

HC18M002/003

数据手册

16/20 引脚 8 位
Touch+ADC 型 MTP 单片机

目录

1	产品简介	5
1.1	功能特性	5
1.2	系统框图	7
1.3	引脚配置	8
1.4	引脚描述	9
2	CPU	15
2.1	CPU 特性	15
2.2	CPU 相关寄存器.....	15
2.3	寻址模式	16
2.4	堆栈	17
3	存储器	18
3.1	程序存储器(ROM).....	18
3.2	掉电数据存储器(EEPROM).....	25
3.3	OPTION.....	27
3.4	数据存储器(RAM).....	29
3.5	特殊功能寄存器(SFR).....	30
4	系统时钟	34
4.1	系统时钟特性	34
4.2	系统时钟相关寄存器	35
5	系统工作模式.....	38
5.1	系统工作模式特性	38
5.2	高低频时钟切换	39
6	复位	40
6.1	复位特性	40
6.2	上电复位	40
6.3	BOR 复位	41
6.4	外部 RST 复位	41
6.5	看门狗(WDT)复位	41
6.6	复位相关寄存器	42
7	通用及复用 I/O.....	45
7.1	通用及复用 I/O 特性	45
7.2	I/O 模式.....	45
7.3	I/O 端口相关寄存器	46
7.4	外设功能映射相关寄存器	52
8	中断	57
8.1	中断特性	57

8.2	中断保护	58
8.3	多中断操作	59
8.4	中断相关寄存器	61
9	定时器/计数器	67
9.1	看门狗定时器	67
9.2	定时器/计数器 T0	69
9.3	定时器/计数器 T1	72
9.4	定时器 2	76
10	脉宽调制 PWM	79
10.1	PWM 特性	79
10.2	PWM 输出模式	80
10.3	PWM 相关寄存器	80
11	通用异步收发器 UART	87
11.1	UART 特性	87
11.2	工作方式	87
11.3	波特率	90
11.4	帧出错检测	90
11.5	UART1 相关寄存器	91
12	串行外部设备接口 SPI	94
12.1	SPI 特性	94
12.2	SPI 信号描述	94
12.3	SPI 时钟速率	95
12.4	SPI 功能框图	95
12.5	SPI 工作模式	95
12.6	SPI 传送形式	97
12.7	SPI 出错检测	98
12.8	SPI 中断	98
12.9	SPI 配置对照	99
12.10	SPI 相关寄存器	100
13	IIC 总线	102
13.1	IIC 特性	102
13.2	IIC 总线工作原理	102
13.3	总线上数据的有效性	103
13.4	总线上的信号	103
13.5	总线上数据初始格式	104
13.6	IIC 总线寻址约定	105
13.7	主机向从机读写 1 个字节数据的过程	105
13.8	IIC 工作模式	107
13.9	IIC 总线相关寄存器	110
14	模数转换 ADC	112
14.1	ADC 特性	112
14.2	ADC 相关寄存器	113

14.3	AD 转换时间	116
15	电容触摸按键 CTK.....	117
16	代码选项 OPTION.....	118
17	指令表	119
18	电气特性	120
18.1	极限参数	120
18.2	DC 特性.....	120
18.3	AC 特性.....	122
18.4	ADC 特性	122
18.5	MTP 内存特性	123
18.6	EEPROM 内存特性	123
18.7	BOR 检测电压特性	123
18.8	其他电气特性	124
19	封装尺寸	125
19.1	SOP16.....	125
19.2	SOP20.....	126
19.3	TSSOP20	127
20	版本记录	128

1 产品简介

HC18M002/003 是一颗采用高速低功耗 CMOS 工艺设计开发的增强型 8 位 ADC 单片机，内部有 4K*16bit MTP 程序存储器，512*8bit RAM，最多 18 个双向 I/O 口，3 个定时器/计数器，1 组 12 位带死区控制互补 PWM，1 个 UART，1 个 SPI，1 个 IIC，1 个外部中断，PORTB 端口支持电平中断，最多 8 通道电容触摸按键检测，最多 11+1 路 12 位 ADC，内置高性能的电源稳压电路，四种系统工作模式（高频、低频、绿色、休眠）和 12 个中断源。

1.1 功能特性

- ◆ CPU
 - 36条高性能精简指令
 - 8级堆栈缓存器
 - 立即、直接和两组间接寻址模式
 - 16位RDT查表
- ◆ ROM
 - 4K*16bit ROM
- ◆ RAM
 - 512 Bytes RAM
- ◆ EEPROM
 - 128 Bytes
- ◆ 时钟
 - 内部高精度 32MHz RC
 - 内部 32KHz RC
- ◆ 多种复位方式
 - 上电复位（POR）
 - 多级低电压复位（BOR）
 - 4.2/3.9/3.6/3.0/2.6/2.4/2.0
 - 看门狗（WDT）复位
- ◆ I/O
 - 最多18个双向I/O口
 - 上下拉、输入输出
 - 外设功能引脚部分映射
- ◆ 中断
 - 12个中断源
 - 1个外部中断
 - PORTB口支持端口电平变化中断
- ◆ 定时器/计数器
 - Timer0: 带有预分频器的8位定时器/计数器
 - Timer1: 带有预分频器的16位定时器/计数器
 - Timer2: 带有8位周期寄存器的8位定时器
- ◆ PWM
 - 1组12位带死区控制互补PWM
 - 可配置为 2 路独立输出
 - 可当定时器使用
 - 具有故障检测功能
- ◆ 通讯模块
 - 1 个 UART
 - 1 个 SPI
 - 1 个 IIC(仅主机模式)
- ◆ ADC 检测电路
 - 支持最多11+1ch 12位ADC检测
 - ADC参考电压可选内部 VREF(1.3V/2V/3V/4V)、外部VREF、VDD
- ◆ 电容触摸按键（CTK）检测电路
 - 支持最多8ch电容触摸按键检测
 - 灵活的灵敏度调节能力
- ◆ 省电模式
 - 低频模式
 - 绿色模式
 - 休眠模式
- ◆ 工作条件
 - 宽电压 2.0V~5.5V
 - 温度范围-40°C~+85°C
- ◆ 封装类型
 - SOP16
 - SOP20
 - TSSOP20

✓ 选型表

Product	ROM Bytes	RAM Bytes	EEPROM Bytes	Max Freq	I/O	TK	A/D	Timer	PWM	INT	UART	IIC	SPI	WDT	Package
HC18M002	4K*16bit	512	128	8M2T	14	6	9+1	8bit*2 16bit*1	12bit*1 组	1	1	1	1	1	SOP16
HC18M003	4K*16bit	512	128	8M2T	18	8	11+1	8bit*2 16bit*1	12bit*1 组	1	1	1	1	1	SOP20
HC18M003	4K*16bit	512	128	8M2T	18	/	8+1	8bit*2 16bit*1	12bit*1 组	1	1	1	1	1	TSSOP20

HC18M002/003 使用注意事项:

- 1、为保证系统的稳定性，建议在 VDD 和 GND 之间接 2 个电容（ $0.1\mu\text{F}+\geq 10\mu\text{F}$ ）；
- 2、当使用 ADC 模块时，内部参考电压选择为 2V 时，VDD 电压须高于 2.7V；内部参考电压选择为 3V 时，VDD 电压须高于 3.5V；内部参考电压选择为 4V 时，VDD 电压须高于 4.5V；
- 3、内部高频起振时间较长，约 1ms，如使用时钟切换请注意；
- 4、PORTB 电平变化中断中，在清零 RBIF 之前必须执行 PORTB 端口读操作；
- 5、仿真时请勿操作 PORTC4（VPP）端口，否则有可能导致芯片异常损坏；
- 6、仿真时无法关闭内部高频 RC，因此仿真时 HSRCDY 会一直为 1；
- 7、使用 EEP 读时，需要在 while(RD)后增加一个 10us 以上的延时，仅 CPU 时钟 $\geq 8\text{M}$ 时需要此操作。
- 8、使用 CTK 时，需要同时配置 ADC 的参考电压（默认电压 2V），当 ADC 参考电压选择 VDD 及 4V 时，需要保证芯片 VDD 电压 $\geq 4.5\text{V}$ ，当 ADC 参考电压选择 3V/2V/1.3V 时，芯片 VDD 电压 $\geq 3.5\text{V}$ 。
- 9、仿真时时钟的 2T/4T 为随机值，使用时请注意。
- 10、从 0V 到芯片工作电压，要求时间在 50ms 以内。

1.2 系统框图

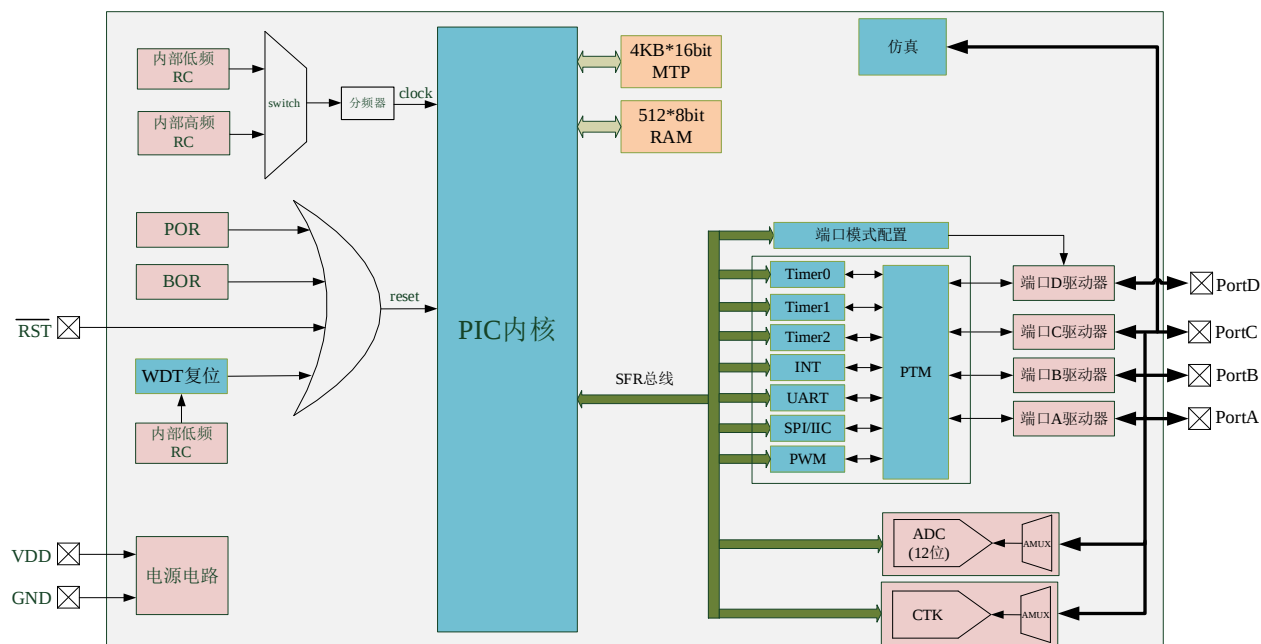


Figure 1-1 系统框图

1.3 引脚配置

1.3.1 SOP16 引脚配置

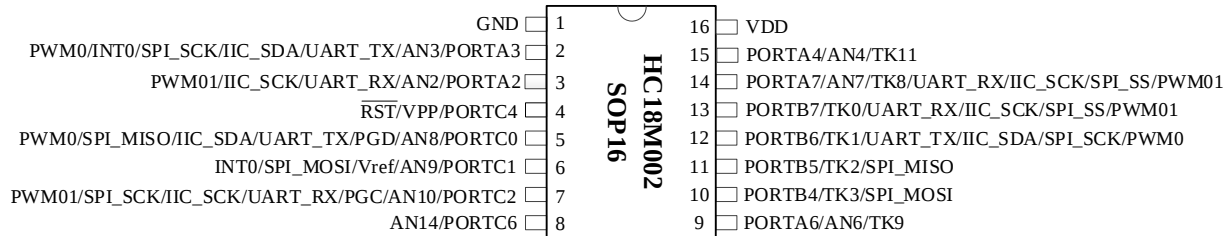


Figure 1-2 SOP16 引脚配置图

1.3.2 SOP20 引脚配置

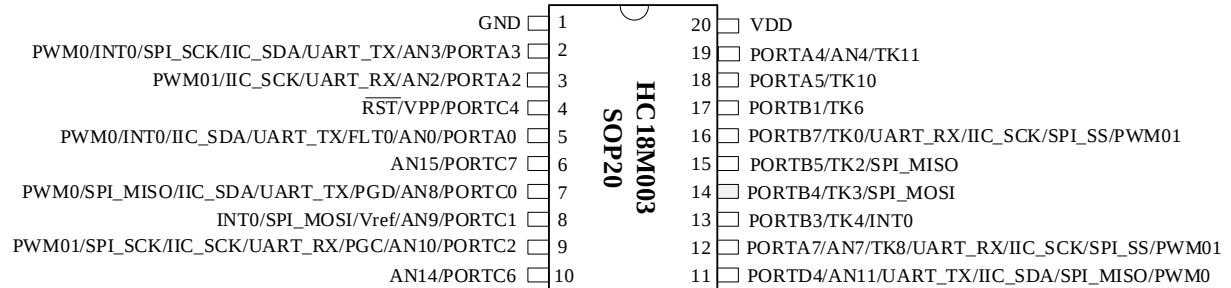


Figure 1-3 SOP20 引脚配置图

1.3.3 TSSOP20 引脚配置

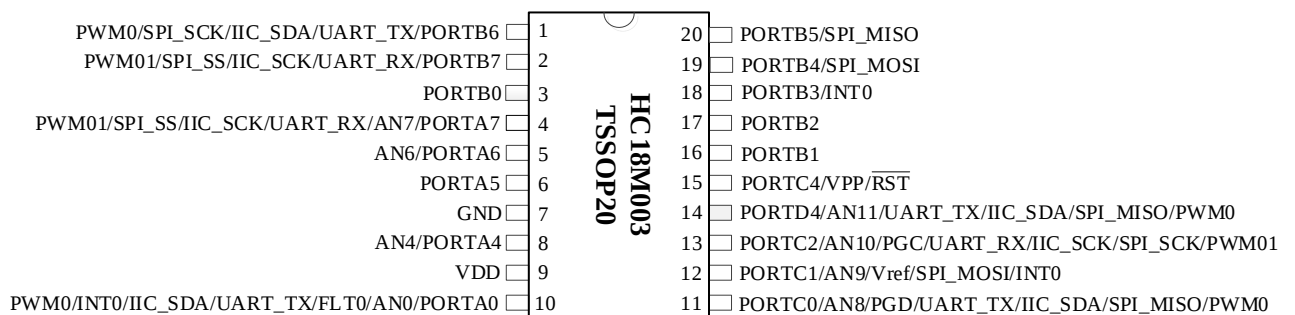


Figure 1-4 TSSOP20 引脚配置图

1.4 引脚描述

1.4.1 SOP16 引脚描述

脚位	名称	类型	说明
1	GND	P	电源地
2	PORTA3 AN3 UART_TX IIC_SDA SPI_SCK INT0 PWM0	I/O AN O I/O I/O I O	输入/输出口 ADC3 输入口 UART 数据传输口 IIC 数据口 SPI 时钟口 外部中断 0 输入引脚 PWM0 输出口
3	PORTA2 AN2 UART_RX IIC_SCK PWM01	I/O AN I/O I/O O	输入/输出口 ADC2 输入口 UART 接收口 IIC 时钟口 PWM01 输出口
4	PORTC4 $\overline{\text{RST}}$ VPP	I I I	输入口 外部复位输入口 编程高压烧录口
5	PORTC0 AN8 PGD UART_TX IIC_SDA SPI_MISO PWM0	I/O AN I/O O I/O I/O O	输入/输出口 ADC8 输入口 编程数据输入/输出口 UART 数据传输口 IIC 数据口 SPI 的数据口，主机的输入和从机的输出 PWM01 输出口
6	PORTC1 AN9 Vref SPI_MOSI INT0	I/O AN AN I/O I	输入/输出口 ADC9 输入口 ADC 参考电压输入口 SPI 的数据口，主机的输出和从机的输入 外部中断 0 输入引脚
7	PORTC2 AN10 PGC UART_RX IIC_SCK SPI_SCK PWM01	I/O AN I I/O I/O I/O O	输入/输出口 ADC10 输入口 编程时钟输入口 UART 接收口 IIC 时钟口 SPI 时钟口 PWM01 输出口
8	PORTC6 AN14	I/O AN	输入/输出口 ADC14 输入口
9	PORTA6	I/O	输入/输出口

	AN6 TK9	AN AN	ADC6 输入口 触摸通道 9
10	PORTB4 TK3 SPI_MOSI	I/O AN I/O	输入/输出口 触摸通道 3 SPI 的数据口，主机的输出和从机的输入
11	PORTB5 TK2 SPI_MISO	I/O AN I/O	输入/输出口 触摸通道 2 SPI 的数据口，主机的输入和从机的输出
12	PORTB6 TK1 UART_TX IIC_SDA SPI_SCK PWM0	I/O AN O I/O I/O O	输入/输出口 触摸通道 1 UART 数据传输口 IIC 数据口 SPI 时钟口 PWM0 输出口
13	PORTB7 TK0 UART_RX IIC_SCK SPI_SS PWM01	I/O AN I/O I/O I O	输入/输出口 触摸通道 0 UART 接收口 IIC 时钟口 SPI 片选口 PWM01 输出口
14	PORTA7 AN7 TK8 UART_RX IIC_SCK SPI_SS PWM01	I/O AN AN I/O I/O I O	输入/输出口 ADC7 输入口 触摸通道 8 UART 接收口 IIC 时钟口 SPI 片选口 PWM01 输出口
15	PORTA4 AN4 TK11	I/O AN AN	输入/输出口 ADC4 输入口 触摸通道 11
16	VDD	P	电源输入口

1.4.2 SOP20 引脚描述

脚位	名称	类型	说明
1	GND	P	电源地
2	PORTA3 AN3 UART_TX IIC_SDA SPI_SCK INT0 PWM0	I/O AN O I/O I/O I O	输入/输出口 ADC3 输入口 UART 数据传输口 IIC 数据口 SPI 时钟口 外部中断 0 输入引脚 PWM0 输出口
3	PORTA2 AN2 UART_RX IIC_SCK PWM01	I/O AN I/O I/O O	输入/输出口 ADC2 输入口 UART 接收口 IIC 时钟口 PWM01 输出口
4	PORTC4 $\overline{\text{RST}}$ VPP	I I I	输入口 外部复位输入口 编程高压烧录口
5	PORTA0 AN0 FLT0 UART_TX IIC_SDA INT0 PWM0	I/O AN I O I/O I O	输入/输出口 ADC0 输入口 PWM0 故障检测输入引脚 UART 数据传输口 IIC 数据口 外部中断 0 输入引脚 PWM0 输出口
6	PORTC7 AN15	I/O AN	输入/输出口 ADC15 输入口
7	PORTC0 AN8 PGD UART_TX IIC_SDA SPI_MISO PWM0	I/O AN I/O O I/O I/O O	输入/输出口 ADC8 输入口 编程数据输入/输出口 UART 数据传输口 IIC 数据口 SPI 的数据口，主机的输入和从机的输出 PWM01 输出口
8	PORTC1 AN9 Vref SPI_MOSI INT0	I/O AN AN I/O I	输入/输出口 ADC9 输入口 ADC 参考电压输入口 SPI 的数据口，主机的输出和从机的输入 外部中断 0 输入引脚
9	PORTC2 AN10 PGC UART_RX	I/O AN I I/O	输入/输出口 ADC10 输入口 编程时钟输入口 UART 接收口

	IIC_SCK SPI_SCK PWM01	I/O I/O O	IIC 时钟口 SPI 时钟口 PWM01 输出口
10	PORTC6 AN14	I/O AN	输入/输出口 ADC14 输入口
11	PORTD4 AN11 UART_TX IIC_SDA SPI_MISO PWM0	I/O AN O I/O I/O O	输入/输出口 ADC11 输入口 UART 数据传输口 IIC 数据口 SPI 的数据口, 主机的输入和从机的输出 PWM0 输出口
12	PORTA7 AN7 TK8 UART_RX IIC_SCK SPI_SS PWM01	I/O AN AN I/O I/O I O	输入/输出口 ADC7 输入口 触摸通道 8 UART 接收口 IIC 时钟口 SPI 片选口 PWM01 输出口
13	PORTB3 TK4 INT0	I/O AN I	输入/输出口 触摸通道 4 外部中断 0 输入引脚
14	PORTB4 TK3 SPI_MOSI	I/O AN I/O	输入/输出口 触摸通道 3 SPI 的数据口, 主机的输出和从机的输入
15	PORTB5 TK2 SPI_MISO	I/O AN I/O	输入/输出口 触摸通道 2 SPI 的数据口, 主机的输入和从机的输出
16	PORTB7 TK0 UART_RX IIC_SCK SPI_SS PWM01	I/O AN I/O I/O I O	输入/输出口 触摸通道 0 UART 接收口 IIC 时钟口 SPI 片选口 PWM01 输出口
17	PORTB1 TK6	I/O AN	输入/输出口 触摸通道 6
18	PORTA5 TK10	I/O AN	输入/输出口 触摸通道 10
19	PORTA4 AN4 TK11	I/O AN AN	输入/输出口 ADC4 输入口 触摸通道 11
20	VDD	P	电源输入口

1.4.3 TSSOP20 引脚描述

脚位	名称	类型	说明
1	PORTB6	I/O	输入/输出口
	UART_TX	O	UART 数据传输口
	IIC_SDA	I/O	IIC 数据口
	SPI_SCK	I/O	SPI 时钟口
	PWM0	O	PWM0 输出口
2	PORTB7	I/O	输入/输出口
	UART_RX	I/O	UART 接收口
	IIC_SCK	I/O	IIC 时钟口
	SPI_SS	I	SPI 片选口
	PWM01	O	PWM01 输出口
3	PORTB0	I/O	输入/输出口
4	PORTA7	I/O	输入/输出口
	AN7	AN	ADC7 输入口
	UART_RX	I/O	UART 接收口
	IIC_SCK	I/O	IIC 时钟口
	SPI_SS	I	SPI 片选口
5	PORTA6	I/O	输入/输出口
	AN6	AN	ADC6 输入口
6	PORTA5	I/O	输入/输出口
7	GND	P	电源地
8	PORTA4	I/O	输入/输出口
	AN4	AN	ADC4 输入口
9	VDD	P	电源输入口
10	PORTA0	I/O	输入/输出口
	AN0	AN	ADC0 输入口
	FLT0	I	PWM0 故障检测输入引脚
	UART_TX	O	UART 数据传输口
	IIC_SDA	I/O	IIC 数据口
	INT0	I	外部中断 0 输入引脚
	PWM0	O	PWM0 输出口
11	PORTC0	I/O	输入/输出口
	AN8	AN	ADC8 输入口
	PGD	I/O	编程数据输入/输出口
	UART_TX	O	UART 数据传输口
	IIC_SDA	I/O	IIC 数据口
	SPI_MISO	I/O	SPI 的数据口，主机的输入和从机的输出
	PWM0	O	PWM01 输出口
12	PORTC1	I/O	输入/输出口
	AN9	AN	ADC9 输入口
	Vref	AN	ADC 参考电压输入口

	SPI_MOSI INT0	I/O I	SPI 的数据口，主机的输出和从机的输入 外部中断 0 输入引脚
13	PORTC2 AN10 PGC UART_RX IIC_SCK SPI_SCK PWM01	I/O AN I I/O I/O I/O O	输入/输出口 ADC10 输入口 编程时钟输入口 UART 接收口 IIC 时钟口 SPI 时钟口 PWM01 输出口
14	PORTD4 AN11 UART_TX IIC_SDA SPI_MISO PWM0	I/O AN O I/O I/O O	输入/输出口 ADC11 输入口 UART 数据传输口 IIC 数据口 SPI 的数据口，主机的输入和从机的输出 PWM0 输出口
15	PORTC4 $\overline{\text{RST}}$ VPP	I I I	输入口 外部复位输入口 编程高压烧录口
16	PORTB1	I/O	输入/输出口
17	PORTB2	I/O	输入/输出口
18	PORTB3 INT0	I/O I	输入/输出口 外部中断 0 输入引脚
19	PORTB4 SPI_MOSI	I/O I/O	输入/输出口 SPI 的数据口，主机的输出和从机的输入
20	PORTB5 SPI_MISO	I/O I/O	输入/输出口 SPI 的数据口，主机的输入和从机的输出

注：I=输入，O=输出，I/O=输入/输出，P=电源，AN=模拟输入输出。

2 CPU

2.1 CPU 特性

HC18M002/003 CPU内核包括:

- 2T/4T 时钟模式
- 8 级堆栈
- 程序存储器
- 寻址方式
- 数据存储器

2.2 CPU 相关寄存器

2.2.1 STATUS 寄存器

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R	R	R/W	R/W	R/W
复位值	0	0	0	1	1	x	X	X
位符号	-	-	RP0	TO	PD	Z	DC	C

位编号	位符号	说明
7-6	-	保留位
5	RP0	BANK 选择位 0: Bank0 1: Bank1
4	TO	超时位 0: 发生了 WDT 溢出 1: 上电、执行了 CLRWDT 指令或 SLEEP 指令
3	PD	掉电位 0: 执行了 SLEEP 指令 1: 上电或执行了 CLRWDT 指令
2	Z	结果为零位 0: 算术或逻辑运算的结果不为零 1: 算术或逻辑运算的结果为零
1	DC	半进位/借位位 0: 加法运算时低四位没有进位/减法运算时有向高四位借位 1: 加法运算时低四位有进位/减法运算时没有向高四位借位
0	C	进位/借位位 0: 加法运算时没有进位/减法运算时有借位发生/移位后移出逻辑 0 1: 加法运算时有进位/减法运算时没有借位发生/移位后移出逻辑 1

2.2.2 PC 寄存器

程序计数器（PC）为12位宽，低字节来自可读写的PCL寄存器，高字节（PC[11:8]）不可读写，可通过PCLATH 寄存器间接写入。

PCL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

PCLATH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PCH11	PCH10	PCH9	PCH8

2.3 寻址模式

HC18M002/003共有三种寻址方式：立即寻址、直接寻址和间接寻址模式

2.3.1 立即寻址

立即数参与运算的寻址方式

➤ 例：立即寻址

ADDLW 06h ;W 的内容加 6，结果放入 W

2.3.2 直接寻址

寄存器参与运算的寻址方式

➤ 例：直接寻址

MOVWF OPTION ;W 的内容装入 OPTION

2.3.3 间接寻址

由指针 FSR 指向的寄存器参与运算的寻址方式。INDF 寄存器不是物理寄存器，对 INDF 寄存器操作可以实现间接寻址

➤ 例：利用间接寻址对 0X100~0X1FF 通用数据存储器进行清零

```

MOV LW                      00h                      ;清零 0X100~0X1FF
MOV WF                      FSR0L
MOV LW                      0x01
MOV WF                      FSR0H                      ;FSR 指向 100h 地址
NEXTBYTE:                      CLRF                      INDF0                      ;对 FSR 指向的数据存储器清零
                              INCF                      FSR0L,F                      ;FSR + 1,指向下一个地址
                              MOV LW                      00h                      ;注意这里的边界值为欲操作 RAM 最大地址 + 1
                              XOR WF                      FSR0L,W                      ;利用间接寻址，注意意外指向特殊寄存器的情况

```

BTFSS STATUS,Z
GOTO NEXTBYTE ;FSR 的值小于 1FFh,循环清零下一个地址

CONTINUE: ... ;完成清零操作

2.4 堆栈

HC18M002/003 具有一个 8 级深度的硬件堆栈。当执行 CALL 指令或由于中断导致程序跳转时，PC 值会被压入堆栈;当执行 RETURN、RETLW 或 RETFIE 指令时，PC 值从堆栈弹出。

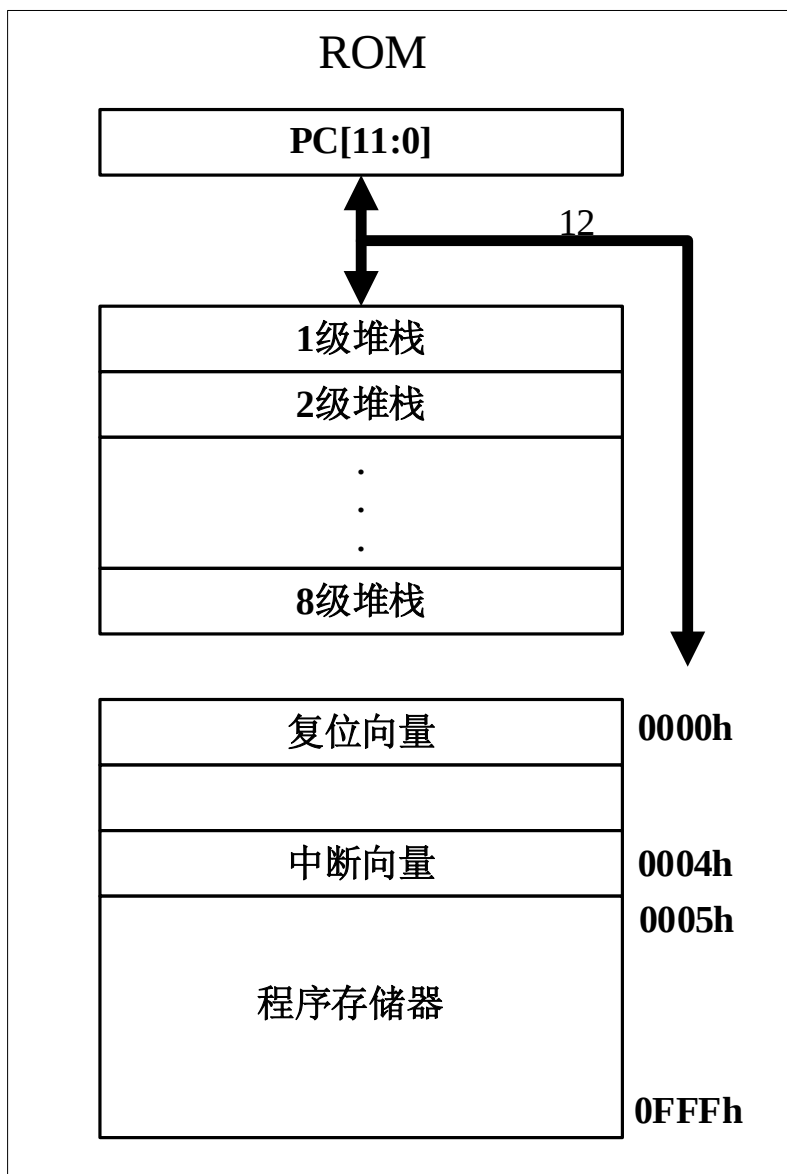


Figure 2-1 堆栈

注:

压栈级数请勿超过 8 级，超过 8 级压栈将导致堆栈溢出，溢出后堆栈指针循环，新的压栈将覆盖原堆栈内容。

3 存储器

3.1 程序存储器(ROM)

HC18M002/003 具有 4K×16 位的 MTP 存储器，下图给出了程序存储器的映射。访问超出物理地址以外的单元时，会导致返回到地址最低单元。

复位向量是 0000H，中断向量是 0004H。

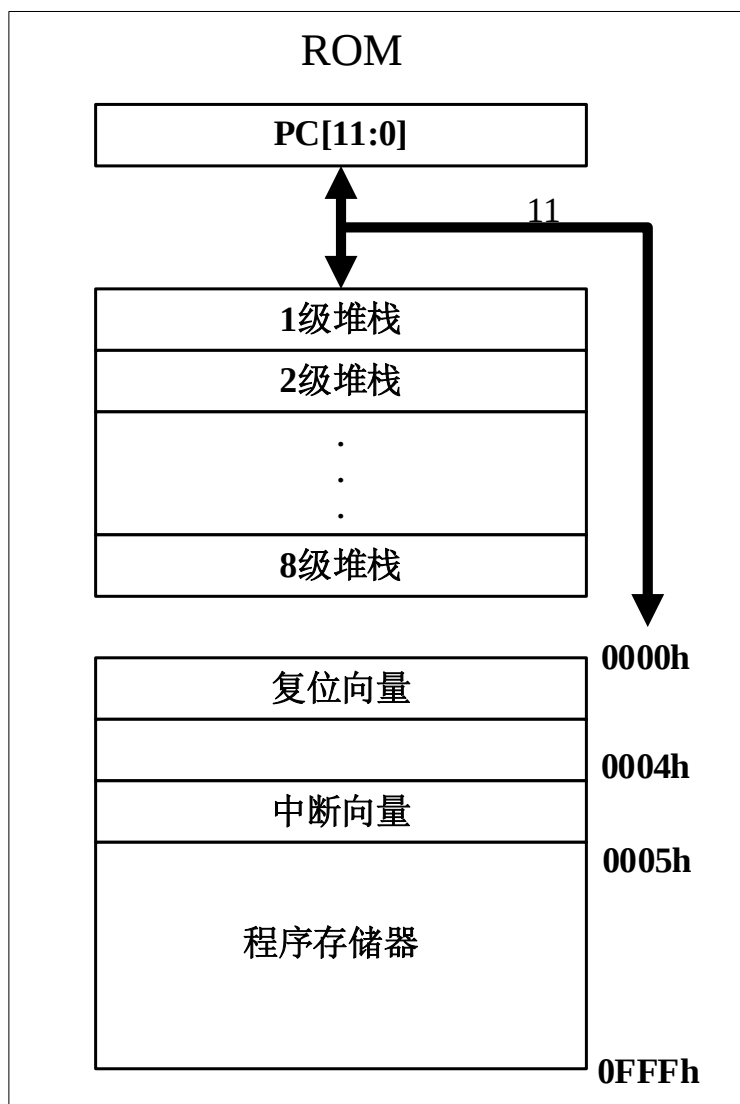


Figure 3-1 MTP 结构图

3.1.1 MTP ICP 操作

用户可以通过 HC-ICD Pro 仿真器的双线方式对 MCU 进行仿真和编程，当 MCU 已经焊在用户板上后，如果用户采用上电复位的方式，只需要连接五根线（VDD、GND、PGC、PGD、VPP），用户系统必须断电，由仿真器提供电源。

另外，因为编程信号非常敏感，用户需要用 4 个跳线将编程引脚（VDD、PGC、PGD、VPP）从应用电路中分离出来，如下图所示。另外，如果使用外部复位引脚进入，也需要将外部复位引脚进行跳线分离。

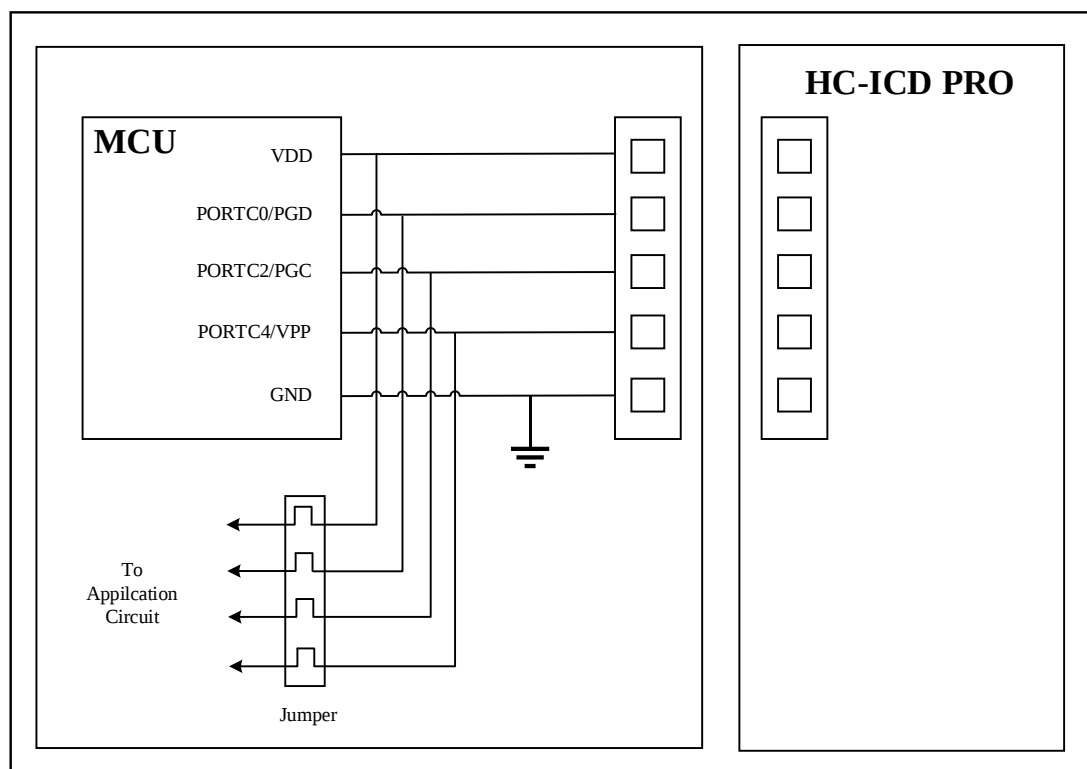


Figure 3-2 HC-ICD Pro 编程硬件连接

3.1.2 复位向量(0000H)

- 上电复位
- 低电压复位
- 看门狗复位
- 外部复位

发生上述任一种复位后，程序将从0000H处重新开始执行，系统寄存器也都将恢复为默认值。

3.1.3 中断向量(0004H)

中断向量地址为0004H。一旦有中断响应，程序计数器PC的当前值就会存入堆栈缓存器并跳转到0004H开始执行中断服务程序。中断服务子程序中需根据程序需要对相应状态寄存器进行适当的断点保护和恢复。下面的示例程序说明了如何编写中断服务程序。

- 例：中断子程序的编写

```

                                ORG      0000H
                                GOTO      MAIN
                                ORG      0004H
                                GOTO      INT_SERVICE

MAIN:
    ...

INT_SERVICE:
    MOVWF    W_TEMP           ;保存W
    SWAPF    STATUS,W
    MOVWF    STATUS_TEMP     ;保存STATUS
    MOVF     PCLATH,W
    MOVWF    PCLATH_TEMP     ;保存PCLATH
    ...
    MOVF     PCLATH_TEMP,W
    MOVWF    PCLATH           ;恢复PCLATH
    SWAPF    STATUS_TEMP,W
    MOVWF    STATUS           ;恢复STATUS
    SWAPF    W_TEMP,F
    SWAPF    W_TEMP,W        ;恢复W
    RETFIE                    ;退出中断
    ...

END
```

对于编写中断服务程序，需要以下几个要点需注意：

1. 中断入口地址为 0x04，响应中断后，程序自动跳转到 0x04 开始执行。
2. 中断服务程序需首先对相应的寄存器进行保护。
3. 保存系统寄存器时使用到的 RAM 建议定义在所有 BANK 均映射的位置。
4. 中断服务子程序返回前对保护的寄存器进行恢复，注意恢复顺序，对 W 必须使用 SWAPF。
5. 程序中使能两个以上的中断源时，程序需对发生中断的中断源进行判断，从而执行相应的服务程序。

6. 需要软件清空对应的中断标志。
7. RETFIE 指令将自动使能 GIE，请勿在中断服务子程序中用其它指令使能 GIE，以免造成中断响应混乱。

3.1.4 查表

方式一：

利用 ADDWF PCL, F 和 RETLW 指令实现数据表，因为以 PCL 为目的操作数的运算将改变程序指针（PC）值，其具体操作为 PC 的低 8 位为 ALU 的运算结果，PC 的高 4 位将从 PC 高位缓冲器 PCLATH 中获得。 如下是数据表实现的一个例子。

➤ 例：数据查表

```

...
    MOVLW    HIGH TAB1    ;获得数据表地址高8位（内部宏指令）
    MOVWF    PCLATH       ;表地址高位赋给PCLATH
    MOVF     TABBUF,W     ;获得表数据偏移量，调用前赋值。
    CALL     TAB1         ;调用数据表
...
TAB1:
    ORG      100H
    ADDWF    PCL,F        ;表头运算
    RETLW    DATA0_TAB1  ;W=0对应数据
    RETLW    DATA1_TAB1  ;W=1对应数据
    RETLW    DATA2_TAB1  ;W=2对应数据
    ...
    RETLW    DATAFE_TAB1 ;W=0XFE 对应数据

```

对于数据查表的编程，需注意：

1. 数据表宽度：8 位
2. 当 PCL 与 W 的加运算有进位时，进位将被舍弃，数据表溢出，将造成查表混乱；故表头尽量放在数据页前端，以免数据表溢出。
3. TABBUF 的值不得大于表长，否则将造成运行混乱。

➤ 例：跳转表

跳转表能够实现多地址跳转功能。由于 PCL 和 W 的值相加即可得到新的 PCL，同时 PCH 从 PCLATH 中载入，因此，可以通过对 PCL 加上不同的 W 值来实现多地址跳转，可参考以下范例。

```

...
    ORG      0100H
    MOVLW    HIGH TAB2    ;获得跳转表地址高位（内部宏指令）
    MOVWF    PCLATH
    MOVF     TABBUF,W
TAB2:
    ADDWF    PCL,F
    GOTO     LABEL0_TAB2  ;TABBUF =0，跳转 LABEL0_TAB2
    GOTO     LABEL1_TAB2  ;以下类推
    GOTO     LABEL2_TAB2

```

GOTO

TABLE3_TAB2

注:

如上跳转表，有 4 个跳转分支，TABBUF 的合法范围为 0X00~0X03

方式二:

可以通过以下5个特殊功能寄存器对ROM区中的数据进行查找:

- PMCON
- PMDATL
- PMDATH
- PMADRL
- PMADRH

寄存器 PMADRH 指向 ROM 区数据地址的高字节 (Bit8~Bit15)，寄存器 PMADRL 指向 ROM 区数据地址的低字节 (Bit0~Bit7)。将 PMCON 寄存器的 RDON 位置 1 启动读操作，使用两条指令来读数据，RDON 位置 1 后的二条指令被自动忽略，建议用户 RDON 位置 1 后的两条指令为 NOP。执行完读操作后，所查找的数据保存在 PMDATLH:PMDATL 寄存器。

➤ 例：查找ROM 地址为“TABLE”的值

```

MOVF    TABLE_ADDR_H, W
MOVWF   PMADRH           ;设置TABLE地址高字节
MOVF    TABLE_ADDR_L, W
MOVWF   PMADRL           ;设置TABLE地址低字节
BSF     PMCON, RDON      ;开始读
NOP
NOP                       ;等待两条指令
MOVF    PMDATL, W
MOVWF   TABLE_DATA_L   ;TABLE_DATA_L= TABLE地址数据低字节
MOVF    PMDATH, W
MOVWF   TABLE_DATA_H   ;TABLE_DATA_H= TABLE地址数据高字节
...
...
TABLE   DW      1234H           ;定义数据表（16位）数据。
        DW      F178H
        DW      2123H
    
```

3.1.4.1 读ROM使能寄存器PMCON

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	-	-	-	RDON

位编号	位符号	说明
7-1	-	保留位
0	RDON	读控制位 0: 不启动 ROM 存储器读操作 1: 启动 ROM 读操作（由硬件清零 RDON；软件只能将 RDON 位置 1，但不能清零）

3.1.4.2 读ROM数据寄存器PMDATL、PMDATH

PMDATL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PMDATL							

位编号	位符号	说明
7-0	PMDATL	读 ROM 数据寄存器低 8 位

PMDATH

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PMDATH							

位编号	位符号	说明
7-0	PMDATH	读 ROM 数据寄存器高 8 位

3.1.4.3 读ROM地址寄存器PMADRL、PMADRH

PMADRL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PMADRL							

位编号	位符号	说明
7-0	PMADRL	读 ROM 地址寄存器低 8 位

PMADRH

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PMADRH							

位编号	位符号	说明
7-0	PMADRH	读 ROM 地址寄存器高 8 位

3.2 掉电数据存储器(EEPROM)

3.2.1 EEPROM 特性

- 内置 128 Bytes 独立 EEPROM 区
- EEPROM 操作时芯片其他功能正常运行
- EEPROM 支持单字节读擦写

3.2.2 EEPROM 相关寄存器

3.2.2.1 EEPROM控制寄存器EEPCON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	EEPEN	-	-	-	WRERR	WREN	WR	RD

位编号	位符号	说明
7	EEPEN	EEPROM 操作使能位 0: 禁止操作 EEPROM 1: 允许操作 EEPROM
6-4	-	保留位
3	WRERR	EEPROM 错误标志位 0: 写过程中未发生异常 1: 写过程中出错（写过程中发生了 WDT 复位、BOR 复位、外部复位等）
2	WREN	EEPROM 写使能位 0: 禁止写 EEPROM 1: 允许写 EEPROM
1	WR	EEPROM 写控制位 0: 写周期完成 1: 启动存储器写操作(由硬件清零 WR，用软件只能将 WR 置 1，但不能清零) 注：写操作完成后同样会置起 EEPIF 标志位
0	RD	EEPROM 读控制位 0: 不启动存储器读操作 1: 启动存储器读操作(由硬件清零 RD，用软件只能将 RD 置 1，但不能清零)

3.2.2.2 EEPROM数据寄存器EEPDAT

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	EEPDAT[7:0]							

位编号	位符号	说明
7-0	EEPDAT[7:0]	EEPROM 数据寄存器

3.2.2.3 EEPROM地址寄存器EEPADR

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	EEPADR[7:0]							

位编号	位符号	说明
7-0	EEPADR[7:0]	EEPROM 地址寄存器

3.2.3 EEPROM 读操作

- 1、使能 EEPEN
- 2、EEPADR 中写入地址
- 3、使能 RD
- 4、判断 RD 是否为 0 即读取完成
- 5、延时 10us(仅 CPU 时钟 $\geq$ 8M 时需要此操作)
- 6、读取 EEPDAT

3.2.4 EEPROM 写操作

- 1、使能 EEPEN
- 2、使能 WREN
- 3、EEPADR 中写入地址
- 4、EEPDAT 中写入待写入数据
- 5、使能 WR
- 6、判断 WR 是否为 0 或 EEPIF 是否为 1
- 7、清除 EEPIF
- 8、判断 WRERR 是否为 1 即写入失败

3.3 OPTION

HC18M002/003 将 0x0000~0x001F 定义为 OPTION 区域，用于存放用户配置的内容。

3.3.1 OPTION 存放内容

地址	名称	地址	名称
0x0000	SYSCFG	0x0010	-
0x0001	BORSEL	0x0011	-
0x0002	-	0x0012	-
0x0003	-	0x0013	-
0x0004	-	0x0014	-
0x0005	-	0x0015	-
0x0006	-	0x0016	-
0x0007	-	0x0017	-
0x0008	-	0x0018	-
0x0009	-	0x0019	-
0x000A	-	0x001A	-
0x000B	-	0x001B	-
0x000C	-	0x001C	-
0x000D	-	0x001D	-
0x000E	-	0x001E	-
0x000F	-	0x001F	-

3.3.1.1 系统配置项SYSCFG

位编号	15	14	13	12	11	10	9	8
位符号	-	PORSELB	CPUT	-	-	-	-	-
默认值	0	0	0	0	0	0	0	0

位编号	7	6	5	4	3	2	1	0
位符号	-	-	-	-	ERST_ENB	WDTENC	-	-
默认值	0	0	0	0	0	1	0	0

位编号	位符号	说明
15	-	保留位
14	PORSELB	复位时间选择: 0: 4.5ms 1: 18ms
13	CPUT	CPU 时钟模式选择 0: 2T 1: 4T
12-4	-	保留位
3	ERST_ENB	复位引脚使能位 0: 外部RST输入 1: PORTC4 为GPIO
2	WDTENC	WDT 时钟使能位 0: 关闭 WDT 时钟 1: 使能 WDT 时钟
1-0	-	保留位

3.3.1.2 BOR选择位BORSEL

位编号	15	14	13	12	11	10	9	8
位符号	-	-	-	-	-	-	-	-
默认值	0	0	0	0	0	0	0	0

位编号	7	6	5	4	3	2	1	0
位符号	-	-	-	-	-	BORVS[2:0]		
默认值	0	0	0	0	0	0	0	1

位编号	位符号	说明
15-3	-	保留位
2-0	BORVS[2:0]	BOR 检测电压点选择位 000: 不可选择 001: 2.0V 010: 2.4V 011: 2.6V 100: 3.0V 101: 3.6V 110: 3.9V 111: 4.2V

3.4数据存储器(RAM)

HC18M002/003 为用户提供了 512 Bytes 内部 RAM 作为数据存储器。下图为数据存储器空间分配。

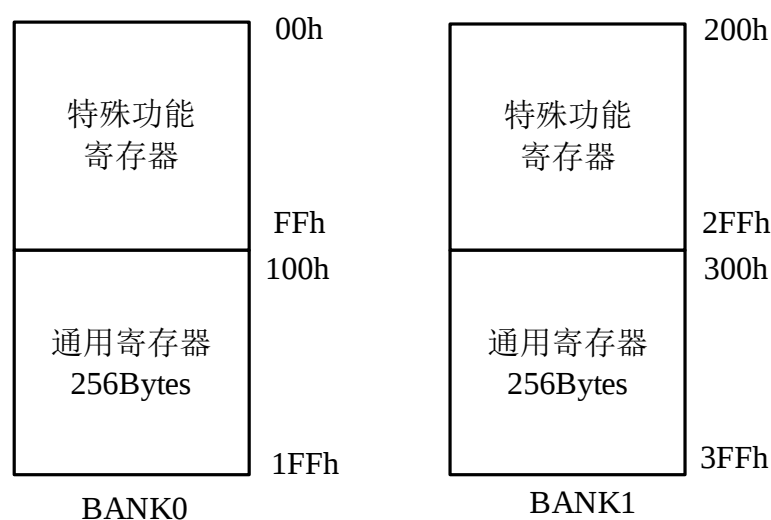


Figure 3-3 数据存储器映射

3.5 特殊功能寄存器(SFR)

HC18M002/003的特殊功能寄存器(SFR)，分在2个存储区Bank0~Bank1，RP0是存储区的选择位。

3.5.1 核心寄存器列表

CORE Register 00-09h&200-209h

00h&200h	INDF0	间接寻址 0 寄存器（不是实际存在的物理寄存器）							
01h&201h	INDF1	间接寻址 1 寄存器（不是实际存在的物理寄存器）							
02h&202h	PCL	程序计数器（PC）低字节							
03h&203h	STATUS	-	-	RP0	TO	PD	Z	DC	C
04h&204h	FSR0L	间接寻址 0 地址低位指针							
05h&205h	FSR0H	间接寻址 0 地址高位指针							
06h&206h	FSR1L	间接寻址 1 地址低位指针							
07h&207h	FSR1H	间接寻址 1 地址高位指针							
08h&208h	PCLATH	-	-	-	-	程序计数器高 4 位			
09h&209h	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF

3.5.2 特殊功能寄存器列表

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位初值
BANK0										
010h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	-	TRISA0	0000 0000
011h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 0000
012h	TRISC	TRISC7	TRISC6	-	TRISC4	-	TRISC2	TRISC1	TRISC0	0000 0000
013h	TRISD	-	-	-	TRISD4	-	-	-	-	0000 0000
01Ch	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	-	PORTA0	0000 0000
01Dh	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	0000 0000

01Eh	PORTC	PORTC7	PORTC6	-	PORTC4	-	PORTC2	PORTC1	PORTC0	0000 0000
01Fh	PORTD	-	-	-	PORTD4	-	-	-	-	0000 0000
028h	WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	-	WPUA0	0000 0000
029h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	0000 0000
02Ah	WPUC	WPUC7	WPUC6	-	WPUC4	-	WPUC2	WPUC1	WPUC0	0000 0000
02Bh	WPUD	-	-	-	WPUD4	-	-	-	-	0000 0000
034h	WPDA	WPDA7	WPDA6	WPDA5	WPDA4	WPDA3	WPDA2	-	WPDA0	0000 0000
035h	WPDB	WPDB7	WPDB6	WPDB5	WPDB4	WPDB3	WPDB2	WPDB1	WPDB0	0000 0000
036h	WPDC	WPDC7	WPDC6	-	WPDC4	-	WPDC2	WPDC1	WPDC0	0000 0000
037h	WPDD	-	-	-	WPDD4	-	-	-	-	0000 0000
038h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000
054h	PIR1	-	ADIF	MODF	SPIF	IICIF	-	T2IF	T1IF	0000 0000
055h	PIR2	CTKIF	WDTIF	PWM0IF	-	RXIF	TXIF	EEPIF	-	0000 0000
058h	T1L	Timer1 计数寄存器低字节								xxxx xxxx
059h	T1H	Timer1 计数寄存器高字节								xxxx xxxx
05Ah	T1CON	T1CS1	T1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	-	T1ON	0000 0000
05Bh	T0	Timer0 计数寄存器								xxxx xxxx
05Ch	T2	Timer2 计数寄存器								xxxx xxxx
05Dh	PR2	Timer2 周期寄存器								1111 1111
05Eh	T2CON	-	T2CKPS3	T2CKPS2	T2CKPS1	T2CKPS0	T2ON	-	-	0000 0000
05Fh	PR1L	Timer1 周期寄存器低字节								xxxx xxxx
060h	PR1CON	-	-	-	-	T1CKPS3	T1CKPS2	-	PR1EN	0000 0000
070h	PIE1	-	ADIE	-	SPIE	IICIE	-	T2IE	T1IE	0000 0000
071h	PIE2	CTKIE	WDTIE	PWM0IE	-	-	UARTIE	EEPIE	-	0000 0000
078h	OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
079h	PCON	-	-	-	WDTENS	-	-	-	-	0001 0000
07Ah	OSCCON	T0OSCEN	RC32M_DIV[1:0]		HXEN	CPU_DIV[2:0]			SCS	0110 0000

07Bh	OSCRDY	-	-	-	-	-	-	HSRCRDY	LSRCRDY	0000 0011
07Ch	TRMEN	-	-	-	-	-	-	-	RCTRMEN	0000 0000
07Dh	TRMV	-	TRMV 校准位							0xxx xxxx
08Ch	ANSELA	ANSELA7	ANSELA6	ANSELA5	ANSELA4	ANSELA3	ANSELA2	-	ANSELA0	0000 0000
08Dh	ANSELB	ANSELB7	ANSELB6	ANSELB5	ANSELB4	ANSELB3	ANSELB2	ANSELB1	ANSELB0	0000 0000
08Eh	ANSELC	ANSELC7	ANSELC6	-	ANSELC4	-	ANSELC2	ANSELC1	ANSELC0	0000 0000
08Fh	ANSELD	-	-	-	ANSELD4	-	-	-	-	0000 0000
092h	ADRESL	ADC 结果寄存器低字节								xxxx xxxx
093h	ADRESH	ADC 结果寄存器高字节								xxxx xxxx
094h	ADCON0	-	-	CHS3	CHS2	CHS1	CHS0	ADON	ADEN	00000000
095h	ADCON1	ADFM	ADCS2	ADCS1	ADCS0	INREFS2	INREFS1	INREFS0	VREFS	0000 0110
096h	ADCLK	-	-	-	-	-	ADCLK2	ADCLK1	ADCLK0	0000 0000
09Ah	PMDATL	程序存储器读数据寄存器的低字节								0000 0000
09Bh	PMDATH	程序存储器读数据寄存器的高字节								0000 0000
09Ch	PMADRL	程序存储器读地址寄存器的低字节								0000 0000
09Dh	PMADRH	程序存储器读地址寄存器的高字节								0000 0000
09Eh	PMCON	-	-	-	-	-	-	-	RDON	1000 0000
0B8h	EEPCON	EEPEN	-	-	-	WPEER	WREN	WR	RD	0000 0000
0B9h	EEPDAT	EEPROM 数据寄存器								0000 0000
0BAh	EEPADR	EEPROM 地址寄存器								0000 0000
0C0h	BORC	BOREN	BORDBCEN	-	-	-	BORVS[2:0]			1000 0001
0C1h	BORDBC	BOR 消抖控制寄存器								0000 0000
0C2h	RSTFR	PORF	EXRSTF	BORF	WDTRF	-	-	-	-	xxxx 0000
0C8h	SLEWC	-	-	-	-	-	-	SLEWC[1:0]		0000 0000
0C9h	UARTMAP	-	RXDMAP2	RXDMAP1	RXDMAP0	-	TXDMAP2	TXDMAP1	TXDMAP0	0000 0000
0CAh	IICMAP	-	SLKMAP[2:0]			-	SDAMAP[2:0]			0000 0000
0CBh	SPIMAP	SSMAP[1:0]		SCKMAP[1:0]		MOSIMAP[1:0]		MOSIMAP1	MOSIMAP0	0000 0000
0CCh	INTMAP	-	MOSIMAP2	TXDMAP4	TXDMAP3	-	INTMAP[2:0]			0000 0000

0CDh	PWMMAP	-	PWM01MAP[2:0]			-	PWM0MAP[2:0]			0000 0000
0D0h	PWMEN	-	-	-	PWM01EN	-	-	-	PWM0EN	0000 0000
0D1h	FLTMODE	-	EFLT0	-	-	-	-	FLT0_MODE[1:0]		0000 0000
0D2h	PWMM	-	-	-	PWM0M	-	-	-	RELOAD0	0000 0001
0D3h	PWM0C	-	-	FLT0S	FLT0C	PWM0S		CK0		0000 0000
0D8h	PWM0PL	PWM0 周期寄存器低 8 位								0000 0000
0D9h	PWM0PH	-	-	-	-	PWM0 周期寄存器高 4 位				0000 0000
0DAh	PWM0DL	PWM0 占空比寄存器低 8 位								0000 0000
0DBh	PWM0DH	-	-	-	-	PWM0 占空比寄存器高 4 位				0000 0000
0DCh	PWM0DTL	PWM0 死区时间寄存器低 8 位								0000 0000
0DDh	PWM0DTH					PWM0 死区时间寄存器高 4 位				0000 0000

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位初值
BANK1										
230h	SCON	FE	RXROV	TXCOL	REN	TB8	RB8	-	-	0000 0000
231h	SCON2	-	-	BRTR	-	-	-	SM1	SM2	0000 0000
232h	SBUF	UART 数据缓冲和寄存器								0000 0000
235h	BRTL	波特率发生器寄存器低 8 位								0000 0000
236h	BRTH	波特率发生器寄存器高 8 位								0000 0000
240h	IICCON	CR2	IICEN	STA	STO	-	AA	CR1	CR0	0000 0000
241h	IICSTA	IIC 状态码					-	-	-	1111 1000
242h	IICDAT	IIC 数据寄存器								0000 0000

4 系统时钟

4.1 系统时钟特性

HC18M002/003 单片机系统时钟有 2 种时钟源可选：

- 内部高频 RC 时钟（32MHz）
- 内部低频 RC 时钟（32KHz）

用户选择后的系统时钟（如果选择的是内部高频 RC，则经 RC32M_DIV[1:0]分频后的时钟）记做 osc_clk ，其频率为 F_{osc} ，周期为 T_{osc} ，主要用于外设模块 PWM， osc_clk 可以进行 1/2/4/8/16/32/64/128 之间分频，分频后的时钟记做 CPU 时钟，其频率为 F_{cpu} ，周期为 T_{cpu} 。

芯片上电复位后，默认选择内部高频 RC 作为系统时钟，其 F_{osc} 为 4MHz， F_{cpu} 为 2MHz，可以通过配置相关寄存器改变 osc_clk 和 cpu_clk 的频率。

注意：CPU 最高可以运行在 8MHz 频率下，如果所选时钟源频率高于 8MHz，需要对其进行分频，使 CPU 时钟频率等于或低于 8MHz。

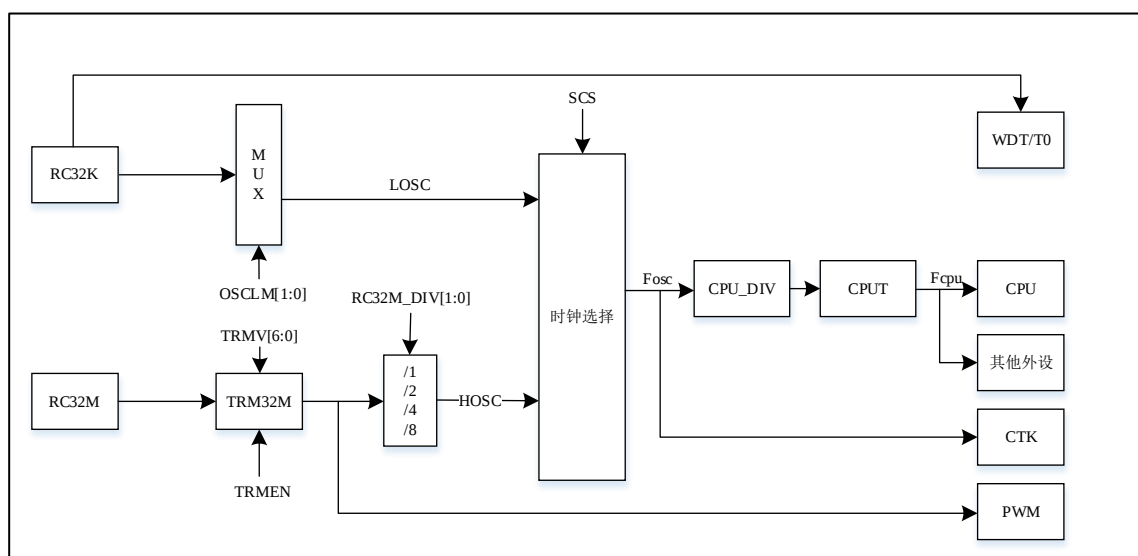


Figure 4-1 系统时钟框图

4.2 系统时钟相关寄存器

4.2.1 时钟控制寄存器 OSCCON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	1	0	0	0	0	0
位符号	T0OSCEN	RC32M_DIV[1:0]		HXEN	CPU_DIV [2:0]			SCS

位编号	位符号	说明
7	其它模块使用	本功能不使用本位，使用时需注意
6-5	RC32M_DIV[1:0]	内部高频 RC 分频系数 00: rc32m_clk 01: rc32m_clk /2 10: rc32m_clk /4 11: rc32m_clk /8（默认）
4	HXEN	内部高速 RC 振荡器使能位 0: 在低速或绿色模式下禁止高频振荡器 1: 在低速或绿色模式下使能高频振荡器
3-1	CPU_DIV[2:0]	CPU 时钟分频系数 000: 1 分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频
0	SCS	高低频时钟选择位 0: 系统时钟选择为高频时钟 1: 系统时钟选择为低频时钟

4.2.2 时钟状态寄存器 OSCRDY

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	1	1
位符号	-	-	-	-	-	-	HSRCRDY	LSRCRDY

位编号	位符号	说明
7-2	-	保留位
1	HSRCRDY	内部高速 RC 振荡器状态位 0: 内部高速 RC 未准备 1: 内部高速 RC 准备就绪
0	LSRCRDY	内部低速 RC 振荡器状态位 0: 内部低速 RC 未准备 1: 内部低速 RC 准备就绪

4.2.3 内部高频 RC 调整使能寄存器 TRMEN

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-							RCTRMEN

位编号	位符号	说明
7-1	-	保留位
0	RCTRMEN	内部高频 RC 调整使能位 1: 使能内部高频 RC 调整 0: 禁止内部高频 RC 调整 注: 使能该寄存器后, 必须立即配置 TRMV 寄存器, 否则这个使能寄存器在执行完下一条指令后会被清零, 内部高频 RC 调整就会失效。

4.2.4 内部高频 RC 调整配置寄存器 TRMV

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	x	x	x	x	x	x	x
位符号	-	RCTRMV						

位编号	位符号	说明
7	-	保留位
6-0	RCTRMV	内部高频 RC 调整配置值 注： 1. x 表示不确定的值，此寄存器的上电复位值为出厂的校准值，校准后的高频 RC 时钟频率为 32MHz（1%）。 2. 在配置这个寄存器值时，需要先将内部高频 RC 调整使能位配置为 1。 3. 根据校准曲线软件先使能 RCTRMEN，紧接着就要配置 RCTRMV，在调整完成后 RCTRMEN 自动清零，防止重复操作。

使用注意：

- 1、系统上电后将高频 RC 出厂校准值读出保存，该值为高频 RC=32MHz 的校准值。
- 2、如需微调高频 RC 频率，可按照步骤，先将 RCTRMEN 使能，再根据调节步长设置新的 RCTRMV 值。
- 3、系统时钟选择内部高频 RC 时，重新设置 RCTRMV 值后，高频 RC 时钟频率发生改变，此时模块（如定时器、UART 的波特率等等）相关频率也会跟随改变。

5 系统工作模式

5.1 系统工作模式特性

HC18M002/003 共有四种工作模式：

- 高频模式
- 低频模式
- 绿色模式
- 休眠模式

系统复位后，工作于高频模式。程序运行过程中，可以通过设置 SCS 位使系统在高频和低频模式之间切换。

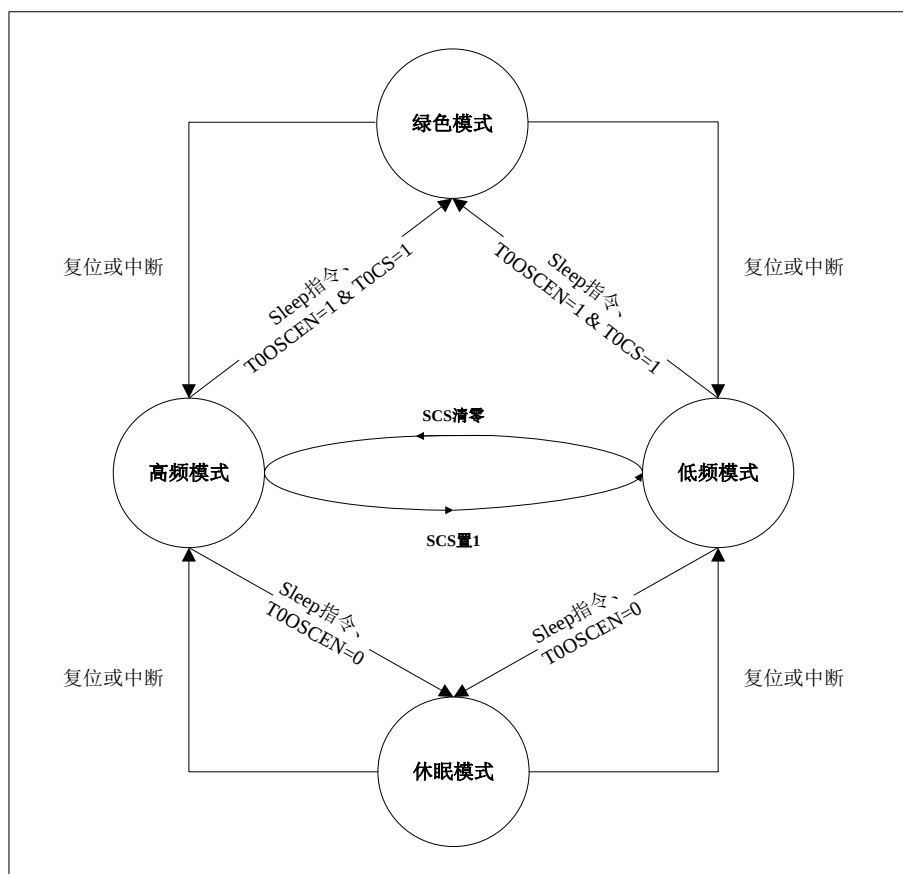


Figure 5-1 工作模式切换框图

注意：

- 从休眠或绿色模式唤醒，中断使能的情况则进入相应中断，否则执行下一句。
- 进入休眠或绿色模式前，关闭 WDT 可降低功耗。

各种模式下振荡器模块及Timer0/Timer1的工作状态表

模块	高频模式	低频模式	绿色模式	休眠模式
高频振荡器	运行	由HXEN决定	由HXEN决定	关闭
低频振荡器	运行	运行	运行	关闭
Timer0	运行	运行	定时唤醒模式下运行	计数器模式下运行
Timer1	运行	运行	异步定时唤醒模式下运行	异步计数器模式下运行
WDT	运行	运行	由WDTEN决定	由WDTEN决定

5.2 高低频时钟切换

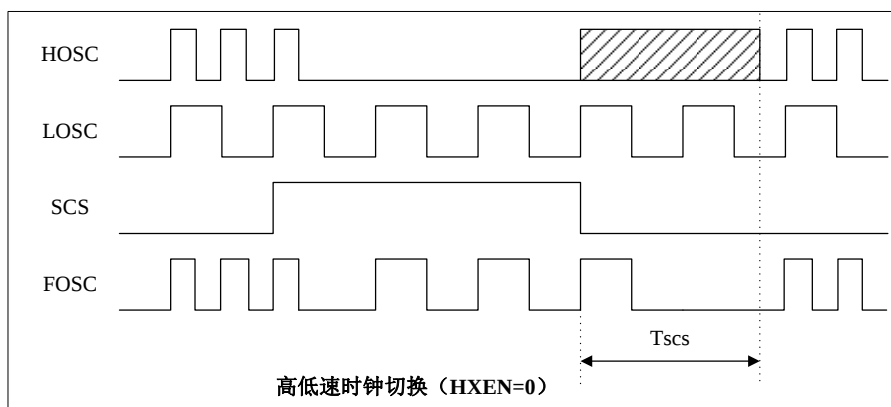


Figure 5-2 高低频切换时序图 (HXEN=0)

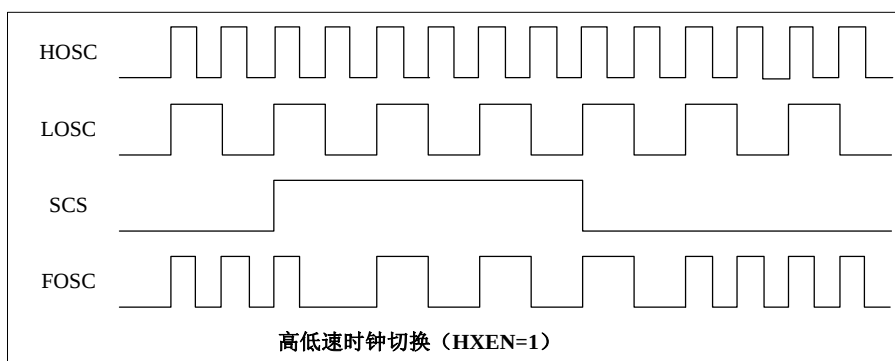


Figure 5-3 高低频切换时序图 (HXEN=1)

时钟切换时间 (Tscs) 计算:

$$T_{scs} = \text{高频振荡器起振时间} + \text{高频振荡器稳定时间}$$

不同类型高频振荡器的稳定时间表

振荡器类型	OST 定时时间
内部 RC 振荡器	16 Clock
低频 RC 振荡器	4 Clock

注意:

系统进入绿色模式后, 低频时钟正常运行。外部或内部中断将系统从绿色模式中唤醒不需要唤醒时间。

时钟切换前, 需要硬件进行 CLK 安全判断, 即判断即将切换至的时钟是否正常允许。

6 复位

6.1 复位特性

- 提供多种方式复位
- 所有的复位方式都有特定标志

当任何一种复位产生时，系统进入复位状态，所有的特殊功能寄存器被初始化，程序停止运行，同时程序计数器（PC）清零。经过上电延时定时器延时后，系统结束复位状态，程序从0000h地址开始执行。

上电延时定时器在复位时提供一个18ms（典型值）的固定延时和一个振荡器起振稳定的时间延时。

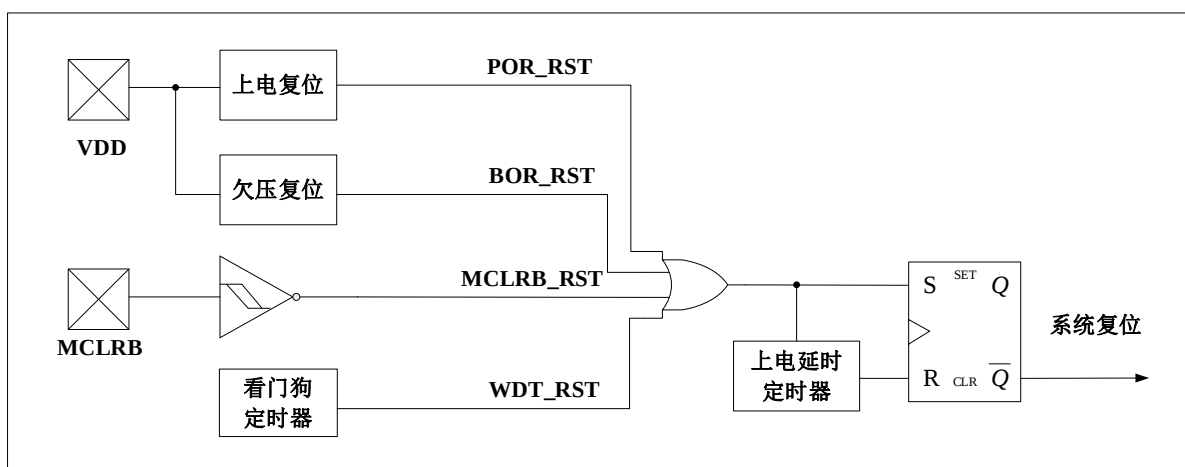


Figure 6-1 复位电路示意图

6.2 上电复位

HC18M002/003单片机在上电过程中，会产生一个POR信号，此信号会复位单片机，同时置位RSTFR寄存器里的PORF位，用户可以判断此标志以来确定是否发生POR复位。

复位结束后会读取OPTION，然后再等待PORSELB代码选项决定的时间，芯片就开始执行用户程序。

注意：完全掉电后（ $VDD < 0.7V$ ），POR复位后的RAM值不稳定，建议重新初始化相应RAM，VDD大于0.7V时RAM数据可以被保存。

6.3 BOR 复位

当 VDD 电压下降到 V_{BOR} 以下,且持续时间超过 T_{BOR} 时,系统产生欠压复位。BOR 复位时, RSTFR 寄存器的 BORF 位将被置 1, 用户可以判断此标志以来确定是否发生 BOR 复位。

HC18M002/003 可以通过代码选项或寄存器来选择 BOR 检测的电压档位, 客户在使用时可根据情况选择合适的 BOR 检测电压。BOR 档位: 4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.8V

欠压复位示意图如下所示, 其中 T_{BOR} 也可以通过寄存器配置, 用来进行电压消抖。

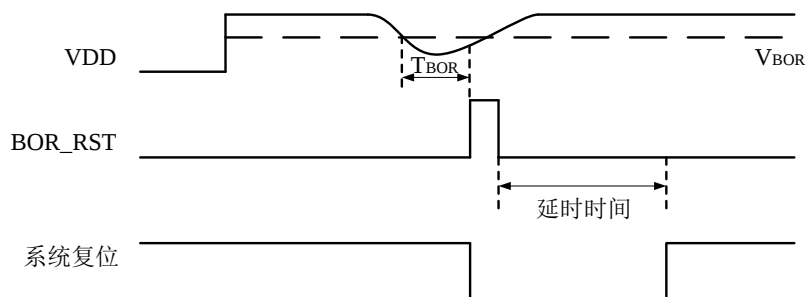


Figure 6-2 BOR 示意图

6.4 外部 RST 复位

外部 $\overline{RST}$ 引脚复位就是从外部向 $\overline{RST}$ 引脚施加一定宽度的复位脉冲, 从而实现单片机的复位, 不使用时可以将其配置为 I/O 口, 需要在代码选项中设置。

做 RST 端口时, 将 $\overline{RST}$ 复位管脚拉低后, 单片机才会进入复位状态, 将 $\overline{RST}$ 复位管脚拉回高电平后, 单片机结束复位状态并从用户程序区的 0000H 处开始正常工作。RST 复位时, RSTFR 寄存器的 EXRSTF 将被置 1, 用户可以判断此标志以来确定是否发生外部 RST 复位。

注意: PORTC4 端口作为外部 $\overline{RST}$ 复位端口时, 无法作为普通 I/O 使用。

6.5 看门狗 (WDT) 复位

为了防止系统在异常情况下受到干扰, MCU 程序跑飞, 导致系统长时间异常工作, 通常是引进看门狗, 如果 MCU 程序中不在规定的时间内按要求操作看门狗, 就认为 MCU 处于异常状态, 看门狗就会强制 MCU 复位, 芯片重新从 0000H 开始运行。

注意: 要使 WDT 复位, 必须置 WDTRST 为 1, 即允许 WDT 复位功能, 否则即使允许 WDT 运行, WDT 也只会置溢出标志, 并不会复位。

6.6 复位相关寄存器

6.6.1 复位标志寄存器 RSTFR

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
POR复位	1	x	x	x	0	0	0	0
EXRST复位	u	1	u	u	0	0	0	0
BOR复位	u	u	1	u	0	0	0	0
WDT复位	u	u	u	1	0	0	0	0
位符号	PORF	EXRSTF	BORF	WDTRF	-	-	-	-

注意：x表示不确定的值，u表示该值由当前复位方式前的值决定，建议在POR复位后清零一下该寄存器。

位编号	位符号	说明
7	PORF	上电复位标志位 0：无上电复位 1：发生上电复位，软件清 0
6	EXRSTF	外部 RST 复位标志位 0：无外部 RST 复位 1：发生外部 RST 复位，软件清 0
5	BORF	欠压复位标志位 0：无欠压复位 1：发生欠压复位，软件清 0
4	WDTRF	WDT 复位标志位 0：无 WDT 复位 1：发生 WDT 复位，软件清 0
3-0	-	保留

6.6.2 BOR 电压检测控制寄存器 BORC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
复位值	1	0	0	0	0	0	0	1
位符号	BOREN	BOR_DBC_EN	-	-	-	BORVS[2:0]		

位编号	位符号	说明
7	BOREN	BOR 使能位 0: 禁止 BOR 1: 允许 BOR
6	BOR_DBC_EN	BOR 消抖使能位 0: 不使能 1: 使能
5-3	-	保留位
2-0	BORVS[2:0]	BOR 检测电压点选择位 000: 禁止配置 001: 2.0V 010: 2.4V 011: 2.6V 100: 3.0V 101: 3.6V 110: 3.9V 111: 4.2V

6.6.3 BOR 电压检测去抖控制寄存器 BORDBC

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	-	BORDBC[2:0]		

位编号	位符号	说明
7-3	-	保留位
2-0	BORDBC[2:0]	BOR 消抖控制位时间选择 000: BORDBC=2 001: BORDBC=4 010: BORDBC=8 011: BORDBC=16 100: BORDBC=32 101: BORDBC=64 110: BORDBC=128 111: BORDBC=256 消抖时间 = BORDBC[2:0] * 8T_{cpu} + 2 T_{cpu} 注: 需要 BOR_DBC_EN 打开, 否则 BOR 不消抖

注意: 绿色模式和休眠模式下自动关闭, 退出绿色模式和休眠模式自动打开。

7 通用及复用I/O

7.1 通用及复用 I/O 特性

- 支持最多 18 个双向 I/O 端口
- 多种模式可配

7.2 I/O 模式

HC18M002/003 所有 I/O 口均可由软件配置成多种工作类型之一，具体为：输入、输出、上下拉。
HC18M002/003 的 PORTC4 为真开漏口。
HC18M002/003 上电复位后，所有 IO 口都默认为模拟输入。

7.3 I/O 端口相关寄存器

7.3.1 IO 口输入输出寄存器 TRISx(x=A、B、C、D)

TRISA

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	-	TRISA0

TRISB

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	TRISBx[7:0]							

TRISC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	TRISC7	TRISC6	-	TRISC4	-	TRISC2	TRISC1	TRISC0

TRISD

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	TRISD4	-	-	-	-

位编号	位符号	说明
7-0	TRISx[7:0]	PORT 端口输入输出控制寄存器 0: 输入状态 1: 输出状态

7.3.2 IO 口模拟模式控制寄存器 ANSELx(x=A、B、C、D)

ANSELA

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	ANSELA7	ANSELA6	ANSELA5	ANSELA4	ANSELA3	ANSELA2	-	ANSELA0

ANSELB

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	ANSELBx[7:0]							

ANSELC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	ANSELC7	ANSELC6	-	ANSELC4	-	ANSELC2	ANSELC1	ANSELC0

ANSELD

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	ANSELD4	-	-	-	-

位编号	位符号	说明
7-0	ANSELx [7:0]	PORT 端口数字模拟输入状态选择器 0: 模拟状态 1: 数字状态

7.3.3 IO 口上拉控制寄存器 WPUx(x=A、B、C、D)

WPUA

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	-	WPUA0

WPUB

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	WPUBx[7:0]							

WPUC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	WPUC7	WPUC6	-	WPUC4	-	WPUC2	WPUC1	WPUC0

WPUD

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	WPUD4	-	-	-	-

位编号	位符号	说明
7-0	WPUx [7:0]	PORT 端口数字上拉电阻选择 0: 禁止上拉 1: 使能上拉

7.3.4 PB 口上拉控制选择寄存器 OPTION

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

位编号	位符号	说明
7	PBPUB	PORTB 上拉使能位 0: 使能 PORTB 上拉（此时无论 WPUB 为何值 PORTB 都为上拉） 1: PORTB 上拉由 WPUB 决定
6-0	其它模块使用	本功能不使用本位，使用时需注意

7.3.5 IO 口下拉控制寄存器 WPDx(x=A、B、C、D)

WPDA

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	WPDA7	WPDA6	WPDA5	WPDA4	WPDA3	WPDA2	-	WPDA0

WPDB

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	WPDBx[7:0]							

WPDC

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	WPDC7	WPDC6	-	WPDC4	-	WPDC2	WPDC1	WPDC0

WPDD

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	WPDD4	-	-	-	-

位编号	位符号	说明
7-0	WPDx [7:0]	PORT 端口数字下拉电阻选择 0: 禁止下拉 1: 使能下拉

7.3.6 IO 口数据寄存器 PORTx(x=A、B、C、D)

PORTA

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	PORTA7	PORTA6	PORTA5	PORTA4	-	-	-	PORTA0

PORTB

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PORTBx[7:0]							

PORTC

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	PORTC0	-	PORTC0	PORTC0	PORTC0

PORTD

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	PORTD4	-	-	-	-

位编号	位符号	说明
7-0	PORTx [7:0]	PORT 口数据寄存器

7.4 外设功能映射相关寄存器

7.4.1 UART 端口通讯端口映射寄存器 UARTMAP

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	RXDMA P2	RXDMA P1	RXDMA P0	-	TXDMA P2	TXDMA P1	TXDMA P0

RXD[2:0]	端口映射
000	无映射
001	RXD-PORTC2
010	保留
011	RXD-PORTA2
100	RXD-PORTA7
101	RXD-PORTB7
11x	保留

TXD[4:0]	端口映射
00000	无映射
00001	TXD-PORTC0
00010	TXD-PORTD4
00100	TXD-PORTA0
01000	TXD-PORTA3
10000	TXD-PORTB6
其他	不可配置

注意： 1、TXD 映射部分寄存器位于 INTMAP 寄存器。
2、配置完成映射后，需要用户手动配置 IO 状态。

7.4.2 IIC 端口通讯端口映射寄存器 IICMAP

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	SCKMAP[2:0]			-	SDAMAP[2:0]		

位编号	位符号	说明
7	-	保留位
6-4	SCKMAP[2:0]	IIC SCK 端口选择 000: 无映射 001: PORTC2 010: 保留 011: PORTA2 100: PORTA7 101: PORTB7 11x: 保留
3	-	保留位
2-0	SDAMAP[2:0]	IIC SDA 端口选择 000: 无映射 001: PORTC0 010: PORTD4 011: PORTA0 100: PORTA3 101: PORTB6 11x: 保留

注意：当使能 IIC 映射后，默认将 IIC_SDA 口设为开漏带上拉输出，但 SCK 依旧需要用户手动配置模式。

7.4.3 SPI 端口通讯端口映射寄存器 SPIMAP

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	SSMAP[1:0]		SCKMAP[1:0]		MOSIMAP[1:0]		MISOM AP1	MISOM AP0

位编号	位符号	说明
7-5	SSMAP[1:0]	SPI SS 端口选择 00: 无映射 01: 保留 10: PORTA7 11: PORTB7
5-4	SCKMAP[1:0]	SPI SCK 端口选择 00: 无映射 01: PORTC2 10: PORTA3 11: PORTB6
3-2	MOSIMAP[1:0]	SPI MOSI 端口选择 00: 无映射 01: MOSI-PORTC1 10: 保留 11: MOSI-PORTB4
1-0	MISOMAP[1:0]	见下表

MISO[2:0]	端口映射
000	无映射
001	MISO-PORTC0
010	MISO-PORTD4
100	MISO-PORTB5
其他	不可配置

注意：1、MISO 映射部分寄存器位于 INTMAP 寄存器。
2、配置完成映射后，需要用户手动配置 IO 状态。

7.4.4 INT 端口映射寄存器 INTMAP

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	MISOM AP2	TXDMA P4	TXDMA P3	-	INTMAP[2:0]		

位编号	位符号	说明
7	-	保留位
6-4	-	此处不使用，使用请参考 UARTMAP&SPIMAP 寄存器
3	-	保留位
2-0	INTMAP[2:0]	INT0 端口选择 000: 无映射 001: PORTC1 010: PORTA0 011: PORTA3 100: PORTB3 其他: 保留

注意：此处需要用户手动设置 IO 口模式。

7.4.5 PWM 端口映射寄存器 PWMMAP

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	PWM01MAP[2:0]			-	PWM0MAP[2:0]		

位编号	位符号	说明
7	-	保留位
6-4	PWM01MAP[2:0]	PWM01 端口选择 000: 无映射 001: PORTC2 010: 保留 011: PORTA2 100: PORTA7 101: PORTB7 11x: 保留
3	-	保留位
2-0	PWM0MAP[2:0]	PWM0 端口选择 000: 无映射 001: PORTC0 010: PORTD4 011: PORTA0 100: PORTA3 101: PORTB6 11x: 保留

注意：此处需要用户手动设置 IO 口模式。

8 中断

8.1 中断特性

HC18M002/003 中断源:

- Timer0定时器中断
- INT0外部中断
- PORTB口电平变化中断
- Timer1定时器中断
- Timer2定时器中断
- AD中断
- UART中断
- PWM中断
- IIC中断
- CTK中断
- EEPROM
- WDT

系统产生中断时，程序计数器（PC）值压入堆栈，程序跳转至0004h，进入中断服务程序。当程序运行到RETFIE指令时，系统退出中断服务程序，程序计数器值出栈，系统执行PC+1地址对应的指令。

为避免误进入中断，在使能中断和退出中断服务程序之前，必须清除中断标志位。

PORTB电平变化中断中，在清零RBIF之前必须执行PORTB端口读操作。

8.2 中断保护

有中断请求发生并被响应后，程序转至0004H执行中断服务程序。

中断服务程序开始执行时，需保存W寄存器、STATUS寄存器、PCLATH寄存器的内容;结束中断服务程序时，恢复PCLATH寄存器、STATUS寄存器、W寄存器的数值，注意顺序。

注意:在退出中断时,由于需要先恢复STATUS,再使用 MOVF 指令恢复 W,可能会改变STATUS,因此必须使用 SWAPF 指令恢复 W。注意在中断中共有两句 SWAPF 指令。

➤ 例：对W、PCLATH 和STATUS 进行入栈保护。

```

                ORG      0000H
                GOTO     START
                ORG      0004H
                GOTO     INT_SERVICE
                ORG      0010H

START:
...

INT_SERVICE:
    MOVWF      W_TEMP           ;保存 W。
    SWAPF      STATUS,W
    MOVWF      STATUS_TEMP      ;保存 STATUS。
    MOVF       PCLATH,W
    MOVWF      PCLATH_TEMP      ;保存 PCLATH。
    CLRF       STATUS           ;切换到BANK0
    ...
    MOVF       PCLATH_TEMP,W
    MOVWF      PCLATH           ;恢复PCLATH。
    SWAPF      STATUS_TEMP,W
    MOVWF      STATUS           ;恢复STATUS。
    SWAPF      W_TEMP,F
    SWAPF      W_TEMP,W         ;恢复W。
    RETFIE                     ;退出中断。
    ...
    END

```

8.3 多中断操作

在同一时刻，系统中可能出现多个中断请求。此时，用户必须根据系统的要求对各中断进行优先权的设置。中断请求标志IF由中断事件触发，当IF处于有效值“1”时，系统并不一定会响应该中断。各中断触发事件如下表所示：

中断	有效触发
T0IF	T0计数器溢出
INTF	由INTEDG控制
RBIF	PORTB电平变化
T1IF	T1计数器溢出
T2IF	T2发生匹配
ADIF	AD数据采集完成
RXIF/TXIF	UART发生发送接收事件
IICIF	IIC发生发送接收事件
PWM0IF	PWM0周期计数溢出中断
CTKIF	CTK转换完成中断
WDTIF	WDT中断溢出标志位
EEPIF	EEPROM擦写完成标志位

多个中断同时发生时，需要注意的是：首先，必须预先设定好各中断的优先权；其次，利用IE和IF控制系统是否响应该中断。在程序中，必须对中断控制位和中断请求标志进行检测。

➤ 例：多中断条件下检测中断请求。

```

ORG    0004H;
GOTO   INT_SERVICE

INT_SERVICE:
...
;保存STATUS、W和PCLATH。

INT0CHK:
;检查是否有INT0 中断请求。
BCF     STATUS,RP0
;BANK0
BTFSS   INTCON,INTE
;检查是否使能INT0 中断。
GOTO    INT0CHK
;跳到下一个中断。
BTFSC   INTCON,INTF
;检查是否有INT0 中断请求。
GOTO    INT0
;进入INT0 中断。

INTT0CHK:
;检查是否有T0 中断请求。
BTFSS   INTCON,T0IE
;检查是否使能T0 中断。
GOTO    INTT2CHK
;跳到下一个中断。
BTFSC   INTCON,T0IF
;检查是否有T0 中断请求。
GOTO    INTT0
;进入T0 中断。

INTT2CHK:
;检查是否有T2 中断请求。
BCF     STATUS,RP0
;BANK0
BTFSS   PIE1,T2IE
;检查是否使能T2 中断。
GOTO    INTRBCHK
;跳到下一个中断。
BCF     STATUS,RP0
;BANK0
BTFSC   PIR1,T2IF
;检查是否有T2 中断请求。
GOTO    INTT2
;进入T2 中断。

```

```
INTRBCHK:
    BTFSS  INTCON,RBIE      ;检查是否使能PORTB电平变化中断。
    GOTO   INT_EXIT        ;跳到中断结束。
    BTFSC  INTCON,RBIF      ;检查是否有PORTB电平变化中断请求。
    GOTO   INTRB           ;进入PORTB电平变化中断。

INTT2:
    BCF    PIR1,T2IF
    ...
    GOTO   INT_EXIT

INT_EXIT:
    ...
    RETFIE                ;恢复STATUS、W和PCLATH。
                          ;退出中断。
```

8.4 中断相关寄存器

8.4.1 内核中断

使能内核中断必须将 GIE 和相应中断的使能位置 1，使能 PORTB 电平变化中断还需要将相应端口配置为输入并且 IOCB 的相应位置 1。INT0 外部中断和 PORTB 电平变化中断可以唤醒 SLEEP，Timer0 中断在计数器模式和定时唤醒模式下可以唤醒 SLEEP。

INTCON

位编号	7	6	5	4	3	2	1	0
R/W	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	x
位符号	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

位编号	位符号	说明
7	GIE	全局中断使能位 0: 禁止所有中断 1: 使能所有未屏蔽的中断
6	其它模块使用	本功能不使用本位，使用时需注意
5	TOIE	Timer0 溢出中断允许位 0: 禁止 T0 中断 1: 允许 T0 中断
4	INTE	INT0 外部中断允许位 0: 禁止 INT0 中断 1: 允许 INT0 中断
3	RBIE	PORTB 电平变化中断允许位 0: 禁止 PORTB 电平变化中断 1: 允许 PORTB 电平变化中断
2	TOIF	Timer0 溢出中断标志位，Timer0 计数寄存器在 FFh 至 00h 时产生溢出信号 0: Timer0 计数寄存器未溢出 1: Timer0 计数寄存器溢出(必须由软件清 0)
1	INTF	INT0 外部中断标志位 0: 未发生 INT0 外部中断 1: 发生 INT0 外部中断(必须由软件清 0)
0	RBIF	PORTB 电平变化中断标志位 0: PORTB[7:0]电平状态没有变化 1: PORTB[7:0]中至少有一个口的电平状态发生了改变(必须由软件清 0)

OPTION

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

位编号	位符号	说明
7	其它模块使用	本功能不使用本位，使用时需注意
6	INTEDG	触发 INT0 外部中断的边沿选择位 0: INT0 引脚下降沿触发中断 1: INT0 引脚上升沿触发中断
5-0	其它模块使用	本功能不使用本位，使用时需注意

8.4.2 外设中断

使能外设中断必须将GIE和PEIE置1，同时将相应中断的使能位置1。Timer1中断在异步计数器模式和异步定时唤醒模式下可以唤醒SLEEP。

INTCON

位编号	7	6	5	4	3	2	1	0
R/W	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	x
位符号	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

位编号	位符号	说明
7	GIE	全局中断使能位 0: 禁止所有中断 1: 使能所有未屏蔽的中断
6	PEIE	外设中断使能位 0: 禁止所有外设中断 1: 使能所有未屏蔽的外设中断
5-0	其它模块使用	本功能不使用本位，使用时需注意

PIE1

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R	R/W	R/W	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	ADIE	-	SPIIE	IICIE	-	T2IE	T1IE

位编号	位符号	说明
7	-	保留位
6	ADIE	ADC 中断使能位 0: 禁止 ADC 中断 1: 使能 ADC 中断
5	-	保留位
4	SPIIE	SPI 中断允许位 0: 禁止 SPI 中断 1: 允许 SPI 中断
3	IICIE	IIC 中断允许位 0: 禁止 IIC 中断 1: 允许 IIC 中断
2	-	保留位
1	T2IE	T2 计数寄存器与 PR2 匹配中断使能位 0: 禁止 T2 匹配中断 1: 使能 T2 匹配中断
0	T1IE	T1 溢出中断使能位 0: 禁止 T1 溢出中断 1: 使能 T1 溢出中断

PIE2

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R
复位值	0	0	0	0	0	0	0	0
位符号	CTKIE	WDTIE	PWMIE	-	-	UARTIE	EEPIE	-

位编号	位符号	说明
7	CTKIE	CTK 中断使能位 0: 禁止 CTK 中断 1: 使能 CTK 中断
6	WDTIE	WDT 中断使能位 0: 禁止 WDT 中断 1: 使能 WDT 中断 注: 使能 WDTIE 后, WDT 溢出后将不再产生复位, 而是产生中断
5	PWMIE	PWM 中断使能位 0: 禁止 PWM 中断 1: 使能 PWM 中断
4-3	-	保留位
2	UARTIE	UART 中断使能位 0: 禁止 UART 中断 1: 使能 UART 中断
1	EEPIE	EEP 中断使能位 0: 禁止 EEPROM 写中断 1: 使能 EEPROM 写中断
0	-	保留位

PIR1

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	ADIF	MODF	SPIIF	IICIF	-	T2IF	T1IF

位编号	位符号	说明
7	-	保留位
6	ADIF	ADC 中断标志位 0: 无 ADC 转换中断 1: 转换结束后, 硬件置 1, 可用于中断请求 (必须软件清 0)
5	MODF	SPI 模式故障标志位 0: 软件写 1 清 0 1: SS 引脚电平与 SPI 模式不一致时, 硬件置 1 (且立即切成从机模式), 也做中断请求标志位
4	SPIIF	SPI 传输完成标志位 0: 软件写 1 清 0 1: 一次传送完成时, 硬件置 1, 也做中断请求标志位
3	IICIF	IIC 串行中断标志位 0: 没有 IIC 串行中断发生 1: 产生 IIC 通信状态码中除 0F8H 之外的状态码时置 1。必须软件清 0
2	-	保留位
1	T2IF	Timer2 计数寄存器与 PR2 匹配中断标志位 0: Timer2 未发生匹配 1: Timer2 发生匹配 (必须用软件清零)
0	T1IF	Timer1 溢出中断标志位, Timer1 计数寄存器在 FFFFh 至 0000h 时产生溢出信号 0: Timer1 计数寄存器未溢出 1: Timer1 计数寄存器溢出 (必须由软件清 0)

PIR2

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R
复位值	0	0	0	0	0	0	0	0
位符号	CTKIF	WDTIF	PWM0IF	-	RXIF	TXIF	EEPIF	-

位编号	位符号	说明
7	CTKIF	电容检测中断标志位 0: 无转换完成中断或软件清 0 1: 检测结束后, 硬件置 1, 可用于中断请求
6	WDTIF	WDT 中断标志位 0: 软件清 0 1: WDT 计数器溢出, 由硬件置 1
5	PWM0IF	PWM0 中断标志位 0: 软件清 0 1: PWM0 周期计数器溢出, 由硬件置 1
4	-	保留位
3	RXIF	接收中断请求中断标志位 0: 软件清 0 1: 方式 0 时, 当串行接收数据第 8 位结束时, 由硬件自动置 1, 其它方式时, 串行接收到停止位开始时刻由硬件置 1
2	TXIF	发送中断请求中断标志位 0: 软件清 0 1: 方式 0 时, 当串行发送数据第 8 位结束时, 由硬件自动置 1, 其它方式时, 在停止位开始发送时由硬件置 1
1	EEPIF	EEPROM 写中断标志位 0: 软件清 0 1: EEPROM 写完成, 由硬件置 1
0	-	保留位

IOCB

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0

位编号	位符号	说明
7-0	IOCB[7:0]	PORTBx 电平变化中断使能位 0: 该端口禁止中断电平变化 1: 该端口允许中断电平变化

9 定时器/计数器

9.1 看门狗定时器

HC18M002/003的看门狗定时器与Timer0定时器/计数器共用一个预分频器。当PSA为0时，看门狗定时器每72ms（典型值）产生一个溢出信号；当PSA为1时，WDT溢出时间由预分频器OPTION[2:0]设置决定，具体请参考Timer0定时器/计数器。

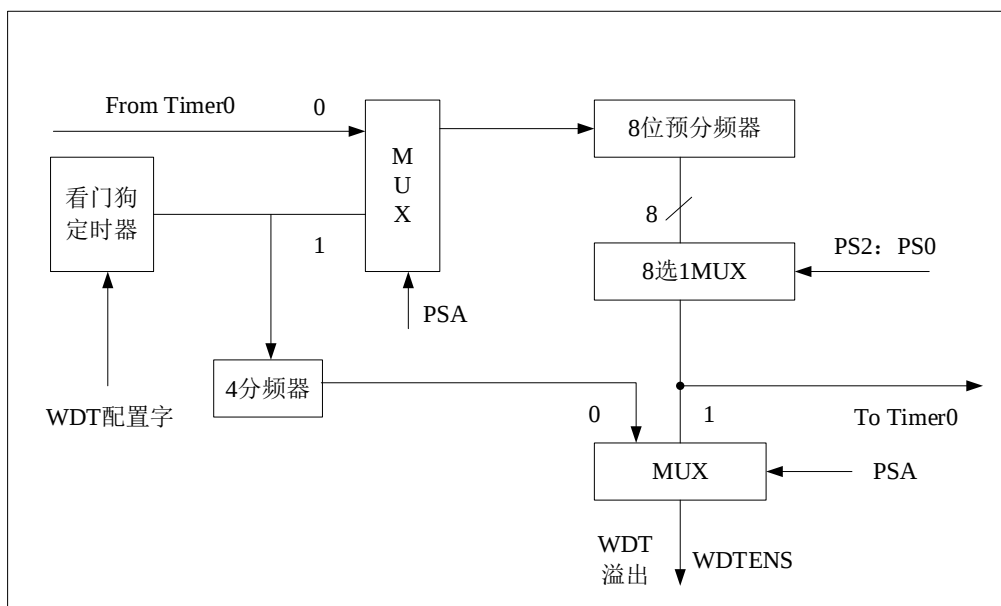


Figure 9-1 看门狗定时器和预分频器框图

9.1.1 WDT 定时器相关寄存器

9.1.1.1 WDT使能寄存器PCON

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R
复位值	0	0	0	1	0	0	0	0
位符号	-	-	-	WDTENS	-	-	-	-

位编号	位符号	说明
7-5	-	保留位
4	WDTENS	WDT 模块使能位 0: 禁止 WDT 定时器 1: 允许 WDT 定时器
3-0	-	保留位

注意:

- 1、看门狗的使能逻辑 看门狗使能 = 芯片配置字使能（WDTEN）&软件使能（WDTENS）
- 2、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;
- 3、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情况;
- 4、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。

WDT预分频比选择

PS[2:0]	WDT预分频比	WDT溢出时间（典型值）
000	1: 2	16ms
001	1: 4	32 ms
010	1: 8	64ms
011	1: 16	128ms
100	1: 32	256ms
101	1: 64	512ms
110	1: 128	1024ms
111	1: 256	2048ms

9.2 定时器/计数器 T0

9.2.1 定时器/计数器 0 特性及系统框图

Timer0 定时器/计数器模块具有如下功能：

- 8 位可编程定时器
- 外部事件计数器
- 绿色模式定时唤醒

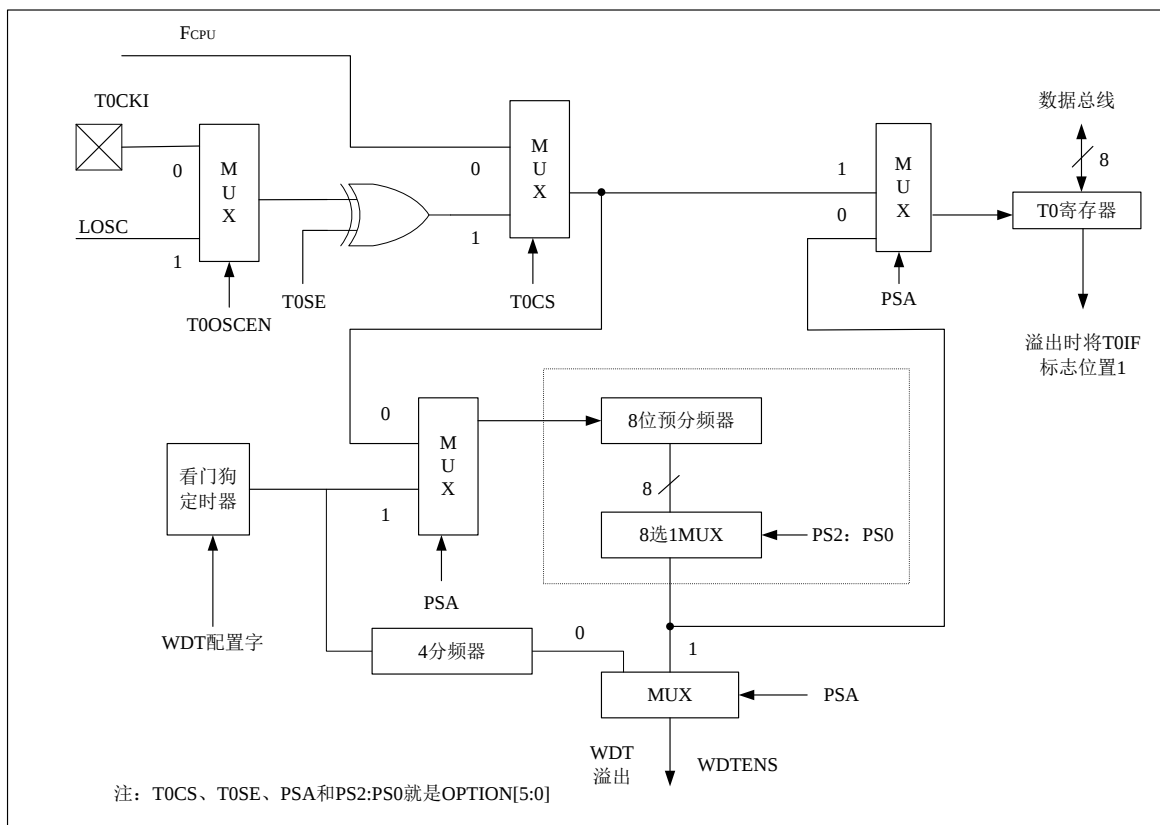


Figure 9-2 TIMER0 方式 0 功能框图

9.2.2 定时器/计数器 T0 相关寄存器

9.2.2.1 定时器T0使能寄存器OSCCON

OSCCON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	T0OSCEN	-	-	-	-	-	HXEN	SCS

位编号	位符号	说明
7	T0OSCEN	定时器 0 模块使能位 0: 禁止定时器 0 1: 允许定时器 0
6-2	-	保留位
1-0	其它模块使用	本功能不使用本位，使用时需注意

9.2.2.2 定时器T0控制寄存器OPTION

OPTION

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	RB PUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

位编号	位符号	说明
7-6	其它模块使用	本功能不使用本位，使用时需注意
5	T0CS	T0 模式选择寄存器，具体见下表
4	T0SE	T0 模式选择寄存器，具体见下表
3	PSA	T0 模式选择寄存器，具体见下表
2	PS2	T0 分频选择寄存器，具体见下表
1	PS1	T0 分频选择寄存器，具体见下表
0	PS0	T0 分频选择寄存器，具体见下表

看门狗定时器与Timer0定时器/计数器共用一个预分频器，当PSA=1预分频器分配给WDT时，Timer0在所选中时钟源的每个周期递增；当PSA=0预分频器分配给Timer0时，Timer0根据PS[2:0]值选择的预分频时钟递增。

Timer0的预分频器不可寻址，当预分频器分配给Timer0时，对Timer0计数寄存器的写操作可以对预分频器清0。

Timer0预分频比选择

PS[2:0]	Timer0预分频比
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

Timer0 工作模式选择

T0CS	T0OSCEN	T0SE	Timer0工作状态
0	x	x	定时器模式，计数时钟 Fcpu，休眠和绿色模式下停止
1	0	0	计数器模式，计数时钟 T0CKI，上升沿计数休眠模式下工作，溢出中断可唤醒 SLEEP
1	0	1	计数器模式，计数时钟 T0CKI，下降沿计数休眠模式下工作，溢出中断可唤醒 SLEEP
1	1	0	定时唤醒模式，计数时钟LOSC，上升沿计数绿色模式下工作，溢出中断可唤醒SLEEP
1	1	1	定时唤醒模式，计数时钟LOSC，下降沿计数绿色模式下工作，溢出中断可唤醒SLEEP

注意：Timer0 工作模式的选择需符合上表描述，选择除上表以外情况可能会造成程序运行混乱，请谨慎操作。

9.2.2.3 定时器T0数据寄存器 T0

TLx (x = 0,1)

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	T0							

位编号	位符号	说明
7-0	T0	T0 数据寄存器

9.3.2 定时器/计数器 T1 相关寄存器

9.3.2.1 定时器T1控制寄存器 T1CON

T1CON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	T1CS1	T1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	-	T1ON

位编号	位符号	说明
7	T1CS1	T1 时钟/模式选择寄存器 1，具体选择见下表
6	T1CS0	T1 时钟/模式选择寄存器 0，具体选择见下表
5	T1CKPS1	定时器 1 时钟源预分频比选择位 1，具体选择见下表
4	T1CKPS0	定时器 1 时钟源预分频比选择位 0，具体选择见下表
3	T1OSCEN	T1 时钟使能寄存器，具体选择见下表
2	T1SYNC	T1 模式选择寄存器，具体选择见下表
1	-	保留位
0	T1ON	定时器 1 模块使能位 0：禁止定时器 1 1：允许定时器 1

PR1CON

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	T1CKPS3	T1CKPS2	-	PR1EN

位编号	位符号	说明
7-4	-	保留位
3	T1CKPS3	定时器 1 时钟源预分频比选择位 3，具体选择见下表
2	T1CKPS2	定时器 1 时钟源预分频比选择位 1，具体选择见下表
1	-	保留位
0	PR1EN	定时器 1 周期使能位 0：Timer1 为 16 Timer 1：Timer1 为可设周期的 8 位 Timer

Timer1时钟源选择

T1CS1	T1CS0	T1OSCEN	时钟源
0	0	x	指令时钟（Fcpu）
0	1	x	系统时钟（Fosc）
1	0	0	T1CKI 引脚上的外部时钟
1	0	1	低频系统时钟

注意：Timer1 时钟源的选择需符合上表描述，选择除上表以外情况会造成程序运行不正常，请谨慎操作。

Timer1输入时钟预分频比选择

T1CKPS[1:0]	Timer1 预分频比
00	1 : 1
01	1 : 2
10	1 : 4
11	1 : 8
0100	1 : 16
0101	1 : 32
0110	1 : 64
0111	1 : 128
1xxx	1 : 256

注：

- 1、Timer1的预分频器不可寻址，可以通过对Timer1计数寄存器写操作将预分频器清0。
- 2、若使用定时器计数器功能时，配置为1:16及以上分频时请先将PR1CON寄存器的PR1EN置1，PR1L寄存器写入0xFF，再进行Timer1输入时钟预分频比选择。

Timer1工作模式选择

T1ON	T1CS[1:0]	T1OSCEN	T1SYNC	Timer1工作模式
1	00	x	x	定时器模式，休眠和绿色模式下停止
1	01	x	x	定时器模式，休眠和绿色模式下停止
1	10	0	0	同步计数器模式，休眠模式下停止
1	10	0	1	异步计数器模式，休眠模式下工作，溢出中断可唤醒SLEEP
1	10	1	0	同步定时唤醒模式，绿色模式下停止，溢出中断不能唤醒SLEEP
1	10	1	1	异步定时唤醒模式，绿色模式下工作，溢出中断可唤醒SLEEP

注意：

- 1、T1 为 16 位计时器，在溢出中断重新赋值时应先 T1H，后 T1L，避免 T1L 在操作中的进位被覆盖；清空时则应先 T1L 后 T1H，避免 T1L 进位意外进入 T1H 造成清空失败。
- 2、Timer1 工作于同步计数器模式和同步定时唤醒模式时，不能唤醒 SLEEP 或绿色模式
- 3、Timer1 工作模式的选择需符合上表描述，选择除上表以外情况可能会造成程序运行混乱，请谨慎操作

9.3.2.2 定时器T1数据寄存器 T1L、T1H

T1L

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	T1L							

位编号	位符号	说明
7-0	T1L	T1 数据寄存器低字节

T1H

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	T1H							

位编号	位符号	说明
7-0	T1H	T1 数据寄存器高字节

9.4 定时器 2

Timer2 定时器具有 8 位预分频器和 8 位周期寄存器（PR2），Timer2 定时器的输入时钟为指令时钟 FCPU，输入时钟通过预分频器产生 Timer2 计数时钟，当计数到与周期寄存器（PR2）的值相同时，在下一指令周期产生 Timer2 溢出信号，可根据实际需要选择不同的预分频比及设置周期寄存器的值，产生不同溢出时间。

Timer2 定时器的输入时钟为指令时钟 FCPU，输入时钟通过预分频器产生 Timer2 计数信号，当计数到与周期寄存器（PR2）的值相同时产生 Timer2 溢出信号。

Timer2 溢出时间 = (PR2 + 1) * 预分频比/Fcpu。

9.4.1 定时器 T2 系统框图

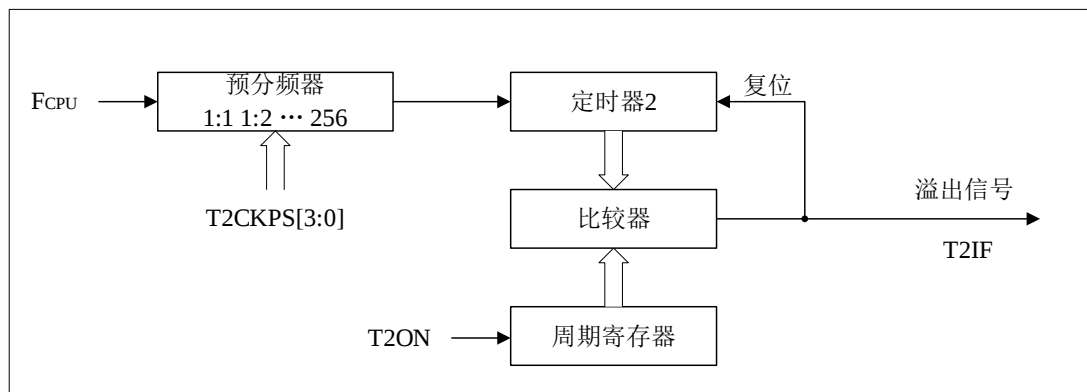


Figure 9-4 TIMER2 系统框图

9.4.2 定时器 T2 相关寄存器

9.4.2.1 定时器2控制寄存器T2CON

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-	T2CKPS3	T2CKPS2	T2CKPS1	T2CKPS0	T2CON	-	-

位编号	位符号	说明
7	-	保留位
6-3	T2CKPS[3:0]	Timer2 预分频比 0000: 1: 1 0001: 1: 2 0010: 1: 4 0011: 1: 8 0100: 1: 16 0101: 1: 32 0110: 1: 64 0111: 1: 128 1xxx: 1: 256
2	T2CON	定时器 2 模块使能位 0: 禁止定时器 2 1: 允许定时器 2
1-0	-	保留位

9.4.2.2 定时器2计数寄存器T2

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	T2							

位编号	位符号	说明
7-0	T2	T2 计数寄存器

9.4.2.3 定时器2周期寄存器PR2

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PR2							

位编号	位符号	说明
7-0	PR2	T2 周期寄存器

10 脉宽调制PWM

10.1 PWM 特性

- 1 组带死区互补 PWM 或 2 路独立 PWM 输出
- 提供每个 PWM 周期溢出中断
- 输出极性可选择
- 提供出错侦测功能可紧急关闭 PWM 输出
- PWM 工作时钟源可设定时钟分频比
- PWM 可做定时器/计数器使用

HC18M002/003 集成了 1 组 12 位 PWM 模块 PWM0, PWM0 的计数器由 PWM0_EN 和 PWM01_EN 来控制, 只要使能 PWM0_EN 和 PWM01_EN 中的任何一个, 计数器就会启动, 计数器的时钟源通过 PWM0C 控制寄存器里的 CK0 来选择。

当需要从芯片管脚输出 PWM 波形时, 还需要将端口设置为输出模式, 同时需要设置复用端口映射寄存器, 将 PWM 映射到相应的端口上。如果不希望从芯片管脚上输出 PWM 波形, 可以不用设置端口模式和复用端口映射寄存器, 这时候 PWM 的计数器可以当一个定时器来使用, 当计数器溢出时, 如果中断允许也会产生 PWM 中断。

如果 EFLT0 置 1, PWM0 输出和其互补输出可由 FLT0 引脚输入信号变化自动关闭。一旦检测到 FLT0 引脚输入有效电平, PWM 输出会立即关闭, 但 PWM 内部计数器仍在继续运行, 这样方便在 FLT0 引脚错误去除后继续 PWM 输出。在 FLT0 输入信号有效期间, FLT0S 位无法清除。只有当 FLT0 输入信号消失后, 才能软件清除 FLT0S 状态位, 此时 PWM 恢复正常输出。

10.2 PWM 输出模式

1 组 PWM0 输出为 PWM0/PWM01，通过控制相关寄存器可使每对 PWM 输出配置成互补输出模式或独立输出模式。

10.2.1 互补输出模式

当 PWM0M 置 0：PWM0 将工作在互补输出模式，互补输出模式时，需要使能 PWM0_OEN 和 PWM01_OEN，此时可以控制对应的周期寄存器、占空比寄存器及死区时间寄存器，从而控制互补波形的输出。互补输出时可选择 PWM0&PWM01 输出极性，方便用户各种电平驱动需求。

10.2.2 独立输出模式

当 PWM0M 置 1：PWM0 将工作在独立输出模式，独立输出模式时，可以控制相关寄存器使能对应 PWM 端口单一输出或同时输出，同时让 PWM0&PWM01 输出时，其周期相同但占空比可单独设置。此时互补输出模式时占空比寄存器将控制 PWM0 的占空比，死区时间控制寄存器将控制 PWM01 的占空比，独立输出时也可控制 PWM0&PWM01 输出极性，方便用户各种电平驱动需求。

10.3 PWM 相关寄存器

10.3.1 PWM 使能寄存器 PWMEN

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	PWM01EN	-	-	-	PWM0EN

位编号	位符号	说明
7-5	-	保留位
4	PWM01EN	PWM01 使能位 0: 禁止 PWM01 1: 使能PWM01
3-1	-	保留位
0	PWM0EN	PWM0 使能位 0: 禁止 PWM0 1: 使能PWM0

10.3.2 FLT 模式寄存器 FLTMODE

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	EFLT0	-	-	-	-	FLT0_MODE	

位编号	位符号	说明
7	-	保留位
6	EFLT0	PWM0 FLT0 控制引脚使能位 0: 禁止故障检测, GPIO 功能或其它功能 1: 允许故障检测, PWM0 故障检测输入引脚 注: 互补输出模式及独立输出模式都可受故障检测脚控制
5-2	-	保留位
1-0	FLT0_MODE	PWM0 故障输出预定状态选择位 00: PWM0&PWM01故障期间均为低电平 01: PWM0故障期间低电平, PWM01故障期间高电平 10: PWM0故障期间高电平, PWM01故障期间低电平 11: PWM0&PWM01故障期间均为高电平

10.3.3 PWM 模式寄存器 PWMM

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R/W
复位值	0	0	0	0	0	0	0	1
位符号	-	-	-	PWM0M	-	-	-	RELOAD0

位编号	位符号	说明
7-5	-	保留位
4	PWM0M	PWM0 工作模式选择位 0: PWM0&PWM01 工作于互补输出模式 1: PWM0&PWM01 工作于独立输出模式 注: 修改 PWM0 工作模式时建议先关闭 PWM0 模块
3-1	-	保留位
0	RELOAD0	PWM0 自动重载使能位 0: 禁止自动重载 1: 使能自动重载 注: 默认值为 1, 默认状态下修改周期、占空比、死区等参数后, 参数会自动重载, 并在下一个 PWM 周期使用这些参数。 在修改周期、占空比、死区等参数前设置此位为 0 将禁止自动重载, PWM 不会使用任何新的参数, 输出状态保持不变, 在软件修改参数完成, 再设置此位为 1, 新修改的参数将在下一个 PWM 周期统一使用。

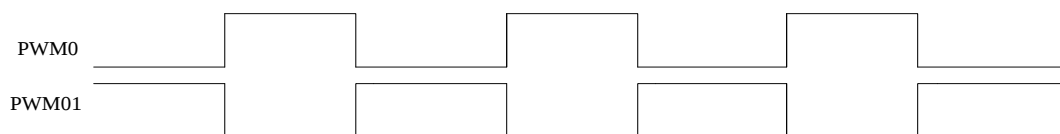
10.3.4 PWM0 相关寄存器

10.3.4.1 PWM0控制寄存器PWM0C

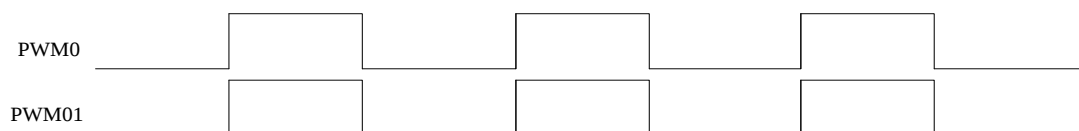
位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	PWM0IE	PWM0IF	FLT0S	FLT0C	PWM0S		CK0	

位编号	位符号	说明
7	PWM0IE	PWM0 中断允许位 0: 禁止 PWM0 中断 1: 允许 PWM0 中断
6	PWM0IF	PWM0 中断标志位 0: 软件清 0 1: PWM0 周期计数器溢出, 由硬件置 1
5	FLT0S	PWM0 FLT 状态位 0: PWM 正常状态, 软件清 0 1: PWM 输出关闭, 硬件置 1
4	FLT0C	PWM0 FLT 引脚配置位 0: FLT0 为低电平时, PWM 输出关闭 1: FLT0 为高电平时, PWM 输出关闭
3-2	PWM0S	PWM0 和 PWM01 输出模式选择位 00: PWM0和PWM01均为高有效 01: PWM0为高有效, PWM01为低有效 10: PWM0为低有效, PWM01为高有效 11: PWM0 和 PWM01 均为低有效 注: 对于独立模式, 输出模式选择位同样有效, 但与互补模式不同的是: 有效期间为占空比期间; 而互补模式中对于 PWM0 的有效期间为占空比期间, PWM01 的有效期间为占空比的互补期间。
1-0	CK0	PWM0 时钟源选择位 00: RC32M/1 01: RC32M /8 10: RC32M /32 11: RC32M /128

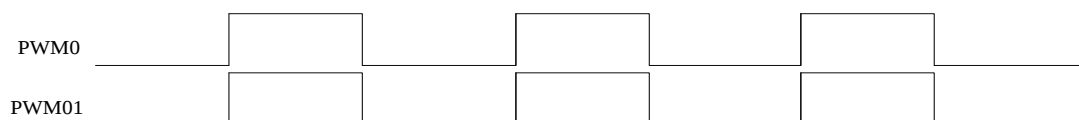
PWM0S=00& PWM0M=0: PWM0和PWM01工作于互补模式且均为高有效



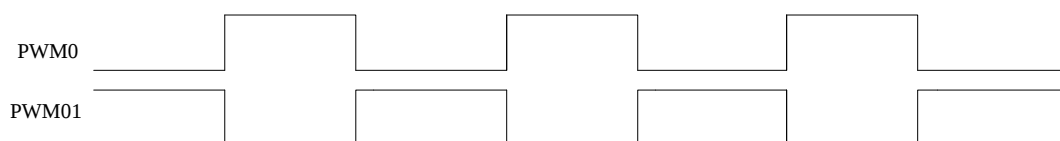
PWM0S=00& PWM0M=1: PWM0和PWM01工作于独立模式且均为高有效



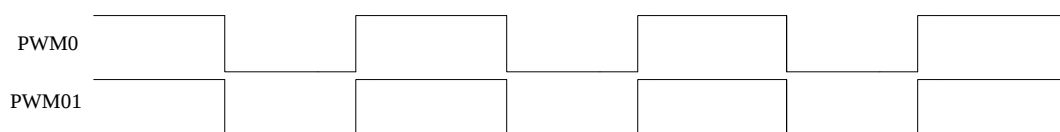
PWM0S=01& PWM0M=0: PWM0和PWM01工作于互补模式且PWM0为高有效、PWM01为低有效



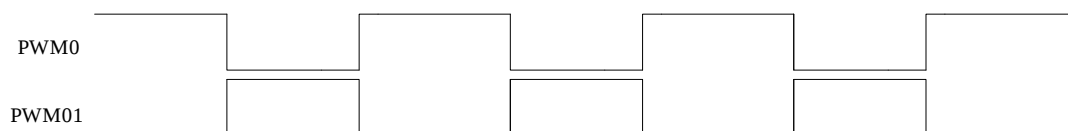
PWM0S=01& PWM0M=1: PWM0和PWM01工作于独立模式且PWM0为高有效、PWM01为低有效



PWM0S=10& PWM0M=0: PWM0和PWM01工作于互补模式且PWM0为低有效、PWM01为高有效



PWM0S=10& PWM0M=1: PWM0和PWM01工作于独立模式且PWM0为低有效、PWM01为高有效



10.3.4.2 PWM0周期寄存器PWM0PL、PWM0PH

PWM0PL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PWM0PL[7:0]							

位编号	位符号	说明
7-0	PWM0PL[7:0]	PWM0 周期寄存器低 8 位

PWM0PH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM0PH[3:0]			

位编号	位符号	说明
7-4	-	保留位
3-0	PWM0PH[3:0]	PWM0 周期寄存器高 4 位

注：修改PWM0周期时先修改高位，后修改低位，读时不受限制，例如

- (1) PWM0PH = 0x05;
- (2) PWM0PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508
- (3) PWM0PH = 0x06; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0508
- (4) PWM0PL = 0x08; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0608
- (5) PWM0PL = 0x09; //此时 PWM 计数器溢出，则下一个周期开始周期计算数据为 0x0609

显然只要修改 PWM 周期，无论低位寄存器是否需要修改，低位都必须写入一次，且周期修改都只会在下一个 PWM 周期才会生效。

$PWM0周期 = [PWM0PH : PWM0PL] * PWM0工作时钟源周期$

10.3.4.3 PWM0占空比寄存器PWM0DL、PWM0DH

PWM0DL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PWM0DL[7:0]							

位编号	位符号	说明
7-0	PWM0DL[7:0]	PWM0 占空比寄存器低 8 位

PWM0DH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM0DH[3:0]			

位编号	位符号	说明
7-4	-	保留位
3-0	PWM0DH[3:0]	PWM0 占空比寄存器高 4 位

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。

$PWM0 \text{ 占空比} = [PWM0DH : PWM0DL] * PWM0 \text{ 工作时钟周期}$

10.3.4.4 PWM0死区时间寄存器PWM0DTL、PWM0DH

PWM0DTL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	PWM0DTL[7:0]							

位编号	位符号	说明
7-0	PWM0DTL[7:0]	PWM0 死区时间寄存器低 8 位

PWM0DTH

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	PWM0DTH[3:0]			

位编号	位符号	说明
7-4	-	保留位
3-0	PWM0DTH[3:0]	PWM0 死区时间寄存器高 4 位

当 PWM0_M=1 时，PWM0 工作在 2 路独立模式，此时的死区时间寄存器被用来当做 PWM01 的占空比寄存器，即独立模式的 PWM0 可以产生 2 路周期相同，但占空比可以不同的 PWM 波形。

互补模式下：PWM0 死区时间 = [PWM0DTH : PWM0DTL] * PWM0 工作时钟周期

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

独立模式下：PWM01 占空比时间 = [PWM0DTH : PWM0DTL] * PWM0 工作时钟周期

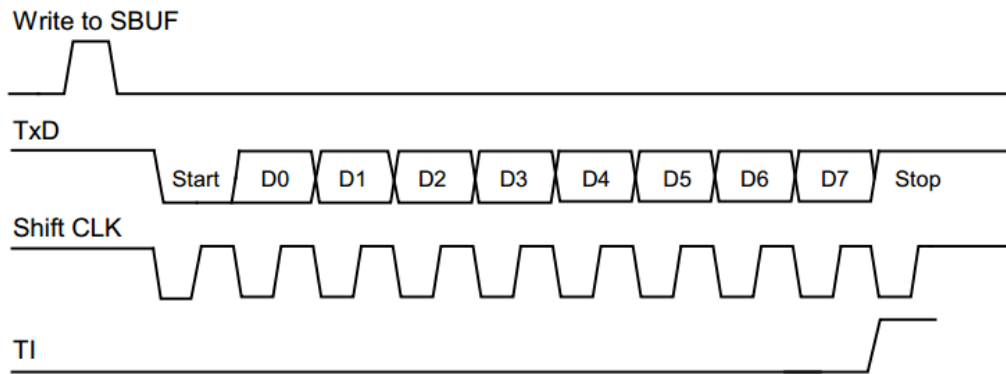


Figure 11-2 Send Timing of Mode 1

只有REN置1时才允许接收。当RXD引脚检测到下降沿时串行口开始接收串行数据。为此，CPU对RXD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位，这有助于16分频计数器与RXD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RXD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RXD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8个数据位和1个停止位（包含错误的停止位，详细见寄存器SM2位说明）移入之后，移位寄存器的内容和停止位(包含错误的停止位)被分别装入SBUF和RB8中，RI置1，但必须满足下列条件：

- (1) RI = 0
- (2) SM2 = 0或者接收的停止位= 1

如果这些条件被满足，那么停止位（包含错误的停止位）装入 RB8，8 个数据位装入 SBUF，RI 被置位。否则接收的帧会丢失。这时，接收器将重新去探测 RXD 端是否另一个下降沿。用户必须用软件清零 RI，然后才能再次接收。

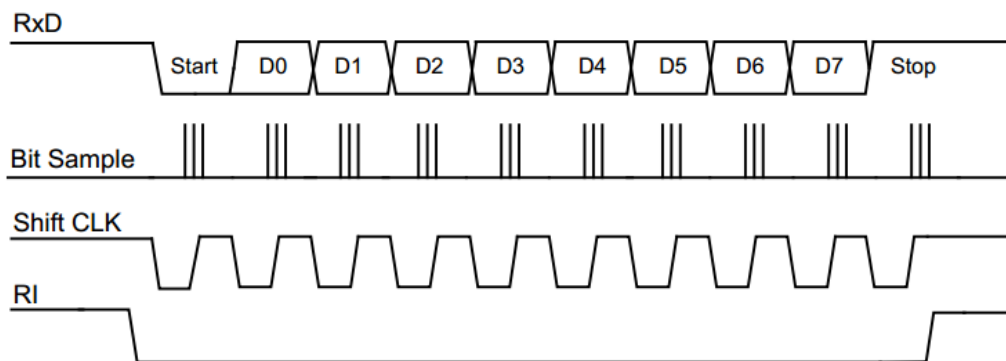


Figure 11-3 Receive Timing of Mode 1

11.2.2 方式 1: 9 位 UART, 可变波特率, 异步全双工

这个方式使用异步全双工通信中的 11 位。一帧由一个起始位（逻辑 0），8 个数据位（低位在前），一个可编程的第 9 数据位和一个停止位（逻辑 1）组成。方式 1 支持多机通信，在数据传送时，第 9 数据位（TB8 位）可以写 0 或 1，例如，可写入 PSW 中的奇偶位 P，或用作多机通信中的数据/地址标志位。当接收到数据时，第 9 数据位移入 RB8 而停止位不保存。

任何将 SBUF 作为目标寄存器的写操作都会启动发送，同时也将 TB8 载入到发送移位寄存器的第 9 位中。实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TXD 引脚上移出，然后是 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TXD 引脚上移出，在停止位开始发送时 TI 标志置位。

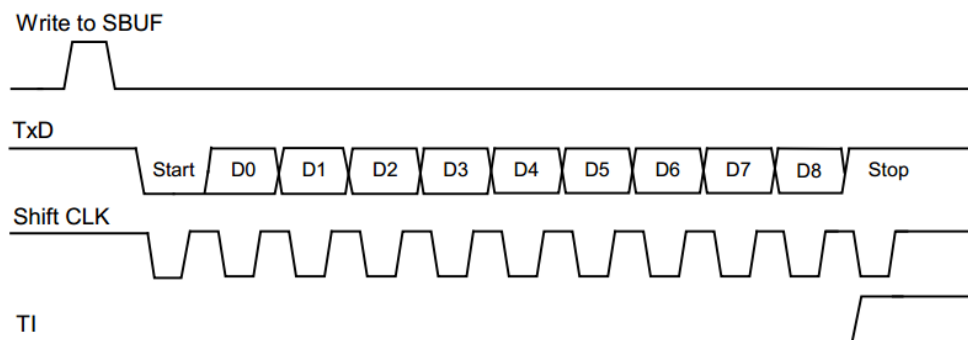


Figure 11-4 Send Timing of Mode 2

只有 REN 置位时才允许接收。当 RXD 引脚检测到下降沿时串行口开始接收串行数据。为此，CPU 对 RXD 不断采样，采样速率为波特率的 16 倍。当检测下降沿时，16 分频计数器立即复位。这有助于 16 分频计数器与 RXD 引脚上的串行数据位同步。16 分频计数器把每一位的时间分为 16 个状态，在第 7、8、9 状态时，位检测器对 RXD 端的电平进行采样。为抑制噪声，在这 3 个状态采样中至少有 2 次采样值一致数据才被接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RXD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9 个数据位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，但必须满足下列条件：

- (1) RI = 0
- (2) SM2 = 0 或者接收的第 9 位 = 1

如果这些条件被满足，那么第 9 位移入 RB8，8 位数据移入 SBUF。但还需要检测停止位，只有停止位为 1，才能置位 RI，如果停止位为 0，则 RI 不会置位。

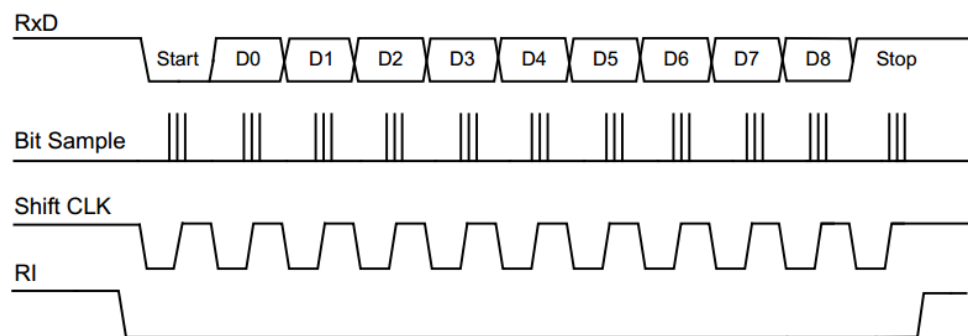


Figure 11-5 Receive Timing of Mode 2

11.3 波特率

UART 自带一个波特率发生器，它实质上就是一个 16 位递增计数器。

波特率公式如下：

$$\text{Baud} = \frac{F_{\text{cpu}}}{16 \times (65536 - \text{SBRT})}, \quad \text{SBRT} = [\text{SBRTH} : \text{SBRTL}]$$

11.4 帧出错检测

3个错误标志位被置位后，只能通过软件清零，尽管后续接收的帧没有任何错误也不会自动清零。

11.4.1 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（TXCOL位）置1。如果发生了冲突，新数据会被忽略，不能被写入发送缓冲器（即不影响传送）。

11.4.2 接收溢出

RI置1，接收缓冲器中的数据未被读取，RI被清0，又开始新的数据接收，若在新的数据接收完成前（RI置1）还未读取之前接收缓冲区中的数据，在那么接收溢出位（RXROV位）置位。如果发生了接收溢出，接收缓冲器中原来的数据不影响，后面的数据则丢失。

11.4.3 帧出错

如果检测到一个无效（低）停止位，那么帧出错位（FE位）置1。

11.5 UART1 相关寄存器

11.5.1 UART1 控制寄存器 SCON、SCON2

SCON

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0
位符号	FE	RXROV	TXCOL	REN	TB8	RB8	-	-

位编号	位符号	说明
7	FE	侦错误检测位 0: 无侦错误或软件清 0 1: 有侦错误, 硬件置 1
6	RXROV	接收溢出标志位 0: 无接收溢出或软件清 0 1: 接收溢出, 硬件置 1
5	TXCOL	发送冲突标志位 0: 无发送冲突或软件清 0 1: 有发送冲突, 硬件置 1
4	REN	串行接收使能控制位 0: 禁止串行接收 1: 允许串行接收
3	TB8	方式 1 时, 为要发送的第 9 位数据, 由软件置 1 或清 0
2	RB8	方式 1 时, 为接收到的第 9 位数据, 作为奇偶校验位或地址帧/数据帧的标志位
1-0	-	保留位

SCON2

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	BRTR	-	-	URATEN	SM1	SM2

位编号	位符号	说明
7-6	-	保留位
5	BRTR	独立波特率发生器 BRT 运行控制位 0: 停止独立波特率发生器 BRT 工作 1: 启动独立波特率发生器 BRT 工作
4-3	-	保留位
2	URATEN	UART 使能位: 0: 禁止 UART 1: 使能 UART
1	SM1	串口工作方式选择位, 详细见下表 0: 8 位 UART 1: 9 位 UART
0	SM2	第九位检测使能位 0: 在方式 0 下, 不检测停止位, 停止位无论是 0 还是 1 都会置位 RI 在方式1下, 不检测第9位, 第9位无论是0还是1都会置位RI 1: 在方式 0 下, 只有停止位为 1 才能置位 RI 在方式1下, 只有第9位为1才能置位RI

11.5.2 UART1 数据缓冲寄存器 SBUF

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	SBUF[7:0]							

位编号	位符号	说明
7-0	SBUF[7:0]	串口缓冲寄存器 写为需要发送的数据, 读为接收到的数据

11.5.3 UART 波特率发生器 BRTL、BRTH

BRTL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	SBRTL[7:0]							

位编号	位符号	说明
7-0	SBRTL[7:0]	波特率发生器寄存器 BRT 低 8 位，用于保存重装时间常数

BRTH

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	SBRTH[7:0]							

位编号	位符号	说明
7-0	SBRTH[7:0]	波特率发生器寄存器 BRT 高 8 位，用于保存重装时间常数

12 串行外部设备接口SPI

12.1 SPI 特性

- 全双工，三/四线同步传输
- 主从机操作
- 4级可编程主时钟频率
- 极性相位可编程的串行时钟
- 可选择数据传输方向
- 写冲突及接收溢出标志
- 带MCU中断的主模式故障出错标志
- 带MCU中断的传输结束标志
- 主模式支持高达0.5Mbps通信速率（ $F_{cpu}=8MHz$ ），从模式下通信速率须在 $F_{cpu}/16$ 及 $F_{cpu}/16$ 以下

12.2 SPI 信号描述

主输出从输入（MOSI）：该信号连接主设备和一个从设备，数据通过 MOSI 从主设备串行传送到从设备，主设备输出，从设备输入。

主输入从输出（MISO）：该信号连接主设备和一个从设备。数据通过 MISO 从从设备串行传入到主设备，从设备输出，主设备输入。若该设备为从设备且未被选时，从设备的 MISO 引脚处于高阻状态。

串行时钟（SCK）：该信号用作控制 MOSI 和 MISO 线上输入输出数据的同步移动，每 8 个时钟周期 MOSI 和 MISO 线上传送一个字节，如果从设备未被选中，SCK 信号将被此设备忽略。注意：只有主设备才能产生 SCK 信号。

从设备选择引脚（ $\overline{SS}$ ）：每个从属外围设备由一个从选择引脚 $\overline{SS}$ 选择，当引脚信号为低电平时，表明该从设备被选中。主设备可以通过软件控制连接于从设备 $\overline{SS}$ 引脚的端口电平选择每个从设备，很明显，只有一个主设备可以驱动通讯网络。为了防止 MISO 总线冲突，同一时间只允许一个从设备与主设备通讯。在主设备模式中， $\overline{SS}$ 引脚状态关联 SPI 状态寄存器 SPSTAT 中 MODF 标志位以防止多个主设备驱动 MOSI 和 SCK。

下列情况， $\overline{SS}$ 引脚可以作为普通端口或其它功能使用：

（1）设备作为主设备，SPI 控制寄存器 SPCTL 寄存器的 SSIG 位置 1。这种配置仅仅存在于通讯网络中只有一个主设备的情况，因此，SPI 状态寄存器 SPSTA 中 MODF 标志位不会被置 1。

（2）设备配置为从设备，SPI 控制寄存器 SPCTL 的 CPHA 位和 SSIG 位置 1。这种配置情况存在于只有一个主设备一个从设备的通讯网络中，因此，设备总是被选中的，主设备也不需要控制从设备的 $\overline{SS}$ 引脚选择其作为通讯目标。

从设备的 $\overline{SS}$ 引脚被使能时，其它主设备可通过使该引脚维持低电平，从而选中该从设备。为防止 MISO 总线冲突，原则上不允许两个及以上的从设备被选中。

主设备的 $\overline{SS}$ 引脚被使能时，若 $\overline{SS}$ 被拉低将置模式错误标志 MODF（可中断），且 MSTR 位也将被清 0，从而使该设备强制切换成从设备。

当 MSTR = 0（从模式）及 CPHA = 0 时，SSIG 必须为 0，因为此时数据传送需要 $\overline{SS}$ 引脚配合，才

能完成多数据传送。

12.3 SPI 时钟速率

在主模式下，SPI 的速率有 4 级选择，分别是系统时钟的 4、16、64 或 128 分频，可通过 SPCTL 寄存器的 SPR[1:0]位进行选择。

12.4 SPI 功能框图

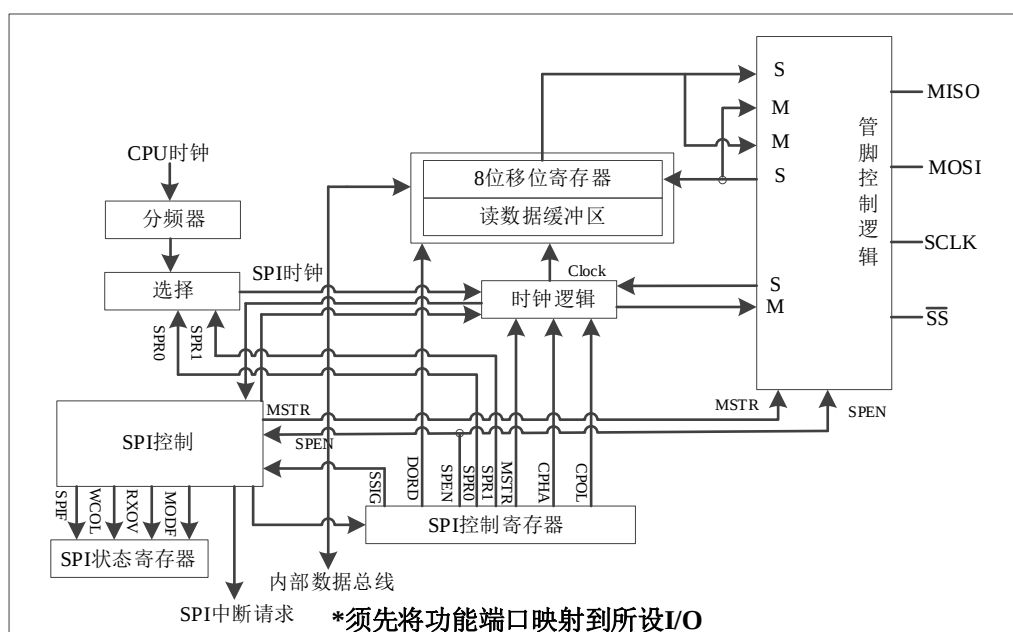


Figure 12-1 SPI 功能方框图

12.5 SPI 工作模式

SPI 可配置为主模式或从模式中的一种。SPI 模块的配置和初始化通过设置相关寄存器来完成。进一步设置相关寄存器即可完成数据传送。

在 SPI 通讯期间，数据同步地被串行的移进移出，串行时钟线（SCK）使两条串行数据线（MOSI&MISO）上数据的移动和采样保持同步。从设备选择线（ $\overline{SS}$ ）可以独立地选择从属设备；如果从设备没有被选中，则不能参与 SPI 总线上的活动。

当 SPI 主设备通过 MOSI 线传送数据到从设备时，从设备通过 MISO 线发送数据到主设备作为相应，从而实现在同一时钟下数据发送与接收的同步全双工传输。发送移位寄存器和接收寄存器使用相同的 SFR 地址，对 SPI 数据寄存器 SPDAT 进行写操作将写入发送移位寄存器，对 SPDAT 寄存器进行读操作将获得接收移位寄存器的数据。注：写入的数据不会影响到需要读出的数据

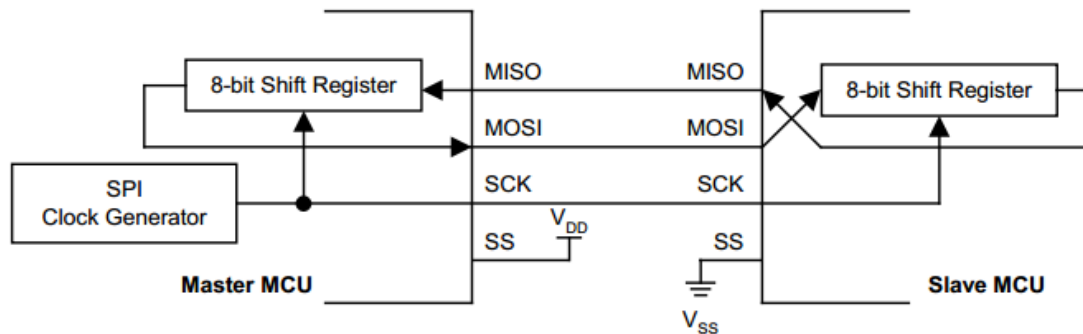


Figure 12-2 全双工主从互联图

主模式

(1) 模式启动

SPI 主设备控制 SPI 总线上的所有数据传送的启动。一个 SPI 总线中只允许一个主设备可以启动传送。

(2) 发送

在 SPI 主模式下，写一个字节数据到 SPI 数据寄存器 SPDAT，数据将会写入发送移位缓冲器。如果发送移位寄存器中已经存在一个数据或正在传送一个数据，那么主 SPI 将产生一个 WCOL 信号以表明写入太快。但是发送移位寄存器中的数据不会受到影响，发送也不会中断。

(3) 接收

当主设备通过 MOSI 线传送数据到从设备时，同时对应的从设备也可以通过 MISO 线将其发送移位寄存器的数据传送给主设备的接收移位寄存器，实现全双工操作。故 SPIF 标志置 1 即表示数据发送完成也表示数据接收完成。本 SPI 模块接收为双缓冲器，即数据可以在 SPIF 置 1 后读出，但必须在下一字节数据接收完成前读出，否则将置接收溢出标志 RXOV，如果发生接收溢出，则后面的数据将不会被移入接收寄存器，接收溢出时，SPIF 可正常置 1。

从模式

(1) 模式启动

将 MSTR 置 0（若 $\overline{SS}$ 被使能则必须拉低）时，设备处于从模式下运行，数据传送过程中设备模式不能改变（ $\overline{SS}$ 引脚必须维持低电平），否则数据传送将失败（SPIF 不会被置 1）。

(2) 发送

SPI 从设备下不能启动数据传送，所以 SPI 从设备必须在主设备开始一次新的数据传送之前将要传送给主设备的数据写入发送移位寄存器。若发送前未写入数据到发送移位寄存器，从设备将传送数据“0x00”给主设备。若写入数据时发送移位寄存器已经存在数据（或发生在传送过程中），那么 SPI 从设备的 WCOL 标志位将置 1，表示发生写 SPDAT 冲突。但是移位寄存器的数据不受影响，传送也不会被中断，传送完成 SPIF 将被置 1。

(3) 接收

从模式下，按照主设备控制的 SCK 信号，数据通过 MOSI 引进移入，当计数器计数 SCK 边缘数到 8 时，表示一个字节数据接收完毕，SPIF 将置 1，数据可以通过此时读取 SPDAT 寄存器获得，但必须在下一数据接收完成前被读出，否则将置接收溢出标志 RXOV，如果发生接收溢出，则后面的数据将不会被移入接收寄存器，接收溢出时，SPIF 可正常置 1。

12.6 SPI 传送形式

通过软件设置寄存器的 CPOL 位和 CPHA 位,用户可以选择 SPI 时钟极性和相位的四种组合方式。CPOL 位定义时钟的极性,即空闲时的电平状态。CPHA 位定义时钟相位,即定义允许数据移位采样的时钟边沿。在通信的两个主从设备中,时钟极性相位设置应当保持一致。

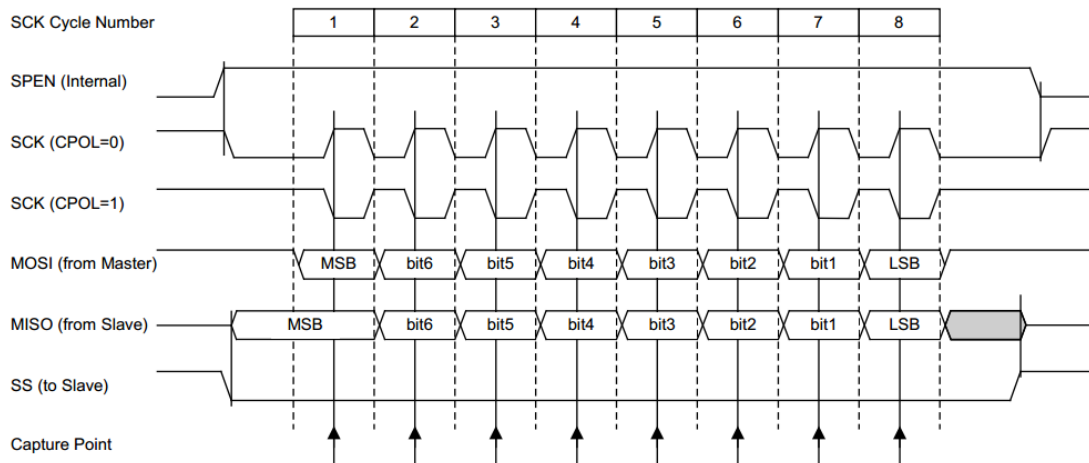


Figure 12-3 数据传送形式 (CPHA=0)

如果 CPHA = 0, 数据在 SCK 的第一沿就被捕获, 所以从设备必须在 SCK 的第一个沿之前就准备好数据, 因此, $\overline{SS}$ 引脚的下降沿从设备就开始数据。 $\overline{SS}$ 引脚在每次传送完一个字节后必须拉高, 在发送下一字节之前重新又被拉低, 故 CPHA = 0 时, SSIG 位无效, 即 $\overline{SS}$ 脚被强制使能。

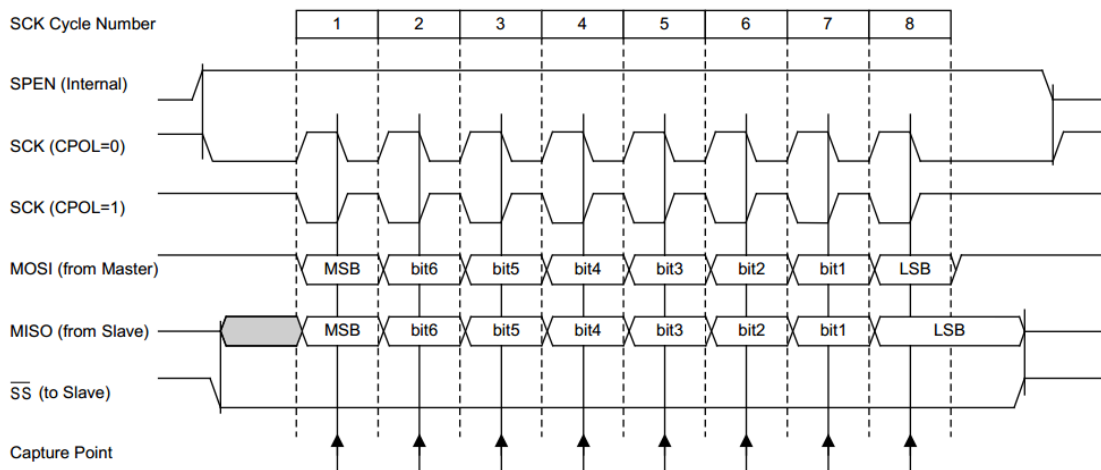


Figure 12-4 数据发送形式 (CPHA=1)

如果 CPHA = 1, 主设备在 SCK 的第一个沿将数据输出到 MOSI 线上, 从设备把 SCK 的第一个沿作为开始发送信号。用户必须在第一个 SCK 的前 2 个沿内完成对 SPDAT 完成写操作。传送过程中彼此模式不能改变, 否则数据发送接收将失败, 模式被改变的寄存器数据 (发送数据) 及状态 (接收为空) 不变。这种数据传送形式为单一主从设备间通信的首选形式。

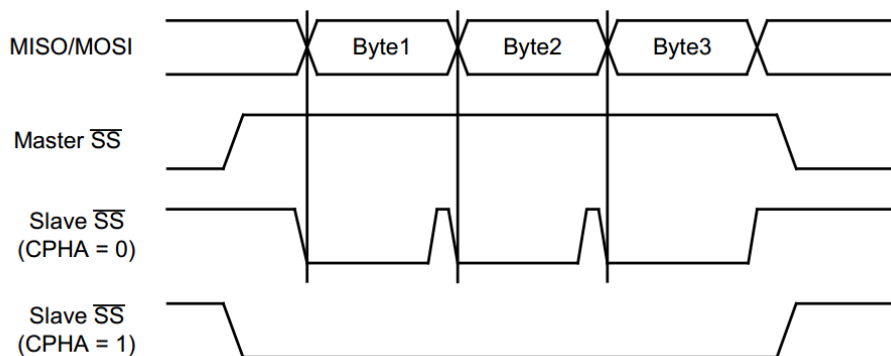


Figure 12-5 CPHA/SS 时序

12.7 SPI 出错检测

SPSTA 寄存器中的一些标志位表示 SPI 通信中的通信错误情况：

(1) 模式故障 (MODF)

SPI 主模式下的模式故障出错表明 $\overline{SS}$ 引脚上的电平状态与实际设备模式不一致，MODF 标志位将被置 1 (可产生中断)，以来表明 SPI 控制系统中存在多主设备的冲突情况，此时硬件将自动清除 SPEN 位，即先关闭 SPI 模块；同时硬件也将自动清除 MSTR 位。需要重启 SPI 模块时，MODF 必须先软件写 1 清 0，再置 SPEN 位。

(2) 写冲突 (WCOL)

在数据未发送或发送期间继续对 SPDAT 做写入操作会引起写冲突，WCOL 位会被置 1，但发送不会终止。需软件写 1 清 0。

(3) 接收溢出 (RXOV)

在接收第二数据完成前仍未清除之前接收数据产生的 SPIF 标志，将置接收溢出标志 RXOV，SPIF 被置 1 时，后面的数据将不会被传入接收寄存器，故接收的数据存入 SPDAT 前必须清除 SPIF，RXOV 位需软件写 1 清 0。

12.8 SPI 中断

两种 SPI 状态标志 SPIF & MODF 都能产生一个 CPU 中断请求。

串行数据传输完成标志 SPIF：完成一个字节数据发送/接收后由硬件置 1。

故障模式标志 MODF：该位被置 1 是指设备模式（主机）与 $\overline{SS}$ 引脚电平不一致，SSIG 位为 1 ($\overline{SS}$ 未被使能) 时，无 MODF 中断请求。

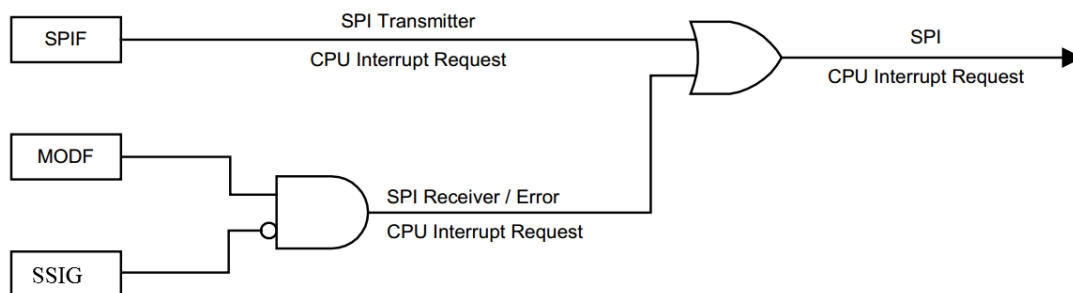


Figure 12-6 SPI 中断请求的产生

12.9 SPI 配置对照

SPEN	SSIG	$\overline{SS}$	MSTR	主或从模式	MISO	MOSI	SCK	备注
0	x	I/O	x	SPI功能禁止	I/O	I/O	I/O	SPI禁止
1	0	0	0	从机模式	输出	输入	输入	选择从机
1	0	1	0	从机模式未被选中	高阻	输入	输入	未被选中。MISO为高阻，以避免总线冲突
1→0	0	0	1→0	关闭SPI	输出	输入	输入	SS配置为输入，SSIG为0。如果SS被驱动为低电平。则被选择作为从机。此时MSTR将清零，并置模式错误标志MODF，可用于请求中断。
1	0	1	1	主（空闲）	输入	高阻	高阻	当主机空闲时MOSI和SCK为高阻态以避免总线冲突。用户必须将SCK上拉或下拉（根据CPOL的取值）以避免SCK出现悬浮状态。
				主（激活）		输出	输出	作为主机激活时，MOSI和SCK为推挽输出。
1	1	I/O	0	从	输出	输入	输入	CPHA不能为0
1	1	I/O	1	主	输入	输出	输出	-

12.10 SPI 相关寄存器

12.10.1 SPI 控制寄存器 SPCTL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR[1:0]	

位编号	位符号	说明
7	SSIG	$\overline{SS}$ 引脚使能位 0: $\overline{SS}$ 脚被使能用于确定器件为主机还是从机 1: MSTR 确定器件为主机还是从机, $\overline{SS}$ 脚作为普通 I/O 使用
6	SPEN	SPI 使能位 0: 禁止 SPI 模块, 相关管脚为普通 I/O(建议 I/O 时设为高阻) 1: 使能 SPI 模块, 相关管脚为 SPI 通信管脚
5	DORD	主/从机模式选择位 0: MSB先发送 1: LSB先发送
4	MSTR	传送方向选择位 0: 从机模式 1: 主机模式
3	CPOL	SPI 时钟极性选择位 0: SCK 空闲时为低电平 1: SCK 空闲时为高电平
2	CPHA	SPI 时钟相位选择位 0: 数据在 SPI 时钟的第一个边沿采样 1: 数据在 SPI 时钟的第二个边沿采样 注: SSIG = 0 & CPHA = 0 时, 数据在 $\overline{SS}$ 为低被驱动; CPHA = 1 时, 数据在 SCK 的前时钟沿驱动
1-0	SPR[1:0]	SPI 时钟速率选择控制位 00: 禁止配置此值 01: $F_{cpu}/16$ 10: $F_{cpu}/64$ 11: $F_{cpu}/128$

12.10.2 SPI 状态寄存器 SPSTAT

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0
位符号	-	WCOL	RXOV	-	-	-	-	-

位编号	位符号	说明
7	-	保留位
6	WCOL	SPI 写冲突标志位 0: 软件写 1 清 0 1: 传送过程中对 SPDAT 执行写操作硬件置 1（正在传送的数据不受影响） 注：当 OSC_CLK 的频率 F_{osc} 和 CPU 的频率 F_{cpu} 不一致时，此标志位会错误置位，但不影响正常的 SPI 通信。
5	RXOV	SPI 接收溢出标志位 0: 软件写 1 清 0 1: 发生接收溢出，硬件置 1 注：接收为双 BUFF，接收溢出发生在第二个数据传送完成前仍未清除之前接收数据产生的 SPIF 标志，故每次准备接收下一个数据前必须先清除 SPIF，否则 RXOV 将置 1，RXOV 置 1 不会影响 SPI 接收
4-0	-	保留位

12.10.3 SPI 数据寄存器 SPDAT

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	SPDAT[7:0]							

位编号	位符号	说明
7-0	SPDAT[7:0]	SPI 数据寄存器

13 IIC总线

13.1 IIC 特性

- 双线通信
- 支持主机模式
- 支持多主机通信时仲裁功能
- 支持地址可编程
- 支持标准速率（最多 100kbps）

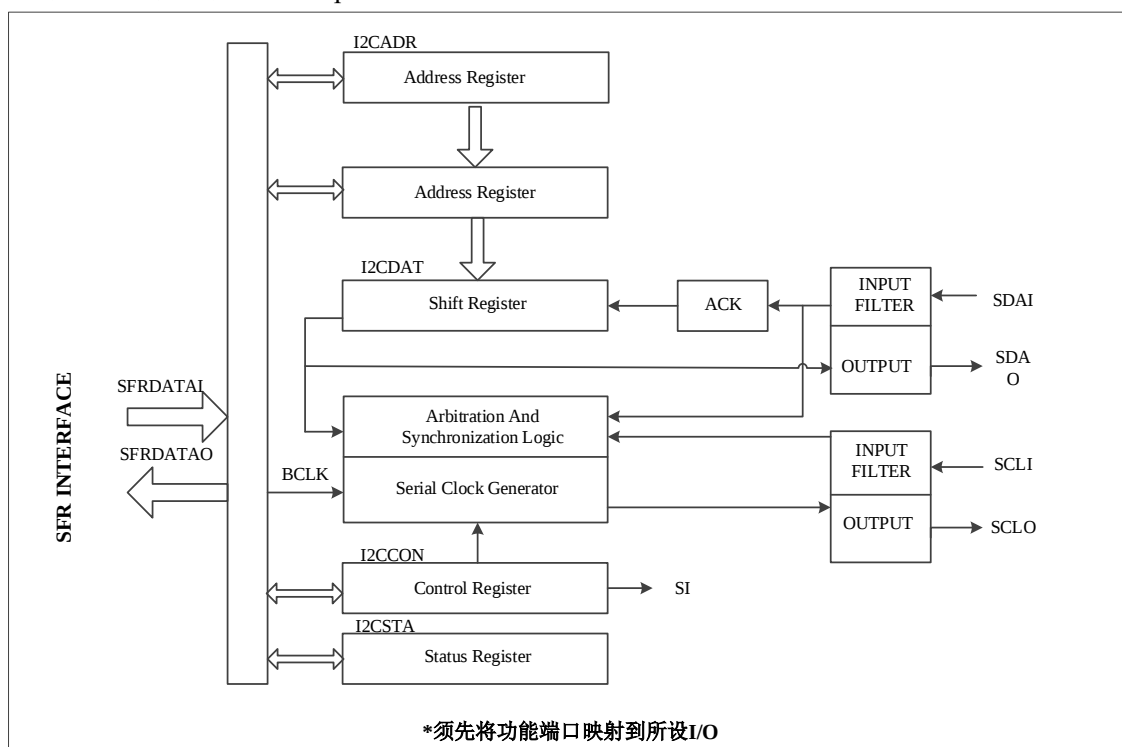


Figure 13-1 IIC 功能框图

13.2 IIC 总线工作原理

物理结构上，IIC 系统由一条串行数据线 SDA 和一条串行时钟线 SCL 组成。主机按一定的通信协议向从机寻址和进行信息传输，在数据传输时，由主机初始化一次数据传输，主机使数据在 SDA 线上传输的同时还通过 SCL 线传输时钟。信息传输的对象和方向以及信息传输的开始和终止均由主机决定。

每个器件都有一个唯一的地址，而且可以是单接收的器件或者可以接收也可以发送的器件。发送器或接收器可以在主模式或从模式下操作，这取决于芯片是否必须启动数据的传输还是仅仅被寻址。

下图是最常用、最典型的 IIC 总线连接方式。

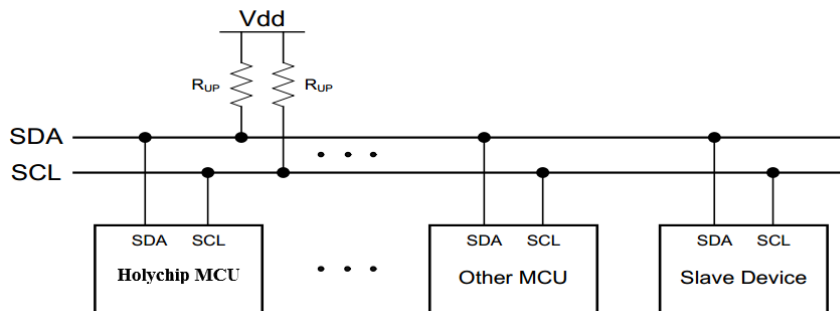


Figure 13-2 IIC 总线连接图

13.3 总线上数据的有效性

IIC 总线是以串行方式传输数据，从数据字节的最高位开始传送，每一个数据位在 SCL 上都有一个时钟脉冲相对应。在时钟线高电平期间数据线上必须保持稳定的逻辑电平状态，高电平为数据 1，低电平为数据 0。只有在时钟线为低电平时，才允许数据线上的电平状态变化，如 Figure13-3 所示。

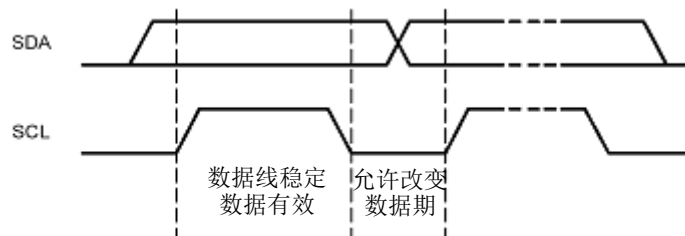


Figure 13-3 IIC 总线上数据的有效性

13.4 总线上的信号

IIC 总线在传送数据过程中共有四种类型信号，它们分别是：开始信号、停止信号、重新开始信号和应答信号。

开始信号（START）：如 Figure 13-4 所示，当 SCL 为高电平时，SDA 由高电平向低电平跳变，产生开始信号。当总线空闲的时候，例如，没有主动设备在使用总线（SDA 和 SCL 都处于高电平），主机通过发送开始（START）信号建立通信。

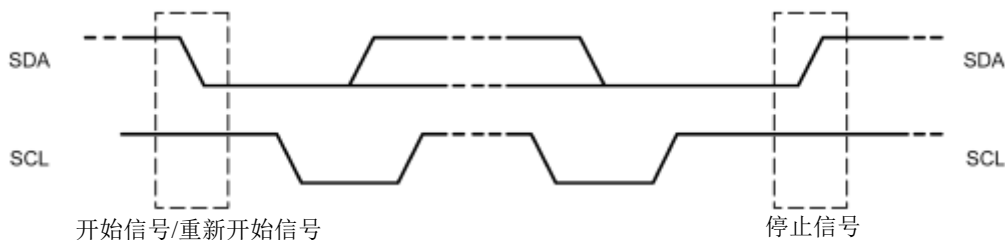


Figure 13-4 开始、重新开始、停止信号

停止信号（STOP）：如 Figure13-4 所示，当 SCL 为高电平时，SDA 由低电平向高电平跳变，产

生停止信号。主机通过发送停止信号，结束数据通信。

重新开始信号（Repeated START）：在 IIC 总线上，由主机发送一个开始信号启动一次通信后，在首次发送停止信号之前，主机通过发送重新开始信号，可以转换与当前从机的通信模式，或是切换到与另一个从机通信。如 Figure13-5 所示，当 SCL 为高电平时，SDA 由高电平向低电平跳变，产生重新开始信号，它的本质就是一个开始信号。

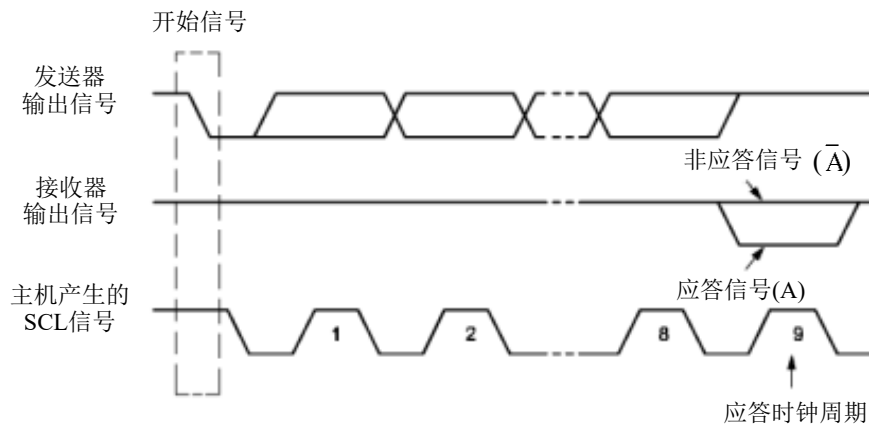


Figure 13-5 IIC 总线的应答信号

应答信号（A）：接收数据的 IIC 在接收到 8 位数据后，向发送数据的 IC 发出的特定的低电平脉冲。每一个数据字节后面都要跟一位应答信号，表示已收到数据。应答信号在第 9 个时钟周期出现，这时发送器必须在这一时钟位上释放数据线，由接收设备拉低 SDA 电平来产生应答信号，由接收设备保持 SDA 的高电平来产生非应答信号（ $\bar{A}$ ），如 Figure13-5 所示。所以，一个完整的字节数据传输需要 9 个时钟脉冲。如果从机作为接收方向主机发送非应答信号，这样，主机方就认为此次数据传输失败；如果是主机作为接收方，在从机发送器发送完一个字节数据后，发送了非应答信号，从机就认为数据传输结束，并释放 SDA 线。不论是以上哪种情况都会终止数据传输，这时，主机或是产生停止信号释放总线，或是产生重新开始信号，开始一次新的通信。开始信号、重新开始信号和停止信号都是由主控制器产生，应答信号由接收器产生，总线上带有 IIC 总线接口的器件很容易检测到这些信号。

13.5 总线上数据初始格式

一般情况下，一个标准的 IIC 通信由四部分组成：开始信号、从机地址传输、数据传输、停止信号。

由主机发送一个开始信号，启动一次 IIC 通信；在主机对从机寻址后，再在总线上传输数据。IIC 总线上传送的每一个字节均为 8 位，首先发送的数据位为最高位，每传送一个字节后都必须跟随一个应答位，每次通信的数据字节数是没有限制的；在全部数据传送结束后，由主机发送停止信号，结束通信。

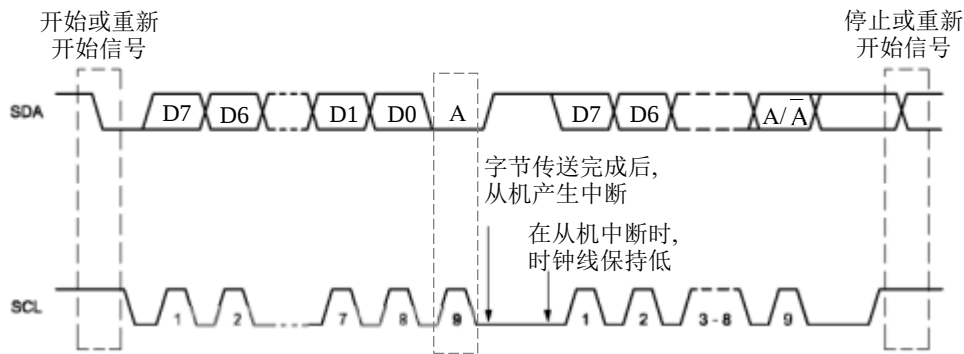


Figure 13-6 IIC 总线的数据传输格式

如 Figure 13-6 所示，时钟线为低电平时数据传输将停止进行。这种情况可以用于当接收器接收到一个字节数据后要要进行一些其它工作而无法立即接收下一个数据时，迫使总线进入等待状态，直到接收器准备好接收新数据时，接收器再释放时钟线使数据传输得以继续正常进行。例如，当接收器接收完主控制器的一个字节数据后，产生中断信号并进行中断处理，中断处理完毕才能接收下一个字节数据，这时接收器在中断处理时将钳住 SCL 为低电平，直到中断处理完毕才释放 SCL。

13.6 IIC 总线寻址约定

IIC 总线系统中挂接的所有外围器件，一般均拥有一个专用的 7 位从器件地址码。由于 7 位从器件地址码，其编码空间最多只有 128 个，后来在原有的 7 位地址码格式基础上，又发展了 10 位地址码格式。10 位地址格式仍然符合总线协议。

“广播呼叫”是个例外，它可以通过将第一个字节的数据全部赋值为 0 来寻址所有器件。广播呼叫用于当主机希望发送相同信息到几个从机时情况。当该地址在使用时，其他器件根据软件配置可能响应应答或忽略。如果器件响应广播呼叫，其操作就像从机接收器模式。

13.7 主机向从机读写 1 个字节数据的过程

如 Figure13-7 所示，主机要向从机写 1 个字节数据时，主机首先产生 START 信号，然后紧跟着发送一个从机地址，这个地址共有 7 位，紧接着的第 8 位是数据方向位 (R/W)，0 表示主机发送数据（写），1 表示主机接收数据（读），这时候主机等待从机的应答信号 (A)，当主机收到应答信号时，发送要访问的地址，继续等待从机的应答信号，当主机收到应答信号时，发送 1 个字节的数据，继续等待从机的应答信号，当主机收到应答信号时，产生停止信号，结束传送过程。

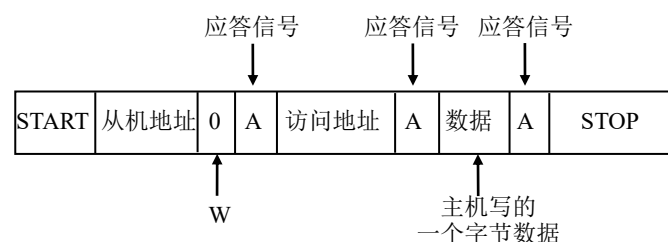


Figure 13-7 主机向从机写数据

如 Figure13-8 所示，主机要从从机读 1 个字节数据时，主机首先产生 START 信号，然后紧跟着发

送一个从机地址，注意此时该地址的第 8 位为 0，表明是向从机写命令，这时候主机等待从机的应答信号（A），当主机收到应答信号时，发送要访问的地址，继续等待从机的应答信号，当主机收到应答信号后，主机要改变通信模式（主机将由发送变为接收，从机将由接收变为发送）所以主机发送重新开始信号，然后紧跟着发送一个从机地址，注意此时该地址的第 8 位为 1，表明将主机设置成接收模式开始读取数据，这时候主机等待从机的应答信号，当主机收到应答信号时，就可以接收 1 个字节的数据，当接收完成后，主机发送非应答信号，表示不在接收数据，主机进而产生停止信号，结束传送过程。

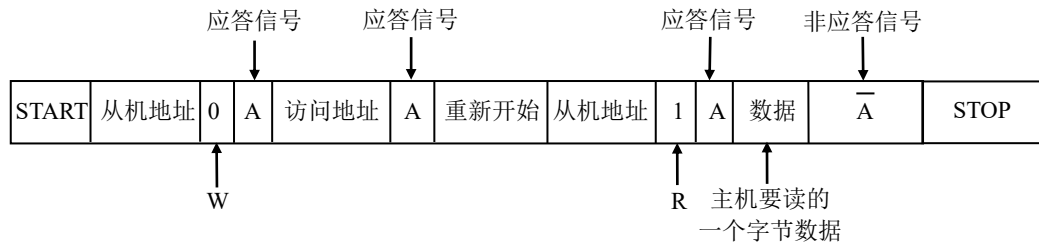


Figure 13-8 主机向从机读写数据 1 个字节数据

13.8 IIC 工作模式

13.8.1 主机发送模式

在主机发送模式下，向从机接收器发送几个数据字节。主机通过 CR[2:0]设置期望时钟速率并向 IICEN 位写 1 使能 IIC 总线，设置 STA 位为 1 进入主机发送模式，只要总线空闲，硬件将测试总线并产生起始信号，成功产生起始信号后，SI 标志位将置位且 IICSTA 的状态码为 08H，之后就是给 IICDAT 载入目标从机地址和数据方向位“写”（SLA+W），SLA+W 开始传输时 SI 位必须清零。

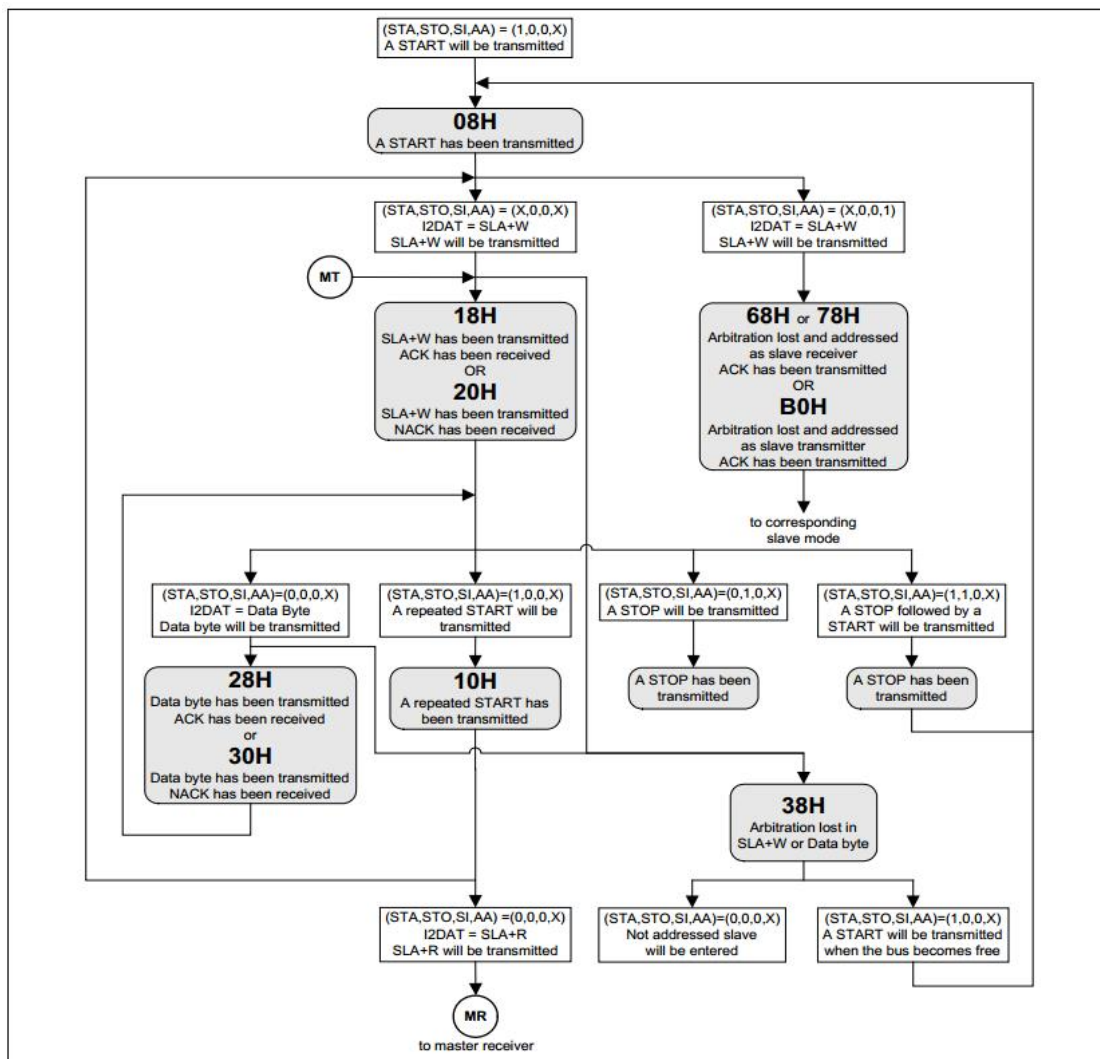


Figure 13-9 主机发送模式流程与状态

13.8.2 主机接收模式

在主机接收模式下，从从机发送器接收几个字节的数据。传输开始与主机发送模式相似，在起始信号之后，IICDAT 应该加载目标从机地址和数据方向位“读”（SLA+R），SLA+R 字节发送后，且返回应答位，重新置位 SI 标志且 IICSTA 读出为 40H，SI 标志应该被清零以便接收从机发送过来的数据，如果 AA 标志位置位，主机接收器将应答从机发送器，如果清零 AA，主机接收器将不会应答从机，并释放从机发送器为不被寻址的从机，然后主机产生停止信号或重复起始信号中止传输或开始另一次传输。

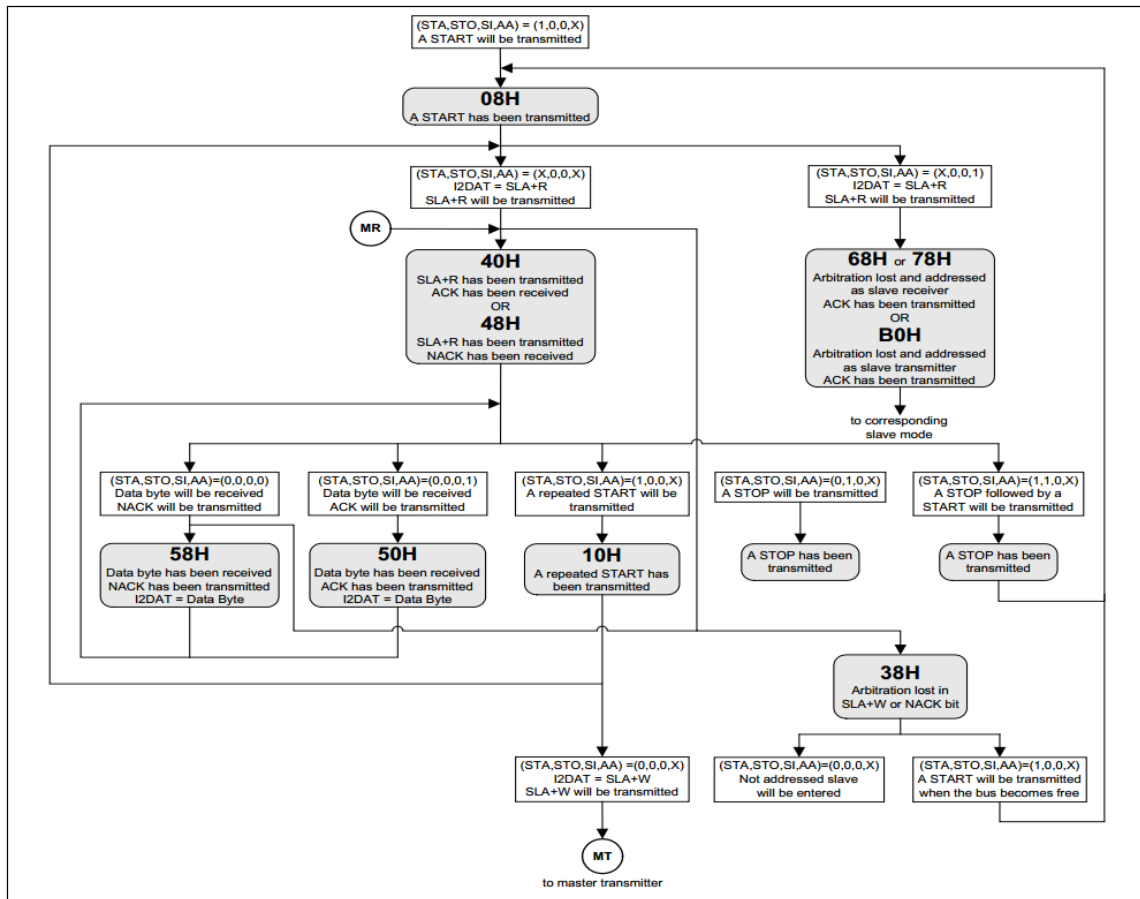


Figure 13-10 主机接收模式流程与状态

13.8.3 其他状态

有两个 IICSTA 状态码与 24 个定义状态不一致，即前面提到的 0F8H 和 00H 状态。

第一个状态码 0F8H 表示在每次传输期间没有得到相关信息，同时，SI 标志为 0 且没有 IIC 中断请求。

另一个标志码 00H 意味在传输过程中发生错误，总线错误是由 START 或停止信号暂时出现在一个非法的位置，如地址字节里第 2 位换到第 8 位，或数据字节包括应答位，当出现总线错误时，SI 标志立即置位，当在 IIC 总线上检测到总线错误，工作器件立即切换到不被寻址从机模式，释放 SDA 和 SCL 总，置位 SI 标志，将 00H 载入 IICSTA。要从总线错误恢复，STO 位必须设置为逻辑 1 且 SI 必须清零，然后，STO 由硬件清零且在没有停止信号就释放 IIC 总线。

特例：如果没有成功产生 START 或重复起始信号，IIC 总线被 SDA 的低电平阻挡，如一个从 CPU 时钟件没有位同步，可以通过在 SCL 总线上发送额外时钟脉冲解决这个问题。当 STA 位置位时，IIC 硬件发送额外时钟脉冲，但是由于 SDA 被拉低，不能产生起始信号，当 SDA 总线最终被释放，发送一个普通的 START 条件，进入状态 08H，继续进行串行传输。当 SDA 为低，如果发送重复起始信号，IIC 硬件也执行以上相同的动作。此情况下，在成功发送起始信号后，进入状态 08H，而不是进入 10H。
注：软件不能解决这类总线问题。

13.9 IIC 总线相关寄存器

13.9.1 IIC 控制寄存器 IICCON

位编号	7	6	5	4	3	2	1	0
R/W	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	0
位符号	CR2	IICEN	STA	STO	-	AA	CR1	CR0

位编号	位符号	说明
7	CR2	IIC 通信时钟选择位 2
6	IICEN	IIC 模块使能位 0: 禁止 IIC 模块 1: 启动 IIC 模块
5	STA	起始位 0: 不发送起始信号 1: 总线空闲时产生起始信号。忙时, 等待停止信号后产生一个起始信号。主机模式下, IIC 准备好发送或接收一个或多个字节时, 置 1 产生一个重复的起始信号
4	STO	停止位 0: 不发送停止信号 1: 主机模式时产生停止信号, 当检测到总线上出现停止信号。IIC 硬件清除 STO 标志。STO 标示的设置也用于将 IIC 设备从错误状态 (IICSTA 为 00H) 恢复, 此条件下, 没有停止信号发送 IIC 总线上。若 STA 和 STO 都置 1, 且在主机模式下设备为原始的, IIC 总线将产生停止信号并立即伴随着起始信号。如果设备为从机模式, 置 STO 恢复到非寻址从机, STO 将会硬件清 0。
3	-	保留位
2	AA	应答标志位 0: 回复 NACK (SDA 上为高电平) 1: 回复 ACK (SDA 上为低电平)
1	CR1	IIC 通信时钟选择位 1
0	CR0	IIC通信时钟选择位0

CR[2:0] IIC 通信时钟选择位:

CR2	CR1	CR0	F _{cpu}				分频系数
			1MHz	2MHz	4MHz	8MHz	
0	0	0	3.91KHz	7.81KHz	15.63KHz	31.25KHz	256
0	0	1	4.46KHz	8.93KHz	17.86KHz	35.71KHz	224
0	1	0	5.21KHz	10.42KHz	20.83KHz	41.67KHz	192
0	1	1	6.25KHz	12.5KHz	25KHz	50KHz	160
1	0	0	1.04KHz	2.08KHz	4.17KHz	8.33KHz	960
1	0	1	8.33KHz	16.67KHz	33.33KHz	66.67KHz	120
1	1	0	16.67KHz	33.33KHz	66.67KHz	133.33KHz	60
1	1	1	无效位				

13.9.2 IIC 状态寄存器 IICSTA

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R
复位值	1	1	1	1	1	0	0	0
位符号	IICSTA[7:3]					-		

位编号	位符号	说明
7-3	IICSTA[7:3]	IIC 状态码，共有 26 个可能的状态码，状态码中除 0F8H 外都可置 SI 标志
2-0	-	保留位

13.9.3 IIC 数据寄存器 IICDAT

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	IICDAT[7:0]							

位编号	位符号	说明
7:0	IICDAT[7:0]	IIC 数据 IICDAT 包含一个字节的将被发送或刚接收到的 IIC 数据。只要 SI 为逻辑 1，IICDAT 中的数据保持不变，在 IIC 发送接收过程中，读或写 IICDAT 的结果都是不确定的。 当 IICDAT 的数据被移出，总线上的数据同步被移入以更新 IICDAT。IICDAT 常显示当前 IIC 总线上的最后字节。因此失去仲裁，在传输之后的 IICDAT 原始值被改变。

14 模数转换ADC

14.1 ADC 特性

HC18M002/003具有一个12位转换分辨率的模数转换器，共有8个外部模拟输入通道，1个内部电池检测通道。

ADC的等效电路：

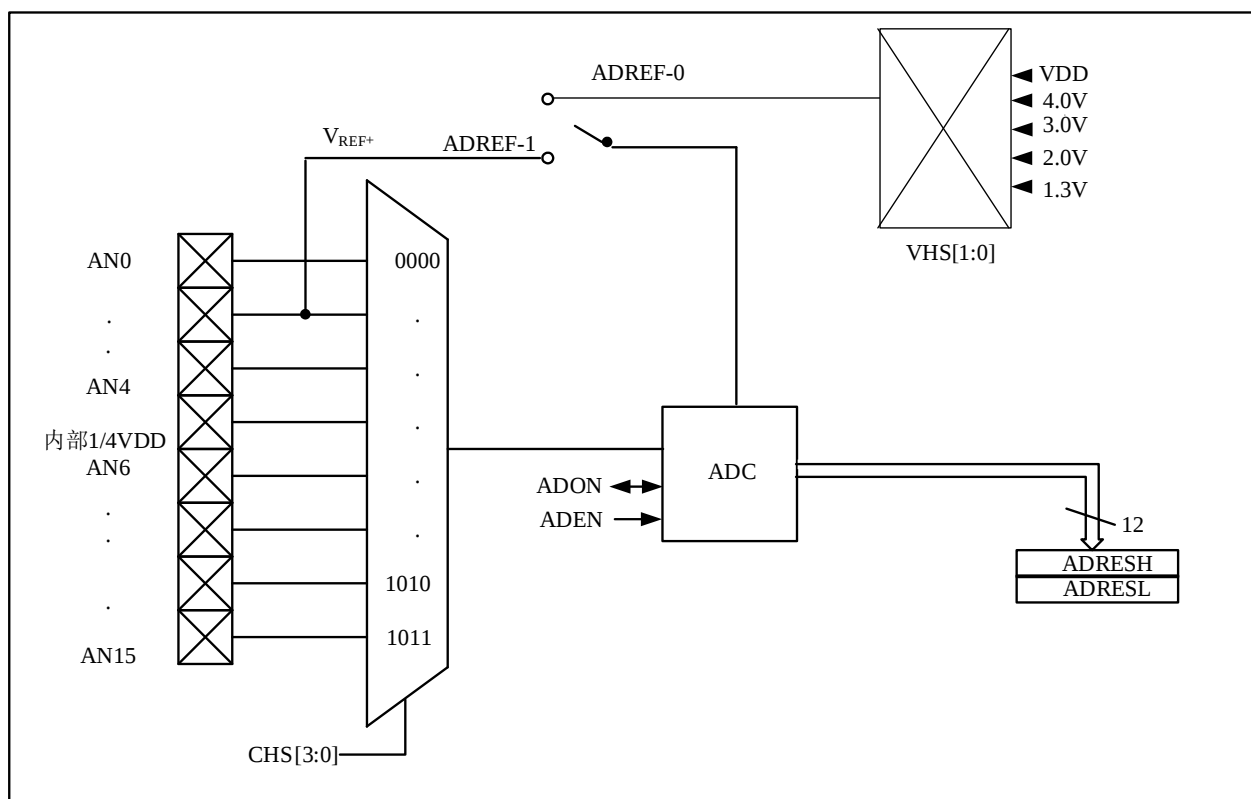


Figure 14-1 ADC 等效电路

14.2 ADC 相关寄存器

14.2.1 ADC 控制寄存器 ADCON0

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	CHS[3:0]				ADON	ADEN

位编号	位符号	说明
7-6	-	保留位
5-2	CHS[3:0]	ADC 通道选择寄存器 0000: AN0 0001: 保留位 0010: AN2 0011: AN3 0100: AN4 0101: 内部 1/4VDD 0110: AN6 0111: AN7 1000: AN8 1001: AN9 1010: AN10 1011: AN11 1100: 保留位 1101: 保留位 1110: AN14 1111: AN15
1	ADON	ADC 转换启动控制位 0: 转换结束后, 硬件自动清 0, 在转换过程中, 软件清 0 将终止转换 1: 启动转换
0	ADEN	ADC 模块使能位 0: 关闭 ADC 模块 1: 使能ADC模块 注: 在使能ADC模块后, 建议延时20us再启动ADC转换, 以保证ADC模块稳定在SLEEP模式下, 建议关闭ADC模块, 以降低待机功耗

14.2.2 ADC 控制寄存器 ADCON1

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	0
位符号	ADFM	ADCS[2:0]			INREFS [2:0]			VREFS

位编号	位符号	说明
7	ADFM	ADC 数据存放格式选择 0: ADC 转换结果为 12 位数据, 数据格式: ADRESH[3:0]-ADRESL[7:0] 1: ADC 转换结果为 10 位数据, 数据格式: ADRESH[1:0]-ADRESL[7:0]
6-4	ADCS[2:0]	ADC 时钟分频选择 000: F_{cpu} 001: $F_{cpu}/2$ 010: $F_{cpu}/4$ 011: $F_{cpu}/8$ 100: $F_{cpu}/16$ 101: $F_{cpu}/32$ 110: $F_{cpu}/64$ 111: FRC
1	INREFS[2:0]	ADC 内部参考电压选择 000: VDD 001: 4.0V 010: 3.0V 011: 2.0V 1xx: 1.3V 注: 为降低系统功耗, 建议在 SLEEP 模式下请勿选择 VDD 作为参考电压
0	VREFS	ADC 内参外参选择 0: 选择内部参考电压 1: 选择外部参考电压

14.2.3 ADC 时钟选择寄存器 ADCLK

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	-	ADCLK[2:0]		

位编号	位符号	说明
7-3	-	保留位
2-0	ADCLK[2:0]	ADC 时钟选择位 000: ADC 转换频率在 4MHz 及以上 001: ADC 转化频率为 1MHz 010: ADC 转化频率为 2MHz 011/1xx: ADC 转化频率为 1MHz 以下

14.2.4 ADC 转换结果寄存器 ADRESL、ADRESH

ADRESL

位编号	7	6	5	4	3	2	1	0
R/W	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
位符号	ADCRL[7:0]							

ADRESH

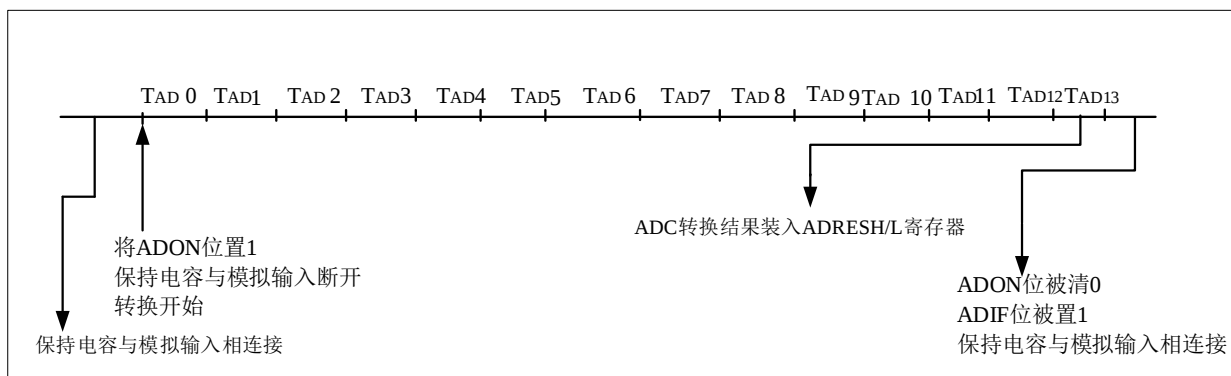
位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	ADCRH[7:0]							

位编号	位符号	说明
7-0	ADRESH[7:0]	ADFM = 0 时 ADRESH[7:0]为 ADC 转换的高 4 位, ADRESL[7:0]为 ADC 转换的低 8 位
7-0	ADRESL[7:0]	ADFM = 1 时 ADRESH[7:0]为 ADC 转换的高 2 位, ADRESL[7:0]为 ADC 转换的低 8 位

14.3 AD 转换时间

ADC转换一位数据所需的时间定义为TAD，转换一次完整的10位数据需要14个TAD。为确保ADC正确转换，必须满足适当的TAD时间。

模数转换TAD 周期



ADC转换时间(TAD)与工作频率关系表

ADC 转换时间 (TAD)		系统频率 (F_{cpu}) :4MHz
ADC 时钟源	ADCS[2:0]	典型值
F_{cpu}	000	4us
$F_{cpu}/2$	001	8us
$F_{cpu}/4$	010	16us
$F_{cpu}/8$	011	32us
$F_{cpu}/16$	100	64us
$F_{cpu}/32$	101	128us
$F_{cpu}/64$	110	256us
FRC	111	500KHz

注意:

1. 为了加快转换速度，建议选用较快时钟源（ADC转换时钟不能超过4MHz）。
2. 当系统频率高于1 MHz 时，仅当在休眠和绿色模式下进行转换时才推荐使用FRC时钟源。

选择FRC时钟源后，ADC需等待一个指令周期后才能启动转换操作，这使得可以执行SLEEP 指令，以降低转换期间的系统噪声。如果使能了ADC中断，转换完成时将唤醒SLEEP。如果禁止了ADC中断，尽管ADEN位仍保持为1，转换完成后ADC模块将关闭。ADC时钟源不是FRC时，尽管ADEN位仍保持为1，SLEEP指令会导致当前转换中止，ADC模块关闭。

除了选择FRC时钟源，改变系统时钟频率均会改变ADC的时钟频率，从而影响ADC转换时间。

ADEN位置1将使能ADC模块，ADON位置1将启动一次ADC转换。ADC转换完成，ADON位硬件清零，ADIF中断标志位置1，ADRESH/ADRESL寄存器值被更新。如果必须在转换完成前终止转换，可用软件将ADON位清零，ADRESH/ ADRESL寄存器将保持前次ADC转换的结果。

15 电容触摸按键CTK

HC18M002/003 内置一个支持最多 8 个通道的触摸电路，配合官网上提供的触摸库可实现弹簧按键、滑轮、滑条、空气触摸等应用。

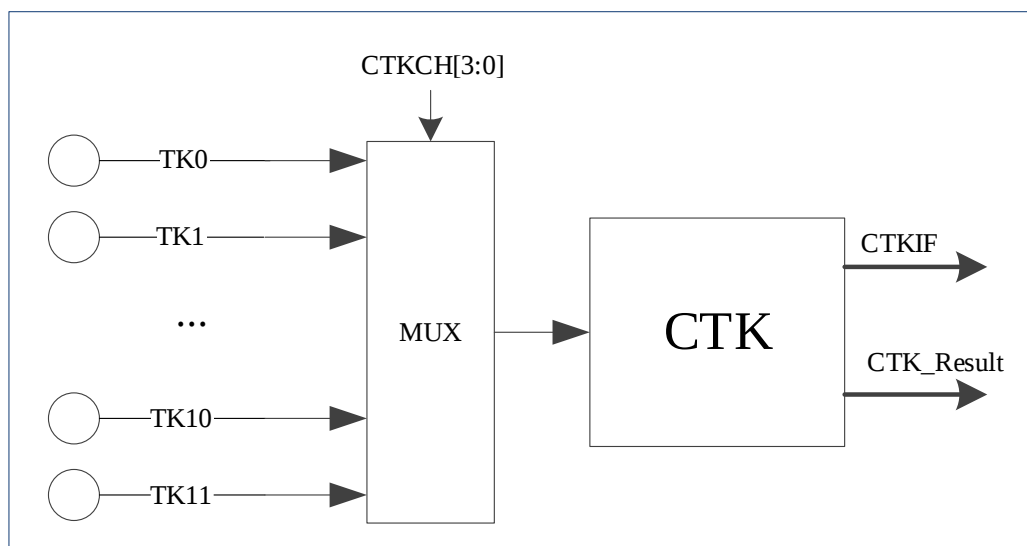


Figure 15-1 CTK 功能框图

16 代码选项OPTION

1、外部复位使能

0: PORTC4 为外部复位引脚（默认），该口作为外部复位引脚时，无法作为普通 I/O 使用

1: PORTC4 为普通 IO 引脚

2、CPU 时钟模式选择

0: 2T

1: 4T

3、复位时间选择

0: 4.5ms

1: 18ms

4、BOR 复位电压选择

1: 2.0V

2: 2.4V

3: 2.6V

4: 3.0V

5: 3.6V

6: 3.9V

7: 4.2V

5、WDT 复位功能使能

0: 关闭 WDT 时钟

1: 使能 WDT 时钟

17 指令表

助记符	操作数	说明	周期数	受影响状态位
ADDWF	f,d	W 和 f 相加	1	C, DC, Z
ADDLW	k	将立即数和 W 相加	1	C, DC, Z
SUBWF	f,d	f 减去 W	1	C, DC, Z
SUBLW	k	立即数减去 W	1	C, DC, Z
DAW	-	W 寄存器值进行 BCD 调整	1	C, DC
ANDWF	f,d	W 和 f 作逻辑与运算	1	Z
ANDLW	k	立即数和 W 作逻辑与运算	1	Z
IORWF	f,d	W 和 f 作逻辑或运算	1	Z
IORLW	k	立即数和 W 作逻辑或运算	1	Z
XORWF	f,d	W 和 f 作逻辑异或运算	1	Z
XORLW	k	立即数和 W 作逻辑异或运算	1	Z
COMF	f,d	f 取反	1	Z
CLRW	-	将 W 清零	1	Z
CLRF	f	将 f 清零	1	Z
INCF	f,d	f 加 1	1	Z
INCFSZ	f,d	f 加 1, 为 0 则跳过	1(2)	-
DECF	f,d	f 减 1	1	Z
DECFSZ	f,d	f 减 1, 为 0 则跳过	1(2)	-
BCF	f,d	将 f 中的 d 位清 0	1	-
BSF	f,d	将 f 中的 d 位置 1	1	-
BTFSC	f,d	检测 f 中的 d 位, 为 0 则跳过	1(2)	-
BTFSS	f,d	检测 f 中的 d 位, 为 1 则跳过	1(2)	-
MOVWF	f	将 W 的内容传送到 f	1	-
MOVF	f,d	将 f 的内容送到目标寄存器	1	Z
MOVLW	k	将立即数 k 传送到 W	1	-
RLF	f,d	对 f 执行带进位的循环左移	1	C
RRF	f,d	对 f 执行带进位的循环右移	1	C
SWAPF	f,d	将 f 的两个半字节进行交换	1	-
CALL	k	调用子程序	2	-
GOTO	k	无条件跳转	2	-
RETFIE	-	从中断返回	2	GIE
RETURN	-	从子程序返回	2	-
RETLW	k	返回时将立即数传送到 W	2	-
CLRWDI	-	清零看门狗定时器	1	TO, PD
SLEEP	-	进入待机模式	1	TO, PD
NOP	-	空操作	1	-

18 电气特性

除非另外说明，以下数据测试条件均为：VDD=5.0V，GND=0V，25°C。

18.1 极限参数

参数	符号	最小值	典型值	最大值	单位
直流供电电压	VDD	-0.3	-	+6.0	V
输入/输出电压	V _I /V _O	GND-0.3	-	VDD+0.3	V
工作环境温度	T _{OTG}	-40	-	+85	°C
存储温度	T _{STG}	-55	-	+125	°C

注：

- （1）流过 VDD 的最大电流值在 5.0V，25°C 下须小于 100mA。
- （2）流过 GND 的最大电流值在 5.0V，25°C 下须小于 150mA。

18.2 DC 特性

参数	符号	条件（VDD=5V）	最小值	典型值	最大值	单位
工作电压	VDD	F _{CPU} =0~8MHz，ADC 模块关闭	4.5	5.0	5.5	V
		F _{CPU} =0~4MHz，ADC 模块关闭	3.0	5.0	5.5	
		F _{CPU} =0~2MHz，ADC 模块关闭	2.0	5.0	5.5	
工作电流	I _{OP1}	F _{OSC} =32MHz，F _{CPU} =8MHz，无负载，高频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	5	-	mA
		F _{OSC} =16MHz，F _{CPU} =8MHz，无负载，高频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	4.6	-	
		F _{OSC} =16MHz，F _{CPU} =4MHz，无负载，高频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	3	-	
		F _{OSC} =8MHz，F _{CPU} =4MHz，无负载，高频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	2.8	-	
		F _{OSC} =8MHz，F _{CPU} =2MHz，无负载，高频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	2	-	
		F _{OSC} =4MHz，F _{CPU} =2MHz，无负载，高频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	1.8	-	
		F _{OSC} =4MHz，F _{CPU} =1MHz，无负载，高频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	1.5	-	

		令，其它模块关闭				
	I _{OP2}	F _{cpu} =16KHz，低频模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭	-	20	-	μA
		F _{cpu} =16KHz，绿色模式，无浮动输入管脚，执行 NOP 指令，其它模块关闭		15		
	I _{PD1}	休眠模式，无负载，无浮动输入管脚，所有模块关闭，WDT 使能	-	7	-	
	I _{PD3}	休眠模式，无负载，无浮动输入管脚，所有模块关闭	-	1	-	
BOR 电流	I _{BOR}	VDD = 5V	4.0	6.0	8.0	μA
输入低电压 1	V _{IL1}	I/O 端口输入	GND	-	0.3*VDD	V
输入高电压 1	V _{IH1}	I/O 端口输入	0.7*VDD	-	VDD	V
输入漏电流	I _{ILC}	I/O端口输入模式，V _{IN} = VDD 或GND	-1	0	1	μA
输出漏电流	I _{OLC}	I/O端口输出模式，V _{OUT} = VDD 或GND	-1	0	1	μA
灌电流	I _{OL1}	0.1VDD，VDD=5V	24.5	35	45.5	mA
拉电流	I _{OH1}	0.9VDD，VDD=5V	6	10	14	
上拉电阻	R _{PU}	VIN=VDD	25	50	75	kΩ
下拉电阻	R _{PD}	VIN=VDD	25	50	75	kΩ
RAM 保持电压	V _{RAM}	-		0.7	1	V
Vbg 电压精度	V _{BG}	VDD=5.0V/3.0V	1.19	1.2	1.21	V
CTK_VDD	V _{CTKVDD}	VDD=5.0V	2.5	2.87	3.1	V
CTK_Vref	V _{CTKVREF}	VDD=5.0V	0.55	0.6	0.65	V

18.3 AC 特性

参数	符号	条件	最小值	典型值	最大值	单位
内部 RC32M 启动时间	Tset1	常温, VDD=5V	-	-	2	ms
内部 RC44K 启动时间	Tset2	常温, VDD=5V	-	-	150	μs
频率精度	FIRC1	VDD=2V~5.5V, 25°C	32(1-1%)	32	32(1+1%)	MHz
	FIRC2	VDD=5.0V, -40°C ~+85°C	32(1-2%)	32	32(1+2%)	MHz
	FWRC	-	16	32	48	KHz

18.4 ADC 特性

参数	符号	条件	最小值	典型值	最大值	单位
供电电压	VAD	-	2	5.0	5.5	V
精度	NR	$GND \leq VAIN \leq V_{ref}$	-	10	12	bit
ADC 输入电压	VAIN	-	GND	-	Vref	V
ADC 输入电阻	RAIN	VAIN=5V	2	-	-	MΩ
模拟电压源推荐阻抗	ZAIN	-	-	-	10	kΩ
ADC 转换电流	IAD	ADC 模块打开, VDD=5.0V	-	0.6	1	mA
ADC 输入电流	IADIN	VDD=5.0V	-	-	10	μA
微分非线性误差	DLE	VDD=5.0V, -40°C~+85°C	-2	-	+4	LSB
积分非线性误差 (1MHz 转换频率)	ILE1	VDD=5.0V, Vref =1.3V, 25°C	-5	-	+2	LSB
		VDD=5.0V, Vref =2V, 25°C	-5	-	+2	
		VDD=5.0V, Vref =3V, 25°C	-4	-	+2	
		VDD=5.0V, Vref =4V, 25°C	-3	-	+2	
		VDD=5.0V, Vref =VDD, 25°C	-2	-	+2	
		VDD=5.0V, Vref =外参, 25°C	-2	-	+2	
积分非线性误差 (1MHz 转换频率)	ILE2	VDD=5.0V, Vref =1.3V, -40°C	-8	-	+4	LSB
		VDD=5.0V, Vref =2V, -40°C	-7	-	+3	
		VDD=5.0V, Vref =3V, -40°C	-7	-	+2	
		VDD=5.0V, Vref =4V, -40°C	-4	-	+3	
		VDD=5.0V, Vref =VDD, -40°C	-2	-	+2	
积分非线性误差 (1MHz 转换频率)	ILE3	VDD=5.0V, Vref =1.3V, +85°C	-12	-	+4	LSB
		VDD=5.0V, Vref =2V, +85°C	-12	-	+2	
		VDD=5.0V, Vref =3V, +85°C	-9	-	+2	
		VDD=5.0V, Vref =4V, +85°C	-7	-	+2	
		VDD=5.0V, Vref =VDD, +85°C	-2	-	+3	
满刻度误差	EF	VDD=5.0V	-5	-	+5	LSB
偏移量误差	EZ	VDD=5.0V	-3	-	+5	LSB
总绝对误差	EAD	VDD=5.0V	-5	-	+5	LSB

18.5 MTP 内存特性

参数	符号	条件	最小值	典型值	最大值	单位
读写测试	MN _{ENDUR}	-	100	-	-	Cycle
数据保存时间	MT _{RET}	T=25°C	-	10	-	year
字节写入电压	MV _{WRITE}	-	8.8	9.0	9.2	V
字节读取电压	MV _{READ}	-	2.0	-	5.5	V
字节写入时间	MT _{WRITE}	1 个字节@4.5V~5.5V		0.5		ms
字节读取时间	MT _{READ1}	1 个字节@4.5V~5.5V	80	-		ns
	MT _{READ2}	1 个字节@3.0V~4.5V	124	-		ns
	MT _{READ3}	1 个字节@2.0V~3.0V	250	-		ns
读取耗电流	MI _{DD1}	F _{cpu} =8MHz@5V	-	4	-	mA
写入耗电流	MI _{DD2}	-	-	1.5	-	mA

18.6 EEPROM 内存特性

参数	符号	条件	最小值	典型值	最大值	单位
读写测试	EN _{ENDUR}	-	10000	-	-	Cycle
数据保存时间	ET _{RET}	T=25°C	10	-	-	year
字节写入电压	EV _{WRITE}	-	3.0	-	5.5	V
字节读取电压	EV _{READ}	-	2.0	-	5.5	V
字节写入时间	ET _{PROG}	1 个字节	-	2	-	ms
字节读取时间	FT _{READ1}	1 个字节@3.0V~5.5V		4		us
	FT _{READ2}	1 个字节@2.0V~3.0V		7		us
读取耗电流	EI _{DD1}	VDD=5.0V	-	1	-	mA
写入耗电流	EI _{DD2}	-	-	10	-	mA

18.7 BOR 检测电压特性

参数	符号	条件	最小值	典型值	最大值	单位
BOR 设定电压 2	V _{BOR2}	BOR 使能, VDD=2V~5.5V	1.8	2.0	2.2	V
BOR 设定电压 3	V _{BOR3}		2.2	2.4	2.6	V
BOR 设定电压 4	V _{BOR4}		2.4	2.6	2.8	V
BOR 设定电压 5	V _{BOR5}		2.8	3.0	3.2	V
BOR 设定电压 6	V _{BOR6}		3.4	3.6	3.8	V
BOR 设定电压 7	V _{BOR7}		3.7	3.9	4.1	V
BOR 设定电压 8	V _{BOR8}		4.0	4.2	4.4	V

18.8 其他电气特性

- 1、ESD (HBM) : CLASS 3A ($\geq 4000V$)
- 2、Latch_up: CLASS I ($\geq 500mA$)
- 3、现有 EFT 测试环境，过 4000V 标准。（VDD 端口加 104 电容，程序跑马灯程序）

19 封装尺寸

19.1 SOP16

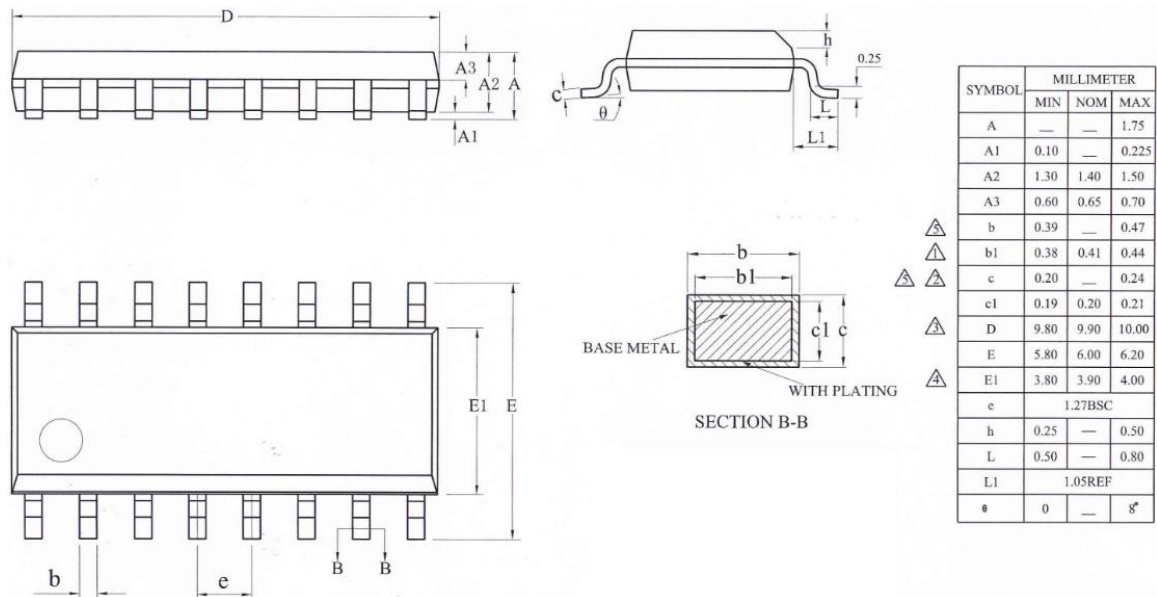


Figure 19-1 SOP16 封装尺寸

19.2 SOP20

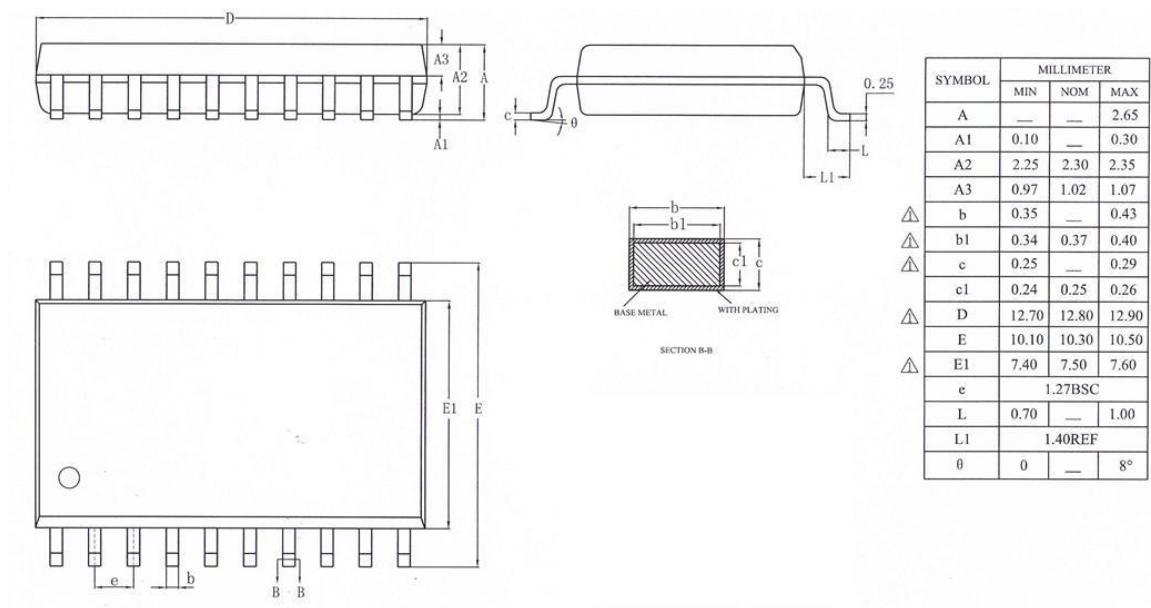


Figure 19-2 SOP20 封装尺寸

19.3 TSSOP20

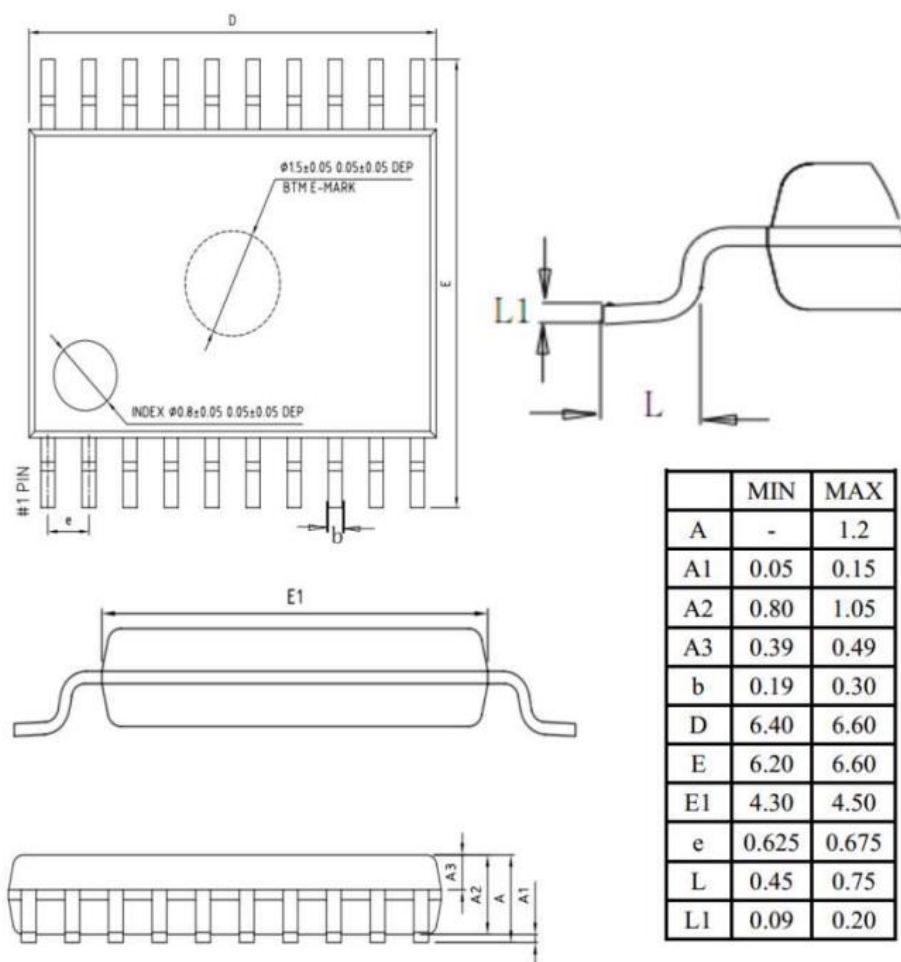


Figure 19-3 TSSOP20 封装尺寸

20 版本记录

版本	日期	描述
Ver1.00	2023-02-08	第一版
Ver1.01	2023-03-20	1、 增加 SOP16/SOP20 封装 2、 增加部分注意事项 3、 修改端口映射部分寄存器 4、 修改部分错误
Ver1.02	2023-03-24	1、 SOP16 以及 SOP20 封装增加 CTK 描述 2、 修改部分错误
Ver1.03	2023-05-23	1、 修改部分注意事项 2、 增加 SOP14 封装 3、 修改部分错误
Ver1.04	2023-05-23	1、 修改部分注意事项 2、 修改部分寄存器
Ver1.05	2023-07-03	1、 修改部分错误 2、 增加部分注意事项
Ver1.06	2023-07-24	1、 修改部分错误 2、 增加部分注意事项 3、 修改 OPTION 部分说明 4、 修改部分电气特性参数
Ver1.07	2023-09-07	1、 修改 IIC 相关描述
Ver1.08	2023-11-29	1、 增加注意事项 2、 修改 SPI 相关配置

HOLYCHIP 公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。HOLYCHIP 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，HOLYCHIP 的产品不是专门设计来应用于外科植入、生命维持和任何 HOLYCHIP 产品产生的故障会对个体造成伤害甚至死亡的领域。如果将 HOLYCHIP 的产品用于上述领域，即使这些是由 HOLYCHIP 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用，并且用户保证 HOLYCHIP 及其雇员、子公司、分支机构和销售商与上述事宜无关。

芯圣电子

2023 年 11 月