

深圳市晶曜电子科技有限公司

SHEN ZHEN MESH STAR ELECTRONICS TECHNOLOGY CO., LTD

WL2610F003_MCU CORE

深圳市晶曜电子科技有限公司

WL2610F003_MCU CORE

用户手册

Ver1.12

目 录

1. 概述.....	5
2. 特征.....	5
3. 管脚信息.....	6
3.1 管脚图.....	6
3.2 管脚说明.....	7
4. 系统框图及存储器结构.....	11
4.1 系统框图.....	11
4.2 程序存储器.....	11
4.3 EEPROM 数据存储区.....	12
4.4 RAM 数据存储区.....	12
5. SFR 特殊寄存器地址.....	13
5.1 SFR 寄存器列表.....	13
5.2 SFR 寄存器映像.....	14
6. SFR 标准功能.....	19
6.1 CPU 内核功能寄存器.....	19
6.1.1 PSW.....	19
6.1.2 DPS.....	20
6.2 系统寄存器和时钟频率.....	20
6.2.1 OSCM（模式控制寄存器）.....	20
6.2.2 PCON（电源模式寄存器）.....	21
6.2.3 FLRDS.....	21
6.2.4 CPUM.....	22
6.2.5 SIHRC.....	22
6.2.6 CHIPCON.....	22
6.2.7 RSTIQ.....	23
6.2.8 SRST.....	23
6.2.9 BTMCON.....	24
6.2.10 INTPS.....	24
6.2.11 WDTC.....	25
6.2.12 CRC 原理和应用简介.....	26
6.2.12.1 CRC 硬件功能的 C 语言程序实现过程.....	28
6.2.12.2 CRCCON 循环冗余校验控制寄存器.....	29
6.2.12.3 CRCDL, CRCDH 循环冗余校验数据寄存器.....	29
6.3 I/O 端口.....	31
6.3.1 外部地址特殊寄存器使能 XSFR.....	31
6.3.2 PnM 端口模式.....	32
6.3.3 Pn 数据寄存器.....	32
6.3.4 PnON NMOS 开漏.....	32
6.3.5 PnPD 端口下拉.....	33
6.3.6 PnPH 端口上拉.....	33
6.3.7 PnRS 驱动增强.....	33
6.3.8 PnW 唤醒寄存器.....	34
6.4 中断.....	35
6.4.1 中断寄存器.....	35
6.4.1.1 IE（中断使能 1）.....	35
6.4.1.2 IE2（中断使能 2）.....	36

6.4.1.3	INT23CON（外中断控制）	36
6.4.2	中断向量入口地址	36
6.4.3	中断处理	37
6.4.4	中断优先级	37
6.5	定时器	39
6.5.1	T0、T1 定时/计数器寄存器	39
6.5.2	T2 定时/计数器寄存器	42
7	SFR 专用功能模块	48
7.1	串行口外部设备接口 SPI	48
7.1.1	SPI（SPI 控制器）	48
7.1.2	SPIBUF（SPI 的缓冲器）	49
7.1.3	SPIC0（SPI 控制器 0）	49
7.1.4	SPIC1（SIM 控制器 1）	50
7.1.5	SPI 通信	51
7.2	增强型通用异步收发器（EUART）	52
7.2.1	工作方式	53
7.2.2	波特率	57
7.2.3	多机通信	57
7.2.4	帧出错检测	58
7.3	增强型通用异步收发器（EUART1）	59
7.3.1	工作方式	60
7.3.2	波特率 SBRTL、SBRTH	63
7.4	EEPROM	65
7.4.1	EECON（EE 控制寄存器）	65
7.4.2	EEADRL（EE 地址）	65
7.4.3	EEDATA（EE 数据）	66
7.4.4	EE 示例程序	66
7.5	FLASH	68
7.5.1	FLCON（FL 控制寄存器 1）	68
7.5.2	FLENA（FL 控制寄存器）	68
7.5.3	FLADRH/L（FL 地址高低位）	68
7.5.4	FLDATH/L（FL 数据高低位）	69
7.5.5	FLASH 操作示例程序	69
7.6	14 通道 ADC	70
7.6.1	概述	70
7.6.2	ADM（ADC 模式寄存器）	71
7.6.3	ADB（ADC 数据寄存器）	72
7.6.4	VREFH（ADC 参考电压寄存器）	73
7.6.5	ADC 操作说明和注意事项	73
7.6.5.1	ADC 信号格式	73
7.6.5.2	AD 转换时间	74
7.6.5.3	ADC 引脚配置	74
7.6.6	P2/P1/P0 口 ADC 共用引脚	75
7.6.7	ADT（ADC 结果微调寄存器）	76
7.6.8	ADF0（ADC 基准修调）	76
7.6.9	ADC 应用电路	77
7.6.10	ADC 程序示例	77
7.7	带死区互补的 12 位 PWM（TM1）	79

7.7.1	TM1 相关寄存器.....	79
7.7.1.1	使能寄存器 PWMEN.....	79
7.7.2.1	输出选择寄存器 PWMPS.....	79
7.7.2.2	极性寄存器 PWMINV.....	80
7.7.2.3	PWM0 计数寄存器 PWMTL、PWMTH.....	80
7.7.2.4	PWMCON 控制寄存器.....	80
7.7.2.5	PWM0 周期寄存器 PWM0PL、PWM0PH.....	81
7.7.2.6	PWM0 占空比寄存器 PWM0DL、PWM0DH.....	81
7.7.2.7	PWM1 死区时间寄存器 PWMD1L、PWMD1H.....	81
7.7.2.8	PWM 占空比寄存器 PWM2DL/PWM2DH~PWM5DL/PWM5DH.....	82
7.7.2.9	PWM 独立模式示例程序及示例图.....	82
7.7.2.10	PWM 互补模式示例程序及示例图.....	83
8.	系统时钟.....	84
8.1	概述.....	84
8.2	指令周期 Fcpu.....	84
9.	复位.....	85
9.1	概述.....	85
9.2	上电、掉电复位.....	85
9.3	看门狗复位.....	86
9.4	REST 脚复位 (P1.7).....	86
9.5	软件复位.....	86
10.	指令集.....	87
10.1	数据传送指令.....	87
10.2	逻辑运算指令.....	88
10.3	算术运算指令.....	89
10.4	转移指令.....	90
11.	配置说明.....	91
12.	烧写编程与调试模式.....	92
12.1	在线编程示意图.....	92
12.2	调试模式.....	92
13.	电气参数.....	94
13.1	极限参数.....	94
13.2	DC 特性 (TEMP=25℃、VDD=5V、全指令周期).....	94
13.3	ADC 参数.....	96
14.	封装及尺寸.....	100
14.1	TSSOP20 封装图及尺寸.....	100
14.2	SOP16 封装图及尺寸.....	101
14.3	QFN20 封装图及尺寸.....	102
15.	修正记录.....	102

WL2610F003_MCU CORE

用户手册

Ver1.12

1. 概述

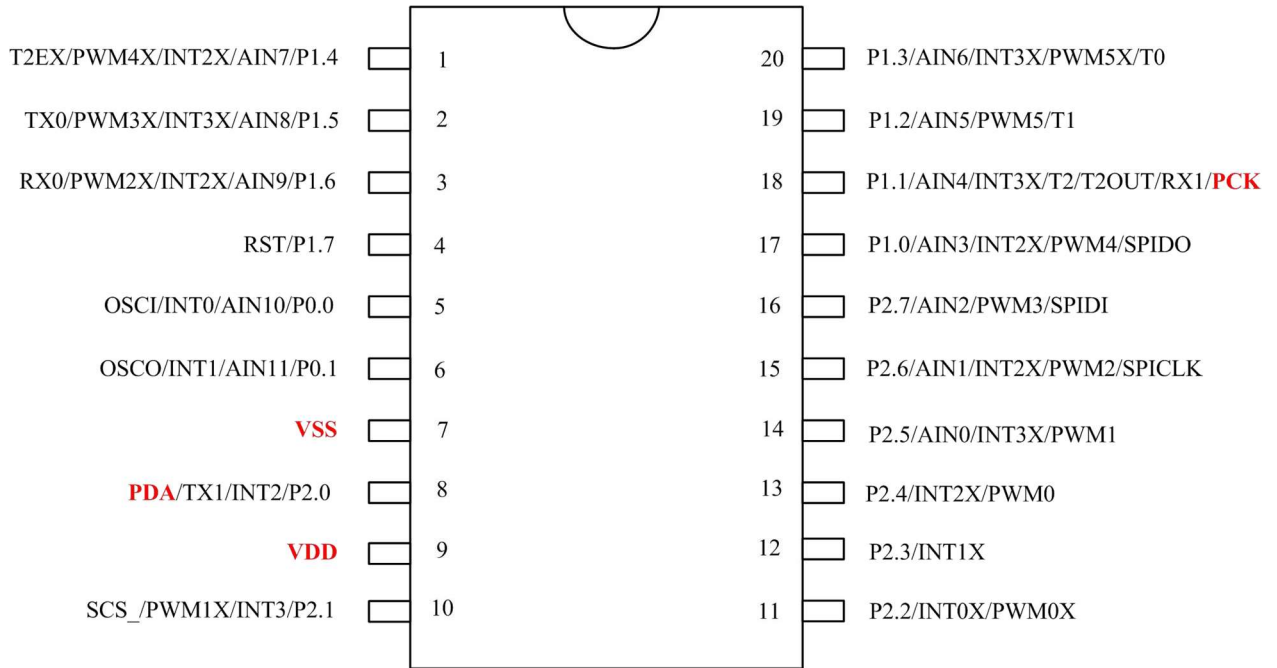
MCU_CORE是采用低功耗高速CMOS工艺制造的51内核单片机，它内建了16Kx8-bit的FLASH、256 Byte的EEPROM、256Byte+512byte的SRAM，包含6路12位PWM，集成了一个12位SAR ADC。

2. 特征

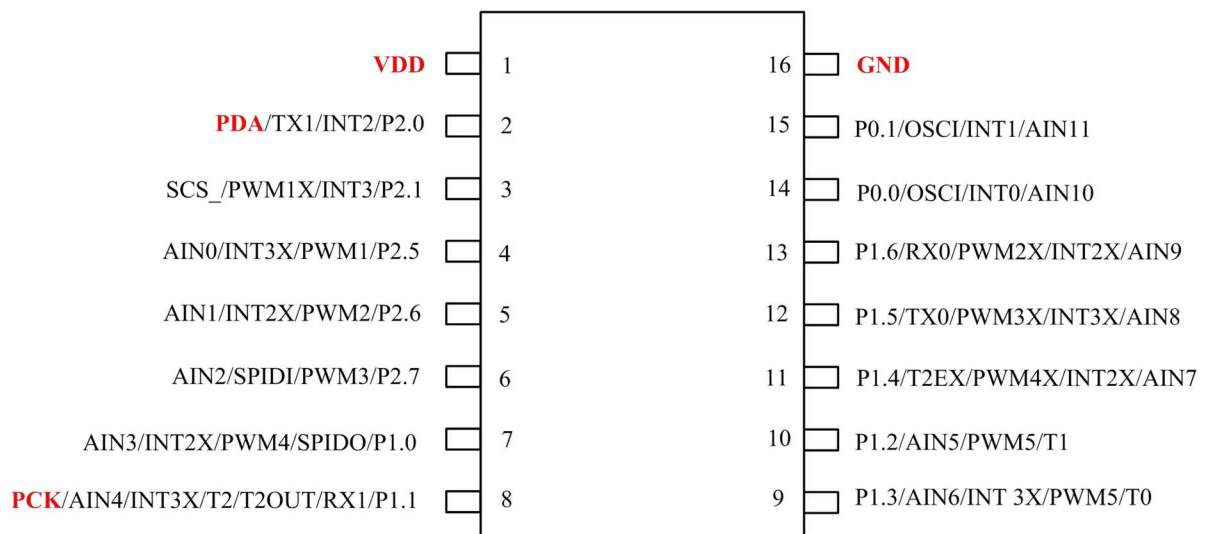
- ◆ 8051 指令，4 级中断优先设置，双数据指针(DPTRS)
- ◆ 16K×8 的程序存储器，灵活代码保护
- ◆ 256×8 位 EEPROM 数据存储区，10 年以上保存寿命
- ◆ 256×8 位 IRAM+ 512×8 位 XRAM
- ◆ T0、T1 定时器兼容传统 8051 的 T0、T1 定时/计数器
- ◆ T2 定时器兼容主流厂商的 T2 定时/计数器
- ◆ 一个增强型 EUART
- ◆ 一个兼容传统 8051 的 UART
- ◆ 4 组功能 IO 口，共 18 个 IO，驱动最大可达 100mA
- ◆ 内嵌片上调试功能，可进行在线仿真，2 线烧录和仿真接口
- ◆ 多组中断，T0/T1/T2,TI/RI,INT0/INT1/INT2/INT3/ADC/PWM/SPI/BTM，4 级中断优先级设置
- ◆ 内置 16MHz±1.5%高速振荡器，内置 128KHz±7%低速晶体振荡器。
- ◆ 支持 ISP 和 IAP 模式
- ◆ 内置 6 级 LVR
- ◆ SPI 通信接口
- ◆ WDT 可设置分频和时钟源
- ◆ 6 路共用周期 12bit PWM，可分为 3 组互补带死区 PWM
- ◆ 12 路 12bit ADC，可内建 1.5v、2v、3v 参考基准/可测 VDD
- ◆ 省电模式 IDLE，可任意端口变化唤醒
- ◆ sleep 模式，可外中断唤醒、端口变化唤醒、BTM 溢出唤醒
- ◆ 软件可设置主频分频和时钟源切换
- ◆ 宽工作电压范围：2.2V 至 5.5V
- ◆ 工作温度：-40~105℃
- ◆ ESD：6000V；Latch up：300mA
- ◆ 基于 KEIL™ 开发环境

3. 管脚信息

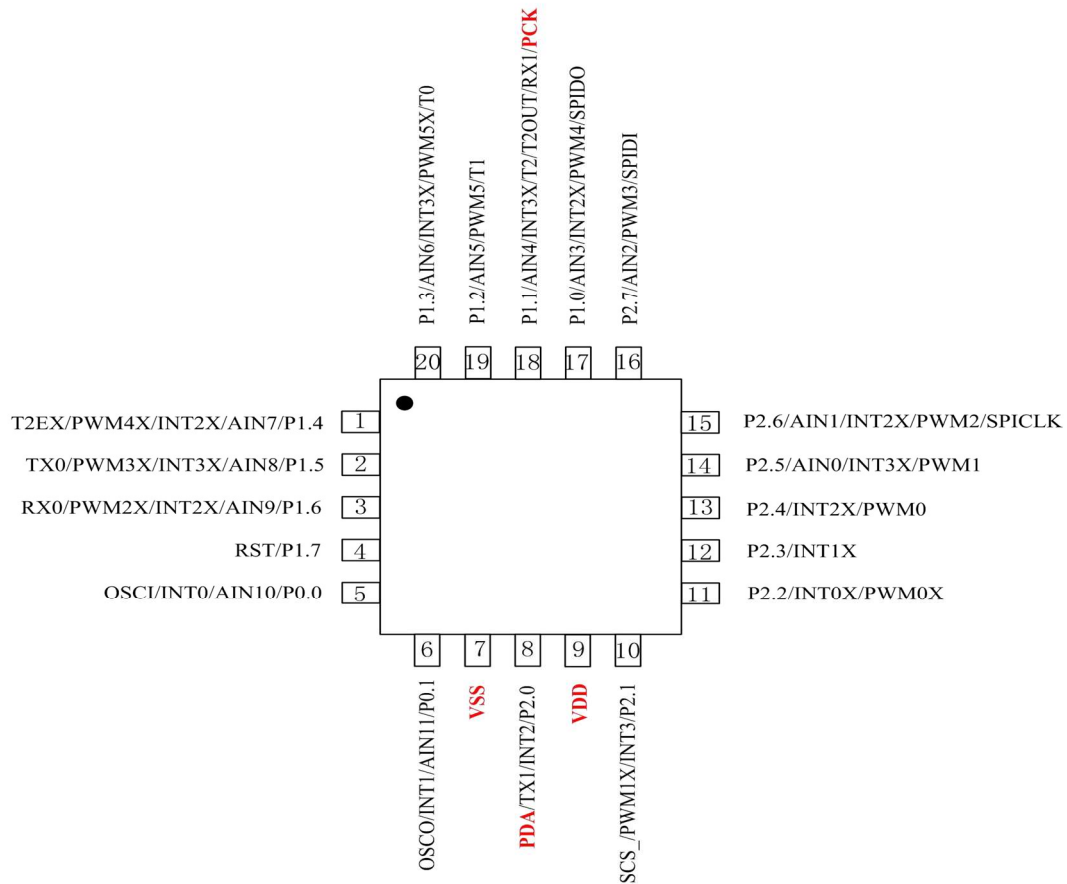
3.1 管脚图



管脚信息（TSSOP20）



管脚信息（SOP16）



管脚信息（QFN20）

注：X 表示该功能另外的选择脚位，和不带 X 的同时只能使用一个。

3.2 管脚说明

管脚名称	I/O	说 明
VDD	I	电源正端（+）输入端；
VSS	I	电源负端（-）输入端；
P0.0	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN10 可配置外中断 INT0 脚 可配置晶振输入脚
P0.1	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻；

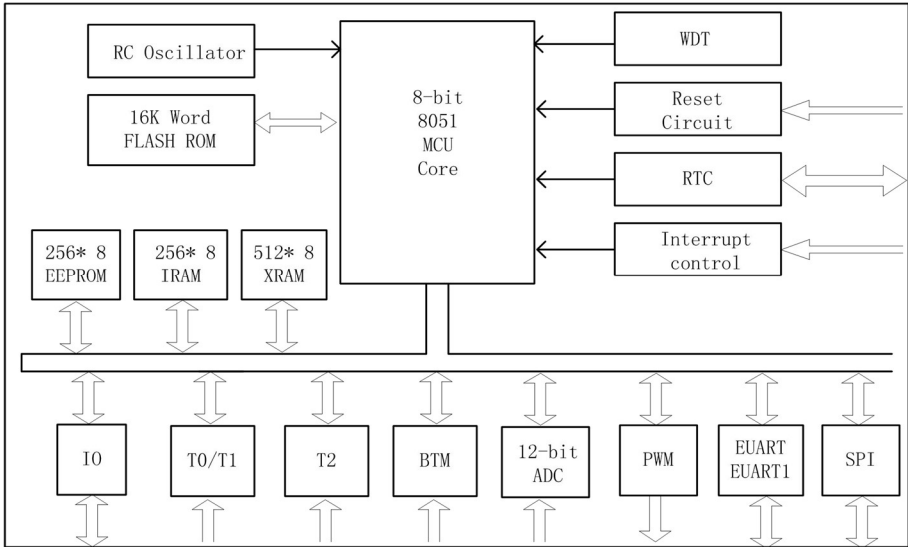
		可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN11 可配置外中断 INT1 脚 可配置晶振输出脚
P1.0	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN3 可配置外中断 INT2 脚 可配置 PWM4 输出口 可配置 SPI 数据输出口
P1.1	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN4 可配置外中断 INT3 脚 可配置 T2 外部时钟输入脚 可配置 T2 1/2 占空比输出脚 可配置 UART1 数据输入脚 烧写时钟脚 PCK
P1.2	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN5 可配置 PWM5 输出口 可配置 T1 外部时钟输入脚
P1.3	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN6 可配置外中断 INT3 脚 可配置 PWM5 输出口 可配置 T0 外部时钟输入脚
P1.4	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN7 可配置外中断 INT2 脚

		可配置 PWM4 输出口 可配置 T2EX 脚
P1.5	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN8 可配置外中断 INT3 脚 可配置 PWM3 输出口 可配置 UART0 数据输出脚
P1.6	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN9 可配置外中断 INT2 脚 可配置 PWM2 输出口 可配置 UART0 数据输入脚
P1.7	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置外部 RST 复位脚
P2.0	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置外中断 INT2 脚 可配置 UART1 数据输出脚 烧写数据脚 PDA
P2.1	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置外中断 INT3 脚 可配置 PWM1 输出口 可配置 SPI 的 SCS_脚
P2.2	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置外中断 INT0 脚 可配置 PWM0 输出口
P2.3	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出

		驱动 4 级可选 可配置外中断 INT1 脚
P2.4	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置外中断 INT2 脚 可配置 PWM0 输出口
P2.5	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN0 可配置外中断 INT3 脚 可配置 PWM1 输出口
P2.6	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN1 可配置外中断 INT2 脚 可配置 PWM2 输出口 可配置 SPI 的时钟脚
P2.7	I/O	双向输入输出脚，输入时施密特触发，内置上下拉电阻； 可开漏输出 驱动 4 级可选 可配置 AD 采样脚 AIN2 可配置外中断 INT3 脚 可配置 SPI 的数据输入脚

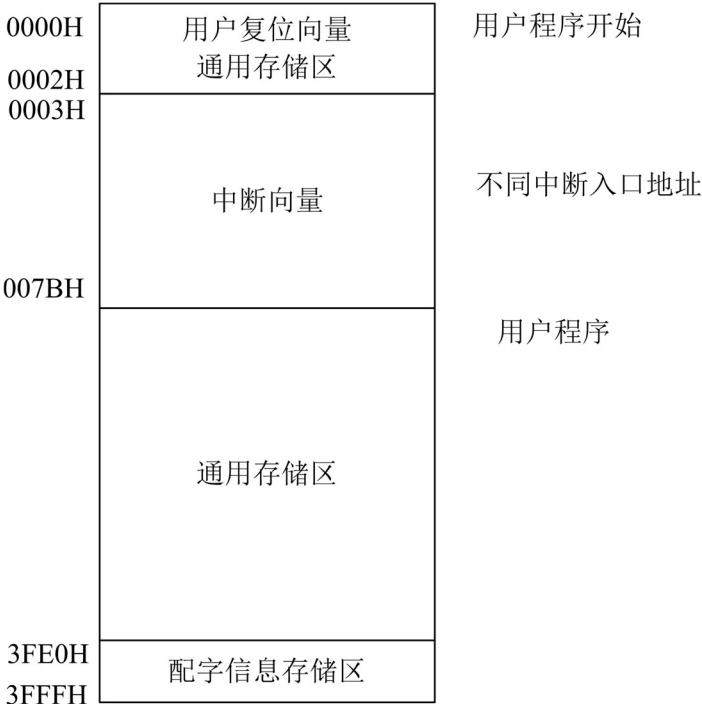
4. 系统框图及存储器结构

4.1 系统框图



4.2 程序存储器

MCU_CORE 单片机程序计数器 PC 为 14 位，能够访问 16K*8-bit 的程序存储器区域，复位地址 0000H，不同中断对应中断入口不同，分别为 0003H~007BH，相邻两个中断入口间隔 8 个字节地址。程序存取器结构如下图所示：



注：3FE0~3FFF 为配置信息存储区，不可使用。

4.3 EEPROM 数据存储区

单片机内建了 256*8 的 EEPROM 数据存储区，可以通过用户程序对此数据区进行访问操作，也可使用外部程序烧录器对此区域进行读写访问。

4.4 RAM 数据存储区

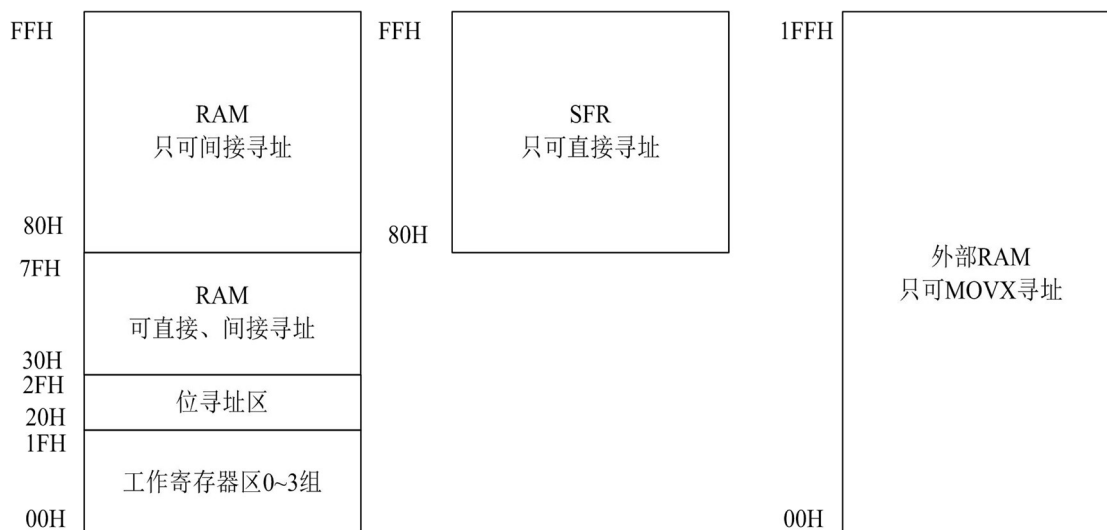
MCU_CORE 的内部 RAM 区分为以下三个独立空间：

- ◆ 低 128 字节 RAM（00H~7FH）可直接或间接寻址。
- ◆ 高 128 字节 RAM（80H~FFH）只能间接寻址。
- ◆ 特殊功能寄存器 SFR（80H~FFH）只能直接寻址。

注：高 128RAM 字节占用的空间和 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 字节的数据 RAM 还是 SFR。未使用的 SFR 禁止读写。

为了提高系统数据处理能力，MCU_CORE 提供了额外的外部 RAM 空间：

- ◆ 地址 00H~1FFH 为外部 RAM 空间，只能使用 MOVX 进行寻址。



5. SFR 特殊寄存器地址

5.1 SFR 寄存器列表

内部 SFR（起始地址 0080）：

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	P0	SP	DPL	DPH	DPL1	DPH1	DPS	PCON	TCON	TMOD	TL0	TL1	TH0	TH1	TCON1	SIHRC
9	P1	XSFR	OSCM	WDTC	RSTIQ	SRST	FLRDS	CPUM	SCON	SBUF	SADDR	SADREN	CHIPCON	CRCCON	CRCDL	CRCDH
A	P2	FLCON	FLADRL	FLADRH	FLDATL	FLDATH	FLENA	IE2	IE					SPIC0	SPIC1	SPIBUF
B		EECON	EEADRL		EEDATA	IP1H	IP1	IPH	IP							
C		ADM	ADB	ADR	ADT	VREFH	ADPCON	ADPCON1	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	INTPS	BTMCON
D	PSW	PWMTL	PWMTH	PWMPL	PWMPH	PWMINV	PWMPS	INT23CON								
E	A	PWM0DL	PWM0DH	PWM1DL	PWM1DH	PWM2DL	PWM2DH	PWMCON								
F	B	PWM3DL	PWM3DH	PWM4DL	PWM4DH	PWM5DL	PWM5DH	PWMEN	SCON1	SBUF1	SADDR1	SADREN1	SBRTL	SBRTH		

注： 域为可位寻址寄存器，其余寄存器皆不可位寻址。

外部地址 SFR（起始地址 FF00）：

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	P0M	P0OD	P0PD	P0PH	P0RS	P0W										
1	P1M	P1OD	P1PD	P1PH	P1RS	P1W										
2	P2M	P2OD	P2PD	P2PH	P2RS0	P2W	P2RS1	ADF0								
3																
4																
5																
6																
7																

注：访问外部地址的 SFR 需通过 MOVX 进行操作，并将 XSFR 寄存器设置为 F6H（上电默认值为 F6H）。

5.2 SFR 寄存器映像

内部 SFR:

地址	寄存器	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	读写	注释
80H	P0	P0<7:0>								R/W	P0_I/O 口
81H	SP	SP<7:0>								R/W	堆栈指针寄存器
82H	DPL	DPL<7:0>								R/W	数据指针 DPTR 低 8 位
83H	DPH	DPH<7:0>								R/W	数据指针 DPTR 高 8 位
84H	DPL1	DPL1<7:0>								R/W	数据指针 DPTR1 低 8 位
85H	DPH1	DPH1<7:0>								R/W	数据指针 DPTR1 高 8 位
86H	DPS	—	—	—	—	—	—	—	DPSEL	R/W	DPTR 切换寄存器
87H	PCON	SMOD	SSTAT	GF2	LVDTE	GF1	GF0	PD	IDL	R/W	电源模式寄存器
88H	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	R/W	定时器/计数器控制寄存器
89H	TMOD	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	R/W	定时器工作方式控制寄存器
8AH	TL0	TL0<7:0>								R/W	定时器 T0 低 8 位
8BH	TL1	TL1<7:0>								R/W	定时器 T1 低 8 位
8CH	TH0	TH0<7:0>								R/W	定时器 T0 高 8 位
8DH	TH1	TH1<7:0>								R/W	定时器 T1 高 8 位
8EH	TCON1	T1X12	T1PS2	T1PS1	T1PS0	T0X12	T0PS2	T0PS1	T0PS0	R/W	T0/T1 时钟源、分频控制寄存器
8FH	SIHRC	SIHRC<7:0>								R/W	寄存器配置 IHRC 的大小
90H	P1	P1<7:0>								R/W	P1_I/O 口
91H	XSFR	XSFR<7:0>								R/W	外部特殊寄存器使能
92H	OSCM	HIRCRDY	-	R128RDY	RTCEN	-	R128KEN	R16MEN	R32KEN	R/W	振荡器模式控制寄存器
93H	WDTC	WDTEN	—	WDTS	WDTPD	WDTCLR	WDTS2	WDTS1	WDTS0	R/W	WDT 控制寄存器
94H	RSTIQ	PORIF	EXRSTIF	BORIF	WDTIF	SRSTIF	—	—	—	R/W	系统工作状态标志
95H	SRST	SRST<7:0>									软件复位寄存器
96H	FLRDS	—	—	—	—	—	—	RDS1	RDS0	R/W	FLASH 读取速度寄存器
97H	CPUM	—	—	SCLKS1	SCLKS0	—	SCKQ2	SCKQ1	SCKQ0	R/W	系统时钟、分频控制寄存器

深圳市晶曜微电子科技有限公司
 SHEN ZHEN MESH STAR ELECTRONICS TECHNOLOGY CO., LTD
 WL2610F003_MCU CORE

98H	SCON	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI	R/W	串行口 0 控制寄存器
-----	------	--------	----------	-----------	-----	-----	-----	----	----	-----	-------------

地址	寄存器	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	读写	注释
99H	SBUF	SBUF<7:0>								R/W	串行口 0 数据缓存器
9AH	SADDR	SADDR<7:0>								R/W	定义从机地址
9BH	SADREN	SADEN<7:0>								R/W	位屏蔽寄存器，检验SADDR 有效位
9CH	CHIPCON	SMOD1	SSTAT1	—	IFENR	—	—	—	—	R/W	程序控制寄存器
9DH	CRCCON	CRC_ST	CRCIF	—	—	—	—	—	CRC_RL	R/W	CRC 冗余控制寄存器
9EH	CRCDL	CRCDL<7:0>								R/W	CRC 数据输出低 8 位
9FH	CRCDH	CRCDH<7:0>								R/W	CRC 数据输出高 8 位
A0H	P2	P2<7:0>								R/W	P2_I/O 口
A1H	FLCON	FLEN	—	DLYS1	DLYS0	—	ERASE	PROG	—	R/W	Flash 控制寄存器
A2H	FLADRL	FLADRL<7:0>								R/W	Flash 地址低位
A3H	FLADRH	FLADRH<7:0>								R/W	Flash 地址高位
A4H	FLDATL	FLDATL<7:0>								R/W	Flash 数据低位
A5H	FLDATH	FLDATH<7:0>								R/W	Flash 数据高位
A6H	FLENA	FLENA<7:0>								R/W	Flash 使能控制寄存器
A7H	IE2	-	ESPI	ES1	EX3	EX2	EPWM	—	—	R/W	中断控制寄存器 1
A8H	IE	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0	R/W	中断控制寄存器
ADH	SPIC0	SPIM2	SPIM1	SPIM0	—	—	—	SPIEN	—	R/W	SPI 控制寄存器 0
AEH	SPIC1	—	—	CKP	CKEG	MLS	SSEN	—	TRF/SPIF	R/W	SPI 控制寄存器 1
AFH	SPIBUF	SPIBUF<7:0>								R/W	SPI 数据缓冲器
BIH	EECON	—	—	DLYS1	DLYS0	ERSALL_EE	ERASE_EE	PROG_EE	READ_EE	R/W	EE 控制寄存器
B2H	EEADRL	EEADRL<7:0>								R/W	EE 地址低 8 位
B4H	EEDATA	EEDATA<7:0>								R/W	EE 数据寄存器
B5H	IP1H	IP1H<7:0>								R/W	中断优先级 1H
B6H	IP1	IP1<7:0>								R/W	中断优先级 1
B7H	IPH	IPH<7:0>								R/W	中断优先级 H

深圳市晶曜微电子科技有限公司
SHEN ZHEN MESH STAR ELECTRONICS TECHNOLOGY CO., LTD
WL2610F003_MCU CORE

地址	寄存器	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	读写	注释
B8H	IP	IP<7:0>								R/W	中断优先级
C1H	ADM	ADENB	ADS	EOC/ADCIF	GCHS	CHS3	CHS2	CHS1	CHS0	R/W	ADC 模式寄存器
C2H	ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	R/W	ADC 数据寄存器 11bit~4bit
C3H	ADR	—	ADCKS1	—	ADCKS0	ADB3	ADB2	ADB1	ADB0	R/W	ADC 数据寄存器 3bit~0bit
C4H	ADT	ADV1	ADV0	—	ADTS	ADT<3>	ADT<2>	ADT<1>	ADT<0>	R/W	ADC 结果微调寄存器
C5H	VREFH	EVHENB	—	—	—	—	—	VHS1	VHS0	R/W	ADC 参考电压寄存器
C6H	ADPCON	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	R/W	ADC 通道管脚配置寄存器
C7H	ADPCON1	—	—	—	—	ADPC11	ADPC10	ADPC9	ADPC8	R/W	ADC 通道管脚配置寄存器 1
C8H	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C_T2	CP_RL2	R/W	T2 控制寄存器
C9H	T2MOD	T2X12	T2PS2	T2PS1	T2PS0	—	—	T2OE	DCEN	R/W	T2 时钟源、分频控制寄存器
CAH	RCAP2L	RCAP2L<7:0>								R/W	定时器T2 重装、捕获寄存器低 8 位
CBH	RCAP2H	RCAP2H<7:0>								R/W	定时器T2 重装、捕获寄存器高 8 位
CCH	TL2	TL2<7:0>								R/W	定时器T2 低 8 位
CDH	TH2	TH2<7:0>								R/W	定时器T2 高 8 位
CEH	INTPS	INT0S	INT1S	INT2S<5:3>			INT3S<2:0>			R/W	外部中断端口选择寄存器
CFH	BTMCON	BTMEN	EBTM	BTMIF	BTMCKS	BTMS3	BTMS2	BTMS1	BTMS0	R/W	BTM 控制寄存器
D0H	PSW	Cy	AC	F0	RS1	RS0	OV	—	P	R/W	程序状态字控制寄存器
D1H	PWMTL	PWMTL<7:0>								R/W	PWM0 计数寄存器PWMTL
D2H	PWMTH	—	—	—	—	PWMTH[3:0]				R/W	PWM0 计数寄存器PWMTH
D3H	PWMPL	PWM0PL<7:0>								R/W	PWM0 周期寄存器低位
D4H	PWMPH	—	—	—	—	PWM0PH[3:0]				R/W	PWM0 周期寄存器高位
D5H	PWMINV	—	—	PWM5INV	PWM4INV	PWM3INV	PWM2INV	PWM1INV	PWM0INV	R/W	PWM 极性寄存器
D6H	PWMPS	—	—	PWM5PS	PWM4PS	PWM3PS	PWM2PS	PWM1PS	PWM0PS	R/W	PWM 输出端口选择寄存器
D7H	INT23CON	-	-	INT3EG	INT2EG	-	-	INT3F	INT2F	R/W	外中断 2、3 控制寄存器
E0H	A	A<7:0>								R/W	累加器
E1H	PWM0DL	PWM0DL<7:0>								R/W	PWM0 占空比寄存器低字节

深圳市晶曜微电子科技有限公司
SHEN ZHEN MESH STAR ELECTRONICS TECHNOLOGY CO., LTD
WL2610F003_MCU CORE

地址	寄存器	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	读写	注释
E2H	PWM0DH	—	—	—	—	PWM0DH<7:0>				R/W	PWM0 占空比寄存器高字节
E3H	PWM1DL	PWM1DL<7:0>								R/W	PWM1 占空比寄存器低字节
E4H	PWM1DH	—	—	—	—	PWM1DH<7:0>				R/W	PWM1 占空比寄存器高字节
E5H	PWM2DL	PWM2DL<7:0>								R/W	PWM2 占空比寄存器低字节
E6H	PWM2DH	—	—	—	—	PWM2DH<7:0>				R/W	PWM2 占空比寄存器高字节
E7H	PWMCON	PWMTR	PWMIF	—	PWMD	CKSEL	CK0[2]	CK0[1]	CK0[0]	R/W	PWM 控制寄存器
F0H	B	B<7:0>								R/W	B 寄存器
F1H	PWM3DL	PWM3DL<7:0>								R/W	PWM3 占空比寄存器低字节
F2H	PWM3DH	—	—	—	—	PWM3DH<7:0>				R/W	PWM3 占空比寄存器高字节
F3H	PWM4DL	PWM4DL<7:0>								R/W	PWM4 占空比寄存器低字节
F4H	PWM4DH	—	—	—	—	PWM4DH<7:0>				R/W	PWM4 占空比寄存器高字节
F5H	PWM5DL	PWM5DL<7:0>								R/W	PWM5 占空比寄存器低字节
F6H	PWM5DH	—	—	—	—	PWM5DH<7:0>				R/W	PWM5 占空比寄存器高字节
F7H	PWMEN	—	—	PWM5_EN	PWM4_EN	PWM3_EN	PWM2_EN	PWM1_EN	PWM0_EN	R/W	PWM 使能寄存器
F8H	SCON1	SM0_1/ FE_1	SM1_1/ RXOV_1	SM2_1/ TXCOL_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	R/W	串行口 1 控制寄存器
F9H	SBUF1	SBUF1<7:0>								R/W	串行口 1 数据缓存器
FAH	SADDR1	SADDR1<7:0>								R/W	串行口 1 定义从机地址
FBH	SADREN1	SADEN1<7:0>								R/W	串行口 1 位屏蔽寄存器
FCH	SBRTL	SBRTL<7:0>								R/W	串行口 1 波特率控制低字节
FDH	SBRTH	TREN	SBRTH<6:0>							R/W	串行口 1 波特率控制高字节

外部地址 SFR:

地址	寄存器	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	读写	注释
FF00H	P0M	—	—	—	—	—	—	P0M1	P0M0	R/W	P0 端口模式控制寄存器
FF01H	P0OD	—	—	—	—	—	—	P0OD1	P0OD0	R/W	P0 端口NMOS 开漏寄存器
FF02H	P0PD	—	—	—	—	—	—	P0PD1	P0PD0	R/W	P0 端口下拉寄存器
FF03H	P0PH	—	—	—	—	—	—	P0PH1	P0PH0	R/W	P0 端口上拉寄存器
FF04H	P0RS	—	—	—	—	—	—	P0RS1	P0RS0	R/W	P0 端口驱动增强寄存器
FF05H	P0W	—	—	—	—	—	—	P0W1	P0W0	R/W	P0 端口端口变化唤醒使能寄存器
FF10H	P1M	P1M <7:0>								R/W	P1 端口模式控制寄存器
FF11H	P1OD	P1OD <7:0>								R/W	P1 端口NMOS 开漏寄存器
FF12H	P1PD	P1PD <7:0>								R/W	P1 端口下拉寄存器
FF13H	P1PH	P1PH <7:0>								R/W	P1 端口上拉寄存器
FF14H	P1RS	P1RS <7:0>								R/W	P1 端口驱动增强寄存器
FF15H	P1W	P1W <7:0>								R/W	P1 端口端口变化唤醒使能寄存器
FF20H	P2M	P2M <7:0>								R/W	P2 端口模式控制寄存器
FF21H	P2OD	P2OD <7:0>								R/W	P2 端口NMOS 开漏寄存器
FF22H	P2PD	P2PD <7:0>								R/W	P2 端口下拉寄存器
FF23H	P2PH	P2PH <7:0>								R/W	P2 端口上拉寄存器
FF24H	P2RS0	P2RS0 <7:0>								R/W	P2 端口驱动增强寄存器 0
FF25H	P2W	P2W <7:0>								R/W	P2 端口端口变化唤醒使能寄存器
FF26H	P2RS1	P2RS1 <7:0>								R/W	P2 端口驱动增强寄存器 1
FF27H	ADF0	ADF0 <7:0>								R/W	ADF0 VREF 的修调值控制寄存器

6. SFR 标准功能

6.1 CPU 内核功能寄存器

累加器 ACC

ACC 是 8051 系列单片机最常用的特殊功能寄存器，在汇编指令代码中可以简写为 A，CPU 进行算术或者逻辑运算过程中，可以将操作数和运算结果存在 A 寄存器中。

B 寄存器

B 寄存器经常用在乘法或者除法运算中，对于 CPU 其他指令运算，B 寄存器也可以作为暂存寄存器。

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 (00-FF) 的任意地址，系统复位后，SP 初始化 07H，使得堆栈事实上由 08H 开始。

注：执行 LCALL/ACALL 时，SP+2；执行 RET/RETI 时 SP-2。

双数据指针 (DPTR)

DPTR 是一个 16 位专用寄存器，其高、低位字节寄存器用 DPH、DPL 表示。既可以联合使用，也可以作为 2 个独立 8 位寄存器 DPH 和 DPL 来处理，系统复位后，初始化为 00H。

DPTR1 是一个 16 位专用寄存器，其高、低位字节寄存器用 DPH1、DPL1 表示。既可以联合使用，也可以作为 2 个独立 8 位寄存器 DPH1 和 DPL1 来处理，系统复位后，初始化为 00H。

程序计数器 (PC)

程序计数器 PC 是一个 16 位的数据寄存器，不属于特殊功能寄存器范围，其内部装载需要执行指令序列的地址，用来控制指令执行的顺序，当单片机复位后，系统强制清除 PC 寄存器为 0x0000。

程序状态字 (PSW) 寄存器

程序状态字寄存器 PSW 存储了与程序运行相关的信息。

6.1.1 PSW

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
D0H(r/w)	PSW	Cy	AC	F0	RS1	RS0	OV	-	P
reset	00H	0	0	0	0	0	0	0	0

Cy: 进位标志位

0: 算术或逻辑运算中，没有进位或借位发生

1: 算术或逻辑运算中，没有进位或借位发生

Ac: 辅助标志位

0: 算术或逻辑运算中，D3 没有向 D4 进位或借位

1: 算术或逻辑运算中，D3 向 D4 进位或借位

Ac 标志位在 BCD 码运算时，配合 DA 指令用作十进制调整。

F0: 用户可自定义标志位。

RS1、RS0: R0~R7 寄存器页选择位

00: 页 0 (内部 RAM 地址 00H~07H)

01: 页 1 (内部 RAM 地址 08H~0FH)

10: 页 2 (内部 RAM 地址 10H~17H)

11: 页 3 (内部 RAM 地址 18H~1FH)

OV: 溢出标志位

0: 没有溢出发生

1: 有溢出发生

P: 奇偶校验位

0: 累加器 A 中值为 1 的个数为偶数

1: 累加器 A 中值为 1 的个数为奇数

6.1.2 DPS

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
86H(r/w)	DPS	-	-	-	-	-	-	-	DPSEL
reset	-	r0	r0	r0	r0	r0	r0	r 0	rw 0

DPSEL: DPTR 和 DPTR1 数据指针选择位

0: 选择当前数据指针为 DPTR

1: 选择当前数据指针为 DPTR1

6.2 系统寄存器和时钟频率

6.2.1 OSCM (模式控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
92H(r/w)	OSCM	HIRCRDY	-	R128RDY	RTCEN	-	R128KEN	R16MEN	R32KEN
reset	A0H	1	0	1	0	0	0	0	0

R32KEN: 低速 32K 振荡使能, 精度较低

0: 有效 (32K 内部低功耗振荡器起振,)

1: 无效 (SLEEP 模式下)

注: 低速 32K 仅在 R32KEN=1, 并且睡眠时, 才会关闭。

R16MEN: 高速振荡器控制位 (内部高速振荡)

0: 有效 (需和上位机配字控制位同时有效)

1: 无效

R128KEN: 低速 128K 振荡使能

0: 有效 (128K 内部低功耗振荡器起振)

1: 无效 (WDT 选择此信号源或者 BTM 开启时 128K 振荡使能自动有效)

注: 128K 做系统时钟时在 sleep 时会自动关闭, 但当 BTM 开启或者 WDT 选择此信号源后会自动开启。

注: 上电默认情况下, 高速 RC、低速 32K、低速 128K 都有效。

RTCEN: RTC 模式使能

0: 有效 (系统时钟为高速 RC, 外置低速晶振运行)

1: 无效 (系统时钟为高速 RC, 外置低速晶振停振)

注：需各上位机配字控制位同是时有效，RTC模式时sleep不会关闭外置低速晶振。

R128RDY：内部 128K 低功耗振荡器准备就绪标志，只读

0：未准备好

1：准备好

HIRCRDY：内部高速振荡器准备就绪标志，只读

0：未准备好

1：准备好

6.2.2 PCON（电源模式寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
87H(r/w)	PCON	SMOD	SSTAT	GF2	LVDTE	GF1	GF0	PD	IDL
reset	10H	0	0	0	1	0	0	0	0

SMOD：UART 波特率加倍位

0：UART 波特率不加倍

1：UART 波特率加倍

SSTAT：UART 模块 SCON 寄存器功能选择位

0：SCON[7:5]工作方式为 SM0，SM1，SM2

1：SCON[7:5]工作方式为 FE，RXOV，TXCOL

FE 功能使能后，UART 可以工作在数据帧错误检测模式；SM0 功能使能，UART 可以配合 SCON SM1 位进行不同串口通信模式的设置。

LVDTE：低压复位使能

0：低压复位功能关闭

1：低压复位功能打开

GF2、GF1、GF0：可做通用寄存器

PD：睡眠模式

0：无效，正常模式

1：有效，进入掉电模式（Sleep）。

注：Sleep 状态时，该位自动清 0。CPU 和振荡器停止工作，特殊部件可以工作。外中断、端口变化、BTM 溢出、WDT 溢出复位、MCLR 复位、BOR 复位可以唤醒睡眠模式。

IDL：空闲模式

0：无效，正常模式。

1：有效，进入空闲状态。

注：进入空闲状态后，该位自动清 0。此时 CPU 停止工作，振荡器继续工作。端口变化、WDT 溢出复位、MCLR 复位、BOR 复位可以唤醒空闲状态。

注：若 PD 和 IDL 同时为 1，则进入掉电模式。唤醒后标志同时被清除。

6.2.3 FLRDS

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
96H(r/w)	FLRDS	-	-	-	-	-	-	RDS1	RDS0

reset	00h	0	0	0	0	0	0	0	0
-------	-----	---	---	---	---	---	---	---	---

RDS<1:0>: 系统时钟选择（系统时钟频率）

00:flash 读取速度，全速；flash 读取功耗最大。

01:flash 读取速度，1/2；主频低于 4M 时。

10:flash 读取速度，1/2；主频低于 4M 时。

11:flash 读取速度，低功耗模式；（一般主频低于 1MHz 时，可以使用）

6.2.4 CPUM

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
97H(r/w)	CPUM	-	-	SCLKS1	SCLKS0	-	SCKQ2	SCKQ1	SCKQ0
reset		0	0	0	X	0	X	X	X

SCLKS<1:0>: 系统时钟选择（选择的对应时钟应在 sleep 状态时自动关闭）

00:系统时钟为内部高速 RC 振荡器，16MHz。

01:不可使用。

10:系统时钟为内部低速低功耗 RC 振荡器，128KHz；应设置低功耗 flash 读取。

11:系统为内部低速超低功耗 RC 振荡器，32KHz；应设置低功耗 flash 读取。

SCKQ<2:0>: 选择系统时钟时的系统输入频率的分频倍数（初始值由上位机配置确定）

000: $F_{osc}/1$ F_{osc} 为系统原始输入时钟，内外部振荡器。

001: $F_{osc}/2$

010: $F_{osc}/4$

011: $F_{osc}/8$

100: $F_{osc}/16$

101: $F_{osc}/32$

110: $F_{osc}/64$

111: $F_{osc}/128$

注：X 表示初始值由上位机配置确定，

注： **$F_{osc}/1$** 不可使用。

6.2.5 SIHRC

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
8FH(r/w)	SIHRC	寄存器配置IHRC 的大小							
reset	C0H	1	1	0	0	0	0	0	0

可以调整内部高速振荡器 HIRC 频率，SIHRC <7:0>数值减小则振荡器频率增大，反之则减小。

寄存器值默认 C0H，频率为 16MHZ；寄存器设置为 00H 频率最大，约 26MHZ；寄存器设置为 0FFH 频率最小，约 12.25MHZ。

6.2.6 CHIPCON

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----

9CH(r/w)	CHIPCON	SMOD1	SSTAT1	-	IFENR	-	-	-	-
reset	00H	0	0	0	0	0	0	0	0

SMOD1: UART1 波特率加倍位

0 : UART1 波特率不加倍

1 : UART1 波特率加倍

SSTAT1: UART1 模块 SCON1 寄存器功能选择位

0: SCON1[7:5]工作方式为 SM0_1, SM1_1, SM2_1

1: SCON1[7:5]工作方式为 FE_1, RXOV_1, TXCOL_1

FE_1 功能使能后, UART1 可以工作在数据帧错误检测模式; SM0_1 功能使能, UART1 可以配合 SCON1 SM1_1 位进行不同串口通信模式的设置。

IFENR: MOVC 指令访问区域设置

0: MOVC 指令访问正常区域

1: MOVC 指令访问 IFEN 区域

6.2.7 RSTIQ

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
94H(r/w)	RSTIQ	PORIF	EXRSTIF	BORIF	WDTIF	SRSTIF	-	-	-
reset	80H	1	0	0	0	0	0	0	0

PORIF: 上电复位标志位

0: 无上电复位;

1: 有上电复位, 软件清 0

EXRSTIF: 外部管脚复位标志位

0: 无外部 RST 管脚复位;

1: 外部 RST 管脚复位, 软件清 0

BORIF: 欠压复位标志位

0: 无欠压 BOR 复位;

1: BOR 复位 (BOR 值可上位机设置), 软件清 0

WDTIF: WDT 溢出标志位

0: 无 WDT 溢出

1: WDT 溢出, 软件清 0

SRSTIF: 软件复位标志位

0: 无软件复位

1: 软件复位发生

6.2.8 SRST

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
95H(r/w)	SRST	软件复位寄存器							
reset	00h	0	0	0	0	0	0	0	0

执行 MOV SRST,#5AH 即可复位。

执行 MOV SRST,#A5H 即可复位。

6.2.9 BTMCON

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
CFH(r/w)	BTMCON	BTMEN	EBTM	BTMIF	BTMCKS	BTMS3	BTMS2	BTMS1	BTMS0
reset	00H	0	0	0	0	0	0	0	0

BTMEN: BTM 定时器使能

0: 禁止 BTM;

1: 使能 BTM;

EBTM: BTM 定时器中断使能

0: 禁止 BTM 中断;

1: 使能 BTM 中断;

BTMIF: BTM 溢出标志

0: BTM 没有溢出;

1: BTM 溢出; 中断程序执行后, 自动清 0。如果不开中断, 只能软件清 0。

BTMCKS: BTM 定时器时钟源选择

0: 选择内置 128K 振荡器;

1: 选择外部 32.768K 晶体振荡器 (IRC&RTC 模式开启)

BTMS<3:0>: BTM 溢出时间选择

0000: 15.625ms

0001: 31.25ms

0010: 62.5ms

0011: 125ms

0100: 0.25s

0101: 0.5s

0110: 1s

0111: 2s

1000: 4s

BTM 定时器定时溢出后会自动清除, 重新计数。

6.2.10 INTPS

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
CEH(r/w)	INTPS	INT0S	INT1S	INT2S<2:0>			INT3S<2:0>		
reset		0	0	0	0	0	0	0	0

通过更改该寄存器参数, 可以将 INT0, INT1, INT2, INT3 映射到芯片不同的管脚上, 扩展了芯片大部分管脚接收外部中断触发信号的能力。

INT0S: INT0 管脚选择

0: 选择 P0.0 作为 INT0 中断脚;

1: 选择 P2.2 作为 INT0 中断脚。

INT1S: INT1 管脚选择

0: 选择 P0.1 作为 INT1 中断脚;

1: 选择 P2.3 作为 INT1 中断脚。

INT2<2:0>: INT2 管脚选择

000: P2.0 作为 INT2 中断脚
 001: P2.4 作为 INT2 中断脚
 010: P2.6 作为 INT2 中断脚
 011: P1.0 作为 INT2 中断脚
 100: P1.2 作为 INT2 中断脚
 101: P1.4 作为 INT2 中断脚
 110: P1.6 作为 INT2 中断脚
 111: P1.6 作为 INT2 中断脚

INT3<2:0>: INT3 管脚选择

000: P2.1 作为 INT3 中断脚
 001: P2.5 作为 INT3 中断脚
 010: P2.7 作为 INT3 中断脚
 011: P1.1 作为 INT3 中断脚
 100: P1.3 作为 INT3 中断脚
 101: P1.5 作为 INT3 中断脚
 110: P1.7 作为 INT3 中断脚
 111: P1.7 作为 INT3 中断脚

注：外中断切换端口后，建议手动清除对应中断的标志位。

6.2.11 WDTC

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
93H(r/w)	WDTC	WDTEN	-	WDTS	WDTPD	WDTCLR	WDTS2	WDTS1	WDTS0
reset	83H	1	0	0	0	0	0	1	1

WDTEN: WDT 使能位

0: 禁止 WDT;
 1: 使能 WDT;

WDTS: WDT 时钟选择

0: WDT 的时钟源为低精度 32K, 未修调 (LRC32K);
 1: WDT 的时钟源为高精度 32K, 已修调 (LRC128K/4 分频);

WDTPD: (上位机选择 disable when sleep)

0: sleep 时无法唤醒
 1: sleep 时 wdt 溢出会唤醒系统, 但不会复位, 也会置位 WDTIF

WDTCLR: 清 WDT

0: 无影响;
 1: 清 0 WDT 计数器, 一个指令周期后, 硬件自动复位此位;

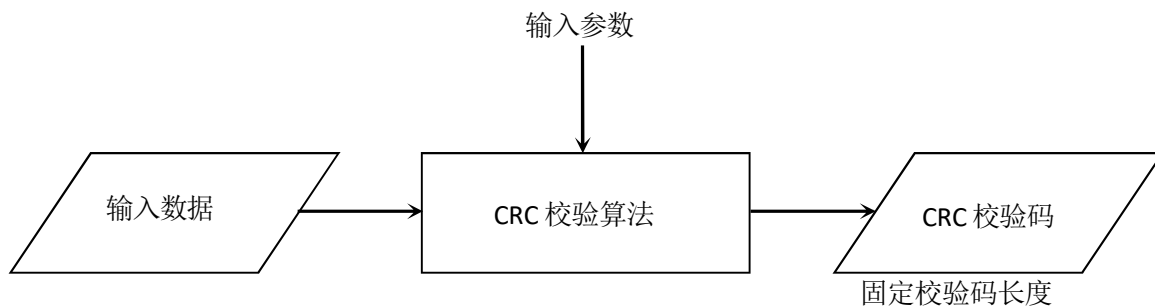
WDTS<2:0>: WDT 复位时间选择

000: 128ms
 001: 256ms
 010: 512ms
 011: 1.02s (上电默认)
 100: 2.05s
 101: 4.10s
 110: 8.2s

看门狗定时器（WDT）是一个递减计数器，独立内部 RC 振荡器作为时钟源。当定时器溢出时，将芯片复位，通过寄存器 WDTEN 位以及上位机选项打开或关闭该功能。看门狗溢出时间通过 WDTIS[2:0]选择，看门狗溢出时将 WDTIF 硬件置位，溢出之后重新开始计数。

6.2.12 CRC 原理和应用简介

单片机在数据传输过程中，因信号干扰或者信号传输通道不稳定等因素，会造成数据传输错误。为了确保发送端和接收端传送接收数据的一致性，经常采用的方法是引入数据 CRC 循环冗余校验算法。数据发送端首先将要发送的数据块进行 CRC 校验并生成 CRC 校验码，然后将 CRC 校验码与数据块合并发送给数据接收端，接收端使用同样的 CRC 校验算法对接收的数据块进行 CRC 校验并生成 CRC 校验码，将结果 CRC 校验码与数据块中发送端 CRC 校验码对比是否相等，判断数据块是否存在接收传输错误。



CRC 数据校验原理图

信息数据的按位异或多项式除法技术是 CRC 算法比较常用的，这种技术非常适合低阶的硬件操作，比如应用在单片机微控制器数据处理领域。

CRC 算法模块的输入参数有以下三个部分组成：

<1>. 被除数：也叫输入数据，可以缩写为“Input_Data[bLength]”

<2>. 除数：也叫多项式产生器，可以缩写为“POLY”

<3>. 一个被初始化的 CRC 值：可以缩写为“Initial_Crc”

下列步骤和 CRC 数据校验流程图阐明了 CRC 运算的详细过程：

<01>. CRC 运算开始

<02>. 数据缓存区长度 bLength 清 0，CRC 初始变量 Initial_Crc 清 0

<03>. 初始变量 Initial_Crc 与输入数据 Input_Data[bLength] 异或结果赋值给 CRC 运算变量 CrcVariable

<04>. 位变量 BitIndex 清 0

<05>. 判断运算变量 CrcVariable MSB 最高位是否为 1

<06>. 变量 CrcVariable 左移 1 位后与多项式系数 POLY 异或结果赋值给变量 CrcVariable

<07>. 变量 CrcVariable 左移 1 位

<08>. 位变量 BitIndex 增加 1

<09>. 判断位变量 BitIndex 是否小于输入数据位宽(输入数据所有位是否全部运算完成)

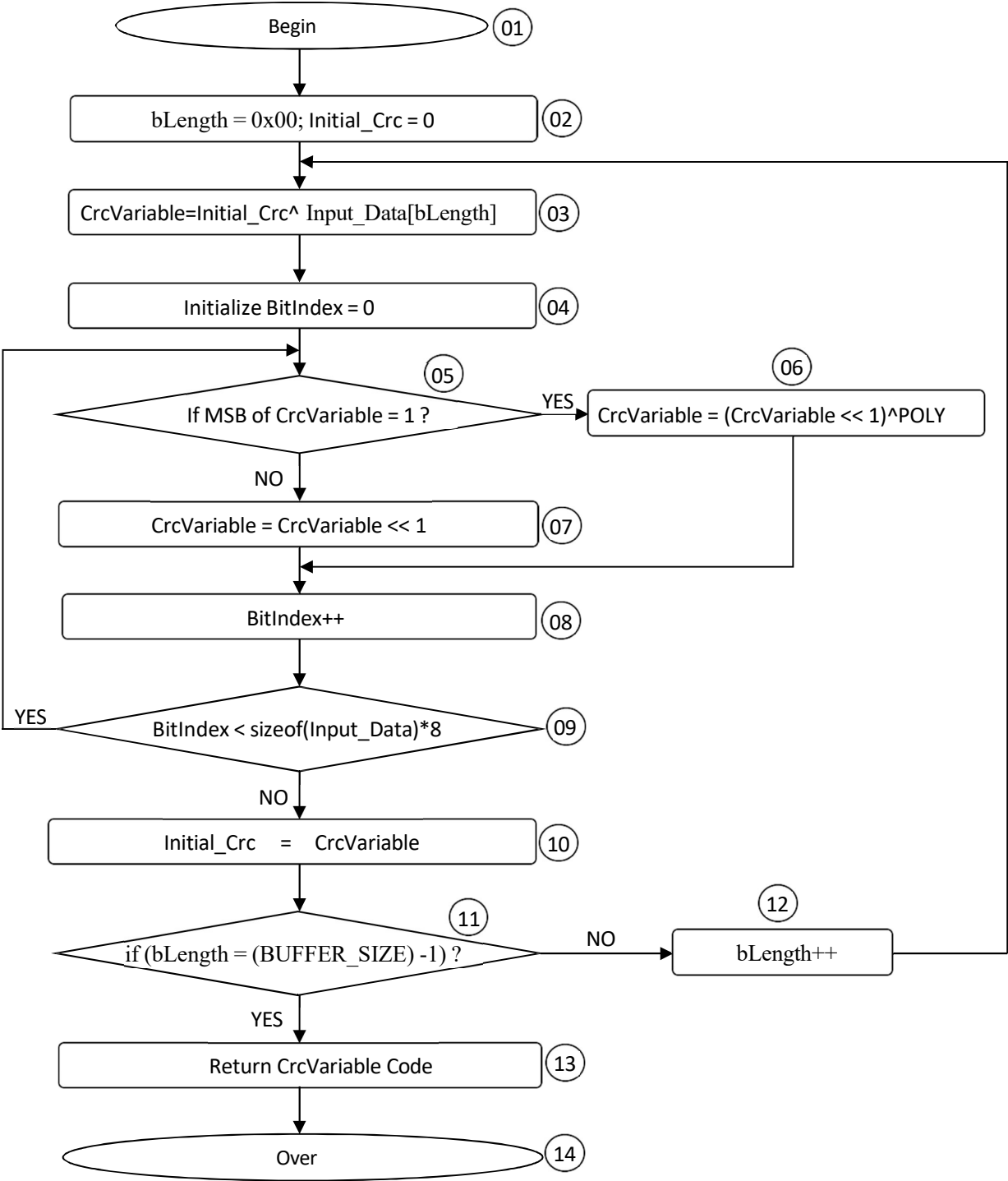
<10>. 运算变量 CrcVariable 赋值给初始变量 Initial_Crc

<11>. 判断输入缓存所有数据是否全部运算完成

- <12>. 数据缓存区长度 bLength 增加 1，指向下一个输入数据

<13>. 返回数据缓存区 CRC 校验码

<14>. CRC 运算结束



CRC 数据块校验流程

下表列举了初始变量 Initial_Crc=0x00，输入数据 Input_Data = 0xC1，多项式系数 POLY=0xCB，经过 8 次移位异或运算后，结果 CRC 校验码 CrcVariable = 0x9B 的运算过程：

BitIndex	Execution step	Binary format	Hex
----------	----------------	---------------	-----

	Initial_Crc=0x00;Input_Data=0xC1	0	0	0	0	0	0	0	0										00	
	CrcVariable = Initial_Crc^Input_Data	1	1	0	0	0	0	0	1										C1	
0	CrcVariable << 1	1	1	0	0	0	0	0	1										C1	
			1	0	0	0	0	0	1	0									82	
	CrcVariable = CrcVariable^ POLY		1	1	0	0	1	0	1	1	POLY								CB	
1	CrcVariable << 1		0	1	0	0	1	0	0	1									49	
2	CrcVariable << 1			1	0	0	1	0	0	1	0								92	
					0	0	1	0	0	1	0	0							24	
	CrcVariable = CrcVariable^ POLY				1	1	0	0	1	0	1	1	POLY						CB	
3	CrcVariable << 1				1	1	1	0	1	1	1	1							EF	
						1	1	0	1	1	1	1	0						DE	
	CrcVariable = CrcVariable^ POLY					1	1	0	0	1	0	1	1	POLY					CB	
4	CrcVariable << 1					0	0	0	1	0	1	0	1						15	
5	CrcVariable << 1						0	0	1	0	1	0	1	0					2A	
6	CrcVariable << 1							0	1	0	1	0	1	0	0				54	
7	CrcVariable << 1								1	0	1	0	1	0	0	0			A8	
									0	1	0	1	0	0	0	0			50	
	CrcVariable = CrcVariable^ POLY									1	1	0	0	1	0	1	1	POLY		CB
	CrcVariable(Returned value)										1	0	0	1	1	0	1	1	9B	

CRC 计算数据列表举例

6.2.12.1 CRC 硬件功能的 C 语言程序实现过程

MCU_CORE 内部集成硬件 CRC 运算模块，模块分为左移运算和右移运算两种运算方式，与模块功能等效的 C 语言范例如下：

左移运算方式：

```

unsigned short crc16_ccitt(unsigned char dat,unsigned short crc)
{
    unsigned short ccitt16=0x1021;
    int i;
    crc ^= (dat<<8);
    for(i=0;i<8;i++)
    {
        if(crc & 0x8000)
        {
            crc <<= 1;
            crc ^= ccitt16;
        }
        else crc <<= 1;
    }
    return crc;
}

```

右移运算方式：

```

unsigned short crc16_ccitt_r(unsigned char dat,unsigned short crc)
{
    unsigned short ccitt16_r=0x8408;
    int i;
    crc ^= dat;

```

```
for(i=0;i<8;i++)
{
    if(crc & 1)
    {
        crc >>= 1;
        crc ^= ccitt16_r;
    }
    else crc >>= 1;
}
return crc;
```

在实际数据包进行 CRC 运算过程中，为了提高数据处理速率，可以采用 MCU 内部集成的 CRC 硬件运算模块对数据缓存进行 CRC 运算，程序设计步骤如下：

- <1>. 在 CRC 模块控制寄存器 CRCCON 中设置 CRC_RL 位决定采用左移或者右移算法
- <2>. 在 CRCCON 中置位 CRC_ST 位打开 CRC 模块冗余校验功能
- <3>. 初始化 CRC 数据寄存器 CRCDH 和 CRCDL 为 0
- <4>. 检测 CRCCON 寄存器运算完成位 CRCIF 是否置位
- <5>. CRCIF 置位则清除该标志位，如果数据全部运算完成则执行<6>.步骤，否则将 1byte 数据输入 CRCDL 数据寄存器，跳到<4>.步骤执行
- <6>. 数据寄存器 CRCDH 和 CRCDL 即为数据区的 CRC 校验码

6.2.12.2 CRCCON 循环冗余校验控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
9DH(w/r)	CRCCON	CRC_ST	CRCIF	-	-	-	-	-	CRC_RL
reset	00H	0	0	0	0	0	0	0	0

CRC_ST: CRC 运算使能控制位

- 0: 关闭 CRC 运算模块，写入 CRCDL, CRCDH 寄存器的数据与读出的数据相同
- 1: 打开 CRC 运算模块，读出的 CRCDL, CRCDH 的内容是写入该寄存器数据的运算结果

CRCIF: CRC 运算完成标志位

- 0: CRC 运算模块未启动或者转换未完成
- 1: CRC 运算转换完成

CRC_RL: CRC 运算过程左移或右移控制位

- 0: MSB First, CRC 模块为左移运算
- 1: LSB First, CRC 模块为右移运算

6.2.12.3 CRCDL, CRCDH 循环冗余校验数据寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
9EH(w/r)	CRCDL	CRC 数据输出低 8 位							
9FH(w/r)	CRCDH	CRC 数据输出高 8 位							
reset	00H	0	0	0	0	0	0	0	0

CRCDL,CRCDH 数据寄存器可以在 CRC_ST=0 时赋初值, 且 CRC_ST=1 时只能写 CRCDL 寄存器, CRC 模块单次数据处理时间为 8 个系统时钟周期, 如果系统时钟为 16MHz, 则完成一次运算只需要 0.5us; CRC 运算模块分为左移运算和右移运算两种运算方式, 运算多项式采用 CRC-CCITT 标准: MSB First 左移模式的多项式系数 POLY = 0x1021(参阅 C 语言范例左移运算方式), LSB First 右移模式的多项式系数 POLY = 0x8408(参阅 C 语言范例右移运算方式)。

注：IO 口相关寄存器在外部 SFR 区，操作 IO 口相关寄存器需将 XSFR 设为 F6H 或者不操作 XSFR（默认为 F6H）

6.3.2 PnM 端口模式

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FF00(w/r)	P0M	-	-	-	-	-	-	P0M1	P0M0
FF10(w/r)	P1M	P1M<7:0>							
FF20(w/r)	P2M	P2M<7:0>							
reset	00H	0	0	0	0	0	0	0	0

PnM[7:0]: Pn 模式控制位 (n=0、1、2)

0: 输入模式;

1: 输出模式。

注：端口方向，1 为输出，0 为输入。默认 0，输入可唤醒。

6.3.3 Pn 数据寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
80(w/r)	P0	-	-	-	-	-	-	P01	P00
90(w/r)	P1	P1<7:0>							
A0(w/r)	P2	P2<7:0>							
reset	FFH	1	1	1	1	1	1	1	1

注：端口数据寄存器上电默认是高。

6.3.4 PnON NMOS 开漏

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FF01(w/r)	P0ON	-	-	-	-	-	-	P0ON1	P0ON0
FF11(w/r)	P1ON	P1ON<7:0>							
FF21(w/r)	P2ON	P2ON<7:0>							
reset	00H	0	0	0	0	0	0	0	0

PnON[7:0]: Pn 开漏输出控制位 (n=0、1、2)

0: 开漏无效;

1: 开漏有效。

注：NMOS 开漏，只能输出 0。默认 0，置 1 有效。

6.3.5 PnPD 端口下拉

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FF02(w/r)	P0PD	-	-	-	-	-	-	P0PD1	P0PD0
FF12(w/r)	P1PD	P1PD<7:0>							
FF22(w/r)	P2PD	P2PD<7:0>							
reset	00H	0	0	0	0	0	0	0	0

PnPD[7:0]: Pn 下拉控制位 (n=0、1、2)

0: 下拉无效;

1: 下拉有效。

注: 输入下拉设置, 默认 0, 置 1 有效。

6.3.6 PnPH 端口上拉

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FF03(w/r)	P0PH	-	-	-	-	-	-	P0PH1	P0PH0
FF13(w/r)	P1PH	P1PH<7:0>							
FF23(w/r)	P2PH	P2PH<7:0>							
reset	00H	0	0	0	0	0	0	0	0

PnPH[7:0]: Pn 上拉控制位 (n=0、1、2)

0: 上拉无效;

1: 上拉有效。

注: 输入上拉设置, 默认 0, 置 1 有效。

6.3.7 PnRS 驱动增强

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FF04(w/r)	P0RS	P0RS<7:0>							
FF14(w/r)	P1RS	P1RS<7:0>							
FF24(w/r)	P2RS0	P2RS0<7:0>							
FF26(w/r)	P2RS1	P2RS1<7:0>							
reset	00H	0	0	0	0	0	0	0	0

P0RSEN&P0RS<0>: P0.0 口驱动选择

00: P0.0 口默认驱动

01: P0.0 口驱动加大

10: P0.0 口驱动更大

11: P0.0 口驱动最大

P0RSEN&P0RS<1>: P0.1 口驱动选择

00: P0.1 口默认驱动
01: P0.1 口驱动加大
10: P0.1 口驱动更大
11: P0.1 口驱动最大

P1RSEN&P1RSx (x=0~7) : P1 口驱动选择

00: P1.x 口默认驱动
01: P1.x 口驱动加大
10: P1.x 口驱动更大
11: P1.x 口驱动最大

P2RS1x&P2RS0x (x=0~7) : P2 口驱动选择

00: P2.x 口默认驱动
01: P2.x 口驱动加大
10: P2.x 口驱动更大
11: P2.x 口驱动最大

注：上位机配置 P0RSEN 与 P0RS<1:0>控制 P0 口驱动，上位机配置 P1RSEN 与 P1RS<7:0>控制 P0 口驱动，寄存器 P2RS1<7:0>与 P2RS0<7:0>控制 P2 口驱动。

6.3.8 PnW 唤醒寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FF05(w/r)	P0W	-	-	-	-	-	-	P0W1	P0W0
FF15(w/r)	P1W	P1W<7:0>							
FF25(w/r)	P2W	P2W<7:0>							
reset	00H	0	0	0	0	0	0	0	0

PnW[7:0]: Pn 唤醒控制位 (n=0、1、2)

0: 唤醒无效;

1: 唤醒有效。

注：睡眠或 IDLE 模式，至少 50us 变化才能唤醒，默认 0，置 1 唤醒使能。

6.4 中断

本设计共有 13 个中断源，每个中断源都存在对应的中断地址。当程序执行过程中，产生中断请求后并被许可后，PC 跳转到对应的中断程序中，执行完中断服务程序后返回源程序的中断点。13 个中断分为四层中断优先级，优先级高的中断可以在优先级低的中断正在执行时，跳转到高优先级中断服务程序。

6.4.1 中断寄存器

6.4.1.1 IE (中断使能 1)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
A8H(r/w)	IE	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
reset	00H	0	0	0	0	0	0	0	0

EA: 总中断允许控制位

0: 总中断禁止

1: 总中断允许

EADC: ADC 中断使能位

0: 禁止 ADC 中断

1: 允许 ADC 中断

ET2: T2 中断使能位

0: 禁止 T2 定时器溢出中断

1: 允许 T2 定时器溢出中断

ES: UART 中断使能位

0: 禁止 UART 串口中断

1: 允许 UART 串口中断

ET1: T1 中断使能位

0: 禁止 T1 定时器溢出中断

1: 允许 T1 定时器溢出中断

EX1: INT1 中断使能位

0: 禁止外中断 INT1 中断

1: 允许外中断 INT1 中断

ET0: T0 中断使能位

0: 禁止 T0 定时器溢出中断

1: 允许 T0 定时器溢出中断

EX0: INT0 中断使能位

0: 禁止外中断 INT0 中断

1: 允许外中断 INT0 中断

当 INT0 或 INT1 选择外部中断引脚低电平触发中断，中断引脚为低电平时，触发中断后中断标志位置 1，之后被硬件清 0，进入中断。中断服务程序结束后返回到主程序，此时若中断引脚还是外接低电平，则再次触发中断。

注：INT0 和 INT1 只有下降沿中断或者低电平中断，无上升沿中断。

6.4.1.2 IE2 (中断使能 2)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
A7H(r/w)	IE2	-	ESPI	ES1	EX3	EX2	EPWM	-	-
reset	00H	0	0	0	0	0	0	0	0

ESPI: SPI 中断使能位

0: 禁止 SPI 串口中断

1: 允许 SPI 串口中断

ES1: UART1 中断使能位

0: 禁止 UART1 串口中断

1: 允许 UART1 串口中断

EX3: INT3 中断使能位

0: 禁止外中断 INT3 中断

1: 允许外中断 INT3 中断

EX2: INT2 中断使能位

0: 禁止外中断 INT2 中断

1: 允许外中断 INT2 中断

EPWM: PWM (TM1) 中断使能位

0: 禁止 PWM 中断

1: 允许 PWM 中断

6.4.1.3 INT23CON (外中断控制)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
D7H(r/w)	INT23CON	-	-	INT3EG	INT2EG	-	-	INT3F	INT2F
reset	00H	0	0	0	0	0	0	0	0

INT3EG: INT3 边沿触发选择

0: INT3 下降沿中断

1: INT3 上升沿中断

INT2EG: INT2 边沿触发选择

0: INT2 下降沿中断

1: INT2 上升沿中断

INT3F: 中断标志位

0: INT3 中断无效

1: INT3 中断有效，中断响应后，硬件自动清 0

INT2F: 中断标志位

0: INT2 中断无效

1: INT2 中断有效，中断响应后，硬件自动清 0

6.4.2 中断向量入口地址

中断名称	SFR	中断标志位	SFR	中断地址	序号
EX0	IE	IE0	TCON	0003H	0
ET0	IE	TF0	TCON	000BH	1
EX1	IE	IE1	TCON	0013H	2

ET1	IE	TF1	TCON	001BH	3
ES	IE	TI、RI	SCON	0023H	4
ET2	IE	TF2	T2CON	002BH	5
EADC	IE	ADCIF	ADM	0033H	6
EBTM	BTMCON	BTMIF	BTMCON	003BH	7
—	—	—	—	0043H	8
—	—	—	—	004BH	9
EPWM	IE2	PWMIF	PWMCON	0053H	10
EX2	IE2	INT2F	I/O	005BH	11
EX3	IE2	INT3F	I/O	0063H	12
ES1	IE2	UART1IF		006BH	13
ESPI	IE2	SPIIF	SPIC1	0073H	14
—	—	—	—	007BH	15

其中：IE、TCON、SCON、T2CON 可以位寻址，可以对寄存器单个位进行读/写。其余寄存器的位不能独立读写。

两个中断入口间只相隔 8 字节，一般情况下难以安放一个完整的中断服务程序。因此通常在中断入口地址处放置一条无条件转移指令，使程序执行转向在其他地址存放的中断服务程序。

注：— 表示未使用。

6.4.3 中断处理

中断标志位被置后，就会根据条件执行中断，但要遵行以下条件：

中断执行只能在指令的最后一个周期内才会有效；如有的指令是 3 周期的，则中断不会在前两个周期执行，只会在第三个周期才能开始执行；

执行 ret, reti, 读写 IE, IE0, IP, IP0, IP2, IP2H 时，不能执行中断

如果比本中断优先级更高的中断在执行，则必须等待优先级高的中断执行完毕后才能执行本中断。

执行中断时，最好执行 push a; push psw 压栈，reti 时执行 pop psw; pop a。

中断调用子程序时，最好是一个完整的子程序，即子程序返回必须执行 ret 指令。

系统响应中断最少需要 4 个时钟周期，最多需要 10 个时钟周期。

6.4.4 中断优先级

每个中断源都被分成 4 个优先级，可根据不同需要设置对应独立的优先级。分别通过 IP、IPH、IP1、IP1H 中相应位来实现。

当系统在响应一个中断服务程序时，可响应更高优先级的中断，但不能响应同优先级或低优先级的另一个中断。

响应最高级中断服务时，不响应其他任何中断。如果不同中断优先级的中断源同时申请时，响应较高优先级的中断申请。

如果同优先级的中断源在指令周期开始时同时申请中断，那么内部查询序列确定中断请求响应

顺序。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
B8H(w/r)	IP	中断优先级							
B7H(w/r)	IPH	中断优先级h							
B6H(w/r)	IP1	中断优先级 1							
B5H(w/r)	IP1H	中断优先级 1h							
reset	00H	0	0	0	0	0	0	0	0

IPH, IP: 控制前 8 个中断优先级

00: 优先级最低

01: 低

10: 高

11: 最高

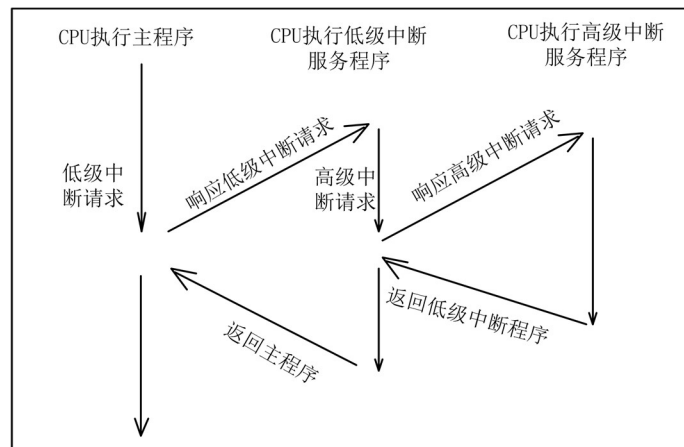
IP1H, IP1: 控制另外 8 个中断优先级

00: 优先级最低

01: 低

10: 高

11: 最高



两级中断嵌套过程

6.5 定时器

6.5.1 T0、T1 定时/计数器寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
8AH(w/r)	TL0	定时器 T0 低 8 位, 复位初始化为 00H							
8BH(w/r)	TL1	定时器 T1 低 8 位, 复位初始化为 00H							
8CH(w/r)	TH0	定时器 T0 高 8 位, 复位初始化为 00H							
8DH(w/r)	TH1	定时器 T1 高 8 位, 复位初始化为 00H							
89H(w/r)	TMOD	定时器 T0、T1 模式控制							
88H(w/r)	TCON	T0、T1 控制							
8EH(w/r)	TCON1	T0、T1 模式控制 1							

TMOD (模式控制)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
89H(r/w)	TMOD	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0
reset	00H	0	0	0	0	0	0	0	0

TxGATE: 门控制 $x=0,1$

0: 仅用 TR_x ($x=0,1$) 来控制定时器/计数器计数

1: 用外中断 ($/INT0$ 或 $INT1$) 上的高电平与 TR_x 共同控制定时器/计数器计数

TxM1、TxM0: 工作方式选择位 $x=0,1$

M1、M0 对应定时器 4 种工作方式,

M1	M0	工作方式
0	0	方式 0, 为 13 位定时器/计数器
0	1	方式 1, 为 16 位定时器/计数器
1	0	方式 2, 8 位常数自动重装载定时器/计数器
1	1	方式 3, 仅适用于 T0, 此时 T0 分为两个 8 位计数器, T1 停止计数

TxC/T: 定时、计数模式 $x=0,1$

0: 定时模式, 对内部时钟分频后的脉冲计数

1: 计数模式, 计数器对外部输入引脚 T0 (P1.3) 或 T1 (P1.2) 外部脉冲负跳变计数

TCON

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
88H(r/w)	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
reset	00H	0	0	0	0	0	0	0	0

TF0、TF1: 溢出标志位

定时器 0 或 1 溢出标志位, 计数溢出时, 硬件使 TF0、TF1 置“1”, 并申请中断。进入中断服务程序后, 由硬件自动清“0”, 在查询方式时, 需要软件清“0”。

TR0、TR1: 定时器 0、1 使能位

0: 停止 (T0 或 T1) 定时器/计数器工作

1: 启动 (T0 或 T1) 定时器/计数器工作

该位可由软件置“1”或清“0”。

IT0、IT1: INT0/INT1 触发选择位

0: 外部中断 0 或 1 (/INT0 或/INT1) 为低电平触发方式

1: 外部中断 0 或 1 (/INT0 或/INT1) 为下降沿跳变触发方式

IE0、IE1:

外部中断 0、1 请求标志位，硬件自动清“0”

TCON 中只有高 4 位与 T0、T1 定时器相关。

TCON1 (时钟控制)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
8EH(r/w)	TCON1	T1X12	T1PS2	T1PS1	T1PS0	T0X12	T0PS2	T0PS1	T0PS0
reset	00H	0	0	0	0	0	0	0	0

T0X12: T0 时钟源选择

0: T0 时钟源为 $F_{OSC}/12$; (可选分频系数)

1: T0 时钟源为 F_{OSC} 。(可选分频系数)

T1X12: T1 时钟源选择

0: T1 时钟源为 $F_{OSC}/12$; (可选分频系数)

1: T1 时钟源为 F_{OSC} 。(可选分频系数)

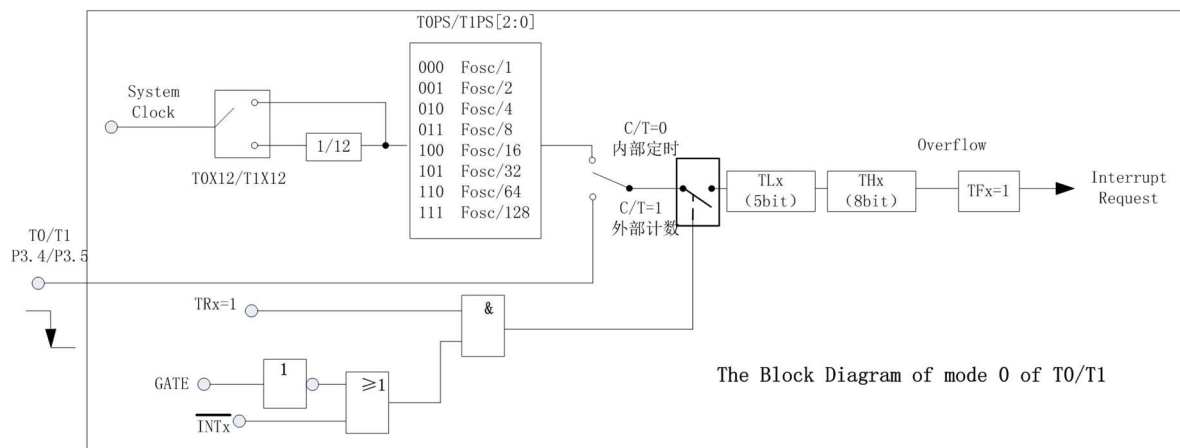
TxPS[2:0]: T0/T1 分频选择位。x=0,1

T0PS/T1PS[2:0]	分频系数	备注
000	$F_{osc} / 1$	分频选择只适用于选择内部时钟源时。当选择T0、T1脚作为时钟源计数时，分频不可选。
001	$F_{osc} / 2$	
010	$F_{osc} / 4$	
011	$F_{osc} / 8$	
100	$F_{osc} / 16$	
101	$F_{osc} / 32$	
110	$F_{osc} / 64$	
111	$F_{osc} / 128$	

方式 0: 13 位定时器/计数器 x=0,1

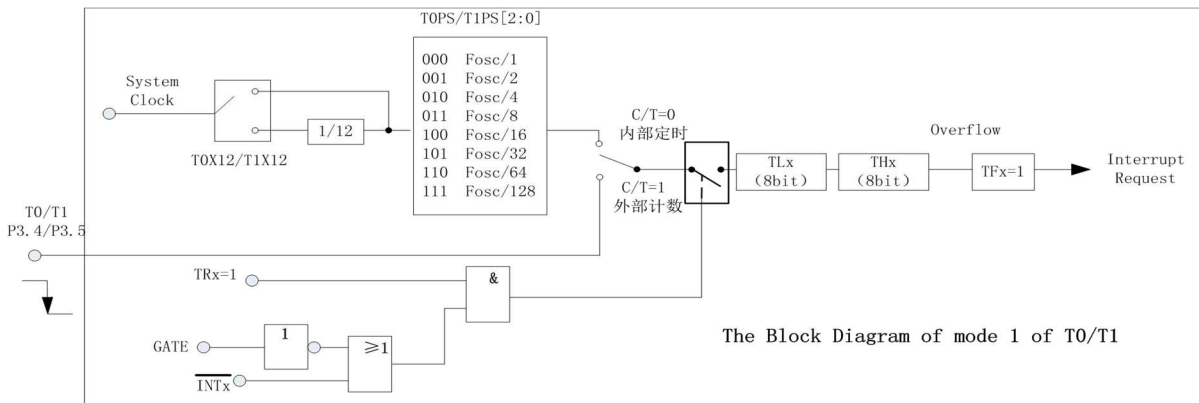
THx 寄存器存放 13 位定时器/计数器的高 8 位，TLx 存放低 5 位 (TLx4~TLx0)，TLx 高 3 位不确定可忽略。当计数溢出时，中断标志位 TFX 被置“1”，如果定时器中断被允许，则进入中断服务程序。

如果 TxC/T=1，定时器输入引脚 Tx 下降沿跳变，使定时器 Tx 计数加 1。如果 TxC/T=0，选择内部系统时钟作为定时器的时钟源。



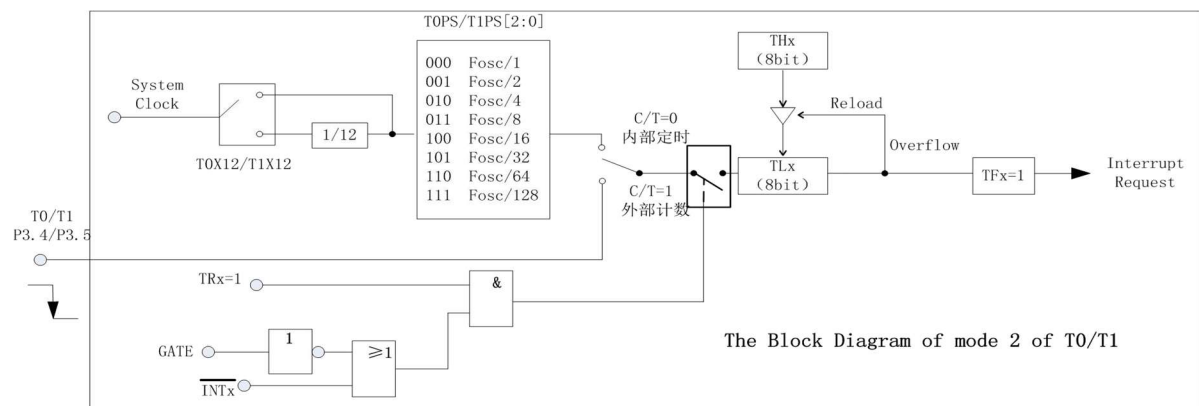
方式 1: 16 位定时器/计数器 x=0,1

THx、TLx 寄存器分别存 16 位定时器/计数器的高 8 位和低 8 位，其余和方式 0 相同



方式 2: 8 位自动重载定时器/计数器 x=0,1

TLx 存放计数值，THx 存放重载值。当 TLx 计数器溢出至 0x00 时，溢出标志位 TFX 置 1，THx 值被重载入 TLx 中。如果定时器中断使能，当 TFX 置 1 时产生中断。重载方式解决了在中断程序中对计数器反复赋初值而影响定时精度问题。

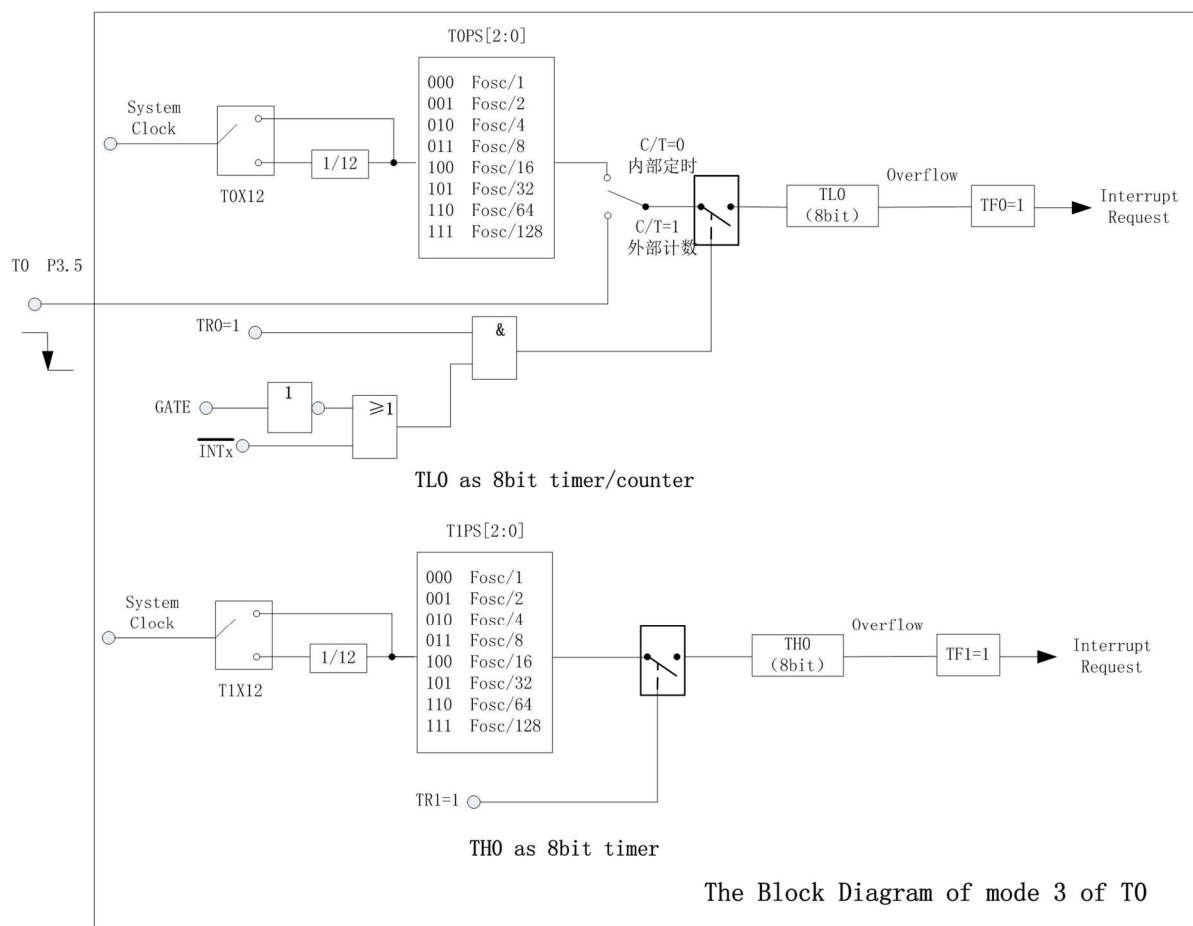


方式 3: 仅是适用 T0，T0 分成两个 8 位计数器

定时器 T0 分成两个独立的 8 位定时器/计数器，分别由 TL0 和 TH0 控制。TL0 由 TR0，T0C/T 和 TF0 控制。

TH0 只能作为定时器功能，使用 T1 的定时时钟。TH0 由定时器 1 的 TR1 控制使能，溢出时 TF1 置 1，控制定时器 T1 中断。

当 T0 工作在方式 3 时，T1 可以工作在方式 0、1 和 2，但是不能置 TF1 标志位和产生中断。可以用于不需要中断的场合。



6.5.2 T2 定时/计数器寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C8H(w/r)	T2CON	定时器 T2 控制寄存器							
C9H(w/r)	T2MOD	定时器 T2 模式寄存器							
CAH(w/r)	RCAP2L	定时器 T2 重装、捕获寄存器低 8 位，复位后初始状态为 00H							
CBH(w/r)	RCAP2H	定时器 T2 重装、捕获寄存器高 8 位，复位后初始状态为 00H							
CCH(w/r)	TL2	定时器 T2 低 8 位，复位后初始状态为 00H							
CDH(w/r)	TH2	定时器 T2 高 8 位，复位后初始状态为 00H							

定时器 2 是一个 16 位定时/计数器，既可以当定时器使用，也可作为外部事件计数器使用，其工作方式由特殊寄存器 T2CON 的 T2C/T 选择。定时器 2 有三种工作方式：捕获方式，自动重装载（向上或向下计数）方式和波特率发生器方式，工作方式由 T2CON 控制位选择。

T2CON

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----

C8H(r/w)	T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/_T2	CP/_RL2
reset	00H	0	0	0	0	0	0	0	0

TF2: 定时器溢出标志位

T2 溢出时，硬件置位，必须软件清 0。当 RCLK=1 或 TCLK=1 时，T2 溢出，不对 TF2 置位

EXF2: T2 外部标记

当 EXEN2=1 且 T2EX (P1.4) 引脚负跳变而出现捕捉或重装载时，EXF2 置位。如果中断使能，则响应中断，EXF2 只能软件清 0。当 T2 工作在向上或向下计数模式时 (DCEN=1)，EXF2 置位不引起中断。

RCLK: 接收时钟允许

0: T1 的溢出脉冲作为接收脉冲

1: T2 溢出脉冲作为串行口 (工作方式 1 或 3 时) 的接收时钟。

TCLK: 发送时钟允许

0: T1 的溢出脉冲作为发送脉冲

1: T2 溢出脉冲作为串行口 (工作方式 1 或 3 时) 的发送时钟。

EXEN2: T2 外部事件允许标志

0: T2EN 端的外部信号无效

1: 如果 T2 未作为串行口波特率发生器，在 T2EX 端出现负跳变脉冲时，激活 T2 捕捉或重装载。

TR2: T2 定时器使能位

0: 关闭 T2 定时器

1: 启动定时器 T2

C/_T2: T2 定时、计数选择位

0: 选择 T2 定时模式

1: 选择对外部事件计数 (P1.1 下降沿触发)

CP/_RL2: T2 模式选择位

CP/_RL2=1 且 ENEX2=1 时，T2EN/P1.1 端下降沿触发捕获

CP/_RL2=0 且 ENEX2=1 时，T2EN/P1.1 端下降沿触发重装

CP/_RL2=0 且 ENEX2=0 时，T2 溢出触发 T2 自动重装载

当 RCLK=1 或 TCLK=1 时，该位无效，T2 溢出时强制工作于自动重装载

RCLK+TCLK	CP/_RL2	TR2	MODE
0	0	1	16-bit 自动重装载
0	1	1	16_bit 捕获
1	×	1	波特率发生器
×	×	0	(off)

T2MOD

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C9H(r/w)	T2MOD	T2X12	T2PS2	T2PS1	T2PS0			T2OE	DCEN

reset	00H	0	0	0	0	0	0	0	0
-------	-----	---	---	---	---	---	---	---	---

T2X12: T2 时钟源选择位

- 0: T2 时钟源为 $F_{OSC}/12$; (可选分频系数)
1: T2 时钟源为 F_{OSC} 。(可选分频系数)

T2PS[2:0]: T2 分频选择位。

T2PS[2:0]	分频系数	备注
000	$F_{osc} / 1$	分频选择只适用于选择内部时钟源时。当选择T2脚作为时钟源计数时，分频不可选。
001	$F_{osc} / 2$	
010	$F_{osc} / 4$	
011	$F_{osc} / 8$	
100	$F_{osc} / 16$	
101	$F_{osc} / 32$	
110	$F_{osc} / 64$	
111	$F_{osc} / 128$	

分频系数对于 T2X12 为 0,1 都是有效的。

T2OE: T2 时钟输出使能位

- 0: 禁止 T2 输出
1: 允许 T2 输出

T2 输出使能用于对 **P1.1** 输出占空比 50%时钟信号。当 T2 自动重载时，T2OE 需要置 0。

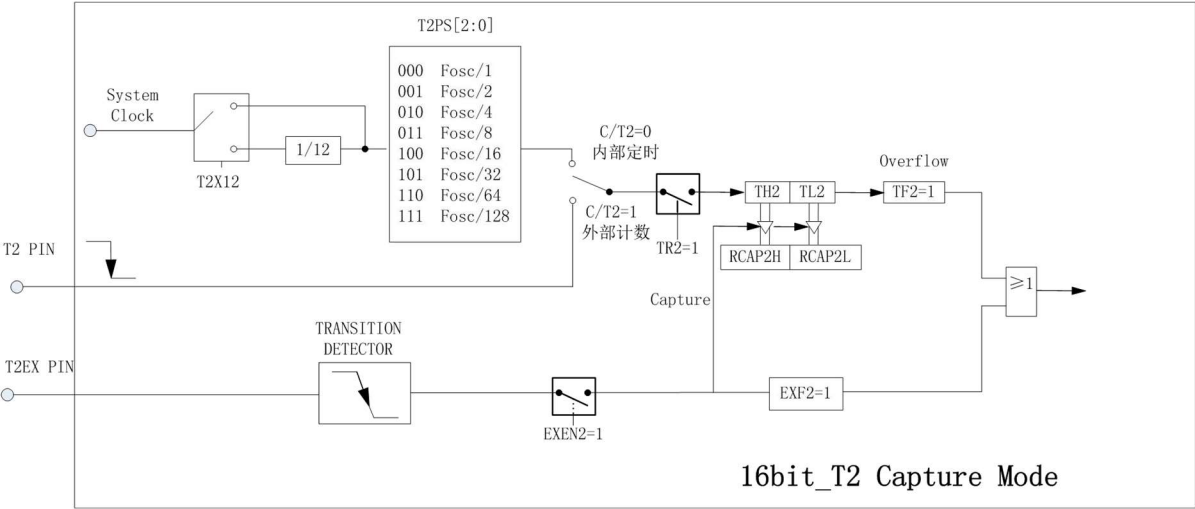
注: P1.1 需设置为输出模式才能输出占空比 50%时钟信号。

DCEN: T2 向上、向下计数选择位

- 0: 禁止 T2 向上和向下计数
1: 允许 T2 向上和向下计数

捕获方式

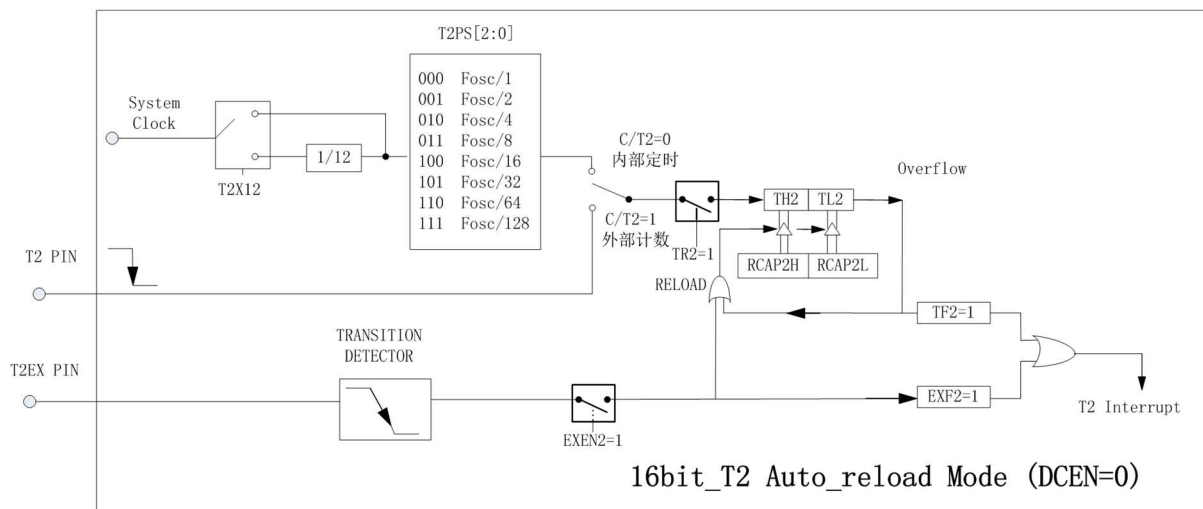
在捕获方式下，EXEN2控制位来选择两种方式。当 EXEN2=0，T2 是一个 16 位定时器或计数器。计数溢出时对 TF2 置位，同时触发中断。当 EXEN2=1，T2 完成相同动作。而当 T2EX 引脚外部输入信号发生负边沿跳变，TH2 和 TL2 的值分别被捕获到 RCAP2H 和 RCAP2L 中。另外，T2EX 引脚信号跳变使得 EXF2 置位，与 TF2 相似也会触发中断。捕获方式见下图。



定时器 T2 捕获方式

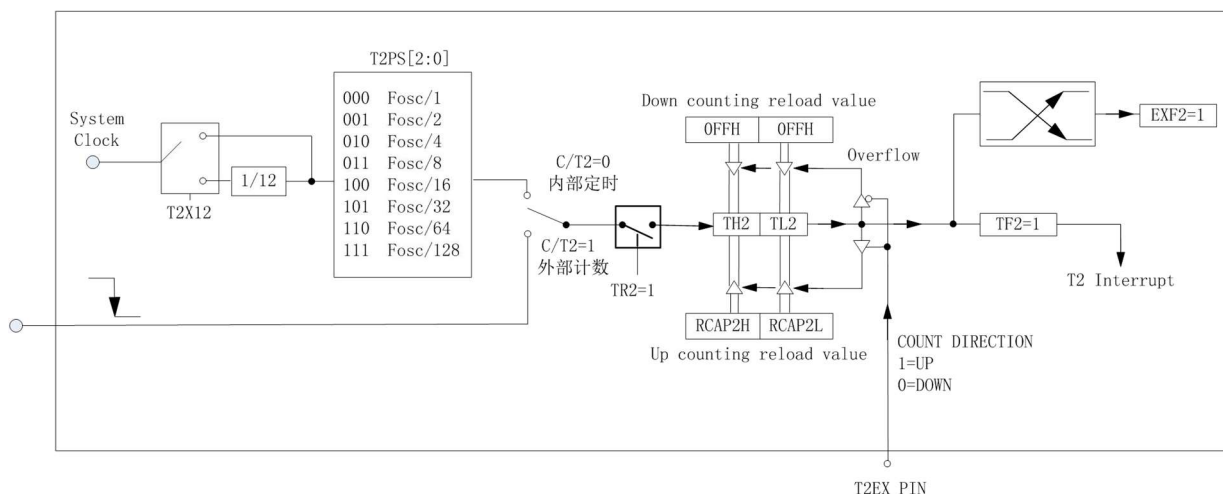
自动重载（向上或向下计数器）方式

自动重载计数方式受 DCEN 位，EXEN2 位和外部输入端 T2EX 边沿跳变影响。见下表可以查看具体控制。



定时器 T2 自动重载方式（DCEN=0）

DCEN	EXEN2	T2EX	MODE
0	0	X	默认向上计数至 0FFFFH 溢出，置位 TF2 且触发中断。同时把 RCAP2H、RCAP2L 值重载，该值软件预置。
	1	X/下降沿	T2 的 16 位重载由溢出或外部 T2EX 下降沿触发，该脉冲使 EXF2 置位，触发中断。
1	X	X	T2 向下计数至 TH2/TL2 等于 RCAP2H、RCAP2L 值时，计数溢出置位 TF2，同时将 0FFFFH 值重新装入定时寄存器 TH2/TL2 中。
	X	1	T2 向上计数 TH2/TL2 等于 0FFFF 值，计数溢出置位 TF2，同时将 RCAP2H、RCAP2L 重载到 TH2、TL2 中。

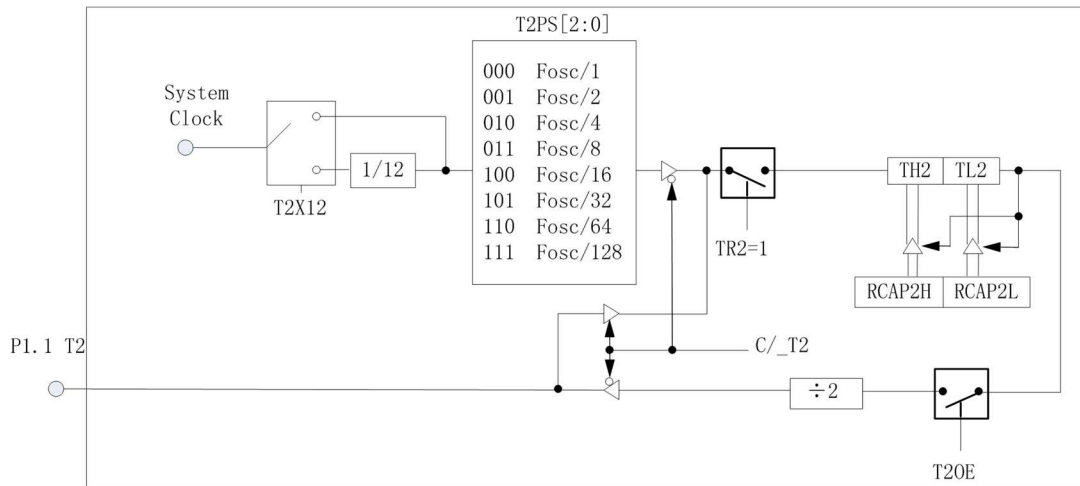


定时器 T2 自动重载方式（DCEN=1）

波特率发生器

当 RCLK 和 TCLK 置位时，则定时器 T2 工作于波特率发生器方式。如果定时器 T2 作为发送器

RCAP2L。



可编程时钟输出工作方式

注意：

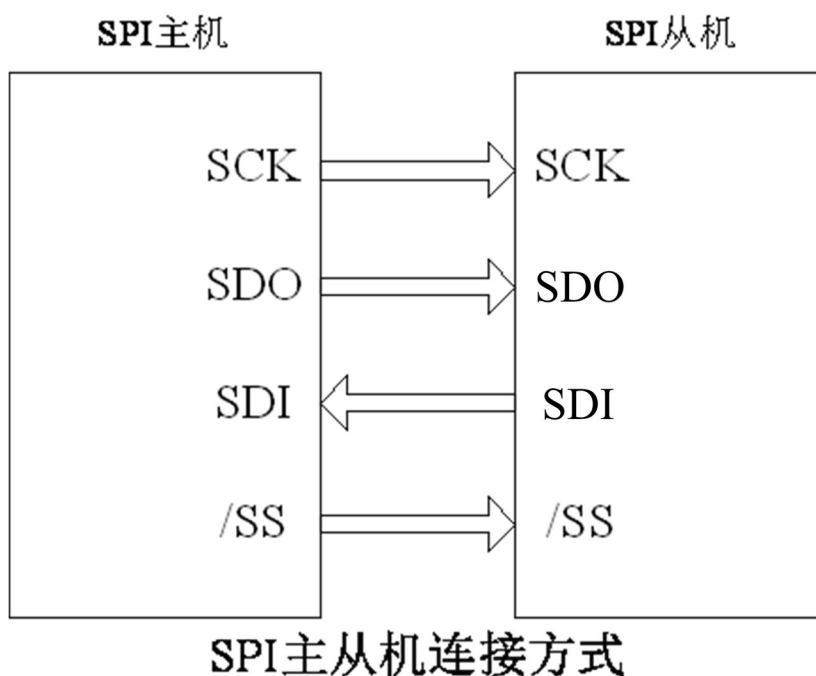
- (1) TF2 和 EXF2 都能触发定时器 2 的中断请求，两者有相同的向量地址。
- (2) 当事件发生时或其他任何时间都能有软件设置 TF2 和 EXF2 为 1，只有软件以及硬件复位才能使之清 0。
- (3) 当 EA=1 且 ET2=1 时，设置 TF2 或 EXF2 为 1 能引起定时器 2 中断。

7. SFR 专用功能模块

7.1 串行口外部设备接口 SPI

7.1.1 SPI（SPI 控制器）

SPI 接口是一个全双工串行数据传输器。SPI 接口的四线为：SDI、SDO、SCK 和/SS。SDI 和 SDO 是数据的输入和输出线。SCK 是串行时钟线，/SS 是从机的选择线。SPI 的接口引脚与普通 I/O 口。通过设定 SPIC0、SPIC1 寄存器的对应位，来使能 SPI 接口。连接到 SPI 接口的单片机以主/从模式进行通信，且主机完成所有的数据传输初始化，并控制时钟信号。由于单片机只有一个/SS 引脚，所以只能拥有一个从机设备。可通过软件控制/SS 引脚使能，设置 SSEN 位为“1”使能/SS 引脚功能，设置 SSEN 位为“0”/SS 引脚将作为普通输入/输出引脚。（管脚的输入输出需要对应设置一下管脚的方向。）



名称	描述
SDO/MOSI	主设备数据输出、从设备数据输入
SDI/MISO	主设备数据输入、从设备数据输出
SCLK	时钟信号，主设备产生
SCS	片选信号，主设备控制

SPI 功能具有以下特点：

- 全双工同步数据传输
- 主从模式

最低有效位先传或最高有效位先传的数据传输模式
传输完成标志位
时钟源上升沿或下降沿有效

7.1.2 SPIBUF（SPI 的缓冲器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
A7H(r/w)	SPIBUF	SPI 缓冲器							
reset	00H	0	0	0	0	0	0	0	0

SPI 的缓冲器，用以存放接收和发送的数据。在单片机尚未将数据写入到 SPI 总线中时，要传输的数据应先存在 SPIBUF 寄存器中。SPI 总线接收到数据之后，单片机就可以从 SPIBUF 寄存器中读取。所有通过 SPI 总线传输或接收的数据都必须通过 SPIBUF 寄存器实现。

7.1.3 SPIC0（SPI 控制器 0）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
ADH(r/w)	SPIC0	SPIM2	SPIM1	SPIM0	-	-	-	SPIEN	-
reset	00H	0	0	0	0	0	0	0	0

SPIM2~SPIM0：SPI 工作模式控制位

- 000：SPI 主机模式；SPI 时钟为 Fsys/4
- 001：SPI 主机模式；SPI 时钟为 Fsys/16
- 010：SPI 主机模式；SPI 时钟为 Fsys/64
- 011：SPI 主机模式；SPI 时钟为 ILRC=32KHz
- 100：SPI 主机模式；SPI 时钟为 TC0/2(模式2_8位自动重装载)
- 101：SPI 从机模式；时钟源从外部主机而得
- 110：未使用模式
- 111：未使用模式

这几位用于设置 SPI 功能的工作模式，用于选择 SPI 的主从模式和 SPI 的主机时钟频率。SPI 时钟源可来自于系统时钟也可以选择来自 TMR0。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

SPIEN：SPI 控制位。

- 0：无效
- 1：有效

此位为 SIM 接口的开/关控制位。选择 SPI 模式时，此位为“0”时，SPI 接口除能，SDI、SDO、SCK 和/SS 脚用于输入/输出功能，SPI 工作电流减小到最小值。此位为“1”时，SPI 接口使能。配置选项中首先将 SPI 接口使能才能使此位有效。若 SPI 经由 SIMM2~SIMM0位设置为工作在 SPI 接口，

当 SIMEN 位由低到高转变时，SPI 控制寄存器中的设置不会发生变化，其首先应在应用程序中初始化。

7.1.4 SPIC1 (SIM 控制器 1)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
AEH(r/w)	SPIC1			CKP	CKEG	MLS	SSEN	-	TRF/SPIF
reset	00H	0	0	0	0	0	0	0	0

CKEG: SPI 的 SCK 有效时钟边沿类型位 (时钟的相位)

CKP=0 时:

- 0: SCK 为高电平且在 SCK 上升沿抓取数据
- 1: SCK 为高电平且在 SCK 下降沿抓取数据

CKP=1 时:

- 0: SCK 为低电平且在 SCK 下降沿抓取数据
- 1: SCK 为低电平且在 SCK 上升沿抓取数据

CKEG 和 CKP 位用于设置 SPI 总线上时钟信号输入和输出方式。在执行数据传输前，这两位必须被设置，否则将产生错误的时钟边沿信号。CKP 位决定时钟线的基本状态，若时钟无效且此位为高，则 SCK 为低电平，若时钟无效且此位为低，则 SCK 为高电平。CKEG 位决定有效时钟边沿类型，取决于 CKP 的状态。

CKP: 时钟线的基础状态位 (时钟的极性)

- 0: 当时钟无效时，SCK 引脚为高电平
- 1: 当时钟无效时，SCK 引脚为低电平

此位决定了时钟线的基础状态，当时钟无效时，若此位为高，SCK 为低电平，若此位为低，SCK 为高电平。

CKP (极性)	CKEG (相位)	描述
0	0	SCK 空闲时为高电平，且在 SCK 上升沿采样数据
0	1	SCK 空闲时为高电平，且在 SCK 下降沿采样数据
1	0	SCK 空闲时为低电平，且在 SCK 下降沿采样数据
1	1	SCK 空闲时为低电平，且在 SCK 上升沿采样数据

MLS: SPI 数据移位选择位

- 0: LSB 低位优先传输
- 1: MSB 高位优先传输

SSEN: SPI /SS 引脚控制位

- 0: 禁止，/SS 处于悬空状态
- 1: 使能，/SS 作为选择脚

模式	SSEN	描述
主机	0	/SS 引脚作为 I/O 口

主机	1	在 SPIEN=1 时, /SS 变低
从机	0	SPI 模块不受 /SS 控制, 都可以接受数据
从机	1	/SS 为低时, SPI 才会接收数据

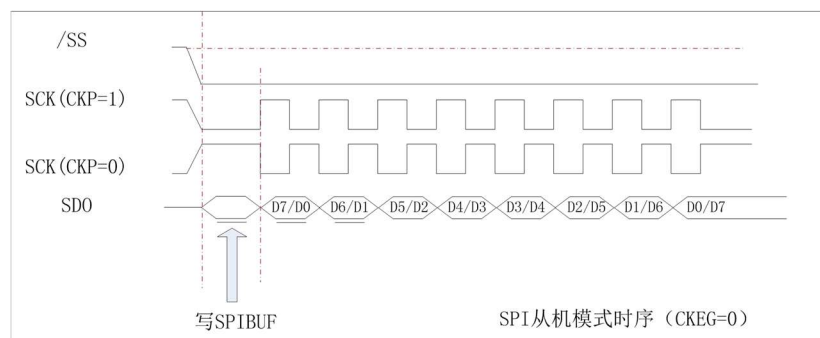
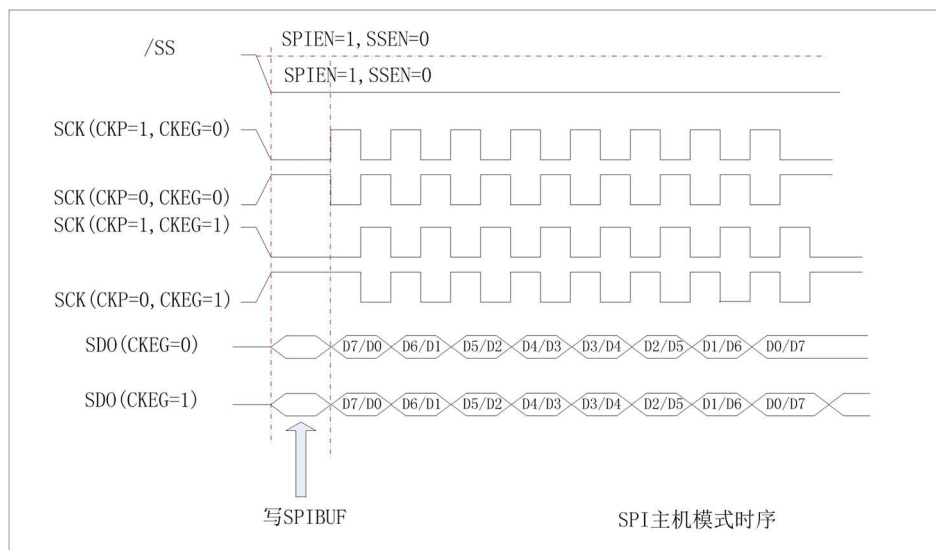
TRF: SPI 发送/接收数据结束标志位

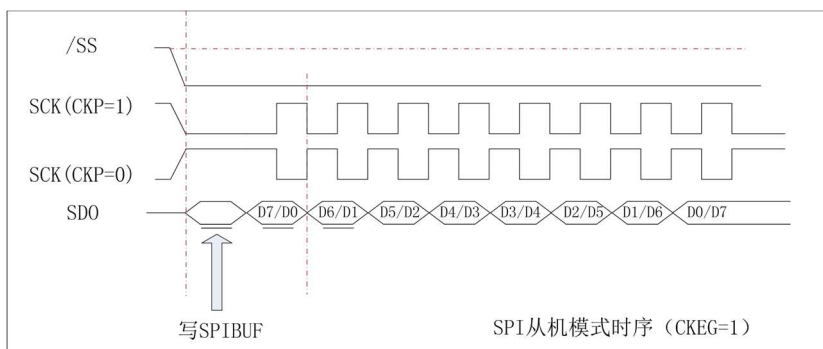
0: 正在发送

1: 发送结束

7.1.5 SPI 通信

将 SPIEN 设置为高, 使能 SPI 功能之后, 单片机处于主机模式, 当数据写入到寄存器 SIMBUF 的同时传输/接收开始进行。数据传输完成时, TRF 位将自动被置位但清除只能通过应用程序完成。单片机处于从机模式时, 收到主机发来的信号之后, 会传输 SIMBUF 中的数据, 而且在 SDI 引脚上的数据也会被移位到 SIMBUF 寄存器中。主机应在输出时钟信号之前先输出一个 /SS 信号以使能从机, 从机的数据传输功能也应在与 /SS 信号相关的适当时候准备就绪, 这由 CKP 和 CKEG 位决定。所附时序图表明了 CKP 和 CKEG 位各种设置情况下从机数据与 /SS 信号的关系。





注：使用 SPI 前，应先定义 CKEG,CKP，然后才使能 SPIEN 信号。

7.2 增强型通用异步收发器 (EUART)

增强型通用异步收发器 EUART 自带一个 15 位向上计数的波特率发生器，其增强功能包括帧出错检测及自动地址识别，EUART 有四种工作方式。

SCON

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
98H(r/w)	SCON	SM0 /FE	SM1 /RXOV	SM2 /TXCOL	REN	TB8	RB8	TI	RI
reset	00H	0	0	0	0	0	0	0	0

SM0, SM1: 串行口工作方式选择位

- 00: 方式 0 同步移位寄存器方式（用于扩展 I/O 口）
- 01: 方式 1 8 位异步收发，波特率可变（由定时器控制）
- 10: 方式 2 9 位异步收发，波特率位 $f_{osc}/64$ 或 $f_{osc}/32$
- 11: 方式 3 9 位异步收发，波特率可变（由定时器控制）

FE: UART 帧出错标志位，当 FE 位被读时，SSTAT 位必须被置位

- 0: 无错，软件清 0。
- 1: 帧出错，由硬件置位

RXOV: UART 接收溢出标志位，当 RXOV 位被读时，SSTAT 位必须被置位

- 0: 无溢出，软件清 0。
- 1: 接收溢出，由硬件置位

SM2: 多机通信控制位（方式 2 和 3）

- 0: 不论第 9 位数据是 0 还是 1，都将前 8 位数据送入 SBUF，并使 RI 置 1，产生中断请求。
- 1: 只有当第 9 位数据（RB8）为 1 时，才将接收到的前 8 位数据送入 SBUF，并使 RI 置 1，产生中断请求；当接收到第 9 位数据为 0 时，则将前 8 位数据丢弃。

在方式 1，如果 SM2=1，则只有收到有效的停止位时，才会激活 RI。如果 SM2=0，收到低电平停止位，也会激活 RI。

在方式 0，SM2 必须为 0。

TXCOL: UART 发送冲突标志位，当 TXCOL 位被读时，SSTAT 位必须被置位

- 0: 无发送冲突，软件清 0。
- 1: 发送冲突，由硬件置位

REN:

- 0: 禁止串行口接收数据
- 1: 允许串行口接收数据

TB8: 发送位的第 9 位数据

方式 2 和 3 时，TB8 由软件设置为 1 或 0；在双机通信时，TB8 作为奇偶校验位使用；多机通信中，用来表示主机发送时地址帧还是数据帧，TB8=1 为地址帧，TB8=0 为数据帧。

RB8: 接收的第 9 位数据

方式 2 和 3 时，RB8 存放接收到的第 9 位数据；在方式 1，如果 SM2=0，RB8 时接收到的停止位；在方式 0，不使用 RB8。

TI: 发送中断标志位

方式 0 时，串行发送第 8 位数据结束时由硬件置 1；在其他方式中，串行口发送停止位的开始时置 1。TI=1，表示一帧数据发送结束，可供软件查询，也可申请中断。在中断程序中，向 SBUF 写入要发送的下一帧数据。TI 必须由软件清 0。

RI: 接收中断标志位

方式 0 时，接受完第 8 位数据时，RI 由硬件置 1。在其他工作方式中，串行口接收到停止位时，RI 置 1。RI=1，表示一帧数据接收完毕，并申请中断，要求 CPU 从接收 SBUF 取走数据。RI 的状态可供软件查询，RI 必须由软件清 0。

SBUF

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
99H(r/w)	SBUF	SBUF<7:0>							
reset	00H	0	0	0	0	0	0	0	0

物理独立的接收、发送缓冲器 SBUF（属于特殊功能寄存器），可同时发送、接收数据。发送缓冲器只能写入不能读出，接收缓冲器只能读出不能写入，两个缓冲器共用一个特殊功能寄存器字节地址（99H）。

SADDR、SADREN

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
9AH(r/w)	SADDR	SADDR<7:0>							
9BH(r/w)	SADREN	SADREN<7:0>							
reset	00H	0	0	0	0	0	0	0	0

SADDR<7:0>：定义 EUART 的从机地址。

SADREN<7:0>：位屏蔽寄存器，决定检验 SADDR 的哪些位对应接收地址

- 0: 在 SADDR 中的相应位被忽略
- 1: SADDR 中相应位被检验时是否是对应接收地址

7.2.1 工作方式

本芯片 EUART 有 3 中工作方式，通信之前需要先初始化 SCON，选择方式和波特率。如果使用方式 1 或 3，应先初始化定时器 T1 或 T2。

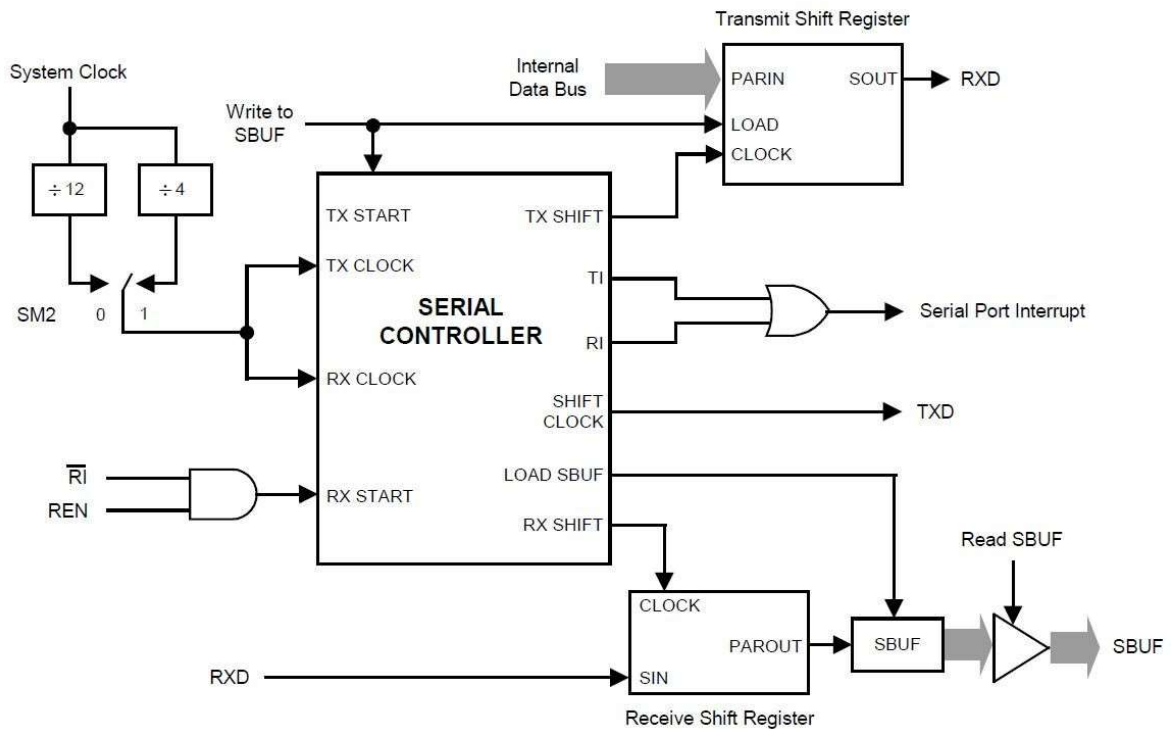
三种方式工作时，任何将 SBUF 作为目标寄存器写操作都会启动发送。由起始位初始化接收（如果 REN=1），通过发送起始位，外部发送器开始通信。

SM0	SM1	方式	类型	波特率	帧长度	起始位	停止位	第 9 位
0	0	0	同步	SYSCLK/4(12)	8 位	无	无	无
0	1	1	异步	T1 或 T2 溢出率/（16 或 32）	10 位	1	1	无
1	0	2	异步	SYSCLK/（32 或 64）	11 位	1	1	0,1
1	1	3	异步	T1 或 T2 溢出率/（16 或 32）	11 位	1	1	0,1

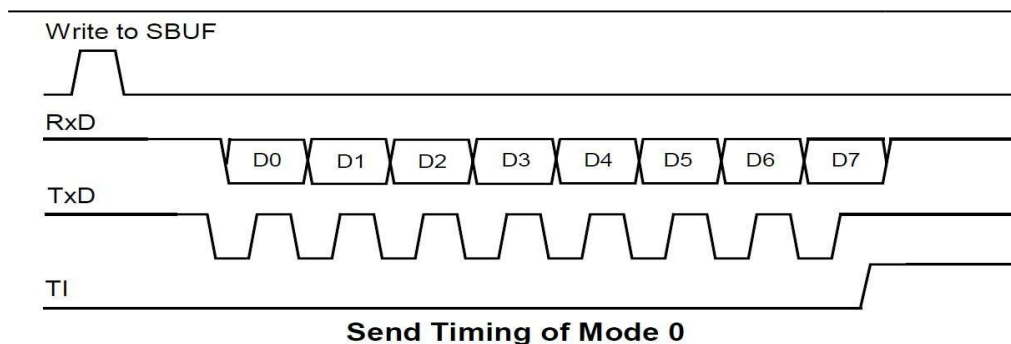
方式 0: 8 位 EUART，同步半双工

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

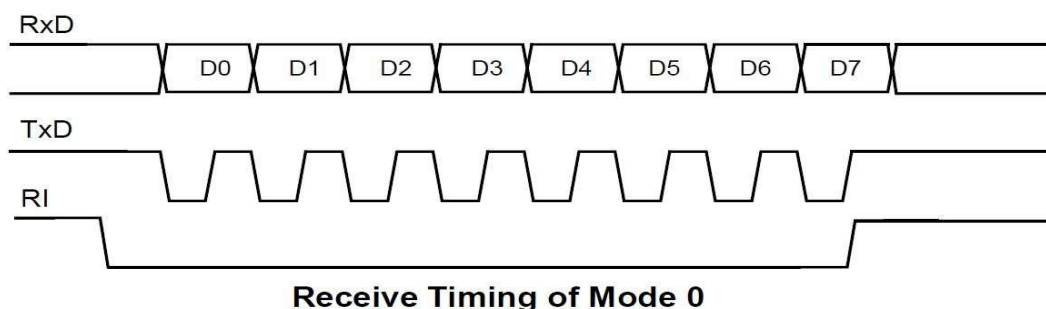
通过置 SM2 位（SCON.5）为 0 或 1，波特率固定为系统时钟的 1/12 或 1/4。当 SM2 位为 0 时，串行端口以系统时钟的 1/12 运行。当置 1 时，串行端口以系统时钟的 1/4 运行。与标准 8051 唯一不同的是，MCU_CORE 在方式 0 中有可变波特率。功能块框图如下图所示。数据通过 RXD 引脚进入和移出串行端口。移位时钟由 TXD 引脚输出。



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元时钟的下降沿，移位寄存器的内容逐次从左往右移位，空位置 0。当移位寄存器中的所有 8 位都发送后，Tx 控制模块停止发送操作，然后在下一个系统时钟的上升沿将 TI 置 1（SCON.1）。



REN (SCON.4) 置 1 和 RI (SCON.0) 清 0 初始化接收。下一个系统时钟启动接收，在移位元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元。当所有 8 位都接收到接收移位寄存器中后，Rx 控制块停止接收，然后在下一个系统时钟的上升沿上 RI 置 1，直到被软件清 0 才允许接收。

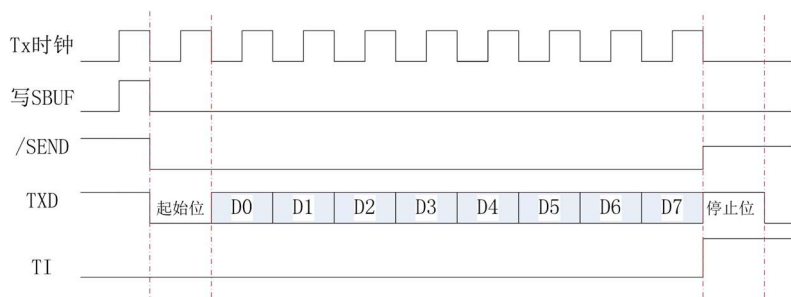


方式 1: 8 位 EUART, 可变波特率, 异步全双工

方式 1 提供 10 位全双工异步通信，10 位由一个起始位（逻辑 0）、8 个数据位（低位为第一位）、一个终止位（逻辑 1）组成。在接收时，这 8 个数据位存储在 SBUF 中，停止位存在 RB8 中。波特率可被设置为 $T1/16$ 、 $T1/32$ 或者 $T2/16$ 。

(1) 方式 1 发送

任何对 SBUF 作为目标寄存器的写操作都会启动发送，TX 的时钟频率就是波特率。发送开始时，内部控制信号 /SEND 变为低有效，将起始位传递到 TXD 脚输出，此后每经过一个 TX 时钟周期，便产生一个移位脉冲，并由 TXD 引脚输出一个数据位。8 位数据位全部发送完毕，中断标志位 TI 置 1，然后 /SEND 失效。



方式 1 发送时序

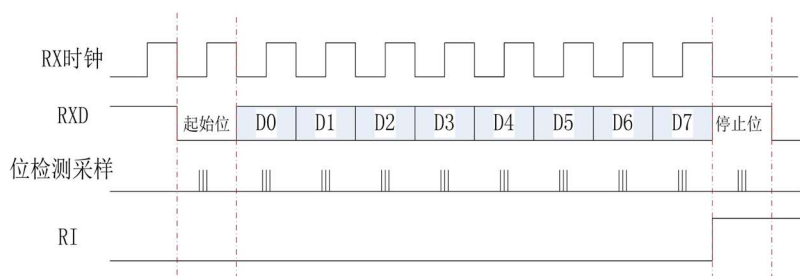
(2) 方式 1 接收

只有 REN=1 时才允许接收。当 RxD 引脚检测到下降沿时串口开始接收串行数据。接收时，定时控制信号有两种。一种是接收移位时钟（RX 时钟），它的频率和传送的波特率相同。另一种是位检测器采样脉冲，采样频率是 RX 时钟 16 倍。也就是在一位数据期间有 16 个采样脉冲，来采样 RxD

脚状态。当采样到 RXD 有从 1 到 0 的跳变时就启动检测器，接收的值时 3 次连续采样（7、8、9 脉冲时采样）取其中 2 次值相同，以确认是否是真正的起始位。如果不是则被忽略，接收电路被复位，等待下一个下降沿。如果是起始位，则移入移位寄存器，并开始接收其他位数据。同样对每一位数据也采用 3 次采样的方式。8 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，RI 置 1，但必须满足下面两个条件：

- a. RI=0
- b. SM2=0，或者接收的停止位=1

如果条件都满足，那么停止位移入 RB8，8 个数据位移入 SBUF，RI 被置 1。否则接收的帧丢失，这是接收器将重新检测 RXD 端是否另一个下降沿。用户必须软件清除 RI 才能再次接收。



方式 1 接收时序

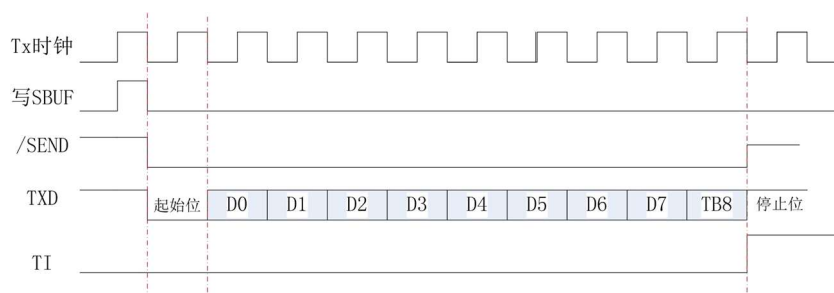
方式 2: 9 位 UART，固定波特率，异步全双工

方式 2 使用异步全双工通信，一帧包含一个起始位（逻辑 0），8 个数据位（低位为第一位），一个可编程的第 9 位和一个停止位（逻辑 1）组成。方式 2 支持多机通信和硬件地址识别。PCON 中的 SMOD 位选择波特率位系统工作频率（F_{cpu}）的 1/32 或 1/64。

$$\text{方式2波特率} = \frac{2^{SMOD}}{64} \times f_{osc}$$

（1）方式 2 发送

发送前，根据通信协议软件设置 TB8（如双机通信时的奇偶校验位或多机通信时的地址/数据的标志位）。然后将要发送的数据写入到 SBUF，即可启动发送过程。串行口能自动提取 RB8 装入到第 9 位数据位的位置，在逐一发送出去。发送完成后，将 TI 置 1。



方式 2 发送时序

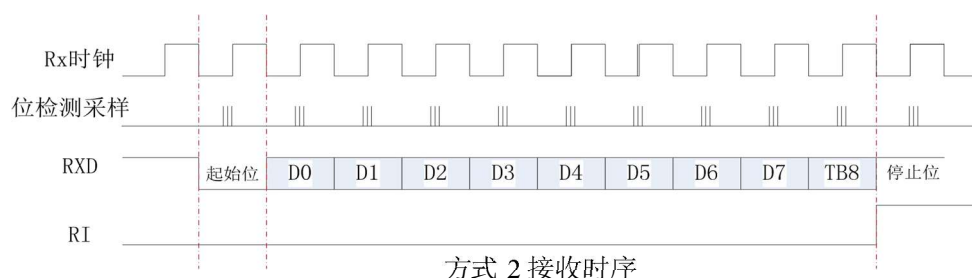
（2）方式 2 接收

只有在 REN=1 时才允许接收数据。接收时，数据由 RXD 脚输入接收 11 位信息。当位检测逻辑采样到 RXD 引脚从 1 到 0 的负跳变，并判断起始位有效后，便开始接收一帧信息。在接收完第 9 位数据后，需满足一下两个条件，才能将接收到的数据送入 SBUF。

- a. RI=0，意味着接收器缓冲器为空
- b. SM2=0，或者接收到第 9 位数据位 RB8=1 时

当满足上述两个条件时，接收到的数据送入 SBUF，第 9 位数据送入 RB8，且 RI 置 1。否则，

数据被丢弃。



方式 3: 9 位 EUART, 可变波特率, 异步全双工

方式 3 的传输协议与方式 2 相同, 而波特率与方式 1 产生方法相同。

7.2.2 波特率

串行口每秒钟发送(或接收)的位数即为波特率, 设发送一位所需时间为 T , 则波特率位 $1/T$ 。对于定时器的不同工作方式, 得到波特率的范围也不一样, 这是因为定时器计数位数不同决定的。

在方式 1 和方式 3 中, 波特率可选择来自定时器 T1 或 T2 的溢出率。分别置 TCLK 和 RCLK 位为 1 来选择定时器 T2 作为 TX 和 RX 的波特时钟源。无论 TCLK 还是 RCLK 为逻辑 1, T2 都为比特率发生器方式。如果 TCLK 和 RCLK 为逻辑 0, 定时器 T1 作为 TX 和 RX 的波特时钟源

$$BaudRate = \frac{2^{SMOD}}{32} \times \frac{f_{T1}}{256 - TH1}, \text{ 用工作在方式 2 的定时器 1 作为波特率发生器。}$$

$$BaudRate = \frac{1}{16} \times \frac{f_{T2}}{65536 - [RCAP2H, RCAP2L]}, \text{ 用定时器 2 作为波特率发生器, 定时器 T2}$$

时钟源为系统时钟。

$$BaudRate = \frac{1}{16} \times \frac{f_{T2}}{65536 - [RCAP2H, RCAP2L]}, \text{ 用定时器 2 作为波特率发生器, 定时器 2}$$

时钟源为 T2 引脚输入时钟。

在方式 2 中, 波特率固定为系统工作时钟 (F_{cpu}) 的 $1/32$ 或 $1/64$, 由 SMOD 为决定。当 SMOD 位为 0 时, EUART 以系统时钟的 $1/64$ 运行, 当 SMOD 位为 1 时, EUART 以系统时钟的 $1/32$ 运行。

7.2.3 多机通信

软件地址识别

方式 2 和方式有一个专门适用于多机通讯的功能, 接收的是 9 位数据, 第 9 位移入 RB8 中, 然后再来一个停止位。当接收到停止位时, 只有在 RB8=1 的条件下, 串行口中断才会有效 (RI 标志置 1)。可以通过将 SCON 寄存器的 SM2 位置 1, 时 EUART 具有该功能。

在多机通讯系统中, 当主机要发送数据块给几个从机中的一个时, 它先送出一地址字节, 以辨认目标地址。地址字节和数据字节时通过第 9 位 TB8 来区别的。TB8=1 为地址帧, TB8=0 为数据帧。

如果从机 SM2=1, 则不会响应数据字节中断。地址字节可以是所有从机产生中断, 每一个从机都检查所收到的地址字节, 以判别本机是不是目标从机。被寻到的从机对 SM2 为清 0, 并准备接受数据字节。当接收完毕后, 从机再次将 SM2 置 1。没有被寻址的从机, 则保持 SM2 位为 1, 不响应数据字节。

多机通信时, 一般主机第一帧的 TB8=1, 后面帧 TB8=0。这样表示第一帧是地址, 后面是数据

帧。从机开始 SM2=1，第一帧接收到地址匹配后，SM2=0，否则 SM2=1。这样地址匹配的从机能够

接收到后面的数据，不匹配的接收不到后面的数据。

自动（硬件）地址识别

在方式 2 和方式 3 中，SM2=1，EUART 运行状态如下：RB8 的第 9 位为 1（地址字节），且接收到的数据字节符合 EUART 的从机地址，EUART 产生一个中断。从机将 SM2 清 0 后，接收后续数据字节。

第 9 位为 1 表明该字节是地址而不是数据。当主机要发送一组数据给多个从机中的一个时，必须先发送目标从机地址。所有从机等待接收地址字节，为了确保仅在接收地址字节时产生中断，SM2 位必须=1。自动地址识别的特点是只有地址匹配的从机才能产生中断，硬件完成地址比较。

中断产生后，地址匹配的从机将 SM2=0，继续接收数据字节。地址不匹配的从机不受影响，将继续等待接收和它匹配的地址字节。全部信息接收完毕后，地址匹配的从机应再次把 SM2 置位，忽略所有传送的非地址字节，指导接收到下一个地址字节。

使用自动地址识别功能时，需要使用两个特殊寄存器，从机地址（SADDR）和地址屏蔽（SADEN）。从机地址是一个 8 位字节，存于 SADDR 寄存器中。SADEN 用于定义 SADDR 各位是否有效，如果 SADEN 中某一位为 0，则 SADDR 中相应位被忽略，如果 SADEN 中某一位为 1，则 SADDR 中相应位将用于产生约定地址。这可以使用户在不改变 SADDR 寄存器中的从机地址的情况下灵活地寻址多个从机。

	从机 1	从机 2
SADDR	10100100	10100111
SADEN（为 0 的位被忽略）	11111010	11111001
约定地址	10100x0x	10100xx1
广播地址（SADDR 或 SADEN）	1111111x	11111111

主机可以通过广播地址与所有从机同时通讯，这个地址等于 SADDR 和 SADEN 的位或，结果中的 0 表示该位被忽略。多数情况下，广播地址位 0xFFh 该地址可被所有从机应答。系统复位后，SADDR 和 SADEN 两个寄存器初始化为 0。

7.2.4 帧出错检测

当寄存器 PCON 中的 SSTAT 位为逻辑 1 时，帧出错检测功能才有效。3 个错误标志位被置位后，只能软件清 0，尽管后续接收的帧没有任何错误也不会自动清 0。

发送冲突：

如果在一个发送正在进行时，用户软件写数据到 SBUF 寄存器时，发送冲突位（SCON 寄存器中的 TXCOL 位）置位。如果发生冲突，新数据会被忽略，不能被写入发送缓冲器。

接收溢出：

如果在接收缓冲器中的数据未被读取之前，RI 清 0，又有新的数据存入接收缓冲器，那么接收溢出位（SCON 中的 RXOV 位）置位。如果发生了接收溢出，接收缓冲器中原来的数据将丢失。

帧出错：

如果检测到一个无效（低）停止位，那么帧出错位（SCON 中的 FE）置位。

注意：在发送之前 TXD 引脚必须被设置为输出高电平。

7.3 增强型通用异步收发器（EUART1）

增强型通用异步收发器 EUART1 自带一个 15 位向上计数的波特率发生器，其增强功能包括帧出错检测及自动地址识别，EUART 有四种工作方式。

SCON1

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
F8H(r/w)	SCON1	SM0_1 /FE_1	SM1_1 /RXOV_1	SM2_1 /TXCOL_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
reset	00H	0	0	0	0	0	0	0	0

SM0_1, SM1_1: 串行口工作方式选择位

见下面表说明

FE_1: UART 帧出错标志位，当 FE_1 位被读时，SSTAT1 位必须被置位

0: 无错，软件清 0。

1: 帧出错，由硬件置位

RXOV_1: UART 接收溢出标志位，当 RXOV_1 位被读时，SSTAT1 位必须被置位

0: 无溢出，软件清 0。

1: 接收溢出，由硬件置位

SM2_1: 多机通信控制位（方式 2 和 3）

0: 不论第 9 位数据是 0 还是 1，都将前 8 位数据送入 SBUF1，并使 RI_1 置 1，产生中断请求。

1: 只有当第 9 位数据（RB8_1）为 1 时，才将接收到的前 8 位数据送入 SBUF1，并使 RI_1 置 1，产生中断请求；当接收到第 9 位数据为 0 时，则将前 8 位数据丢弃。

在方式 1，如果 SM2_1=1，则只有收到有效的停止位时，才会激活 RI_1。如果 SM2_1=0，收到低电平停止位，也会激活 RI_1。

在方式 0，SM2_1 必须为 0。

TXCOL_1: UART 发送冲突标志位，当 TXCOL_1 位被读时，SSTAT1 位必须被置位

0: 无发送冲突，软件清 0。

1: 发送冲突，由硬件置位

REN_1:

0: 禁止串行口接收数据

1: 允许串行口接收数据

TB8_1: 发送位的第 9 位数据

方式 2 和 3 时，TB8_1 由软件设置为 1 或 0；在双机通信时，TB8_1 作为奇偶校验位使用；多机通信中，用来表示主机发送时地址帧还是数据帧，TB8_1=1 为地址帧，TB8=0 为数据帧。

RB8_1: 接收的第 9 位数据

方式 2 和 3 时，RB8_1 存放接收到的第 9 位数据；在方式 1，如果 SM2_1=0，RB8_1 时接收到的停止位；在方式 0，不使用 RB8_1。

TI_1: 发送中断标志位

方式 0 时，串行发送第 8 位数据结束时由硬件置 1；在其他方式中，串行口发送停止位的

开始时置 1。TI_1=1，表示一帧数据发送结束，可供软件查询，也可申请中断。在中断程序中，向 SBUF1 写入要发送的下一帧数据。TI_1 必须由软件清 0。

RI_1：接收中断标志位

方式 0 时，接受完第 8 位数据时，RI_1 由硬件置 1。在其他工作方式中，串行口接收到停止位时，RI_1 置 1。RI_1=1，表示一帧数据接收完毕，并申请中断，要求 CPU 从接收 SBUF1 取走数据。RI_1 的状态可供软件查询，RI_1 必须由软件清 0。

SBUF1

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
F9H(r/w)	SBUF1	SBUF1<7:0>							
reset	00H	0	0	0	0	0	0	0	0

物理独立的接收、发送缓冲器 SBUF1（属于特殊功能寄存器），可同时发送、接收数据。发送缓冲器只能写入不能读出，接收缓冲器只能读出不能写入，两个缓冲器共用一个特殊功能寄存器字节地址。

SADDR、SADREN

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FAH(r/w)	SADDR1	SADDR1<7:0>							
FBH(r/w)	SADREN1	SADREN1<7:0>							
reset	00H	0	0	0	0	0	0	0	0

SADDR1<7:0>：定义 EUART1 的从机地址。

SADREN1<7:0>：位屏蔽寄存器，决定检验 SADDR1 的哪些位对应接收地址

0：在 SADDR1 中的相应位被忽略

1：SADDR1 中相应位被检验时是否是对应接收地址

7.3.1 工作方式

本芯片 EUART1 有 3 中工作方式，通信之前需要先初始化 SCON1，选择方式和波特率。如果使用方式 1 或 3，应先初始化波特率发生器。

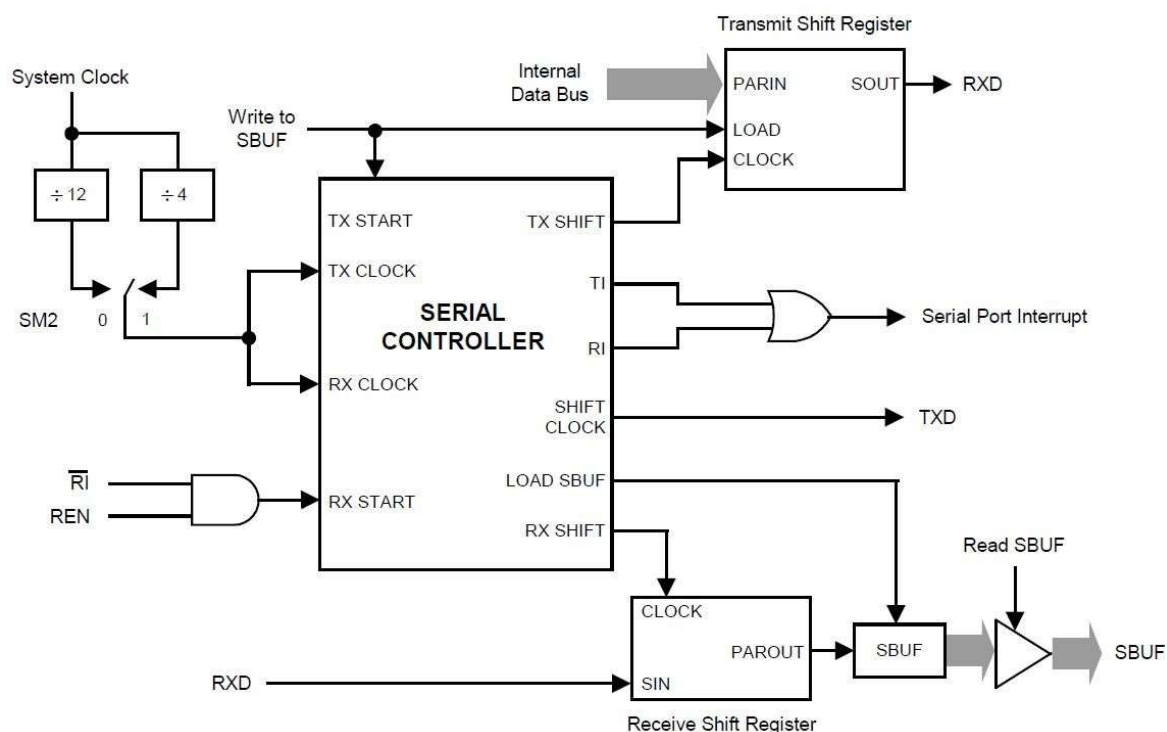
三种方式工作时，任何将 SBUF1 作为目标寄存器写操作都会启动发送。由起始位初始化接收（如果 REN=1），通过发送起始位，外部发送器开始通信。

SM0	SM1	方式	类型	波特率	帧长度	起始位	停止位	第 9 位
0	0	0	同步	SYSCLK/4(12)	8 位	无	无	无
0	1	1	异步	自带波特率溢出率/（16 或 32）	10 位	1	1	无
1	0	2	异步	SYSCLK/（32 或 64）	11 位	1	1	0,1
1	1	3	异步	自带波特率溢出率/（16 或 32）	11 位	1	1	0,1

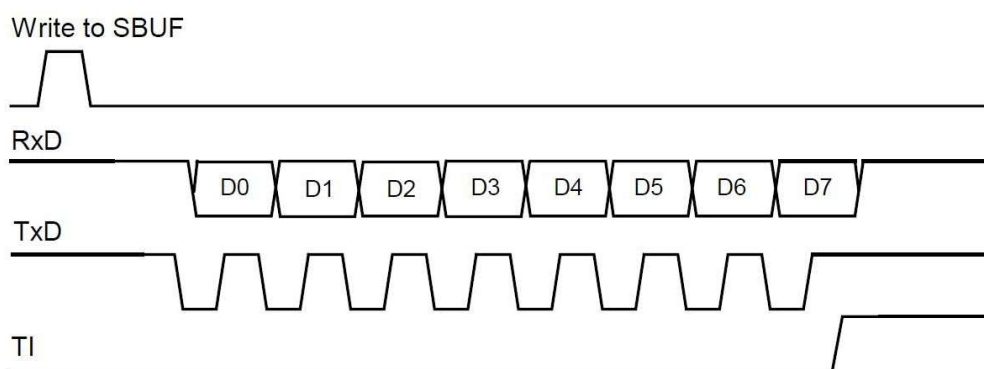
方式 0: 8 位 EUART，同步半双工

方式 0 支持与外部设备的同步通信。在 RXD1 引脚上收发串行数据。TXD1 引脚用作发送移位时钟。MCU_CORE 提供 TXD1 引脚上的移位时钟。因此这个方式是串行通信的半双工方式。在这个方式中，每帧收发 8 位，低位先接收或发送。

通过置 SM2_1 位 (SCON1.5) 为 0 或 1, 波特率固定为系统时钟的 1/12 或 1/4。当 SM2_1 位为 0 时, 串行端口以系统时钟的 1/12 运行。当置 1 时, 串行端口以系统时钟的 1/4 运行。功能块框图如下图所示。数据通过 RXD1 引脚进入和移出串行端口。移位时钟由 TXD 引脚输出。

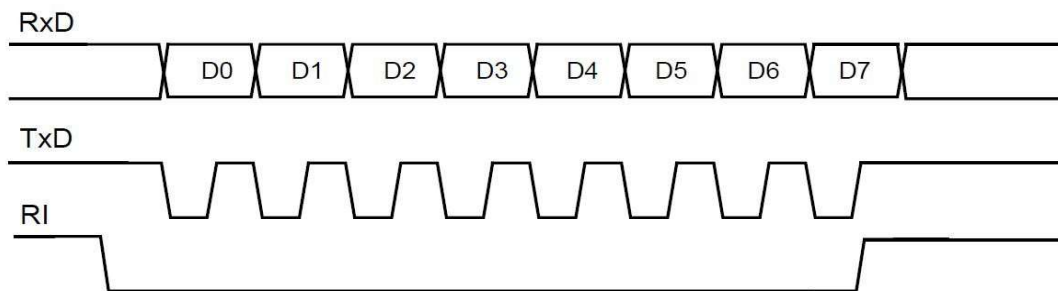


任何将 SBUF1 作为目标寄存器的写操作都会启动发送。下一个系统时钟 Tx 控制块开始发送。数据转换发生在移位元时钟的下降沿, 移位寄存器的内容逐次从左往右移位, 空位置 0。当移位寄存器中的所有 8 位都发送后, Tx 控制模块停止发送操作, 然后在下一个系统时钟的上升沿将 TI_1 置 1 (SCON1.1)。



Send Timing of Mode 0

REN_1 (SCON1.4) 置 1 和 RI_1 (SCON1.0) 清 0 初始化接收。下一个系统时钟启动接收, 在移位元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元。当所有 8 位都接收到接收移位寄存器中后, Rx 控制块停止接收, 然后在下一个系统时钟的上升沿上 RI_1 置 1, 直到被软件清 0 才允许接收。



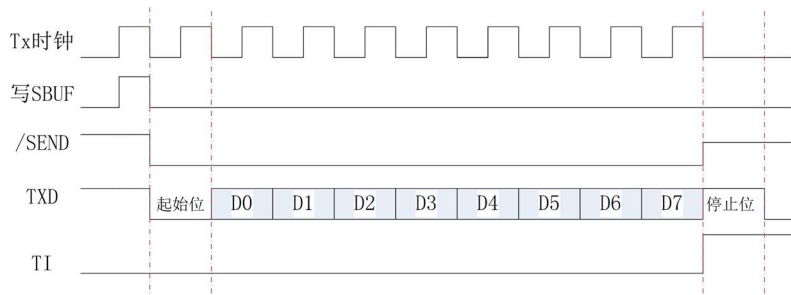
Receive Timing of Mode 0

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 (逻辑 0)、8 个数据位 (低位为第一位)、一个终止位 (逻辑 1) 组成。在接收时, 这 8 个数据位存储在 SBUF1 中, 停止位存在 RB8_1 中。

(1) 方式 1 发送

任何对 SBUF1 作为目标寄存器的写操作都会启动发送, TX 的时钟频率就是波特率。发送开始时, 内部控制信号/SEND 变为低有效, 将起始位传递到 TXD1 脚输出, 此后每经过一个 TX 时钟周期, 便产生一个移位脉冲, 并由 TXD1 引脚输出一个数据位。8 位数据位全部发送完毕, 中断标志位 TI 置 1, 然后/SEND 失效。



方式 1 发送时序

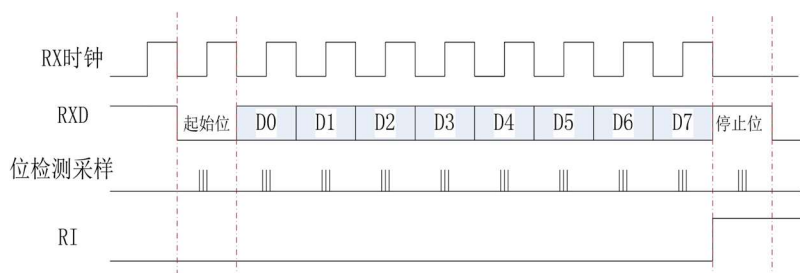
(2) 方式 1 接收

只有 REN_1=1 时才允许接收。当 RXD1 引脚检测到下降沿时串口开始接收串行数据。接收时, 定时控制信号有两种。一种是接收移位时钟 (RX 时钟), 它的频率和传送的波特率相同。另一种是位检测器采样脉冲, 采样频率是 RX 时钟 16 倍。也就是在一位数据期间有 16 个采样脉冲, 来采样 RXD1 脚状态。当采样到 RXD1 有从 1 到 0 的跳变时就启动检测器, 接收的值时 3 次连续采样 (7、8、9 脉冲时采样) 取其中 2 次值相同, 以确认是否是真正的起始位。如果不是则被忽略, 接收电路被复位, 等待下一个下降沿。如果是起始位, 则移入移位寄存器, 并开始接收其他位数据。同样对每一位数据也采用 3 次采样的方式。8 个数据位和 1 个停止位移入之后, 移位寄存器的内容被分别装入 SBUF1 和 RB8_1 中, RI_1 置 1, 但必须满足下面两个条件:

c. RI_1=0

d. SM2_1=0, 或者接收的停止位=1

如果条件都满足, 那么停止位装入 RB8_1, 8 个数据位装入 SBUF1, RI_1 被置 1。否则接收的帧丢失, 这是接收器将重新检测 RXD1 端是否另一个下降沿。用户必须软件清除 RI_1 才能再次接收。



方式 1 接收时序

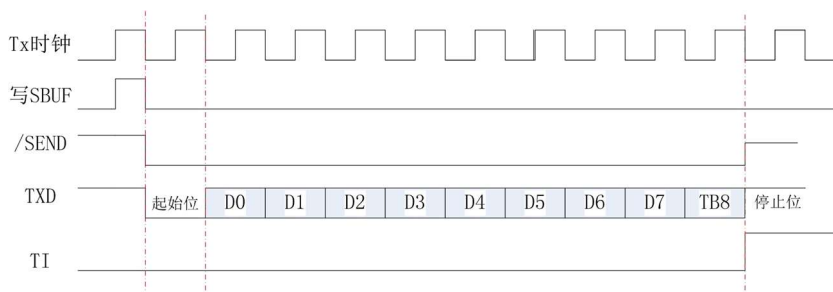
方式 2: 9 位 EUART, 固定波特率, 异步全双工

方式 2 使用异步全双工通信, 一帧包含一个起始位 (逻辑 0), 8 个数据位 (低位为第一位), 一个可编程的第 9 位和一个停止位 (逻辑 1) 组成。方式 2 支持多机通信和硬件地址识别。CHIPCON 中的 SMOD1 位选择波特率位系统工作频率 (Fcpu) 的 1/32 或 1/64。

$$\text{方式2波特率} = \frac{2^{SMOD}}{64} \times f_{osc}$$

(1) 方式 2 发送

发送前, 根据通信协议软件设置 TB8_1 (如双机通信时的奇偶校验位或多机通信时的地址/数据的标志位)。然后将要发送的数据写入到 SBUF1, 即可启动发送过程。串行口能自动提取 RB8_1 装入到第 9 位数据位的位置, 在逐一发送出去。发送完成后, 将 TI_1 置 1。



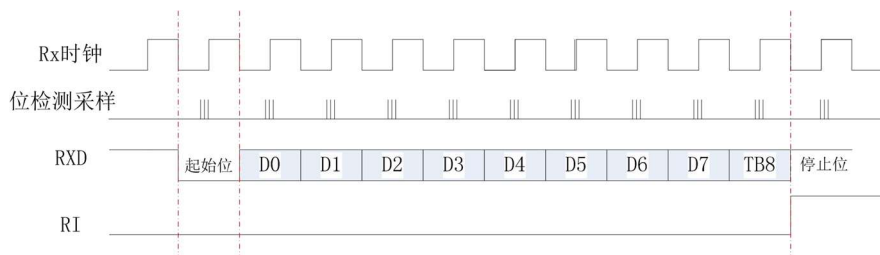
方式 2 接收时序

(2) 方式 2 接收

只有在 REN_1=1 时才允许接收数据。接收时, 数据由 RXD1 脚输入接收 11 位信息。当位检测逻辑采样到 RXD1 引脚从 1 到 0 的负跳变, 并判断起始位有效后, 便开始接收一帧信息。在接收完第 9 位数据后, 需满足一下两个条件, 才能将接收到的数据送入 SBUF1。

- c. RI_1=0, 意味着接收器缓冲器为空
- d. SM2V=0, 或者接收到第 9 位数据位 RB8_1=1 时

当满足上述两个条件时, 接收到的数据送入 SBUF1, 第 9 位数据送入 RB81, 且 RI1 置 1。否则, 数据被丢弃。



方式 2 接收时序

方式 3: 9 位 EUART, 可变波特率, 异步全双工

方式 3 的传输协议与方式 2 相同, 而波特率与方式 1 产生方法相同。

7.3.2 波特率 SBRTL、SBRTH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
FCH(r/w)	SBRTL	SBRTL<7:0>							
FDH(r/w)	SBRTH	TREN	SBRTH<6:0>						

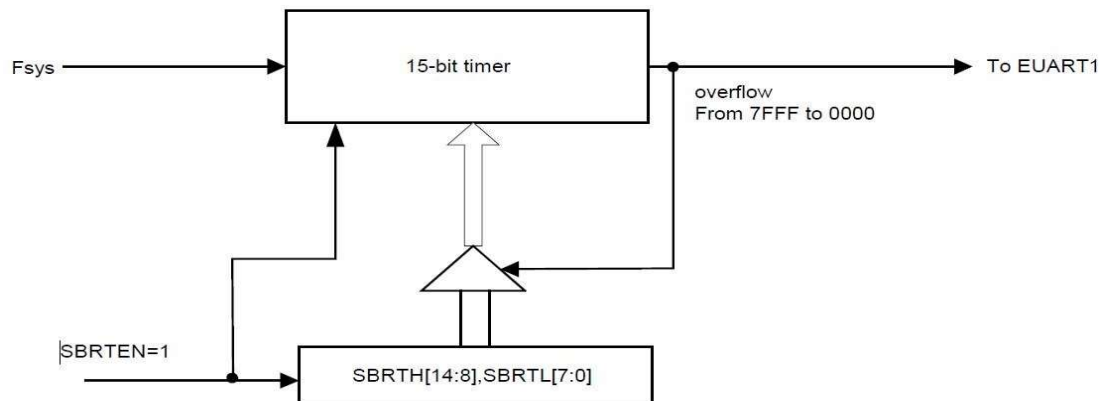
reset	00H	0	0	0	0	0	0	0	0
-------	-----	---	---	---	---	---	---	---	---

BSBTH/BSBTL: EUART1 波特率控制寄存器。

TREN: EUART1 使能位

- 0: EUART1 关闭
- 1: EUART1 使能

EUART1 自带一个波特率发生器，它实质上就是一个 15 位递增计数器。



$$\text{SBRT overflowrate} = \frac{f_{T1}}{32768 - \text{SBRT}}, \text{ 其中 SBRT}=[\text{SBRTH}, \text{SBRTL}].$$

因此，EUART1 在各模式下的波特率计算公式如下：

在方式 0 中，波特率可编程为系统时钟的 1/12 或 1/4，由 SM2_1 位决定。当 SM2_1 为 0 时，串行端口在系统时钟的 1/12 下运行。当 SM2_1 为 1 时，串行端口在系统时钟的 1/4 下运行。

在方式 1 和方式 3 中，波特率可微调，精度为一个系统时钟，公式如下：

$$\text{BaudRate} = \frac{F_{\text{sys}}}{16 \times (32768 - \text{SBRT})}$$

例如：Fsys = 4MHz，需要得到 19200Hz 的波特率，SBRT 值计算方法如下：

$$4000000/16/19200 = 13.02$$

$$\text{SBRT} = 32768 - 13 = 32755$$

$$19230 = 4000000/(16 \times 13), \text{存在误差。}$$

注：Fsys 和 Fosc 是不一样的，Fosc 是时钟源频率，如 16MHz，Fsys 为指令时钟频率，可能 2 分频，4 分频等，如 8MHz，4Mhz。和配置及 CPUM 寄存器有关。

7.4 EEPROM

7.4.1 EECON (EE 控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
B1H(r/w)	EECON	-	DLYS1	DLYS0	ERS16BTE	ERSALL_EE	ERASE_EE	PROG_EE	READ_EE
reset	00H	0	0	0	0	0	0	0	0

DLYS<1:0>: EEPROM 操作时长选择位。

0X: EEPROM 操作为默认时长

10: EEPROM 操作为默认时长 0.7 倍

11: EEPROM 操作为默认时长 1.3 倍

ERS16BTE: 一次擦除 16 个 byte 操作使能。

0: 未开始或已结束操作

1: 使能 16byte 擦除操作 (使能后, ERASE_EE 随后使能)

ERSALL_EE: ERSALL (全擦) 操作使能。

0: 未开始或已结束 ERSALL 操作

1: 使能 ERSALL 操作 (使能后, ERASE_EE 随后使能)

ERASE_EE: ERASE (字擦) 操作使能。

0: 未开始或已结束 ERASE 操作

1: 使能 ERASE 操作

PROG_EE: PROG 操作使能。

0: 未开始或已结束 PROG 操作

1: 使能 PROG 操作

READ_EE: READ 操作使能。

0: 禁止 READ 操作

1: 使能 READ 操作

7.4.2 EEADRL (EE 地址)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
B2H(r/w)	EEADRL	EE 地址低位							
reset	00H	0	0	0	0	0	0	0	0

7.4.3 EEDATA (EE 数据)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
B4H(r/w)	EEDATA	EE 数据低位							
reset	00H	0	0	0	0	0	0	0	0

7.4.4 EE 示例程序

```

//*****WRITE
MOV     EEADRL,#05H
MOV     EEDATA,#055H
MOV     EECON,#062H
LOOP_WRITE:
MOV     A,EECON
ANL     A,#02H
JNZ     LOOP_WRITE
//*****read_all
READ_EE_ALL:
MOV     R7,#0FFH
LOOP_EERD:
MOV     EEADRL,R7
MOV     EECON,#61H
NOP
MOV     A,EEDATA
DJNZ    R7,LOOP_EERD
MOV     EEADRL,#00H
MOV     EECON,#61H
NOP
MOV     A,EEDATA
RET
//*****erase_all
EEALL_ER:
MOV     EECON,#68H
MOV     EECON,#6CH
LOOP_ERA:
MOV     A,EECON
ANL     A,#04H
JNZ     LOOP_ERA
RET
//*****erase_byte
EEBYTE_ER:
MOV     EEADRL,A
MOV     EECON,#64H
LOOP_ERA_BYTE:
MOV     A,EECON
ANL     A,#04H

```

```
JNZ      LOOP_ERA_BYTE
RET
//*****erase_16X
ERASE_16X:
    MOV    EEADRL,A
    MOV    EECON,#70H
    MOV    EECON,#74H
LOOP_ERA16X:
    MOV    A,EECON
    ANL    A,#04H
    JNZ    LOOP_ERA16X
    RET
```

7.5 FLASH

MCU_CORE 支持在应用编程（IAP）功能，即可通过片内固件更新程序代码。IAP 功能可以对数据存储区进行读写操作，同时读数据也可以通过 MOV_C 指令来实现。

7.5.1 FLCON（FL 控制寄存器 1）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
A1H(r/w)	FLCON	FLEN	-	DLYS1	DLYS0	-	ERASE	PROG	-
reset	00H	0	0	0	0	0	0	0	0

FLEN: flash 的 IAP 使能位。

- 0: 不允许
- 1: 允许

DLYS<1:0>: FLASH 操作时长选择位。

- 0X: FLASH 操作为默认时长
- 10: FLASH 操作为默认时长 0.7 倍
- 11: FLASH 操作为默认时长 1.3 倍

ERASE: FLASH（页擦）操作使能。

- 0: 未开始或已结束 ERASE 操作
- 1: 使能 ERASE 操作

PROGL: PROG 操作使能。

- 0: 未开始或已结束 PROG 操作
- 1: 使能 PROG 操作

7.5.2 FLENA（FL 控制寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
A6H(r/w)	FLENA	FL 使能控制寄存器							
reset	00H	0	0	0	0	0	0	0	0

注：若想对 FLASH 进行操作，需将 FLENA 寄存器固定设为 5AH。

7.5.3 FLADRH/L（FL 地址高低位）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
A3H(r/w)	FLADRH	FL 地址高位							
A2H(r/w)	FLADRL	FL 地址低位							
reset	00H	0	0	0	0	0	0	0	0

注：对 FLASH 页擦时，FADRH<5:0>控制页擦地址，一页 256*8bit。

7.5.4 FLDATH/L (FL 数据高低位)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
A5H(r/w)	FLDATH	FL 数据高位							
A4H(r/w)	FLDATL	FL 数据低位							
reset	00H	0	0	0	0	0	0	0	0

写 FLASH 时，是一次写入 2 byte 的。即一次写入 2 个地址数据。

地址按 0000h, 0001h.....顺序，FLDATH=(0000H)，FLDATL=(0001H)，如果有效数据从奇地址开始，那么 FLDATH=00H。写完当前地址后，下一个地址要在当前地址加 2。

读 FLASH 数据需要使用 MOVC 指令读取。

注：EE 和 Flash 不能同时进行擦除和写操作！

7.5.5 FLASH 操作示例程序

```

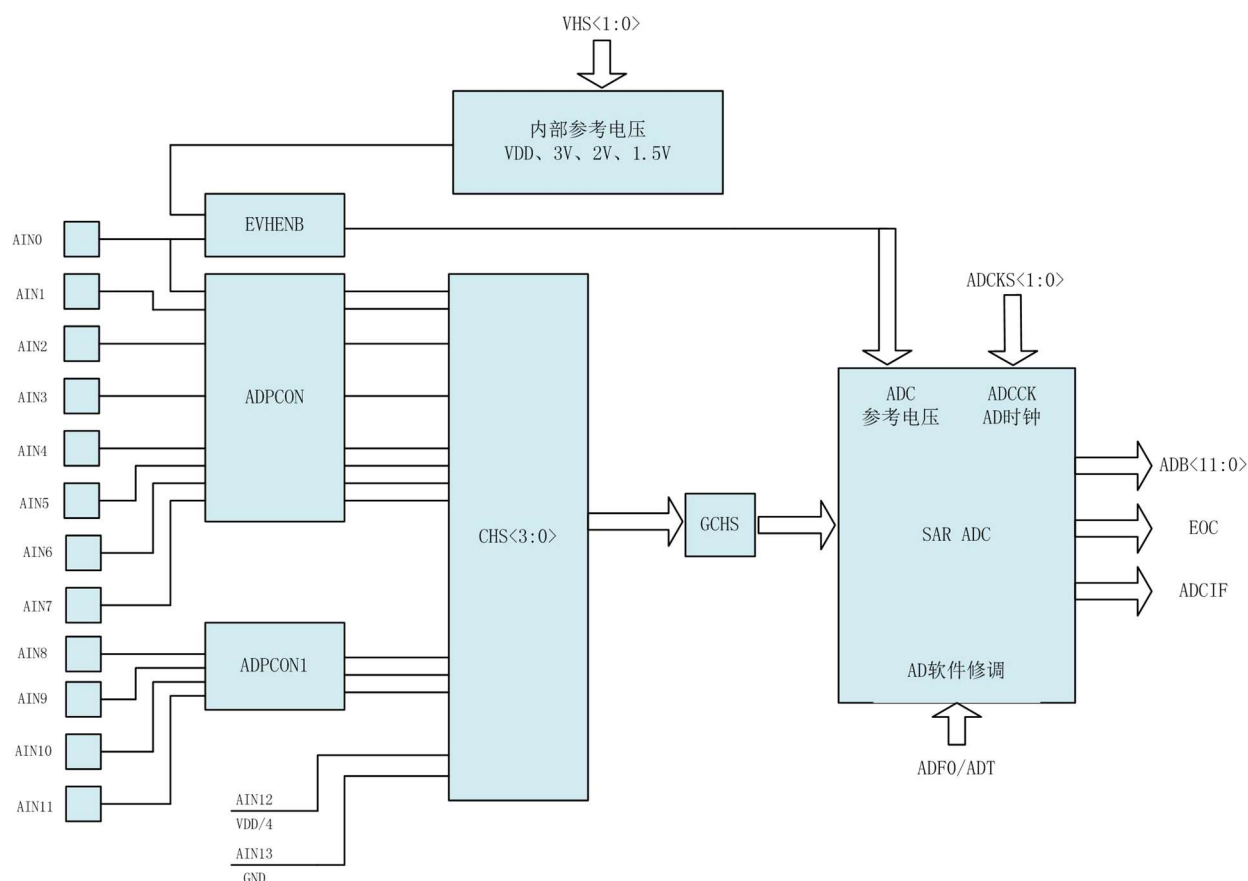
MOV      FLENA,#05AH
//*****erase_256*8BIT
MOV      FLADRH,#01H
MOV      FLADRL,#00H                ;addr=0100~1FFh
MOV      FLCON,#0B4H
MOV      A,#0B0H
LOOP_ERASE:
  CJNE    A,FLCON,LOOP_ERASE
//*****WRITE
MOV      FLADRH,#01H
MOV      FLADRL,#10H                ;addr=0110h
MOV      FLDATH,#055H
MOV      FLDATL,#0AAH
MOV      FLCON,#0B2H
MOV      A,#0B0H
LOOP_WRITE:
  CJNE    A,FLCON,LOOP_WRITE
//*****read
MOV      DPTR,#010FH
MOV      A,#01H
MOVC     A,@A+DPTR                ;ADDR=0110H
MOV      A,#02H
MOVC     A,@A+DPTR                ;ADDR=0111H

```

7.6 14 通道 ADC

7.6.1 概述

模拟数字转换，内置 14 个模拟通道，高达 4096 阶的分辨率，能将一个模拟信号转换成相应的 12 位数字信号。通过 CHS[3:0] 选择模拟信号输入引脚（AIN 引脚），内部 $1/4 \times V_{dd}$ 电压源，GCHS 位使能全部 ADC 通道，模拟信号输入至 SAR ADC。ADC 的分辨率为 12 位；可以通过 ADCKS[1:0] 位选择 ADC 的转换速率以决定 ADC 的转换时间。ADC 参考电压的高电平包括 2 种，内部参考源，包括 VDD、3V、2V、1.5V（EVHENB=0），外部参考源，由 P2.5 提供（EVHENB=1）。ADC 内置 ADPCON、ADPCON1 寄存器来设置模拟输入引脚，必须由程序将 ADC 输入引脚设为不带上拉电阻的输入引脚。设置好 ADENB 和 ADS 位后，ADC 开始转换，转换结束时，ADC 电路将 EOC 和 ADCIF 置 1，并将转换结果存入 ADB 和 ADR 寄存器中。若 EADC=1，ADC 请求中断，AD 转换完成后，ADCIF=1 时，程序计数器跳转中断向量地址执行中断服务程序。



7.6.2 ADM（ADC 模式寄存器）

ADC模式寄存器ADM设置ADC的相关配置：包括ADC启动，ADC通道选择，ADC的参考源选择和ADC处理状态显示等。必须在AD开始转换前将这些配置设置完毕。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C1H(r/w)	ADM	ADENB	ADS	EOC/ADCIF	GCHS	CHS3	CHS2	CHS1	CHS0
reset	00H	0	0	0	0	0	0	0	0

ADENB: ADC控制位。睡眠模式下，禁止ADC以省电。

- 0：禁止；
- 1：使能。

ADS: ADC启动位。ADC处理完成后，ADS位自动清零。

- 0：停止；
- 1：开始。

EOC: ADC状态控制位。

- 0：转换过程中；
- 1：转换结束，ADS复位。如果中断打开，则会进入中断。

GCHS: 通道选择位。

- 0：禁止AIN通道；
- 1：使能AIN通道。

CHS[3:0]: ADC输入通道选择位。

0000 = AIN0, 0001 = AIN1, 0010 = AIN2, 0011 = AIN3, 0100 = AIN4, 0101 = AIN5, 0110 = AIN6, 0111 = AIN7, 1000 = AIN8, 1001 = AIN9, 1010 = AIN10, 1011 = AIN11, 1100 = AIN12, 1101 = AIN13。

AIN12 通道没有连接外部管脚，连接的是 $1/4 \times VDD$ 。这种处理方式可以使 ADC 测得 VDD 的电压。这对于 VDD 变化的，特别是手持设备非常有用。AIN13 通道没有连接外部管脚，连接的是 GND。

ADR 寄存器包括 ADC 模式控制和 ADC 低字节数据缓存器，ADC 配置包括 ADC 时钟速率和 ADC 分辨率。必须在启动 ADC 之前设置好这些配置。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C3H(r/w)	ADR	-	ADCKS1	-	ADCKS0	ADB3	ADB2	ADB1	ADB0
reset	A0H	1	0	1	0	0	0	0	0

ADCKS1	ADCKS0	ADC 时钟源
0	0	Fcpu/16
0	1	Fcpu/8
1	0	Fcpu
1	1	Fcpu/2

ADB[3:0]: ADC12 位分辨率的低字节数据缓存器。

7.6.3 ADB（ADC 数据寄存器）

ADC 数据缓存器共 12 位，用来存储 AD 转换结果，8 位只读寄存器 ADB 存放结果的高字节（bit11~bit4），ADR（ADR[3:0]）存放低字节（bit3~bit0）。ADC 数据缓存器是只读寄存器，系统复位后处于未知状态。

ADB[11:4]: 8 位 ADC 模式下，ADC 数据存放于 ADB 寄存器中。

ADB[3:0]: 12 位 ADC 模式下，ADC 数据存放于 ADB 和 ADR 寄存器中。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C2H(r)	ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
resrt		X	X	X	X	X	X	X	X

ADB[11:4]: ADC12 位分辨率的高字节数据缓存器，系统复位后处于未知状态 X。

AIN n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
0/4096*VRE FH	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VRE FH	0	0	0	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...
4094/4096*V REFH	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*V REFH	1	1	1	1	1	1	1	1	1	1	1	1

针对不同的应用，用户可能需要精度介于 8 位到 12 位之间的 AD 转换器。对于这种情况，可以通过对保存在 ADR 和 ADB 中的转换结果进行处理得到，用户在转换结果中去掉最低的几位得到需要的结果。如下表所示：

ADC 分辨率	ADB								ADR			
	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
8-bit	O	O	O	O	O	O	O	O	X	X	X	X
9-bit	O	O	O	O	O	O	O	O	O	X	X	X
10-bit	O	O	O	O	O	O	O	O	O	O	X	X
11-bit	O	O	O	O	O	O	O	O	O	O	O	X
12-bit	O	O	O	O	O	O	O	O	O	O	O	O

O=可选位，X=未使用的位

7.6.4 VREFH（ADC 参考电压寄存器）

ADC 内置 5 种参考电压，由 VREFH 寄存器控制：包括一个外部参考电压和 4 个内部参考源（VDD、3V、2V、1.5V）。EVHENB=1 时，ADC 参考电压由外部参考源提供（P2.5），必须输入一个电压作为 ADC 参考电压的高电平，且不能低于 2V。EVHENB=0 时，ADC 参考电压由内部参考源提供，并由 VHS[1:0]选择控制。VHS[1:0]=11 时，ADC 参考源选择 VDD；VHS[1:0]=10 时，ADC 参考源选择 3V；VHS[1:0]=01 时，ADC 参考源选择 2V；VHS[1:0]=00 时，ADC 参考源选择 1.5V。外部参考源的限制条件为，最高为 VDD，最低为内部最低电平，否则默认为 VDD。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C5H(r)	VREFH	EVHENB	-	-	-	-	-	VHS1	VHS0
reset	00H	0	0	0	0	0	0	0	0

VHS[1:0]: ADC 内部参考电压选择位。

VHS1	VHS0	内部 VREFH 电压
1	1	VDD
1	0	3.0V
0	1	2.0V
0	0	1.5V

EVHENB: ADC 内部参考电压控制位。

- 0: 允许 ADC 内部 VREFH 功能，VREFH 引脚是 P2.5/AIN0 引脚；
- 1: 禁止 ADC 内部 VREFH 功能，P2.5/AIN0/VREFH 引脚来自外部 VREFH 输入引脚。

7.6.5 ADC 操作说明和注意事项

7.6.5.1 ADC 信号格式

ADC 采样电压范围为参考电压高/低电平之间，ADC 参考低电压为 VSS，高电压包括 VDD/4V/3V/2V，外部参考电压由 P2.5/AVREFH 引脚提供（由 EVHENB 控制）。EVHENB = 0 时，ADC 参考电压选择内部参考源；EVHENB = 1 时，ADC 参考电压选择外部参考源（P2.5/AVREFH）。ADC 参考电压的范围为：（ADC 参考高电压-ADC 参考低电压） $\geq$ 2V，ADC 参考低电压为 VSS=0V，故 ADC 参考高电压范围为 2V~VDD，外部参考电压需在此范围之内。

ADC 内部参考低电压=0V。

ADC 内部端口低电压=VDD/3V/2V/1.5V。（EVHENB = 0）

ADC 外部参考电压=2V~VDD。（EVHENB = 1）

ADC 采样输入信号电压必须在 ADC 参考低电压和 ADC 参考高电压之间，若 ADC 输入信号的电压不在此范围内，则 ADC 的转换结果会出错（满量程或者为 0）。

ADC 参考低电压 $\leq$ ADC 采用输入信号电压 $\leq$ ADC 参考高电压

7.6.5.2 AD 转换时间

ADC转换时间是指从ADS=1（开始ADC）到EOC=1（ADC结束）所用的时间，由ADC分辨率和ADC时钟Rate控制，ADC的转换时间为 $1/(\text{ADC时钟}/4) * 16 \text{ S}$ 。ADC的时钟源为Fcpu，包括Fcpu/1，Fcpu/2，Fcpu/8，Fcpu/16，由ADCKS[1:0]位控制。

ADC 的转换时间会影响 ADC 的性能，如果输入高 Rate 的模拟信号，必须要选择一个高 Rate 的 ADC 转换 Rate。如果 ADC 的转换时间比模拟信号的转换 Rate 慢，则 ADC 的结果出错。故选择合适的 ADC 时钟 Rate 和 ADC 分辨率才能得到合适的 ADC 转换 Rate。

12 位 ADC 转换时间 = $1/(\text{ADC 时钟 Rate}/4) * 16 \text{ sec}$

示例如下：（Fcpu=Fosc/2=8MHZ 为例）

ADCKS1 ADCKS0	ADC 时钟 rate	Fcpu=8MHz	
		AD 转换时间	AD 转换 Rate
00	Fcpu/16	$1/(8\text{MHz}/16/4) * 16 = 128\mu\text{s}$	7.81KHz
01	Fcpu/8	$1/(8\text{MHz}/8/4) * 16 = 64\mu\text{s}$	15.625KHz
10	Fcpu	$1/(8\text{MHz}/4) * 16 = 8\mu\text{s}$	125KHz
11	Fcpu/2	$1/(8\text{MHz}/2/4) * 16 = 16\mu\text{s}$	62.5KHz

7.6.5.3 ADC 引脚配置

ADC输入引脚与IO口共用，ADC输入通道的选择由ADCHS[3:0]控制，ADCCHS[3:0]=0000时选择AIN0，ADCCHS[3:0]=0001时选择AIN1.....同一时间设置AIN口的一个引脚作为ADC的输入引脚，该引脚必须设置为输入引脚，禁止内部上拉，并首先由程序使能ADPCON，ADPCON1寄存器。通过ADCHS[3:0]选择好ADC输入通道后，GCHS置1以使能ADC功能。

ADC输入引脚为GPIO引脚时必须设为输入模式。

必须禁止ADC输入引脚的内部上拉电阻。

ADC输入通道的ADPCON，ADPCON1位必须置1。

EVHENB = 1时，P2.5/AIN0为ADC外部参考源的输入引脚，此时，P2.5必须设为输入模式，并禁止其上拉电阻。

ADC外部参考源输入引脚为GPIO引脚时必须设为输入模式。

必须禁止ADC外部参考源输入引脚的内部上拉电阻。

ADC 输入引脚与普通 I/O 引脚共用。当输入一个模拟信号到 CMOS 结构端口时，尤其当模拟信号为 $1/2 V_{DD}$ 时，可能产生额外的漏电流。当端口输入多个模拟信号时，也会产生额外的漏电流。睡眠模式下，上述漏电流会严重影响到系统的整体功耗。ADCON 为 AIN 口的配置寄存器，将ADPCON[7:0]，ADPCON1[3:0]置 1，其对应的 AIN 引脚将被设为纯模拟信号输入引脚，从而避免上述漏电流的产生。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C6H(r)	ADPCON	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

reset	00H	0	0	0	0	0	0	0	0
-------	-----	---	---	---	---	---	---	---	---

ADPCON[7: 0]: ADC 输入 AIN7~AIN0 脚配置控制位。

0 : P1.4~P1.0,P2.7~P2.5 可以作为模拟输入（ADC 输入）引脚或者 GPIO 引脚；

1 : P1.4~P1.0,P2.7~P2.5 只能作为模拟输入引脚，不能作为 GPIO 引脚。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C7H(r)	ADPCON1	-	-	-	-	ADPC11	ADPC10	ADPC9	ADPC8
reset	00H	0	0	0	0	0	0	0	0

ADPCON1[3: 0]: ADC 输入 AIN11~AIN8 脚配置控制位。

0 : P0.1~P0.0,P1.6~P1.5 可以作为模拟输入（ADC 输入）引脚或者 GPIO 引脚；

1 : P0.1~P0.0,P1.6~P1.5 只能作为模拟输入引脚，不能作为 GPIO 引脚。

注：当 ADC 的 AIN_n 作为普通 I/O 口而不是 ADC 输入引脚时，ADPCON.n 必须置为 0，否则 P3.n/P2.7~P2.4 的普通 I/O 信号会被隔离开来。

ADC 输入通道与 ADC 采样脚配置对应关系如下表：

CHS<3:0>	ADPCON	ADPCON1	AD 脚
0000	01H	00H	P2.5/AIN0
0001	02H	00H	P2.6/AIN1
0010	04H	00H	P2.7/AIN2
0011	08H	00H	P1.0/AIN3
0100	10H	00H	P1.1/AIN4
0101	20H	00H	P1.2/AIN5
0110	40H	00H	P1.3/AIN6
0111	80H	00H	P1.4/AIN7
1000	00H	01H	P1.5/AIN8
1001	00H	02H	P1.6/AIN9
1010	00H	04H	P0.0/AIN10
1011	00H	08H	P0.1/AIN11
1100	00H	00H	VDD/4
1101	00H	00H	GND

注：使用 AIN8（P1.5）进行采样，采样电压需大于 0.7V，否则禁止使用。

7.6.6 P2/P1/P0 口 ADC 共用引脚

P2 口、P1 口、P0 口和 ADC 的输入口共用，施密特触发。同一时间只能设置 P2 口、P1 口、P0 口的一个引脚作为 ADC 的测量信号输入口（通过 ADM 寄存器来设置），其它引脚则作为普通 I/O

使用。具体应用中，当输入一个模拟信号到 CMOS 结构端口，尤其当模拟信号为 $1/2 V_{DD}$ 时，将可能产生额外的漏电流。同样，当 P2 口、P1 口、P0 口外接多个模拟信号时，也会产生额外的漏电流。在睡眠模式下，上述漏电流会严重影响到系统的整体功耗。ADPCON，ADPCON1 为 P2 口、P1 口、P0 口的配置寄存器。将 ADPCON，ADPCON1 置“1”，其对应的 P2 口、P1 口、P0 口将被设置为纯模拟信号输入口，从而避免上述漏电流的情况。

7.6.7 ADT（ADC 结果微调寄存器）

ADC模式寄存器ADM设置ADC的相关配置：包括ADC启动，ADC通道选择，ADC的参考源选择和ADC处理状态显示等。必须在AD开始转换前将这些配置设置完毕。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
C4H(r/w)	ADT	ADV1	ADV0	-	ADTS	ADT<3>	ADT<2>	ADT<1>	ADT<0>
reset	00H	0	0	0	0	0	0	0	0

ADV1~ADV0: AD修调控制寄存器

当两个值不同时，可以使用ADF0寄存器控制VREF的修调值（软件修调）。当两个值相同时，则VREF的修调值受配置字控制（硬件自动修调）。

ADTS: 修调方向控制。

0 : 负方向(ADB转换制变小);

1 : 正方向(ADB转换制变大)。

ADT<3:0>: ADT修调位

0000 : 修调0mv

0001 : 修调1mv

0010 : 修调2mv

0011 : 修调3mv

0100 : 修调4mv

0101 : 修调5mv

0110 : 修调6mv

0111 : 修调7mv

1000 : 修调8mv

1001 : 修调9mv

1010 : 修调10mv

1011 : 修调11mv

1100 : 修调12mv

1101 : 修调13mv

1110 : 修调14mv

1111 : 修调15mv

注：如果得到的 AD 转换值偏小，可以使 ADTS<0>=1,并适当增加 ADT<3:0>的值做调整，反之亦然。

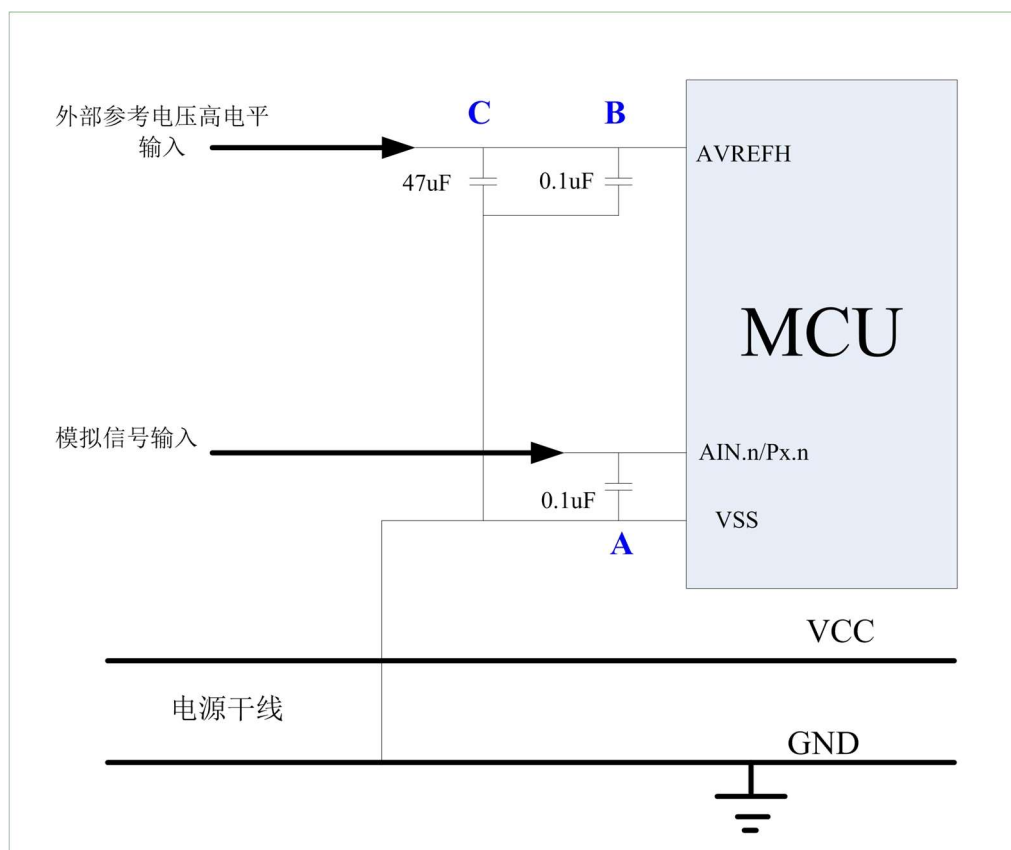
7.6.8 ADF0（ADC 基准修调）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----

FF27H(r/w)	ADF0	ADF0<7:0>							
reset	00H	0	0	0	0	0	0	0	0

ADC 基准修调可以软件修调，可以硬件修调，默认情况下硬件自动修调基准。若 ADV1 与 ADV0 值不相等，则进行软件修调。

7.6.9 ADC 应用电路



模拟信号从 ADC 输入引脚 AIN_n 输入。在 ADC 输入引脚和 VSS 之间 (A) 必须连接一个 0.1uF 的电容，且要尽可能的靠近 ADC 输入引脚。不能将电容的 GND 直接连接到电源干线的 GND，必须通过 VSS 引脚。该电容可以减小电源干扰对模拟信号的影响。

ADC 参考电压高电平由外部参考源提供，外部参考源连接到 AVREFH 引脚 (P2.5)。在 AVREFH 引脚和 VSS 之间连接电容，首先在图中 C 处连接一个 47uF 的电解电容，再在 B 处连接一个 0.1uF 的电容，且要尽可能的靠近 AVREFH 引脚。不能将电容的 GND 直接连接到电源干线上的 GND，必须通过 VSS 引脚。

7.6.10 ADC 程序示例

```

SETB    EA
SETB    EADC                ;使能 ADC 中断
MOV     ADM,#00010100B      ;ADC 输入通道选择 AIN4,并打开通道
MOV     ADR,#01010000B      ;ADC 时钟选择 Fcpu/2
    
```

```

MOV      VREFH,#00000001B      ;选择 2v 内部参考电压
MOV      ADPCON,#08H           ;P1.0;选择 P1.0 作为 ADC 采样脚
MOV      ADM,#10010100B        ;ADENB
MOV      ADT,#015H             ;ADC 转换结果加 5mv
START_AD:
ORL      ADM,#040H             ;ADS
MOV      A,ADM
ANL      A,#20H
WAIT_LOOP:
CJNE     A,#20H,WAIT_LOOP      ;转换完成
. . .                          ;处理 ADC 转换结果
    
```

7.7 带死区互补的 12 位 PWM (TM1)

7.7.1 TM1 特性

- 3 组带死区互补 PWM 或 6 路独立 PWM 输出
- 提供每个 PWM 周期溢出中断，但中断共用同一个向量入口
- 输出极性可选择
- PWM 工作时钟源可设定时钟分频比
- TM1 可做定时器/计数器使用

7.7.2 TM1 相关寄存器

7.7.2.1 使能寄存器 PWMEN

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
F7H(r/w)	PWMEN	-	-	PWM5_EN	PWM4_EN	PWM3_EN	PWM2_EN	PWM1_EN	PWM0_EN
reset	00H	0	0	0	0	0	0	0	0

PWM0~5_EN: PWM0~5 输出控制位

0: 禁止 PWM0~5 输出

1: 允许 PWM0~5 输出

注: PWM0~5 允许输出, 必须在 PWM0~5_EN 置 1 下才有效, 否则为 PWM0~5 输出关闭状态。
当所有 PWM 都不输出时, 可以当做普通定时器用。

7.7.2.1 输出选择寄存器 PWMPS

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
D6H(r/w)	PWMPS	-	-	PWM5PS	PWM4PS	PWM3PS	PWM2PS	PWM1PS	PWM0PS
reset	00H	0	0	0	0	0	0	0	0

PWM0PS: PWM0 输出口选择

0: P2.4 口输出 PWM0

1: P2.2 口输出 PWM0

PWM1PS: PWM1 输出口选择

0: P2.5 口输出 PWM1

1: P2.1 口输出 PWM1

PWM2PS: PWM2 输出口选择

0: P2.6 口输出 PWM2

1: P1.6 口输出 PWM2

PWM3PS: PWM3 输出口选择

0: P2.7 口输出 PWM3

1: P1.5 口输出 PWM3

PWM4PS: PWM4 输出口选择

0: P1.0 口输出 PWM4

1: P1.4 口输出 PWM4

PWM5PS: PWM5 输出口选择

0: P1.2 口输出 PWM5

1: P1.3 口输出 PWM5

注: 输出 PWM 时对应端口必须设为输出模式。

7.7.2.2 极性寄存器 PWMINV

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
D5H(r/w)	PWMINV	-	-	PWM5INV	PWM4INV	PWM3INV	PWM2INV	PWM1INV	PWM0INV
reset	00H	0	0	0	0	0	0	0	0

PWM0~5INV: PWM0~5 输出极性控制位

0: PWM0~5 不取反

1: PWM0~5 输出取反

7.7.2.3 PWM0 计数寄存器 PWMTL、PWMTH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
D1H(r/w)	PWMTL	PWMTL[7:0]							
D2H(r/w)	PWMTH	-	-	-	-	PWMTH[3:0]			
reset	00H	0	0	0	0	0	0	0	0

注: PWMTH:PWMTL 是加 1 计数器, 从 PWMPH:PWMPL 开始计数至 0FFF 溢出。

7.7.2.4 PWMCON 控制寄存器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
E7H(r/w)	PWMCON	PWMTR	PWMIF	-	PWMD	CKSEL	CK0[2]	CK0[1]	CK0[0]
reset	00H	0	0	0	0	0	0	0	0

PWMTR: PWM 使能

0: 禁止

1: 使能

PWMIF: PWM 溢出中断标志

0: 无溢出

1: 溢出 (软件清 0)

PWMD: PWM 模式设置

0: PWM 独立模式, 6 个通道可以不同占空比

1: PWM 互补模式, 6 个通道分成 3 个互补通道, 可设置死区时间。PWM1/3/5 当做死区时间控制器。

CK0[2:0]	CKSEL=0	CKSEL=1
111	Fcpu/2	Fosc/1
110	Fcpu/4	Fosc/2
101	Fcpu/8	Fosc/4
100	Fcpu/16	Fosc/8
011	Fcpu/32	Fosc/16
010	Fcpu/64	Fosc/32
001	Fcpu/128	Fosc/64
000	Fcpu/256	Fosc/128

7.7.2.5 PWM0 周期寄存器 PWM0PL、PWM0PH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
D3H(r/w)	PWMPL	PWM0PL[7:0]							
D4H(r/w)	PWMPH	-	-	-	-	PWM0PH[3:0]			
reset	00H	0	0	0	0	0	0	0	0

PWM0 周期=[0XFFF-PWM0PH:PWM0PL]*PWM0 工作时钟源周期。

注：修改 PWM0 周期寄存器，须先修改高位后修改低位，且修改都在下一个周期才有效。

注：PWM0PH: PWM0PL 组成 12 位数据。

7.7.2.6 PWM0 占空比寄存器 PWM0DL、PWM0DH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
E1H(r/w)	PWM0DL	PWM0DL[7:0]							
E2H(r/w)	PWM0DH	-	-	-	-	PWM0DH[3:0]			
reset	00H	0	0	0	0	0	0	0	0

PWM0 占空比=[PWM0DH:PWM0DL-PWM0PH:PWM0PL]*PWM0 工作时钟周期

注：修改 PWM0 占空比寄存器，操作类似修改 PWM0 周期寄存器，都是必须先修改高位后修改低位，且修改都在下一个周期才有效。

注：定时器 TM1 计数到 FFF 溢出后重装载 PWM0PH/L 的值，计数到 PWM0DH/L 时 PWM0 翻转(溢出到翻转这段时间为占空比)，计数到 FFF 溢出再翻转。

7.7.2.7 PWM1 死区时间寄存器 PWMD1L、PWMD1H

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
E3H(r/w)	PWMD1L	PWMD1L[7:0]							
E4H(r/w)	PWMD1H	-	-	-	-	PWMD1H[3:0]			

reset	00H	0	0	0	0	0	0	0	0
-------	-----	---	---	---	---	---	---	---	---

当 PWMD=0 时, PWM0/1 工作在 2 路独立模式, 此时的死区时间寄存器控制 PWM1 的占空比, 即独立模式的 PWM0/1 可以产生 2 路周期相同, 但占空比可以不同的 PWM 波形。当 PWMD=1 时, PWM0/1 工作在互补模式, 此时的死区时间寄存器控制 PWM1 的死区时间。

互补模式下: PWM1 死区时间=[PWMD1H:PWMD1L]*PWM0 工作时钟周期。

互补模式下: PWM1 死区时间必须小于 PWM0 占空比时间, 死区时间与占空比时间的和必须小于 PWM0 周期。

独立模式下: PWM1 占空比时间=[PWMD1H:PWMD1L-PWM0PH:PWM0PL]*PWM0 工作时钟周期。

注: 修改 PWM1 占空比寄存器 (死区寄存器), 操作类似修改 PWM1 周期寄存器, 都是必须先修改高位后修改低位, 且修改都在下一个周期才有效。

注: 独立模式下定时器 TM1 计数到 FFF 溢出后重装载 PWM0PH/L 的值, 计数到 PWM1DH/L 时 PWM1 翻转(溢出到翻转这段时间为占空比), 计数到 FFF 溢出再翻转。

7.7.2.8 PWM 占空比寄存器 PWM2DL/PWM2DH~PWM5DL/PWM5DH

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
E5H(r/w)	PWM2DL	PWM2DL[7:0]							
E6H(r/w)	PWM2DH	-	-	-	-	PWM2DH[3:0]			
F1H(r/w)	PWM3DL	PWM3DL[7:0]							
F2H(r/w)	PWM3DH	-	-	-	-	PWM3DH[3:0]			
F3H(r/w)	PWM4DL	PWM4DL[7:0]							
F4H(r/w)	PWM4DH	-	-	-	-	PWM4DH[3:0]			
F5H(r/w)	PWM5DL	PWM5DL[7:0]							
F6H(r/w)	PWM5DH	-	-	-	-	PWM5DH[3:0]			
reset	00H	0	0	0	0	0	0	0	0

注: PWM2DH/L 和 PWM3DH/L 是一组, PWM4DH/L 和 PWM5DH/L 是一组, 他们操作方式与 PWM0DH/L 和 PWM1DH/L 是一样的。

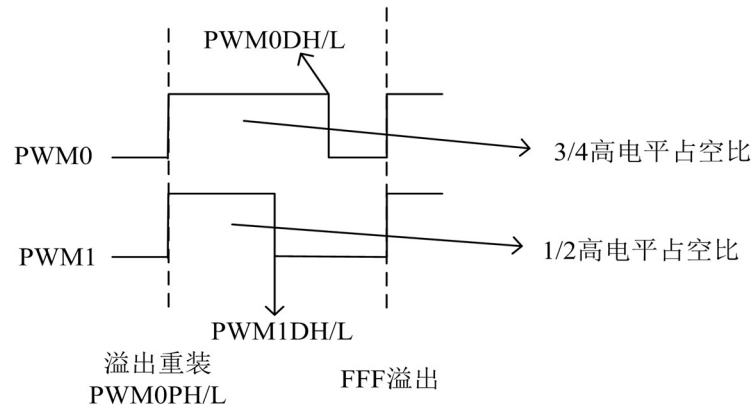
7.7.2.9 PWM 独立模式示例程序及示例图

```

MOV      IE2,#04H          ;PWMIE=1
SETB     EA
MOV      PWMEN,#03H        ;PWM0/PWM1 使能
MOV      PWMINV,#00H       ;极性不取反, 高电平占空比
MOV      PWMPS,#00H        ;默认 P2.4 输出 PWM0, P2.5 输出 PWM1
MOV      PWMPH,#08H
MOV      PWMPH,#00H        ;周期寄存器, 800~FFF
MOV      PWM0DH,#0EH
MOV      PWM0DL,#00H       ;PWM0, 3/4 高电平占空比

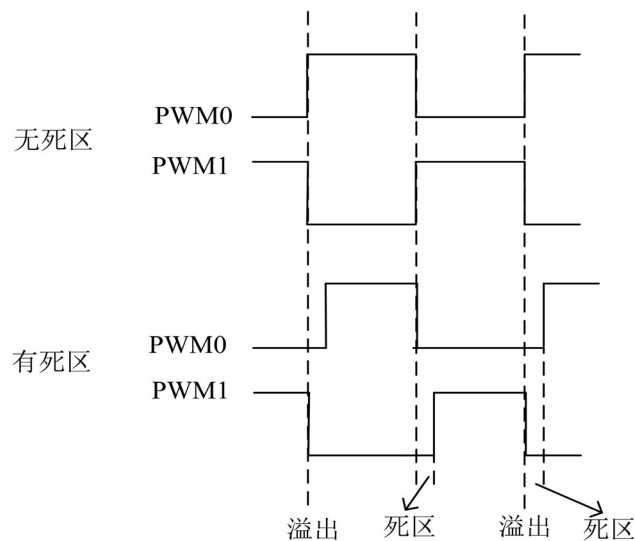
```

```
MOV    PWM1DH,#0CH
MOV    PWM1DL,#00H    ;PWM1, 1/2 高电平占空比
MOV    PWMCON,#088H   ;独立模式, PWM 总使能, Fosc/128
```



7.7.2.10 PWM 互补模式示例程序及示例图

```
MOV    PWMEN,#00H
MOV    PWMINV,#00H
MOV    PWMPSC,#00H
MOV    PWMPH,#08H
MOV    PWMPL,#00H    ;周期寄存器, 800~FFF
MOV    PWM0DH,#0CH
MOV    PWM0DL,#00H   ;PWM0, 1/2 高电平占空比
MOV    PWM1DH,#00H
MOV    PWM1DL,#50H   ;死区寄存器, 死区时间为计数 80 次的时间
MOV    PWMCON,#098H  ;互补模式, PWM 总使能, Fosc/128
```



8. 系统时钟

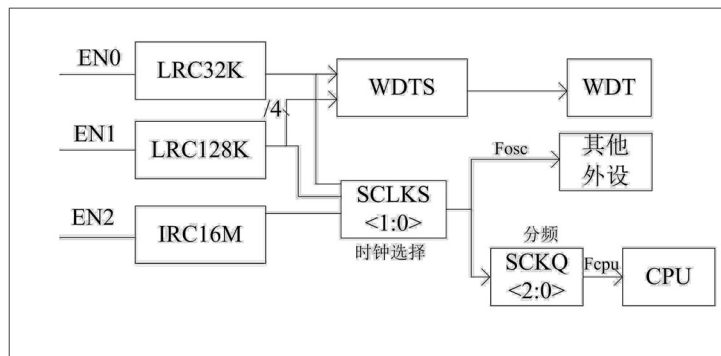
8.1 概述

MCU_CORE内置双时钟系统：高速时钟和低速时钟。低速时钟由内部低速振荡器提供，由CPUM寄存器控制，高、低速时钟都可以作为系统时钟源。内部高速振荡器：IRC16M。

内部低速振荡器：LRC32K以及LRC128K。

Fosc：系统时钟频率。

Fcpu：指令执行频率。



8.2 指令周期 Fcpu

系统时钟速率，即指令周期（Fcpu），从系统时钟源分离出来，决定系统的工作速率。Fcpu的速率由Fcpu编译选项决定，正常模式下， $F_{cpu} = F_{osc}/2 \sim F_{osc}/128$ 。若高速时钟源为内部16MHz振荡器，Fcpu编译选项选择Fosc/4，则Fcpu频率为16MHz/4=4MHz。

若选择LRC32K作为系统时钟，则Fosc为32K，参与分频，Fcpu编译选项选择Fosc/2，即 $F_{cpu} = 32KHz/2 = 16KHz$ 。

若选择LRC128K作为系统时钟，则Fosc为128K，参与分频，Fcpu编译选项选择Fosc/2，即 $F_{cpu} = 128KHz/2 = 64KHz$ 。

高干扰环境下，强烈建议设置 code option 的 Fcpu 选项在 $F_{osc}/4$ 以下，以减少干扰的影响。

注：Fosc/1 不可使用

9. 复位

9.1 概述

MCU_CORE有以下几种复位方

式：上电复位；

看门狗复位；

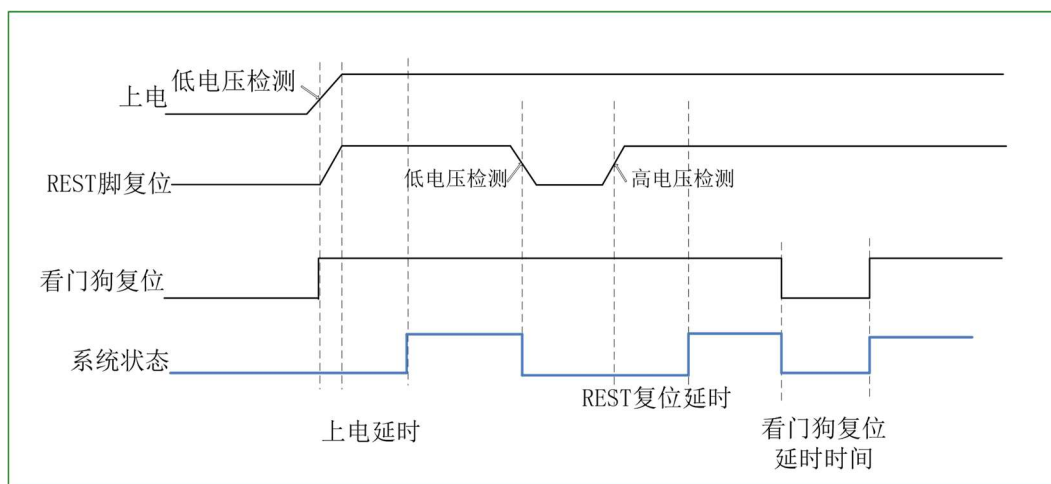
掉电复位；

REST脚复位；

软件复位；

上述任一种复位发生时，所有的系统寄存器恢复默认状态，程序停止运行，同时程序计数器PC清零。复位结束后，系统从向量0000H处重新开始运行。RSTIQ寄存器的标志位能够给出系统复位状态的信息。

任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。对于不同类型的振荡器，完成复位所需要的时间也不同。因为，VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的起振时间最短，晶体振荡器的起振时间则较长。在用户终端使用的过程中，应注意考虑主机对上电复位时间的要求。



9.2 上电、掉电复位

系统上电的过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。下面给出上电复位的正常时序：

上电：系统检测到电源电压上升并等待其稳定；

系统初始化：所有的系统寄存器被置为初始值；

振荡器开始工作：振荡器开始提供系统时钟；

执行程序：上电结束，PORIF复位标志变高，程序开始运行。

系统掉电会发生复位，BORIF复位标志变高：

9.3 看门狗复位

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，看门狗定时器溢出，此时系统复位。看门狗复位后，系统重启进入正常状态。看门狗复位的时序如下：

看门狗定时器状态：系统检测看门狗定时器是否溢出，若溢出，则系统复位；

系统初始化：所有的系统寄存器被置为默认状态；

振荡器开始工作：振荡器开始提供系统时钟；

执行程序：上电结束，程序开始运行。

看门狗定时器应用注意事项：

对看门狗清零之前，检查I/O口的状态和RAM的内容可增强程序的可靠性；

不能在中断中对看门狗清零，否则无法侦测到主程序跑飞的状况；

程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。

注：看门狗溢出则WDTIF变高，若在disable when sleep模式下，WDTPD=1，则睡眠模式下，WDT溢出不复位，会唤醒睡眠模式。

9.4 REST 脚复位（P1.7）

外部复位（仅限于外部复位引脚使能状态）：系统检测外部复位引脚状态。如果不为高电平，系统保持复位状态直到外部复位引脚释放，发生外部复位则EXRSTIF变高。

9.5 软件复位

执行 MOV SRST,#5AH 即可软件复位，SRSTIF 变高。

执行 MOV SRST,#A5H 即可软件复位，SRSTIF 变高。

10. 指令集

10.1 数据传送指令

助 记 符	说 明	字节	周期	对标志位影响			
				P	OV	AC	CY
MOV A,Rn	寄存器送 A	1	1	√	—	—	—
MOV A,direct	直接字节送 A	2	2	√	—	—	—
MOV A,@Ri	间接 RAM 送 A	1	2	√	—	—	—
MOV A,#data	立接数送 A	2	2	√	—	—	—
MOV Rn,A	A 送寄存器	1	1	—	—	—	—
MOV Rn,direct	直接数送寄存器	2	2	—	—	—	—
MOV Rn,#data	立即数送寄存器	2	2	—	—	—	—
MOV direct,A	A 送直接字节	2	2	—	—	—	—
MOV direct,Rn	寄存器送直接字节	2	2	—	—	—	—
MOV direct,direct	直接字节送直接字节	3	3	—	—	—	—
MOV direct,@Ri	间接 Rn 送直接字节	2	3	—	—	—	—
MOV direct,#data	立即数送直接字节	3	3	—	—	—	—
MOV @Ri,A	A 送间接 Rn	1	2	—	—	—	—
MOV @Ri,direct	直接字节送间接 Rn	2	3	—	—	—	—
MOV @Ri,#data	立即数送间接 Rn	2	2	—	—	—	—
MOV DPTR,#data16	16 位常数送数据指针	3	3	—	—	—	—
MOV C,bit	直接位送进位位	2	2	—	—	—	√
MOV bit,C	进位位送直接位	2	2	—	—	—	—
MOVC A,@A+DPTR	A+DPTR 寻址程序存贮字节送 A	1	3	√	—	—	—
MOVC A,@A+PC	A+PC 寻址程序存贮字节送 A	1	3	√	—	—	—
MOVX A,@Ri	外部数据送 A (8 位地址)	1	2	√	—	—	—
MOVX A,@DPTR	外部数据送 A (16 位地址)	1	2	√	—	—	—
MOVX @Ri,A	A 送外部数据 (8 位地址)	1	2	—	—	—	—
MOVX @DPTR,A	A 送外部数据 (16 位地址)	1	2	—	—	—	—
PUSH direct	直接字节进栈道, SP 加 1	2	3	—	—	—	—
POP direct	直接字节出栈, SP 减 1	2	3	—	—	—	—
XCH A,Rn	寄存器与 A 交换	1	1	√	—	—	—
XCH A,direct	直接字节与 A 交换	2	2	√	—	—	—
XCH A,@Ri	间接 Rn 与 A 交换	1	2	√	—	—	—
XCHD A,@Ri	间接 Rn 与 A 低半字节交换	1	2	√	—	—	—

10.2 逻辑运算指令

助 记 符	说 明	字节	周期	对标志位影响			
				P	OV	AC	CY
ANL A,Rn	寄存器与到 A	1	1	√	—	—	—
ANL A,direct	直接字节与到 A	2	2	√	—	—	—
ANL A,@Ri	间接 RAM 与到 A	1	2	√	—	—	—
ANL A,#data	立即数与到 A	2	2	√	—	—	—
ANL direct,A	A 与到直接字节	2	2	—	—	—	—
ANL direct,#data	立即数与到直接字节	3	3	—	—	—	—
ANL C,bit	直接位与到进位位	2	2	—	—	—	√
ANL C,/bit	直接位的反码与到进位位	2	2	—	—	—	√
ORL A,Rn	寄存器或到 A	1	1	√	—	—	—
ORL A,direct	直接字节或到 A	2	2	√	—	—	—
ORL A,@Ri	间接 RAM 或到 A	1	2	√	—	—	—
ORL A,#data	立即数或到 A	2	2	√	—	—	—
ORL direct,A	A 或到直接字节	2	2	—	—	—	—
ORL direct,#data	立即数或到直接字节	3	3	—	—	—	—
ORL C,bit	直接位或到进位位	2	2	—	—	—	√
ORL C,/bit	直接位的反码或到进位位	2	2	—	—	—	√
XRL A,Rn	寄存器异或到 A	1	1	√	—	—	—
XRL A,direct	直接字节异或到 A	2	2	√	—	—	—
XRL A,@Ri	间接 RAM 异或到 A	1	2	√	—	—	—
XRL A,#data	立即数异或到 A	2	2	√	—	—	—
XRL direct,A	A 异或到直接字节	2	2	—	—	—	—
XRL direct,#data	立即数异或到直接字节	3	3	—	—	—	—
SETB C	进位位置 1	1	1	—	—	—	1
SETB bit	直接位置 1	2	2	—	—	—	—
CLR A	A 清 0	1	1	√	—	—	—
CLR C	进位位清 0	1	1	—	—	—	0
CLR bit	直接位清 0	2	2	—	—	—	—
CPL A	A 求反码	1	1	—	—	—	—
CPL C	进位位取反	1	1	—	—	—	√
CPL bit	直接位取反	2	2	—	—	—	—
RL A	A 循环左移一位	1	1	—	—	—	—
RLC A	A 带进位左移一位	1	1	√	—	—	√
RR A	A 右移一位	1	1	—	—	—	—
RRC A	A 带进位右移一位	1	1	√	—	—	√
SWAP A	A 半字节交换	1	1	—	—	—	—

10.3 算术运算指令

助 记 符	说 明	字 节	周 期	对标志位影响			
				P	OV	AC	CY
ADD A,Rn	寄存器加到A	1	1	√	√	√	√
ADD A,direct	直接字节加到A	2	2	√	√	√	√
ADD A,@Ri	间接RAM 加到A	1	2	√	√	√	√
ADD A,#data	立即数加到A	2	2	√	√	√	√
ADDC A,Rn	寄存器带进位加到 A	1	1	√	√	√	√
ADDC A,direct	直接字节带进位加到A	2	2	√	√	√	√
ADDC A,@Ri	间接RAM 带进位加到A	1	2	√	√	√	√
ADDC A,#data	立即数带进位加到 A	2	2	√	√	√	√
SUBB A,Rn	从 A 中减去寄存器和进位	1	1	√	√	√	√
SUBB A,direct	从 A 中减去直接字节和进位	2	2	√	√	√	√
SUBB A,@Ri	从 A 中减去间接RAM 和进位	1	2	√	√	√	√
SUBB A,#data	从 A 中减去立即数和进位	2	2	√	√	√	√
INC A	A 加 1	1	1	√	—	—	—
INC Rn	寄存器加 1	1	1	—	—	—	—
INC direct	直接字节加 1	2	2	—	—	—	—
INC @Ri	间接RAM 加 1	1	2	—	—	—	—
INC DPTR	数据指针加 1	1	1	—	—	—	—
DEC A	A 减 1	1	1	√	—	—	—
DEC Rn	寄存器减 1	1	1	—	—	—	—
DEC direct	直接字节减 1	2	2	—	—	—	—
DEC @Ri	间接RAM 减 1	1	2	—	—	—	—
MUL AB	A 乘 B	1	4	√	√	—	0
DIV AB	A 被 B 除	1	4	√	0/1	—	0
DA A	A 十进制调整	1	1	√	—	—	√

10.4 转移指令

助 记 符	说 明	字 节	周 期	对标志位影响			
				P	OV	AC	CY
AJMP addr 11	绝对转移	2	3	—	—	—	—
LJMP addr 16	长转移	3	4	—	—	—	—
SJMP rel	短转移	2	3	—	—	—	—
JMP @A+DPTR	相对于 DPTR 间接转移	1	2	—	—	—	—
JZ rel	若 A=0 则转移	2	2/3#	—	—	—	—
JNZ rel	若 A≠0 则转移	2	2/3#	—	—	—	—
JC rel	若 C=1 则转移	2	2/3#	—	—	—	—
JNC rel	若 C≠1 则转移	2	2/3#	—	—	—	—
JB bit,rel	若直接位=1 则转移	3	3/4#	—	—	—	—
JNB bit,rel	若直接位=0 则转移	3	3/4#	—	—	—	—
JBC bit,rel	若直接位=1 则转移且清除	3	3/4#	—	—	—	—
CJNE A,direct,rel	直接数与 A 比较, 不等转移	3	3/4#	—	—	—	√
CJNE A,#data,rel	立即数与 A 比较, 不等转移	3	3/4#	—	—	—	√
CJNE @Ri,#data,rel	立即数与间接 RAM 比较, 不等转移	3	3/4#	—	—	—	√
CJNE Rn,#data,rel	立即数与寄存器比较不等转移	3	3/4#	—	—	—	√
DJNZ Rn,rel	寄存器减 1 不为 0 转移	2	2/3#	—	—	—	—
DJNZ direct,rel	直接字节减 1 不为 0 转移	3	3/4#	—	—	—	—
ACALL addr 11	绝对子程序调用	2	4	—	—	—	—
LCALL addr 16	子程序调用	3	5	—	—	—	—
RET	子程序调用返回	1	4	—	—	—	—
RETI	中断程序调用返回	1	4	—	—	—	—
NOP	空操作	1	1	—	—	—	—

备注:

i = 0,1;

rrr = 000~111;

* = a10a9a8;

= 跳转;

√ = 标志有影响;

- = 标志无影响;

0/1 =除法若 B=00H, 则 OV=1, 否则=0;

11. 配置说明

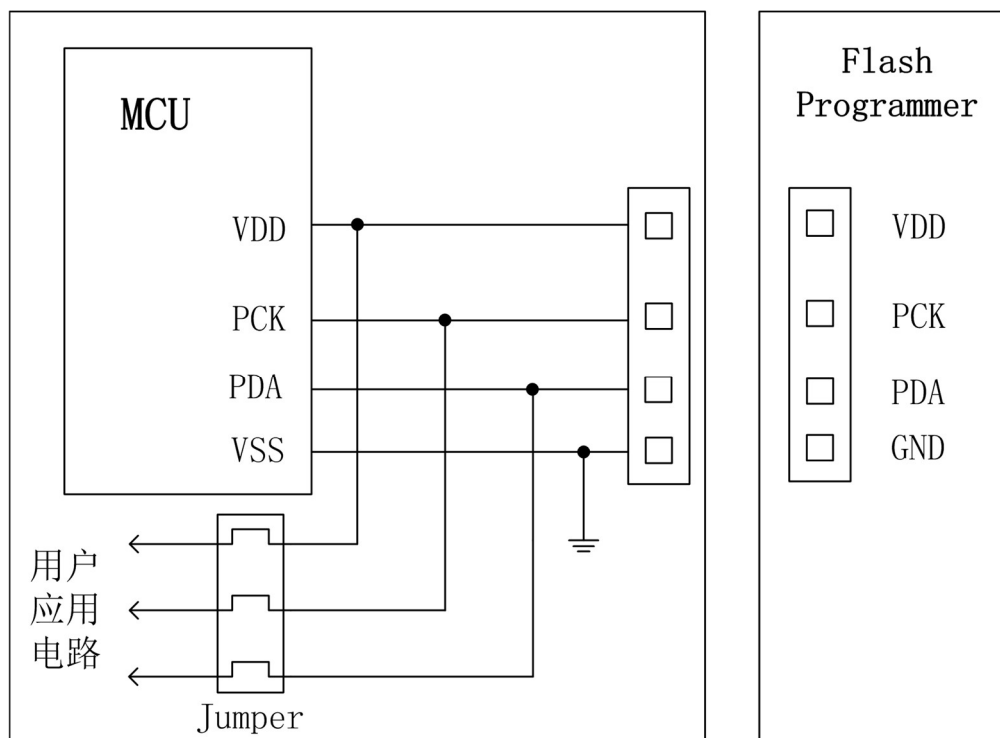
Filter	时钟滤波器	主频在 4M 及以下可选择 enable
RESTB	as IO	P1.7 可选择作为IO 口或者复位脚。 进入仿真模式时，reset pin 强制失效
	as RESET pin	
Fsys<2:0>	Fsys=Fosc (不可使用)	这些设置可以用 CPUM 寄存器重新设置
	Fsys=Fosc/2	
	Fsys=Fosc/4	
	Fsys=Fosc/8	
	Fsys=Fosc/16	
	Fsys=Fosc/32	
	Fsys=Fosc/64	
	Fsys=Fosc/128	
WDTEN	Always on	WDT 设置，和软件一起控制
	Disable when sleep	
	disable	
PWRT<1:0>	6ms	上电延时复位
	20ms	
	75ms	
	290ms	
LVD<2:0>	1.7v	选择 LVR 低压复位点
	2.2v	
	2.4v	
	2.7v	
	3.0v	
	4.2v	
时钟选择	HIRC	睡眠关闭，也可软件关闭
	HIRC & RTC	内部 HIRC 和 RTC32K 振荡，RTC 睡眠时不关闭
PIRSEN	P1 口驱动不增强	P1 口驱动增强位
	P1 口驱动增强	
PORSEN	P0 口驱动不增强	P0 口驱动增强位
	P0 口驱动增强	
DEBUG	Disable	烧录仿真使能，若有效，相关 IO 功能关闭。无效时，如果想 IO 口设置为输入口，则也可以进行芯片烧录仿真
	Enable	

12. 烧写编程与调试模式

12.1 在线编程示意图

在在线编程模式(ISP)中,通过4线接口编程器能完成所有Flash操作。烧写时,编程信号非常敏感,用户需要用3个跳线将编程引脚(VDD, PCK, PDA)从应用电路中分离出。

4线烧写如下图所示:



Flash 4 线烧写编程连接示意图

注:

- (1) 在程序接口连通前,必须断开跳线,从应用电路中分离编程引脚才能开始编程;
- (2) 编程器连接编程接口后,开始编程;
- (3) 在编程完成后断开编程器连线并连接跳线。
- (4) 烧写脚不接电容或者电容小于200pF。
- (5) 串接电阻在100欧姆到150欧姆。
- (6) 烧写数据线小于30cm。
- (7) VDD和GND之间电容小于220uF。

12.2 调试模式

MCU_CORE 内嵌片上调试功能(OCD),这为软件开发者提供了低成本调试方法。OCD 具有完整的调试程序控制流程,包括 4 个硬件断点、单步运行、全速运行、执行到光标处、执行到断点处、复位等内存访问。OCD 系统并不占用任何本地内存,也不和片上外设共享资源。

当 DEBUG 配置为 Enable，OCD 才能有效。OCD 系统使用两线串行接口，PDA 和 PCK，让目标设备和调试控制器建立通讯。PDA 是输入/ 输出引脚，调试时用作数据传输，PCK 是输入口，调试时传输数据同步用的时钟。MCU_CORE 的 OCD 和 ISP 功能是共用这 2 个引脚。PDA 与 P2.0 共用一个引脚，因此该引脚 I/O 功能或其他功能都不能使用；PCK 与 P1.1 共用一个引脚，因此该引脚 I/O 功能或其他功能都不能使用。

13. 电气参数

13.1 极限参数

参数	符号	参数范围	单位
电源电压	Vcc	-0.3~5.5	V
输入电压	Vi	-0.3~Vcc+0.3	V
输出电压	Vo	-0.3~Vcc+0.3	V
最大功耗（Vcc=5V,空载）	Pa	5	mW
工作温度	Topr	-40~+105	℃
贮存温度	Tstg	-40~+125	℃
端口静电放电电压	ESD（HBM）	6000	V
I/O 锁存电流	LU	300	mA

用户于使用中达到或超过“极限参数”中所列值有可能会造成芯片永久性损坏。本规格仅采用芯片在该条件内测试完成，如超过本规范部分不做保证。芯片长期处于绝对最大额定值条件下，可能会影响器件的可靠性。

13.2 DC 特性（TEMP=25℃、VDD=5V、全指令周期）

参数	符号	测试条件	最小值	典型值	最大值	单位
电源电压	Vcc	16MHz, Fosc/4	1.7	5	5.5	V
		16MHz, Fosc/2, 滤波 disable	2.2	5	5.5	V
静态电流	IDD	IRC 模式睡眠, 关闭 ILRC/ILRC1		1.4		uA
		IRC 模式睡眠, 关闭 ILRC, 开启 ILRC1		8.2		uA
		IRC 模式睡眠, 开启 ILRC, 关闭 ILRC1		3.5		uA
		IRC 模式睡眠, 开启 ILRC/ILRC1		10.2		uA
		IRC&RTC 模式, 睡眠		10		uA
工作电流	ICC	16MHz/4 分频, 端口输出方波		1.9		mA
		16MHz/2 分频, 端口输出方波		2.9		mA
		32KHz 低速, 关闭高速, 关闭 128KHz 低速, VDD=5V		80		uA
		32KHz 低速, 关闭高速, 关闭 128KHz 低速, VDD=3V		45		uA
		128Hz 低速, 关闭高速, 关闭 32KHz 低速, VDD=5V		130		uA
		128Hz 低速, 关闭高速, 关闭 32KHz 低速, VDD=3V		70		uA

深圳市晶曜微电子科技有限公司

SHEN ZHEN MESH STAR ELECTRONICS TECHNOLOGY CO., LTD

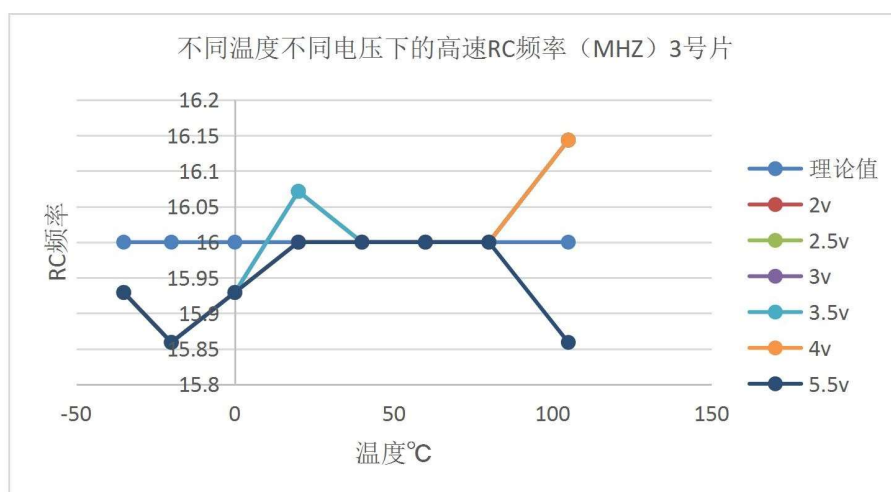
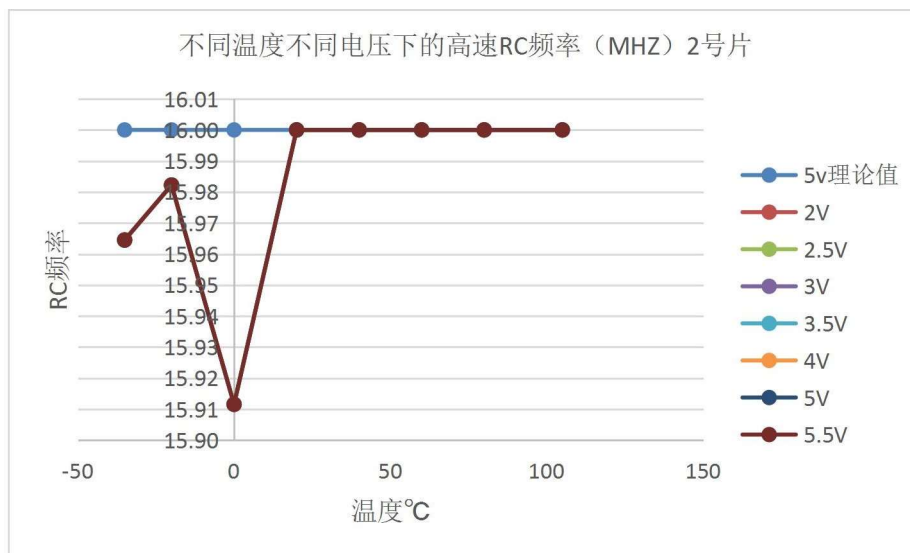
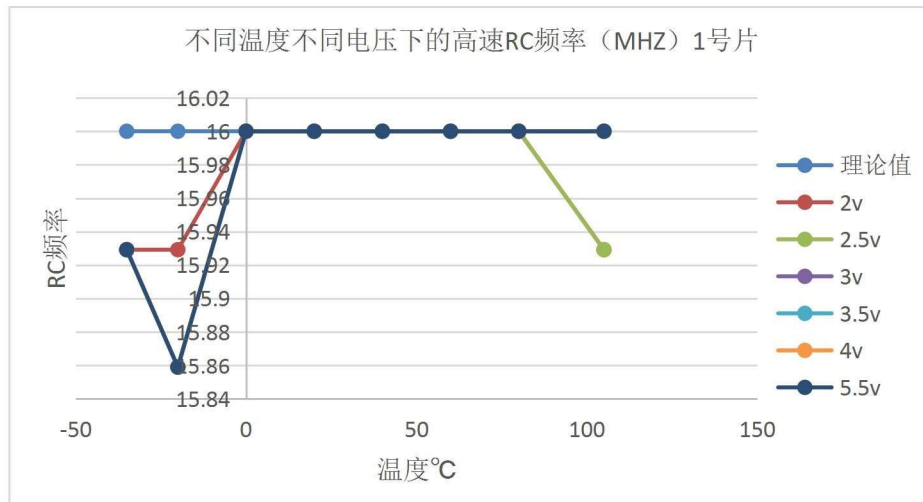
WL2610F003_MCU CORE

内置高频 RC	F_{IHRC}	温度: -40~105℃ 电压: 2~5.5V	15.76	16	16.24	MHz
内置低 LRC32K	F_{ILRC}	VDD=5V		32		KHz
		VDD=3V		64		
内置低 LRC128K	F_{ILRC1}	温度: -40~105℃ 电压: 2~5.5V	119	128	137	KHz
I/O 管脚输入高电压	V_{IH}	VDD=5V		3.6		V
I/O 管脚输入低电压	V_{IL}			1.0		V
I/O 管脚输入高电压	V_{IH}	VDD=3V		2.16		V
I/O 管脚输入低电压	V_{IL}			0.67		V
I/O 管脚 (P2.0 口 1 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		23.6		mA
	I_{OH}	$V_{OH}=0.8VDD$		11.7		mA
I/O 管脚 (P2.0 口 2 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		53.2		mA
	I_{OH}	$V_{OH}=0.8VDD$		29.2		mA
I/O 管脚 (P2.0 口 3 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		53.2		mA
	I_{OH}	$V_{OH}=0.8VDD$		29.2		mA
I/O 管脚 (P2.0 口 4 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		62.4		mA
	I_{OH}	$V_{OH}=0.8VDD$		38.5		mA
I/O 管脚 (P1.0 口 1 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		23.6		mA
	I_{OH}	$V_{OH}=0.8VDD$		11.1		mA
I/O 管脚 (P1.0 口 2 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		43.2		mA
	I_{OH}	$V_{OH}=0.8VDD$		21.1		mA
I/O 管脚 (P1.0 口 3 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		66.8		mA
	I_{OH}	$V_{OH}=0.8VDD$		26.7		mA
I/O 管脚 (P1.0 口 4 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		81.2		mA
	I_{OH}	$V_{OH}=0.8VDD$		33.3		mA
P2.4/P2.5 (1 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		28.4		mA
	I_{OH}	$V_{OH}=0.8VDD$		13.2		mA
P2.4/P2.5 (2 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		75.6		mA
	I_{OH}	$V_{OH}=0.8VDD$		44.4		mA
P2.4/P2.5 (3 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		75.6		mA
	I_{OH}	$V_{OH}=0.8VDD$		44.4		mA
P2.4/P2.5 (4 级驱动)	I_{OL}	$V_{OL}=0.2VDD$		105.2		mA
	I_{OH}	$V_{OH}=0.8VDD$		65.6		mA
端口上/下拉电阻	R	0.5VDD		60		K Ω
BOR disable	V	温度: -40~105℃	1.55	1.65	1.7	V
BOR 2.2V	V	温度: 25℃		2.15		V
BOR 2.4V	V	温度: 25℃		2.40		V
BOR 2.7V	V	温度: 25℃		2.73		V
BOR 3.0V	V	温度: 25℃		3.15		V
BOR 4.2V	V	温度: 25℃		4.50		V

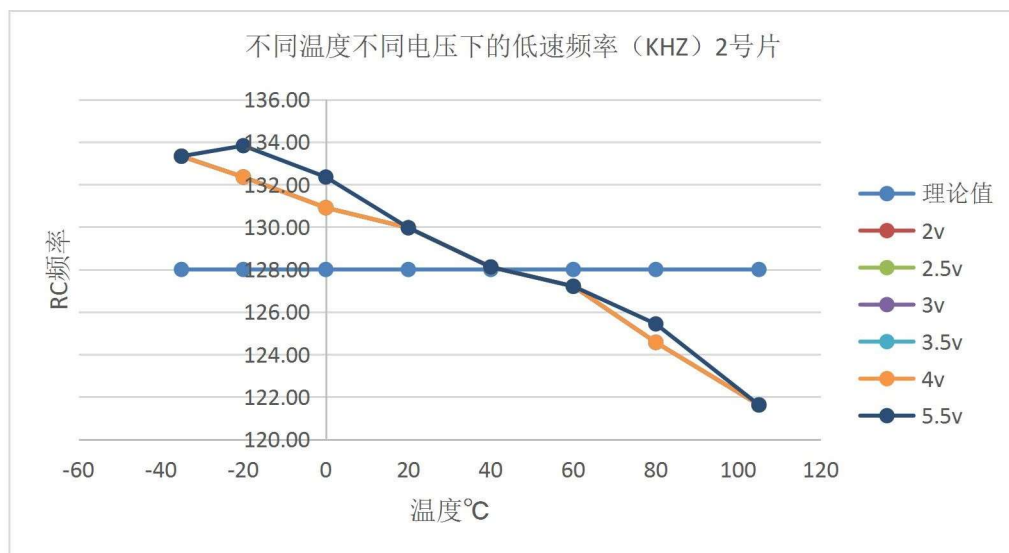
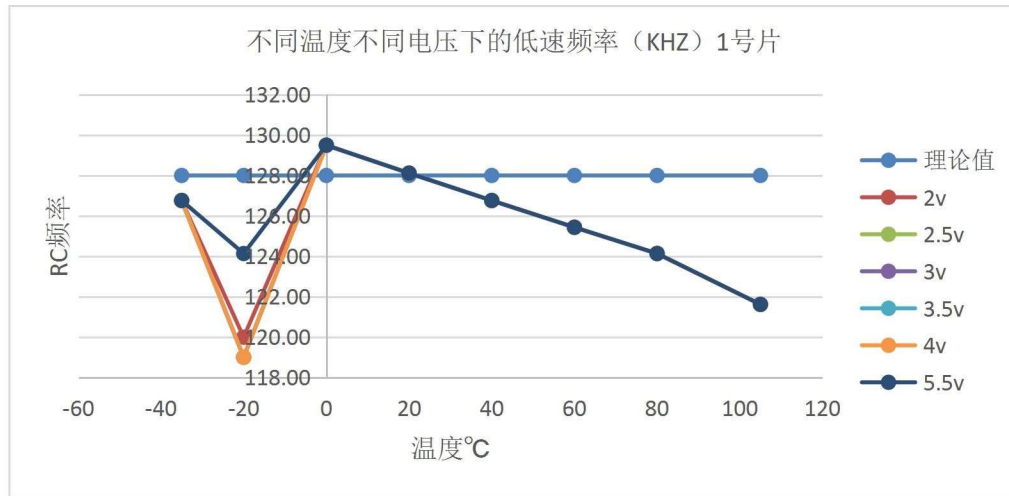
13.3 ADC 参数

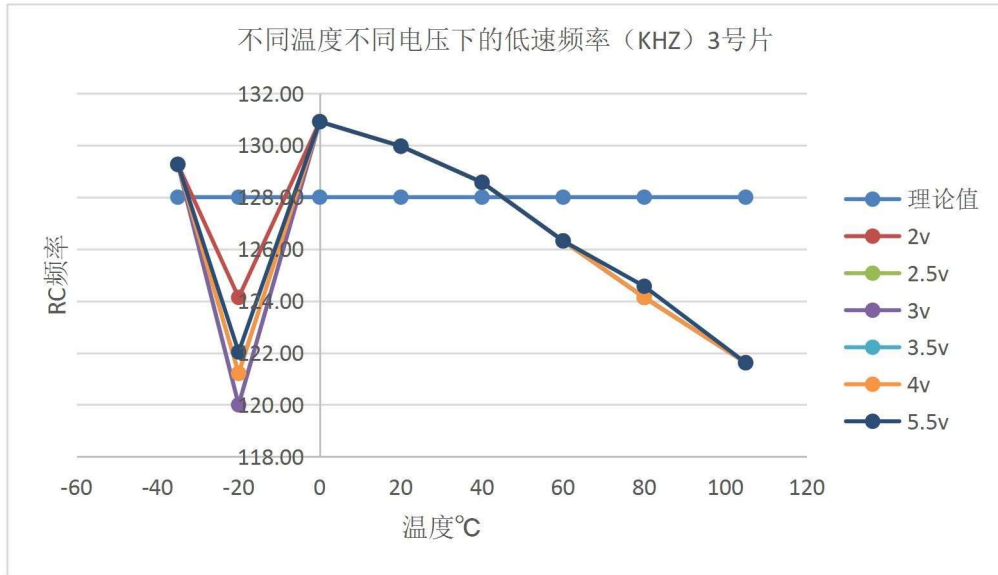
参数	符号	参数范围	单位
ADC 工作电压	V_{AVDD}	2.0~5.5	V
ADC 工作电流	I_{AVDD}	300	uA
模拟信号输入电压	V_{AIN}	0~ V_{AVDD}	V
分辨率	N_R	12	bit
有效精度	EA	10	bit
转换率	F_C	250	ksps
内部基准	V_{REF}	1.5/2/3 (±1%)	V

以下图标显示不同电源电压温度条件下 **IHRC** 的特性:

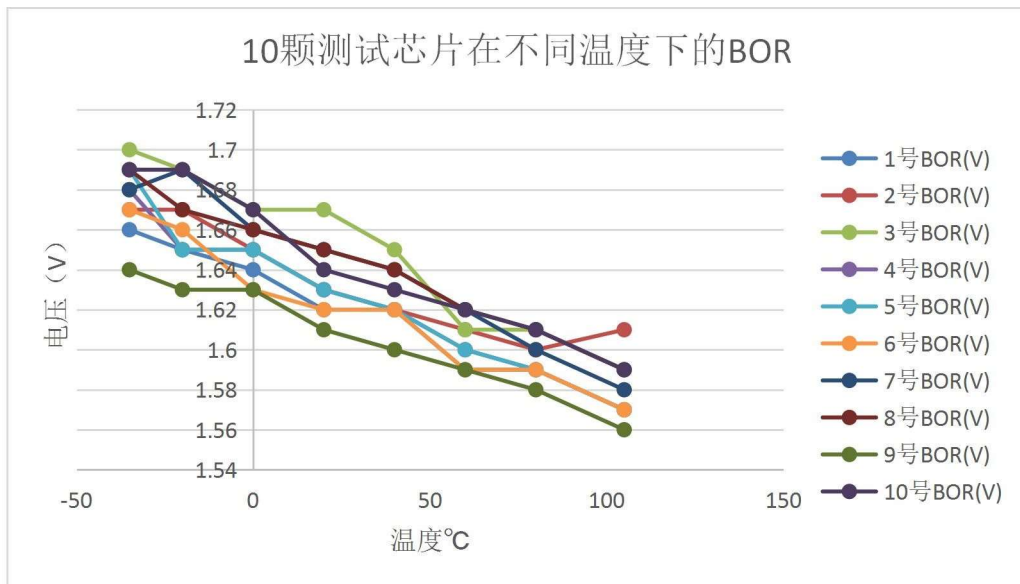


以下图标显示不同电源电压温度条件下 **ILRC1** 的特性:



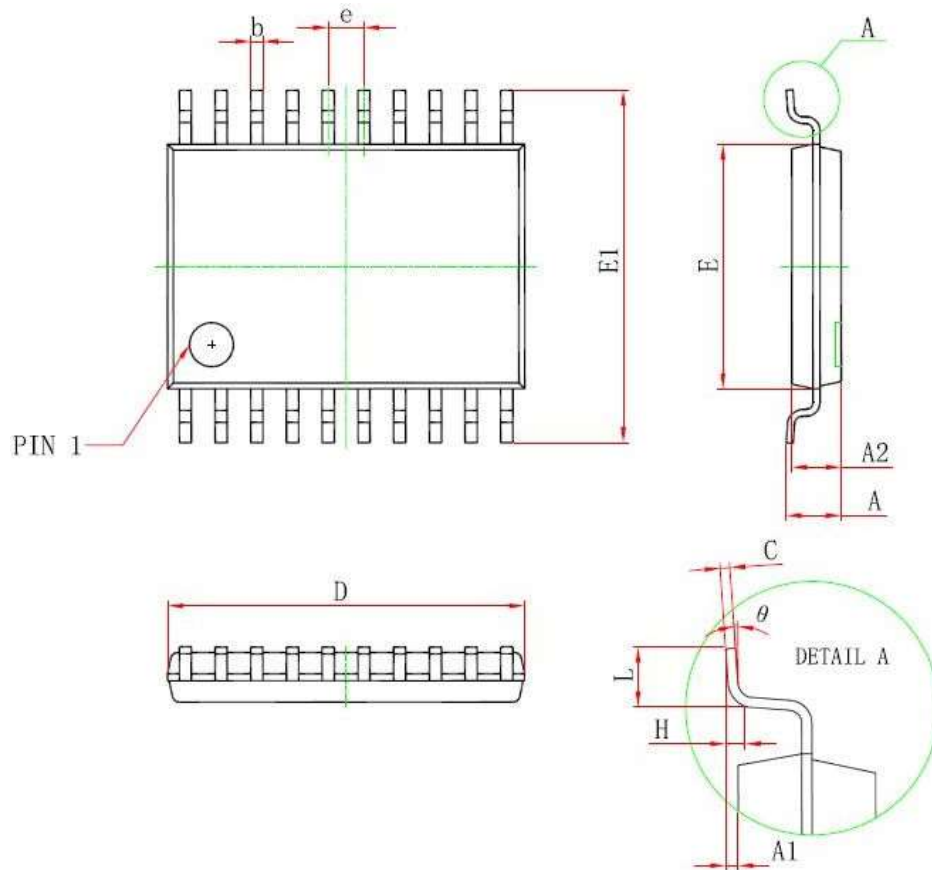


以下图标显示不同温度条件下 **BOR (disable)** 的特性:



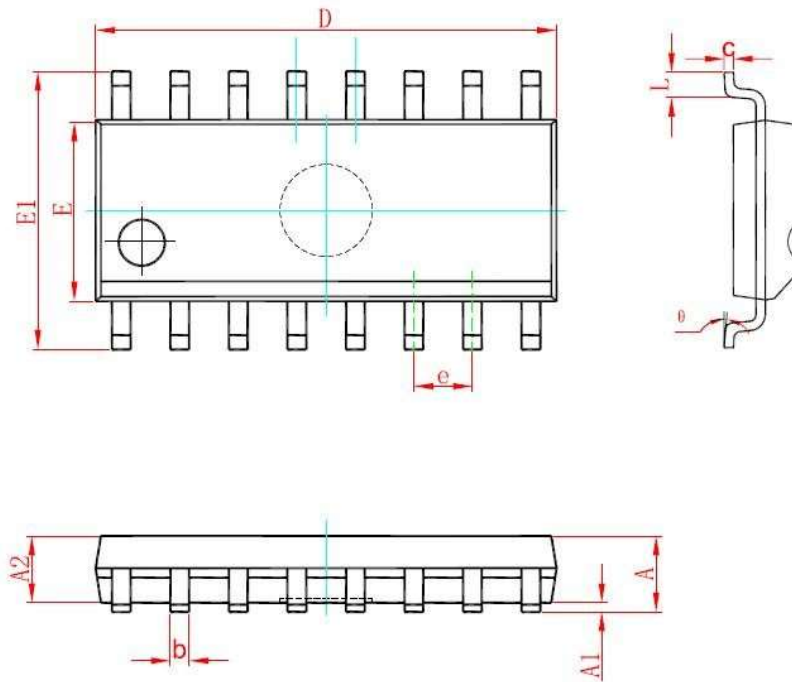
14. 封装及尺寸

14.1 TSSOP20 封装图及尺寸



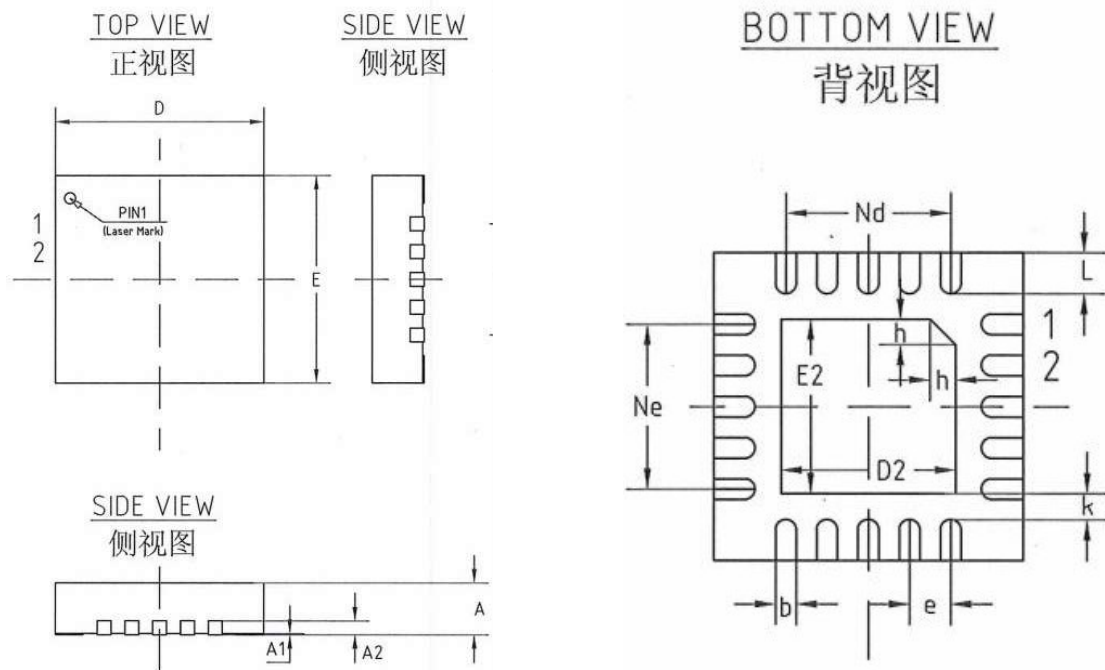
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
D	6.400	6.600	0.252	0.259
E	4.300	4.500	0.169	0.177
b	0.190	0.300	0.007	0.012
c	0.090	0.200	0.004	0.008
E1	6.250	6.550	0.246	0.258
A		1.200		0.047
A2	0.800	1.000	0.031	0.039
A1	0.050	0.150	0.002	0.006
e	0.65 (BSC)		0.026 (BSC)	
L	0.500	0.700	0.020	0.028
H	0.25(TYP)		0.01(TYP)	
θ	1°	7°	1°	7°

14.2 SOP16 封装图及尺寸



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.007	0.010
D	9.800	10.200	0.386	0.402
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

14.3 QFN20 封装图及尺寸



机械尺寸/mm			
字符 SYMBOL	最小值 MIN	典型值 NOMINAL	最大值 MAX
A	0.70	0.75	0.80
A1	-	0.02	0.05
A2	0.203 REF		
b	0.15	0.20	0.25
D	2.90	3.00	3.10
D2	1.55	1.65	1.75
E	2.90	3.00	3.10
E2	1.55	1.65	1.75
e	0.40 BSC		
K	0.175	0.275	0.375
L	0.35	0.40	0.45
h	0.20	0.25	0.30
Ne	1.60 BSC		
Nd	1.60 BSC		

15. 修正记录

版本	时间	内容
Ver1.00	2022.04.18	初版
Ver1.01	2022.05.18	完善
Ver1.02	2022.06.01	定稿
Ver1.03	2022.06.10	删除 BOOT 相关内容
Ver1.04	2022.06.24	调整 CRC 流程图
Ver1.05	2022.07.13	修改 IRC 频率为 16MHZ，限制 Fosc/1 使用条件
Ver1.06	2022.08.19	修改中断优先级说明，修改 SPI 时序图
Ver1.07	2022.11.10	修改 PWM 独立、互补模式部分错误说明
Ver1.08	2023.03.31	修改 IO 端口引脚框图
Ver1.09	2023.04.21	修改 IAP 说明描述，删除部分无关描述，添加 SOP16 脚位图及封装信息
Ver1.10	2023.04.24	添加 QFN20 脚位图及封装信息
Ver1.11	2023.04.25	修改 SOP16 脚位图
Ver1.12	2023.06.08	Page75 添加使用 AIN8(P1.5)作为采样脚使用说明