

EN8F5113 数据手册

内置 EEPROM 和 Flash
带 12Bit ADC/PWM/SPI/I2C/EUART 的
低功耗增强型 8051 微控制器

(Version 2.9)

目录

1. 产品概述.....	8
2. 功能.....	8
3. 功能框图.....	11
4. 管脚说明.....	12
4.1. 管脚封装图 (TSSOP20).....	12
4.2. 管脚封装图 (QFN20).....	13
4.3. 管脚封装图 (SOP16).....	14
4.4. 管脚封装图 (QFN16).....	14
4.5. 管脚封装图 (DFN8).....	15
4.6. 管脚功能说明.....	15
5. 特殊功能寄存器.....	18
5.1. SFR (特殊功能寄存器) 汇总.....	18
5.2. SFR 映像图.....	19
6. CPU.....	20
6.1. 特性.....	20
6.2. 存储器结构.....	22
6.2.1. SRAM.....	22
6.2.2. Flash 存储器.....	22
7. 系统时钟和复位系统.....	24
7.1. 系统时钟.....	24
7.1.1. 寄存器.....	24
7.1.2. 内置 RC 时钟振荡器.....	26
7.1.3. 外部晶体振荡器.....	26
7.2. 复位系统.....	27
7.2.1. 看门狗复位.....	27
7.2.2. 低电压复位.....	28
7.2.3. 内置高速比较器.....	29

8. I/O 端口.....	30
8.1. 特性.....	30
8.2. 端口状态控制.....	30
8.3. 端口上/下拉电阻.....	31
8.4. 端口复用.....	32
8.4.1. 端口复用控制寄存器.....	32
8.4.2. 端口复用说明.....	35
8.5. 端口中断.....	36
9. 定时器.....	39
9.1. 定时器 0, 1.....	39
9.1.1. 定时器 0, 1 工作方式.....	39
9.1.2. Timer 模块图.....	39
9.1.3. 定时器 0, 1 寄存器.....	40
9.2. RTC 计数器.....	41
10. 脉宽调制模块 (PWM).....	44
10.1. PWM 结构框图.....	46
10.2. PWM 模块控制寄存器.....	47
10.2.1. PWMCON0 (D4H).....	47
10.2.2. PWMCON1 (D5H).....	47
10.2.3. PWMMASKEN (D6H).....	48
10.2.4. PWMMASKDATA (CCH).....	48
10.2.5. PWMINTCON (CDH).....	49
10.2.6. PWMINTF (A4H).....	50
10.2.7. PWMFBCON0 (A5H).....	50
10.2.8. PWMFBCON1 (A6H).....	51
10.2.9. PWMDTCN (9EH).....	51
10.2.10. PWMDTCYC (96H).....	52
10.2.11. PWMPH/PWMPH (CE/C4H).....	52
10.2.12. PWMODL/PWMODH (C5/C6H).....	53
10.2.13. PWM1DL/PWM1DH (BC/BDH).....	53
10.2.14. PWM2DL/PWM2DH (BE/B4H).....	54
10.2.15. PWM3DL/PWM3DH (B5/B6H).....	54
10.2.16. PWM4DL/PWM4DH (AB/ACH).....	54
10.2.17. PWM5DL/PWM5DH (AD/AEH).....	55

11. 蜂鸣器.....	56
12. 增强型串口 (EUART0/1)	57
12.1. 特性.....	57
12.2. 寄存器.....	57
12.3. 数据接收&发送.....	60
12.4. EUART0 工作模式.....	60
12.5. EUART1 工作模式.....	63
12.6. 多机通讯.....	64
12.7. 波特率.....	64
13. I2C 控制器.....	65
13.1. I2C Slave 控制器.....	65
13.1.1. 功能框图.....	66
13.1.2. 控制寄存器.....	66
13.2. I2C Master 控制器.....	71
13.2.1. 功能框图.....	72
13.2.2. 寄存器.....	72
13.3. I2C 帧数据格式.....	77
13.4. 应答.....	79
13.5. 时钟.....	79
13.5.1 时钟系统.....	79
13.5.2. I2C 时钟.....	80
13.6. 主模式：传输.....	81
13.6.1 初始化.....	81
13.6.2 起点.....	81
13.6.3 数据传输.....	82
13.6.4 停止重复启动.....	82
13.6.5 传输错误.....	82
13.7. 主模式：接收.....	83
13.7.1 初始化.....	83
13.7.2 起点.....	83
13.7.3 接收.....	83
13.7.4 停止重复启动.....	83
13.7.5 接收错误.....	84

14. SPI.....	85
14.1. 功能框图.....	85
14.2. 特性.....	85
14.3. SPI 寄存器.....	86
14.3.1. SPICR 寄存器.....	86
14.3.2. SPIDR 寄存器.....	88
14.3.3. SPISR 寄存器.....	88
14.3.4 清除 SPSR's 标志.....	89
14.4 SPI master 模式.....	89
14.4.1 I/O pin 配置.....	89
14.4.2 SPI 控制寄存器配置.....	89
14.4.3 Master 模式数据传输.....	89
14.5 SPI slave 模式.....	90
14.5.1 I/O pin 配置.....	90
14.5.2 SPI 控制寄存器配置.....	90
14.5.3 Slave 模式下数据传输.....	90
14.6 SPI 传输格式.....	91
14.6.1 CPHA bit 复位时.....	91
14.6.2 CPHA bit 置位时.....	92
14.7 SPI 系统错误.....	93
14.7.1 写冲突.....	93
14.7.2 Master mode 故障错误.....	93
14.7.3 Overrun condition.....	93
15. 模数转换(ADC).....	95
15.1. 特性.....	95
15.2. ADC 模块图.....	95
15.3. ADC 寄存器.....	95
15.4. ADC 工作参考流程.....	99
16. IAP 控制单元 (Data EEPROM).....	100
16.1. IAP 相关控制寄存器.....	101
16.1.1. IAPCTRL(C8H): IAP 控制寄存器.....	101
16.1.2. IAPLOCK(F5H): IAP 解锁寄存器.....	102
16.1.3. IAPDATA(EDH): IAP 数据寄存器.....	102

16.1.4. IAPADDRH (DDH): IAP 高位地址寄存器.....	102
16.1.5. IAPADDRL (E5H): IAP 低位地址寄存器.....	103
16.2. IAP 操作流程.....	103
17. 省电模式.....	104
17.1. 特性.....	104
17.2. 寄存器.....	104
17.3. Idle Mode.....	105
17.4. Sleep Mode.....	105
17.4. DeepSleep Mode.....	105
18. 中断.....	107
18.1. 特性.....	107
18.2. 寄存器.....	107
18.3. 中断汇总.....	108
19. 96 位序列号 (UID).....	109
20. 指令集.....	110
20.1. 指令操作数说明.....	110
20.2. 算术操作指令.....	110
20.3. 逻辑操作指令.....	111
20.4. 数据传送指令.....	112
20.5. 控制程序转移指令.....	113
20.6. 位操作指令.....	114
21. 电气特性.....	115
21.1. 极限参数.....	115
21.2. 直流电气特性.....	115
21.3. ADC 电气特性.....	116
21.4. 内建高频 IHRC 电气特性.....	116
21.5. 内建 LDO 电气特性.....	116

22. 封装信息.....	117
规格更新记录.....	118

1. 产品概述

EN8F5113 系列是通用型 1T 8051 Core MCU。在同样的系统时钟下，比传统的 8051 运行更快速，性能更优越，指令代码完全兼容传统 8051；

保留了标准 8051 的主要特性，包括 256 字节 RAM 和 2 个 16 位定时器；芯片还集成了 1K 字节外部扩展 RAM，可灵活配置的 EEPROM，RTC，和 8K Flash，并且可实现在应用中编程（IAP）功能；

集成了 10 路 12 位带数字比较功能的 ADC，两个自带波特率发生器的串口，1 路 8Mbit/s 主从可配的 SPI 接口；6 路 16 位分辨率的 PWM 等其他丰富的外设资源；EN8F5113 系列内部集成了 RC 振荡器（IHRC），看门狗定时器，上电复位和低电压复位等功能，从而省掉了外部晶振以及复位等外围电路；支持多种低功耗工作模式；

EN8F5113 系列是一颗高性能，低成本，高可靠性的 MCU。合理的资源配置、极佳的低功耗与超强抗干扰性能，BOM 精简，nA 级别的待机功耗，是一颗高性能，低成本，高可靠性的 MCU。

2. 功能

- ◆ 基于 1T 8051 指令流水线结构 8 位单片机
- ◆ FlashROM:
 - FlashROM 最大至 8K 字节；
 - 读/写/擦除速度快：读（40nS）；写（Max. 20uS）；擦除（Page:Max. 2mS；Macro:Max. 10mS）；
 - 功耗低：读（150uA/Mhz）；写（TYP. 5mA）；擦除（TYP. 5mA）；Standby（TYP. 1uA）；
 - 产品耐久性高：
最少 100000 个程序/擦除周期；最少 40 年数据保留期；
 - 固定 1024 字节 EEPROM，也可灵活扩展 1024/512/256/0 字节 EEPROM；
 - 支持 ISP（在系统编程）和 IAP（在应用编程）；
 - Flash 数据硬件安全电路保护机制；
- ◆ RAM: Idata 256 字节，Xdata 1024 字节；
- ◆ 序列号（UID）：支持 96bit 唯一序列号；
- ◆ 超宽工作电压：1.8~5.5V ；
- ◆ 超强的 ESD/Latch up 性能：
HBM pass Level: ±8KV (Min.)；
CDM pass Level: ±900V (Min.)；
Latch up pass Trigger current: ±500mA (Min.)；
- ◆ 时钟源：
 - 内部高频 RC 振荡器（IHRC）：
16MHz（2.0V~5.5V: ±1% @ -25℃~85℃；±2% @ -45℃~125℃）
 - 内部低频 RC 振荡器（ILRC）：128KHz (@25℃，2.0V~5.5V)
 - 外部时钟输入：2~16MHz
 - 晶体谐振器：（LOSC）32768Hz、（HOSC）2-16MHz
 - 多种可配置时钟输出（与 BEEP IO 复用）
- ◆ 17 个双向 CMOS I/O 管脚

- 内建独立控制上拉、下拉电阻
- 输出驱动可选：8mA/12mA/16mA/24mA
- 可选择 Open drain 输出
- 所有 IO 都支持外部中断功能
- 所有 IO 都支持 IDLE、Sleep 模式下的唤醒功能
- 10 个 IO 支持作为 1/2 BIAS 的软件 LCD COM/SEG 输出
- ◆ 2 个 16 位定时器 T0/T1
- ◆ 低功耗 RTC
 - 与外部 LOSC 配合实现 RTC 功能
 - 带有报警中断
 - 定时中断唤醒
- ◆ 1 路主从可配置的 SPI，接口时钟速度为 8M ($f_{sys}/2$)
- ◆ 1 路主从可配置的 I2C 接口
- ◆ 增强性 PWM 输出
 - 16bit 计数器，6 个独立 PWM 输出，周期相同，占空比可灵活配置；
 - 可配置为带死区控制的组合模式，每对的 PWM 输出波形相反互补，插入的死区可配置；
 - 可配置为同步模式、分组模式；
 - 支持中心对齐、边沿对齐模式；
 - One-shot (只有边沿对齐模式支持) 和 Auto-reload 模式；
 - 硬件紧急故障保护：P1.4，ADC 数据比较触发；
 - 灵活的中断配置。
- ◆ 两个增强型串口 EUART0/EUART1
- ◆ 12bit ADC
 - 与 GPIO 复用的 10 通道输入可选；
 - 输入电压范围：0~参考电压；
 - DNL: ± 2 LSB; INL: ± 4 LSB;
 - Up to 500 ksps 转换速率；
 - ADC 工作时钟可选：系统时钟的 2/4/8/16/32/64/128/256 分频；
 - 采样/保持时间可选：4/8/12/20/36/68/132/260（单位：ADC 工作时钟周期）；
 - 灵活的参考电压选择：
 - ◇ 外接基准电压（与 AIN<10>通道复用）
 - ◇ 内建基准电压（1.5V/2V/3V/4V），校正后精度： $\pm 1\%$ @ $-40\sim 85^{\circ}\text{C}$ ，可配置 AIN<3>口 (P13) 输出
 - ◇ 内部电源电压
 - 内置 16 step Offset 校正电路；
 - 支持单通道连续比较模式；
 - 可选择配置内部 4/8 滑动平均输出；
 - 支持可编程上下限阈值的模拟看门狗功能；
 - 低功耗模式可选择，低频下（ADC 工作时钟<1MHz）典型电流<300uA
- ◆ 看门狗定时器（WDT）
- ◆ 预热计数器（用于上电复位&中断唤醒等）
- ◆ 蜂鸣音发生器 (BEEP)

- ◆ 两组输入可选的内置高速比较器（输入通道和 AIN<3:0>复用）
- ◆ 中断源：
 - 定时器 T0/T1
 - RTC 定时中断
 - EUART0/EUART1
 - ADC
 - 可配置比较器中断
 - LVD
 - PWM 周期中断
 - SPI 中断
 - I2C 中断
 - 17 个 IO 中断
 - 10 个 IO 支持 Deep Sleep Mode 下外部唤醒(与 GPIO 复用)
- ◆ 复位源：
 - POR（上电复位）
 - LVD（掉电复位）
 - 看门狗复位
 - PIN Reset
- ◆ 内建低电压检测模块（16 step 可调）
- ◆ 省电模式支持
 - Active Mode: <250 μ A/MIPS;
 - Sleep Mode: TYP <5 μ A; MAX <10 μ A;
 - Deep Sleep Mode: TYP <20nA; MAX <500nA;
- ◆ 封装
最大支持 20Pin 封装。

3. 功能框图

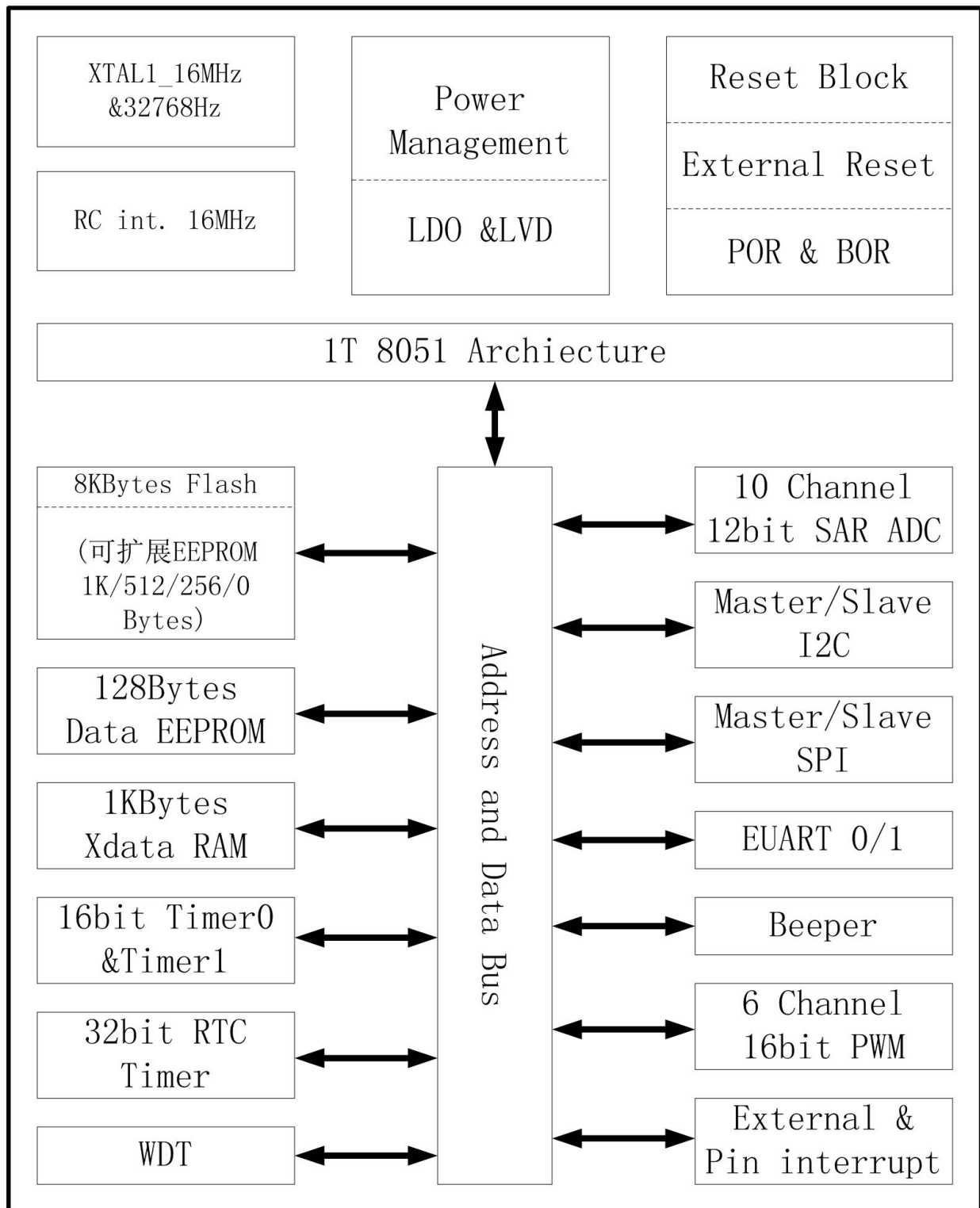


图 3-0

4. 管脚说明

4.1. 管脚封装图 (TSSOP20)

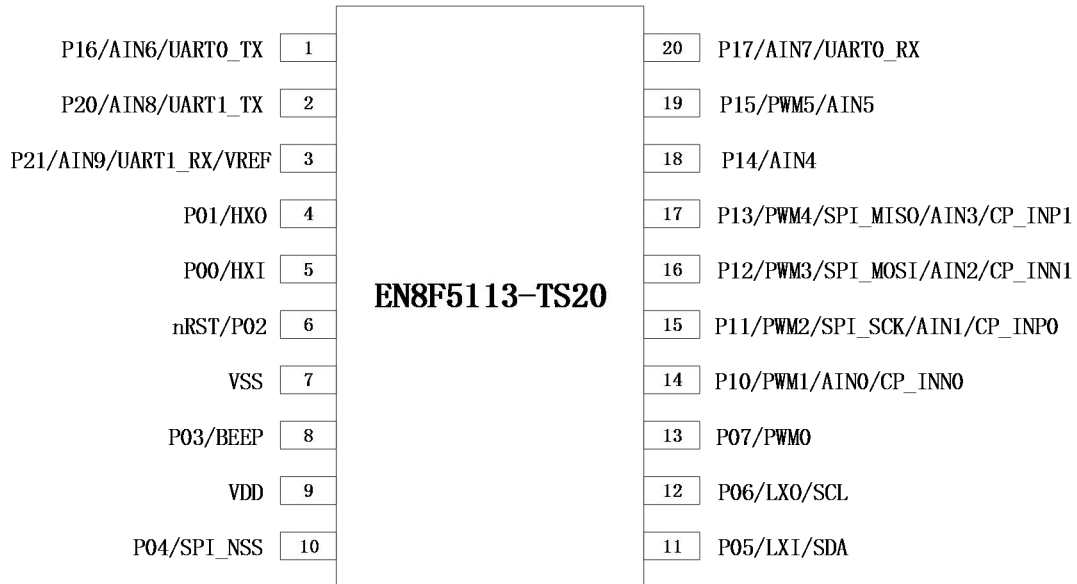


图 4-1

注意: 和 GPIO 复用功能的引脚, 默认功能都是 GPIO. 第二、三、四功能通过相应的寄存器打开.

4.2. 管脚封装图(QFN20)

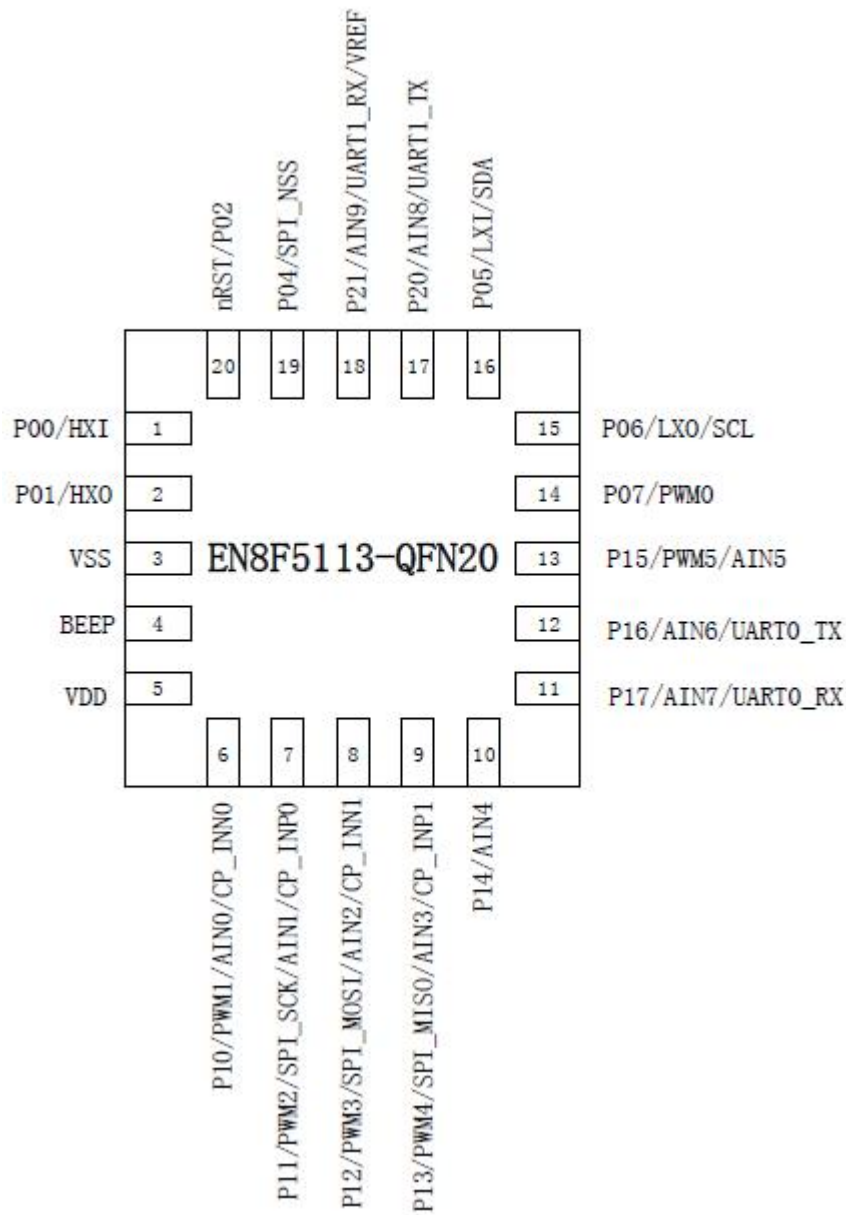


图 4-2

4.3. 管脚封装图(SOP16)

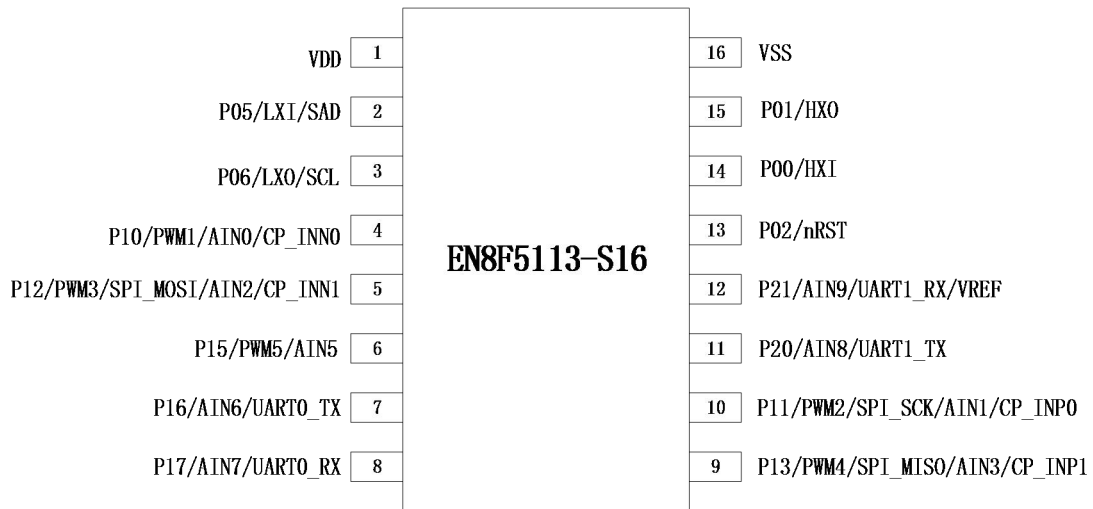


图 4-3

4.4. 管脚封装图(QFN16)

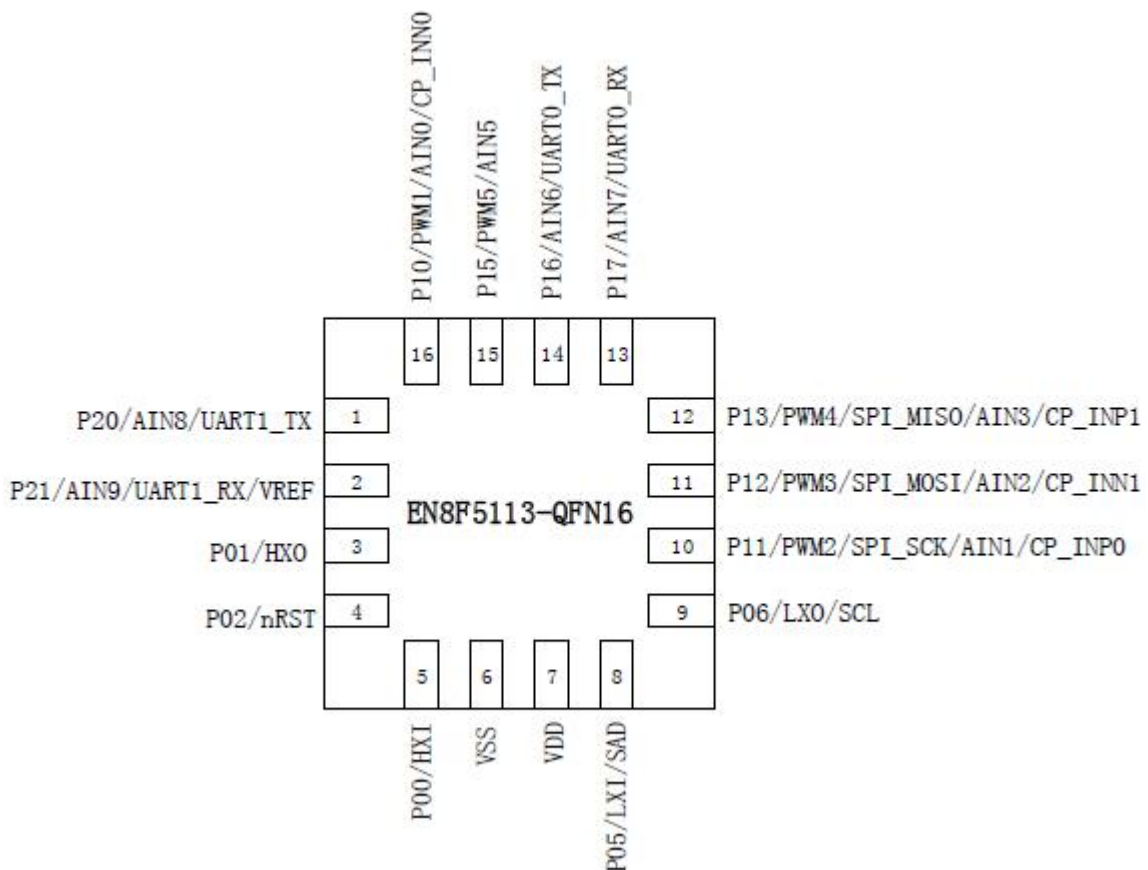
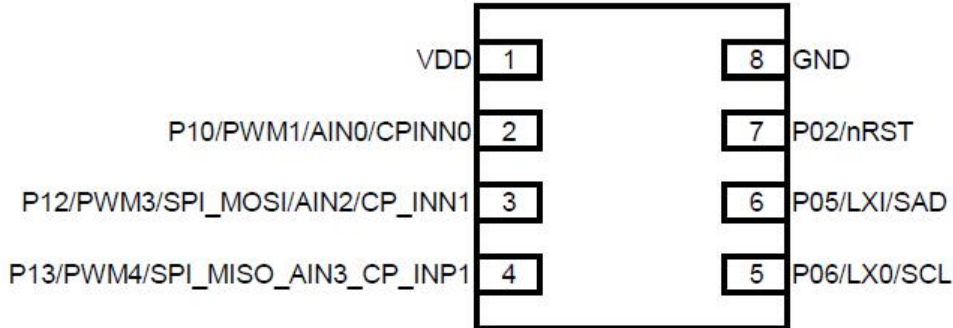


图 4-4

4.5. 管脚封装图(DFN8)

EN8F5113-DFN8-31



EN8F5113-DFN8-32

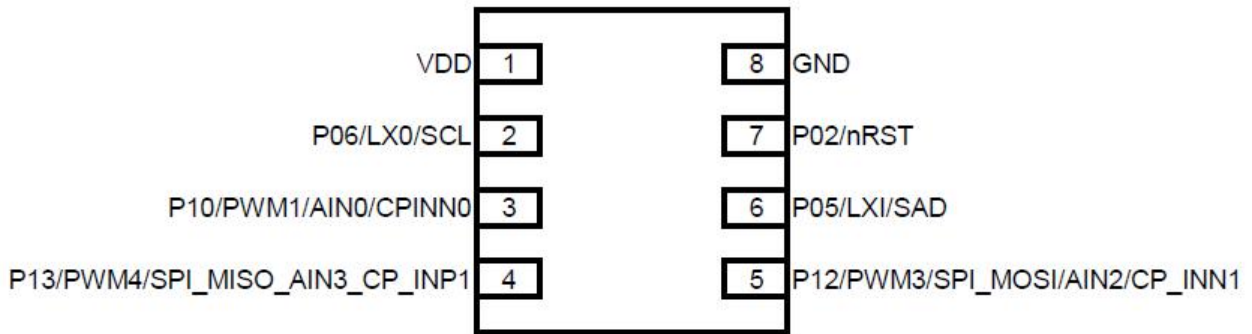


图 4-5

4.6. 管脚功能说明

Pin Name	Pin # TSSOP20	Pin# QFN20	Pin# SOP16	Pin# QFN16	Pin# DFN8	Type	Description
P16 /AIN6 /UART0_TX	1	12	7	14		I/O	P1.6
							ADC 输入 Channel 6
							UART0 TX
							LCD COM/SEG □
P20 /AIN8 /UART1_TX	2	17	11	1		I/O	P2.0
							ADC 输入 Channel 8
							UART1TX
							LCD COM/SEG □

Pin Name	Pin # TSSOP20	Pin# QFN20	Pin# SOP16	Pin# QFN16	Pin# DFN8	Type	Description
P21 /AIN9 /UART1_RX /VREF	3	18	12	2		I/O	P2.1
							ADC 输入 Channel 9
							UART1 RX
							ADC 外部参考输入
nRST /P02	6	20	13	4	7	I/O	Reset Pin ,低电平复位, 内建上拉电阻
							P0.2
P00 /HXI	5	1	14	5		I/O	P0.0
							HOSC 输入
P01 /HXO	4	2	15	3		I/O	P0.1
							HOSC 输出
VSS	7	3	16	6	8	G	Ground
BEEP	8	4				O	BEEP 输出
VDD	9	5	1	7	1	P	Power(1.8~5.5)
P04 /SPI_NSS	10	19				I/O	P0.4
							SPI master/slave select input
P05 /LXI /SDA	11	16	2	8	2	I/O	P0.5
							LOSC 输入
							I ² C 数据
P06 /LXO /SCL	12	15	3	9	3	I/O	P0.6
							LOSC 输出
							I ² C 时钟
P07 /PWM0	13	14				I/O	P0.7
							PWM0 输出
P10 /PWM1 /AIN0 /CP_INN0	14	6	4	16		I/O	P1.0
							PWM1 输出
							ADC 输入 Channel 0 比较器 0 负端输入脚
							LCD COM/SEG 口
P11 / PWM2 /SPI_SCK /AIN1 /CP_INP0	15	7	10	10		I/O	P1.1
							PWM2 输出
							SPI 时钟
							ADC 输入 Channel 1 比较器 0 正端输入脚
							LCD COM/SEG 口
P12 / PWM3 /SPI_MOSI /AIN2 /CP_INN1	16	8	5	11	6	I/O	P1.2
							PWM3 输出
							SPI master 输出/slave 输入
							ADC 输入 Channel 2

Pin Name	Pin # TSSOP20	Pin# QFN20	Pin# SOP16	Pin# QFN16	Pin# DFN8	Type	Description
							比较器 1 负端输入脚 LCD COM/SEG 口
P13 / PWM4 /SPI_MISO /AIN3 /CP_INP1	17	9	9	12	5	I/O	P1.3 PWM4 输出 SPI master 输入/slave 输出 ADC 输入 Channel 3 比较器 1 正端输入脚 LCD COM/SEG 口
P14 /AIN4	18	10				I/O	P1.4 ADC 输入 Channel 4 LCD COM/SEG 口
P15 / PWM5 /AIN5	19	13	6	15	4	I/O	P1.5 PWM5 输出 ADC 输入 Channel 5 LCD COM/SEG 口
P17 /AIN7 /UART0_RX	20	11	8	13		I/O	P1.7 ADC 输入 Channel 7 Uart0 RX LCD COM/SEG 口

5. 特殊功能寄存器

5.1. SFR(特殊功能寄存器) 汇总

EN8F5113 系列包含 128 字节可直接寻址的特殊功能寄存器,分类如下:

CPU 内核寄存器	ACC,B,PSW,SP,DPS,DPL,DPH,DPL1,DPH1
RTC 寄存器	RTCSECOND,RTCMINUTE,RTCHOUR,RTCCON
PORT 寄存器	P0,P1,P2,P0OEN,P1OEN,P2OEN,P0PUN,P1PUN,P2PUN,P0PDN,P1PDN, P2PDN,P0ODN,P1ODN,P2ODN,P0HC0,P1HC0,P2HC0,P0HC1,P1HC1, P2HC1,P0IEN,P1IEN,P2IEN,P0IRQ,P1IRQ,P2IRQ,P12IRQCON, PORTMUX1L,PORTMUX1H,PORTMUX_2,PORTMUX_3,PORTMUX_4.
PMU 寄存器	CLKON,CLKDIV,RSTSTAT,PCON
ANALOG 控制寄存器	LVDLVRCON,LVDCP,ADVREFSEL, ADDH,ADDL,ADCON1,ADCON2, ADCON3,RC16MITUNE,RC16MADJ,ADT,DPWAKE
IAP 控制寄存器	IAPCTRL,IAPLOCK,IAPDATA,IAPADDRL,IAPADDRH
PWM 控制寄存器	PWMCON0,PWMCON1,PWMMASKEN,PWMMASKDATA,PWMINTCON, PWMPH,PWMPH,PWMODL,PWMODH,PWM1DL,PWM1DH, PWM2DL,PWM2DH,PWM3DL,PWM3DH,PWM4DL,PWM4DH, PWM5DL,PWM5DH,PWMINTF,PWMFBCON0,PWMFBCON1,PWMDTCO N,PWMDTCYC
SPI 控制寄存器	SPICR,SPIDR,SPISR
I2CM/S 控制寄存器	MSCON (MCON/STCON) ,MSRXBUF (MRXBUF/SRXBUF) ,MSTXBUF (MTXBUF/STXBUF) ,MPRESC,MSSTAT0 (MSTAT0/SSTAT0) , MSSTAT1 (MSTAT1/SSTAT1) ,MSIEN0 (MIEN0/SIEN0) ,MSIEN1 (MIEN1/SIEN1),MSADDR (MCADDR/SSADDR)
INTERRUPT 控制寄存器	IEN0,IEN1,IP0,IP0H,IP1,IP1H
PHERI 控制寄存器	S0CON,S0BUF,S1CON,S1BUF,S0RELL,S0RELLH,S1RELL,S1RELLH,TCON, TMOD,TL0,TL1,TH0,TH1,BEEP
Global 控制寄存器	UID,TOPCON,EEPROMCON

5.2. SFR 映像图

	可位寻址	不可位寻址							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8H	PORTMUX1L	P00EN	P10EN	P20EN	P012IRQCON	LVDLVRCON	ADVREFSEL	ADT	FFH
F0H	B	POPUN	P1PUN	P2PUN	CLKCON	IAPLOCK	ADDH	DPWAKE	F7H
E8H	PORTMUX1H	POPDN	P1PDN	P2PDN	CLKDIV	IAPDATA	ADDL	ADCON1	EFH
E0H	ACC	POODN	P1ODN	P2ODN	RSTSTAT	IAPADDRL	RC16MITUNE	ADCON2	E7H
D8H	PORTMUX_2	POHC1	P1HC1	P2HC1	EEPROMCON	IAPADDRH	RC16MADJ	ADCON3	DFH
D0H	PSW	POHC0	P1HC0	P2HC0	PWMCON0	PWMCON1	PWMMASKEN	MSCON (MCON/ STCON)	D7H
C8H	IAPCTRL	POIEN	P1IEN	P2IEN	PWMMASKDAT A	PWMINTCON	PWMPPL	MSRXBUF (MRX BUF/SRXBUF)	CFH
C0H	PORTMUX_3	POIRQ	P1IRQ	P2IRQ	PWMPH	PWMDL	PWMDH	MSTXBUF (MTX BUF/STXBUF)	C7H
B8H	IEN1	IP1	SORELH	S1RELH	PWM1DL	PWM1DH	PWM2DL	MPRESC	BFH
B0H	PCON	IP1H	RTCSEC OND	RTCMI NUTE	PWM2DH	PWM3DL	PWM3DH	MSSTAT0 (MST AT0/SSTAT0)	B7H
A8H	IEN0	IP0	SORELL	PWM4DL	PWM4DH	PWM5DL	PWM5DH	MSSTAT1 (MST AT1/SSTAT1)	AFH
A0H	P2	IPOH	RTCHOU R	RTCCON	PWMINTF	PWMFBCON0	PWMFBCON1	MSIEN0 (MIEN 0/SIEN0)	A7H
98H	SOCON	S0BUF		S1CON	S1BUF	S1RELL	PWMDTCN	MSIEN1 (MIEN 1/SIEN1)	9FH
90H	P1	PORTMU X_4	DPS	SPICR	SPIDR	SPISR	PWMDTCYC	MSADDR (MCAD DR/SSADDR)	97H
88H	TCON	TMOD	TLO	TL1	TH0	TH1	UID	TOPCON	8FH
80H	P0	SP	DPL	DPH	DPL1	DPH1	BEEP	LVDCP	87H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

6. CPU

6.1. 特性

◆ CPU 内核寄存器：ACC, B, PSW, SP, DPS, DPL, DPH, DPL1, DPH1

累加器 (ACC)

累加器 ACC 是一个常用的专用寄存器，指令系统中采用 A 作为累加器的助记符。

B 寄存器

在乘除指令中，会用到 B 寄存器，在其他指令中，B 寄存器可作为暂存器使用

栈指针 (SP)

栈指针 SP 是一个 8 位专用寄存器，在执行 PUSH、各种子程序调用、中断响应等指令时，SP 先加 1，再将数据压栈；执行 POP、RET、RETI 等指令时，数据退出堆栈后 SP 再减 1。堆栈栈顶可以是片上内部 RAM (00H-FFH) 的任意地址，系统复位后，SP 初始化为 07H，使得堆栈事实上由 08H 地址开始。

程序状态字 (PSW) 寄存器

程序状态字包含了程序状态信息。

数据指针 (DPTR)

数据指针 DPTR 是一个 16 位专用寄存器，其高位字节寄存器用 DPH 表示，低位字节寄存器用 DPL 表示。它们既可以作为一个 16 位寄存器 DPTR 来处理，也可以作为 2 个独立的 8 位寄存器 DPH 和 DPL 来处理，EN8F5113 系列包含双数据指针 DPTR&DPTR1，通过 DPS(bit0)寄存器选择。

Table 5.1 PSW 寄存器

D0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	CY	进位标志： 当最后一次算术操作产生进位（加法）或借位（减法）时，该位置 1。其它算术操作将其清 0。
6	AC	辅助进位标志 当最后一次算术操作向高半字节有进位（加法）或借位（减法）时，该位置 1。其它算术操作将其清 0
5	F0	用户标志 0： 这是一个可位寻址、用于软件控制的通用标志位
4-3	RS[1:0]	RS1-RS0: 寄存器区选择 00: 页 0（映射到 00H-07H） 01: 页 1（映射到 08H-0FH） 10: 页 2（映射到 10H-17H） 11: 页 3（映射到 18H-1FH）

2	OV	溢出标志： 该位在下列情况下被置 1： ADD、ADDC 或 SUBB 指令引起符号位变化溢出 MUL 指令引起溢出（结果大于 255）。 DIV 指令的除数为 0。 ADD、ADDC、SUBB、MUL 和 DIV 指令的其它情况使该位清 0
1	F1	用户标志 1： 这是一个可位寻址、用于软件控制的通用标志位
0	P	奇偶校验位： 0：累加器中 8 个位的和为偶数 1：累加器中 8 个位的和为奇数

6.2. 存储器结构

EN8F5113 系列具有 8KB Flash ROM, 1024 字节 SRAM, 128 字节数据 EEPROM, 以及用户可扩展 EEPROM 区。这些存储器的地址空间与结构示意图如下:

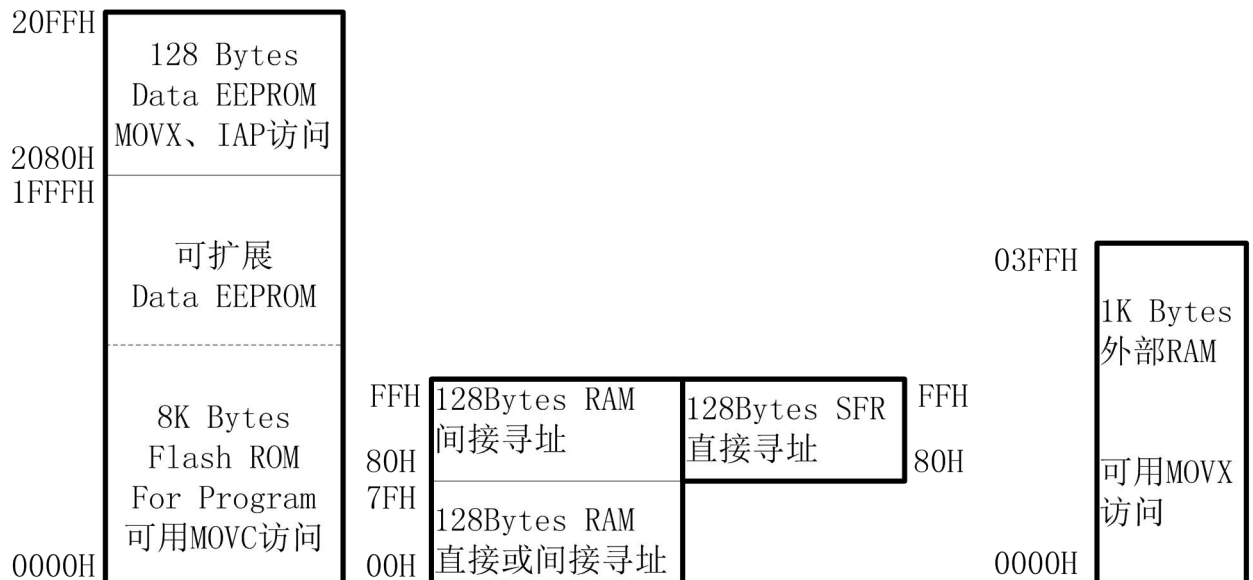


图 6-0

6.2.1. SRAM

EN8F5113 系列为数据存储提供了内部 RAM 和外部 RAM, 下列为存储器空间分配:

- ◆ 低位 128 字节的 RAM(地址从 00H 到 7FH), 可直接或间接寻址
- ◆ 高位 128 字节的 RAM(地址从 80H 到 FFH) 只能间接寻址
- ◆ 特殊功能寄存器 (SFR, 地址从 80H 到 FFH) 只能直接寻址
- ◆ 外部 RAM 可通过 MOVX 指令直接访问

高位 128 字节的 RAM 占用的地址空间和 SFR 相同, 但在物理上与 SFR 的空间是分离的。当一个指令访问高于地址 7FH 的内部位置时, CPU 可以根据访问的指令类型来区分是访问高位 128 字节数据 RAM 还是访问 SFR。

EN8F5113 系列在外部数据空间提供了 1k bytes 字节 RAM, 支持高级语言。

6.2.2. Flash 存储器

EN8F5113 系列有 8KB 的 Flash ROM, 地址是 0000H~1FFFH。此 4KB/8KB 存储器可以反复擦写读, 擦写次数可达 10 万次, 数据保持能力为 10 年。EN8F5113 系列通过专用烧写器对 Flash ROM 进行擦除、写入、读出以及加密等操作。EN8F5113 系列采用清晰的权限控制和先进的数据加密技术, 大大提高 Flash Rom 数据安全性。

地址为 2080H~20FFH 区域的 Flash ROM 可以作为 Data EEPROM 使用(即支持用户在 ICP 时对此区域进行擦除、写入以及读取操作, 详细操作请看 ICP/IAP 章节), 此 128 字节 Flash 可反复擦写 10 万次, 也可以通过 IAP、MOVX 指令读取数据。

另外, Flash 数据区最后一个 Page (地址: 1C00H~1FFFH) 可配置为 ROM 区+EEPROM 区, EEPROM 区可提供给用户存储需要保持的变量。EEPROM 区位于 8K 空间的最后空间。

用户可根据需求来扩展 Data EEPROM，支持 1024/512/256/0 字节可选；对于有扩展 Data EEPROM 的情况，编程 Flash ROM 将减少相应的字节。

扩展 Data EEPROM 控制寄存器 **EEPROMCON(DCH)**:

DCH	Bit7-2	Bit1	Bit0
EEPROMCON	-	EEPROMCON[1:0]	
R/W	-	R/W	
Default	-	00b	

位编号	位符号	说明		
7:4	-	-		
1:0	EEPROMCON	EEPROMCON	扩展 EEPROM	地址
		00	0	-
		01	256 Bytes	1F00H~1FFFH
		10	512 Bytes	1E00H~1FFFH
		11	1024 Bytes	1C00H~1FFFH

7. 系统时钟和复位系统

7.1. 系统时钟

EN8F5113 系列采用三时钟系统:

- 内置 16MHz RC 时钟振荡器(IHRC, Internal High-Speed RC Oscillator)
- 外接高速晶体振荡器(HOSC)
- 和外接低速晶体振荡器(LOSC)

这四个时钟都可用作系统时钟源 Fsys, 并可通过时钟管理寄存器实现切换, 默认是 IHRC 振荡时钟。IHRC 在芯片出厂前被精确地校准到 16MHz@3.3V/25°C, 此 IHRC 受工作温度和工作电压影响会有一些的漂移, 在 1.8V~5.5V 的电压范围、-25°C~85°C 温度范围内, 频率漂移小于 ±1%。

128KHz ILRC 时钟经 4 分频后再做为低频时钟使用。

Fsys 为系统时钟频率即 CPU 内核工作频率, 该时钟频率可以通过时钟管理模块实现 IHRC 与外部晶振时钟之间切换。

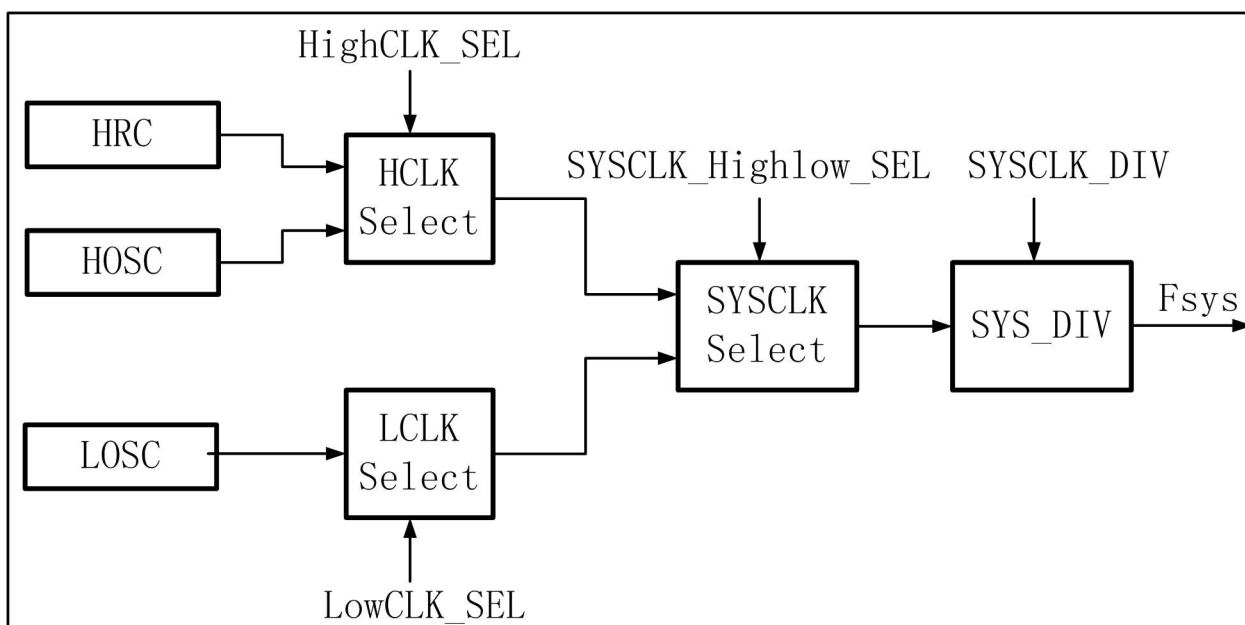


图 7-0

7.1.1. 寄存器

Table1. 外部晶体振荡器选择控制寄存器 PORTMUX1L(F8H):

F8H	Bit7-4	Bit3	Bit2	Bit1	Bit0
PORTMUX1L	PWMPORTEN[3:0]	LOSCPORTEN	BEEPPORTEN	RSTPORTEN	HOSCPORTEN
R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0

位编号	位符号	说明
7	PWMPORTEN[3]	1:PWM3 (第一复用) ; 0: Other
6	PWMPORTEN[2]	1:PWM2 (第一复用) ; 0: Other
5	PWMPORTEN[1]	1:PWM1 (第一复用) ; 0: Other
4	PWMPORTEN[0]	1:PWM0 (第一复用) ; 0: Other
3	LOSCPORTEN	1:LOSC XI/XO (第一复用); 0: 0: Other Flash Infor 初始化。
2	BEEPPORTEN	上电后软件需要将该位置为 1 。 1:输出 BEEP (第一复用) ; 0: Reserve 该信号也直接控制 BEEP 模块 1:打开 BEEP 模块, 0:关闭 BEEP 模块
1	RSTPORTEN	1:NRST (第一复用) ; 0: P0.2 (MainFunc) Flash Infor 初始化
0	HOSCPORTEN	1:HOSC XI/XO (第一复用); 0: P0.0/P0.1 (MainFunc) Flash Infor 初始化

Table2. 时钟选择控制寄存器 CLKCON(F4H):

F4H	Bit7-Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	-	WDTCKEN	RC16MEN	LowCLK_SEL	HighCLK_SEL	SYSClk_HighLow_SEL
R/W	-	R/W	R/W	R/W	R/W	R/W
Default	-	1	1	1	1	1

位编号	位符号	说明
7: 5	-	-
4	WDTCKEN	1: 打开 ILRC; 0: 关闭 ILRC 芯片处于 Normal 工作模式下, 必须一直置 1, 即 ILRC 一直使能
3	RC16MEN	1: 打开 IHRC; 0: 关闭 IHRC
2	LowCLK_SEL	1: 选择 ILRC 的 4 分频 时钟作为低速时钟源 0: 选择 LOSC 作为低速时钟源
1	HighCLK_SEL	1: 选择 IHRC 作为高速时钟源 0: 选择 HOSC 作为高速时钟源
0	SYSClk_HighLow_SEL	1: 选择 HOSC 或 IHRC 时钟作为系统时钟源 0: 选择低速 LOSC 或 ILRC 的 4 分频 作为系统时钟源

Table3. 时钟分频控制寄存器 CLKDIV(ECH):

ECH	Bit7-3	Bit2	Bit1	Bit0
CLKDIV	-	SYSCLKDIV[2]	SYSCLKDIV[1]	SYSCLKDIV[0]
R/W	-	R/W	R/W	R/W
Default	-	0	0	0

位编号	位符号	说明
7: 3	-	-
2	SYSCLKDIV[2]	高速时钟分频后作为系统时钟:
1	SYSCLKDIV[1]	000:1 分频; 001:2 分频; 010 : 4 分频
0	SYSCLKDIV[0]	011:8 分频; 100:16 分频; 101: 32 分频 110:64 分频; 111:128 分频

7.1.2. 内置 RC 时钟振荡器

EN8F5113 系列内置两个 RC 振荡器:

一个标称值 16MHz、可校准 IHRC 振荡器,校准后 IHRC 振荡器输出精度 $\pm 1\%$ ($-25^{\circ}\text{C}\sim+85^{\circ}\text{C}$)的时钟信号。

EN8F5113 系列上电启动后缺省选择 IHRC 振荡器时钟作为系统时钟。

EN8F5113 系列正常上电复位后, IHRC 振荡器即开始稳定输出时钟信号。该时钟信号可以通过内部校准控制寄存器或者 Flash 闪存信息区中的校准字来达到频率值校准,以得到精准的 16MHz 时钟信号。

- 一个输出频率为 128KHz ILRC 振荡器,不可校准,其输出时钟经 4 分频后供给系统,即 32KHz。

7.1.3. 外部晶体振荡器

EN8F5113 系列支持外接晶振时钟、陶瓷谐振器(HOSC、LOSC)外部时钟信号。当选用外部时钟信号时,用户首先打开外部时钟选择信号 PORTMUX1H (E8H)即将所选择的晶振对应的 IO 设置为晶振管脚,然后再设置 CLKCON 的 Bit2、Bit1,确定 CPU 时钟 F_{sys} 外部时钟的选择。

外部晶振时钟(HOSC、LOSC)和内部 RC(IHRC)能同时工作,通过上述寄存器配置来选择一路时钟作为 F_{sys}。

7.2. 复位系统

EN8F5113 系列有以下复位方式：

- 上电复位(BOR, 内含 LVR 模块);
- 外部复位(内建上拉电阻);
- 看门狗复位 (WDT);
- 低电压复位 LVD (Low Voltage Detect)。

7.2.1. 看门狗复位

看门狗定时器是一个递减计数器。

读或者写 RSTSTAT 寄存器均自动清除看门狗计数器 (喂狗)。

复位寄存器 RSTSTAT (E4H)

E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2-0
RSTSTAT	WDTOF	WDTEN	-	PORF	ERSTF	WDTCYCLE[2:0]
R/W	R/W	R/W	-	R/W	R/W	R/W
复位值(tPOR)	0	0	-	1	x	0
复位值(tWDT)	1	0	-	x	x	0
复位值(tERST)	x	0	-	x	1	0

编号	位符号	说明
7	WDTOF	看门狗溢出标志位 看门狗溢出时由硬件置 1，可由软件或上电复位清 0 0: 未发生 WDT 溢出 1: 发生 WDT 溢出
6	WDTEN	看门狗使能控制位 0: 关闭看门狗功能 1: 使能看门狗功能
5	-	-
4	PORF	上电复位标志位 上电复位后硬件置 1，只能由软件清 0 0: 没有发生上电复位 1: 发生过上电复位
3	ERSTF	Reset 引脚复位标志位 引脚复位后置 1，由软件或上电复位清 0 0: 没有发生引脚复位 1: 发生过引脚复位
2-0	WDTCYCLE[2:0]	WDT 溢出最小周期: 000: 溢出周期最小值= 2048ms 001: 溢出周期最小值= 512ms 010: 溢出周期最小值= 128ms

	011: 溢出周期最小值= 64ms 100: 溢出周期最小值= 32ms 101: 溢出周期最小值= 8ms 110: 溢出周期最小值= 2ms 111: 溢出周期最小值= 0.5ms 注意: 应用中如果看门狗打开, 程式请看门狗的最大间隔时间不能大于以上所列数值。 读或者写 RSTSTAT 寄存器均可清看门狗
--	---

7.2.2. 低电压复位

EN8F5113 系列内建低电压复位电路模块, 缺省为使能状态。当电源电压低于设置的复位门限电压, LVD 模块工作并产生复位信号, 整个芯片进入复位状态。而当电源电压回升到复位电压门限以上, 复位信号释放, 芯片重新进入正常工作状态。

由于此模块需要基准参考电压, 所以只能工作在 Normal 状态下。

LVD 模块控制寄存器 **LVDLVRCON(FDH)**

FDH	Bit7-4	Bit3	Bit2	Bit1	Bit0
LVDLVRCON	LVDS[3:0]	LVDFRSTEN	LVDF	LVR_ENB	LVD_ENB
R/W	R/W	R/W	R	R/W	R/W
复位值	0	0	0	0	0

编号	位符号	说明																																				
7-4	LVDS[3:0]	LVD 复位门限电压点设置: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LVD_S</th> <th>LVD point</th> <th>LVD_S</th> <th>LVD point</th> </tr> </thead> <tbody> <tr><td>0000</td><td>1.5V</td><td>1000</td><td>3.1V</td></tr> <tr><td>0001</td><td>1.7V</td><td>1001</td><td>3.3V</td></tr> <tr><td>0010</td><td>1.9V</td><td>1010</td><td>3.5V</td></tr> <tr><td>0011</td><td>2.1V</td><td>1011</td><td>3.7V</td></tr> <tr><td>0100</td><td>2.3V</td><td>1100</td><td>3.9V</td></tr> <tr><td>0101</td><td>2.5V</td><td>1101</td><td>4.1V</td></tr> <tr><td>0110</td><td>2.7V</td><td>1110</td><td>4.3V</td></tr> <tr><td>0111</td><td>2.9V</td><td>1111</td><td>4.5V</td></tr> </tbody> </table>	LVD_S	LVD point	LVD_S	LVD point	0000	1.5V	1000	3.1V	0001	1.7V	1001	3.3V	0010	1.9V	1010	3.5V	0011	2.1V	1011	3.7V	0100	2.3V	1100	3.9V	0101	2.5V	1101	4.1V	0110	2.7V	1110	4.3V	0111	2.9V	1111	4.5V
LVD_S	LVD point	LVD_S	LVD point																																			
0000	1.5V	1000	3.1V																																			
0001	1.7V	1001	3.3V																																			
0010	1.9V	1010	3.5V																																			
0011	2.1V	1011	3.7V																																			
0100	2.3V	1100	3.9V																																			
0101	2.5V	1101	4.1V																																			
0110	2.7V	1110	4.3V																																			
0111	2.9V	1111	4.5V																																			
3	LVDFRSTEN	软件置 1 表示 LVDF 作为系统复位信号,POR/ETST 复位期间 HW 从 FlashInfor 区读值填写。																																				
2	LVDF	1: 表示检测到低电,接模拟信号 LVDO。 如果 LVDRSTEN=1, 该信号送到复位模块; 如果 LVDRSTEN=0, 只产生中断, 进入中断服务程序需要清该标准位																																				
1	LVR_ENB	0: LVR (POR 模块中)有效; 1: LVR (POR 模块中)关闭。 软件配置该位寄存器。																																				
0	LVD_ENB	0:Enable LVD; 1:disable LVD。																																				

低功耗模式需关闭 LVD。

7.2.3. 内置高速比较器

EN8F5113 系列中 LVD 和内置高速比较器共用，当作为内部高速比较器使用时：

LVDCP 模块控制寄存器 LVDCP (87H)

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCP	LVD_CP_INTEN	LVD_CP_INTCON	LVD_CP_IF	CP_O		LVD_CP_INP_SEL	LVD_CP_INN_SEL	LVD_CP_EN
R/W	R/W	R/W	R/W	R		R/W	R/W	R/W
复位值	0	0	0	0		0	0	0

编号	位符号	说明
7	LVD_CP_INTEN	0: LVD_CP 不产生中断; 1: LVD_CP 中断打开
6	LVD_CP_INTCON	0: CP_O=0 产生中断; 1: CP_O=1 产生中断
5	LVD_CP_IF	LVD_CP 中断标准位, 软件需要清 0
4	CP_O	比较结果: 正端>负端, 输出 1; 否则输出 0
3		
2	LVD_CP_INP_SEL	比较器正端选择 0: ANA_OUT[1]; 1: ANA_OUT[3]
1	LVD_CP_INN_SEL	比较器负端选择 0: ANA_OUT[0]; 1: ANA_OUT[2]
0	LVD_CP_EN	1: 比较器使能。(LVD_ENB 必须为 1, 即 LVD 功能关闭) 0: 比较器关闭

8. I/O 端口

8.1. 特性

- ◆ 17 个可编程双向 I/O 端口
- ◆ 输出驱动可编程控制
- ◆ 可选择 Open Drain 输出
- ◆ 全部 IO 支持中断功能：上升沿、下降沿、高/低电平触发
- ◆ 内建上拉、下拉电阻
- ◆ 10 个 IO 都支持作为 1/2 BIAS 的 LCD COM/SEG 输出：
P1.0~P1.7, P2.0~P2.1
- ◆ I/O 口可与其他功能共用

8.2. 端口状态控制

Table 端口方向控制寄存器 (PxOEN)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0OEN(F9H)	P0OEN.7	P0OEN.6	P0OEN.5	P0OEN.4	-	P0OEN.2	P0OEN.1	P0OEN.0
P1OEN(FAH)	P1OEN.7	P1OEN.6	P1OEN.5	P1OEN.4	P1OEN.3	P1OEN.2	P1OEN.1	P1OEN.0
P2OEN(FBH)	-	-	-	-	-	-	P2OEN.1	P2OEN.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	PxOENy X=0~2, y = 0~7	端口方向控制位 0: output 1: input

Table Open drain 使能控制寄存器 (PxODN)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0ODN(E1H)	P0ODN.7	P0ODN.6	P0ODN.5	P0ODN.4	-	P0ODN.2	P0ODN.1	P0ODN.0
P1ODN(E2H)	P1ODN.7	P1ODN.6	P1ODN.5	P1ODN.4	P1ODN.3	P1ODN.2	P1ODN.1	P1ODN.0
P2ODN(E3H)	-	-	-	-	-	-	P2ODN.1	P2ODN.0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	PxODNy X=0~2, y = 0~7	端口下拉使能控制位 0: 使能 Open Drain 输出 1: 关闭 Open Drain 输出

Table IO 输出驱动能力控制寄存器 (PxHC1y / PxHC0y)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0HC1(D9H) /P0HC0(D1H)	P0HC1.7 /P0HC0.7	P0HC1.6 /P0HC0.6	P0HC1.5 /P0HC0.5	P0HC1.4 /P0HC0.4	-	P0HC1.2 /P0HC0.2	P0HC1.1 /P0HC0.2	P0HC1.0 /P0HC0.0
P1HC1(DAH) /P1HC0(D2H) /P2HC0(D3H)	P1HC1.7 /P1HC0.7	P1HC1.6 /P1HC0.6	P1HC1.5 /P1HC0.5	P1HC1.4 /P1HC0.4	P1HC1.3 /P1HC0.3	P1HC1.2 /P1HC0.2	P1HC1.1 /P1HC0.2	P1HC1.0 /P1HC0.0 P2HC0.2 P2HC0.0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0

位编号	位符号	说明
7:0	PxHC1y / PxHC0y X=0~2 ,y = 0~7	端口输出能力控制位 00, 8mA; 01, 12mA; 10, 16mA; 11, 24mA P2 端口支持 8mA/12mA

Table 端口数据寄存器 (Px)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0(80H)	P0.7	P0.6	P0.5	P0.4	-	P0.2	P0.1	P0.0
P1(90H)	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P2(A0H)	-	-	-	-	-	-	P2.1	P2.0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	Px.y X=0~2 ,y = 0~7	端口数据寄存器 当端口配置为输出状态时 0: 输出低电平, 1: 输出高电平 当端口配置为输入状态时, 读到的是端口状态

8.3. 端口上/下拉电阻

Table 上拉使能控制寄存器 (PxPUN)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POPUN(F1H)	POPUN.7	POPUN.6	POPUN.5	POPUN.4	-	POPUN.2	POPUN.1	POPUN.0
P1PUN(F2H)	P1PUN.7	P1PUN.6	P1PUN.5	P1PUN.4	P1PUN.3	P1PUN.2	P1PUN.1	P1PUN.0
P2PUN(F3H)	-	-	-	-	-	-	P2PUN.1	P2PUN.0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	PxPUNy	端口上拉使能控制位

	X=0~2 ,y = 0~7	0: 使能内部上拉电阻 1: 关闭内部上拉电阻 <i>内部上拉电阻设计参考值 75k Ω</i>
--	----------------	--

Table 下拉使能控制寄存器 (PxPDN)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POPDN(E9H)	POPDN.7	POPDN.6	POPDN.5	POPDN.4	-	POPDN.2	POPDN.1	POPDN.0
P1PDN(EAH)	P1PDN.7	P1PDN.6	P1PDN.5	P1PDN.4	P1PDN.3	P1PDN.2	P1PDN.1	P1PDN.0
P2PDN(EBH)	-	-	-	-	-	-	P2PDN.1	P2PDN.0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	PxPDNy X=0~2 ,y = 0~7	端口下拉使能控制位 0: 使能内部下拉电阻 1: 关闭内部下拉电阻 <i>内部下拉电阻设计参考值 75k Ω</i>

可以控制 IO (P1.0~P1.7,P2.0~P2.1), 分别输出 VDD、VDD/2、VSS 三种不同的电位, 实现 1/2 BIAS 的软件控制 LCD COM/SEG 输出:

- 首先将选择的 IO 设置为 Analog IO;
- 其次软件控制其内部上下拉电阻。

8.4. 端口复用

EN8F5113 系列的端口可复用做其他功能。最多有五级复用, 第一复用功能优先级最高, 默认功能优先级最低。

8.4.1. 端口复用控制寄存器

Table PORTMUX1L

F8H	Bit7-4	Bit3	Bit2	Bit1	Bit0
PORTMUX1L	PWMPORTEN[3:0]	LOSCPORTEN	BEEPPORTEN	RSTPORTEN	HOSCPORTEEN
读/写	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	

位编号	位符号	说明
7	PWMPORTEN[3]	1:PWM3 (第一复用); 0: Other
6	PWMPORTEN[2]	1:PWM2 (第一复用); 0: Other
5	PWMPORTEN[1]	1:PWM1 (第一复用); 0: Other
4	PWMPORTEN[0]	1:PWM0 (第一复用); 0: Other
3	LOSCPORTEN	1:LOSC XI/XO (第一复用); 0: 0: Other、Other Flash Infor 初始化。

2	BEEPPORTEN	上电后软件需要置该位为 1 。 该信号也直接控制 BEEP 模块，1 打开 BEEP 模块，0 关闭 BEEP 模块 1:输出 BEEP（第一复用）； 0: Reserve
1	RSTPORTEN	1:nRst（第一复用）； 0: P0.2（MainFunc） Flash Infor 初始化
0	HOSCPORTEN	1:HOSC XI/XO（第一复用）； 0: P0.0/P0.1（MainFunc） Flash Infor 初始化

Table PORTMUX1H

E8H	Bit7	Bit6-2	Bit1-0
PORTMUX1H	-	ADPORTEN[9:4]	PWMPORTEN[5:4]
读/写	R/W		
复位值	0		

位编号	位符号	说明
7	-	-
6	ADPORTEN[9]	1:AIN9（第一复用）； 0: Other
5	ADPORTEN[8]	1:AIN8（第一复用）； 0: Other
4	ADPORTEN[7]	1:AIN7（第一复用）； 0: Other
3	ADPORTEN[6]	1:AIN6（第一复用）； 0: Other
2	ADPORTEN[4]	1:AIN4（第一复用）； 0: P1.4（MainFunc）
1	PWMPORTEN[5]	1:PWM5（第一复用）； 0: Other
0	PWMPORTEN[4]	1:PWM4（第一复用）； 0: Other

Table PORTMUX_2

D8H	Bit7-4	Bit3	Bit2	Bit1	Bit0
PORTMUX_2	-	SPIPORTEN	I2CPORTEN	UART1PORTEN	UART0PORTEN
读/写	-	R/W	R/W	R/W	R/W
复位值	-	0	1	0	0

位编号	位符号	说明
7-4		
3	SPIPORTEN	0: Other、Other、Other、Other； 1:在第一复用没开启的前提下，这 4 个 PAD 为 SPI_NSS、SPI_SCK、SPI_MOSI、SPI_MISO
2	I2CPORTEN	0: Other； 1:在第一复用没开启的前提下，这两个 PAD 为 SDA、SCL
1	UART1PORT_EN	0: P2.0、Other； 1:在第一复用没开启的前提下，这两个 PAD 为 UART1 TX0、RX0

0	UART0PORTEN	0: P1.6、P1.7; 1:在第一复用没开启的前提下, 这两个 PAD 为 UART0 TX0、RX0
---	-------------	--

Table PORTMUX_3

C0H	Bit7-5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTMUX_3	-	ADPORTEN[5:0]				
读/写	-	R/W				
复位值	-	0				

位编号	位符号	说明
7-5	-	-
4	ADPORTEN[5]	1:AIN5 (第三复用); 0: P1.5 (MainFunc)
3	ADPORTEN[3]	1:AIN3 (第三复用); 0: Other
2	ADPORTEN[2]	1:AIN2 (第三复用); 0: Other
1	ADPORTEN[1]	1:AIN1 (第三复用); 0: Other
0	ADPORTEN[0]	1:AIN0 (第三复用); 0: Other

Table PORTMUX_4

91H	Bit7-3	Bit2	Bit1	Bit0
PORTMUX_4	-	VREFPORTEN	CP1PORTEN	CP0PORTEN
读/写	-	R/W	R/W	R/W
复位值	-	0	0	0

位编号	位符号	说明
7-3	-	-
2	VREFPORTEN	0: P2.1; 1:在第一、二、三复用没开启的前提下, 这个 PAD 为 VREF
1	CP1PORTEN	0: P1.2、P1.3; 1:在第一、二、三复用没开启的前提下, 这 2 个 PAD 为 CP_INN1、CP_INP1
0	CP0PORTEN	0: P1.0、P1.1; 1:在第一、二、三复用没开启的前提下, 这 2 个 PAD 为 CP_INN0、CP_INP0

8.4.2. 端口复用说明

- 一级复用控制寄存器：PORTMUX1L/H；
- 二级复用控制寄存器：PORTMUX2；
- 三级复用控制寄存器：PORTMUX3；
- 四级复用控制寄存器：PORTMUX4；
- 默认功能。

一级复用	二级复用	三级复用	四级复用	默认功能	Pad name
if(RSTPORTEN ==1) ==>nRst				else P0.2	pad_p02_nrst
if(HOSCPORTEN ==1) ==> XI				else P0.0	pad_p00_hxin
if(HOSCPORTEN ==1) ==> XO				else P0.1	pad_p01_hxout
if(BEEPEN ==1) ==> BEEP					pad_beep
	else if(SPIPORTEN==1) ==> SPI_NSS			else P0.4	pad_p04
if(LOSCPORTEN==1) ==> LOSC XI	else if(I2CPORTEN==1) ==> SDA			else P0.5	pad_p05_lxin
if(LOSCPORTEN==1) ==> LOSC XO	else if(I2CPORTEN==1) ==> SCL			else P0.6	pad_p06_lxout
if(PWM0PORTEN ==1) ==> PWM0_OUT				else P0.7	pad_p07_pwm0
if(PWM1PORTEN ==1) ==> PWM1_OUT		if(ADPORTEN[0] ==1) ==> AIN0(A)	else if(CPOPORTEN ==1) ==> CP_INNO	else P1.0	pad_p10_ain0
if(PWM2PORTEN ==1) ==> PWM2_OUT	else if(SPIPORTEN==1) ==> SPI_SCK	if(ADPORTEN[1] ==1) ==> AIN1(A)	else if(CPOPORTEN ==1) ==> CP_INP0	else P1.1	pad_p11_ain1
if(PWM3PORTEN ==1) ==> PWM3_OUT	else if(SPIPORTEN==1) ==> SPI_MOSI	if(ADPORTEN[2] ==1) ==> AIN2(A)	else if(CP1PORTEN ==1) ==> CP_INN1	else P1.2	pad_p12_ain2
if(PWM4PORTEN ==1) ==> PWM4_OUT	else if(SPIPORTEN==1) ==> SPI_MISO	if(ADPORTEN[3] ==1) ==> AIN3(A)	else if(CP1PORTEN ==1) ==> CP_INP1	else P1.3	pad_p13_ain3
if(ADPORTEN[4]==1)=				else P1.4	pad_p14_ain4

=> AIN4(A)					
if(PWM5PORTEN ==1) ==> PWM5_OUT		if(ADPORTEN[5] ==1) ==> AIN5(A)		else P1.5	pad_p15_ain5
if(ADPORTEN[6]==1) ==> AIN6(A)	else if(UART0PORTEN==1) ==> UART0_TX			else P1.6	pad_p16_ain6
if(ADPORTEN[7]==1) ==> AIN7(A)	else if(UART0PORTEN==1) ==> UART0_RX			else P1.7	pad_p17_ain7
if(ADPORTEN[8]==1) ==> AIN8(A)	else if(UART1PORTEN==1) ==> UART1_TX			else P2.0	pad_p20_ain8
if(ADPORTEN[9]==1) ==> AIN9(A)	else if(UART1PORTEN==1) ==> UART1_RX		else if(VREFPORTEN==1) ==> VREF(A)	else P2.1	pad_p21_ain9

8.5. 端口中断

EN8F5113 系列全部端口支持中断功能，如果使能中断，上升沿或者下降沿均会触发中断，全部端口中断共用 INTO 中断入口。

如果端口中断被使能，当 MCU 进入掉电模式时，端口电平的变化，可以中断唤醒 MCU

Table 端口中断使能控制位 (PxIEN)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POIEN(C9H)	POIEN.7	POIEN.6	POIEN.5	POIEN.4	-	POIEN.2	POIEN.1	POIEN.0
P1IEN(CAH)	P1IEN.7	P1IEN.6	P1IEN.5	P1IEN.4	P1IEN.3	P1IEN.2	P1IEN.1	P1IEN.0
P2IEN(CBH)	-	-	-	-	-	-	P2IEN.1	P2IEN.0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	PxIENy X=0~2 ,y = 0~7	端口中断使能控制位 0: 关闭端口中断功能 1: 使能端口中断功能。打开中断功能的时候，使能端口的去抖动功能。

Table 端口中断标志位 (PxIRQ)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POIRQ(C1H)	POIRQ.7	POIRQ.6	POIRQ.5	POIRQ.4	-	POIRQ.2	POIRQ.1	POIRQ.0

P1IRQ(C2H)	P1IRQ.7	P1IRQ.6	P1IRQ.5	P1IRQ.4	P1IRQ.3	P1IRQ.2	P1IRQ.1	P1IRQ.0
P2IRQ(C3H)	-	-	-	-	-	-	P2IRQ.1	P2IRQ.0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	PxIRQy X=0~2 ,y = 0~7	端口中断标志位 0: 端口未产生中断 1: 端口产生了中断

Table 端口中断触发方式 (P012IRQCON)

	Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P012IRQCON(FCH)	-	P2IRQ Config[1]	P2IRQ Config[0]	P1IRQ Config[1]	P1IRQ Config[0]	P0IRQ Config[1]	P0IRQ Config[0]
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W
复位值	-	0	0	0	0	0	0

位编号	位符号	说明
7:6	-	-
5	P2IRQConfig[1]	配置 P2 口 2 个 I0 中断触发方式
4	P2IRQConfig[0]	00: 下降沿触发; 01: 上升沿触发 10: 高电平触发; 11: 低电平触发
3	P1IRQConfig[1]	配置 P1 口 8 个 I0 中断触发方式
2	P1IRQConfig[0]	00: 下降沿触发; 01: 上升沿触发 10: 高电平触发; 11: 低电平触发
1	P0IRQConfig[1]	配置 P0 口 8 个 I0 中断触发方式
0	P0IRQConfig[0]	00: 下降沿触发; 01: 上升沿触发 10: 高电平触发; 11: 低电平触发

Table 端口中断标志位清除使能 (PORTIRQCLREN)

	Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PORTIRQCLREN (DBH)	-						PORTIRQCLREN
读/写	-						R/W
复位值	-						0

位编号	位符号	说明
7:1	-	-

0	PORTIRQCLREN	在清 PORTIRQ 前写 0x01 到该寄存器，然后再执行清 POIRQ/P1IRQ/P2IRQ 的动作，清完后必须写 0x00 到该寄存器
---	--------------	---

具体操作参考如下：

清外部中断0 信号：

```

... ..
PORTIRQCLREN |= 0x01;           //清外部中断信号ENABLE
POIRQ=0;
PORTIRQCLREN &= 0xFE;          //清外部中断信号DISABLE

```

清外部中断1 信号：

```

... ..
PORTIRQCLREN |= 0x01;           //清外部中断信号ENABLE
P1IRQ=0;
PORTIRQCLREN &= 0xFE;          //清外部中断信号DISABLE

```

清外部中断2 信号：

```

... ..
PORTIRQCLREN |= 0x01;           //清外部中断信号ENABLE
P2IRQ=0;
PORTIRQCLREN &= 0xFE;          //清外部中断信号DISABLE

```

清外部所有中断信号：

```

... ..
PORTIRQCLREN |= 0x01;           //清外部中断信号ENABLE
POIRQ=0;
P1IRQ=0;
P2IRQ=0;
PORTIRQCLREN &= 0xFE;          //清外部中断信号 DISABLE
... ..

```

9. 定时器

EN8F5113 系列 有:

- 两个 16 位定时器/计数器 T0、T1，它们具有计数方式和定时方式两种工作模式；定时器 0, 1 是 16 位自动重载定时器，通过两个数据寄存器 TH0/1 和 TL0/1 访问，由 TCON 寄存器控制，IEN0 寄存器的 bit1 et0 位置 1 允许定时器 0 中断，IEN0 寄存器的 bit3 et1 位置 1 允许定时器 1 中断(详见**中断**章节)
- 一个 32 位 RTC 计数器，分别有秒、分、小时的计数，并可产生半秒、秒、分、小时以及天中断。

9.1. 定时器 0, 1

9.1.1. 定时器 0, 1 工作方式

- 定时器 0, 1 的工作方式完全一样，以下以定时器 0 来说明定时器 0, 1 的工作方式。
- 定时器 0 只有一种工作方式:16 位自动重载定时器，可以设置预分频比，定时器 0 有一个 16 位计数器/定时器 (TH0, TL0). 当 TH0 和 TL0 被写时，用作定时器重载寄存器，当被读时，用作计数寄存器. TRO 位置 1 使定时器 0 开始递增计数. 定时器在 0xFFFF 到 0x0000 溢出并置位 TFO. 溢出同时，定时器重载寄存器的 16 位数据被重新载入计数寄存器中，TH0 写操作也导致重载寄存器的数据重新载入计数寄存器。
- TH0 和 TL0 读写操作遵循以下顺序：
写操作:先低位后高位
读操作:先高位后低位
- 定时器 0, 1 必须选择系统时钟为时钟源

9.1.2. 定时器 0, 1 模块图

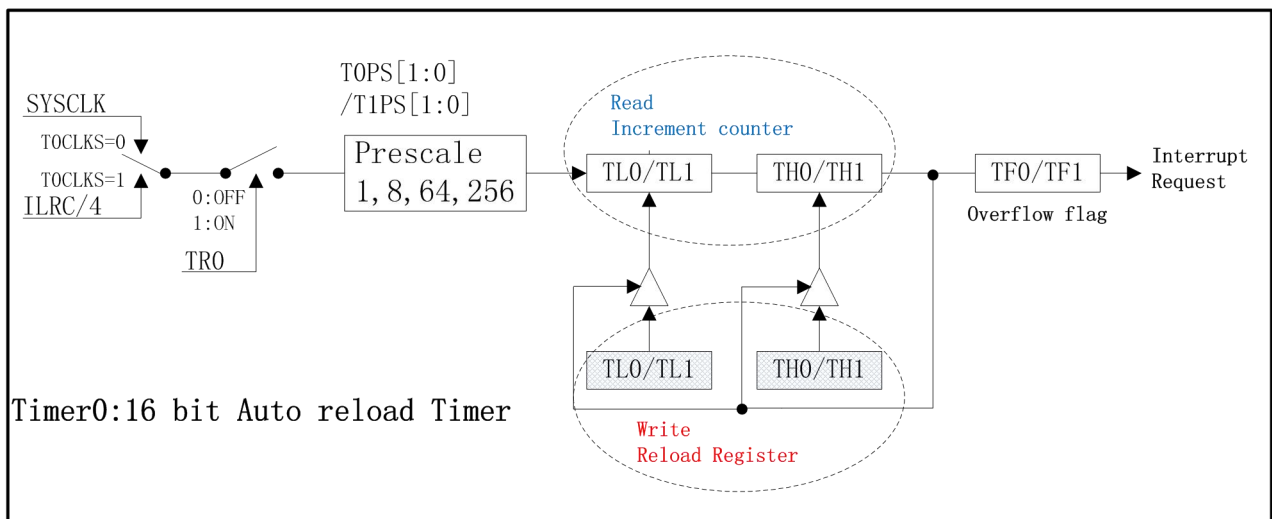


图 9-0

9.1.3. 定时器 0, 1 寄存器

定时器 0,1 重载/计数寄存器

8AH~8DH	Bit7-0
TL0(8AH)	Timer0 低 8 位 [7:0]; Default 是 8'b0000_0000
TH0(8BH)	Timer0 高 8 位[15:8]; Default 是 8'b0000_0000
TL1(8CH)	Timer1 低 8 位 [7:0]; Default 是 8'b0000_0000
TH1(8DH)	Timer1 高 8 位[15:8]; Default 是 8'b0000_0000
读/写	R/W

编号	位符号	说明
7: 0	TL0/TH0	定时器 0:重载/计数寄存器 写: 重载寄存器 读: 计数寄存器

编号	位符号	说明
7: 0	TL1/TH1	定时器 1:重载/计数寄存器 写: 重载寄存器 读: 计数寄存器

定时器 0, 1 控制寄存器 TCON:

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	-	TR1	T1CLKSel	TF0	-	TR0	TOCLKSel
读/写	R/W	-	R/W	R/W	R/W	-	R/W	R/W
复位值	0	-	0	0	0	-	0	0

编号	位符号	说明
7	TF1	定时器 1 溢出标志 0: 无溢出 (必须软件清 0) 1: 溢出 (硬件置位, 软件写 1 无效, 写 0 清 0)
6	-	-
5	TR1	定时器 1 使能位: 0: 停止定时器 1 1: 开始定时器 1
4	T1CLKS	定时器 1 计数时钟选择位: 必须写 0 选择系统时钟作为定时器 1 的计数时钟
3	TF0	定时器 0 溢出标志 0: 无溢出 (必须软件清 0) 1: 溢出 (硬件置位, 软件写 1 无效, 写 0 清 0)
2	-	-

1	TRO	定时器 0 使能位： 0: 停止定时器 0 1: 开始定时器 0
0	TOCLKS	定时器 0 计数时钟选择位： 必须写 0 选择系统时钟作为定时器 0 的计数时钟

定时器 0, 1 分频选择寄存器 TMOD:

89H	Bit7-6	Bit5	Bit4	Bit3-2	Bit1	Bit0
TMOD	-	T1PS[1]	T1PS[0]	-	TOPS[1]	TOPS[0]
读/写	-	读/写	读/写	-	读/写	读/写
复位值	-	0	0	-	0	0

编号	位符号	说明
7-6		
5-4	T1PS[1:0]	定时器 1 预分频比选择位： 00 :1/1 01:1/8 10:1/64 11:1/256
3-2		
1-0	TOPS[1:0]	定时器 0 预分频比选择位： 00 :1/1 01:1/8 10:1/64 11:1/256

9.2. RTC 计数器

RTC 计数器使用外部 LOSC 作为时钟源，分别有秒、分、小时的计数，并可产生半秒、秒、分、小时以及天中断。

RTC 时钟选择寄存器 CLKCON(F4H):

F4H	Bit7-Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	-	WDTCKEN	RC16MEN	LowCLK_SEL	HighCLK_SEL	SYSClk_HighLow_SEL
R/W	-	R/W	R/W	R/W	R/W	R/W
Default	-	1	1	1	1	1

位编号	位符号	说明
7: 5	-	-
4	WDTCKEN	1: 打开 ILRC; 0: 关闭 ILRC 芯片处于 Normal 工作模式下，必须一直置 1，即 ILRC 一直使能
3	RC16MEN	1: 打开 IHRC; 0: 关闭 IHRC

2	LowCLK_SEL	1: 选择 <i>ILRC 的 4 分频</i> 时钟作为低速时钟源 0: 选择 LOSC 作为低速时钟源
1	HighCLK_SEL	1: 选择 IHRC 作为高速时钟源 0: 选择 HOSC 作为高速时钟源
0	SYSCLK_HighLow_SEL	1: 选择 HOSC 或 IHRC 时钟作为系统时钟源 0: 选择低速 LOSC 或 <i>ILRC 的 4 分频</i> 作为系统时钟源

RTC 秒计数寄存器 RTCSECOND(B2H):

B2H	Bit7-Bit6	Bit5-0
RTCSECOND	-	Second[5:0]
R/W	-	R/W
Default	-	0

位编号	位符号	说明
7: 6	-	-
5	Second[5]	秒设置寄存器，表示当前时间的秒钟值.最大值为 59
4	Second[4]	
3	Second[3]	
2	Second[2]	
1	Second[1]	
0	Second[0]	

RTC 分计数寄存器 RTCMINUTE(B3H):

B3H	Bit7-6	Bit5-0
RTCMINUTE	-	Minute[5:0]
R/W	-	R/W
Default	-	0

位编号	位符号	说明
7: 6	-	-
5	Minute[5]	分设置寄存器，表示当前时间的秒钟值.最大值为 59
4	Minute[4]	
3	Minute[3]	
2	Minute[2]	
1	Minute[1]	
0	Minute[0]	

RTC 小时计数寄存器 RTCHOUR(A2H):

A2H	Bit7-Bit5	Bit4-0
HOUR	-	Minute[5:0]
R/W	-	R/W
Default	-	0

位编号	位符号	说明
7: 5	-	-
4	Minute[4]	小时设置寄存器，表示当前时间的秒钟值.最大值为 23
3	Minute[3]	
2	Minute[2]	
1	Minute[1]	
0	Minute[0]	

RTC 中断寄存器 RTCCON(A3H):

A3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RTCCON	RTSEN	-	RTCIRQ	DayIntEn	HourIntEn	MinIntEn	SecIntEn	HalfSecIntEn
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	-	0	0	0	0	0	0

位编号	位符号	说明
7	RTCEN	1:RTC 打开; 0: RTC 禁止
6	-	-
5	RTCIRQ	RTC 中断标志位，软件中断服务程序需要清 0
4	DayIntEn	1: 允许每天产生一次中断; 0: 关闭天中断
3	HourIntEn	1: 允许每小时产生一次中断; 0: 关闭小时中断
2	MinIntEn	1: 允许每分钟产生一次中断; 0: 关闭分中断
1	SecIntEn	1: 允许每秒钟产生一次中断; 0: 关闭秒中断
0	HalfSecIntEn	1: 允许每半秒产生一次中断; 0: 关闭半秒中断

10. 脉宽调制模块 (PWM)

PWM (脉宽调制) 信号是有广泛应用领域。它可用于电机驱动, 风扇控制, 背光亮度调节, LED 调光, 或模拟作为一个简单的数字到模拟转换器通过低通滤波电路输出。

EN8F5113 系列 PWM 是专为提供三对具有可编程周期和占空比, 最大分辨率为 16 位的 PWM 输出实现电机控制。该结构使用户易于驱动单相或三相无刷直流电动机或三相交流异步电动机。每六个 PWM 可配置为独立模式, 互补模式, 或同步模式之一。如果采用互补模式, 则可编程死区时间插入可同时保护 MOS 导通。PWM 波形可以是边对齐的, 也可以是中心对齐的可变中断点。

EN8F5113 系列带有 PWM 控制器 (PWMC), 内部带有一个 16 位独立计数器, 可以支持 6 路独立的 PWM 输出, 完成 6 路 PWM 控制功能。

该 PWMC 主要功能如下:

- 16bit 计数器, 产生 6 个独立 PWM 输出, 周期相同, 占空比可灵活配置
- 可配置为带死区控制的组合模式, 3 组互补的 PWM 对, 一对中的 PWM 输出波形相反互补, 插入的死区可配置 (01, 23, 45)。死区控制只在组合模式下有效。
- 可配置为同步模式, 输出三组相同的 PWM 信号, PWM1/3/5 分别等同于 PWM0/2/4
- 可配置为分组模式, 输出三队相同的 PWM 信号, PWM2=PWM4=PWM0, PWM3=PWM5=PWM1
- 支持中心对齐和边沿对齐模式
- One-shot (只有边沿对齐模式支持) 和 Auto-reload 模式
- 硬件故障保护: P1.4, ADC 数据比较触发
- 灵活的中断配置, 可配置产生上升沿、下降沿、中心点 (中心对齐才支持)、结束点 (中心对齐才支持)、故障刹车几类中断

PWM 输出至管脚, 需要先配置 IO 端口复用寄存器 PORTMUX1L、PORTMUX1H:

Table PORTMUX1L

F8H	Bit7-4	Bit3	Bit2	Bit1	Bit0
PORTMUX1L	PWMPORTEN[3:0]	LOSCPORTEN	BEEPPORTEN	RSTPORTEN	HOSCPORTEN
读/写	读/写	读/写	读/写	读/写	
复位值	0	0	0	0	

位编号	位符号	说明
7	PWMPORTEN[3]	1:PWM3 (第一复用); 0: Other
6	PWMPORTEN[2]	1:PWM2 (第一复用); 0: Other
5	PWMPORTEN[1]	1:PWM1 (第一复用); 0: Other
4	PWMPORTEN[0]	1:PWM0 (第一复用); 0: Other
3	LOSCPORTEN	1:LOSC XI/XO (第一复用); 0: Other Flash Infor 初始化。
2	BEEPPORTEN	上电后软件需要将该位置为 1。 该信号也直接控制 BEEP 模块, 1 打开 BEEP 模块, 0 关闭 BEEP 模块 1:输出 BEEP (第一复用); 0: Reserve
1	RSTPORTEN	1:nRst (第一复用); 0: P0.2 (MainFunc) Flash Infor 初始化

0	HOSCPORTEN	1:HOSC XI/XO (第一复用); 0: P0.0/P0.1 (MainFunc) Flash Infor 初始化
---	------------	---

Table PORTMUX1H

E8H	Bit7	Bit6-2	Bit1-0
PORTMUX1H	-	ADPORTEN[9:4]	PWMPORTEN[5:4]
读/写		读/写	
复位值		0	

位编号	位符号	说明
7	-	-
6	ADPORTEN[9]	1:AIN9 (第一复用); 0: Other
5	ADPORTEN[8]	1:AIN8 (第一复用); 0: Other
4	ADPORTEN[7]	1:AIN7 (第一复用); 0: Other
3	ADPORTEN[6]	1:AIN6 (第一复用); 0: Other
2	ADPORTEN[4]	1:AIN4 (第一复用); 0: P1.4 (MainFunc)
1	PWMPORTEN[5]	1:PWM5 (第一复用); 0: Other
0	PWMPORTEN[4]	1:PWM4 (第一复用); 0: Other

10.1. PWM 结构框图

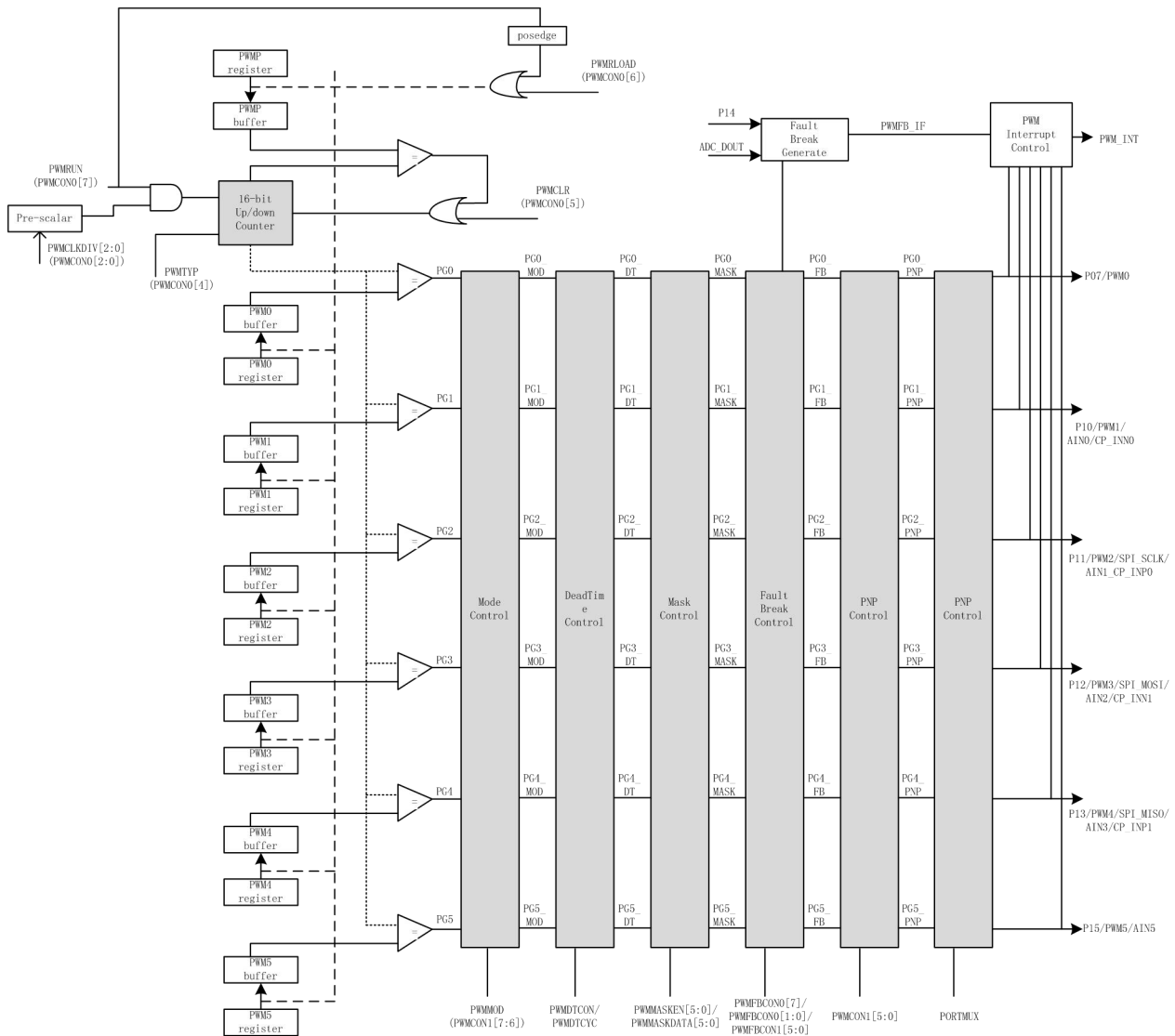


图 10-1 PWM 结构框图

PWM 发生器的时钟由系统时钟或定时器 1 溢出除以 PWM 时钟预标量选择从 1/1 到 1/128。有寄存器 PWMCON0.PWMCLKDIV[2:0]定义。

对于共享一个 16 位周期计数器，所有 PWM 通道的周期相同。

每个 PWM 的占空比是独立占空比寄存器确定，即 {PWMxDH, PWMxDL}，x 为 0~5。

有六个占空比寄存器，六个 PWM 输出可以独立产生不同的占空比。

PWM 信号的间隔和占空比是由 16 位计数器与周期和占空比寄存器相比得出。

PWM 计数器产生六路 PWM 信号称为 PG0, PG1, PG2, PG3, PG4 和 PG5。这些信号将通过 PWM 和故障制动输出控制电路。它在 I/O 引脚上产生真正的 PWM 输出。输出控制电路决定 PWM 模式、死区时间插入、Mask Output、故障制动控制和 PWM 极性。

最后一级是 PWM 输出或 I/O 功能的多路复用器。

10.2. PWM 模块控制寄存器

10.2.1. PWMCON0 (D4H)

D4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2-0
PWMCON0	PWMRUN	PWMLOAD	PWMCLR	PWMTYP	PWMOS	PWMCLKDIV[2:0]
读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0

位编号	位符号	说明
7	PWMRUN	PWM run enable 0: PWM Idle; 1: PWM 工作 FB 时硬件会清 0 RUN, 停止 PWM 工作并将计数器清为 0000
6	PWMLOAD	软件写 1 表示有新的 PWMP、PWMD 需要更新, 硬件在周期结束时再更新 PWMP、PWMD。更新后硬件清 0
5	PWMCLR	软件置 1 表示需要将计数器清为 0000, 硬件执行完该操作将 PWMCLR 清 0
4	PWMTYP	0: 边沿对齐模式, 占空比为 PWMD/PWMP, 周期为 PWMP 1: 中心对齐模式, 占空比为 PWMD/PWMP, 周期为 PWMPx2
3	PWMOS	0: 关闭 One-short 模式 1: 打开 One-short 模式。该模式在边沿对齐模式下才有效, 一个周期结束后 PWM 停止工作, 硬件清 PWMRUN 及计数器
2	PWMCLKDIV[2]	PWM 时钟分频设置: 000: 1/1; 001:1/2; 010:1/4; 011:1/8 100:1/16; 101:1/32; 110:1/64; 111:1/128
1	PWMCLKDIV[1]	
0	PWMCLKDIV[0]	

10.2.2. PWMCON1 (D5H)

D5H	Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON1	PWMMOD[1:0]	PWM5PNP	PWM4PNP	PWM3PNP	PWM2PNP	PWM1PNP	PWM0PNP
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0

位编号	位符号	说明
7	PWMMOD[1]	00: 独立模式, PWM0/1/2/3/4/5 分别独立输出
6	PWMMOD[0]	01: 带死区控制的组合模式, 0/2/4 还是独立输出, 1/3/5 是 0/2/4 的取反
		10: 同步模式, 0/2/4 还是独立输出, 1/3/5 等于 0/2/4,

		11: 分组模式, 0/1 独立输出, 2/4 等于 0, 3/5 等于 1
5	PWM5PNP	PWMn 输出数据取反选择: 0: PWMn 原始输出 1: PWMn 取反输出
4	PWM4PNP	
3	PWM3PNP	
2	PWM2PNP	
1	PWM1PNP	
0	PWM0PNP	

10.2.3. PWMMASKEN (D6H)

D6H	Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKEN	-	PWM5MEN	PWM4MEN	PWM3MEN	PWM2MEN	PWM1MEN	PWM0MEN
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值	-	0	0	0	0	0	0

位编号	位符号	说明
7-6	-	-
5	PWM5MEN	PWMn 输出数据选择: 0: PWMn 输出对应的 PWM 输出的值 1: PWMn 输出 PWMnD 的固定值
4	PWM4MEN	
3	PWM3MEN	
2	PWM2MEN	
1	PWM1MEN	
0	PWM0MEN	

10.2.4. PWMMASKDATA (CCH)

CCH	Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKDATA	-	PWM5MD	PWM4MD	PWM3MD	PWM2MD	PWM1MD	PWM0MD
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值	-	0	0	0	0	0	0

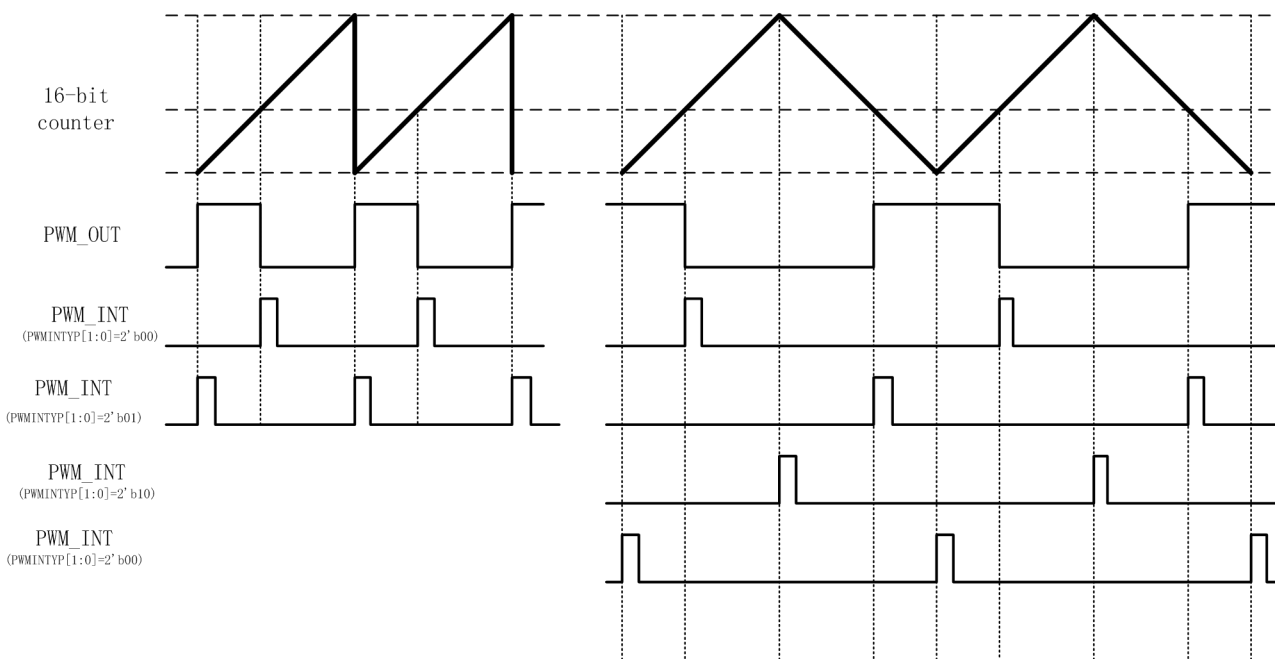
位编号	位符号	说明
7-6	-	-
5	PWM5MD	在 MASK 模式下, PWMn 输出数据: 0: PWMn 输出 0 1: PWMn 输出 1
4	PWM4MD	
3	PWM3MD	

2	PWM2MD	
1	PWM1MD	
0	PWM0MD	

10.2.5. PWMINTCON (CDH)

CDH	Bit7-6	Bit5-0
PWMINTCON	PWMINTTYP[1:0]	PWM5IEN~PWM0IEN
读/写	读/写	读/写
复位值	0	0

位编号	位符号	说明
7	PWMINTTYP[1]	00: PWM0/1/2/3/4/5 下降沿产生中断
6		01: PWM0/1/2/3/4/5 上升沿产生中断
	PWMINTTYP[0]	10: PWM0/1/2/3/4/5 中心点产生中断, 中心对齐模式才有效
		11: PWM0/1/2/3/4/5 结束点产生中断, 中心对齐模式才有效
5	PWM5IEN	1: PWM5 中断使能 (需 IEN0 EPWM=1); 0: 关闭 PWM5 中断
4	PWM4IEN	1: PWM4 中断使能 (需 IEN0 EPWM=1); 0: 关闭 PWM4 中断
3	PWM3IEN	1: PWM3 中断使能 (需 IEN0 EPWM=1); 0: 关闭 PWM3 中断
2	PWM2IEN	1: PWM2 中断使能 (需 IEN0 EPWM=1); 0: 关闭 PWM2 中断
1	PWM1IEN	1: PWM1 中断使能 (需 IEN0 EPWM=1); 0: 关闭 PWM1 中断
0	PWM0IEN	1: PWM0 中断使能 (需 IEN0 EPWM=1); 0: 关闭 PWM0 中断



10.2.6. PWMINTF (A4H)

A4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMINTF	-	FBIF	PWM5IF	PWM4IF	PWM3IF	PWM2IF	PWM1IF	PWM0IF
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	-	0	0	0	0	0	0	0

位编号	位符号	说明
7	-	-
6	FBIF	硬件置 1 表示有紧急故障中断请求，需要软件清 0
5	PWM5IF	硬件置 1 表示对应的 PWM 有中断请求，需要软件清 0。 FBIF, PWMnIF 组合生成 PWM 中断送到中断控制器
4	PWM4IF	
3	PWM3IF	
2	PWM2IF	
1	PWM1IF	
0	PWM0IF	

10.2.7. PWMFBCON0 (A5H)

A5H	Bit7-6	Bit5-4	Bit3-2	Bit1	Bit0
PWMFBCON0	-	GPIOFBCON[1:0]	-	GPIOFBEN	ADCFBEN
读/写	-	读/写	-	读/写	读/写
复位值	-	0	-	0	0

位编号	位符号	说明
7-6	-	-
5	GPIOFBCON[1]	00/11: P14 上升沿触发
4	GPIOFBCON[0]	01: P14 下降沿触发 10: P14 上下沿触发
3-2	-	-
1	GPIOFBEN	0: 关闭 P14 紧急故障功能 1: 打开 P14 紧急故障功能，P1.4 的上下沿 () 发生会停止 PWM 工作，清 PWMRUN 和计数器，同时产生 FB 中断。需要软件清 FBIF 并写 PWMRUN 才能恢复 PWM 工作
0	ADCFBEN	0: 关闭 ADC 比较紧急故障功能

		1: 打开 ADC 比较紧急故障功能，ADC 比较触发会停止 PWM 工作，清 PWMRUN 和计数器，同时产生 FB 中断。需要软件清 FBIF 并写 PWMRUN 才能恢复 PWM 工作 参考 ADC 数据比较功能
--	--	--

10.2.8. PWMFBCON1 (A6H)

A6H	Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBCON1	-	PWM5FBD	PWM4FBD	PWM3FBD	PWM2FBD	PWM1FBD	PWM0FBD
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值	-	0	0	0	0	0	0

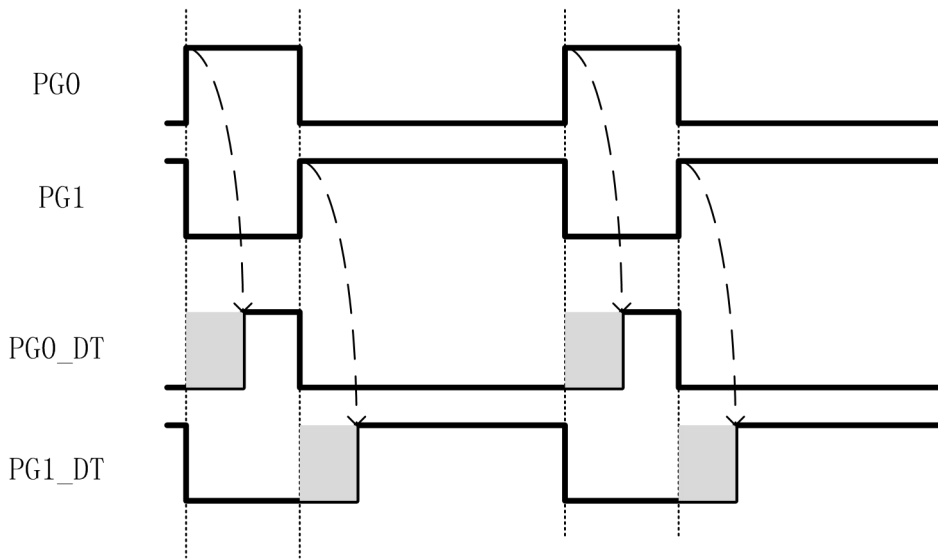
位编号	位符号	说明
7-6	-	-
5	PWM5FBD	紧急故障后，PWMn 输出对应的 PWMnFBD 的值。FBIF 清 0 后恢复 PWMn 输出
4	PWM4FBD	
3	PWM3FBD	
2	PWM2FBD	
1	PWM1FBD	
0	PWM0FBD	

10.2.9. PWMDTCN (9EH)

9EH	Bit7-4	Bit3	Bit2	Bit1	Bit0
PWMDTCN	PWMDTCYC[11:8]	-	PWM45DTEN	PWM23DTEN	PWM01DTEN
读/写	读/写	-	读/写	读/写	读/写
复位值	0	-	0	0	0

位编号	位符号	说明
7	PWMDTCYC[11]	PWMDTCYC 高 4 位
6	PWMDTCYC[10]	
5	PWMDTCYC[9]	
4	PWMDTCYC[8]	
3		
2	PWM45DTEN	1: PWM4、PWM5 死区控制使能
1	PWM23DTEN	1: PWM2、PWM3 死区控制使能

0	PWM01DTEN	1: PWM0、PWM1 死区控制使能
---	-----------	---------------------



10. 2. 10. PWMDTCYC (96H)

96H	Bit7-0
PWMDTCYC	PWMDTCYC[7:0]
读/写	读/写
复位值	0

位编号	位符号	说明
7	PWMDTCYC[7]	PWMDTCYC[11:0]:PWM 死区控制计数周期, 以 PWM 模块输入时钟计数 PWM 运行期间不能修改这组值
6	PWMDTCYC[6]	
5	PWMDTCYC[5]	
4	PWMDTCYC[4]	
3	PWMDTCYC[3]	
2	PWMDTCYC[2]	
1	PWMDTCYC[1]	
0	PWMDTCYC[0]	

10. 2. 11. PWMPH/PWMPH (CE/C4H)

	Bit7-0
PWMPH(CEH)	PWMPH[7:0]

PWMPH(C4H)	PWMP[15:8]
读/写	读/写
复位值	0

位编号	位符号	说明
	PWMP[15:0]	PWM 周期寄存器。 下一周期起效 如果出现 PWMP 不大于 PWMD 的情况，PGn 输出 0

10.2.12. PWM0DL/PWM0DH (C5/C6H)

	Bit7-0
PWM0DL(C5H)	PWMD0[7:0]
PWM0DH(C6H)	PWMD0[15:8]
读/写	读/写
复位值	0

位编号	位符号	说明
	PWMD0[15:0]	PWM 比较寄存器用于调整 PWM0 占空比。 下一周期起效

10.2.13. PWM1DL/PWM1DH (BC/BDH)

	Bit7-0
PWM1DL(BCH)	PWMD1[7:0]
PWM1DH(BDH)	PWMD1[15:8]
读/写	读/写
复位值	0

位编号	位符号	说明
	PWMD1[15:0]	PWM 比较寄存器用于调整 PWM1 占空比。 下一周期起效

10.2.14. PWM2DL/PWM2DH (BE/B4H)

	Bit7-0
PWM2DL(BEH)	PWMD2[7:0]
PWM2DH(B4H)	PWMD2[15:8]
读/写	读/写
复位值	0

位编号	位符号	说明
	PWMD2[15:0]	PWM 比较寄存器用于调整 PWM2 占空比。 下一周期起效

10.2.15. PWM3DL/PWM3DH (B5/B6H)

	Bit7-0
PWM3DL(B5H)	PWMD3[7:0]
PWM3DH(B6H)	PWMD3[15:8]
读/写	读/写
复位值	0

位编号	位符号	说明
	PWMD3[15:0]	PWM 比较寄存器用于调整 PWM3 占空比。 下一周期起效

10.2.16. PWM4DL/PWM4DH (AB/ACH)

	Bit7-0
PWM4DL(ABH)	PWMD4[7:0]
PWM4DH(ACH)	PWMD4[15:8]
读/写	读/写
复位值	0

位编号	位符号	说明
	PWMD4[15:0]	PWM 比较寄存器用于调整 PWM4 占空比。 下一周期起效

10. 2. 17. PWM5DL/PWM5DH (AD/AEH)

	Bit7-0
PWM5DL(ADH)	PWMD5[7:0]
PWM5DH(AEH)	PWMD5[15:8]
读/写	读/写
复位值	0

位编号	位符号	说明
	PWMD5[15:0]	PWM 比较寄存器用于调整 PWM5 占空比。 下一周期起效

11. 蜂鸣器

EN8F5113 系列内建蜂鸣器信号发生器，可软件配置输出各种不同频率的方波：

$F_{BEEP}=F_{IHRC}/2^M$ ，（M 可选择：1/2/3/4/5/6/7/8/9/10/11/12/13/14）。

寄存器

蜂鸣器控制寄存器 BEEPCTR:

86H	Bit7-4	Bit3-0
BEEP	-	BEEPSEL[3: 0]
读/写	-	读/写
复位值	-	0H

编号	位符号	说明																																						
7-4	-	-																																						
3-0	BEEPSEL[3: 0]	BEEP 电路使用 IHRC 时钟分频: $F_{BEEP}=F_{IHRC}/2^M$																																						
		<table border="1"> <thead> <tr> <th>BEEPSEL</th> <th>M</th> <th>BEEPSEL</th> <th>M</th> <th>BEEPSEL</th> <th>M</th> <th>BEEPSEL</th> <th>M</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>14</td> <td>0100</td> <td>10</td> <td>1000</td> <td>6</td> <td>1100</td> <td>2</td> </tr> <tr> <td>0001</td> <td>13</td> <td>0101</td> <td>9</td> <td>1001</td> <td>5</td> <td>1101</td> <td rowspan="3">1</td> </tr> <tr> <td>0010</td> <td>12</td> <td>0110</td> <td>8</td> <td>1010</td> <td>4</td> <td>1110</td> </tr> <tr> <td>0011</td> <td>11</td> <td>0111</td> <td>7</td> <td>1011</td> <td>3</td> <td>1111</td> </tr> </tbody> </table>	BEEPSEL	M	BEEPSEL	M	BEEPSEL	M	BEEPSEL	M	0000	14	0100	10	1000	6	1100	2	0001	13	0101	9	1001	5	1101	1	0010	12	0110	8	1010	4	1110	0011	11	0111	7	1011	3	1111
		BEEPSEL	M	BEEPSEL	M	BEEPSEL	M	BEEPSEL	M																															
		0000	14	0100	10	1000	6	1100	2																															
		0001	13	0101	9	1001	5	1101	1																															
		0010	12	0110	8	1010	4	1110																																
0011	11	0111	7	1011	3	1111																																		

12. 增强型串口 (EUART0/1)

12.1. 特性

- EUART0/1 均自带波特率发生器
- EUART0 有四种工作模式
- EUART1 有两种工作模式

12.2. 寄存器

EUART 使能控制寄存器 PORTMUX_2:

D8H	Bit7-4	Bit3	Bit2	Bit1	Bit0
PORTMUX_2	-	SPIPORTEN	I2CPORTEN	UART1PORTEN	UART0PORTEN
读/写	-	读/写	读/写	读/写	读/写
复位值	-	0	0	0	0

位编号	位符号	说明
7-4	-	-
3	SPIPORTEN	0: P1.0、P1.1、P1.2、P1.3; 1:在第一复用没开启的前提下, 这4个 PAD 为 SPI_NSS、SPI_SCK、SPI_MOSI、SPI_MISO
2	I2CPORTEN	0: P0.5、P0.6; 1:在第一复用没开启的前提下, 这两个 PAD 为 SDA、SCL
1	UART1PORT_EN	0: P2.0、P2.1; 1:在第一复用没开启的前提下, 这两个 PAD 为 UART1 TX0、RX0
0	UART0PORTEN	0: P1.6、P1.7; 1:在第一复用没开启的前提下, 这两个 PAD 为 UART0 TX0、RX0

EUART0 控制寄存器 S0CON:

98H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0CON	SM0	SM1	SM20	RENO	TB80	RB80	TIO	RIO
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

编号	位符号	说明																
7	SM0	EUART0 模式选择控制位:																
6	SM1		<table border="1"> <thead> <tr> <th>SM0</th> <th>SM1</th> <th>Mode</th> <th>描述</th> <th>Baund Rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Shift register</td> <td>SYSCLK/12</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>8-bit UART</td> <td>可配置</td> </tr> </tbody> </table>	SM0	SM1	Mode	描述	Baund Rate	0	0	0	Shift register	SYSCLK/12	0	1	1	8-bit UART	可配置
			SM0	SM1	Mode	描述	Baund Rate											
			0	0	0	Shift register	SYSCLK/12											
0	1	1	8-bit UART	可配置														

		1	0	2	9-bit UART	SYSCLK/16
		1	1	3	9-bit UART	可配置
5	SM20	EUART0 多机通讯使能控制位: 0: 关闭多机通讯功能, 1: 使能多机通讯功能只接收 RB80 = 1 的 9 位数据				
4	REN0	EUART0 接收允许使能控制位: 0: 不允许串口 0 接收 1: 允许串口 0 数据接收				
3	TB80	EUART0 Mode2,3 第 9 位发送数据位				
2	RB80	EUART0 Mode2,3 第 9 位接收数据位				
1	TI0	EUART0 发送中断标志位: 当发送数据完成后, 硬件置位, 必须由软件清 0				
0	RI0	EUART0 接收中断标志位: 当完成一次数据接收, 硬件置位, 必须由软件清 0				

EUART1 控制寄存器 S1CON:

9BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1CON	SM	-	SM21	REN1	TB81	RB81	TI1	RI1
读/写	读/写	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

编号	位符号	说明												
7	SM	EUART1 模式选择控制寄存位: <table border="1" style="margin-left: 20px;"> <tr> <th>SM</th> <th>Mode</th> <th>描述</th> <th>Baund Rate</th> </tr> <tr> <td>0</td> <td>A</td> <td>9-bit UART</td> <td>可配置</td> </tr> <tr> <td>1</td> <td>B</td> <td>8-bit UART</td> <td>可配置</td> </tr> </table>	SM	Mode	描述	Baund Rate	0	A	9-bit UART	可配置	1	B	8-bit UART	可配置
SM	Mode	描述	Baund Rate											
0	A	9-bit UART	可配置											
1	B	8-bit UART	可配置											
6	-	-												
5	SM21	EUART1 多机通讯使能控制位: 0: 关闭多机通讯功能 1: 使能多机通讯功能, 只接收 RB81= 1 的 9 位数据												
4	REN1	EUART1 接收允许使能控制位: 0: 不允许串口 1 接收 1: 允许串口 1 数据接收												
3	TB81	EUART1 第 9 位发送数据位												
2	RB81	EUART1 第 9 位接收数据位												
1	TI1	EUART1 发送中断标志位: 当发送数据完成后, 硬件置位, 必须由软件清 0												
0	RI1	EUART1 接收中断标志位: 当完成一次数据接收, 硬件置位, 必须由软件清 0												

EUART0 数据寄存器:

99H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SOBUF	SOBUF.7	SOBUF.6	SOBUF.5	SOBUF.4	SOBUF.3	SOBUF.2	SOBUF.1	SOBUF.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

编号	位符号	说明
7-0	SOBUF [7: 0]	EUART0 数据寄存器: 读 SOBUF 返回 UART0 接收到的数据, 写 SOBUF 启动 UART0 数据发送

EUART1 数据寄存器:

9CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1BUF	S1BUF.7	S1BUF.6	S1BUF.5	S1BUF.4	S1BUF.3	S1BUF.2	S1BUF.1	S1BUF.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

编号	位符号	说明
7-0	S1BUF [7: 0]	EUART1 数据寄存器: 读 S1BUF 返回 UART1 接收到的数据, 写 S1BUF 启动 UART1 数据发送

EUART0 波特率配置寄存器:

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SORELL(AAH)	SORELL.7	SORELL.6	SORELL.5	SORELL.4	SORELL.3	SORELL.2	SORELL.1	SORELL.0
SORELH(BAH)	-	-	--		-	-	SORELH.1	SORELH.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

编号	位符号	说明
7-0	SOBUF [7: 0]	EUART0 Mode1,3 波特率配置寄存器: SORELH +SORELL 构成波特率设置 SOREL[9:0] Baud rate =
1-0	SORELH[1: 0]	

EUART1 波特率配置寄存器:

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1RELL(9DH)	S1RELL.7	S1RELL.6	S1RELL.5	S1RELL.4	S1RELL.3	S1RELL.2	S1RELL.1	S1RELL.0
S1RELH(BBH)	-	-	-	-	-	-	S1RELH.1	S1RELH.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

编号	位符号	说明
7-0	S1RELL[7: 0]	EUART1 波特率配置寄存器:
1-0	S1RELH[1: 0]	S1RELH +S1RELL 构成波特率设置 S1REL[9:0] Baud rate =

12.3. 数据接收&发送

串口有两个独立的 Buffer，一个发送 Buffer,一个接收 Buffer。

向 S0BUF 或 S1BUF 写数据启动串口数据发送，读 S0BUF 或 S1BUF 返回接收 Buffer 的数据。

串口可连续的发送数据或接收数据。

接收 Buffer 可以缓存 1 byte 数据，避免在 CPU 未完成第一 byte 数据读取，此时第 2 byte 数据传输已完成的情况下，数据会丢失。

12.4. EUART0 工作模式

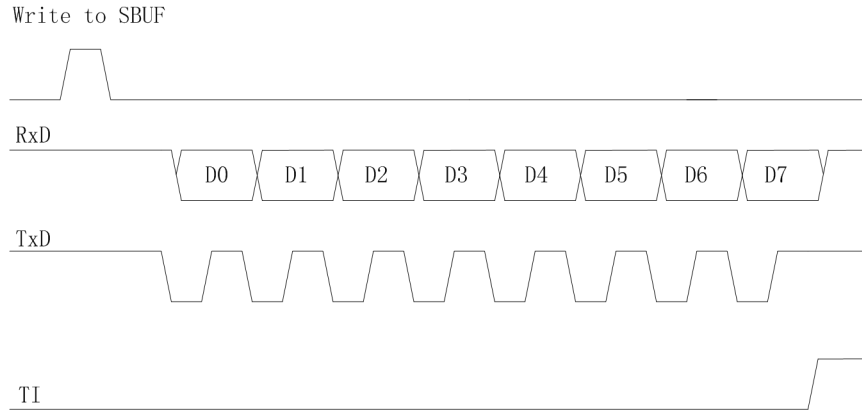
EUART0 有四种工作模式通过 SM0, SM1 选择

SM0	SM1	Mode	描述	Baud Rate
0	0	0	Shift register	SYSCLK /12
0	1	1	8-bit UART	可配置
1	0	2	9-bit UART	SYSCLK/16
1	1	3	9-bit UART	可配置

➤ 方式 0: 同步，半双工通讯

方式 0 支持与外部设备的同步通信。在 RX 引脚上收发串行数据，TX 引脚发送移位时钟。在这个方式中，每帧收发 8 位，低位先接收或发送。

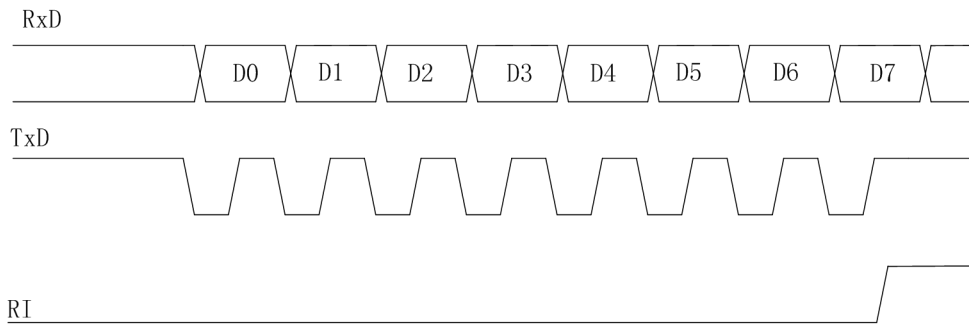
任何将 SBUF 作为目标寄存器的写操作都会启动发送，下一个系统时钟 TX 控制块开始发送。数据转换发生在移位时钟下降沿，移位寄存器内容逐次从左往右移位，空位置 0。发送完成后，TX 控制模块停止发送操作，然后在下一个系统时钟的上升沿将 TI 置位。



Send Timing of Mode 0

图 12-0

REN 置 1 和 RI 清 0 初始化接收。在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有的 8 位数据都移到移位寄存器后，RX 控制块停职接收，下一个系统时钟上升沿 RI 置位，直到软件清零才允许下一次接收。



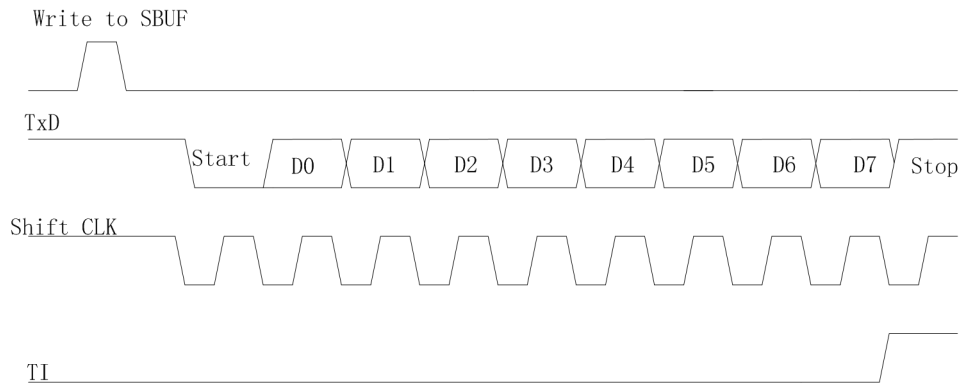
Receive Timing of Mode 0

图 12-1

➤ 方式 1: 8 位 EUART, 可变波特率, 异步全双工

方式 1 提供 10 位全双工异步通信, 10 位由一个起始位, 8 个数据位, 一个停止位组成。在接收时, 8 个数据位存在 SBUF 中, 停止位储存在 RB8 中。

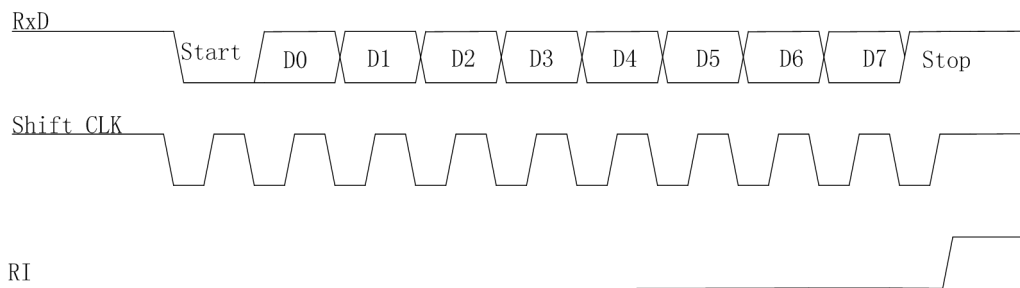
任何将 SBUF 作为目标寄存器的写操作都会启动发送。起始位首先在 TX 引脚上移出, 然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据位都发送完后, 停止位在 TX 引脚上移出, 在停止位发出的同时 TI 标志置位发出中断请求。



Send Timing of Mode 1

图 12-2

只有 REN 置位时才允许接收。当 RX 引脚检测到下降沿时串行口开始接收串行数据。若起始位有效，则移入移位寄存器，并接着移入其他位到移位寄存器。8 个数据位和 1 个停止位移入之后，若此时 RI=0，且 SM2=0 或者接收到的停止位为 1，（若不满足则丢掉此帧）则将停止位装入 RB8，8 个数据位装入 SBUF，随后 RI 被置位。这时，接收器继续探测 RX 的下一个下降沿。用户需要软件清零 RI，然后才能再次接收。



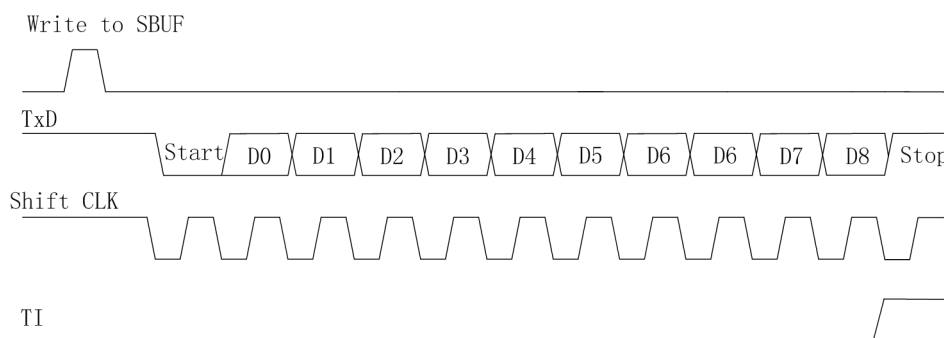
Receive Timing of Mode 1

图 12-3

➤ 方式 2：9 位 EUART，固定波特率，异步全双工

方式 2 提供 11 位全双工异步通信，波特率固定为系统时钟的 1/16。一帧由一个起始位，8 个数据位，一个可编程位，一个地址位组成。

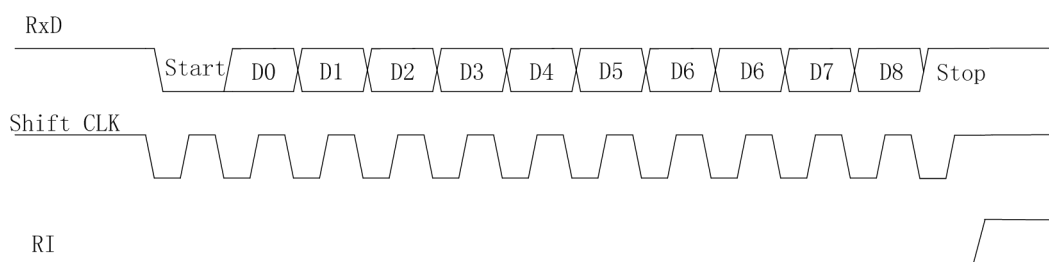
任何将 SBUF 作为目标寄存器的写操作都会启动发送，同时将 TB8 载入发送移位寄存器中的第九位。起始位首先在 TX 引脚移出，然后是 9 位数据位。在所有数据发送完成后，停止位在 TX 引脚上移出，在停止位开始发送时 TI 标志置位发出中断请求。



Send Timing of Mode 2

图 12-4

只有 REN 置位时才允许接收。当 RX 引脚检测到下降沿时串行口开始接收串行数据。若起始位有效，则移入移位寄存器，并接着移入其他位到移位寄存器。9 个数据位和 1 个停止位移入之后，若此时 RI=0，且 SM2=0 或者接收到的停止位为 1，（若不满足则丢掉此帧）则将第九位装入 RB8，8 个数据位装入 SBUF，随后 RI 被置位。这时，接收器继续探测 RX 的下一个下降沿。用户需要软件清零 RI，然后才能再次接收。



Receive Timing of Mode 2

图 12-5

➤ 方式 3: 9 位 EUART,可变波特率, 异步全双工

方式 3 使用方式 2 的传输协议以及方式 1 的波特率产生方式

12.5. EUART1 工作模式

EUART1 有两种工作模式通过 SM 选择

SM	Mode	描述	Baund Rate
0	A	9-bit UART	可配置
1	B	8-bit UART	可配置

EUART1 的这两种方式可以参考 EUART0 的方式 3 和方式 1

12.6. 多机通讯

EUART0 的方式 2 和方式 3 以及 EUART1 的方式 A 具有多机通讯功能。在采用多机通讯系统时，当主机要发送一数据块给数个从机中的一个时，先发送一个地址字节，以寻址目标从机。地址字节与数据字节可用第 9 数据位来区别，地址字节的第 9 位为 1，数据字节的第 9 位为 0。接收方根据第 9 位信息判断是否接收，多机通讯过程如下：

EN8F5113 系列设置为主机通信发送流程如下：

- 1) 设置为 9 bit Mode, 发送接收方地址，且置 TB8 = 1,
- 2) 根据自定的协议发送数据，且置 TB8 = 0。

EN8F5113 系列设置为主机通信接收流程如下：

- 1) 设置 SM2=0，（无条件接收全部数据），主机根据自定协议解析数据。

EN8F5113 系列设置为从机通信接收流程如下：

- 1) 置从机的 SM2=1，处于只接收地址帧状态，此时 EN8F5113 系列只会接收第 9bit 为 1 的地址数据；
- 2) 当接收到数据时，软件判断是否与本机设定的串口地址匹配；
- 3) 所有从机接收到地址帧后，各自将接收的地址与本机地址比较：若匹配即为目标从机，清除 SM2=0，准备接收主机即将发送的数据帧，接收完毕后再置 SM2=1；若不匹配，保持 SM2=1，忽略接下来的所有数据帧，不产生中断请求，直到接收到地址帧再次进行比较确认。

12.7. 波特率

◆ EUART0 modes 1 and 3

Baud rate =

◆ EUART1

Baud rate =

SYSClk 选择内部高频 IHRC，常见波特率 SxREL 配置值以及实际误差如下表

目标波特率	SxREL	实际波特率	误差
115200	1015	111111	3.5%
57600	1007	58824	-2.1%
38400	998	38462	-0.2%
19200	972	19231	-0.2%
9600	920	9615	-0.2%

13. I2C 控制器

EN8F5113 系列内部集成电路（I2C）总线是一种简单的双向 2 线总线。提供了 8051 和 I2C 总线的接口。

有主、从控制器组成，两部分控制寄存器复用情况如下表格：

寄存器名	地址	对应主控制器	对应从控制器
MSCON	D7H	MCON	STCON
MSRXBUF	CFH	MRXBUF	SRXBUF
MSTXBUF	C7H	MTXBUF	STXBUF
MPRESC	BFH	MPRESC	-
MSSTAT0	B7Hi2i	MSTAT0	SSTAT0
MSSTAT1	AFH	MSTAT1	SSTAT1
MSIEN0	A7H	MIEN0	SIEN0
MSIEN1	9FH	MIEN1	SIEN1
MSADDR	97H	MCADDR	SSADDR

I2C Slave/Master 选择控制寄存器 TOPCON(8FH):

8FH	7-3	2	1	0
TOPCON	-	I2CMRST	I2CSRST	I2CSEL
读/写	-	读/写	读/写	读/写
RESET	-	0	0	0

Bit Number	Bit Mnemonic	Function
7-3	-	-
2	I2CMRST	软件写 1 复位 I2C Master, 写 0 结束复位;
1	I2CSRST	软件写 1 复位 I2C Slave, 写 0 复位结束;
0	I2CSEL	0: I2C Slave select; 1: I2C Master select.

13.1. I2C Slave 控制器

使用前将 I2CSEL 置为 0。从控制器处理所有必要的功能，以响应从 I2C 主控制器的请求，支持快速和标准传输速率。

主要功能：

- I2C 位寻址。
- Slave 操作。
- 时钟展宽与等待状态生成。
- 工作范围广泛的输入频率。
- 中断的产生。

接收和发送数据分别存储在接收缓冲区和传输缓冲器中。每个缓冲区中只能存储一个字节。在 I2C 传输期间，CPU 需要定期读取接收缓冲区并定期写入传输缓冲区

13.1.1. 功能框图

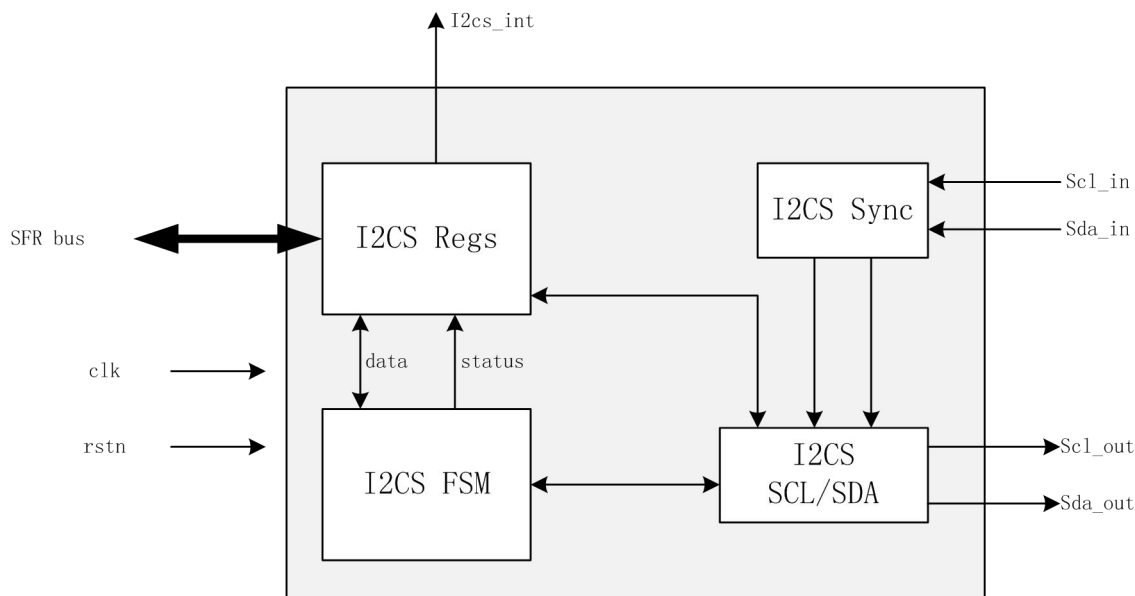


图 13-0

13.1.2. 控制寄存器

8 个 SFR 寄存器如下:

Mnemonic	Address	Description	Reset value
STCON	D7H	I2CS 传输控制寄存器	00h
SRXBUF	CFH	I2CS 接受缓冲区	00h
STXBUF	C7H	I2CS 传输缓冲区	00h
SSTAT0	B7H	I2CS 状态寄存器 0	00h
SSTAT1	AFH	I2CS 状态寄存器 1	00h
SIEN0	A7H	I2CS 中断使能寄存器 0	00h
SIEN1	9FH	I2CS 中断使能寄存器 1	00h
SSADDR	97H	I2CS 自身地址寄存器	00h

13.1.2.1. STCON 寄存器

I2CSlave 传输控制寄存器:

BIT	7	6	5	4-1	0
FIELD	-	I2CEN	SWS	-	TIG
RESET	0000 0000b				

Bit Number	Bit Mnemonic	Function
------------	--------------	----------

7	-	-
6	I2CEN	I2CS 使能. 设置为 1 时, 激活 I2CS; 清除为 0 时, 停用 I2CS
5	SWS	I2CS 等待状态 设置为 1 时, 在 RX 溢出时在 SCL 行上生成等待状态。 当清除为 0 时, I2CS 发出 “not acknowledge” 停止传输时, RX 溢出。
4:1	-	-
0	TIG	正在进行中的传输 当被硬件设置为 1 时, I2C 正在 I2C 总线进行传输。清楚, 否则被清除为 0。

13. 1. 2. 2. SRXBUF 寄存器(只读)

I2CSlave 接受 Buffer:

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	0000 0000b							

Bit Number	Bit Mnemonic	Function
7: 0	-	I2CS 接受数据

13. 1. 2. 3. STXBUF 寄存器(只读)

I2CSlave 传输 Buffer:

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	0000 0000b							

Bit Number	Bit Mnemonic	Function
7: 0	-	I2CS 传输数据

13. 1. 2. 4. SSTAT0 寄存器

I2CSlave 状态寄存器 0:

BIT	7	6	5-3	2	1	0
FIELD	-	GC	-	SUNF	SOVF	SNE
RESET	0000 0000b					

Bit Number	Bit Mnemonic	Function
7	-	
6	GC	一般调用 设置为 1，指示已检测到一般调用。
5:3	-	-
2	SUNE	传输不足 传输数据字节未就绪（传输缓冲区为空），而需要发送新数据字节。等待状态生成，直到数据可用为止。
1	SOVF	接收溢出 接收到的数据字节不能写入（接收缓冲区已满），同时接收到新的位。设置为 1 时，RX 溢出和 STCON.SWS = 0。它指示接收缓冲区在接收新字节时已满。一个不承认发送如果 STCON.SWS = 1，当一个新的字节被接收时，等待状态生成且 OVF 没有设置。
0	SNE	正常结束（没有错误的访问结束） 当成功访问结束时发送一个停止时将被设置。当一个新的 I2C 存取开始时自动清除。它也可以通过软件清除。

当 GC, UNF 或 SOVF 标志已设置，接收和传输过程中被禁用，直到 CPU 读取 SSTAT0 寄存器。此读操作自动清除这些标志。

这些中断源都可以被 SIEN0 寄存器设置为单独启用/禁用。输出信号 OTXRXINT 设置为“1”时，一个或多个中断源主动和启用。

当禁用中断发生，OTXRXINT 仍然不变，但相应的中断点还是被设置。当鉴别到一般调用时，Slave 控制器将 SSTAT0.GC 设置为“1”。CPU 必须处理接收到的数据作为一般调用信息。

13.1.2.5. SSTAT1 寄存器

I2CSlave 状态寄存器 1:

BIT	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

FIELD	-	-	STBE	-	STBF	SRBE	-	SRBF
RESET	0000 0000b							

Bit Number	Bit Mnemonic	Function
7-6	-	
5	STBE	I2CS 传输缓冲区是空的 当传输缓冲区为空时，设置为 1。当至少有一个字节是准备好数据传输时，这个标志被清除为 0。当 CPU 执行写访问 STXBUF 寄存器时，这个标志位也被清除。
4	-	
3	STBF	I2CS 传输缓冲区已满 当传输缓冲区满时设置为 1。当没有更多的写操作到传输缓冲区或内存执行（CPU 写请求 STXBUF 不考虑）时被设置。当传输缓冲区为空时，清除 0。当 I2C 传输控制器请求一个新的数据字节时，此标志将被清除。
2	SRBE	I2CS 接收缓冲区是空的 当接收缓冲区为空时设置为 1。没有更多的写操作到接收缓冲区或内存执行（CPU 读取请求 SRXBUF 不考虑）。一个新的数据字节的 TXRX 控制器接收时，这个标志位被清除。当至少一个接收到的数据字节可用时，清除为 0。
1	-	
0	SRBF	I2CS 接收缓冲区已满 当接收缓冲区满时设置为 1。这个标志被清除时，CPU 执行一个读操作 SRXBUF 寄存器。当接收缓冲区为空时，清除为 0。

这些中断源都可以被 SIEN1 寄存器设置为单独启用/禁用。输出信号 OFIFOINT 设置为“1”时，一个或多个中断源主动和启用。当禁用中断发生时，OFIFOINT 不变，但相应的中断位被设置。当设置消失时，这些中断源将被清除。

13.1.2.6. SIEN0 寄存器

I2CSlave 中断使能寄存器 0:

BIT	7	6	5	4	3	2	1	0
FIELD	-	EGC	-	-	-	ESUNF	ESOVF	ESNE

RESET	0000 0000b
-------	------------

Bit Number	Bit Mnemonic	Function
7	-	
6	EGC	I2CS 一般调用中断使能位 (SSTAT0.SGC) 清除时, 禁用 SSTAT0.SGC 点产生一个中断请求 设置时, 启用 SSTAT0.SGC 点产生一个中断请求
5:3	-	
2	ESUNF	I2CS 下溢中断使能位 清除时, 禁用 SSTAT0.SUNF 点产生一个中断请求 设置时, 启用 SSTAT0.SUNF 点产生一个中断请求
1	ESOVF	I2CS 溢出中断使能位 清除时, 禁用 SSTAT0.SOVF 点产生一个中断请求 设置时, 启用 SSTAT0.SOVF 点产生一个中断请求
0	ESNE	I2CS 正常结束中断使能位 清除时, 禁用 SSTAT0.SNE 点产生一个中断请求 设置时, 启用 SSTAT0.SNE 点产生一个中断请求

13.1.2.7. SIEN1 寄存器

I2CSlave 中断使能寄存器 1:

BIT	7	6	5	4	3	2	1	0
FIELD	-	-	ESBE	-	ESTBF	ESRBE	-	ESRBF
RESET	0000 0000b							

Bit Number	Bit Mnemonic	Function
7:6	-	
5	ESBE	I2CS 传输缓冲区空的中断使能位 清除时, 禁用 SSTAT1.STBE 点产生一个中断请求 设置时, 启用 SSTAT1.STBE 点产生一个中断请求
4	-	
3	ESTBF	I2CS 传输缓冲区满中断使能位 清除时, 禁用 SSTAT1.STBF 点产生一个中断请求 设置时, 启用 SSTAT1.STBF 点产生一个中断请求
2	ESRBE	I2CS 接收缓冲区空的中断使能位 清除时, 禁用 SSTAT1.SRBE 点产生一个中断请求 设置时, 启用 SSTAT1.SRBE 点产生一个中断请求
1	-	

0	ESRBF	I2CS 接收缓冲区满中断使能位 清除时，禁用 SSTAT1.SRBF 点产生一个中断请求 设置时，启用 SSTAT1.SRBF 点产生一个中断请求
---	-------	---

13.1.2.8. SSADDR 寄存器

I2CSlave 自身地址寄存器:

BIT	7	6	5	4	3	2	1	0
FIELD	-	SADDR						
RESET	0000 0000b							

Bit Number	Bit Mnemonic	Function
7	-	
6: 0	SADDR	7 位自地址 此寄存器必须在 I2C 传输开始之前写入。

13.2. I2C Master 控制器

使用前将 I2CSEL 置为 1。内部集成一个 I2C 主控制器提供了一个 8051 和 I2C 总线的接口。处理所有必要功能来建立和维护数据链路:

- 快速标准传输速率。
- I2C 位寻址。
- 简单主操作。
- 时钟展宽与等待状态生成。
- 工作范围广泛的输入频率。
- 中断的产生。

13.2.1. 功能框图

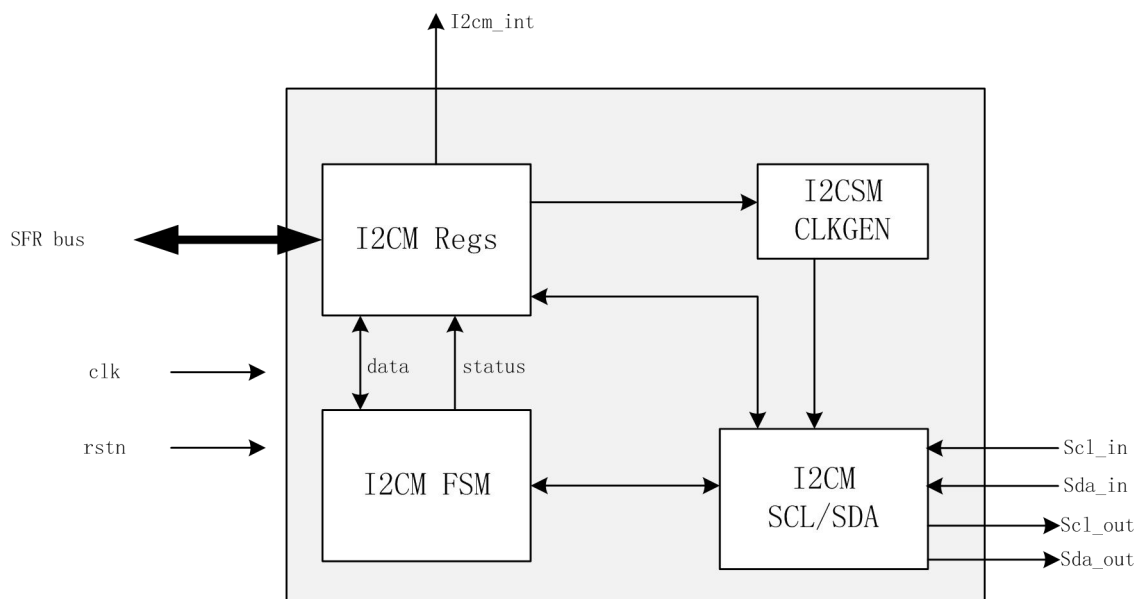


图 14-0

13.2.2. 寄存器

Mnemonic	Name	Address
MCON	I2CM 控制寄存器	D7H
MRXBUF	I2CM 接受缓冲器	CFH
MTXBUF	I2CM 传输缓冲器	C7H
MPRESC	I2CM Pre-scalar 时钟寄存器	BFH
MSTAT0	I2CM 状态寄存器 0	B7H
MSTAT1	I2CM 状态寄存器 1	AFH
MIEN0	I2CM 中断使能寄存器 0	A7H
MIEN1	I2CM 中断使能寄存器 1	9FH
MCADDR	I2CM 调用地址寄存器	97H

13.2.2.1. MCON 寄存器

I2C Master 控制寄存器:

BIT	7	6	5	4	3	2	1	0
FIELD	-	-	WAIT	-	STOP	SRST	STA	BUSY
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:6	-	-
5	WAIT	等待状态模式 “1”：当RX溢出时，在SCL线上生成等待状态。

		“0”：当RX溢出时，发送“Not Acknowledge”停止传输。
4	-	
3	STOP	产生停止条件 当设置此位时，当前字节正常结束，并在确认周期之后生成停止条件。当发送停止条件时，控制器将自动清除此位。
2	SRST	软件复位 一旦达到空闲状态，此位将自动清除。
1	STA	产生启动条件 当传输开始或检测到错误时，控制器将自动清除此位。
-	BUSY	BUSY标志位 当I2C帧在I2C总线上传输时，这个位被设置为“1”。

13.2.2.2. MRXBUF 寄存器

I2C Master Receive Buffer: Read Only

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:0	RXBUF	Data received by I2C

13.2.2.3. MTXBUF 寄存器

I2C 主传输缓冲器: Write Only

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:0	TXBUF	Data transmitted by I2C

13.2.2.4. MPRESC 寄存器

MPRESC register enables to generate *OSCL* output from a large range of CLK frequency.

BIT	7	6	5	4	3	2	1	0
FIELD								
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:0	PRESC	Clock pre scale register: $F_{scl}=F_{clk}/[10*(PRESC+1)]$

This register should not be written during a transmission.

13.2.2.5. MSTAT0 寄存器

I2C Master Status Register0.

BIT	7	6	5	4	3	2	1	0
FIELD	-	-	-	DNA	SANA	UNF	OVF	NEND
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:5	-	
4	DNA	未确认数据字节 传输期间未确认的数据字节。停止状态发送。
3	SANA	未接收从属地址 未接收从属地址。停止状态发送。
2	UNF	Under Flow 传输数据字节未就绪（传输缓冲区为空），而需要发送新的数据字节。发送停止条件。
1	OVF	Receive Overflow 接收到的数据字节无法写入（接收缓冲区已满），同时接收到新字节。发送一个不确认和一个停止条件。
0	NEND	Normal End (End of access with no error) 当成功访问结束时发送一个停止时设置。当一个新的I2C存取开始时自动清除。

这些中断源在对该寄存器的读访问之后自动清除。

当 DNA、SANA、UNF 或 OVF 标志被设置，接收和传输过程禁用直到 CPU 读 MASTAT1 寄存器。如果这些错误位设置，这个读操作自动复位 MASTAT1 寄存器和 MCON.ST 位。如果读操作没有错误位设置完成，MCON.STA 位不清零。

这些中断源都可以被 MIENO 寄存器设置为单独启用/禁用。OTXRINT 输出信号设置为“1”时，一个或多个中断源主动和启用。当禁用中断发生时，OTXRINT 不变，但相应的中断位被设置。

13.2.2.6. MSTAT1 寄存器

I2C Master Status Register1. Read only

BIT	7	6	5	4	3	2	1	0
FIELD	-	-	TBE	-	TBF	RBE	-	RBF
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:6	-	
5	DNA	Transmission buffer is empty “1”：传输缓冲区空。这个标志被清除时，CPU执行写访问给TXDATA寄存器。 “0”：至少有一个传输数据字节可用。
4	-	
3	TBF	Transmission buffer is full “1”：传输缓冲区满。没有更多的写操作为发射进行缓冲或内存（CPU写请求给TXDATA不考虑）。这个标志被清除时，一个新的数据字节的TXRX控制器的要求。 “0”：传输缓冲区是空的。
2	RBE	Reception buffer is empty “1”：接收缓冲区空。没有更多的写操作到接收缓冲区或进行内存（CPU读取请求RXDATA不考虑）。这个标志被清除时，一个新的数据字节的TXRX控制器接收。 “0”：至少有一个接收的数据字节可用。
1	-	
0	RBF	Reception buffer is full “1”：接收缓冲区满。这个标志被清除时，CPU执行一个读操作RXDATA登记。 “0”：接收缓冲区是空的

13.2.2.7. MIENO 寄存器

I2CM Interrupt Enable Register 0

BIT	7	6	5	4	3	2	1	0
FIELD	-	-	-	ETNA	ESANA	ETUR	EROVER	ENEND
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:5	-	
4	ETNA	Data byte Not Acknowledged Interrupt enable bit 清除时，禁用MSTAT0.DNA位产生一个中断请求 设置时，启用MSTAT0.DNA位产生一个中断请求
3	ESANA	Slave Address Not Acknowledged Interrupt enable bit 清除时，禁用MSTAT0.SANA位产生一个中断请求 设置时，启用MSTAT0.SANA位产生一个中断请求
2	ETUR	Underflow Interrupt enable bit 清除时，禁用MSTAT0.UNF位产生一个中断请求 设置时，启用MSTAT0.UNF位产生一个中断请求
1	EROVER	Overflow Interrupt enable bit 清除时，禁用MSTAT0.OVF位产生一个中断请求 设置时，启用MSTAT0.OVF位产生一个中断请求
0	ENEND	Normal End Interrupt enable bit 清除时，禁用MSTAT0.NEND位产生一个中断请求 设置时，启用MSTAT0.NEND位产生一个中断请求

13.2.2.8. MIEN1 寄存器

I2CM Interrupt Enable Register1

BIT	7	6	5	4	3	2	1	0
FIELD	-	-	ETBE	-	ETBF	ERBE	-	ERBF
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7:6	-	
5	ETBE	Transmission Buffer Empty Interrupt enable bit 清除时，禁用MSTAT1.TBE位产生一个中断请求 设置时，启用MSTAT1.TBE位产生一个中断请求
4	-	
3	ETBF	Transmission Buffer Full Interrupt enable bit

		清除时，禁用MSTAT1.TBF位产生一个中断请求 设置时，启用MSTAT1.TBF位产生一个中断请求
2	ERBE	Reception Buffer Empty Interrupt enable bit 清除时，禁用MSTAT1.RBE位产生一个中断请求 设置时，启用MSTAT1.RBE位产生一个中断请求
1	-	
0	ERBF	Reception Buffer Full Interrupt enable bit 清除时，禁用MSTAT1.RBF位产生一个中断请求 设置时，启用MSTAT1.RBF位产生一个中断请求

13.2.2.9. MCADDR 寄存器

I2CM Call Address Register

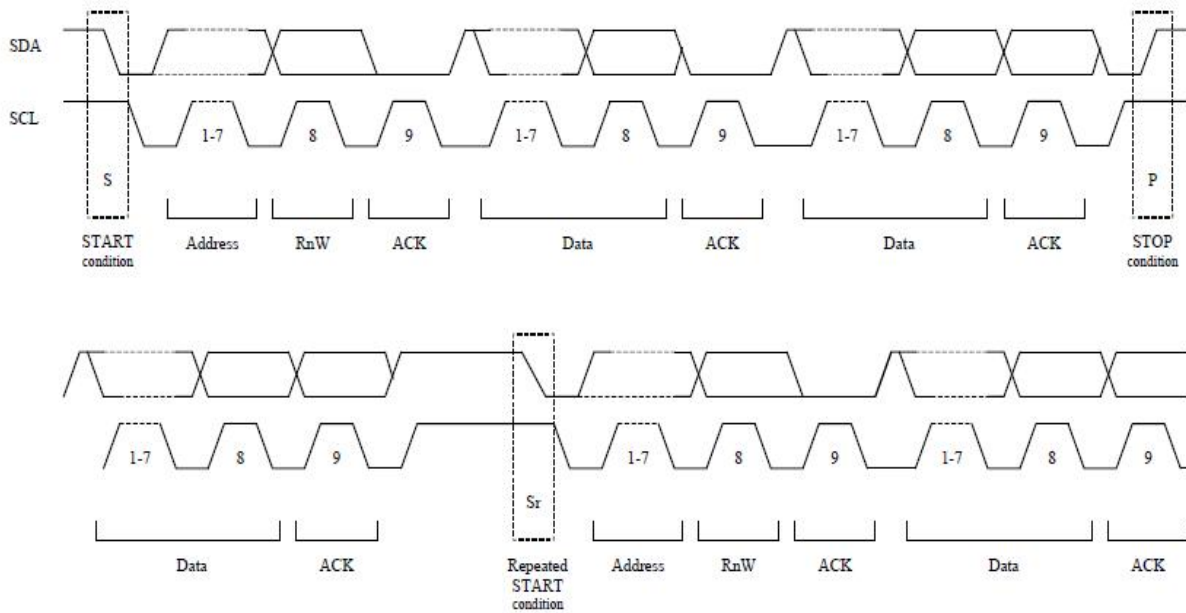
BIT	7	6	5	4	3	2	1	0
FIELD	RWN	CADDR						
RESET	0000 0000b							

Bit NO.	Bit Menemonic	Function
7	RWN	Read/write control bit for I2C transaction 设置时，从寻址从设备读取数据 清除时，将数据写入到寻址从设备
6:0	CADDR	7-bit Call Address 此寄存器必须在I2C传输开始之前写入。

13.3. I2C 帧数据格式

I2C 总线支持两种格式：7 位地址格式和 10 位的地址格式。

该 I2C 控制器只支持 7 位格式如下图所示：



START Condition : A HIGH to LOW transition on the SDA line while SCL is HIGH.
 STOP Condition : A LOW to HIGH transition on the SDA line while SCL is HIGH.
 DATA validity : The data on the SDA line must be stable during the HIGH period of SCL. The state of the data line can only change when SCL is LOW.

图 13-1

所有的字节在 SDA 线上长度都是 8 位。在 I2C 线路上传输的字节数是不受限制。每个字节后面跟着一个由接收器设置的应答位。数据是从最高位 (MSB) 首位开始传输。在开始条件 (S) 之后，发送一个从地址。这个地址长 7 位。第八位确定消息的方向 (R/WN): “0” 意味着 Master 将数据写入到一个选定的 Slave; “1” 意味着 Master 会从选择的 Slave 中读取数据。数据传输总是以停止位终止 (P)。但是，如果 Master 仍然希望在总线上通信，它可以生成重复启动 (SR)，这表示不生成第一个停止位而产生另一个开始。在这种传输中，可以进行各种读/写格式组合。

值得注意的是，两组八个地址 (0000xxx 和 1111xxx) 预留的用途如下表所示。

SLAVE ADDRESS	RnW Bit	Description
0000 000	0	General call address. It is used to address every device connected to I2C-bus.
0000 000	1	START byte(1)
0000 001	X	CBUS address(2).
0000 010	X	Reserved for different bus format.
0000 011	X	Reserved for future purposes.
0000 1XX	X	High Speed master code.
1111 1 XX	X	Reserved for future purposes.
1111 0XX	X	10 bit slave addressing.
(1) : No device allowed to acknowledge at the reception of the START byte.		
(2) : The CBUS address has been reserved to enable the intermixing of CBUS compatible and the I2C-bus compatible devices in the same system. I2C-bus compatible devices are not allowed to		

respond on reception of this address.

13.4. 应答

带有应答的数据传输是强制性的。与应答有关的时钟脉冲由 Master 产生。发射机在应答时钟脉冲期间释放 SDA 线（高电平）。接收机必须在确认脉冲期间拉下 SDA 线，以便在时钟脉冲的高周期期间保持稳定的 Low。

当 Slave 设备不接收从地址或数据时，数据线 SDA 必须由 Slave 置高。然后 Master 可以生成停止条件来中止传输。

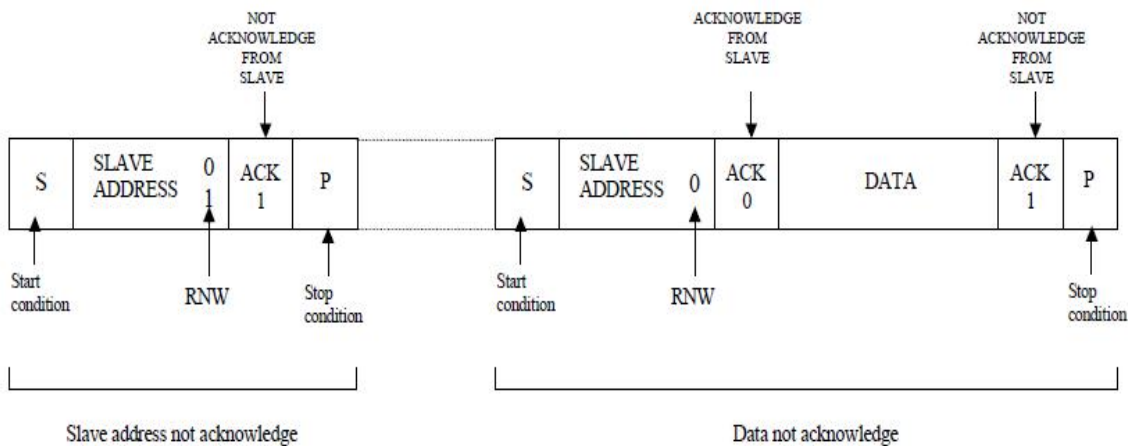


图 13-2 "not acknowledge" by slave device

如果主接收器与传输有关，它必须通过对最后一个字节不产生应答而将数据的结束信号发送到从发射机。Slave 将释放 SDA 线，让 Master 产生停止或重复的条件。

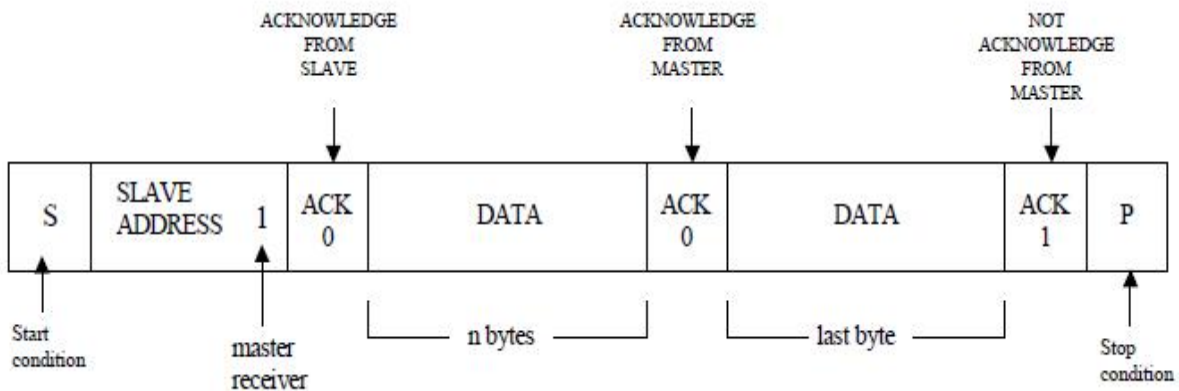


图 13-3 "not acknowledge" by Master (end of transmission)

13.5. 时钟

13.5.1 时钟系统

The input clock CLK is used to generate OSCL using PRESC register.

13.5.2. I2C 时钟

13.5.2.1. 时钟同步

对于该设备, 时钟同步将用于使接收设备在接收和确认字节后保持 SCL 线低, 从而迫使 Master 进入等待状态。这使得 Slave 设备获得更多的时间来存储所接收的字节或准备发送另一个字节。

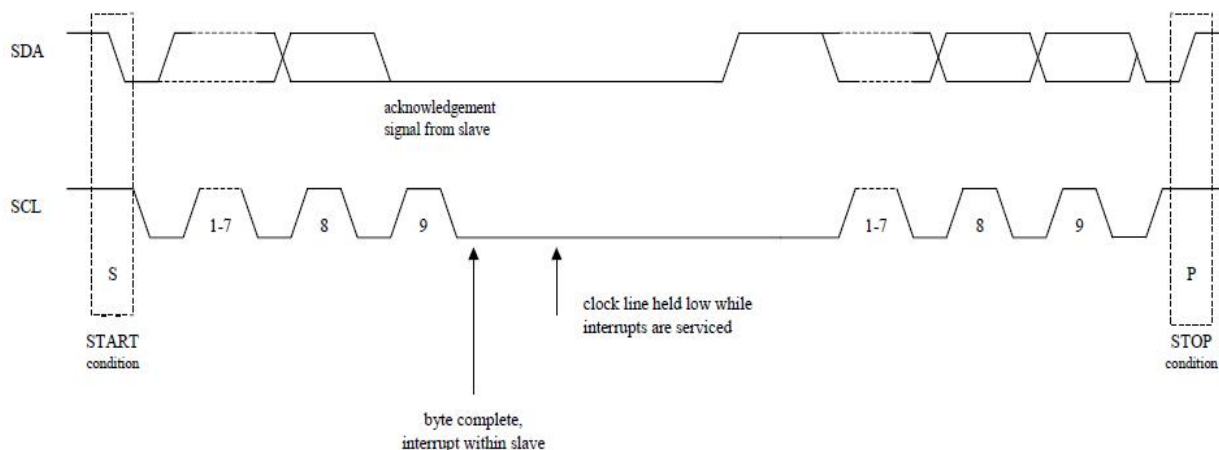


图 13-4 Clock synchronization as handshake.

时钟同步是利用 I2C 接口到 SCL 线连接进行 wired_and。这意味着 SCL 线上的高到低转换将导致相关设备开始计数它们的低周期。在他们自己的低周期结束时, 设备将设置他们的时钟为高。然而, 只要一个时钟仍处于低周期内, SCL 线将保持低电平。因此, SCL 线将被最长的低周期设备保持低电平。在此期间, 较短的低周期设备进入高等待状态。当所有相关设备都计算掉它们的低周期时, SCL 线将被释放并上升。设备时钟和 SCL 线之间没有什么区别, 所有设备都将开始计数它们的高周期。完成高周期的第一个设备将再次把 SCL 线拖低。这样, 就产生了一个同步时钟。

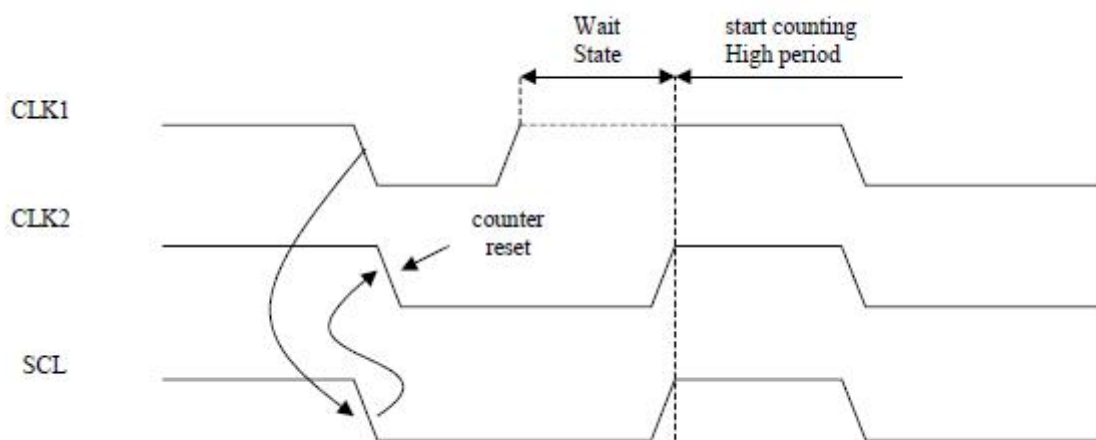


图 13-5 Clock synchronization

13.5.2.2 时钟产生

系统时钟用于产生I2C总线时钟（OSCL输出），其比率为5，如下图所示：

$$F_{oscl} = F_{clk} / (5 * 2 * (presc + 1))$$

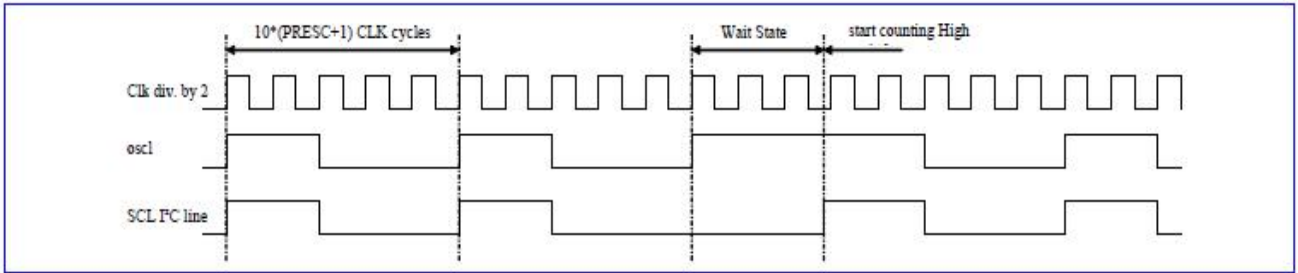


图14-7 OSCL output generation

MPRESC 寄存器可以从大范围的时钟频率产生 OSCL 输出。MPRESC 寄存器只能在两个 I2C 传输之间被设置。

写操作 MPRESC 寄存器是采取的帐户只有一般的有限状态机处于闲置状态。当 MPRESC 值被修改，时钟频率修正没有任何故障信号。注意，OSCL 输出保持高在空闲模式。

13.6. 主模式：传输

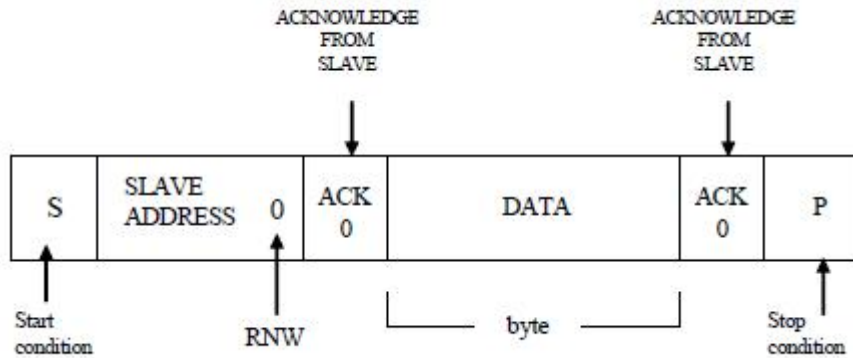


图14-8 Typical transmission.

13.6.1 初始化

在开始传输，CPU写从地址为MCADDR寄存器。注意，对于写请求，从地址的LSB必须设置为“0”。

CPU将要写一个字节的的数据到发送缓冲区（MTXBUF寄存器）在交易（必须注意避免溢出）。TBF（发送缓冲区满）或TBE（发送缓冲区空）标志可用来检查发送缓冲区的状态。

13.6.2 起点

初始化后，CPU可以通过设置在MCON寄存器中STA位开始传输。然后，主控制器在I2C总线上生成启动条件。当传输开始从站地址传输时，STA位将自动清除。

在开始发送之后，从地址寄存器加载在移位寄存器上，以在I2C总线上传输，主控制器请求第

一个数据字节发送。一旦从地址发送，主控制器就从从控制器等待从地址应答。

13.6.3 数据传输

如果从控制器返回从地址应答，主控制器将在移位寄存器中加载数据字节以在I2C总线上传输。如果这个数据字节不是最后一个，则主发送一个请求读取下一个数据字节。一旦数据字节被发送，主控制器就等待来自控制器的数据确认。在确认情况下，如果数据字节发送不是最后一个，则控制器发送另一个数据字节。

检测最后一个数据字节：

如果设置停止位，则数据字节是最后一个数据字节。

注意：由于传输缓冲区（1字节）的大小，第一个传输的数据字节也是最后一个数据字节。

13.6.4 停止重复启动

一旦发送了最后一个数据字节，主控制器就等待来自控制器的数据确认。在确认的情况下，如果 STA 被 CPU 设置为“1”，则控制器将产生重复启动，以便访问另一个从设备或改变传输的方向（主模式接收），然后发送停止条件以完成通信。一旦停止状态或重复启动，停止位将自动清除。

在重复启动的情况下，CPU 必须在当前传输结束之前初始化下一个传输（从地址、长度和数据字节的写入）。

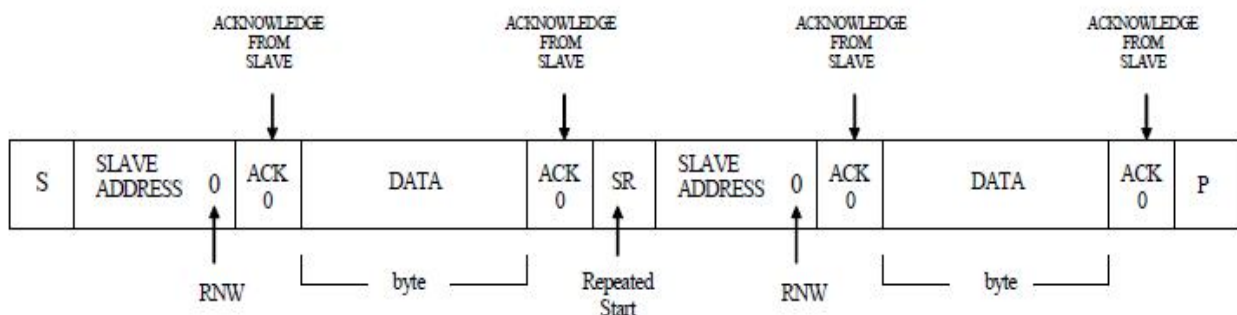


图14-9 Repeated Start or Stop condition after last byte.

13.6.5 传输错误

13.6.5.1 Not acknowledge from Slave Controller

如果从地址不是由主控制器承认，中断传输发送一个停止条件，并设置该标志。如果数据不被从控制器接收，主通过发送停止条件中断传输，并设置DNA标志。

13.6.5.2 Transmit Underflow

如果没有数据字节是有效的从发送缓冲器时，控制器需要发送一个字节的的数据，主中断传输发送一个停止条件，并设置的标志。

这种下溢发生在传输缓冲区为空时（CPU 没有及时填充传输缓冲区）。

13.6.5.3 End of error

当检测到错误时，OTXRINT 输出设置是否启用相应的中断源。控制器被阻塞直到 MSTAT0 寄存器由 CPU 读取。这个读操作重置 MSTAT0 寄存器和 STA 位禁用潜在的重复启动。为了进行传输，CPU 必须只设置 STA。重新启动从一开始就同一传输，CPU 必须设置软件复位，然后设置 STA 和填充 MTXBUF。

13.7. 主模式：接收

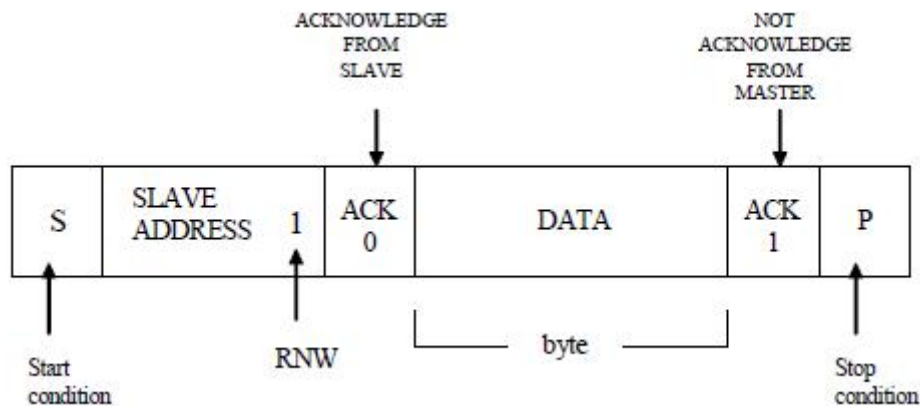


图14-10

13.7.1 初始化

在开始接收，CPU写从地址到MCADDR寄存器。注意，对于一个读请求，从地址的LSB必须设置为“1”。

CPU将接收到的数据字节读取接收缓冲区中（mrxbuf登记）经常在交易（必须注意避免溢出）。

13.7.2 起点

初始化后，CPU可以通过设置在MCON寄存器STA位开始接受登记。然后，主控制器在I2C总线上生成启动条件。传输开始时，STA位将自动清除。

现在，从地址加载在移位寄存器中，以便在I2C总线上传输。一旦从地址发送，主控制器就从从控制器等待从地址应答。如果从控制器返回从地址确认，主控制器正在等待第一个接收的数据字节。

13.7.3 接收

一旦接收到一个数据字节，它就由主控制器在接收端存储。更重要的是，如果接收到的数据字节不是最后一个，主控制器在I2C总线上发送一个确认。否则，将发送一个“不确认”来指示它是最后一个读取请求，而从控制器必须释放I2C总线以允许生成停止条件。

在数据确认传输，一个新的数据接收可以做到和CPU可以读取存储的数据使用mrxbuf登记。

检测最后一个数据字节：

如果设置停止位，则数据字节是最后一个数据字节。

注意：由于接收缓冲区的大小（1字节），第一个接收到的数据字节也是最后一个数据字节。

13.7.4 停止重复启动

在“数据未确认”传输之后，如果cpu将STA设置为“1”，则控制器将产生重复启动，以便访问另一个从设备或改变传输的方向（主模式传输），然后发送停止条件以完成通信。一旦停止状态或重复启动，停止位将自动清除。

在重复启动的情况下，CPU 必须在当前接收结束之前初始化下一个传输（从地址、长度和数据字节的写入）。

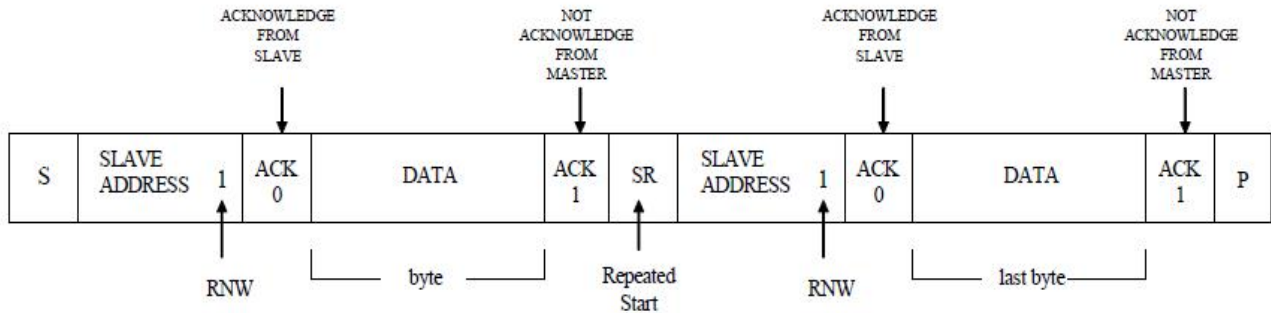


图14-11 Repeated Start or Stop condition after last byte

13.7.5 接收错误

13.7.5.1 Not acknowledge from Slave Controller

如果从地址不是由主控制器承认，中断传输发送一个停止条件，并设置SANA标志。

13.7.5.2 Receive Overflow

当一个字节的数据被接收，该收发控制器检查以前的数据字节已被处理。如果不是这样（RXBUF溢出），主中断接收发送“不承认”和停止条件，设置OVF标志。

13.7.5.3 End of error

当检测到错误时，OTXRINTt 输出设置是否启用相应的中断源。控制器被阻塞直到 MSTAT0 寄存器由 CPU 读取。这个读操作重置 MSTAT0 寄存器和 STA 位禁用一个潜在的重启。如果 CPU 想丢弃以前接收到的数据字节，它必须设置软件重置。要重新启动相同的传输，CPU 只需设置 STA。

14. SPI

串行外设接口（SPI）是一种串行同步通信系统，它可以通过三根线与外部设备交换 8 位数据。SPI 设备可以配置为主设备或从设备。

主要特点是：

- 全双工三线串行同步传输
- 主、从模式
- 可编程时钟极性和相位
- 8 位可编程主模式比特速率
- 从设备最高频率 = FCPU/8
- 传输结束中断标志
- 写冲突标志
- 主模式故障保护能力和中断标志

14.1. 功能框图

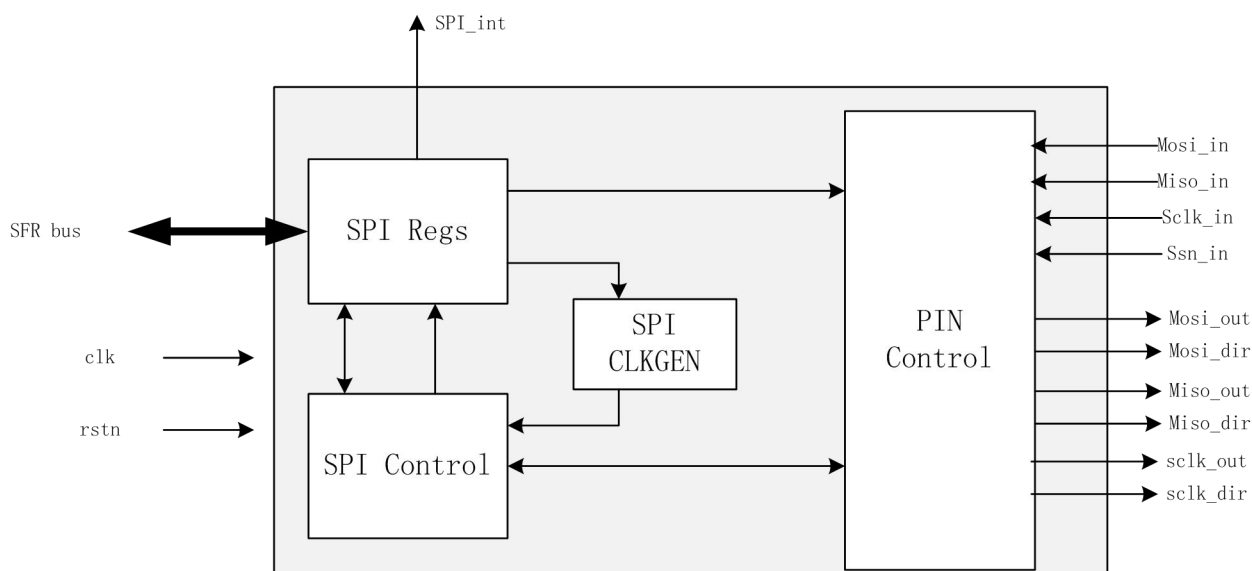


图15-0

14.2. 特性

SPI设备通过4个IO连接外部设备：

MISO: Master In Slave Out

MOSI: Master Out Slave In

SCK: Serial Clock

SSN: Slave Select

显示 SPI 主设备和 SPI 从设备之间的互连：

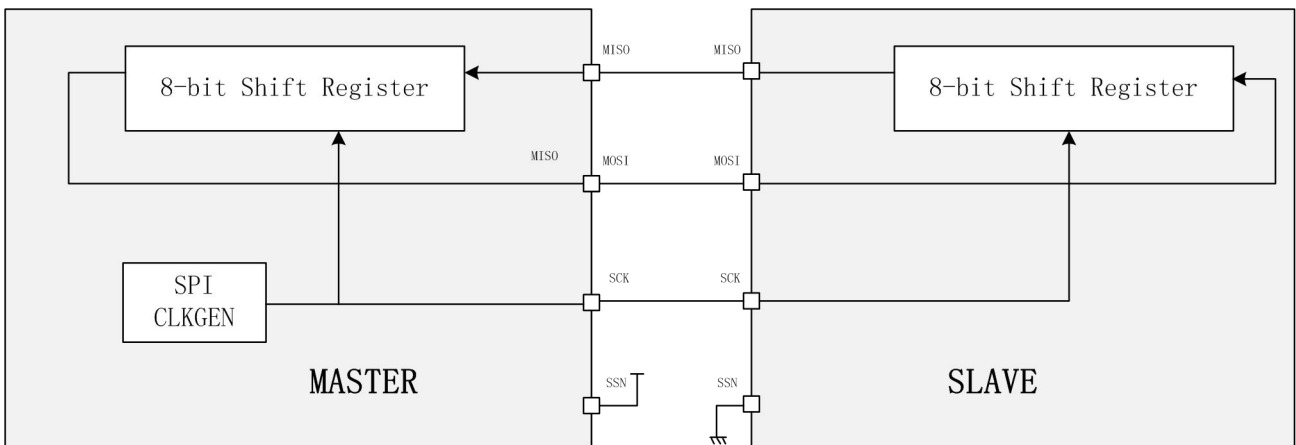


图15-1 SPI Master/SPI Slave

- MOSI连接在一起；
- MISO 连接在一起；
- SCK 连接在一起。

在一个8位字符传输期间，主设备SSN必须连接到逻辑“1”和通信从设备的SSN必须连接到逻辑“0”。

当一个SPI传送时，主设备通过其MOSI向从设备传输数据（高位在前），提供同步时钟通过SCK信号。从设备回应通过MISO发送数据（高位在前）。

在传输结束时，主设备和从设备交换了它们8位字符移位寄存器的值。

14.3. SPI 寄存器

SPI的实现需要三个专用寄存器SFR：

SPICR: 控制寄存器 (SPIE, SPE, SPR2, MSTR, CPOL, CPHA, SPR1 and SPR0 bits)

SPIDR: 数据寄存器

SPISR: 状态寄存器 (SPIF, WCOL and MODF bits)

Register	Address	Description	Reset value
SPICR	0x93	SPI Control Register	04h
SPIDR	0x94	SPI Data Register	00h
SPISR	0x95	SPI Status Register	00h

14.3.1. SPICR 寄存器

SPICR寄存器，它可以在任何时间读取或写入通过SFR总线，用于配置SPI系统。

但是建议遵循这个协议去改变MSTR, CPOL, CPHA, SPR2, SPR1 或SPR0位：

当SPE = '0', MSTR, CPOL, CPHA, SPR2, SPR1 or SPR0位没有限制。

当SPE = '1',建议复位SPE位，未了改变MSTR位；

在主模式（SPE =MSTR= 1”），建议改变 CPOL, CPHA, SPR2, SPR1 or SPR0，当不在传输过程（一个传输过程从 SPDR 写入到 SPIF 点设置）。

从模式（SPE = 1”和 MSTR=0”），这是有可能改变 CPOL or CPHA 位，当 SS_N = “1”，但 SS_N 不由 MCU 控制，建议以 SPE 位复位改变其中的一位。SPR2、SPR1 和 SPR0 位是没有限制的，因为他们不用于从模式。

如果不遵守该协议，则不保证当前和以下传输的功能。要恢复 SPI 系统的良好操作，建议重置 SPE 位或设置 RST 引脚。

BIT	7	6	5	4	3	2	1	0
FIELD	SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
RESET	0	0	0	0	0	1	0	0

Bit NO.	Bit Menemonic	Function
7	SPIE	SPI 中断使能 清除时，SPI中断阻止（interrupt_n = “1”）时 设置时，一个SPI中断产生（interrupt_n = 0），假如SPSR寄存器中的 SPIF = “1” 或MODF “1”。
6	SPE	SPI 系统使能 When clear, SPI system is off. When set, SPI system is on.
5	SPR2	SPI Baud Rate Select bit 2
4	MSTR	Master/Slave 模式选择 When clear, SPI system is configured as a slave. When set, SPI system is configured as a master
3	CPOL	Clock Polarity Select When clear, the serial clock idles low - Active high clock selected. When set, the serial clock idles high - Active low clock selected.
2	CPHA	Clock Phase Select When clear, the first clock transition is the first capture edge When set, the second clock transition is the first capture edge
1:0	SPR[1:0]	Baud Rate Select bit [1:0]

下表显示了可用的可编程波特率（主模式）当 SPI 连接单片机时，Fcpu = 48 MHz。

SPR2	SPR1	SPR0	Fsck	Fsck (Fcpu = 48MHz)
0	0	0	Fcpu /8	6.0 MHz
0	0	1	Fcpu /8	6.0 MHz
0	1	0	Fcpu /16	3.0 MHz
0	1	1	Fcpu /32	1.5 MHz
1	0	0	Fcpu /64	750 KHz
1	0	1	Fcpu /128	375 KHz
1	1	0	Fcpu /256	187.5 KHz
1	1	1	Fcpu /512	93.75 KHz

SPI programmable baud rate for Fcpu = 48 MHz

14.3.2 SPDR 寄存器

SPDR寄存器，这是用来发送和接收数据在串行总线上，是SPI系统的中心单元。它可以在任何时候可读和写的限制通过SFR总线。

它是在发送方向和接收方向的双缓冲单缓冲：写入SPDR的地方直接从移位寄存器数据[7:0]值，当读到SPDR返回最后一个接收到的数据放入缓冲区的读数据[7:0]（而不是移位寄存器数据）。

在传输过程中是不可能写 SPDR，或当 SPIF=1”（除非 SPIF=1” 已经先读）：

- 当 SPI 系统关闭（SPE = '0'）不可以发生传输，所以可以写几次 SPDR，如果 SPIF 之前没有设置（除非 SPIF=1” 已经先读）。
- 在主模式（SPE =MSTR= 1），传输开始时 SPDR 被写，传输结束时 SPDR 被设定。如果没有传输的过程中，和 SPIF 没有被设置（除非 SPIF=1” 已经先读），SPDR 是有可能的被写一次。
- 在从模式（SPE = 1” 和 MSTR=0”），如果没有传输的过程中，和 SPIF 没有被设置（除非 SPIF=1” 已经先读），SPDR 是有可能的被写几次。

BIT	7	6	5	4	3	2	1	0
FIELD	D[7:0]							
RESET	0x00							

14.3.3. SPISR 寄存器

SPISR 是只读寄存器（通过 SFR 总线）。它用于指示 SPI 传输的结束和两种类型的系统错误。

BIT	7	6	5	4	3	2	1	0
FIELD	SPIF	WCOL	-	MODF	-	-	-	-
RESET	0x00							

Bit NO.	Bit Menemonic	Function
7	SPIF	SPI Transfer Complete Flag 在SPI传输结束时由硬件设置。
6	WCOL	Write Collision Error Flag Set by hardware if there is a write access to the SPDR register while a transfer is in progress. Clear by a read to the SPSR register (while WCOL = '1') followed by an access (read or write) to the SPDR register.
5	-	
4	MODF	Mode Fault Error Flag Set by hardware if the SS_N pin goes low while FlipSPI is configured as a master. Clear by a read to the SPSR register (while MODF = '1') followed by an access (read or write) to the SPDR register.

14.3.4 清除 SPSR's 标志

SPISR 寄存器的三状态标志是用同样的方法清除：

1. 访问 SPISR 读取，以确保状态标志被应用软件考虑。
2. 访问（读或写）的 SPIDR 寄存器，意味着此前 SPI 传输（无论成功或错误）已由软件考虑和一个新的传输开始

14.4 SPI master 模式

14.4.1 I/O pin 配置

MOSI：输出 (MOSIDIR = '0')

MISO：输入 (MISODIR = '1')

SCK：输出 (SCKDIR = '0')

SS_N：连接至 '1' (in a single master system)

14.4.2 SPI 控制寄存器配置

SPCR.SPE 和 SPCR.MSTR 位必须设置。

SPCR.CPHA 和 SPCR.CPOL 位定义时钟相位和极性。

SPCR.SPR[2:0] 定义串行时钟波特率。

14.4.3 Master 模式数据传输

发送序列开始时，一个字节写入 SPDR 寄存器：字节并行加载的数据 [7:0] 到 8 位的移位寄存器在写周期（它是由下面的图 SFRLOAD 脉冲表示）。

然后这个数据从串行 MSB 首先对 MOSIOUT 引脚，和串行时钟是在 SCKOUT 引脚提供同步传输。MISOIN 引脚接收所选设备的响应。

SS_N 一直保持为高直至 8 位数据传输结束。

Note: The delay between the write to SPDR and start of the transmission of the data is 1/2 SCKOUT cycle time.

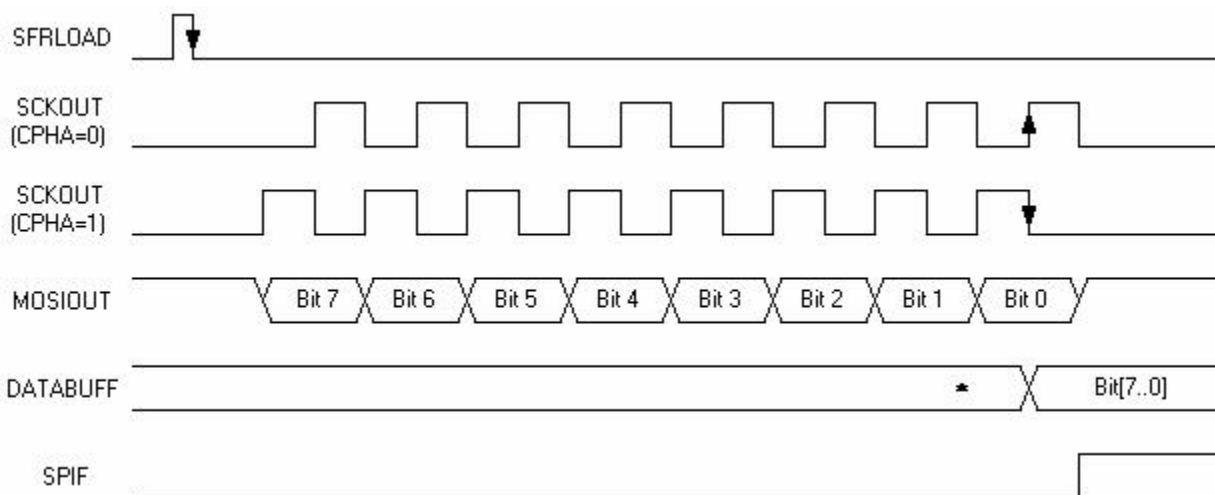


图 15-2 Beginning and end of transfer for a SPI master (CPOL = 0)

当数据传输完成后（图15-2）：从MISOIN引脚收到一个移位寄存器中拷贝的字节，进入读取缓冲区（DATABUFF）在SCK的最后捕捉边缘（SCKOUT），和SPIF位设置在最后SCK周期。中断生成，如果SPIE位设置。收到的字节的值返回DATAOUT[7:0]当SPDR寄存器已读。

清除SPIF位，要求读到SPSR寄存器（而SPIF位设置）后跟一个读或写的SPDR寄存器。
当SPIF为设置，所有写到SPDR寄存器被忽略（除非SPIF = “1” 已经先读）

14.5 SPI slave 模式

当SS_N引脚为高电平，SPI从逻辑和位计数器复位。

14.5.1 I/O pin 配置

MOSI : 输入 (MOSIDIR = ‘1’)

MISO : 输出, 如果SS_N=0 (MISODIR = SS_N)

SCK : 输入(SCKDIR = ‘1’)

14.5.2 SPI 控制寄存器配置

- SPCR.SPE位必须设置、SPCR.MSTR位必须复位。
- SPCR.CPHA 和 SPCR.CPOL位定义时钟相位和极性
(SPI从时钟时钟相位和极性必须与SPI主时钟相同)。
- SPCR.SPR[2:0] 由于SPI主设备提供串行时钟，所以不使用。

14.5.3 Slave 模式下数据传输

一个字节可以写在 SPDR 寄存器但它不是必需的。发送序列开始时，从设备收到来自其 MOSIIN 引脚的数据 MSBIT 和从它的 SCKIN 引脚的串行时钟信号。从设备回应发出连续高位在前的 8 位移位数据寄存器到其 MISOOUT 脚。SS_N 引脚必须保持在低电平在 8 位传输完成期间。

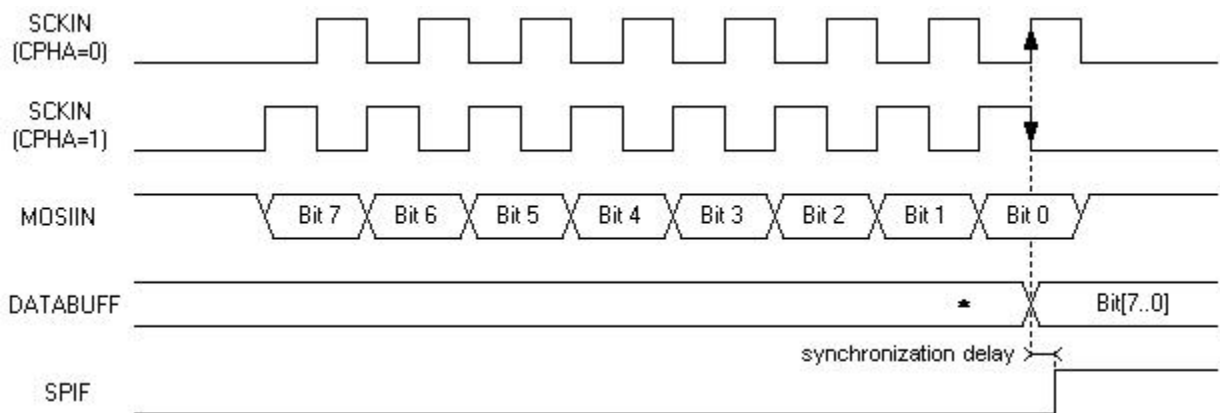


图15-3 Beginning and end of transfer for a SPI slave (CPOL = 0)

当数据传输完成后（图15-3）：从MOSIIN引脚收到一个移位寄存器中拷贝的字节，进入读取缓冲区（DATABUF）在SCK的最后捕捉边缘（SCKIN），和SPIF位设置。中断生成，如果SPIE位设置。收到的字节的值返回DATAOUT[7:0]当SPDR寄存器已读。

当SCK可以异步的SPI主时钟，同步延迟需要在SCK的最后捕捉边缘设置SPIF位。

清除SPIF位，要求读到SPSR寄存器（而SPIF位设置）后跟一个读或写的SPDR寄存器。

当SPIF位被设置，所有写入SPDR寄存器被忽略（除非SPIF = “1” 已经先读）。

SPIF位可以清除在第二传输期间，但必须之前清除第二个最后SCK捕捉边缘避免溢出条件。

当CPHA位复位，SS_N引脚必须设置在每8位传输之间，为了写到SPDR寄存器而不产生写冲突。

14.6 SPI 传输格式

为了适应大多数可用的同步串行外围设备，可以通过串行时钟相位和极性的四种组合选择四种可能的数据/时钟关系。在传输过程中，SPI 主设备和通信从设备的时钟相位和极性必须相同。
请注意，在从模式或 SPE 位未设置时，没有串行时钟输出。

14.6.1 CPHA bit 复位时

串行时钟的第一个边沿是MSBit捕获边缘（图15-4）：

当CPOL设置，捕获边缘串行时钟的下降沿

当CPOL复位，捕获边缘是串行时钟的上升沿

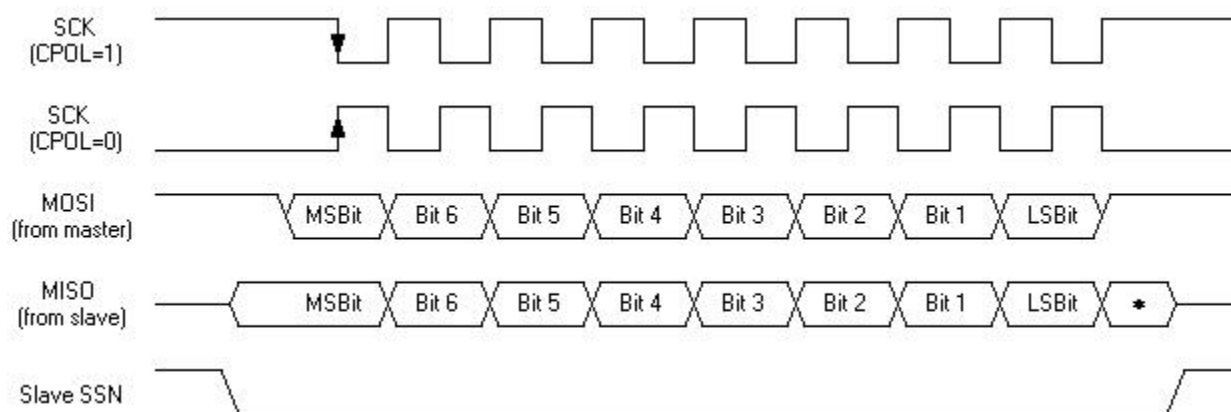


图15-4 Data/clock timing diagram (CPHA = 0)

一个主设备，传输开始于一个字节写在其 SPDR 寄存器；结束于 SPIF 位设置。

一个从属设备且 CPHA=0，传输开始于 SS_N 引脚变为低；结束于 SS_N 回到高，即使 SPIF 位已被设置。

作为从设备没有办法知道传输将何时开始，其移位寄存器的 MSBit 发送通过 MISOOOUT 引脚发送，一旦 SS_N 引脚变低。

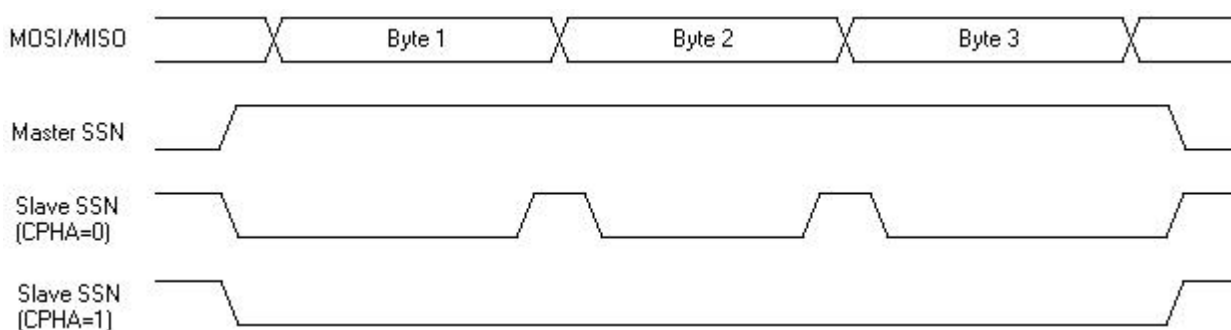


图15-5SSN timing diagram (CPHA = 0)

从设备 SS_N 引脚必须设置在每 8 位传输之间（图 15-5）。事实上，在一个从设备，如果 CPHA=“0”，一个写 SPDR 寄存器时 SS_N 引脚连接到一个低电平产生写冲突。此外，只有第八个 SCKIN 边缘后 SS_N 引脚被拉低，才被考虑在内。

14.6.2 CPHA bit 置位时

串行时钟的第二边缘是MSBit捕获边缘（图15-6）：

当CPOL设置，捕获边缘是串行时钟的上升沿

当CPOL复位，捕获边缘是串行时钟的下降沿

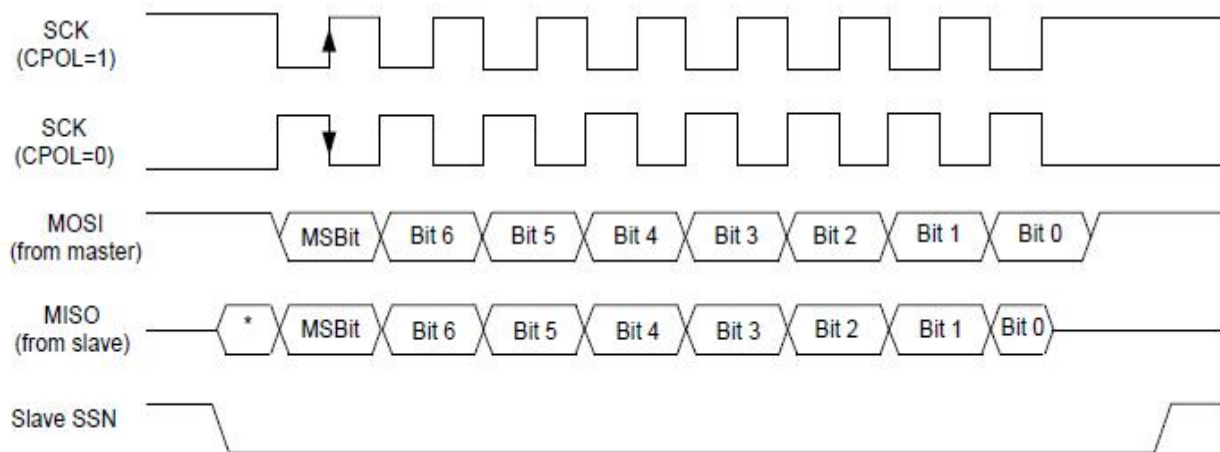


图15-6 Data/clock timing diagram (CPHA = 1)

一个主设备，一个传输开始于一个字节写在其 SPDR 寄存器；结束于 SPIF 位设置。

一个从设备且 CPHA="1"，传输开始于串行时钟第一边缘后；结束于最后 SCKIN 边缘，当 SPIF 位设置。

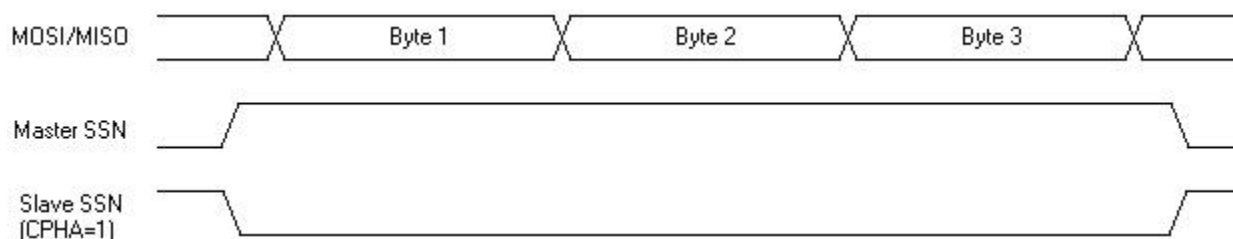


图15-7 SSN timing diagram (CPHA = 1)

Note: SS_N pin can stay low during several 8-bit transfers (Figure 15-7).

14.7 SPI 系统错误

14.7.1 写冲突

一个写冲突发生当尝试写入SPDR寄存器已完成，而传输正在进行中。在这种情况下，该WCOL位设置（中断生成，即使SPIE位设置），传输继续不中断，并导致错误的的数据被忽略。

在主和从模式下都可能发生写冲突。

In master mode:

传输开始于一个字节写在SPDR寄存器，结束于SPIF位设置。尝试写入SPDR寄存器在传输过程中产生写冲突。

In slave mode:

when CPHA is reset:

一个写阻塞发生时，尝试写入SPDR寄存器已在SS_N引脚处于低电平：SS_N引脚必须为高在每一个数据字节传输允许写入SPDR寄存器，不产生写冲突。

when CPHA is set:

传输开始于SCKIN引脚的第一个边缘后，结束于SPIF位设置。尝试写入SPDR寄存器在传输过程中产生写冲突。

清除WCOL位，要求读到SPSR寄存器（WCOL位设置）后跟一个读或写的SPDR寄存器。

Note: the WCOL bit is not cleared if the second step is a write to the SPDR register while a transfer is in progress, and the SPSR register must be read once again in this case.

14.7.2 Master mode 故障错误

主模式错误发生在一个SPI主设备时，其SS_N引脚被拉低：

这意味着另一个设备试图选择SPI主设备作为一个从设备。

这可能导致IO之间的竞争并导致灾难性的死锁（如果有IO的话，和如果这些IO用于标准推挽模式）。

当发生模式错误时：

方向控制信号MOSIDIR，MISODIR和SCKDIR设置去配置MOSI，MISO和SCK作为输入 I/O；

MODF被设置，和中断产生假如SPIE位被设置；

SPE被重置以禁用SPI系统；

MSTR位复位，配置SPI设备在从模式下。

清除MODF位，要求读到SPSR寄存器（同时MODF位设置）后紧跟一个访问（读或写）的SPDR寄存器。

但主模式错误机制并不能完全保护SPI IO损害：

它可能是一个设备试图成为主设备但不立即拉低第一主设备的SSN IO，或它可能是两个从设备通过错误的系统的选择。在这两种情况下，未检测到主模式故障错误。

在这种情况下，建议以推挽方式驱动输出。使用串一个电阻与SPI IO是可能的（价值取决于IO驱动能力）。在这两种情况下，可用波特率都受到影响（取决于SPI线路的负载电容）。

14.7.3 Overrun condition

这种情况未被SPI系统错误检测出来（在SPSR寄存器中没有标志）。

当一个从设备在结束之前传输后清除SPIF位之前接收到第二个数据字节时，Overrun条件突变，该SPIF位必须在第七串行时钟捕获边缘清除（在第七位被采样在移位寄存器中）。

在这种情况下，缓冲区和该装置的移位寄存器包含字节在SPIF位最后清除后接收，和其他所有

的字节数发送后消失。

如果SPIF位第七和第八串行时钟捕获边缘之间清除，SPIF位不在第八串行时钟捕获边缘设置（作为最后一个字节接收丢失）。

移位器包含字节发送后SPIF点最后清除，为了让下面传输的安全方法：

- 主设备向从设备发送一个字节。
- 从设备的响应发送之前接收字节（从设备没有写 SPDR 寄存器）
- 主设备生成一个附加的传输周期来接收它发送的最后一个字节（因为从设备在发送之前必须先收到一个字节）
- 两个连续词必须不同（由于两个相同的词之间的传输可能发生溢出情况，主设备不应该检测第二个词已被从设备丢失）

15. 模数转换 (ADC)

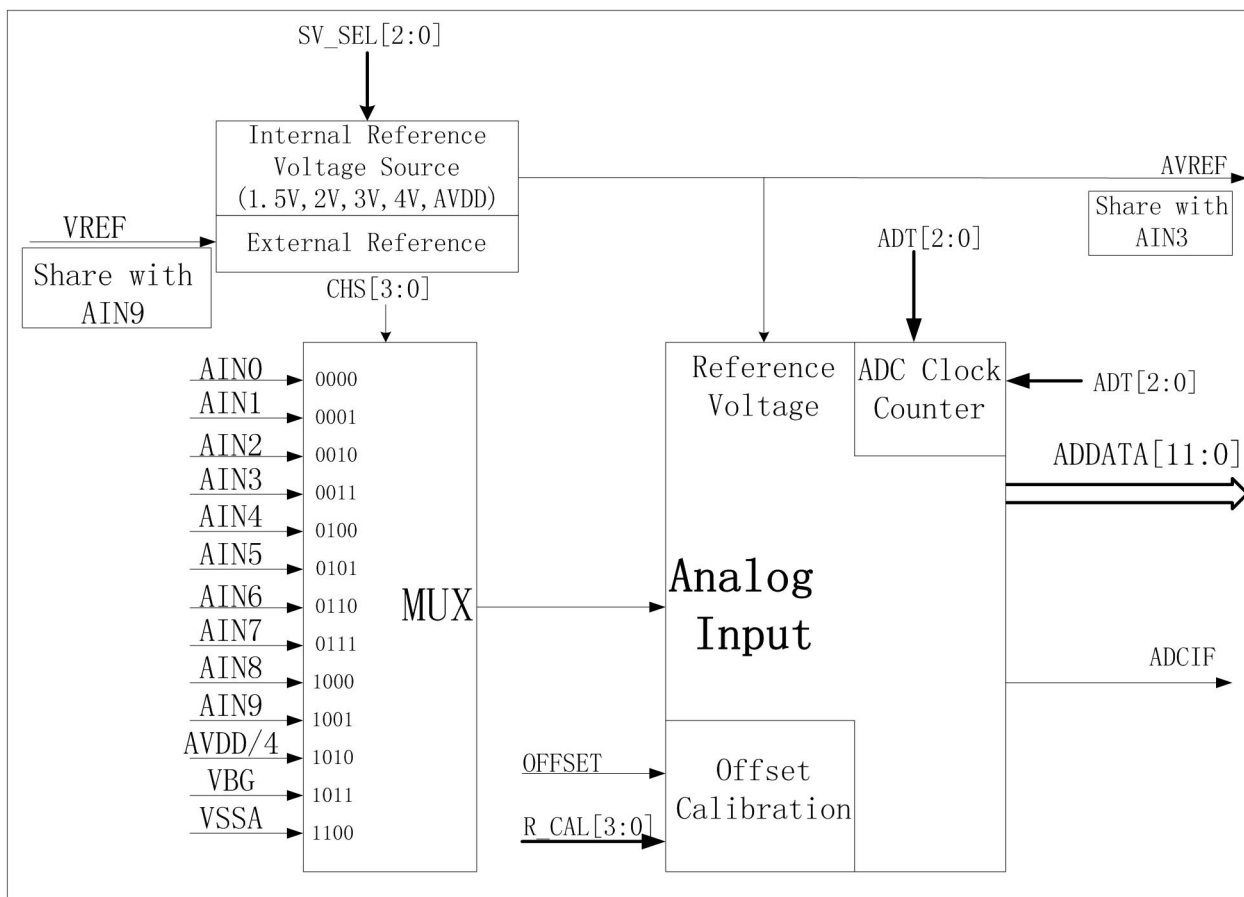
15.1. 特性

- ◆ 12 位分辨率
- ◆ 内建参考电压：1.5V/2.0V/3.0V/4.0V/
- ◆ 外部输入参考电压，以及电源电压
- ◆ 10 通道输入

EN8F5113 系列包含一个单端输入、12 位逐次逼近模/数转换器 (ADC)。10 个 ADC 通道都可以输入独立的模拟信号，但是每次只能使用一个通道。GO 信号控制开始转换，提示转换结束。当转换完成时，更新 ADC 数据寄存器，设置 ADCON 寄存器中的 ADCIF 位，并产生一个中断（如果 ADC 中断被允许）。

ADC 模块整合了数字比较功能。

15.2. ADC 模块图



15.3. ADC 寄存器

ADC 控制寄存器 ADCON1

EFH	Bit7-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	Filter_en[1:0]	GO	OFFSET	VREF_T	AD_LP_EN	VREF_EN	ADCEN
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	1	0	0	0	0

编号	位符号	说明
7:6	Filter_en[1:0]	00/11:直接输出原始数据; 01: 4 倍平均后输出; 10: 8 倍平均后输出
5	GO	ADC 状态标志位 0:关闭 ADC 转换 1:使能 ADC 转换, 如果非比较模式, 转换完成后硬件置位 ADCIF,当软件清 0 ADCIF 后, 硬件 会再次更新 ADC 数据寄存器。 比较模式, 会持续转换, 转换结果与 ADC 数据寄存器比较, 根据 datax.判断 置位 OR 清 0 ADCIF
4	OFFSET	ADC offset 控制: OFFSET=1, +OFFSET; 0, -OFFSET.POR/ETST 复位期间 HW 从 FlashInfor 区读值填写
3	VREF_T	1: 测试 ADC 参考电压。此时参考电压从 AIN[3]输入, AIN[3]不能作为模拟 输入
2	AD_LP_EN	1:enable LP。此模式下 ADCLK 时钟频率要低于 1MHz
1	VREF_EN	SW 控制。1 使能 ADC 参考电压模块、Bias 模块; 0 关闭。
0	ADCEN	1: 打开 ADC 模块; 0: 关闭 ADC 模块

ADC 控制寄存器 ADCON2

E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	ADCIF	datax	MODE	EC	CHS[3]	CHS[2]	CHS[1]	CHS[0]
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	1	1	0	0	0	0	0

编号	位符号	说明																								
7	ADCIF	中断标志。ADC 转换结束或者模拟输入大于 ADDH/L 的时候置 1 并触发 ADC 中断, 软件清 0																								
6	datax	1:比较模式 ADC_DOUT>比较数据置 1; 0: ADC_DOUT<比较数据置 1																								
5	MODE	1:12bit 0:10bit																								
4	EC	1: 打开比较功能; 0: 关闭比较																								
3	CHS[3]	<table border="1"> <thead> <tr> <th>CHS</th> <th>ADC input select</th> <th>CHS</th> <th>ADC input select</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AIN0</td> <td>1000</td> <td>AIN8</td> </tr> <tr> <td>0001</td> <td>AIN1</td> <td>1001</td> <td>AIN9</td> </tr> <tr> <td>0010</td> <td>AIN2</td> <td>1010</td> <td>(AVDDI)/4</td> </tr> <tr> <td>0011</td> <td>AIN3</td> <td>1011</td> <td>VBG</td> </tr> <tr> <td>0100</td> <td>AIN4</td> <td>1100</td> <td>GNDA</td> </tr> </tbody> </table>	CHS	ADC input select	CHS	ADC input select	0000	AIN0	1000	AIN8	0001	AIN1	1001	AIN9	0010	AIN2	1010	(AVDDI)/4	0011	AIN3	1011	VBG	0100	AIN4	1100	GNDA
CHS		ADC input select	CHS	ADC input select																						
0000		AIN0	1000	AIN8																						
0001		AIN1	1001	AIN9																						
0010	AIN2	1010	(AVDDI)/4																							
0011	AIN3	1011	VBG																							
0100	AIN4	1100	GNDA																							
2	CHS[2]																									
1	CHS[1]																									

0	CHS[0]	0101	AIN5	1101	X
		0110	AIN6	1110	X
		0111	AIN7	1111	X

ADC 电压基准选择寄存器 ADCON3

DFH	Bit7	Bit6-4			Bit3-0			
ADCON3	-	SV_SEL[2:0]			R_CAL[3:0]			
读/写	-	读/写			读/写			
复位值	-	1	0	1	0	0	0	0

编号	位符号	说明
7	-	-
6	SV_SEL[2]	ADC 参考电源选择： 000: 1.5V (Int); 001: 2.0V (Int); 010: 3V (Int); 011: 4.0V (Int); 100: AVDDI (Int); 101: ANA_OUT[9]
5	SV_SEL[1]	
4	SV_SEL[0]	
3	R_CAL[3]	
2	R_CAL[2]	
1	R_CAL[1]	
0	R_CAL[0]	

ADC 参考电压控制 ADVREF_SEL

FEH	Bit7	Bit6-0						
ADVREF_SEL	-	VREF_SEL[6:0]						
读/写	-	读/写						
复位值	-	0	0	1	0	0	0	0

编号	位符号	说明
7	-	-
6:0	VREF_SEL[6:0]	Default 是 7'b001_0000. POR/ETST 复位期间 HW 从 FlashInfor 区读值读取 调整电压值: 3.2mV/bit(2V 参考); 4.7mV/bit(3V 参考) [6:5]为粗调位, 部分芯片需要通过设置[6:5]来调整

ADC 采样时间选择、转换时钟控制位 ADT

FFH	Bit7	Bit6-4		Bit3	Bit2-0	
ADT	-	SH_CTR[2:0]		-	ADT[2:0]	
读/写	-	读/写		-	读/写	

复位值	-	0	-	0
-----	---	---	---	---

编号	位符号	说明
7	-	-
6:4	SH_CTR[2:0]	采样保持时间： 000: 4 *ADCLK; 001: 8 *ADCLK; 010: 12* ADCLK 011: 12 *ADCLK; 100: 36* ADCLK; 101: 68 *ADCLK 110: 132 *ADCLK; 111: 260* ADCLK
3	-	-
2-0	ADT[2:0]	ADC 时钟周期选择位 000:ADC 时钟周期 $T_{ad}=2T_{sys}$ 001:ADC 时钟周期 $T_{ad}=4T_{sys}$ 010:ADC 时钟周期 $T_{ad}=8T_{sys}$ 011:ADC 时钟周期 $T_{ad}=16T_{sys}$ 100:ADC 时钟周期 $T_{ad}=32T_{sys}$ 101:ADC 时钟周期 $T_{ad}=64T_{sys}$ 110:ADC 时钟周期 $T_{ad}=128T_{sys}$ 111:ADC 时钟周期 $T_{ad}=256T_{sys}$

ADC 数据寄存器低 8 位 ADDATAL

EEH	Bit7-4	Bit3-0
ADDATAL	-	ADDATA[3:0]
读/写	-	读/写
复位值	-	0

ADC 数据寄存器高 8 位 ADDATAH

F6H	Bit7-0
ADDATAH	ADDATA[11:4]
读/写	读/写
复位值	0

编号	位符号	说明
ADDATAH	7:0	在非比较模式 (EC = 0) 时, HW 将 ADC 输出的 Digital 数据更新到该寄存器, 软件 Read。 在比较模式时 (EC = 1), 软件将阈值值写到该寄存器.
ADDATAL	3:0	

15.4. ADC 工作参考流程

启动 ADC 转换步骤:

- 1)配置并选择模拟输入通道
 - 2)使能 ADC 转换,并清 0 ADCIF
 - 3)GO 置位开始 ADC 转换
 - 4)等待 ADCIF=1, 如果 ADC 中断使能, 则 ADC 中断将会产生, 用户需要软件清 0 ADCIF
 - 5)从 ADDH/ADDL 获得转换数据
- 重复步骤 4-5 开始下一次转换

启动数字比较功能步骤如下:

- 1) 配置并选择模拟输入通道
- 2) 写入 ADDH/ADDL 获得转换数据
- 3) EC 置 1 使能数字比较功能
- 4) 使能 ADC 模块
- 5) GO 置 1 开始数字比较功能
- 6) 如果模拟输入的值比设置的比较值大, ADIF 会被置 1。如果 ADC 中断使能, 则 ADC 中断将会产生, 用户需要软件清 0 ADCIF
- 7)数字比较功能会持续工作, 直到 GO 清 0

16. IAP 控制单元 (Data EEPROM)

EN8F5113 系列在 8KB 程序存储器空间之外，另有两个存储空间，可以进行 IAP (In Application Programming) 操作，即用户可在程序中将数据写入该区域或从该区域读出可以实现 Data EEPROM 功能：

1、Flash 空间由 8 个 1K/Page main 区和 1 个 128Byte Information 区组成，规划为 **ROM+扩展 EEPROM+客户信息区**，如下图所示。

		IAP Erase	IAP WR	IAP RD	MOVC RD	MOVX RD	MOVX WR	
Page0	1K, 0000-03FF ROM 区	Yes	Yes	Yes	Yes	No	No	
Page1	1K, 0400-07FF ROM 区	Yes	Yes	Yes	Yes	No	No	
Page2	1K, 0800-0BFF ROM 区	Yes	Yes	Yes	Yes	No	No	
Page3	1K, 0C00-0FFF ROM 区	Yes	Yes	Yes	Yes	No	No	
Page4	1K, 1000-13FF ROM 区	Yes	Yes	Yes	Yes	No	No	
Page5	1K, 1400-17FF ROM 区	Yes	Yes	Yes	Yes	No	No	
Page6	1K, 1800-1BFF ROM 区	Yes	Yes	Yes	Yes	No	No	
Page7	1K, ROM+扩展 EEPROM 区	1C00-1XXX ROM 区	Yes	Yes	Yes	Yes	No	No
		1XXX-1FFF EEPROM 区	Yes	Yes	Yes	No	Yes	No
EEPROM0	2080-20FF 客户固定信息区	No	Yes	Yes	No	Yes	No	

- 2、其中前 7K 固定为 ROM 区，第 8K 可配置为 ROM+EEPROM 区。
- 3、其中前 7K 固定为 ROM 区，第 8K 可配置为 ROM+EEPROM 区。
- 3、EEPROMCON=00：无扩展 EEPROM 区，8K Bytes Flash 都作为 ROM 用；
EEPROMCON=11：后 1K 为扩展 EEPROM 区，7K Bytes ROM 空间。
- 4、Information 区为 128Byte 客户信息区。

读写操作如下：

1、ROM 区可在 ICP 模式下由烧录器擦/写/读；软件可以通过 IAP 方式擦/写/读，可以通过 MOVC 指令读取数据

2、EEPROM 区可在 ICP 模式下由烧录器擦/写/读；软件可以擦/写/读，在擦前需要备份 ROM 数据；可以通过 MOVX 指令读取数据，地址为 1xxx-1FFF，KEIL 软件需要根据

EEPROMCON 寄存器配置外部 RAM 区间

3、客户信息区可在 ICP 模式下由烧录器擦/写/读；软件可以通过 IAP 写一次 ICP 未写过的空间，可以 IAP 读，可以通过 MOVX 指令读取数据，地址为 2080-20FF，KEIL 软件需要配置外部 RAM 区间为 2080-20FF

扩展 Data EEPROM 控制寄存器 EEPROMCON(DCH):

DCH	Bit7-2	Bit1-0
EEPROMCON	-	EEPROMCON[1:0]
R/W	-	R/W
Default	-	00b

位编号	位符号	说明			
7:4	-	-			
1:0	EEPROMCON	EEPROMCON	扩展 EEPROM 空间	ROM 空间	扩展 EEPROM 地址
		00	0	8K Bytes	-
		01 & 10	-	-	-
		11	1K Bytes	7K Bytes	1C00H~1FFFH

16. 1. IAP 相关控制寄存器

16. 1. 1. IAPCTRL(C8H): IAP 控制寄存器

IAP 控制寄存器:

地址	名称	7	6-5	4	3	2	1	0	上电复位值
C8H	IAPCTRL	IAPLOCKEN	-	IAPFail	IAPRD	IAPER	IAPWR	IAPGO	1000,0000b
读/写		W/R	-	W/R	W/R	W/R	W/R	W/R	
位编号	位符号	说明							
7	IAPLOCKEN	1:IAP 锁死，任何 IAP 操作无效；0: IAP 解锁，允许通过流程开始 IAP 动作，每次 IAP 动作结束硬件自动置 1 锁死 IAP 功能. 完成解锁流程后读该位返回 0，否则一直返回 1							
6-5									
4									
3	IAPFail	以下 5 种情况，置 IAPGO 无效同时 IAPFail 位置 1，表示错误操作，该位需要软件清 0. 1、未完成解锁流程开始 IAP，硬件置 IAPFail 为 1，IAPGO 置 1 无效； 2、如果 IAP Erase 与程序运行在同一 Page 的时候，硬件置 IAPFail 为 1，							

		IAPGO 置 1 无效; 3、擦 Infor 区 IAPGO 置 1 无效; 4、读写 Infor 区低 128Byte, IAPGO 置 1 无效。
2	IAPRD	1: 当前 IAP 为 Read 操作
1	IAPER	1: 当前 IAP 为 Erase 操作
0	IAPWR	1: 当前 IAP 为 Program 操作

16.1.2. IAPLOCK(F5H): IAP 解锁寄存器

IAP 解锁寄存器:

地址	名称	7	6	5	4	3	2	1	0	上电复位值
F5H	IAPLOCK	IAPLock[7:0]								00H
读/写		W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	
位编号	位符号	说明								
7~0	IAPLock	软件置 IAPLOCKEN=0 后, 依次写 5A、A5 到 IAPLock 寄存器, 完成 IAP 解锁流程, 读这个寄存器一直返回 FF								

16.1.3. IAPDATA(EDH): IAP 数据寄存器

IAP 数据寄存器: 该寄存器存放待写入 Data EEPROM 的 8 位数据

地址	名称	7	6	5	4	3	2	1	0	上电复位值
EDH	IAPDATA	IAPDATA[7:0]								00H
读/写		W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	
位编号	位符号	说明								
7~0	IAPDATA	存放待写入 Data EEPROM 的 8 位数据								

16.1.4. IAPADDRH (DDH): IAP 高位地址寄存器

IAP 高位地址寄存器:

地址	名称	7	6	5	4	3	2	1	0	上电复位值
DDH	IAPADDRH	-	-	IAPAddr[13:8]						00H
读/写		-	-	W/R	W/R	W/R	W/R	W/R	W/R	

位编号	位符号	说明
7-6	-	-
5	IAPAddr[13]	IAP 写的时候, 该 bit 置 1 表示写 Information 区
4-0	IAPAddr[12:8]	IAP 地址。在 Erase 操作时, IAPAddress[12:10]三位表示 Page 号

16.1.5. IAPADDRL (E5H): IAP 低位地址寄存器

IAP 低位地址寄存器：该寄存器存放待写 Data EEPROM 地址的低 8 位

地址	名称	7	6	5	4	3	2	1	0	上电复位值
E5H	IAPADDRL	IAPADDRL[7:0]								00H
读/写		W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	

位编号	位符号	说明
7~0	IAPADDRL	IAP 地址。

16.2. IAP 操作流程

IAP 流程：

- 1、写 IAPLOCKEN=0;
- 2、写 IAPLOCK=5A, 再写 A5, er_reg/wr_reg 必须写 0;
- 3、将需要 Program 的数据写到 IAPDATA, 如果是 Erase 操作则没有本步骤;
- 4、将需要操作的地址写到 IAPAddrH/IAPAddrL, 如果是 Erase 则只需要写 IAPAddrH;
- 5、写 IAPGO=1, IAPER=1 或者 IAPWR=1 或者 IAPRD=1;
- 6、判断 IAPGO=0 本次 IAP 结束, 通过判断 IAPFail 可以知道本次操作是否成功, 如果是 IAPRD, IAPFail=0, 则可以去 IAPDATA 读取数据;

17. 省电模式

17.1. 特性

- **Active Mode:** 典型电流值 $<250 \mu\text{A}/\text{MIPS}$
- **Sleep Mode1:** 典型电流值 $<5 \mu\text{A}$ 、MAX $<10\mu\text{A}$;
 - 停止系统时钟 (IHRC、HOSC) ;
 - Timer/RTC 使用 LOSC 或 ILRC 时钟
 - 可通过 Timer 计数溢出中断、RTC 中断、外部 GPIO 中断 (P05/P06 口除外) 退出 Sleep 模式。退出 Sleep 模式需要等待系统稳定后继续进入 Sleep 模式的地址开始运行程序。
- **Sleep Mode2:** 典型电流值 $<2 \mu\text{A}$ 、MAX $<10\mu\text{A}$;
 - 停止系统时钟 (IHRC、HOSC、LOSC) ;
 - 通过外部 GPIO 中断退出 Sleep 模式。退出 Sleep 模式需要等待系统稳定后继续进入 Sleep 模式的地址开始运行程序。
- **Deep Sleep Mode:** 典型电流值 $<20\text{nA}$ 、MAX $<500\text{nA}$;
 - 停止系统时钟 (IHRC、HOSC、LOSC) ;
 - 禁止所有功能仅支持外部特定 IO 口唤醒。

17.2. 寄存器

省电模式控制寄存器 PCON(B0H)

B0H	7	6	5	4	3	2	1	0
PCON	Lowclk_en_sleep	LDO_LP_EN_IDLE-	PFLASH_ENB	PRAM_ENB	-	DeepSleep	Sleep	Idle
读/写	读/写	-读/写	读/写	读/写	-	读/写	读 / 写	读 / 写
复位值	1	-1	0	0	-	0	0	0

编号	位符号	说明
7	Lowclk_en_sleep	Sleep 模式下是否关闭低速时钟, 0 关闭(硬件自动关闭 LOSC)
6	LDO_LP_EN_IDLE	1: IDLE 模式下 LDO 进入低功耗模式; 0: IDLE 模式下 LDO 正常工作。
5		
4		
3	-	-
2	DeepSleep	置该位前先要置 Wakeup_en。软件读该位返回 0
1	Sleep	置该位的同时要控制 PRAM_ENB, PFLASH_ENB。软件读该位返回 0
0	Idle	置 1 进入 IDLE 状态。软件读该位返回 0

17.3. Idle Mode

- ◆ 将 Idle(PCON.0)位置 1, CPU 将进入 idle Mode;
 - Idle(PCON.0)位置 1, 是 CPU 进入 idle Mode 前的最后一条指令;
 - 可以选择不同的唤醒方式, 在进入 Idle 前要进行设置;
 - Idle Mode 能够降低系统的功耗, 在此模式下, 程序中止运行, CPU 时钟停止, 但外设时钟没有停止;

以下两种方式可退出 Idle Mode:

- 1) 产生一个中断 (INT0, Timer0/1, EUART0/1, RTC, LVDCP), 可退出 IDLE Mode, 当中断发生后, 立即回复 CPU 时钟, 硬件清除 PCON 的 idle 位, 然后执行中断服务程序, 随后跳转到进入 Idle 模式指令之后的指令。
- 2) 复位信号 (复位引脚上的低电平, WDT 复位, LVD 复位, LVR (BOR 模块中) 复位), 复位信号产生后, 系统复位全部寄存器, 然后程序从地址为 000H 开始执行;

17.4. Sleep Mode

- ◆ 将 Sleep(PCON.1)位置 1, CPU 将进入 SleepMode;
 - Sleep(PCON.1)位置 1, 是 CPU 进入 StopMode 前的最后一条指令;
 - 可以选择不同的唤醒方式, 在进入 SleepMode 前要进行设置;
 - SleepMode 能够进入较低功耗模式, 在此模式下, 程序中止运行, CPU 时钟停止, 系统高频时钟将全部被关闭, 内部 LDO 将进入低功耗模式;

以下两种方式可退出 SleepMode:

- 1) 中断 (INT0, Timer0,1 (当时钟源选择为低频), RTC, LVDCP), 当中断发生后, 系统先将内部 LDO 切换到 Normal 模式, 然后恢复系统时钟 (需等待预热时间), 然后执行中断服务程序, 随后跳转到进入 Stop 模式指令之后的指令。
- 2) 复位信号 (复位引脚上的低电平, WDT 复位, LVD 复位, LVR (BOR 模块中) 复位), 复位信号产生后, 系统复位全部寄存器, 恢复时钟, 然后程序从地址为 000H 开始执行;

17.4. DeepSleep Mode

- ◆ 将 DeepSleep(PCON.2)位置 1, CPU 将进入 DeepSleep Mode;
- ◆ DeepSleep(PCON.2)位置 1, 是 CPU 进入 DeepSleepMode 前的最后一条指令;
- ◆ 由于在 DeepSleep Mode 下只支持特定 IO 唤醒, 因此在进入 DeepSleepMode 前要进行如下设置:

特定 IO 唤醒控制寄存器 DPWAKE (F7H)

F7H	7	6	5	4	3	2	1	0
DPWAKE	Wakeup_cs[3:0]				-	-	-	Wake_en
读/写	读/写	读/写	读/写	读/写	-	-	-	读/写
复位值	0	0	0	-	-	-	-	0

编号	位符号	说明
7	Wakeup_cs[3]	外部中断唤醒 IO 选择:
6	Wakeup_cs[2]	0000:P1.0; 0001:P1.1; 0010:P1.2;

5	Wakeup_cs[1]	0011:P1.3; 0100:P1.4; 0101:P1.5;
4	Wakeup_cs[0]	0110:P1.6; 0111:P1.7; 1000:P2.0; other:P2.1;
3	-	-
2	-	-
1	-	-
0	Wakeup_en	1:外部中断唤醒使能; 0:外部中断唤醒关闭。

***:

- 在进入 DeepSleepMode 时，所有 GPIO 默认为上拉输入口；
- 支持 DeepSleepMode 唤醒 GPIO 为 P1.0~P1.7,P2.0,P2.1；
- 此 10 个 GPIO（P1.0~P1.7,P2.0,P2.1）和 ADC 模拟输入复用，如果外部有接模拟信号或低电平输入，务必将此口设置为模拟 IO 口功能（除去设定的唤醒 GPIO）；

18. 中断

18.1. 特性

- ◆ 13 个中断源: GPIO, Timer0/1, UART0/1, PWM, ADC, SPI,I2CTXRX, I2CFIFO, CP, LVD, RTC
- ◆ 4 级中断优先级可配

18.2. 寄存器

初级中断允许控制 IEN0(A8H):

A8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EA	EADC	EPWM	ESO	ET1	ES1	ET0	EX0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

编号	位符号	说明
7	EA	总中断开关, 1 打开; 0 关闭
6	EADC	ADC 中断开关, 1 打开; 0 关闭
5	EPWM	PWM 中断开关, 1 打开; 0 关闭
4	ESO	UART0 中断开关, 1 打开; 0 关闭
3	ET1	Timer1 中断开关, 1 打开; 0 关闭
2	ES1	UART1 中断开关, 1 打开; 0 关闭
1	ET0	Timer0 中断开关, 1 打开; 0 关闭
0	EX0	PORT 中断开关, 1 打开; 0 关闭。 如果要使能外部中断, 需要同时使能 EX0, 以及端口中断使能 PxIEN, 配置好沿中断还是电平中断 P0I2IRQConfig, 在外部中断服务程序中通过软件查询, 外部中断到底由哪个端口产生

中断允许控制 IEN1(B8H):

B8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	-	-	ELVD	ELVDCP	ERTC	EI2CFIFO	EI2CTXRX	ESPI
读/写	-	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值	-	-	0	0	0	0	0	0

编号	位符号	说明
7	-	-
6	-	-
5	ELVD	LVD 中断开关, 1 打开; 0 关闭
4	ELVDCP	LVDPCP 中断开关, 1 打开; 0 关闭
3	ERTC	RTC 中断开关, 1 打开; 0 关闭

2	EI2CFIFO	I2C FIFO 中断开关, 1 打开; 0 关闭
1	EI2CTXR	I2C TXRX 中断开关, 1 打开; 0 关闭
0	ESPI	SPI 中断开关, 1 打开; 0 关闭

中断优先级控制寄存器 IP0(A9H), IP0H(A1H), IP1(B9H), IP1H(B1H):

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP0(A9H)	-	IP0.6	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0
IP0H(A1H)	-	-	IP0.12	IP0.11	IP0.10	IP0.9	IP0.8	IP0.7
IP1(B9H)	-	IP1.6	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0
IP1H(B1H)	-	-	IP1.12	IP1.11	IP1.10	IP1.9	IP1.8	IP1.7
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	-	0	0	0	0	0	0	0

编号	IP0(A9H)		IP0H(A1H)		IP1(B9H)		IP1H(B1H)	
	位符号	说明	位符号	说明	位符号	说明	位符号	说明
7	-	-	-	-	-	-	-	-
6	IP0.6	ADC.IP0	-	-	IP1.6	ADC.IP1	-	-
5	IP0.5	PWM.IP0	IP0.12	LVD.IP0	IP1.5	PWM.IP1	IP1.12	LVD.IP1
4	IP0.4	UART0.IP0	IP0.11	LVDCP.IP0	IP1.4	UART0.IP1	IP1.11	LVDCP.IP1
3	IP0.3	Timer1.IP0	IP0.10	RTC1.IP0	IP1.3	Timer1.IP1	IP1.10	RTC1.IP1
2	IP0.2	UART1.IP0	IP0.9	I2CFIFO.IP0	IP1.2	UART1.IP1	IP1.9	I2CFIFO.IP1
1	IP0.1	Timer0.IP0	IP0.8	I2CTXR.IP0	IP1.1	Timer0.IP1	IP1.8	I2CTXR.IP1
0	IP0.0	PORT.IP0	IP0.7	SPI.IP0	IP1.0	PORT.IP1	IP1.7	SPI.IP1

IP1.x 与 IP0.x 的值共同设置对应中断的优先级,4 级优先级可配:

IP1.x	IP0.x	Priority Level
0	0	Level0(lowest)
0	1	Level1
1	0	Level2
1	1	Level3(highest)

18.3. 中断汇总

中断源	入口地址	允许位	标志位	轮询优先级	中断号(C语言)
Reset	0000H	-	-	0(最高级)	-
PORT	0003H	EX0+PxIENy	PxIRQy	1	0
Timer0	000BH	ET0	TF0	2	1
EUART1	0013H	ES1	RI1+TI1	3	2
Timer1	001BH	ET1	TF1	4	3

EUART0	0023H	ES0	RI0+TI0	5	4
PWM	002BH	EPWM+PWMxI E	PWMxIF+FBIF	6	5
ADC	0033H	EADC	ADCIF	7	6
SPI	003BH			8	8
I2CTXRX	0043H			9	9
I2CFIFO	004BH			10	10
RTC	0053H	ERTC	RTCCConfig.RTCIRQ	11	11
LVDCP	005BH	ELVDCP	LVD_CP.LVDCP_IF	12	12
LVD	0063H	ELVD	LVDLVR_Config.LVDF	13	13

19. 96 位序列号 (UID)

EN8F5113 系列出厂前，每颗都会预烧一个 96 位的代码，用以确保该芯片的唯一性，这个唯一代码被称为序列号 UID (Unique Code)。用户获得此序列号方式参考如下：

```

... ..
unicode[0] = UID;
unicode[3] = UID;
unicode[6] = UID;
unicode[9] = UID;
_nop_();
unicode[1] = UID;
unicode[4] = UID;
unicode[7] = UID;
unicode[10] = UID;
_nop_();
unicode[2] = UID;
unicode[5] = UID;
unicode[8] = UID;
unicode[11] = UID;
... ..

```

UID 使用说明注意：

建议在初始化时读取，取得 UID 数据后保存到定义的数组变量中，在程序运行中如果需要 UID 数据，直接从数组变量中读取，不要再进行读 UID 寄存器操作。

20. 指令集

EN8F5113 系列机器周期为 1 个时钟周期，且多数指令只需一个机器周期执行完成。

20.1. 指令操作数说明

Rn	Working register R0-R7
direct	256 internal RAM locations, any Special Function Registers
@Ri	Indirect internal or external RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction
#data 16	16-bit constant included as bytes 2 and 3 of instruction
bit	256 software flags, any bit-addressable I/O pin, control or status bit
A	Accumulator
addr16	Destination address for LCALL and LJMP may be anywhere within the 64-Kbyte of program memory address space
addr11	Destination address for ACALL and AJMP will be within the same 2-Kbytes page of program memory as the first byte of the following instruction
rel	SJMP and all conditional jumps include an 8-bit offset byte. Range is +127/-128 bytes relative to the first byte of the following instruction

20.2. 算术操作指令

Mnemonic	Description	Code	Bytes	Cycles
ADD A, Rn	Add register to accumulator	28-2F	1	1
ADD A, direct	Add direct byte to accumulator	25	2	2
ADD A, @Ri	Add indirect RAM to accumulator	26-27	1	2
ADD A, #data	Add immediate data to accumulator	24	2	2
ADDC A, Rn	Add register to accumulator with carry flag	38-3F	1	1
ADDC A, direct	Add direct byte to A with carry flag	35	2	2
ADDC A, @Ri	Add indirect RAM to A with carry flag	36-37	1	2
ADDC A, #data	Add immediate data to A with carry flag	34	2	2
SUBB A, Rn	Subtract register from A with borrow	98-9F	1	1
SUBB A, direct	Subtract direct byte from A with borrow	95	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	96-97	1	2
SUBB A, #data	Subtract immediate data from A with borrow	94	2	2
INC A	Increment accumulator	04	1	1
INC Rn	Increment register	08-0F	1	2
INC direct	Increment direct byte	05	2	3
INC @Ri	Increment indirect RAM	06-07	1	3
INC DPTR	Increment data pointer	A3	1	1

DEC A	Decrement accumulator	14	1	1
DEC Rn	Decrement register	18-1F	1	2
DEC direct	Decrement direct byte	15	2	3
DEC @Ri	Decrement indirect RAM	16-17	1	3
MUL AB	Multiply A and B	A4	1	5
DIV	Divide A by B	84	1	5
DA A	Decimal adjust accumulator	D4	1	1

20.3. 逻辑操作指令

Mnemonic	Description	Code	Bytes	Cycles
ANL A, Rn	AND register to accumulator	58-5F	1	1
ANL A, direct	AND direct byte to accumulator	55	2	2
ANL A, @Ri	AND indirect RAM to accumulator	56-57	1	2
ANL A, #data	AND immediate data to accumulator	54	2	2
ANL direct, A	AND accumulator to direct byte	52	2	3
ANL direct, #data	AND immediate data to direct byte	53	3	4
ORL A, Rn	OR register to accumulator	48-4F	1	1
ORL A, direct	OR direct byte to accumulator	45	2	2
ORL A, @Ri	OR indirect RAM to accumulator	46-47	1	2
ORL A, #data	OR immediate data to accumulator	44	2	2
ORL direct, A	OR accumulator to direct byte	42	2	3
ORL direct, #data	OR immediate data to direct byte	43	3	4
XRL A, Rn	Exclusive OR register to accumulator	68-6F	1	1
XRL A, direct	Exclusive OR direct byte to accumulator	65	2	2
XRL A, @Ri	Exclusive OR indirect RAM to accumulator	66-67	1	2
XRL A, #data	Exclusive OR immediate data to accumulator	64	2	2
XRL direct, A	Exclusive OR accumulator to direct byte	62	2	3
XRL direct, #data	Exclusive OR immediate data to direct byte	63	3	4
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate accumulator left	23	1	1
RLC A	Rotate accumulator left through carry	33	1	1
RR A	Rotate accumulator right	03	1	1

RRC A	Rotate accumulator right through carry	13	1	1
SWAP A	Swap nibbles within the accumulator	C4	1	1

20.4. 数据传送指令

Mnemonic	Description	Code	Bytes	Cycles
MOV A, Rn	Move register to accumulator	E8-E F	1	1
MOV A, direct	Move direct byte to accumulator	E5	2	2
MOV A, @Ri	Move indirect RAM to accumulator	E6-E 7	1	2
MOV A, #data	Move immediate data to accumulator	74	2	2
MOV Rn, A	Move accumulator to register	F8-F F	1	2
MOV Rn, direct	Move direct byte to register	A8-A F	2	4
MOV Rn, #data	Move immediate data to register	78-7 F	2	2
MOV direct, A	Move accumulator to direct byte	F5	2	3
MOV direct, Rn	Move register to direct byte	88-8 F	2	3
MOV direct1, direct2	Move direct byte to direct byte	85	3	4
MOV direct, @Ri	Move indirect RAM to direct byte	86-8 7	2	4
MOV direct, #data	Move immediate data to direct byte	75	3	3
MOV @Ri, A	Move accumulator to indirect RAM	F6-F 7	1	3
MOV @Ri, direct	Move direct byte to indirect RAM	A6-A 7	2	5
MOV @Ri, #data	Move immediate data to indirect RAM	76-7 7	2	3
MOV DPTR, #data16	Load data pointer with a 16-bit constant	90	3	3
MOVC A, @A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOVC A, @A+PC	Move code byte relative to PC to accumulator	83	1	3
MOVX A, @Ri	Move external RAM(8-bit addr.) to A	E2-E 3	1	3-10

MOVX A, @DPTR	Move external RAM(16-bit addr.) to A	E0	1	3-10
MOVX @Ri, A	Move A to extern RAM(8-bit addr.)	F2-F3	1	4-11
MOVX @DPTR, A	Move A to extern RAM(16-bit addr.)	F0	1	4-11
PUSH direct	Push direct byte onto stack	C0	2	4
POP direct	Pop direct byte from stack	D0	2	3
XCH A, Rn	Exchange register with accumulator	C8-CF	1	2
XCH A, direct	Exchange direct byte with accumulator	C5	2	3
XCH A, @Ri	Exchange indirect RAM with accumulator	C6-C7	1	3
XCHD A, @Ri	Exchange low-order nibble indir. RAM with A	D6-D7	1	3

20.5. 控制程序转移指令

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	xxx11	2	6
LCALL addr16	Long subroutine call	12	3	6
RET	from subroutine	22	1	4
RETI	from interrupt	32	1	4
AJMP addr11	Absolute jump	xxx01	2	3
LJMP addr16	Long jump	02	3	4
SJMP rel	Short jump (relative addr.)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
JZ rel	Jump if accumulator is zero	60	2	3
JNZ rel	Jump if accumulator is not zero	70	2	3
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit, rel	Jump if direct bit is set	20	3	4
JNB bit, rel	Jump if direct bit is not set	30	3	4
JBC bit, directrel	Jump if direct bit is set and clear bit	10	3	4
CJNE A, directrel	Compare direct byte to A and jump if not equal	B5	3	4
CJNE A, #datarel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn, #datarel	Compare immed. to reg. and jump if	B8-BF	3	4

	not equal			
CJNE @Ri, #data:rel	Compare immed. to ind. and jump if not equal	B6-B7	3	4
DJNE Rn, rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct, rel	Decrement direct byte and jump if not zero	D5	3	4
NOP	No operation	00	1	1

20.6. 位操作指令

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	C3	1	1
CLR bit	Clear direct bit	C2	2	3
SETB C	Set carry flag	D3	1	1
SETB bit	Set direct bit	D2	2	3
CPL C	Complement carry flag	B3	1	1
CPL bit	Complement direct bit	B2	2	3
ANL C, bit	AND direct bit to carry flag	82	2	2
ANL C, /bit	AND complement of direct bit to carry	B0	2	2
ORL C, bit	OR direct bit to carry flag	72	2	2
ORL C, /bit	OR complement of direct bit to carry	A0	2	2
MOV C, bit	Move direct bit to carry flag	A2	2	2
MOV bit, C	Move carry flag to direct bit	92	2	3

21. 电气特性

21.1. 极限参数

直流供电电压.....-0.3V to +5.5V
 输入/输出电压..... GND-0.3V to VDD+0.3V
 工作环境温度.....-40°C to +85°C
 存储温度.....-55°C to +125°C

注意: 如果器件的工作条件超过左列“极限参数”的范围, 将造成器件永久性破坏。只有当器件工作在说明书所规定的范围内时功能才能得到保障。器件在极限参数列举的条件下工作将会影响到器件工作的可靠性。

21.2. 直流电气特性

(VDD = 1.8 - 5.5V, GND = 0V, TA = +25°C, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
工作电压		1.8	3.3	5.5	V	
工作电流					mA	
待机电流(Idle 模式)					mA	
待机电流(Sleep 模式)			5	10	μA	无负载,IO 无 floating, LOSC 和 ILRC 任一使能即可
			2	10	μA	无负载,IO 无 floating, LOSC 关闭、ILRC 关闭
待机电流(Deep sleep 模式)			20	500	nA	无负载,IO 无 floating
输入低电压		GND		0.3 *V _{DD}	V	
输入高电压		0.7 *V _{DD}		V _{DD}	V	
输入漏电流	-1			1	μA	
上拉电阻			50		kΩ	V _{DD} = 3.3V
输出高电压	V _{H1}	V _{DD} - 0.5			V	I _{OH} = -16mA, V _{DD} = 3.3V
输出高电压	V _{H2}	V _{DD} - 0.5			V	I _{OH} = -4mA, V _{DD} = 3.3V
输出低电压	V _{L1}			GND+0.5V	V	I _{OL} = 16mA, V _{DD} = 3.3V
输出低电压	V _{L2}			GND+0.5V	V	I _{OL} = 4mA, V _{DD} = 3.3V

Note: 流过 VDD 的最大电流小于 150mA, 流过 GND 的最大电流小于 150mA,

21.3. ADC 电气特性

(VDD = 1.8 - 5.5V, GND = 0V, TA = +25° C, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
精度			12		bit	
A/D 输入内阻		2			MΩ	
微分非线性误差	DNL			±2	LSB	
积分非线性误差	INL			±4	LSB	
总绝对误差				±6	LSB	
转换时间		2			μs	
工作电流				2	mA	

21.4. 内建高频 IHRC 电气特性

(VDD = 1.8 - 5.5V, GND = 0V, TA = +25° C, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
频率	F		16		Mhz	
振荡输出稳定时间	Tstb		2 ⁷		clk	
频率稳定性	ΔF/F		±1		%	
工作电流	Id			0.6	mA	

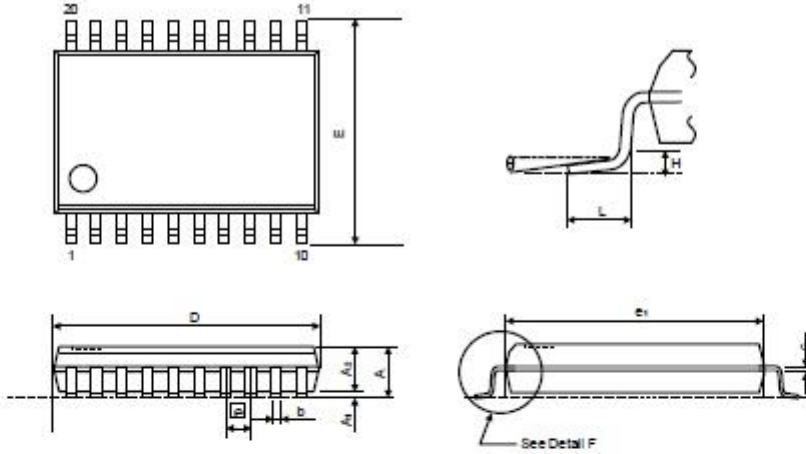
21.5. 内建 LDO 电气特性

(VDD = 1.8 - 5.5V, GND = 0V, TA = +25° C, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
输出电压	V		1.8		V	
驱动能力	Id		30		mA	
无负载电流	Iq			1	μA	低功耗模式

22. 封装信息

TSSOP20L 外形尺寸 单位：英寸/毫米



符号	mm(毫米)		
	最小	正常	最大
A	-	-	1.200
A1	0.050	-	0.150
A2	0.800	-	1.000
b	0.190	-	0.300
c	0.090	-	0.200
D	6.400	-	6.600
E	6.250	-	6.550
e1	4.300	-	4.500
$\frac{e}{n}$	0.65(BSC)		
L	0.500	-	0.700
e	1°	-	7°
H	0.25(TYP)		

规格更新记录

版本	记录	日期
V1.0	初始版本	2018.09.08
V1.6	增加 ESD/Latch Up 性能参数	2019.10.23
V1.8	增加 SOP16/QFN16/DFN8 封装	2020.03.28
V2.0	修改 IAPCTRL 寄存器描述	2020.04.18
V2.1	修改 QFN20A 封装信息	2020.05.13
V2.2	修改 PWMINTCON(CDH)寄存器	2020.07.19
V2.3	增加端口中断标志位清除使能 (PORTIRQCLREN) 寄存器	2020.08.04
V2.4	增加 ILRC 的描述和使用; 增加 96 bit 序列号描述	2021.3.11
V2.5	修改 IO 上拉使能控制寄存器 PxPUN 复位值	2021.03.29
V2.6	修改 TIMER0/1 框图	2021.04.01
V2.7	修改 IAP 操作流程中寄存器名称	2021.05.11
V2.8	修改地址 F8H 寄存器描述, Page 32 &Page 44	2021.06.10
V2.9	修改部分已知错误	2021.06.17

智能电子产品整体解决方案商，单片机集成芯片定！

缔造价值！让“芯”方案更智能！

全球销售及服务网点联系信息：

深圳市英锐恩科技有限公司

ENROO-TECH (SHENZHEN) CO.LTD

中国·深圳市龙岗区坂田街道环城南路坂田国际中心 C2 栋 8 楼 815

Enroo-Tech Technologies CO., Limited

香港新界葵涌工业街 24-28 号威信物流中心 13 楼 A 室

联系电话：86-755-82543411,83167411,83283911,88845951

联系传真：86-755-82543511

全国热线：4007-888-234

联系邮件：enroo@enroo.com

公司网站：<http://www.enroo.com> <http://www.enroo-tech.com>

企业官网二维码



企业微信公众号二维码

