

SD3003/SD3004 应用手册

目录

1	概述	5
1.1	文档说明	5
1.2	名词定义	5
1.2.1	常用词汇索引	5
1.2.2	寄存器相关字	5
2	产品简介	6
2.1	产品描述及基本配置表	6
2.2	主要特性	6
2.3	功能框图	7
2.4	管脚配置	8
3	CPU	11
3.1	处理器内核, CPU CORE	11
3.2	存储器, MEMORY	11
3.2.1	概述	11
3.3	程序计数器, PC	14
3.4	堆栈, STK	14
3.5	数据存储空间	15
3.6	数据存储寄存器说明	16
3.7	寄存器列表	17
4	指令集	20
4.1	操作码字符介绍	20
4.2	指令集介绍	20
4.3	指令说明	23
5	时钟系统	36
5.1	概述	36
5.2	CPU及外围模块时钟源	36
5.2.1	CPU时钟源	36
5.3	时钟相关寄存器	38
6	复位	39
6.1	概述	39
6.1.1	外部复位	40
6.1.2	低电压检测复位 (LVD)	40
6.1.3	WDT看门狗计数器复位	41
6.1.4	STK堆栈错误复位	41
6.1.5	指令复位	41
6.2	状态寄存器	42

6.3 数据寄存器复位值	42
7 中断.....	44
7.1 概述	44
7.2 中断说明	45
7.3 寄存器说明	46
8 I/O口	47
8.1 概述	47
8.2 PORT相关寄存器说明	48
8.3 输入/输出口 1, PORT1	49
8.4 输入/输出口 2, PORT2.....	51
8.5 输入/输出口 3, PORT3	52
8.6 输入/输出口 4, PORT4	54
9 看门狗WDT (WATCH DOG).....	59
9.1 WDT概述	59
9.2 WDT使用说明	59
9.3 寄存器说明-WDT	59
10 计时/定时器.....	59
10.1 概述	59
10.2 定时器 0, TMR0.....	59
10.2.1 概述.....	59
10.2.2 定时器0 使用说明.....	60
10.2.3 寄存器说明-TMR0	61
10.3 定时器 1, TMR1.....	61
10.3.1 概述.....	61
10.3.2 TMR1 使用说明.....	62
10.3.3 寄存器说明-TMR1	62
10.3.4 捕捉功能说明.....	63
11 电量计量.....	64
11.1 概述	64
11.2 寄存器说明	64
11.3 电量计数器	67
12 USART通信.....	68
12.1 概述	68
12.2 波特率设置	68
12.3 异步模式	68
12.3.1 启动.....	68
12.3.2 波特率设置.....	69
12.3.3 发送.....	69
12.3.4 接收.....	70
12.4 同步主控模式	71

12.4.1 同步主机发送模式.....	71
12.4.2 同步主机接收模式.....	72
12.5 同步从动模式	73
12.5.1 同步从动发送.....	73
12.5.2 同步从动接收.....	73
12.6 寄存器说明	74
13 LCD DRIVER.....	75
13.1 LCD DRIVER概述	75
13.2 LCD DRIVER的时钟	76
13.3 LCD驱动模块设置.....	76
13.4 COM/SEG 设置.....	77
13.5 显示数据设置	77
13.6 LCD输出波形.....	78
14. RTC.....	80
14.1 概述	80
14.2 RTC使用	80
14.2.1 RTC初始设置.....	80
14.2.2 RTC中断事件服务.....	80
14.2.3 RTC使能.....	80
15. 串行通信-I2C主从机接口	81
15.1 I2C概述	81
15.2 I2C从动模式使用说明	82
15.2.1 启动.....	82
15.2.2 地址匹配.....	83
15.2.3 接收.....	83
15.2.4 发送.....	83
15.3 全局呼叫地址支持	84
15.4 主机模式	84
15.4.1 I2C主机模式工作方式.....	85
15.4.2 波特率发生器.....	85
15.4.3 发送和接收时序.....	86
15.4.3.1 主机发送流程.....	86
15.4.3.2 主机接收流程.....	86
15.4 多主机通信、总线冲突和总线仲裁.....	87
15.5 寄存器说明	87
16 程序脉冲发生器PPG.....	90
16.1 概述	90
16.2 寄存器说明	91
17 工作模式.....	93
17.1 工作模式概述	93

17.1.1	正常工作模式.....	93
17.1.2	待机模式.....	93
17.1.3	休眠模式.....	93
17.2	正常工作中模式转换	93
18	测试模式说明	94
18.1	OTP串行烧录模式.....	94
18.1.1	串行烧录管脚.....	94
18.1.2	串行烧录时序.....	94
18.1.3	串行读取时序.....	95
18.2	并行烧录	96
18.2.1	并行烧录管脚.....	96
18.2.2	并行烧录时序.....	97
18.2.3	并行读取时序.....	98
18.3	外部灌指令模式	99
18.3.1	外灌指令接口.....	100
18.4	查看测试信号模式	100
19.	典型应用	101
20	电气参数	102
21	封装规格	104

1 概述

1.1 文档说明

本文档为 SD3003/SD3004 的应用手册，包含功能说明，寄存器使用说明，存储器空间等。

1.2 名词定义

1.2.1 常用词汇索引

ADC	Analog to Digital Converter	模数转换器
Bit	Bit	位
BOR	Brown-Out Reset	低压复位
Byte	Byte	字节
CPU	Central Processing Unit	中央处理器
DM	Data Memory	数据存储
FSR	File Select Register	间接寻址寄存器
GPR	General Purpose Register	一般寄存器
MSB	Most Significant Bit	最高有效位
OTP	One Time Program-EEPROM	一次性写入存储器
PC	Program Counter	程序计数器
SR	Special Register	特殊寄存器
SRAM	Static Random Access Memory	静态随机访问存储器
STK	Stack	堆栈
WDT	Watch Dog Timer	看门狗计数器
WREG	Work Register	工作寄存器

1.2.2 寄存器相关字

R/W	Read/Write	可读可写
R	Readonly	只读
W	Writeonly	只写
--	Not use	未使用
u	Unchanged	不改变
x	Unknown	未知

2 产品简介

2.1 产品描述及基本配置表

本产品是 LCD/LED 显示的电能计量的 SOC 解决方案，可以极大的降低电表或者计量插座等产品的设计复杂度，降低成本。

表 2-1 SD3004/SD3003 资源对比

单片机 型号	OTP (Words)	RAM (Bytes)	Stack	LCD	Timer	PPG	烧录电 压 (V)	外部中 断	电能计 量	交流有 效值计 算	RTC
SD3004	2k	128	6 级	4*24 段	T0,T1	有	12.5V	2 个	有	有	有
SD3003	2k	128	6 级	4*15 段	T0,T1	有	12.5V	2 个	有	有	有

2.2 主要特性

- 高精度电能计量，计量精度满足一级电表使用要求
- 提供电压有效值和电流有效值
- 可以计算有功功率、功率因数
- 可以计算交流电频率
- 提供高频校验脉冲输出，用于校表
- 可以对电量进行累计计算
- 24×4 的 LCD 驱动电路，可切换为 I/O，支持 LED 驱动
- 带 RTC，可以提供秒信号输出
- 带 USART 和 I2C 接口
- 程序存储器为 2K*16 OTP，可以在线烧录；数据存储器 128Bytes
- 工作电压：计量部分 4.5V-5.5V，其他部分 2.4V-5.5V

2.3 功能框图

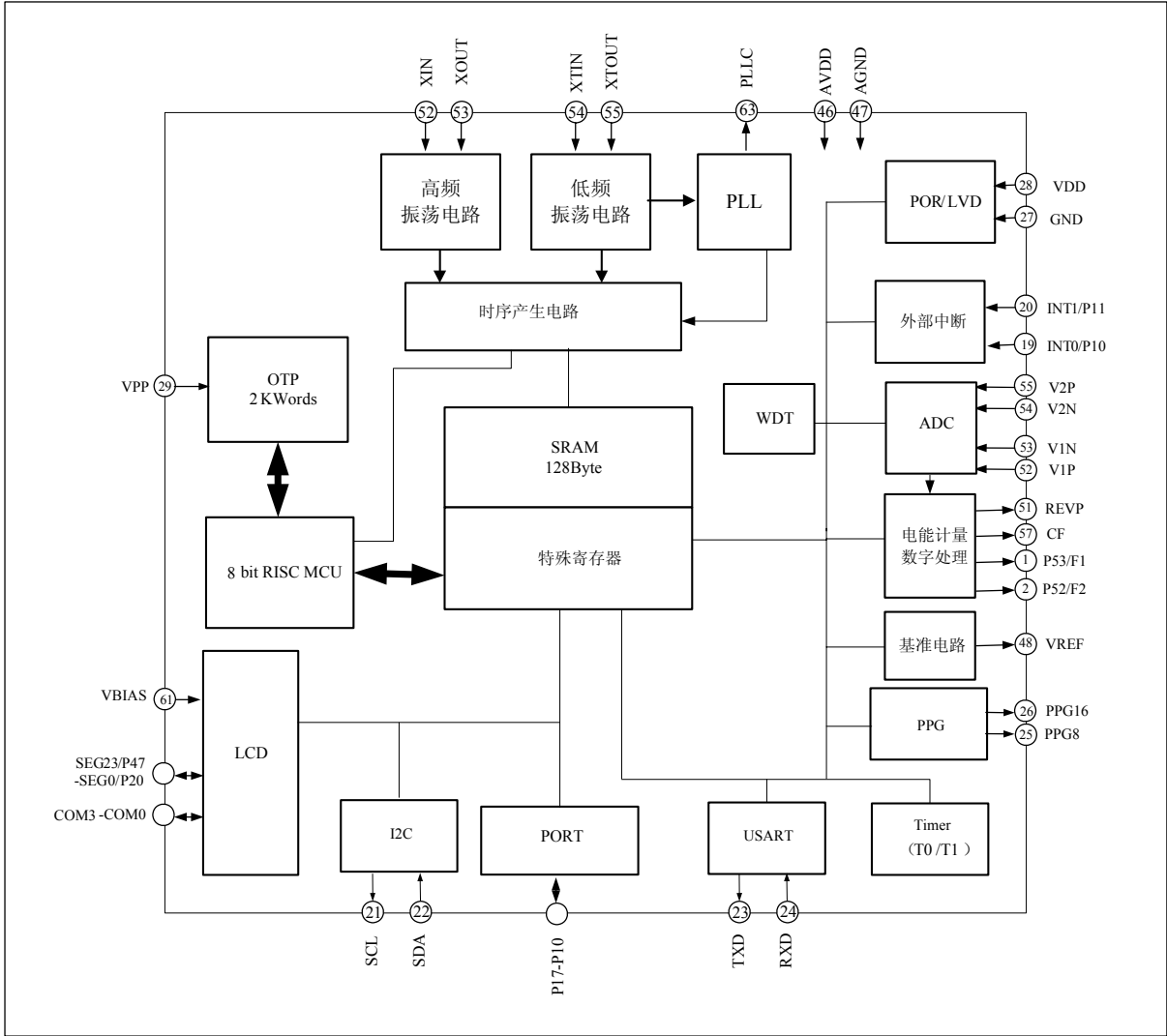


图 2-1. 功能框图



2.4 管脚配置

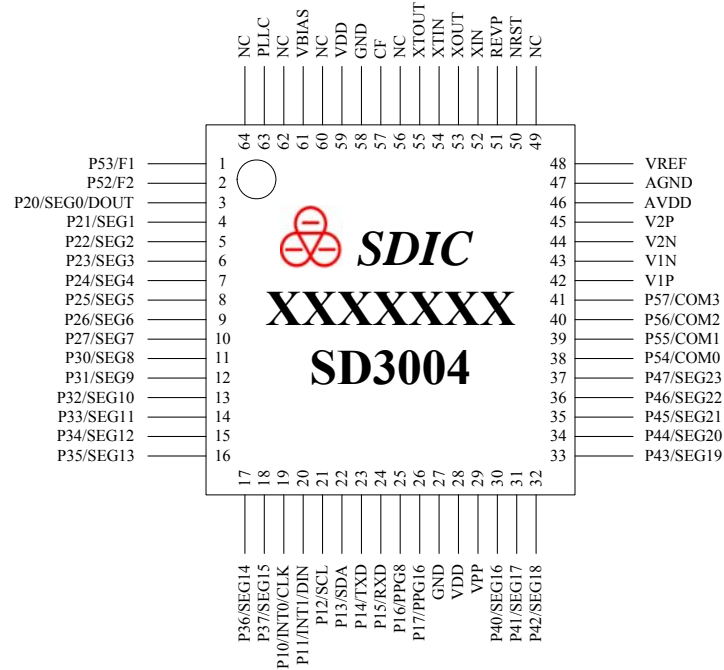


图 2-2. SD3004 管脚图

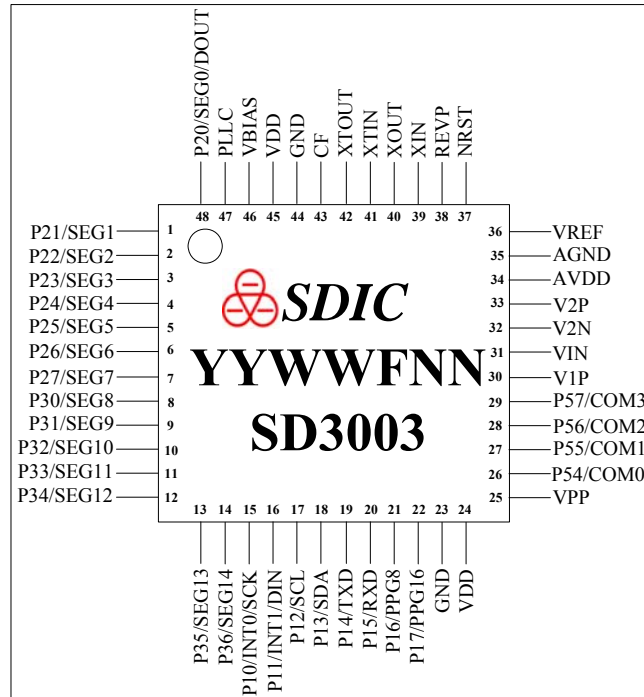


图 2-3. SD3003 管脚图

表 1. SD3004 管脚描述

序号	管脚名称	属性	管脚描述
1	P53/F1	I/O	驱动电机的输出, 跟 P53 复用
2	P52/F2	I/O	驱动电机的输出, 跟 P52 复用
3-18	P20/SEG0-P37/SEG15	I/O	LCD SEG15-SEG0。SEG7-0 跟 PORT27-PORT20 复用; SEG0 还跟 OTP 烧录的数据输出端复用; SEG15-8 跟 PORT37-PORT30 复用
19-26	P10/INT0/SCK-P17/PPG16	I/O	I/O P17-P10。P10 复用 INT0, 中断沿可选; 也跟 OTP 烧录的时钟输入脚复用; P11 复用 INT1, 中断沿可选; 也跟 OTP 烧录的数据输入端复用; P12 复用 I2C 时钟端 SCL; P13 复用 I2C 数据端 SDA; P14 复用 USART 的数据发送 TXD/CK; P15 复用 USART 的数据接收 RXD/DT; P16 复用 8 位的 PPG 输出; P17 复用 16 位的 PPG 输出
27	GND	地	数字地
28	VDD	电源	数字电源
29	VPP	I	OTP 编程电源脚
30~37	P40/SEG16-P47/SEG23	I/O	LCD SEG16-23, 跟 PORT40-PORT47 复用
38~41	COM0-COM3	模拟	LCD 的 COM 端, 跟 PORT54-P57 复用
42	V1P	模拟	电流通道正向输入
43	V1N	模拟	电流通道反向输入
44	V2N	模拟	电压通道反向输入
45	V2P	模拟	电压通道正向输入
46	AVDD	电源	计量部分模拟电源
47	AGND	地	计量部分模拟地
48	VREF	模拟	电压基准输出
49	NC	--	空脚, 可接地或电源
50	NRST	I	复位信号, 低电平复位
51	REVP	O	反向指示信号, 高电平指示反向信号
52	XIN	模拟	3.58MHz 晶振输入
53	XOUT	模拟	3.58MHz 晶振输出
54	XTIN	模拟	32.768KHz 晶振输入
55	XTOUT	模拟	32.768KHz 晶振输出
56	NC	--	空脚, 可接地或电源
57	CF	O	高频脉冲输出
58	GND	地	数字地
59	VDD	电源	数字电源
60	NC	--	空脚, 可接地或电源
61	VBIAS	模拟	LCD 显示的调整电阻
62	NC	--	空脚, 可接地或电源
63	PLL	模拟	PLL 外接电容
64	NC	--	空脚, 可接地或电源

表 2. SD3003 管脚描述

序号	管脚名称	属性	描述
48,1-14	P20/SEG0/DOUT-- P36/SEG14	LCD 驱动, I/O	LCD SEG0-SEG14 SEG0-7 跟 P20-P27 复用; SEG0 跟 OTP 烧录的数据输出脚复用 SEG8-14 跟 P30-P36 复用
15	P10	I/O	复用 INT0, 中断沿可选, 也跟 OTP 烧录的时钟输入脚 SCK 复用
16	P11	I/O	复用 INT1, 中断沿可选, 也跟 OTP 烧录的数据输入脚 DIN 复用
17	P12	I/O	复用 I2C 时钟端 SCL
18	P13	I/O	复用 I2C 数据端 SDA
19	P14	I/O	复用 UART 的数据发送 TXD
20	P15	I/O	复用 UART 的数据接收 RXD
21	P16	I/O	复用 8 位 PPG 输出
22	P17	I/O	复用 16 位 PPG 输出复用
23	GND	地	数字地
24	VDD	电源	数字电源
25	VPP	电源	OTP 编程电源脚
26-29	COM0--COM3	LCD 驱动, I/O	LCD 的 COM 端, 跟 P54-P57 复用
30	V1P	模拟	电流通道正相输入
31	V1N	模拟	电流通道反相输入
32	V2N	模拟	电压通道反相输入
33	V2P	模拟	电压通道正相输入
34	AVDD	电源	计量部分模拟电源
35	AGND	地	计量部分模拟地
36	VREF	模拟	电压基准输出
37	NRST	I	复位信号, 低电平复位
38	REVP	O	反向指示信号, 高电平指示反向信号
39	XIN	模拟	3.58MHz 高频晶振输入
40	XOUT	模拟	3.58MHz 高频晶振输出
41	XTIN	模拟	32.768kHz 晶振输入
42	XTOUT	模拟	32.768kHz 晶振输出
43	CF	O	计量高频脉冲输出
44	GND	地	数字地
45	VDD	电源	数字电源
46	VBIAS	模拟	LCD 显示的调整电阻
47	PLLC	模拟	PLL 外接电容

3 CPU

3.1 处理器内核，CPU core

8BITS RISC CPU，49 条指令，指令支持直接寻址和间接寻址，低功耗设计。绝大多数指令为一个指令周期（4 个时钟周期），个别指令为 2 个指令周期，有 6 级硬件堆栈。

3.2 存储器，Memory

3.2.1 概述

存储器的有 2 种，一种是程序存储器，由 OTP 构成，另一种为数据存储器（DM），由 SRAM 和特殊寄存器组成。

程序存储器包括：

主存储区（Main Program Memory，MPM）

程序计数器（Program Counter，PC）

堆栈（Stack，STK）

数据存储器包括：

特殊寄存器（Special Register，SR）

一般寄存器（General Purpose Register，GPR）

存储器相关寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BSR	--	--	--	--	存储区选择寄存器			
FSR0L	间接寻址寄存器 0 的低字节							
FSR0H	--	--	--	--	间接数据存储器地址指针 0 的高字节			
INDF0	使用 FSR0 的内容寻址数据存储器— FSR0 的值不改变（不是物理寄存器）							
PCL	程序计数器低字节							
PCLATH	--	--	--	--	程序计数器高字节			
STKPTR	stkful	stkunf	--	--	--	stkptr_reg[2:0]		
TOSL	栈顶寄存器低字节							
TOSH	--	--	栈顶寄存器高字节					

地址指针低位字节

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL (FF9H)								
R/W	R/W							
Reset	0000 0000							

地址指针高位字节

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCLATH(FFAH)	--	--	--	--				

R/W	--	--	--	--	R/W
Reset	--	--	--	--	0000

压栈指针寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STKPTR(FFCH)	stkful	stkunf	--	--	--	stkptr_reg[2:0]		
R/W	R	R	--	--	--	R/W		
Reset	0	0	--	--	--	000		

stkptr_reg[2:0]: 压栈的地址值, 从 000 到 111 依次代表堆栈 0 级到 7 级, 1-6 为有效堆栈空间。

stkful: 压栈满溢出标志位, “1”有效

stkunf: 压栈空溢出标志位, “1”有效. (仅上电或者掉电时才会复位)

栈顶低字节

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TOSL(FFDH)								
R/W	R/W							
Reset	0000 0000							

栈顶高字节

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TOSH(FFE H)	--	--	--	--				
R/W	--	--			R/W			
Reset	--	--	--		0000			

3.2.2 程序存储空间

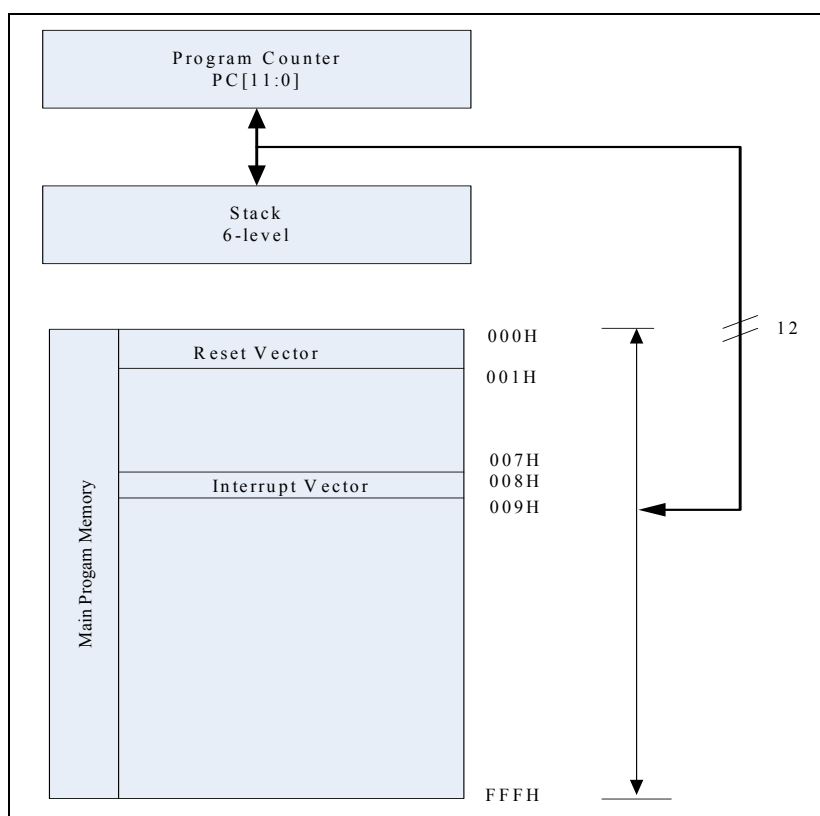


图3-1 程序存储器结构图

如图 2-1 所示，程序存储空间最大寻址范围从 000H 到 FFFH，共 4096 字节。

- 复位向量入口地址：0000H
- 中断向量入口地址：0008H

程序存储器为 OTP，在未进行烧录时，所有位均为“1”，写入后，各位根据写入的状态显示“0”或“1”。

用户程序举例：

```

org      0x0000          ;程序入口地址
goto     Start
org      0x0008          ;中断入口地址
goto     Hint
org      0x0020
Start:
    movlb 0x01
    movlw 0x5A
    ...
Hint:          ;中断服务程序
    ...
    retfie 1    ;中断返回
    ...
end            ;程序结束标志

```

3.3 程序计数器,PC

程序计数器 PC 由 PCLATH、PCL 组成。共 12 位。可直接读写。

- 读取 PC[11:0]，必须先读 PCL[7:0]后再读 PCLATH[3:0]，才能读到正确的数据，否则将读不到正确的数据。
- 写入 PC[11:0]，必须先写 PCLATH[3:0]后再写 PCL[7:0]，否则将写入错误的的数据。

用户程序举例（使程序跳转到 There 执行）：

```
movlw   high(There)      ;取 There 高字节，放入 PCLATH
movwf   PCLATH,0
movlw   low(There)       ;取 There 低字节，放入 PCL，放入后 PC 会立即跳转到 There
movwf   PCL,0
```

...

There:

...

3.4 堆栈, STK

堆栈 STK 主要由堆栈指示寄存器 STKPTR 和栈顶寄存器 TOSH/TOSL 组成。共 6 级堆栈，也就是说最多使用 6 级程序嵌套，如果开中断，最多只能 5 级嵌套。如采用开发片调试，最多采用 4 级程序嵌套。

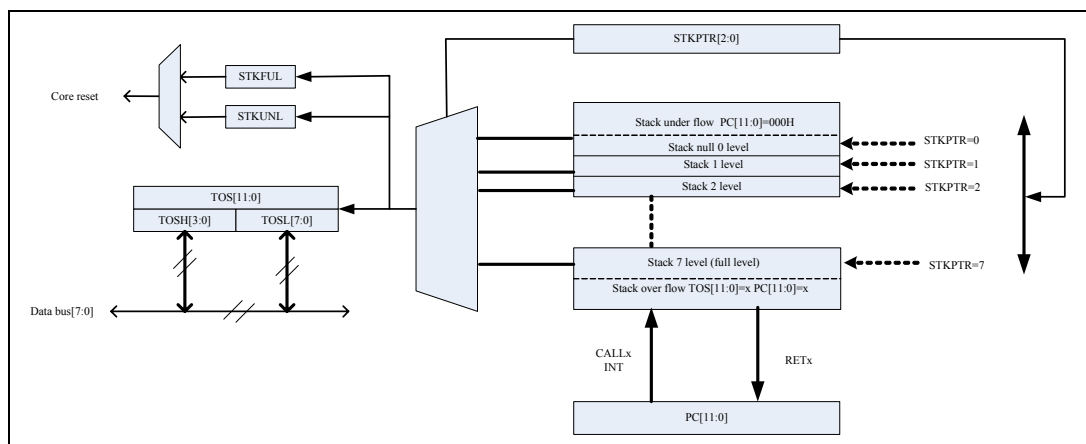


图 3-2 堆栈结构图

堆栈的栈顶寄存器 TOS[11:0] 有 12 位数据长度，有 2 个寄存器 TOSH[3:0]/TOSL[7:0] 组成。stkptr_reg[2:0]=0 时，堆栈空，当程序执行 CALL 或发生中断时，堆栈指示寄存器 stkptr_reg[2:0] 将做加一动作并将程序计数器 PC 地址写入 TOS[11:0] 寄存器；当程序执行 RETx 指令时，堆栈指示寄存器 stkptr_reg[2:0] 将做减一动作，在做减一动作前会将 TOS[11:0] 数据先写入 PC[11:0]，写入完成后，stkptr_reg[2:0] 才做减一动作，并使当下的 TOS[11:0] 值改变。

- 读取 TOS[11:0] 寄存器，直接读取即可。
- 写入 TOS[11:0] 寄存器，可以直接写入，也可通过 CALL 指令或中断(INT)将 PC[11:0] 数据写入。

在堆栈的过程中，会发生堆栈溢出或堆栈空等事件。

- 堆栈溢出：stkful 置“1”，产生复位信号，将 PC[11:0] 复位到 0000H。stkptr_reg [2:0]

- 堆栈空: `stkunf` 置“1”, 产生复位信号, 将 `PC[11:0]` 复位到 0000H。 `stkptr_reg [2:0]` 指示寄存器不受影响。用户需在清 `stkptr_reg [2:0]` 后, 再对寄存器 `stkunf` 清零。

3.5 数据存储空间

BANK 0 BSR<0000>	特殊寄存器I (000~04F)	000H
	一般寄存器 (050~0FF)	0FFH 100H
BANK 1 BSR<0001>	一般寄存器	17FH 180H
	一般寄存器 (Not Used)	1FFH
⋮	⋮ Not Used	
BANK 14 BSR<1110>	一般寄存器(Not Used)	E00H
BANK 15 BSR<1111>	一般寄存器 (F00~F4F)	EFFH F00H
	特殊寄存器II (F50~FFF)	FFFH

3.5.1 存储器与指令

指令集中位操作功能的指令最多会有“F”、“D”、“A”等三个参数。

“D”是指运算后的数据要放的地方。D=0 放在 WREG 寄存器、D=1 放在数据存储器

3.5.2 区域选择寄存器 BSR

- 当 A=0 时，无论 BSR[3:0]的指向何区域，对数据寄存器的读/写只会在特殊寄存器 I 和 II 区域。
- 当 A=1 时，对数据存储空间的读/写会依照 BSR[3:0]所指定的地址。对于本芯片，BSR[3:0]只能设置为 0001b，也就是只在地址 100H-17FH 范围内的 128 字节可以操作。

3.5.3 特殊寄存器

特殊寄存器包含 CPU core 与外围功能的相关寄存器，在特殊寄存器中常用于配合指令操作的寄存器有：一个是工作寄存器 WREG，另一个是间接寻址寄存器 FSR。

3.5.4 工作寄存器 WREG

工作寄存器简称 W，为配合指令使用最多的寄存器，例如数据搬移、运算与判断等。

3.5.5 间接寻址寄存器，FSR0 与 INDF0

FSR0 间接寻址寄存器由寄存器 FSR0[11:0]与索引寄存器 INDF0[7:0]组成。FSR0[11:0]分为 FSR0H[3:0]、FSR0L[7:0]两个寄存器。INDF0[7:0]为虚拟寄存器，对 INDF0 操作，实际是对以 FSR0[11:0]的数据作为地址的数据寄存器进行操作。对 INDF0 操作，BSR 的值不影响数据寄存器的寻址。

3.5.6 一般寄存器

一般寄存器为用户进行数据存储、运算等自由使用地址范围，在本芯片里有 128 字节的 SRAM 组成，地址范围 100H-17FH。

3.5.7 中断快速恢复寄存器

有三个寄存器：区域选择寄存器 BSR 的备份寄存器；状态寄存器 STATUS 的备份寄存器；工作寄存器 WREG 的备份寄存器。当发生中断或者使用快速调用时，WREG, BSR 和 STATUS 寄存器会自动备份到它们的备份寄存器中。

3.6 数据存储寄存器说明

间接寻址寄存器0的低字节

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR0L(FE9H)								
R/W	R/W							
Reset	0000 0000							

间接寻址寄存器0的高字节

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR0H(FEAH)	--	--	--	--				
R/W	--	--	--	--	R/W			
Reset	--	--	--	--	0000			

索引寄存器0

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INDF0(FEFH)	不是物理寄存器							

区域选择寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BSR(FE0H)	--	--	--	--	BSR[3:0]			
R/W	--	--	--	--	R/W			
Reset	--	--	--	--	0000			

工作寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WREG(FE8H)								
R/W	R/W							
Reset	XXXX XXXX							

3.7 寄存器列表

表 3-1. 寄存器描述

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
F58H	LCDCOM	--	--	--	--	com[3:0]			
F59H	LCDCON	lcden	--	--	biassel	cs1	cs0	--	--
F5AH	LCDSE0	SEG 选择寄存器 se[7:0]							
F5BH	LCDSE1	SEG 选择寄存器 se[15:8]							
F5CH	LCDSE2	SEG 选择寄存器 se[23:16]							
F60H	LCDDATA0	S01C3	S01C2	S01C1	S01C0	S00C3	S00C2	S00C1	S00C0
F61H	LCDDATA1	S03C3	S03C2	S03C1	S03C0	S02C3	S02C2	S02C1	S02C0
F62H	LCDDATA2	S05C3	S05C2	S05C1	S05C0	S04C3	S04C2	S04C1	S04C0
F63H	TMR1U	TMR1 的 24 位计数预重载寄存器的最高 8 位数据							
F64H	RD_TMR1U	TMR1 的 24 位当前计数值的最高 8 位数据							
F65H	CCPR1U	24 位捕捉寄存器的最高 8 位数据							
F66H	LCDDATA3	S07C3	S07C2	S07C1	S07C0	S06C3	S06C2	S06C1	S06C0
F67H	LCDDATA4	S09C3	S09C2	S09C1	S09C0	S08C3	S08C2	S08C1	S08C0
F68H	LCDDATA5	S11C3	S11C2	S11C1	S11C0	S10C3	S10C2	S10C1	S10C0
F69H	PPG2H_H	16 位 PPG 模块的高电平时间设置的高 8 位数据							
F6AH	PPG2H_L	16 位 PPG 模块的高电平时间设置的低 8 位数据							
F6BH	SSPADDD	I2C 波特率设置寄存器							
F6CH	SSPBUF	I2C 串行接收/ 发送缓冲器							
F6DH	SSPCON1	wcol	sspov	sspen	ckp	sspm3	sspm2	sspm1	sspm0
F6EH	SSPCON2	gcen	ackstat	ackdt	acken	rcen	pen	rsen	sen
F6FH	SSPSTAT	--	--	d/na	p	s	r/w	--	bf
F70H	PPG2CON	PPG2EN	--	--	--	--	--	ppg2s[1]	ppg2s[0]
F71H	LCDDATA6	S13C3	S13C2	S13C1	S13C0	S12C3	S12C2	S12C1	S12C0
F72H	LCDDATA7	S15C3	S15C2	S15C1	S15C0	S14C3	S14C2	S14C1	S14C0
F73H	LCDDATA8	S17C3	S17C2	S17C1	S17C0	S16C3	S16C2	S16C1	S16C0
F74H	PPG_CON	ppg_en	--	--	--	--	--	ppg_sel[1:0]	

F75H	PPG_SETH	8 位 PPG 模块的高电平时间设置寄存器							
F76H	PPG_SETP	8 位 PPG 模块的周期时间设置寄存器							
F77H	LCDDATA9	S19C3	S19C2	S19C1	S19C0	S18C3	S18C2	S18C1	S18C0
F78H	LCDDATA10	S21C3	S21C2	S21C1	S21C0	S20C3	S20C2	S20C1	S20C0
F79H	LCDDATA11	S23C3	S23C2	S23C1	S23C0	S22C3	S22C2	S22C1	S22C0
F7BH	RCSTA	spen	rx9	sren	cren	--	ferr	oerr	rx9d
F7CH	RCREG	UART 接收数据寄存器							
F7DH	TXSTA	csrc	tx9	txen	sync	--	--	trmt	tx9d
F7EH	TXREG	UART 发送数据寄存器							
F7FH	SPBRG	UART 波特率设置寄存器							
F80H	PT1	读 PORT1 引脚，写 PORT1 数据寄存器							
F81H	PT2	读 PORT2 引脚，写 PORT2 数据寄存器							
F82H	PT3	读 PORT3 引脚，写 PORT3 数据寄存器							
F83H	PT4	读 PORT4 引脚，写 PORT4 数据寄存器							
F84H	PT5	读 PORT5 引脚，写 PORT5 数据寄存器							
F85H	PT1PU	PORT1 端口上拉使能寄存器							
F86H	PPG2P_H	16 位 PPG 模块的周期时间设置的高 8 位数据							
F87H	PPG2P_L	16 位 PPG 模块的周期时间设置的低 8 位数据							
F89H	PT1EN	PORT1 端口方向设置寄存器							
F8AH	PT2EN	PORT2 端口方向设置寄存器							
F8BH	PT3EN	PORT3 端口方向设置寄存器							
F8CH	PT4EN	PORT4 端口方向设置寄存器							
F8DH	PT5EN	PORT5 端口方向设置寄存器							
F92H	LAT1	读 PORT1 数据寄存器，写 PORT1 数据寄存器							
F93H	LAT2	读 PORT2 数据寄存器，写 PORT2 数据寄存器							
F94H	LAT3	读 PORT3 数据寄存器，写 PORT3 数据寄存器							
F95H	LAT2	读 PORT4 数据寄存器，写 PORT4 数据寄存器							
F96H	LAT3	读 PORT5 数据寄存器，写 PORT5 数据寄存器							
F9BH	OSCTUNE	rcoscen	cryoscen	osc_en	a2	a1	a0	pllen	--
F9EH	PIR	--	--	sspif	tmr0if	secif	--	int1if	int0if
FA0H	PIE2	--	--	rcie	txie	belie	lvdie	tmr1ie	ccpie
FA1H	PIR2	--	--	rcif	txif	belif	lvdif	tmr1if	ccpif
FA9H	VOL_RESULTU	电压 24 位瞬时结果的最高 8 位							
FAAH	CURRENT_RESULTU	电流 24 位瞬时结果的最高 8 位							
FABH	COUNTER_CON	--	--	--	revp	up_ov	down_ov	up_en	down_en
FACH	COUNT_UP_H	电能脉冲 16 为正向计数器的高 8 位							
FADH	COUNT_UP_L	电能脉冲 16 为正向计数器的低 8 位							
FAEH	COUNT_DOWN_H	电能脉冲 16 为反向计数器的高 8 位							
FAFH	COUNT_DOWN_L	电能脉冲 16 为反向计数器的低 8 位							
FB1H	RD_TMR0L	定时器 0 的 8 位当前计数值寄存器							
FB2H	RD_TMR1H	TMR1 的 24 位当前计数值的中间 8 位数据							
FB3H	RD_TMR1L	TMR1 的 24 位当前计数值的最低 8 位数据							
FBEH	CCPR1L	24 位捕捉寄存器的最低 8 位数据							

FBFH	CCPR1H	24 位捕捉寄存器的中间 8 位数据							
FC0H	ADCON	scf	ac_dc	shutdown	cf_fast	s1	s0	g1	g0
FC1H	ADC_CON2	bitsel1	adc_en	mul_sel1	mul_sel0	ac_dc2	bitsel0	sel_ch1	sel_ch0
FC2H	RLT_ADJ0	平方和累加结果寄存器 add[7:0]							
FC3H	RLT_ADJ1	平方和累加结果寄存器 add[15:8]							
FC4H	RLT_ADJ2	平方和累加结果寄存器 add[23:16]							
FC5H	RLT_ADJ3	平方和累加结果寄存器 add[31:24]							
FC6H	RLT_ADJ4	平方和累加结果寄存器 add[39:32]							
FC7H	ADD_COUNT0	平方和累加次数寄存器 c[7:0]							
FC8H	ADD_COUNT1	--	--	--	--	--	--	c[9]	c[8]
FC9H	VOL_RESULTH	电压 24 位瞬时结果的中间 8 位							
FCAH	VOL_RESULTL	电压 24 位瞬时结果的最低 8 位							
FCBH	CURRENT_RESULTH	电流 24 位瞬时结果的中间 8 位							
FCCH	CURRENT_RESULTL	电流 24 位瞬时结果的最低 8 位							
FCDH	T1CON	--	--	t1ps[1]	t1ps[0]	--	--	cf_cap_en	tmr1_en
FCEH	TMR1L	计数器预设值寄存器低 8 位							
FCFH	TMR1H	计数器预设值寄存器中间 8 位							
FD1H	WDTCN	--	--	--	--	--	wdtsel[1:0]	wdten	
FD2H	LVDCON	fen	--	--	lvdreset_en	lvd_4v	lvd_2o2v	--	lvd2o2v_en
FD3H	OSCCON	--	ircf2	ircf1	ircf0	--	idlen	clk_sel[1:0]	
FD5H	T0CON	tmr0_en	--	--	--	--	t0ps[2]	t0ps[1]	t0ps[0]
FD6H	TMR0L	定时器 0 预设值寄存器							
FD8H	STATUS	un_ins	pd	to	--	--	zero	dc	carry
FE0H	BSR	--	--	--	--	存储区选择寄存器			
FE8H	WREG	工作寄存器							
FE9H	FSR0L	间接数据存储器地址指针 0 的低字节							
FEAH	FSR0H	--	--	--	--	间接数据存储器地址指针 0 的高字节			
FEFH	INDF0	使用 FSR0 的内容寻址数据存储器— FSR0 的值不改变（不是物理寄存器）							
FF1H	INTCON2	--	--	--	--	--	--	intedg1	intedg0
FF2H	INTCON	Gie	0	sspie	tmr0ie	secie	--	intl1ie	int0ie
FF5H	TABLAT	表读/写寄存器							
FF6H	TBLPTRL	表地址低字节							
FF7H	TBLPTRH	--	--	表地址高字节					
FF9H	PCL	程序计数器低字节							
FFAH	PCLATH	--	--	--	--	程序计数器高字节			
FFCH	STKPTR	stkful	stkunf	--	--	--	stkptr_reg[2:0]		
FFDH	TOSL	栈顶寄存器低字节							
FFEH	TOSH	--	--	栈顶寄存器高字节					

4 指令集

4.1 操作码字符介绍

A	快速操作 RAM 位 A = 0: 快速操作寄存器单元 (BSR 寄存器被忽略) A = 1: 由 BSR 寄存器指定的寄存器存储区
F	8 位寄存器地址 (00H 到 FFH), 或 2 位 FSR 标识符 (0H 到 3H)。
B	8 位文件寄存器内的位地址 (0 到 7)。
D	目标寄存器选择位 D = 0: 结果保存至 WREG 寄存器 D = 1: 结果保存至文件寄存器 F。
K	立即数、常数或者标号 (可能是 8 位、12 位或 20 位的值)
N	相对跳转指令的相对地址 (二进制补码), 或 CALL/BRANCH 和 RETURN 指令的直接地址。
S	快速调用 / 返回模式选择位。 S = 0: 不对影子寄存器进行更新, 也不用影子寄存器的内容更新其他寄存器 S = 1: 将寄存器的值存入影子寄存器或把影子寄存器中的值载入寄存器 (快速模式)
C、DC、Z	ALU 状态位: 进位标志位、辅助进位标志位、全零标志位。
PC	程序计数器。
PCL	程序计数器低字节
PCH	程序计数器高字节
PCLATH	程序计数器高字节锁存器
TBLPTR	14 位表指针 (指向程序存储器地址)
TABLAT	低 8 位表锁存器
TABLATH	高 8 位表锁存器
TOS	栈顶
WDT	看门狗定时器
WREG	工作寄存器

4.2 指令集介绍

表 4-1. 指令集列表

指令	说明	周期	MSB		LSB		受影响的状态位
寄存器操作指令							
ADDWF F, D, A	WREG 与 F 相加	1	0010	01DA	FFFF	FFFF	C, DC, Z,
ADDWFC F, D, A	WREG 与 F 带进位相加	1	0010	00DA	FFFF	FFFF	C, DC, Z,
ANDWF F, D, A	WREG 和 F 做与运算	1	0001	01DA	FFFF	FFFF	Z,
CLRF F, A	F 清零	1	0110	101A	FFFF	FFFF	Z
COME	F 取反	1	0001	11DA	FFFF	FFFF	Z,

F, D, A							
DECf F, D, A	F 减 1	1	0000	01DA	FFFF	FFFF	C, DC, Z,
DECfSZ F, D, A	F 减 1, 为 0 则 跳过	1 (2 或3)	0010	11DA	FFFF	FFFF	无
DCfSNZ F, D, A	F 减 1, 非 0 则 跳过	1 (2 或3)	0100	11DA	FFFF	FFFF	无
INCF F, D, A	F 加 1	1	0010	10DA	FFFF	FFFF	C, DC, Z,
INCFSZ F, D, A	F 加 1, 为 0 则 跳过	1 (2 或3)	0011	11DA	FFFF	FFFF	无
INFSNZ F, D, A	F 加 1, 非 0 则 跳过	1 (2 或3)	0100	10DA	FFFF	FFFF	无
CPFSEQ F, A	将 F 与 WREG 做比较, 相等则 跳过	1 (2 或3)	0110	001A	FFFF	FFFF	无
IORWF F, D, A	WREG 和 F 做 或运算	1	0001	00DA	FFFF	FFFF	Z,
MOVF F, D, A	移动 F	1	0101	00DA	FFFF	FFFF	Z,
MOVWF F, A	将 WREG 移入 F	1	0110	111A	FFFF	FFFF	无
RLCF F, D, A	对 F 执行带进位 的循环左移	1	0011	01DA	FFFF	FFFF	C, Z,
RLNCF F, D, A	F 循环左移 (不 带进位)	1	0100	01DA	FFFF	FFFF	Z,
RRCF F, D, A	对 F 执行带进位 的循环右移	1	0011	00DA	FFFF	FFFF	C, Z,
RRNCF F, D, A	F 循环右移 (不 带进位)	1	0100	00DA	FFFF	FFFF	Z,
SETE F, A	将 F 置为全 1	1	0110	100A	FFFF	FFFF	无
SUBWF F, D, A	F 减去 WREG	1	0101	11DA	FFFF	FFFF	C, DC, Z,
SUBWFB F, D, A	F 减去 WREG (带借位)	1	0101	10DA	FFFF	FFFF	C, DC, Z,
TSTFSZ F, A	测试 F, 为 0 则 跳过	1 (2 或3)	0110	011A	FFFF	FFFF	无
XORWF F, D, A	WREG 和 F 异 或运算	1	0001	10DA	FFFF	FFFF	Z
位操作指令							
BCF F, B, A	将 F 中的某位清 零	1	1001	BBBA	FFFF	FFFF	无
BSF	将 F 中的某位置	1	1000	BBBA	FFFF	FFFF	无

F, B, A	1						
BTFS F, B, A	测试 F 中的某 位, 为 0 则跳过	1 (2 或 3)	1011	BBBA	FFFF	FFFF	无
BTFS F, B, A	测试 F 中的某 位, 为 1 则跳过	1 (2 或 3)	1010	BBBA	FFFF	FFFF	无
控制指令							
BRA N	无条件跳转	2	1101	0NNN	NNNN	NNNN	无
CALL N, S	调用子程序 (第 一个字) (第二个字)	2	1110	110S	KKKK	KKKK	无
			1111	KKKK	KKKK	KKKK	
CLRWD T	看门狗定时器清 零	1	0000	0000	0000	0100	TO, PD
GOTO N	跳转到地址 (第 一个字) (第二个字)	2	1110	1111	KKKK	KKKK	无
			1111	KKKK	KKKK	KKKK	
NOP —	空操作	1	0000	0000	0000	0000	无
POP —	从返回堆栈栈顶 (TOS) 出栈	1	0000	0000	0000	0110	无
PUSH —	从返回堆栈栈顶 (TOS) 进栈	1	0000	0000	0000	0101	无
NOP —	空操作	1	1111	XXXX	XXXX	XXXX	无
RCALL N	相对调用	2	1101	1NNN	NNNN	NNNN	无
RESET	用软件使器件复 位	1	0000	0000	1111	1111	全部
RETFIE S	中断返回使能	2	0000	0000	0001	000S	GIE
RETLW K	返回时将立即数 送入 WREG	2	0000	1100	KKKK	KKKK	无
RETURN S	从子程序返回	2	0000	0000	0001	001S	无
SLEEP	进入待机模式	1	0000	0000	0000	0011	TO, PD
立即数操作类指令							
ADDLW K	WREG 与立即 数相加	1	0000	1111	KKKK	KKKK	C, DC, Z,
ANDLW K	WREG 和立即 数进行与运算	1	0000	1011	KKKK	KKKK	Z,
IORLW K	WREG 和立即 数进行或运算	1	0000	1001	KKKK	KKKK	Z,
MOVLB	将立即数移入	1	0000	0001	0000	KKKK	无

K	BSR<3:0>						
MOVLW K	将立即数移入 WREG	1	0000	1110	KKKK	KKKK	无
RETLW K	返回时将立即数 送入 WREG	2	0000	1100	KKKK	KKKK	无
SUBLW K	立即数减去 WREG	1	0000	1000	KKKK	KKKK	C, DC, Z,
XORLW K	WREG 和立即 数进行异或运算	1	0000	1010	KKKK	KKKK	Z
数据存储器↔ 程序存储器操作							
TBLRD*	表读	2	0000	0000	0000	1000	无
TBLWT*	表写	2	0000	0000	0000	1100	无
寄存器间接寻址							
对寄存器 INDF0 的操作实际上是以寄存器 FSR0H, FSR0L 的内容作为地址去操							

备注：1. POP, PUSH 只用于仿真模式，不提供用户使用。

4.3 指令说明

语法： ADDLW K

操作数： $0 \leq K \leq 255$

操作： $(WREG) + K \rightarrow WREG$

说明： 将WREG 寄存器的内容与8位立即数K相加，结果保存在WREG寄存器。

示例： ADDLW 0X15

指令执行前，WREG = 0X10；

指令执行后，WREG = 0X25；

4.3.2 ADDWF – WREG 与F 寄存器相加

语法： ADDWF F,D,A

操作数： $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作： $(WREG) + (F) \rightarrow DEST$

说明： 将WREG 的内容和F 寄存器的内容相加。如果D 为0，结果存储到WREG 中，如果D 为1，结果存回寄存器F（默认）。如果A为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例： ADDWF REG,0,0

指令执行前，WREG = 0X17，REG = 0XC2；

指令执行后，WREG = 0XD9，REG = 0XC2。

4.3.3 ADDWFC – WREG 与F 寄存器带进位相加

语法： ADDWFC F,D,A

操作数： $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(WREG) + (F) + (C) \rightarrow DEST$

说明: 将WREG 的内容、进位标志位C 和F 寄存器的内容相加。如果D为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: ADDWFC REG,1,1

指令执行前, $C = 1$, $WREG = 0X4D$, $REG = 0X02$;

指令执行后, $C = 0$, $WREG = 0X4D$, $REG = 0X50$ 。

4.3.4 ANDLW – 立即数和WREG 寄存器作逻辑与运算

语法: ANDLW K

操作数: $0 \leq K \leq 255$

操作: $(WREG) \& K \rightarrow WREG$

说明: 将WREG 寄存器的内容与8 位立即数K 相与, 结果保存在WREG寄存器。

示例: ANDLW 0X00

指令执行前, $WREG = 0XFF$;

指令执行后, $WREG = 0X00$ 。

4.3.5 ANDWF – 将WREG 和F 作逻辑与运算

语法: ANDWF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(WREG) \& (F) \rightarrow DEST$

说明: 将WREG 的内容和F 寄存器的内容相与。如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: ANDWF REG,1,0

指令执行前, $WREG = 0X55$, $REG = 0XAA$;

指令执行后, $WREG = 0X55$, $REG = 0X00$ 。

4.3.6 BCF – 将F 寄存器中的某位清零

语法: BCF F,B,A

操作数: $0 \leq F \leq 255$

$0 \leq B \leq 7$

$A \in [0,1]$

操作: $0 \rightarrow F \langle B \rangle$

说明: 将寄存器F 中的位B 清零。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: BCF REG,7,1

指令执行前, $REG = 0XFF$;

指令执行后, $REG = 0X7F$ 。

4.3.7 BSF – 将F 寄存器中的某位置1

语法: BSF F,B,A

操作数: $0 \leq F \leq 255$

$0 \leq B \leq 7$

$A \in [0,1]$

操作: $0 \rightarrow F \leftarrow B$

说明: 将寄存器F 中的位B 置1。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: BSF REG,7,0

指令执行前, REG = 0X00;

指令执行后, REG = 0X80。

4.3.8 BRA – 无条件跳转

语法: BRA N

操作数: $-1024 \leq N \leq 1023$

操作: $(PC) + 2 + 2N \rightarrow (PC)$

说明: 二进制补码“2N”与PC 相加, 由于PC 要先递增才能取下一条指令, 所以新地址将为

$PC + 2 + 2N$ 。

示例: HERE:

BRA THERE

指令执行前, PC = 地址(HERE);

指令执行后, PC = 地址(THERE)。

4.3.9 BTFSC – 测试寄存器中的位, 为0 则跳过后条指令

语法: BTFSC F,B,A

操作数: $0 \leq F \leq 255$

$0 \leq B \leq 7$

$A \in [0,1]$

操作: 如果 $F \leftarrow B = 0$ 则跳过后条指令

说明: 如果寄存器F 的位B 为0, 则跳过后条指令。即在位B 为0 时, 丢弃下一条指令而执行一条NOP 指令。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: BTFSC REG,7,0

BSF REG,6,0

BCF REG,0,0

指令执行前, REG = 0X01;

指令执行后, REG = 0X00。

4.3.10 BTFSS – 测试寄存器中的位, 为1 则跳过后条指令

语法: BTFSS F,B,A

操作数: $0 \leq F \leq 255$

$0 \leq B \leq 7$

$A \in [0,1]$

操作: 如果 $F \leftarrow B = 0$ 则跳过后条指令

说明: 如果寄存器F 的位B 为1, 则跳过后条指令。即在位B 为1 时, 丢弃下一条指

令而执行一条NOP 指令。如果A 为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例： BTFSS REG,7,1

BSF REG,6,1

BCF REG,0,1

指令执行前，REG = 0X01；

指令执行后，REG = 0X40。

4.3.11 CALL – 调用子程序

语法： CALL N{,S}

操作数： $0 \leq N \leq 1048575$

$S \in [0,1]$

操作： (PC) + 4 -> TOS, N -> PC<20:1>, 如果S = 1: (WREG) -> WS, STATUS -> STATUSS, BSR -> BSRS

说明： 可在整个2MB 的存储范围内进行子程序调用。首先，将返回地址(PC + 4)压栈，如果S = 1，还会将WREG、STATUS 和BSR 存入对应的影子寄存器，如果S = 0（默认），则不会，然后将20位的值N 装入PC<20:1>。

示例： HERE:

CALL THERE

指令执行前，PC = 地址(HERE)；

指令执行后，PC = 地址(THERE)

TOS = 地址 (HERE + 4)，其中TOS为栈顶。

4.3.12 CLRF – 将F清零

语法： CLRF F,A

操作数： $0 \leq F \leq 255$

$A \in [0,1]$

操作： 0X00 -> F

说明： 清零寄存器F 中。如果A 为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例： CLRF REG,0

指令执行前，REG = 0XFF；

指令执行后，REG = 0X00。

4.3.13 CLRWD – 清零看门狗定时器

语法： CLRWD

操作数： 无

操作： 0X00 -> WDT

说明： 清零看门狗定时器（喂狗）。

示例： CLRWD

指令执行前，WDT 计数器 = 0XXX；

指令执行后，WDT 计数器 = 0X00。

4.3.14 COMF – 将F 取反

语法: COMF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(\sim F) \rightarrow \text{DEST}$

说明: 将寄存器F 的内容取反。如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: COMF REG,0,0

指令执行前, WREG = 0X00, REG = 0X55;

指令执行后, WREG = 0XAA, REG = 0X55。

4.3.15 CPFSEQ – 比较F 和WREG, 如果F = WREG 则跳过下条指令

语法: CPFSEQ F,A

操作数: $0 \leq F \leq 255$

$A \in [0,1]$

操作: $(F) - \text{WREG}$, 如果F = WREG 则跳过下条指令 (无符号比较)

说明: 通过执行无符号的减法, 将寄存器F 的内容与WREG 进行比较, 相等则跳过下条指令, 即丢弃下一条指令而执行一条NOP 指令。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: CFPSEQ REG,1

BSF REG,6,1

BCF REG,0,1

指令执行前, WREG = 0X00, REG = 0X00;

指令执行后, WREG = 0X00, REG = 0X00。

4.3.16 DECF – F 减1

语法: DECF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(F) - 1 \rightarrow \text{DEST}$

说明: 将寄存器F 的内容减1。如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: DECF REG,1,1

指令执行前, WREG = 0X00, REG = 0X00;

指令执行后, WREG = 0X00, REG = 0XFF。

4.3.17 DECFSZ – F 减1, 为0 则跳过下条指令

语法: DECFSZ F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: (F) - 1 -> DEST, 结果为0 时跳过下条指令

说明: 将寄存器F 的内容减1。如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果结果为0, 则跳过下条指令, 即丢弃下条指令, 改为执行NOP。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: DECFSZ REG,0,1

BSF REG1,7,0

...

指令执行前, WREG = 0X00, REG = 0XFF, REG1 = 0X00;

指令执行后, WREG = 0XFE, REG = 0XFF, REG1 = 0X80。

4.3.18 DCFSNZ - F 减1, 不为0 则跳过下条指令

语法: DCFSNZ F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: (F) - 1 -> DEST, 结果不为0 时跳过下条指令

说明: 将寄存器F 的内容减1。如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果结果不为0, 则跳过下条指令, 即丢弃下条指令, 改为执行NOP。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: DCFSNZ REG,1,1

BSF REG1,7,0

...

指令执行前, WREG = 0XFF, REG = 0X01, REG1 = 0X00;

指令执行后, WREG = 0XFF, REG = 0X00, REG1 = 0X00。

4.3.19 GOTO – 无条件跳转

语法: GOTO N

操作数: $0 \leq N \leq 1048575$

操作: $N \rightarrow PC<20:1>$

说明: GOTO 指令运行无条件跳转到整个2MB 存储范围中的任何位置。将20 位值N 装入PC<20:1>。

示例: HERE:

GOTO THERE

指令执行前, PC = 地址(HERE);

指令执行后, PC = 地址(THERE)。

4.3.20 INCF – F 加1

语法: INCF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: (F) + 1 -> DEST

说明：将寄存器F 的内容加1。如果D 为0，结果存储到WREG 中，如果D 为1，结果存回寄存器F（默认）。如果A 为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例：INCF REG1,1

指令执行前，WREG = 0XFF，REG = 0XFF；

指令执行后，WREG = 0XFF，REG = 0X00。

4.3.21 INCFSZ – F 加1，为0 则跳过下条指令

语法：INCFSZ F,D,A

操作数：0≤F≤255

D ∈ [0,1]

A ∈ [0,1]

操作：(F) + 1 -> DEST，结果为0 时跳过下条指令

说明：将寄存器F 的内容加1。如果D 为0，结果存储到WREG 中，如果D 为1，结果存回寄存器F（默认）。如果结果为0，则跳过下条指令，即丢弃下条指令，改为执行NOP。如果A 为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例：INCFSZ REG0,1

BSF REG1,7,0

...

指令执行前，WREG = 0X00，REG = 0XFE，REG1 = 0X00；

指令执行后，WREG = 0XFF，REG = 0XFE，REG1 = 0X80。

4.3.22 INFSNZ – F 加1，不为0 则跳过下条指令

语法：INFSNZ F,D,A

操作数：0≤F≤255

D ∈ [0,1]

A ∈ [0,1]

操作：(F) - 1 -> DEST，结果不为0 时跳过下条指令

说明：将寄存器F 的内容加1。如果D 为0，结果存储到WREG 中，如果D 为1，结果存回寄存器F（默认）。如果结果不为0，则跳过下条指令，即丢弃下条指令，改为执行NOP。如果A 为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例：INFSNZ REG1,1

BSF REG1,7,0

...

指令执行前，WREG = 0X55，REG = 0X00，REG1 = 0X00；

指令执行后，WREG = 0X55，REG = 0X01，REG1 = 0X00。

4.3.23 IORLW – 将立即数与W 作逻辑或运算

语法：IORLW K

操作数：0≤K≤255

操作：(WREG) | K -> WREG

说明：将WREG 寄存器的内容与8 位立即数K 相或，结果保存在WREG寄存器。

示例: IORLW 0XAA

指令执行前, WREG = 0X55;

指令执行后, WREG = 0XFF。

4.3.24 IORWF – 将WREG 和F 作逻辑或运算

语法: IORWF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(WREG) | (F) \rightarrow DEST$

说明: 将WREG 的内容和F 寄存器的内容相或。如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: IORWF REG,1,0

指令执行前, WREG = 0X55, REG = 0XAA;

指令执行后, WREG = 0X55, REG = 0XFF。

4.3.25 MOVF – 移动F

语法: MOVF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(F) \rightarrow DEST$

说明: 移动寄存器F, 如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: MOVF REG,0,1

指令执行前, WREG = 0X55, REG = 0XAA;

指令执行后, WREG = 0XAA, REG = 0XAA。

4.3.26 MOVLB – 将立即数移入BSR 的低半字节

语法: MOVLB K

操作数: $0 \leq K \leq 255$

操作: $K \rightarrow BSR$

说明: 将8 位立即数K 移入存储区选择寄存器BSR, 不管K7:K4 的值如何, BSR<7:4> 的值保持为0。

示例: MOVLB 0X01

指令执行前, BSR = 0X00;

指令执行后, BSR = 0X01。

4.3.27 MOVLW – 将立即数移入WREG

语法: MOVLW K

操作数: $0 \leq K \leq 255$

操作: K -> WREG

说明: 将8 位立即数K 移入工作寄存器WREG。

示例: MOVLW 0X01

指令执行前, WREG = 0X00;

指令执行后, WREG = 0X01。

4.3.28 MOVWF – 将WREG 内容移入F

语法: MOVWF F,A

操作数: $0 \leq F \leq 255$

$A \in [0,1]$

操作: (WREG) -> F

说明: 将WREG 中的数据移入寄存器F。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: MOVWF REG1

指令执行前, WREG = 0X55, REG = 0X00;

指令执行后, WREG = 0X55, REG = 0X55。

4.3.29 NOP – 执行空操作

语法: NOP

操作数: 无

操作: 空操作

说明: 不进行任何操作。

4.3.30 RCALL – 相对调用

语法: RCALL N

操作数: $-1024 \leq N \leq 1023$

操作: $(PC) + 2 \rightarrow TOS$

$(PC) + 2 + 2N \rightarrow (PC)$

说明: 从当前地址跳转 (最多1K) 来调用子程序。首先, 将返回地址(PC + 2)压入返回堆栈, 然后, 将二进制补码“2N”与PC 相加, 因为PC 要先递增才能取下条指令, 因此新的地址将为

$PC + 2 + 2N$ 。

示例: HERE:

RCALL THERE

指令执行前, PC = 地址(HERE);

指令执行后, PC = 地址(THERE); TOS = 地址(HERE + 2)

4.3.31 RESET – 复位

语法: RESET

操作数: 无

操作: 将所有受复位影响的寄存器和标志位复位, 0X00 -> PC。

说明: 软件复位。

示例: HERE:

RESET

指令执行前, PC = 地址 (HERE);

指令执行后, PC = 0X00; 寄存器 = 复位值; 标志位 = 复位值

4.3.32 RETFIE – 从中断返回

语法: RETFIE {S}

操作数: $S \in [0,1]$

操作: (TOS) \rightarrow PC, 如果 $S = 1$, (WREGS) \rightarrow WREG; (STATUS) \rightarrow STATUS; (BSRS) \rightarrow BSR; 如果 $S = 0$ (默认), 则不更新。PCLATU和PCLATH保持不变。

说明: 从中断返回, 执行出栈操作, 将栈顶(TOS)的内容装入PC。通过将全局中断使能位置1, 来重新使能中断。

4.3.33 RETLW – 将立即数返回给WREG

语法: RETLW K

操作数: $0 \leq K \leq 255$

操作: $K \rightarrow$ WREG; (TOS) \rightarrow PC; PCLATU和PCLATH保持不变。

说明: 将8 位立即数K 装入WREG, 将栈顶(TOS)的内容装入PC。

4.3.34 RETURN – 从子程序返回

语法: RETURN {S}

操作数: $S \in [0,1]$

操作: (TOS) \rightarrow PC; 如果 $S = 1$: (WREGS) \rightarrow WREG; (STATUS) \rightarrow STATUS; (BSRS) \rightarrow BSR; PCLATU和PCLATH保持不变。如果 $S = 0$ (默认), 则不更新。

说明: 从子程序返回, 执行出栈操作, 将栈顶(TOS)的内容装入PC。

4.3.35 RLCF – F 带进位循环左移

语法: RLCF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(F \langle N \rangle) \rightarrow \text{DEST} \langle N+1 \rangle$

$(F \langle 7 \rangle) \rightarrow C$

$(C) \rightarrow \text{DEST} \langle 0 \rangle$

说明: 将寄存器F 的内容连同进位标志位一起循环左移1 位, 如果D为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: RLCF REG,0,0

指令执行前, $C = 0$, WREG = 0X00, REG = 0XAA(1010 1010);

指令执行后, $C = 1$, WREG = 0X55(0101 0101), REG = 0XAA。

4.3.36 RLNCF – F 循环左移 (不带进位)

语法: RLNCF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(F<N>) \rightarrow DEST<N+1>$

$(F<7>) \rightarrow DEST<0>$

说明: 将寄存器F 的内容循环左移1 位, 如果D 为0, 结果存储到WREG中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: RLNCF REG,1,0

指令执行前, WREG = 0X00, REG = 0XAA(1010 1010);

指令执行后, WREG = 0X00, REG = 0X55(0101 0101)。

4.3.37 RRCF – F 带进位循环右移

语法: RRCF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(F<N>) \rightarrow DEST<N-1>$

$(F<0>) \rightarrow C$

$(C) \rightarrow DEST<7>$

说明: 将寄存器F 的内容连同进位标志位一起循环右移1 位, 如果D为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: RRCF REG,0,1

指令执行前, C = 1, WREG = 0X00, REG = 0XAA(1010 1010);

指令执行后, C = 0, WREG = 0XD5(1101 0101), REG = 0XAA。

4.3.38 RRNCF – F 循环右移 (不带进位)

语法: RRNCF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(F<N>) \rightarrow DEST<N-1>$

$(F<0>) \rightarrow DEST<7>$

说明: 将寄存器F 的内容循环右移1 位, 如果D 为0, 结果存储到WREG中, 如果D 为1, 结果存回寄存器F (默认)。如果A 为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: RRNCF REG,1,0

指令执行前, WREG = 0X00, REG = 0XAA(1010 1010);

指令执行后, WREG = 0X00, REG = 0X55(0101 0101)。

4.3.39 SETF – 将F 的内容置为全1

语法: SETF F,A

操作数: $0 \leq F \leq 255$

$A \in [0,1]$

操作: $0XFF \rightarrow F$

说明：将寄存器F 的内容置为0XFF。如果A 为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例：SETF REG,0

指令执行前，REG = 0X00；

指令执行后，REG = 0XFF。

4.3.40 SLEEP – 进入休眠模式

语法：SLEEP

操作数：无

操作：0X00 -> WDT

MCU 停止工作。

说明：如果设置为待机模式（OSCCON 寄存器的IDLEN 为0），则MCU 停止工作，但振荡器仍然振荡；如果设置为休眠模式（IDLEN 为1），则MCU 停止工作，振荡器停止振荡。

4.3.41 SUBLW – 立即数减去WREG

语法：SUBLW K

操作数： $0 \leq K \leq 255$

操作： $K - (WREG) \rightarrow WREG$

说明：将8 位立即数K 减去WREG，结果保存在WREG 寄存器。

示例：SUBLW 0X15

指令执行前，WREG = 0X14；

指令执行后，WREG = 0X01；

4.3.42 SUBWF - F 减去WREG（不带借位）

语法：SUBWF F,D,A

操作数： $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作： $(F) - (WREG) \rightarrow DEST$

说明：将F 寄存器的内容减去WREG 中的内容。如果D 为0，结果存储到WREG 中，如果D 为1，结果存回寄存器F（默认）。如果A为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例：SUBWF REG,0,0

指令执行前，WREG = 0X17，REG = 0XC9；

指令执行后，WREG = 0XB2，REG = 0XC9。

4.3.43 SUBWFB – F 减去WREG（带借位）

语法：SUBWFB F,D,A

操作数： $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作： $(F) - (WREG) - (\sim C) \rightarrow DEST$

说明：将F 寄存器的内容减去WREG 的内容和进位（借位）（通过二进制补码进行运算）。

如果D 为0，结果存储到WREG 中，如果D为1，结果存回寄存器F（默认）。如果A 为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例： SUBWFB REG,1,1

指令执行前，C = 1，WREG = 0X02，REG = 0X05；

指令执行后，C = 1，WREG = 0X02，REG = 0X03。

4.3.44 TBLRD

语法： TBLRD *

操作数： 无

操作： (程序存储区(TBLPTR)) -> TABLAT

说明： 该指令用于读取程序存储器的内容，使用表指针(TBLPTR)对程序存储器进行寻址。TBLPTR(21 位指针)，寻址范围为2MB。

示例：

MOVLW HIGH(HERE) ;取高字节

MOVWF TBLPTRH,0

MOVLW LOW(HERE) ;取低字节

MOVWF TBLPTRL,0

TBLRD *

MOVF TABLAT,0,0

NOP

...

HERE:

DB 0X55 ;定义字节0X55 在HERE 地址处

指令执行前，WREG = 0X00，TABLAT = 0X00；

指令执行后，WREG = 0X55，TABLAT = 0X55。

4.3.45 TSTFSZ – 测试F，为0 则跳过下条指令

语法： TSTFSZ F,A

操作数： $0 \leq F \leq 255$

$A \in [0,1]$

操作： 如果F = 0X00 则跳过下条指令。

说明： 如果F = 0X00，丢弃下一条指令而执行一条NOP 指令。如果A为0，选择快速操作存储区（SFR），如果A 为1，使用BSR 选择GPR 存储区（SRAM）。

示例： TSTFSZ REG,1

BSF REG,6,1

BCF REG,0,1

指令执行前，REG = 0X00；

指令执行后，REG = 0X00。

4.3.46 XORLW – 立即数与WREG 做逻辑异或运算

语法： XORLW K

操作数： $0 \leq K \leq 255$

操作： $(WREG) \wedge K \rightarrow WREG$

说明： 将WREG 寄存器的内容与8 位立即数K 相异或，结果保存在WREG寄存器。

示例: XORLW 0XAF

指令执行前, WREG = 0XB5;

指令执行后, WREG = 0X1A。

4.3.47 XORWF -将WREG 和F 作逻辑异或运算

语法: XORWF F,D,A

操作数: $0 \leq F \leq 255$

$D \in [0,1]$

$A \in [0,1]$

操作: $(WREG) \wedge (F) \rightarrow DEST$

说明: 将WREG 的内容和F 寄存器的内容相异或。如果D 为0, 结果存储到WREG 中, 如果D 为1, 结果存回寄存器F (默认)。如果A为0, 选择快速操作存储区 (SFR), 如果A 为1, 使用BSR 选择GPR 存储区 (SRAM)。

示例: XORWF REG,1,0

指令执行前, WREG = 0XB5, REG = 0XAF;

指令执行后, WREG = 0XB5, REG = 0X1A。

5 时钟系统

5.1 概述

SD3004 提供了丰富的时钟选择。主要提供了四个时钟源: 有 32.768K 晶振/3.58MHz 晶振/32K RC 振荡/PLL, 可以选择作为系统时钟。

- RC32K, 为内部低频 RC 振荡器, 振荡频率 32kHz;
- OSC32K, 为外部 32.768kHz 晶振;
- HOSC, 为 3.58MHz 晶振;
- PLLOSC, 为内部 3.604MHz 锁相环时钟。

四个时钟源根据寄存器配置都可以用作系统时钟。芯片上电时, 所有时钟默认开启。启动时系统时钟默认为 RC32K/2, 由于 RC32K 上电后 可以很快启动, 使得芯片上电后能够迅速进入工作状态。

时钟控制相关寄存器:

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCON	--	ircf2	ircf1	ircf0	--	idlen	clk_sel[1:0]	
OSCTUNE	rcoscen	cryoscen	osc_en	a2	a1	a0	pllen	--
WDTCON	--	--	--	--	--	wdtsel[1:0]		wdten

5.2 CPU 及外围模块时钟源

四个时钟源 RC32K、OSC32K、HOSC、PLLOSC 或者它们的分频信号进入芯片的 CPU 与外围模块。

5.2.1 CPU 时钟源

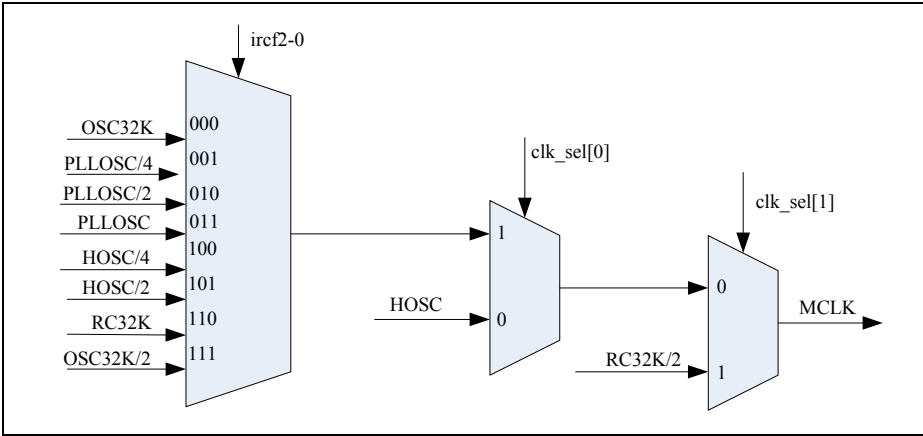


图 5-1 系统时钟选择

系统时钟选择寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCON (FD3H)	--	ircf2	ircf1	ircf0	--	idlen	clk_sel[1:0]	
R/W	--	R/W	R/W	R/W	--	R/W	R/W	
Reset	--	0	0	0	--	0	11	

ircf2-0 (clk_sel[1:0]=01) 时，系统时钟选择情况:

ircf2-0	时钟选择
000	OSC32K
001	PLLOSC/4
010	PLLOSC /2
011	PLLOSC
100	HOSC/4
101	HOSC/2
110	RC32K
111	OSC32K/2

idlen: 执行 SLEEP 指令后，进入模式选择。

0: 待机模式: MCU 停止工作，仍然有振荡时钟，部分外围模块可以工作。

1: 休眠模式: 所有时钟停止，功耗最低。

clk_sel[1:0]: 系统时钟总选择位:

clk_sel[1]	时钟选择
0	根据 clk_sel[0]选择时钟
1	RC32K/2
clk_sel[0]	时钟选择
0	HOSC
1	根据寄存器 ircf2-0 选择

在做时钟切换时，必须:

- 先选择 ircf2-0，再切换 clk_sel[1]。
- 每 2 个时钟频率间切换，必须以默认的 RC32K/2 作为衔接，即先从时钟 A 切换到默认的 RC32K/2，再从默认的 RC32K/2 切换到时钟 B。
- 若要关闭某时钟，必须等系统时钟切换完毕后可以关闭。
- 若要新使能一个时钟并作为系统时钟，在使能该时钟后必须等待一段时间方可做切换。

5.2.2 CPU 外围模块时钟源

SD3004 外围模块的时钟源有不同的控制寄存器与分频器进行配置，该配置将在各外围模块的说明部分详细说明，因此现在只附上外围模块工作时钟配置图。如图 5-2。

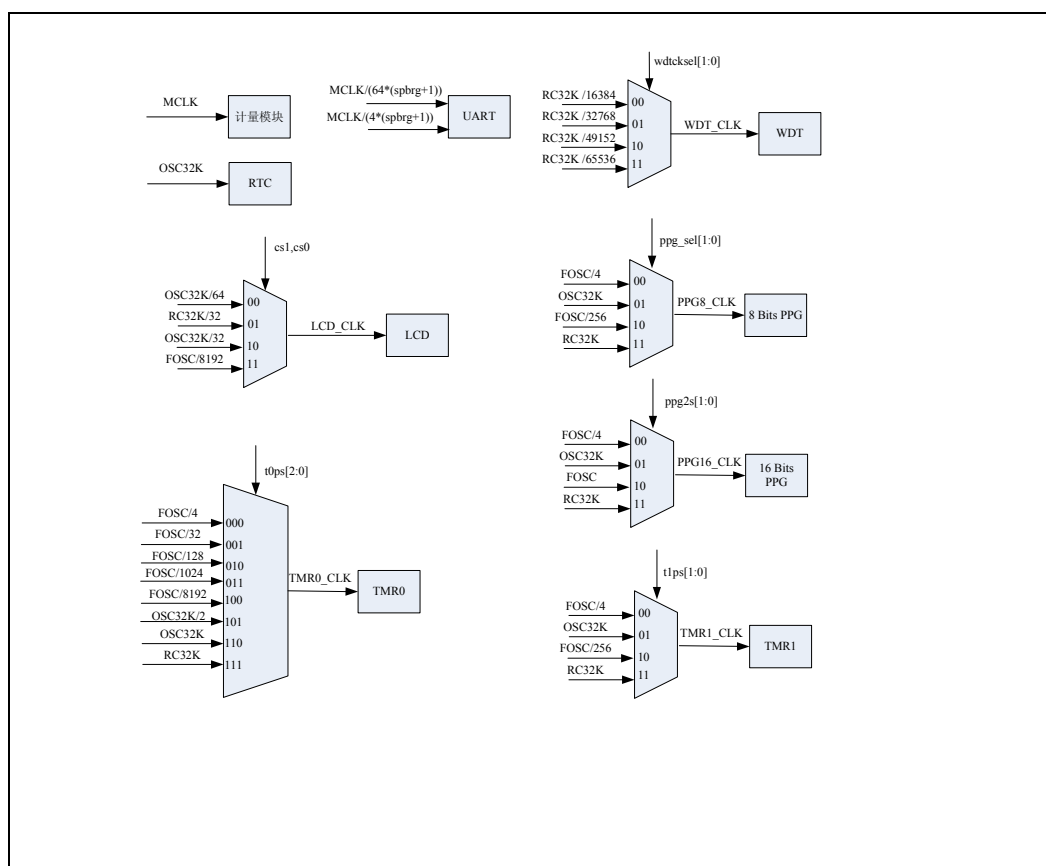


图 5-2 各外围模块时钟源

5.3 时钟相关寄存器

时钟使能控制寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCTUNE (F9BH)	rcoscen	cryoscen	osc_en	a2	a1	a0	pllen	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	--
Reset	1	1	1	1	1	1	1	--

rcoscen: 内部 32KHz RC 振荡时钟使能位，高有效

cryoscen: 高频晶振时钟使能位, 高有效

osc_en: OSC32K 时钟使能位, 高有效

a2-a0: OSC32K 时钟电流能力选择, 全“1”时电流最大。使用的时候请选择默认值。

a2-a0	电流 (nA)
000	25
001	50
010	100
011	200
100	400
101	800
110	15
111	1600

pllen: PLL 时钟使能信号, 高有效

WDT 控制寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCON (FD1H)	--	--	--	--	--	wdtsel[1]	wdtsel[0]	wdten
R/W	--	--	--	--	--	R/W	R/W	R/W
Reset	--	--	--	--	--	0	0	0

wdten: WDT 使能信号, 高电平使能

wdtsel[1:0] : WDT 时间选择寄存器

wdtsel [1:0]	溢出频率
00	RC32K /16384
01	RC32K /32768
10	RC32K /49152
11	RC32K /65536

6 复位

6.1 概述

SD3004 的复位电路包含以下五种事件来触发复位信号, 包括低压检测复位 BOR 和外部引脚复位 EXT_RST, 看门狗复位 WDT, 堆栈错误复位 STK_RST 和软件复位 SOFT_RST。

复位事件如下, 复位关系见图 4-1:

- BOR 低压复位
- EXT_RST 外部复位
- WDT 看门狗复位
- SOFT_RST 软件复位
- STK_RST 堆栈错误复位

相关操作状态寄存器:

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	un_ins	pd	to	--	--	zero	dc	carry
STKPTR	stkful	stkunf	--	--	--	stkptr_reg[2:0]		

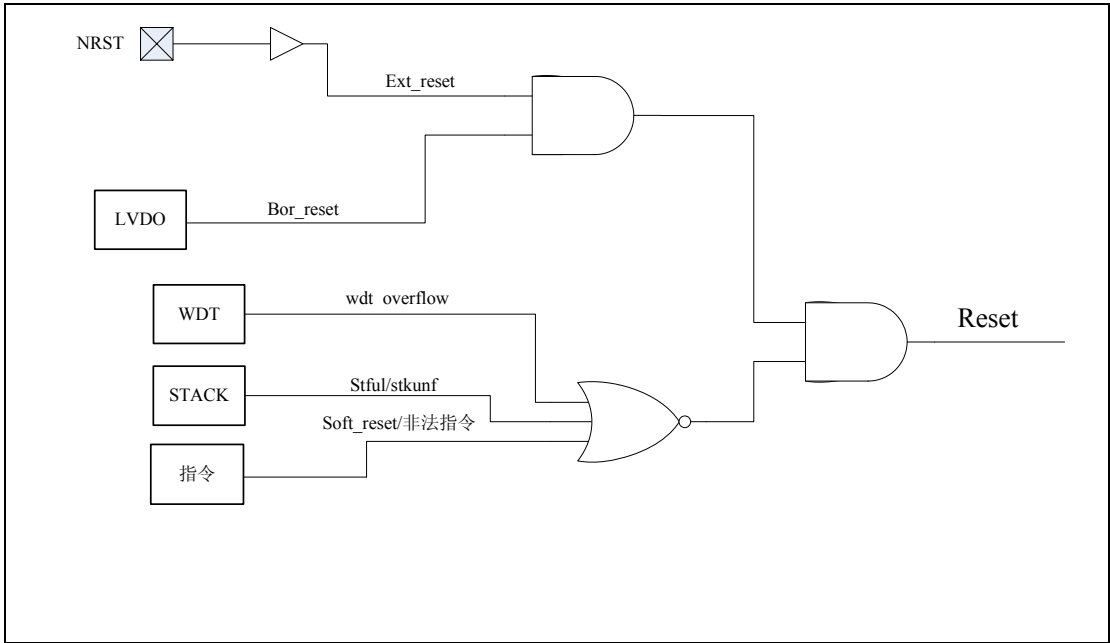


图6-1 复位关系示意图

表 6-1 复位等级表

复位种类	事件	说明
异步复位	EXT_RST	CPU 重新启动，须等待内部低频振荡启动计数器计数完成后才能进入正常工作状态。
	BOR	
内部复位	WDT	复位寄存器，CPU 快速回到正常工作状态。
	STK	
	SOFT_RST	

6.1.1 外部复位

外部复位指用户通过 VPP/RST 功能复用引脚输入低电平对芯片进行复位。外部复位的低电平时间必须超过 10us 才有效。

复位引脚输入具有迟滞，高低电平转换点分别为 0.2VDD 与 0.8VDD。即当复位电压上升到 0.8VDD 的时候才退出复位状态，当复位电压下降到 0.2VDD 才进入复位状态。

6.1.2 低电压检测复位（LVD）

芯片内部有低压检测电路 LVD，当芯片工作电压因干扰和或者电源自身电压下降到低于一定阈值时使芯片处于复位状态，以防止 MCU 死机或外围电路工作异常。直到电源电压恢复到正常范围内才解除复位，使芯片进入正常工作模式。

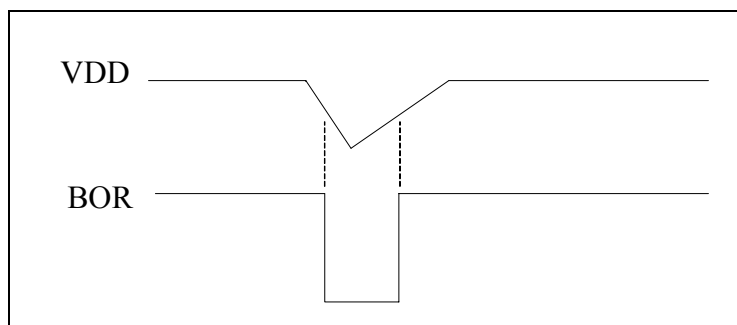


图6-3 低压检测复位过程

具有 4.0V/2.2V/2.0V 三路低压检测功能:

4.0V 电压检测用于对 AVDD 电压的检测，AVDD 电压低于 4.0V 会发出中断，MCU 会根据该中断信号保存数据，关计量部分 A/D；

2.2V 电压检测用于对 VDD 电压的检测，检测到 VDD 电压低到 2.2V 时，会置位相应标志位；

2.0V 电压检测用于对 VDD 电压的检测，检测到 VDD 电压低到 2.0V 时复位 MCU。

低压检测控制寄存器：

寄存器	Bit7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
LVDCON	FEN	--	--	LVDRESET_EN	LVD_4V	LVD_2O2V	--	LVD2O2V_EN
default	0	--	--	1			--	0

- LVD_4V: 4.0V 低压检测标志位，“0”: AVDD 电压低于 4.0V；
- LVD_2O2V: 2.2V 低压检测标志位，“0”: DVDD 电压低于 2.2V；
- LVD2O2V_EN: 2.2V 低压检测使能位，“1”工作，默认低电平；
- LVDRESET_EN: 低压检测复位使能，“1”工作；系统电源电压跌至 2.0V，系统复位，默认是高电平；
- FEN: F1, F2 的输出使能，为高电平时，PORT53, PORT52 端口的输出信号为 F1, F2。

应用注意：由于 LVD 需要消耗一些功耗，进入待机模式时，建议关闭。

6.1.3 WDT 看门狗计数器复位

WDT 看门狗计数器在正常工作模式下，当程序跑飞导致没有定时对 WDT 清零时，WDT 计数会溢出产生复位信号，使芯片进入复位状态。当发生 WDT 看门狗复位时，STATUS 寄存器的 to 标志位会置“1”。

6.1.4 STK 堆栈错误复位

程序发生堆栈满溢出或空溢出时，会产生复位信号使芯片复位。当发生堆栈错误复位时，STKPTR 寄存器的 stkful 或 stkunf 会置“1”。

6.1.5 指令复位

用户操作指令执行 RESET 指令时，会产生复位信号使芯片进入快速启动状态。

6.2 状态寄存器

寄存器说明与复位状态:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS(FD8H)	un_ins	pd	to	--	--	zero	dc	carry
R/W	R	R	R	--	--	R	R	R
Reset	0	0	0	--	--	U	U	U

un_ins: 非法指令标志位

“1”发生过非法指令

“0”未发生过非法指令

pd: 待机标志位

“1” 执行 sleep 指令

“0” 上电、执行 clrwtd 指令

to: 看门狗超时标志位

“1” 发生 WDT 超时溢出

“0” 上电、执行 sleep 或 clrwtd 指令

zero: 零位

“1” 算术运算或逻辑运算结果为零

“0”算术运算或逻辑运算结果不为零

dc: 辅助进位/ 借位标志位

“1” 结果的第4 个低位发生了进位

“0”结果的第4 个低位未发生进位

carry: 进位/ 借位标志位

“1”结果中最高位发生了进位

“0”结果中最高位未发生进位

复位状态跟复位源的相互关系见下表:

6.3 数据寄存器复位值

表 6-2 复位状态寄存器关系表

寄存器	地址	复位	寄存器	地址	复位
LDCOM	F58H	---- 0000	OSCTUNE	F9BH	1111 111-
LDCON	F59H	0--0 00--	PIR	F9EH	--00 0-00
LCDSE0	F5AH	0000 0000	PIE2	FA0H	--00 0000
LCDSE1	F5BH	0000 0000	PIR2	FA1H	--00 0000
LCDSE2	F5CH	0000 0000	VOL_RESULTU	FA9H	---- xxxx
LCDDATA0	F60H	0000 0000	CURRENT_RESULTU	FAAH	---- xxxx
LCDDATA1	F61H	0000 0000	COUNTER_CON	FABH	---0 0000
LCDDATA2	F62H	0000 0000	COUNT_UP_H	FACH	0000 0000

TMR1U	F63H	0000 0000	COUNT_UP_L	FADH	0000 0000
RD_TMR1U	F64H	0000 0000	COUNT_DOWN_H	FAEH	0000 0000
CCPR1U	F65H	0000 0000	COUNT_DOWN_L	FAFH	0000 0000
LCDDATA3	F66H	0000 0000	RD_TMR0L	FB1H	0000 0000
LCDDATA4	F67H	0000 0000	RD_TMR1H	FB2H	0000 0000
LCDDATA5	F68H	0000 0000	RD_TMR1L	FB3H	0000 0000
PPG2H_H	F69H	0000 0000	CCPR1L	FBEH	0000 0000
PPG2H_L	F6AH	0000 0000	CCPR1H	FBFH	0000 0000
SSPAD	F6BH	0000 0000	ADCON	FC0H	0100 1111
SSPBUF	F6CH	xxxx xxxx	ADC_CON2	FC1H	1111 1110
SSPCON1	F6DH	0001 0000	RLT_ADJ0	FC2H	0000 0000
SSPCON2	F6EH	0000 0000	RLT_ADJ1	FC3H	0000 0000
SSPSTAT	F6FH	--00 0000	RLT_ADJ2	FC4H	0000 0000
PPG2CON	F70H	0--- --00	RLT_ADJ3	FC5H	0000 0000
LCDDATA6	F71H	0000 0000	RLT_ADJ4	FC6H	0000 0000
LCDDATA7	F72H	0000 0000	ADD_COUNT0	FC7H	0010 0010
LCDDATA8	F73H	0000 0000	ADD_COUNT1	FC8H	---- --00
PPG_CON	F74H	0--- --00	VOL_RESULTH	FC9H	xxxx xxxx
PPG_SETH	F75H	0000 0000	VOL_RESULTL	FCAH	xxxx xxxx
PPG_SETP	F76H	0000 0000	CURRENT_RESULTH	FCBH	xxxx xxxx
LCDDATA9	F77H	0000 0000	CURRENT_RESULTL	FCCH	xxxx xxxx
LCDDATA10	F78H	0000 0000	T1CON	FCDH	0000 --00
LCDDATA11	F79H	0000 0000	TMR1L	FCEH	0000 0000
RCSTA	F7BH	0000 -00x	TMR1H	FCFH	0000 0000
RCREG	F7CH	xxxx xxxx	WDTCON	FD1H	---- -000
TXSTA	F7DH	0000 --10	LVDCON	FD2H	0—1 0000
TXREG	F7EH	xxxx xxxx	OSCCON	FD3H	1111 111-
SPBRG	F7FH	xxxx xxxx	T0CON	FD5H	0--- -000
PT1	F80H	xxxx xxxx	TMR0L	FD6H	0000 0000
PT2	F81H	xxxx xxxx	STATUS	FD8H	000- -xxx
PT3	F82H	xxxx xxxx	BSR	FE0H	---- 0000
PT4	F83H	xxxx xxxx	WREG	FE8H	xxxx xxxx
PT5	F84H	xxxx xx--	FSR0L	FE9H	0000 0000
PT1PU	F85H	1111 1111	FSR0H	FEAH	---- 0000
PPG2P_H	F86H	0000 0000	INDF0	FEFH	xxxx xxxx
PPG2P_L	F87H	0000 0000	INTCON2	FF1H	---- --11
PT1EN	F89H	0000 0000	INTCON	FF2H	0-00 0-00
PT2EN	F8AH	0000 0000	TABLAT	FF5H	0000 0000
PT3EN	F8BH	0000 0000	TBLPTRL	FF6H	0000 0000
PT4EN	F8CH	0000 0000	TBLPTRH	FF7H	---- 0000
PT5EN	F8DH	0000 00--	PCL	FF9H	0000 0000
LAT1	F92H	0000 0000	PCLATH	FFAH	---- 0000
LAT2	F93H	0000 0000	STKPTR	FFCH	00-- -000

LAT3	F94H	0000 0000	TOSL	FFDH	0000 0000
LAT4	F95H	0000 0000	TOSH	FFEh	---- 0000
LAT5	F96H	0000 00--			

7 中断

7.1 概述

本芯片有多个中断源，一个中断优先级，由中断使能控制寄存器 INTCON/PIE 与中断事件标志位寄存器 PIR 组成。中断发生时产生中断事件，全局中断使能和对应中断使能为有效时，响应中断：将当前程序计数器 PC 值压栈，PC 跳至程序存储器的中断向量 0008H 执行中断程序。

中断相关寄存器：

表 7-1 中断寄存器列表

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR	--	--	sspif	timer0if	secif	--	int1if	int0if
PIE2	--	--	rcie	txie	bclie	lvdie	tmr1ie	ccpie
PIR2	--	--	rcif	txif	bclif	lvdif	tmr1if	ccpif
INTCON2	--	--	--	--	--	--	intedge1	intedge0
INTCON	gie	--	sspie	tmr0ie	secie	--	int1ie	int0ie

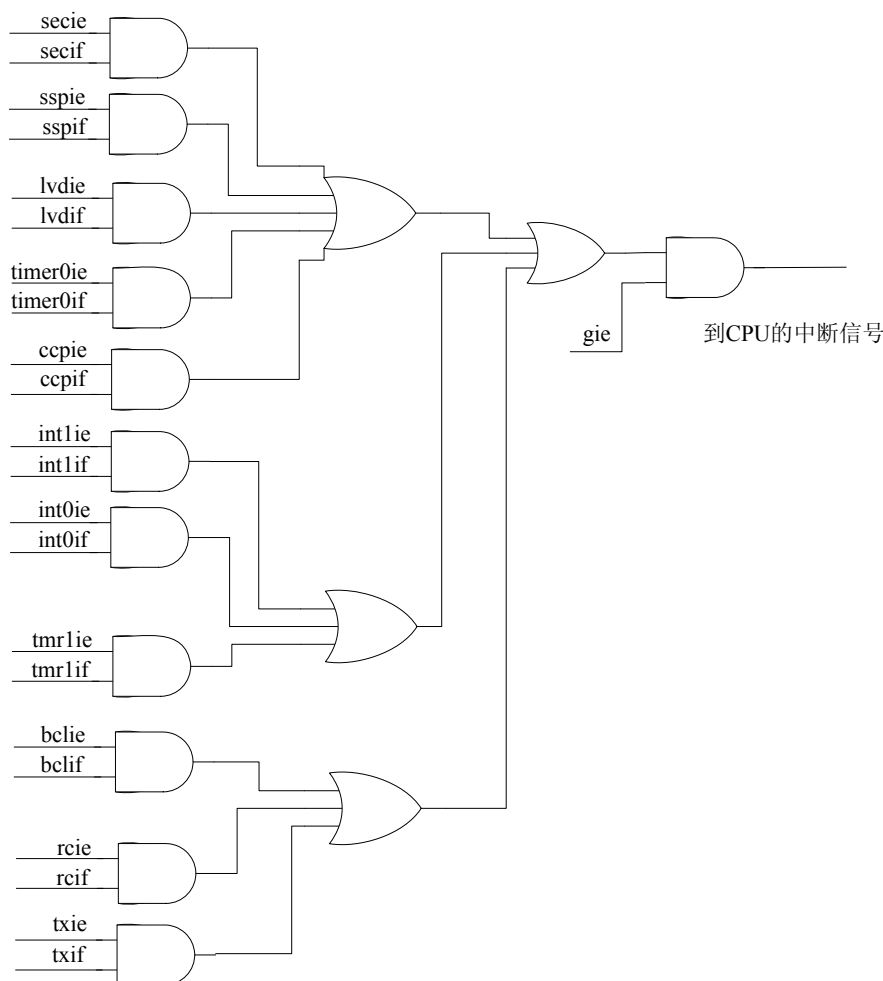


图 7-1 中断向量图

7.2 中断说明

中断的产生需要设置全局中断使能 **gie** 和每个中断事件的各自使能位。使能中断事件只需要将对应于中断事件的使能寄存器 **INTCON/PIEx** 相应位置“1”；反之，设置零关闭中断事件。中断产生时，程序计数器跳转到地址 **0008H** 处运行。在中断服务程序里，可以通过查询中断标志位确认中断源，中断返回前用软件清中断标志位。

当进入中断服务向量时，**gie** 会自动清零禁止其他中断进入，当中断服务程序执行完返回到中断入口时，**gie** 自动置 1，允许新的中断产生。

中断的现场保护：中断期间，PC 的返回地址被保存到堆栈中，**WREG**, **BSR** 和 **STARTUS** 寄存器会自动备份到它们的备份寄存器中，如果有其他需要保存的寄存器，在中断程序里做保护，如果不是有快速返回功能，**WREG**, **BSR** 和 **STARTUS** 寄存器也需要保护。

用户程序举例：

Hint:

```

btfss    PIR2,TMR1IF,0      ;判断是否是 TMER1 中断
bra      IntReturn
bcf      PIR2,TMR1IF,0      ;清中断标志位
...                               ;现场保护（可选）及其它代码

```

HintReturn:

retfie 1 ;中断退出

7.3 寄存器说明

中断使能寄存器 1

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCON(FF2H)	gie	--	sspie	tmr0ie	secie	--	int1ie	int0ie
R/W	R/W	--	R/W	R/W	R/W	--	R/W	R/W
Reset	0	--	0	0	0	--	0	0

gie: 全局中断使能位, 高有效

sspie: I2C 中断使能位, 高电平有效

tmr0ie: 定时器 0 中断使能位, 高有效

secie: 0.5 秒中断使能位, 高电平有效

int1ie: 外部中断 1 使能位, 高有效

int0ie: 外部中断 0 使能位, 高有效

外部中断0/1沿选择寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCON2(FF1H)	--	--	--	--	--	--	intedg1	intedg0
R/W	--	--	--	--	--	--	R/W	R/W
Reset	--	--	--	--	--	--	1	1

intedge1: 外部中断 1 边沿选择位

“1”上升沿触发中断

“0”下降沿触发中断

intedge0: 外部中断 0 边沿选择位

“1”上升沿触发中断

“0”下降沿触发中断

注: 在中断里切换沿选择时操作顺序: 进入中断后先清中断标志位, 再关相应的中断使能位, 再切换沿选择, 然后再清一次中断标志位, 开中断使能位, 退出中断。

中断标志位寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR(F9EH)	--	--	sspif	timer0if	secif	--	int1if	int0if
R/W	--	--	R/W	R/W	R/W	--	R/W	R/W
Reset	--	--	0	0	0	--	0	0

sspif: I2C 中断标志位, “1”发生中断, 软件清零

tmr0if: 定时器 0 溢出中断标志位, “1”发生中断, 软件清零

secif: RTC 半秒中断标志位, “1”发生中断, 软件清零

Int1if: 外部中断 1 标志位, “1”发生中断, 软件清零

Int0if: 外部中断 0 标志位, “1”发生中断, 软件清零

中断使能寄存器2

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIE2(FA0H)	--	--	rcie	txie	bclie	lvdie	tmrlie	ccpie
R/W	--	--	R/W	R/W	R/W	R/W	R/W	R/W
Reset	--	--	0	0	0	0	0	0

rcie: UART 接收中断使能位, 高有效

txie: UART 发送中断使能位, 高有效

bclie: I2C 多主机模式总线冲突中断使能位, 高有效

lvdie: 4.0V 低压检测中断使能位, 高有效

tmrlie: 定时器 1 溢出中断使能位, 高有效

ccpie: CAPTURE 中断使能位, 高有效

中断标志位寄存器2:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR2(FA1H)	--	--	rcif	txif	bclif	lvdif	tmrlif	ccpif
R/W	--	--	R	R	R/W	R/W	R/W	R/W
Reset	--	--	0	0	0	0	0	0

rcif: UART 接收中断标志位, “1”发生中断, 软件清零

txif: UART 发送中断标志位, “1”发生中断, 软件清零

bclif: I2C 多主机模式总线冲突中断标志位, “1”发生中断, 软件清零

lvdif: 4.0V 低压检测中断标志位, “1”发生中断, 软件清零

tmrlif: 定时器 1 计数溢出标志位, “1”发生中断, 软件清零

ccpif: CAPTURE 中断标志位, “1”发生中断, 软件清零

8 I/O 口

8.1 概述

SD3004 有 5 组输入/输出端口 I/O, 在程序初始化中应该先对 I/O 进行定义, 可作为数字的输入和输出以及功能模块的管脚。每组 I/O 由一组寄存器控制。当用作外围模块的功能管脚时, 不能做输入/输出管脚用。

以下主要描述 I/O 的标准数字输入输出 I/O 控制, 功能复用请参考相应功能模块章节。

SD3004 主要有以下 I/O:

I/O 组	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORT1	P17	P16	P15	P14	P13	P12	P11	P10
PORT2	P27	P26	P25	P24	P23	P22	P21	P20
PORT3	P37	P36	P35	P34	P33	P32	P31	P30
PORT4	P47	P46	P45	P44	P43	P42	P41	P40
PORT5	P57	P56	P55	P54	P53	P52	--	--

I/O相关寄存器:

表 8-1 I/O 寄存器列表

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

PT1	读 PORT1 引脚，写 PORT1 数据锁存器							
PT2	读 PORT2 引脚，写 PORT2 数据锁存器							
PT3	读 PORT3 引脚，写 PORT3 数据锁存器							
PT4	读 PORT4 引脚，写 PORT4 数据锁存器							
PT5	读 PORT5 引脚，写 PORT5 数据锁存器						--	--
PT1PU	PORT1 端口上拉使能寄存器							
PT1EN	PORT1 端口方向设置寄存器							
PT2EN	PORT2 端口方向设置寄存器							
PT3EN	PORT3 端口方向设置寄存器							
PT4EN	PORT4 端口方向设置寄存器							
PT5EN	PORT5 端口方向设置寄存器						--	--
LAT1	读 PORT1 数据寄存器，写 PORT1 数据锁存器							
LAT2	读 PORT2 数据寄存器，写 PORT2 数据锁存器							
LAT3	读 PORT3 数据寄存器，写 PORT3 数据锁存器							
LAT4	读 PORT4 数据寄存器，写 PORT4 数据锁存器							
LAT5	读 PORT5 数据寄存器，写 PORT5 数据锁存器							
INTCON2	--	--	--	--	--	--	intedg1	intedg0

8.2 PORT 相关寄存器说明

1) PT 寄存器控制

当 I/O 被设为输入状态，读 PT 可以读到当前 I/O 的状态，“1”表示输入为高电平，“0”表示输入为低电平。

当 I/O 被设为输出状态，写 PT 可以设置 I/O 的输出状态，“1”表示设置输出高电平，“0”表示设置输出低电平。在输出状态，写 PT 与写 LAT 寄存器效果相同。

2) PTxEN 输入/输出控制寄存器

PT1EN、PT2EN、PT3EN、PT4EN、PT5EN 都是用来设置 I/O 的输入输出方向。

PT1EN、PT2EN、PT3EN、PT4EN、PT5EN 中的这些位被设置“1”时对应 I/O 为输出状态，设置“0”时对应 I/O 为输入状态。默认为“0”，I/O 为输入。

当某个 I/O 设定为输入状态后，应当开启上拉或在外部输入明确高低电平等，不可让 I/O 出现悬浮状态。否则可能产生输入状态不明确以及额外的漏电电流。

3) LAT 输出数据锁存寄存器

写 LAT 和写 PT 寄存器的作用是一样的，都是对输出数据寄存器进行设置，如果输出使能，该寄存器的值会直接输出到端口上去，读 LAT 寄存器是读输出数据寄存器

4) PT1PU 上拉电阻控制寄存器

PT1PU 寄存器中的各位用来设置对应 I/O 中的上拉电阻功能是否开启。

0——开启上拉电阻

1——关闭上拉电阻（默认）

在芯片进入休眠模式之前，若 I/O 设置为输入状态且外部电路连接方式会造成 I/O 有悬空

现象时可启用上拉电阻，以避免 I/O 悬空而导致芯片进入休眠模式后产生漏电流。

5) INTCON2 中断信号产生条件

此寄存器中 intedg1、intedg0 用于设置两路外部中断 INT1/INT0 发生何种变化时产生中断信号，电平变化条件可分为上升沿（0→1）变化、下降沿（1→0）变化。

- Intedg1：外部中断 1 边沿选择位
- “1”上升沿触发中断
- “0”下降沿触发中断
- Intedg0：外部中断 0 边沿选择位
- “1”上升沿触发中断
- “0”下降沿触发中断

8.3 输入/输出口 1，PORT1

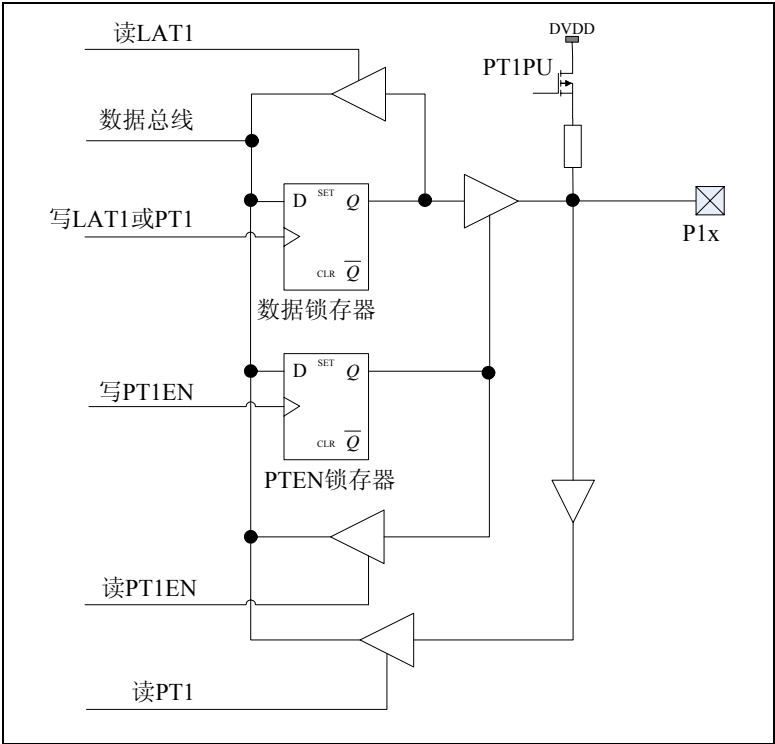


图 8-1 PORT1 逻辑框图

PORT1 功能说明

PORT1 为施密特输入 I/O，带上拉选择。 P11 复用 INT1，中断沿可选；也跟 OTP 烧录的数据输入端复用；P12 复用 I2C 时钟端 SCL；P13 复用 I2C 数据端 SDA；P14 复用 USART 的数据发送 TXD/CK；P15 复用 USART 的数据接收 RXD/DT；P16 复用 8 位的 PPG 输出；P17 复用 16 位的 PPG 输出。

PORT1上拉设置寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT1PU(012H)	pu[7]	pu[6]	pu[5]	pu[4]	pu[3]	pu[2]	pu[1]	pu[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset	1	1	1	1	1	1	1	1
-------	---	---	---	---	---	---	---	---

“0”：上拉电阻启用

“1”：上拉电阻禁止

PORT1端口寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT1(F80H)	PT17	PT16	PT15	PT14	PT13	PT12	PT11	PT10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

读：读 PT1 端口值；写：写 PORT1 的输出数据寄存器 LAT1

1：读/写高电平

0：读/写低电平

PORT1输入/输出控制寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT1EN(F89H)	PT1EN[7]	PT1EN[6]	PT1EN[5]	PT1EN[4]	PT1EN[3]	PT1EN[2]	PT1EN[1]	PT1EN[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1：设置端口为输出口

0：设置端口为输入口

PORT1输出数据寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LAT1(F92H)	LAT1[7]	LAT1[6]	LAT1[5]	LAT1[4]	LAT1[3]	LAT1[2]	LAT1[1]	LAT1[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1：读/写高电平

0：读/写低电平

8.4 输入/输出口 2, PORT2

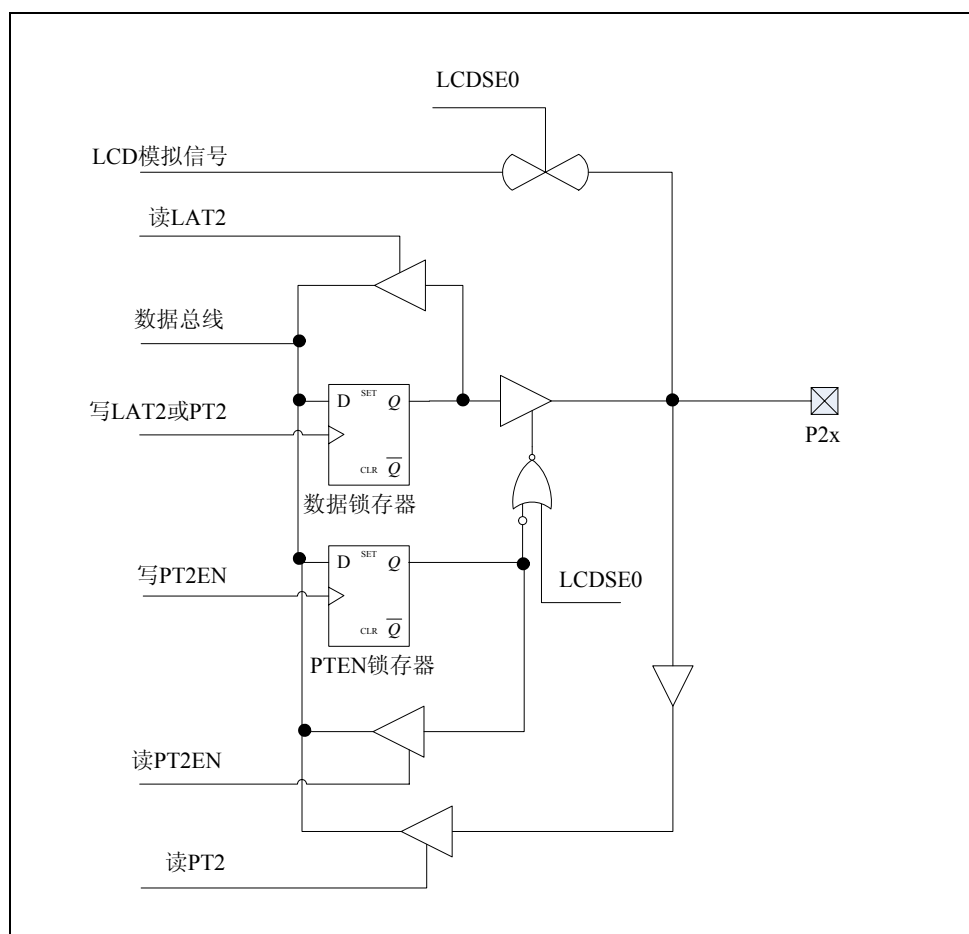


图 8-2 PORT2 逻辑框图

PORT2 功能说明

P27~P20 跟液晶的 SEG7~SEG0 复用

寄存器说明：

PORT2端口寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT2(F80H)	PT27	PT26	PT25	PT24	PT23	PT22	PT21	PT20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

读：读 PT2 端口值；写：写 PORT2 的输出数据寄存器 LAT2

1：读/写高电平。

0：读/写低电平

PORT2输入/输出控制寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT2EN(F89H)	PT2EN[7]	PT2EN[6]	PT2EN[5]	PT2EN[4]	PT2EN[3]	PT2EN[2]	PT2EN[1]	PT2EN[0]

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1: 设置端口为输出口。

0: 设置端口为输入口。

PORT2输出数据寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LAT2(F92H)	LAT2[7]	LAT2[6]	LAT2[5]	LAT2[4]	LAT2[3]	LAT2[2]	LAT2[1]	LAT2[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1: 读/写高电平。

0: 读/写低电平。

8.5 输入/输出端口 3, PORT3

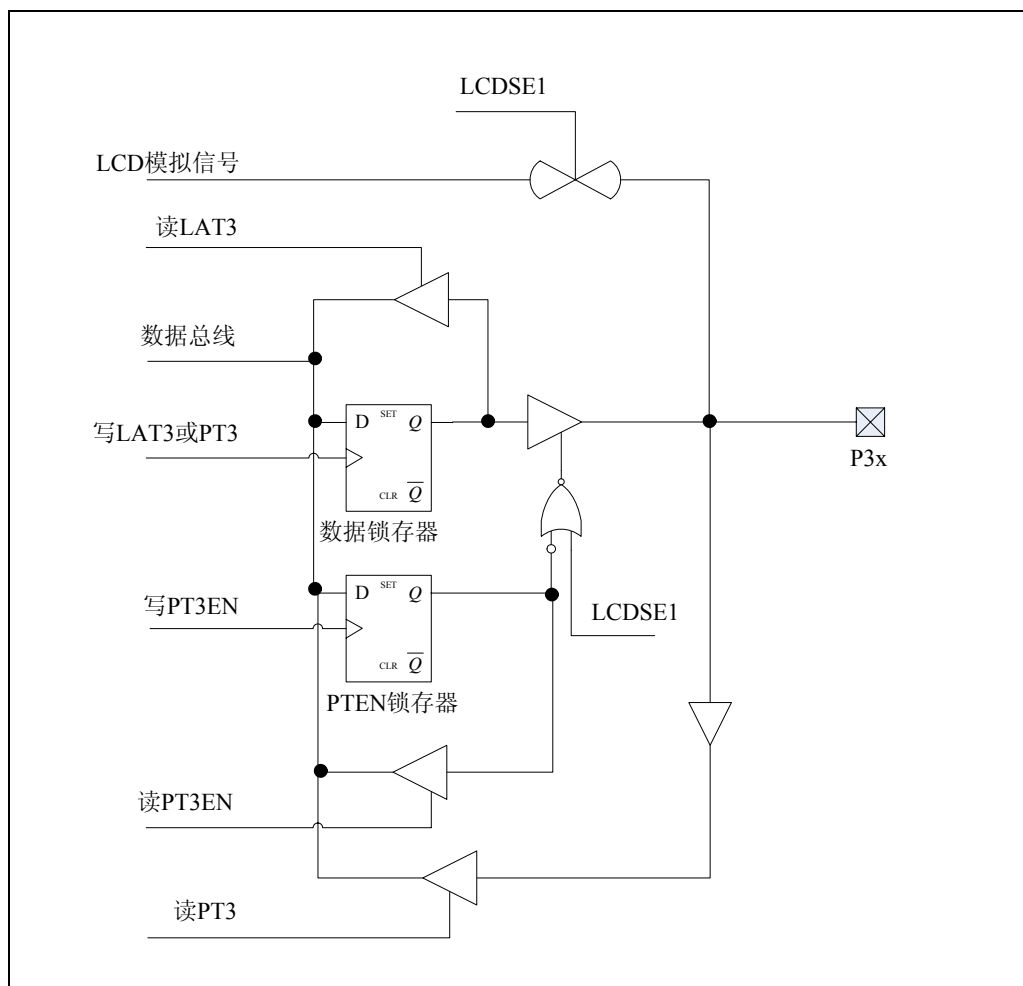


图 8-3 PORT3 逻辑框图

PORT3 功能说明

P37~P30 跟液晶的 SEG15~SEG8 复用

PORT3端口寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT3(F81H)	PT37	PT36	PT35	PT34	PT33	PT32	PT31	PT30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

读：读 PORT3 端口值；写：写 PORT3 的输出寄存器 LAT3

1：读/写高电平。

0：读/写低电平

PORT3输入/输出控制寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT3EN(F8AH)	PT3EN[7]	PT3EN[6]	PT3EN[5]	PT3EN[4]	PT3EN[3]	PT3EN[2]	PT3EN[1]	PT3EN[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1：设置端口为输出口。

0：设置端口为输入口。

PORT3输出数据锁存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LAT3(F93H)	LAT3[7]	LAT3[6]	LAT3[5]	LAT3[4]	LAT3[3]	LAT3[2]	LAT3[1]	LAT3[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1：读/写高电平。

0：读/写低电平。

8.6 输入/输出端口 4, PORT4

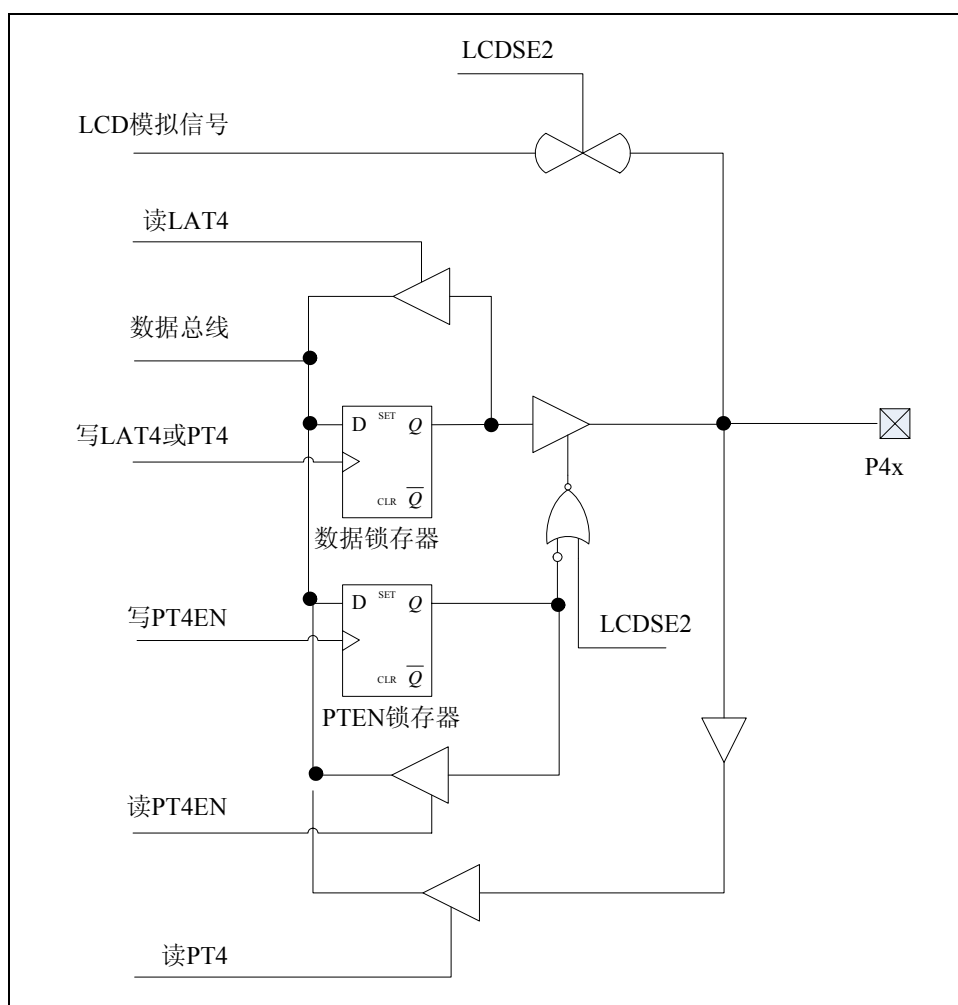


图8-4 PORT4 逻辑框图

PORT4 功能说明

P47~P40 跟 SEG23~SEG16 复用

PORT4端口寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT4(F82H)	PT47	PT46	PT45	PT44	PT43	PT42	PT41	PT40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

读：读 PT4 端口值；写：写 PORT4 的输出寄存器 LAT4。

1：读/写高电平

0：读/写低电平

PORT4输入/输出控制寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT4EN(F8BH)	PT4EN[7]	PT4EN[6]	PT4EN[5]	PT4EN[4]	PT4EN[3]	PT4EN[2]	PT4EN[1]	PT4EN[0]

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1: 设置端口为输出口。

0: 设置端口为输入口。

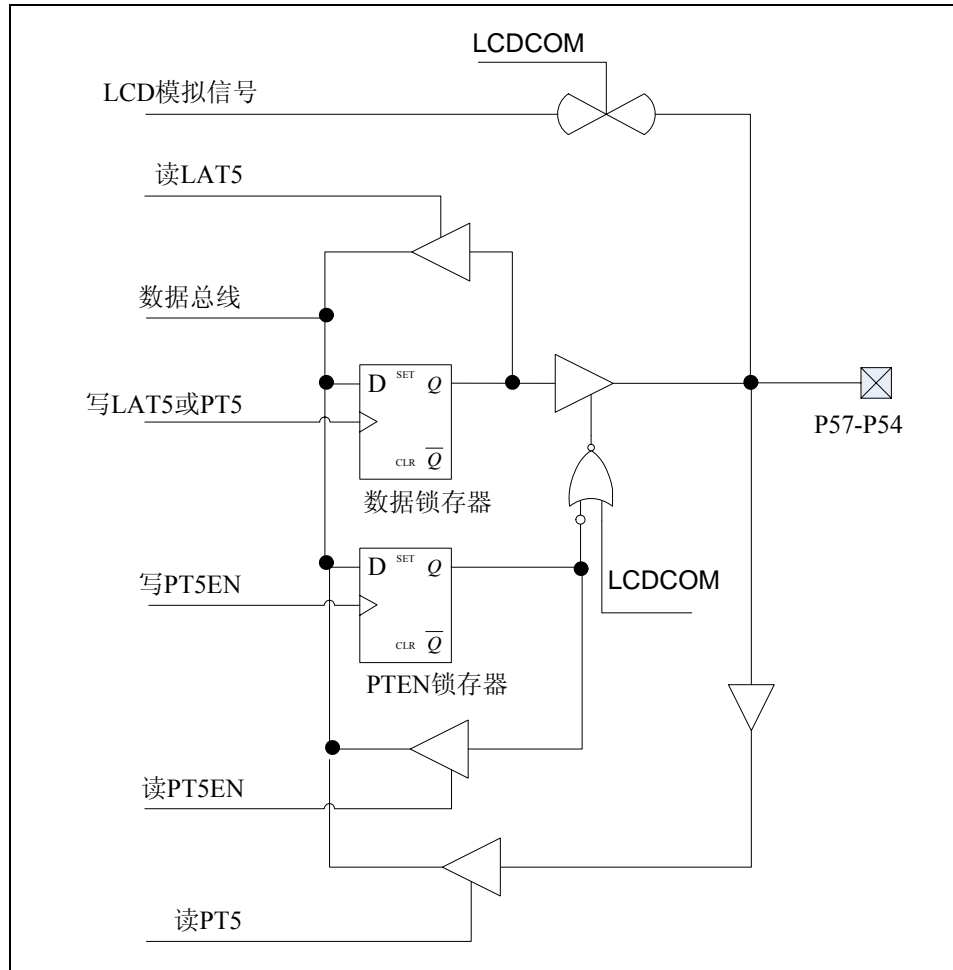
PORT4输出数据锁存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LAT4(F94H)	LAT4[7]	LAT4[6]	LAT4[5]	LAT4[4]	LAT4[3]	LAT4[2]	LAT4[1]	LAT4[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

1: 读/写高电平。

0: 读/写低电平。

8.7 输入输出端口 5, PORT5



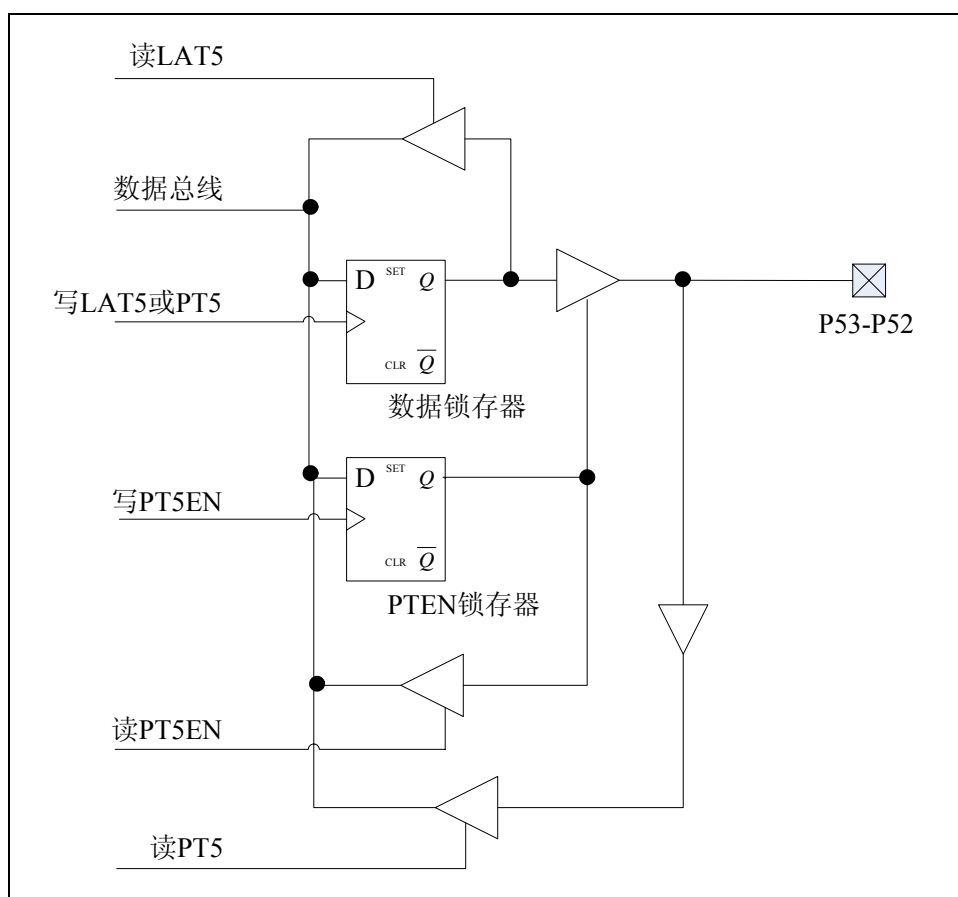


图8-5 PORT5 逻辑框图

PORT5 功能说明

P57~P54 跟液晶的 COM3~COM0 复用；P53~P52 跟驱动电解的 F1/F2 复用

PORT5端口寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT5(F83H)	PT57	PT56	PT55	PT54	PT53	PT52	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	--	--
Reset	x	x	x	x	x	x	--	--

读：读 PT5 端口值；写：写 PORT5 的输出寄存器。

1：读/写高电平

0：读/写低电平

PORT5输入/输出控制寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PT5EN(F8CH)	PT5EN[7]	PT5EN[6]	PT5EN[5]	PT5EN[4]	PT5EN[3]	PT5EN[2]	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	--	--
Reset	0	0	0	0	0	0	--	--

1：设置端口为输出口。

0：设置端口为输入口。

PORT5输出数据锁存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LAT5(F95H)	LAT5[7]	LAT5[6]	LAT5[5]	LAT5[4]	LAT5[3]	LAT5[2]	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	--	--
Reset	0	0	0	0	0	0	--	--

1: 读/写高电平。

0: 读/写低电平。

9 看门狗 WDT (Watch dog)

9.1 WDT 概述

SD3004 中的 WDT 包括一个 15Bits 计数器。运行模式下产生的信号用来复位，而在待机模式下不工作。

9.2 WDT 使用说明

WDT 时钟源采用内部低频 RC 振荡的二分频，寄存器 wdtssel[1:0]可选择 WDT 计数器的溢出时间，计数器溢出后可产生 WDT 复位信号 to。看门狗的溢出时间可以设置的范围从 0.5 秒到 2 秒。

9.3 寄存器说明-WDT

WDT控制寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTCON (FD1H)	--	--	--	--	--	wdtsel[1]	wdtsel[0]	wdten
R/W	--	--	--	--	--	R/W	R/W	R/W
Reset	--	--	--	--	--	0	0	0

wdten: WDT 的工作状态使能控制位，高电平使能。

wdtsel[1:0]: WDT 溢出频率选择。

wdtsel[1:0]	溢出频率
00	RC32K /16384
01	RC32K /32768
10	RC32K /49152
11	RC32K /65536

10 计时/定时器

10.1 概述

有两个计数器/定时器：TM0和TM1。

10.2 定时器 0，TMR0

10.2.1 概述

定时器 0 为 8-bit 的设计结构，TMR0 可工作于工作模式和待机模式。8 位定时器/计数器

0 是递增型计数，溢出产生中断，计数器值可读，计数起始值可以通过寄存器 TMR0L 设置，计数时钟可以通过寄存器选择晶振时钟、内部低频 RC 振荡时钟、内部 MCU 工作时钟或其分频，可工作于正常工作模式和待机模式。框图如 10-1

TMR0相关寄存器:

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0CON	tmr0_en	--	--	--	--	t0ps[2]	t0ps[1]	t0ps[0]
TMR0L	定时器 0 计数预设值寄存器							
RD_TMR0L	读定时器 0 当前计数值							
PIR	--	--	sspif	timer0if	secif	--	int1if	int0if
INTCON	gie	--	sspie	tmr0ie	secie	--	int1ie	int0ie

阴影单元不使用

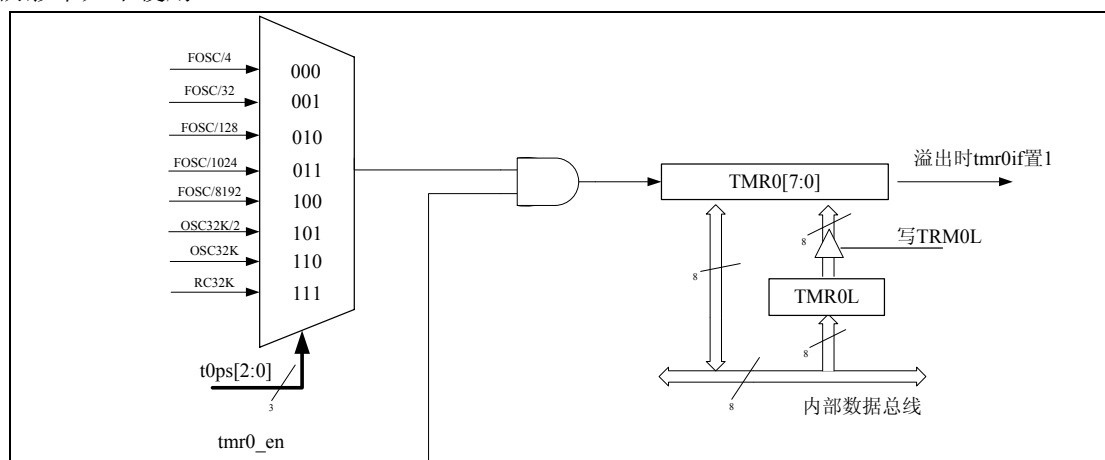


图 10-1 定时器 0 框图

10.2.2 定时器 0 使用说明

首先对定时器时钟源进行选择，由 t0ps[2:0] 进行控制。无论选择哪种时钟，在待机模式下都可以正常工作。

设置寄存器 TMR0L，如果需要产生中断，将中断总使能位 gie 和定时器 0 中断位 tmr0ie 置 1，将定时器 0 的使能位 tmr0_en 置 1，定时器 0 从 TMR0L 的值开始递增计数，定时器 0 计数到 FFH 后溢出产生中断，tmr0if 置 1。计数器重新回到 TMR0L 设定值开始计数。读 RD_TMR0L 寄存器可以读到定时器 0 的当前值。

用户程序举例：

```

movlw    0x02                ;选择定时器 0 时钟源
movwf    T0CON,0
movlw    0x3C                ;设置定时器 0 溢出时间
movwf    TMR0L,0
bsf      T0CON,TMR0EN,0      ;开启定时器 0
bcf      PIR1,TMR0IF,0       ;清定时器 0 中断标志，使能定时器 0 中断（可选）
bsf      INTCON,TMR0IE,0

```

10.2.3 寄存器说明-TMR0

定时器0控制寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0CON (FD5H)	tmr0_en	--	--	--	--	t0ps[2]	t0ps[1]	t0ps[0]
R/W	R/W	--	--	--	--	R/W	R/W	R/W
Reset	0	--	--	--	--	0	0	0

tmr0_en: TMR0 使能位, 高有效, 缺省为 0;

t0ps [2:0]: TMR0 时钟频率选择:

000	FOSC/4
001	FOSC/32
010	FOSC/128
011	FOSC/1024
100	FOSC/8192
101	OSC32K/2
110	OSC32K
111	RC32K

定时器0计数预设值寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0L(FD6H)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

定时器0当前计数值

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RD_TMR0L(FB1H)								
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

10.3 定时器 1, TMR1

10.3.1 概述

24 位定时器 1 是递增型计数, 溢出产生中断, 计数器值可读, 计数起始值可以通过寄存器 TMR1U/TMR1H/TMR1L 设置, 计数时钟可以通过寄存器选择内部低频 RC 振荡时钟、晶振时钟、内部 MCU 工作时钟或其分频, 可工作于工作模式和待机模式, 支持 CAPTURE 功能。

TMR2相关寄存器:

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CON	cap_sel[1]	cap_sel[0]	t1ps[1]	t1ps[0]	--	--	cf_cap_en	tmr1_en
TMR1L	TMR1 预设值寄存器低 8 位							
TMR1H	TMR1 预设值寄存器中间 8 位							

TMR1U	TMR1 预设值寄存器最高 8 位							
RD_TMR1L	TMR1 的 24 位当前计数值的最低 8 位数据							
RD_TMR1H	TMR1 的 24 位当前计数值的中间 8 位数据							
RD_TMR1U	TMR1 的 24 位当前计数值的最高 8 位数据							
CCPR1L	24 位捕捉寄存器的最低 8 位数据							
CCPR1H	24 位捕捉寄存器的中间 8 位数据							
CCPR1U	24 位捕捉寄存器的最高 8 位数据							
PIE2	--	--	rcie	txie	bclie	lvdie	tmrlie	ccpie
PIR2	--	--	rcif	txif	bclif	lvdif	tmrlif	ccpif
INTCON	gie							

阴影单元未使用

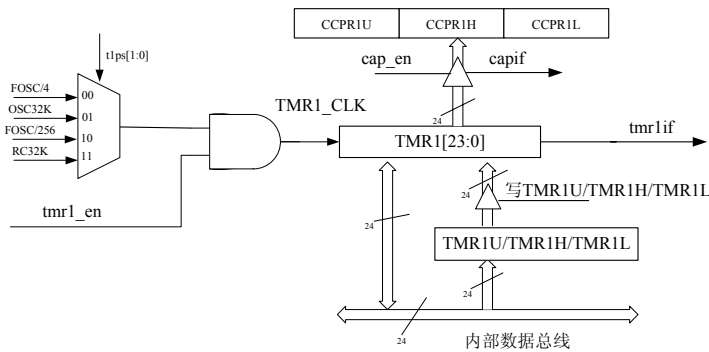


图 10-2 TMR1 结构图

10.3.2 TMR1 使用说明

首先对定时器时钟源进行选择，可以选择 4 种时钟，由 t1ps[1:0] 进行控制，具体见寄存器说明。无论选择哪种时钟，在待机模式下都可以正常工作。

设置寄存器 TMR1U/TMR1H/TMR1L，如果需要产生中断，将中断总使能位 gie 和定时器 1 中断位 tmrlie 置 1，将定时器 1 的使能位 tmrl_en 置 1，定时器 1 从 TMR1U/TMR1H/TMR1L 的值开始递增计数，定时器 1 计数到 FFFFFFFH 后溢出，计数器重新回到 TMR1U/TMR1H/TMR1L 设定值开始计数，如果不使用中断，可以将 tmrlie 置 0，通过查询 tmrlif 位来判断是否溢出事件产生。

10.3.3 寄存器说明-TMR1

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CON(FADH)	cap_sel[1]	cap_sel[0]	t1ps[1]	t1ps[0]	--	--	cf_cap_en	tmrl_en
R/W	R/W	R/W	R/W	R/W	--	--	R/W	R/W
Reset	0	0	0	0	--	--	0	0

tmrl_en: TMR1 使能位，高有效。

cf_cap_en: CF 捕捉使能信号，高有效，默认 0。

t1ps[1:0]: 频率选择控制位。

00	FOSC/4
----	--------

01	OSC32K
10	FOSC/256
11	RC32K

cap_sel[1:0]: 捕捉信号选择位, “00” 选择电量脉冲 CF, 可以选择这个选项来测量有功功率; “01” 选择电压信号符号位, 可以用这个选项来测量电压信号的工作频率; “10” 选择 I/O P10, 可以用来测量外部信号的脉宽; “11” 可以选择这个测频率。选择 11 测频率稳定性比 01 高。

00	电量脉冲 CF 输出
01	电压信号符号位
10	P10
11	低通滤波结果符号位

配置成01或11时, 中断信号不受中断使能ccpie控制。在需要捕捉电压信号符号位时, 再去配置为01或者11, 最好不要初始化就配置。也可以在进入中断的时候做两次判断, 先判断中断使能位, 再判中断标志位。

TMR1计数器预设值寄存器,

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR1L(FCEH)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TMR1H(FCFH)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
TMR1U(F63H)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

定时器1计数器当前值:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RD_TMR1L(FB3H)								
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
RD_TMR1H(FB2H)								
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
RD_TMR1U(F64H)								
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

10.3.4 捕捉功能说明

在捕捉模式下, 有外部输入引脚 P10 或者内部捕捉事件发生时, 寄存器 CCPIU/CCPIH/CCPIL 存储 TMR1 的 24 位计数值, 并产生中断信号 ccpif 置 1。捕捉时刻是

捕捉信号的上升沿。

捕捉寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPR1L(FBEH)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
CCPR1H(FBFH)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
CCPR1U(F65H)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

11 电量计量

11.1 概述

电量计量模块是一个高精度的单相有功功率计量电路，具有空载防潜动功能。包括对电压/电流有效值计算，电量的计算，跟 TMR1 的 CAPTURE 功能配合可以用来测量有功功率或者电压信号的频率。

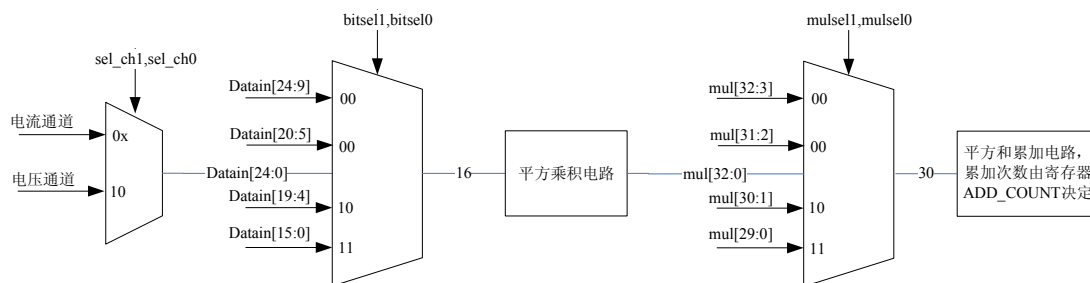


图 11-1 电能计量框图

11.2 寄存器说明

计量控制寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON (FC0H)	scf	ac_dc	shutdown	cf_fast	s1	s0	g1	g0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	1	1	1	1

shutdown: 计量模块使能控制。“1”，关计量模块

cf_fast: 电量脉冲 CF 输出加速信号，“1”，CF 输出频率为原来的 128 倍。

ac_dc: 电流通道交直流选择，“1”选择交流

g1~g0: 电流通道 (VIP、VIN) 的增益选择位;

g1~g0	通道 V1 增益	通道 V1 最大差动输入
00	1	$\pm 450\text{mV}$
01	2	$\pm 230\text{mV}$
10	8	$\pm 60\text{mV}$
11	16	$\pm 30\text{mV}$

scf、s1、s0、CF 的关系见下表:

序号	scf	s1	s0	CF(Hz)
1	1	0	0	$128 \times \text{Freq}$
2	0	0	0	$64 \times \text{Freq}$
3	1	0	1	$64 \times \text{Freq}$
4	0	0	1	$32 \times \text{Freq}$
5	1	1	0	$32 \times \text{Freq}$
6	0	1	0	$16 \times \text{Freq}$
7	1	1	1	$16 \times \text{Freq}$
8	0	1	1	$2048 \times \text{Freq}$

经实际测试: 前面 7 种设置 CF 输出频率是一样的。第 8 种设置 CF 输出频率被放大 128 倍。

其中 Freq 的频率跟电流通道的输入电压有效值、电压通道输入电压有效值、电流通道 PGA 增益、工作时钟频率成正比; 跟基准电压 V_{REF} 的平方成反比。

计量控制寄存器 2

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADC_CON2 (FC1H)	bitsel1	adc_en	mul_sel1	mul_sel0	ac_dc2	bitsel0	sel_ch1	sel_ch0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	0

sel_ch1, sel_ch0: 电压、电流通道选择, 具体见下表

sel_ch1, sel_ch0	选择的信号
0X	选择电流通道进行平方和计算
11	Not Used
10	选择电压通道进行平方和计算

adc_en: 电压, 电流平方和计算使能, “1” 使能

ac_dc: 电压通道交直流选择, “1” 选择交流

mul_sel1, mul_sel0: 选择进入累计和电路的位数, 可以根据电流通道输入信号的大小选择合适的的数据, 从而实现最大的测量精度, 建议选择默认值。

mul_sel1, mul_sel0	数据选择
00	最低 30 位数据
01	次低 30 位数据
10	次高 30 位数据
11	最高 30 位数据

bitsel1,bitsel0: 用来选择进入平方乘积运算电路的数据, 可以根据电流通道输入信号的大小选择合适的数据, 从而实现最大的测量精度。

bitsel1,bitsel0	数据选择	建议测量信号范围
00	最低 16 位数据	Not Used
01	次低 16 位数据	测试小信号如 50uV
10	次高 16 位数据	测试 500uV~50mV
11	最高 16 位数据	测试 50mV 及以上信号

累加结果寄存器

该寄存器是 40 位的平方和累加结果, 如果要得到有效值, 需要软件开平方。

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLT_ADJ0 (FC2H)	ADD[7]	ADD[6]	ADD[5]	ADD[4]	ADD[3]	ADD[2]	ADD[1]	ADD[0]
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
RLT_ADJ1 (FC3H)	ADD[15]	ADD[14]	ADD[13]	ADD[12]	ADD[11]	ADD[10]	ADD[9]	ADD[8]
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
RLT_ADJ2 (FC4H)	ADD[23]	ADD[22]	ADD[21]	ADD[20]	ADD[19]	ADD[18]	ADD[17]	ADD[16]
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
RLT_ADJ3 (FC5H)	ADD[31]	ADD[30]	ADD[29]	ADD[28]	ADD[27]	ADD[26]	ADD[25]	ADD[24]
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
RLT_ADJ4 (FC6H)	ADD[39]	ADD[38]	ADD[37]	ADD[36]	ADD[35]	ADD[34]	ADD[33]	ADD[32]
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

累计次数寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADD_COUNT0 (FC7H)	C[7]	C[6]	C[5]	C[4]	C[3]	C[2]	C[1]	C[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	1	0
ADD_COUNT1 (FC8H)	--	--	--	--	--	--	C[9]	C[8]

R/W	--	--	--	--	--	--	R/W	R/W
Reset	--	--	--	--	--	--	0	0

累加次数寄存器用来决定平方后的结果累加多少次，一般来说累加次数越多，累加后的结果抖动越小，但响应时间也会变长，使用的时候需要折中考虑。

电压瞬时结果寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VOL_RESULTL (FCAH)								
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
VOL_RESULTH (FC9H)								
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
VOL_RESULTU (FA9H)	--	--	--	--				
R/W	--	--	--	--	R	R	R	R
Reset	--	--	--	--	x	x	x	x

电流瞬时结果寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CURRENT_RESULTL(FCCH)								
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
CURRENT_RESULTH (FCBH)								
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x
CURRENT_RESULTU (FAAH)	--	--	--	--				
R/W	--	--	--	--	R	R	R	R
Reset	--	--	--	--	x	x	x	x

11.3 电量计数器

电量计数器为 2 个 16 位的计数器，分别对电能计量电路的正反向电能脉冲 CF 计数，当计数器溢出时，有标志位产生。可以利用这两个计数器进行电量的累计。

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
COUNTER_CON (FABH)	--	--	--	revp	up_ov	down_ov	up_en	down_en

R/W	--	--	--	R/W	R/W	R/W	R/W	R/W
Reset	--	--	--	0	0	0	0	0

revp: 反向功率标志位, “1”表示反向功率, 只读

up_ov: 正向电能脉冲计数溢出标志, “1”表示溢出; 软件可清零

down_ov: 反向电能脉冲计数溢出标志, “1”表示溢出; 软件可清零

up_en: 正向电能脉冲计数器使能位, 高有效, 低电平复位计数器

down_en: 反向电能脉冲计数器使能位, 高有效, 低电平复位计数器

12 USART 通信

12.1 概述

同步异步通信接口, 可以以全双工异步方式、半双工同步主控模式、半双工同步从动模式通信, 波特率可设, 可以连续读写, 8/9 位数据发送/接收, 有发送接收中断。

相关寄存器:

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCSTA	spen	rx9	sren	cren	--	ferr	oerr	rx9d
RCREG	USART 接收数据寄存器							
TXSTA	csrc	tx9	txen	sync	--	--	trmt	tx9d
TXREG	USART 发送数据寄存器							
SPBRG	USART 波特率设置寄存器							
PIE2	--	--	rcie	txie	bclie	lvdie	tmr1ie	ccpie
PIR2	--	--	rcif	txif	bclif	lvdif	tmr1if	ccpif
INTCON	gie	xtsie	sspif	tmr0ie	secie	keyie	int1ie	int0ie

阴影单元未使用

12.2 波特率设置

8-bit 波特率设置。

在异步模式下 ($\text{sync}=0$), $\text{BRG} = \text{FOSC} / (64 * (\text{SPBRG}[7:0] + 1))$

在同步模式下 ($\text{sync}=1$), $\text{BRG} = \text{FOSC} / (4 * (\text{SPBRG}[7:0] + 1))$

12.3 异步模式

12.3.1 启动

通过清零 sync 位选择异步工作模式。设置 spen=1, 并选择 txen/cren, 进行接收/发送通信。

12.3.2 波特率设置

8-bit 波特率设置，波特率公式：

$$BRG = FOSC / (64 * (SPBRG[7:0] + 1))$$

12.3.3 发送

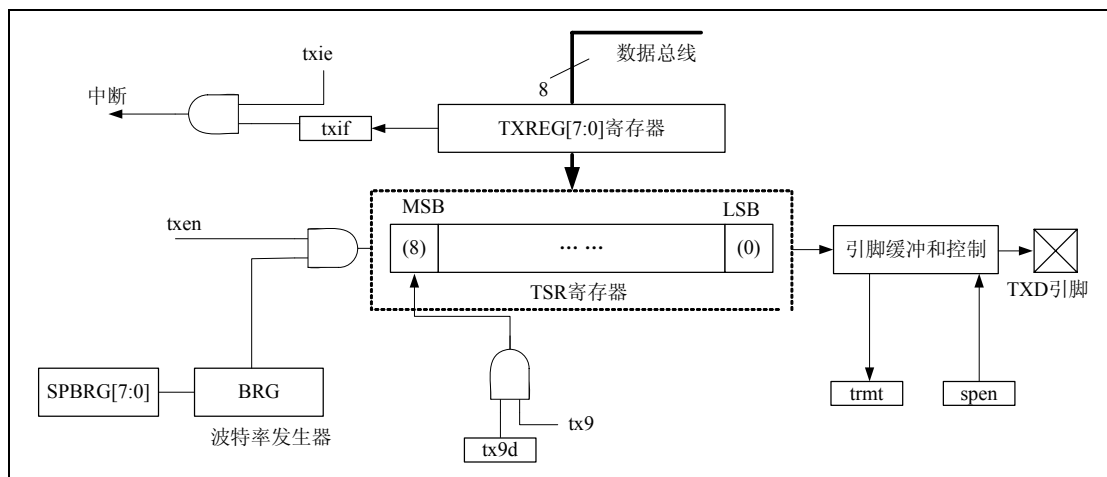


图 12-1 发送结构框图

发送器的核心是发送(串行)移位寄存器 TSR。移位寄存器从读/写发送缓冲寄存器 TXREG 中获取数据。TXREG 寄存器中的数据由软件写入。在前一次装入的停止位发送前，TXREG 寄存器中的新数据（如果有的话）就会被装入 TSR。

一旦 TXREG 寄存器向 TSR 寄存器传输了数据，TXREG 寄存器就为空，同时标志位 txif 置 1。可以通过将中断使能位 txie 置 1 或清零来使能/禁止该中断。不管 txie 的状态如何，只要中断发生，txif 就会置“1”并且不能软件清零。

txif 指示的是 TXREG 寄存器的标志位，而另一个位 trmt 则指示 TSR 寄存器的状态。trmt 是只读位，它在 TSR 寄存器为空时被置 1。trmt 位与任何中断均无关联，因此要确定 TSR 寄存器是否为空，用户只能对此位进行轮询。

注：

- 1) TSR 寄存器并未映射到数据寄存器中，用户不能对它直接访问。
- 2) 当使能位 txen 置 1 时，标志位 txif 也会置“1”。

发送步骤：

- 1) 设置 SPBRG，选择合适的波特率
- 2) 清零 sync，将 spen 置 1，使能异步串口。
- 3) 若需要中断，使能 txie, gie。
- 4) 若需要发送 9 位数据，将发送位 tx9 置“1”。
- 5) 通过将 txen 置 1 使能发送，此操作同时将 txif 置“1”。
- 6) 如果选择发送 9 位数据，将第 9 位数据装入 tx9d 位。
- 7) 将数据装入 TXREG 寄存器（开始发送）

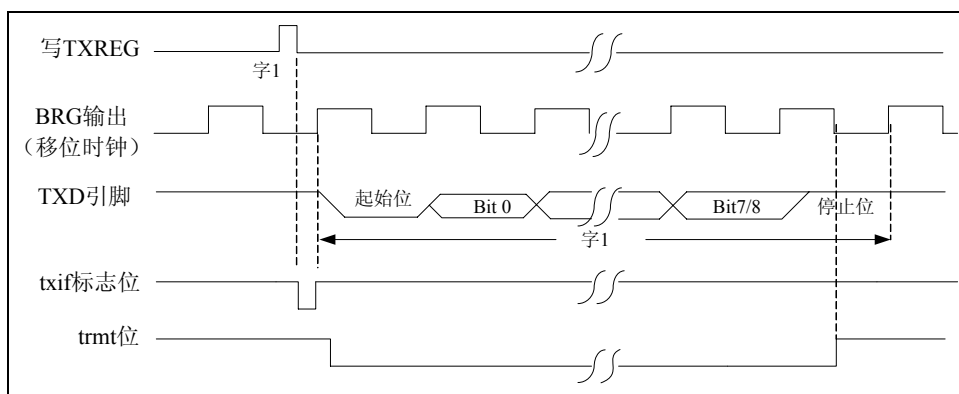


图 12-2 异步发送时序图

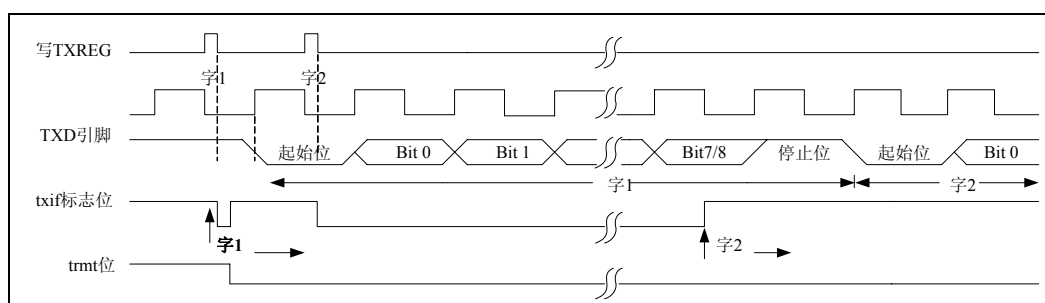


图 12-3 异步发送时序图（连续发送）

12.3.4 接收

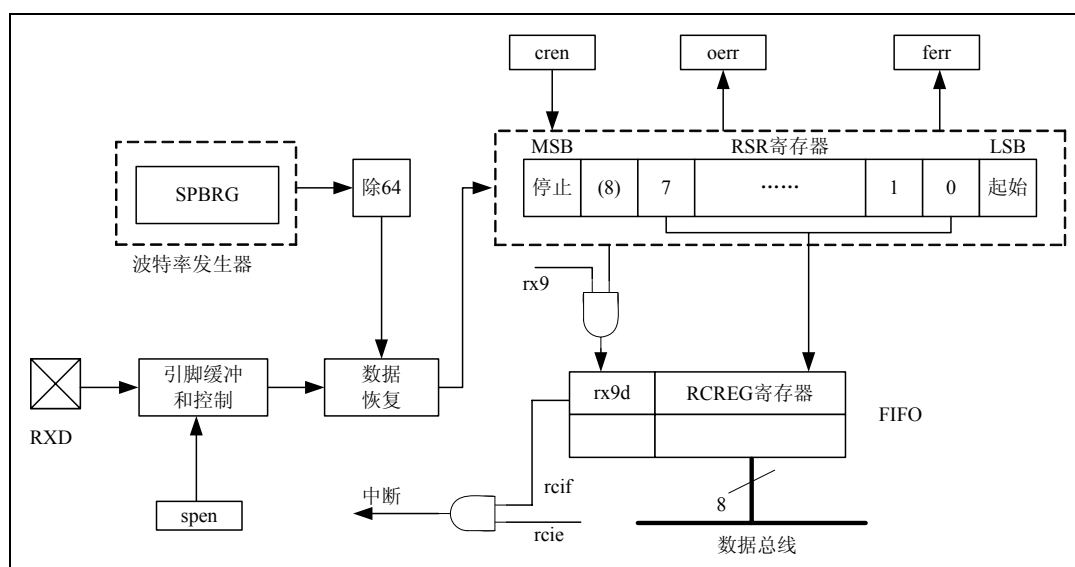


图 12-4 接收框图

接收框图如上，在 RX 引脚上接收数据，并驱动数据恢复电路。数据恢复电路实际上是一个 8 倍波特率的高速移位器，而主接收串行移位寄存器则工作在次波特率下。

设置接收的步骤如下：

- 1) 对 SPBRG 寄存器进行初始化，设置合适的波特率。
- 2) 清零 sync 选择异步模式，spen 置 1 使能端口。

- 3) 若需要中断，开启 rcie,gie
- 4) 将 cren 接收使能
- 5) 当接收完成，rcif 置 1，如果此时中断使能 rcie,gie 置 1，将会进入中断服务。
- 6) 读 RCSTA 获得第 9 位数据 rx9d（如果已使能）并判断在接收过程中是否发生错误。
- 7) 通过读 RCREG 获得接收到的 8 位数据。
- 8) 如果发生错误，通过使能位 cren 来消除错误。

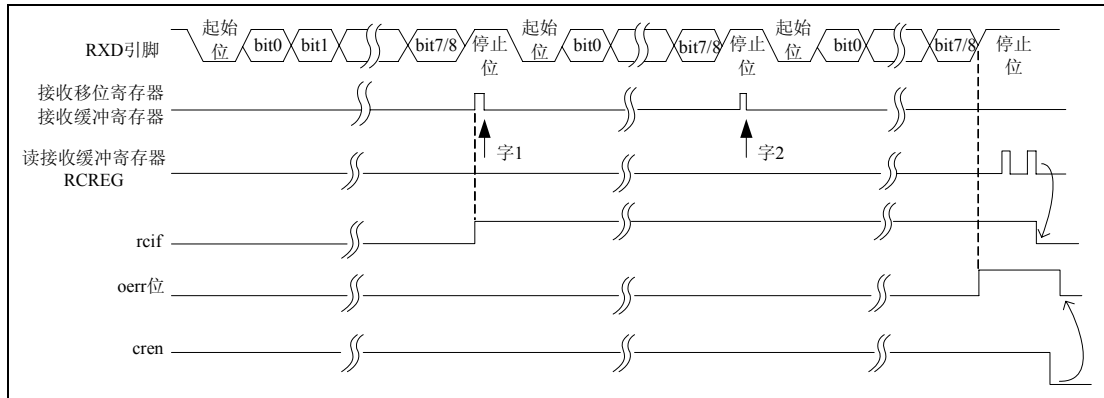


图 12-5 异步接收时序图

上图显示了在 RX 输入上顺序接收 3 个字节的情况，连续接收 3 个字中间没有读 RCREG 会使 oerr（溢出）位置 1。

12.4 同步主控模式

将 csrc 位置 1 可以进入同步主控模式。在此模式下，数据以半双工方式发送（即发送和接收不同时进行）。发送数据是，禁止接收，反之亦然。将 syn 位置 1 可进入同步模式。主控模式意味着芯片将在 CK/TX 时钟线上发送主控时钟信号。图 12-6 和 12-7 为发送框图。

波特率设置公式为： $BRG = FOSC / (4 * (SPBRG[7:0] + 1))$

12.4.1 同步主机发送模式

设置同步主控发送操作的步骤如下：

- 1) 对 SPBRG 寄存器初始化，设置合适的波特率
- 2) 设置 sync, spen, csrc 进入同步主机模式。
- 3) 如果需要中断，设置 txie=1, gie=1。
- 4) 如果需要 9-bit 传输，设置 tx9=1。
- 5) 设置 txen=“1”使能发送。
- 6) 如果选择了 9-bit 传输，第九位数据应写入 tx9d。
- 7) 将数据写入 txreg 寄存器开始发送。

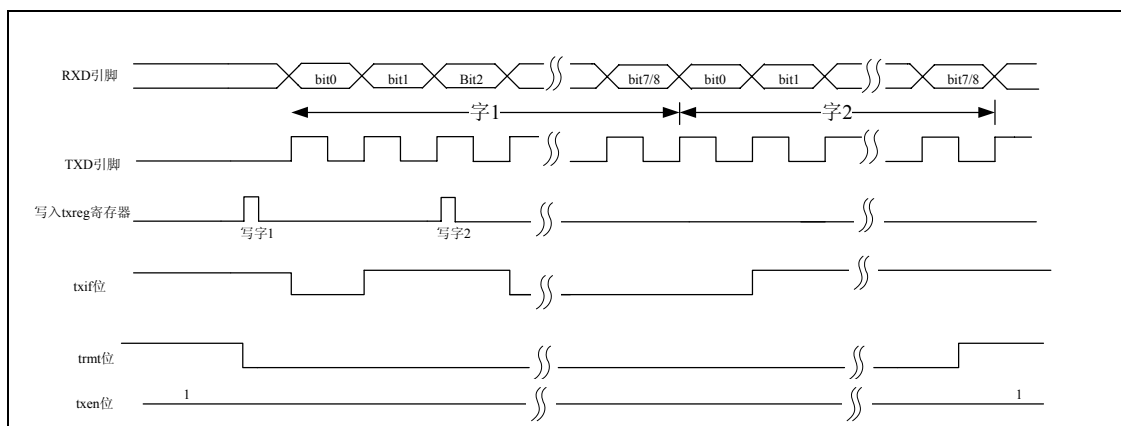


图 12-6 同步发送时序图

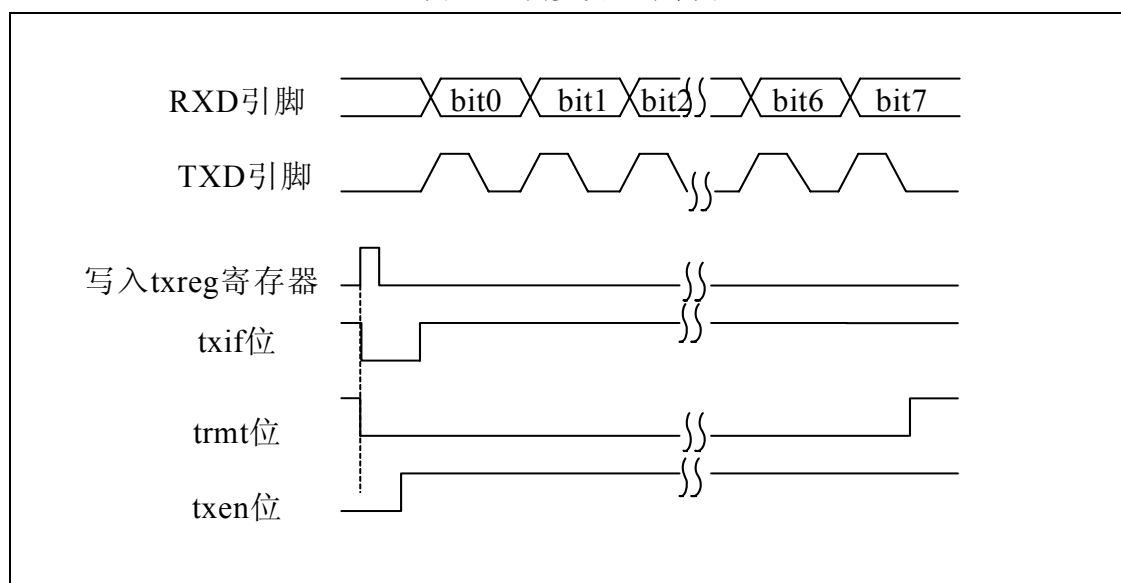


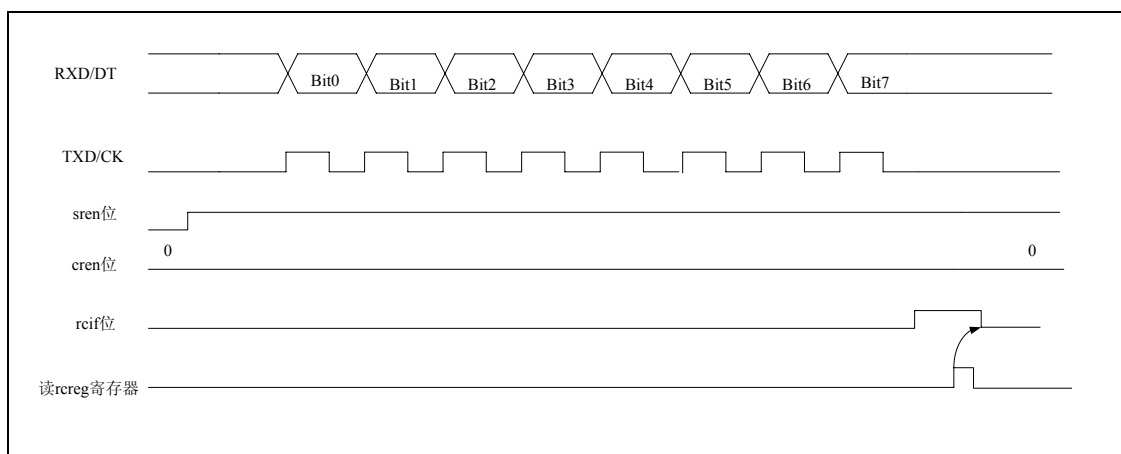
图 12-7 同步发送时序图（由 txen 控制）

12.4.2 同步主机接收模式

在主机接收模式下，由主控方发送时钟，由发送方根据时钟发送数据。

设置同步主机接收的步骤如下：

- 1) 对 SPBRG 初始化，设置合适的波特率。
- 2) 将 sync, spen, csrc 位置 1，使能同步主机串口。
- 3) 确保将 cren, sren 位清零。
- 4) 若需中断，将中断使能位 rcie, gie 置 1。
- 5) 若需要接收 9 位数据，将 rx9 位置 1。
- 6) 若需要单字接收，将 sren 位置 1；若需连续接收，将 cren 位置 1。
- 7) 当接收完成时，中断标志位 rcif 将置 1，此时如果使能中断，将产生一个中断。
- 8) 读 rcsta 寄存器以获取第 9 位数据，并判断在接收过程中是否发生错误。
- 9) 通过读 rcreg 寄存器来读取接收到的 8 位数据，
- 10) 如果发生错误，将 cren 清零来清除错误。



12.5 同步从动模式

将 csrc 清零，进入同步从动模式。此模式与同步主控模式的区别为移位时钟由时钟脚 CK 提供（在同步主控模式下由内部时钟源提供）。这使得器件能在低功耗模式下发送或接收数据。

12.5.1 同步从动发送

除了待机模式以外，同步主控，同步从动模式的工作原理是相同的。

如果向缓冲器 TXREG 写入 2 个字节，然后执行 SLEEP 指令，将发生以下事件：

- 第一个字立即发送到 TSR 寄存器进行发送。
- 第二个字仍保留在 TXREG 寄存器中
- 标志位 txif 不会被置 1
- 当第一个字移出 TSR 后，TXREG 寄存器把第二个字送入 TSR,同时将标志位 TXIF 置“1”。
- 如果已经开启中断使能，中断将把芯片从待机模式下唤醒。

设置同步从动发送的步骤如下：

- 1) 通过将 sync 和 spen 置 1，csrc 清零，使能同步从动串口。
- 2) 将 cren 和 sren 清零
- 3) 若需要中断，将中断使能为 txie 和 gie 置“1”
- 4) 若需要发送 9 位数据，将 tx9 置“1”
- 5) 将使能位 txen 位置“1”使能发送
- 6) 如果选择发送 9 位数据，应该将第 9 位数据装入 tx9d 位。
- 7) 将数据装入 TXREG 寄存器启动发送

12.5.2 同步从动接收

除了待机模式，休眠模式以及在从动模式下忽略 sren 位以外，同步主控模式和从动模式的工作原理相同。

如果在进入待机或休眠模式前将 cren 位置 1 使能接收，那么在低功耗模式下可以接收一个数据字。接收到该字后，RSR 寄存器把数据发送到 RCREG 寄存器，如果中断使能位已经开启，产生的中断将被芯片从低功耗模式中唤醒。

设置同步从动的接收的步骤如下：

- 1) 通过将 sync 和 spen 置 1, crsc 位清零, 使能 同步从动串口。
- 2) 若需要中断, 将中断使能位 rcie 和 gie 置 1
- 3) 若需要接收 9 位数据, 将 rx9 位置 1
- 4) 将使能位 cren 置 1 使能接收
- 5) 当接收完成, rcif 位将置 1, 如果中断使能位已开启, 将产生一个中断
- 6) 读 RCSTA 寄存器以获得第 9 位数据

12.6 寄存器说明

接收设置寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCSTA(F7BH)	spen	rx9	sren	cren	--	ferr	oerr	rx9d
R/W	R/W	R/W	R/W	R/W	--	R	R	R
Reset	0	0	0	0	--	0	0	U

- spen: UART 使能位, “1” 使能;
- rx9: 9 位选择接收使能。“1” 选择 9 位接收, “0” 选择 8 位接收;
- sren: 接收使能位
同步模式: “1” 选择接收, “0” 禁止接收;
异步模式: 无用
- cren: 连续接收使能位, “1” 使能接收, “0” 禁止接收
- ferr: 帧出错标志位, 高电平标志出错
- oerr: 溢出出错标志位, 高电平标志出错
- rx9d: 接收数据的第 9 位

接收数据寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCREG(F7CH)								
R/W	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

发送设置寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TXSTA(F7DH)	csrc	tx9	txen	sync	--	--	trmt	tx9d
R/W	R/W	R/W	R/W	R/W	--	--	R	R/W
Reset	0	0	0	0	--	--	1	0

- csrc: 时钟源选择位,
同步模式: “1” 主机模式; “0” 从机模式;
异步模式: 无用
- tx9: 9 位选择发送使能位。“1” 选择 9 位发送, “0” 选择 8 位发送
- txen: 发送使能位, 高电平使能
- sync: UART 模式选择位, “1” 同步模式, “0”, 异步模式
- trmt: 发送移位寄存器空标志位, 高电平为空
- tx9d: 发送数据的第 9 位 (可以通过软件设置作为奇偶校验位)

波特率设置寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPBRG(F7FH)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

发送数据寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TXREG(F7EH)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

发送数据的设置必须在 **spen** 和 **txen** 设置好后，才能写入值。

13 LCD DRIVER

13.1 LCD driver 概述

SD3004 支持 1/4duty、1/3bias LCD 驱动，内部包括增强 LCD 驱动能力的电路。输出 4 个 COM 信号与 24 个 SEG 信号，支持 4×24 的 LCD。SD3003 支持 4×15 的 LCD

相关寄存器：

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDDATA0	S01C3	S01C2	S01C1	S01C0	S00C3	S00C2	S00C1	S00C0
LCDDATA1	S03C3	S03C2	S03C1	S03C0	S02C3	S02C2	S02C1	S02C0
LCDDATA2	S05C3	S05C2	S05C1	S05C0	S04C3	S04C2	S04C1	S04C0
LCDDATA3	S07C3	S07C2	S07C1	S07C0	S06C3	S06C2	S06C1	S06C0
LCDDATA4	S09C3	S09C2	S09C1	S09C0	S08C3	S08C2	S08C1	S08C0
LCDDATA5	S11C3	S11C2	S11C1	S11C0	S10C3	S10C2	S10C1	S10C0
LCDDATA6	S13C3	S13C2	S13C1	S13C0	S12C3	S12C2	S12C1	S12C0
LCDDATA7	S15C3	S15C2	S15C1	S15C0	S14C3	S14C2	S14C1	S14C0
LCDDATA8	S17C3	S17C2	S17C1	S17C0	S16C3	S16C2	S16C1	S16C0
LCDDATA9	S19C3	S19C2	S19C1	S19C0	S18C3	S18C2	S18C1	S18C0
LCDDATA10	S21C3	S21C2	S21C1	S21C0	S20C3	S20C2	S20C1	S20C0
LCDDATA11	S23C3	S23C2	S23C1	S23C0	S22C3	S22C2	S22C1	S22C0
LCDCOM	--	--	--	--	com[3:0]			
LCDCON	lcsen	--	--	--	cs1	cs0	--	--
LCDSE0	se[7:0]							
LCDSE1	se[15:8]							
LCDSE2	se[23:16]							

13.2 LCD Driver 的时钟

LCD driver 模块能够灵活的选择时钟源以适应不同的应用方案，其内部定义两个时钟：f lcdck、f frame。描述如下：

- 🌈 f lcdck: LCD driver logic 的工作时钟，由 f lcd 分频产生；
- 🌈 f frame: LCD 显示的帧扫描频率，为 f lcdck/4。

LCD driver 模块的时钟电路如下图：

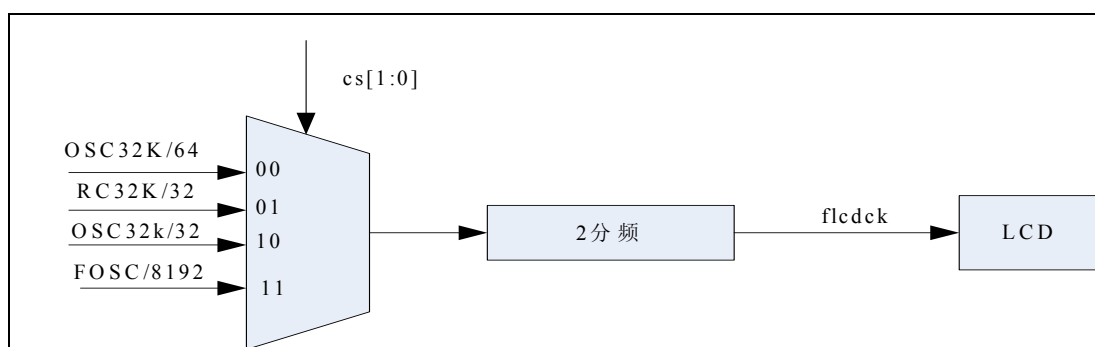


图 13-1 LCD 工作频率选择

13.3 LCD 驱动模块设置

LCD 控制寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON(F59H)	lcden	--	--	biassel	cs1	cs0	--	--
R/W	R/W			R/W	R/W	R/W		
Reset	0			0	0	0		

- cs1,cs0 : 缺省为 00
该两位用于控制 LCD 驱动模块的内部时钟：
cs1,cs0 = (0, 0): OSC32K/64.
 = (0, 1): RC32K/32.
 = (1, 0): OSC32K/32
 = (1, 1): FOSC/8192
- lcden : 缺省为 0
该位为 LCD 驱动模块的使能信号，“0”： Disable； “1”： Enable。
- biassel: “1”： LCD 的偏置电压为电源电压；“0”，LCD 的偏置电压需要外接。

13.4 COM/SEG 设置

COM/SEG 与数字 I/O PORT2、PORT3、PORT4 复用。当用作 LCD 驱动时必须对 I/O 进行设置。COM3~COM0 的设置位分别按数字对应 LCDCOM 寄存器中的 com[3:0]。同样，SEG0~23 的设置位分别按数字对应 LCDSE2、LCDSE1 与 LCDSE0 中的 se[23:0]。

COM选择寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCOM (F58H)	--	--	--	--	com[3]	com[2]	com[1]	com[0]
R/W	--	--	--	--	R/W	R/W	R/W	R/W
Reset	--	--	--	--	0	0	0	0

com[3:0]:

“0”，对应管脚为标准数字 I/O，由数字 I/O 的相应控制位决定功能（默认）。

“1”，选择对应管脚为 LCD driver 的 COM 功能，将数字 I/O 的功能关闭。

SEG选择寄存器：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSE2(F5C)	se[23]	se[22]	se[21]	se[20]	se[19]	se[18]	se[17]	se[16]
LCDSE1(F5B)	se[15]	se[14]	se[13]	se[12]	se[11]	se[10]	se[9]	se[8]
LCDSE0(F5A)	se[7]	se[6]	se[5]	se[4]	se[3]	se[2]	se[1]	se[0]

se[23:0]:

“0”，对应管脚为标准数字 I/O，由数字 IO 的相应控制位决定功能（Reset 后默认）。

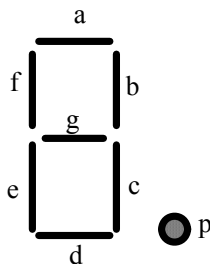
“1”，选择对应管脚为 LCD driver 的 SEG 功能，将数字 I/O 端关闭。

13.5 显示数据设置

SD3004 的 LCD Driver 输出 4 个 COM、24 个 SEG，一共可驱动 96 段显示。由 LCDDATA0~11 一共 12 个寄存器 96bit 进行控制。

每个字节寄存器 LCDDATA 控制 2 个字节引脚 SEG_n，而每个字节引脚具有 4bit 控制位 SEG_n[3:0]。控制位是否产生有效输出与对应的 I/O 是否开启为 COM/SEG 功能有关。

对于通用的段码式 LCD 屏连接方式，对应的段码如下：



LCD SEG 数据寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
对应字段	a	b	c	p	f	g	e	d

LCDDATA0(F60)	S01C3	S01C2	S01C1	S01C0	S00C3	S00C2	S00C1	S00C0
LCDDATA1(F61)	S03C3	S03C2	S03C1	S03C0	S02C3	S02C2	S02C1	S02C0
LCDDATA2(F62)	S05C3	S05C2	S05C1	S05C0	S04C3	S04C2	S04C1	S04C0
LCDDATA3(F66)	S07C3	S07C2	S07C1	S07C0	S06C3	S06C2	S06C1	S06C0
LCDDATA4(F67)	S09C3	S09C2	S09C1	S09C0	S08C3	S08C2	S08C1	S08C0
LCDDATA5(F68)	S11C3	S11C2	S11C1	S11C0	S10C3	S10C2	S10C1	S10C0
LCDDATA6(F71)	S13C3	S13C2	S13C1	S13C0	S12C3	S12C2	S12C1	S12C0
LCDDATA7(F72)	S15C3	S15C2	S15C1	S15C0	S14C3	S14C2	S14C1	S14C0
LCDDATA8(F73)	S17C3	S17C2	S17C1	S17C0	S16C3	S16C2	S16C1	S16C0
LCDDATA9(F77)	S19C3	S19C2	S19C1	S19C0	S18C3	S18C2	S18C1	S18C0
LCDDATA10(F78)	S21C3	S21C2	S21C1	S21C0	S20C3	S20C2	S20C1	S20C0
LCDDATA11(F79)	S23C3	S23C2	S23C1	S23C0	S22C3	S22C2	S22C1	S22C0

所有位:

0---熄灭（默认）

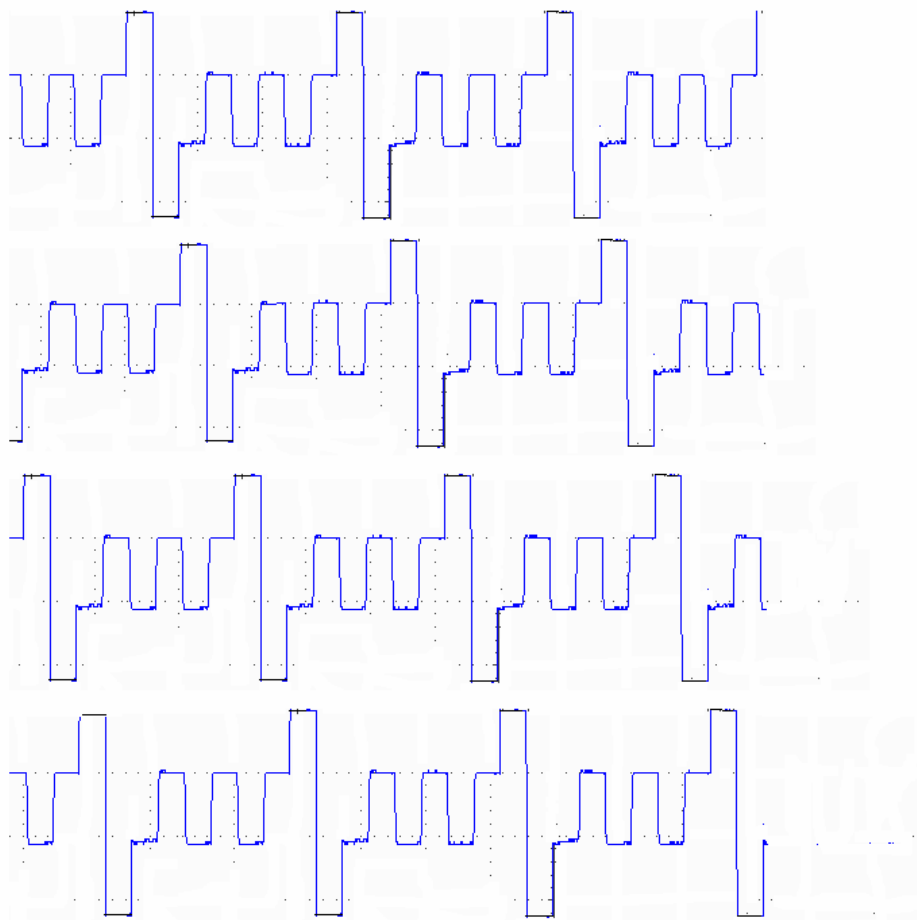
1---点亮。

注：上面的段码对应只是厂家按照常用的连接方式给出。不同的连接方式，应该自行判断段码对应关系。

SD3003 使用的数据寄存器是 LCDDAT0-LCDDAT7，其中 LCDDATA7 只使用该寄存器的低 4 位。

13.6 LCD 输出波形

COM:



一组实际测量的 COM-SEG 波形:

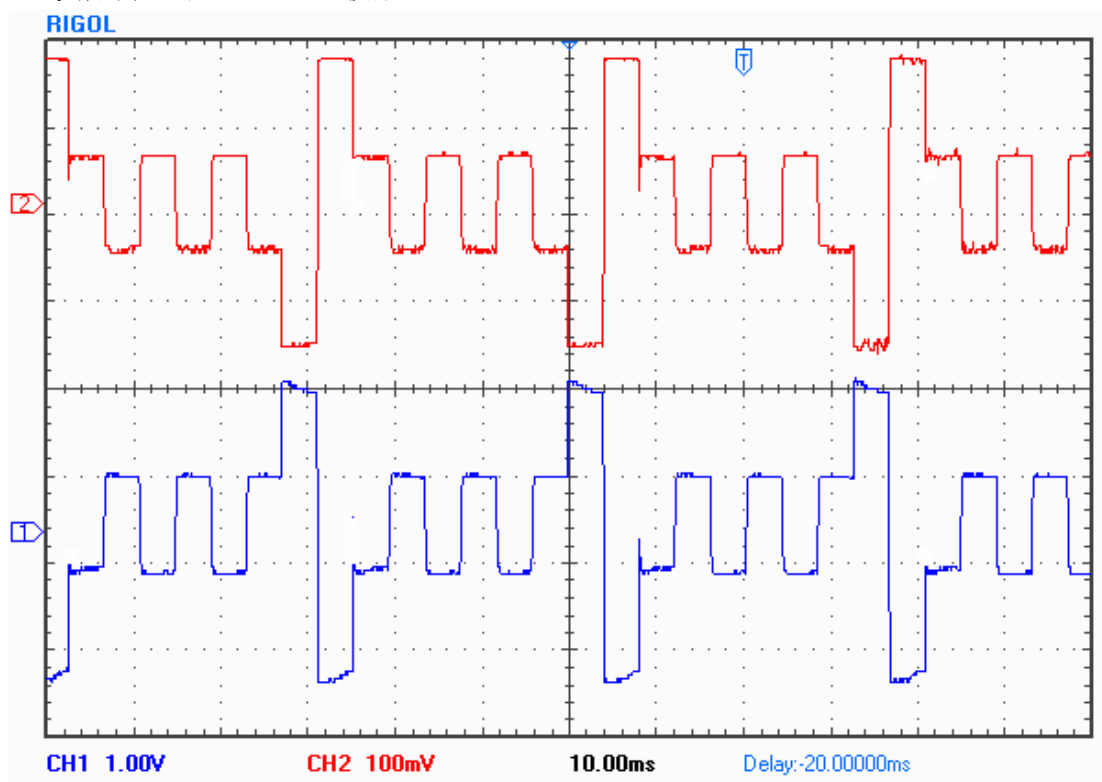


图 13-2 LCD 驱动波形

14. RTC

14.1 概述

SD3004 提供 RTC 功能，输入时钟 32.768KHz，RTC 使用外部低频晶振作为时钟，RTC 模块本身没有使能信号，一旦当且仅当低频晶振使能时，RTC 就开始正常工作，否则关闭。RTC 模块的复位信号仅为异步复位（低压复位或外部复位），只有当该情况的复位，其寄存器值才会恢复初值。

相关寄存器:

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PIR	--	--	sspif	timer0if	secif	--	int1if	int0if
INTCON	gie	--	sspie	tmr0ie	secie	--	int1ie	int0ie

阴影单元未使用。

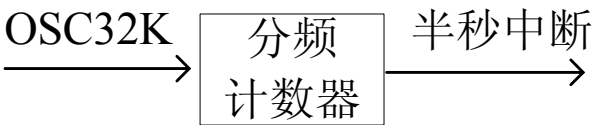


图 14-1 实时时钟 RTC 结构框图

14.2 RTC 使用

14.2.1 RTC 初始设置

RTC 的工作时钟由外部低频晶振提供。

14.2.2 RTC 中断事件服务

当时间计到 0.5，就有一个中断输出。若要开启中断服务，必须设置 secie 和 gie 为 1.

14.2.3 RTC 使能

RTC 没有单独的使能信号，当外部低频晶振时钟设置正确即开始工作。若外部低频晶振设置不正确则不工作。

15. 串行通信-I2C 主从机接口

15.1 I2C 概述

SD3004 提供 I2C 接口，可以实现主控和从动模式（包括全局呼叫模式），并且可以通过硬件提供启动位中断和停止位中断以确定空闲总线（多主控模式）。支持 7 位寻址。可以在工作模式和待机模式（仅从机模式）下工作。

相关寄存器：

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPADD	地址寄存器							
SSPBUF	接收/发送缓存寄存器							
SSPCON1	wcol	sspov	sspen	ckp	sspm3	sspm2	sspm1	sspm0
SSPCON2	gcen	ackstat	ackdt	acken	rcen	pen	rsen	sen
SSPSTAT	--	--	d/na	P	S	r/nw	--	bf
PIE2	--	--	rcie	txie	bclie	Int2ie	Tmr1ie	ccpie
PIR2	--	--	rcif	txif	bclif	Int2if	Tmr1if	ccpif
PIR1	--	--	sspif	timer0if	secif	--	int1if	int0if
INTCON	gie	--	sspie	tmr0ie	secie	--	int1ie	int0ie

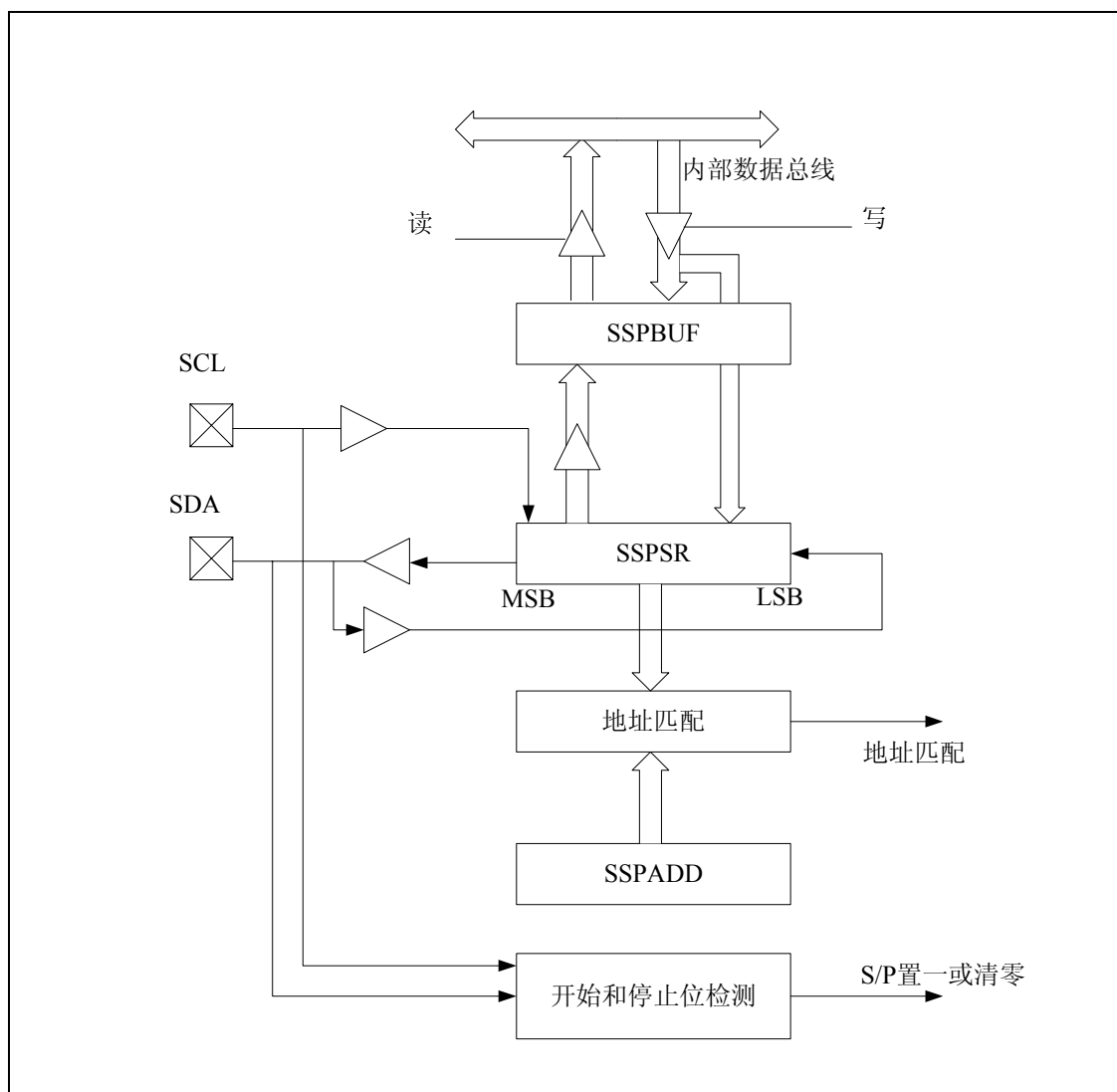


图 15-1 I2C 结构图

15.2 I2C 从动模式使用说明

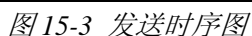
15.2.1 启动

设置 `sspm[3:0]` 为 0110 选择从动模式。设置 `sspen=1` 开启 I2C 功能，并配置端口 P13/P12 自动为 I2C 接口 SDA/SCL，并根据接收发送操作自动切换输入和输出。设置 `sspen=0` 关闭 I2C 功能。

在从动模式下，硬件总是在地址匹配时产生中断。当地址匹配或在地址匹配后发送的数据被接收时，硬件会自动产生一个应答（ACK_b）脉冲，并把当时 SSPSR 寄存器中接收到的值装入 SSPBUF 寄存器。

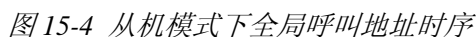
只要满足下列条件之一，I2C 模块就不会产生 ACK_b 脉冲：

- ◆ 在接收到数据前，缓冲器满位 BF 置“1”。
- ◆ 在接收到数据前，溢出位 `sspov` 被置“1”。



在 I2C 总线的寻址过程中，通常由启动条件后第一个字节决定主机器件将寻址哪个从机器件。但全局呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答响应。

全局呼叫使能位 (**gcn**) 使能时, 即可识别全局呼叫地址。检测到启动条件后, 8 位数据会移入 **SSPSR**, 并将该地址与 **SSPADD** 比较, 还会和全局呼叫地址进行比较。如果与全局呼叫地址匹配, **SSPSR** 的值将传递到 **SSPBUF**, **bf** 标志位置 1, 并且 **sspif** 中断标志位在第 9 位的下降沿置 1。当中断得到响应, 可以通过读取 **SSPBUF** 的内容来检查中断源, 并用于判断地址是特定器件还是一个全局呼叫地址。



通过将 SSPCON1 中的 SSPM[3:0] 设为 1000，选择主机模式。将 SSPEN 置 1，使能 I2C 功能。在主机模式下，SCL 和 SDA 由芯片硬件控制。

2012-05-26

都清零，总线处于空闲状态，也可以获得总线的控制权。

一旦使能主机模式，用户可以选择以下 6 项操作：

- 1) 在 SDA 和 SCL 上发出一个启动条件
- 2) 在 SDA 和 SCL 上发出一个重复启动条件
- 3) 写入 SSPBUF 寄存器，启动数据/地址的发送
- 4) 配置 I2C 端口用于接收数据
- 5) 在接收数据字节末尾产生应答条件
- 6) 在 SDA 和 SCL 上产生停止条件

下列事件会使 sspif 中断标志位置 1：

启动条件，停止条件，数据字节发送/接收，应答发送，重复启动。

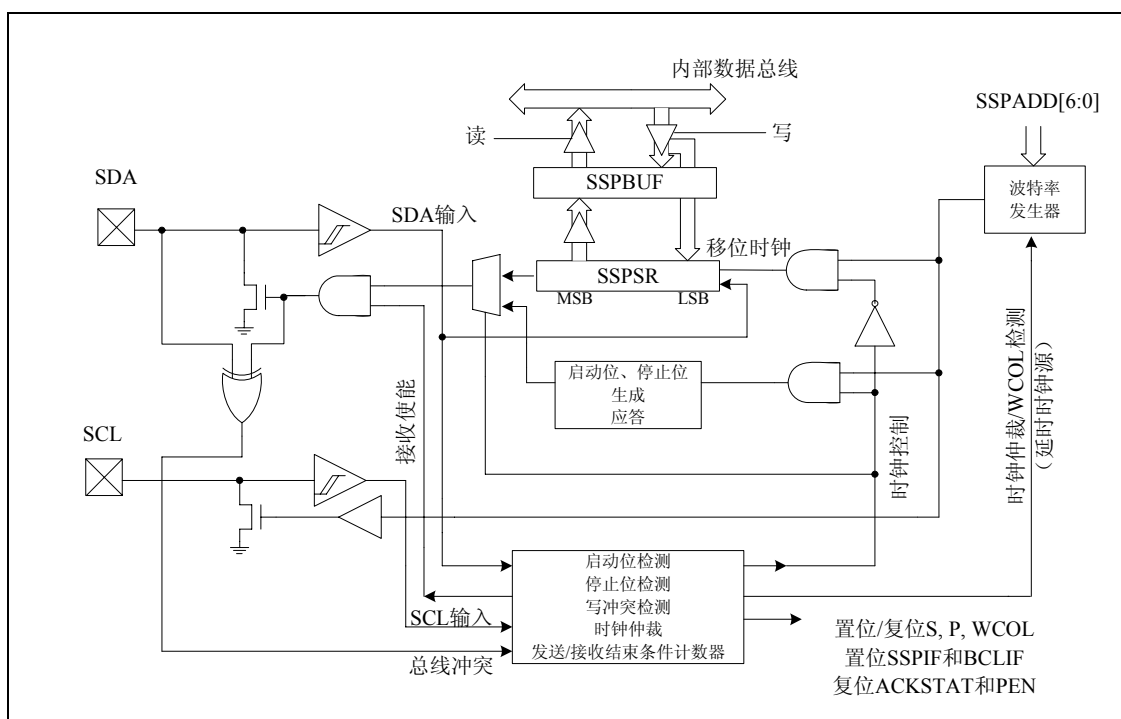


图 15-5 I2C 主机模块框图

15.4.1 I2C 主机模式工作方式

主机器件产生所有串行时钟脉冲和启动/停止信号。以停止信号或重复启动信号接收传输。因为重复启动信号也是下一个串行传输的开始，因此不会释放 I2C 总线。

在主机发送模式下，串行数据通过 SDA 输出，而串行时钟由 SCL 输出。发送的第一个字节包括器件的从机器件地址（7 位）和读/写（r/nw）位，在发送模式下，r/nw 位为 0。

15.4.2 波特率发生器

在 I2C 主机模式下，波特率发生器的重载值在 SSPADD 寄存器的低 7 位，当发生对 SSPBUF 的写操作时，波特率发生器自动开始计数。BRG 会递减计数至 0，然后停止，并等待另一次的重载。一旦完成指定的操作（即，在发送的最后数据位后面带有 nack），内部时钟将自动停止计数，SCL 引脚将保持在其最后状态。

$$FBRG = FOSC / (4 * (SSPADD + 2))$$

在时钟的发送序列上，SCL 的高电平和低电平时间至少保持一个 BRG 时间。如果发生时钟仲裁，即在器件发送的 SCL 由低变高时，SCL 被外部拉低，保持低电平。器件将等待 SCL 恢复为高电平后才开始新的 BRG 计数。

15.4.3 发送和接收时序

15.4.3.1 主机发送流程

步骤如下：

- 1) 设置 SSPADD，设置合适的波特率
- 2) 设置 sen=1 开始发送启动时序。等待 sen 自动清零（同时 sspif 自动置 1），表明启动时序发送完成。
- 3) 将要发送的数据写入 SSPBUF，等待 r/nw 自动清零（同时 sspif 自动置 1），表明数据发送完成。读 ackstat 判断从机是否应答。
- 4) 如要发送多字节，重复（3）。
- 5) 如果要结束发送，设置 pen=1 发送停止时序。等待 pen 自动清零（同时 sspif 自动置 1），表明停止时序发送完成。

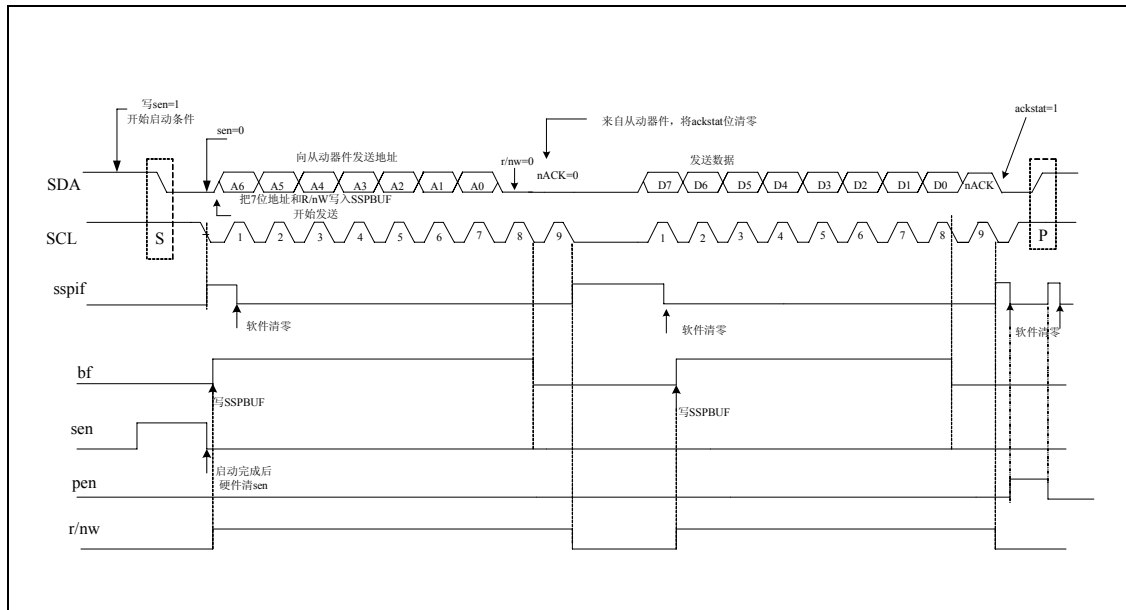


图 15-6 I2C 主机发送时序

15.4.3.2 主机接收流程

步骤如下：

- 1) 设置 SSPADD，选择合适的波特率
- 2) 设置 sen=1, 发送启动时序。等待 sen 自动清零（同时 sspif 自动置 1），表明启动时序发送完成。
- 3) 将要发送的从机地址写入 SSPBUF，开始发送。等待 r/nw 自动清零（同时 sspif 自动置 1），表明数据发送完成。
- 4) 设置 rcen=1, 开始接收从机发送的数据。等待 rcen 自动清零（同时 sspif 自动置 1），表明接收数据完成。从 SSPBUF 寄存器读取数据。
- 5) 将要发送的应答字写入 ackdt。设置 acken=1，开始发送应答序列。等待 acken 自动清零（同时 sspif 自动置 1），表明应答序列发送结束。

- 6) 如果要接收多个字节, 重复步骤 (4) (5)
- 7) 如果要发送结束序列, 设置 **pen=1**, 开始发送结束序列。等待 **pen** 自动清零 (同时 **sspif** 自动置 1), 表明结束序列发送结束。

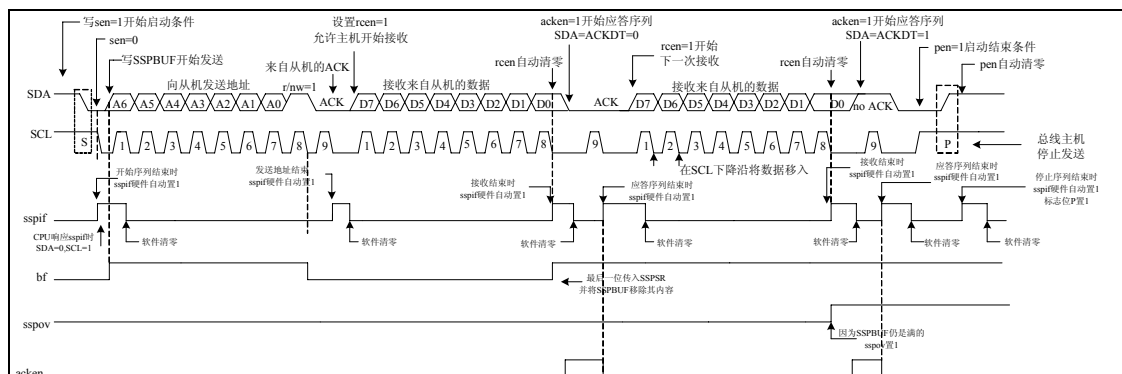


图 15-7 I2C 主机接收时序

15.4 多主机通信、总线冲突和总线仲裁

支持多主机模式是通过总线仲裁来实现的。当主器件将地址/数据位输出到 SDA 引脚时, 如果主机将 SDA 输出 1, 而另一个主器件将 SDA 输出 0, 就发生了总线仲裁。当 SCL 为高电平, SDA 上的数据应当稳定, 如果 SDA 引脚上期望的数据是 1, 而实际采样到的数据是 0 就发生了总线冲突。当发生总线冲突, 主器件将把总线冲突标志位 **BCLIF** 置 1, 并将 I2C 端口复位到空闲状态。

如果在发送状态发生总线冲突, 则中止发生, **bf** 标志位清零, SDA 和 SCL 拉高, 并且 **SSPBUF** 可写。如果在启动、重复启动、停止或应答条件的执行过程中发生总线冲突, 则这种条件中止, SDA 和 SCL 拉高, **SSPCON2** 的相应寄存器清零。

在发生总线冲突后, 可以将 I2C 切换为从机模式, 以检查标志位 **P** 是否置 1, 判断总线空闲。若总线空闲, 可再切换为主机模式进行通信。

15.5 寄存器说明

状态寄存器:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPSTAT(F6FH)	--	--	d/na	P	S	r/nw	--	bf
R/W	--	--	R	R	R	R	--	R
reset	--	--	0	0	0	0	--	0

d/na: 数据/地址位

1= 表明接收或发送的最后字节是数据

0= 表明接收或发送的最后字节是地址

P: 停止位

1= 表示最近检测到停止位

0= 表示未检测到停止位

注: 仅在复位时和 **SSPEN** 清零时, 该位清零。

S: 启动位

1= 表示最后检测到启动位

0= 表示未检测到启动位

注: 仅在复位时和 **SSPEN** 清零时, 该位清零。

r/nw: 读/写位信息

在从动模式下:

1= 读

0= 写

注: 该位用来保存在最近一次地址匹配后的R/W 位信息。仅从本机地址与接收地址匹配开始, 到下一个启动位、停止位或无ACK 位时, 该位有效。

在主机模式下:

1= 正在进行发送

0= 未进行发送

注: 该位与SEN、RSEN、PEN、RCEN 或ACKEN 进行“或”运算的结果表示SD3004 是否处在空闲模式。

bf: 缓冲器满状态位

在发送模式下:

1= 接收完成, SSPBUF 满

0= 接收没有完成, SSPBUF 空

在接收模式下:

1= 数据正在发送(不包括ACK 位和停止位), SSPBUF 满

0= 数据发送完毕(不包括ACK 位和停止位), SSPBUF 空

控制寄存器1:

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPCON1(F6DH)	wcol	sspov	sspen	ckp	sspm3	sspm2	sspm1	sspm0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	1	0	0	0	0

wcol: 写冲突检测位

在主机发送模式下:

1= 当I2C 不满足开始发送数据的条件时有数据要写入SSPBUF 寄存器(该位必须用软件清零)

0= 未发生冲突

在从动发送模式下:

1= 正在发送前一个字时又有数据写入SSPBUF 寄存器(该位必须用软件清零)

0= 未发生冲突

在接收模式(主机或从动模式)下: 这是一个“忽略”位

sspov: 接收溢出标识位

在接收模式下:

1= 当SSPBUF 寄存器中仍保存前一个字节时收到了下一个字节(该位必须用软件清零)

0= 没有溢出

在发送模式下:

这在发送模式下是“忽略”位

sspen: 同步串行口使能位

1= 使能串行口, 并配置SDA 和SCL 引脚为串行口引脚

0= 禁止串行口, 并配置这些引脚为I/O 口引脚

注: 当该位为1 而使能时, 必须正确配置SDA 和SCL 引脚为输入引脚或输出引脚。

ckp: SCK 释放控制位

在从动模式下：

1= 释放时钟

0= 保持时钟线为低电平（时钟延长），用于保证数据的建立时间

在主机模式下：在此模式下未使用

sspm3-sspm0：同步串行口模式选择位

1000=I2C 主控模式，时钟=FOSC / (4 * (SSPADD+2))

0110=I2C 从动模式， 7 位地址

注： 其他模式不可用。

控制寄存器2：

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPCON2(F6EH)	gcen	ackstat	ackdt	acken	rcen	pen	rsen	sen
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

gcen：全局呼叫使能位（仅从动模式）

1=SSPSR 接收到全局呼叫地址（0000h）时使能中断

0= 禁止全局呼叫地址

ackstat：应答状态位（仅主机发送模式）

1= 没有收到来自从动器件的应答

0= 收到来自从动器件的应答

ackdt：应答数据位（仅主机接收模式）

1= 不应答

0= 应答

注： 用户在接收结束时启动一个应答序列，同时发送该值。

acken：应答序列使能位（仅主机接收模式）

1= 在SDA 和SCL 引脚启动应答序列，发送ackdt数据位。

由硬件自动清零。

0= 应答序列空闲

rcen：接收使能位（仅主机模式）

1= 使能I2C 接收模式

0= 接收空闲

pen：停止条件使能位（仅主机模式）

1= 在SDA 和SCL 引脚启动停止条件。由硬件自动清零。

0= 停止条件空闲

rsen：重复启动条件使能位（仅主机模式）

1= 在SDA 和SCL 引脚启动重复启动条件。

由硬件自动清零。

0= 重复启动条件空闲

sen：启动条件使能（仅主机模式）

在主机模式下：

1= 在SDA 和SCL 引脚启动启动条件。由硬件自动清零。

0= 启动条件空闲

注： 对于acken、rcen、pen、rsen 和sen位：如果 I2C 模块不在空闲模式，不对该位置1（不支持并行操作），也不对SSPBUF 进行写操作（或者禁止向SSPBUF写）。

串行接收/发送缓冲器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPBUF(F6CH)								
R/W	R/W							
RESET	XXXX XXXX							

地址寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSPADD(F6BH)								
R/W	R/W							
RESET	0000-0000							

在主机模式下，为波特率设置寄存器。

16 程序脉冲发生器 PPG

16.1 概述

SD3004 有 2 个程序脉冲发生器，8 位和 16 位。程序脉冲发生器可以根据用户设置输出不同占空比的脉冲波形。

脉冲输出脚和端口复用，8 位 PPG 和 P16 复用；16 位 PPG 和 P17 复用。

相关寄存器：

名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPG_CON	ppg_en						ppg_sel[1]	ppg_sel[0]
PPG_SETH	8 位 PPG 高电平时间设置寄存器							
PPG_SETL	8 位 PPG 周期时间设置寄存器							
PPG2CON	ppg2_en						ppg2s[1]	ppg2s[0]
PPG2H_H	16 位 PPG 的高电平时间的高字节							
PPG2H_L	16 位 PPG 的高电平时间的低字节							
PPG2L_H	16 位 PPG 的周期时间的高字节							
PPG2L_L	16 位 PPG 的周期时间的低字节							

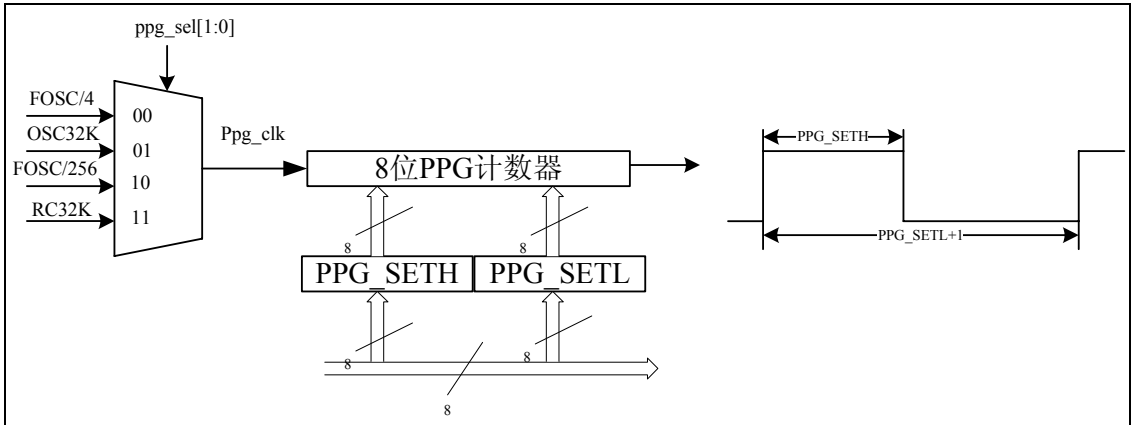


图 16-1 8 位 PPG 结构框图

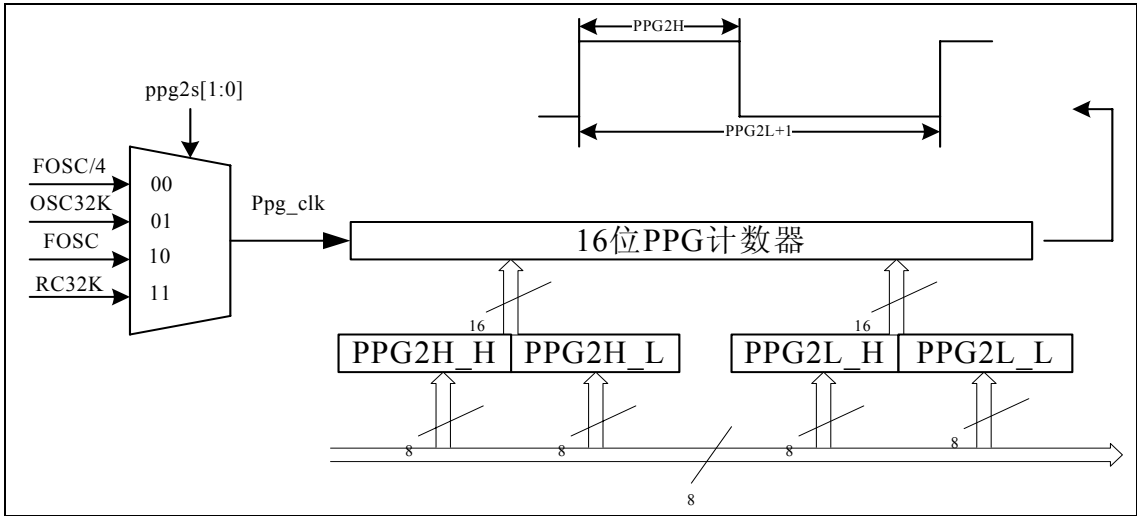


图 16-2 16 位 PPG 结构框图

16.2 寄存器说明

8 位 PPG 控制寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPG_CON (F74H)	ppg_en	--	--	--	--	--	ppg_se l[1]	ppg_se l[0]
R/W	R/W	--	--	--	--	--	R/W	R/W
Reset	0	--	--	--	--	--	0	0

ppg_en: PPG 使能位，高电平使能

ppg_sel[1:0]: 频率选择控制位:

00	FOSC/4
01	OSC32K
10	FOSC/256
11	RC32K

8 位 PPG 占空比设置寄存器：

高电平时间设置，

可读可写，默认 00H

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPG_SETH (F75H)								
Reset	0	0	0	0	0	0	0	0

周期时间设置：可读可写，默认 00H

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPG_SETL(F76H)								
Reset	0	0	0	0	0	0	0	0

周期为 (PPG_SETL + 1) 时间

16 位 PPG 控制寄存器

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPG2CON(F70H)	ppg2en						ppg2s[1]	ppg2s[0]
R/W	R/W						R/W	R/W
Reset	0						0	0

ppg2en: PPG2 使能位，高电平使能

ppg2s[1:0]: 频率选择控制位：

00	FOSC/4
01	OSC32K
10	OSC
11	RC32K

16 位 PPG 占空比设置寄存器：

高电平时间设置寄存器，可读可写，默认 0000H

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPG2H_H(F69H)								
Reset	0	0	0	0	0	0	0	0
PPG2H_L(F6AH)								
Reset	0	0	0	0	0	0	0	0

周期时间设置寄存器：可读可写，默认 0000H。

寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PPG2L_H(F86H)								
Reset	0	0	0	0	0	0	0	0
PPG2L_L(F87H)								
Reset	0	0	0	0	0	0	0	0

周期为 (PPG2L + 1) 时间

17 工作模式

17.1 工作模式概述

提供三种工作模式让用户可以在运行速度与功耗之间做最佳选择，三种模式分别为：正常工作模式、待机模式、休眠模式。

17.1.1 正常工作模式

正常工作模式主要指 CPU 和芯片外围模块根据用户设置正常工作，功耗最大。

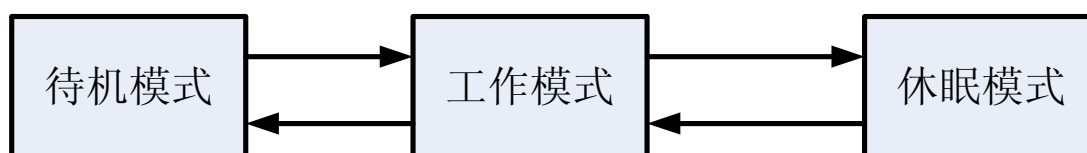
17.1.2 待机模式

待机模式功耗比较小，主要指 CPU 停止工作等待中断或者复位唤醒，芯片外围模块仍正常工作，当外围模块产生中断事件将会唤醒 CPU。中断唤醒 CPU 从待机模式转为正常工作模式，程序从待机停止时的地址执行，复位信号将 CPU 唤醒，程序从地址起始处开始执行。

17.1.3 休眠模式

休眠模式最节省功耗，CPU、内部 RC 振荡，外部晶振将停止工作，外围模块中使用这些时钟源的也将停止工作。只有外部中断和复位可以唤醒 CPU 进入正常工作模式。注意在此模式下，因此为了让芯片功耗最低，不用于唤醒芯片的外围模块功能需要关闭。

17.2 正常工作中模式转换



各种模式的转换如上图所示，待机模式与工作模式可以相互转换，休眠模式也在工作模式可以相互转换，但待机模式与休眠模式不能相互转换。从工作模式到待机模式和休眠模式转换通过指令实现，从待机模式和休眠模式到工作模式通过唤醒中断实现。

1) 工作模式到待机/休眠模式的转换

正常工作中的三种模式转换通过 sleep 指令结合 OSCCON 寄存器中的 idlen 位来实现。需要注意的是，由工作模式进入休眠模式与待机模式有些外设必须先要在程序中关断。

指令：

当 idlen=0，运行 sleep 时进入待机模式；

当 idlen=1，运行 sleep 时进入休眠模式；

2) 待机模式到工作模式转换的中断资源

➤ 所有可用中断源

....

3) 休眠模式到工作模式转换的中断资源

➤ 外部中断 INT0

➤ 外部中断 INT1

18 测试模式说明

SD3004 有四种测试模式：OTP 串行烧录模式、OTP 并行烧录模式、外部灌指令模式和查看测试信号模式。

设置 VPP 引脚电压在不等于 5V 的情况下，配合 P10,P11 脚的信号分别进入 4 种不同的测试模式。

18.1 OTP 串行烧录模式

用户程序通过 OTP 串行烧录，VPP 引脚在烧录时应该接 12.5V 的烧录电压，正常工作时连接 VDD。

在内部复位前，将 P11 和 P10 脚接低电平，VPP 引脚接 12.5V 的烧录电压，进入串行烧录模式。在该测试模式下，使用 SPI 串口进行 OTP 的烧录和读取。SD3004 作为从机使用。

18.1.1 串行烧录管脚

芯片管脚	OTP 烧录管脚	端口方向
P10	SCK	IN
P11	SDI	IN
SEG0/P20	SDO	OUT
VPP	VPP	IN
COM0/P54	1M 时钟	IN

OTP 烧录时钟 SCK 脚，空闲时保持低电平。在通信过程中，SD3004 在 SCK 下降沿采样数据。在 SCK 上升沿发送数据。

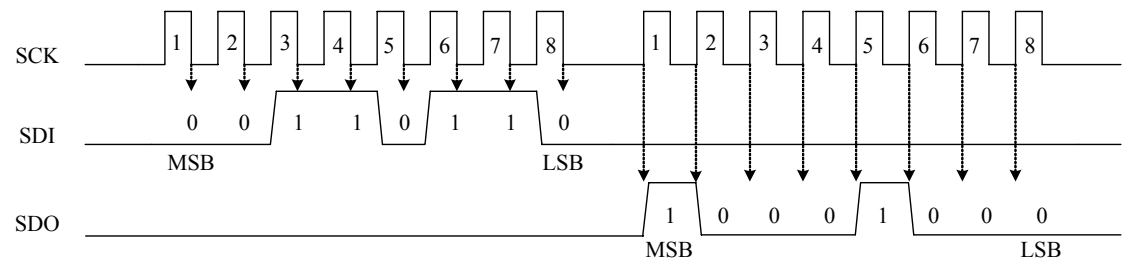
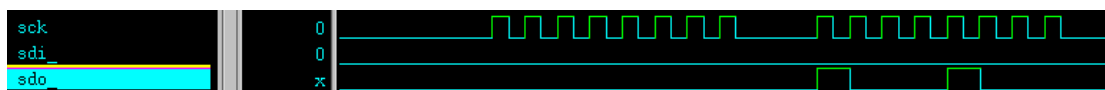


图 17-1 SPI 通信格式

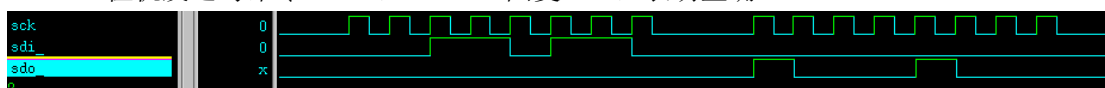
图 17-1 为上位机发送 36H，SD3004 回复 88H，在通信过程中，大部分时间为半双工方式。

18.1.2 串行烧录时序

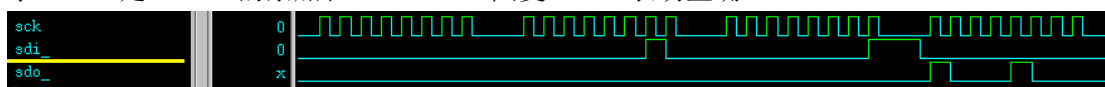
(1) 上位机发送开始命令:00H，SD3004 回复 88H，表明正确（若上位机发生错误，则回复 55H）



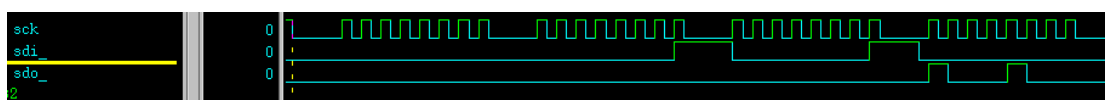
(2) 上位机发送写命令：36H，SD3004 回复 88H，表明正确。



(3) 上位机发送首地址：0002H（分 2 次，先发送高字节 00H，再发送低字节 02H）和校验字 01H（是 0002H 的累加和）。SD3004 回复 88H，表明正确。



(4) 上位机发送要烧录的数据的个数：0001H（分 2 次，先发送高字节 00H，再发送低字节 01H）和校验字 01H（是 0001H 的累加和）。SD3004 回复 88H，表明正确。

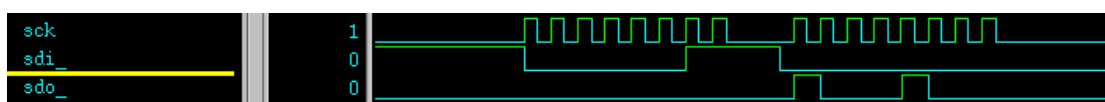


(5) 上位机发送要烧录的数据：0015H（分 2 次，先发送高字节 00H，再发送低字节 15H）和 2 次校验字 03H（0015H 的累加和）。

若第二次发送校验字的同时，SD3004 回复 00H，表明校验字正确，开始等待烧录，等待时间 400us；若校验字错误，在第二次发送校验字的同时，SD3004 回复 55H，需要上位机重新发送。

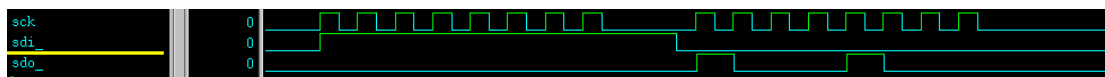


等待时间结束后，上位机再发一次校验字 03H，从机回复 88H 表明烧录完成，可以进行下一步操作。若仍回复 00H，则表明还没有烧录完成。



若有多个数据要烧录，重复步骤（5）

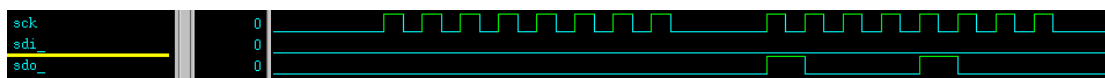
(6) 数据烧录完成后，上位机发送结束命令：FFH。SD3004 回复 88H 表明正确。



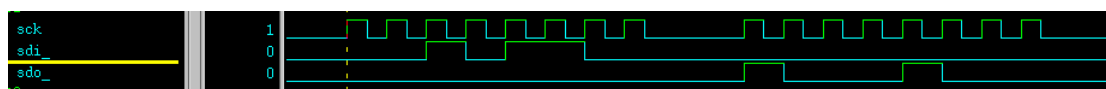
在烧录多个数据的情况下，SD3004 会自动计算完成烧录多少次，如果在不正确的时候发送结束命令，SD3004 回复 55H。

18.1.3 串行读取时序

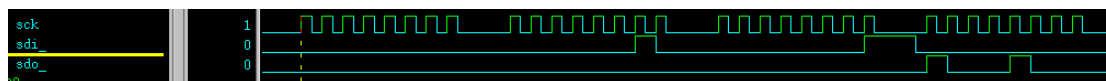
(1) 上位机发送开始命令:00H，SD3004 回复 88H，表明正确（错误回复 55H）



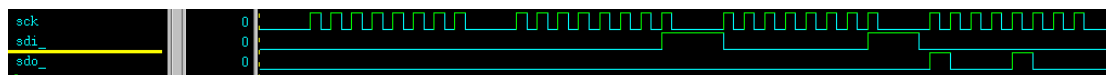
(2) 上位机发送读命令：2CH，SD3004 回复 88H，表明正确。



(3) 上位机发送首地址：0002H（分 2 次，先发送高字节 00H,再发送低字节 02H）和校验字 01H（0002H 的累加和），SD3004 回复 88H，表明正确。



(4) 上位机发送要读取数据的个数：0001H（分 2 次，先发送高字节 00H，再发送低字节 01H）和校验字 01H(0001H 的累加和)，SD3004 回复 88H，表明正确。

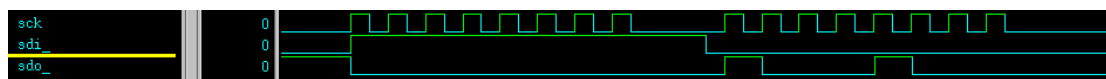


(5) SD3004 发送数据 0015H（分 2 次，先发送高字节 00H，再发送低字节 15H）和校验字 03H（0015H 的累加和）



若有多组数据要读取，重复步骤（5）

(6) 数据读取完成后，上位机发送结束命令 FFH，SD3004 回复 88H，表明正确。



18.2 并行烧录

在内部复位前，P11 和 P10 引脚接高电平，VPP 引脚接 12.5V 的烧录电压，进入并行烧录模式。

在并行烧录模式，OTP 接口信号被依次引出，由外部直接添加激励驱动烧录 OTP。

18.2.1 并行烧录管脚

	名称	OTP 接口说明		端口方向
		烧录模式	读模式	
P57	Progb	PROGB	未使用，保持高电平	I
P56	PWE	未使用，保持低电平	READB 的反	I
P55	PCE	CEB	CEB	I
P54	PA_IO	AD	AD	I
P41	TM1	一般使用设为 0	一般使用设为 0	I
P40	TM0	一般使用设为 0	一般使用设为 0	I
P37	Irdata15	DIN[15]	DOUT[15]	I/O
P36	Irdata14	DIN[14]	DOUT[14]	I/O
P35	Irdata13	DIN[13]	DOUT[13]	I/O

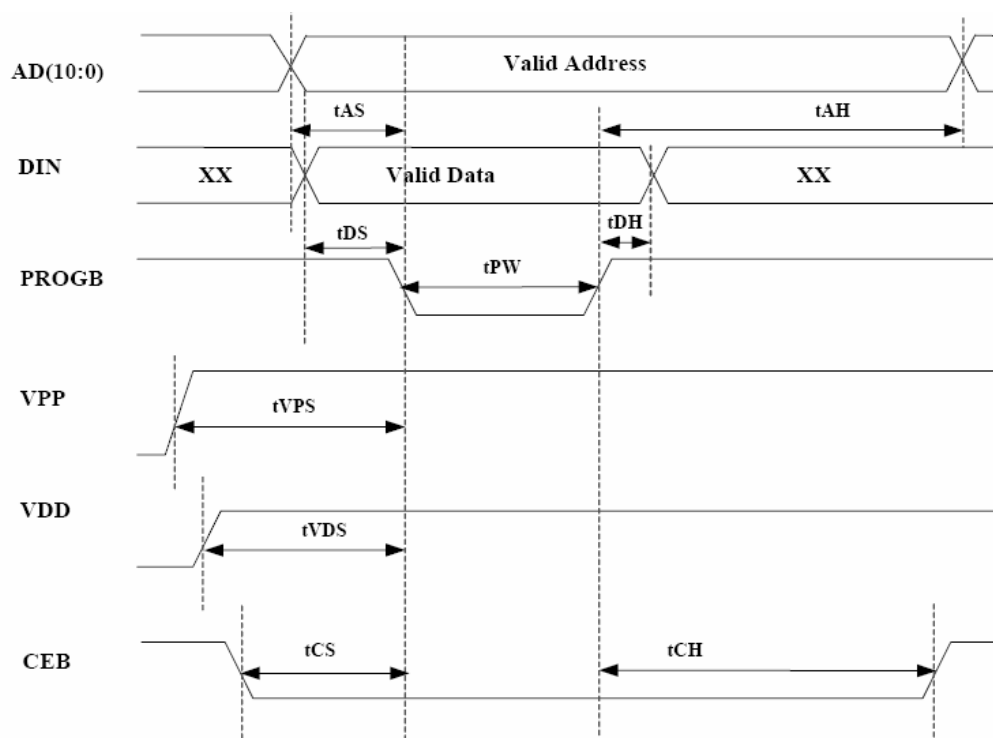
P34	Irdata12	DIN[12]	DOUT[12]	I/O
P33	Irdata11	DIN[11]	DOUT[11]	I/O
P32	Irdata10	DIN[10]	DOUT[10]	I/O
P31	Irdata9	DIN[9]	DOUT[9]	I/O
P30	Irdata8	DIN[8]	DOUT[8]	I/O
P27	Irdata7	DIN[7]	DOUT[7]	I/O
P26	Irdata6	DIN[6]	DOUT[6]	I/O
P25	Irdata5	DIN[5]	DOUT[5]	I/O
P24	Irdata4	DIN[4]	DOUT[4]	I/O
P23	Irdata3	DIN[3]	DOUT[3]	I/O
P22	Irdata2	DIN[2]	DOUT[2]	I/O
P21	Irdata1	DIN[1]	DOUT[1]	I/O
P20	Irdata0	DIN[0]	DOUT[0]	I/O

注：

（1）AD 位，在并行处理中，AD 给一个时钟信号，数字内部将根据该时钟信号的翻转，对 OTP 接口的 AD[10:0]从 000H 地址开始进行加一动作（上升沿加一）。地址只能顺序加一操作。

（2）DIN 和 DOUT 的选择根据 PWE 的电平选择。PWE=1,选择输出，DOUT 有效。PWE=0,选择输入，DIN 有效。

18.2.2 并行烧录时序

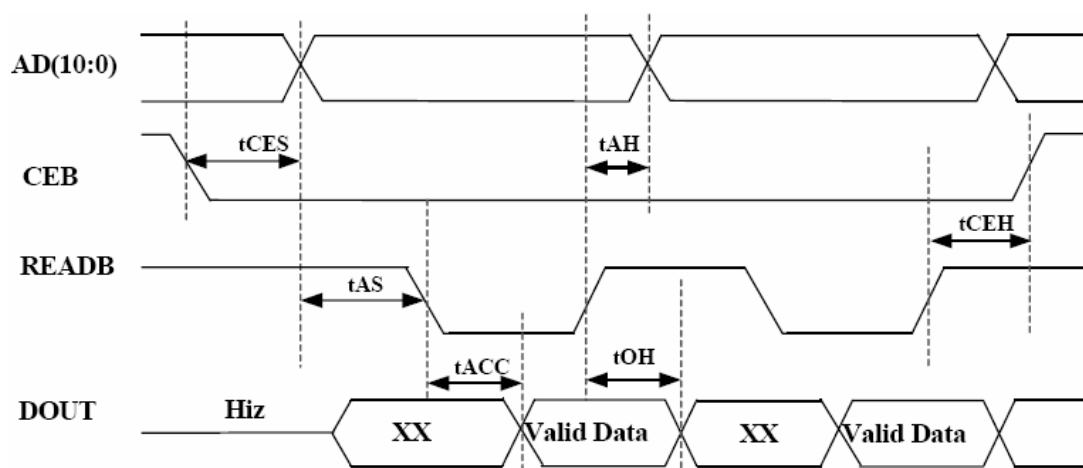


Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Address Setup Time	tAS		1			us
CEB Setup Time	tCS		1			us
Data Setup Time	tDS		1			us
CEB Setup Time	tCH		1			us
Address Hold Time	tAH		1			us
Data Hold Time	tDH		1			us
VPP Setup Time	tVPS		1			us
VDD Setup Time	tVDS		1			us
Program Pulse Width ^{Note}	tPW		100		800	us

操作顺序：

- (1) 各控制信号初始态为高电平。PWE 为低电平。IRDATA 根据 PWE 的设置输入状态。
Vpp 加高压
- (2) 将 CEB 设为低电平。
- (3) 控制 AD 时钟从低电平翻转到高电平。
- (4) 给 DIN 并行有效数据。
- (5) PROGB=0 并保持 100us 以上。
- (6) 将 PROGB=1，结束一次烧录。
- (7) 将 AD 从高电平翻转到低电平。若烧录多次，重复 (3) 到 (7)

18.2.3 并行读取时序



时序要求:

Table3-4 Read characteristics (VDD=2.3V to 5.5V, Ta=-40 to 85°C, load of DOUT: 0.5pF)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
CEB to Address setup time	tCES	VDD=2.3~5.5V	2			us
Address to READB setup time	tAS	VDD=3.6~5.5V	50			ns
		VDD=2.7~3.6V	100			ns
		VDD=2.3~2.7V	100			ns
READB to output delay	tACC	VDD=3.6~5.5V			50	ns
		VDD=2.7~3.6V			100	ns
		VDD=2.3~2.7V			500	ns
CEB to READB hold time	tCEH	VDD=2.3~5.5V	0			
Address hold time from READB	tAH	VDD=2.3~5.5V	0			
Output hold time from READB	tOH	VDD=2.3~5.5V			0	ns

操作顺序:

- (1) 各控制信号初始状态为高电平。PWE 为低电平。VPP 给电源电压。
- (2) 将 CEB 设为低电平。
- (3) 设置 AD 时钟从高电平翻转到低电平。
- (4) 设置 READB=0, 即 PWE 为高电平。此时 IRDATA 口自动设置为输出状态。
- (5) 最久 500ns 后读到数据。
- (4) 将 READB 恢复高电平, 即 PWE 为低电平。此时 IRDATA 口自动恢复为输入状态。结束一次读操作。在读操作期间, 保证 IRDATA 不接信号输入。
- (5) 设置 AD 时钟从低电平翻转到高电平

18.3 外部灌指令模式

在内部复位前, P11 接高电平, P10 接低电平, VPP 引脚接低电平, 进入外灌指令模式。

在该模式下, 在指令时钟 stclk4 的每个上升沿, 直接将 16 位指令灌入, 进行执行。内部地址自动进行加一操作, 因此不能执行跳转指令。

18.3.1 外灌指令接口

SD3004 端口	说明	方向
P54	系统时钟 TEST_OSC	I
P40	指令时钟 stclk4	O
P37	Ext_Irdata15	I
P36	Ext_Irdata14	I
P35	Ext_Irdata13	I
P34	Ext_Irdata12	I
P33	Ext_Irdata11	I
P32	Ext_Irdata10	I
P31	Ext_Irdata9	I
P30	Ext_Irdata8	I
P27	Ext_Irdata7	I
P26	Ext_Irdata6	I
P25	Ext_Irdata5	I
P24	Ext_Irdata4	I
P23	Ext_Irdata3	I
P22	Ext_Irdata2	I
P21	Ext_Irdata1	I
P20	Ext_Irdata0	I

18.4 查看测试信号模式

在内部复位前，P11 接低电平，P10 接高电平，VPP 引脚接低电平，进入查看测试信号模式。下表为可以查看的信号列表。

SD3004 引脚	测试信号	方向	说明
P57	lvd_4o6v	O	4V 的低压检测标志位
P56	osc	O	系统时钟
P55	reset_b	O	内部复位
P54	Ext_osc	I	上电默认系统时钟源
P53	lvd_2o2v	O	2V 的低压检测标志位
P52	pllosc	O	PLL 时钟输出
P41	rcosc32k	O	低频 RC 震荡时钟输出
P40	Osc32k	O	低频晶振时钟输出
P37	Cic_out [15]	O	计量模块瞬时数据
P36	Cic_out [14]	O	计量模块瞬时数据
P35	Cic_out [13]	O	计量模块瞬时数据
P34	Cic_out [12]	O	计量模块瞬时数据
P33	Cic_out [11]	O	计量模块瞬时数据
P32	Cic_out [10]	O	计量模块瞬时数据

P31	Cic_out [9]	O	计量模块瞬时数据
P30	Cic_out [8]	O	计量模块瞬时数据
P27	Cic_out [7]	O	计量模块瞬时数据
P26	Cic_out [6]	O	计量模块瞬时数据
P25	Cic_out [5]	O	计量模块瞬时数据
P24	Cic_out [4]	O	计量模块瞬时数据
P23	Cic_out [3]	O	计量模块瞬时数据
P22	Cic_out [2]	O	计量模块瞬时数据
P21	Cic_out [1]	O	计量模块瞬时数据
P20	Cic_out [0]	O	计量模块瞬时数据

19. 典型应用

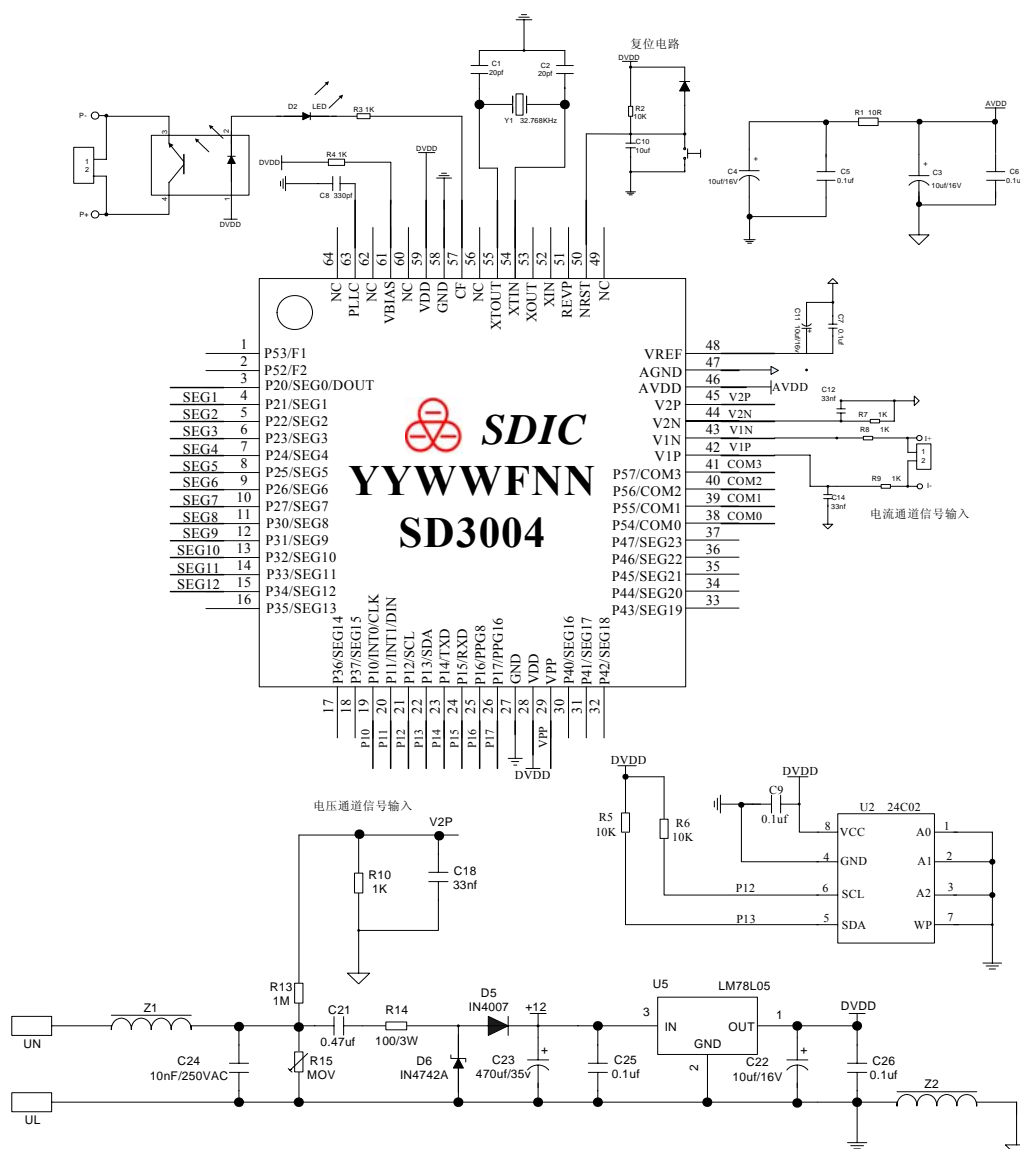


图18-1 典型应用图

20 电气参数

表 19.1 最大绝对值

标识	参数	最小值	最大值	单位
T_A	环境温度	-40	+85	°C
T_S	储存温度	-55	+150	°C
V_{DD}	供电电压	-0.2	+7	V
V_{pp}	烧录电压	-0.2	+13	V
V_{IN}, V_{OUT}	直流输入、输出	-0.2	$V_{DD}+0.3$	V
T_L	回流焊温度曲线		Per IPC/JEDECJ-STD-020C	°C

注意:

1. CMOS 器件易被高能静电损坏, 设备必须储存在导电泡沫, 注意避免工作电压超出范围。
2. 在插拔电路前请关闭电源。

表 19.2 电气参数 ($V_{DD}=5V$, $AV_{DD}=5V$)

符号	参数名称	最小值	典型值	最大值	单位	条件/备注
PLLOSC	工作频率 1	--	3.604	--	MHz	锁相环时钟工作
OSC32K	工作频率 2	--	32.768	--	KHz	外部晶振时钟
HOSC	工作频率 3	--	3.58	--	MHz	外部高频晶振时钟
RC32K	工作频率 4	16	--	--	KHz	内部 RC 振荡
FOSC	工作时钟	--	3.58	--	MHz	工作频率 1 到 4 或者他们的分频时钟
VDD		2.4	--	5.25	V	数字电源
AVDD		4.75	5	5.25	V	模拟电源
IDD1	工作电流 1	--	5	--	mA	工作频率 3.58MHz, MCU 工作, 计量部分工作, 液晶显示,
IDD2	工作电流 2	--	15	30	uA	采用 32KHz/2 时钟工作, MCU 进入 SLEEP 状态, LCD 维持显示, 计量部分不工作
IDD3	工作电流 3	--	--	1	uA	所有模块停止工作
VIL	输入低电压	--	--	0.3VDD		PT2/PT3/PT4/PT5
		--	--	0.2VDD		PT1
		--	--	0.2VDD		NRST
VIH	输入高电压	--	--	0.7VDD		PT2/PT3/PT4/PT5
		--	--	0.8VDD		PT1
		--	--	0.8VDD		NRST
Rpu	上拉电阻	50K	--	100K	欧姆	PT1/NRST
VOL	输出低电平	--	--	0.3VDD	V	
VOH	输出高电平	0.7VDD	--	--	V	

VPP	编程电压	11.75	12	12.25	V	
VREF	基准电压范围	2.3	2.5	2.7	V	
T_VREF	基准随温度变化温漂	--	30	60	PPM/度	-40 度到+85 度
I_ACCU	电压/电流通道的测量精度	--	0.5	1	%full scale	
RANGE2	通道 2 输入信号范围	--	--	650	mV	
RANGE1	通道 1 输入信号范围	--	--	450	mV	

21 封装规格

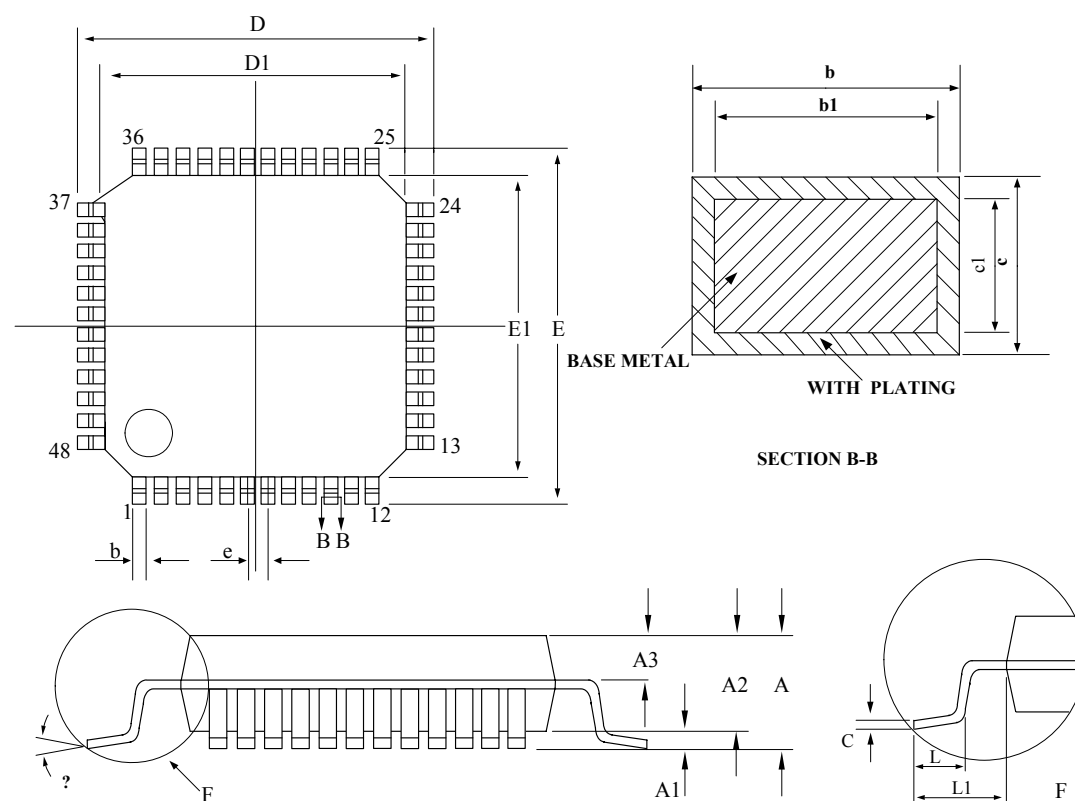


图 21-1. SD3003 封装外形图

Dimensions: mm

Symbol	Min.	Nom.	Max.
A	—	—	1.60
A1	0.05	—	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19	—	0.27
b1	0.18	0.20	0.23
c	0.13	—	0.14
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
e	0.50BSC		
L	0.40	—	0.65
L1	1.00BSC		
Θ	0	—	7

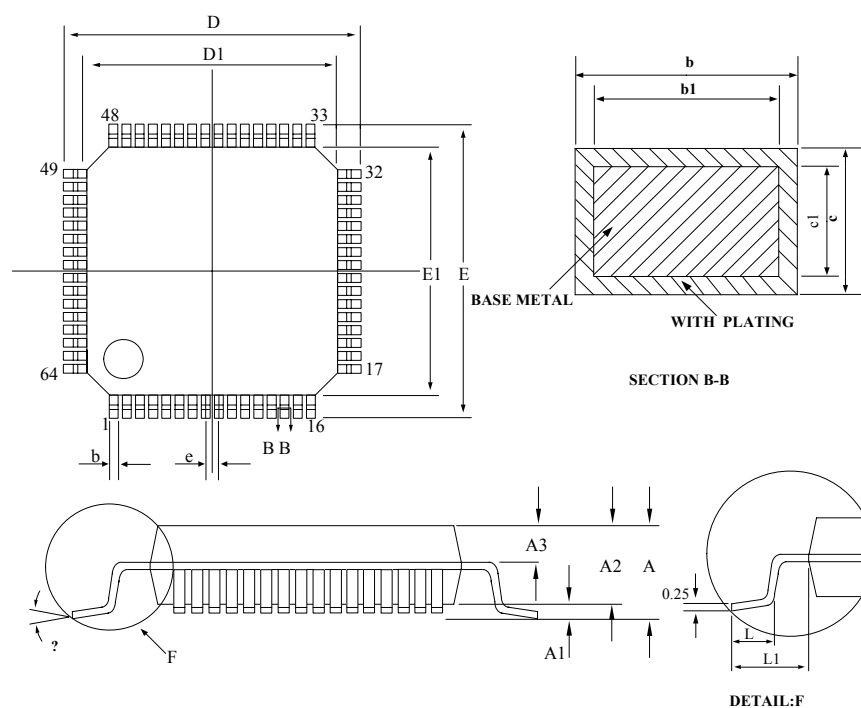


图 21-2. SD3004 封装外形图

Dimensions: mm

Symbol	Min.	Nom.	Max.
A	—	—	1.60
A1	0.05	—	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19	—	0.27
b1	0.18	0.20	0.23
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	6.90	7.00	7.10
e	0.50BSC		
L	0.45	—	0.75
L1	1.00BSC		
Θ	0	—	7