



深圳市芯海科技

CSU8RP1185 用户手册

基于 OTP ROM 的 8 位 RISC MCU

REV 0.6

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](http://www.chipsea.com)



## 37 引脚 8 位 OTP ROM 单片机产品介绍

### 高性能的 RISC CPU

- 8 位单片机 MCU
- 内置 2K×16 位一次性可编程存储器 (OTP ROM)
- 128 字节数据存储器 (SRAM)
- 只有 39 条单字指令
- 6 级存储堆栈
- 128Byte EPROM

### 振荡器

- 内带 4MHz 振荡器

### 模拟特性

- 模数转换器 (ADC)
  - 1 路全差分模拟输入
  - 24 位分辨率
  - 内部集成的可编程增益放大器
  - ADC 的输出速率 30Hz~3.9KHz
- 内带电荷泵
- 内带稳压器供传感器和调制器

### 专用微控制器的特性

- 上电复位 (POR)
- 上电复位延迟定时器 (158ms)
- 内带低电压复位 (LVR)
- Timer
  - 8 位可编程预分频的 8 位的定时计数器
- 扩展型看门狗定时器 (WDT)
  - 可编程的时间范围

### 外设特性

- 12 位双向 I/O 口

- 1 路蜂鸣器输出
- 4×12 的 LCD 驱动
  - 可选择内部晶振或 WDT 晶振作为时钟源
  - 可选择两种不同的 LCD 驱动波形
  - 可选择不同的偏置电压产生方式
- 2 个外部中断
- 低电压检测 (LVD) 引脚 (内部提供 2.4V、2.5V、2.6V、2.7V、2.8V、3.2V、3.6V 电压比较)
- 内置低电压烧录控制电路, 最低 2.5V 可以自烧录

### 低功耗特性

- MCU 工作电流
  - 正常模式 500uA@500KHz (工作电压 3.3V)
  - 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

- 37-PIN dice

### 应用场合

- 电子衡器
- 精密测量及控制系统

---

## 历史修改记录

2012年5月25日	初稿完成	版本号为 0.6

历史修改记录

## 目 录

历史修改记录.....	3
<b>1 产品概述.....</b>	<b>6</b>
1.1 主要特性.....	6
1.2 PIN配置.....	6
1.3 功能模块原理图.....	8
<b>2 标准功能.....</b>	<b>10</b>
2.1 CPU核.....	10
2.1.1 存储器.....	12
2.1.2 状态寄存器.....	14
2.1.3 INTE及INTF中断寄存器.....	15
2.2 SFR.....	17
2.2.1 系统专用寄存器.....	17
2.2.2 辅助专用寄存器.....	17
2.3 时钟系统.....	18
2.3.1 振荡器状态.....	18
2.3.2 CPU指令周期.....	18
2.3.3 蜂鸣器时钟.....	19
2.3.4 TMCLK（定时器模块输入时钟）.....	20
2.3.5 LCDCLK（LCD模块输入时钟）.....	21
2.4 定时器.....	23
2.5 I/O PORT.....	24
2.5.1 带模拟输入通道的数字I/O口：PT1[4].....	27
2.5.2 数字I/O口：PT1[7:5].....	29
2.5.3 数字I/O口与外部中断输入：PT2[1:0].....	31
2.5.4 数字I/O口PT2[6:2].....	33
2.5.5 数字I/O接口或者蜂鸣器输出：PT2[7].....	35
<b>3 增强功能.....</b>	<b>37</b>
3.1 电源系统.....	37
3.1.1 Regulator.....	37
3.1.2 低电压比较器.....	39
3.1.3 电荷泵.....	40
3.2 HALT与SLEEP模式.....	42
3.3 复位系统.....	43
3.4 看门狗.....	44
3.5 ADC模块.....	46
3.5.1 ADC寄存器说明.....	46
3.5.2 ADC增益以及时钟.....	48
3.5.3 ADC输出速率.....	48
3.5.4 ADC工作电流.....	48
3.5.5 ADC增益的温度特性调整.....	48
3.6 LCD DRIVER.....	49
3.6.1 LCD控制模式.....	49
3.6.2 LCD帧频选择.....	50
3.6.3 LCD偏置电压.....	52
3.6.4 LCD驱动波形.....	54

3.6.5	LCD寄存器说明 .....	62
3.6.6	LCD操作步骤 .....	63
3.7	OTP模块.....	65
3.8	OTP在线烧录.....	66
<b>4</b>	<b>MCU指令集 .....</b>	<b>67</b>
<b>5</b>	<b>电气特性.....</b>	<b>79</b>
5.1	最大极限值 .....	79
5.2	直流特性 (DVDD, AVDD = 3.3V, T <sub>A</sub> = 25°C, 如无其他说明则都是此条件) .....	79
5.3	ADC的特性 (VREF= 3V, T <sub>A</sub> = 25°C, 如无其他说明则都是此条件) .....	80
<b>6</b>	<b>BONDING说明 .....</b>	<b>81</b>
6.1	PIN排布.....	81
6.2	PIN坐标 .....	82

## 1 产品概述

### 1.1 主要特性

CSU8RP1185 芯片是一个 8 位 CMOS 单芯片 MCU，内置 2K×16 位一次性可编程（OTP）ROM，一个带有 1 路全差分模拟信号输入的 24 位 ADC，低噪声放大器及 4×12 的 LCD 驱动。

### 1.2 PIN 配置

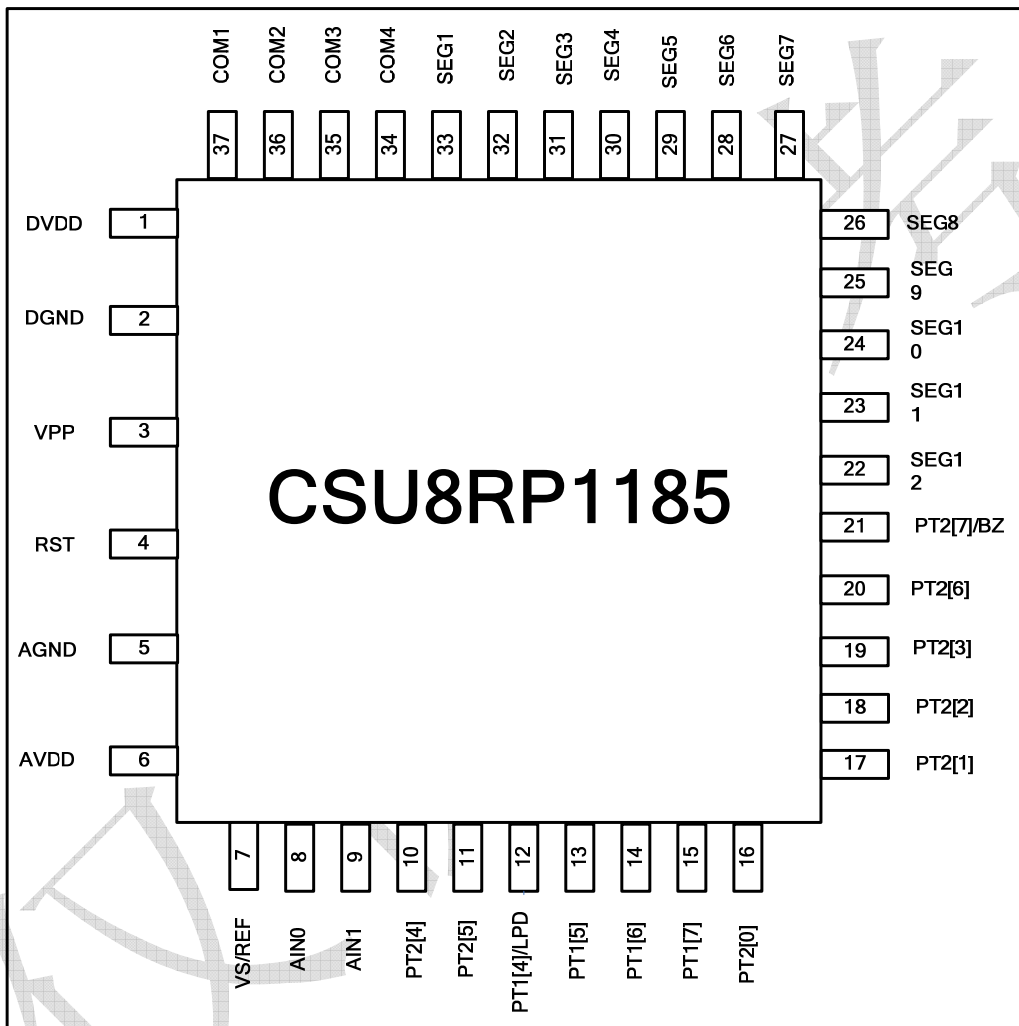


图 1-1 功能引脚图

表 1-1 引脚说明表

管脚名称	输入/输出	管脚序号	描述
DVDD	P	1	数字电源
DGND	P	2	数字地
VPP	P	3	烧录电源接口
RST	I	4	复位信号输入
AGND	P	5	模拟地
AVDD	P	6	模拟电源
VS/REF	O/I	7	稳压输出/参考电压输入
AIN0~1	I	8~9	模拟差分输入端
PT2[4]~[5]	I	10~11	I/O
PT1[4]/LPD	I/O	12	I/O 或者低电压检测输入端
PT1[5]	I/O	13	I/O
PT1[6]	I/O	14	I/O
PT1[7]	I/O	15	I/O
PT2[0]~[1]	I/O	16~17	OTP 烧录时钟及数据或作为 I/O/中断
PT2[2]	I/O	18	I/O
PT2[3]	I/O	19	I/O
PT2[6]	I/O	20	I/O
PT2[7] /BZ	I/O	21	I/O 或者蜂鸣器输出
SEG12~1	O	22~33	LCD Segment 输出
COM4~1	O	34~37	LCD Com 输出

### 1.3 功能模块原理图

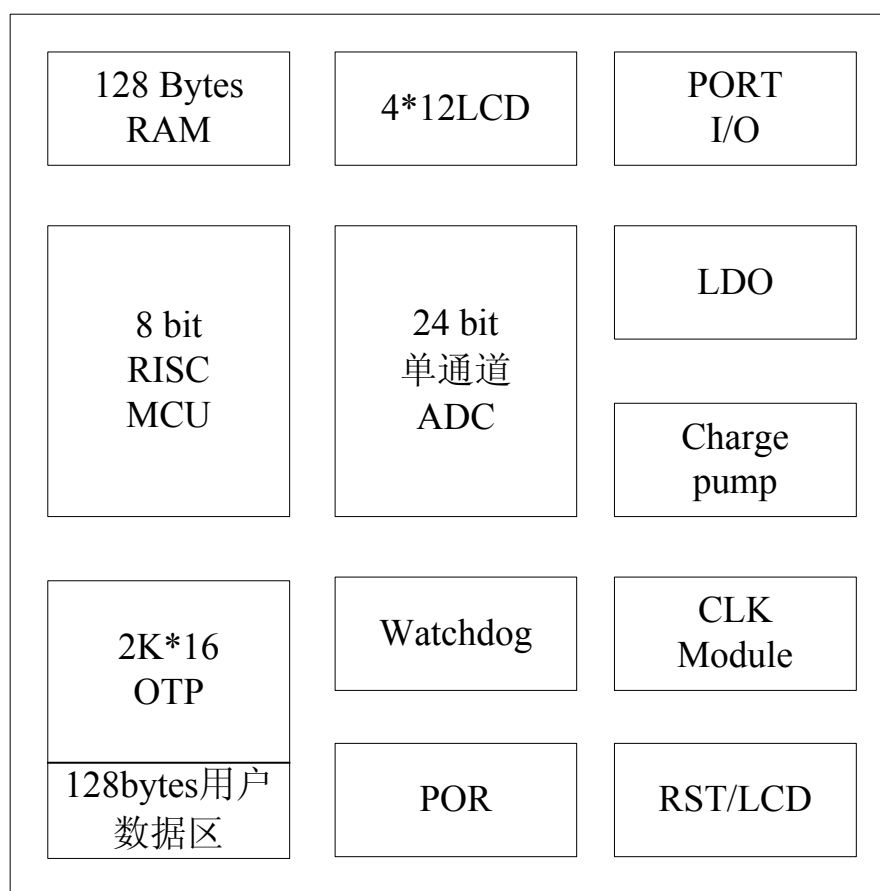


图 1-2 CSU8RP1185 功能模块

从功能模块原理图中可看到有 5 个功能模块，其描述见 表 1-2 CSU8RP1185 主要功能描述



表 1-2 CSU8RP1185 主要功能描述

项目	子项目	描述
CPU 核	RISC CPU Core	详细描述见 1.11 节
	OTP 程序存储器	OTP: 一次性可编程 2K 条编程指令
	数据存储器	CSU8RP1185 带有 256Bytes SRAM (128 Bytes 寄存器, 128 Bytes 普通数据存储器)
	时钟系统	CSU8RP1185 有一个内部 4M 晶振。
数据功能模块	定时器模块	用于定时中断及看门狗的时钟计数器
	LCD 模块	内带 4×12 的 LCD 驱动器
	Buzzer	用户连接一个蜂鸣器到内带的蜂鸣器接口以接收警告或提醒信号
	Ext.INT	CSU8RP1185 提供 2 个外部中断接口
模拟功能模块	ADC	内带 Sigma-Delta 的 ADC 将传感器的模拟信号转换为数字信号
电源功能模块	电源模块	CSU8RP1185 有一个专用的电源系统。此电源系统能为 ADC 提供固定的电压。芯片的输入电压可以在一个范围内浮动
普通用途 I/O	PT1	PT1 接口有 4 位。
	PT2	PT2 接口有 8 位。用户可以定义这 8 位接口用于普通用途或某些专用功能, 比如外部中断, 蜂鸣器

## 2 标准功能

### 2.1 CPU 核

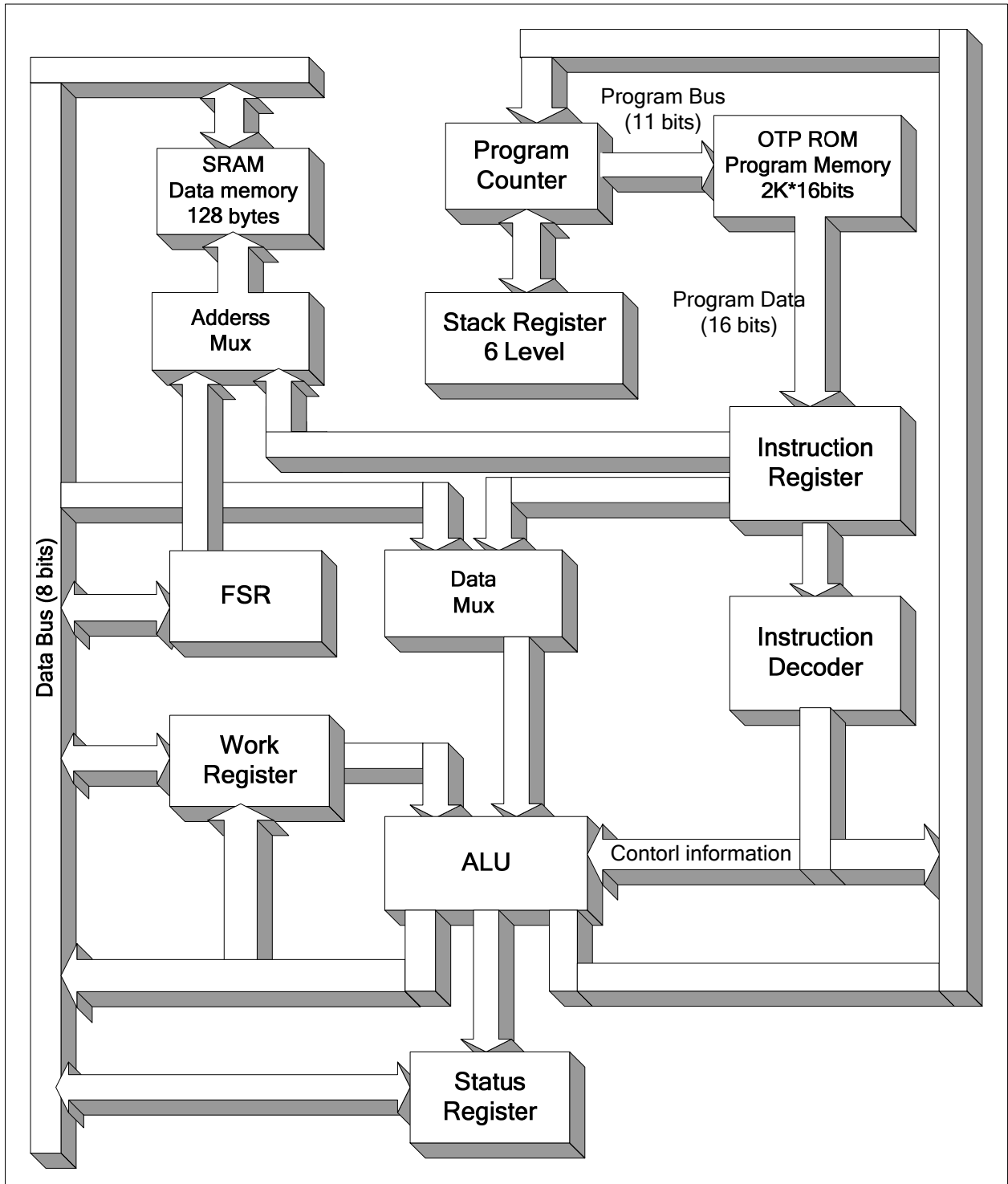


图 2-1 CSU8RP1185 CPU 核的功能模块图

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

表 2-1 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（11bits），指令指针推送到程序存储器，然后自动加 1 以进行下一次周期。
栈寄存器	堆栈寄存器是用来记录程序返回的指令指针。当程序调用函数，程序计数器会将指令指针推送到堆栈寄存器。在函数执行结束之后，堆栈寄存器会将指令指针送回到程序计数器以继续原来的程序处理。
指令寄存器	<p>程序计数器将指令指针（程序存储器地址）推送到程序存储器，程序存储器将程序存储器的数据（16bits）及指令推送到指令寄存器。</p> <p>CSU8RP1185 的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。</p> <p>CPU 能将立即数推送到工作寄存器，或者进行某些处理后，根据控制信息，将立即数存储到直接地址所指向的数据存储器寄存器中。</p> <p>直接地址（8bits） 数据寄存器的地址。CPU 能利用此地址来对数据存储器进行操作。</p> <p>直接数据（8bits） CPU 通过 ALU 利用此数据对工作寄存器进行操作。</p> <p>控制信息 它记录着 ALU 的操作信息。</p>
指令译码器	指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。
算术逻辑单元	算术逻辑单元不仅能完成 8 位二进制的加，减，加 1，减 1 等算术计算，还能对 8 位变量进行逻辑的与，或，异或，循环移位，求补，清零等逻辑运算。
工作寄存器	工作寄存器是用来缓存数据存储器中某些存储地址的数据。
状态寄存器	当 CPU 利用 ALU 处理寄存器数据时，如下的状态寄存器将会根据不同的指令而变化：PD，TO，DC，C 及 Z。
文件选择寄存器	在 CSU8RP1185 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	CSU8RP1185 内带 4K bytes 的 OTP ROM 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 2K 的指令。程序存储器的地址总线是 11bits，数据总线是 16bits。
数据存储器	CSU8RP1185 内带 128bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 8bits，数据总线是 8bits。除了 SRAM 之外，CSU8RP1185 还带有 128bytes 的 EPROM 可以作为用户数据存储区，寻址空间紧跟程序存储器。

2.1.1 存储器

1. 程序存储器主要用于指令的存储，在 CSU8RP1185 中，该程序存储器是 2046\*16bit 的 OTP。(范围为 000H~7FDH)，7FEH 和 7FFH 为保留地址。800H 到 87FH 为 EPROM 区 系统的 reset 地址为 000H，中断入口地址为 004H，需要注意的一点就是所有的中断共用同一个中断入口地址。

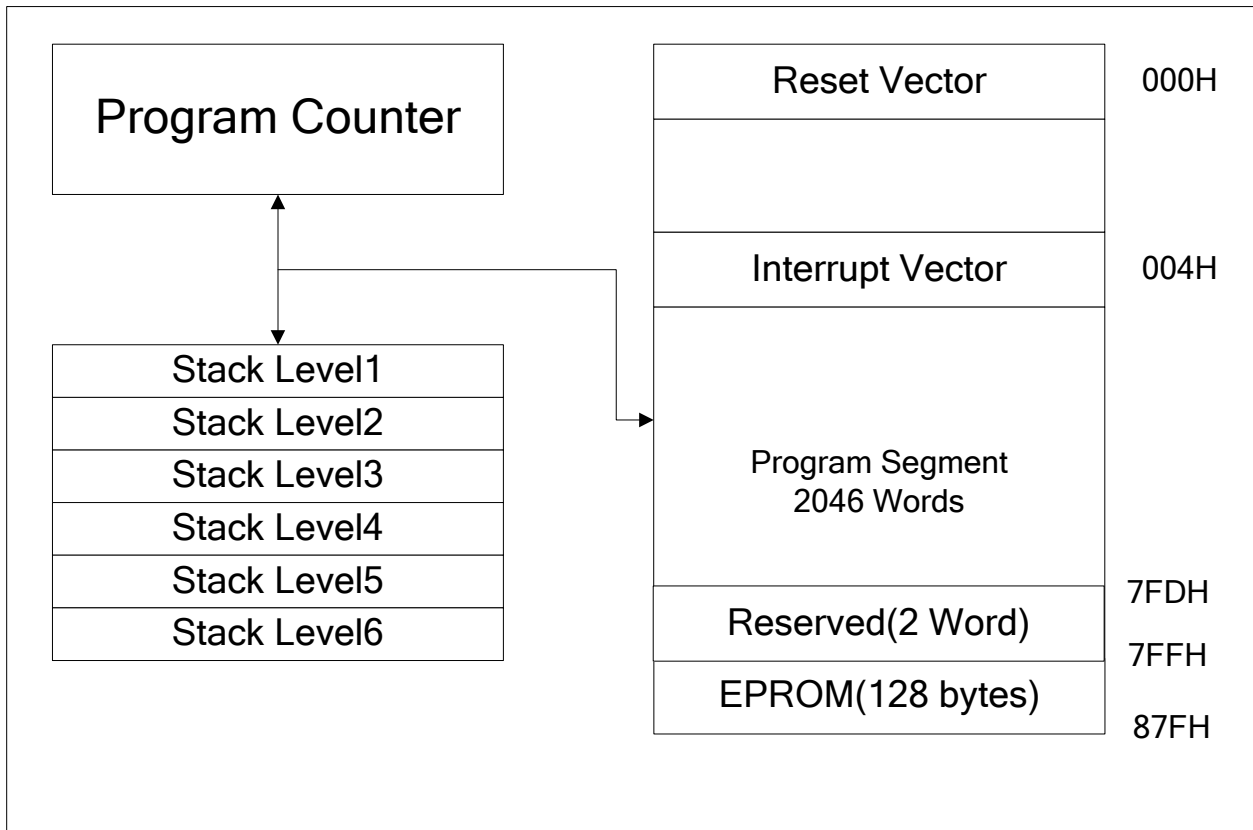


图 2-2 程序存储器

2. 数据存储器主要用于程序运行过程中，全局以及中间变量的存储。该存储器分为三个部分。地址的 00H 至 07H 是系统特殊功能寄存器，例如间接地址，间接地址指针，状态寄存器，工作寄存器，中断标志位，中断控制寄存器。地址的 08H 至 7FH 外设特殊功能寄存器，例如 IO 端口，定时器，ADC，LCD 驱动，系统特殊功能寄存器和外设特殊功能寄存器是用寄存器实现，而通用数据存储器是 RAM 实现，可以读出也可以写入。

表 2-2 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	00H	07H
外设特殊功能寄存器	08H	7FH
通用数据存储器	80H	FFH

3. 通过IND0 以及FSR0 这两个寄存器可以对数据存储器以及特殊功能寄存器进行间接访问。当从间接地址寄存器(IND0)读入数据时，MCU实际上是以FSR0 中的值作为地址去访问数据存储器得到数据。当向间接寄存器(IND0)写入数据时，MCU实际上是以FSR0 中的值作为地址去访问数据存储器将值存入该地址。其访问方式见图 2-3 间接地址访问。

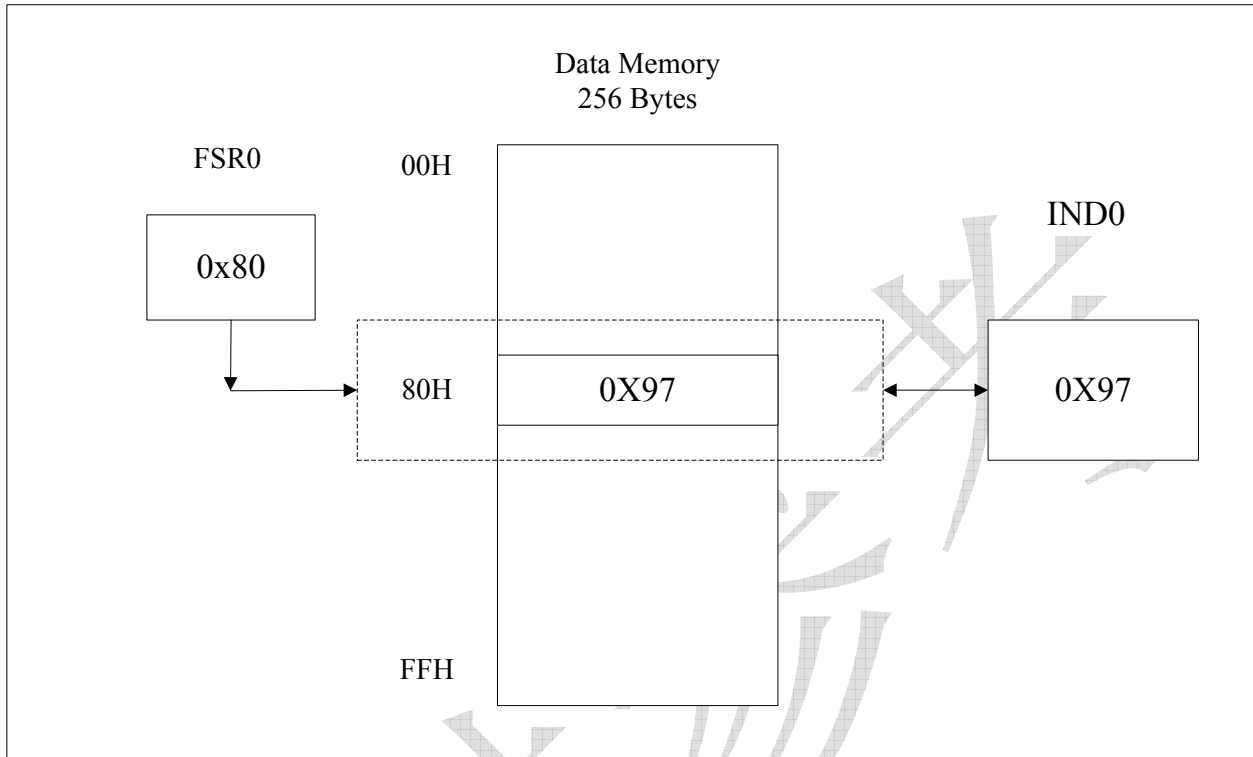


图 2-3 间接地址访问

### 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，则会影响到 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-X	U-X	U-X	R-0	R-0	R/W-X	R/W-X	R/W-X
STATUS				PD	TO	DC	C	Z
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 4 PD: 掉电标志位

1 = 执行 SLEEP 指令

0 = 上电复位后

Bit 3 TO: 看门狗定时溢出标志。

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 = 算术或逻辑操作结果为 0

0 = 算术或逻辑操作结果不为 0

#### 特性 (Property) :

R = 可读位

W = 可写位

U = 无效位

-n = 上电复位后的值

'1' = 位已设置

'0' = 位已清零

X = 不确定位

### 2.1.3 INTE 及 INTF 中断寄存器

中断系统的入口地址为 004H，各个中断之间没有优先级，靠程序控制各个中断的优先级。只要有中断标志位，就会有中断响应，响应中断之后需要软件将中断标志位清除，否则会不断响应中断。INTE 及 INTF 寄存器是可读、可写的，包括使能位及标志位，用于中断器件。

INTE 寄存器（地址为 07H）

特性	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTE	GIE			TMIE		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 4 TMIE: 8-Bit 定时器中断使能标志  
1 = 使能定时器中断  
0 = 不使能定时器中断
- Bit 2 ADIE: ADC 中断使能标志  
1 = 使能 ADC 中断  
0 = 不使能 ADC 中断
- Bit 1 E1IE: PT2.1 外部中断使能标志  
1 = 使能 PT2.1 外部中断  
0 = 不使能 PT2.1 外部中断
- Bit 0 E0IE: PT2.0 外部中断使能标志  
1 = 使能 PT2.0 外部中断  
0 = 不使能 PT2.0 外部中断

#### 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

INTF 寄存器（地址为 06H）

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

- Bit 4 TMIF: 定时中断标志  
1 = 发生定时中断, 必须软件清 0  
0 = 没发生定时中断
- Bit 2 ADIF: ADC 中断标志  
1 = 发生 ADC 中断, 必须软件清 0  
0 = 没发生 ADC 中断
- Bit 1 E1IF: PT2.1 外部中断标志  
1 = 发生 PT2.1 外部中断, 必须软件清 0  
0 = 没发生 PT2.1 外部中断
- Bit 0 E0IF: PT2.0 外部中断标志  
1 = 发生 PT2.0 外部中断, 必须软件清 0  
0 = 没发生 PT2.0 外部中断

**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	上电复位后的值
00H	IND0	以 FSR0 中内容作为地址的数据存储器中的数据								uuuuuuuu
02H	FSR0	间接数据存储器的地址指针 0								00000000
04H	STATUS				PD	TO	DC	C	Z	00u00uuu
05H	WORK	工作寄存器								00000000
06H	INTF				TMIF		ADIF	E11F	E0IF	00000000
07H	INTE	GIE			TMIE		ADIE	E11E	E0IE	00000000

### 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，ADC，信号的条件控制寄存器，LCD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	
0BH	EADRL					PARL[6:0]				u0000000	
0DH	WDTCON	WDTEN			Wdt_lcd		WDTS[3:0]			0uu00000	
0EH	TMOUT					TMOUT[7:0]				xxxxxxx	
0FH	TMCON	TRST				TMEN		INS[2:0]		1uuu0000	
10H	AD0H				AD0[23:16]				xxxxxxx		
11H	AD0L				AD0[15:8]				xxxxxxx		
12H	AD0LL				AD0[7:0]				xxxxxxx		
13H	ADCON					ADSC		ADM[2:0]		uuuu0000	
14H	MCK						M2 CK	M1 CK		uuuuu00u	
15H	PCK			LCDSCK[3:0]			S BEEP[1:0]			u000000u	
18H	NETA	SINL[1:0]								00uuuuuu	
19H	NETB					ERV				uuuuxuuu	
1AH	NETC			ADGAIN[5:0]				ADEN		0000000u	
1BH	NETD					VLCDX[1:0]		LCDREF[1:0]		uuuu0000	
1CH	NETE	LDOS[1:0]				SILB[2:0]		ENLB		00u0000u	
1DH	NETF	CHP_VPP		ENVDDA				BGID[1:0]		ENVB	0u0uu000
1FH	SVD								LBOU	uuuuuuux	
20H	PT1		PT1[7:4]							xxxxuuuu	
21H	PT1EN		PT1EN[7:4]							0000uuuu	
22H	PT1PU		PT1PU[7:4]							0000uuuu	
23H	AIENB					AIENB1				uuuu0uuu	
24H	PT2		PT2[7:0]							xxxxxxx	
25H	PT2EN		PT2EN[7:0]							00000000	
26H	PT2PU		PT2PU[7:0]							00000000	
27H	PT2MR	BZEN				E1M[1:0]		E0M[1:0]		0uuu0000	
40H	LCD1						SEG1[3:0]			uuuu0000	
41H	LCD2						SEG2[3:0]			uuuu0000	
42H	LCD3						SEG3[3:0]			uuuu0000	
43H	LCD4						SEG4[3:0]			uuuu0000	
44H	LCD5						SEG5[3:0]			uuuu0000	
45H	LCD6						SEG6[3:0]			uuuu0000	
46H	LCD7						SEG7[3:0]			uuuu0000	
47H	LCD8						SEG8[3:0]			uuuu0000	
48H	LCD9						SEG9[3:0]			uuuu0000	
49H	LCD10						SEG10[3:0]			uuuu0000	
4AH	LCD11						SEG11[3:0]			uuuu0000	
4BH	LCD12						SEG12[3:0]			uuuu0000	
58H	LCDENR	LCDCK[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL	00001110	
59H	TEMPC	TEMPC[7:0]								00000000	

## 2.3 时钟系统

### 2.3.1 振荡器状态

内置振荡器只有执行 Sleep 指令后才能被关闭。

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

### 2.3.2 CPU 指令周期

表 2-6 CSU8RP1185 CPU 指令周期寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
14H	MCK						M2_CK	M1_CK		uuuuu00u

用户可以通过设置 M1\_CK, M2\_CK 来选择指令周期（用户必须保证切换指令周期时，时钟切换是稳定的；一般在切换后加一条 NOP 指令）。

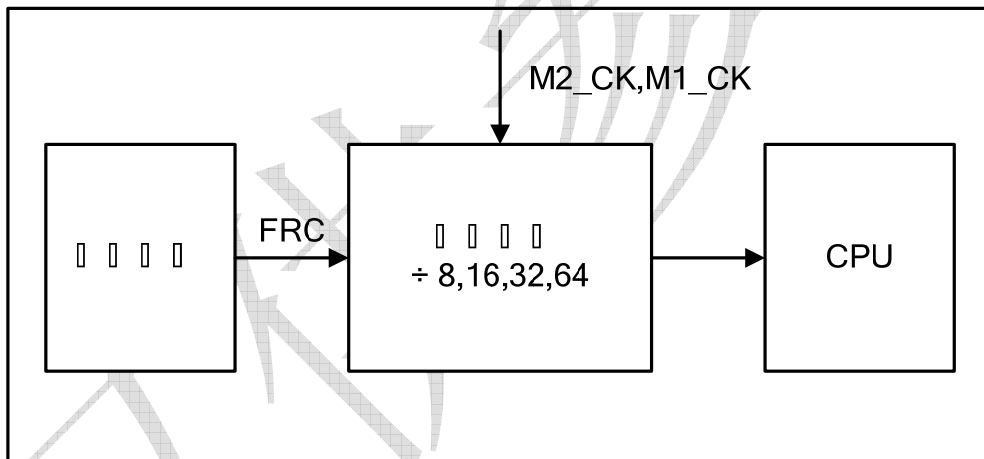


图 2-4 CPU 指令周期时钟示意图

表 2-7 指令周期选择列表

M2_CK	M1_CK	指令周期(KHz)	
0	0	$F_{RC}/32$	125
0	1	$F_{RC}/64$	62.5
1	0	$F_{RC}/8$	500
1	1	$F_{RC}/16$	250

2.3.3 蜂鸣器时钟

表 2-8 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]			uuuuu00u

CSU8RP1185 有一个蜂鸣器时钟用于蜂鸣器源。用户通过设置S\_BEEP寄存器标志位来改变蜂鸣器时钟，设置如表 2-9。

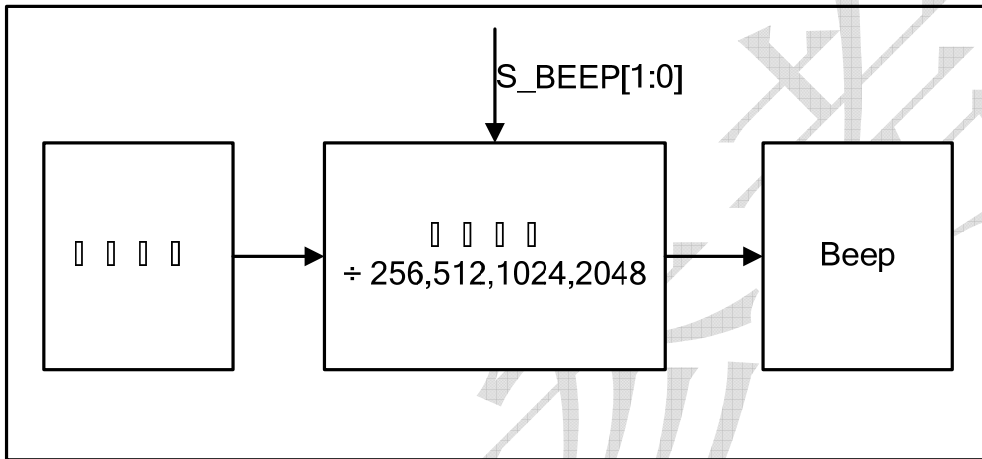


图 2-5 Beep 时钟选择示意图

表 2-9 蜂鸣器时钟选择列表

S_BEEP		时钟源(KHz)		BEEP CLOCK(KHz)	
0	0	ICK	4000	ICK/256	16
0	1	ICK	4000	ICK/512	8
1	0	ICK	4000	ICK/1024	4
1	1	ICK	4000	ICK/2048	2

### 2.3.4 TMCLK（定时器模块输入时钟）

TMCLK用于CSU8RP1185 定时器模块。根据表 2-10，用户通过正确设置M1\_CK 标志位以选择TMCLK的频率。

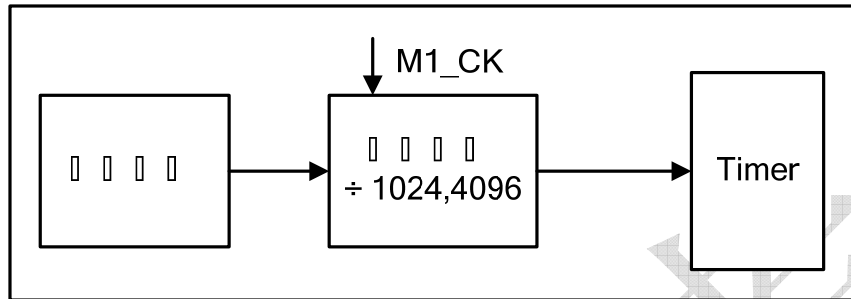


图 2-6 定时器时钟分频示意图

表 2-10 TMCLK 选择列表

M1_CK	时钟源(KHz)		TMCLK(Hz)	
0	ICK	4000	ICK/1024	3906
1	ICK	4000	ICK/4096	976

### 2.3.5 LCDCLK (LCD 模块输入时钟)

LCD 的帧频率可以通过设置寄存器标志 LCDCK[1:0]确定。CSU8RP1185 对 LCD 模块的输入时钟进行分频以获得 LCDCK。

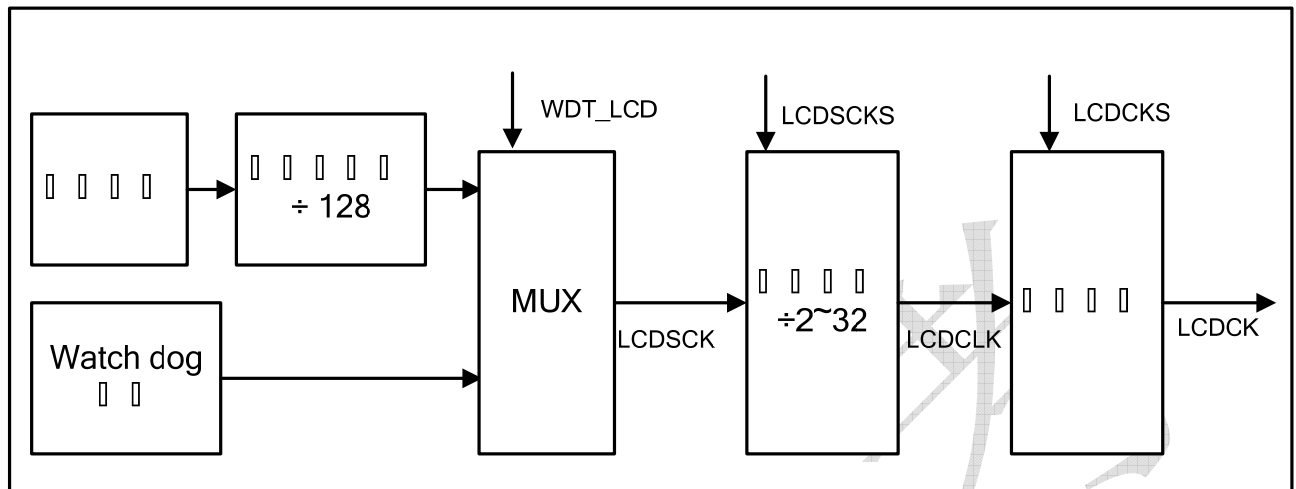


图 2-7 LCD 帧频时钟选择

表 2-11 LCDCLK 选择列表

Wdt lcd	LCDSCKS				LCDCLK(KHz)	
0	0	0	0	0	LCDSCK/32	1
0	0	0	0	1	LCDSCK/30	1.067
0	0	0	1	0	LCDSCK/28	1.143
0	0	0	1	1	LCDSCK/26	1.231
0	0	1	0	0	LCDSCK/24	1.333
0	0	1	0	1	LCDSCK/22	1.455
0	0	1	1	0	LCDSCK/20	1.6
0	0	1	1	1	LCDSCK/18	1.778
0	1	0	0	0	LCDSCK/16	2
0	1	0	0	1	LCDSCK/14	2.286
0	1	0	1	0	LCDSCK/12	2.667
0	1	0	1	1	LCDSCK/10	3.2
0	1	1	0	0	LCDSCK/8	4
0	1	1	0	1	LCDSCK/6	5.333
0	1	1	1	0	LCDSCK/4	8
0	1	1	1	1	LCDSCK/2	16
1	0	0	0	0	WTDCLK/32	0.094
1	0	0	0	1	WTDCLK/30	0.1
1	0	0	1	0	WTDCLK/28	0.107
1	0	0	1	1	WTDCLK/26	0.115
1	0	1	0	0	WTDCLK/24	0.125
1	0	1	0	1	WTDCLK/22	0.136
1	0	1	1	0	WTDCLK/20	0.15
1	0	1	1	1	WTDCLK/18	0.167
1	1	0	0	0	WTDCLK/16	0.188
1	1	0	0	1	WTDCLK/14	0.214
1	1	0	1	0	WTDCLK/12	0.25
1	1	0	1	1	WTDCLK/10	0.3
1	1	1	0	0	WTDCLK/8	0.375
1	1	1	0	1	WTDCLK/6	0.5
1	1	1	1	0	WTDCLK/4	0.75
1	1	1	1	1	WTDCLK/2	1.5

表 2-12 LCD 帧频选择列表

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

芯海科技

## 2.4 定时器

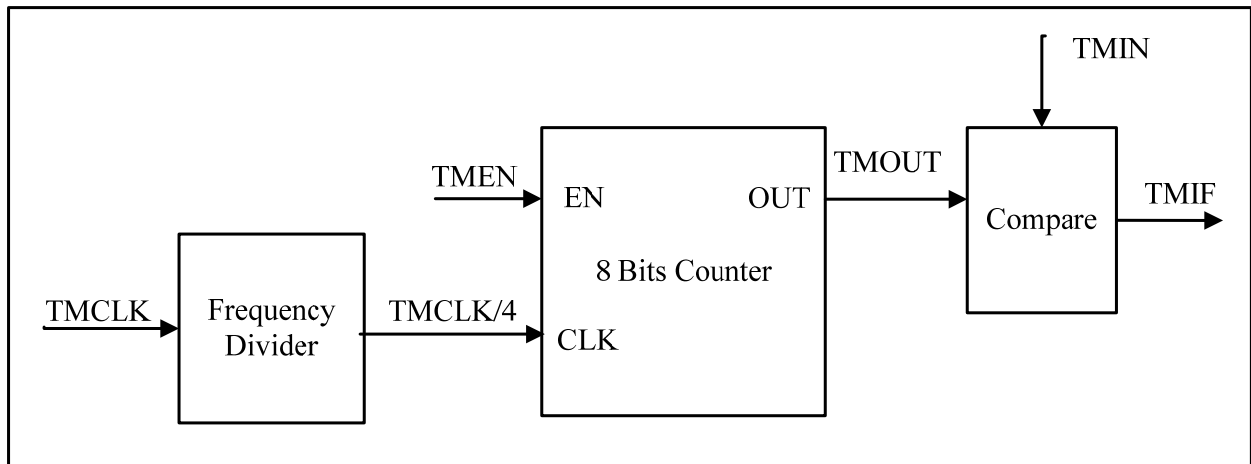


图 2-8 定时模块的功能框图

定时器模块的输入是 TMCLK。在定时器模块集成了一个分频器对 TMCLK 进行 4 分频，分频的时钟作为 8 bits 计数器的输入时钟。当用户设置了定时器模块的使能标志，8 bits 计数器将启动，TMOUT[7:0]将会从 00H 递增至 0FFH。用户需要设置 INS（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，中断标志位会自设置，程序计数器会跳转到 004H 以执行中断服务程序。

表 2-13 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				TMIF					00000000
07H	INTE	GIE			TMIE					00000000
0EH	TMOUT	TMOUT[7:0]								00000000
0FH	TMCON	TRST				TMEN		INS[2:0]		1uuu0000

操作：

1. 设置 TMCLK，为定时器模块选择输入。
2. 设置 INS[2:0]，选择定时器中断源。请看表 2-14。
3. 设置寄存器标志位：TMIE 与 GIE，使能定时器中断。
4. 设置寄存器标志位：TMEN，使能定时器模块的 8 bits 计数器。
5. 清零寄存器标志位：TRST，复位定时器模块的计数器。
6. 当定时超时发生时，寄存器标志位 TMIF 会自复位，程序计数器会复位为 004H。

表 2-14 定时器选择列表

INS[2:0]	中断源	时间 (TMCLK = 976Hz)
000	TMOUT[0]	1/128 s
001	TMOUT[1]	1/64 s
010	TMOUT[2]	1/32 s
011	TMOUT[3]	1/16 s
100	TMOUT[4]	1/8 s
101	TMOUT[5]	1/4 s
110	TMOUT[6]	1/2 s
111	TMOUT[7]	1 s

## 2.5 I/O port

表 2-15 I/O 口寄存器表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF				--	--		E1IF	E0IF	00000000
07H	INTE	GIE			--	--		E1IE	E0IE	00000000
20H	PT1	PT1[7:4]								00000000
21H	PT1EN	PT1EN[7:4]								00000000
22H	PT1PU	PT1PU[7:4]								00000000
23H	AIENB					AIENB1				00000000
24H	PT2	PT2[7:0]								00000000
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000
27H	PT2MR	BZEN				E1M[1:0]		E0M[1:0]		00000000

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。CSU8RP1185 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	R/W-X	R/W-X	R/W-X	R/W-X	U-0	U-0	U-0	U-0
PT1	PT1[7:4]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1[7:4]: GPIO1 口数据标志  
 PT1[7] = GPIO1 bit 7 数据标志位  
 PT1[6] = GPIO1 bit 6 数据标志位  
 PT1[5] = GPIO1 bit 5 数据标志位  
 PT1[4] = GPIO1 bit 4 数据标志位

**PT1EN 寄存器（地址为 21H）**

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
PT1EN	PT1EN[7:4]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1EN[7:4]: GPIO1 口输入/输出控制标志  
 PT1EN[7] = GPIO1 bit 7 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口  
 PT1EN[6] = GPIO1 bit 6 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口  
 PT1EN[5] = GPIO1 bit 5 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口  
 PT1EN[4] = GPIO1 bit 4 的 I/O 控制标志位；0 = 定义为输入口，1 = 定义为输出口



**PT1PU 寄存器（地址为 22H）**

特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
PT1PU	PT1PU[7:4]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PU[7:4]: GPIO1 口上拉电阻使能标志

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

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

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

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

**AENB 寄存器（地址为 23H）**

特性	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
AENB					AIENB1			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3 AIENB1: PT1[4]数模通道选择信号

0 = PT1[4]定义为模拟通道

1 = PT1[4]定义为数字通道

**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]: GPIO 2 口输入/输出控制标志

PT2EN[7] = GPIO2 bit 7 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

PT2EN[6] = GPIO2 bit 6 的 I/O 控制标志位; 0 = 定义为输入口, 1 = 定义为输出口

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

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

**PT2MR 寄存器（地址为 27H）**

特性	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2MR	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

1 = 使能蜂鸣器功能, GPIO2 口 bit 7 定义为蜂鸣器输出接口

0 = 不使能蜂鸣器功能, GPIO2 口 bit 7 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 (GPIO2 口 bit 1) 在状态改变时触发

10 = 外部中断 1 (GPIO2 口 bit 1) 在状态改变时触发

01 = 外部中断 1 (GPIO2 口 bit 1) 为上升沿触发

00 = 外部中断 1 (GPIO2 口 bit 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 (GPIO2 口 bit 0) 在状态改变时触发

10 = 外部中断 0 (GPIO2 口 bit 0) 在状态改变时触发

01 = 外部中断 0 (GPIO2 口 bit 0) 为上升沿触发

00 = 外部中断 0 (GPIO2 口 bit 0) 为下降沿触发

**特性 (Property):**

R = 可读位

W = 可写位

U = 无效位

-n = 上电复位后的值

'1' = 位已设置

'0' = 位已清零

X = 不确定位

## 2.5.1 带模拟输入通道的数字 I/O 口：PT1[4]

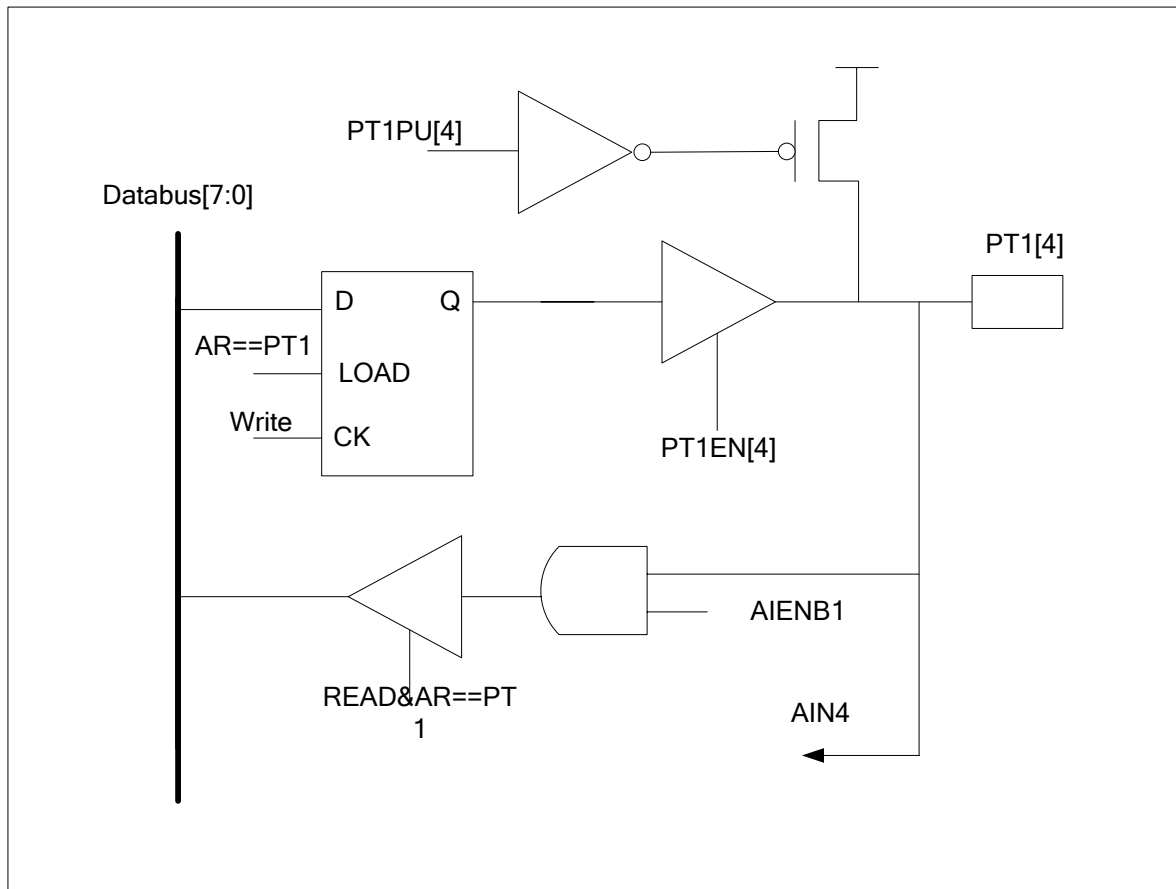


图 2-9 PT1[4] 功能框图

GPIO1 口（PT1[4]）功能框图如图 2-9 PT1[4] 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 4（PT1[4]）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

CSU8RP1185 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（CSU8RP1185 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

CSU8RP1185 在 PT1 口集成内部上拉电阻功能，上拉电阻大约为 100KΩ（上拉电流大约为 30uA。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[4] 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 2-16 PT1 寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
20H	PT1				PT1[4]					uuuuuuuu
21H	PT1EN				PT1EN[4]					00000000
22H	PT1PU				PT1PU[4]					00000000
23H	AENB					AIENB1				00000000

## 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[4]置低（设置为数字输入），PT1UP[4]置低（没有上拉电阻）。
5. 需先使能 ENVB，模拟输入才能正常工作。
6. 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。

## 注意操作：

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

## 2.5.2 数字 I/O 口：PT1[7:5]

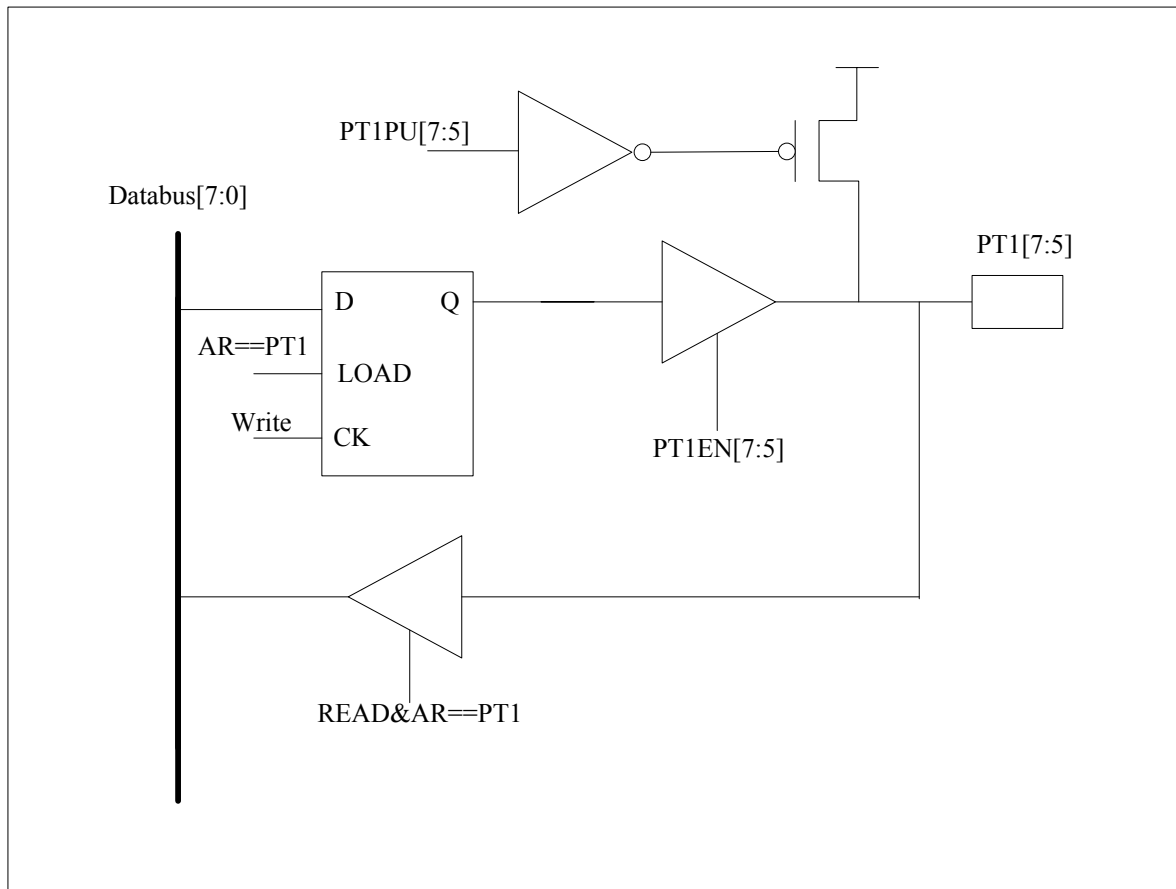


图 2-10 PT1[7:5] 功能框图

GPIO1 口（PT1[7:5]）功能框图如图 2-10 PT1[7:5] 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[7:5] 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 7~bit 5（PT1[7:5]）可用于输入数字。当 PT1EN[n] 置为 0 时，PT1[7:5] 设置为数字输入。

- 输出

CSU8RP1185 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（CSU8RP1185 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

CSU8RP1185 在 PT1 口集成内部上拉电阻功能，上拉电阻大约为 100KΩ（上拉电流大约为 30uA。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[7:5] 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 2-17 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[7:5]								uuuuuuuu
21H	PT1EN	PT1EN[7:5]								00000000
22H	PT1PU	PT1PU[7:5]								00000000

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：

4. 置位寄存器标志 PT1EN[n]。PT1 [n]被定义为输出接口。
5. 置位相应的寄存器标志 PT1PU[n]。PT1 [n]连接到内部的上拉电阻。
6. 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。

注意操作：

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



表 2-18 PT2 寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
06H	INTF				--	--	--	E1IF	E0IF	00000000
07H	INTE	GIE			--	--	--	E1IE	E0IE	00000000
24H	PT2	PT2[7:0]								uuuuuuuu
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000
27H	PT2MR					E1M[1:0]		E0M[1:0]		00000000

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”。

注意操作：

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



## 2.5.4 数字 I/O 口 PT2[6:2]

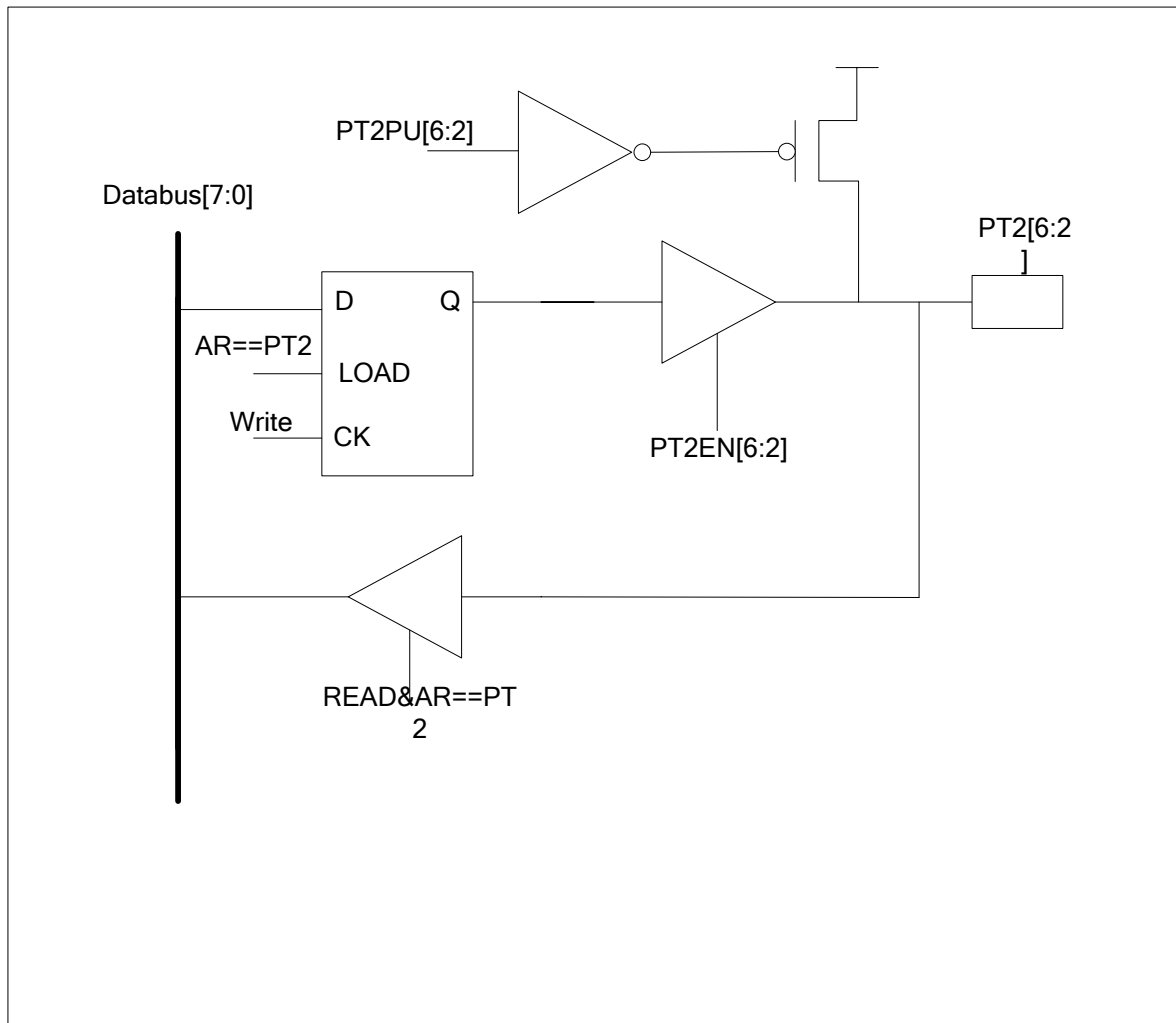


图 2-12 PT2[6:2] 功能框图

GPIO2 口 bit 6:2 (PT2[6:2]) 的功能框图如图 2-12 PT2[6:2] 功能框图所示。此 GPIO 口的主要功能是用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[6:2] 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 口 bit 6:2 (PT2[6:2]) 可以作为普通用途的 I/O 口。

- 输出

CSU8RP1185 使用内部 D 锁存器输出数字数据。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 (CSU8RP1185 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

CSU8RP1185 在 PT2 口集成内部上拉电阻功能，上拉电阻大约为 100KΩ (上拉电流大约为 30uA)。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[6:2] 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 2-19 PT2 寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
24H	PT2	PT2[7:0]								uuuuuuuu
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000

## 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. 置位相应的寄存器标志位：PT2EN[n]。PT2[n]被定义为输出接口。
2. 置位相应的寄存器标志位：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。

## 注意操作：

1. 在 I/O 口与 VDD 之间并联一个小电阻（大约 10K $\Omega$ ），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 接口或者蜂鸣器输出：PT2[7]

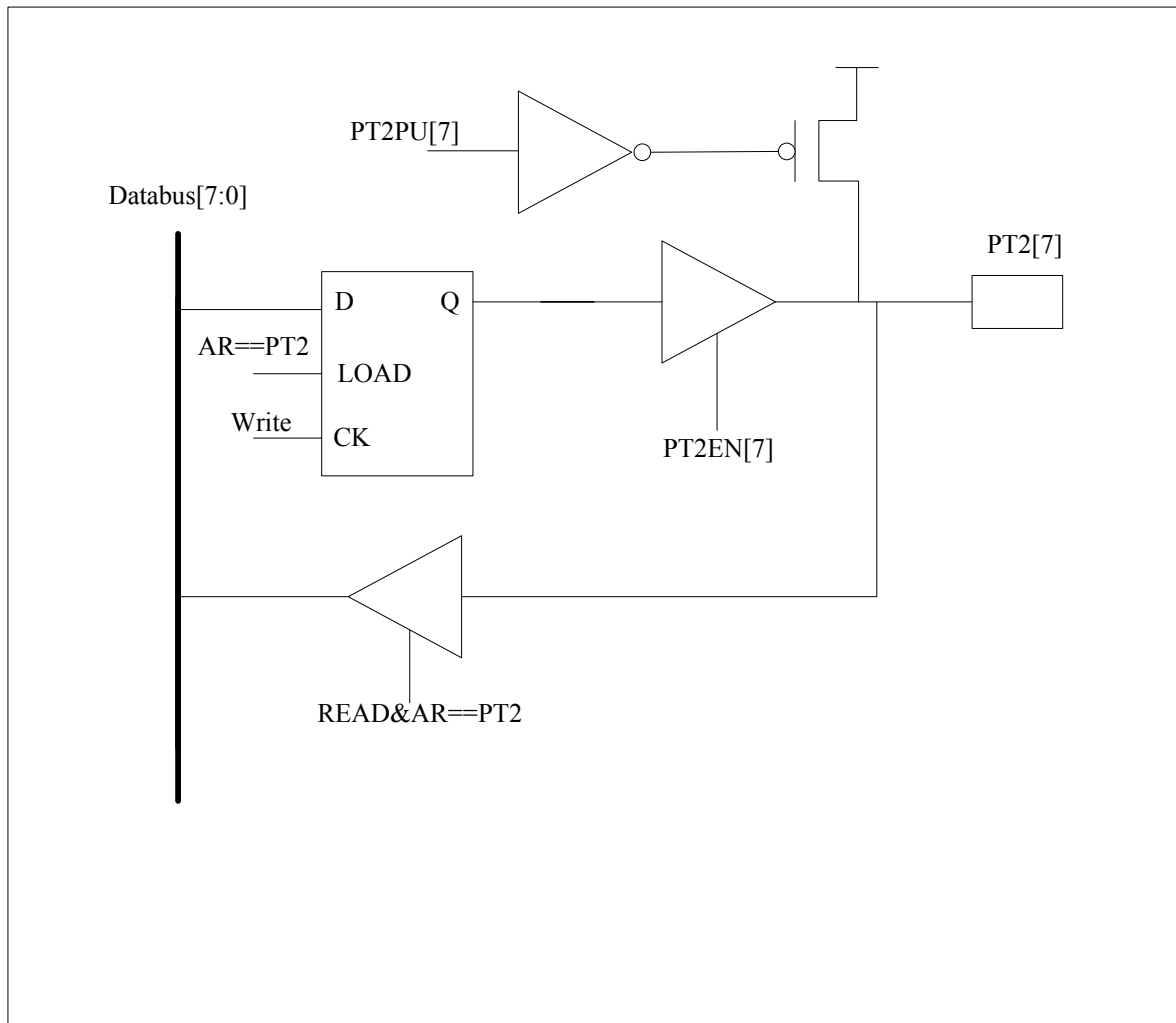


图 2-13 PT2[7] 功能框图

GPIO2 口 bit7 (PT2[7]) 的功能框图如图 2-13 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] 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 口 bit 7 (PT2[7]) 可以作为蜂鸣器输出接口，或者作为普通 I/O 接口。通过设置寄存器标志 BZEN 决定是否使能蜂鸣器输出。

输出

CSU8RP1185 使用内部 D 锁存器输出数字数据。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 (CSU8RP1185 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

CSU8RP1185 在 PT2 口集成内部上拉电阻功能，上拉电阻大约为 100KΩ (上拉电流大约为 30uA。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] 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 2-20 PT2[7]寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
24H	PT2	PT2[7:0]								uuuuuuuu
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000
27H	PT2MR	BZEN				E1M[1:0]		E0M[1:0]		00000000

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. 置位相应的寄存器标志位：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. 置位相应的寄存器标志位：PT2EN[n]。PT2[n]被定义为输出接口。
2. 置位相应的寄存器标志位：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。

蜂鸣器输出操作：

1. 置位寄存器标志位 PT2EN[7]。PT2[7]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。PT2[7]就作为蜂鸣器输出接口。
4. 将一个蜂鸣器与 PT2 bit7 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 10K $\Omega$ ），当 PT2PU[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

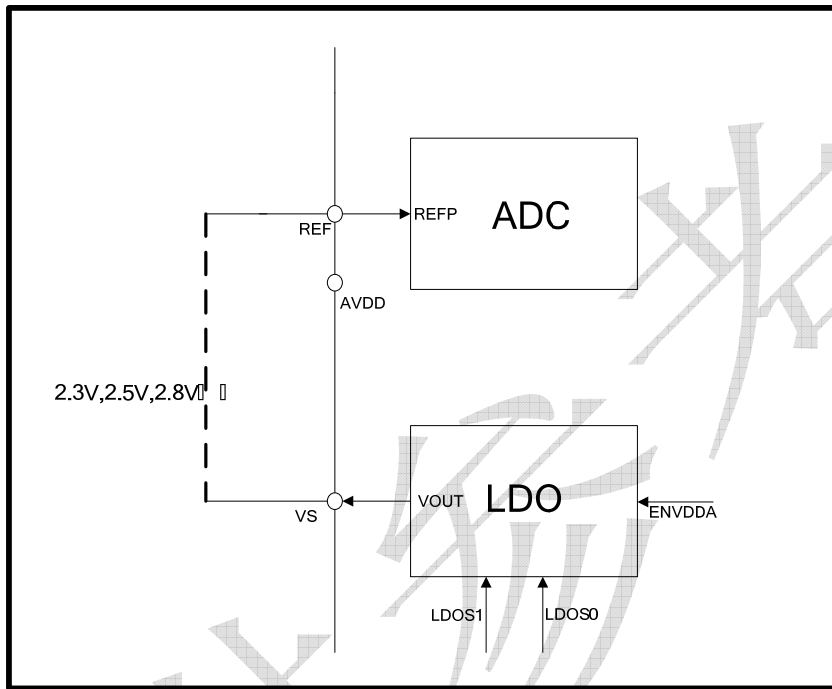


图 3-1 稳压电路

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 2.3V，2.5V，2.8V，3.0V可选。ENVDDA作为LDO的使能信号。LDO的控制寄存器标志是ENVDDA与LDOS。输出电压是VS。ENVB作为整个模拟电源部分的使能信号，关断之后ADC和LCD 电荷泵等将会不工作。

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]								00000000
1DH	NETF			ENVDDA					ENVB	00000000

NETE 寄存器 (地址=1CH)

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

Bit7~6 LDOS[1:0]: VS 电压值选择

- LDOS[1:0] 00 VS=3.0
- LDOS[1:0] 01 VS=2.8
- LDOS[1:0] 10 VS=2.5
- LDOS[1:0] 11 VS=2.3

NETF 寄存器 (地址=1DH)

特性	W-0	W-0	R/W-0	U-X	U-X	U-X	U-X	R/W-0
NETF			ENVDDA					ENVB
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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

Bit0 ENVB: 模拟电源使能信号  
 ENVB=1: 模拟电源使能  
 ENVB=0: 模拟电源不使能

操作:

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

3.1.2 低电压比较器

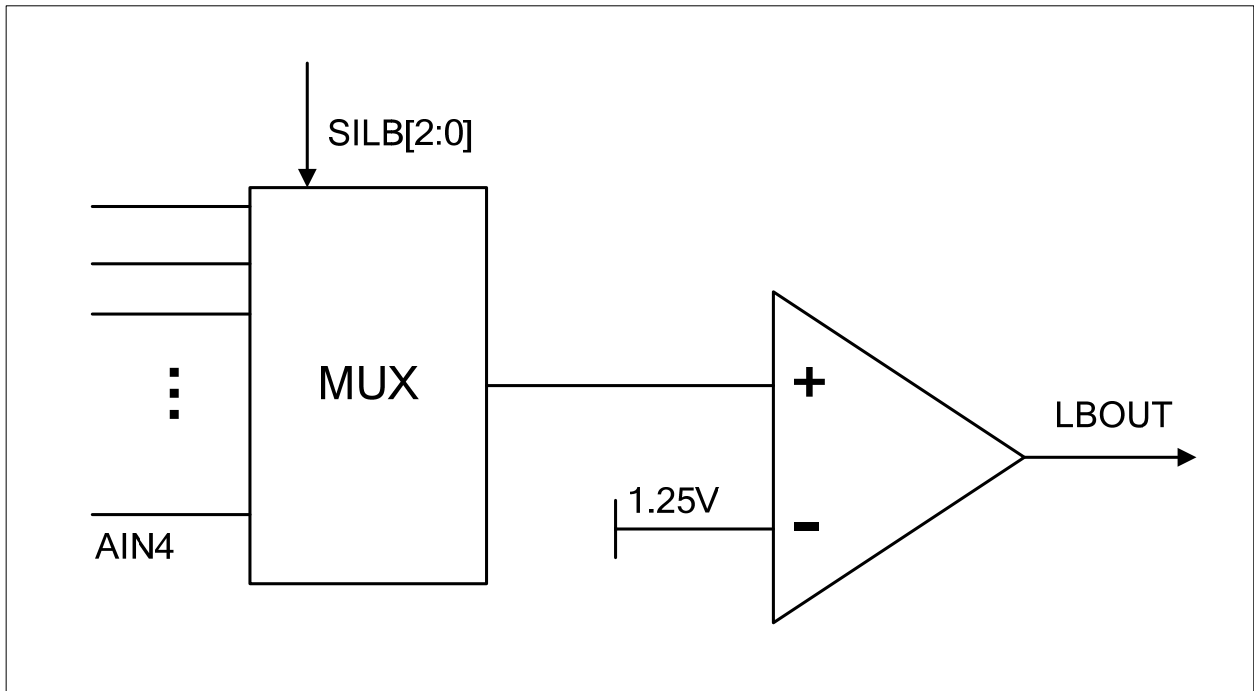


图 3-2 低电压比较功能模块框图

低电压比较器用于VDD的低电压检测。CSU8RP1185 集成一个可产生 1/2VDD及 1/3VDD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 1.25V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。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				SILB[2:0]			ENLB		00000000
1FH	SVD								LBOUT	uuuuuuuu

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。

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	VDD	VDD>2.4V	LBOUT=1
001	VDD	VDD>2.5V	LBOUT=1
010	VDD	VDD>2.6V	LBOUT=1
011	VDD	VDD>2.7V	LBOUT=1
100	VDD	VDD>2.8V	LBOUT=1
101	VDD	VDD>3.6V	LBOUT=1
110	AIN4	AIN4>1.25V	LBOUT=1
111	VDD	VDD>3.6V	LBOUT=1

3.1.3 电荷泵

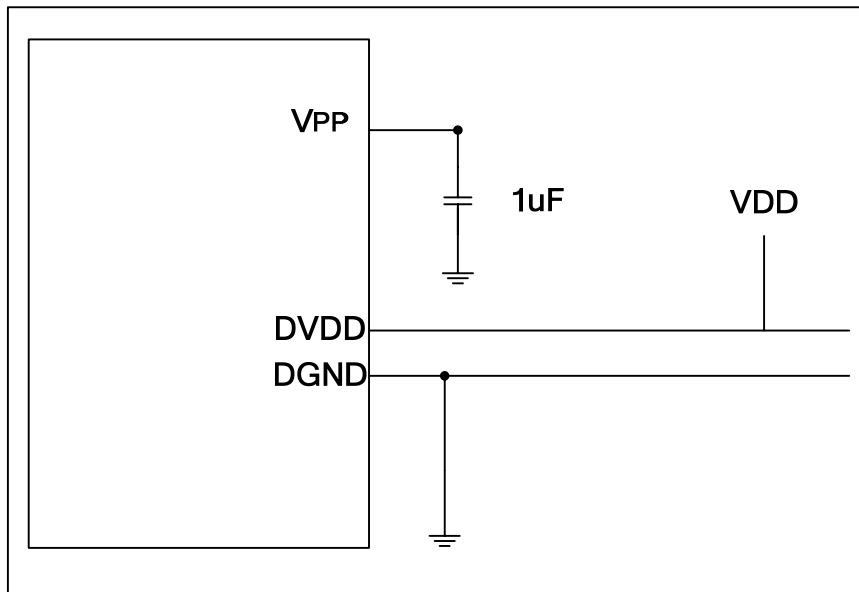


图 3 电荷泵外围电路示意图

电荷泵电路主要有两种使用方式，一种是提供 LCD 的显示电压源，一种是作为自烧录时提供烧录电压。当使用电荷泵电路时，需要在 VPP 引脚处接入一个 1uF 的电容。

不同应用情况下的寄存器配置和外置电容接法。

功能	LCDEN	CHP_VPP	ENPMPL	外置电容 (VPP 引脚)
LCD 关闭, 不升压, 不外部供电	0	0	0	无需外接电容
内部供电显示,LCD 接 DVDD	1	X	0	无需外接电容
自烧录,LCD 关闭	0	1	1	接外置电容
自烧录,LCD 显示	1	1	1	接外置电容
内部供电显示,LCD 接 Pump	1	0	1	接外置电容
Pump 开,LCD 关闭	0	0	1	接外置电容

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	上电复位值
1BH	NETD					VLCDX[1:0]				00000000
1DH	NETF	CHP_VPP							ENVB	00000000
58H	LCDENR								ENPMPL	00000000

NETF 寄存器 (地址=1DH)

特性	R/W-0	W-0	R/W-0	U-X	U-X	U-X	U-X	R/W-0
NETF	CHP_VPP							ENVB

Bit 7 CHP\_VPP: 升压泵电压选项

0 = 升压泵电压受 VLCDX 选择

1 = 升压泵电压泵到烧录电压 7.5V



NETD 寄存器（地址为 1BH）

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
NETD					VLCDX[1:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit3-2 VLCDX: VLCD 输出电压选择

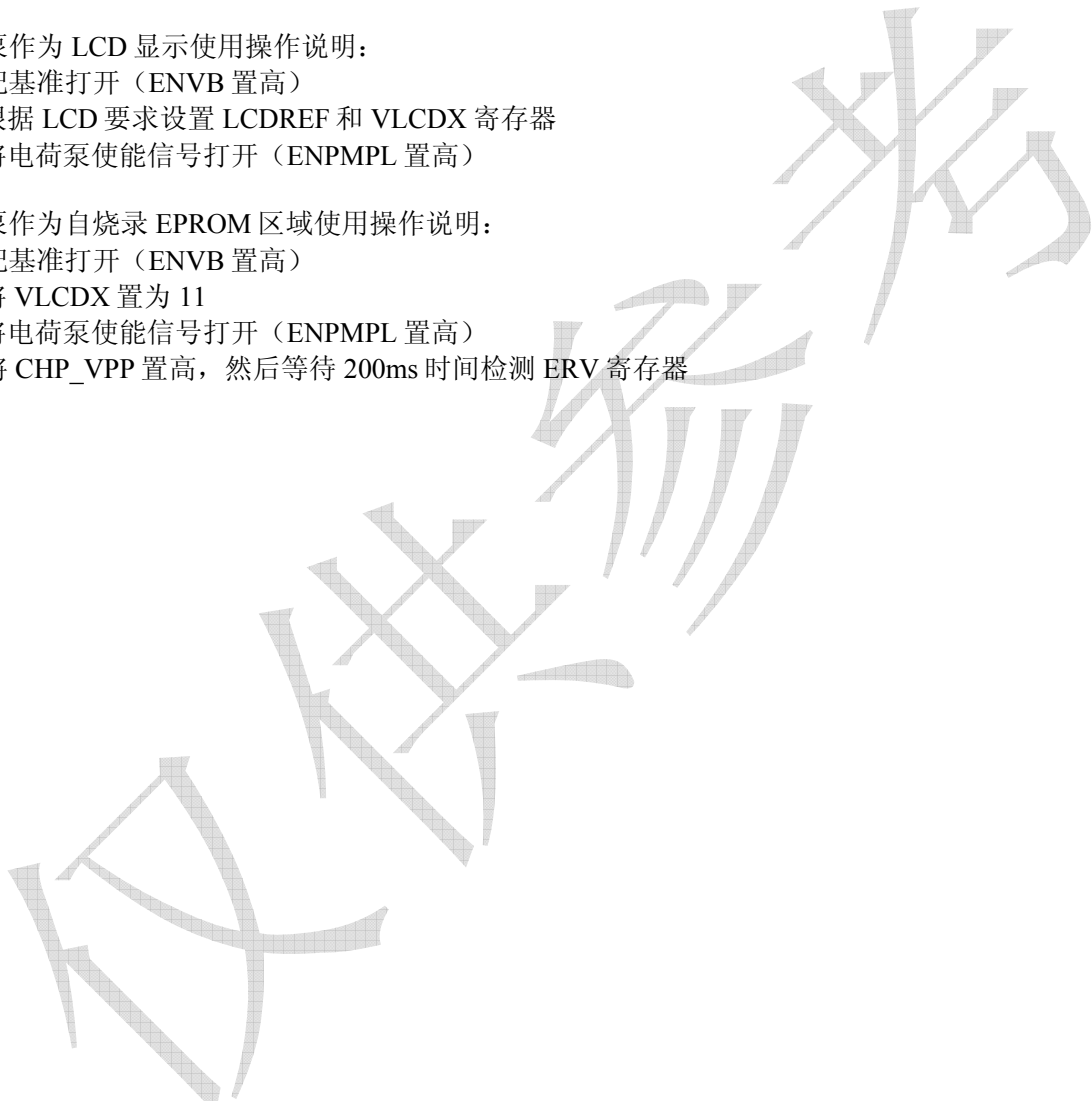
- 00 = 保留
- 01 = 2.8V
- 10 = 3.0V
- 11 = 3.2V

电荷泵作为 LCD 显示使用操作说明:

- 1 把基准打开 (ENVB 置高)
- 2 根据 LCD 要求设置 LCDREF 和 VLCDX 寄存器
- 3 将电荷泵使能信号打开 (ENPMPL 置高)

电荷泵作为自烧录 EPROM 区域使用操作说明:

- 1 把基准打开 (ENVB 置高)
- 2 将 VLCDX 置为 11
- 3 将电荷泵使能信号打开 (ENPMPL 置高)
- 4 将 CHP\_VPP 置高, 然后等待 200ms 时间检测 ERV 寄存器



### 3.2 Halt 与 Sleep 模式

CSU8RP1185 支持低功耗工作模式。为了使 CSU8RP1185 处于待机状态，可以让 CPU 停止工作使 CSU8RP1185 进行停止或睡眠模式，减低功耗。这两种模式描述如下：

停止模式

CPU 执行停止指令后，程序计数器停止计数直到出现中断指令。为了避免由中断返回（Interrupt Return）引起的程序错误，建议在停止指令之后加一 NOP 指令以保证程序返回时能正常运行。

睡眠模式

CPU 执行睡眠指令后，所有的振荡器停止工作直到出现一个外部中断指令复位 CPU。为了避免由中断返回（Interrupt Return）引起的程序错误，建议在停止指令之后加一 NOP 指令以保证程序的正常运行。在睡眠模式下的功耗大约有 2uA。

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，需要关闭所有的电源模块及模拟电路，并且保证所有的 I/O 口是接到 DVDD 或 DGND 电平。

在执行睡眠指令之前，先执行下面的程序。

```
CLRF NETA      ;复位状态
CLRF NETC      ;复位状态
CLRF NETE      ;复位状态
CLRF NETF      ;复位状态
CLRF PT1PU     ;断开 PT1 上拉电阻
MOVLW FFH
MOVWF PT1EN    ;PT1[7:4]用作输出接口
CLRF PT1       ;将 PT1[7:4]输出为低
MOVLW 01H
MOVWF PT2PU    ;断开 PT2 口除 bit0 (PT2[0]) 外的其它接口的上拉电阻
MOVLW 0FEH
MOVWF PT2EN    ;除 bit0 (PT2[0]) 外，PT2[7:0]用作输出接口
CLRF PT2       ;将 PT2[7:1]输出为低
CLRF INTF      ;清零中断标志
MOVLW 081H
MOVWF INTE     ;使能外部中断
SLEEP          ;使 CSU8RP1185 进行睡眠模式
NOP            ;保证 CPU 重启后程序能正常工作
```

### 3.3 复位系统

CSU8RP1185 包括以下几种复位方式：

- 上电复位
- 看门狗复位
- 掉电复位
- 外部硬件复位

上述复位方式中除看门狗复位以外的复位方式发生时，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 000H 处重新开始运行。当看门狗复位发生时，系统寄存器值仍然保持不变，程序停止运行，同时程序计数器 PC 清零。复位结束后，系统从向量 000H 处重新开始运行。

系统复位需要一定的时间，并提供完整的上电复位过程。对于不同类型的振荡器，完成复位所需要的时间也不同。因此，DVDD 的上升速度和不同晶振的起振时间都不固定。晶体振荡器类型不同则复位时间亦存在差别，这使得 DVDD 上升时间和启动时间不是确定值。

在 CSU8RP1185 中，除看门狗复位以外的复位方式发生以后，系统需要等待 158ms 的时间，才能开始正常工作。

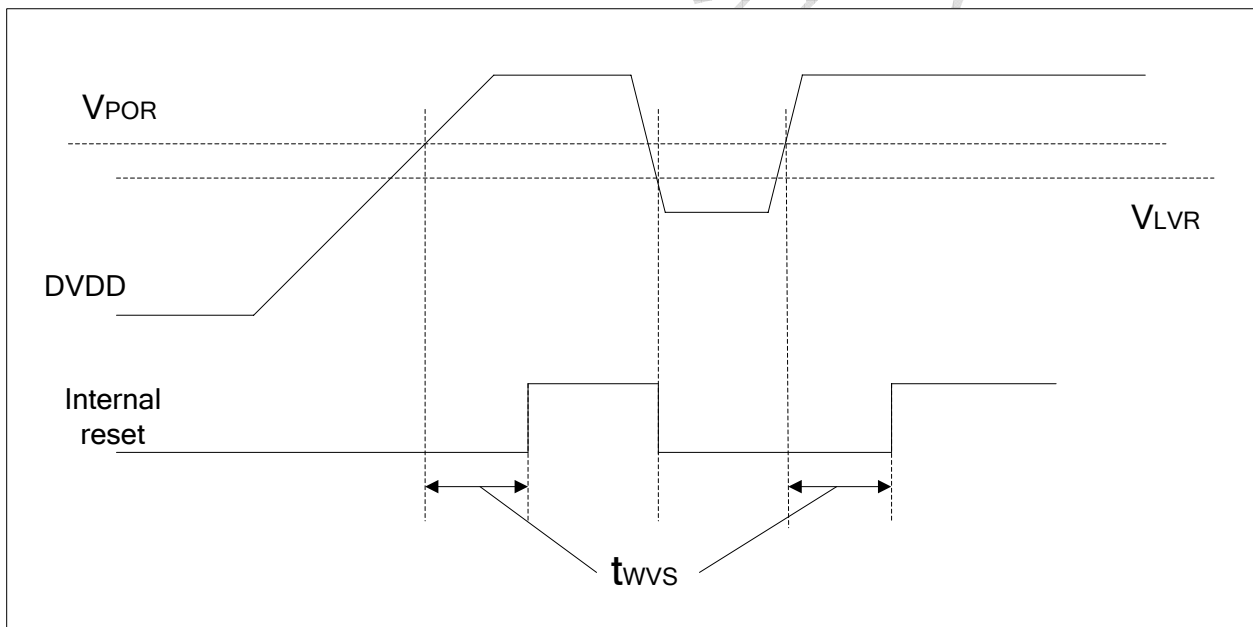


图 3-4 上电复位电路示例及上电过程

参数	典型值
VPOR	2.2V
VLVR	2.0V
tWVS	158ms

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

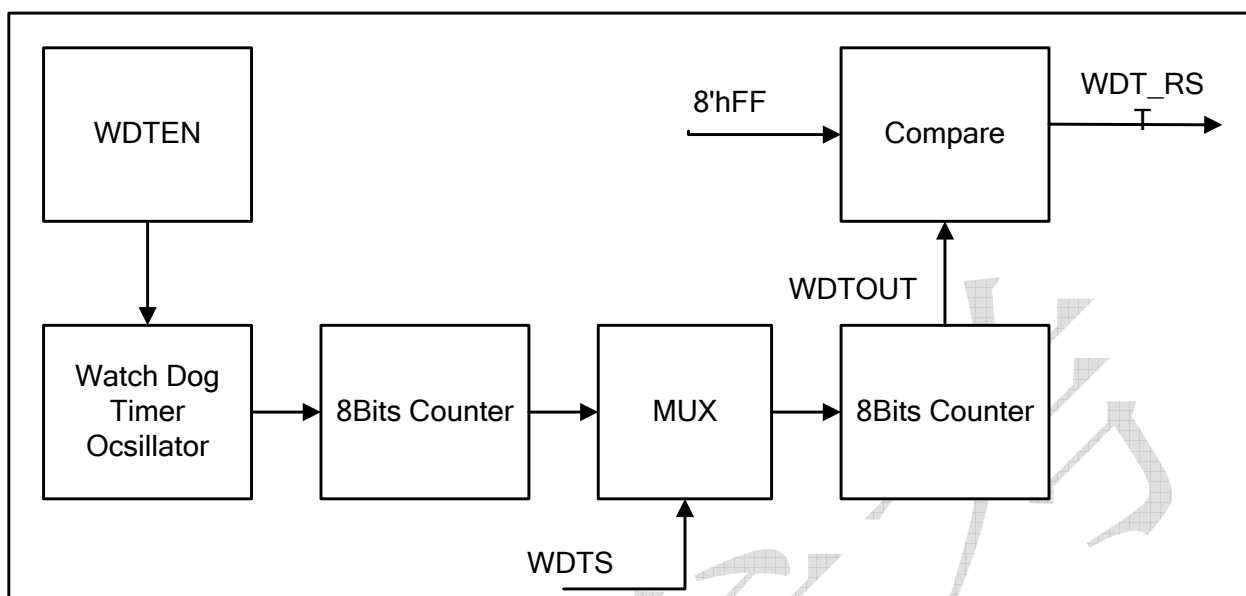


图 3-5 看门狗定时器功能框图

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 CPU 复位。在正常运行时，程序一般在 WDT 复位 CPU 之前先复位 WDT。当出现某些故障时，程序会被 WDT 复位到正常状态下。

看门狗定时器的输入是寄存器标志位：WDTEN 与 WDTS[2:0]，WDT 的输出是寄存器标志位：TO。当用户置位 WDTEN 时，则内部的看门狗定时器振荡器（3KHz）将会启动，产生的时钟被送到“8 bits 计数器 1”，如图 3-5 所示。“8 bits 计数器 1”的输出是虚信号 WDTA[7:0]，被发送到一个受寄存器标志位 WDTS[2:0] 控制的多路选择器，选择器的输出作为“8 bits 计数器 2”的时钟输入。当“8 bits 计数器 2”溢出时，它会发送 WDTOUT 信号复位 CPU（程序计数器将会跳转到 000H 以复位程序）及置位 TO 标志位。

当 WDTS[3] 为 1 时，8 bits 计数器 2 的计数到 93 时溢出，当 WDTS[3] 为 0 时，8 bits 计数器 2 的计数到 255 时溢出。用户可以使用指令 CLRWDT 复位 WDT。

表 3-4 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				00u00uuu
0DH	WDTCN	WDTEN			Wdt_lcd	WDTS[3:0]				0uuuu000

操作:

1. 设置 WDTS[3:0], 选择 WDT 超时频率。
2. 置位寄存器标志位: WDTEN, 使能 WDT。
3. 在程序中执行 CLRWDT 指令复位 WDT。

当wdt\_lcd标志位置高以后, LCD将采用wdt的 2 分频时钟作为LCD\_CLK, 此时LCSSCKS的寄存器对于LCD\_CLK的分频作用无效, 但是LCDCKS的帧频分频仍然有效。具体时钟分频请参考 3.6.2LCD帧频选择

表 3-5 看门狗时钟选择列表

WDTS[3]	WDTS[2:0]	计数器时钟	时间
0 (8 bits 计数器 1)	000	WDTIN[7]	21.8 s
	001	WDTIN [6]	10.9 s
	010	WDTIN [5]	5.5 s
	011	WDTIN [4]	2.7 s
	100	WDTIN [3]	1.4 s
	101	WDTIN [2]	0.68 s
	110	WDTIN [1]	0.34 s
	111	WDTIN [0]	0.17 s
1 (6bits 计数器 1)	000	WDTIN[7]	8 s
	001	WDTIN [6]	4 s
	010	WDTIN [5]	2s
	011	WDTIN [4]	1 s
	100	WDTIN [3]	0.5s
	101	WDTIN [2]	0.25s
	110	WDTIN [1]	0.125s
	111	WDTIN [0]	0.0625s

### 3.5 ADC 模块

#### 3.5.1 ADC 寄存器说明

表 3-6 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				--	--	ADIF	--	--	00000000	
07H	INTE	GIE			--	--	ADIE	--	--	00000000	
10H	ADOH	ADO[23:16]								00000000	
11H	ADOL	ADO[15:8]								00000000	
12H	ADOLL	ADO[7:0]									
13H	ADCON					ADSC		ADM[2:0]		uuuu0000	
18H	NETA	SINL								00000000	
1AH	NETC	ADGAIN[5:0]						ADEN			00000000
1DH	NETF						BGID[1:0]			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	
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	
ADOLL	ADO[7:0]								
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 23-0 ADO[23:0]: ADC 数字输出

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

ADO[22] = ADC 数字输出数据 bit 22

~

~

~

ADO[0] = ADC 数字输出数据 bit 0

**NETA 寄存器（地址为 18h）**

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

Bit 7-6 SINL :

00 = ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 Vin+, AIN1 为 Vin-;

01 = 同 00;

10 = 保留;

11 = ADC 输入端连接到 AIN0 和 AIN1, AIN0 为 Vin-, AIN1 为 Vin+;

**NETC 寄存器（地址为 1AH）**

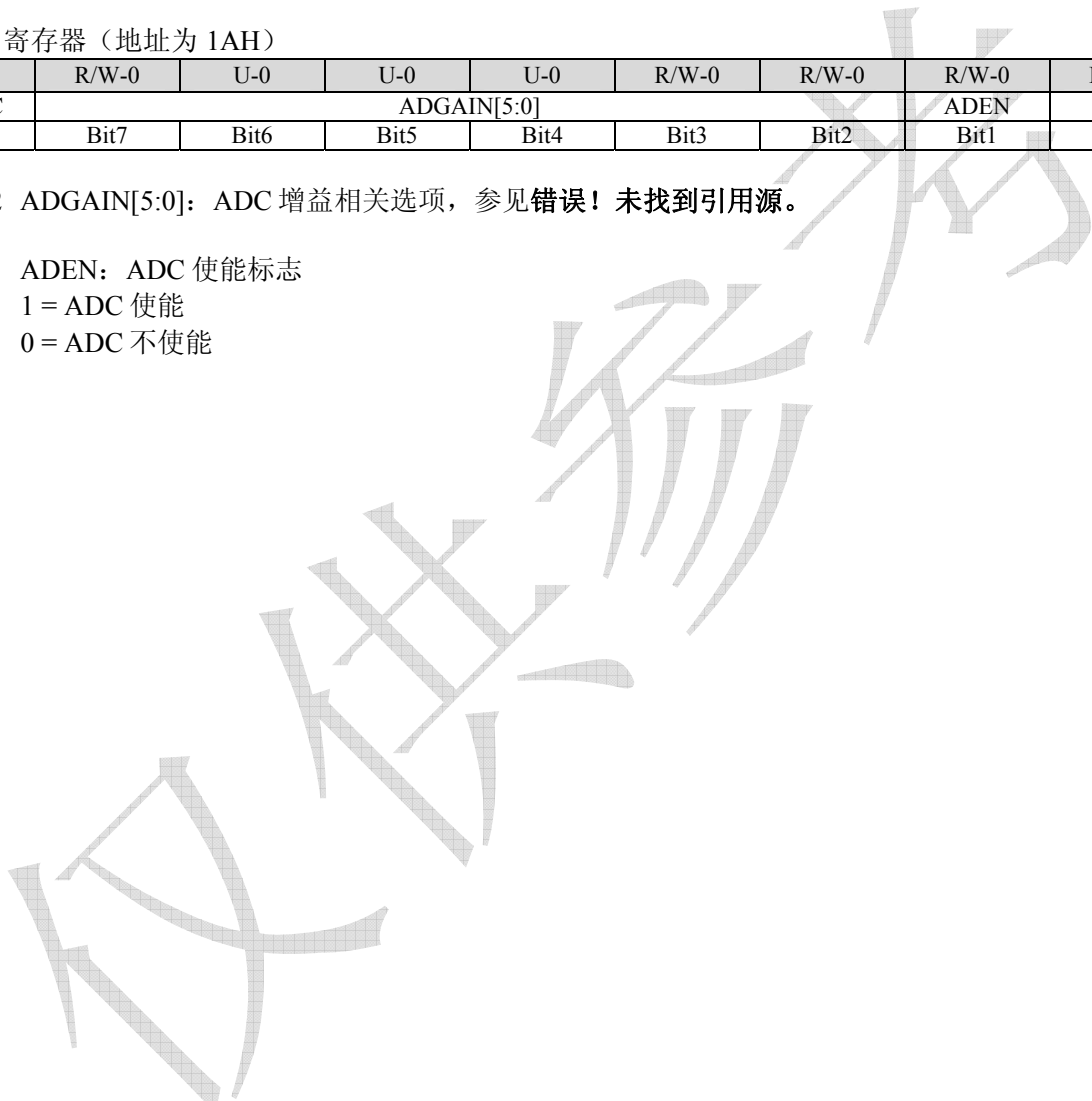
特性	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
NETC	ADGAIN[5:0]						ADEN	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-2 ADGAIN[5:0]: ADC 增益相关选项, 参见**错误! 未找到引用源。**

Bit 1 ADEN: ADC 使能标志

1 = ADC 使能

0 = ADC 不使能



### 3.5.2 ADC 增益以及时钟

表 3-7 ADC 增益选择列表

VS	ADSC	ADGAIN	BGID	TEMPC	ADCF	PGA	ENOB
2.3V	0	110101	01	01110000	250K	1	18.3
	0	000001	01	01110000	250K	64	17.3
2.5V	0	110101	01	01110000	250K	1	18.3
	0	000001	01	01110000	250K	64	17.3
2.8V	0	110101	01	01110000	250K	1	18.3
	0	000001	01	01110000	250K	64	17.3
	1	000001	01	01110000	125K	128	17.4
3.0V	0	110101	01	01110000	250K	1	18.3
	0	000001	01	01110000	250K	64	17.3
	1	000001	01	01110000	125K	128	17.4

注：

- 1.信号源内阻为 1KΩ，ADM=111，信号测试范围 0mV~5mV
- 2.ENOB 计算中所选取的 AD 个数为连续的 1024 个 AD 值，ENOB 结果不包含符号位。所有 ENOB 数据为单一样片测试结果，对批量生产只起参考作用，实际应用中，由于传感器及芯片的批次不同，精度将有所偏差。

### 3.5.3 ADC 输出速率

表 3-8 ADC 输出速率选择列表

ADM[2:0]	ADC 输出速率 (ADCF 参照表 3-7 ADC 增益选择列表)
000	ADCF/64
001	ADCF/128
010	ADCF/256
011	ADCF/512
100	ADCF/1024
101	ADCF/2048
110	ADCF/4096
111	ADCF/8192

注：ADCF 的大小请参照错误！未找到引用源。

### 3.5.4 ADC 工作电流

NETF 寄存器 (地址为 1DH)

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

注：详细使用参见错误！未找到引用源。

### 3.5.5 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[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

注：详细使用参见表 3-7 ADC 增益选择列表



### 3.6 LCD Driver

#### 3.6.1 LCD 控制模式

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

表 3-9 LCD 的 duty 选择列表

LCD_DUTY[1:0]	控制模式	SEG1-16							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
01	1/2duty					-	-	COM2	COM1
10	1/3duty					-	COM3	COM2	COM1
11	1/4duty					COM4	COM3	COM2	COM1

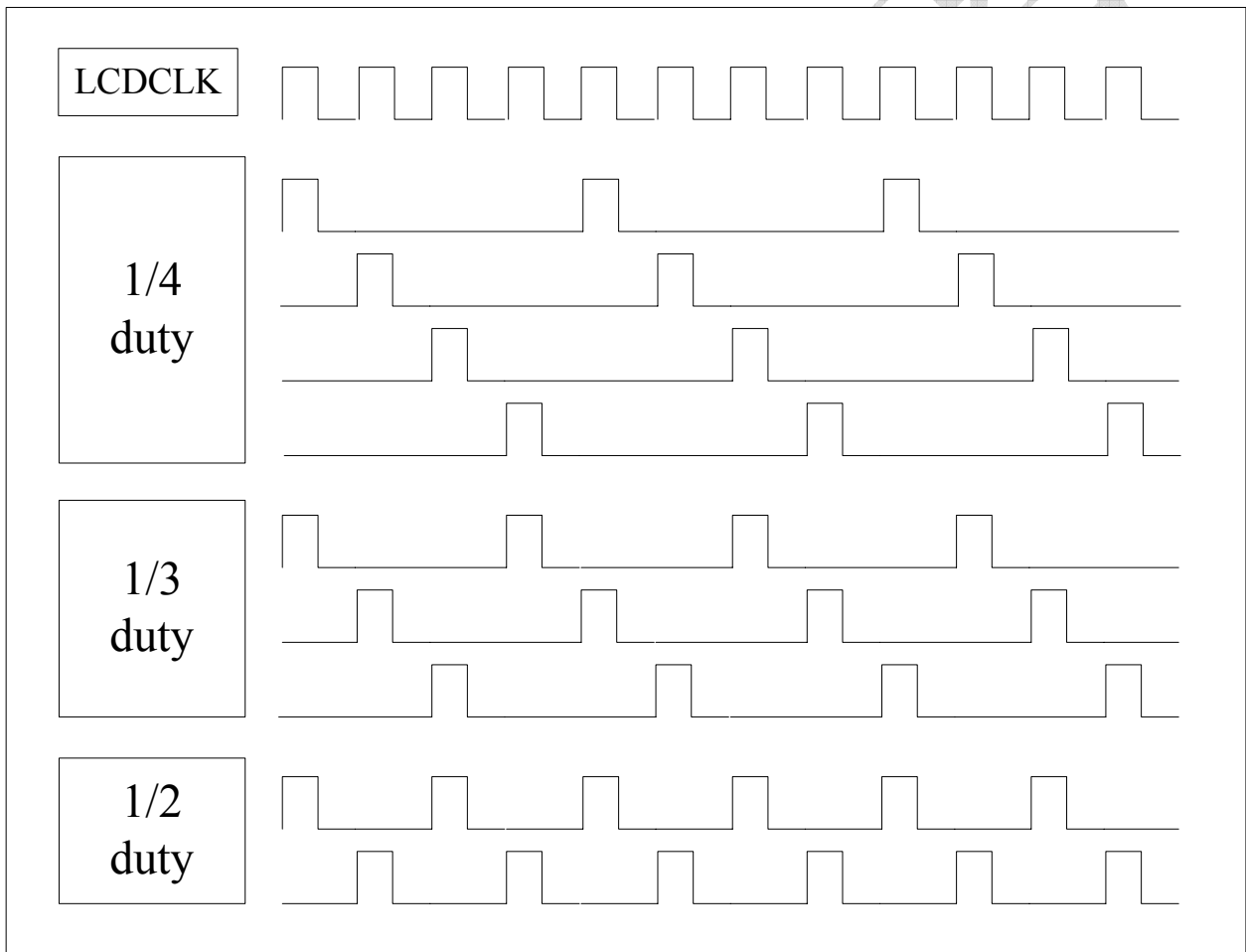


图 3-6 LCD 的 duty 模式工作周期

3.6.2 LCD 帧频选择

LCD 的帧频率可以通过设置寄存器标志 LCDCKS[1:0]确定。CSU8RP1185 对 LCD 模块的输入时钟进行分频以获得 LCDCK。

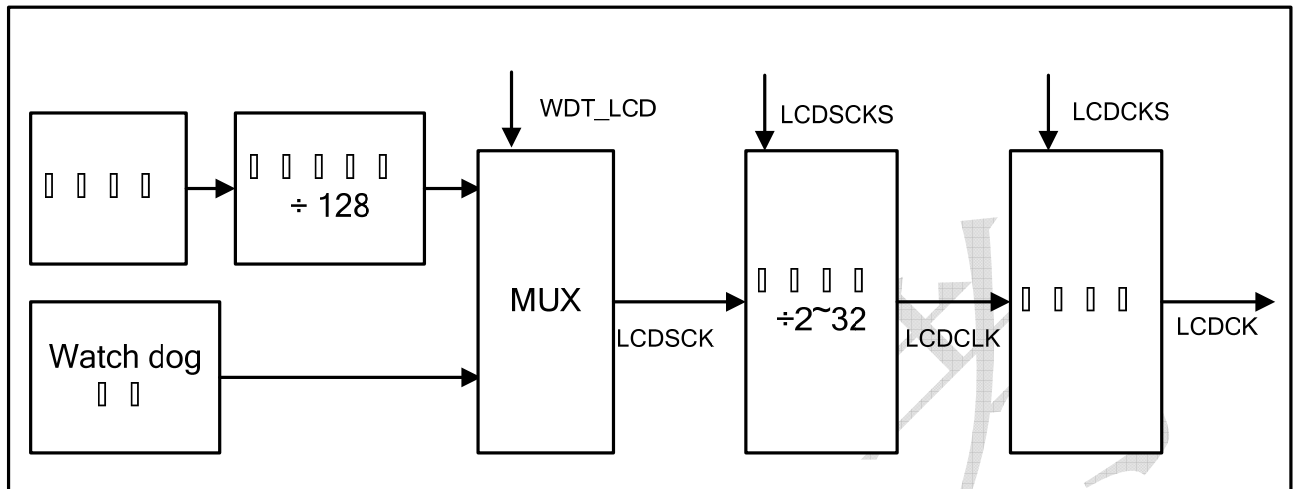


图 3-7 LCD 帧频时钟选择

表 3-10 LCDCLK 选择列表

Wdt lcd	LCDSCKS					LCDCLK(KHz)	
0	0	0	0	0	LCDCK/32	1	
0	0	0	0	1	LCDCK/30	1.067	
0	0	0	1	0	LCDCK/28	1.143	
0	0	0	1	1	LCDCK/26	1.231	
0	0	1	0	0	LCDCK/24	1.333	
0	0	1	0	1	LCDCK/22	1.455	
0	0	1	1	0	LCDCK/20	1.6	
0	0	1	1	1	LCDCK/18	1.778	
0	1	0	0	0	LCDCK/16	2	
0	1	0	0	1	LCDCK/14	2.286	
0	1	0	1	0	LCDCK/12	2.667	
0	1	0	1	1	LCDCK/10	3.2	
0	1	1	0	0	LCDCK/8	4	
0	1	1	0	1	LCDCK/6	5.333	
0	1	1	1	0	LCDCK/4	8	
0	1	1	1	1	LCDCK/2	16	
1	0	0	0	0	WTDCLK/32	0.094	
1	0	0	0	1	WTDCLK/30	0.1	
1	0	0	1	0	WTDCLK/28	0.107	
1	0	0	1	1	WTDCLK/26	0.115	
1	0	1	0	0	WTDCLK/24	0.125	
1	0	1	0	1	WTDCLK/22	0.136	
1	0	1	1	0	WTDCLK/20	0.15	
1	0	1	1	1	WTDCLK/18	0.167	
1	1	0	0	0	WTDCLK/16	0.188	
1	1	0	0	1	WTDCLK/14	0.214	
1	1	0	1	0	WTDCLK/12	0.25	
1	1	0	1	1	WTDCLK/10	0.3	
1	1	1	0	0	WTDCLK/8	0.375	
1	1	1	0	1	WTDCLK/6	0.5	
1	1	1	1	0	WTDCLK/4	0.75	
1	1	1	1	1	WTDCLK/2	1.5	

表 3-11 LCD 帧频选择列表

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

### 3.6.3 LCD 偏置电压

LCD 驱动器有 3 个偏置电压，V1、V2 及 V3，有 2 种电源模式：1/3bias、1/2bias。偏置电压的产生电路采用内部电阻分压，优点是可以节省 V2 与 V1 pin 的外部电容，静态功耗的大小与分压电阻的阻值有关，分压电阻越大驱动能力越弱。

采用电阻分压的方式产生偏置电压

- 1/3bias 电源系统

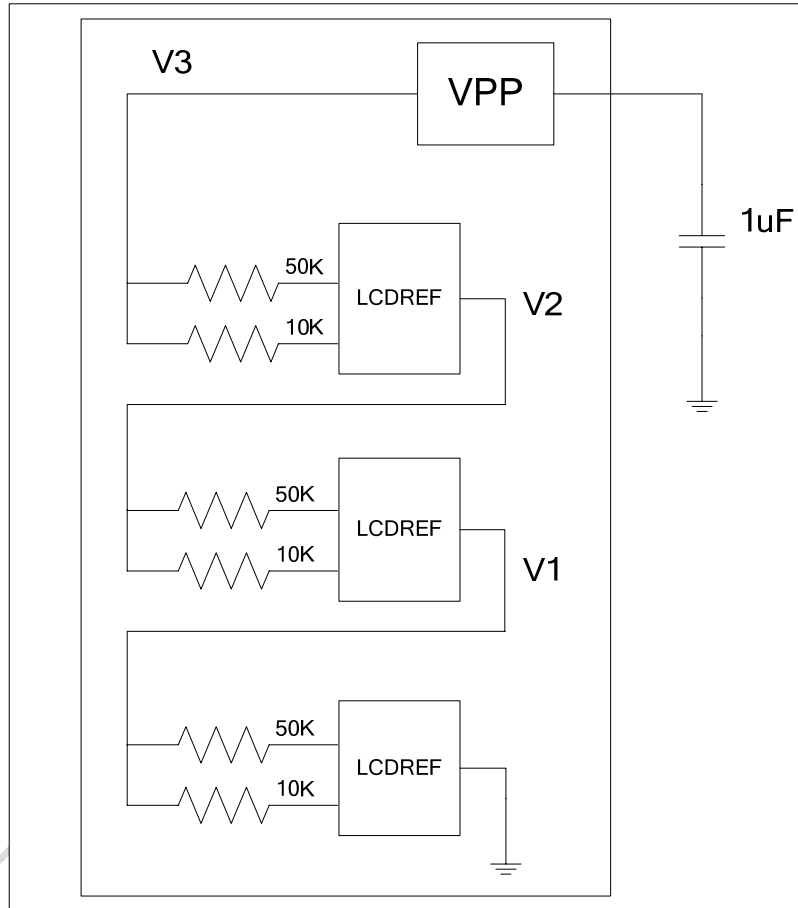


图 3-8 LCD 的 1/3bias 电源系统电路连接图（电阻分压）

● 1/2bias 电源系统

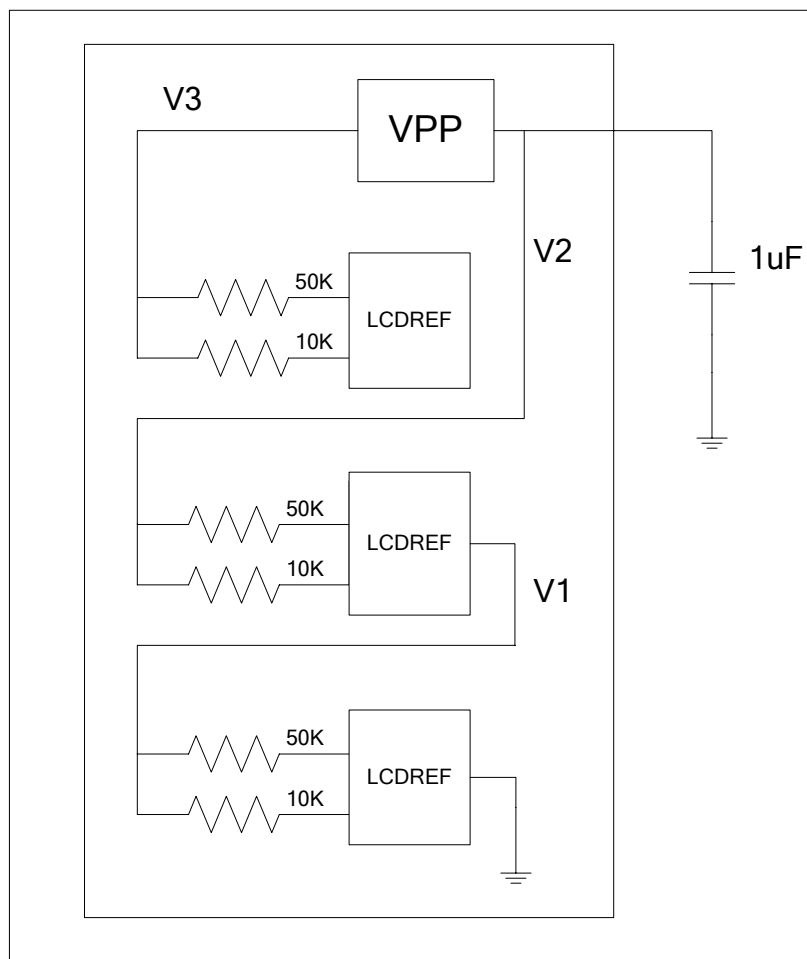


图 3-9 LCD 的 1/2bias 电源系统电路连接图（电阻分压）

3.6.4 LCD 驱动波形

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

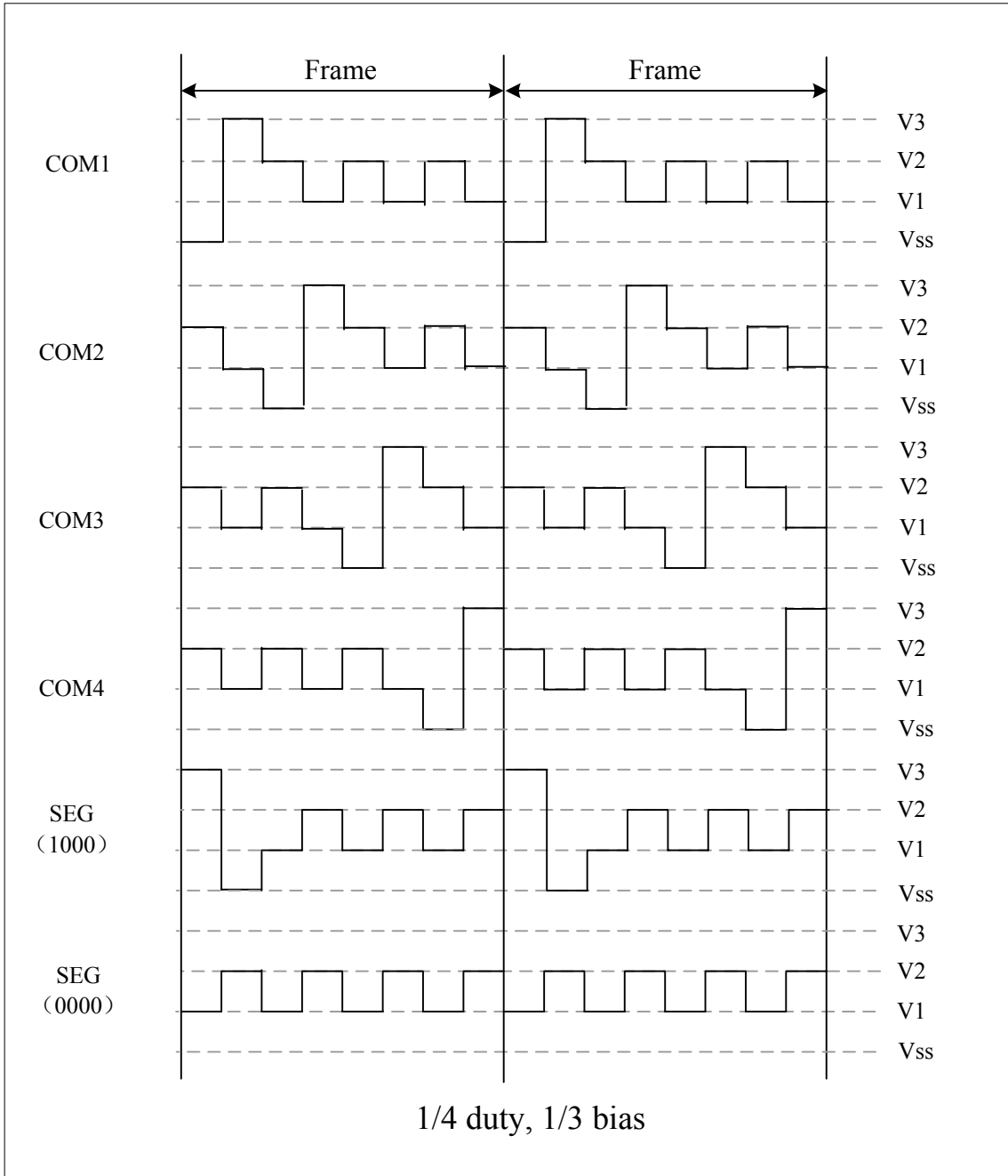


图 3-10 LCD 的 1/4duty 1/3bias 电源系统的时钟（A 波形）

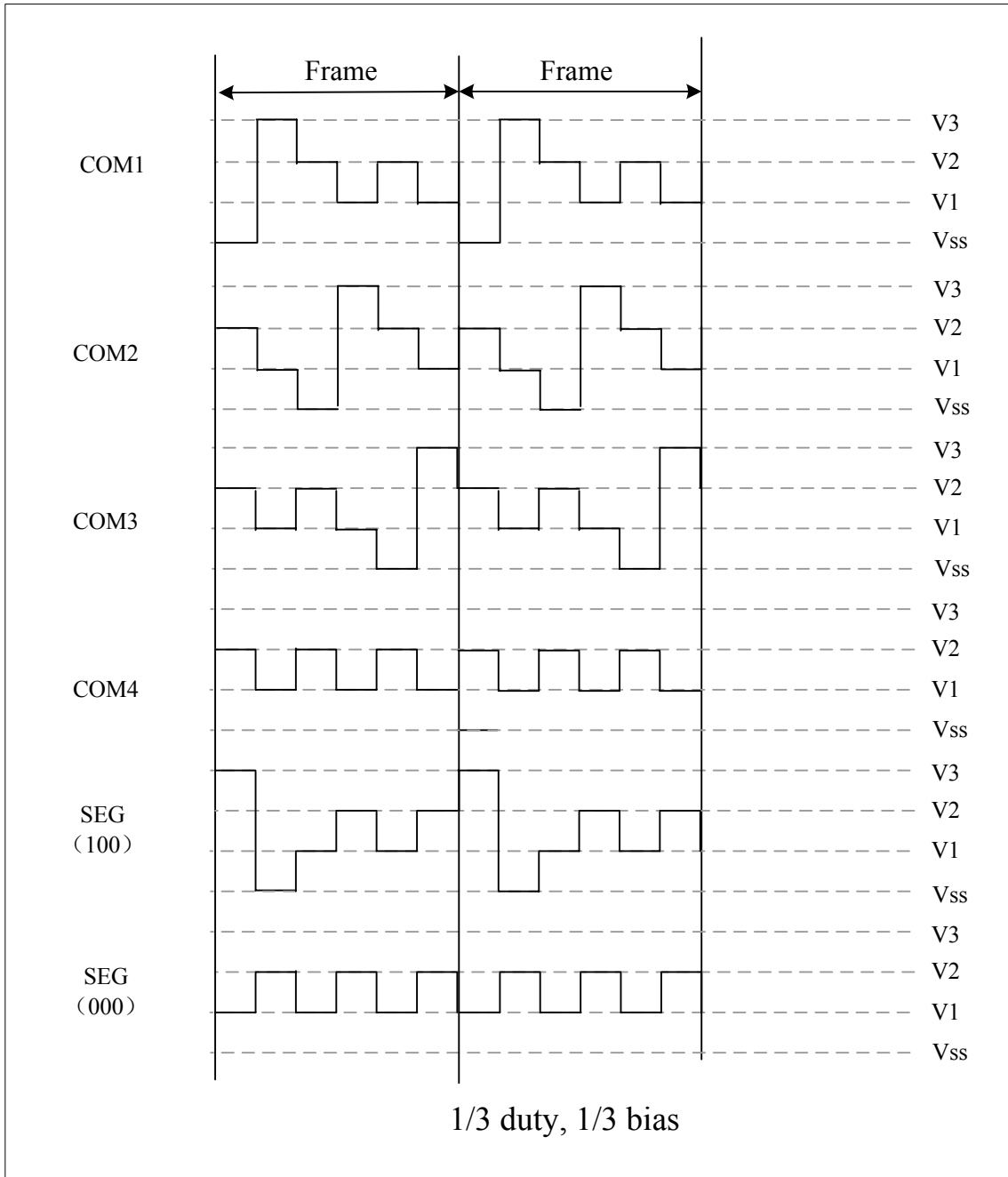


图 3-11 LCD 的 1/3duty 1/3bias 电源系统的时钟（A 波形）

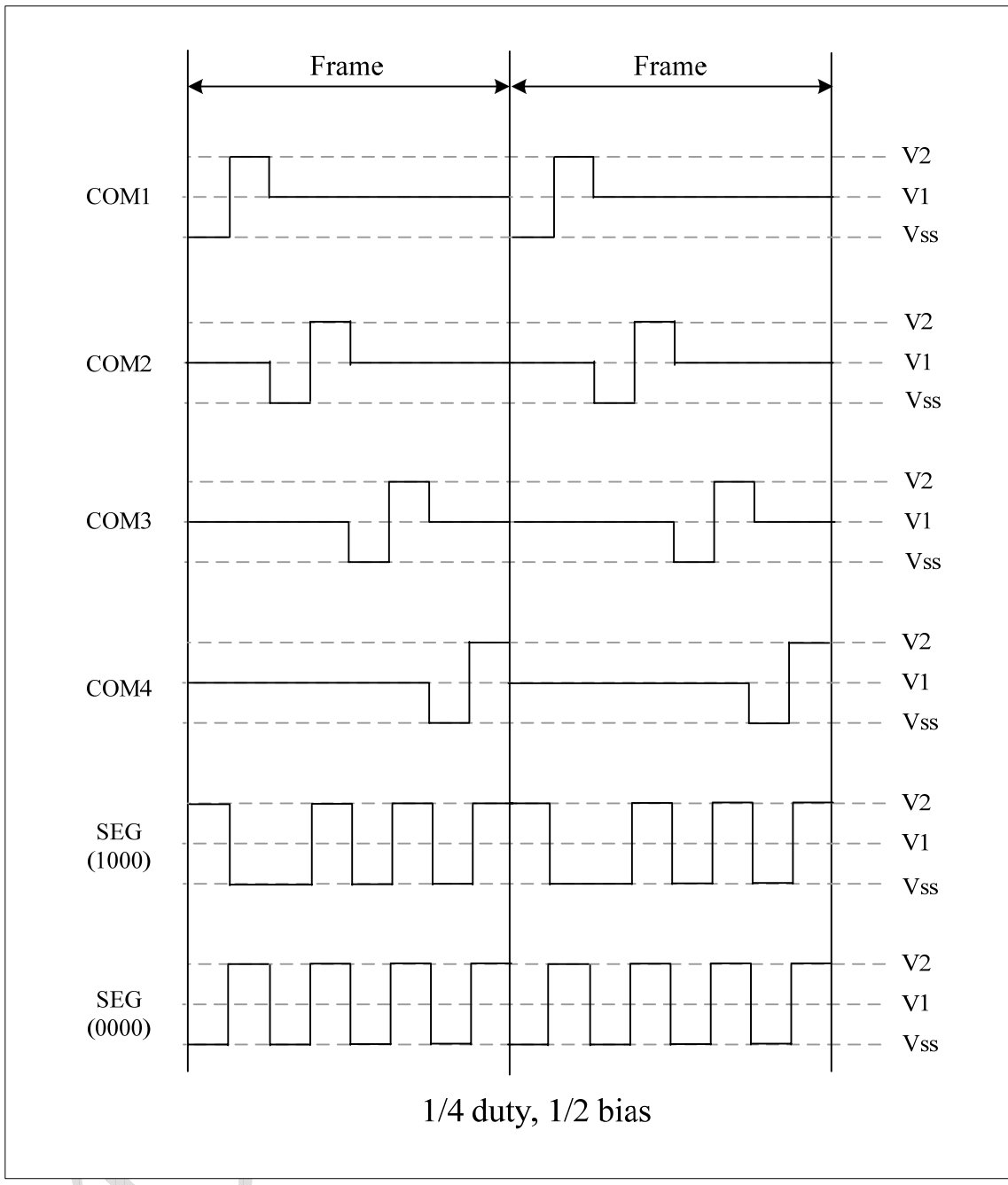


图 3-12 LCD 的 1/4duty 1/2bias 电源系统的时钟 (A 波形)



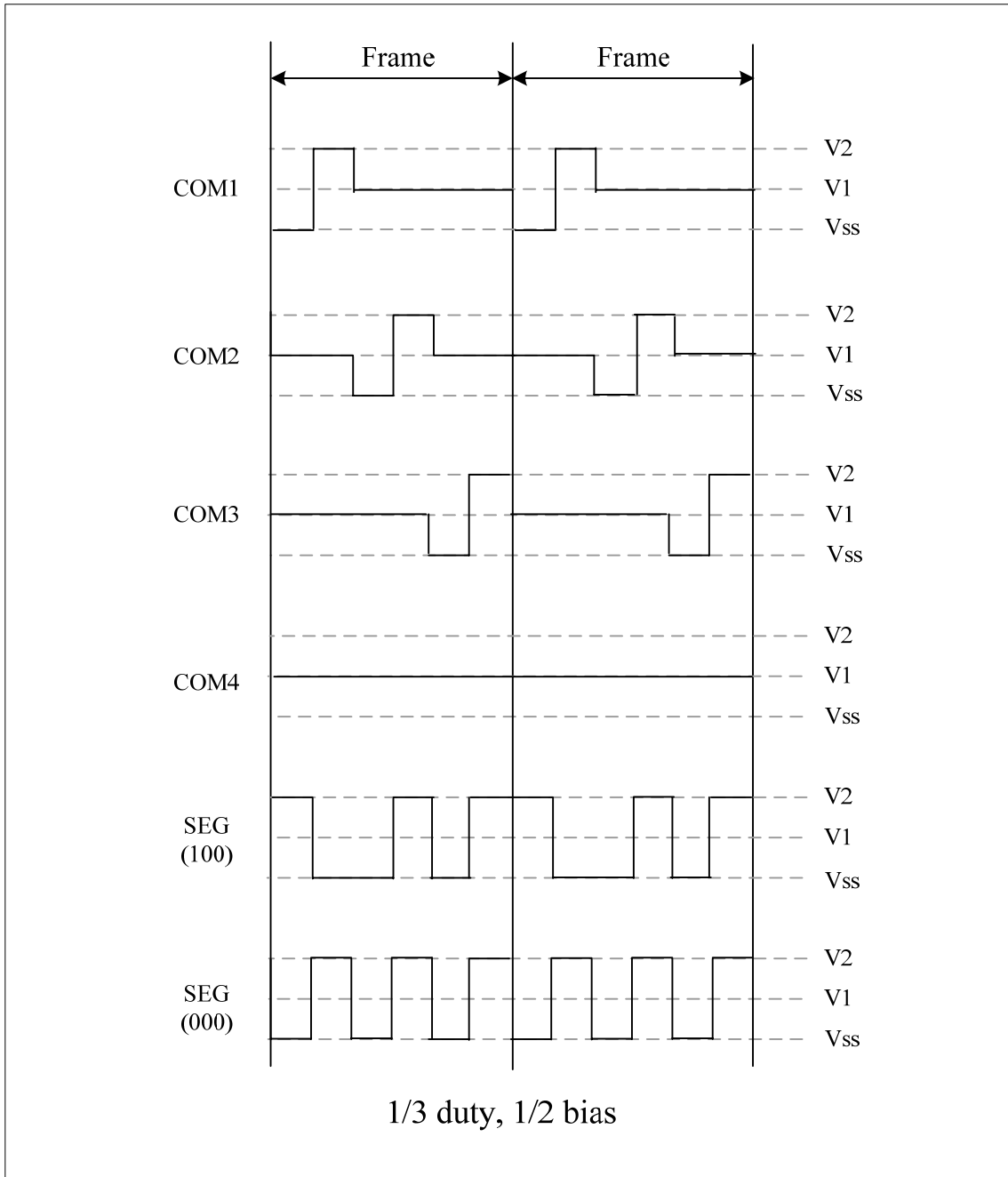


图 3-13 LCD 的 1/3duty 1/2bias 电源系统的时钟（A 波形）

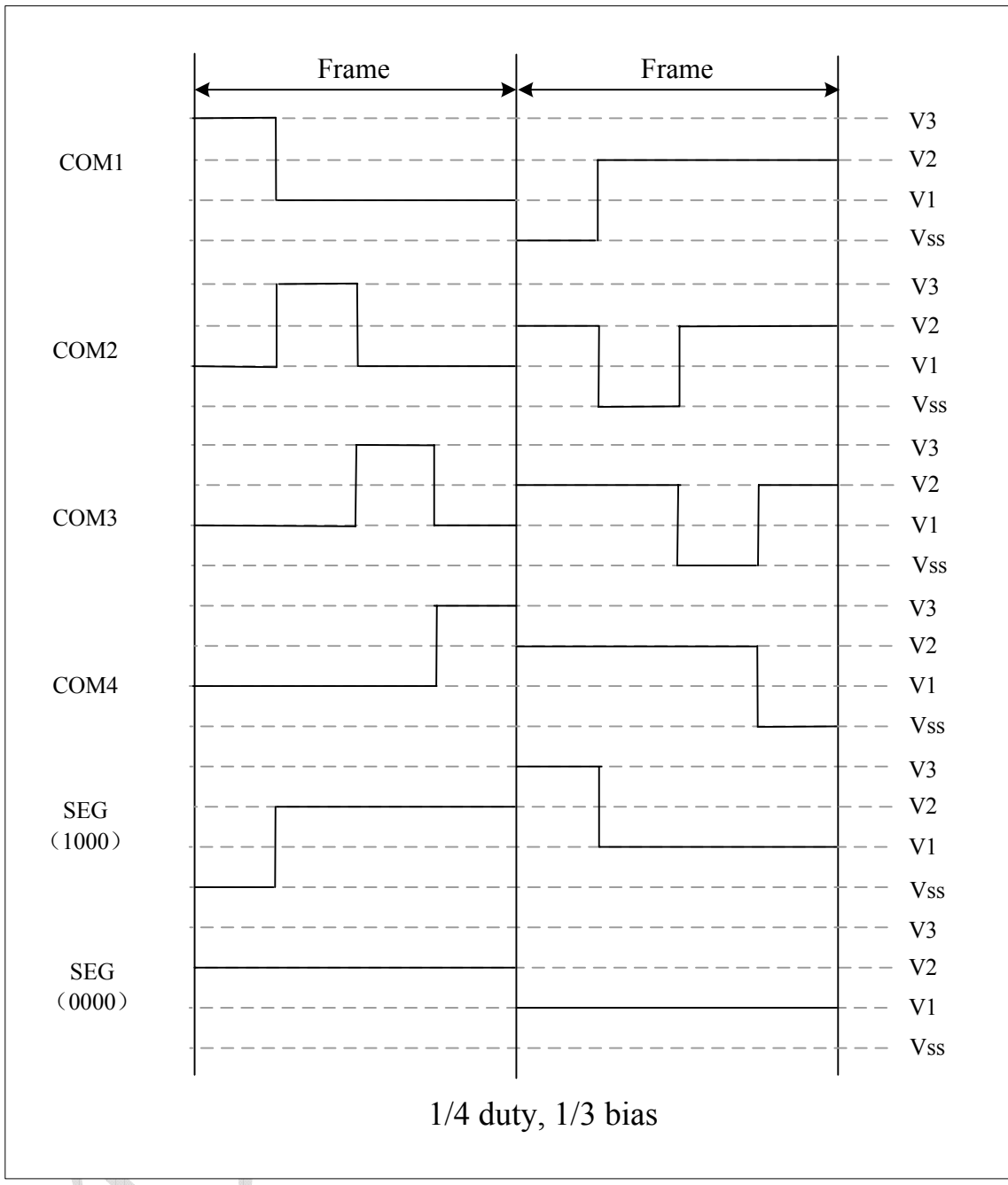


图 3-14 LCD 的 1/4duty 1/3bias 电源系统的时钟 (B 波形)

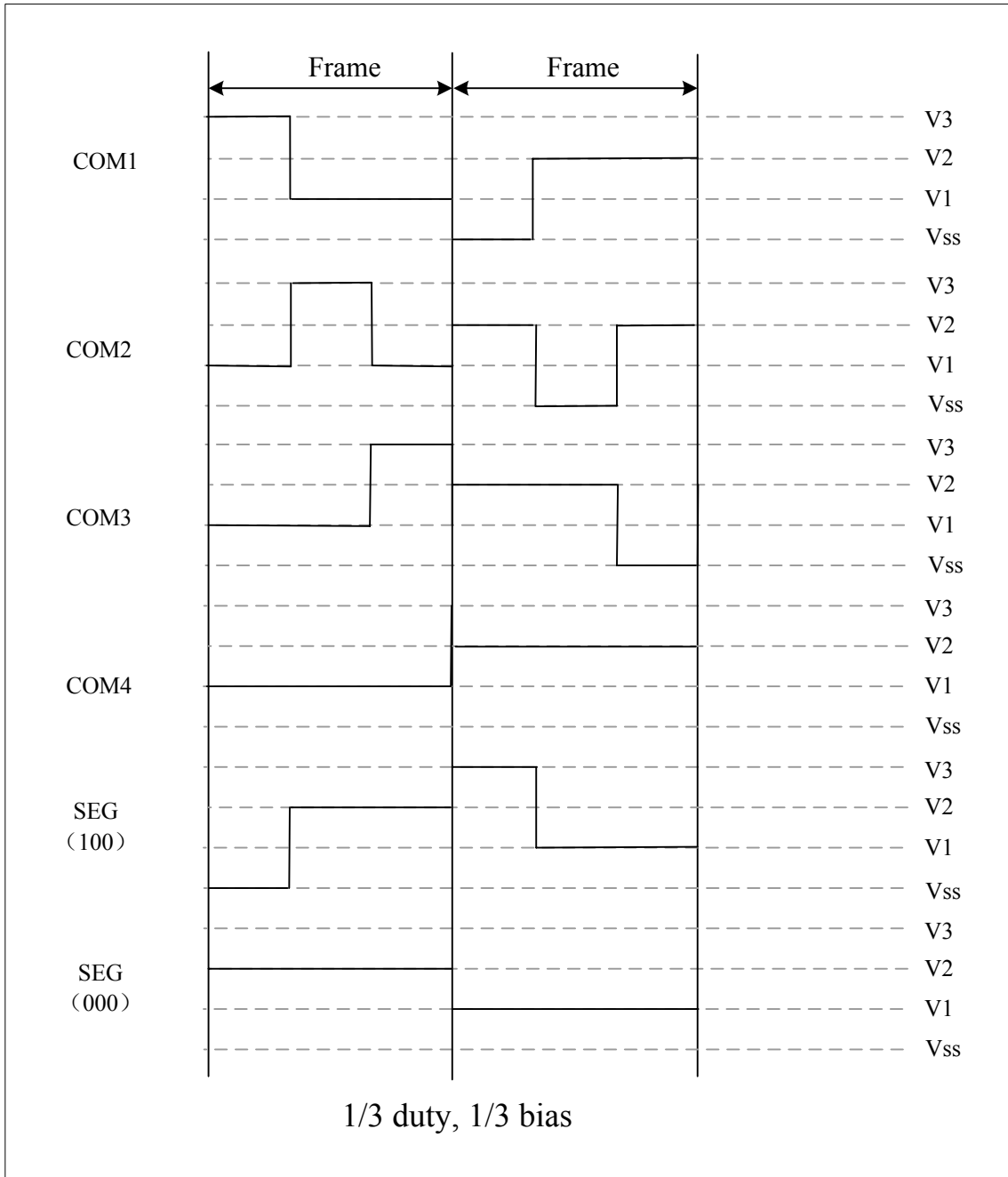


图 3-15 LCD 的 1/3duty 1/3bias 电源系统的时钟 (B 波形)

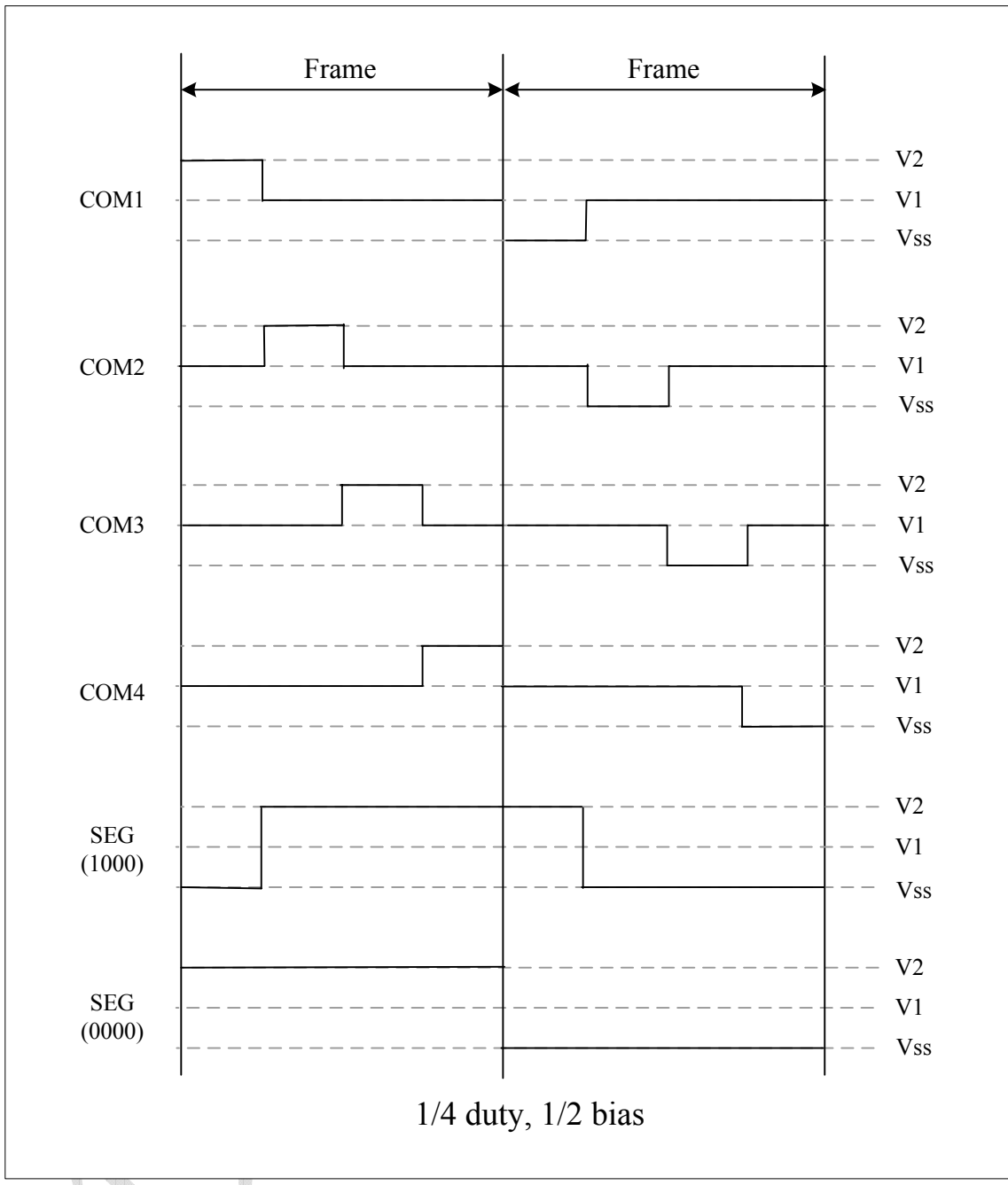


图 3-16 LCD 的 1/4duty 1/2bias 电源系统的时钟 (B 波形)

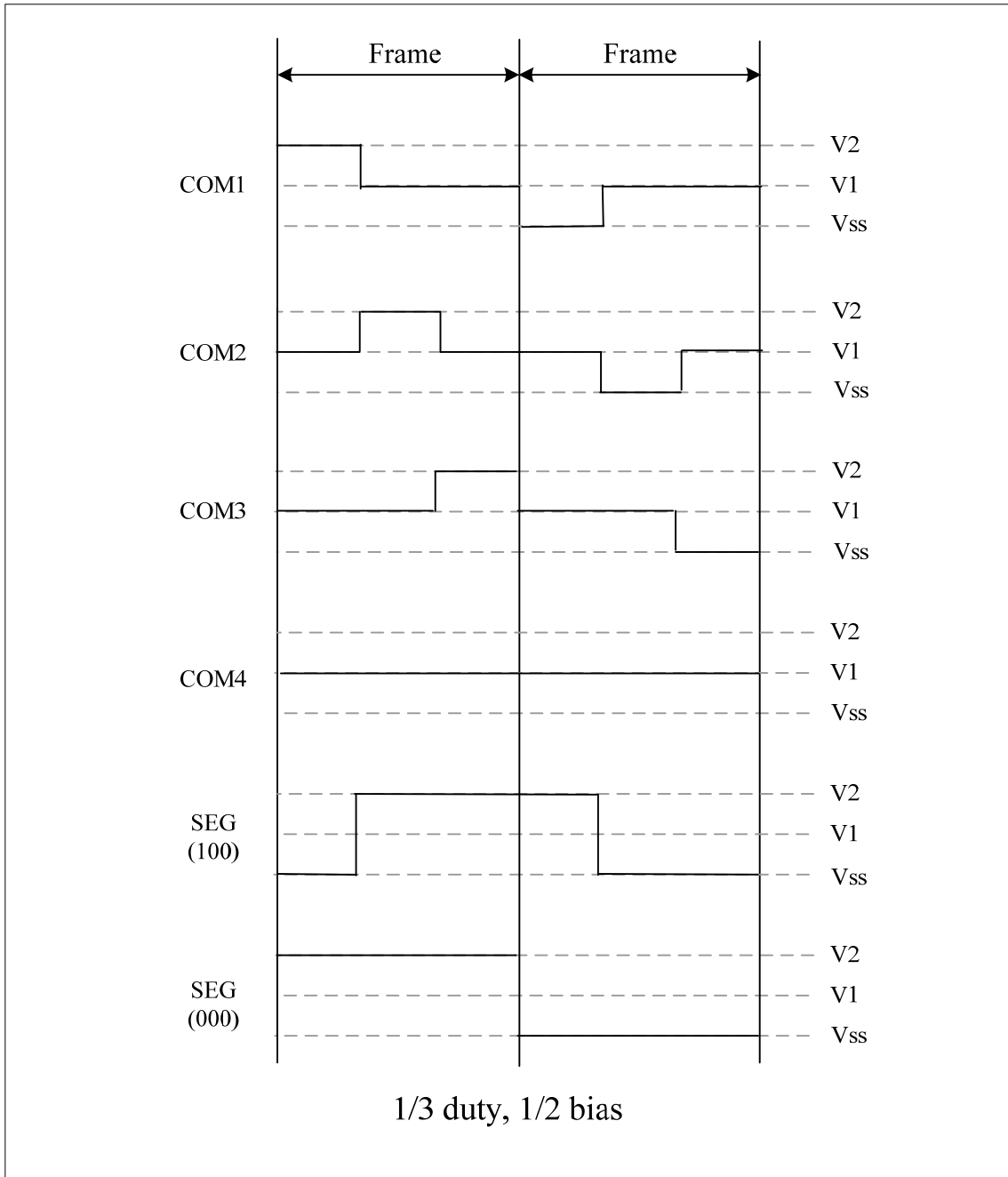


图 3-17 LCD 的 1/3duty 1/2bias 电源系统的时钟（B 波形）

3.6.5 LCD 寄存器说明

表 3-12 CSU8RP1185 的 LCD 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
15H	PCK		LCDSCK[3:0]							00000000
1BH	NETD					VLCDX[1:0]		LCDREF[1:0]		00000000
40H	LCD1						SEG1[3:0]			00000000
41H	LCD2						SEG2[3:0]			00000000
42H	LCD3						SEG3[3:0]			00000000
43H	LCD4						SEG4[3:0]			00000000
44H	LCD5						SEG5[3:0]			00000000
45H	LCD6						SEG6[3:0]			00000000
46H	LCD7						SEG7[3:0]			00000000
47H	LCD8						SEG8[3:0]			00000000
48H	LCD9						SEG9[3:0]			00000000
49H	LCD10						SEG10[3:0]			00000000
4AH	LCD11						SEG11[3:0]			00000000
4BH	LCD12						SEG12[3:0]			00000000
58H	LCDENR	LCDCK[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL	000u0000

NETD 寄存器（地址为 1BH）

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	R/W-X
NETD					VLCDX[1:0]		LCDREF[1:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit3-2 VLCDX: VLCD 输出电压选择

- 00 = 保留
- 01 = 2.8V
- 10 = 3.0V
- 11 = 3.2V

Bit1-0 LCDREF: LCD V1/V2 产生电路分压电阻选择

- 0X=保留
- 10 = 50Kohm
- 11 = 10Kohm

LCD1 寄存器（地址为 40H）

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	R/W-X
LCD1					SEG1[3:0]				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

- Bit 3-0 SEG1[3]: LCD 驱动器控制信号: SEG1 带 COM4。
- SEG1[2]: LCD 驱动器控制信号: SEG1 带 COM3。
- SEG1[1]: LCD 驱动器控制信号: SEG1 带 COM2。
- SEG1[0]: LCD 驱动器控制信号: SEG1 带 COM1。

LCD2 寄存器（地址为 41H）

~

LCD3 寄存器（地址为 42H）

~

LCD12 寄存器（地址为 4BH）

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
LCD12	SEG12[3:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 3-0 SEG12[3]: LCD 驱动器控制信号: SEG12 带 COM4。  
 SEG12[2]: LCD 驱动器控制信号: SEG12 带 COM3。  
 SEG12[1]: LCD 驱动器控制信号: SEG12 带 COM2。  
 SEG12[0]: LCD 驱动器控制信号: SEG12 带 COM1。

LCDENR 寄存器（58H）

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
LCDENR	LCDCKS[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

- Bit 7-6 LCDCKS[1:0]: LCD 帧频选择器  
 11 = LCD 的帧频是 LCD 输入时钟频率的 1/32  
 10 = LCD 的帧频是 LCD 输入时钟频率的 1/16  
 01 = LCD 的帧频是 LCD 输入时钟频率的 1/8  
 00 = LCD 的帧频是 LCD 输入时钟频率的 1/4
- Bit 5 LCDEN: LCD 驱动器使能标志  
 1 = LCD 驱动器使能。LCD 的时钟被启动  
 0 = LCD 驱动器不使能。LCD 的时钟被停止
- Bit 4 LCDWS: LCD 波形选择  
 1 = 波形 B  
 0 = 波形 A
- Bit 3 LEVEL: LCD 驱动器的偏置电压选择器  
 0 = LCD 驱动器的偏置电压是 1/3bias  
 1 = LCD 驱动器的偏置电压是 1/2bias
- Bit 2-1 LCD\_DUTY[1:0]: LCD 驱动器控制模式（SEG duty 周期）  
 11 = LCD 驱动器控制模式是 1/4duty 周期模式  
 10 = LCD 驱动器控制模式是 1/3duty 周期模式  
 01 = LCD 驱动器控制模式是 1/2duty 周期模式  
 00 = 不可用
- Bit 0 ENPMPL: 电荷泵 使能标志位  
 1 = 电荷泵 打开（此时 ENVB 必须置 1，否则 PUMP 无法正常工作）  
 0 = 电荷泵 关闭

### 3.6.6 LCD 操作步骤

LCD 的操作:

1. 将段接口连接到 LCD 面板。
2. 设置寄存器标志 LEVEL 选择 LCD 驱动器电源系统。（0 = 1/3bias, 1 = 1/2bias）
3. 如果使用芯片提供 LCD 的电源时，设置 ENPMPL 使能 LCD 电荷泵。（同时必须打开 ENVB）  
 如果使用外部电源提供 LCD 的电源时，将外部电源与 VLCD 相连即可，不需要打开 LCD 电荷泵。
4. 选择 LCD 输入时钟的频率。
5. 设置寄存器标志 LCD\_DUTY[1:0]，选择控制模式。（SEG duty 周期）

表 3-13 LCD 的 duty 控制模式选择列表

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

6. 置位 LCDEN 以使能 LCD 驱动器

仅供参考



### 3.7 OTP 模块

OTP 烧写器的接口：

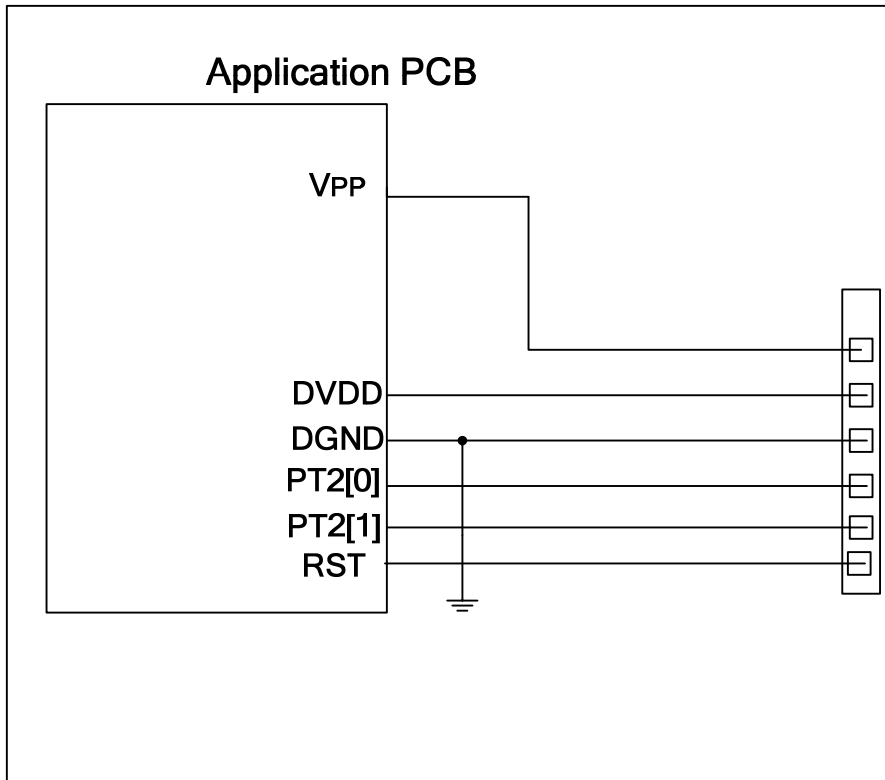


图 3-18 OTP 烧写器接口图

表 3-14 OTP 接口说明

端口名称	说明	备注
VPP	烧录电压	
DVDD	电源正端	
DGND	电源负端	
PT2[0]	数据端口	
PT2[1]	数据端口	
Reset	控制端口	

### 3.8 OTP 在线烧录

电路要求：在在线烧录时，VPP 必须采用内部的电荷泵。

地址要求：CSU8RP1185 芯片只有从 0800H 至 087FH 共 128 bytes 地址可以作为数据存储区。通过 PARL 寄存器来选择地址。例如当 PARL[6:0]为 00H 时选择对 0800H 地址进行烧录和读取。当 PARL[6:0]为 7FH 时选择对 087FH 地址进行烧录和读取

表 3-15 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
0BH	EADRL	PARL[6:0]								00000000
19H	NETB					ERV				00000000

EADRL: 提供 OTP 在线烧录或者在线读 OTP 的七位地址。

Work: 提供 OTP 在线烧录时的烧录数据,读 OTP 时读出的数据。

ERV: 当 VPP 引脚电压达到烧录电压时, ERV 置高。

操作方式:

在线烧录 OTP 时

1. 把基准打开 (ENVB 置高)
2. 将 VLCDX 置为 11
3. 将电荷泵使能信号打开 (ENPMPL 置高)
4. 将 CHP\_VPP 置高, 然后等待 200ms 时间检测 ERV 寄存器
5. 检查 ERV 电压值是否达到烧录电压, 达到烧录电压后延时 50ms。
6. 将烧录地址写入 EADRL 寄存器。
7. 将烧录的数据写入 work 寄存器。
8. 用在线烧录指令 (TBLP) 烧录。烧录指令中的时间选择 (k) 参见表 3-16 在线烧录时间选择寄存器
9. 烧录完一个 Byte 后, 必须从步骤 5 开始烧录下一个 Byte。

表 3-16 在线烧录时间选择寄存器

M2_CK	M1_CK	时钟源(KHz)		k (十进制)
1	0	ICK	4000	255
1	1	ICK	4000	130

在线读 OTP 数据时

1. 将读 OTP 地址的低八位写入 EADRL 寄存器。
2. 用在线读 OTP 指令 (MOVP) 读出 OTP 数据, 执行该指令后, 数据存放在 work 寄存器。

主要有两条指令:

TBLP k

MOVP

其中 TBLP k 是将寄存器 work 中的数据写到以 EADRL 的内容作为 OTP 的写地址中, 烧录时间是 K 个指令周期

MOVP 是将 EADRL 的内容作为 OTP 的读地址, 读出的数据放到寄存器 work 中。

## 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
ADDPCW	$[PC] \leftarrow [PC] + 1 + [W]$	1	~
ADDWF f,d	$[Destination] \leftarrow [f] + [W]$	1	C,DC,Z
ADDWFC f,d	$[Destination] \leftarrow [f] + [W] + C$	1	C,DC,Z
ANDLW k	$[W] \leftarrow [W] \text{ AND } k$	1	Z
ANDWF f,d	$[Destination] \leftarrow [W] \text{ AND } [f]$	1	Z
BCF f,b	$[f < b >] \leftarrow 0$	1	~
BSF f,b	$[f < b >] \leftarrow 1$	1	~
BTFSC f,b	Jump if $[f < b >] = 0$	1	~
BTFSS f,b	Jump if $[f < b >] = 1$	1	~
CALL k	Push PC+1 and Goto K	1	~
CLRF f	$[f] \leftarrow 0$	1	Z
CLRWDT	Clear watch dog timer	1	~
COMF f,d	$[f] \leftarrow \text{NOT}([f])$	1	Z
DECF f,d	$[Destination] \leftarrow [f] - 1$	1	Z
DECFSZ f,d	$[Destination] \leftarrow [f] - 1$ , jump if the result is zero	1	~
GOTO k	$PC \leftarrow k$	1	~
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	~
IORLW k	$[W] \leftarrow [W] \text{ OR } k$	1	Z
IORWF f,d	$[Destination] \leftarrow [W] \text{ OR } [f]$	1	Z
MOVP	$[EADRH, EADRL] \rightarrow [EDATH, WORK]$	2	~
MOVFW f	$[W] \leftarrow [f]$	1	~
MOVLW k	$[W] \leftarrow k$	1	~
MOVWF k	$[f] \leftarrow [W]$	1	~
NOP	No operation	1	~
RETFIE	Pop PC and GIE = 1	1	~
RETLW k	RETURN and W=k	1	~
RETURN	POP PC	1	~
RLF f,d	$[Destination < n+1 >] \leftarrow [f < n >]$	1	C,Z
RRF f,d	$[Destination < n-1 >] \leftarrow [f < n >]$	1	C,Z
SLEEP	STOP OSC	1	PD
SUBLW k	$[W] \leftarrow k - [W]$	1	C,DC,Z
SUBWF f,d	$[Destinnation] \leftarrow [f] - [W]$	1	C,DC,Z
SUBWFC f,d	$[Destinnation] \leftarrow [f] - [W] + C$	1	C,DC,Z
TBLP k	$[EADRH, EADRL] \leftarrow [EDATH, WORK]$	$2 * k + 1$	~
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(00H ~ FFh)

W: 工作寄存器

k: 立即数

d: 目标地址选择: d=0 结果保存在工作寄存器, d=1: 结果保存在数据存储器 f 单元

b: 位选择(0~7)

[f]: f 地址的内容

PC: 程序计数器

C: 进位标志

DC: 半加进位标志

Z: 结果为零标志

PD: 睡眠标志位

TO:看门狗溢出标志  
WDT:看门狗计数器

表 4-2 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	1
例子 1 ADDPCW	在指令执行之前: W=7FH, PC=0212H 指令执行之后: PC=0292H
例子 2 ADDPCW	在指令执行之前: W=80H, PC=0212H 指令执行之后: PC=0193H
例子 3 ADDPCW	在指令执行之前: W=FEH, PC=0212H 指令执行之后: PC=0211H

3

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

4

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

5

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

6

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

7

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

8

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

9

BTFS	如果 bit 测试为 0 则跳转
指令格式	BTFS f, b 0<=f<=FFH 0<=b<=7
操作	Skip if (f[b])=0
标志位	无
描述	如果 f 的 bit 位是 0, 下一条取到的指令将被丢到, 然后执行一条空指令组成一个两周期的指令。
周期	1
例子 NODE BTFS 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

BTFS	如果 bit 测试为 1, 则跳转
指令格式	BTFS f, b 0<=f<=FFH 0<=b<=7
操作	Skip if (f[b])=1
标志位	无
描述	如果 f 的 bit 位是 1, 下一条取到的指令将被丢到, 然后执行一条空指令组成一个两周期的指令。
周期	1
例子 NODE BTFS 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<=1FFFH
操作	(top stack)←PC+1 PC←K
标志位	无
描述	子程序调用, 先将 PC+1 压入堆栈, 然后把立即数地址下载到 PC 中。
周期	1

12

CLRF	清除 f
指令格式	CLRF f 0<=f<=FFH
操作	(f)<-0
标志位	Z
描述	将 f 的内容清零
周期	1
例子 CLRF WORK	在指令执行之前 WORK=5AH 在指令执行之后 WORK=00H

\*注。当 clrf status 寄存器时，标志位 Z 不会置高

13

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

14

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

15

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

16

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

17

GOTO	无条件跳转
指令格式	GOTO K 0<=K<=1FFFH
操作	PC<-K
标志位	无
描述	立即地址载入 PC
周期	1

18

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

19

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



20

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

21

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

22

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

23

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

24

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

25

MOVP	将 OTP 中的 {EADRH, EADRL} 的数据读出放入 {EDATH, WORK} 中
指令格式	MOVP K
操作	{EDATH, WORK}<←({EADRH, EADRL})
标志位	无
描述	将 OTP 中的 {EADRH, EADRL} 的数据读出放入 {EDATH, WORK} 中
周期	2
例子 MOVP	在指令执行之前 EPROM: 03H: 07H EADRL =03H 在指令执行之后 W=07H

26

MOVWF	将工作寄存器的值传送到 f 中
指令格式	MOVWF f 0<=f<=FFH
操作	(f)<←(W)
标志位	无
描述	将工作寄存器的值传送到 f 中
周期	1
例子 MOVWF f	在指令执行之前 W=88H f=23H 在指令执行之后 W=88H f=88H

27

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

28

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	1

29

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	1

30

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	1

31

RLF	带进位左移
指令格式	RLF f, d 0<=f<=FFH d=0,1
操作	(目标地址[n+1])<←(f[n]) (目标地址[0])<←C C<←(f[7])
标志位	C, Z
描述	F 带进位左移一位 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 RLF f, 1	在指令执行之前 C=0 W=88H f=E6H 在指令执行之后 C=1 W=88H f=CCH

32

RRF	带进位右移
指令格式	RRF f, d 0<=f<=FFH d=0,1
操作	(目标地址[n-1])<←(f[n]) (目标地址[7])<←C C<←(f[7])
标志位	C
描述	F 带进位右移一位 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 RRF f, 0	在指令执行之前 C=0 W=88H f=95H 在指令执行之后 C=1 W=4AH f=95H

33

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

34

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)

35

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

36

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

37

TBLP	将 OTP 的 {EADRH, EADRL} 的地址写入 {EDATH, WORK} 中的数据
指令格式	TBLP k
操作	OTP({EADRH, EADRL})<←({EDATH, WORK})
标志位	无
描述	将 OTP 的 {EADRH, EADRL} 的地址写入 {EDATH, WORK} 中的数据
周期	与工作时钟有关
例子 TBLP 25	在指令执行之前 OTP memory: 17H= FFH EADRL = 17H WORK = 05H 在指令执行之后 OTP memory: 17H = 05H

38

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

39

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

## 5 电气特性

### 5.1 最大极限值

表 5-1 CSU8RP1185 最大极限值

参数	范围	单位
电源 DVDD,AVDD	2.4~3.6	V
引脚输入电压	-0.3~DVDD+0.3 -0.3~AVDD+0.3	V
工作温度	-40~+85	°C
存储温度	-55~+150	°C

### 5.2 直流特性 (DVDD, AVDD = 3.3V, T<sub>A</sub> = 25°C, 如无其他说明则都是此条件)

表 5-2 CSU8RP1185 直流特性

符号	参数	测试条件	最小值	典型值	最大值	单位
VDD	工作电源		2.4	3	3.6	V
IDD1	电源电流 1	指令周期 = 500KHz 电荷泵、ADC 打开		1.5		mA
ISleep	睡眠模式下电源电流	睡眠指令	1.2	1.5	3.5	uA
VIH	数字输入高电平	PT1,2,	VDD-0.6			V
VIL	数字输入低电平	PT1,2,			0.3	V
IPU	上拉电流	PT1,2 Vin = 0		30		uA
IOH	高电平输出电流	VOH=DVDD-0.3V		3		mA
IOL	低电平输出电流	VOL=0.3V		3		mA
IOH	高电平输出电流	VOH=DVDD-0.3V (PT2.2、PT2.3)		10		mA
IOL	低电平输出电流	VOL=0.3V (PT2.2、PT2.3)		10		mA
IREG	VS 稳压器输出电流	AVDD=3.3V VS=2.3V		4	8	mA
VLREF	用于低电压检测的内部参考电压			1.25		V
TCLREF	用于低电压检测的内部参考电压温度系数	T <sub>A</sub> = -40~80°C		50		ppm/°C
VLBAT	低电池检测电压	SILB[2:0]=000		2.4		V
		SILB[2:0]=001		2.5		
		SILB[2:0]=010		2.6		
		SILB[2:0]=011		2.7		
		SILB[2:0]=100		2.8		
		SILB[2:0]=101		3.6		
		SILB[2:0]=111		3.6		
FRC	内置 RC 振荡器		3.2	4.0	4.8	MHz
FWDT	内置看门狗时钟		1.6	3.2	4.8	KHz

5.3 ADC 的特性 (VREF=3V, TA = 25°C, 如无其他说明则都是此条件)

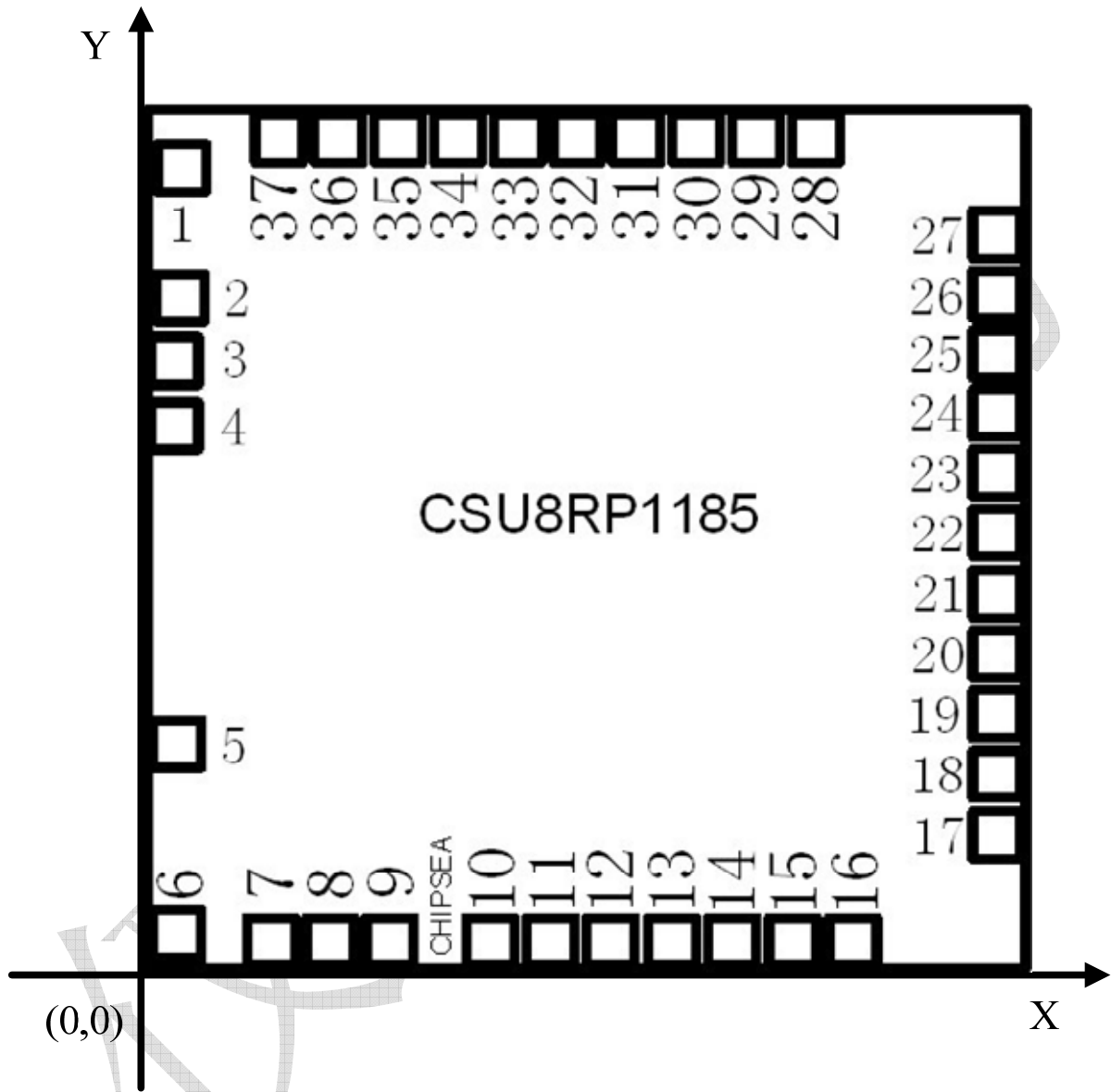
表 5-3 CSU8RP1185 ADC 的特性

参数		条件	最小值	典型值	最大值	单位
模拟输入	模拟输入范围		AGND-0.1		AVDD+0.1	V
	满幅输入电压 (AIN+) - (AIN-)				±VREF/PGA	V
	差分输入阻抗			16/PGA		MΩ
系统性能	分辨率	无失码		24		Bits
	输入噪声 (rms)	增益=1				uv
		增益=64				nV
	积分线性度	增益=64				% of FS
	失调误差	增益=64				uV
	失调误差漂移	增益=64				uV/°C
	增益误差	增益=64				%
增益误差漂移	增益=64				ppm/°C	
参考电压	VS	LDOS[1:0]= 11	2.2	2.3	2.4	V
		LDOS[1:0]= 10	2.4	2.5	2.6	
		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 Bonding 说明

### 6.1 PIN 排布



注意：衬底必须接到 DGND

37pin-dice

X= 1470um Y=1440um

Pad 大小 70um\*75um

其中 DVDD pad 上方有 B01 字样

## 6.2 Pin 坐标

No.	PAD Name	(x,y)	
1	DVDD	(59,1344.5)	
2	DGND	(57,1127.5)	
3	VPP	(47,1021.75)	
4	RST	(47,913)	
5	AGND	(52,367.58)	
6	AVDD	(51.64,65)	
7	VS/REF	(208.2,47)	
8	AIN0	(307.98,47)	
9	AIN1	(407.76,47)	
10	PT2<4>	(580.13,47)	
11	PT2<5>	(680.96,47)	
12	PT1<4>	(781.94,47)	
13	PT1<5>	(884.82,47)	
14	PT1<6>	(985.64,47)	
15	PT1<7>	(1086.45,47)	
16	PT2<0>	(1186.76,47)	
17	PT2<1>	(1423,225.94)	
18	PT2<2>	(1423,328.4)	
19	PT2<3>	(1423,429.22)	
20	PT2<6>	(1423,530.04)	
21	PT2<7>	(1423,630.86)	
22	SEG<12>	(1423,734.14)	
23	SEG<11>	(1423,833.92)	
24	SEG<10>	(1423,933.7)	
25	SEG<9>	(1423,1033.48)	
26	SEG<8>	(1423,1133.26)	
27	SEG<7>	(1423,1233.04)	
28	SEG<6>	(1119.75,1393)	
29	SEG<5>	(1019.97, 1393)	
30	SEG<4>	(920.19, 1393)	
31	SEG<3>	(820.41, 1393)	
32	SEG<2>	(720.63, 1393)	
33	SEG<1>	(620.86, 1393)	
34	COM<4>	(521.07, 1393)	
35	COM<3>	(421.29, 1393)	
36	COM<2>	(321.51, 1393)	
37	COM<1>	(221.73, 1393)	