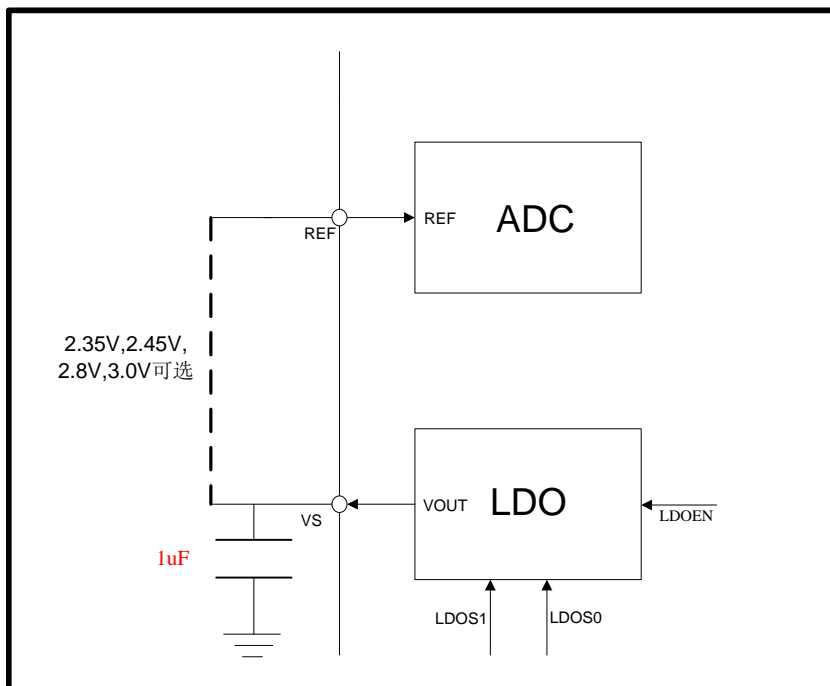


图 3-1 稳压电路

如图 3-1 所示，用于产生VS作为传感器和ADC的参考电压，通过选择LDOS可以使输出 2.35V，2.45V，2.8V，3.0V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 3-1 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
1CH	NETE	LDOS[1:0]		-	SILB[2:0]		ENLB	LB_RST_CON		00u00000
1DH	NETF	-	-	LDOEN	-		BGID[1:0]	-		u00uu01u

NETE 寄存器 (地址=1CH)

特性	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NETE	LDOS[1:0]			SILB[2:0]		ENLB	LB_RST_CON	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7~6 LDOS[1:0]: VS 电压值选择

LDOS[1:0] 00 VS=3.0V

LDOS[1:0] 01 VS=2.8V

LDOS[1:0] 10 VS=**2.45V**

LDOS[1:0] 11 VS=**2.35V**

NETF 寄存器 (地址=1DH)

特性	U-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
NETF		-	LDOEN			BGID[1:0]		-
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit5 LDOEN: LDO 使能信号
 LDOEN=1: LDO 使能
 LDOEN=0: LDO 不使能

操作:

1. 设置 LDOS[1:0], 选择 VS 值。
2. 将 LDOEN 置高。

3.1.2 低电压比较器

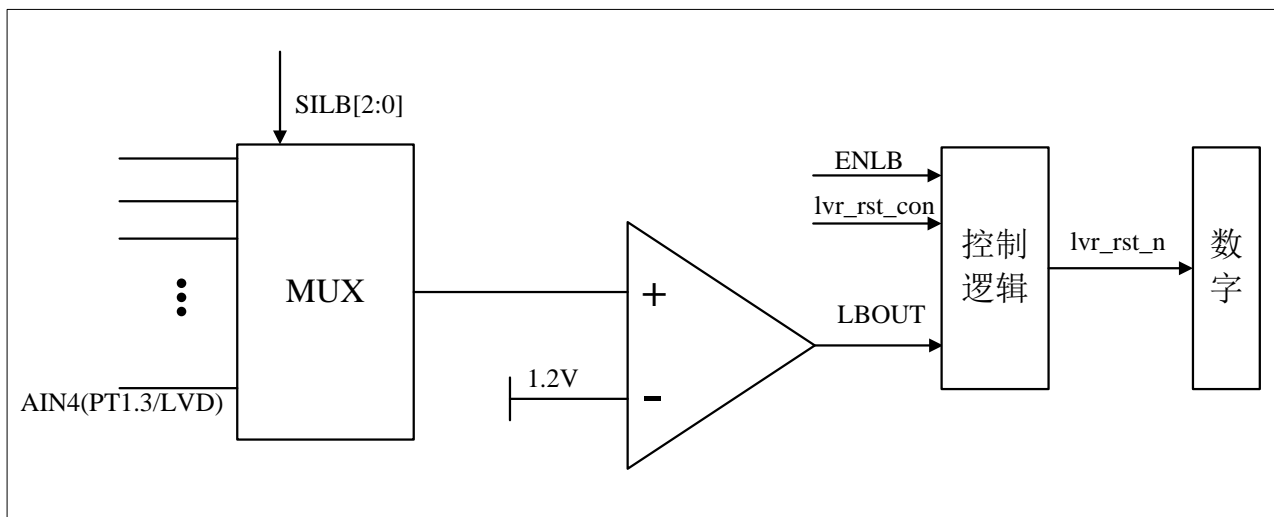


图 3-2 低电压比较功能模块框图

低电压比较器用于 DVDD 的低电压检测。CSU18MB86 系列集成一个可产生 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 ENLB, 比较器的输出是 LBOUT, LBOUT 为只读, 在 ENLB=1 且 LB_RST_CON=1 时, 芯片内部会对 LBOUT 进行判断, 如果 LBOUT=0 的时间持续超过 100us, 芯片产生复位。请看图 3-2。

表 3-2 低电压比较器寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
1CH	NETE	LDOS[1:0]				SILB[2:0]		ENLB	LB_RST_CON	00u00000
1FH	SVD								LBOUT	uuuuuuux

操作:

1. 设置寄存器标志位 ENLB, 使能低电压比较器。
2. 比较器输出是 LBOUT。

注: 在 sleep 模式下 LVD 模块不工作。

表 3-3 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>3.6V	LBOUT=1
110	AIN4	AIN4>1.2V	LBOUT=1
111	DVDD	DVDD>3.6V	LBOUT=1

NETE 寄存器 (地址=1CH)

位地址	标识符	功能																																				
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>3.6V</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>3.6V	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>3.6V	LBOUT=1																																	
110	LPD	LPD>1.2V	LBOUT=1																																			
111	DVDD	DVDD>3.6V	LBOUT=1																																			
1	ENLB	低电压检测使能信号																																				
		<table border="1"> <thead> <tr> <th>ENLB</th> <th></th> </tr> </thead> <tbody> <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> </tbody> </table>	ENLB		0	低电压检测不使能	1	低电压检测使能																														
		ENLB																																				
0	低电压检测不使能																																					
1	低电压检测使能																																					
0	LB_RST_CON	低电压复位使能控制信号，仅在 ENLB=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 ENLB=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。																																				

SVD 寄存器 (地址=1CH)

位地址	标识符	功能
0	LBOUT	低电压检测标志，只读。 参见 SILB[2:0]说明

3.1.3 Charge pump

CSU18MB86 系列内置 Charge pump，可以升压到 3.8V，用做 LED 模式下 PT2 口的驱动电压；

LED 模式下 PT2 口驱动 Charge pump

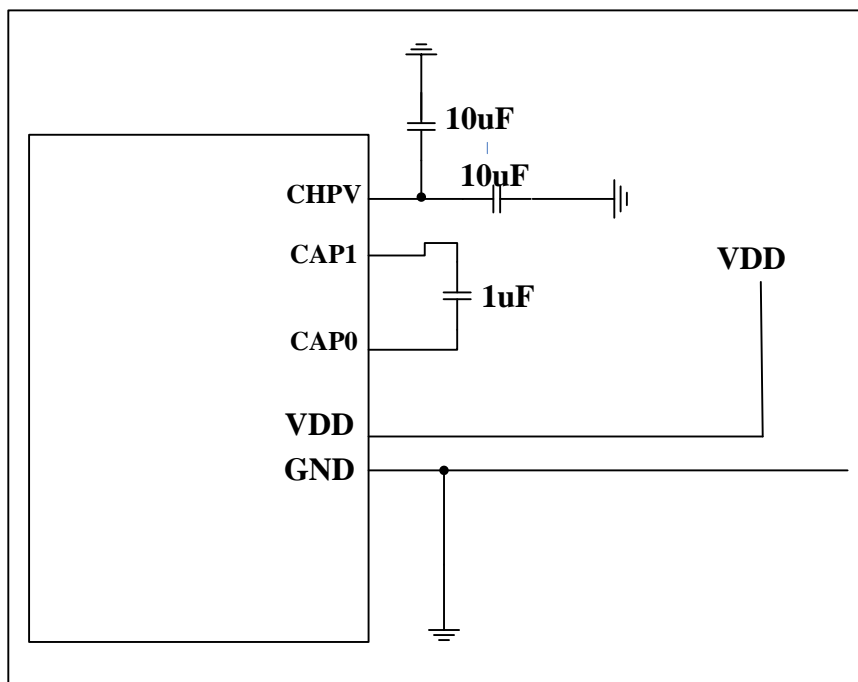


图 3-3 PT2.0 驱动电荷泵外围电路示意图

该电荷泵电路是作为 LED 模式下 PT2 口的驱动电压。当使用该电荷泵电路时，需要在 CAP1 和 CAP0 引脚处跨接入一个 1uF 的电容，CHPV 接 20uF 的电容。

CHPCON 寄存器（地址为 44H）

特性	U-0	U-0	U-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
CHPCON				CHPVS		CHPCLKS[1:0]		CHPEN

Bit4 : 3.8V charge pump 输出电压选择
 CHPVS =0: 输出电压为 3.8V(默认值)
 CHPVS =1: 输出电压为 3.6V

Bit2:1 CHPCLKS: 3.8V charge pump 时钟选择, 其频率选择如下:

CHPCLKS[1:0]	charge pump 时钟
00	500kHz
01	250kHz
10	125kHz(默认值)
11	62.5Hz

备注: 上表 charge pump 时钟是参考 ICK=16MHz 来计算的, 对于 ICK=14.7456MHz 可依比例计算获得。

Bit0 : CHPEN 3.8v charge pump 使能控制
 CHPEN =0: 3.8v charge pump 不使能
 CHPEN =1: 3.8v charge pump 使能

该电荷泵使用操作说明:

- 1 设置 CHPVS[1:0], 选择 charge pump 输出电压
- 2 设置 CHPCLKS[1:0], 选择 charge pump 工作频率
- 3 将 CHPEN 置为 1
- 4 该电荷泵的输出电压维持在 3.8V 或 3.6V

3.2 Halt 与 Sleep 模式

CSU18MB86 系列支持低功耗工作模式。为了使 CSU18MB86 系列处于待机状态，可以让 CPU 停止工作使 CSU18MB86 系列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 2uA 以下。

注：在进入 sleep 时，外接 LED 的 PT2 IO 要设置为输出且输出电平为低，其他 IO 保持固定电平

注：在进入 sleep 时，LED_PMODE=1 的 sleep 电流会比 LED_PMODE=0 时偏大 0.3uA，但在 LED_PMODE=0 时，存在 CHPV 到 DVDD 漏电流的可能性（如果 CHPV>DVDD+0.5V，就会有漏电流）

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) 定时器 0 溢出中断
- 5) 定时器 1 溢出中断
- 6) UART 发送中断
- 7) UART 接收中断
- 8) RTC 中断

睡眠模式

CPU 执行睡眠指令 (Sleep) 后，内置 16M 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。

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) WDT 中断
- 4) 定时器中断 0 (使用 WDT 时钟时)
- 5) RTC 中断

如果定时器 0 时钟源选择 WDT 时钟，需要将 WDT 时钟打开，并将定时器 0 的使能和中断使能标志打开。

3.3 复位系统

CSU18MB86 系列包括以下几种复位方式：

- 上电复位
- 掉电复位
- 低电压检测复位
- 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。

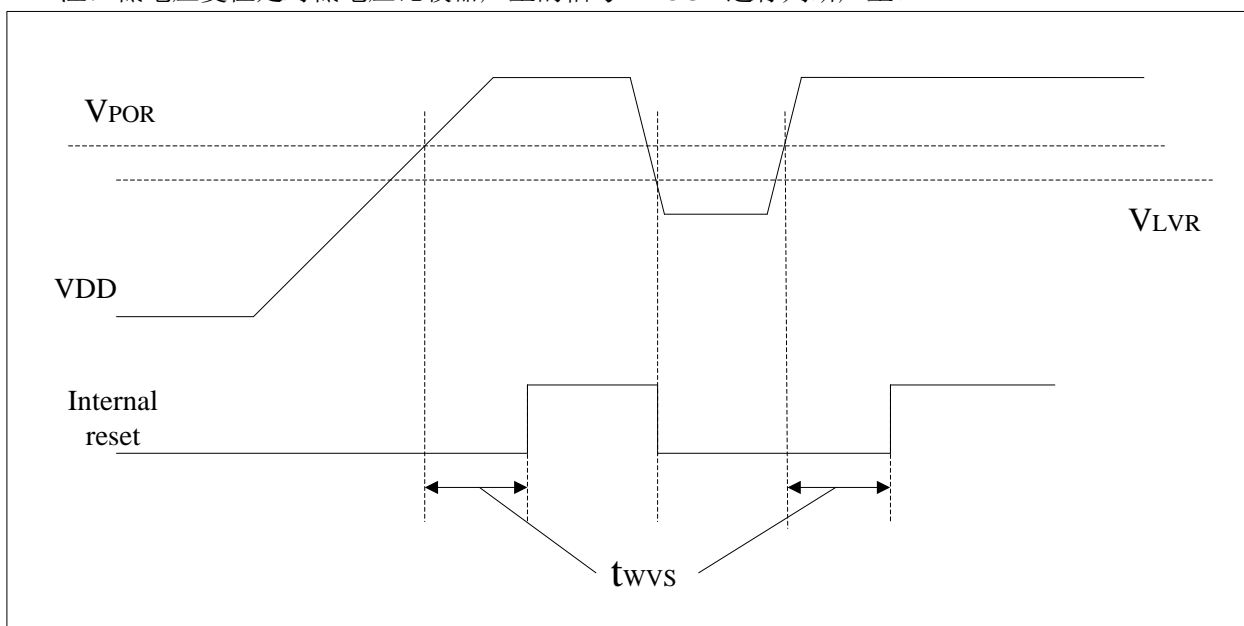


图 3-4 上电复位电路示例及上电过程

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

3.4 看门狗

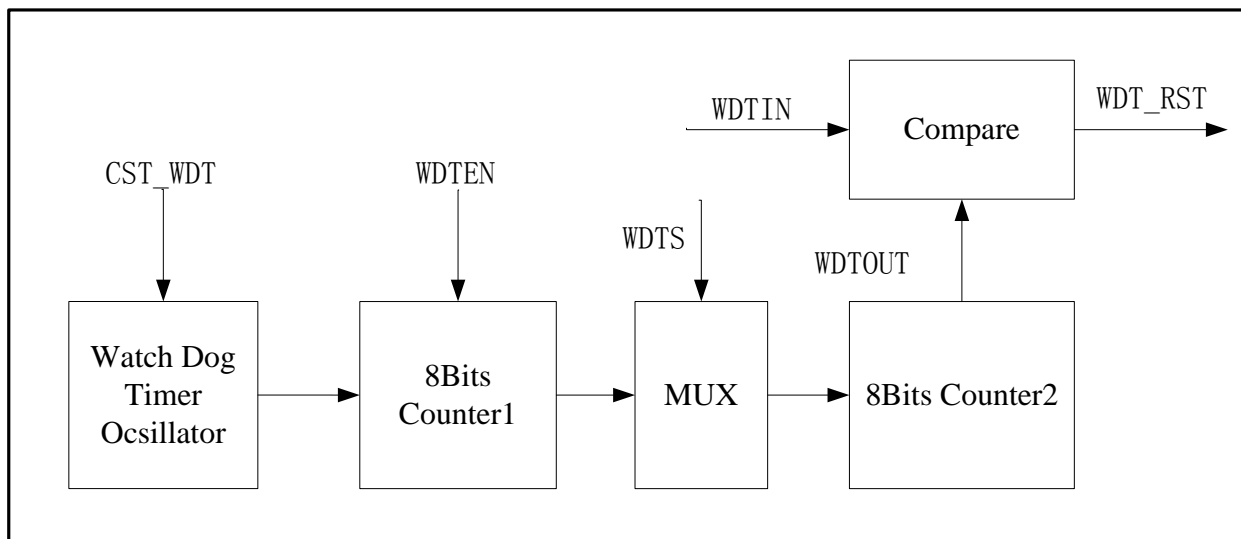


图 3-5 看门狗定时器功能框图

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。

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。

看门狗定时器寄存器表

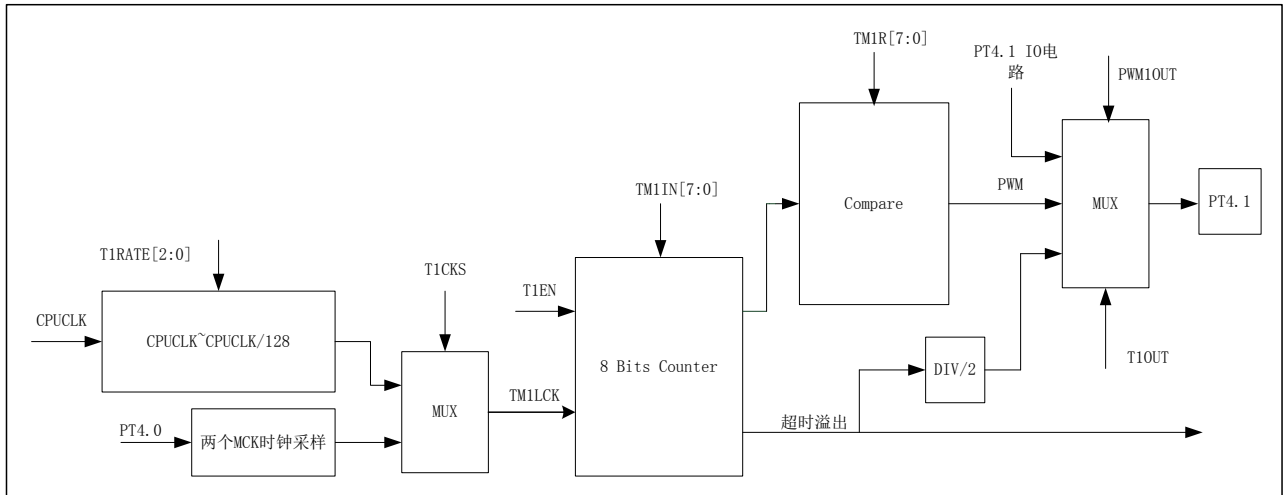
地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
04H	STATUS					TO				uuu00000	
0EH	WDTCON	WDTEN					WDT_S[2:0]			0uuuu000	
0FH	WDTIN	WDT_IN[7:0]									10111011

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

3.5 定时/计数器 1



定时/计数器 2 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；

3.5.1 寄存器说明

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	-	-	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	uu000000
07H	INTE	GIE	-	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	0u000000
37h	TM1CON	T1EN	TIRATE[2:0]			T1CKS	T1RSTB	T1OUT	PWM1OUT	00000100
38h	TM1IN	TM1IN[7:0]								11111111
39h	TM1CNT	TM1CNT[7:0]								00000000
3Ah	TM1R	TM1R[7:0]								00000000

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 <table border="1"> <thead> <tr> <th>T1RATE [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>	T1RATE [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
T1RATE [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	T1CKS	定时/计数器 1 时钟源选择位 1: PT4.0 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	PT4.1 口输出控制 <table border="1"> <thead> <tr> <th>T1OUT</th> <th>PWM1OUT</th> <th>PT4.1 输出控制, 仅当 PT4.1 配置为输出有效</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>IO 输出</td> </tr> <tr> <td>1</td> <td>0</td> <td>蜂鸣器输出</td> </tr> </tbody> </table>	T1OUT	PWM1OUT	PT4.1 输出控制, 仅当 PT4.1 配置为输出有效	0	0	IO 输出	1	0	蜂鸣器输出									
T1OUT	PWM1OUT	PT4.1 输出控制, 仅当 PT4.1 配置为输出有效																		
0	0	IO 输出																		
1	0	蜂鸣器输出																		
0	PWM1OUT	<table border="1"> <tbody> <tr> <td>0</td> <td>1</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	PWM1 输出	1	1	PWM1 取反输出												
0	1	PWM1 输出																		
1	1	PWM1 取反输出																		

TM1IN 寄存器各位功能表

位地址	标识符	功能
7: 0	TM1IN[7:0]	定时/计数器溢出值低 8 位

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 计数寄存器低 8 位, 只读

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低 8 位

3.5.2 定时器

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:

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

3.5.3 蜂鸣器(仅 CSU18MB86-SSOP24 支持)

操作:

- 1) 把 PT4.1 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。
- 7)

蜂鸣器周期计算方法:

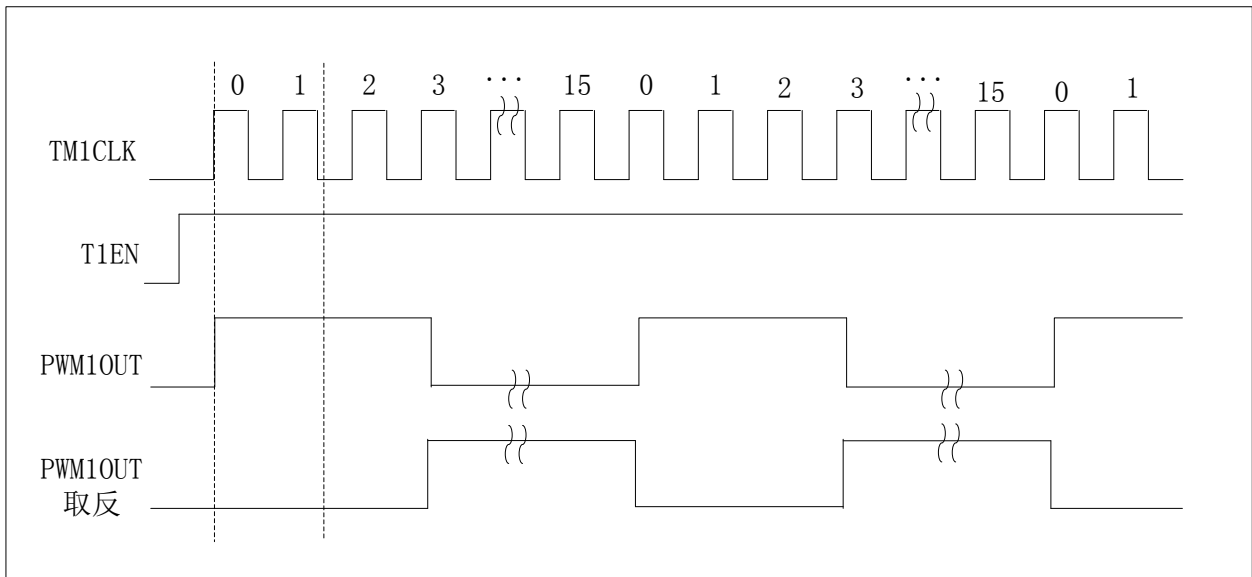
$$\text{蜂鸣器周期} = (\text{TM1IN} + 1) * 2 / \text{TM1CLK.} \quad (\text{TM1IN 不为 } 0)$$

3.5.4 PWM(仅 CSU18MB86-SSOP24 支持)

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 PT4.1 为输出口, 之后把 T1EN 置 1 启动定时器。
- 5) PWM1 从 PT4.1 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:



3.6 24bit-ADC 模块

CSU18MB86 系列包含一个 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 24Bit ADC 的各种配置参数，打开全局中断使能以及 24Bit 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。

3.6.1 24Bit ADC 寄存器说明

表 3-4 24Bit ADC 功能模块相关寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
06H	INTF	-	-	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	uu000000
07H	INTE	GIE	-	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	0u000000
10H	ADOH	ADO[23:16]								00000000
11H	ADOL	ADO[15:8]								00000000
12H	ADOLL	ADO[7:0]								00000000
13H	ADCON					ADSC	ADM[2:0]			uuuu0000
18H	NETA	SINL[1:0]			CM_SEL					0u00uuuu
1AH	NETC	CHOPM[1:0]			-	ADG[1:0]		ADEN	-	0000000u
1CH	NETE	LDOS[1:0]		-	SILB[2:0]			ENLB	Lb_rst_con	00u00000
1DH	NETF	-	-	LDOEN	-		BGID[1:0]		-	uu0uu01u
59H	TEMPC	TEMPC[7:0]								00000000

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

ADOLL 寄存器（地址为 12H）

特性	R-0	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]: 24Bit ADC 数字输出

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

ADO[22] = 24Bit ADC 数字输出数据 bit23

~

ADO[0] = 24Bit ADC 数字输出数据 bit0

ADCON 寄存器（地址为 13H）

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

Bit 3 ADSC: 24Bit ADC 采样速率选择 (参考 ICK=16MHz 计算值)
 =0, 24Bit ADC 采样速率为 250kHz;
 =1, 24Bit ADC 采样速率为 500kHz;

Bit 2-0 ADM:24Bit ADC 降采样速率选择寄存器

表 3-5 24Bit ADC 输出速率选择列表

ADM[2:0]	24Bit ADC 输出速率
000	3906Hz
001	1953Hz
010	976.5Hz
011	488Hz
100	244Hz
101	122Hz
110	61Hz
111	30.5Hz

备注: 上表 ADC 的输出速率是参考 ICK=16MHz 来计算的, 对于 ICK=14.7456MHz 可依比例计算获得。

NETA 寄存器 (地址为 18H)

特性	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
NETA	SINL[1:0]			CM_SEL				
	Bit7	Bit6		Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 SINL: 通道选择信号

00 = 24Bit ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 V_{in+}, AIN1 为 V_{in-};

01 = 内短;

10 = 24Bit ADC 输入端连接到 TEMP;

11 = 24Bit ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 V_{in-}, AIN1 为 V_{in+};

Bit 4 CM_SEL: 工作模式

0 = 工作模式 0。

1 = 工作模式 1。

NETC 寄存器 (地址为 1AH)

特性	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
NETC	CHOPM[1:0]		-		ADG[1:0]		ADEN	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7-6 : 当 ADM=000、001、010, CHOPM[1:0]写入 00; 当 ADM=其他值时, CHOPM[1:0]写入 10;

Bit 3-2 ADG[1:0]: 内部 24Bit ADC 输入增益

ADG	PGA
00	1
01	16
10	64
11	128

Bit 1 ADEN: 24Bit ADC 使能标志

1 = 24Bit ADC 使能

0 = 24Bit ADC 不使能

NETE 寄存器 (地址=1CH)

特性	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NETE	LDOS[1:0]			SILB[2:0]			ENLB	Lb_rst_con
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7~6 LDOS[1:0]: VS 电压值选择

LDOS[1:0] 00 VS=3.0V

LDOS[1:0] 01 VS=2.8V

LDOS[1:0] 10 VS=2.45V

LDOS[1:0] 11 VS=2.35V

NETF 寄存器 (地址=1DH)

特性	U-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
NETF		-	LDOEN			BGID[1:0]		-
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit5 LDOEN: LDO 电源使能信号

LDOEN=1: LDO 电源使能

LDOEN=0: LDO 电源不使能

Bit2~1 BGID[1:0]: 当 ADSC=0 时写入 01; 当 ADSC=1 时写入 10。

3.6.2 24Bit ADC 增益的温度特性调整

TEMPC 寄存器 (地址为 59H)

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

Bit7~5 TEMPC[7:5]: 温度补偿选择。

=000:0ppm

=001:45ppm@2.4V

=010:90ppm@2.4V

=011:130ppm@2.4V

=100:0ppm

=101:-45ppm@2.4V

=110:-90ppm@2.4V

=111:-130ppm@2.4V

Bit4~0 TEMPC[4:0]: 写入 0000

3.6.3 温度传感器

温度传感器有两种校准方式，校准方式描述如下：

温度传感器校准方式 1:

温度传感器精度受限于 VS 的误差，同时忽略 AD 失调温漂。实际误差为测量温度与标定温度的差乘以 VS 误差，假设测量温度与标定温度差是 20 度，实际 VS 电压值与理论 VS (2.35V、2.45V、2.8V、3.0V) 偏差 1%，则引入测量误差是 20*1%=0.2 度。适用于 VS 一致性比较好和理论偏差小的情况。

- 1、配置 SINL<1:0>=10 选择温度传感器输入 24Bit ADC
- 2、ADG=00 选择 PGA=1。
- 3、CHOPM<1:0>=10, BGID<1:0>=01, ADSC=0

4、进行一点标定（用当前环境温度进行标定，记录标定温度和标定温度 AD 值），
注意：当从 24Bit ADC 其他通道切换到温度传感器测量时，需要丢掉前三笔数据。

温度标定计算公式：

$T_{adoffset} = \text{理论上的每度码值变化值} * (273.15 + \text{标定温度}) - \text{实际标定温度 AD 值}$

$\text{温度} = (\text{实际标定温度 AD 值} + T_{adoffset}) / \text{理论上的每度码值变化值} - 273.15$

$\text{测量温度} = (\text{实际测量温度 AD 值} - \text{标定温度 AD 值}) / \text{理论上的每度码值变化值} + \text{标定温度}$
理论上的每度码值变化值为： $28886.63 / V_s$ 。 $V_s = 2.35V$ 或 $2.45V$ 或 $2.8V$ 或 $3.0V$ 。

温度传感器校准方式 2：

温度传感器精度受限于 VS 的温度漂移，同时忽略温度传感器本身的失调和忽略 AD 失调温漂。假设标定温度是 27 度，测量温度点与标定温度点的 VS 电压的温漂偏差 0.05%，则引入测量误差是 $300 * 0.05\% = 0.15$ 度。适用于 VS 离散性大与理论偏差大的情况。

1、配置 $SINL<1:0>=01$ ，选择内短输入 24Bit ADC

2、 $ADG=00$ 选择 $PGA=1$ 。

3、 $CHOPM<1:0>=10$, $BGID<1:0>=01$, $ADSC=0$

4、记录 ADC 失调电压值 $Doff$

5、配置 $SINL<1:0>=10$ ，选择温度传感器输入 24Bit ADC，其他配置维持不变。

6、进行一点标定（用当前环境温度进行标定，记录标定温度和标定温度 AD 值）

注意：当从 24Bit ADC 其他通道切换到温度传感器测量时，需要丢掉前三笔数据。

$\text{测量温度} = (\text{标定温度} + 273.15) * (\text{测量温度 AD 值} - Doff) / (\text{标定温度 AD 值} - Doff) - 273.15$

3.7 10Bit ADC 模块

CSU18MB86 系列内置一个 10bit 的 ADC，输出数据位宽为 10 位，用来测试电池电压，信号由 PT2.2 口输入，输入信号范围是 $1/3 * VDD \sim 11/12 * VDD$ 。

3.7.1 10Bit ADC 寄存器说明

表 3-6 10Bit ADC 功能模块相关寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
06H	INTF	-	-	TM1IF	TM0IF	AD2IF	ADIF	E1IF	E0IF	uu000000
07H	INTE	GIE	-	TM1IE	TM0IE	AD2IE	ADIE	E1IE	E0IE	0u000000
48H	AD2OH	AD2EN		-	-			AD2O[9:8]		0uuuuu00
49H	AD2OL	AD2O[7:0]								00000000

AD2OH 寄存器（地址为 48H）

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

Bit7 AD2EN: 10BitADC 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, 10BitADC 不使能;

=1, 10BitADC 使能;

Bit1-0 AD2O[9:8]: 10BitADC 数据的高两位。

AD2OL 寄存器（地址为 49H）

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

Bit1-0 AD2OL[7:0]: 10BitADC 数据的低八位。

10bit ADC 数据计算方式:

最低分辨率为: $LSB = \text{电源电压} * 7/12 / 2^{10}$,

输入信号计算公式: $1/3 * \text{电源电压} + AD2O[9:0] * LSB$,

例如: 电源电压=3V, AD2O[9:0]十进制值为 590,

那么: $LSB = 3V * 7/12 * 1/2^{10} = 1.709mV$, 输入信号: $1/3 * 3V + 590 * 1.709mV = 2.00831V$

3.7.2 10BitADC 的操作步骤

1. 使能相应的中断
2. 置位 AD2EN
3. 等待中断: AD2EN 置位之后, 经过 100us 之后可以产生中断。
4. 中断响应之后, 清除中断
5. 关闭 AD2EN, 读取 AD2O[9:0]

注:每次数据转换必须把 AD2EN 从 0 变 1.

3.8 LED Driver

CSU18MB86 系列有 8 个 LED 驱动口，最多可以驱动 56 个发光二极管。

连接方式如下图所示，映射逻辑参考寄存器说明。

CSU18MB86 系列的 PT2.0~PT2.7 可以使用 3.8V charge pump 驱动白光和蓝光发光二极管，驱动红光发光二极管时可以不使用 3.8V charge pump。

如下图所示，如果要点亮 LED1，需要把 PT2.0 置低、PT2.1 置高，LEDZCON[7:0] 设置为 FCH，其余 IO 被置为高阻输出，同样道理，要点亮其他 LED 时，把连接 LED 正极的 IO 置高，连接 LED 负极的 IO 置低，其余 IO 为高阻输出。

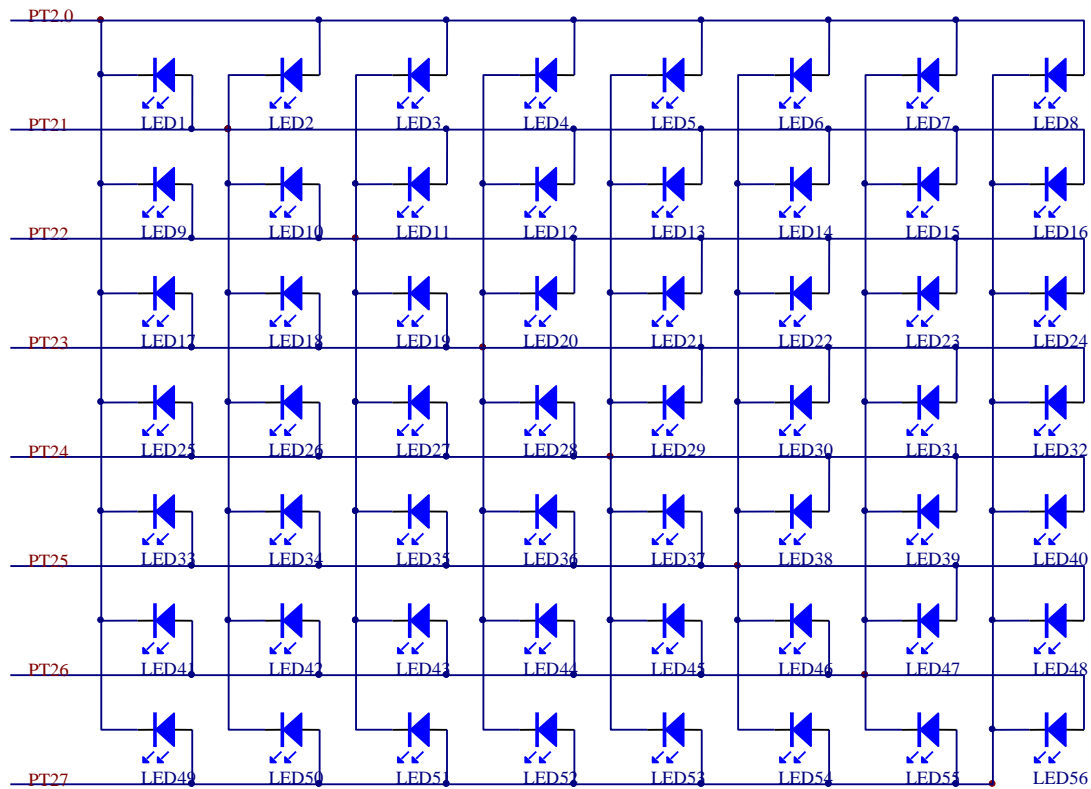


图 3-6 CSU18MB86 系列 LED 驱动连接示意图

3.8.1 寄存器说明

表 3-7 LED 驱动器寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
3BH	LED1	LED1[7:0]								00000000
3CH	LED2	LED2[7:0]								00000000
3DH	LED3	LED3[7:0]								00000000
3EH	LED4	LED4[7:0]								00000000
3FH	LED5	LED5[7:0]								00000000
40H	LED6	LED6[7:0]								00000000
41H	LED7	LED7[7:0]								00000000
42H	LEDCON	LED_CURRENT[2:0]			-	-	-	-	LED_PMODE	00000100
44H	CHPCON	-	-	-	CHPV5	-	CHPCLKS[1:0]	CHPEN	uuu0u100	
7DH	LEDEN	LEDEN[7:0]								00000000
7FH	LEDZCON	LEDZCON[7:0]								00000000

LED1 寄存器（地址为 3BH）

~

LED7 寄存器（地址为 41H）

位地址	标识符	功能
7:0	LED1~LED7	LED 驱动器控制信号

LED1	LED1[0]	LED1	P2.1 高 P2.0 低	LED4	LED4[0]	LED43	P2.6 高 P2.2 低	LED7	LED7[0]	LED49	P2.7 高 P2.0 低
	LED1[1]	LED18	P2.3 高 P2.1 低		LED4[1]	LED35	P2.5 高 P2.2 低		LED7[1]	LED50	P2.7 高 P2.1 低
	LED1[2]	LED44	P2.6 高 P2.3 低		LED4[2]	LED38	P2.4 高 P2.5 低		LED7[2]	LED51	P2.7 高 P2.2 低
	LED1[3]	LED7	P2.0 高 P2.6 低		LED4[3]	LED45	P2.6 高 P2.4 低		LED7[3]	LED52	P2.7 高 P2.3 低
	LED1[4]	LED31	P2.3 高 P2.6 低		LED4[4]	LED37	P2.5 高 P2.4 低		LED7[4]	LED53	P2.7 高 P2.4 低
	LED1[5]	LED12	P2.1 高 P2.3 低		LED4[5]	LED22	P2.2 高 P2.5 低		LED7[5]	LED54	P2.7 高 P2.5 低
	LED1[6]	LED17	P2.3 高 P2.0 低		LED4[6]	LED33	P2.5 高 P2.0 低		LED7[6]	LED55	P2.7 高 P2.6 低
	LED1[7]	LED30	P2.3 高 P2.5 低		LED4[7]	LED21	P2.2 高 P2.4 低		LED7[7]	LED42	P2.6 高 P2.1 低
LED2	LED2[0]	LED2	P2.0 高 P2.1 低	LED5	LED5[0]	LED10	P2.2 高 P2.1 低				
	LED2[1]	LED9	P2.2 高 P2.0 低		LED5[1]	LED20	P2.2 高 P2.3 低				
	LED2[2]	LED25	P2.4 高 P2.0 低		LED5[2]	LED29	P2.3 高 P2.4 低				
	LED2[3]	LED41	P2.6 高 P2.0 低		LED5[3]	LED36	P2.5 高 P2.3 低				
	LED2[4]	LED5	P2.0 高 P2.4 低		LED5[4]	LED28	P2.4 高 P2.3 低				
	LED2[5]	LED3	P2.0 高 P2.2 低		LED5[5]	LED19	P2.3 高 P2.2 低				
	LED2[6]	LED4	P2.0 高 P2.3 低		LED5[6]	LED13	P2.1 高 P2.4 低				
	LED2[7]	LED11	P2.1 高 P2.2 低		LED5[7]	LED27	P2.4 高 P2.2 低				
LED3	LED3[0]	LED23	P2.2 高 P2.6 低	LED6	LED6[0]	LED8	P2.0 高 P2.7 低				
	LED3[1]	LED34	P2.5 高 P2.1 低		LED6[1]	LED16	P2.1 高 P2.7 低				
	LED3[2]	LED47	P2.5 高 P2.6 低		LED6[2]	LED24	P2.2 高 P2.7 低				
	LED3[3]	LED39	P2.4 高 P2.6 低		LED6[3]	LED32	P2.3 高 P2.7 低				
	LED3[4]	LED46	P2.6 高 P2.5 低		LED6[4]	LED40	P2.4 高 P2.7 低				
	LED3[5]	LED14	P2.1 高 P2.5 低		LED6[5]	LED48	P2.5 高 P2.7 低				
	LED3[6]	LED6	P2.0 高 P2.5 低		LED6[6]	LED56	P2.6 高 P2.7 低				
	LED3[7]	LED26	P2.4 高 P2.1 低		LED6[7]	LED15	P2.1 高 P2.6 低				

LEDCON 寄存器（地址为 42H）

位地址	标识符	功能
7:5	LED_CURRENT[2:0]	LED 驱动电流选择 =000: 驱动电流能力为 50mA; =001: 驱动电流能力为 40mA; =010: 驱动电流能力为 30mA; =011: 驱动电流能力为 25mA; =100: 驱动电流能力为 20mA; =101: 驱动电流能力为 15mA; =110: 驱动电流能力为 10mA; =111: 驱动电流能力为 10mA;
4:1	NC	
0	LED_PMODE	LED 驱动电源控制信号 =0: VDD 作为 LED 驱动电源; (默认) =1: 3.8V pump (CHPEN=1) 或者外部输入电源 (CHPEN=0) 作为 LED 驱动电源 注: 在进入 sleep 时, LED_PMODE=1 的 sleep 电流会比 LED_PMODE=0 时偏大 0.3uA, 但在 LED_PMODE=0 时, 存在 CHPV 到 DVDD 漏电流的可能性 (如果 CHPV>DVDD+0.5V, 就会有漏电流)

CHPCON 寄存器（地址为 44H）

位地址	标识符	功能										
4	CHPVS	3.8V charge pump 输出电压选择: =0: 输出电压为 3.8V(默认值) =1: 输出电压为 3.6V										
2:1	CHPCLKS	3.8V charge pump 时钟选择, 其频率选择如下(参考 ICK=16MHz 计算值): <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CHPCLKS[1:0]</th> <th>charge pump 时钟</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>500kHz</td> </tr> <tr> <td>01</td> <td>250kHz</td> </tr> <tr> <td>10</td> <td>125kHz(默认值)</td> </tr> <tr> <td>11</td> <td>62.5Hz</td> </tr> </tbody> </table>	CHPCLKS[1:0]	charge pump 时钟	00	500kHz	01	250kHz	10	125kHz(默认值)	11	62.5Hz
CHPCLKS[1:0]	charge pump 时钟											
00	500kHz											
01	250kHz											
10	125kHz(默认值)											
11	62.5Hz											
0	CHPEN	3.8v charge pump 使能控制 =0: 3.8v charge pump 不使能 =1: 3.8v charge pump 使能										

LEDEN 寄存器（地址为 7DH）

位地址	标识符	功能
7:0	LEDEN[7:0]	LED 功能使能控制: 当 LEDEN 为 8'h26: 开启 LED 功能; 否则, 关闭 LED 功能

LEDZCON 寄存器（地址为 7FH）

位地址	标识符	功能
7:0	LEDZCON [7:0]	PT2 高阻态输出控制寄存器： LEDZCON [0]=1 时；PT2.0 输出高阻态； LEDZCON [1]=1 时；PT2.1 输出高阻态； LEDZCON [2]=1 时；PT2.2 输出高阻态； LEDZCON [3]=1 时；PT2.3 输出高阻态； LEDZCON [4]=1 时；PT2.4 输出高阻态； LEDZCON [5]=1 时；PT2.5 输出高阻态； LEDZCON [6]=1 时；PT2.6 输出高阻态； LEDZCON [7]=1 时；PT2.7 输出高阻态；

3.8.2 LED 的操作步骤

1. 将 PT2 连接到对应的 LED 正负极
2. 将 PT2 口设置为输出口
3. 设置 LED_CURRENT [1:0]选择 LED 驱动电流
4. 设置 LED_PMODE 选择 LED 驱动电源
5. 设置 LEDEN 为 26H，使能 LED 功能
6. 写数据到 PT2 和 LEDZCON 寄存器

3.9 串行通信接口(CSU18MB86-SOP16 不支持)

CSU18MB86-SSOP20 和 CSU18MB86-SSOP24 主要提供一个可编程全双工串行通信接口。该接口能同时进行数据的发送和接收，也可以作为一个同步移位寄存器使用。工作模式同通用 8051。

3.9.1 工作方式

UART 有 4 种工作方式。在通信之前用户必须先初始化 SCON1 和 SCON2，选择方式和波特率。

在所有四种方式中，任何将 SBUF 作为目标寄存器的写操作都会启动发送。在方式 0 中由条件 URRIF=0 和 REN=1 初始化接收。这会在 TX 引脚上产生一个时钟信号，然后在 RX 引脚上移 8 位数据。在其他方式中由输入的起始位初始化接收（如果 REN=1）。通过发送起始位，外部发送器开始通信。

表 3-8 串口通信工作方式

SM0	SM1	方式	类型	波特率	帧长度	起始位	停止位	第 9 位	
0	0	0	同步	cpuclk/6	8bits	无	无	无	
0	1	1	异步	UARTCLK/16 或者 32	10bits	1	1	无	
1	0	2	异步	SMOD	波特率	11bits	1	1	0,1
				0	cpuclk/32				
				1	cpuclk/16				
1	1	3	异步	UARTCLK /16 或者 32	11bits	1	1	0,1	

方式 0: 同步，半双工通信

方式 0 支持与外部设备的同步通信。在 RX 引脚上首发串行数据。TX 引脚用作发送移位时钟。CSU18MB86 系列提供 TX 引脚上的移位时钟。因此这个方式是串行通信的半双工方式。在这个方式中，每帧收发 8 位，低位先接收或发送。

功能框图如下图所示。数据通过 RX 引脚进入和移出串行端口，时钟有 TX 引脚输出，用来移位 RX 上的数据。

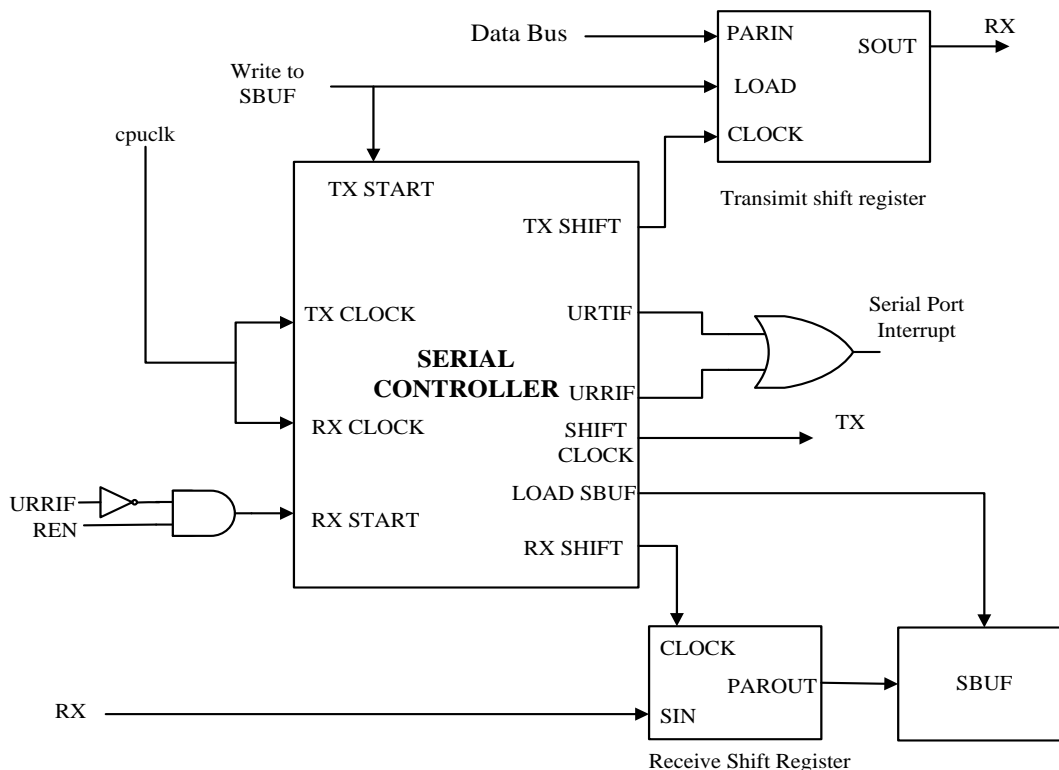


图 3-7 方式 0 功能框图

任何将 SBUF 作为目标寄存器的写操作都会启动发送。数据转换发生在移位时钟的下降沿，移位寄存器的内容逐次从左往右移位，空位置 0。当移位寄存器中的所有 8 位都发送后，TX 控制模块停止发送操作，然后在下一个系统时钟的上升沿将 URTIF 置 1，并且 RX 引脚保持高电平。



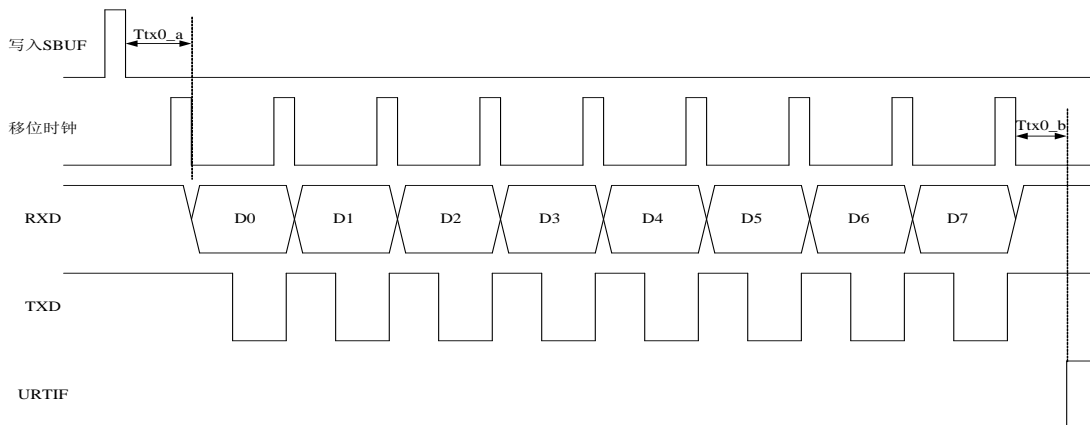


图 3-8 方式 0 发送数据波形

REN 置 1 和 URRIF 清 0 初始化接收。下一个系统时钟启动接收，在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有 8 为都接收到接收移位寄存器中后，RX 控制块停止接收，然后在下一个系统时钟的上升沿上 URRIF 置 1，知道被软件清 0 才允许接收。

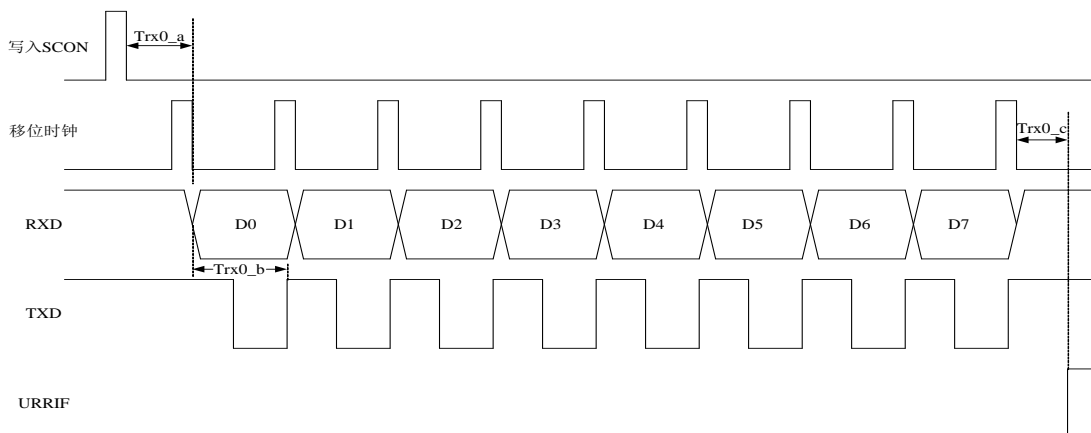


图 3-9 方式 0 接收数据波形

方式 1:8 位 UART，可变波特率，异步全双工

方式 1 提供 10 位全双工异步通信，10 位由一个起始位（逻辑 0），8 个数据位（低位为第一位），和一个停止位（逻辑 1）组成。在接收时，这 8 个数据位存储在 SBUF 中而停止位存在 RB8 中。方式 1 中的波特率是可变的，串行收发波特率可被设置为 UART 时钟的 1/16 或 1/32。

功能块框图如下图所示：

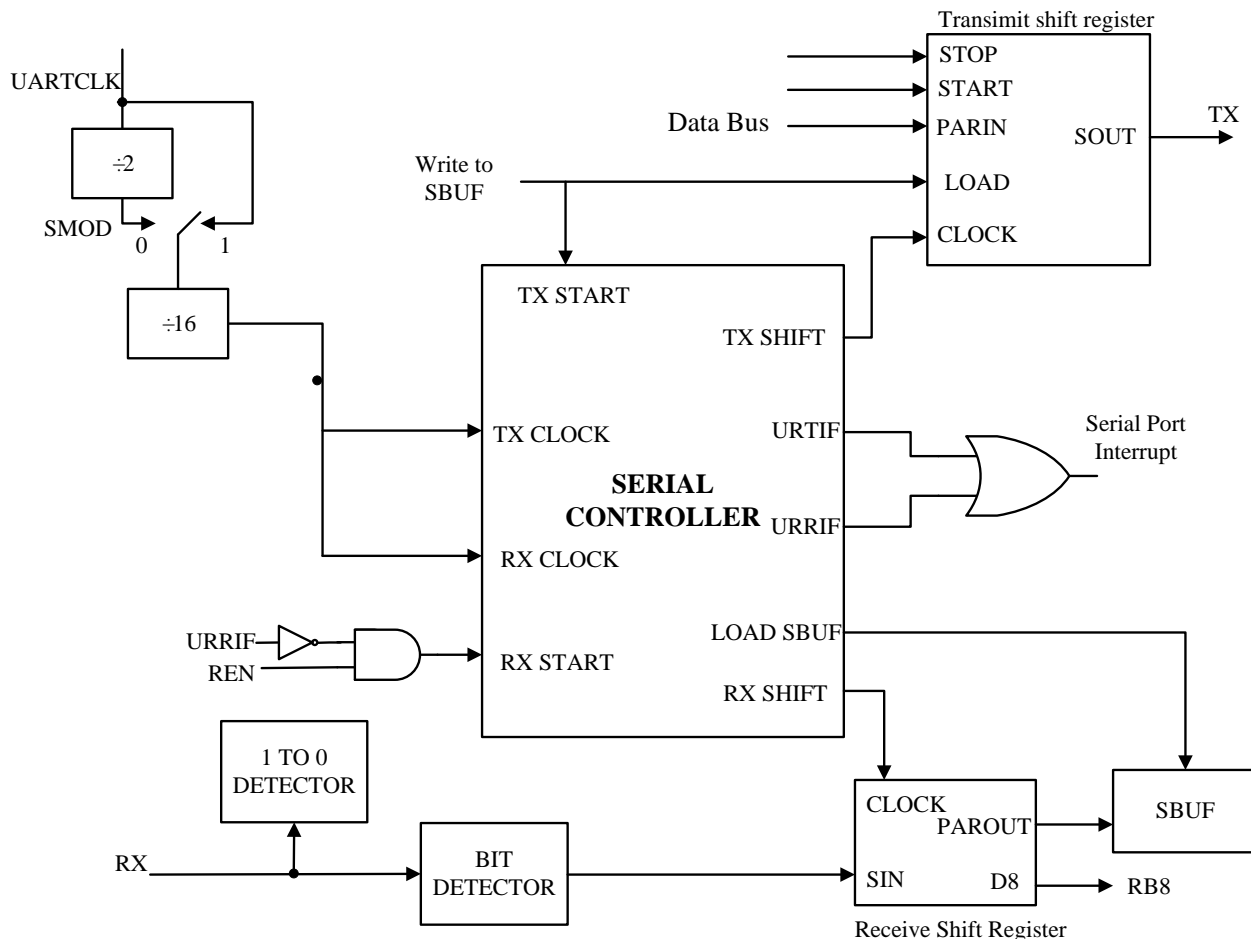


图 3-10 方式 1 功能框图

任何将 SBUF 作为目标寄存器的写操作都会启动发送，实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TX 引脚上移出，然后是 8 为数据位。在发送移位寄存器中的所有 8 为数据都发送完后，停止位在 TX 引脚上移出，在停止位发出的同时 URTIF 标志置 1。

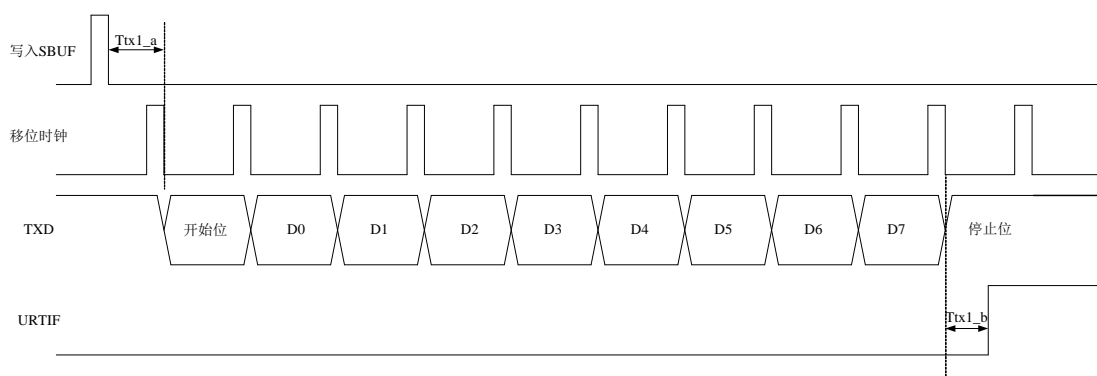


图 3-11 方式 1 发送数据波形

只有 REN 位置 1 时，才允许接收。当 RX 引脚检测到下降沿时串口开始接收串行数据。为此，MCU 对 RX 不断采样，采样速率为波特率的 16 倍。当检测下降沿时，16 分频计数器立即复位，这有助于 16 分频计数器与 RX 引脚上的串行数据位同步。16 分频计数器每一位的时间分为 16 个状态，在第 7、8、9 状态时，位检测器对 RX 端的电平进行采样。为抑制噪声，在这 3 个状态采样中至少有 2 次采样值一致数据才被接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RX 引脚上另一个下降沿的到来。若起始位有效，则移入寄存器，并接着移入其它位到移位寄存器。8 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，URRIF 置 1，但必须满足下列条件：

1. URRIF=0
2. SM2=0 或者接收的停止位=1

如果这些条件满足，那么停止位移入 RB8，8 个数据位移入 SBUF，URRIF 被置 1。否则接收的帧会丢失。这时接收器将重新去探测 RX 端是否另一个下降沿。用户必须用软件清除 URRIF，然后才能再次接收。

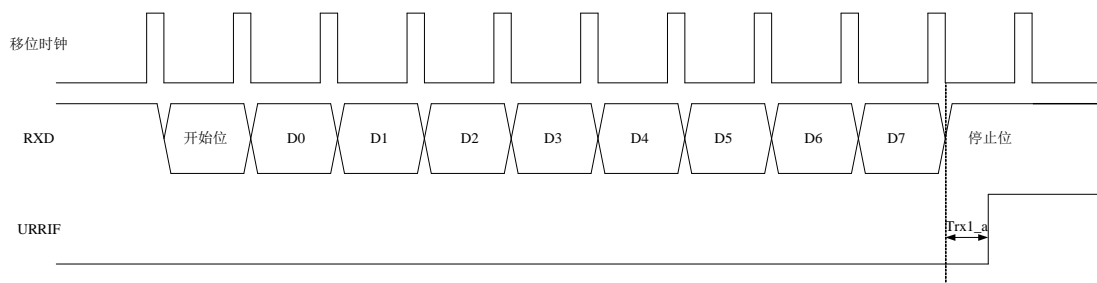


图 3-12 方式 1 接收数据波形

方式 2: 9 位 UART，固定波特率，异步全双工

这个方式使用异步全双工通信中的 11 位。一帧由一个起始位（逻辑 0），8 个数据位（低位为第一位），一个可编程的第 9 数据位和一个停止位（逻辑 1）组成。方式 2 支持多机通信（详见）。在数据传送时，第 9 位数据位（SCON 中的 TB8）可以写成 0 或 1，例如，可写入 PSW 中的奇偶位 P，或用作多级通信中的数据/地址标志位。当接收到数据时，第 9 数据位进入 RB8 而停止位不保存。PCON 中的 SMOD 位选择波特率为 CPU 工作时钟的 1/16 或 1/32。功能块框图如下所示：

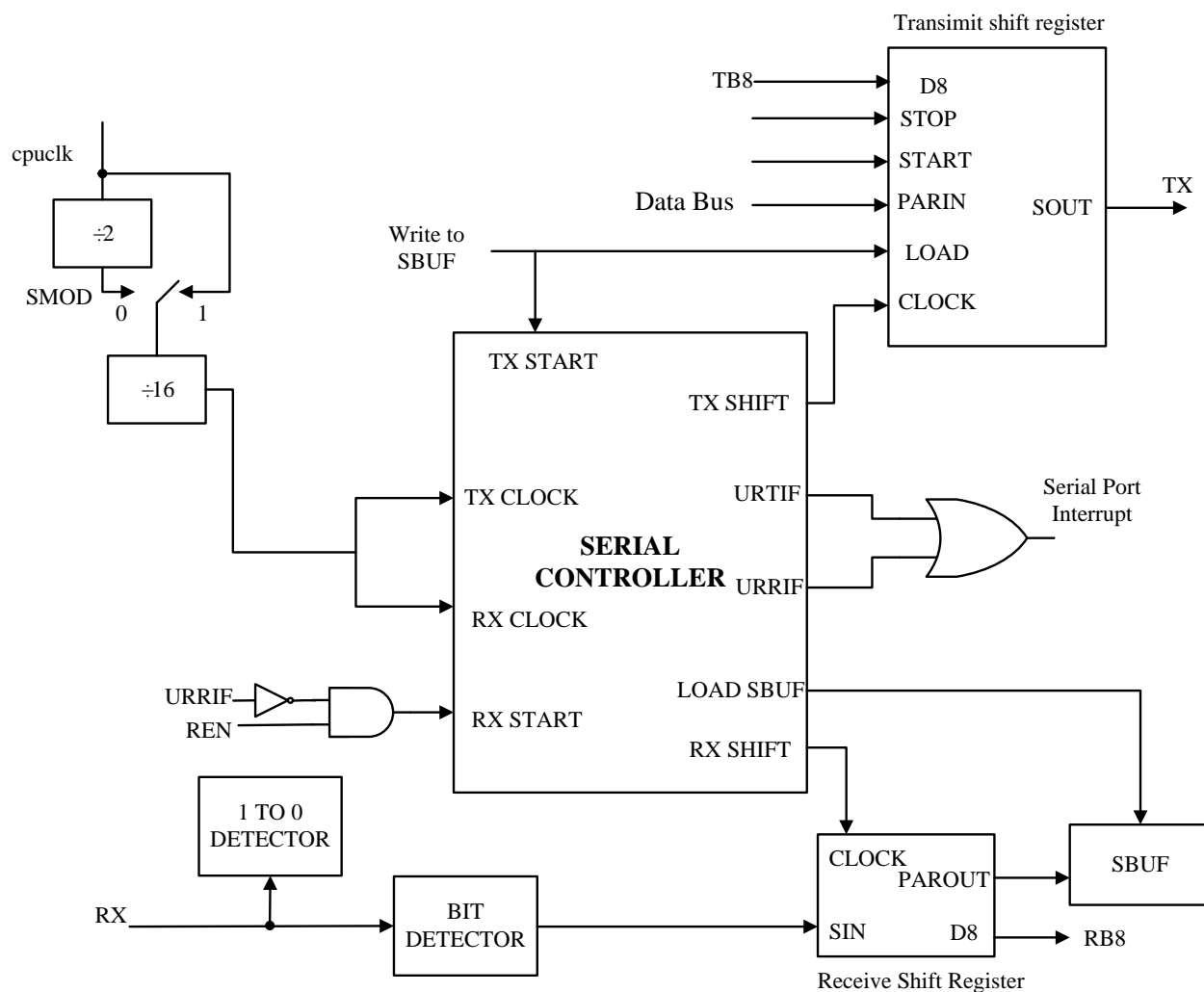


图 3-13 方式 2 功能框图

任何将 SBUF 作为目标寄存器的写操作都会启动发送，同时也将 TB8 载入到发送移位寄存器的第 9 位中。实际上发送是从 16 分频计数器的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TX 引脚上移出，然后是 9 位数据位（低位为第一位）。在发送转换寄存器中的所有 9 为数据都发送完后，停止位在 TX 引脚上移出，在停止位发送后，URTIF 标志置 1。

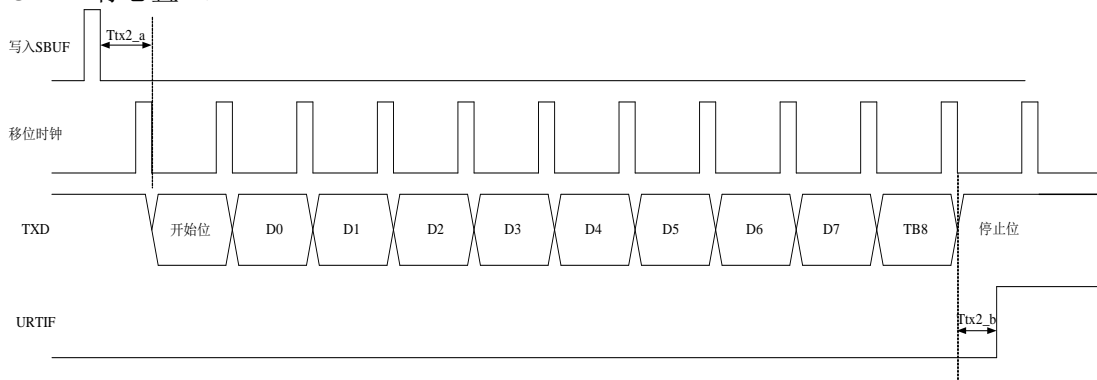


图 3-14 方式 2 发送数据波形

只有 REN 位置 1 时才允许接收。当 RX 引脚检测到下降沿时串行口开始接收串行数据。为此，CPU 对 RX 不断采样，采样速度为波特率的 16 倍。当检测下降沿时，16 位分频计数器立即复位。这有助于 16 分频计数器与 RX 引脚上的串行数据位同步。16 分频计数器把每一位的时间分为 16 个状态，在第 7、8、9 状态时，位检测器对 RX 端的电平进行采样。为抑制噪声，在这 3 个状态采样中至少有 2 次采样值一致数据才被接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，改为被忽略，接收电路被复位，等待 RX 引脚上另一个下降沿的到来。若起始位有效，则移入寄存器，并接着移入其它位到移位寄存器。9 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，URRIF 置 1，但必须满足下列条件：

1. URRIF=0
2. SM2=0 或者接收的第 9 位=1，且接收位符合 UART 地址

如果条件满足，那么第 9 位移入 RB8，8 位数据移入 SBUF，URRIF 被置 1。否则接收的数据帧会丢失。

在停止位的当中，接收器回到寻找 RX 引脚上的另一个下降沿。用户必须用软件清除 URRIF，然后才能再次接收。

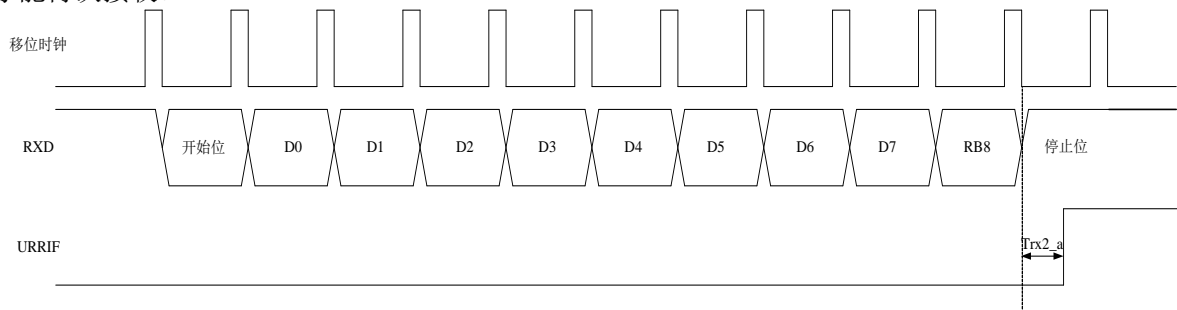


图 3-15 方式 2 接收数据波形

方式 3: 9 位 UART，可变波特率，异步全双工

方式 3 使用方式 2 的传输协议以及方式 1 的波特率产生方式

3.9.2 多机通信

软件地址识别

方式 2 和方式 3 有一个专门的适用于多机通信的功能。在这两个方式下，接收的是 8 位数据，第 9 位移入 RB8 中，然后再来一位停止位。UART 可以这样来设定：当接收到停止位时，只有 RB8=1 的条件下，串行口中断才会有效（请求标志 URRIF 置 1）。可以通过将 SCON 寄存器的 SM2 位置 1 是 UART 具有这个功能。

在多机通信系统中，以如下所述来利用这一功能。当主机要发送一数据给几个从机中的一个时，它先送出一地址字节，以辨认目标从机。地址字节与数据字节可用第 9 为数据位来区别，地址字节的第 9 位为 1，数据字节的第 9 位为 0。

如果从接 SM2 为 1，则不会响应数据字节中断。地址字节可以中断所有从机，这样，每一个从机都检查所接收到的地址字节，以辨别自己是不是目标从机。被寻址的从机 SM2 位清 0，并准备接收即将到来的数据字节，当接收完毕时，从机再一次将 SM2 置 1。没有被寻址的从机，则维持它们的 SM2 位为 1，忽略到来的数据字节，继续做自己的事情。

注意：在方式 0 中，SM2 用来选择波特率加倍。在方式 1 中，SM2 用来检测停止位是否有效，如果 SM2=1，接收中断不会响应直到接收到一个有效的停止位。

3.9.3 寄存器说明

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
32h	INTF2	-	-	-	-	RTCIF	-	URTIF	URRIF	uuuu0u00
33h	INTE2	-	-	-	-	RTCIE	-	URTIE	URRIE	uuuu0u00
7AH	SCON1	SM0	SM1	SM2	REN	TB8	RB8	-	UARTEN	00000000
7BH	SCON2	SMOD	-							0uuuuuuu
7CH	SBUF									00000000

SCON1 寄存器

位地址	标识符	功能
7:6	SM0、SM1	串口通信工作方式选择寄存器 参见表 3-8 串口通信工作方式
5	SM2	保留
4	REN	接收控制选择 1: 允许接收 0: 禁止接收
3	TB8	发送数据第 9 位
2	RB8	接收数据第 9 位 不可写
1	NC	保留
0	UARTEN	串口使能

SCON2 寄存器

位地址	标识符	功能
7	SMOD	波特率选择寄存器 参见表 3-8 串口通信工作方式

注：在 UARTEN 打开之后，为了保证 UART 通信的稳定性，SCON1 和 SCON2 寄存器配置不能更改。

SBUF 寄存器

位地址	标识符	功能
7:0	SBUF	当串口发送数据时，将发送数据写入 SBUF 寄存器。 当串口接收数据时，从 SBUF 寄存器读出接收数据。

3.9.4 波特率

波特率 (K)	ICK=16MHz/14.7456MHz, 模式 2 或模式 4		
	实际波特率 (K)	偏差(%)	SMOD
9.6	9.6153	0.16	=0
19.2	19.2307	0.16	=1
57.6	57.6	0%	=0
115.2	115.2	0%	=1

注：当使用波特率 57.6KHz 或 115.2KHz 时，必须选择 ICK=14.7456MHz。

3.10 RTC(仅 CSU18MB86-SSOP24 支持)

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 32768kHz 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

3.10.1 功能描述

由于 RTC 时钟和系统时钟的不同, 当用户对任一 RTC 寄存器进行写入时, 必须等待 2 个 RTC 时钟周期(60us), 内部 RTC 计时器的值才会被更新, 128 个指令周期之后可以读取写入的数据判断写是否正确。此外, 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。在读取地址为 62H~68H 的 RTC 时间寄存器时, 这些寄存器会被锁定 128 指令周期, 防止读取数据过程中 RTC 时间寄存器发生变化。

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

3.10.2 寄存器说明

表 3-9 RTC 寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
32h	INTF2	-	-	-	-	RTCIF	-	URTIF	URRIF	uuuu0u00
33h	INTE2	-	-	-	-	RTCIE	-	URTIE	URRIE	uuuu0u00
60H	RTCCON	LIR		-		24hr/12hr		-	RTCEN	0uuu0uu0
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	-		10HOUR10(0~2)		1HOUR(0~9)				uu000001
66H	RTCMIN	-		10MIN(0~5)		1MIN(0~9)				u0000000
67H	RTCSEC	-		10SEC(0~5)		1SEC(0~9)				u0000000
68H	RTCDWR	-				-	DWR[2:0] (0~6)			uuuuu000
69H	INTEGER					INTEGER[3:0]				uuuu0111
6AH	FRACTION					FRACTION[5:0]				uu000000

INTF2 寄存器

位地址	标识符	功能
3	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

INTE2 寄存器

位地址	标识符	功能
3	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
[3]	24hr/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
[0]	RTCEN	RTC 使能控制 1=RTC 功能使能 0=RTC 功能禁止

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
[5:4]	10HR	10HourTimeDigit (0~2)
[3:0]	1HR	1HourTimeDigit (0~9)

RTCMIN 寄存器

位地址	标识符	功能
[6:4]	10MIN	10MinTimeDigit (0~5)
[3:0]	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> <tr><td>5</td><td>星期五</td></tr> <tr><td>6</td><td>星期六</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	5	星期五	6	星期六
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.

INTEGER 寄存器

Bits	描述												
[7:4]	- 保留												
[3:0]	RTC 时钟补偿整数部分, 每秒钟补偿一次, 默认值 0111. <table border="1" style="margin-left: 20px;"> <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> </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
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	

RACTION 寄存器

Bits	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。

3.10.3 RTC 的操作步骤

1. 将 PT4.0/4.1 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

3.11 MTP 模块

MTP 烧写器的接口:

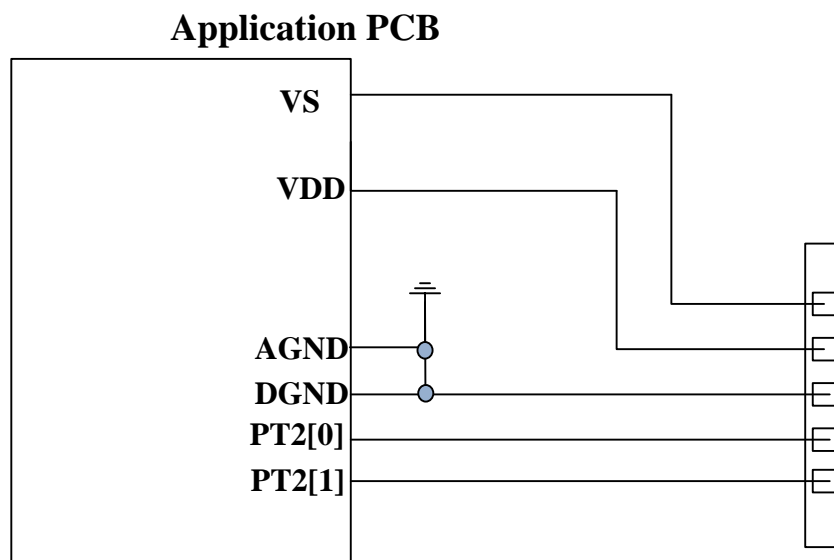


图 3-16 MTP 烧写器接口图

表 3-10 MTP 接口说明

端口名称	说明	备注
VS	内置 LDO 电压输出	用来进行 AD 校准
VDD	芯片电源输入	
AGND	芯片模拟地	
DGND	芯片数字地	
PT2[0]	数据端口	
PT2[1]	数据端口	

- 注意事项: 1. 建议使用 1To4 铁壳烧录器（必须接 VS 且将 AGND 与 DGND 相连，接线如上图所示）。
 2. 若使用 1To1 铁壳烧录器(慎用)，烧录时将不会进行 ADC 校验（此时只需接 PT20、PT21、DVDD、DGND 四线）。
 3. 未进行 ADC 校验可能会导致温度部分工作异常。

3.12 MTP 在线烧录

电路要求：在线烧录时，使用芯片自带的升压电路。

地址要求：CSU18MB86 系列 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)；

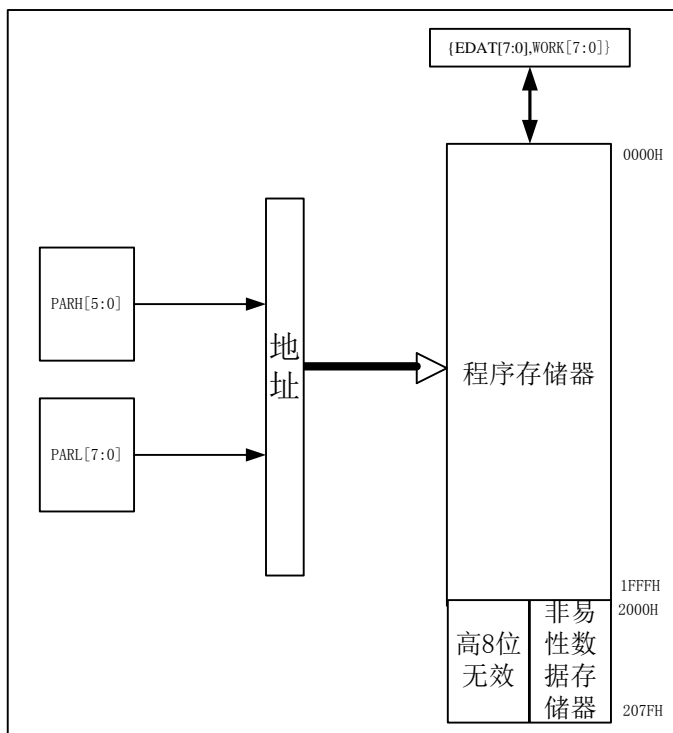


图 3-17 在线烧录地址寻址示意图

表 3-11 在线烧录寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
05H	WORK	工作寄存器								00000000
0AH	EADRH	PARH[5:0]								uu000000
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。

操作方式：

在线烧录时：

1. 使用 CLRWDT 指令复位 WDT；

2. 对 EOPEN 写相应的数据;
 3. 将烧录地址写入 EADRH, EADRL 寄存器;
 4. 高八位数据写入 EDAT 寄存器;
 5. 对 EOPEN 写入相应的数据;
 6. 将烧录数据低八位数据写入 work 寄存器;
 7. 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。

4 MCU 指令集

表 4-1 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
ADDFPCW	$[PC] \leftarrow [PC] + 1 + [W]$	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
MOVFW 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	$(W) \leftarrow (W) + K$
标志位	C, DC, Z
描述	工作寄存器的内容加上立即数 K 结果保存到工作寄存器中
周期	1
例子 ADDLW 08H	在指令执行之前: W=08H 在指令执行之后: W=10H

2

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

3

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

4

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

5

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

6

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

7

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

8

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

9

BTFSC	如果 bit 测试为 0 则跳转
指令格式	BTFSC f, b 0<=f<=1FFH 0<=b<=7
操作	Skip if (f[b])=0
标志位	无
描述	如果 f 的 bit 位是 0，下一条取到的指令将被丢到，然后执行一条空指令组成一个两周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期
例子 NODE BTFSC FLAG 2 OP1: OP2:	在程序执行以前 PC=address(NODE) 指令执行之后 If(FLAG[2])=0 PC=address(OP2) If(FLAG[2])=1 PC=address(OP1)

10

BTFSS	如果 bit 测试为 1，则跳转
指令格式	BTFSS f, b 0<=f<=1FFH 0<=b<=7
操作	Skip if (f[b])=1
标志位	无
描述	如果 f 的 bit 位是 1，下一条取到的指令将被丢到，然后执行一条空指令组成一个两周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期
例子 NODE BTFSS FLAG 2 OP1: OP2:	在程序执行以前 PC=address(NODE) 指令执行之后 If(FLAG[2])=0 PC=address(OP1) If(FLAG[2])=1 PC=address(OP2)

11

CALL	子程序调用
指令格式	CALL K 0<=K<=1FFFFH
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 COMF f, 1	在指令执行之前 W=88H, f=23H 在指令执行之后 W=88H, f=DCH

15

DAW	十进制调整 W 寄存器
指令格式	DAW
操作	十进制调整 W 寄存器
标志位	C,DC
描述	一般与加法一起使用。 如果低半字节的值大于 9 或 DC 为 1 时，低半字节加 6； 如果高半字节的值大于 9 或 C 为 1 时，高半字节加 6
周期	1
例子 若 W=25； ADDLW 39 DAW	在 DAW 指令执行之前 W=25+39 =64=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 $0 \leq K \leq 1FFFH$
操作	$PC \leftarrow K$
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 $0 \leq f \leq 1FFFH$ d=0,1
操作	(目的地址) $\leftarrow (f)+1$
标志位	Z
描述	f 加 1 如果 d 为 0，结果保存到工作寄存器中 如果 d 为 1，结果保存到 f 中。
周期	1
例子 INCF f, 0	在指令执行之前 W=88H f=23H 在指令执行之后 W=24H f=23H
例子 2 INCF f, 1	在指令执行之前 W=88H f=23H 在指令执行之后 W=88H f=24H

21

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

22

IORLW	工作寄存器与立即数或
指令格式	IORLW K 0<=K<=FFH
操作	(W)<-(W) K
标志位	Z
描述	立即数与工作寄存器的内容或。结果保存到工作寄存器中。
周期	1
例子 IORLW 85H	在指令执行之前 W=69H 在指令执行之后 W=EDH

23

IORWF	f 与工作寄存器或
指令格式	IORWF f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<-(W) (f)
标志位	Z
描述	f 和工作寄存器或 当 d 为 0 时, 结果保存到工作寄存器中 当 d 为 1 时, 结果保存到 f 中
周期	1
例子 IORWF f,1	在指令执行前 W=88H f=23H 在指令执行后 W=88H f=ABH

24

MOVFW	传送到工作寄存器
指令格式	MOVFW f 0<=f<=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 带进位左移一位 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 RLF f, 1	在指令执行之前 C=0 W=88H f=E6H 在指令执行之后 C=1 W=88H f=CCH

35

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

RRF f, 0	C=0 W=88H f=95H 在指令执行之后 C=1 W=4AH f=95H
----------	---

36

SLEEP	晶振停止
指令格式	SLEEP
操作	CPU 晶振停止
标志位	PD
描述	CPU 晶振停止。CPU 通过外部中断源重启
周期	1

37

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

38

SUBWF	f 的值减工作寄存器的值
指令格式	SUBWF f, d 0<=f<=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	在指令执行之前

SWAPF f,1	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	$MTP(\{EADRH,EADRL\}) \leftarrow (\{WORK\})$
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	$(W) \leftarrow (W) \wedge K$
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	$(\text{目标地址}) \leftarrow (W) \wedge (f)$
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 最大极限值

表 5-1 CSU18MB86 系列最大极限值

参数	范围	单位
电源 VDD	2.4~3.6	V
引脚输入电压	-0.3~VDD+0.3	V
工作温度	-40~+85	℃
存储温度	-55~+150	℃

5.2 直流特性

(VDD = 3.3V, T_A = 25℃, 如无其他说明则都是此条件)

符号	参数	测试条件	最小值	典型值	最大值	单位
VDD	工作电源		2.4	3	3.6	V
IDD1	电源电流 1	指令周期 = 2MHz		1.2		mA
ISleep	睡眠模式下电源电流	睡眠指令		2		uA
VIH	数字输入高电平	PT2.0\ PT2.1	0.7*V _{DD}			V
VIL	数字输入低电平	PT2.0\ PT2.1			0.3*VDD	V
VIH	数字输入高电平	PT1、PT2(除 PT2.0\ PT2.1)、PT4	0.6*V _{DD}			V
VIL	数字输入低电平	PT1、PT2(除 PT2.0\ PT2.1)、PT4			0.4*VDD	V
IPU	上拉电流	PT1、PT2、PT4		50		uA
IOH	高电平输出电流	VOH=0.9*VDD (除 PT1.2、PT2)		3		mA
IOL	低电平输出电流	VOL=0.1*VDD (除 PT1.2、PT2)		-3		mA
IOH	高电平输出电流	VOH=0.9*VDD 仅 PT1.2		9		mA
IOL	低电平输出电流	VOL=0.1*VDD 仅 PT1.2		-9		mA
IOH	高电平输出电流	PT2: LED 模式		50/40/30/25 /20/15/10		mA
				3: 非 LED 模式		mA
IOL	低电平输出电流	PT2: LED 模式		50/40/30/25 /20/15/10		mA
				3: 非 LED 模式		mA
IREG	VS 稳压器输出电流	VS=2.35V		4	8	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	T _{AB} = -40~80℃		50		ppm/°C
VLBAT	低电池检测电压	SILB[2:0]=000	2.35	2.4	2.45	V
		SILB[2:0]=001	2.45	2.5	2.55	
		SILB[2:0]=010	2.55	2.6	2.65	
		SILB[2:0]=011	2.65	2.7	2.75	
		SILB[2:0]=100	2.75	2.8	2.85	
		SILB[2:0]=101	3.55	3.6	3.65	
		SILB[2:0]=110	1.15	1.2	1.25	

		SILB[2:0]=111	3.55	3.6	3.65	
FRC	内置 RC 振荡器			16		MHz
FWDT	内置看门狗时钟			3		KHz

5.3 24Bit ADC 的特性

表 5-2 Sigma Delta ADC 性能指标

(Vref = 2.35V, 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	
模拟输入	共模输入电压		GND	VREF/2	VREF	V	
	满幅输入电压 (AIN+)-(AIN-)				±VREF/PGA	V	
	差分输入阻抗			8/PGA		MΩ	
系统性能	无失码			24 (带符号位)		Bits	
	有效精度	PGA=64,30Hz		17.5(不带符号位)		Bits	
		PGA=1,30Hz		19(不带符号位)		Bits	
	输入噪声 (rms)	增益=1			4.48		uv
		增益=64			198		nV
	积分线性度	增益=64			±0.003		% of 0.7*FS
	失调误差	增益=64			5		uV
失调误差漂移	增益=64			-0.03		uV/°C	
增益误差	增益=64			-8		%	
参考电压	VS	LDOS[1:0]= 11	2.25	2.35	2.45	V	
		LDOS[1:0]= 10	2.35	2.45	2.55		
		LDOS[1:0]= 01	2.7	2.8	2.9		
		LDOS[1:0]= 00		3.0			
	参考电压温度系数					±100	ppm/°C

6 封装

CSU18MB86 有三种封装形式，分别为 SOP16L、SSOP20 和 SSOP24，对应的产品名称为 CSU18MB86-SOP16、CSU18MB86-SSOP20 和 CSU18MB86-SSOP24。

6.1 SOP16L 封装尺寸

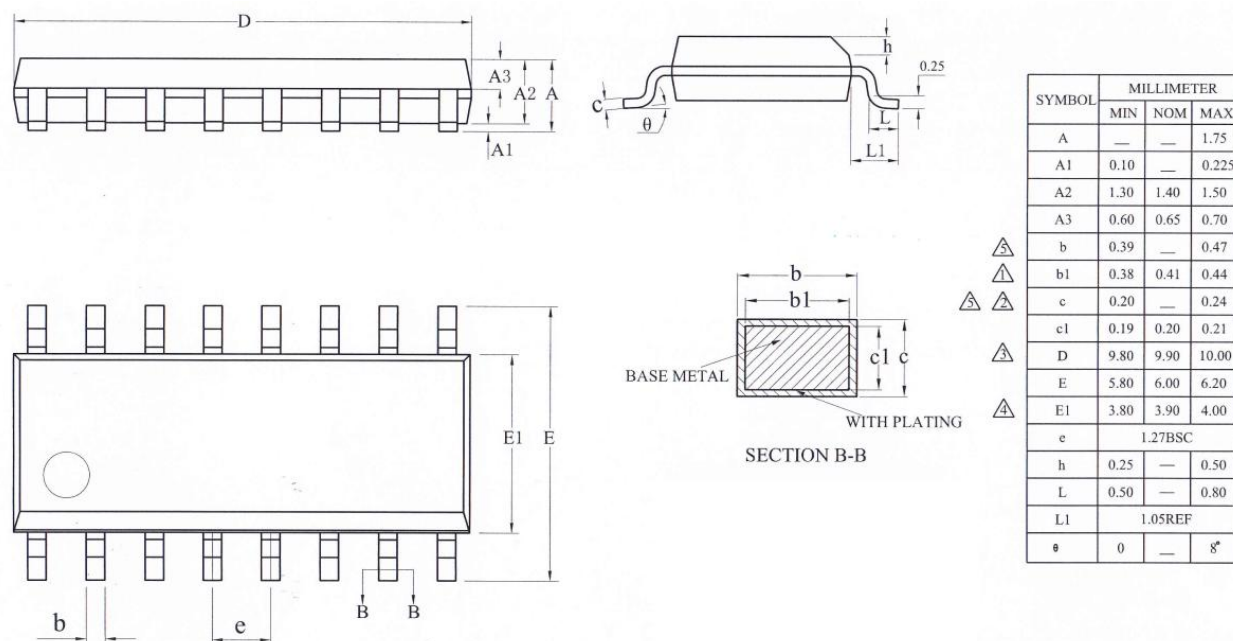
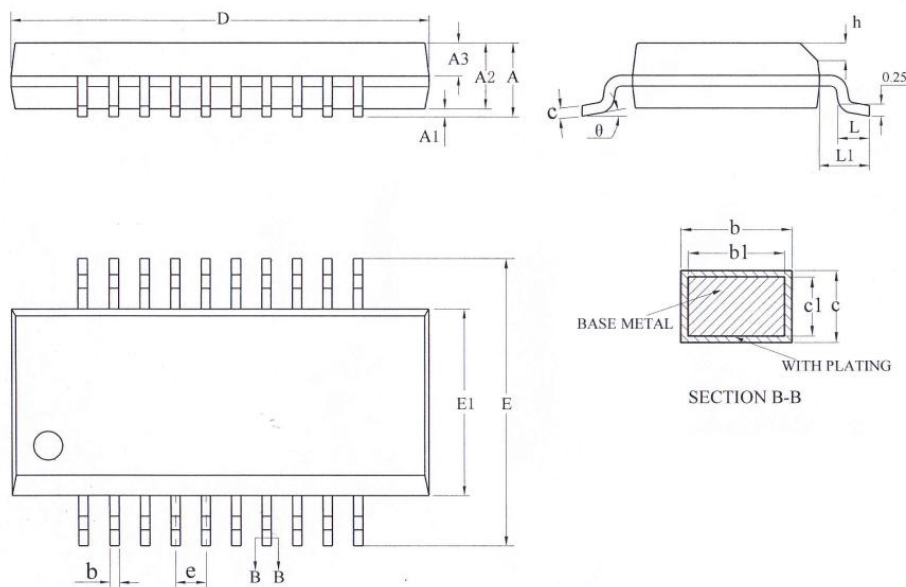


图 6-1 SOP16L 封装尺寸图

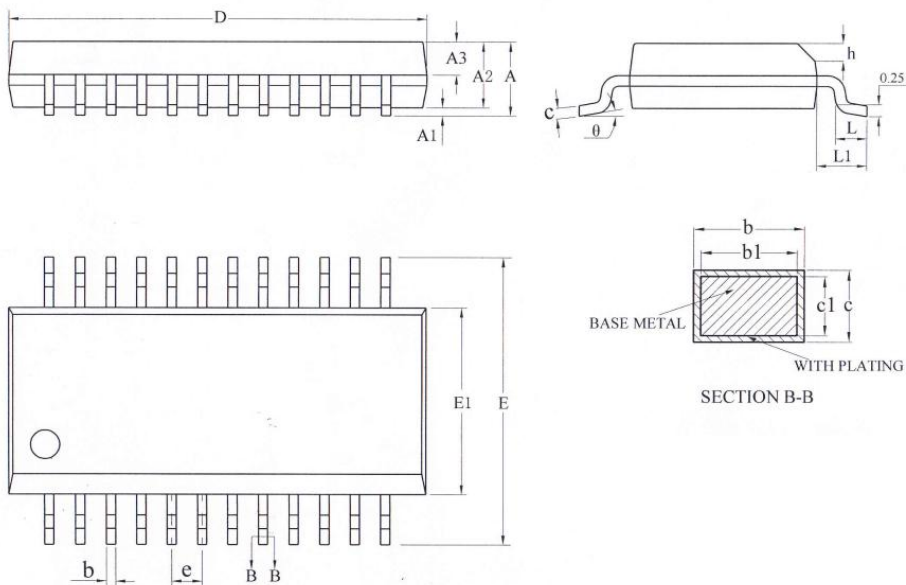
6.2 SSOP20 封装尺寸



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	0.15	0.25
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.23	—	0.31
b1	0.22	0.25	0.28
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	8.55	8.65	8.75
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.635BSC		
h	0.30	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

图 6-2 SSOP20 封装尺寸图

6.3 SSOP24 封装尺寸



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	0.15	0.25
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.23	—	0.31
b1	0.22	0.25	0.28
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	8.55	8.65	8.75
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.635BSC		
h	0.30	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

图 6-3 SSOP24 封装尺寸图