



CSU1281用户手册

基于 OTP ROM的 8位 RISC MCU

Rev.1.8

通讯地址:深圳市福田区新闻路景苑大厦A1002-1003室
邮政编码:518029
公司电话:+(86 755)83063040
传真:+(86 755)83065035
公司网站: www.sictech.com.cn
企业QQ: 2742541955

版本历史

历史版本	修改说明	版本时间
REV1.0	创建文档	
REV1.1	1增加高速外部晶振功能，以及外部晶振起振时间（参见 2.2.1） 2增加 EO_SLP寄存器，可在 sleep模式下不关外部晶振（参见 2.2.1） 3修改了 VS的输出电压选择（参见 3.1.1）	
REV 1.2	修改 timer以及 LCD时钟系统的描述	
REV 1.3	修改了笔误	
REV1.4	1增加了 INTF寄存器说明（参见 2.1.3） 2增加了 IO口模块说明（参见 2.4）	
REV1.5	1.增加了 PT1.3（COM6）和 PT1.4(COM5)作为输出口的功能(参见 2.4.7) 2.修改了复位电路部分的说明(参见 5.4)	
REV1.6	1.取消了除 CSU1281之外的其他 CSU12系列型号	
REV1.7	增加了对 COM5和 COM6作为 PT1.4和 PT1.3使用时的说明，在使用此两个IO的输出功能时，PT1[7:3]寄存器不得使用 bsf和 bcf指令，否则 PT1.3和PT1.4的输出将被清零。(参见 2.4.7)。PT1PU[2]口即 TEST寄存器不得置 1，否则 LCD驱动电压将发生异常。	
REV1.8	增加使用外部高速晶振，睡眠电流达到最小的配置说明 参见 3.2Halt与 Sleep模式	2014-2-12

目录

版本历史	2
目 录	3
1 产品概述	5
1.1 功能描述	5
1.2 BONDING图	6
1.3 功能模块原理图	8
2 标准功能	10
2.1 CPU核	10
2.1.1 存储器	12
2.1.2 状态寄存器	14
2.1.3 INTE及 INTF中断寄存器	15
SFR	17
2.1.4 系统专用寄存器	17
2.1.5 辅助专用寄存器	17
2.2 时钟系统	19
2.2.1 振荡器状态	19
2.2.2 CPU指令周期	21
2.2.3 ADC时钟	22
2.2.4 蜂鸣器时钟	23
2.2.5 TMCLK（定时器及 LCD模块输入时钟）	24
2.3 定时器	25
2.4 I/O PORT	26
2.4.1 数字 OUT PUT口：PT1[4:3]	31
2.4.2 数字 I/O口：PT1[7:5]	33
2.4.3 数字 I/O口与外部中断输入：PT2[0]，PT2[1]	35
2.4.4 数字 I/O口或者 PDM输出：PT2[3:2]	37
2.4.5 数据 I/O接口：PT2[6:4]	39
2.4.6 数据 I/O接口或者蜂鸣器输出：PT2[7]	41
2.4.7 带 LCD Segment驱动输出的的数字 I/O口：PT3[7:0]	43
3 增强功能	45
3.1 电源系统	45
3.1.1 Regulator	45
3.1.2 低电压比较器	47
3.2 HALT与 SLEEP模式	48
3.3 看门狗(WDT)	49
3.4 ADC模块	50
3.5 LCD DRIVER	55
3.6 PDM模块	75
4 CPU指令集	79
5 电气特性	91
5.1 CSU1281极限值	91
5.2 直流特性（VDD = 3.3V，T _A = 25°C，如无其他说明则都是此条件）	91
5.3 ADC特性（VDD = 3.3V，T _A = 25°C，如无其他说明则都是此条件）	92
5.4 复位特性（BROWNOUT及外部复位引脚）	92
6 BONDING说明	93

6.1	PAD排布	94
6.2	PAD坐标	95
7	产品命名规则.....	97
	产品型号说明	97

1 产品概述

1.1 功能描述

CSU1281是一个 8位 CMOS单芯片 MCU，内置 $4K \times 16$ 位一次性可编程（OTP）ROM，带有 2通道 24位全差分输入或 4通道 24位单端输入 ADC，低噪声放大器及 4×24 或 8×22 的 LCD驱动。

高性能的 RISC CPU

（默认值：AVDD，另可以配置为 2.3、2.6、2.8V）

- ✓ 8位单片机 MCU
- ✓ 内置 $4K \times 16$ 位一次性可编程存储器（OTP ROM）
- ✓ 256字节数据存储器（SRAM）
- ✓ 只有 37条单字指令
- ✓ 8级存储堆栈

专用微控制器的特性

- ✓ 上电复位（POR）
- ✓ 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
- ✓ 电压工作范围
 - DVDD 2.4V~3.6V
 - AVDD 2.4V~3.6V

振荡器

- ✓ 内带 1/3.25MHz振荡器
- ✓ 外部晶振（32.768kHz RTC）或 4MHz~16MHz晶振

外设特性

- ✓ 19位双向 I/O口外加 2位输出口
- ✓ 2路 PDM脉冲密度调制器输出
- ✓ 1路蜂鸣器输出
- ✓ 可选 4×24 、 8×22 的 LCD驱动
 - 可选择内部晶振，外部晶振
 - 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
- ✓ 内置温度传感器

低功耗特性

- ✓ MCU工作电流
 - 正常模式 1mA@1MHz 2mA@4MHz(工作电压 3.3V);
 - 休眠模式下的电流小于 3μA

封装

- ✓ 封装：65-PIN dice

模拟特性

- ✓ 模数转换器（ADC）
 - 2路模拟输入
 - 24位分辨率，有效精度 20位(PGA为 1 时，输出速率为 25Hz)。
 - 内部集成的可编程增益放大器能提供 1、64、128、256等不同倍率的增益。
 - ADC的输出速率 3.8Hz~15KHz
- ✓ 内带电荷泵
- ✓ 内带稳压器，提供传感器和调制器的电源

应用场合

- ✓ 传感器或变换器测量应用
- ✓ 电子衡器如高精度口袋秤，厨房称，人体称，脂肪秤
- ✓ 数字仪表
- ✓ 家用医疗仪器
- ✓ 家电控制器

1.2 Bonding图

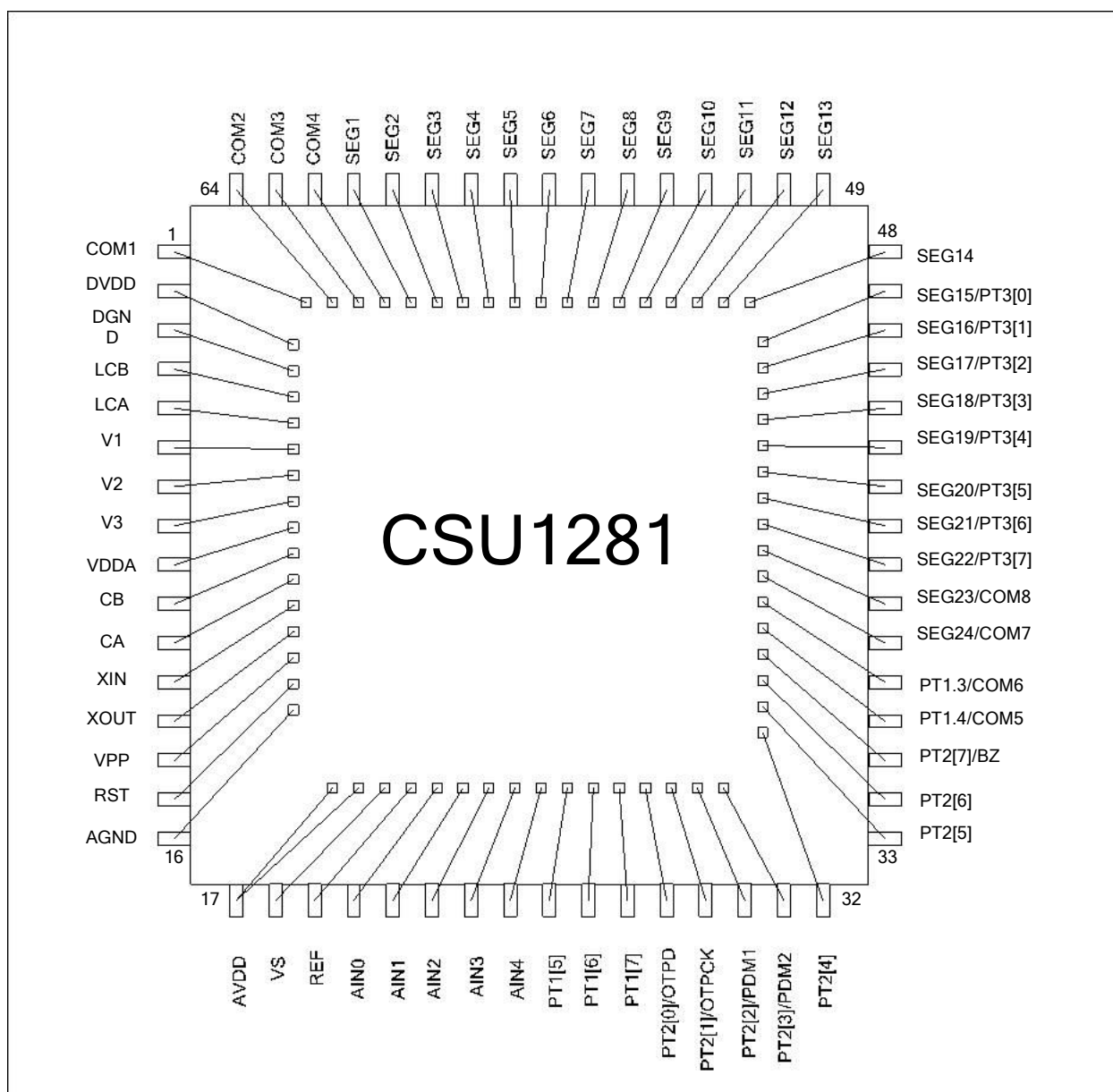


图 1-1引脚配置图

备注：衬底需要接地。

表 1-1 引脚说明表

管脚名称	输入/输出	管脚序号	描述
COM1	O	1	LCD Com输出
DVDD	I	2	数字电源
DGND	I	3	数字地
LCB/LCA	O	4~5	LCD电压产生所需的电荷交换电容
V1/V2/V3	I/O	6~8	LCD显示所使用的 1V/2V/3V电压
VDDA	I	9	升压电路所产生的 3V电压
CB,CA	O	10~11	升压电路所使用的电荷交换电容
XIN,XOUT	I/O	12~13	外部晶振的输入和输出
VPP	I	14	OTP烧写电压输入 烧写 OTP时接 6.5V，正常工作接 VDD
RST	I	15	复位信号输入,低电平有效
AGND	I	16	模拟地
AVDD	I	17	模拟电源
VS	O	18	内部 LDO输出，可用于给传感器供电
REF	I	19	ADC参考电压输入引脚，也可由外部提供 VDD
ANA0	I/O	20	模拟差分输入端
ANA1	I/O	21	模拟差分输入端
ANA2	I/O	22	模拟差分输入端
ANA3	I/O	23	模拟差分输入端
ANA4	I/O	24	低电压检测输入端
PT1[5]~[7]	I/O	25~27	I/O
PT2[0]/OTPD	I/O	28	OTP烧写的数据口或作为 I/O
PT2[1]/OTPC	I/O	29	OTP烧写的时钟输入口或作为 I/O
PT2[2]~[3]/PDM1~2	I/O	30~31	两路 PDM码输出，或作为 I/O
PT2[4]~[6]	I/O	32~34	I/O
PT2[7]/BZ	I/O	35	蜂鸣器输出或者 I/O
COM5	I/O	36	LCD的 COM5输出或者 PT1.4的输出功能(无输入功能，有上拉电阻功能)
COM6	I/O	37	LCD的 COM6输出或者 PT1.3的输出功能(无输入功能，有上拉电阻功能)
SEG24/COM7	I/O	38	LCD的 Segment或 COM7复用引脚
SEG23/COM8	I/O	39	LCD的 Segment或 COM8复用引脚
SEG22~SEG15	I/O	40~47	LCD的 Segment或作为 PT3 I/O口
SEG14~SEG1	I/O	48~61	LCD的 Segment
COM4~2	O	62~64	LCD Com4~2输出

1.3 功能模块原理图

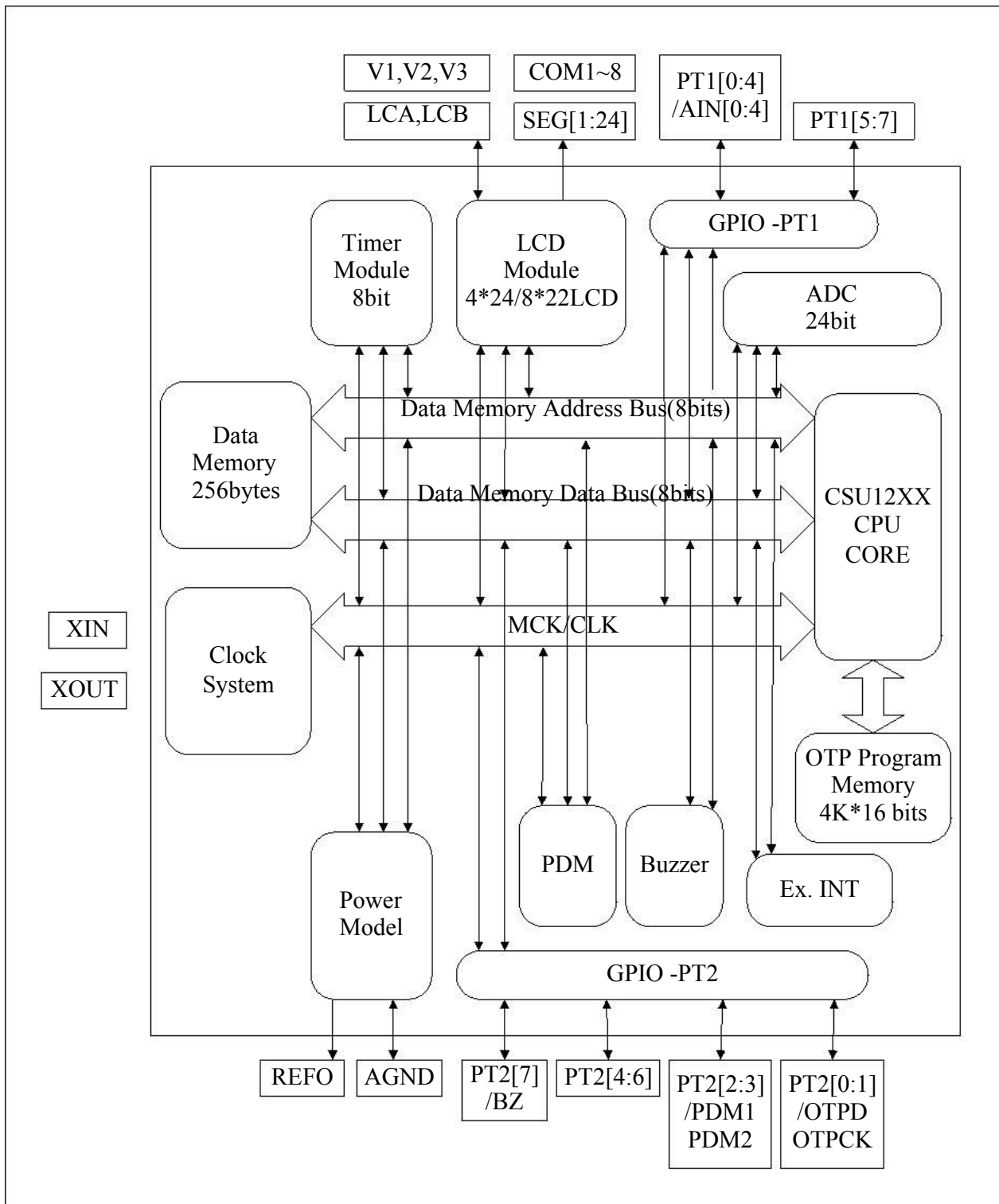


图 1-2 CSU1281能模块

从功能模块原理图中可看到有 5个功能模块，如表 1-2描述

表 1-2 CSU1281主要功能描述

项目	子项目	描述
CPU核	RISC CPU Core	详细描述见 2.1节
	OTP程序存储器	OTP：一次性可编程 8K Bytes用于 4K条编程指令
	数据存储器	CSU1281带有 384 Bytes SRAM（128 Bytes寄存器， 256 Bytes普通数据存储器）
	时钟系统	CSU1281有两种时钟源。一种是 1MHz或者 3.25MHz的内部时钟供 CPU工作，另一个是 32768Hz或者 4MHz~16MHz的外部时钟。
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 4×24或 8×22的 LCD驱动器
	PDM	脉冲密度调制
	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	CSU1281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	CSU1281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 5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， PDM及蜂鸣 器
	PT3	PT3接口有 8位。用户可以定义这 8位接口用于普通 用途或者 LCD的 Segment

2 标准功能

2.1 CPU核

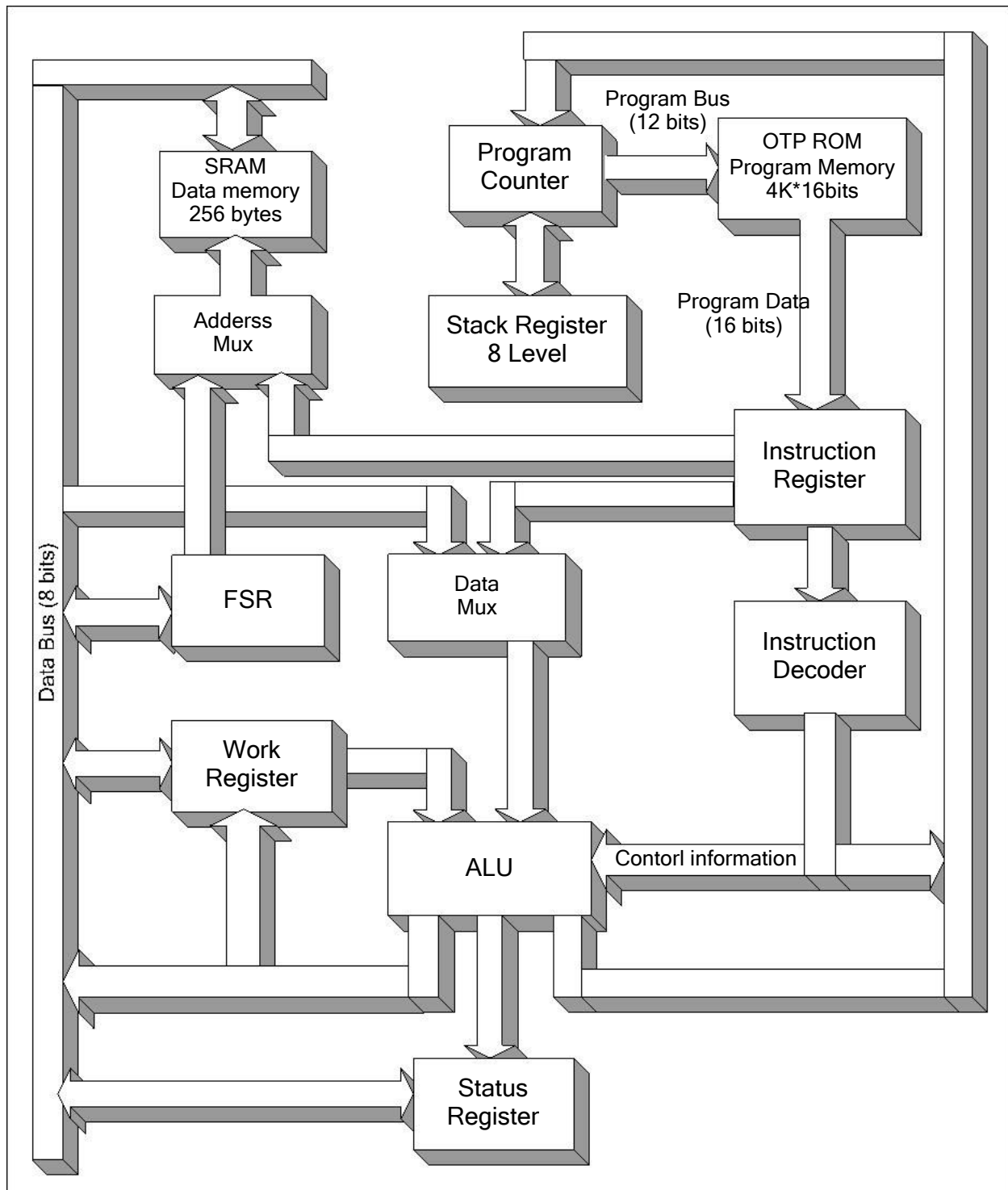


图 2-1 CSU1281 CPU核的功能模块图

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 CPU架构说明

模块名称	描述
程序计数器	程序计数器在 CPU的工作周期期间起到很重要的作用，它记录 CPU每个周期处理程序存储器中指令的指针。在一个 CPU周期中，程序计数器将程序存储器地址（12bits），指令指针推送到程序存储器，然后自动加 1以进行下一次周期。
栈寄存器	堆栈寄存器是用来记录程序返回的指令指针。当程序调用函数，程序计数器会将指令指针推送到堆栈寄存器。在函数执行结束之后，堆栈寄存器会将指令指针送回到程序计数器以继续原来的程序处理。
指令寄存器	程序计数器将指令指针（程序存储器地址）推送到程序存储器，程序存储器将程序存储器的数据（16bits）推送到指令寄存器。 CSU1281的指令是 16bits，包括 3种信息：直接地址，立即数及控制信息。 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。 直接地址（9bits） 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。 立即数（8bits） CPU通过 ALU利用此数据对工作寄存器进行操作。 控制信息 它记录着 ALU的操作信息。
指令译码器	指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。
算术逻辑单元	算术逻辑单元不仅能完成 8位二进制的加，减，加 1，减 1等算术计算，还能对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 CSU1281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	CSU1281内带 8K 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 4K的指令。程序存储器的地址总线是 12bits，数据总线是 16bits。
数据存储器	CSU1281内带 256 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。

8

2.1.1 存储器

1. 程序存储器主要用于指令的存储，在 CSU1281中，该程序存储器是 $4K \times 16\text{bits}$ 的 OTP，对于程序员来说，该存储器只读，不可以写入。

系统的 reset地址 000H，中断入口地址 004H，需要注意的是所有的中断共用同一个中断入口地址。

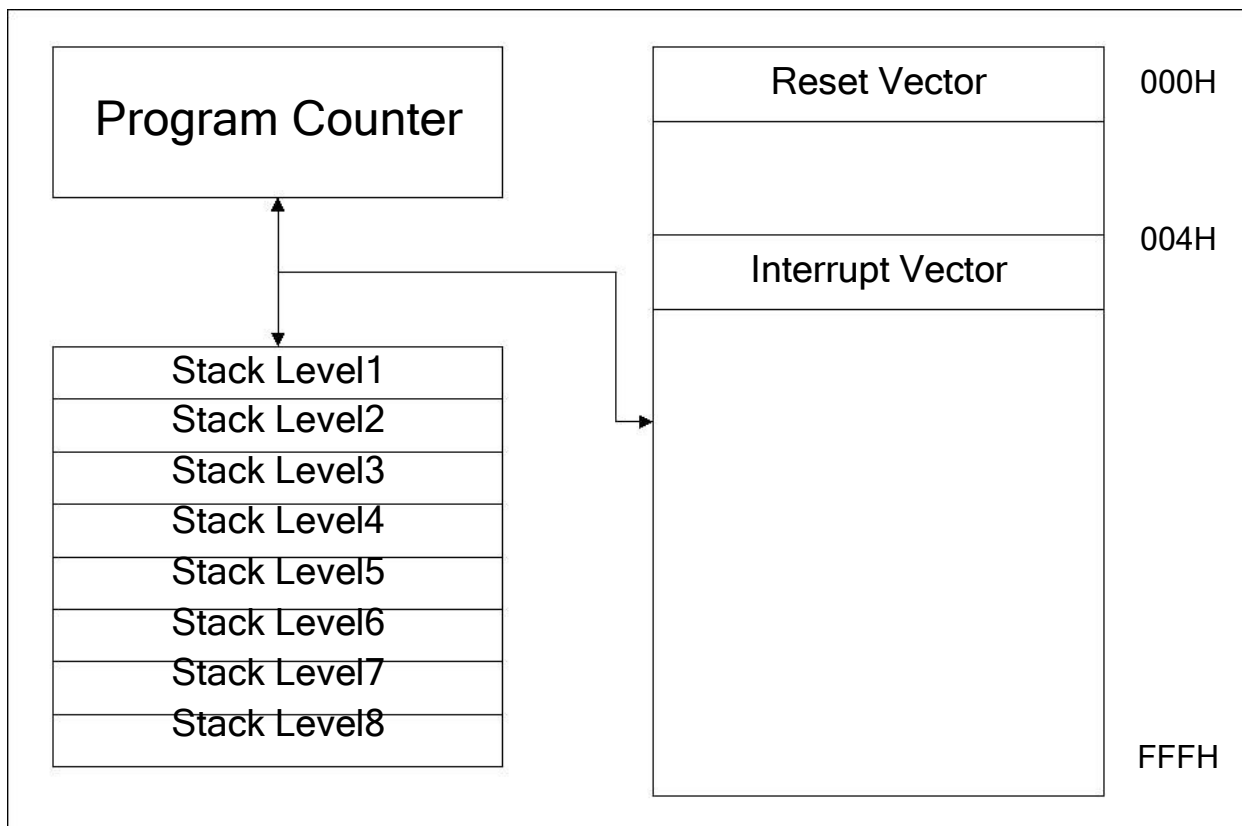


图 2-2程序存储器

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 000H至 007H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 008H至 07FH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	000H	007H
外设特殊功能寄存器	008H	07FH
通用数据存储器	080H	17FH

3. 通过 IND0和 IND1以及 FSR0与 FSR1这四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，CPU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，CPU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。

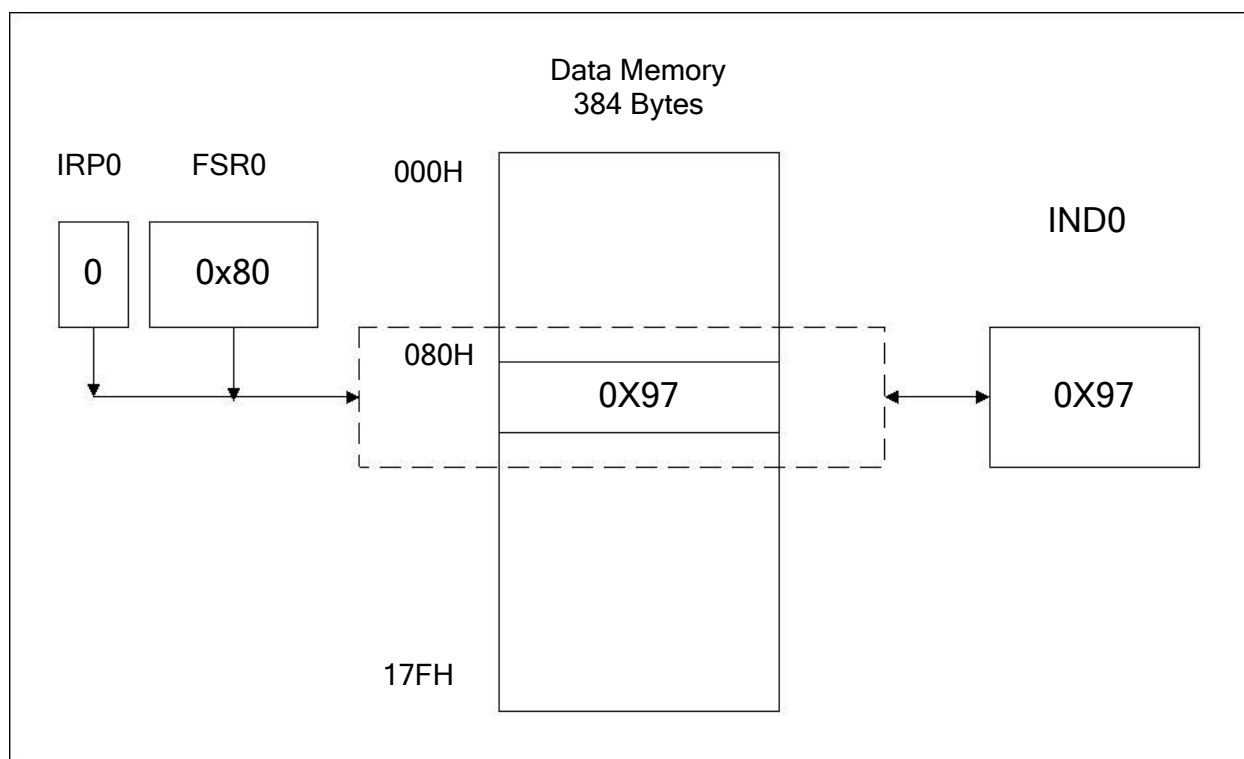


图 2-3间接地址访问

4. 数据存储器后 128byte (100H~17FH) 的访问方式。

- 1) 指令支持直接的访问，例如 MOVWF f 中 f 可以是 9bit 的地址。
- 2) 此外还可以通过间接寻址的方式访问，间接寻址访问时，需要先将 IRP1 或者 IRP0 设置为 1，然后按照间接寻址的方式访问即可，需要注意的是，间接地址访问时 FSR1 或者 FSR0 中保存的地址仍然是实际地址减去 100H。

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=04H）

特性	R/W-0	R/W-0	U-0	R-0	R-0	R/W-0	R/W-0	R/W-0
STATUS	IRP1	IRP0		PD	TO	DC	C	Z
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 IRP1: 间接寻址 1 翻页标志位

1 = 间接寻址 1 访问 100H~17FH 的内容

0 = 间接寻址 1 访问 00H~FFH 的内容

Bit 6 IRP0: 间接寻址 0 翻页标志位

1 = 间接寻址 0 访问 100H~17FH 的内容

0 = 间接寻址 0 访问 00H~FFH 的内容

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。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

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 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=07H）

特性	R/W-0	U-0	U-0	R/W-0	U-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=06H）

特性	U-0	U-0	U-0	R/W-0	U-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: 8-Bit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

注意：对于 INTF寄存器请采用 **movwf**指令进行操作。
 例如：要清除 TMIF寄存器，请使用以下指令
MOVLW 11101111b
MOVWF INTF

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

SFR

2.1.4 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	Bi5	Bi4	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
01H	IND1	以 FSR1 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
03H	FSR1	间接数据存储器的地址指针 1								00000000
04H	STATUS	IRP1	IRP0		PD	TO	DC	C	Z	00u00000
05H	WORK	工作寄存器								uuuuuuuu
06H	INTF				TMIF		ADIF	E1IF	E0IF	uuu0u000
07H	INTE	GIE			TMIE		ADIE	E1IE	E0IE	0uu0u000

2.1.5 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 及以下章节。

注意：PT1PU[2] 为 TEST 寄存器，必须保持默认值 0。

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
0DH	WDTCON	WTDEN					WTS[2:0]			0uuuu000
0EH	TMOUT	TMOUT[7:0]								00000000
0FH	TMCON	TRST				TMEN	INS[2:0]			1uuu0000
10H	ADOH	ADO[23:16]								00000000
11H	ADOL	ADO[15:8]								00000000
12H	ADOLL	ADO[7:0]								00000000
13H	ADCON						ADM[2:0]			uuuuu000
14H	MCK						M2_CK	M1_CK	M0_CK	00000000
15H	PCK	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK		EO_SLP	S_BEEP	uuuuu000
18H	NETA								OSC_H	0000uuuu
1AH	NETC	SINL[1:0]		SENS[1:0]						00uu000u
1CH	NETE	CHOPM[1:0]				ADG[1:0]		ADEN		00uu000u
1DH	NETF	LDOS[1:0]				SILB[1:0]		ENLB		uu0uu000
1FH				ENVDDA			BGID[1:0]		ENVB	uuuuuuux
20H	SVD								LBOUT	xxxxxuuu
21H	PT1	PT1[7:5]			PT1 [4]	PT1[3]				00000uuu
22H	PT1EN	PT1EN[7:5]			PT1EN[4]	PT1EN[3]				00000uuu
24H	PT1PU	PT1PU[7:5]			PT1PU[4]	PT1PU[3]	TEST			xxxxxxxx
25H	PT2	PT2[7:0]								00000000
26H	PT2EN	PT2EN[7:0]								00000000
27H	PT2PU	PT2PU[7:0]								0u000000
28H	PT2MR									00000000
29H	PT3	BZEN		PM2EN	PM1EN	E1M[1:0]		E0M[1:0]		00000000
2AH	PT3EN	PT3[7:0]								00000000
2BH	PT3PU	PT3EN[7:0]								00000000
30H	AOENB3	PT3PU[7:0]								00000000
31H	PMD1H	AOENB3 [7:0]								00000000
32H	PMD1L	PMD1[15:8]								00000000
33H	PMD2H	PMD1[7:0]								00000000
36H	PMD2L	PMD2[15:8]								00000000
40H	PMCON	PMD2[7:0]								00000000
41H	LCD1	PDM2EN	PM2CS[2:0]			PDM1EN	PM1CS[2:0]			00000000
		SEG1[7:4](仅对 8COM)				SEG1[3:0]				
	LCD2	SEG2[7:4] (仅对 8COM)				SEG2[3:0]				
42H	LCD3	SEG3[7:4] (仅对 8COM)				SEG3[3:0]				00000000
43H	LCD4	SEG4[7:4] (仅对 8COM)				SEG4[3:0]				00000000
44H	LCD5	SEG5[7:4] (仅对 8COM)				SEG5[3:0]				00000000
45H	LCD6	SEG6[7:4] (仅对 8COM)				SEG6[3:0]				00000000
46H	LCD7	SEG7[7:4] (仅对 8COM)				SEG7[3:0]				00000000
47H	LCD8	SEG8[7:4] (仅对 8COM)				SEG8[3:0]				00000000
48H	LCD9	SEG9[7:4] (仅对 8COM)				SEG9[3:0]				00000000
49H	LCD10	SEG10[7:4] (仅对 8COM)				SEG10[3:0]				00000000
4AH	LCD11	SEG11[7:4] (仅对 8COM)				SEG11[3:0]				00000000
4BH	LCD12	SEG12[7:4] (仅对 8COM)				SEG12[3:0]				00000000
4CH	LCD13	SEG13[7:4] (仅对 8COM)				SEG13[3:0]				00000000
4DH	LCD14	SEG14[7:4] (仅对 8COM)				SEG14[3:0]				00000000
4EH	LCD15	SEG15[7:4] (仅对 8COM)				SEG15[3:0]				00000000
4FH	LCD16	SEG16[7:4] (仅对 8COM)				SEG16[3:0]				00000000
50H	LCD17	SEG17[7:4] (仅对 8COM)				SEG17[3:0]				00000000
51H	LCD18	SEG18[7:4] (仅对 8COM)				SEG18[3:0]				00000000
52H	LCD19	SEG19[7:4] (仅对 8COM)				SEG19[3:0]				00000000
53H	LCD20	SEG20[7:4] (仅对 8COM)				SEG20[3:0]				00000000
54H	LCD21	SEG21[7:4] (仅对 8COM)				SEG21[3:0]				00000000
55H	LCD22	SEG22[7:4] (仅对 8COM)				SEG22[3:0]				00000000
56H	LCD23					SEG23[3:0](仅对 4COM)				uuuu0000
57H	LCD24					SEG24[3:0](仅对 4COM)				uuuu0000
58H	LCDENR	LCDCKS[1:0]		LCDEN		LEVEL	LCD_DUTY[1:0]		ENPMPL	000u0110

2.2 时钟系统

2.2.1 振荡器状态

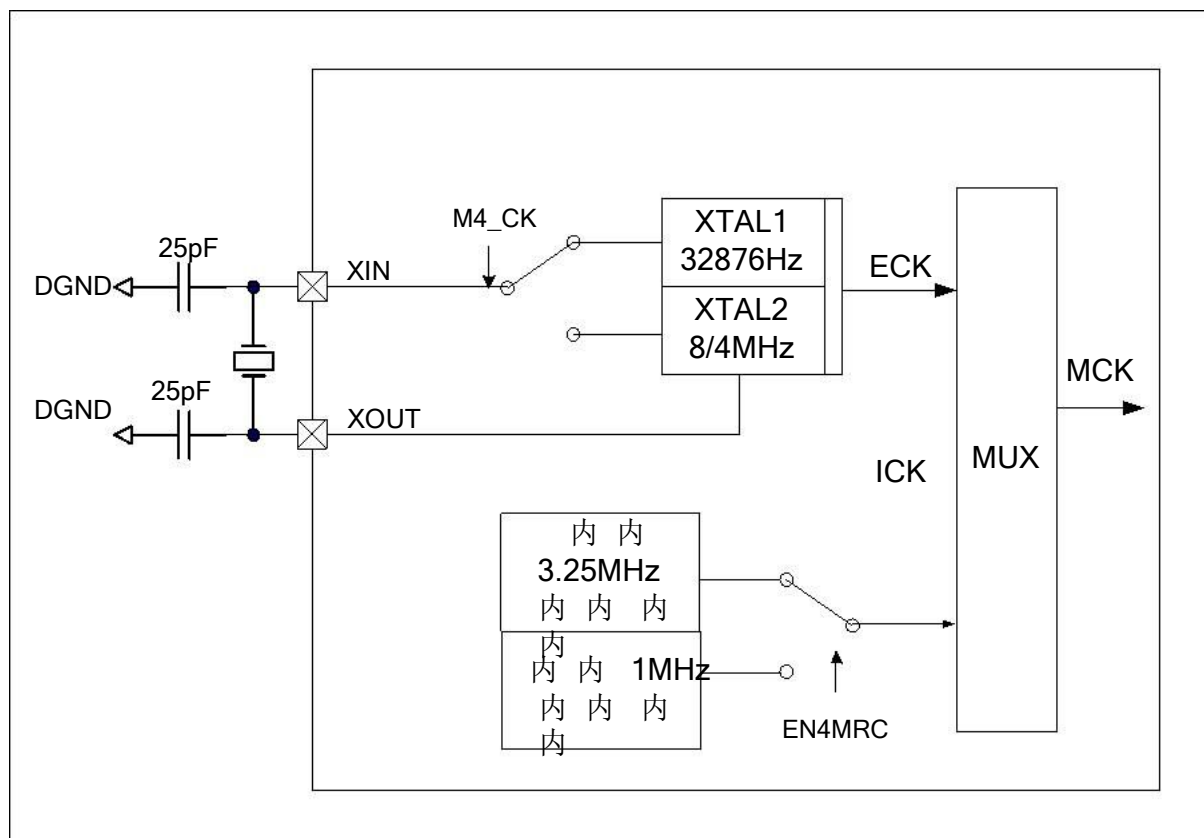


图 2-4 CSU1281振荡器状态框图

表 2-5 CSU1281时钟系统寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
14H	MCK	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK						EO_SLP		OSC_H	uuuuu000

CSU1281有两种时钟源。一种是内部时钟，可以产生 1MHz或者 3.25MHz的时钟供 CPU工作；另一个是外部时钟，能够提供 32768Hz或者 4~16MHz的时钟信号。用户可以通过 MCK、PCK寄存器来选择其中的一个时钟。请看图 2-4。通过 M3_CK和 M0_CK配置 MCK。

表 2-6 MCK选择列表

M3_CK	M0_CK	MCK
X	0	ICK
0	1	ECK/4
1	1	ECK/8

用户必须正确设置 M7_CK与 M6_CK以使能内部及外部振荡器，如表 2-7所示。如果执行睡眠指令来使 CSU1281进行睡眠模式，内部及外部振荡器将不使能，但是如果将 EO_SLP寄存器置高以后，执行睡眠指令后，内部晶振关闭，外部晶振依旧打开。

表 2-7振荡器状态选择列表

输入				振荡器状态	
睡眠指令	EO_SLP	M7_CK	M6_CK	内部振荡器	外部振荡器
1	0	X	X	不使能	不使能
1	1	X	0	不使能	使能
1	1	X	1	不使能	不使能
0	X	0	0	使能	使能
0	X	0	1	使能	不使能
0	X	1	0	不使能	使能
0	X	1	1	不使能	不使能

注：X即该位可为任意值。

注意：在使用外部晶振时，32768Hz的晶振需要 300ms的起振时间。4MHz的晶振需要 10ms的起振时间。

2.2.2 CPU指令周期

表 2-8 CSU1281 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	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	00000000

用户可以通过设置 M0_CK, M1_CK, M2_CK及 M3_CK来选择指令周期（用户必须保证切换振荡器时，振荡器的输出是稳定的；一般在切换后加两条 NOP指令）。其中 M2_CK,M1_CK选择指令周期与 MCK的关系（如表 2-9），M3_CK,M0_CK选择晶振与 MCK关系，M4_CK,OSC_H选择晶振的频率。

表 2-9指令周期与 MCK关系列表

M2_CK	M1_CK	指令周期
0	0	MCK/8
0	1	MCK/16
1	0	MCK/2
1	1	MCK/4

表 2-10指令周期选择列表

OSC_H	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	MCK(KHz)		指令周期(KHz)	
0	X	X	0	0	0	ICK	1000	MCK/8	125
0	X	X	0	1	0	ICK	1000	MCK/16	62.5
0	X	X	1	0	0	ICK	1000	MCK/2	500
0	X	X	1	1	0	ICK	1000	MCK/4	250
1	X	X	0	0	0	ICK	3250	MCK/8	406
1	X	X	0	1	0	ICK	3250	MCK/16	203
1	X	X	1	0	0	ICK	3250	MCK/2	1625
1	X	X	1	1	0	ICK	3250	MCK/4	812
X	0	0	0	0	1	ECK/4	8.192	MCK/8	1.024
X	0	0	0	1	1	ECK/4	8.192	MCK/16	0.512
X	0	0	1	0	1	ECK/4	8.192	MCK/2	4.096
X	0	0	1	1	1	ECK/4	8.192	MCK/4	2.048
X	0	1	0	0	1	ECK/8	4.096	MCK/8	0.512
X	0	1	0	1	1	ECK/8	4.096	MCK/16	0.256
X	0	1	1	0	1	ECK/8	4.096	MCK/2	2.048
X	0	1	1	1	1	ECK/8	4.096	MCK/4	1.024
X	1	0	0	0	1	ECK/4	1000	MCK/8	125
X	1	0	0	1	1	ECK/4	1000	MCK/16	62.5
X	1	0	1	0	1	ECK/4	1000	MCK/2	500
X	1	0	1	1	1	ECK/4	1000	MCK/4	250
X	1	1	0	0	1	ECK/8	500	MCK/8	62.5
X			0	1	1	ECK/8		MCK/16	31.25
X	1	1	1	0	1	ECK/8	500	MCK/2	250
X	1	1	1	1	1	ECK/8	500	MCK/4	
	1	1					500		125

2.2.3 ADC时钟

表 2-11 ADC采样频率选择寄存器

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
MCK					M3_CK		M1_CK	M0_CK
14H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

CSU1281中 ADC的采样频率用于采样信号。用户通过设置 M0_CK, M1_CK, M3_CK, OSC_H 寄存器标志位来改变采样时钟, 设置如表 2-12。由于 32KHz晶振时, ADC的采样速率过低, 已经不可用, 所以 ECK用为 4MHz外部晶振时的情况。

表 2-12 ADC采样频率选择列表

OSC_H	M3_CK	M1_CK	M0_CK	MCK(KHz)		ADCF(KHz)	
0	X	0	0	ICK	1000	MCK/16	62.5
0	X	1	0	ICK	1000	MCK/32	31.25
1	X	0	0	ICK	3250	MCK/16	203
1	X	1	0	ICK	3250	MCK/32	101.5
X	0	0	1	ECK/4	1000	MCK/16	62.5
X	0	1	1	ECK/4	1000	MCK/32	31.25
X	1	0	1	ECK/8	500	MCK/16	31.25
X	1	1	1	ECK/8	500	MCK/32	16

表 2-13 ADC输出速率选择寄存器

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ADCON						ADM[2:0]		
13H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

表 2-14 ADC输出速率选择列表

ADM[2:0]	ADC输出速率
000	ADCF/128
001	ADCF/256
010	ADCF/512
011	ADCF/1024
100	ADCF/2048
101	ADCF/4096
110	ADCF/8192
111	ADCF/8192

2.2.4 蜂鸣器时钟

表 2-15 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
14H	MCK	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK							S_BEEP		uuuuu000

CSU1281有一个蜂鸣器时钟用于蜂鸣器源。用户通过设置 M0_CK, M1_CK, M3_CK, M4_CK及 S_BEEP寄存器标志位来改变蜂鸣时钟, 设置如表 2-16。

表 2-16 蜂鸣器时钟选择列表

OSC_H	M4_CK	M3_CK	M1_CK	M0_CK	S_BEEP	时钟源(KHz)		BEEP CLOCK(KHz)	
0	X	X	0	0	0	ICK	1000	ICK/256	4
0	X	X	0	0	1	ICK	1000	ICK/512	2
0	X	X	1	0	0	ICK	1000	ICK/1024	1
0	X	X	1	0	1	ICK	1000	ICK/2048	0.5
1	X	X	0	0	0	ICK	3250	ICK/256	12.7
1	X	X	0	0	1	ICK	3250	ICK/512	6.3
1	X	X	1	0	0	ICK	3250	ICK/1024	3.2
1	X	X	1	0	1	ICK	3250	ICK/2048	1.6
X	0	0	0	1	0	ECK	32	ECK/512	0.064
X	0	0	1	1	0	ECK	32	ECK/2048	0.016
X	0	1	0	1	0	ECK	32	ECK/1024	0.032
X	0	1	1	1	0	ECK	32	ECK/4096	0.008
X	0	X	X	1	1	ECK	32	ECK/8	4.096
X		0	0	1	0	ECK	4000		8
X	1		1	1	0	ECK	4000	ECK/512	
X	1	0		1	0	ECK	4000	ECK/2048	2
X	1	1	0	1	0	ECK	4000	ECK/1024	4
X	1	1	1	1	0	ECK	4000	ECK/4096	1
X		X	X	1	1	ECK	4000	ECK/8	500

2.2.5 TMCLK（定时器及LCD模块输入时钟）

TMCLK用于 CSU1281定时器及 LCD模块。根据表 2-17，用户通过正确设置 M5_CK标志位以选择 TMCLK的频率。

表 2-17 TMCLK选择列表

OSC_H	M4_CK	M5_CK	M3_CK	M1_CK	M0_CK	时钟源(KHz)		TMCLK(Hz)	
0	X	0	X	0	0	ICK	1000	ICK/1024	976
0	X	0	X	1	0	ICK	1000	ICK/4096	244
1	X	0	X	0	0	ICK	3250	ICK/2048	1586
1	X	0	X	1	0	ICK	3250	ICK/4096	793
X	0	0	0	0	1	ECK	32	ECK/2048	16
X	0	0	0	1	1	ECK	32	ECK/8192	4
X	0	0	1	0	1	ECK	32	ECK/4096	8
X	0	0	1	1	1	ECK	32	ECK/16384	2
X	1	0	0	0	1	ECK	4000	ECK/2048	2000
X	1	0	0	1	1	ECK	4000	ECK/8192	500
X	1	0	1	0	1	ECK	4000	ECK/4096	1000
X	1	0	1	1	1	ECK	4000	ECK/16384	250
X	0	1	X	X	X	ECK	32	ECK/32	1024
X	1	1	X	X	X	ECK	4000	ECK/32	125000

2.3 定时器

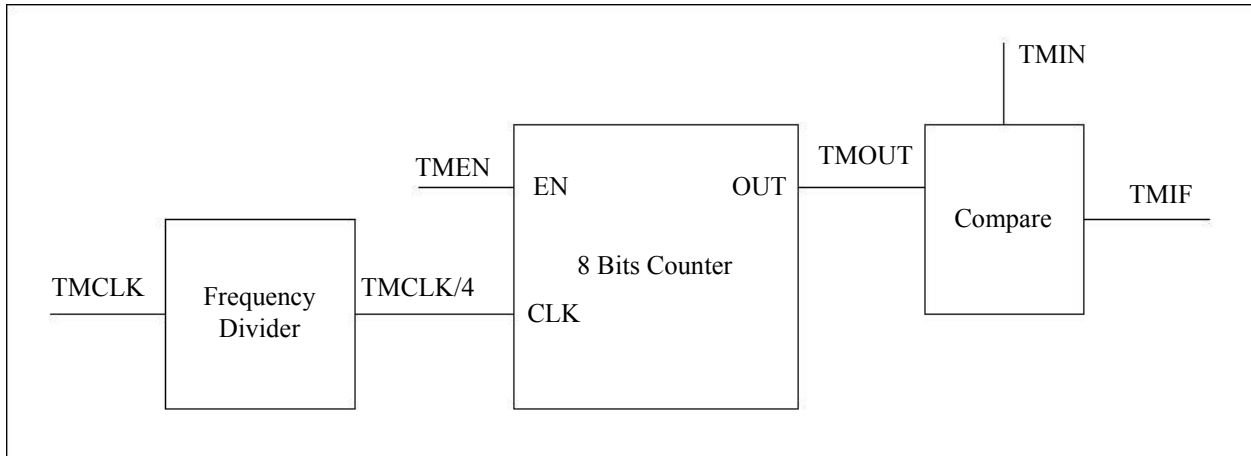


图 2-5 定时模块的功能框图

定时器模块的输入是 TMCLK，定时器分频器对 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 递增到 FFH。用户需要设置 INS（定时器模块中断信号选择器）以选择定时中断信号。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 2-18 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					uuu0u000
07H	INTE	GIE			TMIE					0uu0u000
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 2-19。
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 2-19 定时器选择列表

INS[2:0]	中断源	时间 TMCLK=1024(ECK/32)
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

2.4 I/O port

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							
06H	INTF							E1IF	E0IF	uuu0u000							
07H	INTE	GIE						E1IE	E0IE	0uu0u000							
20H	PT1	PT1[7:5]			PT1[4]	PT1[3]				xxxxxuuu							
21H	PT1EN	PT1EN[7:5]			PT1EN[4]	PT1EN[3]				00000uuu							
22H	PT1PU	PT1PU[7:5]			PT1PU[4]	PT1PU[3]	TEST			000000uu							
24H	PT2	PT2[7:0]								xxxxxxxx							
25H	PT2EN	PT2EN[7:0]								00000000							
26H	PT2PU	PT2PU[7:0]								00000000							
27H	PT2MR	BZEN	PM2EN		PM1EN	E1M[1:0]		E0M[1:0]		0u000000							
28H	PT3				PT3[7:0]								00000000				
29H	PT3EN				PT3EN[7:0]								00000000				
2AH	PT3PU	PT3PU[7:0]								00000000							
2BH	AOENB3	PT3PU[7:0]								00000000							
		AOENB3 [7:0]															

微控制器中的通用 I/O口（GPIO）用于通用的数字的输入与输出功能。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。CSU1281的部分 GPIO可以被定义为其它的特殊功能。在本节，只说明 GPIO的通用 I/O口功能，特殊功能将会在接下来的章节中说明。

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	R/W-X	U-0	U-0	U-0
PT1	PT1[7:5]			PT1[4]	PT1[3]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1[7:5]: 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，仅 LCD选择非 8COM模式时有效(即 LCD_DUTY[1:0]不等于 00)

PT1[3] = GPIO1 bit 3数据标志位，仅 LCD选择非 8COM模式时有效(即 LCD_DUTY[1:0]不等于 00)

注意：如果使用了 PT1[4]和 PT1[3]作为输出，则不得对 PT1[7:6]整组寄存器进行位操作，否则会把这两个两个口的输出值清零。

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

PT1EN寄存器（地址为 21H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
PT1EN	PT1EN[7:5]			PT1EN[4]	PT1EN[3]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1EN[7:5]: 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;注意: 此 IO口只能作为输出, 输入功能无效

PT1EN[3] = GPIO1 bit 3的 I/O控制标志位;注意: 此 IO口只能作为输出, 输入功能无效

PT1PU寄存器（地址为 22H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1PU	PT1PU[7:5]			PT1PU[4]	PT1PU[3]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PU[7:5]: 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, 不受 LCD_DUTY 控制, 因此 LCD为 8COM模式时, 此 bit必须置 0

PT1PU[3] = GPIO1 bit 3控制标志位; 0 =断开上拉电阻, 1 =使用上拉电阻, 不受 LCD_DUTY 控制, 因此 LCD为 8COM模式时, 此 bit必须置 0

PT2寄存器（地址=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

PT2[7] = GPIO2 bit 7的数据

PT2[6] = GPIO2 bit 6的数据

PT2[5] = GPIO2 bit 5的数据

PT2[4] = GPIO2 bit 4的数据

PT2[3] = GPIO2 bit 3的数据

PT2[2] = GPIO2 bit 2的数据

PT2[1] = GPIO2 bit 1的数据

PT2[0] = GPIO2 bit 0的数据

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

PT2EN寄存器（地址=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=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

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

PT2MR寄存器（地址=27H）

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
PT2MR	BZEN		PM2EN	PM1EN	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 5 PM2EN：PDM模块使能标志

1 = PDM模块使能，GPIO2口 bit 3定义为 PDM输出

0 = PDM模块不使能，GPIO2口 bit 3定义为普通用途 I/O口

Bit 4 PM1EN：PDM模块使能标志

1 = PDM模块使能，GPIO2口 bit 2定义为 PDM输出

0 = PDM模块不使能，GPIO2口 bit 2定义为普通用途 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

PT3寄存器（地址=28H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3					PT3[7:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3[7:0]：GPIO3口数据

PT3[7] = GPIO3 bit 7数据

PT3[6] = GPIO3 bit 6数据

PT3[5] = GPIO3 bit 5数据

PT3[4] = GPIO3 bit 4数据

PT3[3] = GPIO3 bit 3数据

PT3[2] = GPIO3 bit 2数据

PT3[1] = GPIO3 bit 1数据

PT3[0] = GPIO3bit 0数据

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

PT3EN寄存器（地址=29H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3EN	PT3EN[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3EN[7:0]: GPIO3口输入/输出控制标志

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

PT3EN[6] = GPIO3 bit 6的 I/O控制标志位; 0 =定义为输入口, 1 =定义为输出口

PT3EN[5] = GPIO3 bit 5的 I/O控制标志位; 0 =定义为输入口, 1 =定义为输出口

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

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

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

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

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

PT3PU寄存器（地址=2AH）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3PU	PT3PU[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3PU[7:0]: GPIO3口上拉电阻使能标志

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

PT3PU[6] = GPIO3 bit 6控制标志位; 0 =断开上拉电阻, 1 =使用上拉电阻

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

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

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

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

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

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

AOENB3寄存器（地址=2BH）

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
AOENB3	AOENB3[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 AOENB3[7:0]: GPIO3模拟/数字控制标志, 与 PT3[7:0]一一对应, 0 =定义为模拟通道(即 LCD Segment输出), 1 =定义为数字通道

特性 (Property) :

R =可读位 W =可写位 U =无效位

-n =上电复位后的值 ‘1’ =位已设置 ‘0’ =位已清零

X =不确定位

2.4.1 数字OUT PUT口：PT1[4:3]

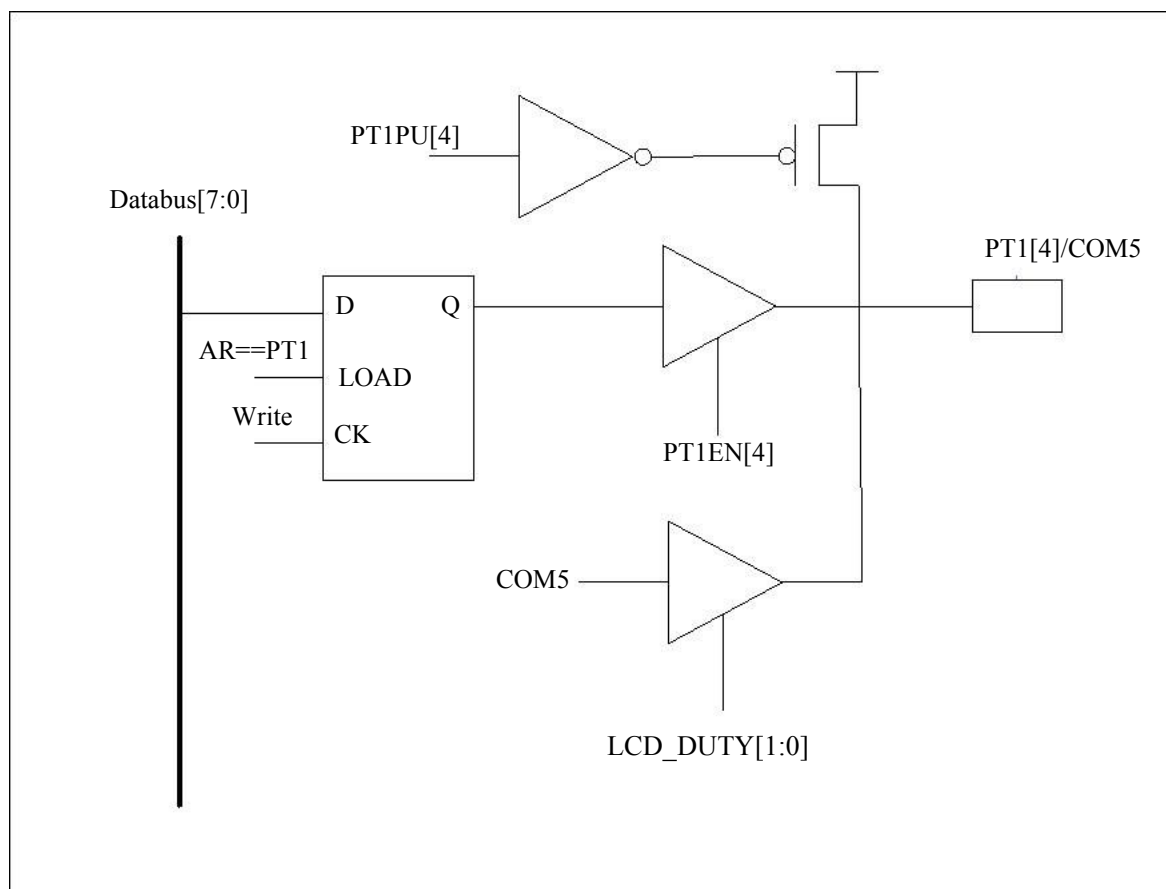


图 2-6 PT1[4:3]功能框图

GPIO1口（PT1[4:3]）功能框图如图 2-6所示。这两个口只能作为 OUTPUT使用，不可作为输入。当 LCD_DUTY[1:0]=00，即 LCD 8COM模式被启用时，此两个 IO口无效。

✓ 输出

CSU1281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（CSU1281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

CSU1281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。当接口接上拉电阻时，输入数据默认为高（即为 1）。

注意：如果把这两个 IO口设置为输出，则对 PT1[7:3]寄存器都不得使用 bsf和 bcf指令。即，对 PT1的所有 IO口，都不得使用 bsf指令和 bcf指令，否则会造成 PT1[3]和 PT1[4]输出结果为 0。

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[7:5]			PT1[4]	PT1[3]				xxxxxuuu
21H	PT1EN	PT1EN[7:5]			PT1EN[4]	PTEN1[3]				00000uuu
22H	PT1PU	PT1PU[7:5]			PT1PU[4]	PT1PU[3]	TEST			000000uu

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。(注意：当 LCD_DUTY[1:0]=00，即使用 8COM模式时，PT1EN[3]和 PT1EN[4]不可置 1，否则会与 COM波形发生冲突)。
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。(注意：当 LCD_DUTY[1:0]=00，即使用 8COM模式时，PT1PU[3]和 PT1PU[4]不可置 1，否则会与 COM波形发生冲突)
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. 在 I/O口与 DVDD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。

2.4.2 数字I/O口：PT1[7:5]

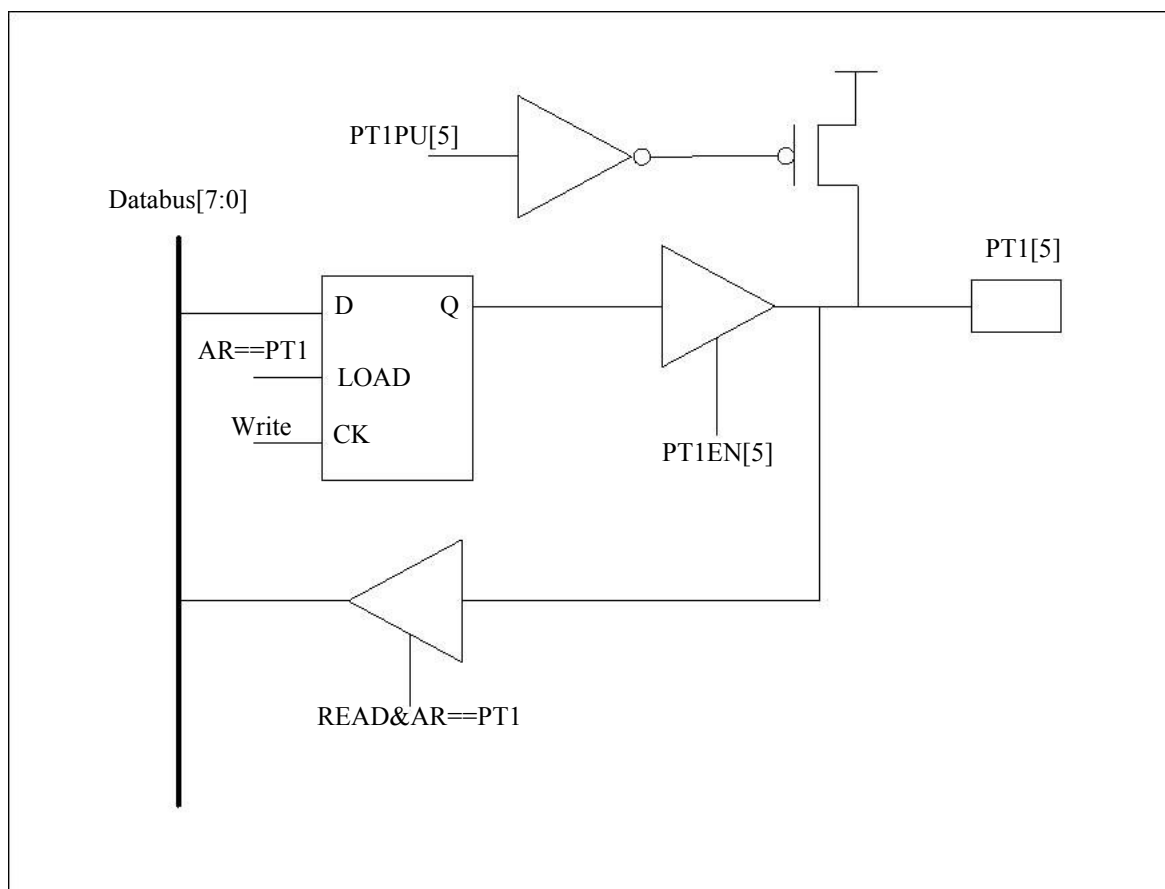


图 2-7 PT1[7:5]功能框图

GPIO1口（PT1[7:5]）功能框图如图 2-6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

CSU1281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（CSU1281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

CSU1281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[7:5]								xxxxxuuu
21H	PT1EN	PT1EN[7:5]								00000uuu
22H	PT1PU	PT1PU[7:5]			PT1PU[4]	PT1PU[3]	TEST			000000uu

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：

2. 在 I/O口与 DVDD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。

2.4.3 数字I/O口与外部中断输入：PT2[0]，PT2[1]

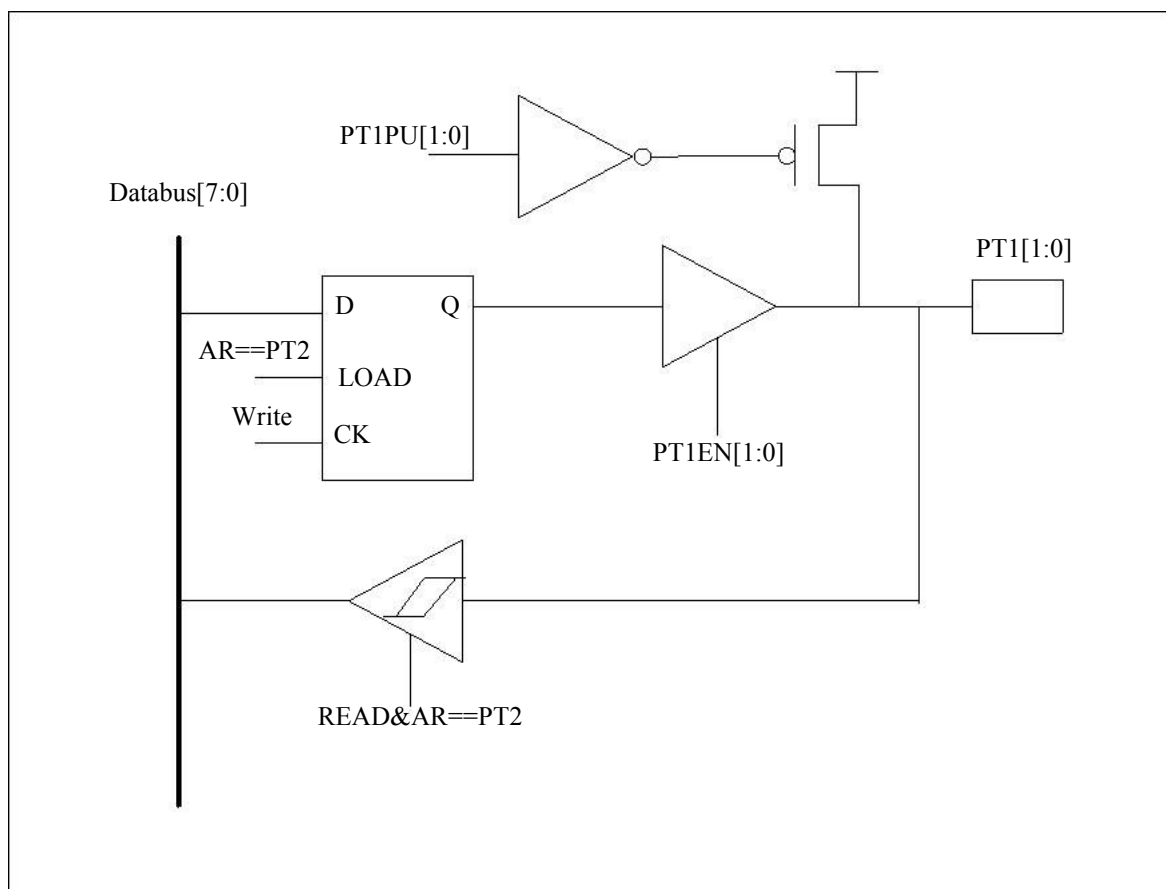


图 2-9 PT2[0]，PT2[1]功能框图

GPIO2口的 bit1与 bit0（PT2[1:0]）功能框图如图 2-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口 bit1与 bit0（PT2[1:0]）可作为外部中断接口 INT1与 INT0，或者作为普通 I/O口。通过控制 INTE寄存器的标志位 E0IE与 E1IE以决定是否使能中断。中断触发模式是由寄存器标志：E0M[1:0]，E1M[1:0]决定。这两个输入接口可以作为施密特触发，上/下触发电平分别为 0.45DVDD/0.2DVDD。

✓ 输出

CSU1281通过内部 D触发器输出数字数据。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（CSU1281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

CSU1281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[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
06H	INTF				--	--	--	E1IF	E0IF	uuu0u000
07H	INTE	GIE			--	--	--	E1IE	E0IE	0uu0u000
24H	PT2	PT2[7:0]								xxxxxxx
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000
27H	PT2MR					E1M[1:0]		E0M[1:0]		0u000000

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 DVDD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。

2.4.4 数字I/O口或者PDM输出：PT2[3:2]

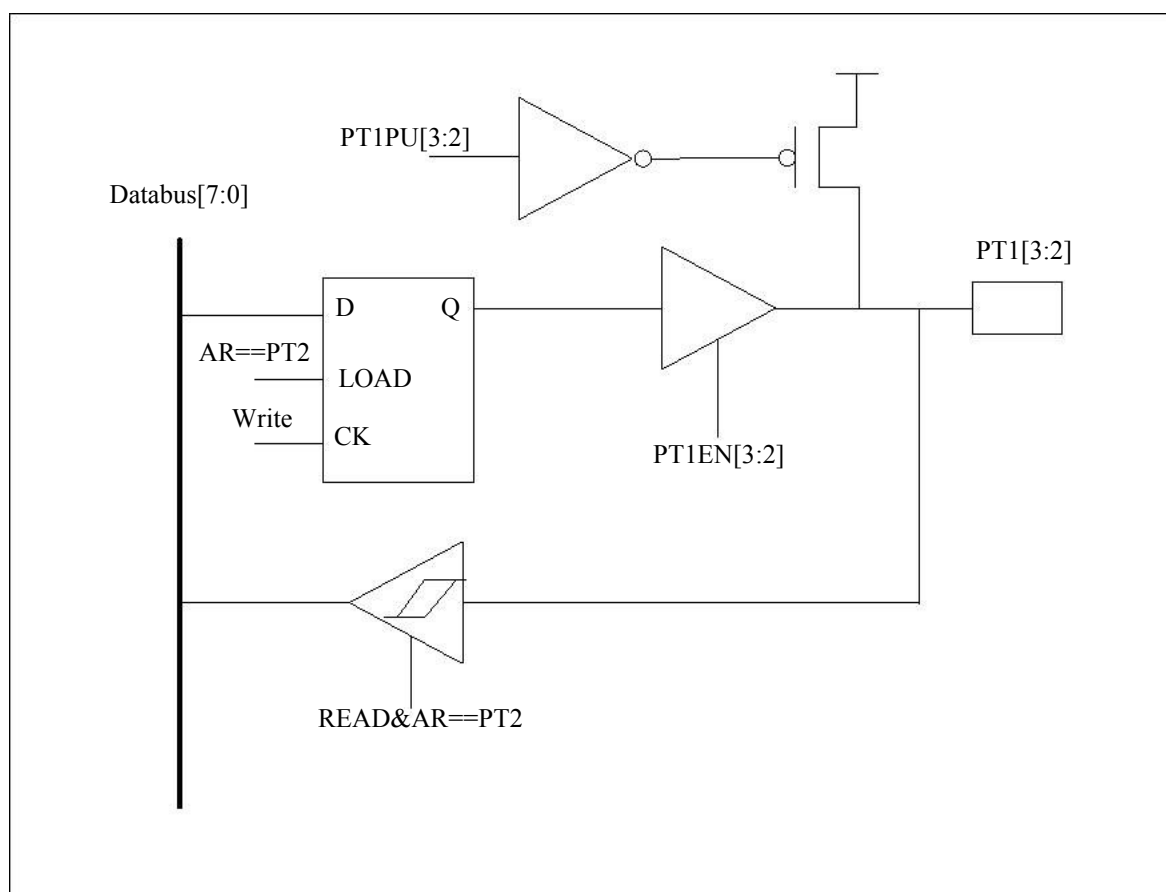


图 2-10 PT2[2]功能框图

GPIO2口 bit 2、bit3 (PT2[2]、PT2[3]) 的功能框图如图 2-10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[3:2]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 bit2、bit3 (PT2[2]、PT2[3]) 可以作为 PDM (脉冲密度调制) 输出接口，或者作为普通用途的 I/O口。通过设置寄存器标志 PM1EN及 PM2EN决定 PDM是否使能。PDM的详细使用描述请参阅 3.6。

✓ 输出

CSU1281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 (CSU1281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

CSU1281在 PT2口集成内部上拉电阻功能，上拉电阻大约为 $100K\Omega$ （上拉电流大约为 $30\mu A$ 。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
24H	PT2	PT2[7:0]								xxxxxxx
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000
27H	PT2MR	BZEN		PM2EN	PM1EN					0u000000

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 DVDD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。

2.4.5 数据 I/O 接口：PT2[6:4]

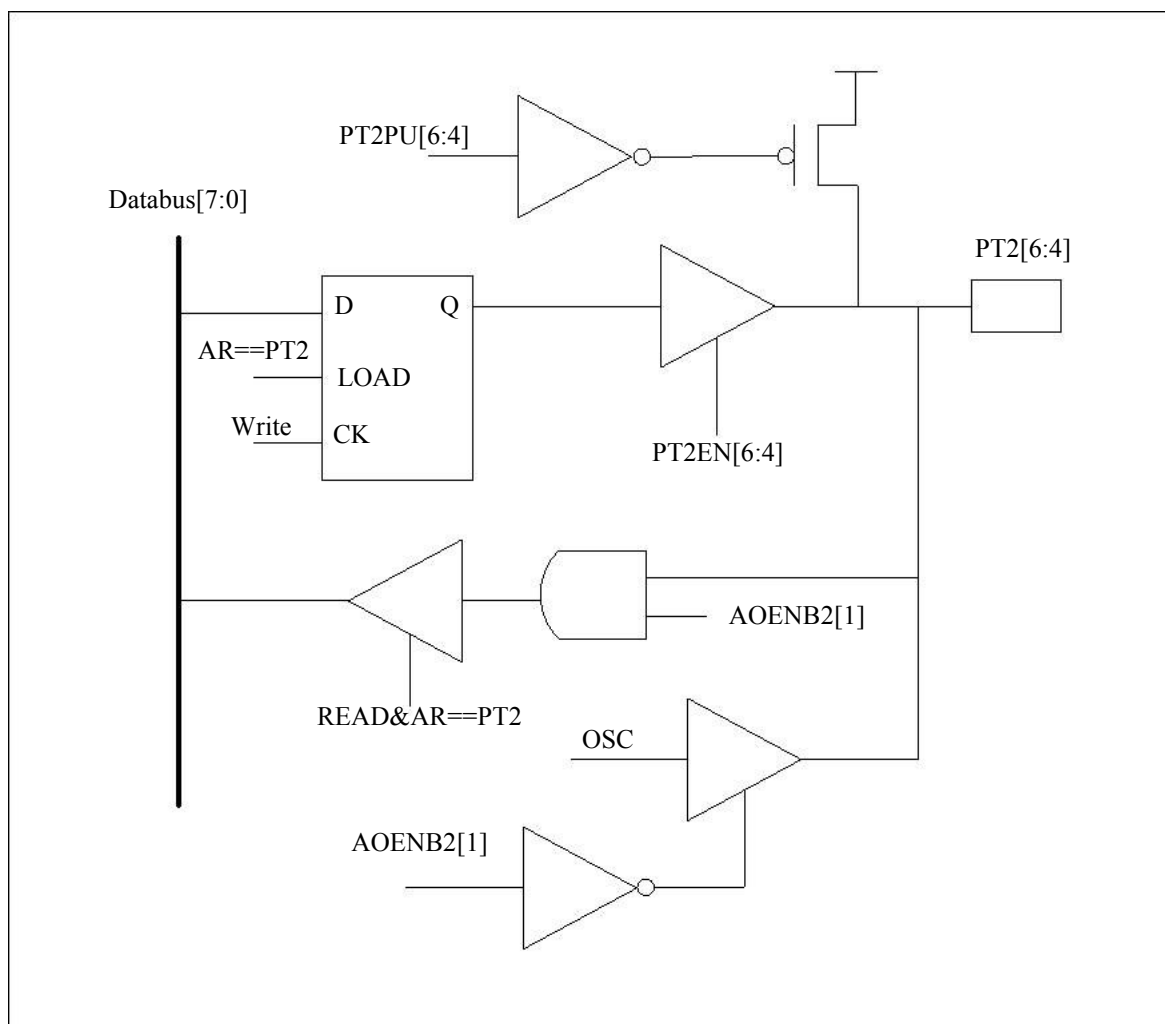


图 2-11 PT2[6:3]功能框图

GPIO2口 bit 6,bit5,bit4 (PT2[6:4]) 的功能框图如图 2-1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[6:3]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口 bit 6,bit5,bit4 (PT2[6:4]) 只能作为普通 I/O接口。

✓ 输出

CSU1281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 (CSU1281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

CSU1281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[6:4]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 2-25 PT2寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
24H	PT2	PT2[7:0]								xxxxxxx
25H	PT2EN	PT2EN[7:0]								00000000
26H	PT2PU	PT2PU[7:0]								00000000
27H	PT2MR	BZEN		PM1EN	PM2EN					0u000000

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。

注意操作：

1. 在 I/O口与 DVDD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。

2.4.6 数据I/O接口或者蜂鸣器输出: PT2[7]

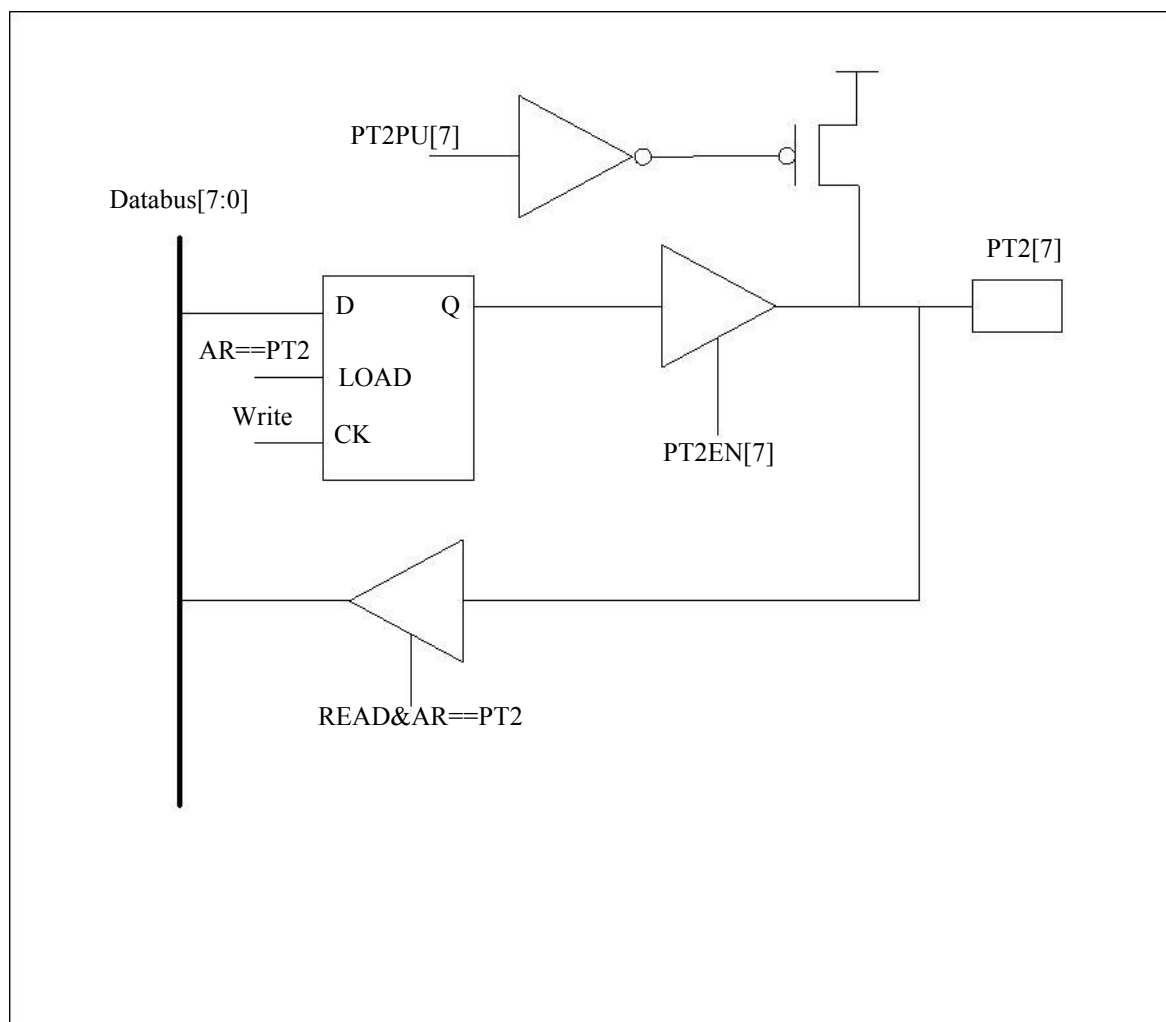


图 2-82 PT2[7]功能框图

GPIO2口 bit 7 (PT2[7])的功能框图如图 2-82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

CSU1281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（CSU1281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

CSU1281在 PT2口集成内部上拉电阻功能，上拉电阻大约为 100K Ω （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 2-26 PT2[7]寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
24H	PT2	PT2[7:0]								xxxxxxx
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 DVDD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。

2.4.7带LCD Segment驱动输出的的数字I/O口: PT3[7:0]

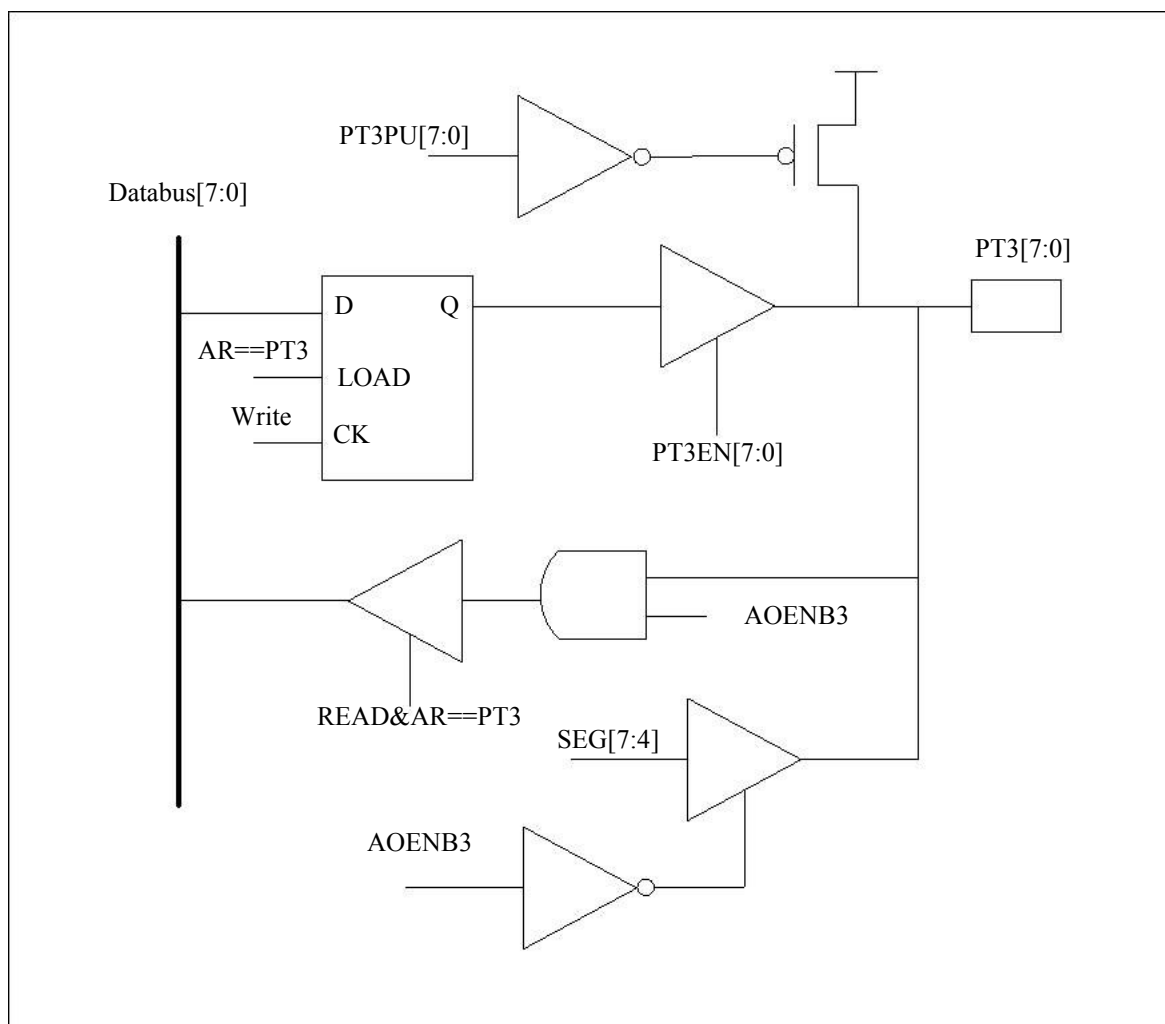


图 2-13 PT3[7:0]功能框图

GPIO3口(PT3[7:0])功能框图如图 2-13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 PT3EN[7:0]以决定接口是输入或输出。同时还可以作为 LCD的 Seg[22:15]的输出。由 AOENB3决定究竟是作为 I/O使用还是作为 LCD驱动输出使用。默认为 LCD驱动输出。输入与输出功能及相关的功能解释如下。

↙ 输入

GPIO3接口 bit 0~bit 7 (PT3[7:0]) 可用于输入数字。用户应该控制寄存器标志 PT3EN为 0。

↙ 输出

GPIO3接口 bit 0~bit 7 (PT3[7:0]) 可用于输入数字或者模拟信号输出 (仅限于 seg 信号输出) 用户应该控制寄存器标志 AOENB3[7:0] 决定输出信号的类型。如果 AOENB3 全被置位 (即为 1), GPIO3 接口中的与门允许数字信号连接到数据总线, 否则, 输出信号被定义为模拟信号。

CSU1281通过内部 D触发器输出数字信号。当程序通过 PT3输出数据时，数据首先被发送到数据总线，当有写信号及 AR（CSU1281内部器件地址指针）指向 PT1时，然后 D触发器会锁存数据从 PT3口输出。

↙ 上拉电阻

CSU1281在 PT3口集成内部上拉电阻功能, 上拉电阻大约为 100K Ω (上拉电流大约为 30 μ A。当程序要运行至睡眠模式之前, 须禁止 PT3PU)。可通过控制寄存器标志 PT3PU[7:0]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 2-27 PT3寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
28H	PT3	PT3[7:0]								00000000
29H	PT3EN	PT3EN[7:0]								00000000
2AH	PT3PU	PT3PU[7:0]								00000000
2BH	AOENB3	AOENB3[7:0]								00000000

读数据操作：

1. 清零寄存器标志位：PT3EN[n]（n是用户要控制的 bit）。PT3 [n]被定义为输入接口。
2. 置位寄存器标志位：PT3PU[n]。PT3 [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 PT3[n]获得数据。

写数据操作：

1. 置位寄存器标志 PT3EN[n]。PT3 [n]被定义为输出接口。
2. 置位相应的寄存器标志 PT3PU[n]。PT3 [n]连接到内部的上拉电阻。
3. 如果输出信号是数字信号，置位寄存器标志位：AOENB3[n]（n = 7~0）。
4. 如果输出信号是模拟信号，清零寄存器标志位：AOENB3[n]（n = 7~0）。
5. 需先使能 LCD电路，AON0~AON7才能正常工作。
6. 设置 PT3[n]作为数据输出，内部的 D触发器将锁存数据直到 PT3[n]的数据改变。

3 增强功能

3.1 电源系统

3.1.1 Regulator

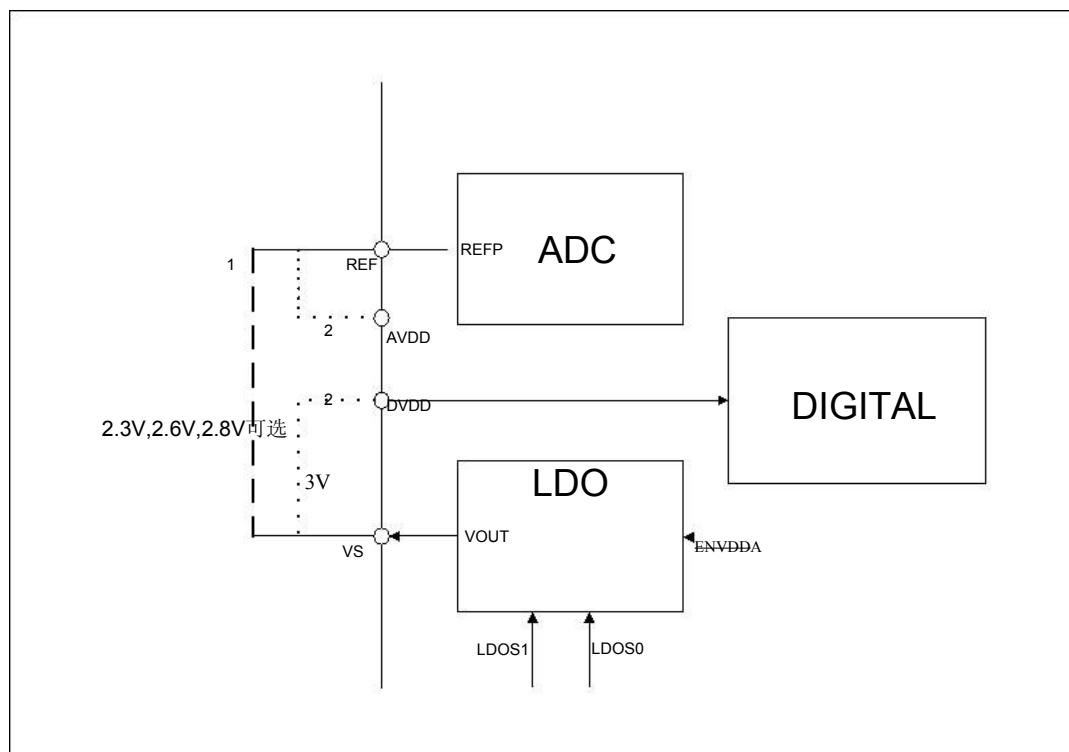


图 3-1 稳压电路

当输入电源电压小于 3.3V 时，LDO 主要应用于方案 1，如图 3-1 稳压电路所示，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，2.6V，2.8V，3V 可选。当输入电源电压为 5V 时，LDO 主要应用于方案 2，如图 3-1 稳压电路所示，LDO 的输出 3V 电压作为芯片数字部分的供电电源，此时 ADC 和传感器的参考源选择 AVDD。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 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]								00uu000u
1DH	NETF			ENVDDA					ENVB	uu0uu000

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=AVDD

LDOS[1:0] 01 VS=2.8

LDOS[1:0] 10 VS=2.6

LDOS[1:0] 11 VS=2.3

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	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

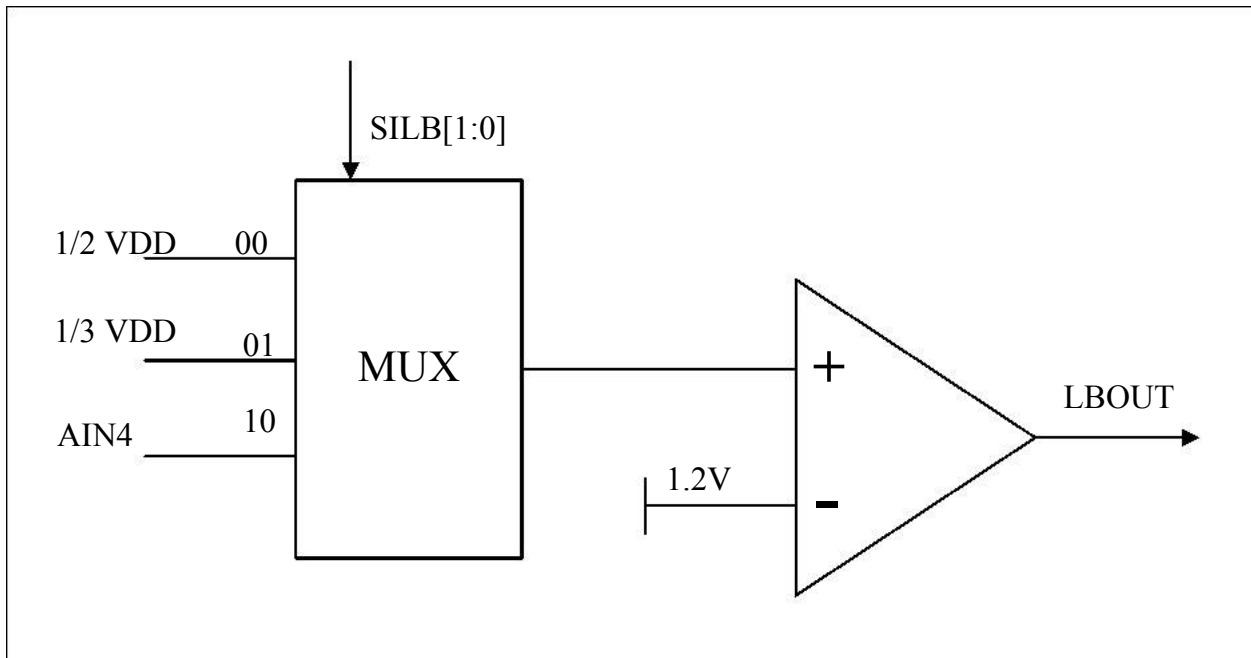


图 3-2 低电压比较功能模块框图

低电压比较器用于 $DVDD$ 的低电压检测。CSU1281 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 $SILB[1:0]$ 及 $ENLB$ ，比较器的输出是 $LBOUT$ ， $LBOUT$ 为只读。请看图 3-2 低电压比较功能模块框图。

表 3-2 低电压比较器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
1CH	NETE					$SILB[1:0]$		ENLB		00uu000u
1FH	SVD								LBOUT	uuuuuuux

Bit1 ENLB：低电压比较器使能标志位

ENLB=1：低电压比较器使能

ENLB=0：低电压比较器不使能

表 3-3 低电压比较器检测电压的选择列表

$SILB[1:0]$	检测电压	满足条件	则
00	$1/2DVDD$	$DVDD < 2.4V$	$LBOUT = 0$
01	$1/3DVDD$	$DVDD < 3.6V$	$LBOUT = 0$
10	AIN4	$AIN4 < 1.2V$	$LBOUT = 0$

操作：

低电压比较器必须要在 $ENVB$ 打开的情况下才能工作

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

3.2 Halt与Sleep模式

CSU1281支持低电压工作模式。为了使 CSU1281处于待机状态，可以让 CPU停止工作使 CSU1281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小于 3uA。

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 0FFH
MOVWF PT1EN    ;PT1[7:0]用作输出接口
CLRF PT1       ;将 PT1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 CSU1281进行睡眠模式
NOP            ;保证 CPU重启后程序能正常工作
```

采用外部高速晶振工作时，进入 sleep模式需要如下操作：

```
MCK x1xx xxxxb ;M6_CK配置成 1关闭外部晶振
PCK xxxx x0xxb ;PCK的 bit2 EO_SLP要清零，睡眠时要关闭外部晶振
MCK xxx0 xxxxb ;M4_CK要清 0切换成外部低速晶振
```

3.3 看门狗(WDT)

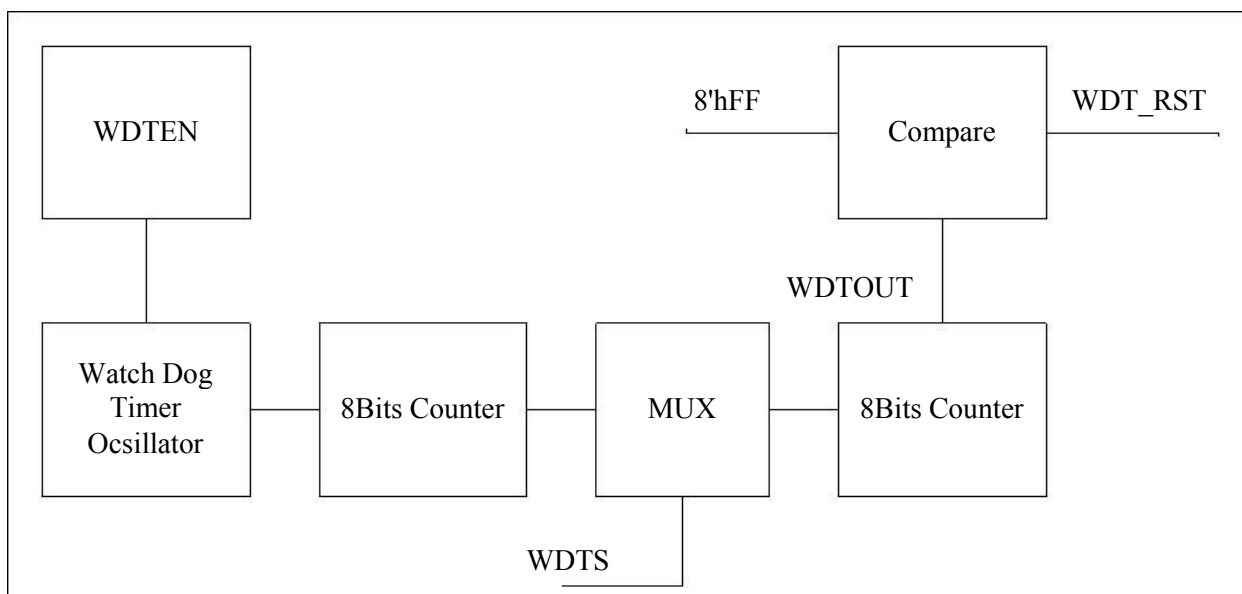


图 3-3看门狗定时器功能框图

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

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 3-3看门狗定时器功能框图所示。“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。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				00u00000
0DH	WDTCON	WDTEN					WDTS[2:0]			0uuuu000

操作：

1. 设置 WDTS[2:0]，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。

表 3-5看门狗计数器选择列表

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

3.4 ADC模块

ADC外围电路典型连接方式

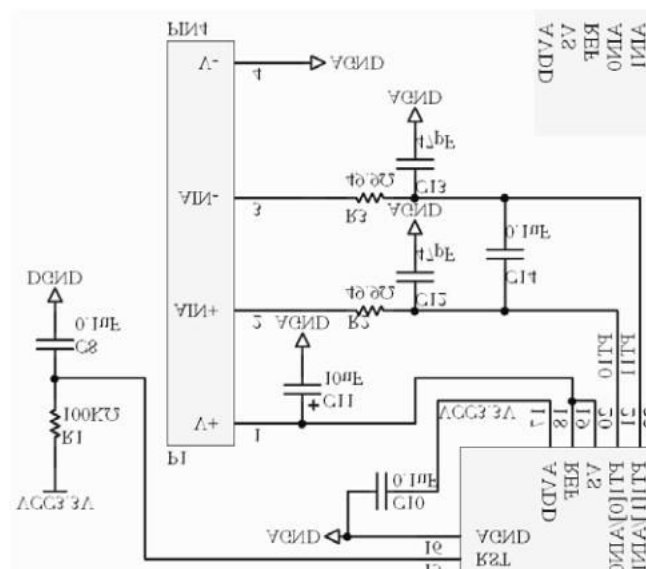


图 3-4采用内部参考电源给传感器供电应用实例

采用内部参考电源给传感器供电时外部电路接法。此时，需将寄存器NETF的ENVDDA(5 bit)置高，使能内部的参考电压VS，并将VS和REF端连接在一起，用VS给传感器提供电源

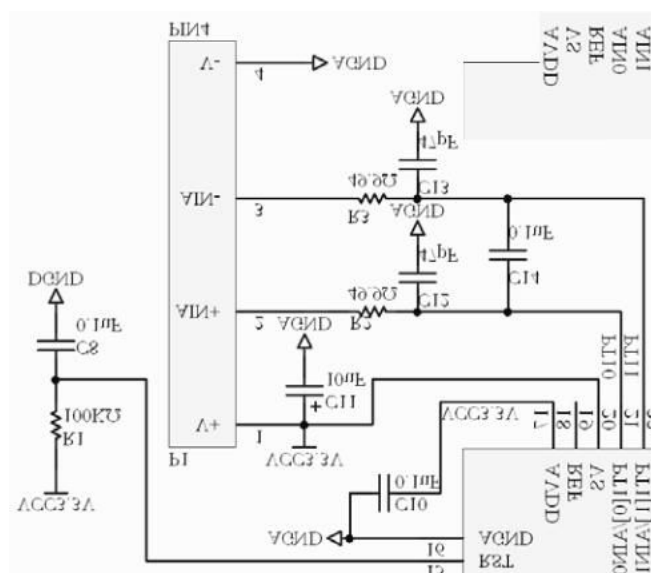


图 3-5采用外部参考电源给传感器供电应用实例

采用外部参考电源给传感器供电时外部电路接法。此时，REF接到外部参考电源。将寄存器NETF的ENVDDA(5 bit)置低可关闭内部的参考电压，此时由外部电源给传感器提供电源，同时提供给芯片参考电压REF。

表 3-6 ADC功能模块相关寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
06H	INTF				--	--	ADIF	--	--	uuu0u000
07H	INTE	GIE					ADIE	--	--	0uu0u000
10H	ADOH	ADO[23:16]								00000000
11H	ADOL	ADO[15:8]								00000000
12H	ADOLL	ADO[7:0]								00000000
13H	ADCON						ADM[2:0]			uuuuu000
14H	MCK	--	--	--	--	--	--	M1_CK	--	00000000
18H	NETA	SINL[1:0]		SENS[1:0]						0000uuuu
1AH	NETC	CHOPM[1:0]				ADG[1:0]		ADEN		00uu000u
1DH	NETF						BGID[1:0]			uu0uu000

ADOH寄存器（地址=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=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=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]：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

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

表 3-7 ADC采样频率选择寄存器

特性	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
MCK							M1_CK	
14H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

表 3-8 ADC采样频率选择列表

M1_CK	ADC采样频率 (ADCF)
0	MCK/16
1	MCK/32

表 3-9 ADC输出速率选择寄存器

特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ADCON							ADM[2:0]	
13H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

表 3-10 ADC输出速率选择列表

ADM[2:0]	ADC输出速率
000	ADCF/128
001	ADCF/256
010	ADCF/512
011	ADCF/1024
100	ADCF/2048
101	ADCF/4096
110	ADCF/8192
111	ADCF/8192

NETA寄存器（地址=18H）

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

Bit 7-6 SINL[1:0]: ADC输入端的信号选择器

11 = ADC选择单端输入信号（如图 3-6）

10 = ADC输入端连接到 TEMP

01 = ADC输入端连接到 AIN2和 AIN3

00 = ADC输入端连接到 AIN0和 AIN1

Bit 5-4 SenS[1:0]:

00 =选择 VINP =AIN0, VINN=AIN4

01 =选择 VINP =AIN1, VINN=AIN4

10 =选择 VINP =AIN2, VINN=AIN4

11 =选择 VINP =AIN3, VINN=AIN4

输入选择电路主要用于选择输入信号的通道以及实现温度传感器的功能，同时 CSU1281的输入选择电路可以实现 4路半桥传感器共用时对单端输入信号的处理。Sinl[1:0]主要用于选择 AIN0和 AIN1、AIN2和 AIN3、温度传感器差分输出以及单端输入信号等四种情况作为 ADC的差分输入信号。当 Sinl[1:0]为 11时，通过选择 SenS[1: 0]来分别对单端输入信号 AIN0, AIN1, AIN2, AIN3进行 AD转换，而 AIN4用作单端输入信号的共模电压，考虑温度特性和输入阻抗匹配，AIN4由相同的半桥传感器输出信号提供。ADC不同通道之间切换时间为 3次 ADC中断间隔，即每次切换通道之后要忽略前 2次 ADC中断的数据。

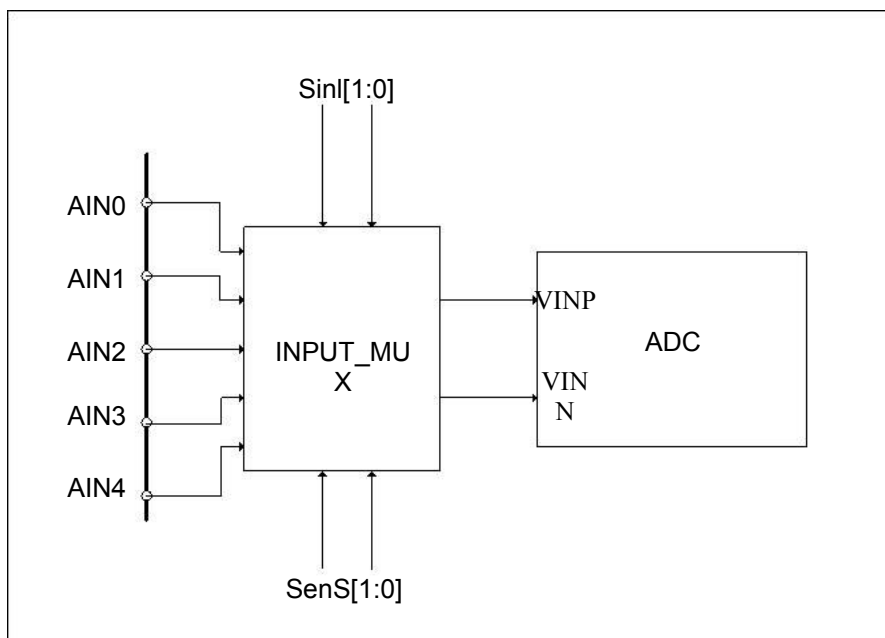


图 3-6 ADC单端输入选择信号

NETC寄存器（地址=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

Bit 7-6 CHOPM: ADC随机斩波选择

11 =关闭斩波

10 =随机斩波。

01 = 1/2采样频率斩波，时钟上升沿为原采样时钟高电平的中间。

00 = 1/2采样频率斩波

Bit 3-2 ADG[1:0]: 内部 ADC输入增益

11 =内部 ADC的输入增益是 256

10 =内部 ADC的输入增益是 128

01 =内部 ADC的输入增益是 64

00 =内部 ADC的输入增益是 1

Bit 1 ADEN: ADC使能标志

1 = ADC使能

0 = ADC不使能

NETF寄存器（地址=1DH）

特性	U-0	U-0	R/W-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

Bit 2-1 BGID[1:0]: 运放偏置电流选择

11 =内部 ADC的运放偏置电流增大 100%

10 =内部 ADC的运放偏置电流增大 50%

01 =内部 ADC的运放偏置电流增大 50%

00 =内部 ADC的运放偏置电流为默认值

内部温度传感器

在使用内部温度传感器时，需要将 ADC输入端的信号选择器（SINL）置为 10，即选择内部温度传感器输入，并将内部 ADC输入增益调整为 1，如果选择其他的增益值会出现 ADC值溢出现象。如果从其他 ADC通道切换到内部温度传感器通道时，需要忽略前 2次 ADC中断（因为 ADC的通道切换时间为 3次 ADC中断间隔）。

在 PGA = 1；基准源：3.3V。

内部温度传感器输出信号	400uV/°C
输出信号最大误差	8.1%

3.5 LCD Driver

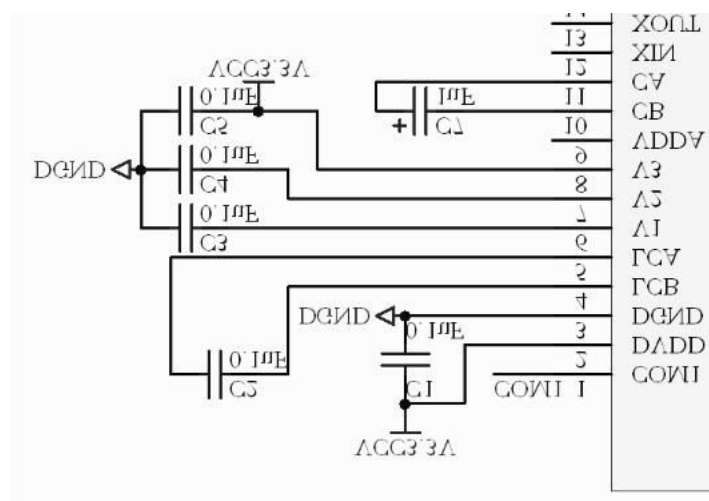


图 3-7采用外部电源给 LCD供电应用实例

采用外部电源给LCD供电时外部电路接法。此时，需将寄存器LCDENR的ENPMPL(0 bit)置零，关掉内部的LCD供电系统。

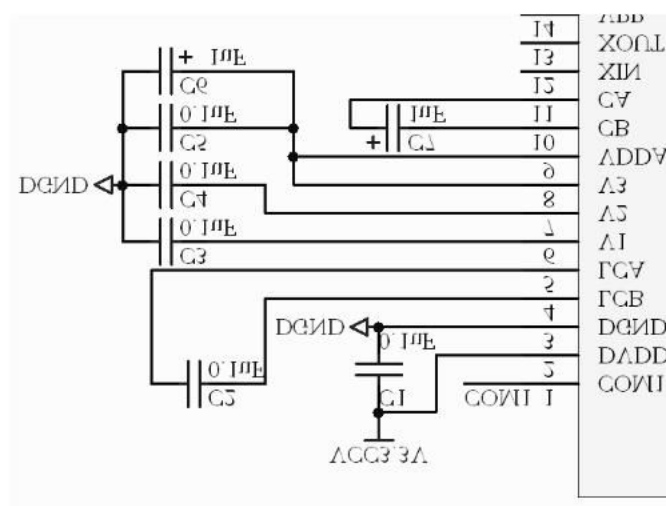


图 3-8采用内部电源给 LCD供电应用实例

采用内部电源给LCD供电时外部电路接法。此时，需将寄存器LCDENR的ENPMPL(0 bit)置1，打开内部的LCD供电系统，并将VDDA和V3相连，由内部电源VDDA提供电源给LCD供电。

LCD驱动器有 4种控制模式：1/8duty，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 3-11 LCD的 duty选择列表

LCD_DUTY[1:0]	控制模式	SEG1-22							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00	1/8duty	COM8	COM7	COM6	COM5	COM4	COM3	COM2	COM1
01	1/2duty	-	-	--	--	-	-	COM2	COM1
10	1/3duty	-	--	--	--	-	COM3	COM2	COM1
11	1/4duty	--	--	--	--	COM4	COM3	COM2	COM1

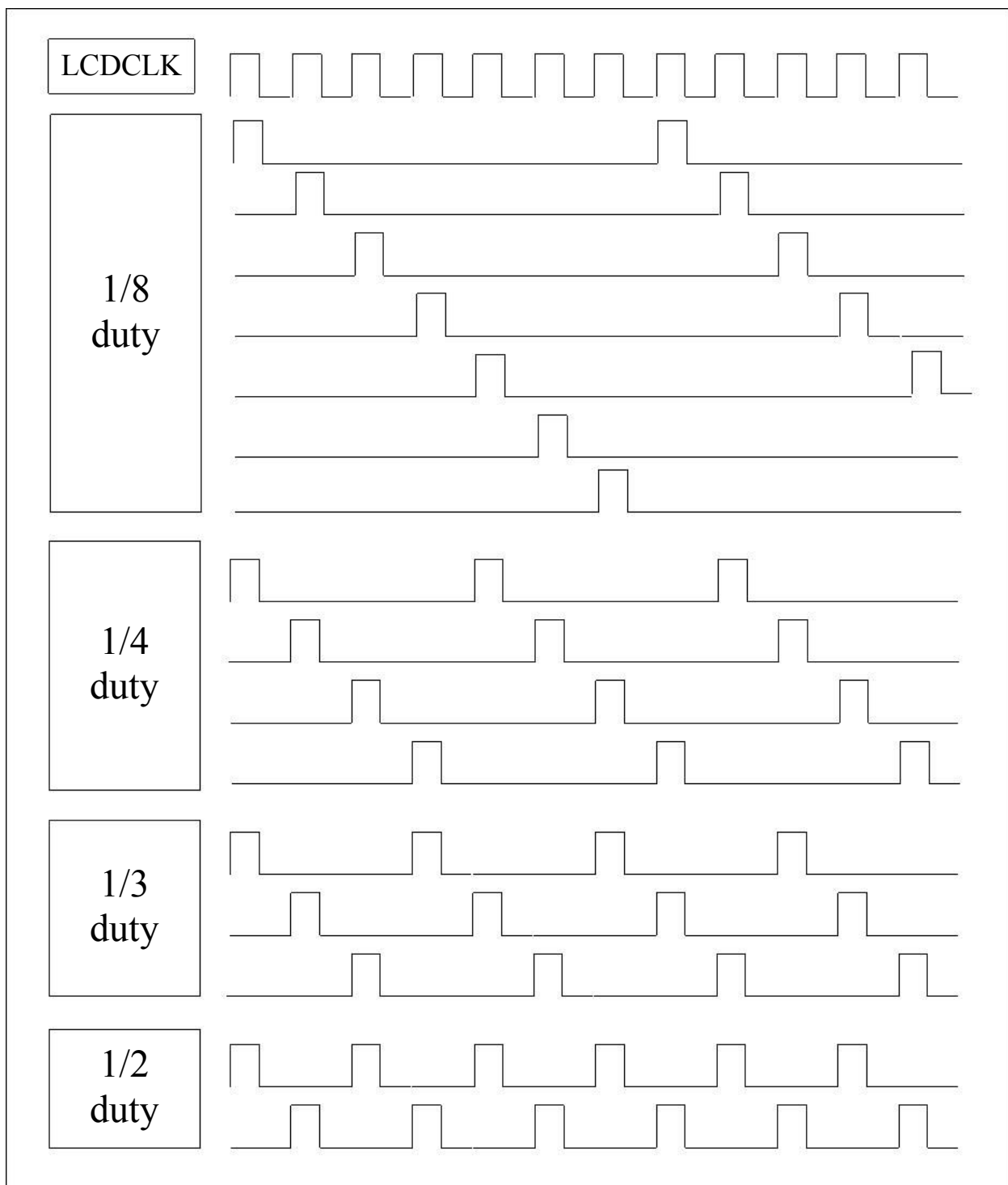
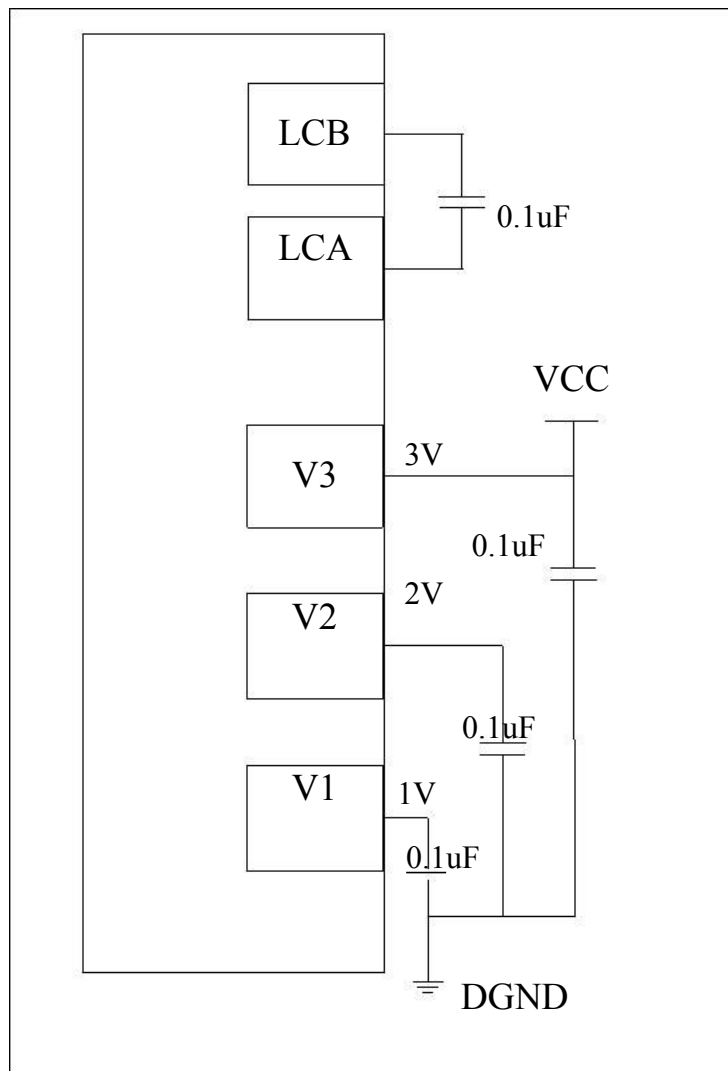


图 3-9 LCD的 duty模式工作周期

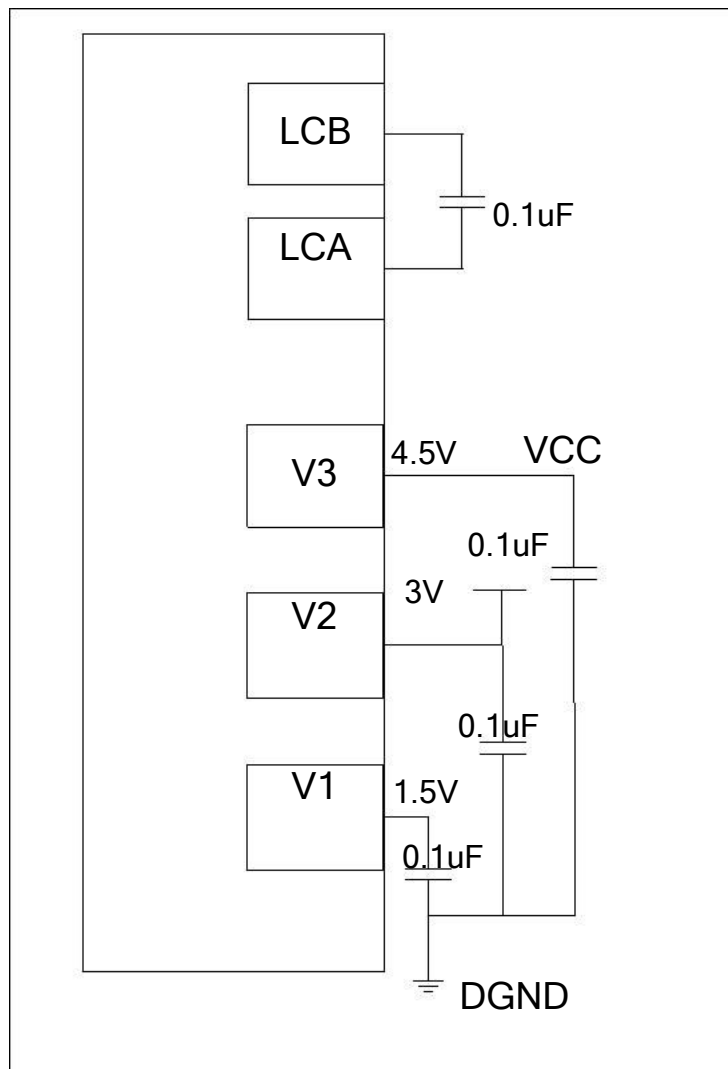
LCD的帧频率可以通过设置寄存器标志 LCDCKS[1:0]确定。CSU1281对 LCD模块的输入时钟进行分频以获得 LCDCLK。

LCD驱动器有 3个偏置电压接口，V1、V2及 V3，有 2种电源模式：1/3bias、1/2bias。请看下面描述，设置 LCD的电源系统。

1/3bias电源系统



(A)



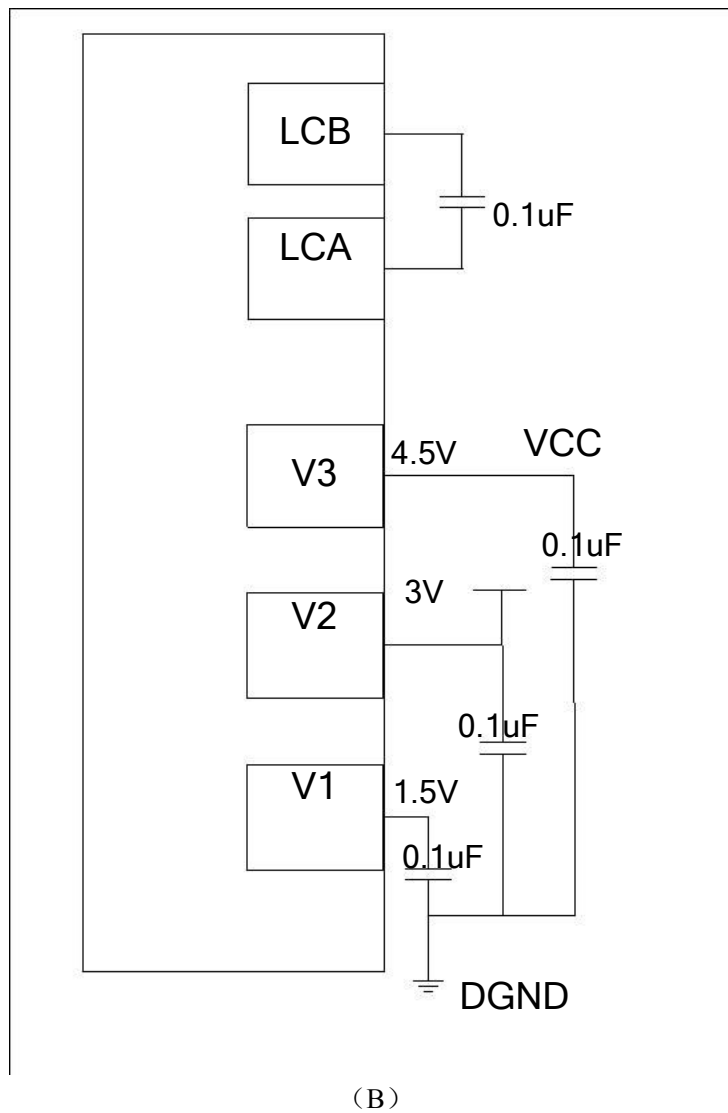
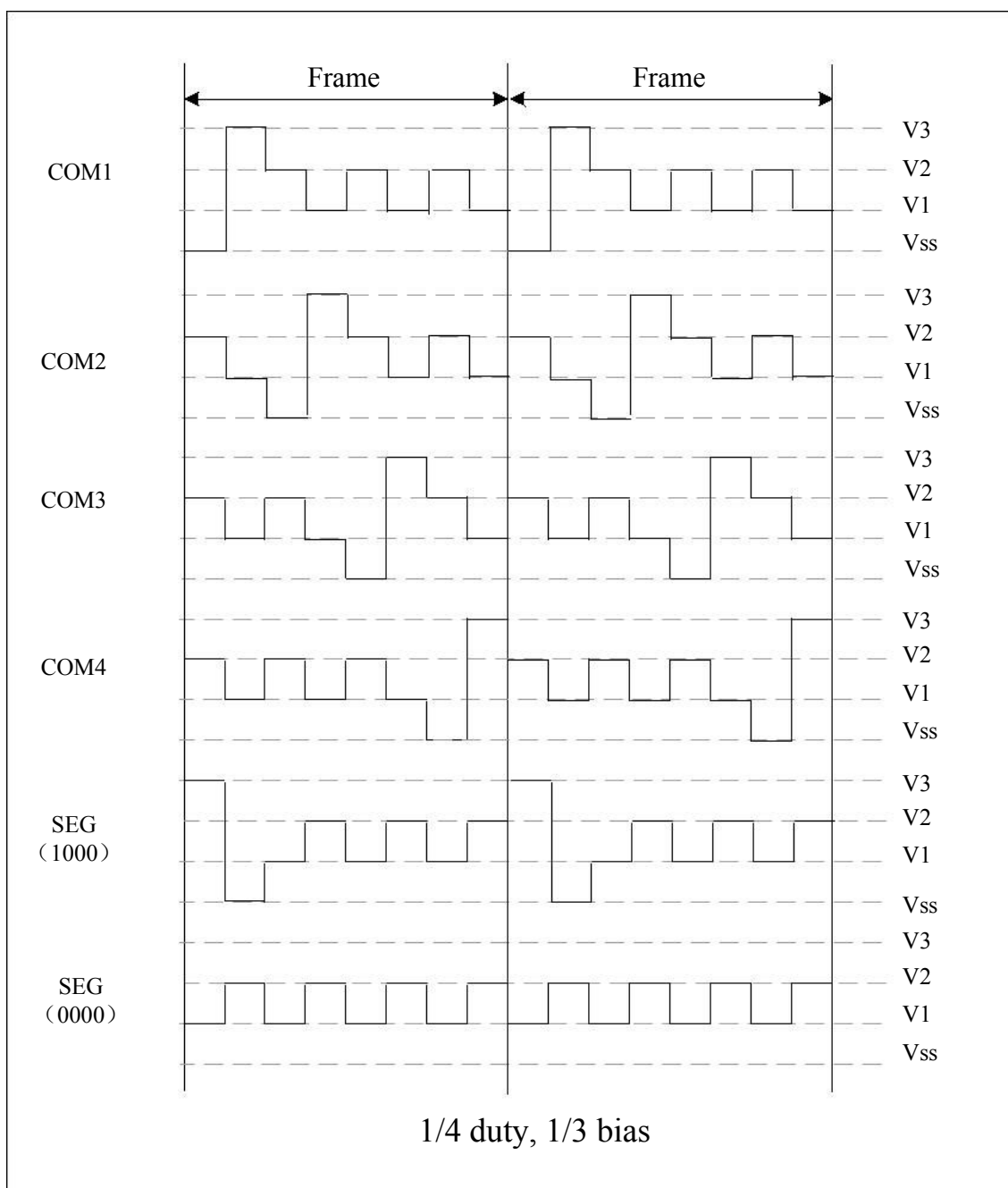
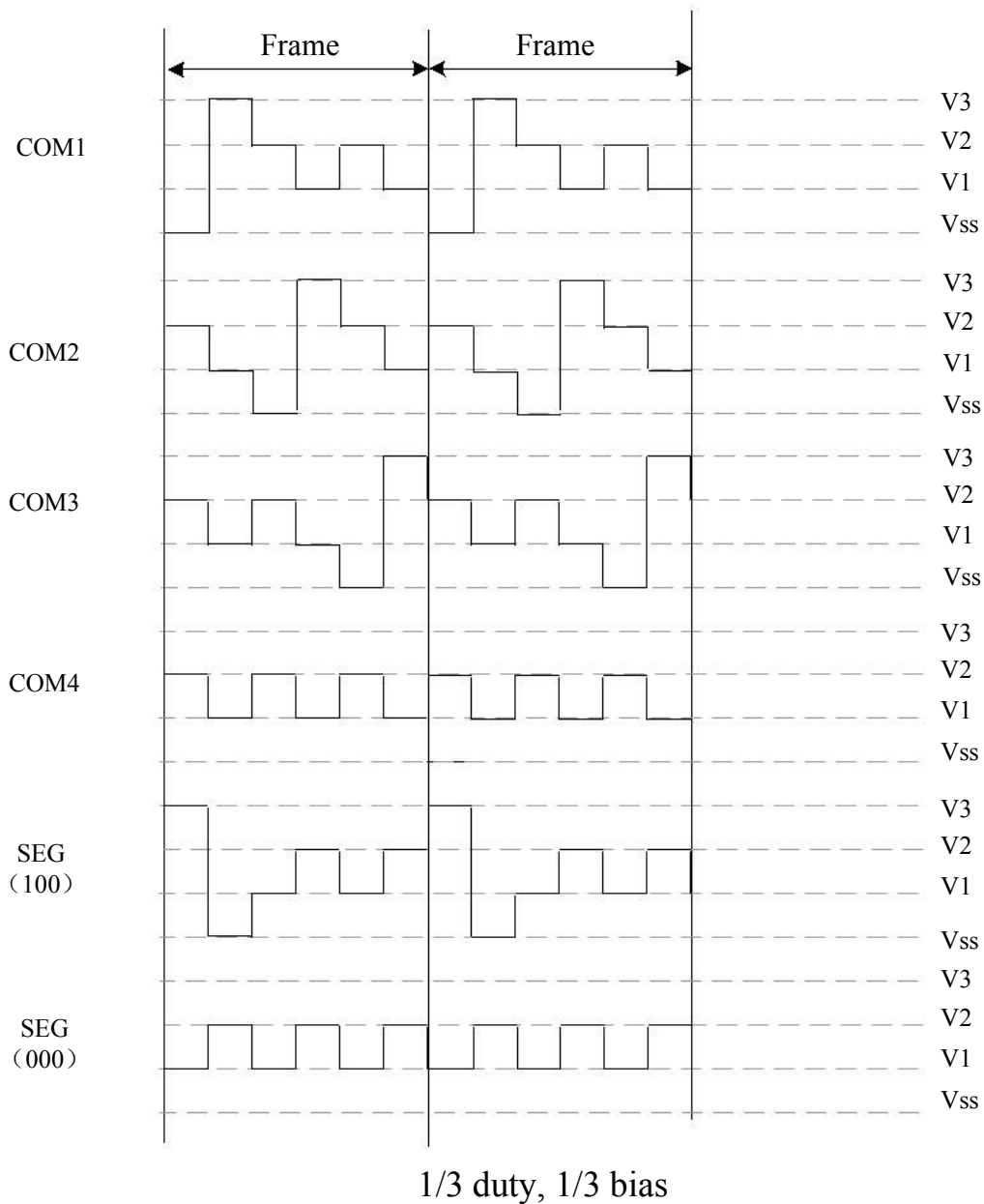


图 3-10 LCD的 1/3bias电源系统电路连接图



(A)



(B)

图 3-11 LCD的 1/3bias电源系统的时钟

✓ 1/2bias电源系统

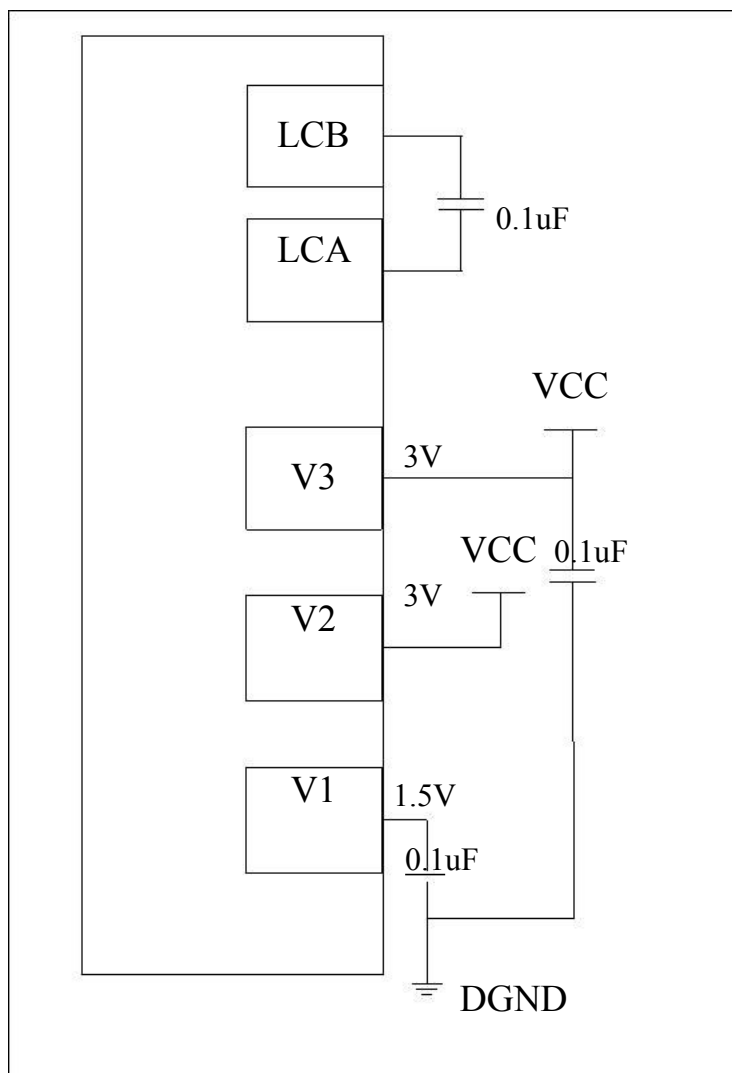
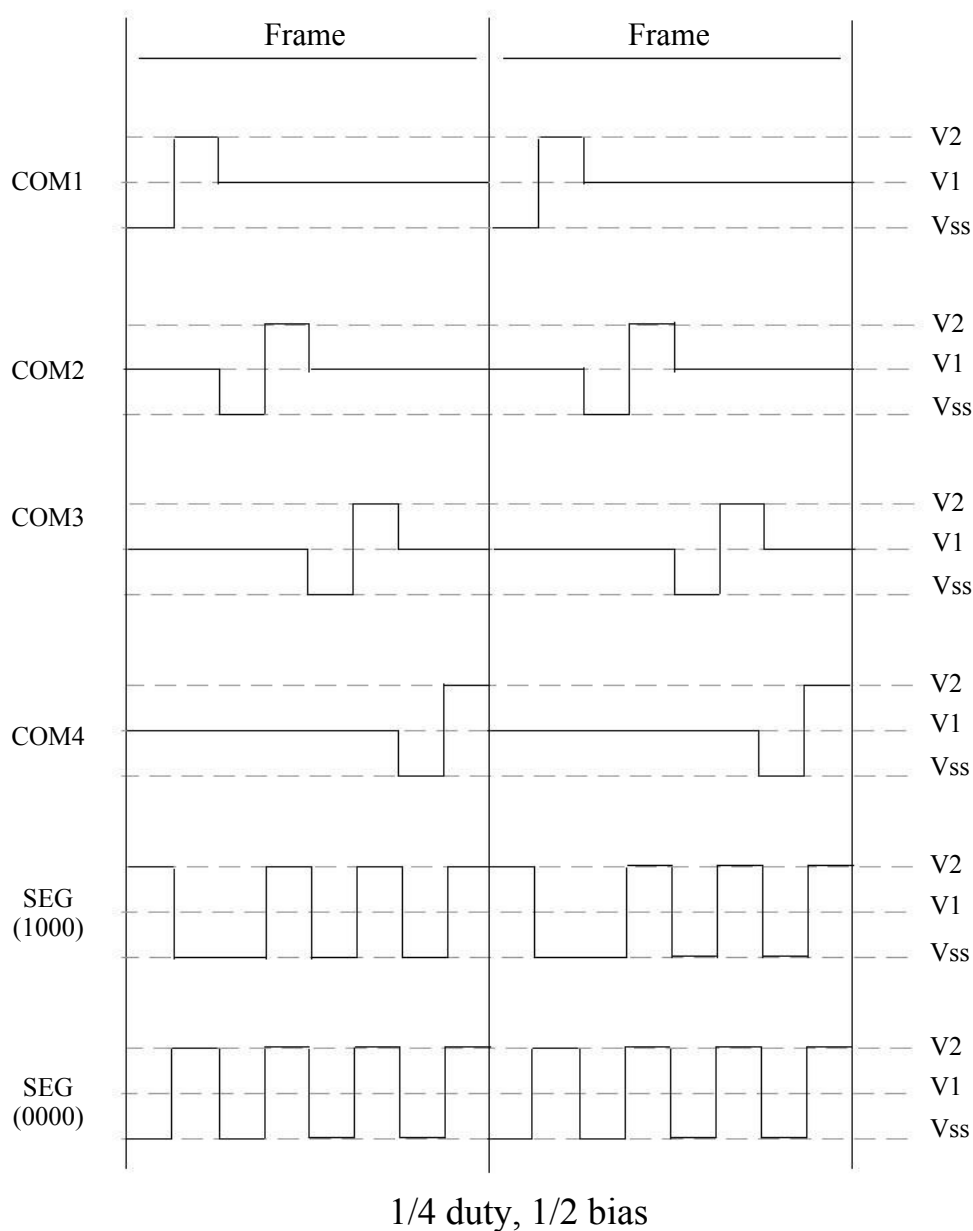
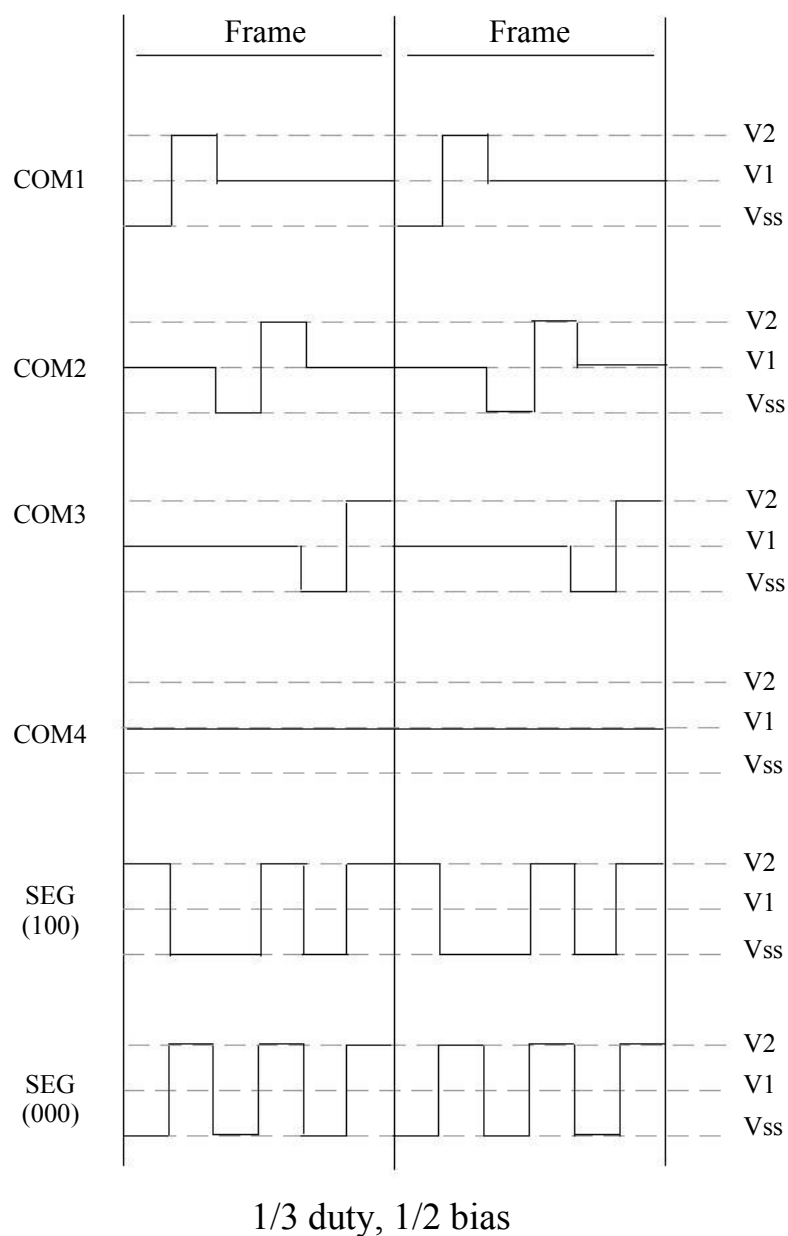


图 3-12 LCD的 1/2bias电源系统电路连接图



(A)



(B)

图 3-13 LCD的 1/2bias电源系统的时钟

表 3-12 CSU1281的 LCD驱动器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
40H	LCD1	SEG1[7:4](仅对 8COM)				SEG1[3:0]				00000000
41H	LCD2	SEG2[7:4](仅对 8COM)				SEG2[3:0]				00000000
42H	LCD3	SEG3[7:4](仅对 8COM)				SEG3[3:0]				00000000
43H	LCD4	SEG4[7:4](仅对 8COM)				SEG4[3:0]				00000000
44H	LCD5	SEG5[7:4](仅对 8COM)				SEG5[3:0]				00000000
45H	LCD6	SEG6[7:4](仅对 8COM)				SEG6[3:0]				00000000
46H	LCD7	SEG7[7:4](仅对 8COM)				SEG7[3:0]				00000000
47H	LCD8	SEG8[7:4](仅对 8COM)				SEG8[3:0]				00000000
48H	LCD9	SEG9[7:4](仅对 8COM)				SEG9[3:0]				00000000
49H	LCD10	SEG10[7:4](仅对 8COM)				SEG10[3:0]				00000000
4AH	LCD11	SEG11[7:4](仅对 8COM)				SEG11[3:0]				00000000
4BH	LCD12	SEG12[7:4](仅对 8COM)				SEG12[3:0]				00000000
4CH	LCD13	SEG13[7:4](仅对 8COM)				SEG13[3:0]				00000000
4DH	LCD14	SEG14[7:4](仅对 8COM)				SEG14[3:0]				00000000
4EH	LCD15	SEG15[7:4](仅对 8COM)				SEG15[3:0]				00000000
4FH	LCD16	SEG16[7:4](仅对 8COM)				SEG16[3:0]				00000000
50H	LCD17	SEG17[7:4](仅对 8COM)				SEG17[3:0]				00000000
51H	LCD18	SEG18[7:4](仅对 8COM)				SEG18[3:0]				00000000
52H	LCD19	SEG19[7:4](仅对 8COM)				SEG19[3:0]				00000000
53H	LCD20	SEG20[7:4](仅对 8COM)				SEG20[3:0]				00000000
54H	LCD21	SEG21[7:4](仅对 8COM)				SEG21[3:0]				00000000
55H	LCD22	SEG22[7:4](仅对 8COM)				SEG22[3:0]				00000000
56H	LCD23					SEG23[3:0](仅对 4COM)				uuuu0000
57H	LCD24					SEG24[3:0](仅对 4COM)				uuuu0000
58H	LCDENR	LCDCKS[1:0]		LCDEN		LEVEL	LCD_DUTY[1:0]		ENPMPL	000u0110

LCD1寄存器（地址=40H）

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
LCD1	SEG1[7:4]				SEG1[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG1[7]: LCD驱动器控制信号: SEG1带 COM8。
 SEG1[6]: LCD驱动器控制信号: SEG1带 COM7。
 SEG1[5]: LCD驱动器控制信号: SEG1带 COM6。
 SEG1[4]: LCD驱动器控制信号: SEG1带 COM5。
 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=41H）

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
LCD2	SEG2[7:4]				SEG2[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG2[7]: LCD驱动器控制信号: SEG2带 COM8。
 SEG2[6]: LCD驱动器控制信号: SEG2带 COM7。
 SEG2[5]: LCD驱动器控制信号: SEG2带 COM6。
 SEG2[4]: LCD驱动器控制信号: SEG2带 COM5。
 Bit 3-0 SEG2[3]: LCD驱动器控制信号: SEG2带 COM4。
 SEG2[2]: LCD驱动器控制信号: SEG2带 COM3。
 SEG2[1]: LCD驱动器控制信号: SEG2带 COM2。
 SEG2[0]: LCD驱动器控制信号: SEG2带 COM1。

LCD3寄存器（地址=42H）

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
LCD3	SEG3[7:4]				SEG3[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG3[7]: LCD驱动器控制信号: SEG3带 COM8。
 SEG3[6]: LCD驱动器控制信号: SEG3带 COM7。
 SEG3[5]: LCD驱动器控制信号: SEG3带 COM6。
 SEG3[4]: LCD驱动器控制信号: SEG3带 COM5。
 Bit 3-0 SEG3[3]: LCD驱动器控制信号: SEG3带 COM4。
 SEG3[2]: LCD驱动器控制信号: SEG3带 COM3。
 SEG3[1]: LCD驱动器控制信号: SEG3带 COM2。
 SEG3[0]: LCD驱动器控制信号: SEG3带 COM1。

LCD4寄存器（地址=43H）

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
LCD4	SEG4[7:4]				SEG4[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG4[7]: LCD驱动器控制信号: SEG4带 COM8。
 SEG4[6]: LCD驱动器控制信号: SEG4带 COM7。
 SEG4[5]: LCD驱动器控制信号: SEG4带 COM6。
 SEG4[4]: LCD驱动器控制信号: SEG4带 COM5。
 Bit 3-0 SEG4[3]: LCD驱动器控制信号: SEG4带 COM4。
 SEG4[2]: LCD驱动器控制信号: SEG4带 COM3。
 SEG4[1]: LCD驱动器控制信号: SEG4带 COM2。
 SEG4[0]: LCD驱动器控制信号: SEG4带 COM1。

LCD5寄存器（地址=44H）

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
LCD5	SEG5[7:4]				SEG5[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG5[7]: LCD驱动器控制信号: SEG5带 COM8。
 SEG5[6]: LCD驱动器控制信号: SEG5带 COM7。
 SEG5[5]: LCD驱动器控制信号: SEG5带 COM6。
 SEG5[4]: LCD驱动器控制信号: SEG5带 COM5。
 Bit 3-0 SEG5[3]: LCD驱动器控制信号: SEG5带 COM4。
 SEG5[2]: LCD驱动器控制信号: SEG5带 COM3。
 SEG5[1]: LCD驱动器控制信号: SEG5带 COM2。
 SEG5[0]: LCD驱动器控制信号: SEG5带 COM1。

LCD6寄存器（地址=45H）

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
LCD6	SEG6[7:4]				SEG6[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG6[7]: LCD驱动器控制信号: SEG6带 COM8。
 SEG6[6]: LCD驱动器控制信号: SEG6带 COM7。
 SEG6[5]: LCD驱动器控制信号: SEG6带 COM6。
 SEG6[4]: LCD驱动器控制信号: SEG6带 COM5。
 Bit 3-0 SEG6[3]: LCD驱动器控制信号: SEG6带 COM4。
 SEG6[2]: LCD驱动器控制信号: SEG6带 COM3。
 SEG6[1]: LCD驱动器控制信号: SEG6带 COM2。
 SEG6[0]: LCD驱动器控制信号: SEG6带 COM1。

LCD7寄存器（地址=46H）

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
LCD7	SEG7[7:4]				SEG7[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG7[7]: LCD驱动器控制信号: SEG7带 COM8。
 SEG7[6]: LCD驱动器控制信号: SEG7带 COM7。
 SEG7[5]: LCD驱动器控制信号: SEG7带 COM6。
 SEG7[4]: LCD驱动器控制信号: SEG7带 COM5。
 Bit 3-0 SEG7[3]: LCD驱动器控制信号: SEG7带 COM4。
 SEG7[2]: LCD驱动器控制信号: SEG7带 COM3。
 SEG7[1]: LCD驱动器控制信号: SEG7带 COM2。
 SEG7[0]: LCD驱动器控制信号: SEG7带 COM1。

LCD8寄存器（地址=47H）

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
LCD8	SEG8[7:4]				SEG8[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG8[7]: LCD驱动器控制信号: SEG8带 COM8。
 SEG8[6]: LCD驱动器控制信号: SEG8带 COM7。
 SEG8[5]: LCD驱动器控制信号: SEG8带 COM6。
 SEG8[4]: LCD驱动器控制信号: SEG8带 COM5。
 Bit 3-0 SEG8[3]: LCD驱动器控制信号: SEG8带 COM4。
 SEG8[2]: LCD驱动器控制信号: SEG8带 COM3。
 SEG8[1]: LCD驱动器控制信号: SEG8带 COM2。
 SEG8[0]: LCD驱动器控制信号: SEG8带 COM1。

LCD9寄存器（地址=48H）

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
LCD9	SEG9[7:4]				SEG9[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG9[7]: LCD驱动器控制信号: SEG9带 COM8。
 SEG9[6]: LCD驱动器控制信号: SEG9带 COM7。
 SEG9[5]: LCD驱动器控制信号: SEG9带 COM6。
 SEG9[4]: LCD驱动器控制信号: SEG9带 COM5。
 Bit 3-0 SEG9[3]: LCD驱动器控制信号: SEG9带 COM4。
 SEG9[2]: LCD驱动器控制信号: SEG9带 COM3。
 SEG9[1]: LCD驱动器控制信号: SEG9带 COM2。
 SEG9[0]: LCD驱动器控制信号: SEG9带 COM1。

LCD10寄存器（地址=49H）

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
LCD10	SEG10[7:4]				SEG10[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG10[7]: LCD驱动器控制信号: SEG10带 COM8。
 SEG10[6]: LCD驱动器控制信号: SEG10带 COM7。
 SEG10[5]: LCD驱动器控制信号: SEG10带 COM6。
 SEG10[4]: LCD驱动器控制信号: SEG10带 COM5。
 Bit 3-0 SEG10[3]: LCD驱动器控制信号: SEG10带 COM4。
 SEG10[2]: LCD驱动器控制信号: SEG10带 COM3。
 SEG10[1]: LCD驱动器控制信号: SEG10带 COM2。
 SEG10[0]: LCD驱动器控制信号: SEG10带 COM1。

LCD11寄存器（地址=4AH）

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
LCD11	SEG11[7:4]				SEG11[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG11[7]: LCD驱动器控制信号: SEG11带 COM8。
 SEG11[6]: LCD驱动器控制信号: SEG11带 COM7。
 SEG11[5]: LCD驱动器控制信号: SEG11带 COM6。
 SEG11[4]: LCD驱动器控制信号: SEG11带 COM5。
 Bit 3-0 SEG11[3]: LCD驱动器控制信号: SEG11带 COM4。
 SEG11[2]: LCD驱动器控制信号: SEG11带 COM3。
 SEG11[1]: LCD驱动器控制信号: SEG11带 COM2。
 SEG11[0]: LCD驱动器控制信号: SEG11带 COM1。

LCD12寄存器（地址=4BH）

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
LCD12	SEG12[7:4]				SEG12[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG12[7]: LCD驱动器控制信号: SEG12带 COM8。
 SEG12[6]: LCD驱动器控制信号: SEG12带 COM7。
 SEG12[5]: LCD驱动器控制信号: SEG12带 COM6。
 SEG12[4]: LCD驱动器控制信号: SEG12带 COM5。
 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。

LCD13寄存器（地址=4CH）

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
LCD13	SEG13[7:4]				SEG13[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG13[7]: LCD驱动器控制信号: SEG13带 COM8。
 SEG13[6]: LCD驱动器控制信号: SEG13带 COM7。
 SEG13[5]: LCD驱动器控制信号: SEG13带 COM6。
 SEG13[4]: LCD驱动器控制信号: SEG13带 COM5。
 Bit 3-0 SEG13[3]: LCD驱动器控制信号: SEG13带 COM4。
 SEG13[2]: LCD驱动器控制信号: SEG13带 COM3。
 SEG13[1]: LCD驱动器控制信号: SEG13带 COM2。
 SEG13[0]: LCD驱动器控制信号: SEG13带 COM1。

LCD14寄存器（地址=4DH）

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
LCD14	SEG14[7:4]				SEG14[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG14[7]: LCD驱动器控制信号: SEG14带 COM8。
 SEG14[6]: LCD驱动器控制信号: SEG14带 COM7。
 SEG14[5]: LCD驱动器控制信号: SEG14带 COM6。
 SEG14[4]: LCD驱动器控制信号: SEG14带 COM5。
 Bit 3-0 SEG14[3]: LCD驱动器控制信号: SEG14带 COM4。
 SEG14[2]: LCD驱动器控制信号: SEG14带 COM3。
 SEG14[1]: LCD驱动器控制信号: SEG14带 COM2。
 SEG14[0]: LCD驱动器控制信号: SEG14带 COM1。

LCD15寄存器（地址=4EH）

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
LCD15	SEG15[7:4]				SEG15[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG15[7]: LCD驱动器控制信号: SEG15带 COM8。
 SEG15[6]: LCD驱动器控制信号: SEG15带 COM7。
 SEG15[5]: LCD驱动器控制信号: SEG15带 COM6。
 SEG15[4]: LCD驱动器控制信号: SEG15带 COM5。
 Bit 3-0 SEG15[3]: LCD驱动器控制信号: SEG15带 COM4。
 SEG15[2]: LCD驱动器控制信号: SEG15带 COM3。
 SEG15[1]: LCD驱动器控制信号: SEG15带 COM2。
 SEG15[0]: LCD驱动器控制信号: SEG15带 COM1。

LCD16寄存器（地址=4FH）

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
LCD16	SEG16[7:4]				SEG16[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG16[7]: LCD驱动器控制信号: SEG16带 COM8。
 SEG16[6]: LCD驱动器控制信号: SEG16带 COM7。
 SEG16[5]: LCD驱动器控制信号: SEG16带 COM6。
 SEG16[4]: LCD驱动器控制信号: SEG16带 COM5。
 Bit 3-0 SEG16[3]: LCD驱动器控制信号: SEG16带 COM4。
 SEG16[2]: LCD驱动器控制信号: SEG16带 COM3。
 SEG16[1]: LCD驱动器控制信号: SEG16带 COM2。
 SEG16[0]: LCD驱动器控制信号: SEG16带 COM1。

LCD17寄存器（地址=50H）

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
LCD17	SEG17[7:4]				SEG17[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG17[7]: LCD驱动器控制信号: SEG17带 COM8。
 SEG17[6]: LCD驱动器控制信号: SEG17带 COM7。
 SEG17[5]: LCD驱动器控制信号: SEG17带 COM6。
 SEG17[4]: LCD驱动器控制信号: SEG17带 COM5。
 Bit 3-0 SEG17[3]: LCD驱动器控制信号: SEG17带 COM4。
 SEG17[2]: LCD驱动器控制信号: SEG17带 COM3。
 SEG17[1]: LCD驱动器控制信号: SEG17带 COM2。
 SEG17[0]: LCD驱动器控制信号: SEG17带 COM1。

LCD18寄存器（地址=51H）

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
LCD18	SEG18[7:4]				SEG18[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG18[7]: LCD驱动器控制信号: SEG18带 COM8。
 SEG18[6]: LCD驱动器控制信号: SEG18带 COM7。
 SEG18[5]: LCD驱动器控制信号: SEG18带 COM6。
 SEG18[4]: LCD驱动器控制信号: SEG18带 COM5。
 Bit 3-0 SEG18[3]: LCD驱动器控制信号: SEG18带 COM4。
 SEG18[2]: LCD驱动器控制信号: SEG18带 COM3。
 SEG18[1]: LCD驱动器控制信号: SEG18带 COM2。
 SEG18[0]: LCD驱动器控制信号: SEG18带 COM1。

LCD19寄存器（地址=52H）

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
LCD19	SEG19[7:4]				SEG19[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG19[7]: LCD驱动器控制信号: SEG19带 COM8。
 SEG19[6]: LCD驱动器控制信号: SEG19带 COM7。
 SEG19[5]: LCD驱动器控制信号: SEG19带 COM6。
 SEG19[4]: LCD驱动器控制信号: SEG19带 COM5。
 Bit 3-0 SEG19[3]: LCD驱动器控制信号: SEG19带 COM4。
 SEG19[2]: LCD驱动器控制信号: SEG19带 COM3。
 SEG19[1]: LCD驱动器控制信号: SEG19带 COM2。
 SEG19[0]: LCD驱动器控制信号: SEG19带 COM1。

LCD20寄存器（地址=53H）

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
LCD20	SEG20[7:4]				SEG20[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG20[7]: LCD驱动器控制信号: SEG20带 COM8。
 SEG20[6]: LCD驱动器控制信号: SEG20带 COM7。
 SEG20[5]: LCD驱动器控制信号: SEG20带 COM6。
 SEG20[4]: LCD驱动器控制信号: SEG20带 COM5。
 Bit 3-0 SEG20[3]: LCD驱动器控制信号: SEG20带 COM4。
 SEG20[2]: LCD驱动器控制信号: SEG20带 COM3。
 SEG20[1]: LCD驱动器控制信号: SEG20带 COM2。
 SEG20[0]: LCD驱动器控制信号: SEG20带 COM1。

LCD21寄存器（地址=54H）

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
LCD21	SEG21[7:4]				SEG21[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG21[7]: LCD驱动器控制信号: SEG21带 COM8。
 SEG21[6]: LCD驱动器控制信号: SEG21带 COM7。
 SEG21[5]: LCD驱动器控制信号: SEG21带 COM6。
 SEG21[4]: LCD驱动器控制信号: SEG21带 COM5。
 Bit 3-0 SEG21[3]: LCD驱动器控制信号: SEG21带 COM4。
 SEG21[2]: LCD驱动器控制信号: SEG21带 COM3。
 SEG21[1]: LCD驱动器控制信号: SEG21带 COM2。
 SEG21[0]: LCD驱动器控制信号: SEG21带 COM1。

LCD22寄存器（地址=55H）

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
LCD22	SEG22[7:4]				SEG22[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-4 SEG22[7]: LCD驱动器控制信号: SEG22带 COM8。
 SEG22[6]: LCD驱动器控制信号: SEG22带 COM7。
 SEG22[5]: LCD驱动器控制信号: SEG22带 COM6。
 SEG22[4]: LCD驱动器控制信号: SEG22带 COM5。
 Bit 3-0 SEG22[3]: LCD驱动器控制信号: SEG22带 COM4。
 SEG22[2]: LCD驱动器控制信号: SEG22带 COM3。
 SEG22[1]: LCD驱动器控制信号: SEG22带 COM2。
 SEG22[0]: LCD驱动器控制信号: SEG22带 COM1。

LCD23寄存器（地址=56H）

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
LCD23					SEG23[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3-0 SEG23[3]: LCD驱动器控制信号: SEG23带 COM4。
 SEG23[2]: LCD驱动器控制信号: SEG23带 COM3。
 SEG23[1]: LCD驱动器控制信号: SEG23带 COM2。
 SEG23[0]: LCD驱动器控制信号: SEG23带 COM1。

LCD24寄存器（地址=57H）

特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
LCD6					SEG24[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3-0 SEG24[3]: LCD驱动器控制信号: SEG24带 COM4。
 SEG24[2]: LCD驱动器控制信号: SEG24带 COM3。
 SEG24[1]: LCD驱动器控制信号: SEG24带 COM2。
 SEG24[0]: LCD驱动器控制信号: SEG24带 COM1。

LCDENR寄存器（地址=58H）

特性	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-1	R/W-1	R/W-0
LCDENR	LCDCKS[1:0]		LCDEN		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 1/64
- 10 = LCD的帧频是 LCD输入时钟频率的 1/32
- 01 = LCD的帧频是 LCD输入时钟频率的 1/16
- 00 = LCD的帧频是 LCD输入时钟频率的 1/8

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 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 = LCD驱动器控制模式是 1/8duty周期模式

Bit 0 ENPMPL: LCD驱动器电荷泵使能标志

- 1 = LCD驱动器的电荷泵使能
- 0 = LCD驱动器的电荷泵不使能

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

表 3-13 LCD驱动器的寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
14H	MCK	--	--	M5_CK		M3_CK	--	M1_CK	M0_CK	00000000
54H	LCDENR	LCDCKS[1:0]		LCDEN		LEVEL	LCD_DUTY[1:0]		ENPMPL	000u0110

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. 设置 ENPMPL使能 LCD电荷泵。
4. 设置 M0_CK、M1_CK、M3_CK及 M5_CK, 选择 LCD输入时钟的频率。

表 3-14 LCDCLK选择列表

OSC_H	M4_CK	M5_CK	M3_CK	M1_CK	M0_CK	时钟源(KHz)		TMCLK(Hz)	
0	X	0	X	0	0	ICK	1000	ICK/1024	976
0	X	0	X	1	0	ICK	1000	ICK/4096	244
1	X	0	X	0	0	ICK	3250	ICK/2048	1586
1	X	0	X	1	0	ICK	3250	ICK/4096	793
X	0	0	0	0	1	ECK	32	ECK/2048	16
X	0	0	0	1	1	ECK	32	ECK/8192	4
X	0	0	1	0	1	ECK	32	ECK/4096	8
X	0	0	1	1	1	ECK	32	ECK/16384	2
X	1	0	0	0	1	ECK	4000	ECK/2048	2000
X	1	0	0	1	1	ECK	4000	ECK/8192	500
X	1	0	1	0	1	ECK	4000	ECK/4096	1000
X	1	0	1	1	1	ECK	4000	ECK/16384	250
X	0	1	X	X	X	ECK	32	ECK/32	1024
X	1	1	X	X	X	ECK	4000	ECK/32	125000

5. 设置寄存器标志 LCDCKL[1:0]选择 LCD的时钟频率

表 3-15 LCD帧频率的选择列表(时钟来源于 TMCLK)

LCDCKS[1:0]	LCD帧频率 (LCDCK)
00	LCD输入时钟频率/8
01	LCD输入时钟频率/16
10	LCD输入时钟频率/32
11	LCD输入时钟频率/64

6. 设置寄存器标志 LCD_DUTY[1:0], 选择控制模式。(SEG duty周期)

表 3-16 LCD的 duty控制模式选择列表

LCD_DUTY[1:0]	控制模式
00	1/8
01	1/2
10	1/3
11	1/4

7. 置位 LCDEN以使能 LCD驱动器

3.6 PDM模块

端口 PT2[3:2]既可以定义为 PDM输出，又可以定义为通用 IO端口。通过 PM1EN, PM2EN信号来控制该端口的定义。用户通过调整 PMCS来调整 PDM的脉宽。PDM的时钟被分为 16条内部的时钟信号 PDM15, PDM14, ..., PDM0。用户通过控制 PMD1和 PMD2寄存器来组合这 16个内部信号。

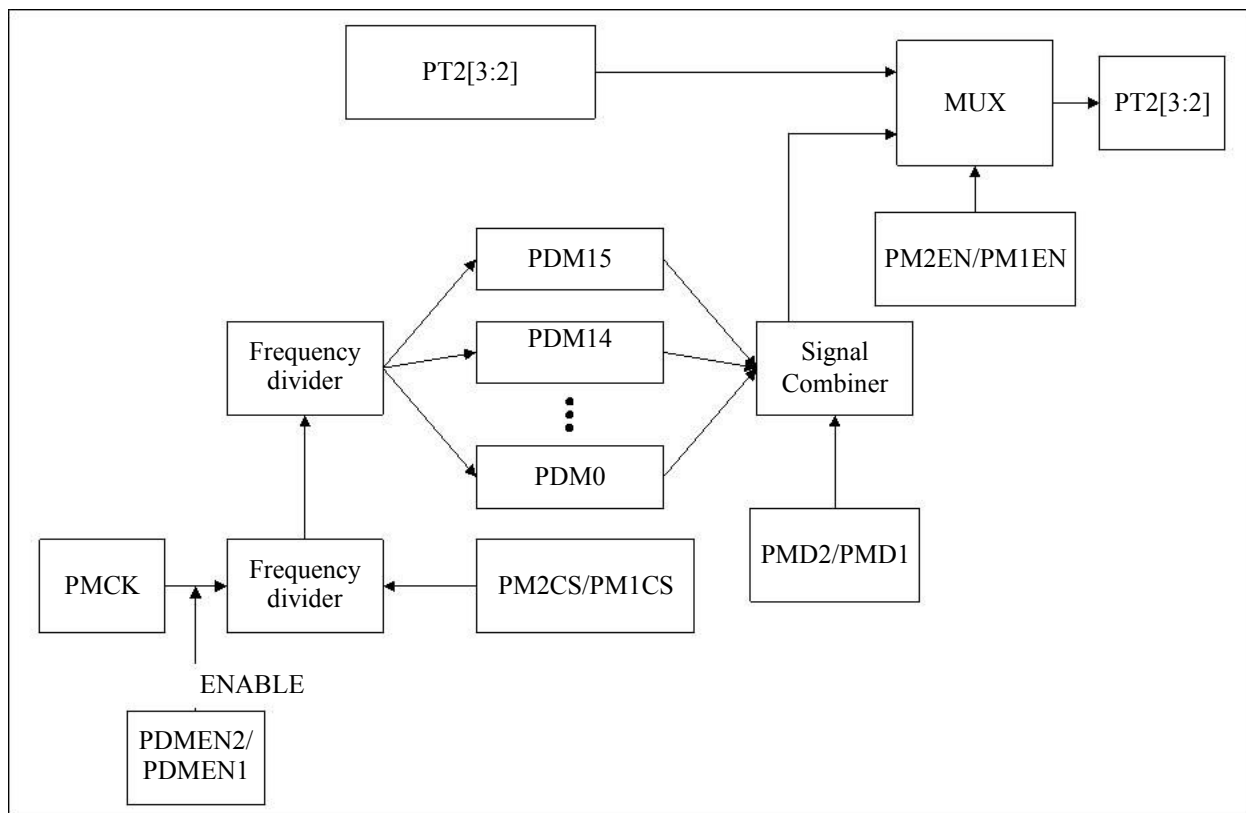


图 3-14 PDM电路结构

这 16条信号的时序关系如下所示。需要注意的一点是这 16条信号，任意两条信号不会同时出现高电平。

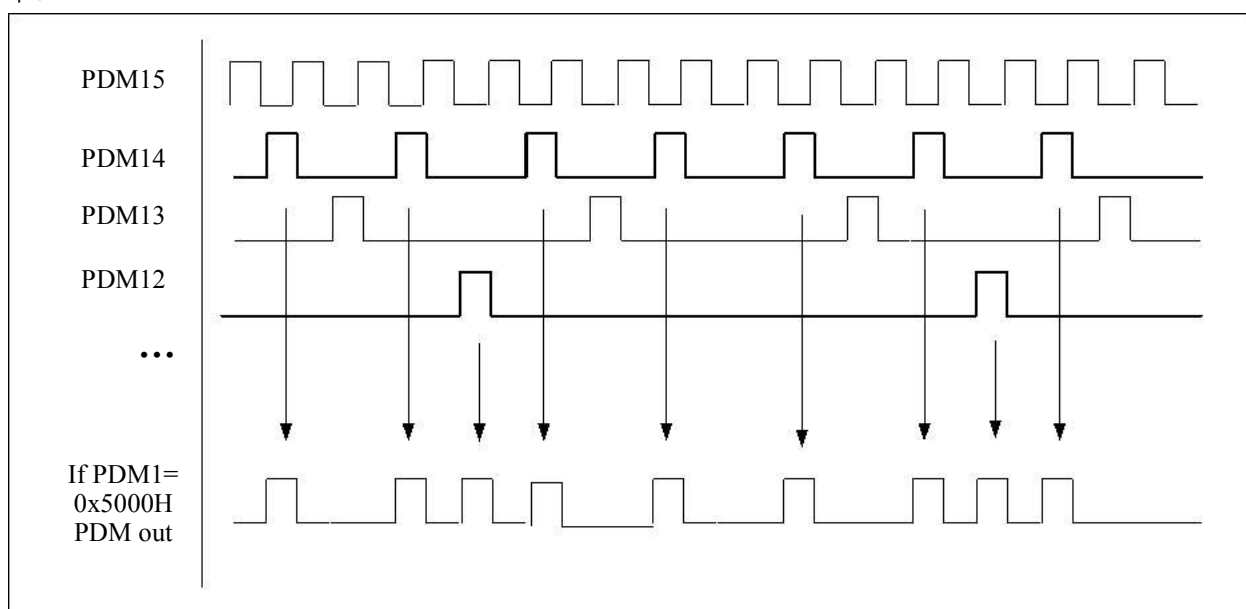


图 3-15 PDM时序图

表 3-17 PDM模块寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
27H	PT2MR	--		PM2EN	PM1EN	--		--		0u000000
30H	PMD1H	PMD1[15:8]								00000000
31H	PMD1L	PMD1[7:0]								00000000
32H	PMD2H	PMD2[15:8]								00000000
33H	PMD2L	PMD2[7:0]								00000000
36H	PMCON	PDM2EN	PM2CS[2:0]			PDM1EN	PM1CS[2:0]			00000000

PT2MR寄存器（地址=27H）

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	R/W-0
PT2MR			PM2EN	PM1EN					
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 5 PM2EN: PT2[3]输出选择器
 1 = GPIO2口 bit 3 (PT2[3]) 定义为 PDM输出
 0 = GPIO2口 bit 3 (PT2[3]) 定义为普通用途 I/O口

Bit 4 PM1EN: PT2[2]输出选择器
 1 = GPIO2口 bit 2 (PT2[2]) 定义为 PDM输出
 0 = GPIO2口 bit 2 (PT2[2]) 定义为普通用途 I/O口

PMD1H寄存器（地址=30H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD1H	PMD1[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

PMD1L寄存器（地址=31H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD1L	PMD1[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 15-0PMD1[15:0]: PDM模块数据输出控制寄存器

PMD1[15] = PDM15 (PDMCLK/2₀) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[14] = PDM14 (PDMCLK/2₁) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[13] = PDM13 (PDMCLK/2₂) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[12] = PDM12 (PDMCLK/2₃) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[11] = PDM11 (PDMCLK/2₄) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[10] = PDM10 (PDMCLK/2₅) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[9] = PDM9 (PDMCLK/2₆) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[8] = PDM8 (PDMCLK/2₇) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[7] = PDM7 (PDMCLK/2₈) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[6] = PDM6 (PDMCLK/2₉) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[5] = PDM5 (PDMCLK/2₁₀) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[4] = PDM4 (PDMCLK/2₁₁) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[3] = PDM3 (PDMCLK/2₁₂) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[2] = PDM2 (PDMCLK/2₁₃) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[1] = PDM1 (PDMCLK/2₁₄) 信号组合使能标志位。1 =使能; 0 =不使能
 PMD1[0] = PDM0 (PDMCLK/2₁₅) 信号组合使能标志位。1 =使能; 0 =不使能

PMD2H寄存器（地址=32H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD2H	PMD2[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

PMD2L寄存器（地址=33H）

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD2L	PMD2[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 15-0PMD2[15:0]: PDM2模块数据输出控制寄存器

PMD2[15] = PDM15 (PDMCLK/2₀) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[14] = PDM14 (PDMCLK/2₁) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[13] = PDM13 (PDMCLK/2₂) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[12] = PDM12 (PDMCLK/2₃) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[11] = PDM11 (PDMCLK/2₄) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[10] = PDM10 (PDMCLK/2₅) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[9] = PDM9 (PDMCLK/2₆) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[8] = PDM8 (PDMCLK/2₇) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[7] = PDM7 (PDMCLK/2₈) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[6] = PDM6 (PDMCLK/2₉) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[5] = PDM5 (PDMCLK/2₁₀) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[4] = PDM4 (PDMCLK/2₁₁) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[3] = PDM3 (PDMCLK/2₁₂) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[2] = PDM1 (PDMCLK/2₁₃) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[1] = PDM1 (PDMCLK/2₁₄) 信号组合使能标志位。1 = 使能; 0 = 不使能
 PMD2[0] = PDM0 (PDMCLK/2₁₅) 信号组合使能标志位。1 = 使能; 0 = 不使能

PMCON寄存器（地址=36H）

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
PMCON	PDM1EN	PM2CS[2:0]			PDM1EN	PM1CS[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 PDM2EN: PDM2模块使能标志

1 = PDM2模块使能, GPIO2口 bit 3可以被定义为 PDM输出接口
 0 = PDM2模块不使能, GPIO2口 bit 3可以被定义为普通用途 I/O接口

Bit 6-4 PM2CS[2:0]: PDMCLK频率选择器

111 = PDM2CLK频率为 MCK/128
 110 = PDM2CLK频率为 MCK/64
 101 = PDM2CLK频率为 MCK/32
 100 = PDM2CLK频率为 MCK/16
 011 = PDM2CLK频率为 MCK/8
 010 = PDM2CLK频率为 MCK/4
 001 = PDM2CLK频率为 MCK/2
 000 = PDM2CLK频率为 MCK

Bit 3 PDM1EN: PDM1模块使能标志

1 = PDM1模块使能, GPIO2口 bit 2可以被定义为 PDM输出接口
 0 = PDM1模块不使能, GPIO2口 bit 2可以被定义为普通用途 I/O接口

Bit 2-0 PM1CS[2:0]: PDMCLK频率选择器

111 = PDM1CLK 频率为 MCK/128
 110 = PDM1CLK 频率为 MCK/64
 101 = PDM1CLK 频率为 MCK/32
 100 = PDM1CLK 频率为 MCK/16
 011 = PDM1CLK 频率为 MCK/8
 010 = PDM1CLK 频率为 MCK/4
 001 = PDM1CLK 频率为 MCK/2
 000 = PDM1CLK 频率为 MCK

表 3-18 PDM 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
14H	MCK	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
25H	PT2EN	PT2EN[7:0]								00000000
27H	PT2MR	--			PM1EN	--		--		00000000
30H	PMD1H	PMD1[15:8]								00000000
31H	PMD1L	PMD1[7:0]								00000000
36H	PMCON			PDMEN			PMCS[2:0]			00000000

PDM 操作：

1. 设置 M0_CK, M3_CK 决定 MCK。
2. 置位 PDMEN 以使能 PDM 模块。
3. 设置 PMCS[2:0] 决定 PDMCLK 的频率。
4. 设置 PMD1[15:0] 决定 PDM 的输出信号。
5. 置位 PT2EN[2] 以使 PT2[2] 作为输出接口。
6. 置位 PM1EN 以使 PT2[2] 作为 PDM 模块的输出接口。

表 3-19 PDMCLK 选择表

PWCS	PDMCLK 频率
000	MCK
001	MCK/2
010	MCK/4
011	MCK/8
100	MCK/16
101	MCK/32
110	MCK/64
111	MCK/128

4 CPU指令集

表 4-1 CPU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	Z
BSF f,b	$[f < b>] \leftarrow 1$	1	Z
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	C,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	C,Z
INCF f,d	$[Destination] \leftarrow [f] + 1$	1	~
INCFSZ f,d	$[Destination] \leftarrow [f] + 1$, jump if the result is zero	1	Z
IORLW k	$[W] \leftarrow [W] \text{ OR } k$	1	Z
IORWF f,d	$[Destination] \leftarrow [W] \text{ OR } [f]$	1	~
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	C,Z
RLF f,d	$[Destination < n+1>] \leftarrow [f < n>]$	1	C,Z
RRF f,d	$[Destination < n-1>] \leftarrow [f < n>]$	1	PD
SLEEP	STOP OSC	1	
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
XORLW k	$[W] \leftarrow [W] \text{ XOR } k$	1	Z
XORWF f,d	$[Destination] \leftarrow [W] \text{ XOR } [f]$		Z

参数说明:

f:数据存储器地址(00h~17Fh)

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 CPU指令集描述

1

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

2

ADDPCW	将 W的内容加到 PC中
指令格式	ADDPCW
操作	(PC)<—(PC)+1+(W)
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

3

ADDWF	加工作寄存器到 f
指令格式	ADDWF f,d 0<=f<=17Fh d=0,1
操作	[目标地址]<—(f)+(W)
标志位描述	C, CD, Z 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 f0	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 f1	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<=17Fh 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<=17Fh 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<=17Fh 0<=b<=7
操作	(f[b])<—0
标志位	Z
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<=17Fh 0<=b<=7
操作	(f[b])<-1
标志位 描述	Z 将 f的 b位置
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<=17Fh 0<=b<=7
操作	Skip if (f[b])=0
标志位 描述	无 如果 f的 bit位是 0, 则 PC值加 2即跳过 BTFSC指令的下一条指令
周期	1
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<=17Fh 0<=b<=7
操作	Skip if (f[b])=1
标志位 描述	无 如果 f的 bit位是 1, 则 PC值加 2即跳过 BTFSS指令的下一条指令
周期	1
例子 NODE BTFSS FLAG 2 OP1: OP2:	在程序执行以前 PC=address(NODE) 指令执行之后 If(FLAG[2])=0 PC=address(OP1) If(FLAG[2])=1 PC=address(OP2)

11

CALL	子程序调用
指令格式	CALL K 0<=K<=FFFh
操作	(top stack)<-PC+1 PC<-K
标志位	无
描述	子程序调用, 先将 PC+1压入堆栈, 然后把立即数地址下载到 PC中。
周期	1

--	--

12

CLRF	清除 f
指令格式	CLRF f 0<=f<=17Fh
操作	(f)<—0
标志位	无
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

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<=17Fh 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<=17Fh d=0,1
操作	(目的地址)<—(f)-1
标志位	C,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<=17Fh d=0,1
操作	(目的地址)<—(f)-1,如果结果为 0跳转
标志位	--
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, 则 PC值加 2即跳过 DECFSZ指令的下一条指令
周期 例子 NodeDECFSZ FLAG, 1 OP1: OP2:	1 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<=FFFh
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<=17Fh d=0,1
操作	(目的地址)<—(f)+1
标志位	C,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<=17Fh d=0,1
操作	(目的地址)<-(f)+1 如果结果为 0 就跳转
标志位	--
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，则 PC 值加 2 即跳过 INCFSZ 指令的下一条指令
周期 例子 Node INCFSZ FLAG, 1 OP1: OP2:	1 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)

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<=17Fh 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<=17Fh
操作	(W)<—(f)
标志位 描述	无 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 描述	无 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

MOVWF	将工作寄存器的值传送到 f中
指令格式	MOVWF f 0<=f<=17Fh
操作	(f)<—(W)
标志位 描述	无 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

26

NOP	无操作
指令格式	NOP
操作	无操作
标志位 描述	无 无操作
周期	1

27

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

28

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

29

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

30

RLF	带进位左移
指令格式	RLF f, d 0<=f<=17Fh d=0,1
操作	(目标地址[n+1])<—(f[n]) (目标地址[0])<—C C<—(f[7])
标志位	C, Z
描述	F带进位位左移一位 如果 d为 0，结果保存到工作寄存器 如果 d为 1，结果保存到 f中
周期	1
例子 RLF f, 1	在指令执行之前 C=0 W=88h f=E6h 在指令执行之后 C=1 W=88h f=CCh

31

RRF	带进位右移
指令格式	RRF f, d 0<=f<=17Fh 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,Z
描述	F带进位位右移一位

	如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 RRF f, 0	在指令执行之前 C=0 W=88h f=95h 在指令执行之后 C=1 W=4Ah f=95h

32

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

33

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 描述	C, DC, Z 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)

34

SUBWF	f的值减工作寄存器的值
指令格式	SUBWF f, d 0<=f<=17Fh d=0,1
操作	(目标地址)<←(f)-(W)
标志位 描述	C, DC, Z 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

35

SUBWFC	带借位的减法
指令格式	SUBWFC f, d 0<=f<=17Fh 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

36

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

37

XORWF	f的值与工作寄存器的值异或
指令格式	XORWF f, d 0<=f<=17Fh 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	在指令执行之前

XORWF f, 1	W=AC _h f=5F _h 在指令执行之后 f=F3 _h
------------	---

5 电气特性

5.1 CSU1281 极限值

表 5-1 CSU1281 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~VDD+0.3	V
工作温度	-40~+85	°C
存贮温度	-55~+150	°C
焊接温度, 时间	220°C, 10秒	

5.2 直流特性 (DVDD, AVDD = 3.3V, T_A = 25°C, 如无其他说明则都是此条件)

表 5-2 CSU1281 直流特性

符号	参数	测试条件	最小值	典型值	最大值	单位
DVDD	工作电源		2.4	3.3	3.6	V
AVDD	工作电源		2.4	3.3	3.6	V
IDD1	电源电流 1	MCK = 1MHz CPUCLK=MCK/2 电荷泵、ADC打 开		3		mA
IDD2	电源电流 2	内部振荡器关闭 MCK = 32768Hz LCD开	8	10		uA
IPO	睡眠模式下电源电流	睡眠指令			3	uA
VIH	数字输入高电平	PT1,2,3	VDD-0.6			V
VIL	数字输入低电平	PT1,2,3			0.3	V
IPU	上拉电流	PT1,2,3 Vin = 0		30		uA
IOH	高电平输出电流	VOH=VDD-0.3V		3		mA
IOL	低电平输出电流	VOL=0.3V		3		mA
VDDA	模拟电源			3		V
I _{REG}	VDDA稳压器输出电流	VDD=3.3V 内部电压倍压 VDDA=2.85V (没 有负载)		0.8		mA
VCVDDA	VDDA电压变化系数		-2		2	%V
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 _A = -40~80°C		50		ppm/°C
VLBAT	低电池检测电压	SILB[1:0]=00		2.4		V
		SILB[1:0]=01		3.6		
FRC	内置 RC振荡器	OSC_H=0	0.8	1.0	1.2	MHz
		OSC_H=1	2.6	3.25	3.9	
FWDT	内置看门狗时钟		1.5	3	4.5	KHz

5.3 ADC特性 (DVDD, AVDD= 3.3V, T_A= 25°C, 如无其他说明则都是此条件)

表 5-3 CSU1281 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	Buffer关闭	AGND-0.1		AVDD+0.1	V
		Buffer打开	AGND+0.4		AVDD-1.5	V
	满幅输入电压 (AIN+) - (AIN-)				±VREF/PGA	V
	差分输入阻抗	Buffer关闭		16/PGA		MΩ
		Buffer打开		5		GΩ
系统性能	分辨率	无失码		24		Bits
	输入噪声 (rms)	增益=1		2.8		uv
		增益=128		120		nV
	积分线性度	增益=128		±0.004		% of FS
	失调误差	增益=128		14		uV
	失调误差漂移	增益=128		0.05		uV/°C
	增益误差	增益=128		0.1		%
参考电压	VS	LDOS[1:0]= 11	2	2.2	2.3	V
		LDOS[1:0]= 10	2.3	2.5	2.6	
		LDOS[1:0]= 01	2.5	2.8	2.8	
		LDOS[1:0]= 00	2.6	3	3	
	参考电压温度系数			100		ppm/°C

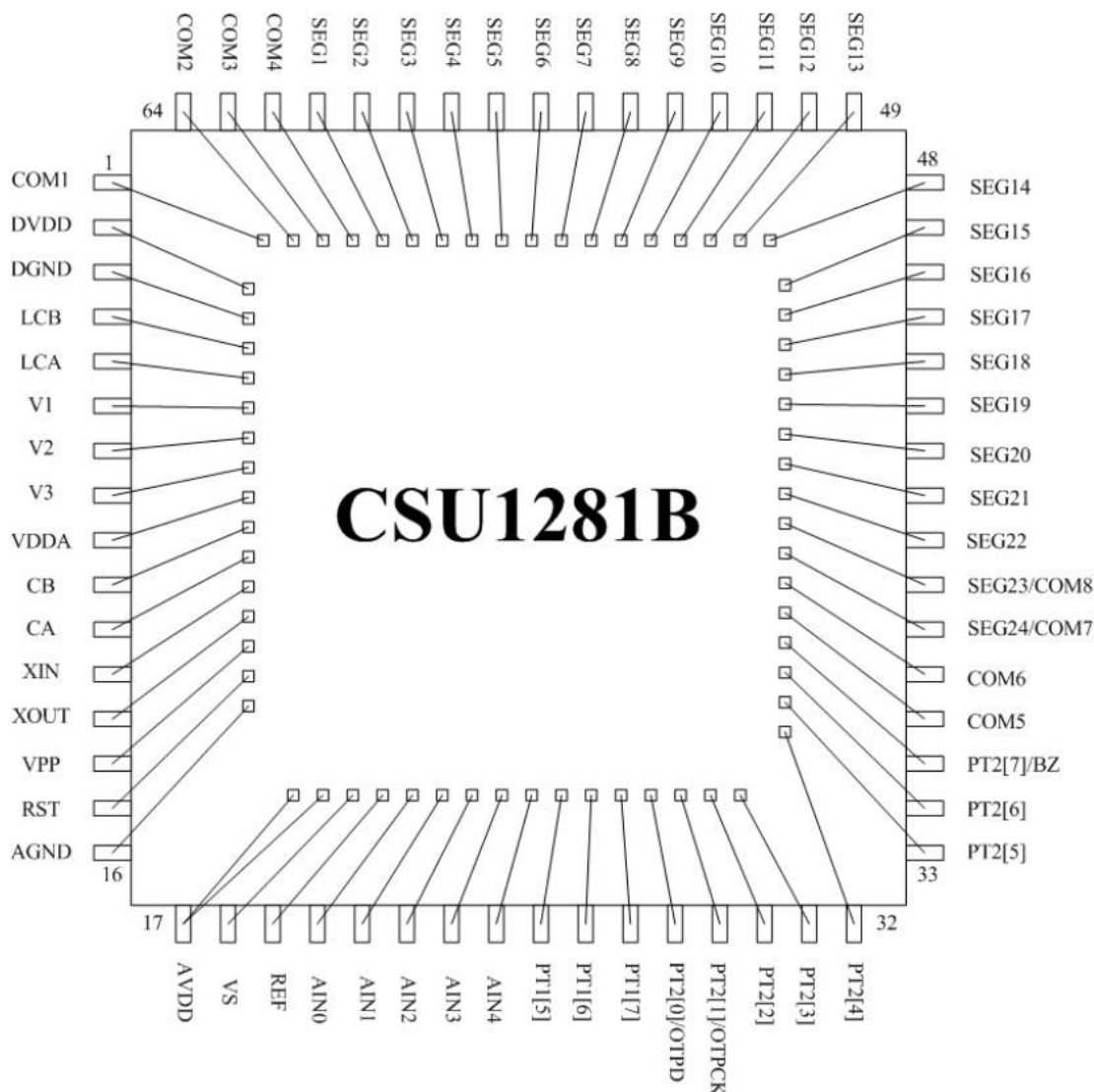
5.4 复位特性 (Brownout及外部复位引脚)

CSU1281芯片的复位电压点为 $2.2V \pm 0.1V$, 当电源电压低于复位电压点时, 芯片处于复位状态。上电复位过程中, 在电源电压未达到复位电压点之前, 内部复位电路会通过 RST pin以 1000欧姆的内阻对外部复位电路中的电容进行放电。电源电压上升至高于复位电压点之后, 内部电路停止对外部复位电容的放电, 外部上电复位过程开始, 此过程的长短, 取决于外部复位电阻和电容的大小。计算公式为: $t=0.38 \cdot R \cdot C$ (例如: $R=1M, C=0.1\mu F$ 时, $T=38ms$)

当电源电压突变, 并低于复位电压点时, 上述上电复位过程会重新开始, 以避免系统电源不稳定的状态。

不允许把 RST PIN不通过电阻直接接到 DVDD

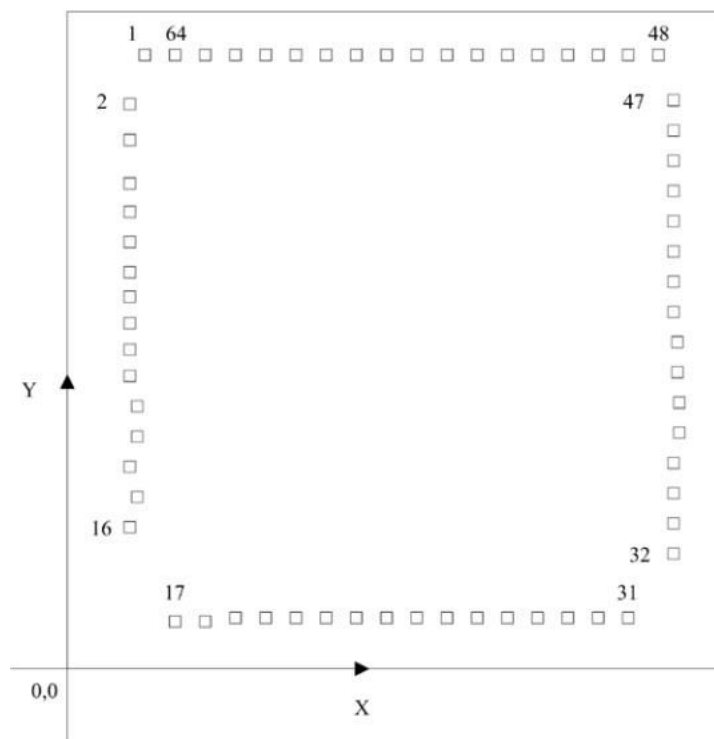
6 bonding说明



由图可知，左边有 15 个 pin，下边有 16 个 pin，右边有 16 个 pin，上边有 18 个 pin。然后在裸片上我们可以看到右下角有芯片的版本号（有 CSU1200 字样），右上角空白，左上角和左下角均有一些电路。通过以上信息可以确定绑定的管脚顺序。

注意：衬底必须接到 **DGND**

6.1 PAD排布



芯片尺寸: X=2.756mm, Y=2.769mm

6.2 PAD坐标

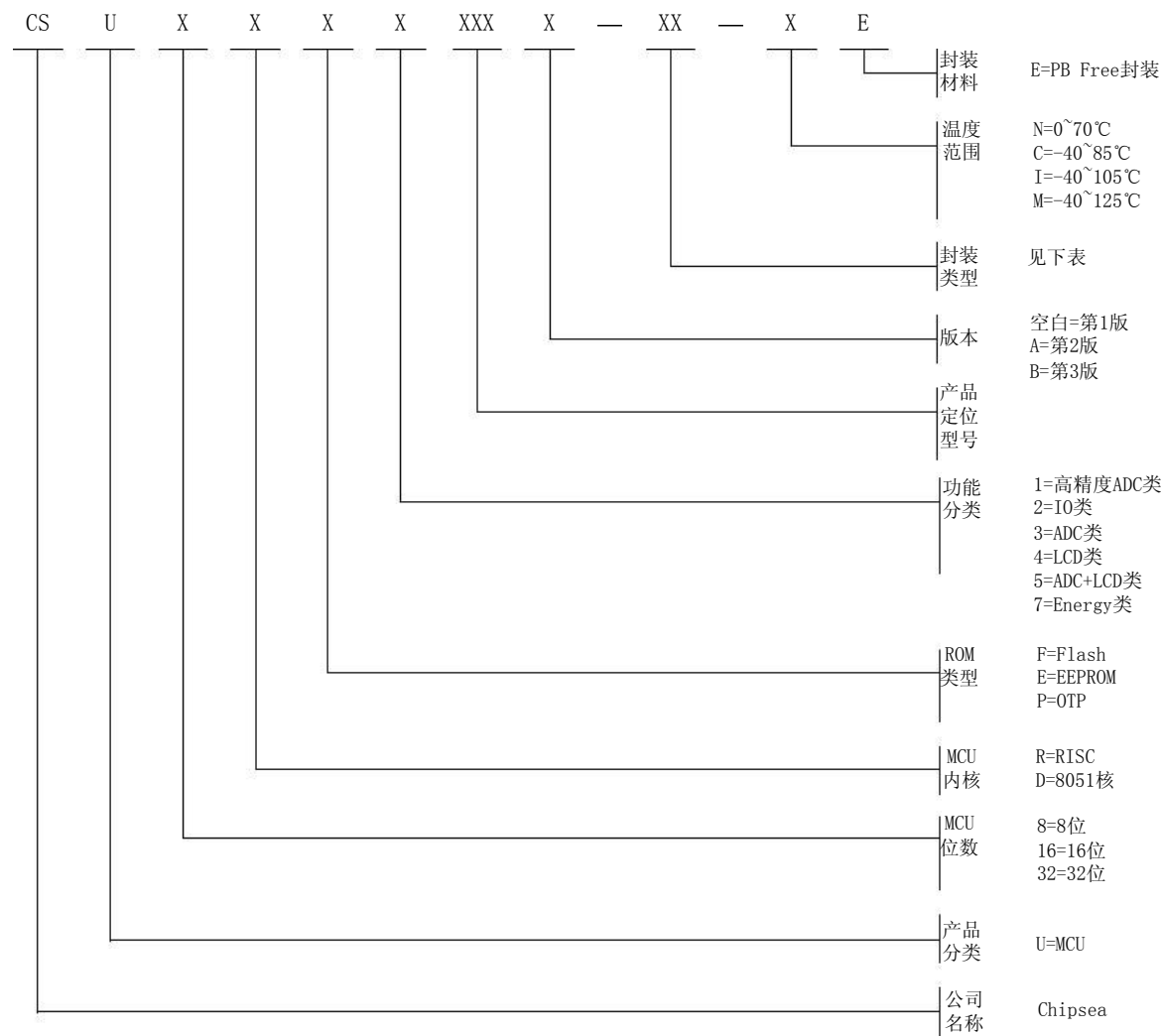
PIN脚 序号	符号	PAD中心坐标(um) (X, Y)
1	COM1	307, 2565
2	DVDD	45, 2233
3	DGND	45, 2095
4	LCB	45, 1948
5	LCA	45, 1828
6	V1	45, 1708
7	V2	45, 1588
8	V3	45, 1468
9	VDDA	45, 1348
10	CB	45, 1228
11	CA	45, 1108
12	XIN	92, 953
13	XOUT	92, 833
14	VPP	45, 668
15	RST	92, 530
16	AGND	45, 303
17	AVDD	330, 46
17	AVDD	467, 46
18	REF	609, 92
19	VS	729, 92
20	PT1_0	1003, 92
21	PT1_1	1124, 92
22	PT1_2	1243, 92
23	PT1_3	1363, 92
24	PT1_4	1483, 92
25	PT1_5	1603, 92
26	PT1_6	1724, 92
27	PT1_7	1843, 92
28	PT2_0	1963, 92
29	PT2_1	2083, 92
30	PT2_2	2204, 92
31	PT2_3	2323, 92
32	PT2_4	2518, 331
33	PT2_5	2518, 451
34	PT2_6	2518, 571
35	PT2_7	2518, 691
36	COM5	2565, 821
37	COM6	2565, 945
38	COM7	2565, 1067
39	COM8	2565, 1190
40	SEG22	2518, 1340
41	SEG21	2518, 1470
42	SEG20	2518, 1600

43	SEG19	2518, 1730
44	SEG18	2518, 1860
45	SEG17	2518, 1991
46	SEG16	2518, 2121
47	SEG15	2518, 2251
48	SEG14	2347, 2565
49	SEG13	2227, 2565
50	SEG12	2107, 2565
51	SEG11	1987, 2565
52	SEG10	1867, 2565
53	SEG9	1747, 2565
54	SEG8	1627, 2565
55	SEG7	1507, 2565
56	SEG6	1387, 2565
57	SEG5	1267, 2565
58	SEG4	1147, 2565
59	SEG3	1027, 2565
60	SEG2	907, 2565
61	SEG1	787, 2565
62	COM4	667, 2565
63	COM3	547, 2565
64	COM2	427, 2565

注意：PAD17和 PAD18都是 Bonding到 AVDD

7产品命名规则

产品型号说明



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