MDT10F1822/1823

数据手册（版本 V1.0）

8 位 MTP
CMOS 单片机

1.0 器件概述

♦ **存储器：**
MTP 空间：2K*14 位，可经受 1000 次写操作。
EEPROM：256 字节，可经受 10000 次写操作。
SRAM 空间：128 字节。

♦ **单片机特性**

• 上电复位（Power-on Reset，POR）、上电延时定时器（Power-up Timer，PWRT）和振荡器起振定时器（Oscillator Start-up Timer ，OST）
• 可编程欠压复位（Programmable Brown-outReset ，BOR）
• 扩展型看门狗定时器（Watchdog Timer，WDT）
•工作电压范围：
- 2.2V-5.5V
•可编程代码保护

♦ **I/O 引脚配置**
具有独立方向控制的 11 个 I/O 引脚：PA 口 5 个、PC 口 6 个。
一个只能作输入的 PA3 口。
高灌/拉电流可直接驱动 LED。
PA 端口引脚电平变化中断。
PA、PC 端口独立的可编程弱上拉。

♦ **模数转换器**
- 10位分辨率，最多8 路通道
- 可在休眠模式下进行转换
• 模拟比较器模块：
- 最多2 个轨到轨模拟比较器
- 功耗模式控制
- 可通过软件控制滞后
• 参考电压模块：
- 具有1.024V、2.048V和4.096V输出电压的固定参考电压（Fixed Voltage Reference，FVR）
- 带有正负参考电压选择的5 位轨到轨电阻式DAC

♦ **双时钟系统**
- 出厂时精度已校准到± 1%，典型值
- 通过软件可选择的频率范围为31kHz至16MHz
• 31kHz低功耗内部振荡器
• 4种晶振模式，频率高达16MHz
• 3种外部时钟模式，频率高达16MHz
• 故障保护时钟监视器（Fail-Safe Clock Monitor，FSCM）：
- 当外设时钟停止时可使器件安全关闭
• 双速振荡器启动
• 参考时钟模块：
- 可编程时钟输出频率和占空比

♦ **高性能的 RISCCPU**
仅需学习49 条指令：
- 除了跳转指令之外，所有指令都是单周期指令
• 工作速度：
- DC —— 16MHz振荡器/时钟输入
- DC ——250ns指令周期
• 带有自动现场保护的中断功能
• 带有可选上溢/下溢复位的16级深硬件堆栈
• 直接寻址、间接寻址和相对寻址模式：
- 2个完整的16位文件选择寄存器（File SelectRegister ，MSR）
- MSR可读程序和数据存储器

♦ **特殊特性**
- 高精度内部振荡器，出厂时精度校准为+/-1%。
- 可用软件选择的频率范围为 31.25kHz 到 16MHz。
- 软件可选的 31kHz 内部振荡器。
- 节能的休眠模式。
- 宽工作电压范围（2.2V 到 5.5V）。
- 工业级温度范围。
- 上电复位（Powe-onReset，POR）。
- 上电延时定时器（Power-upTimer，PWRT）和振荡器起振定时器（OscillatorStart-upTimer，OST）。
- 带软件控制选择的PED低电压侦测选择(侦测电压有1.9V、2.5V 可选)。
- 带片上振荡器（振荡器频率可由软件选择，当预分频比最大时其标称值为 268 秒）并且可软件使能的增强型低电流看门狗定时器（WatchdogTimer，WDT）。
- 带上拉的外部复位，可复用为输入引脚。
- 可编程代码保护。

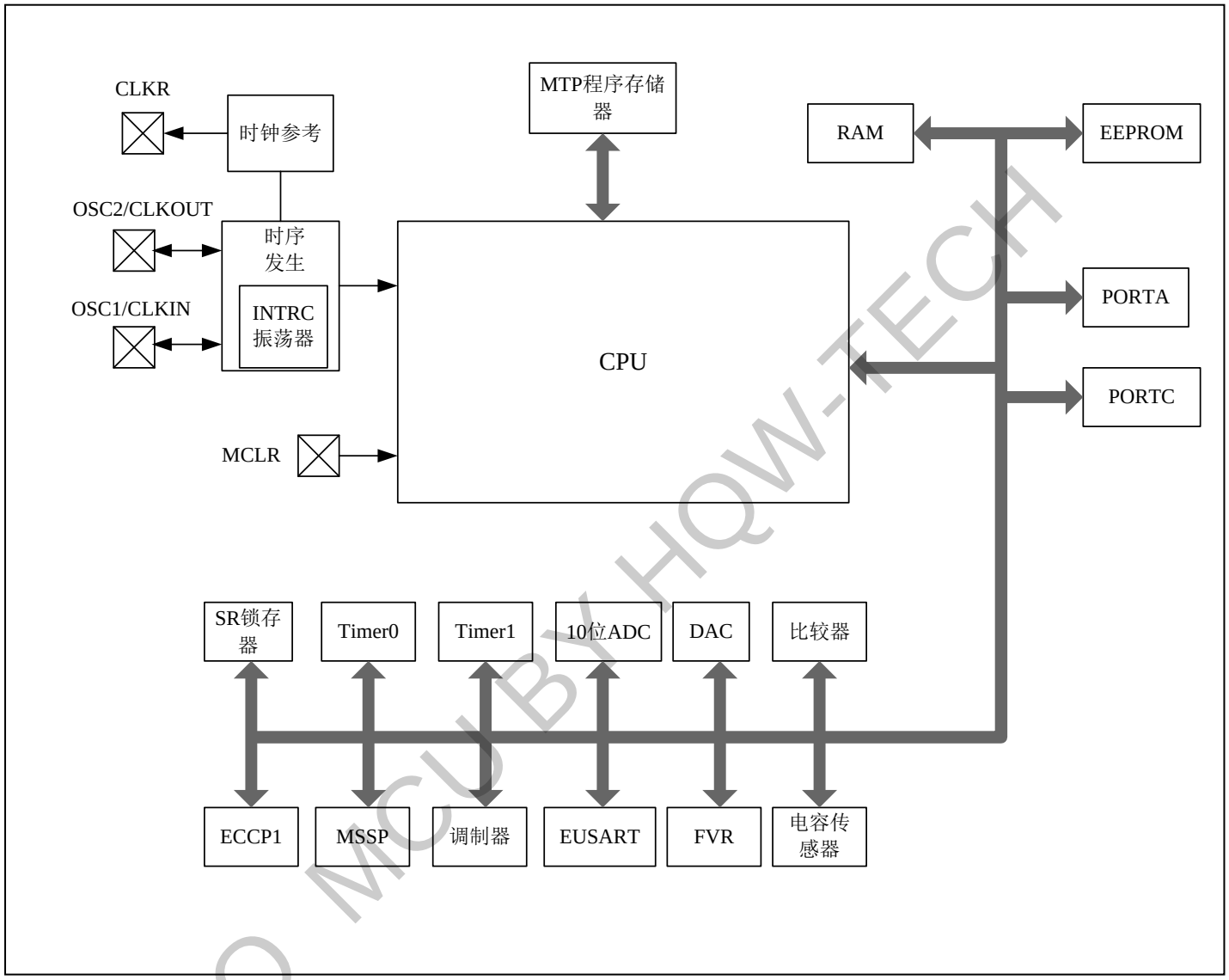
♦ **外设特性**
- 高灌/拉电流为25mA/25mA
- 可编程电平变化中断引脚
• Timer0：带有8位预分频器的8位定时器/计数器
• 增强型Timer1：
- 带有预分频器的16位定时器/计数器
- 外部门控输入模式
- 专用的低功耗32 kHz 振荡器驱动器

• Timer2：带有8位周期寄存器、预分频器和后分频器的8位定时器/计数器
• 增强型CCP（ECCP）模块：
  - 可通过软件选择时基
  - 自动关闭和自动重启
  - PWM转向
• 带有SPI和I2C的主同步串行端口（MasterSynchronous Serial Port， MSSP）：
- 7位地址掩码
• 增强型通用同步/异步收发器（Enhanced UniversalSynchronous Asynchronous ReceiverTransmitter，EUSART）模块：
  - 兼容RS-232、 RS-485和LIN
  - 自动波特率检测
• 电容传感（Capacitive Sensing，CPS）模块：
  -最多8路输入通道
• 数据信号调制器模块：
  - 可选择调制器和载波源
• SR锁存器：
  - 多个置1/复位输入选项
  - 仿真555定时器应用

器件	ROM	数据存储器		I/O	10 位 A/D 转换器	电容传感器通道	比较器	定时器（8/16）	EUSART	MSSP	SR 锁存器	封装
	MTP	EEPROM	SRAM									
MDT10F1822	2048	256	128	6	4	4	1	2/1	1	1	1	DIP、SOP8
MDT10F1823	2048	256	128	12	8	8	2	2/1	1	1	1	DIP、SOP14

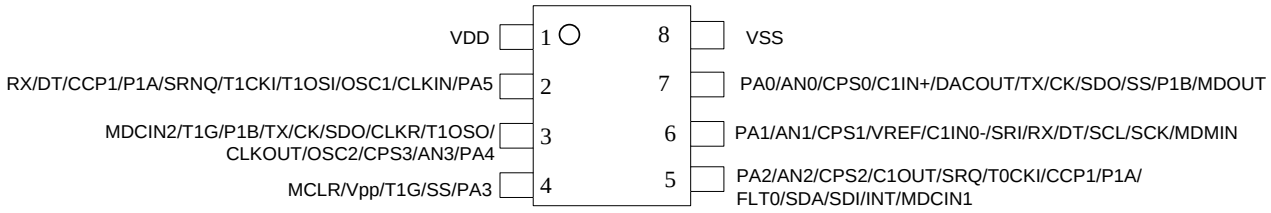
目录

1.1 系统结构图



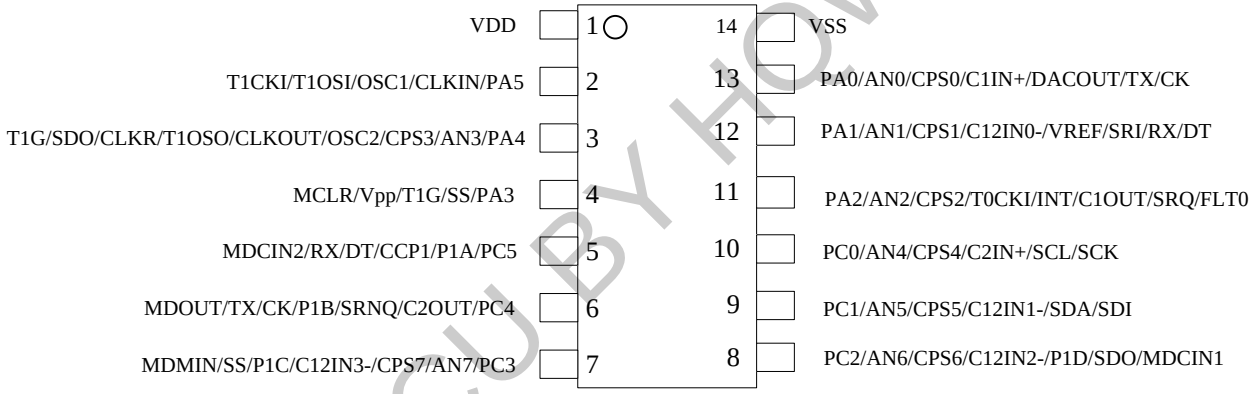
1.2 封装脚位图

1.2.1 8 引脚图（DIP、SOP）



MDT10F1822

1.2.2 14 引脚图（DIP、SOP）



MDT10F1823

1.3 引脚说明

名称	功能	输入类型	输出类型	说明
PA0/AN0/CPS0/C1IN+/DACOUT/TX/CK /ICSPDAT/ICDDAT	PA0	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN0	AN	—	A/D通道0输入
	CPS0	AN	—	电容传感输入0
	C1IN+	AN	—	比较器C1同相输入
	DACOUT	—	AN	数模转换器输出
	TX	—	CMOS	USART异步发送
	CK	ST	CMOS	USART同步时钟
	ICSPDAT	ST	CMOS	串行编程数据
PA1/AN1/CPS1/C12IN0-/VREF/SRI/RX/DT/ICSPCLK	PA1	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN1	AN	—	A/D通道1输入
	CPS1	AN	—	电容传感输入1
	VREF	AN	—	A/D和DAC正参考电压输入
	C12IN0-	AN	—	比较器C1或C2的反相输入
	SRI	ST	—	SR锁存器输入

	RX	ST	—	USART异步输入
	DT	ST	CMOS	USART同步数据
	ICSPCLK	ST	—	串行编程时钟
PA2/AN2/CPS2/T0CKI/INT/ C1OUT/SRQ/FLT0	PA2	ST	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN2	AN	—	A/D通道2输入
	CPS2	AN	—	电容传感输入2
	T0CKI	ST	—	Timer0时钟输入
	INT	ST	—	外部中断
	C1OUT	—	CMOS	比较器C1的输出
	SRQ	—	CMOS	SR锁存器未反相输出
	FLT0	ST	—	ECCP自动关闭故障输入
PA3/$\overline{MCLR}$/Vpp/T1G/$\overline{SS}$	PA3	TTL	—	带电平变化中断的PORTA输入
	$\overline{MCLR}$	ST	—	带有内部上拉的主复位
	Vpp	HV	—	编程电压(9.5V)
	T1G	ST	—	Timer1门控输入
	$\overline{SS}$	ST	—	从选择输入
PA4/AN3/CPS3/OSC2/CLKOUT/ T1OSO/CLKR/SDO/$\overline{T1G}$	PA4	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN3	AN	—	A/D通道3输入
	$\overline{T1G}$	ST	—	Timer1门控输入
	CPS3	AN	—	电容传感输入3
	OSC2	XTAL	XTAL	晶振/谐振（LP、XT和HS模式）
	CLKOUT	—	CMOS	FOSC/4输出
	T1OSO	XTAL	XTAL	Timer1振荡器连接
	CLKR	—	CMOS	时钟参考输出
	SDO	—	CMOS	SPI数据输出
PA5/CLKIN/OSC1/T1OSI/T1CKI	PA5	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	CLKIN	CMOS	—	外部时钟输入（EC模式）
	T1CKI	ST	—	Timer1时钟输入
	OSC1	XTAL	—	晶振/谐振（LP、XT和HS模式）
	T1OSI	XTAL	XTAL	Timer1振荡器连接
PC0/AN4/CPS4/C2IN+/SCL/SCK	PC0	TTL	CMOS	PORTC I/O
	AN4	AN	—	A/D通道4输入
	CPS4	AN	—	电容传感输入4
	C2IN+	AN	—	比较器C2同相输入
	SCL	I2C	OD	I2C时钟
	SCK	ST	CMOS	SPI时钟
PC1/AN5/CPS5/C12IN1-/SDA/ SDI	PC1	TTL	CMOS	PORTC I/O
	AN5	AN	—	A/D通道5输入
	CPS5	AN	—	电容传感输入5
	C12IN1-	AN	—	比较器C1或C2的反相输入
	SDA	I2C	OD	I2C数据输入/输出
	SDI	CMOS	—	SPI数据输入
PC2/AN6/CPS6/C12IN2-/P1D/ SDO/MDCIN1	PC2	TTL	CMOS	PORTC I/O
	AN6	AN	—	A/D通道6输入
	CPS6	AN	—	电容传感输入6
	C12IN2-	AN	—	比较器C1或C2的反相输入
	P1D	—	CMOS	PWM输出
	SDO	—	CMOS	SPI数据输出
	MDCIN1	ST	—	调制器载波输入1
PC3/AN7/CPS7/C12IN3-/P1C /SS/MDMIN	PC3	TTL	CMOS	PORTC I/O
	AN7	AN	—	A/D通道7输入
	CPS7	AN	—	电容传感输入7
	C12IN3-	AN	—	比较器C1或C2的反相输入
	P1C	—	CMOS	PWM输出
	SS	ST	—	从选择输入

	MDMIN	ST	—	调制器源输入
PC4/C2OUT/SRNQ/P1B/CK/TX/MDOUT	PC4	TTL	CMOS	PORTC I/O
	C2OUT	—	CMOS	比较器C2的输出
	SRNQ	—	CMOS	SR锁存器的反相输出
	P1B	—	CMOS	PWM输出
	CK	ST	CMOS	USART同步时钟
	TX	—	CMOS	USART异步发送
	MDOUT	—	CMOS	调制器输出
PC5/P1A/CCP1/DT/RX/MDCIN2	PC5	TTL	CMOS	PORTC I/O
	P1A	—	CMOS	PWM输出
	CCP1	ST	CMOS	捕捉/比较/PWM
	DT	ST	CMOS	USART同步数据
	RX	ST	—	USART异步输入
	MDCIN2	ST	—	调制载波输入2
VDD	VDD	电源	—	正电源端
VSS	VSS	电源	—	接地参考端

图注： AN = 模拟输入或输出 CMOS = CMOS兼容输入或输出 HV = 高压
TTL = TTL兼容输出 XTAL = 晶振 OD = 漏极开漏
ST = 带COMS电平的施密特触发器输入

2.0 增强性中档 CPU

MDT10F1822/1823 包含增强性 8 位 CPU 内核，具有 49 条指令。中断功能包含自动现场保护，具有 16 级内部硬件堆栈，堆栈的上溢与下溢能使 MCU 发生复位操作。MDT10F1822/1823 提供直接寻址、间接寻址和相对寻址模式。

2.1 自动中断现场保护

在中断期间，MDT10F1822/1823 硬件会自动将一些寄存器保存到影子寄存器中，从中断返回时则会恢复这些寄存节省堆空间和用户代码。更多信息，请参见第 8.5 节“自动现场保护”。

2.2 带有上溢和下溢的 16 级堆栈

外部堆栈是具有 15 位宽、16 级深的存储器。在堆栈发生上溢或下溢时，PSTA 寄存器中相应位（STKOVF 或 STKUNF）会置 1，如果使能复位，则会导致软件复位。更多详细信息，请参见第 3.4.5 节“堆栈”。

2.3 文件选择寄存器

有两个 16 位文件选择寄存器（MSR），MSR 可以访问所有文件寄存器和程序存储器，支持对于所有存储器使用一个数据指针。当 MSR 指向程序存储器时，使用 IAR 的指令需要一个额外的指令周期，用于取数据。通用存储器现在可以线性寻址，支持访问大于 80 字节的连续数据。此外，还有一些支持 MSR 新指令。更多详细信息，请参见第 3.5 节“间接寻址”。

2.4 指令集

增强型 CPU 具有 49 条指令，用于支持 CPU 特性。更多详细信息，请参见第 28.0 节“指令表”。

3.0 存储器构成

3.1 程序存储器构成

MDT10F1822/1823 在物理上实现了 2Kx14（0000h-07FFh）的存储空间。访问该边界以外的单元将导致实际访问存储器的第一个 2Kx14 存储空间。复位向量地址为 0000h，中断向量地址为 0004h。

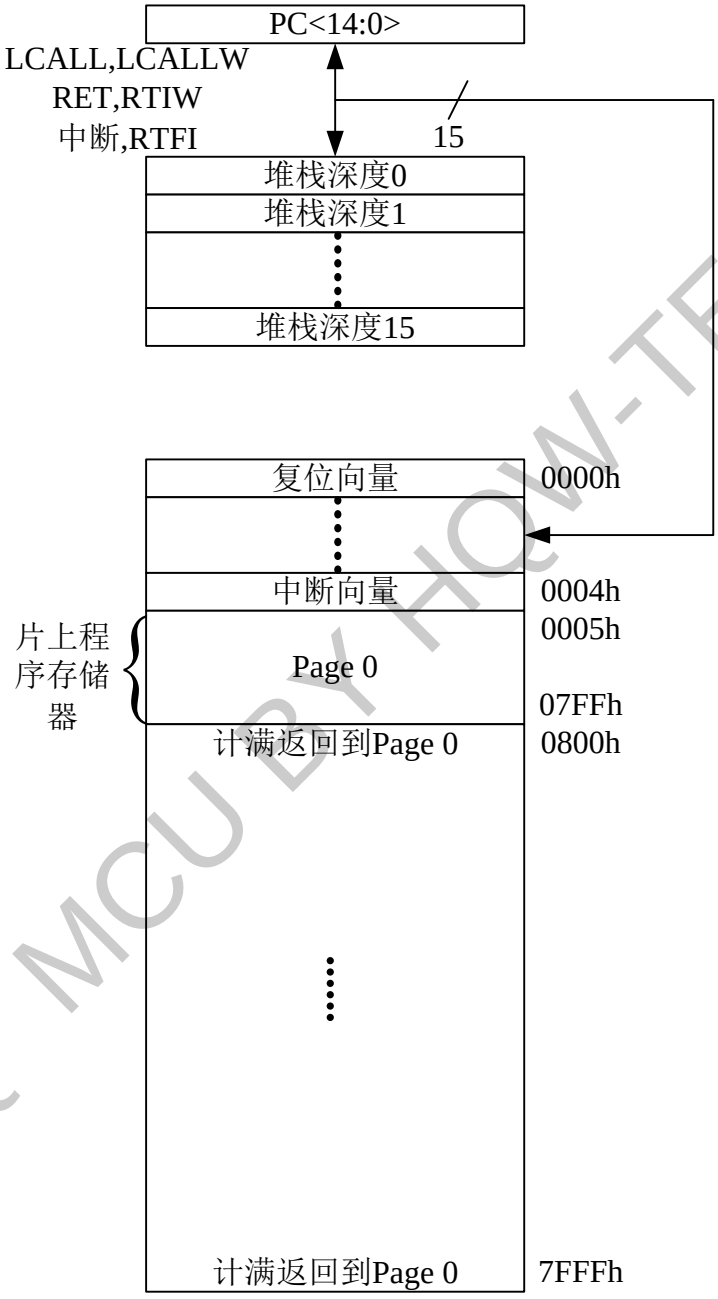


图 3-1：程序存储器映射和堆栈

3.1.1 复位向量（0000H）

具有一个字长的系统复位向量（0000H）。

- 上电复位；
- 欠压复位；
- 外部复位；
- 看门狗复位；
- RESET 复位；
- 堆栈上下溢复位；
- 编程模式退出；

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。根据 STATUS 寄存器中的 TO 和 PD 标志位的内容可以判断系统复位方式。

3.1.2 中断向量（0004H）

中断向量的地址为 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。

3.2 数据存储器构成

数据存储器划分为 32 个存储区，每个存储区有 128 字节。每个存储区都包含（图 3-2）：

• 12个内核寄存器
• 20个特殊功能寄存器（Special Function Registers，SFR）
• 最多80 字节的通用RAM（General Purpose RAM，GPR）
• 16字节的公共RAM

工作存储区的选择通过向存储区选择寄存器（Bank Select Register， BSR）写入存储区编号来进行。未实现的存储器将读为0。所有数据存储器可以直接访问所有数据存储器（通过使用文件寄存器的指令），也可以通过两个文件选择寄存器（MSR）间接访问。

3.2.1 内核寄存器

内核寄存器包含直接影响MDT10F1822/1823基本操作的寄存器。这些寄存器如下所列：

• IAR0
• IAR1
• PCL
• STATUS
• MSR0低字节
• MSR0高字节
• MSR1低字节
• MSR1高字节
• BSR
• WREG
• PCLATH
• INTS

注： 内核寄存器位于每个数据存储区的前 12 个地址处

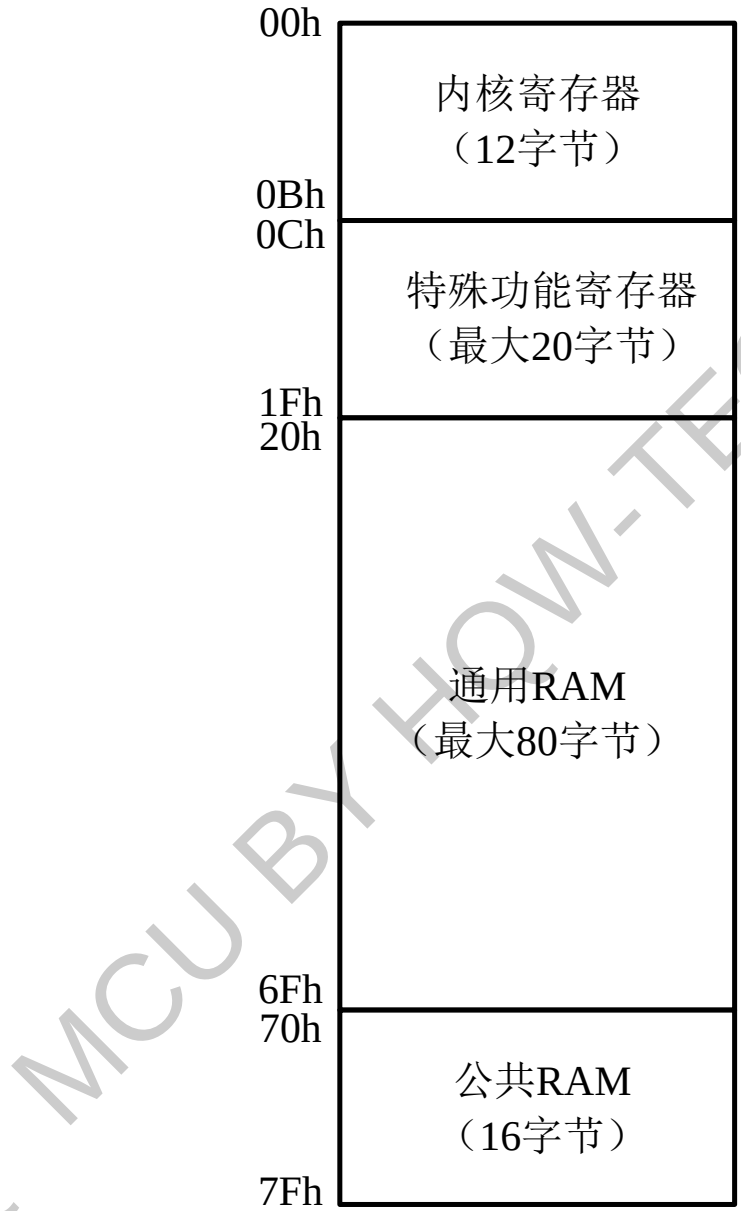


图3-2　存储区分区

3.2.2 特殊功能寄存器汇总

3.2.2.1 特殊功能寄存器位定义（BANK0）

BANK0										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	00H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	01H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	02H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	03H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	04H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	05H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	06H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	07H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	08H(1)	—	—	—	BSR<4:0>					---00000
WREG	09H(1)	工作寄存器								0000 0000
PCLATH	0AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	0BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
PORTA	0CH	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--xx xxxx
—	0DH	未实现								—
PORTC	0EH	—	—	PC5	PC4	PC3	PC2	PC1	PC0	--xx xxxx
—	0FH	未实现								—
—	10H	未实现								—
PIFB1	11H	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000
PIFB2	12H	OSFIF	C2IF	C1IF	EEIF	BCL1IF	—	—	—	0000 0---
—	13H	未实现								—
—	14H	未实现								—
TMR0	15H	Timer0 模块寄存器								xxxx xxxx
TMR1L	16H	16 位 TMR1 寄存器最低有效字节的保持寄存器								xxxx xxxx
TMR1H	17H	16 位 TMR1 寄存器最高有效字节的保持寄存器								xxxx xxxx
T1STA	18H	TMR1CS1	TMR1CS0	T1CKPS<1:0>		T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	0000 00-0
T1GCON	19H	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{DONE}$	T1GVAL	T1GSS<1:0>		0000 0x00
TMR2	1AH	Timer2 模块寄存器								0000 0000
PR2	1BH	Timer2 周期寄存器								1111 1111
T2STA	1CH	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000
—	1DH	未实现								—
CPSCON0	1EH	CPSON	CPSRM	—	—	CPSRNG<1:0>		CPSOUT	T0XCS	00-- 0000
CPSCON1	1FH	—	—	—	—	CPSCH<3:2>		CPSCH<1:0>		---- 0000

图注： - = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.2.2.2 特殊功能寄存器位定义（BANK1）

BANK1										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	80H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	81H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	82H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	83H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	84H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	85H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	86H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	87H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	88H(1)	—	—	—	BSR<4:0>					---00000
WREG	89H(1)	工作寄存器								0000 0000
PCLATH	8AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	8BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
CPIOA	8CH	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111
—	8DH	未实现								—
CPIOC	8EH	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	--11 1111
—	8FH	未实现								—
—	90H	未实现								—
PIEB1	91H	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000
PIEB2	92H	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	—	0000 0---
—	93H	未实现								—
—	94H	未实现								—
OPTION	95H	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111
PSTA	96H	STKOVF	STKUNF	—	—	MCLR	RI	POR	BOR	00-- 11qq
WDTCON	97H	—	—	WDTPS<4:0>					SWDTEN	--01 0110
—	98H	未实现								—
OSCCON	99H	—	IRCF<3:0>				—	SCS<1:0>		0011 1-00
OSCSTAT	9AH	T1OSCR	—	OSTS	HFIOFR	HFIOFL	—	LFIOFR	HFIOFS	10q0 0q00
ADRESL	9BH	A/D 结果寄存器的低字节								xxxx xxxx
ADRESH	9CH	A/D 结果寄存器的高字节								xxxx xxxx
ADC0CN0	9DH	—	CHS<4:0>					GO/DONE	ADON	-000 0000
ADC0CN1	9EH	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		0000 --00
—	9FH	未实现								—

图注：- = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.2.2.3 特殊功能寄存器位定义（BANK2）

BANK2										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	100H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	101H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	102H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	103H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	104H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	105H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	106H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	107H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	108H(1)	—	—	—	BSR<4:0>					---00000
WREG	109H(1)	工作寄存器								0000 0000
PCLATH	10AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	10BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
LATA	10CH	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx
—	10DH	未实现								—
LATC	10EH	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	--xx xxxx
—	10FH	未实现								—
—	110H	未实现								—
CM1CON0	111H	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	0000 -100
CM1CON1	112H	C1INTP	C1INTN	C1PCH<1:0>		—	—	C1NCH1(2)	C1NCH0	0000 ---0
CM2CON0	113H	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	0000 -100
CM2CON1	114H	C2INTP	C2INTN	C2PCH<1:0>		—	—	C2NCH<1:0>		0000 --00
CMOUT	115H	—	—	—	—	—	—	MC2OUT(2)	MC1OUT	---- --00
BORCON	116H	SBOREN	—	—	—	—	—	—	BORRDY	1--- ---q
FVRCON	117H	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		0q00 0000
DACCON0	118H	DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	—	000- 00--
DACCON1	119H	—	—	—	DACR<4:0>					---0 0000
SRCON0	11AH	SRLEN	SRCLK<2:0>			SRQEN	SRNQEN	SRPS	SRPR	0000 0000
SRCON1	11BH	SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRPC2E	SRPC1E	0000 0000
—	11CH	未实现								—
APFCON	11DH	RXDTSEL	SDOSEL	SSSEL	—	T1GSEL	TXCKSEL	P1BSEL	CCP1SEL	000- 0000
—	11EH	未实现								—
—	11FH	未实现								—

图注：- = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.2.2.4 特殊功能寄存器位定义（BANK3）

BANK3										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	180H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	181H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	182H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	183H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	184H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	185H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	186H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	187H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	188H(1)	—	—	—	BSR<4:0>					---00000
WREG	189H(1)	工作寄存器								0000 0000
PCLATH	18AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	18BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
ADINSA	18CH	—	—	—	AN3	—	AN2	AN1	AN0	---1 -111
—	18DH	未实现								—
ADINSC	18EH	—	—	—	—	AN7	AN6	AN5	AN4	---- 1111
—	18FH	未实现								—
—	190H	未实现								—
EEADRL	191H	EEPROM 存储器地址寄存器的低字节								0000 0000
EEDATL	193H	EEPROM 存储器读数据寄存器的低字节								xxxx xxxx
EECON1	195H	EEPGD	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	0000 x000
EECON2	196H	EEPROM 控制寄存器 2								0000 0000
—	197H	未实现								—
—	198H	未实现								—
RCREG	199H	USART 接收数据寄存器								0000 0000
TXREG	19AH	USART 发送数据寄存器								0000 0000
SPBRGL	19BH	波特率发生器数据寄存器的低字节								0000 0000
SPBRGH	19CH	波特率发生器数据寄存器的高字节								0000 0000
RCSTA	19DH	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x
TXSTA	19EH	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010
BAUDCON	19FH	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00

图注：- = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.2.2.5 特殊功能寄存器位定义（BANK4）

BANK4										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	200H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	201H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	202H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	203H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	204H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	205H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	206H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	207H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	208H(1)	—	—	—	BSR<4:0>					---00000
WREG	209H(1)	工作寄存器								0000 0000
PCLATH	20AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	20BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
PAPHR	20CH	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0	--11 1111
—	20DH	未实现								—
PCPHR	20EH	—	—	PCPHR5	PCPHR4	PCPHR3	PCPHR2	PCPHR1	PCPHR0	--11 1111
—	20FH	未实现								—
—	210H	未实现								—
SSP1BUF	211H	同步串行口接收缓冲/ 发送寄存器								xxxx xxxx
SSP1ADD	212H	ADD<7:0>								0000 0000
SSP1MSK	213H	MSK<7:0>								1111 1111
SSP1STAT	214H	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000
SSP1CON1	215H	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000
SSP1CON2	216H	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000
SSP1CON3	217H	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000
—	218H	未实现								—
—	219H	未实现								—
—	21AH	未实现								—
—	21BH	未实现								—
—	21CH	未实现								—
—	21DH	未实现								—
—	21EH	未实现								—
—	21FH	未实现								—

图注： - = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.2.2.6 特殊功能寄存器位定义（BANK5）

BANK5										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	280H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	281H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	282H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	283H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	284H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	285H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	286H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	287H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	288H(1)	—	—	—	BSR<4:0>					---00000
WREG	289H(1)	工作寄存器								0000 0000
PCLATH	28AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	28BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
—	28CH	未实现								—
—	28DH	未实现								—
—	28EH	未实现								—
—	28FH	未实现								—
—	290H	未实现								—
CCPR1L	291H	捕捉/ 比较/PWM 寄存器 1 （LSB）								xxxx xxxx
CCPR1H	292H	捕捉/ 比较/PWM（从动）寄存器 1 （MSB）								xxxx xxxx
CCP1CON	293H	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				0000 0000
PWM1CON	294H	P1RSEN	P1DC<6:0>							0000 0000
CCP1AS	295H	CCP1ASE	CCP1AS<2:0>			PSS1AC<1:0>		PSS1BD<1:0>		0000 0000
PSTR1CON	296H	—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A	---0 0001
—	297H	未实现								—
—	298H	未实现								—
—	299H	未实现								—
—	29AH	未实现								—
—	29BH	未实现								—
—	29CH	未实现								—
—	29DH	未实现								—
—	29EH	未实现								—
—	29FH	未实现								—

图注：- = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.2.2.7 特殊功能寄存器位定义（BANK7）

BANK7										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	380H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	381H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	382H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	383H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	384H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	385H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	386H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	387H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	388H(1)	—	—	—	BSR<4:0>					---00000
WREG	389H(1)	工作寄存器								0000 0000
PCLATH	38AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	38BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
—	38CH	未实现								—
—	38DH	未实现								—
—	38EH	未实现								—
—	38FH	未实现								—
—	390H	未实现								—
IOCAP	391H	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000
IOCAN	392H	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000
IOCAF	393H	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000
—	394H	未实现								—
—	395H	未实现								—
—	396H	未实现								—
—	397H	未实现								—
—	398H	未实现								—
—	399H	未实现								—
CLKRCON	39AH	CLKREN	CLKROE	CLKRSLR	CLKRDC<1:0>		CLKRDIV<2:0>			0011 0000
—	39BH	未实现								—
MDCON	39CH	MDEN	MDOE	MDSLR	MDOPOL	MDOUT	—	—	MDBIT	0010 ---0
MDSRC	39DH	MDMSODIS	—	—	—	MDMS<3:0>				x--- xxxx
MDCARL	39EH	MDCLODIS	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				xxx- xxxx
MDCARH	39FH	MDCHODIS	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				xxx- xxxx

图注：- = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.2.2.8 特殊功能寄存器位定义（BANK31）

BANK31										
名称	地址	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
IAR0	F80H(1)	通过用 MSR0H/MSR0L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
IAR1	F81H(1)	通过用 MSR1H/MSR1L 的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx
PCL	F82H(1)	程序计数器（Program Counter， PC）的最低有效字								0000 0000
STATUS	F83H(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000
MSR0L	F84H(1)	间接数据存储器地址 0 低字节指针								0000 0000
MSR0H	F85H(1)	间接数据存储器地址 0 高字节指针								0000 0000
MSR1L	F86H(1)	间接数据存储器地址 1 低字节指针								0000 0000
MSR1H	F87H(1)	间接数据存储器地址 1 高字节指针								0000 0000
BSR	F88H(1)	—	—	—	BSR<4:0>					---00000
WREG	F89H(1)	工作寄存器								0000 0000
PCLATH	F8AH(1)	—	程序计数器高 7 位的写缓冲区							-0000000
INTS	F8BH(1)	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x
—	F8CH — FE3H	未实现								—
STATUS_SHAD	FE4H	—	—	—	—	—	Z_SHAD	DC_SHAD	C_SHAD	---- -xxx
WREG_SHAD	FE5H	工作寄存器的影子寄存器								0000 0000
BSR_SHAD	FE6H	—	—	—	存储区选择寄存器的影子寄存器					---x xxxx
PCLATH_SHAD	FE7H	—	程序计数器锁存器高字节寄存器的影子寄存器							-xxx xxxx
MSR0L_SHAD	FE8H	间接数据存储器地址 0 低字节指针的影子寄存器								xxxx xxxx
MSR0H_SHAD	FE9H	间接数据存储器地址 0 高字节指针的影子寄存器								xxxx xxxx
MSR1L_SHAD	FEAH	间接数据存储器地址 1 低字节指针的影子寄存器								xxxx xxxx
MSR1H_SHAD	FEBH	间接数据存储器地址 1 高字节指针的影子寄存器								xxxx xxxx
—	FECH	未实现								—
STKPTR	FEDH	—	—	—	当前堆栈指					---1 1111
TOSL	FEEH	栈顶低字节								xxxx xxxx
TOSH	FEFH	—	栈顶高字节							-xxx xxxx

图注： - = 未实现单元读为 0， u = 不变， x = 未知， q = 取值视情况而定，r = 保留，阴影 = 未实现。

注 1：可从任何存储区访问这些寄存器。

3.3 状态寄存器（STATUS）

STATUS寄存器包括：

• ALU的算术运算状态

• 复位状态

STATUS寄存器与任何其他寄存器一样，可作为任何指令的目标寄存器。如果一条影响Z、HC 或C 位的指令以STATUS寄存器作为目标寄存器，那么对这三个位的写操作将被禁止。这些位根据器件逻辑被置1 或清零。而且，TO和 PD位均为不可写位。因此，当执行一条将STATUS寄存器作为目标寄存器的指令时，运行结果可能会与预想的不同。例如， CLRR STATUS将会清零高3 位，并将Z位置1。这将使STATUS寄存器中的值成为000u u1uu（其中u = 不变）。因此，建议仅使用BCR、BSR、SWAPR 和STWR 指令来改变STATUS 寄存器的值，因为这些指令不会影响任何状态位。关于其他不影响任何状态位的指令，请参见第28.0节 “ 指令表”。

注1： 在减法运算中，C 和HC 位分别作为借位位和半借位位。

寄存器03H：状态寄存器（STATUS）

U-0	U-0	U-0	R-1	R-1	R/W-x	R/W-x	R/W-x
—	—	—	**$\overline{TO}$**	**$\overline{PD}$**	**Z**	**HC**	**C**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-5 未实现

bit4 **$\overline{TO}$**：超时状态位
1= 上电后，执行了CLRWT指令或SLEEP指令
0= 发生WDT超时溢出

bit3 **$\overline{PD}$**：掉电标志位
1= 上电复位后或执行了CLRWT指令
0= 执行了SLEEP指令

bit2 **Z**：零标志位
1= 算术运算或逻辑运算的结果为零
0= 算术运算或逻辑运算的结果不为零

bit1 **HC**：半进位/借位位（ADDWR、ADDWI、SUBWR和SUBWI指令）。对于借位，极性是相反的。
1= 结果的第4低位向高位发生了进位
0= 结果的第4低位未向高位发生进位

bit0 **C**：进位/$\overline{借位位}$(1)（ADDWR、ADDWI、SUBWR和SUBWI指令）
1= 结果的最高位发生了进位（减法时，没有发生借位时为1）
0= 结果的最高位未发生进位

注 1：极性是相反的。减法是通过加上第二个操作数的二进制补码（Two’sComplement）来实现的。对于移位指令（RRR和RLR），此位的值来自源寄存器的最高位或最低位。

3.4 PCL 和 PCLATH

程序计数器（PC）为 15 位宽。其低字节来自可读写的 PCL 寄存器，高字节（PC<14:8>）来自 PCLATH，不能直接读写。任何复位都将清零 PC。图 3-3 显示了装载 PC 值的 5 种情形

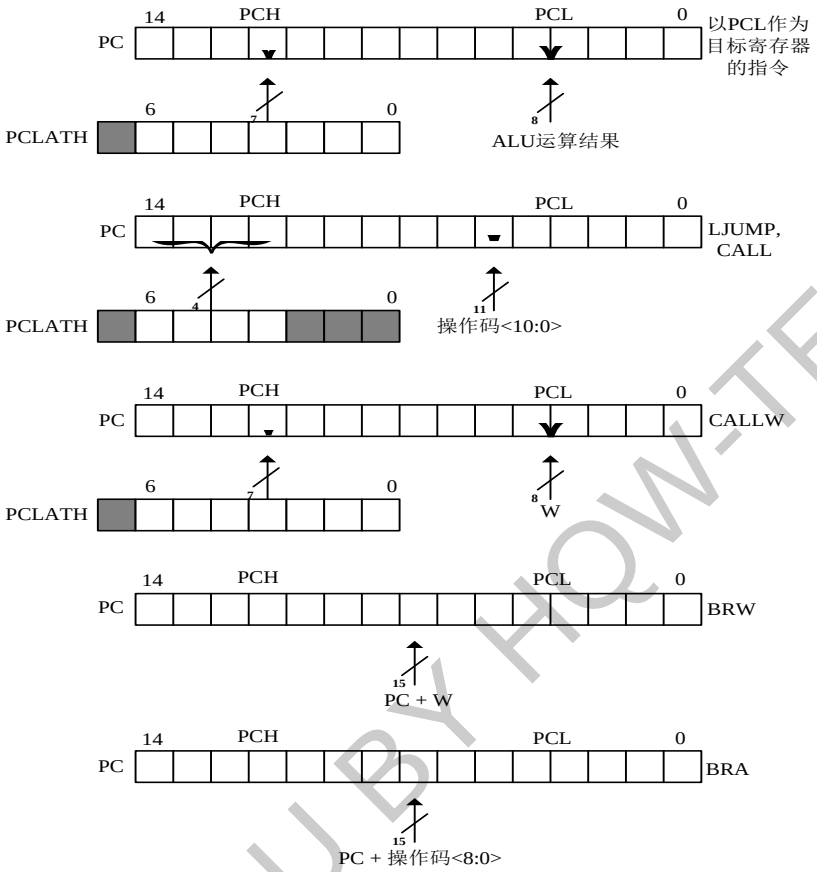


图3-3

3.4.1 修改 PCL

在执行以PCL 寄存器作为目标寄存器的任何指令的同时，也会使程序计数器的PC<14:8> 位（PCH）被PCLATH 寄存器的内容所代替。这使得可以通过将所需的高7 位写入PCLATH寄存器来改变程序计数器的整个内容。当将低8 位写入PCL 寄存器时，程序计数器的所有15 位都将变为PCLATH 寄存器中和那些被写入PCL寄存器的值。

3.4.2 计算 LJUMP

计算LJUMP是通过向程序计数器加一个偏移量（ADDWR PCL）来实现的。当使用计算LJUMP方法执行表读操作时， 应注意表地址是否超出PCL 存储器边界（每个存储块为 256 字节）

3.4.3 计算函数调用

利用计算函数LCALL，程序可以维护一些函数表，并提供另一种执行状态机或查找表的方式。当使用计算函数LCALL执行表读操作时，应注意表地址是否超出了PCL存储器边界（每个存储块为256 字节）。如果使用LCALL指令，PCH<2:0> 和PCL寄存器中将装入LCALL 指令的操作数。PCH<6:3> 中将装入PCLATH<6:3>。CALLW指令通过将PCLATH和W组合构成目标地址来支持计算调用。计算CALLW 通过向W 寄存器中装入所需地址并执行CALLW来实现。PCL寄存器中装入W的值，PCH 中装入PCLATH 的值。

3.4.4 跳转

跳转指令会将一个偏移量与PC相加。这使得可以实现可重定位代码和跨越页边界的代码。存在两种跳转形式：BRW和BRA。在两种形式中，PC都会发生递增，以便取下一条指令。使用任一跳转指令时，都可以跨越PCL存储器边界。如果使用BRW，则向W寄存器中装入所需的无符号地址，然后执行BRW。整个PC中将装入地址PC+1+W。如果使用BRA，整个PC中将装入PC+1+BRA指令操作数的有符号值。

3.4.5 堆栈

所有器件都具有16 级x 15 位宽的硬件堆栈（见图3-4至3-7）。堆栈既不占用程序存储空间，也不占用数据存储空间。当执行LCALL或CALLW指令，或者中断导致程序跳转时，PC值将被压入堆栈。而在执行RET、RTIW或RTFI指令时，将从堆栈中弹出PC值。PCLATH不受压栈或出栈操作的影响。如果STVREN位被设置为0（配置字2），堆栈将作为循环缓冲区工作。这意味着当压栈16次后，第17次压入堆栈的值将会覆盖第一次压栈时所保存的值，而第18次压入堆栈的值将覆盖第二次压栈时所保存的值，依此类推。无论是否使能了复位，STKOVF和STKUNF标志位都将在上溢/下溢时置1。

注：1： 不存在被称为PUSH 或 POP 的指令/ 助记符。堆栈的压入或弹出是源于执行了LCALL、CALLW、RET、RTIW 和 RTFI指令，或源于跳转到中断向量地址。

3.4.6 访问堆栈

通过TOSH、TOSL 和STKPTR 寄存器可以使用堆栈。STKPTR 是堆栈指针的当前值。TOSH:TOSL 寄存器对是指向栈顶。两个寄存器都是可读写的。由于PC 的大小为15 位，所以TOS 拆分为TOSH 和TOSL。要访问堆栈，可以调整STKPTR 的值（它会决定TOSH:TOSL位置），然后读写TOSH:TOSL。STKPTR 的宽度为5位，以允许检测上溢和下溢。在正常程序操作期间， LCALL、CALLW 和中断会使STKPTR 递增，而RTIW、RET 和RTFI 会使STKPTR 递减。在任意时刻，都可以通过检查STKPTR来确定所剩余的堆栈空间。STKPTR 总是指向堆栈中当前使用的位置。因此， LCALL 或LCALLW 会先递增STKPTR，然后再写入PC，而返回操作则会先取出PC，然后再递减STKPTR。访问堆栈示意图如下图：

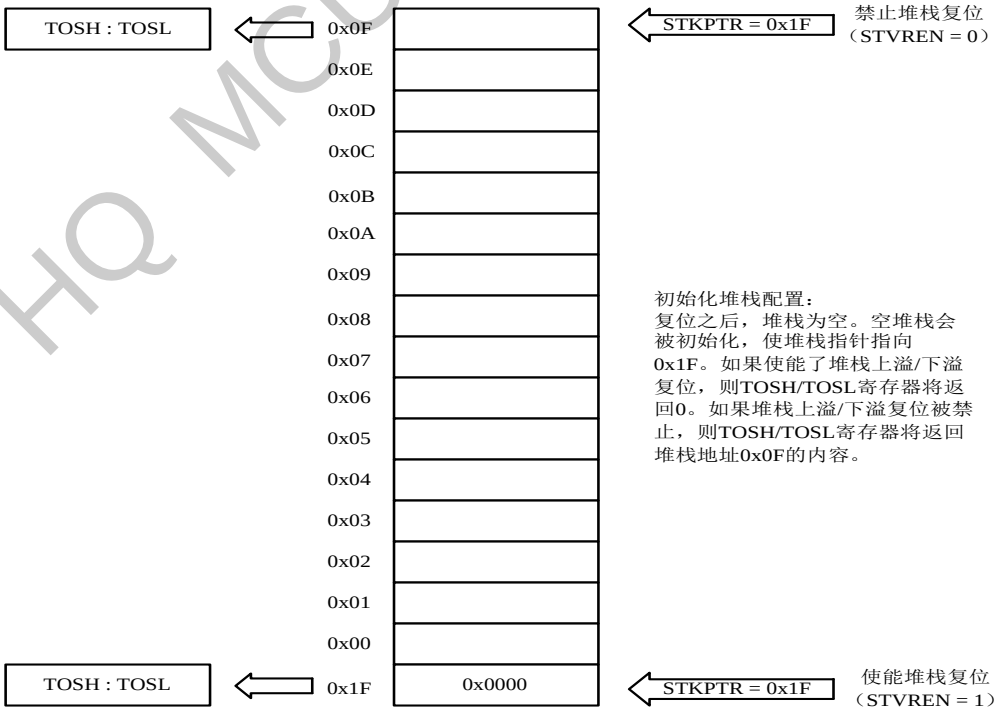


图3-4

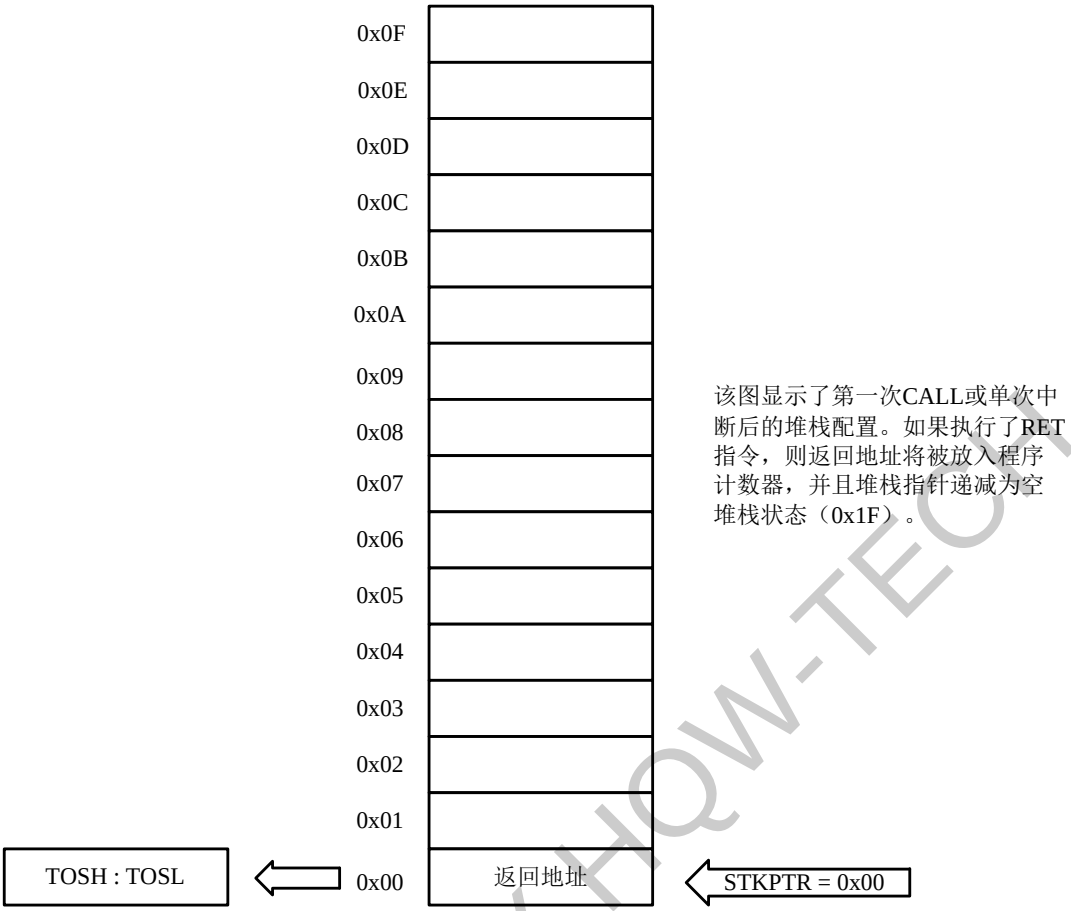


图3-5

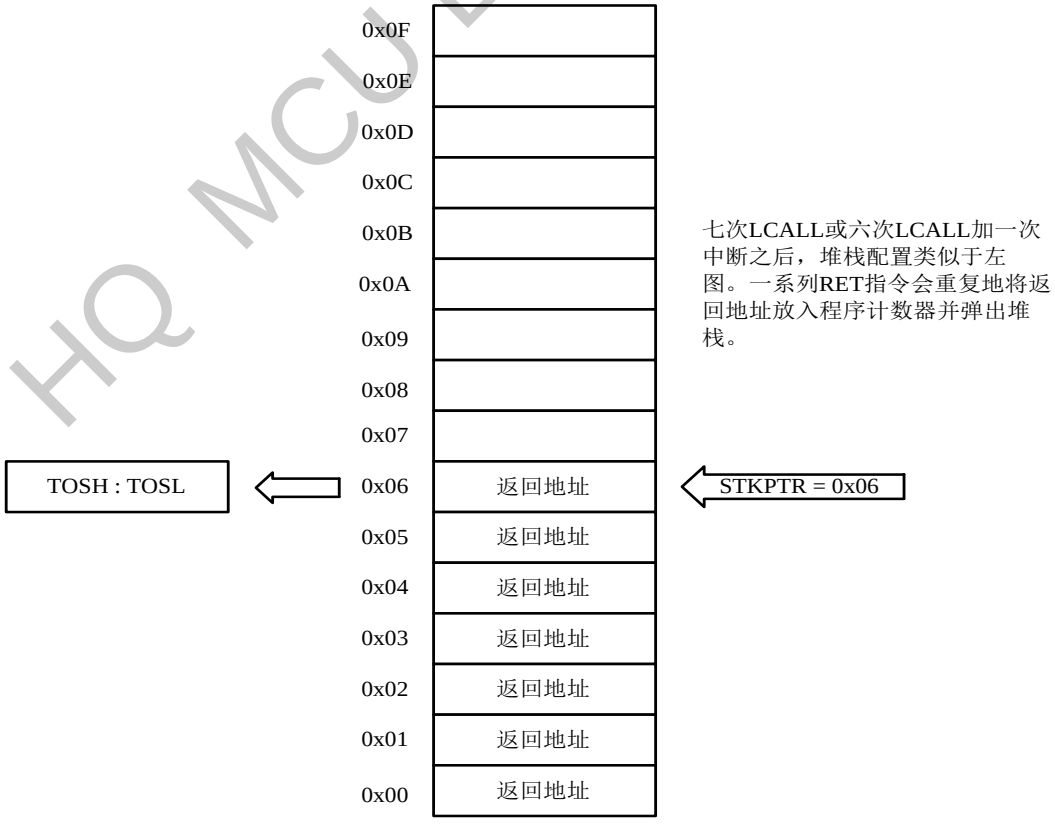


图3-6

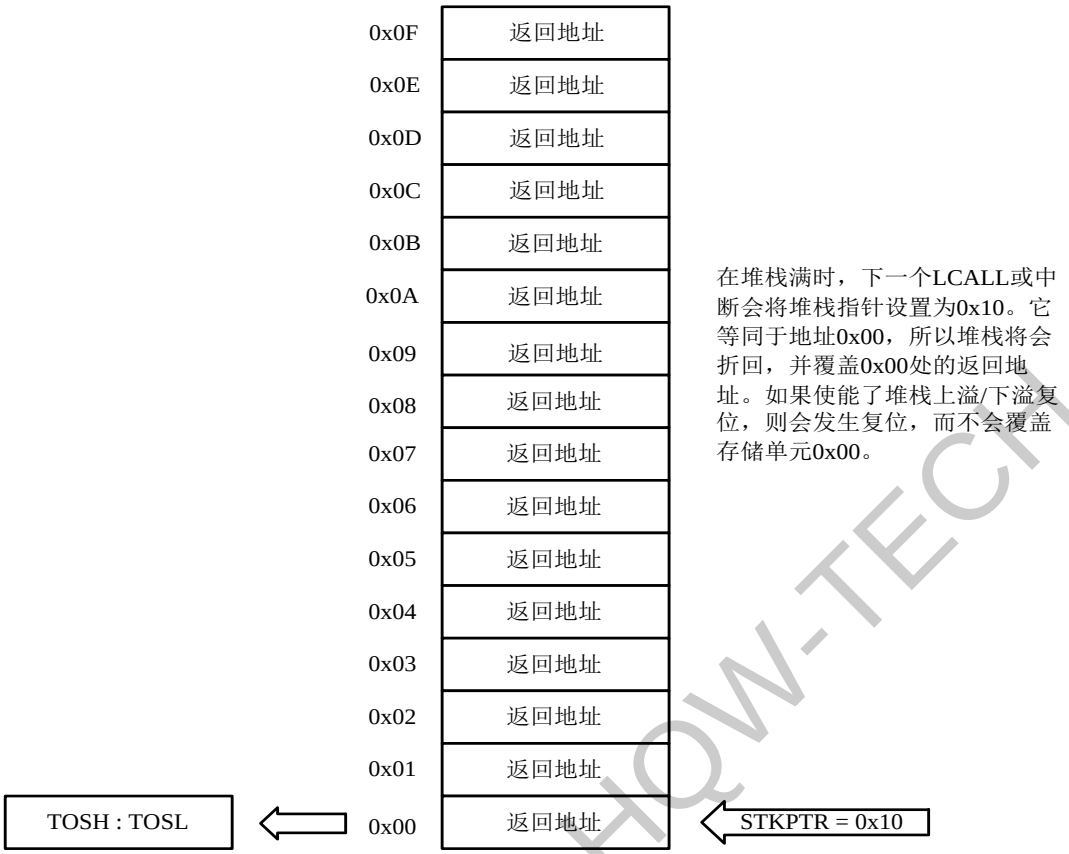


图3-7

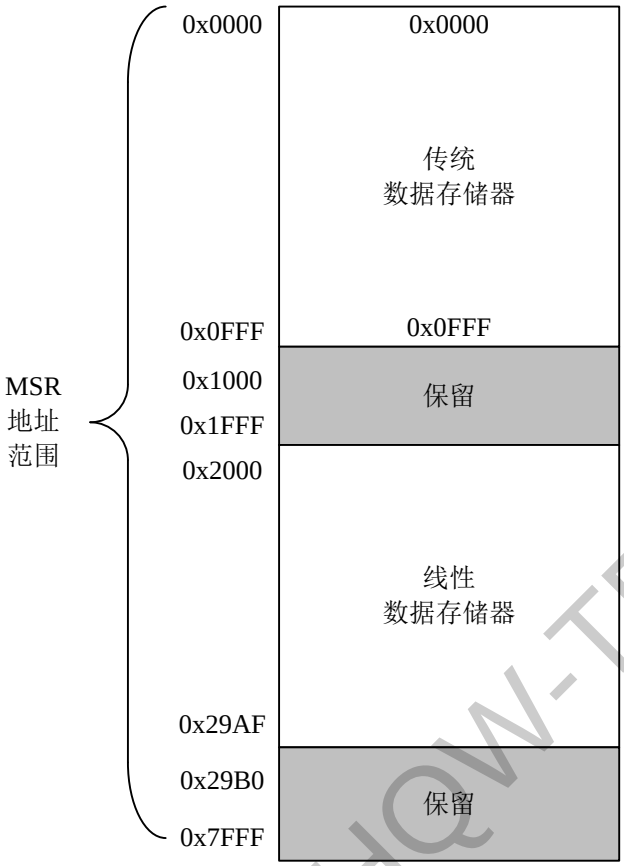
3.4.7 上溢/下溢复位

如果配置字寄存器2 中的STVREN 位被设置为1，则在压栈操作超过堆栈第16 级或出栈操作超过堆栈第1级时，器件会发生复位，并将PSTA 寄存器中的相应位（分别为STKOVF 或STKUNF）置1。

3.5 间接寻址

IARn寄存器不是物理寄存器。访问IARn寄存器的所有指令实际上访问的是由文件选择寄存器（MSR）指定的地址处的寄存器。如果MSRn地址指定了两个IARn寄存器中的一个，则读操作将返回0，写操作将不会发生（虽然状态位可能会受影响）。MSRn寄存器值由MSRnH和MSRnL对构成。MSR 寄存器构成一个16 位地址，支持65536 个存储单元的寻址空间。这些存储单元分为2 个存储器区域：

- 传统数据存储器
- 线性数据存储器



3.5.1 传统数据存储器

传统数据存储器是从MSR地址0x000至MSR地址0xFFF的区域。这些地址对应于所有SFR、GPR 和公共寄存器的绝对地址。

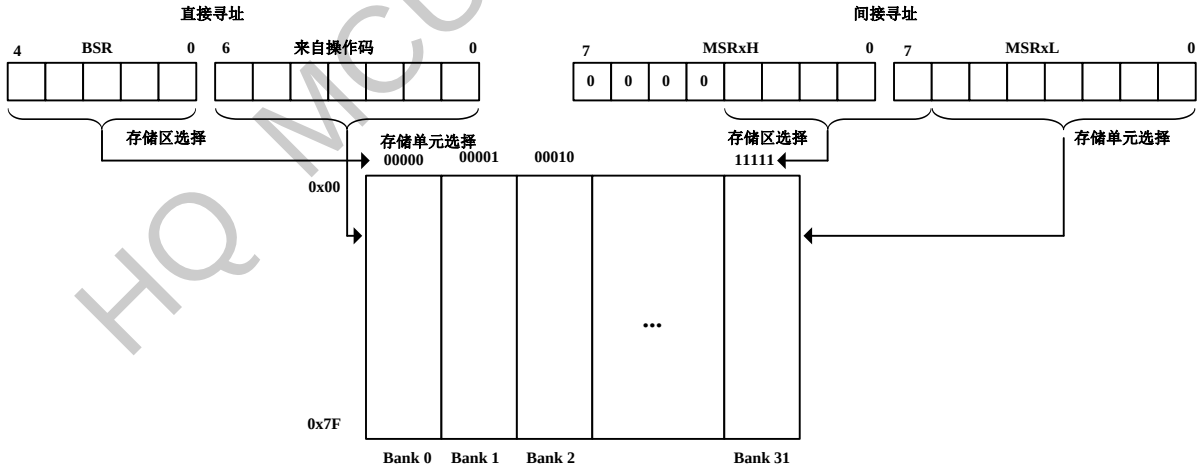


图3-8 传统数据存储器映射

4.0 器件配置

器件配置功能由配置字寄存器1 和配置字寄存器2、代码保护以及器件ID 组成。

4.1 配置字

有几个配置字位可用于选择不同的振荡器和存储器保护选项。这些位实现为位于8007h 的配置字寄存器1 和位于8008h 的配置字寄存器2。

寄存器 8007H：配置字寄存器1（Config1 Option）

U-0	U-0	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
—	—	**FCMEN**	**IESO**	**$\overline{\text{CLKOUTEN}}$**	**BOREN1**	**BOREN0**	**$\overline{\text{CPD}}$**
bit15							bit8

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
$\overline{\text{CP}}$	**MCLRE**	**$\overline{\text{PWRTE}}$**	**WDTE1**	**WDTE0**	**FOSC2**	**FOSC1**	**FOSC0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	x= 未知
-n=POR时的值	1= 置1	0= 清零	P= 可编程位

bit15-14 未实现：保留

bit13 **FCMEN：** 故障保护时钟监视器使能位
1 = 使能故障保护时钟监视器
0 = 禁止故障保护时钟监视器

bit12 **IESO：** 内/ 外部切换位
1 = 使能内/ 外部切换模式
0 = 禁止内/ 外部切换模式

bit11 **$\overline{\text{CLKOUTEN}}$**：时钟输出使能
如果FOSC 配置位被设置为LP、XT 或HS 模式：
该位被忽略，禁止CLKOUT 功能。CLKOUT 引脚为振荡器功能。
所有其他FOSC 模式：
1 = 禁止CLKOUT 功能。CLKOUT 引脚为I/O 功能。
0 = 在CLKOUT 引脚使能CLKOUT 功能

bit10-9 **BOREN<1:0>：** 欠压复位使能位(1)
11 = 使能BOR
10 = 在工作期间使能BOR，在休眠期间禁止BOR
01 = BOR 由BORCON 寄存器的SBOREN 位控制
00 = 禁止BOR

bit8 **$\overline{\text{CPD}}$**：数据代码保护位(2)
1 = 禁止数据存储器代码保护
0 = 使能数据存储器代码保护

bit7 **$\overline{\text{CP}}$**：代码保护位(3)
1 = 禁止程序存储器代码保护
0 = 使能程序存储器代码保护

bit6 **MCLRE**：PA3/MCLR引脚功能选择
1 = PA3/MCLR引脚为外部复位引脚MCLR,使能弱上拉
0 = PA3/MCLR引脚为数字输入，MCLR功能在内部被禁止，弱上拉有PAPHR控制

bit5 **$\overline{\text{PWRTE}}$**：上电延时定时器使能位(1)
1 = 禁止PWRT
0 = 使能PWRT

bit4-3 **WDTE<1:0>：** 看门狗定时器使能位
11 = 使能WDT
10 = 在运行时使能WDT，在休眠时禁止WDT
01 = WDT 由WDTCON 寄存器的SWDTEN 位控制
00 = 禁止WDT

bit2-0 **FOSC <2:0>**：振荡器选择位
111 = ECH：外部时钟，高功耗模式（4-16 MHz）：器件时钟提供给CLKIN 引脚
110 = ECM：外部时钟，中等功耗模式（0.5-4 MHz）：器件时钟提供给CLKIN 引脚
101 = ECL：外部时钟，低功耗模式（0-0.5 MHz）：器件时钟提供给CLKIN 引脚
100 = INTOSC 振荡器：CLKIN 引脚为I/O 功能
011 = EXTRC 振荡器：外部RC 电路连接到CLKIN 引脚
010 = HS 振荡器：高速晶振/ 谐振器连接在OSC1 和OSC2 引脚之间
001 = XT 振荡器：晶振/ 谐振器连接在OSC1 和OSC2 引脚之间
000 = LP 振荡器：低功耗晶振连接在OSC1 和OSC2 引脚之间

注 1：使能欠压复位不会自动使能上电延时定时器。
2：关闭代码保护时，整个数据EEPROM都将被擦除。

寄存器 8008H：配置字寄存器2（Config1 Option）

U-1	U-1	R/P-1/1	R/P-1/1	U-1	R/P-1/1	R/P-1/1	R-0
—	—	—	—	—	**BORV**	**STVREN**	—
bit15							bit8

U-1	U-1	U-1	U-1	U-1	U-1	R/P-1/1	R/P-1/1
—	—	—	保留	—	—	**WRT1**	**WRT0**
bit7							bit0

图注：
R= 可读位 W= 可写位 U= 未实现位，读为0 x= 未知
-n=POR时的值 1= 置1 0= 清零 P= 可编程位

bit15-14 未实现
bit13 该位应被编程为0
bit12 该位应保持为1
bit11 未实现
bit10 **BORV：** 欠压复位电压选择位
1 = 欠压复位电压（1.9V 典型值），选择低跳变点
0 = 欠压复位电压（2.5V 典型值），选择高跳变点
bit9 **STVREN：** 堆栈上溢/ 下溢复位使能位
1 = 堆栈上溢或下溢将导致复位
0 = 堆栈上溢或下溢不会导致复位
bit8 未实现：读为0
bit7-5 未实现
bit4 该位应被编程为1。
bit3-2 未实现：读为1
bit1-0 **WRT<1:0>：** ROM自写保护位
11 = 写保护关闭
（必须写“11”）

4.2 代码保护

通过代码保护，可以防止对器件的未授权访问。程序存储器保护和数据EEPROM 保护独立进行控制。对数据EEPROM 的内部访问不会受任何代码保护设置影响。

4.2.1 程序存储器保护

整个程序存储空间都通过配置字寄存器1 中的 $\overline{CP}$ 位来防止外部读写操作。当$\overline{CP}$= 0 时，将禁止对程序存储器的外部读写操作，读取时将返回全0。

4.2.2 数据 EEPROM 保护

整个数据EEPROM 通过$\overline{CPD}$位来防止外部读写操作。当$\overline{CPD}$ = 0 时，将禁止对数据EEPROM 的外部读写操作。无论保护位的设置如何，CPU 都可以继续读写数据EEPROM。

5.0 振荡器模块（带故障保护时钟监视）

5.1 概述

振荡器模块具有多种时钟源和选择特性，从而使之可广泛使用于各种应用中，同时最大限度地发挥性能并降低功耗。图5-1 给出了振荡器模块的框图。时钟源可由外部振荡器、石英晶体谐振器、陶瓷谐振器以及阻容（Resistor-Capacitor，RC）电路提供。此外，系统时钟源可由两个内部振荡器之一电路提供，并通过软件来选择速度。其他时钟特性包括：

- 可通过软件选择外部或内部系统时钟源。
- 双速启动模式，最大限度地缩短外部振荡器起振与代码执行之间的延时。
- 故障保护时钟监视器（FSCM），用来检测外部时钟源（LP、XT、HS、EC 或RC 模式）故障并自动切换到内部振荡器。
- 振荡器起振定时器（OST）可确保晶振源的稳定性。

振荡器模块可配置为以下8 种时钟模式之一。

1. ECL—— 外部时钟低功耗模式
（0 MHz 至0.5 MHz）
2. ECM—— 外部时钟中等功耗模式
（0.5 MHz 至4 MHz）
3. ECH—— 外部时钟高功耗模式
（4 MHz 至16 MHz）
4. LP——32 kHz 低功耗晶振模式
5. XT—— 中等增益晶振或陶瓷谐振器模式
（高达4 MHz）
6. HS——高增益晶振或陶瓷谐振器模式
（4 MHz 至 16 MHz）
7. RC—— 外部阻容（RC）
8. INTOSC—— 内部振荡器（31 kHz 至16 MHz）

时钟源模式通过配置字寄存器1中的FOSC<2:0>位进行选择。FOSC 位决定在器件初次上电时使用的振荡器类型。EC 时钟模式依靠外部逻辑电平信号作为器件时钟源。LP、XT 和HS 时钟模式要求器件在外部连接一个晶振或谐振器。每种模式都针对不同频率范围而优化。RC时钟模式需要一个外部电阻和电容来设置振荡器频率。

INTOSC 内部振荡器模块可以产生低频、高频时钟源，分别用LFINTOSC和HFINTOSC表示。（见内部振荡器模块，图5-1）。基于这两个时钟源，可以产生多种器件时钟频率选择。

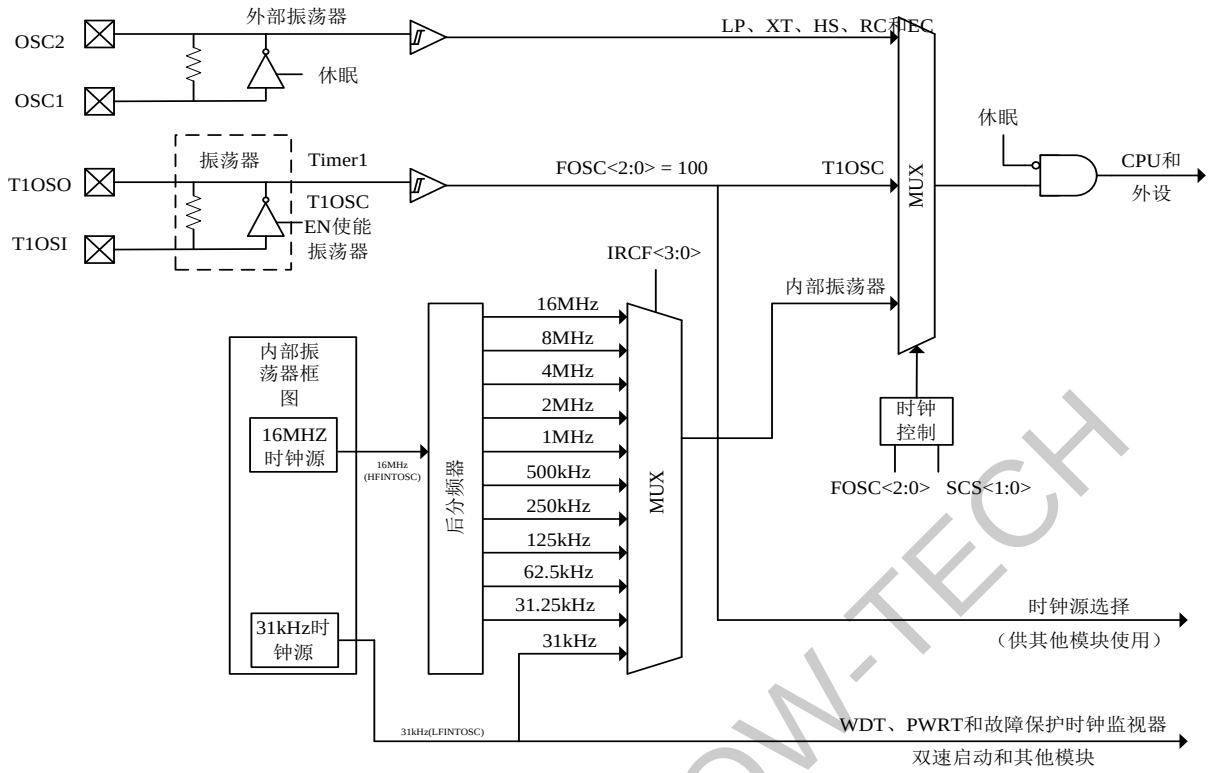


图5-1：MCU的时钟源简化图

5.2 时钟类型

时钟源可分为外部和内部模式。外部时钟源依靠外部电路提供时钟源工作。例如：振荡器模块（EC 模式）、石英晶体谐振器或陶瓷谐振器（LP、XT 和HS 模式）以及阻容（RC）模式电路。内部时钟源内置于振荡器模块中。内部振荡器模块具有两个内部振荡器，用于产生2个内部系统时钟源：16 MHz 高频内部振荡器（HFINTOSC）和31 kHz 低频内部振荡器（LFINTOSC）。通过OSCCON寄存器中的系统时钟选择（SCS）位在外部和内部时钟源之间选择系统时钟。

5.2.1 外部时钟模式

通过执行以下操作之一，可以使用外部时钟源作为器件系统时钟：

• 编程配置字寄存器1 中的FOSC<2:0> 位，选择在器件复位时用作默认系统时钟的外部时钟源。

• 写入OSCCON 寄存器中的SCS<1:0> 位，将系统时钟源切换为：

- Timer1振荡器（在运行时），或者

- 由FOSC 位的值决定的外部时钟源。

5.2.1.1 EC 模式

外部时钟（EC）模式允许外部产生的逻辑电平信号作为系统时钟源。工作在该模式下时，外部时钟源连接到OSC1 输入。OSC2/CLKOUT 可用作通用I/O 或CLKOUT。EC 模式具有3 种功耗模式，可通过配置字寄存器1 进行选择 ：

• 高功耗， 4-16 MHz （FOSC = 111）

• 中等功耗， 0.5-4 MHz （FOSC = 110）

• 低功耗， 0-0.5 MHz （FOSC = 101）

当选取了EC 模式时，振荡器起振定时器（OST）被禁止。因此，上电复位（POR）后或者从休眠中唤醒后的操作不存在延时。因为MCU的设计是完全静态的，停止外部时钟输入将使器件暂停工作并保持所有数据完整。当再次启动外部时钟时，器件恢复工作，就好像没有停止过一样。

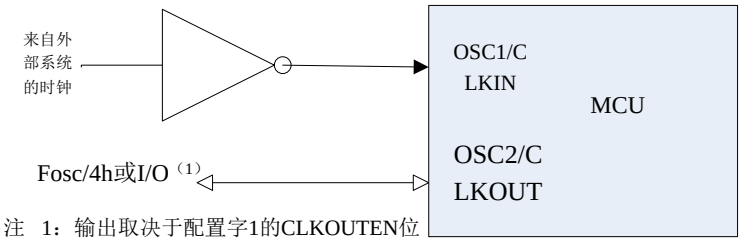
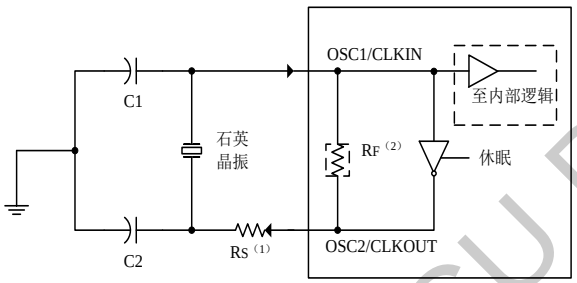


图5-2：外部时钟（EC）模式的工作原理

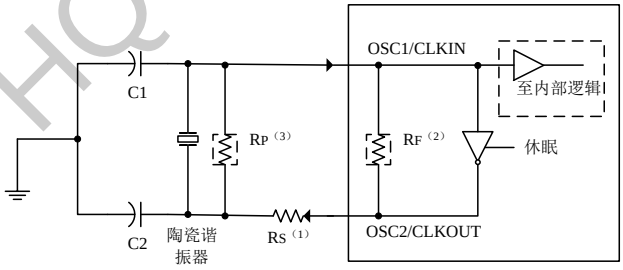
5.2.1.2 LP、XT 和 HS 模式

LP、XT和HS模式支持使用连接到OSC1和OSC2的石英晶体谐振器或陶瓷谐振器（图5-3或图5-4）。这三种模式选择内部反相放大器的低、中或高增益设定，以支持各种谐振器类型及速度。LP 振荡器模式选择内部反相放大器的最低增益设定。LP模式的电流消耗在三种模式中最小。该模式设计用来驱动仅32.768 kHz 的音叉（Tuning Fork）型晶振（钟表晶振）。XT 振荡器模式选择内部反相放大器的中等增益设定。XT 模式的电流消耗在三种模式中居中。该模式最适合驱动具备中等驱动电平规格要求的谐振器。HS 振荡器模式选择内部反相放大器的最高增益设定。HS 模式的电流消耗在三种模式中最大。该模式最适合驱动需要高驱动设定的谐振器。

图5-3和 图5-4 分别给出了石英晶体谐振器和陶瓷谐振器的典型电路。



注 1： 对于具有低驱动电平规格的石英晶振，可能需要串联一个电阻（RS）。
2： RF 的值根据所选的振荡器模式变化（典型值在2 MΩ 至10 MΩ 之间）。



注 1： 对于具有低驱动电平规格的陶瓷谐振器，可能需要串联一个电阻（RS）。
2： RF 的值根据所选的振荡器模式变化（典型值在2 MΩ 至10 MΩ 之间）。
3： 为使陶瓷谐振器正常工作，可能需要并联一个反馈电阻（RP）。

图 5-3： 石英晶振的工作原理（LP、XT 或 HS 模式） 图 5-4：陶瓷谐振器的工作原理（XT 或 HS 模式）

5.2.1.3 振荡器起振定时器（OST）

如果振荡器模块被配置为LP、XT 或HS 模式，则振荡器起振定时器（OST）对来自OSC1 的振荡计数1024次。这发生在上电复位（POR）和上电延时定时器（PWRT）延时结束后（如果配置了） ，或从休眠中唤醒后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经起振并为振荡器模块提供稳定的系统时钟。为了使外部振荡器起振和代码执行之间的延时最小，可选择双速时钟启动模式（见第5.4节“双速时钟启动模式”）。

5.2.1.4 TIMER1 振荡器

Timer1 振荡器是与Timer1 外设关联的独立晶振。它针对计时操作而优化，并且需要在T1OSO 和T1OSI器件引脚之间连接一个32.768 kHz 晶振。Timer1 振荡器可以用作备用系统时钟源，并且可以在运行时通过时钟切换选择。更多信息，请参见第5.3 节 “ 时钟切换”。

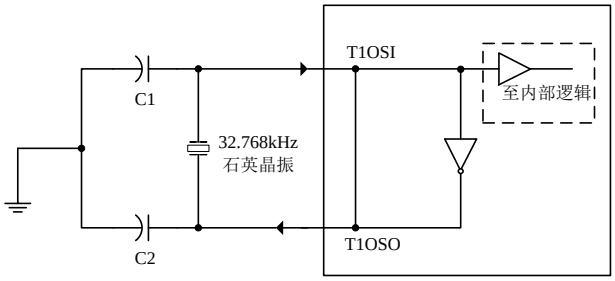


图 5-5: 石英晶振的工作原理（Timer 振荡器）

5.2.1.5 外部 RC 振荡器

外部阻容（RC）模式支持使用外部RC 电路。对时钟精度要求不高时，这使设计人员有了很大的精度选择空间，且保持成本最低。RC电路连接到OSC1。OSC2/CLKOUT 可用作通用I/O或CLKOUT。OSC2/CLKOUT 引脚的功能由配置字寄存器1的CLKOUTEN 位的状态决定。

图5-6 给出了外部RC 模式的连接图。

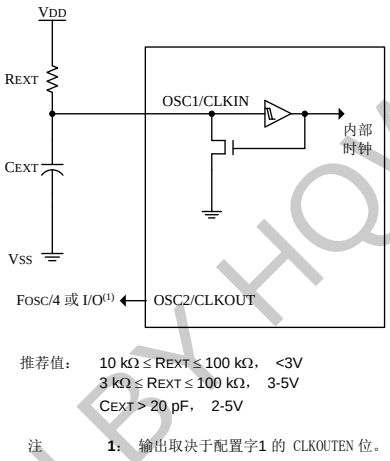


图5-6: 外部RC 模式的连接图

RC振荡器频率是供电电压、电阻（REXT）和电容（CEXT）值以及工作温度的函数。影响振荡器频率的其他因素有：

- 阈值电压变化
- 元件容差
- 不同封装类型的电容差异

用户还应考虑因所使用的外部RC 元件的容差而导致的差异。

5.2.2 内部时钟源

通过执行以下操作之一，可以将器件配置为使用内部振荡器模块作为系统时钟：

- 编程配置字寄存器1中的FOSC<2:0>位来选择INTOSC时钟源，在器件复位时将使用该时钟源作为默认系统时钟。
- 在运行时写入OSCCON寄存器中的SCS<1:0> 位，将系统时钟源切换为内部振荡器。更多信息，请参见第5.3节“ 时钟切换”。

在INTOSC 模式下， OSC1/CLKIN 可用作通用I/O。OSC2/CLKOUT 可用作通用I/O 或CLKOUT。OSC2/CLKOUT引脚的功能由配置字寄存器1中的CLKOUTEN位的状态决定。

内部振荡器模块具有两个独立振荡器：

1. HFINTOSC （高频内部振荡器）出厂时已校准，工作频率为16 MHz。
2. LFINTOSC （低频内部振荡器）未经校准，工作频率为31 kHz。

5.2.2.1 HFINTOSC

高频内部振荡器（HFINTOSC）在出厂时已校准，为16 MHz 内部时钟源。HFINTOSC 的输出连接到后分频器和多路开关（见图5-1）。使用OSCCON 寄存器的IRCF<3:0> 位，可通过软件选择基于 HFINTOSC 产生的9 个频率中的一个。更多信息，请参见 第5.2.2.5 节“ 内部振荡器时钟切换时序”。发生以下情况时,HFINTOSC 被使能：

- 根据所需的HF 频率配置OSCCON 寄存器的IRCF<3:0> 位，并且
- FOSC<2:0> = 100，或者
- 将OSCCON 寄存器的系统时钟源（SCS）位设置为1x。

OSCSTAT 寄存器的高频内部振荡器就绪位（HFIOFR）指示HFINTOSC 何时运行并可使用。OSCSTAT 寄存器的高频内部振荡器状态锁定位（HFIOFL）指示HFINTOSC 何时在距离其最终值2%的范围内运行。OSCSTAT 寄存器的高频内部振荡器状态稳定位（HFIOFS）指示HFINTOSC何时在距离其最终值0.5%的范围内运行。

5.2.2.2 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准的31 kHz内部时钟源。LFINTOSC 的输出连接到多路开关（见图 5-1）。使用OSCCON 寄存器的IRCF<3:0> 位，通过软件选择31 kHz。LFINTOSC 还是上电延时定时器（PWRT）、看门狗定时器（WDT）以及故障保护时钟监视器（FSCM）的时钟源。LFINTOSC 可以通过选择 31 kHz（OSCCON 寄存器的IRCF<3:0> 位= 000）作为系统时钟源（OSCCON 寄存器的SCS 位= 1x）进行使能，也可以通过以下方式使能：

- 根据所需的LF 频率配置OSCCON 寄存器的IRCF<3:0> 位，并且
- FOSC<2:0> = 100，或者
- 将OSCCON 寄存器的系统时钟源（SCS）位设置为1x

使用LFINTOSC 的外设有：

- 上电延时定时器（PWRT）
- 看门狗定时器（WDT）
- 故障保护时钟监视器 （FSCM）

OSCSTAT 寄存器的低频内部振荡器就绪位（LFIOFR）指示LFINTOSC 何时运行并可使用。

5.2.2.3 内部振荡器频率选择

使用OSCCON 寄存器的内部振荡器频率选择位IRCF<3:0>，可通过软件选择系统时钟速度。16 MHzHFINTOSC 和31 kHz LFINTOSC 的输出连接到后分频器和多路开关（见图5-1）。OSCCON 寄存器的内部振荡器频率选择位IRCF<3:0>用于选择内部振荡器的频率输出。可通过软件选择以下频率中的一个：

- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz （复位后的默认值）
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz （LFINTOSC）

通过OSCCON 寄存器的IRCF<3:0> 位，可以重复选择一些频率。重复选择可以为系统设计提供权衡的空间。对于某个给定的频率，可以通过更改振荡器源来降低功耗。在使用同一振荡器源的情况下改变频率时，可以实现更快的转换速度。

5.2.2.4 内部振荡器时钟切换时序

当在HFINTOSC、和LFINTOSC 之间切换时，新振荡器可能已经关闭以节省功耗（见图5-7）。如果是这种情况，则在修改OSCCON寄存器的 IRCF<3:0>位之后，进行频率选择之前，存在一定的延时。OSCSTAT 寄存器将反映HFINTOSC、和LFINTOSC 振荡器的当前状态。频率选择序列如下：

1. 修改OSCCON 寄存器的IRCF<3:0> 位。
2. 如果新时钟是关闭的，开始时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿出现。
4. 当前时钟保持为低电平，时钟切换电路等待新时钟上升沿出现。
5. 新时钟现在开始工作。
6. OSCSTAT 寄存器按需要进行更新。
7. 时钟切换完成。

更多详细信息，请参见图5-7。

如果内部振荡器速度在同一时钟源的两个时钟之间进行切换，则选取新频率不存在起振延时。表5-1 列出了时钟切换延时。

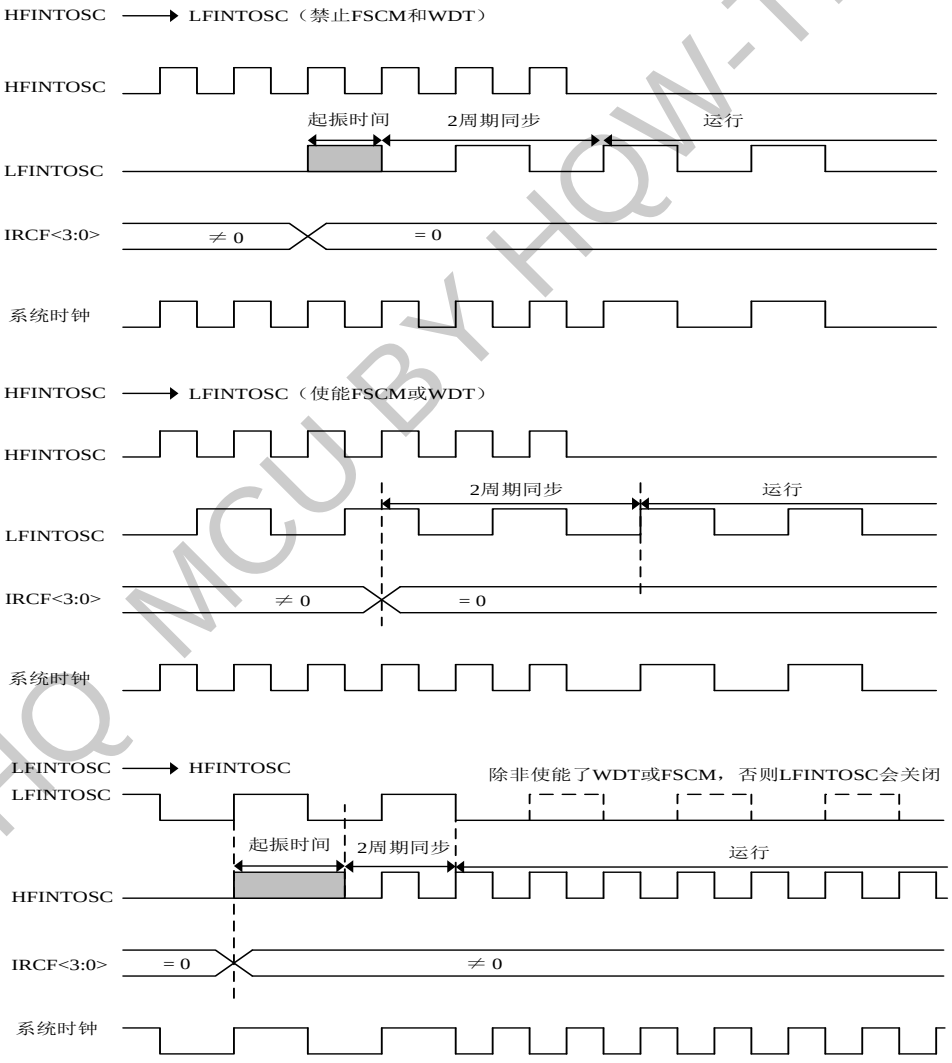


图5-7: 内部振荡器切换时序

5.3 时钟切换

使用OSCCON 寄存器的系统时钟选择（SCS）位，可通过软件在外部和内部时钟源之间切换系统时钟源。使用SCS 位可以选择以下时钟源：

- 由配置字寄存器1 中的FOSC 位决定的默认系统振荡器
- Timer1 32 kHz 晶振
- 内部振荡器模块（INTOSC）

5.3.1 系统时钟选择（SCS）位

OSCCON 寄存器的系统时钟选择（SCS）位选择用于CPU 和外设的系统时钟源。

- 当OSCCON 寄存器的SCS 位= 00 时，系统时钟源由配置字寄存器1 中的FOSC<2:0> 位的值决定。
- 当OSCCON 寄存器的SCS 位= 01 时，系统时钟源为Timer1 振荡器。
- 当OSCCON 寄存器的SCS位= 1x时，系统时钟源由通过OSCCON 寄存器的 IRCF<3:0> 位选择的内部振荡器频率选择。复位之后，OSCCON 寄存器的SCS 位总是被清零。

> 注：任何自动时钟切换（可能由双速启动或故障保护时钟监视器产生）都不会更新OSCCON 寄存器的SCS 位。用户可以监视OSCSTAT 寄存器的OSTS 位，以确定当前的系统时钟源。

当在时钟源之间切换时，需要一定的延时以使新时钟稳定。表5-1 给出了各种振荡器延时。

5.3.2 振荡器起振超时状态（OSTS）位

OSCSTAT寄存器的振荡器起振超时状态（OSTS）位用于指示系统时钟是来自外部时钟源（由配置字寄存器1中的FOSC<2:0> 位定义），还是来自内部时钟源。OSTS还用于特别指示在LP、XT或HS模式下，振荡器起振定时器（OST）是否已超时。OST不会反映Timer1 振荡器的状态。

5.3.3 TIMER1 振荡器就绪（T1OSCR）位

在选择Timer1 振荡器作为系统时钟源之前，用户必须确保它已就绪备用。OSCSTAT 寄存器的Timer1 振荡器就绪（T1OSCR）位指示Timer1 振荡器是否已就绪备用。在T1OSCR 位置1 之后，可以将SCS 位配置为选择Timer1 振荡器。

5.4 双速时钟启动模式

双速启动模式通过最大限度地缩短外部振荡器起振与代码执行之间的延时，进一步节省了功耗。对于频繁使用休眠模式的应用，双速启动模式将从器件唤醒的时间中去除外部振荡器的起振时间，从而可降低器件的总体功耗。该模式使得能够将应用从休眠中唤醒，将INTOSC内部振荡器模块用作时钟源执行数条指令，然后再返回休眠状态而无需等待外部振荡器的稳定。

当振荡器模块被配置为LP、XT 或HS 模式时，双速启动可以带来一些益处。对于这些模式，振荡器起振定时器（OST）会被使能，并且它必须在计数1024 次振荡之后，振荡器才能用作系统时钟源。如果振荡器模块被配置为除LP、XT 或HS 模式以外的任何模式，则双速启动将被禁止。这是因为POR 后或从休眠中退出时，外部时钟振荡器不需要花时间稳定。如果在器件进入休眠模式之前OST 计数到1024，并且OSCSTAT 寄存器的OSTS 位置1，则程序执行切换到外部振荡器。但是，如果唤醒所需的时间极短，系统可能永远不会使用外部振荡器工作。

> 注：执行SLEEP指令将中止振荡器起振时间，并使OSCSTAT 寄存器的OSTS 位保持清零。

表5-1: 振荡器切换延时

切换自	切换到	频率	振荡器延时
休眠/POR	LFINTOSC(1) HFINTOSC(1)	31kHz 31.25kHz-16MHz	振荡器预热延时
休眠/POR	EC或RC	DC-16MHz	2个周期
LFINTOSC	EC或RC	DC-16MHz	每次一周期
休眠/POR	Timer1振荡器 LP、XT或HS	32kHz-16MHz	1024个时钟周期（OST）
任何时钟源	HFINTOSC	31.25kHz-16MHz	2ms（近似值）
任何时钟源	LFINTOSC	31kHz	每次一周期
任何时钟源	Timer1振荡器	32kHz	1024个时钟周期（OST）

5.4.1 双速启动模式配置

通过以下设置来配置双速启动模式：

• IESO（在配置字寄存器1中） = 1；内部/外部切换位 （使能双速启动模式）。

• SCS （在OSCCON 寄存器中） = 00。

• 配置字寄存器1 中的FOSC<2:0> 位被配置为 LP、XT 或HS 模式。

在以下事件之后，进入双速启动模式：

• 上电复位（POR）以及在上电延时定时器（PWRT）延时结束（如果使能）后，或者

• 从休眠中唤醒。

5.4.2 双速启动顺序

1. 从上电复位或休眠中唤醒。
2. 使用内部振荡器以OSCCON寄存器的IRCF<3:0>位设置的频率开始执行指令。
3. OST 使能，计数1024 个时钟周期。
4. OST 超时，等待内部振荡器下降沿出现。
5. OSTS 置1。
6. 系统时钟保持为低电平，直到新时钟下一个下降沿出现（LP、XT 或HS 模式）。
7. 系统时钟切换到外部时钟源

5.4.3 检查双速状态

通过检查OSCSTAT 寄存器的OSTS 位的状态，可以确认单片机是依靠外部时钟源运行（由配置字寄存器1 中的FOSC<2:0> 位定义），还是依靠内部振荡器运行。

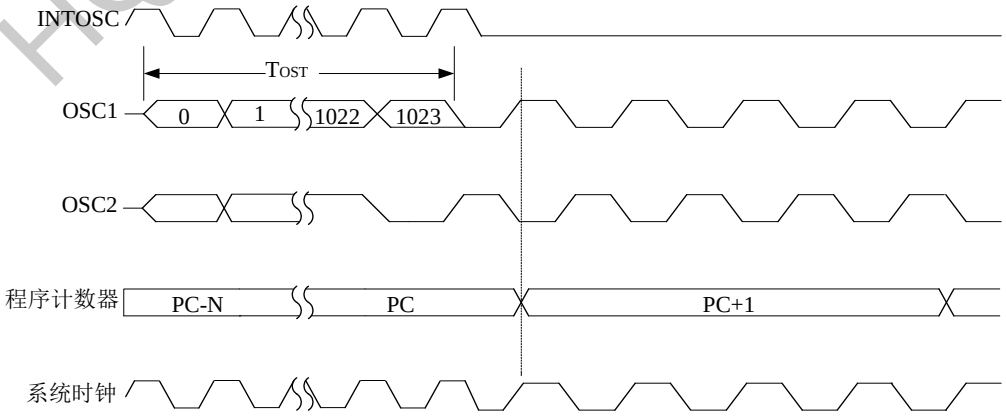


图5-8: 双速启动

5.5 故障保护时钟监视器

故障保护时钟监视器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM 能在振荡器起振定时器（OST）延时结束后的任一时刻检测振荡器故障。FSCM通过将配置字寄存器1 中的FCMEN 位置1 来使能。FSCM 可用于所有外部振荡器模式（LP、XT、HS、EC、Timer1 振荡器和RC）。

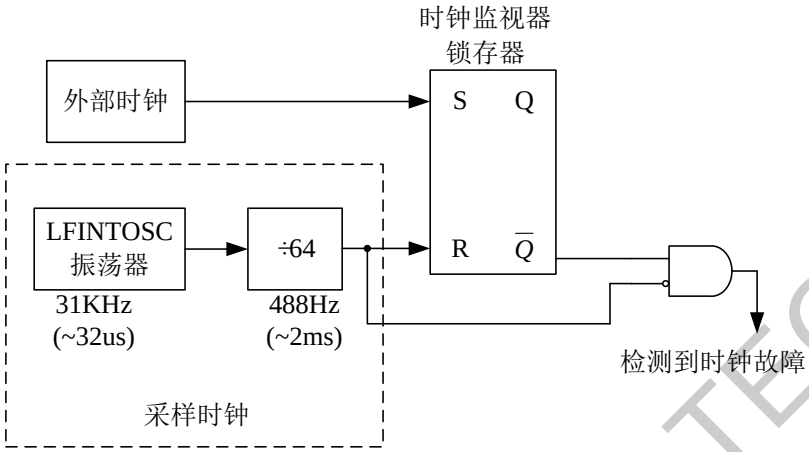


图5-9: FSCM框图

5.5.1 故障保护检测

FSCM模块通过将外部振荡器与FSCM采样时钟比较来检测振荡器故障。将LFINTOSC 64 分频，就产生了采样时钟。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，外部时钟将锁存器置1。在采样时钟的每个上升沿，采样时钟将锁存器清零。如果已经经过采样时钟的整个半周期，但外部时钟仍未变为低电平，则会检测到故障。

5.5.2 故障保护操作

当外部时钟出现故障时，FSCM 将器件时钟切换到内部时钟源，并将PIFB2 寄存器的OSFIF 标志位置1。如果PIEB2 寄存器的OSFIE 位也置1，则OSFIF 标志置1会产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续由内部时钟源提供，直到器件固件成功重启外部振荡器并切换回外部振荡器进行工作。FSCM选定的内部时钟源由OSCCON寄存器的IRCF<3:0>位决定。这使得可以在故障发生前配置内部振荡器。

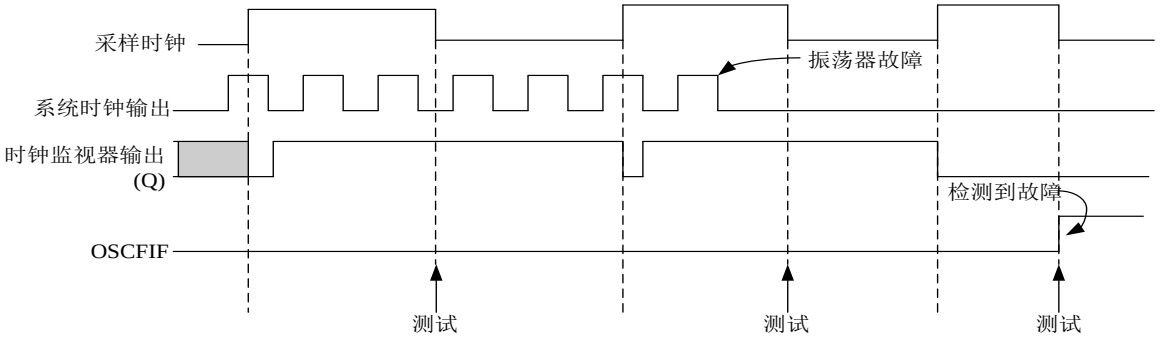
5.5.3 故障保护条件清除

在复位、执行SLEEP 指令或更改OSCCON寄存器的SCS 位之后，故障保护条件被清除。SCS 位被更改后，OST 将重新启动。OST 运行时，器件将依靠OSCCON中选定的INTOSC 工作。OST 超时后，在器件成功切换到外部时钟源后，故障保护条件被清除。 在器件切换到外部时钟源前，OSFIF 位应清零。 如果仍存在故障保护条件，那么硬件会再次将OSFIF 标志置1。

5.5.4 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振定时器（OST）延时结束后的任一时刻检测振荡器故障。从休眠状态唤醒后以及任何类型的复位后使用OST。OST 不能在EC 或RC 时钟模式下使用，所以一旦复位或唤醒完成，FSCM 就处于活动状态。当FSCM 被使能时，双速启动也被使能。因此，当OST 运行时，器件总是处于代码执行阶段。

> 注：由于振荡器起振时间范围较大，在振荡器起振期间（即，从复位或休眠中退出后），故障保护电路不处于活动状态。经过一段适当的时间后，用户应检查OSCSTAT寄存器中的状态位，以验证振荡器是否已成功起振以及系统时钟是否切换成功。



注：系统时钟频率通常远高于采样时钟频率。本例选用的相对频率仅是为了便于说明

图5-10: FSCM 时序图

5.6 寄存器说明

寄存器99H：振荡器控制寄存器（OSCCON）

U-0	R/W-0	R/W-1	R/W-1	R/W-1	U-0	R/W-0	R/W-0
—	**IRCF3**	**IRCF2**	**IRCF1**	**IRCF0**	—	**SCS1**	**SCS0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **未实现：**写为0

bit6-3 **IRCF<3:0>：**内部振荡器频率选择位
000x = 31 kHz LF
0010 = 31.25 kHz
0011 = 31.25 kHz
0100 = 62.5 kHz
0101 = 125 kHz
0110 = 250 kHz
0111 = 500 kHz （复位时的默认值）
1000 = 125 kHz
1001 = 250 kHz
1010 = 500 kHz
1011 = 1 MHz
1100 = 2 MHz
1101 = 4 MHz
1110 = 8 MHz
1111 = 16 MHz

bit2 **未实现：读**为0

bit1-0 **SCS:** 系统时钟选择位
1x = 内部振荡器模块
01 = Timer1 振荡器
00 = 由配置字寄存器1 中FOSC<2:0> 决定的时钟

寄存器9AH：振荡器状态寄存器（OSCSTAT）

R-1/q	R-0	R-q/q	R-0/q	R-0/q	R-q/q	R-0/0	R-0/q
T1OSCR	—	**OSTS**	**HFIOFR**	**HFIOFL**	—	**LFIOFR**	**HFIOFS**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	x= 未知
-n=POR时的值	1= 置1	0= 清零	q= 条件值

bit7 **T1OSCR**：Timer1就绪位
如果T1OSCEN = 1：
1 = Timer1 振荡器就绪
0 = Timer1 振荡器未就绪
如果T1OSCEN = 0：
1 = Timer1 时钟源始终就绪

bit6 未实现，读为0

bit5 **OSTS**：振荡器起振延时状态位
1 = 依靠由配置字寄存器1 中FOSC<2:0> 位定义的时钟运行
0 = 依靠内部振荡器（FOSC<2:0> = 100）运行

bit4 **HFIOFR**：高频内部振荡器就绪位
1 = HFINTOSC 就绪
0 = HFINTOSC 未就绪

bit3 **HFIOFL**：高频内部振荡器锁定位
1 = HFINTOSC 的精度至少在2% 以内
0 = HFINTOSC 的精度在2% 以外

bit2 **保留位**

bit1 **LFIOFR**：低频内部振荡器就绪位
1 = LFINTOSC 就绪
0 = LFINTOSC 未就绪

bit0 **HFIOFS**: 高频内部振荡器稳定位
1 = HFINTOSC 的精度至少在0.5% 以内
0 = HFINTOSC 的精度在0.5% 以外

表 5-2：与时钟源相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
OSCCON	—	IRCF3	IRCF2	IRCF1	IRCF0	—	SCS1	SCS0	0011 1xxx	0011 1xxx
OSCSTAT	T1OSCR	—	OSTS	HFIOFR	HFIOFL	—	LFIOFR	HFIOFS	--011111	xxxx xxxx
PIEB2	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	—	0000 0---	xxxx xxxx
PIFB2	OSFIF	C2IF	C1IF	EEIF	BCL1IF	—	—	—	0000 0xxx	xxxx xxxx
T1STA	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	0000 0000	xxxx xxxx

图注： x = 未知， u = 不变， — =未实现（读为 0）。不使用阴影单元。

6.0 参考时钟模块

参考时钟模块可将返聘后的时钟发送到器件的时钟输出引脚（CLKR）,并为调制器模块提供辅助内部时钟源。该模块在所有振荡器配置中都可用，允许用户选择更大范围的时钟分频比来驱动应用中的外部器件。参考时钟模块具有以下特性：

- 系统时钟源作为时钟源。
- 在所有振荡器配置中都可用。
- 可编程的时钟分频比。
- 端口引脚输出使能。
- 可选的占空比。
- 压摆率控制。

参考时钟模块通过CLKRCON寄存器进行控制，并在CLKREN置一时使能。要将分频后的时钟信号输出到CLKR端口引脚，CLKROE位必须置一。CLKRDIV<2:0>位允许选择8种不同时钟的分频比选项。CLKRDC<1:0>位可用于修改输出时钟的占空比。CLKRSLR位用于控制压摆率控制。

注 1：如果不使用分配器，而是选择了基本时钟速率，则除非选择的占空比为0%，否则输出时钟的占空比将重视等于源时钟的占空比。如果时钟分频比设置为基本时钟/2，则25%和75%占空比的精度将取决于源时钟。

6.1 压摆率

可以禁止对于输出端口引脚的压摆率限制。压摆率限制可以通过将CLKRCON寄存器中的CLKRSLR位清零而取消。

6.2 复位影响

在发生任何器件复位时，参考时钟模块都会被禁止。用户的固件负责在使能输出之前初始化模块，寄存器会复位到默认值。

6.3 与 CLKR 引脚冲突

在一下两种情况下，参考时钟输出信号将无法输出到CLKR引脚：

- 选择了LP、XT和HS振荡器模式。
- 使能了CLKOUT功能。

即使以上任一情况为真，任然可以使能模块，参考时钟信号可以与调制器模块配合使用。

6.3.1 CLKOUT 功能

CLKOUT功能的优先级高于参考时钟模块，因此配置字寄存器1中的CLKOUTEN位使能了CLKOUT功能，则在端口引脚上将总是输出FOSC/4。

6.4 寄存器说明

寄存器39AH：参考时钟控制寄存器（CLKRCON）

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
CLKREN	**CLKROE**	**CLKRSLR**	**CLKRDC1**	**CLKRDC0**	**CLKRDIV2**	**CLKRDIV1**	**CLKRDIV0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **CLKREN**：参考时钟模块使能位
1 = 使能参考时钟模块
0 = 禁止参考时钟模块

bit6 **CLKROE**：参考时钟输出使能位(3)
1 = 在CLKR引脚使能参考时钟输出
0 = 在CLKR引脚禁止参考时钟输出

bit5 **CLKRSLR**：参考时钟压摆率控制限制使能位
1 = 使能压摆率限制
0 = 禁止压摆率限制

bit4-3 **CLKRDC <1:0>**：参考时钟占空比
11 = 时钟输出占空比为75%
10 = 时钟输出占空比为50%
01 = 时钟输出占空比为25%
00 = 时钟输出占空比为0%

bit2-0 **CLKRDIV <2:0>**：参考时钟分频比
111 = 基本时钟值被128分频
110 = 基本时钟值被64分频
101 = 基本时钟值被32分频
100 = 基本时钟值被16分频
011 = 基本时钟值被8分频
010 = 基本时钟值被4分频
001 = 基本时钟值被2分频(1)
000 = 基本时钟值(2)

注
1：在该模式下，25%和75%占空比的精度将取决于源时钟的占空比。
2：在该模式下，除非选择的占空比为0%，否则占空比将总是等于源时钟的占空比。
3：要将CLKR送到引脚上，必须使配置字1的CLKOUTEN = 1。配置字1的CLKOUTEN = 0时，将产生FOSC/4。

表 6-1：与参考时钟相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
CLKRCON	CLKREN	CLKROE	CLKRSLR	CLKRDC1	CLKRDC0	CLKRDIV2	CLKRDIV1	CLKRDIV0	xxxx00XX	uuuuuuxx

图注： x = 未知， u = 不变， — =未实现（读为 0）。不使用阴影单元。

7.0 复位

该器件的复位有几种方式：

- 上电复位（POR）
- 欠压复位（BOR）
- MCLR复位
- WDT复位
- RESET 指令
- 堆栈上溢
- 堆栈下溢
- 编程模式退出

要使VDD 稳定下来，可以使能可选的上电延时定时器来延长BOR 或POR 事件之后的复位时间。

7.1 上电复位（POR）

POR电路会将器件一直保持在复位状态，直到VDD 达到最低工作条件可接受的电平为止。在VDD 上升缓慢、高速运行或要求一定模拟性能时，所需的电压可能高于最低VDD。可以使用PWRT、BOR 或MCLR 功能来延长启动周期，直到满足所有器件工作条件为止。

7.1.1 上电延时定时器（PWRT）

上电延时定时器在POR 或欠压复位时提供一个64 ms标称值的延时。只要PWRT 处于活动状态，器件就保持在复位状态。PWRT 延时使VDD 有额外的时间上升到所需的电平。通过清零配置字寄存器1 中的PWRTE位使能上电延时定时器。上电延时定时器会在POR 和BOR 释放之后启动。

7.2 欠压复位（BOR）

当VDD 达到可选的最低电平时， BOR 电路会将器件保持在复位状态。在POR 和BOR 之间，可在整个电压范围内对器件的执行进行保护。欠压复位模块具有4 种工作模式，它们由配置字寄存器1中的BOREN<1:0>位控制。这4 种工作模式是：

- BOR总是开启
- BOR在休眠模式下关闭
- BOR通过软件进行控制
- BOR总是关闭

对配置字寄存器2中的BORV位进行配置来选择欠压复位电平。VDD 噪声抑制滤波器可以防止BOR 在发生轻微事件时产生触发。如果VDD 降至低于VBOR 的时间大于参数TBORDC，器件将会发生复位。

BOR工作模式

BOREN配置位	SBOREN	器件模式	BOR模式	POR释放时的器件操作	从休眠模式唤醒时的器件操作
BOR_ON（11）	X	X	有效	等待BOR就绪 (1)	
BOR_NSLEEP（10）	X	唤醒	有效	等待BOR就绪	
BOR_NSLEEP（10）	X	休眠	禁止		
BOR_SBOREN（01）	1	X	有效	立即开始	
BOR_SBOREN（01）	0	X	禁止	立即开始	
BOR_OFF（00）	X	X	禁止	立即开始	

注1： 在这两种特殊情况（“POR释放”和“从休眠模式中唤醒”）下，不会延迟起振。因为BOR电路由BOREN<1:0>位来强制启动，所以在CPU准备执行指令之前，BOR就绪标志会置1（BORRDY = 1）。

7.2.1 BOR 总是开启

当配置字寄存器1 的BOREN 位设置为11 时，BOR 将总是开启。器件启动会被延迟，直到BOR 就绪，且VDD 高于BOR 阈值为止。BOR保护在休眠期间有效。BOR不会延迟从休眠中唤醒。

7.2.2 BOR 在休眠模式下关闭

当配置字寄存器1的BOREN位设置为10时，除非处于休眠模式，否则BOR将开启。器件启动会被延迟，直到BOR就绪，且VDD高于BOR阈值为止。BOR 保护在休眠期间无效。器件唤醒会被延迟，直到BOR就绪为止。

7.2.3 通过软件对 BOR 控制

当配置字寄存器1的BOREN位设置为01时， BOR将通过BORCON寄存器的SBOREN位进行控制。器件启动不会受BOR就绪条件或VDD电平条件影响而延迟。BOR保护会在BOR电路就绪时立即开始。BOR电路的状态在BORCON寄存器的BORRDY位中反映。BOR保护在休眠期间不变

7.3 外部引脚复位（MCLR）

MCLR是器件可选的外部复位引脚，MCLR功能由配置字寄存器1的MCLRE位控制。当使能MCLR并且引脚保持低电平时，器件保持复位状态。MCLR引脚通过内部弱上拉与VDD连接。当MCLR功能禁止时，引脚将作为通用输入，内部弱上拉由软件控制。

7.4 看门狗定时器复位（WDT）

如果固件未在超时周期内发出CLRWDT 指令，看门狗定时器会产生复位。STATUS 寄存器中的TO 和PD 位会改变，指示发生WDT 复位。

7.5 RESET 指令

RESET 指令会引起器件复位。PSTA 寄存器中RI位将设置为0。

7.6 堆栈上溢/下溢复位

器件可以在堆栈上溢或下溢时复位。PSTA 寄存器的STKOVF或STKUNF位用于指示复位条件。这些复位通过将配置字寄存器2 中的STVREN 位置1 来使能。

7.7 上电延时定时器

上电延时定时器可用于在BOR 或POR 事件之后延迟器件执行。该定时器通常用于使VDD 在允许器件开始运行之前先稳定下来。上电延时定时器由配置字寄存器1的PWRTE 位控制。

7.8 启动序列

在POR 或BOR 释放时，只有先发生以下事件，器件才会开始执行：

1、上电延时定时器运行完毕（如果使能）。

2、振荡器起振定时器运行完毕（如果对于振荡器需要）。

3、MCLR引脚必须被释放（如果使能）。

上电延时定时器和振荡器起振定时器的运行与MCLR复位无关。如果MCLR保持低电平时间足够长，上电延时定时器和振荡器起振定时器将会延时结束。

7.9 电源控制寄存器（PSTA）

电源控制寄存器包含区分一下各种复位标志：

- 上电复位（POR）
- 欠压复位（BOR）
- RESET指令复位
- 堆栈上溢复位
- 堆栈下溢复位
- 外部引脚复位（MCLR）

7.10 寄存器说明

寄存器116H：欠压复位控制寄存器（BORCON）

R/W-1	U-0	U-0	U-0	U-0	U-0	U-0	R-u
SBOREN	—	—	—	—	—	—	**BORRDY**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **SBOREN**：软件欠压复位使能位
如果配置字BOREN<1:0>≠ 01：
SBOREN 可读/写，但对BOR没有任何作用
如果配置字BOREN<1:0>＝ 01：
1 = 使能BOR
0 = 禁止BOR

bit6-1 **未实现**

bit0 **BORRDY:** 欠压复位电路就绪状态位
1 = 欠压复位电路有效
0 = 欠压复位电路无效

寄存器96H：电源控制寄存器（PSTA）

R/W/HS-0/q	R/W/HS-0/q	U-0	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q
STKOVF	**STKUNF**	—	—	**RMCLR**	**RI**	**POR**	**BOR**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
HS=硬件置一	1= 置1	0= 清零	HC=硬件清零位

bit7 **STKOVF**：堆栈上溢标志位
1 = 发生了堆栈上溢
0 = 未发生堆栈上溢或由固件设置为0

bit6 **STKUNF**：堆栈下溢标志位
1 = 发生了堆栈下溢
0 = 未发生堆栈下溢或由固件设置为0

bit5-4 **未实现**

bit3 **RMCLR**：MCLR复位标志位
1 = 未发生MCLR复位或由硬件设置为1

0 = 发生MCLR复位，（当发生MCLR复位时，由硬件设置为0）

bit2 **RI：** RESET指令位

1 = 未执行RESET 指令或由固件设置为1

0 = 执行了RESET 指令（当执行RESET 指令时，由硬件设置为0）

bit1 **POR：** 上电复位状态位

1 = 未发生上电复位

0 = 发生了上电复位（发生上电复位后必须用软件置1）

bit0 **BOR:** 欠压复位状态位

1 = 未发生欠压复位

0 = 发生了欠压复位（发生上电复位或欠压复位后必须用软件置1）

表 7-1：与复位有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	1xxx xxxq	xxxx xxxx
PSTA	STKOVF	STKUNF	—	—	RMCLR	RI	POR	BOR	0000 11xx	xxxx xxxx
STATUS	—	—	—	TO	PD	Z	HC	C	0001 100x	xxxq quuu

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。 BOR 不使用阴影单元。

注 1： 其他（非上电）复位包括正常工作时的$\overline{\text{MCLR}}$复位和看门狗定时器复位。

8.0 中断

通过中断功能，一些事件可以抢占正常的程序流。固件用于确定中断源，并执行相应的操作。有些中断可配置为将MCU 从休眠模式唤醒。本章包含了关于中断的以下信息：

- 工作原理
- 中断延时
- 休眠期间的中断
- INT引脚
- 自动现场保护

图8-1和图8-2给出了中断逻辑框图

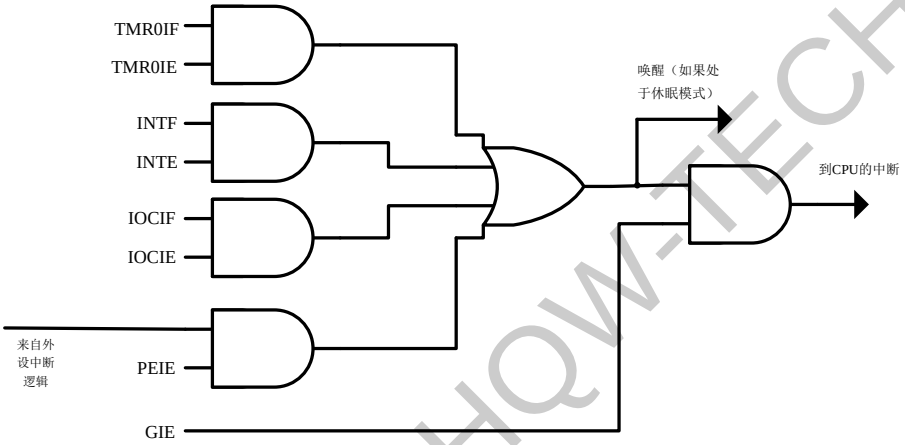


图8-1：中断逻辑

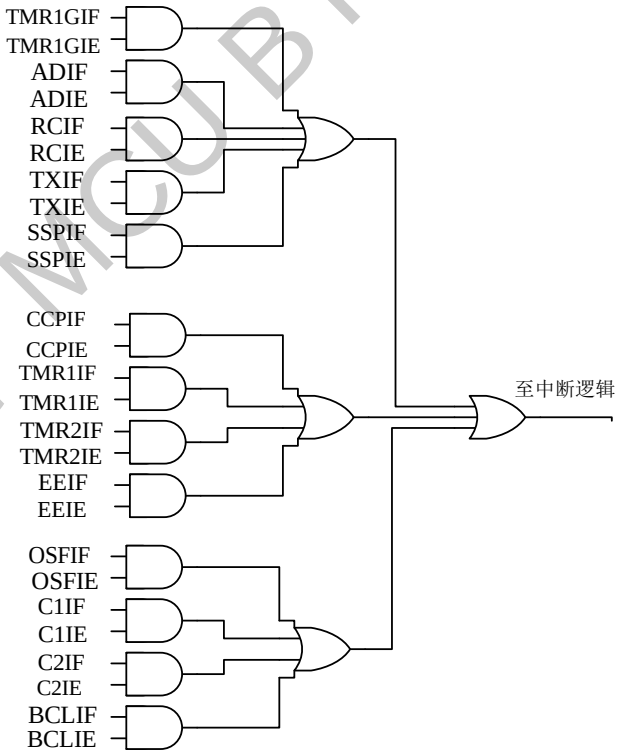


图8-2：外设中断逻辑

8.1 工作原理

任何器件复位时都会禁止中断。通过将以下位置1 允许相应中断：

- INTS寄存器的GIE 位
- 特定中断事件的中断允许位
- INTS寄存器的PEIE 位（如果中断事件的中断允许位包含在PIEB1 或PIEB2 寄存器中）

INTS、PIFB1 和PIFB2 寄存器通过中断标志位记录各个中断。无论GIE、PEIE 和各个中断允许位的状态如何，中断标志位都会在中断发生时置1。当中断事件发生时，若GIE 位置1，将发生以下事件：

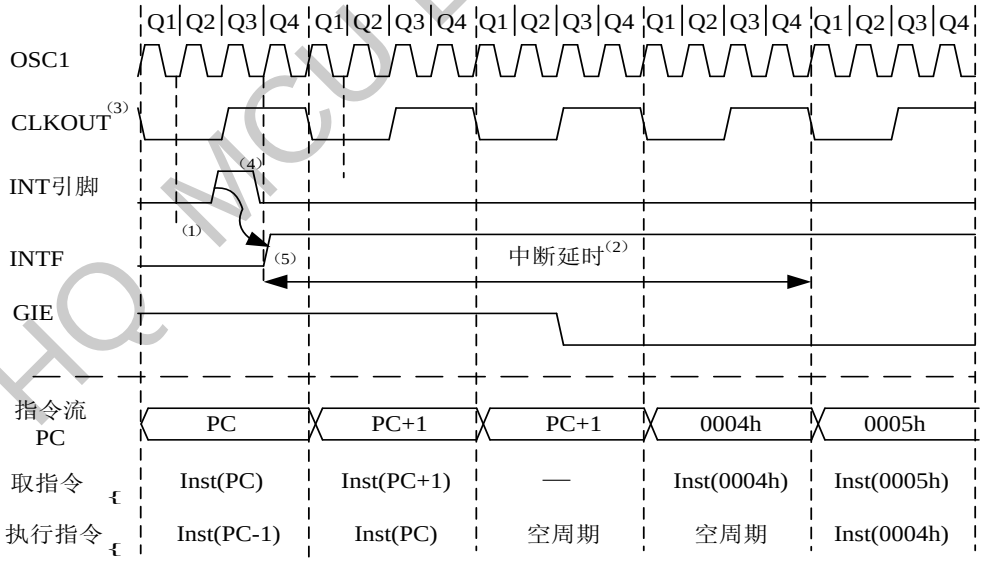
- 清除当前的预取指令
- GIE位清零
- 程序计数器（PC）的当前值压入堆栈
- 自动将关键寄存器保存到影子寄存器中
- 将中断向量0004h 装入PC

中断服务程序（Interrupt Service Routine， ISR）中的固件应通过查询中断标志位来确定中断源。退出ISR前必须清零中断标志位，以避免重复的中断。由于GIE 位清零，执行ISR 期间发生的任何中断都会通过其中断标志位记录下来，但不会使处理器重定位到中断向量。通过从堆栈弹出先前保存的地址、从影子寄存器恢复保存的现场数据并将GIE 位置1，RTFI 指令退出ISR。

注：1： 无论中断允许位状态如何，各中断标志位都会在中断发生时置1。
2：GIE 位清零时，将忽略所有中断。GIE 位清零期间发生的任何中断，都会在GIE 位再次置1 时得到处理。

8.2 中断延时

中断延时定义为从发生中断事件到开始执行中断向量处代码经过的时间。同步中断的延时为3 或4 个指令周期。对于异步中断，延时为3 至5 个指令周期，这取决于中断何时发生。



注 1：在此时采样INTF标志（每个Q1周期）
2：异步中断延时为3-5TCY。同步中断延时为3-4TCY，其中TCY为一个指令周期时间。无论Inst(PC)是单 周期指令还是双周期指令，中断延时都是一样的。
3：CLKOUT不适用于所以振荡器模式
4：关于INT脉冲的最小宽度，请参见第29.0节“电气规范”中的交流规范
5：允许在Q1-Q4周期内的任意时刻将INTF置1

图8-3：INT引脚中断时序

8.3 休眠期间的中断

有些中断可用于将器件从休眠模式唤醒。要从休眠模式唤醒器件，外设必须能在没有系统时钟的情况下工作。进入休眠模式前，必须将相应中断源的中断允许位置1。从休眠模式唤醒时，如果GIE 位也置1，则处理器将跳转到中断向量。否则，处理器将继续执行SLEEP 指令后的指令。紧接SLEEP 指令后的指令总是会在跳转到ISR前执行。

8.4 INT 引脚

INT 引脚可用于产生异步边沿触发中断。可以通过将INTS寄存器的INTE位置1 来允许该中断。OPTION寄存器的INTEDG 位决定中断在哪个边沿发生。INTEDG 位置1 时，上升沿将引起中断。INTEDG 位清零时，下降沿将引起中断。INTS 寄存器的INTF 位将在INT 引脚上出现有效边沿时置1。如果GIE 和INTE位也置1，则处理器会将程序执行重定位到中断向量。

8.5 自动现场保护

进入中断时PC的返回地址被保存在堆栈中。此外，以下寄存器会被自动保存到影子寄存器中：

- W寄存器
- STATUS寄存器（TO 和PD 除外）
- BSR 寄存器
- MSR寄存器
- PCLATH寄存器

在退出中断服务程序时，将会自动恢复这些寄存器。在ISR 期间对这些寄存器进行的任何修改都会丢失。如果需要修改其中的任意寄存器，则应修改相应的影子寄存器，该值在退出ISR 时将会被恢复。影子寄存器位于Bank 31 中，它们是可读写寄存器。根据用户的应用，可能还需要保存其他寄存器。

8.6 寄存器说明

寄存器0BH：中断控制寄存器（INTS）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	**PEIE**	**TMR0IE**	**INTE**	**PAIE**	**TMR0IF**	**INTF**	**PAIF**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **GIE**：全局中断允许位
1 = 允许所有有效中断
0 = 禁止所有中断

bit6 **PEIE**：外设中断允许位
1 = 允许所有有效外设中断
0 = 禁止所有外设中断

bit5 **TMR0IE**：Timer0 上溢中断允许位
1 = 允许Timer0 中断
0 = 禁止Timer0 中断

bit4 **INTE**：INT 外部中断允许位
1 = 允许INT 外部中断
0 = 禁止INT 外部中断

bit3 **PAIE**：**PORTA**电平变化中断允许位

1 = 允许电平变化中断
0 = 禁止电平变化中断

bit2 **TMR0IF**：Timer0 上溢中断标志位
1 = TMR0 寄存器已上溢
0 = TMR0 寄存器未上溢

bit1 **INTF**：INT 外部中断标志位
1 = 发生了INT 外部中断
0 = 未发生INT 外部中断

bit0 **PAIF**：电平变化中断标志位(1)
1 = 至少有一个电平变化中断引脚改变了状态
0 = 没有任何电平变化中断引脚的状态发生改变

注 **1**：PAIF 标志位为只读，且当IOCAF 寄存器中的所有电平变化中断标志由软件清零后，PAIF 标志位被清零。

寄存器91H：外设中断允许寄存器1（PIEB1）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1GIE	**ADIE**	**RCIE**	**TXIE**	**SSP1IE**	**CCP1IE**	**TMR2IE**	**TMR1IE**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **TMR1GIE**：Timer1 门控中断允许位
1 = 允许Timer1 门控采集中断
0 = 禁止Timer1 门控采集中断

bit6 **ADIE**：A/D 转换器（ADC）中断允许位
1 = 允许ADC 中断
0 = 禁止ADC 中断

bit5 **RCIE**：USART 接收中断允许位
1 = 允许USART 接收中断
0 = 禁止USART 接收中断

bit4 **TXIE**：USART 发送中断允许位
1 = 允许USART 发送中断
0 = 禁止USART 发送中断

bit3 **SSP1IE**：同步串行口（MSSP）中断允许位
1 = 允许MSSP 中断
0 = 禁止MSSP 中断

bit2 **CCP1IE**：CCP1 中断允许位
1 = 允许CCP1 中断
0 = 禁止CCP1 中断

bit1 **TMR2IE**：TMR2 与PR2 匹配中断允许位
1 = 允许Timer2 与PR2 匹配中断
0 = 禁止Timer2 与PR2 匹配中断

bit0 **TMR1IE:** Timer1 上溢中断允许位
1 = 允许Timer1 上溢中断
0 = 禁止Timer1 上溢中断

寄存器92H：外设中断允许寄存器2（PIEB2）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
OSFIE	**C2IE**	**C1IE**	**EEIE**	**BCLIE**	—	—	—
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **OSFIE**：振荡器故障中断允许位
1 = 允许振荡器故障中断
0 = 禁止振荡器故障中断

bit6 **C2IE**：比较器C2 中断允许位
1 = 允许比较器C2 中断
0 = 禁止比较器C2 中断

bit5 **C1IE**：比较器C1 中断允许位
1 = 允许比较器C1 中断
0 = 禁止比较器C1 中断

bit4 **EEIE**：EEPROM 写操作完成中断允许位
1 = 允许EEPROM 写操作完成中断
0 = 禁止EEPROM 写操作完成中断

bit3 **BCLIE**：MSSP 总线冲突中断允许位
1 = 允许MSSP 中断
0 = 禁止MSSP 中断

bit2-0 **未实现**

寄存器11H：外设中断请求寄存器1（PIFB1）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1GIF	**ADIF**	**RCIF**	**TXIF**	**SSP1IF**	**CCP1IF**	**TMR2IF**	**TMR1IF**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **TMR1GIF**：Timer1 门控中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit6 **ADIF**：A/D 转换器（ADC）中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit5 **RCIF**：USART 接收中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit4 **TXIF**：USART 发送中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit3 **SSP1IF**：同步串行口（MSSP）中断标志位
1 = 允许MSSP 中断

0 = 禁止MSSP 中断

bit2 **CCP1IF**：CCP1 中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit1 **TMR2IF**：TMR2 与PR2 匹配中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit0 **TMR1IF:** Timer1 上溢中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

寄存器12H：外设中断请求寄存器2（PIFB2）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
OSFIF	**C2IF**	**C1IF**	**EEIF**	**BCLIF**	**—**	**—**	**—**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **OSFIF**：振荡器故障中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit6 **C2IF**：比较器C2 中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit5 **C1IF**：比较器C1 中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit4 **EEIF**：EEPROM 写操作完成中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit3 **BCLIF**：MSSP 总线冲突中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit2-0 未实现

表 8-1：与中断有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TMROIE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x	xxxx xxxq
OPTION	WPUEN	INTEDG	TMROCS	TMR0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIEB2	OSFIE	C2IE	C1IE	EEIE	BCLIE	—	—	—	0000 0xxx	0000 0xxx
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIFB2	OSFIF	C2IF	CSIF	EEIF	BCLIF	—	—	—	0000 0000	0000 0xxx

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

9.0 掉电模式

通过执行SLEEP 指令可进入掉电模式。在进入休眠模式时，会存在以下条件：

1. 如果在休眠期间使能WDT，则WDT 会清零，但保持运行。
2. STATUS寄存器的PD 位被清零。
3. STATUS寄存器的TO 位被置1。
4. CPU 时钟被禁止。
5. 31 kHz LFINTOSC 不受影响，使用它工作的外设可以在休眠模式下继续工作。
6. Timer1 振荡器不受影响，使用它工作的外设可以在休眠模式下继续工作。
7. 如果选择了专用FRC 时钟，则ADC 不受影响。
8. 电容传感振荡器不受影响。
9. I/O 端口保持执行SLEEP 指令之前的状态（驱动为高电平、低电平或高阻态）。
10. WDT 之外的其他复位都不会受休眠模式影响。

要最大程度降低电流消耗，应考虑以下条件：

- I/O引脚不应悬空
- 来自I/O 引脚的外部电路灌电流
- 来自I/O 引脚的内部电路拉电流
- 从带内部弱上拉的引脚汲取的电流
- 使用31 kHz LFINTOSC 的模块
- 使用Timer1 振荡器的模块

为了避免输入引脚悬空而引入开关电流，应在外部将高阻抗输入的I/O 引脚拉为 VDD 或VSS。

可能产生拉电流的内部电路示例包括诸如DAC 和FVR之类的模块。关于这些模块的更多信息，请参见第17.0节“ 数模转换器（DAC）模块”和第14.0 节“ 固定参考电压（FVR） ”。

9.1 从休眠状态唤醒

发生以下任一事件将器件从休眠状态唤醒：

1. MCLR 引脚上的外部复位输入（如果使能）
2. BOR 复位（如果使能）
3. POR 复位
4. 看门狗定时器（如果使能）
5. 任何外部中断
6. 由可以在休眠期间运行的外设所产生的中断（更多信息，请参见各个外设）

前三个事件会导致器件复位。后三个事件被认为是程序执行的继续。要确定是发生了器件复位还是唤醒事件。 当执行SLEEP 指令时，下一条指令（PC + 1）被预先取出。如果希望通过中断事件唤醒器件，则必须允许相应的中断允许位。唤醒与GIE 位的状态无关。如果GIE 位被禁止，器件将继续执行SLEEP 指令之后的指令。如果GIE位被允许，器件将执行SLEEP指令之后的指令，器件将调用中断服务程序。如果不希望执行SLEEP指令之后的指令，用户应在SLEEP 指令后面放置一条NOP 指令。器件从休眠状态唤醒时， WDT 都将被清零，而与唤醒原因无关。

9.1.1 使用中断唤醒

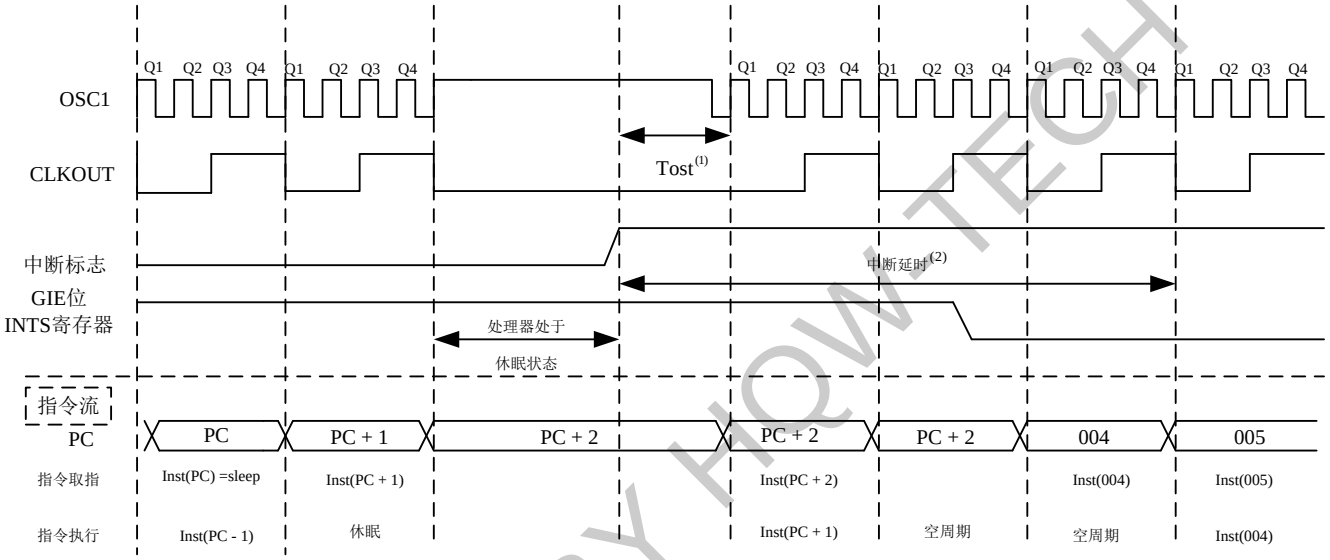
当禁止全局中断（GIE 被清零）时，并且任一中断源的中断允许位和中断标志位都置1，将会发生以下事件之一：

- 如果在执行SLEEP 指令之前发生中断
  - SLEEP 指令将作为NOP 指令执行
  - WDT和WDT 预分频器不会被清零
  - STATUS寄存器的TO 位不会被置1

- STATUS寄存器的PD 位不会被清零

• 如果在执行SLEEP 指令期间或之后发生中断
- 将完整执行SLEEP 指令
- 器件将立即从休眠状态唤醒
- WDT和WDT 预分频器将被清零
- STATUS寄存器的TO 位将被置1
- STATUS寄存器的PD 位将被清零

即使在执行SLEEP 指令之前检查到标志位为0，这些标志位也有可能在SLEEP 指令执行完毕之前被置1。要确定是否执行了SLEEP 指令，可测试PD 位。如果PD 位置1，则说明SLEEP 指令被当作一条NOP 指令执行了。



注：1、Tost = 1024 Tosc （图中未按比例绘制）。该延时仅适用于XT、HS或LP振荡器模式
2、中断延时在GIE = 1处理器唤醒后，将调用004H的ISR程序，如果GIE=0时，程序将继续执行。

图9-1：通过中断从休眠状态唤醒

表 9-1：与掉电模式有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TMROIE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x	0000 000q
OPTION	WPUEN	INTEDG	TMROCS	TMR0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIEB2	OSFIE	C2IE	C1IE	EEIE	BCLIE	—	—	—	0000 0xxx	0000 0xxx
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIFB2	OSFIF	C2IF	CSIF	EEIF	BCLIF	—	—	—	0000 0xxx	0000 0xxx
STATUS	—	—	—	TO	PD	Z	HC	C	---1 1000	---1 1000
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000	--00 0000
IOCP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
WDTCON	—	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	--00 0000	--00 0000

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

10.0 看门狗定时器

看门狗定时器是一个系统定时器，以31 kHz LFINTOSC内部振荡器作为其工作的时基。如果固件未在超时周期内发出CLRWDT指令，看门狗定时器会产生复位。看门狗定时器通常用于使系统从意外事件中恢复。

WDT 具有以下特性：

- 独立时钟源
- 多种工作模式
  - WDT总是开启
  - WDT在休眠模式下关闭
  - WDT通过软件进行控制
  - WDT总是关闭
- 超时周期可配置为从1 ms 至256 秒（典型值）
- 多种复位条件
- 休眠期间的操作

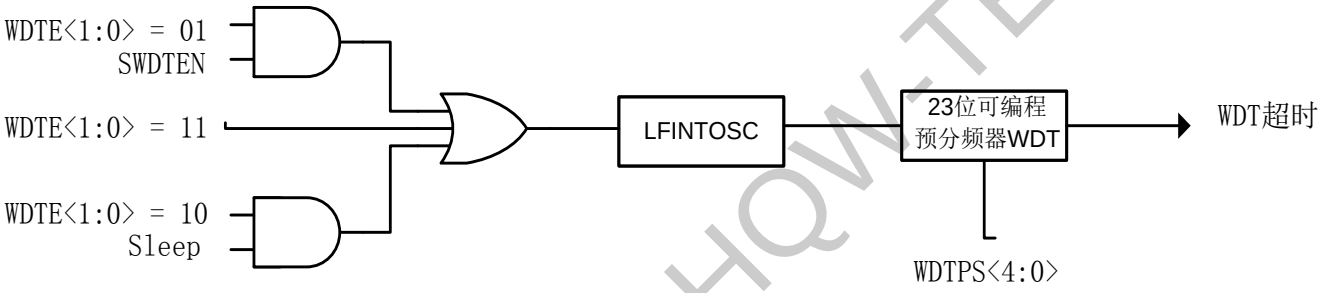


图 10-1： 看门狗定时器框图

10.1 WDT 工作模式

看门狗定时器模块具有4 种工作模式，这些工作模式由配置字寄存器1 中的WDTE<1:0> 位控制。请参见下表10-1。当配置字寄存器1的WDTE位设置为10时，除非处于休眠模式，否则WDT将开启，WDT保护在休眠期间无效；当配置字寄存器1的WDTE 位设置为11，WDT 将总是开启，WDT 保护在休眠期间有效；当配置字寄存器1的WDTE 位设置为01 时， WDT 将通过WDTCON寄存器的SWDTEN 位进行控制，WDT 保护在休眠期间不变。

表 10-1： WDT 工作模式

WDTE配置位	SWDTEN	器件模式	WDT模式
WDT_ON(11)	x	X	有效
WDT_NSLEEP(10)	x	唤醒	有效
WDT_NSLEEP(10)	x	休眠	禁止
WDT_SWDTEN(01)	1	X	有效
WDT_SWDTEN(01)	0	X	禁止

10.2 清零 WDT

当发生以下任何条件时， WDT 被清零：

- 任何复位
- 执行了CLRWDT 指令
- 器件进入休眠模式
- 器件从休眠状态唤醒
- 振荡器故障事件
- WDT被禁止
- OST正在运行

表 10-2： WDT清零条件

条件	WDT
WDTE<1:0> = 00	清零
WDTE<1:0> = 01 且SWDTEN = 0	
WDTE<1:0> = 10 并进入休眠状态	
CLRWDT 命令	
检测到振荡器故障	
退出休眠 + 系统时钟 = T1OSC、 EXTRC、INTOSC和EXTCLK	
退出休眠 + 系统时钟 = XT、HS 和LP	清零直到OST 延时结束
更改INTOSC 分频比（IRCF 位）	不受影响

10.3 寄存器说明

寄存器97H：看门狗定时器控制寄存器（WDTCON）

U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0
—	—	**WDTPS4**	**WDTPS3**	**WDTPS2**	**WDTPS1**	**WDTPS0**	**SWDTEN**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-6 **未实现**

bit5-1 **WDTPS <4:0>**：看门狗定时器周期选择位

00000 = 1:32 （时间间隔典型值为1 ms）
00001 = 1:64 （时间间隔典型值为2 ms）
00010 = 1:128 （时间间隔典型值为4 ms）
00011 = 1:256 （时间间隔典型值为8 ms）
00100 = 1:512 （时间间隔典型值为16 ms）
00101 = 1:1024 （时间间隔典型值为32 ms）
00110 = 1:2048 （时间间隔典型值为64 ms）
00111 = 1:4096 （时间间隔典型值为128 ms）
01000 = 1:8192 （时间间隔典型值为256 ms）
01001 = 1:16384 （时间间隔典型值为512 ms）
01010 = 1:32768 （时间间隔典型值为1s）
01011 = 1:65536 （时间间隔典型值为2s）（复位值）
01100 = 1:131072 （217）（时间间隔典型值为4s）

01101 = 1:262144 （218）（时间间隔典型值为8s）
01110 = 1:524288 （219）（时间间隔典型值为16s）
01111 = 1:1048576 （220）（时间间隔典型值为32s）
10000 = 1:2097152 （221）（时间间隔典型值为64s）
10001 = 1:4194304 （222）（时间间隔典型值为128s）
10010 = 1:8388608 （223）（时间间隔典型值为256s）
10011 = 保留。产生最小的时间间隔（1:32）

-
-
-

11111 = 保留。产生最小的时间间隔（1:32）

bit0 **SWDTEN:** 看门狗定时器软件使能/ 禁止位
如果WDTE<1:0> = 00：
该位被忽略。
如果WDTE<1:0> = 01：
1 = WDT开启
0 = WDT关闭
如果WDTE<1:0> = 1x：
该位被忽略。

11.0 数据 EEPROM 存储器控制

在整个VDD 范围内的正常工作期间，数据EEPROM都是可读写的。这些存储器并不直接映射到寄存器文件空间，而是通过特殊功能寄存器（SFR）来间接寻址。有4 个SFR 用于访问这些存储器，如下：

- EECON1
- EECON2
- EEDATL
- EEADRL

当与数据存储器模块接口时，EEDATL 存放8 位读/ 写数据，而EEADRL 存放被访问的EEDATL 存储单元的地址。这些器件具有256 字节的数据EEPROM，地址范围从0h 至0FFh。EEPROM数据存储器允许以字节为单位读写。EEPROM字节写操作会自动擦除存储单元并写入新数据（在写入前擦除）。写入时间由片上定时器控制。写入/ 擦除电压是由片上电荷泵产生的，此电荷泵在器件字节或字操作的电压范围内工作。当器件被代码保护时，器件编程器将不再能访问数据或程序存储器。当器件被代码保护时，CPU 仍可继续读写数据EEPROM 存储器。

11.1 EEADRL 寄存器

EEADRL 寄存器能寻址最大256 字节的数据EEPROM。当选择EEPROM地址值时，只将地址的LSB 写入EEADRL 寄存器。

11.2 EECON1 和 EECON2 寄存器

EECON1 是访问EE 存储器的控制寄存器。控制位EEPGD 决定访问的是程序存储器还是数据存储器。当清零时，任何后续操作都将针对EEPROM 存储器进行。当置1 时，任何后续操作都将针对程序存储器进行。在复位时，默认情况下会选择EEPROM。控制位RD 和WR 分别用于启动读和写操作。用软件只能将这两位置1 而无法清零。在读或写操作完成后，它们由硬件清零。由于无法用软件将WR 位清零，可避免意外地过早终止写操作。当WREN 位置1 时，允许进行写操作。上电时，WREN位被清零。在正常工作期间，如果写操作被复位中断，WRERR 位会置1。在这些情况下，复位后用户可以检查WRERR 位并执行相应的错误处理程序。在写操作完成时，PIFB2寄存器的中断标志位EEIF将置1。它必须用软件清零。读EECON2 将得到全0。EECON2寄存器仅在数据EEPROM写序列中使用。要使能写操作，必须向EECON2中写入特定模式。

11.3 使用数据 EEPROM

数据EEPROM 是高耐用性可字节寻址的阵列，已将其优化以便存储频繁变动的信息（例如，程序变量或其他经常更新的数据）。如果一个段中的变量经常发生改变，而另一个段中的变量不发生改变，就可能造成超出对EEPROM 的总写次数，而不超出对某个字节的总写次数。如果发生这种情况，必须执行一次阵列刷新。出于这个原因，不经常更改的变量（如常量、ID 和校准值等）应存放在闪存程序存储器中。

11.3.1 读取 EEPROM 存储器

要读取数据存储单元，用户必须将地址写入EEADRL寄存器，清零EECON1 寄存器的EEPGD 和CFGS 控制位，然后将控制位RD 置1。在紧接着的下一个周期，EEDATL 寄存器中的数据即可使用；因此，可在下一条指令读取。EEDATL 将保留该值直到另一次读操作开始或用户写入新值（在写操作期间）。

例 11-1： 读取数据EEPROM

```
BANKSEL   EEADRL          ;
LDWI      DATA_EE_ADDR    ;
STWR      EEADRL          ;Data Memory
                          ;Address to read
BCR       EECON1,CFGS     ;Deselect Config space
BCR       EECON1,EEPGD    ;Point to DATA memory
BSR       EECON1,RD       ;EE Read
LDR       EEDATL,W        ;W = EEDATL
```

注：不管CPD 位的设置如何，总是可以读取数据EEPROM。

11.3.2 写入数据 EEPROM 存储器

要写入EEPROM数据存储单元，用户必须先将地址写入EEADRL寄存器，并将数据写入EEDATL寄存器。然后用户必须按特定序列开始写入每个字节。如果没有完全按照上述序列（即先将55h写入EECON2，随后将AAh写入EECON2，最后将WR位置1）写入每个字节，将不会启动写操作。在该代码段中应禁止中断。此外，必须将EECON1中的WREN 位置1以使能写操作。该机制可防止由于意外（非预期的）执行代码（即程序失控）导致误写数据EEPROM。除了对EEPROM进行更新之外，用户应始终保持WREN位清零。WREN位不会被硬件清零。在写操作序列开始后，将WREN位清零将不会影响写周期。除非WREN位置1，否则WR位将禁止置1。写周期完成时， WR位由硬件清零并且EE写操作完成中断标志位（EEIF）置1。用户可以允许该中断或查询该位。EEIF必须用软件清零。

例 11-2： 写入数据EEPROM

```
BANKSEL   EEADRL          ;
LDWI      DATA_EE_ADDR    ;
SWTR      EEADRL          ;Data Memory Address to write
LDWI      DATA_EE_DATA    ;
STWR      EEDATL          ;Data Memory Value to write
BCR       EECON1, CFGS    ;Deselect Configuration space
BCR       EECON1, EEPGD   ;Point to DATA memory
BSR       EECON1, WREN    ;Enable writes
BCR       INTS, GIE       ;Disable INTs.
LDWI      55h             ;
STWR      EECON2          ;Write 55h
LDWI      0AAh            ;
STWR      EECON2          ;Write AAh
BSR       EECON1, WR      ;Set WR bit to begin write
BSR       INTS, GIE       ;Enable Interrupts
BCR       EECON1, WREN    ;Disable writes
BTSC      EECON1, WR      ;Wait for write to complete
LJUMP     $-2             ;Done
```

11.3.3 防止误写操作的保护模式

有些情况下，用户并不希望写入数据EEPROM存储器。为防止EEPROM误写操作，芯片内嵌了各种保护机制。上电时，WREN 被清零。此外，上电延时定时器（延时64 ms）也会阻止对EEPROM 进行写操作。写操作启动序列以及WREN 位可一起用来防止在以下情况下的意外写操作：

- 欠压
- 电源毛刺
- 软件故障

11.3.4 代码保护期间的数据 EEPROM 的操作

通过将配置字寄存器1中的CPD 位编程为0，可对数据存储器进行代码保护。当数据存储器被代码保护时，只有CPU 仍能对数据EEPROM 进行读写操作。对数据存储器进行代码保护的同时，建议用户也对程序存储器采取代码保护。这可以防止有人将您的程序替换为可访问数据EEPROM 内容的程序。

11.4 寄存器说明

寄存器193H：EEPROM数据寄存器（EEDATL）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EEDATL7	**EEDATL 6**	**EEDATL 5**	**EEDATL 4**	**EEDATL 3**	**EEDATL 2**	**EEDATL 1**	**EEDATL 0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-0 **EEDATL <7:0>**：EEPROM数据字节读写值

寄存器191H：EEPROM地址寄存器（EEADRL）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EEADDR7	**EEADDR6**	**EEADDR5**	**EEADDR4**	**EEADDR3**	**EEADDR2**	**EEADDR1**	**EEADDR0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-0 **EEADDR <7:0>**：EEPROM地址寄存器

寄存器195H：EEPROM控制1寄存器（EECON1）

R/W-0	R/W-0	R/W-0	R/W/HC-0	R/W-0	R/W-0	R/W/HC-0	R/W/HC-0
EEPDG	**CFGS**	**LWLO**	**FREE**	**WRERR**	**WREN**	**WR**	**RD**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
HC = 硬件清零位	1 = 置1	0 = 清零	x = 未知

bit7 **EEPDG**：数据EEPROM 存储器选择位
1 = 访问闪存程序存储器
0 = 访问数据EEPROM 存储器

bit6 **CFGS**：数据EEPROM 存储器或配置寄存器选择位
1 = 访问配置、用户ID 和器件ID 寄存器

0 = 数据EEPROM 存储器

bit5 **LWLO**：可以忽略

bit4 **FREE**：可以忽略

bit3 **WRERR**：EEPROM 错误标志位

1 = 条件指示试图执行不合法的编程或擦除序列，或者发生终止（试图将WR 位置1 （写入1）时自动将该位置1）

0 = 编程或擦除操作正常完成

bit2 **WREN**：编程/ 擦除使能位

1 = 允许编程/ 擦除周期

0 = 禁止对程序闪存和数据EEPROM 的编程/ 擦除操作

bit1 **WR**：写控制位

1 = 启动程序闪存或数据EEPROM 的编程/ 擦除操作。
操作是自定时的，一旦该操作完成，该位即由硬件清零。
用软件只能将WR 位置1 （不能清零）。

0 = 对闪存或数据EEPROM 的编程/ 擦除操作已完成并且变为无效。

bit0 **RD**：读控制位

1 = 启动程序闪存或数据EEPROM 的读操作。读操作需要一个周期。RD 由硬件清零。用软件只能将RD 位置1 （不能清零）。

0 = 不启动程序闪存或数据EEPROM 的读操作。

寄存器196H：EEPROM控制2寄存器（EECON2）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EEADDR7	**EEADDR6**	**EEADDR5**	**EEADDR4**	**EEADDR3**	**EEADDR2**	**EEADDR1**	**EEADDR0**
bit7							bit0

图注：
R = 可读位　W = 可写位　U = 未实现位，读为0
-n = POR时的值　1 = 置1　0 = 清零　x = 未知

bit7-0 **EECON2 <7:0>**：数据EEPROM解锁模式位，不是物理寄存器
要对写操作进行解锁，必须先写入55h，接着写入AAh，然后再将EECON1 寄存器的WR 位置1。写入该寄存器的值用于对写操作进行解锁。对于这些写操作，存在一些特定的时序要求。

表 11-1：与 EEPROM 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
EECON1	EEPDG	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	0000 x000	0000 q000
EECON2	EEPROM 控制寄存器 2								0000 0000	0000 0000
EEADRL	EEADDR7	EEADDR6	EEADDR5	EEADDR4	EEADDR3	EEADDR2	EEADDR1	EEADDR0	0000 0000	0000 0000
EEDATL	EEDATL7	EEDATL6	EEDATL5	EEDATL4	EEDATL3	EEDATL2	EEDATL1	EEDATL0	xxxx xxxx	xxxx xxxx
INTS	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x	0000 000q
PIEB2	OSFIE	C2IE	C1IE	EEIE	BCLIE	—	—	—	0000 0xxx	0000 0xxx
PIFB2	OSFIF	C2IF	C1IF	EEIF	BCLIF	—	—	—	0000 0xxx	0000 0xxx

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

12.0 I/O 端口

根据选定的器件和使能的外设，最多有两个端口可供使用。通常而言，当某个外设被使能时，其相关引脚可能不能用作通用I/O 引脚。每个端口都有三个用于工作的标准寄存器。这些寄存器是：

• CPIOx寄存器（数据方向）
• PORTx寄存器（读取器件引脚的电平）
• LATx寄存器（输出锁存器）

这些端口可能具有以下一个或多个额外的寄存器。这些寄存器是：

• ADINSx （模拟选择）
• P x PHR（弱上拉）
• INLVLx（输入级控制）

数据锁存器（LATx 寄存器）对I/O 引脚驱动值进行读-修改-写操作时非常有用。对LATx 寄存器的写操作与写入相应PORTx 寄存器的效果相同。读取LATx 寄存器时，将会读取I/O 端口锁存器中保存的值，而读取PORTx 寄存器时，将会读取实际的I/O 引脚值。带有模拟功能的端口还具有ADINSx 寄存器，该寄存器可用于禁止数字输入和节省功耗。图12-1 给出了通用I/O 端口的简化模型，没有给出与其他外设的接口。

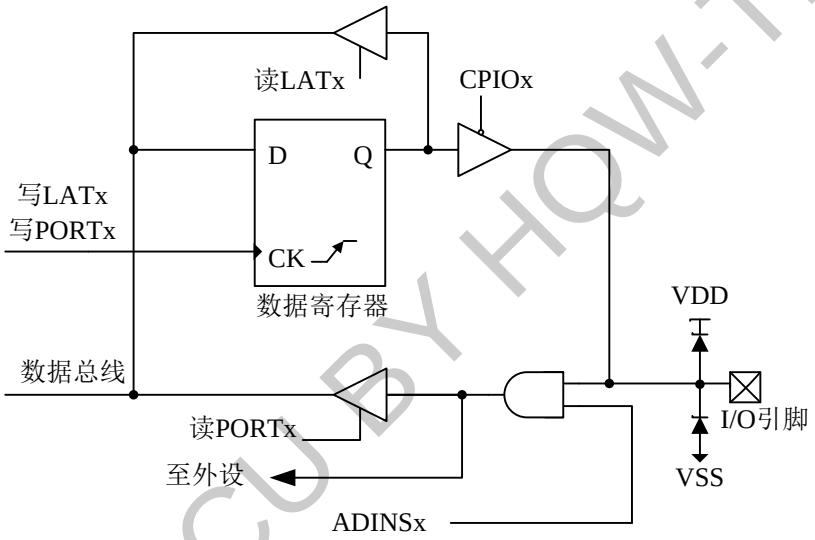


图 12-1： 通用I/O 端口的工作原理

12.1 备用引脚功能

备用引脚功能控制（APFCON）寄存器用于将特定的外设输入和输出功能配置到不同的引脚上。APFCON寄存器如寄存器11DH所示。对于本器件，以下功能可以配置到不同的引脚上。

• RX/DT
• TX/CK
• SDO
• SS（从选择）
• T1G
• P1B
• CCP1/P1A

这些位对于任意CPIO寄存器的值没有任何影响。PORT和CPIO 改写会被送到正确的引脚。未选择的引脚不会受影响。

12.2 PORTA 寄存器

PORTA 是一个6 位宽的双向端口。对应的数据方向寄存器是CPIOA。将CPIOA 某位置1（= 1）时，会将PORTA 的相应引脚设为输入（即，禁止输出驱动器）。将CPIOA 某位清零（= 0）时，会将PORTA的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选定的引脚）。PA3 是个例外，它仅可作为输入引脚，其CPIO 位总是读为1。例12-1 显示了如何初始化PORTA 。读PORTA寄存器将读出相应引脚的状态，而对PORTA 寄存器进行写操作则将写入端口锁存器。所有写操作都是读-修改-写操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器（LATA）。CPIOA寄存器控制PORTA 引脚输出驱动器，即使它们被用作模拟输入。当引脚用于模拟输入时，用户应确保CPIOA寄存器中的各位保持置1。配置为模拟输入的I/O 引脚总是读为0。

12.2.1 模数选择寄存器

ADINSA 寄存器用于将I/O 引脚的输入模式配置为模拟。将相应的ADINSA位设置为高电平将使引脚上的所有数字读操作都读为0，并允许引脚上的模拟功能正确工作。ADINSA位的状态不会影响数字输出功能。CPIO清零且ADINS 置1 的引脚将仍作为数字输出工作，但输入模式将变为模拟。当在受影响的端口上执行读- 修改-写指令时，这会引起意外操作。

注：必须对 ADINSC 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 0。

例 12-1： 初始化PORTA

```
BANKSEL   PORTA      ;
CLRR      PORTA      ;Init PORTA
BANKSEL   LATA       ;Data Latch
CLRR      LATA       ;
BANKSEL   ADINSA     ;
CLRR      ADINSA     ;digital I/O
BANKSEL   CPIOA      ;
LDWI      B'00111000'  ;Set RA<5:3> as inputs
SWTR      CPIOA      ;and set RA<2:0> as
                     ;outputs
```

12.2.2 PORTA 功能和输出优先级

每个PORTA 引脚都与其他功能复用。这里将简要说明引脚及其复用功能和输出优先级。当使能多个输出时，实际引脚控制权将属于以下列表中编号最小的外设。优先级列表中未列出模拟输入功能，例如ADC、比较器和电容传感输入。这些输入在使用ADINSx 寄存器将I/O引脚设置为模拟模式时有效。当引脚处于模拟模式时，数字输出功能可以按照下面列出的优先级控制引脚。

PA0

1. ICSPDAT
2. ICDDAT
3. DACOUT （DAC）
4. TX/CK （EUSART）

PA1

1. ICSPCLK
2. ICDCLK

3. RX/DT （EUSART）

PA2

1. SRQ
2. C1OUT （比较器）

PA3

无输出优先级。仅用作输入的引脚。

PA4

1. OSC2
2. CLKOUT
3. T1OSO （Timer1 振荡器）
4. CLKR
5. SDO

PA5

1. OSC1
2. T1OSI （Timer1 振荡器）

12.3 PORTC 寄存器

PORTC 是一个6 位宽的双向端口。对应的数据方向寄存器是CPIOC。将CPIOC 某位置1（=1）时，会将PORTC 的相应引脚设为输入（即，使相应的输出驱动器呈高阻态）。将CPIOC 某位清零（= 0）时，会将PORTC 的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选定的引脚） 。例12-2显示了如何初始化PORTC。读 PORTC 寄存器将读出相应引脚的状态，而对PORTC 寄存器进行写操作则是将数据写入端口锁存器。所有写操作都是读- 修改- 写操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器（LATC）。CPIOC 寄存器控制PORTC 引脚输出驱动器，即使它们被用作模拟输入。当引脚用于模拟输入时，用户应确保CPIOC 寄存器中的各位保持置1。配置为模拟输入的I/O 引脚总是读为0。

例 12-2： 初始化PORTC

```
BANKSEL     PORTC       ;
CLRR        PORTC       ;Init PORTC
BANKSEL     LATC        ;Data Latch
CLRR        LATC        ;
BANKSEL     ADINSC
CLRR        ADINSC      ;Make RC<5:0> digital
BANKSEL     CPIOB       ;
LDWI        B’00110000’ ;Set RC<5:4> as inputs
                        ;and RC<3:0> as outputs
STWR        CPIOC       ;
```

12.3.1 PORTC 功能和输出优先级

每个PORTC 引脚都与其他功能复用。这里将简要说明引脚及其复用功能和输出优先级。当使能多个输出时，实际引脚控制权将属于以下列表中编号最小的外设。下面的列表中未包含模拟输入和一些数字输入功能。这些输入功能会在引脚配置为输出时保持有效。一些数字输入功能会改写一些其他端口功能，优先级列表中包含了这些功能。

PC0
1. SCL （MSSP）
2. SCK （MSSP）
PC1
1. SDA （MSSP）
PC2
1. SDO （MSSP）
2. P1D
PC3
1. P1C
PC4
1. MDOUT
2. SRNQ
3. C2OUT
4. TX/CK
5. P1B
PC5
1. RX/DT
2. CCP1/P1A

12.4 寄存器说明

寄存器0CH：PORTA寄存器（PORTA）

U-0	U-0	R/W-x	R/W-x	R-x	R/W-x	R/W-x	R/W-x
—	—	**PA5**	**PA4**	**PA3**	**PA2**	**PA1**	**PA0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-6 未实现：读为0
bit5-0 **PA<5:0>**：PORTAI/O引脚位
1=PORTA引脚电平>V_{IH}
0=PORTA引脚电平<V_{IL}

寄存器8CH：PORTA三态寄存器（CPIOA）

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	**CPIOA5**	**CPIOA4**	—	**CPIOA2**	**CPIOA1**	**CPIOA0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-6 未实现：读为0
bit5-0 **CPIOA<5:0>**：PORTA三态控制位
1=PORTA引脚配置为输入（三态）
0=PORTA引脚配置为输出

注1：在XT、HS和LP振荡模式下，CPIOA<5:4>始终读为1。

寄存器10CH：PORTA数据锁存寄存器（LATA）

U-0	U-0	R/W-x	R/W-x	U-0	R/W-x	R/W-x	R/W-x
—	—	**LATA5**	**LATA 4**	—	**LATA2**	**LATA 1**	**LATA 0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-6 未实现：读为0

bit5-4 **LATA <5：4>**：PA <5：4>输出锁存值位

bit3 未实现：读为0

bit2-0 **LATA <2：0>**：PA <2：0>输出锁存值位

注：写入PORTA 时，实际上会写入相应的LATA 寄存器。读取PORTA 寄存器时，将返回实际的I/O 引脚值。

寄存器18CH：PORTA模拟选择寄存器（ADINSA）

U-0	U-0	U-0	R/W-x	U-0	R/W-x	R/W-x	R/W-x
—	—	—	**AN3**	—	**AN2**	**AN1**	**AN0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-5 未实现：读为0

bit4 **AN3**：PA 4端口选择模拟或数字功能
0 = 数字I/O。引脚被配置为端口或数字特殊功能。
1 = 模拟输入。引脚被配置为模拟输入(1)。 数字输入缓冲器被禁止。

bit3 未实现：读为0

bit2-0 **AN <2:0>**：PA <2:0>端口选择模拟或数字功能
0 = 数字I/O。引脚被配置为端口或数字特殊功能。
1 = 模拟输入。引脚被配置为模拟输入(1)。 数字输入缓冲器被禁止。

注 1：当将某个引脚设置为模拟输入时，必须将相应的CPIO 位设置为输入模式，以允许从外部控制引脚电压。

寄存器20CH：PORTA弱上拉寄存器（PAPHR）

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	**PAPHR5**	**PAPHR4**	**PAPHR3**	**PAPHR2**	**PAPHR1**	**PAPHR0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 未实现：读为0

bit5-0 **PAPHR<5:0>**：弱上拉寄存器位
1 = 使能上拉
0 = 禁止上拉

注 1： 必须清零OPTION 寄存器的全局WPUEN 位，从而使能各个上拉功能。
2： 如果引脚被配置为输出，则自动禁止弱上拉器件。

寄存器0EH：PORTC寄存器（PORTC）

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
-	-	**PC5**	**PC4**	**PC3**	**PC2**	**PC1**	**PC0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-0 **PC<5:0>**：PORTCI/O引脚位
1 = PORTC引脚电平>VIH
0 = PORTC引脚电平<VIL

寄存器8EH：PORTC三态寄存器（CPIOC）

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	**CPIOC5**	**CPIOC4**	**CPIOC3**	**CPIOC2**	**CPIOC1**	**CPIOC0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 未实现：读为0
bit5-0 **CPIOC<5:0>**：PORTC三态控制位
1 = PORTC引脚配置为输入（三态）
0 = PORTC引脚配置为输出

寄存器10EH：PORTC数据锁存寄存器（LATC）

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	**LATC5**	**LATC4**	**LATC3**	**LATC2**	**LATC1**	**LATC0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 未实现：读为0
bit5-0 **LTAC<5:0>**：PORTC 输出锁存值位
注1：写入PORTC时，实际上会写入相应的LATC寄存器。读取PORTC寄存器时，将返回实际的I/O引脚值

寄存器18EH：PORTC模拟选择寄存器（ADINSC）

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	**AN7**	**AN6**	**AN5**	**AN4**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-4 未实现：读为0
bit3-0 **AN<7:4>**：将PC<3:0>引脚选择位模拟或数字功能
0 = 数字I/O。引脚被配置为端口或数字特殊功能。
1 = 模拟输入。引脚被配置为模拟输入(1)。 数字输入缓冲器被禁止。

注 1:当将某个引脚设置为模拟输入时，必须将相应的CPIO 位设置为输入模式，以允许从外部控制引脚电压。

寄存器20EH：PORTC弱上拉寄存器（PCPHR）

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	**—**	**PCPHR5**	**PCPHR4**	**PCPHR3**	**PCPHR2**	**PCPHR1**	**PCPHR0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 未实现：读为0

bit5-0 **PCPHR<5:0>**：弱上拉寄存器位
1 = 使能上拉
0 = 禁止上拉

注 1： 必须清零OPTION 寄存器的全局WPUEN 位，从而使能各个上拉功能。
2： 如果引脚被配置为输出，则自动禁止弱上拉器件。

寄存器11DH：备用引脚功能控制寄存器（APFCON）

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RXDTSEL	**SDOSEL**	**SSSEL**	**—**	**T1GSEL**	**TXCKSEL**	**P1BSEL**	**CCP1SEL**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7 **RXDTSEL**：引脚选择位
0 = PC5 上具有RX/DT 功能
1 = PA1 上具有RX/DT 功能

bit6 **SDOSEL**：引脚选择位
0 = PC2 上具有SDO 功能
1 = PA4 上具有SDO 功能

bit5 **SSSEL**：引脚选择位
0 = PC3 上具有SS 功能
1 = PA3 上具有SS 功能

bit4 未实现： 读为0

bit3 **T1GSEL**：引脚选择位
0 = PA4 上具有T1G 功能
1 = PA3 上具有T1G 功能

bit2 **TXCKSEL**：引脚选择位
0 = PC4 上具有TX/CK 功能
1 = PA0 上具有TX/CK 功能

bit1 **P1BSEL**：引脚选择位
PC4 上总是具有P1B 功能

bit0 **CCP1SEL:** 引脚选择位
PC5 上总是具有CCP1/P1A 功能

表 12-1：与 PORTA 和 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	--11-111	--11-111
ADINSC	—	—	—	—	AN7	AN6	AN5	AN4	----1111	----1111
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	00000000	00000000
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--000000	--000000
OPTION	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	11111111	11111111
PORTC	—	—	PC5	PC4	PC3	PC2	PC1	PC0	--000000	--uuuuuu
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	--111111	--111111
PAPHR	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0	--111111	--111111
PCPHR	—	—	PCPHR5	PCPHR4	PCPHR3	PCPHR2	PCPHR1	PCPHR0	--111111	--111111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	--111111	--111111
LATC	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	--000000	--uuuuuu

图注： x = 未知， u = 不变， — =未实现（读为 0）。

13.0 电平变化中断

PORTA 引脚可以配置为作为电平变化中断（Interrupt-On-Change， IOC）引脚工作。中断可以通过检测具有上升沿或下降沿的信号而产生。任意一个PORTA 引脚或PORTA 引脚组合都可以配置为产生中断。电平变化中断模块具有以下特性：

- 电平变化中断允许（主开关）
- 独立的引脚配置
- 上升沿和下降沿检测
- 独立的引脚中断标志

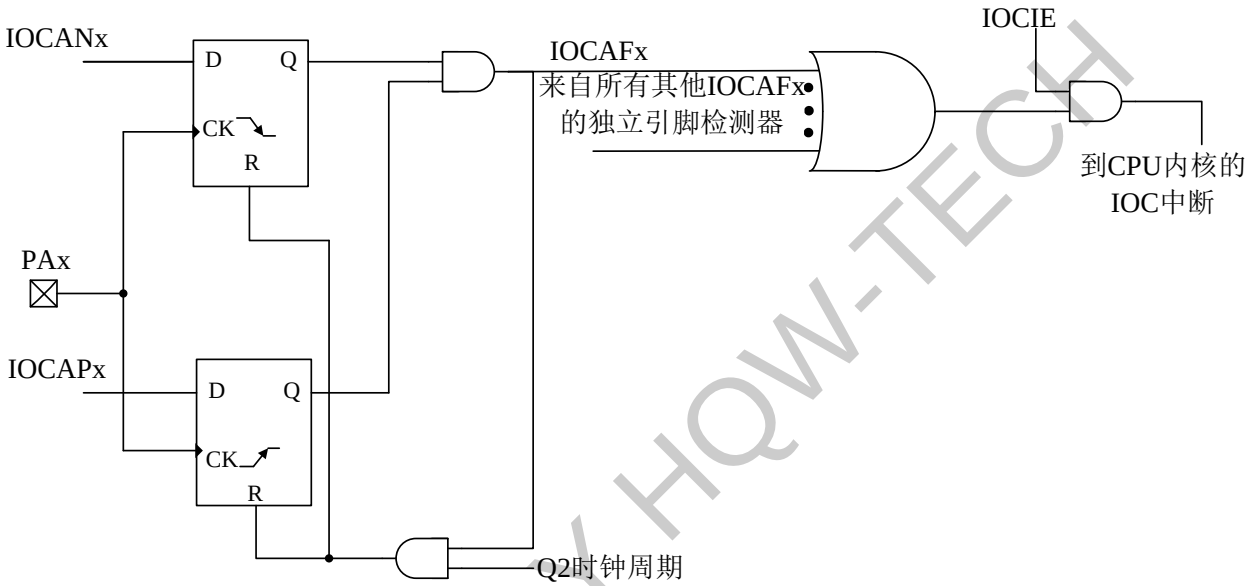


图13-1： 电平变化中断框图

13.1 使能模块

要允许各个PORTA 引脚产生中断， INTS 寄存器的PAIE 位必须置1。如果PAIE 位被禁止，在引脚上仍然会发生边沿检测，但不会产生中断。

13.2 独立的引脚配置

对于每个PORTA 引脚，都提供了上升沿检测器和下降沿检测器。要使某个引脚检测上升沿，需要将IOCAP寄存器中的相关IOCAPx 位置1。要允许引脚检测下降沿，需要将IOCAN 寄存器中的相关IOCANx 位置1。通过同时将IOCAPx位和IOCANx位置1（分别在IOCAP和IOCAN寄存器中），一个引脚可以配置为同时检测上升沿和下降沿。

13.3 中断标志

位于IOCAF 寄存器中的IOCAFx 位是对应于PORTA 的电平变化中断引脚的状态标志。如果在正确使能的引脚上检测到期望的边沿，则对应于该引脚的状态标志会置1，并且如果PAIE 位置1，则还会产生中断。INTS寄存器的PAIF 位会反映所有IOCAFx 位的状态。

13.4 清零中断标志

各个状态标志（IOCAFx 位）可以通过将其复位为零的方式清零。如果在该清零操作期间检测到另一个边沿，则无论实际写入的值如何，关联的状态标志都会在序列结束时置1。为了确保在清零标志时不会丢失任何已检测的边沿，应当仅执行可屏蔽已知更改位的与操作。以下序列是一个说明应执行何种操作的示例。

例13-1： 清零中断标志（以PORTA为例）

```
LDWI    0xff
XORWR   IOCAF, W
ANDWR   IOCAF, F
```

13.5 休眠模式下的操作

如果PAIE 位置1，电平变化中断的中断序列会将器件从休眠模式唤醒。如果在处于休眠模式时检测到边沿，则在退出休眠模式执行第一条指令之前，会先更新IOCAF 寄存器。

13.6 寄存器说明

寄存器391H：PORTA电平变化中断正边沿寄存器（IOCAP）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	**—**	**IOCAP5**	**IOCAP4**	**IOCAP3**	**IOCAP2**	**IOCAP1**	**IOCAP0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 未实现：读为0

bit5-0 **IOCAP<5:0>**：电平变化中断PORTA 正边沿使能位

1 = 在引脚上对于正向边沿允许电平变化中断。关联的状态位和中断标志将在检测到边沿时置1。

0 = 禁止关联引脚的电平变化中断。

寄存器392H：PORTA电平变化中断负边沿寄存器（IOCAN）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	**—**	**IOCAN5**	**IOCAN4**	**IOCAN3**	**IOCAN2**	**IOCAN1**	**IOCAN0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 未实现：读为0

bit5-0 **IOCAN<5:0>**：电平变化中断PORTA 负边沿使能位

1 = 在引脚上对于负向边沿允许电平变化中断。关联的状态位和中断标志将在检测到边沿时置1。

0 = 禁止关联引脚的电平变化中断。

寄存器393H：PORTA电平变化中断负边沿寄存器（IOCAF）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	**—**	**IOCAF5**	**IOCAF4**	**IOCAF3**	**IOCAF2**	**IOCAF1**	**IOCAF0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 未实现：读为0

bit5-0 **IOCAF<5:0>**：电平变化中断PORTA 标志位

1 = 在关联引脚上检测到使能的电平变化。在IOCAPx = 1，并在PAx上检测到上升沿时置1，或者在IOCANx = 1，并在PAx上检测到下降沿时置1。

0 = 未检测到电平变化，或者用户清除了检测到的电平变化。

表 13-1：与电平变化中断有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TMROIE	INTE	PAIE	TMR0IF	INTF	PAIF	0000 000x	0000 000q
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	xxx1 x111	xxx1 x111
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	xx00 0000	xx00 0000
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	xx00 0000	xx00 0000
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	xx00 0000	xx00 0000
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xx00 0000	xx00 0000

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

14.0 固定参考电压（FVR）

固定参考电压（FVR）是独立于VDD 的稳定参考电压，可选的输出电压有1.024V、2.048V 或4.096V。FVR 的输出可以配置为向以下对象提供参考电压：

- ADC 输入通道
- ADC 正参考电压
- 比较器的同相输入
- 数模转换器（DAC）
- 电容传感（CPS）模块

FVR 可以通过将FVRCON 寄存器的FVREN 位置1 来使能。

14.1 独立的增益放大器

送到ADC、比较器、DAC 和CPS 模块的FVR 输出会经过两个独立的可编程增益放大器。每个放大器都可以配置为将参考电压放大1 倍、2 倍或4 倍，产生三种可能电压。FVRCON 寄存器的ADFVR<1:0> 位用于使能和配置送到ADC 模块的参考电压的增益放大器设置。FVRCON 寄存器的CDAFVR<1:0> 位用于使能和配置送到比较器、DAC 和CPS 模块的参考电压的增益放大器设置。

14.2 FVR 稳定周期

当固定参考电压模块使能时，参考电压和放大电路需要一段时间才能达到稳定。在电路稳定下来、可供使用时，FVRCON 寄存器的FVRRDY 位将会置1。

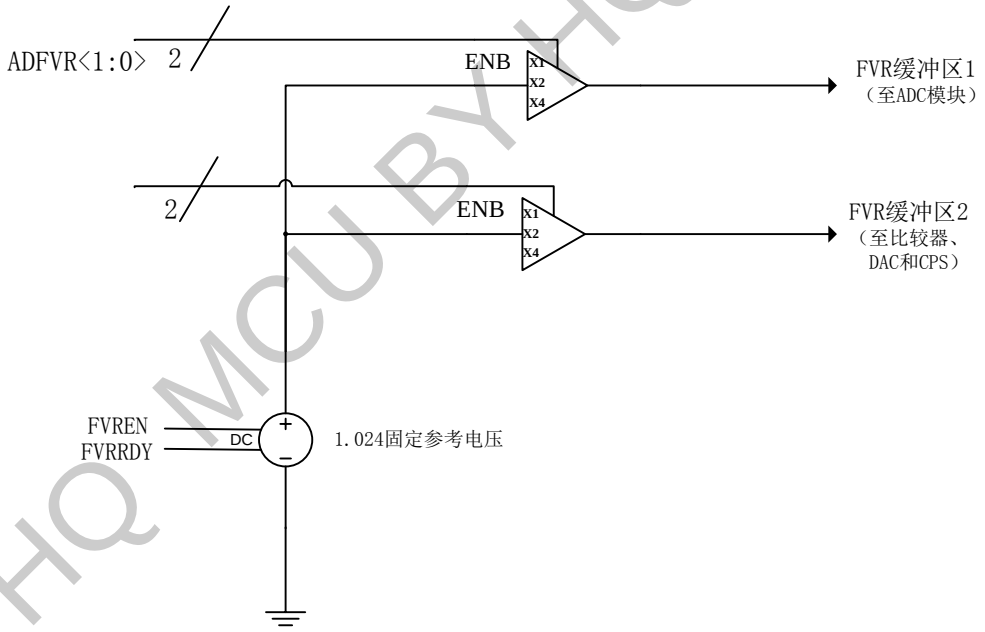


图14-1：参考电压框图

14.3 寄存器说明

寄存器117H：固定参考电压控制寄存器（FVRCON）

R/W-0	R-q	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FVREN	**FVRRDY**	**TSEN**	**TSRNG**	**CDAFVR1**	**CDAFVR0**	**ADFVR1**	**ADFVR0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
q=取决具体条件	1= 置1	0= 清零	x= 未知

bit7 **FVREN：**固定参考电压使能位
0 = 禁止固定参考电压
1 = 使能固定参考电压

bit6 **FVRRDY：**固定参考电压就绪标志位
0 = 固定参考电压输出未就绪或未使能
1 = 固定参考电压输出就绪备用

bit5 **TSEN：**温度指示器使能位
0 = 禁止温度指示器
1 = 使能温度指示器

bit4 **TSRNG：**温度指示器范围选择位
0 = VOUT = VDD - 2VT （低电压范围）
1 = VOUT = VDD - 4VT （高电压范围）

bit3-2 **CDAFVR<1:0>：**比较器和DAC 固定参考电压选择位
00 = 比较器、DAC 和CPS 模块固定参考电压外设输出关闭
01 = 比较器、DAC 和CPS 模块固定参考电压外设输出为1x （1.024V）
10 = 比较器、DAC 和CPS 模块固定参考电压外设输出为2x （2.048V）(1)
11 = 比较器、DAC 和CPS 模块固定参考电压外设输出为4x （4.096V）(1)

bit1-0 **ADFVR<1:0>：**ADC 固定参考电压选择位
00 = ADC 固定参考电压外设输出关闭
01 = ADC 固定参考电压外设输出为1x （1.024V）
10 = ADC 固定参考电压外设输出为2x （2.048V） (1)
11 = ADC 固定参考电压外设输出为4x （4.096V）(1)

注：1. 固定参考电压输出不能超出VDD。

表 14-1：与固定参考电压有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	0q00 0000	0q00 0000

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

15.0 温度指示器模块

本器件系列配备了用于测量硅裸片工作温度的温度电路。电路的工作温度量程介于-40°C 和+85°C 之间。其输出是与器件温度成正比的电压。温度指示器的输出在内部与器件ADC 连接。电路可以用作温度阈值检测器，也可以用作更精确的温度指示器，这取决于所执行的校准级别。执行单点校准时，电路可以指示邻近该点的温度。执行双点校准时，电路可以更精确地检测整个温度量程。

15.1 电路工作原理

图15-1 给出了温度电路的简化框图。与温度成正比的电压输出通过测量多个硅结的正向电压降而得到。
公式15-1 描述了温度指示器的输出特性。

公式15-1： VOUT 范围

高电压范围 ： $V_{OUT} = V_{DD} - 4V_T$
低电压范围 ： $V_{OUT} = V_{DD} - 2V_T$

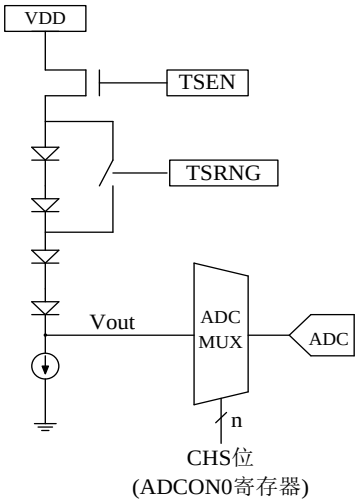


图15-1： 温度指示器电

温度检测电路集成了固定参考电压（FVR）模块。可以通过将FVRCON 寄存器的TSEN 位置1 来使能该电路。在禁止时，电路不会消耗任何电流。电路可以工作于高电压范围或低电压范围。高电压范围（选择方式是将FVRCON 寄存器的TSRNG 位置1）可提供较宽的输出电压。这可以在整个温度量程中提供更高的分辨率，但不同部分的一致性较低。该电压范围需要较高的偏置电压才能工作，所以需要较高的VDD。低电压范围的选择方式是将FVRCON寄存器的TSRNG位清零。低电压范围产生的电压降较小，所以只需较低的偏置电压就可以让电路工作。低电压范围旨在用于进行低电压操作。

15.2 最小工作电压与最低检测温度

当温度电路工作于低电压范围时，器件可以使用规范范围内的任意工作电压工作。当温度电路工作于高电压范围时，器件工作电压VDD 必须足够高，以确保正确地偏置温度电路。
表15-1给出了建议的最小VDD 与范围设置

表15-1： 建议的VDD 与范围

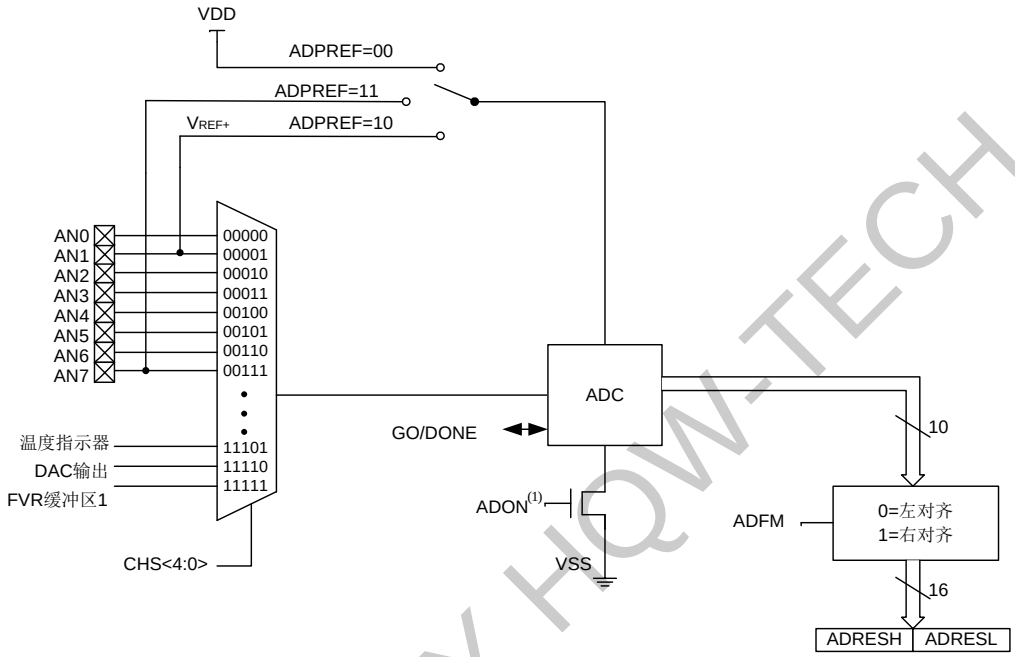
最小VDD， TSRNG = 1	最小VDD， TSRNG = 0
3.6V	1.8V

15.3 温度输出

电路的输出使用内部模数转换器测量。保留一路通道用于温度电路输出。为了确保温度的测量，在转换开始之前，用户必须在复用器连接到温度指示器之后至少等待200 微秒。另外，用户必须在两个温度指示器输出的连续两次转换之间等待200 微秒。

16.0 模数转换器（ADC）模块

模数转换器（ADC）可将模拟输入信号转换为信号的10 位二进制表示。该模块使用模拟输入，这些输入通过多路开关连接到同一个采样和保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生10 位二进制结果，并将转换结果存储在ADC 结果寄存器（ADRESH:ADRESL 寄存器对）中。图16-1 给出了ADC 的框图。可通过软件方式选择内部产生的电压或外部提供的电压作为ADC 参考电压。ADC可在转换完成时产生中断。该中断可用于将器件从休眠状态唤醒。



注 1：当ADON=0时，所有多路开关输入都会被断开

图16-1： ADC 框图

16.1 ADC 配置

配置和使用ADC 时必须考虑以下功能：

- 端口配置
- 通道选择
- ADC 参考电压选择
- ADC 转换时钟源
- 中断控制
- 结果格式

16.1.1 端口配置

ADC 可用于转换模拟和数字信号。转换模拟信号时，应通过设置相关的CPIO 和ADINS 位将I/O引脚配置为模拟。

注：在任何定义为数字输入的引脚上施加模拟电压可能导致输入缓冲器消耗的电流过大。

16.1.2 通道选择

有最多11 个通道选择可供使用：

- AN<3:0> 引脚
- AN<7:0> 引脚
- 温度指示器

• DAC 输出
• FVR （固定参考电压）输出

ADC0CN0 寄存器的CHS 位决定与采样保持电路相连接的通道。
当改变通道时，在开始下一次转换前需要一段延时。

16.1.3 ADC 参考电压

ADC0CN1 寄存器的ADPREF 位用于控制正参考电压。
负参考电压可以是：
• VREF+ 引脚
• VDD
• FVR 2.048V
• FVR 4.096V

16.1.4 转换时钟

可通过软件方式设置ADC0CN1 寄存器的ADCS 位来选择转换时钟源。有以下7 种时钟频率可供选择：
• FOSC/2
• FOSC/4
• FOSC/8
• FOSC/16
• FOSC/32
• FOSC/64
• FRC （专用内部振荡器）

完成一个位转换所需的时间定义为TAD。一次完整的10 位转换需要11.5 个TAD 周期，如图16-2 所示。为正确转换，必须满足合适的TAD 规范。表16-1 给出了适当的ADC 时钟选择的示例。

注：除非使用FRC，否则系统时钟频率的任何改变都会改变ADC 时钟频率，这会影响ADC 结果。

表16-1： ADC 时钟周期（TAD）与器件工作频率关系表

ADC 时钟周期（T_{AD}）		器件频率（FOSC）			
ADC 时钟源	ADCS<2:0>	16MHz	8MHz	4MHz	1MHz
FOSC/2	000	125 ns(2)	250 ns(2)	500 ns(2)	2.0 μs
FOSC/4	100	250 ns(2)	500 ns(2)	1.0 μs	4.0 μs
FOSC/8	001	0.5 μs(2)	1.0 μs	2.0 μs	8.0 μs(3)
FOSC/16	101	1.0 μs	2.0 μs	4.0 μs	16.0 μs(3)
FOSC/32	010	2.0 μs	4.0 μs	8.0 μs(3)	32.0 μs(3)
FOSC/64	110	4.0 μs	8.0 μs(3)	16.0 μs(3)	64.0 μs(3)
F_{RC}	X11	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)	1.0-6.0 μs(1,4)

图注： 阴影单元表示超出了建议范围。
注 1： 对于VDD， FRC 时钟源具有1.6 μs 的典型TAD 时间。
2： 这些值均违反了所需的最小T_{AD} 时间。
3： 为了加快转换速度，建议选用另一个时钟源。
4： 通过系统时钟FOSC 来产生ADC 时钟时，可以最大程度降低ADC 时钟周期（T_{AD}）和ADC 总转换时间。但是，如果要在器件处于休眠模式时执行转换，则必须使用FRC 时钟源。

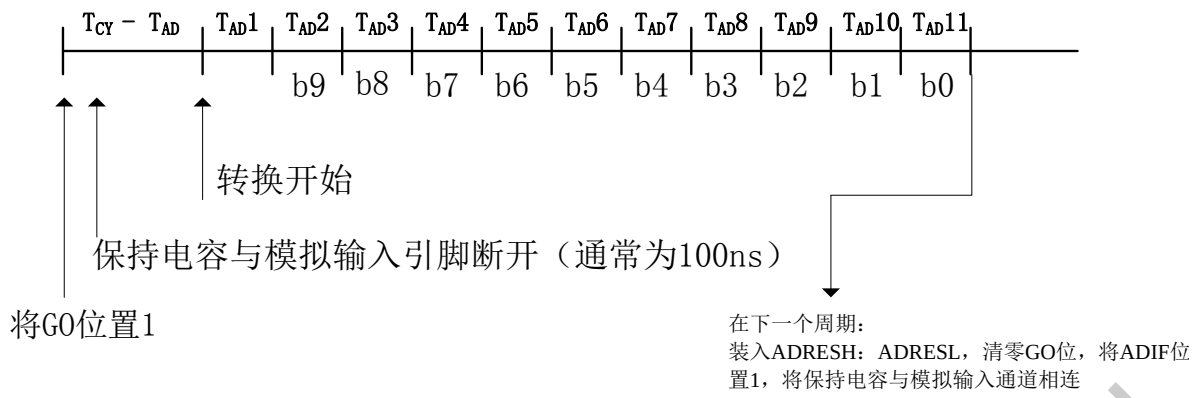


图16-2： 模数转换TAD 周期

16.1.5 中断

ADC 模块可在模数转换完成时产生中断。ADC 中断标志位是PIFB1 寄存器中的ADIF 位。ADC 中断允许位是PIEB1 寄存器中的ADIE 位。ADIF 位必须用软件清零。

注 1：ADIF 位在每次转换完成时置1，与是否允许ADC 中断无关。 2： 仅当选择了FRC 振荡器时， ADC 才能在休眠模式下工作。

器件工作或休眠时都可产生该中断。如果器件处于休眠状态，该中断会唤醒器件。从休眠状态唤醒时，总是执行紧跟SLEEP指令后的下一条指令。如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止INTS寄存器的GIE 和PEIE 位。如果使能了INTS 寄存器的GIE 和PEIE 位，执行将切换到中断服务程序。

16.1.6 结果格式

10 位A/D 转换结果可以两种格式提供：左对齐或右对齐。ADC0CN1 寄存器的ADFM 位控制输出格式。图16-3 给出了两种输出格式。

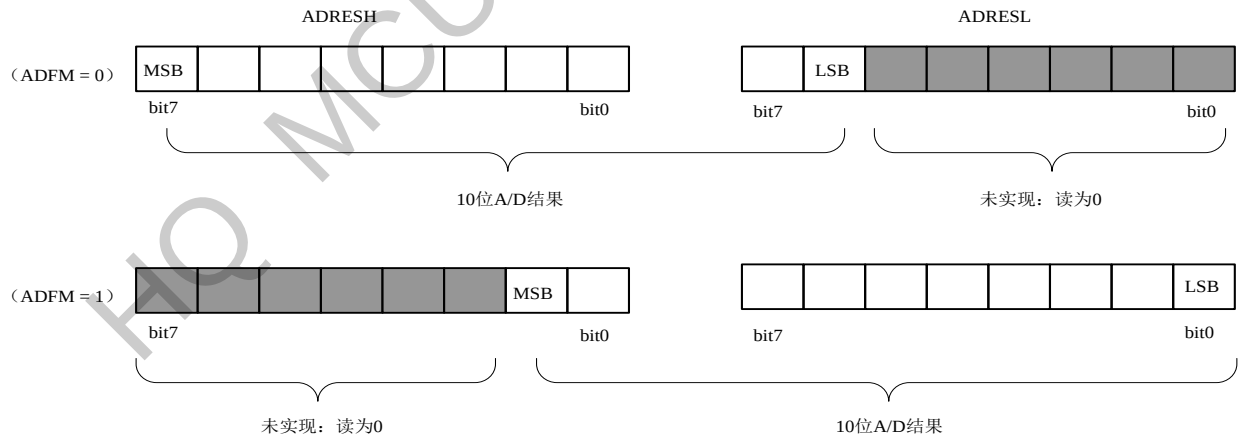


图16-3：10位A/D转换结果格式

16.2 工作原理

16.2.1 启动转换

要使能ADC 模块，ADC0CN0 寄存器的ADON 位必须设置为1。将ADC0CN0 寄存器的GO/DONE 位设置为1将启动模数转换。

注： 不应在启动ADC的同一条指令中将GO/DONE位置1。

16.2.2 转换完成

转换完成时， ADC 模块将：

- 清零GO/DONE 位
- 将ADIF 中断标志位置1
- 用新的转换结果更新ADRESH 和ADRESL 寄存器

16.2.3 终止转换

如果必须在转换完成前终止转换，可用软件将GO/DONE位清零。会用部分完成的模数转换结果更新ADRESH 和ADRESL 寄存器。未完成的位将用最后转换的位替代。

注：器件复位将强制所有寄存器进入复位状态。因此，ADC 模块被关闭，任何待处理的转换操作被终止。

16.2.4 休眠期间的 ADC 操作

ADC模块可以在休眠模式下工作。这需要将ADC时钟源设置为FRC 选项。当选择FRC时钟源时，ADC需等待一个额外的指令周期后才能启动转换。这使得可以执行SLEEP指令，这将降低转换期间的系统噪声。如果允许了ADC中断，转换完成时器件将从休眠状态唤醒。如果禁止了ADC中断，尽管ADON位仍保持置1，转换完成后ADC模块将关闭。ADC时钟源不是FRC时，尽管ADON位仍保持置1，SLEEP指令会导致当前转换中止，ADC模块被关闭。

16.2.5 特殊事件触发

CCPx/ECCPx 模块的特殊事件触发器允许定期进行ADC 转换而无需软件干预。当出现触发信号时，GO/DONE 位由硬件置1， Timer1 计数器复位为零。使用特殊事件触发器不能确保正确的ADC 时序。用户需负责确保满足ADC 时序要求。

16.2.6 A/D 转换步骤

以下是用ADC 执行模数转换的示例步骤：

1. 配置端口：
   - 禁止引脚输出驱动器（见CPIO 寄存器）
   - 将引脚配置为模拟功能（见ADINS 寄存器）
2. 配置ADC 模块：
   - 选择ADC 转换时钟
   - 配置参考电压
   - 选择ADC 输入通道
   - 开启ADC 模块
3. 配置ADC 中断（可选）：
   - 清零ADC 中断标志
   - 允许ADC 中断
   - 允许外设中断
   - 允许全局中断(1)
4. 等待所需采集时间(2)。
5. 通过将GO/DONE 位置1 启动转换。

6. 通过以下方式之一等待ADC 转换完成：
   - 查询GO/DONE 位
   - 等待ADC 中断（已允许中断）
7. 读取ADC 结果。
8. 清零ADC 中断标志（如果已允许中断则需要）。

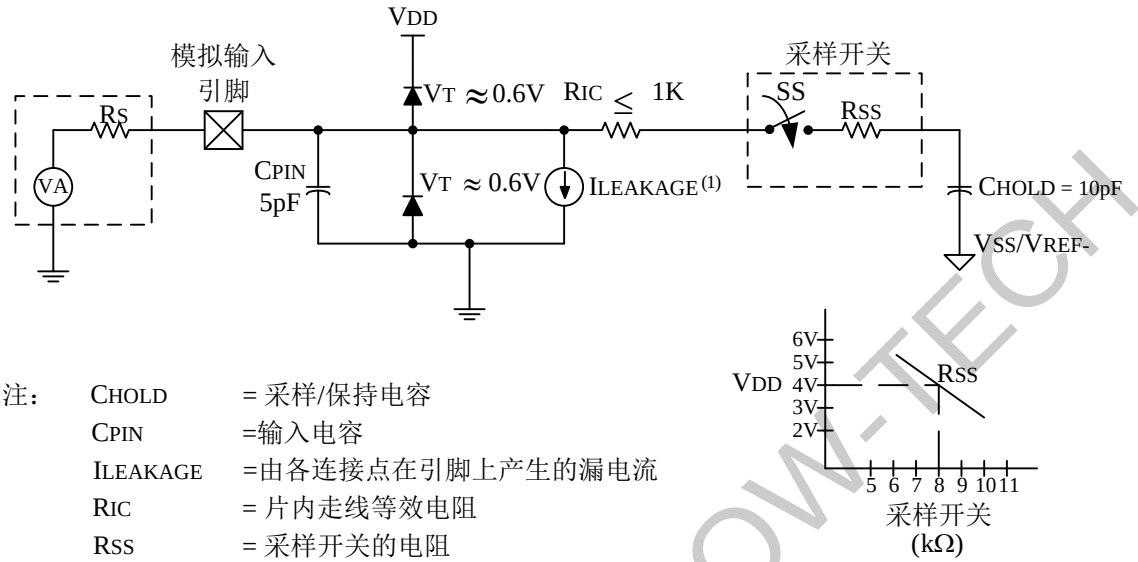
注 1： 如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止全局中断。
2： 请参见第14.3 节“A/D 采集要求”。

例16-1： A/D 转换

```
;This code block configures the ADC
;for polling, Vdd and Vss references, Frc
;clock and AN0 input.
;Conversion start & polling for completion
; are included.
;
BANKSEL     ADC0CN1          ;
LDWI        B’11110000’      ;Right justify, Frc
                               ;clock
STWR        ADC0CN1          ;Vdd and Vss Vref
BANKSEL     CPIOA            ;
BSR         CPIOA,0          ;Set PA0 to input
BANKSEL     ADINS            ;
BSR         ADINS,0          ;Set PA0 to analog
BANKSEL     ADC0CN0          ;
LDWI        B’00000001’      ;Select channel AN0
STWR        ADC0CN0          ;Turn ADC On
LCALL       SampleTime       ;Acquisiton delay
BSR         ADC0CN0,ADGO     ;Start conversion
BTSC        ADC0CN0,ADGO     ;Is conversion done?
LJUMP       $-1              ;No, test again
BANKSEL     ADRESH           ;
LDR         ADRESH,W         ;Read upper 2 bits
STWR        RESULTHI         ;store in GPR space
BANKSEL     ADRESL           ;
LDR         ADRESL,W         ;Read lower 8 bits
STWR        RESULTLO         ;Store in GPR space
```

16.3 A/D 采样要求

为了使ADC 达到规定的精度，必须使充电保持电容（CHOLD）完全充电至输入通道的电平。模拟输入模型如图16-4 所示。模拟信号源阻抗（RS）和内部采样开关阻抗（RSS）直接影响电容CHOLD 的充电时间。采样开关阻抗（RSS）随器件电压（VDD）的变化而变化，参见图16-4。模拟信号源的最大阻抗推荐值为10 k 。采集时间可能随着源阻抗的降低而缩短。在选择（或改变）模拟输入通道后，必须在启动转换前完成A/D 采集。



图注：

CHOLD	= 采样/保持电容
CPIN	=输入电容
ILEAKAGE	=由各连接点在引脚上产生的漏电流
RIC	= 片内走线等效电阻
RSS	= 采样开关的电阻
SS	= 采样开关
VT	= 阈值电压

图16-4： 模拟输入模型

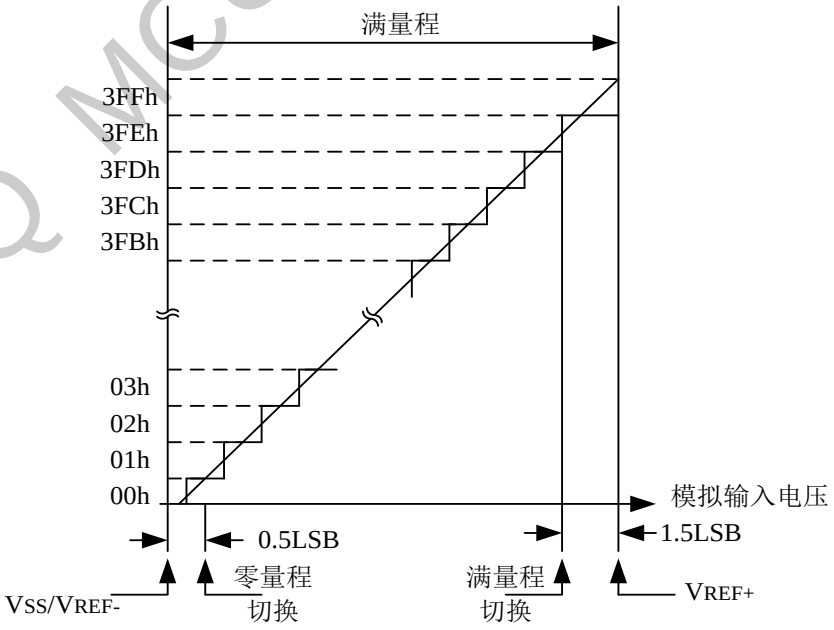


图16-5： ADC 传递函数

16.3 寄存器说明

寄存器9DH：A/D控制寄存器0（ADC0CN0）

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	**CHS4**	**CHS3**	**CHS2**	**CHS1**	**CHS0**	**GO/DONE**	**ADON**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 未实现：读为0

bit6-2 **CHS<4:0>**：模拟通道选择位

00000 = AN0
00001 = AN1
00010 = AN2
00011 = AN3
00100 = AN4
00101 = AN5
00110 = AN6
00111 = AN7
01001 = 保留。不连接任何通道。
•
•
•
11101 = 保留。不连接任何通道。
11101 = 温度指示器
11110 = DAC 输出
11111 = FVR （固定参考电压）缓冲区1 输出

bit1 **GO/DONE**：A/D 转换状态位

1 = A/D 转换正在进行。将该位置1 可启动A/D 转换周期。A/D 转换完成后，该位由硬件自动清零。
0 = A/D 转换已完成/ 未进行

bit0 **ADON**：ADC 使能位

1 = 使能ADC
0 = 禁止ADC，不消耗工作电流

寄存器9EH：A/D控制寄存器1（ADC0CN1）

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
ADFM	**ADCS2**	**ADCS1**	**ADCS0**	—	—	**ADPREF1**	**ADPREF0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **ADFM**：A/D 结果格式选择位

1 = 右对齐。当装入转换结果时， ADRESH 的高6 位设置为0。
0 = 左对齐。当装入转换结果时， ADRESL 的低6 位设置为0。

bit6-4 **ADCS<2:0>**：A/D 转换时钟选择位
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
011 = FRC （由专用RC 振荡器提供的时钟）
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64
111 = FRC （由专用RC 振荡器提供的时钟）

bit3-2 未实现：读为0

bit0 **ADPREF<1:0>**：A/D 正参考电压配置位
00 = VREF+ 连接到AVDD
01 = 保留
10 = VREF+ 连接到外部VREF+(1)
11 = VREF+ 连接到内部固定参考电压（FVR）模块(1)

注1 当选择FVR 或VREF+ 引脚作为正参考电压源时，注意存在一个最小电压规范。

寄存器9CH：ADC结果寄存器高字节（ADRESH）ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-0 **ADRESH <9:2>**：ADC 结果寄存器位
10 位转换结果的高8 位

寄存器9BH：ADC结果寄存器低字节（ADRESL）ADFM = 0

R/W-x	R/W-x	U-0	U-0	U-0	U-0	U-0	U-0
		—	—	—	—	—	—
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 **ADRESL <1:0>**：ADC 结果寄存器位
10 位转换结果的低2 位

bit5-0 未实现：读为0

寄存器9CH：ADC结果寄存器高字节（ADRESH）ADFM = 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x
—	—	—	—	—	—		
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-2 未实现：读为0

bit1-0 **ADRESH <9:8>**：ADC 结果寄存器位
10 位转换结果的高2 位

寄存器9BH：ADC结果寄存器低字节（ADRESL）ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
bit7							bit0

图注：
R = 可读位　W = 可写位　U = 未实现位，读为0
-n = POR时的值　1 = 置1　0 = 清零　x = 未知

bit7-0 **ADRESL <7:6>**：ADC 结果寄存器位
10 位转换结果的低8 位

表 16-2：与 ADC 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADC0CN0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	x000 0000	x000 0000
ADC0CN1	ADFM	ADCS2	ADCS1	ADCS0	—	—	ADPREF1	ADPREF0	0000 xx00	0000 xx00
ADRESH	ADC 结果寄存器高字节								xxxx xxxx	qqqq qqqq
ADRESL	ADC 结果寄存器低字节								xxxx xxxx	qqqq qqqq
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	xxx1 x111	xxx1 x111
ADINSC	—	—	—	—	AN7	AN6	AN5	AN4	xxxx 1111	xxxx 1111
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	0q00 0000	0q00 0000
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS1	DACPSS0	—	—	000x 00xx	000x 00xx
DACCON1	—	—	—	DACR4	DACR3	DACR2	DACR1	DACR0	xxx0 0000	xxx0 0000
INTS	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	0000 000x	0000 000q
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xx11 x111	xx11 x111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	xx11 1111	xx11 1111

图注： u = 不变， x = 未知， — = 未实现位，读为 0， q = 取值视具体情况而定。

17.0 数模转换器（DAC）模块

数模转换器提供了一个可变参考电压，它与输入源成比例，具有32 个可选输出电压。

DAC 的输入可以连接到：

- 外部VREF 引脚
- VDD 供电电压
- FVR （固定参考电压）

DAC 的输出可以配置为向以下对象提供参考电压：

- 比较器的同相输入
- ADC 输入通道
- DACOUT 引脚

数模转换器（DAC）可以通过将DACCON0 寄存器的DACEN 位置1 来使能。

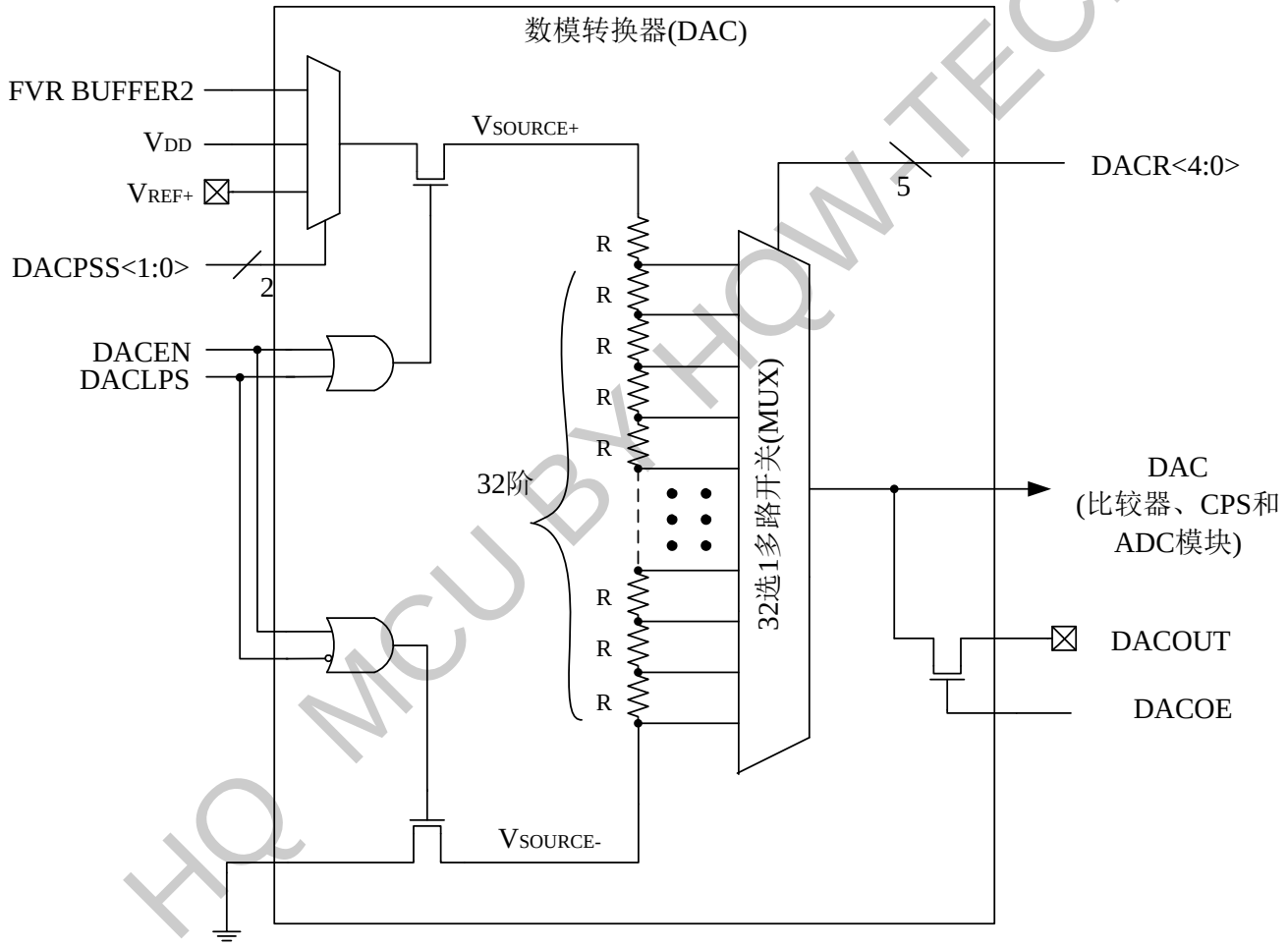


图17-1：数模转换器框图

17.1 输出电压选择

DAC 具有32个电平范围。32个电平通过DACCON1寄存器的DACR<4:0> 位进行设置。
DAC 输出电压由以下公式17-1确定：

公式17-1： DAC 输出电压

如果 *DACEN=1*

如果 *DACEN=0&DACLPS = 1 & DACR[4:0] = 11111*

如果 *DACEN=0&DACLPS = 0 & DACR[4:0] = 00000*

17.2 比例输出输出电压

DAC 输出值通过使用一个梯形电阻网络产生，梯形电阻网络的每一端分别与正参考电压和负参考电压输入源连接。如果任一输入源的电压发生波动，DAC 输出值中会产生类似的波动。

17.3 DAC 参考电压输出

可以通过将DACCON0 寄存器的DACOE 位设置为1，将DAC 输出到DACOUT 引脚。选择将DAC 参考电压输出到DACOUT引脚会自动改写数字输出缓冲器和该引脚的数字输入阈值检测器功能。当DACOUT引脚已被配置为DAC 参考电压输出时，读取该引脚将总是返回0。要提高电流驱动能力，DAC 参考电压输出端DACOUT必须外接缓冲器。图17-2 举例说明了这一缓冲技术。

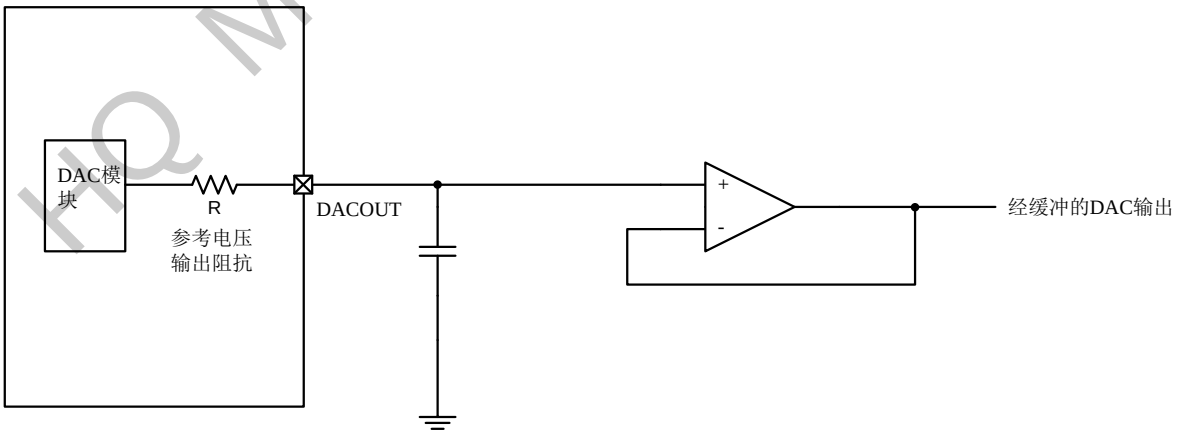


图17-2： 参考电压输出缓冲器示例

17.4 功耗电压状态

为了让DAC模块的功耗降至最低，必须将梯形电阻网络的两个参考电压输入源之一断开。禁止正电压源（$V_{SOURCE}+$）或负电压源（$V_{SOURCE}-$）都是可以的。负电压源通过将DACCON0 寄存器中的DACLPS 位置1来禁止。而将DACCON0 寄存器中的DACLPS 清零则可以禁止正电压源。

17.4.1 输出钳位至正电源

通过执行以下操作，可以将DAC 输出电压设置为$V_{SOURCE}+$，使功耗降至最低：

- 将DACCON0 寄存器中的DACEN 位清零。
- 将DACCON0 寄存器中的DACLPS 位置1。
- 将DACPSS 位配置为适当的正电压源。
- 将DACCON1 寄存器中的DACR<4:0> 位配置为11111。

这种方法也可用于将FVR 的电压输出到输出引脚上。更多信息，请参见图17-2。
关于输出钳位示例，请参见图17-3。

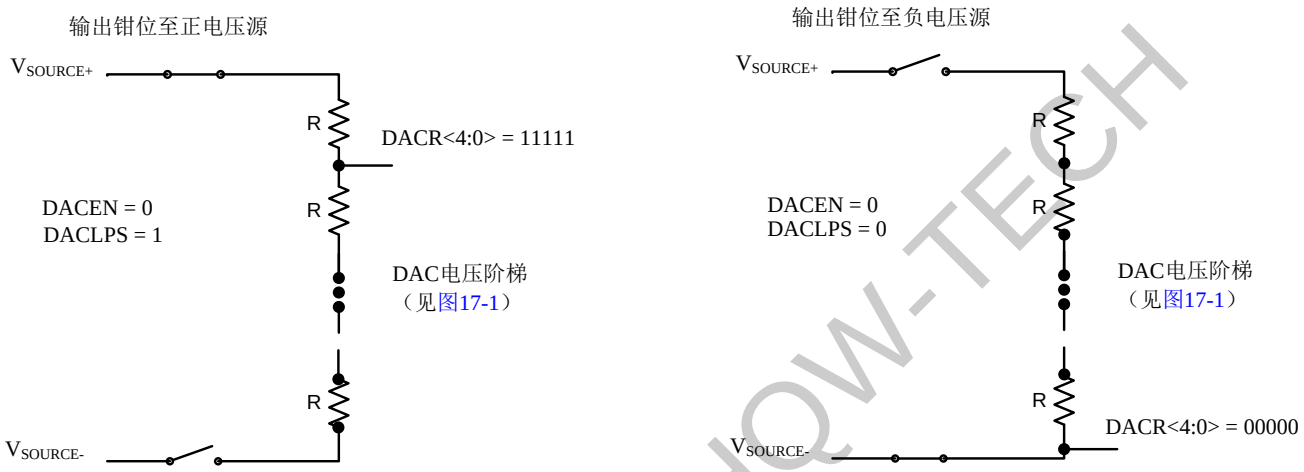


图17-3： 输出电压钳位示例

17.4.2 输出钳位至负电源

通过执行以下操作，可以将DAC 输出电压设置为$V_{SOURCE}-$，使功耗降至最低：

- 将DACCON0 寄存器中的DACEN 位清零。
- 将DACCON0 寄存器中的DACLPS 位清零。
- 将DACNSS 位配置为适当的负电压源。
- 将DACCON1 寄存器中的DACR<4:0> 位配置为00000。

这使得比较器可以检测到过零点，且不额外消耗流经DAC 模块的电流。
关于输出钳位示例，请参见图17-3。

17.5 休眠期间的操作

如果因中断或看门狗定时器超时将器件从休眠模式唤醒，DACCON0 寄存器的内容将不受影响。为了最大程度降低休眠模式下的电流消耗，应禁止参考电压模块。

17.6 复位影响

器件复位会产生以下影响：

- DAC 被禁止。
- DAC 输出电压从DACOUT 引脚上被移除。
- DACR<4:0> 范围选择位被清零。

17.7 寄存器说明

寄存器118H：参考电压控制寄存器0（DACCON0）

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0
DACEN	**DACLPS**	**DACOE**	—	**DACPSS1**	**DACPSS0**	—	—
bit7							bit0

图注：
R= 可读位　W= 可写位　U= 未实现位，读为0
-n=POR时的值　1= 置1　0= 清零　x= 未知

bit7 **DACEN**：DAC 使能位
1 = 使能DAC
0 = 禁止DAC
bit6 **DACLPS**：DAC 低功耗电压状态选择位
1 = 选择DAC 正参考电压源
0 = 选择DAC 负参考电压源
bit5 **DACOE**：DAC 电压输出使能位
1 = DAC 电平也从DACOUT 引脚输出
0 = DAC 电平从DACOUT 引脚断开
bit4 未实现：读为0
bit3-2 **DACPSS<1:0>**：DAC 正参考电压源选择位
00 = VDD
01 = VREF+ 引脚
10 = FVR 缓冲区2 输出
11 = 保留，未使用
bit1-0 未实现：读为0

寄存器119H：参考电压控制寄存器1（DACCON1）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	**DACR4**	**DACR3**	**DACR2**	**DACR1**	**DACR0**
bit7							bit0

图注：
R = 可读位　W = 可写位　U = 未实现位，读为0
-n = POR时的值　1 = 置1　0 = 清零　x = 未知

bit7-5 未实现：读为0
bit4-0 **DACR<4:0>**：DAC 电压输出选择位

表 17-1：与 DAC 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR1	ADFVR0	0q00 0000	0q00 0000
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS1	DACPSS0	—	—	000x 00xx	000x 00xx
DACCON1	—	—	—	DACR4	DACR3	DACR2	DACR1	DACR0	xxx0 0000	xxx0 0000

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

18.0 SR 锁存器

模块由单个SR 锁存器组成，该锁存器具有多个置1和复位输入，以及独立的锁存器输出。SR 锁存器模块具有以下特性：

- 可编程输入选择
- 有外部SR 锁存器输出
- 独立的Q 和$\overline{Q}$输出
- 固件置1 和复位

SR 锁存器可用于各种模拟应用，包括振荡器电路、单次电路、滞后控制器和模拟计时应用。

18.1 锁存器操作

锁存器是不依赖于时钟源的置1- 复位锁存器。每个置1和复位输入均为高电平有效。锁存器可以通过以下方式置1 或复位：

- 软件控制（SRPS 和SRPR 位）
- 比较器C1 的输出（SYNCC1OUT）
- 比较器C2 的输出（SYNCC2OUT）
- SRI 引脚
- 可编程时钟（SRCLK）

SRCON0 寄存器的SRPS 和SRPR 位可以分别用于置1或复位SR 锁存器。锁存器是复位优先型的。因此，如果置1 和复位输入同时为高电平，则锁存器将进入复位状态。SRPS 和SRPR 位都是自复位的，也就是说，对两个位中的任一个位执行一次写操作是完成锁存器置1或复位操作的必要条件。

比较器C1 或C2 的输出可以用作SR 锁存器的置1 或复位输入。其中任一比较器的输出都可以与Timer1 时钟源进行同步。更多信息，请参见第19.0 节“ 比较器模块”和第21.0 节“ 带门控控制的Timer1 模块”。

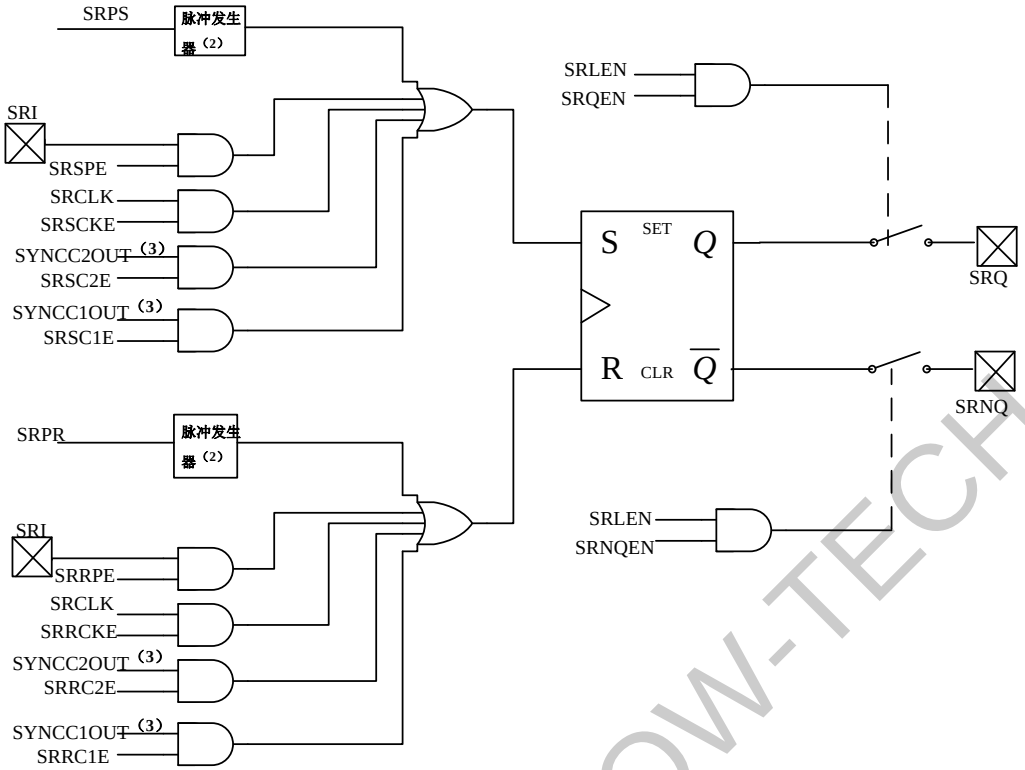
SRI 引脚上的外部源可以用作SR 锁存器的置1 或复位输入。内部时钟源可用于定期置1或复位SR锁存器。SRCON0寄存器中的SRCLK<2:0> 位用于选择时钟源周期。SRCON1 寄存器的SRSCKE 和SRRCKE位分别用于使能时钟源置1 或复位SR 锁存器。

18.2 锁存器输出

SRCON0寄存器的SRQEN和SRNQEN位用于控制Q 和$\overline{Q}$锁存器输出。SR锁存器的两个输出可以同时直接输出到I/O引脚。（必须清零相应端口对应的CPIO位以使能端口引脚输出驱动器。）

18.3 复位影响

在发生任何器件复位时， SR 锁存器输出不会初始化为某个已知状态。用户的固件负责在使能输出引脚之前初始化锁存器输出。



注 1： 如果R = 1 且S = 1， 则Q = 0， Q_N = 1.
2： 脉冲发生器产生1个Q状态脉冲宽度.
3： 名称表示比较器输出的连接点.

图18-1： SR 锁存器的简化框图

表18-1： SRCLK 频率表

SRCLK	分频比	FOSC= 16MHz	FOSC = 4MHz	FOSC = 1MHz
111	512	31.3 kHz	7.81 kHz	1.95 kHz
110	256	62.5 kHz	15.6 kHz	3.90 kHz
101	128	125 kHz	31.25 kHz	7.81 kHz
100	64	250 kHz	62.5 kHz	15.6 kHz
011	32	500 kHz	125 kHz	31.3 kHz
010	16	1 MHz	250 kHz	62.5 kHz
001	8	2 MHz	500 kHz	125 kHz
000	4	4 MHz	1 MHz	250 kHz

18.4 寄存器说明

寄存器11AH：SR锁存器控制寄存器0（SRCON0）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/S-0	R/S-0
SRLEN	**SRCLK2**	**SRCLK1**	**SRCLK0**	**SRQEN**	**SRNQEN**	**SRPS**	**SRPR**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
S = 只可置1	1 = 置1	0 = 清零	x = 未知

bit7 **SRLEN**：SR 锁存器使能位

1 = 使能SR 锁存器
0 = 禁止SR 锁存器

bit6-4 **DACPSS<2:0>**：SR 锁存器时钟分频比位
000 = 每4 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲
001 = 每8 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲
010 = 每16 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲
011 = 每32 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲
100 = 每64 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲
101 = 每128 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲
110 = 每256 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲
111 = 每512 个FOSC 周期时钟产生1 个FOSC 宽度的脉冲

Bit3 **SRQEN**：SR 锁存器Q 输出使能位
如果SRLEN = 1：
1 = Q 出现在SRQ 引脚
0 = 禁止外部Q 输出
如果SRLEN = 0：
禁止SR 锁存器

Bit2 **SRNQEN**：SR 锁存器$\overline{Q}$ 输出使能位
如果SRLEN = 1：
1 = $\overline{Q}$ 出现在SRnQ 引脚
0 = 禁止外部Q 输出
如果SRLEN = 0：
禁止SR 锁存器

Bit1 **SRPS**：使SR 锁存器置1 的脉冲输入位
1 = 为置1 输入提供1 个Q 时钟周期的脉冲
0 = 对置1 输入没有影响

bit0 **SRPR**：使SR 锁存器复位的脉冲输入位
1 = 为复位输入提供1 个Q 时钟周期的脉冲
0 = 对复位输入没有影响

寄存器11BH：SR锁存器控制寄存器1（SRCON1）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SRSPE	**SRSCKE**	**SRSC2E**	**SRSC1E**	**SRRPE**	**SRRCKE**	**SRPC2E**	**SRPC1E**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
S = 只可置1	1 = 置1	0 = 清零	x = 未知

bit7 **SRSPE**：SR 锁存器外设置1 使能位
1 = 当SRI 引脚为高电平时， SR 锁存器被置1
0 = SRI 引脚对SR 锁存器的置1 输入没有影响

Bit6 **SRSPE**：SR 锁存器置1 时钟使能位
1 = SRCLK 为SR 锁存器的置1 输入提供脉冲
0 = SRCLK 对SR 锁存器的置1 输入没有影响

Bit5 **SRSC2E**：SR 锁存器C2 置1 使能位
1 = 当比较器C2 的输出为高电平时， SR 锁存器被置1

0 = 比较器C2 的输出对SR 锁存器的置1 输入没有影响

Bit4 **SRSC1E**：SR 锁存器C1 置1 使能位
1 = 当比较器C1 的输出为高电平时， SR 锁存器被置1
0 = 比较器C1 的输出对SR 锁存器的置1 输入没有影响

Bit3 **SRRPE**：SR 锁存器外设复位使能位
1 = 当SRI 引脚为高电平时， SR 锁存器被复位
0 = SRI 引脚对SR 锁存器的复位输入没有影响

Bit2 **SRRCKE**：SR 锁存器复位时钟使能位
1 = SRCLK 为SR 锁存器的复位输入提供脉冲
0 = SRCLK 对SR 锁存器的复位输入没有影响

Bit1 **SRPC2E**：SR 锁存器C2 复位使能位
1 = 当比较器C2 的输出为高电平时， SR 锁存器被复位
0 = 比较器C2 的输出对SR 锁存器的复位输入没有影响

bit0 **SRPC1E**：SR 锁存器C1 复位使能位
1 = 当比较器C1 的输出为高电平时， SR 锁存器被复位
0 = 比较器C1 的输出对SR 锁存器的复位输入没有影响

表 18-1：与 SR 锁存器有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	xxx1 x111	xxx1 x111
SRCON0	SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR	0000 0000	0000 0000
SRCON1	SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRPC2E	SRPC1E	0000 0000	0000 0000
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xx11 x111	xx11 x111

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

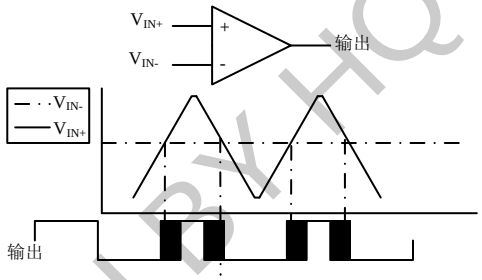
19.0 比较器模块

比较器模块通过比较两个模拟电压并提供其相对幅值的数字表示，用于建立模拟电路与数字电路的接口。比较器是非常有用的混合信号模块，因为它们提供了与程序执行相独立的模拟功能。模拟比较器模块具有以下特性：

- 独立的比较器控制
- 可编程输入选择
- 有内部/ 外部比较器输出
- 可编程输出极性
- 电平变化中断
- 从休眠状态唤醒
- 可编程的速度/ 功耗优化
- PWM 关闭
- 可编程和固定参考电压

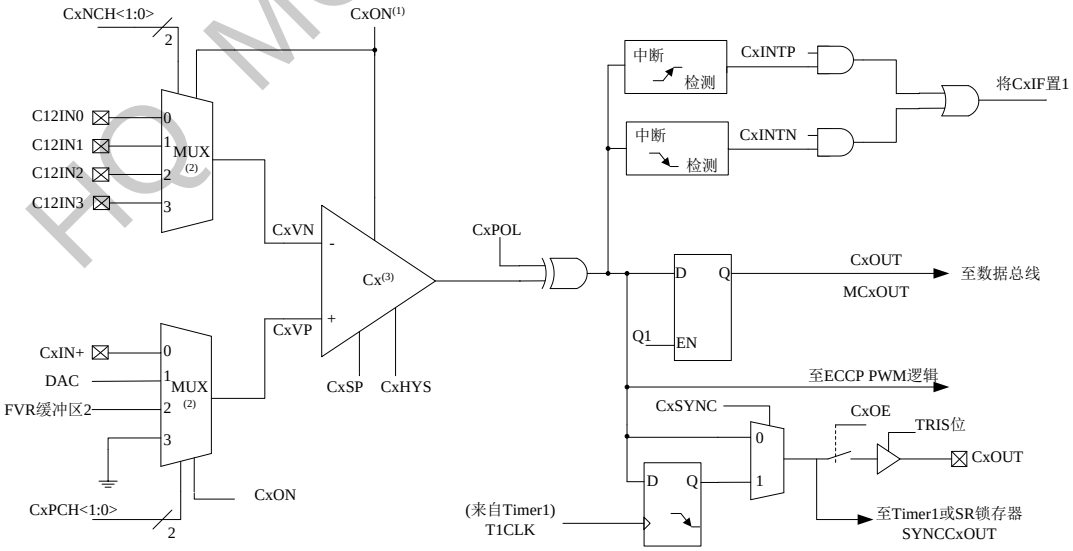
19.1 比较器概述

图19-1所示为单比较器以及模拟输入电平与数字输出之间的关系。当VIN+ 上的模拟电压小于VIN- 上的模拟电压时，比较器输出为数字低电平。当VIN+ 上的模拟电压大于VIN- 上的模拟电压时，比较器输出为数字高电平。



注：比较器输出的黑色区域表示因输入失调电压和响应时间所造成的输出不确定区域

图19-1： 单比较器



注： 1：当CxON = 0时，比较器将产生0输出。
2：当CxON = 0时，所有多路开关输入都会被切断。
3：在调试期间可以冻结比较器输出。

图19-2： 比较器1 和2 模块的简化框图

19.2 比较器控制

每个比较器都具有2 个控制寄存器：CMxCON0 和CMxCON1。

CMxCON0 寄存器包含以下控制和状态位：

- 使能
- 输出选择
- 输出极性
- 速度/ 功耗选择
- 滞后使能
- 输出同步

CMxCON1 寄存器包含以下控制位：

- 中断允许
- 中断边沿极性
- 同相输入通道选择
- 反相输入通道选择

19.2.1 比较器使能

将CMxCON0 寄存器的CxON 位置1 可以使能比较器操作。清零CxON 位可以禁止比较器，以使电流消耗降至最低。

19.2.2 比较器输出选择

可以通过读CMxCON0 寄存器的CxOUT 位或CMOUT寄存器的MCxOUT 位监视比较器的输出。为了使输出可用于外部连接，必须满足以下条件：

- 必须将CMxCON0 寄存器的CxOE 位置1
- 必须清零相应的CPIO 位
- 必须将CMxCON0 寄存器的CxON 位置1

注 1：CMxCON0 寄存器的 CxOE 位会改写端口数据锁存器。将 CMxCON0 寄存器的 CxON 位置 1 对端口改写没有影响。
2：比较器的内部输出在每个指令周期被锁存。除非另外指定，否则不锁存外部输出。

将比较器的输出反相在功能上等效于交换比较器输入。可以通过将CMxCON0 寄存器的CxPOL 位置1 来使比较器输出的极性反相。清零CxPOL 位得到的是未反相的输出信号。

表19-1 给出了输出状态与输入条件的关系（包括极性控制）。

表19-1： 比较器输出状态与输入条件

输入条件	CxPOL	CxOUT
$CxV_N > CxV_P$	0	0
$CxV_N < CxV_P$	0	1
$CxV_N > CxV_P$	1	1
$CxV_N < CxV_P$	1	0

19.2.3 比较器速度/功耗选择

在程序执行期间通过CxSP 控制位可以最佳地权衡速度与功耗。该位的默认状态为1，选择正常速度模式。器件功耗可以通过将CxSP 位清零进行优化，代价是比较器传输延时变长。

19.3 比较器滞后

通过在每个比较器的输入引脚上加上一个可选的分离电压量，可以为整体操作提供滞后功能。滞后功能通过将CMxCON0 寄存器的CxHYS 位置1 来使能。

19.4 Timer1 门控操作

比较器操作产生的输出可以用作Timer1 的门控源。该功能可用于对模拟事件的持续时间或间隔时间进行计时。建议将比较器输出与Timer1 进行同步。这可以确保在比较器中发生变化时， Timer1 不会递增。

19.4.1 比较器输出同步

通过将CMxCON0 寄存器的CxSYNC 位置1，可以使比较器C1 或C2 的输出与Timer1 保持同步。使能比较器的输出时，比较器的输出在Timer1 时钟源的下降沿被锁存。如果Timer1 使用了预分频器，则比较器的输出在经过预分频后被锁存。为了防止发生竞争，比较器的输出在Timer1时钟源的下降沿被锁存，而Timer1在其时钟源的上升沿递增。更多信息，请参见比较器框图（图19-2）和Timer1 框图（图21-1）。

19.5 比较器中断

比较器可以在输出值发生改变时产生中断；对于每个比较器，都提供了上升沿检测器和下降沿检测器。当触发任一边沿检测器时，如果它关联的允许位已置1（CMxCON1 寄存器的CxINTP 和/ 或CxINTN 位），则相应的中断标志位（PIFB2 寄存器的CxIF 位）会置1。要允许中断，必须将以下位置1：

- CMxCON0 寄存器的CxON、CxPOL 和CxSP 位
- PIEB2 寄存器的CxIE 位
- CMxCON1 寄存器的CxINTP 位（对于上升沿检测）
- CMxCON1 寄存器的CxINTN 位（对于下降沿检测）
- INTS 寄存器的PEIE 和GIE 位

关联的中断标志位（PIFB2 寄存器的CxIF 位）必须用软件清零。如果在清零该标志时检测到另一个边沿，则标志仍然会在序列结束时置1。

> 注： 即使比较器被禁止，还是可以通过使用CMxCON0寄存器的CxPOL位更改输出极性来产生中断，或者通过使用CMxCON0寄存器的CxON位开启或关闭比较器来产生中断。

19.6 比较器同相输入选择

通过配置CMxCON1 寄存器的CxPCH<1:0> 位，将内部参考电压或模拟引脚连接到比较器的同相输入。

- C1IN+ 或C2IN+ 模拟引脚
- DAC
- FVR （固定参考电压）
- VSS （地）

每当禁止比较器（CxON = 0）时，所有比较器输入都会被禁止。

19.7 比较器反相输入选择

CMxCON0 寄存器的CxNCH<1:0> 位指示4 个模拟引脚中的一个连接到比较器的反相输入。

> 注：要将CxIN+ 和CxINx- 引脚用作模拟输入，必须将ADINS 寄存器中的相应位置1，同时也必须将相应的CPIO位置1来禁止输出驱动器。

19.8 比较器响应时间

在改变输入源或选择新的参考电压后，一段时间内比较器的输出状态都是不确定的。这段时间被称为响应时间。比较器的响应时间不同于参考电压的稳定时间。因此，在确定比较器输入改变的总响应时间时，必须考虑这两个时间。

19.9 与 ECCP 逻辑交互

C1和C2比较器可以用作通用比较器。它们的输出可以送到C1OUT和C2OUT引脚上。当ECCP自动关闭有效时，它可以使用一个比较器信号，也可以同时使用两个比较器信号。如果同时还使能了自动重启，则可以将比较器配置为ECCP的闭环模拟反馈，从而构成一个模拟控制PWM。.

注：当第一次初始化比较器模块时，输出状态是未知的。初始化后，用户在可靠使用结果（主要是在使用与其他外设特性（例如，ECCP 自动关断模式）有关的结果时）之前，应先验证比较器的输出状态。

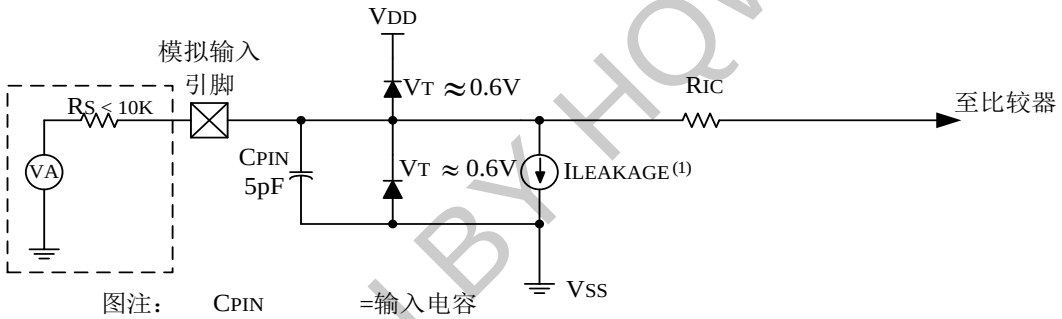
19.10 模拟输入连接注意事项

模拟输入的简化电路如图19-3 所示。由于模拟输入引脚与数字输入共用连接，它们在VDD 和VSS 之间连有反向偏置的ESD 保护二极管。因此，模拟输入必须在VSS和VDD 之间。如果输入电压与这一范围偏离的绝对值超过0.6V，就可能发生一个二极管正向导通，从而可能导致锁死发生。

模拟信号源的最大阻抗推荐值为10 k。任何连接到模拟输入引脚的外部元件（如电容或齐纳二极管），应保证其泄漏电流极小以使引入的误差降至最低。

注 1： 读端口寄存器时，所有配置为模拟输入的引脚均读为0。配置为数字输入的引脚将根据输入规范转换为模拟输入。

2： 定义为数字输入引脚上的模拟电平可能会使输入缓冲器的电流消耗超过规定值。



图注： CPIN =输入电容
ILEAKAGE =由各连接点在引脚上产生的泄漏电流
RIC = 片内走线等效电阻
RS = 信号源阻抗
VA = 模拟电压
VT = 阈值电压

注 1：请参见第29.0节 “电气规范”

图19-3： 模拟输入模型

19.11 寄存器说明

寄存器111H/113H：比较器Cx控制寄存器0（CMxCON0）

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
CxON	**CxOUT**	**CxOE**	**CxPOL**	—	**CxSP**	**CxHYS**	**CxSYNC**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **CxON**：比较器使能位
1 = 使能比较器，并且比较器不消耗有功功率
0 = 禁止比较器

bit6 **CxOUT**：比较器输出位
如果CxPOL = 1 （极性反相）：
1 = CxVP < CxVN
0 = CxVP > CxVN
如果CxPOL = 0 （极性不反相）：
1 = CxVP > CxVN
0 = CxVP < CxVN

bit5 **CxOE**：比较器输出使能位
1 = CxOUT 出现在CxOUT 引脚。只有关联的CPIO 位清零时才能实际驱动引脚。
0 = CxOUT 仅在内部有效

bit4 **CxPOL**：比较器输出极性选择位
1 = 比较器输出反相
0 = 比较器输出不反相

bit3 未实现：读为0

bit2 **CxSP**：比较器速度/ 功耗选择位
1 = 比较器工作在正常功耗、高速模式下
0 = 比较器工作在低功耗、低速模式下

bit1 **CxHYS**：比较器滞后使能位
1 = 使能比较器滞后
0 = 禁止比较器滞后

bit0 **CxSYNC**：比较器输出同步模式位
1 = 送到Timer1 和I/O 引脚的比较器输出与Timer1 时钟源的变化进行同步。输出在Timer1 时钟源的下降沿进行更新。
0 = 送到Timer1 和I/O 引脚的比较器输出是异步的。

寄存器112H/114H：比较器Cx控制寄存器1（CMxCON1）

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
CxINTP	**CxINTN**	**CxPCH1**	**CxPCH0**	**—**	**—**	**CxNCH1**	**CxNCH0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **CxINTP**：比较器正向边沿中断允许位
1 = 在CxOUT 位的正向边沿， CxIF 中断标志将置1
0 = 在CxOUT 位的正向边沿， CxIF 中断标志不会置1

bit6 **CxINTN**：比较器负向边沿中断允许位
1 = 在CxOUT 位的负向边沿， CxIF 中断标志将置1
0 = 在CxOUT 位的负向边沿， CxIF 中断标志不会置1

bit5-4 **CxPCH<1:0>**：比较器同相输入通道选择位
00 = CxVP 连接到CxIN+ 引脚
01 = CxVP 连接到DAC 参考电压
10 = CxVP 连接到FVR 参考电压

bit3-2 未实现：读为0

bit1-0 **CxNCH<1:0>**：比较器反相输入通道选择位 (1)
MDT10F1822：
0 = C1VN连接到C1IN0- 引脚

MDT10F1823：
00 = CxVN 连接到C12IN0- 引脚
01 = CxVN 连接到C12IN1- 引脚
10 = CxVN 连接到C12IN2- 引脚
11 = CxVN 连接到C12IN3- 引脚

注 1：使用MDT10F1822时，CMxCON1这一寄存器的bit1未实现，读为0，bit0只能为0。

寄存器115H：比较器输出寄存器（CMOUT）

U-0	U-0	U-0	U-0	U-0	U-0	R-0	R-0
—	—	—	—	—	—	**MC2OUT**	**MC1OUT**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-2 未实现：读为0
bit1 **MC2OUT**：C2OUT 的镜像副本位
bit0 **MC1OUT**：C1OUT 的镜像副本位

表 19-1：与比较器有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	xxx1 x111	xxx1 x111
ADINSC	—	—	—	—	AN7	AN6	AN5	AN4	xxxx 1111	xxxx 1111
INTS	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	0000 000x	0000 000q
PIEB2	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	—	0000 0xxx	0000 0xxx
PIFB2	OSFIF	C2IF	C1IF	EEIF	BCL1IF	—	—	—	0000 0xxx	0000 0xxx
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xx11 x111	xx11 x111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	xx11 1111	xx11 1111
CMxCON0	CxON	CxOUT	CxOE	CxPOL	—	CxSP	CxHYS	CxSYNC	0000 x100	0000 x100
CMxCON1	CxINTP	CxINTN	CxPCH1	CxPCH0	—	—	CxNCH1	CxNCH0	0000 xx00	0000 xx00
CMOUT	—	—	—	—	—	—	C2OUT	C1OUT	xxxx xx00	xxxx xx00

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

20.0 TIMER0 模块

Timer0 模块是8 位定时器/ 计数器，具有以下特性：

- 8 位定时器/ 计数器寄存器（TMR0）
- 8 位预分频器（独立于看门狗定时器）
- 可编程内部或外部时钟源
- 可编程外部时钟边沿选择
- 上溢时产生中断
- TMR0 可用于门控Timer1

图20-1 给出了Timer0 模块的框图。

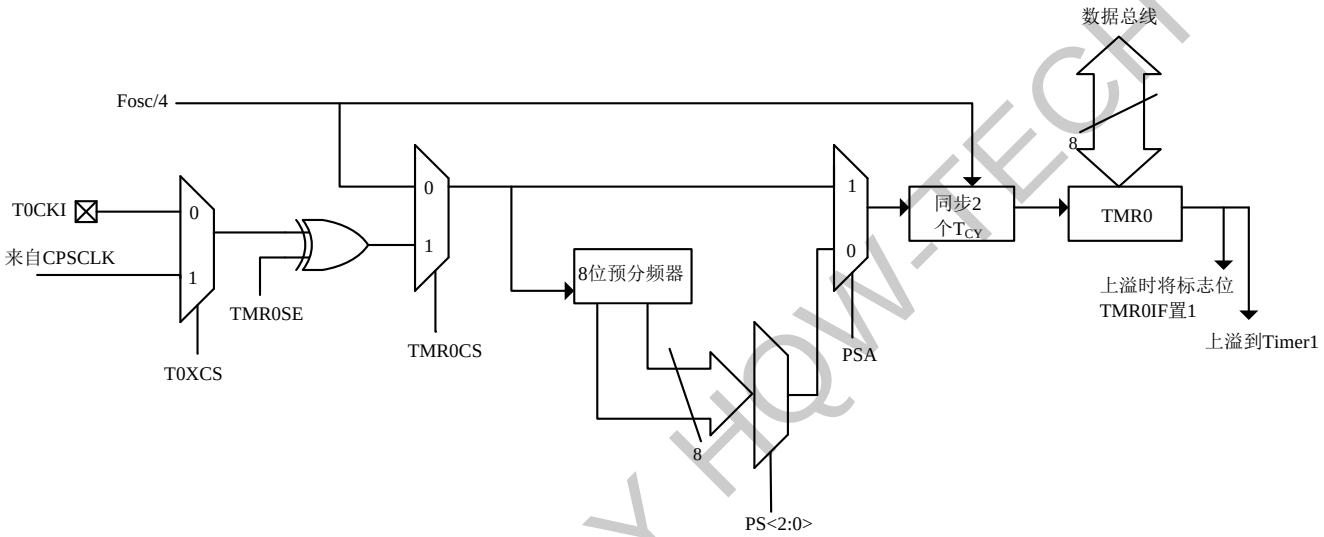


图20-1： TIMER0 框图

20.1 Timer0 工作原理

Timer0 模块可被用作8 位定时器或8 位计数器。

20.1.1 8 位定时器模式

如果在没有预分频器的情况下使用Timer0 模块，它将在每个指令周期递增。8 位定时器模式可通过清零OPTION 寄存器的TMR0CS 位选择。当写TMR0 时，紧跟写操作之后的两个指令周期内禁止TMR0 递增。

注：当写TMR0 时，考虑到存在两个指令周期的延时，可以调整写入TMR0寄存器的值。

20.1.2 8 位计时器模式

在8 位计数器模式下，Timer0 模块将在T0CKI 引脚或电容传感振荡器（CPSCLK）信号的每个上升沿或下降沿递增。使用T0CKI 引脚的8 位计数器模式，可通过将OPTION 寄存器中的TMR0CS 位设置为1 并CPSCON0 寄存器中的T0XCS 位重新设置为0 进行选择。使用电容传感振荡器（CPSCLK）信号的8 位计数器模式，可通过将OPTION 寄存器中的TMR0CS 位设置为1并将CPSCON0 寄存器中的T0XCS位设置为1进行选择。两个输入源递增边沿是上升沿还是下降沿由OPTION 寄存器中的TMR0SE 位决定。

20.1.3 软件可编程的预分频器

软件可编程的预分频器只能用于Timer0。可通过清零OPTION 寄存器的PSA 位来使能预分频器。

注：看门狗定时器（WDT）使用它自己的独立预分频器。

Timer0 模块有8 个预分频比选项，范围从1:2 至1:256。预分频值可通过OPTION 寄存器的PS<2:0> 位进行选择。为了让Timer0 模块使用1:1 预分频值，必须通过将OPTION 寄存器的PSA 位置1 来禁止预分频器。预分频器是不可读写的。写TMR0 寄存器的所有指令都会清零预分频器。

20.1.4TIMER0 中断

TMR0 寄存器从FFh 上溢到00h 时，将产生Timer0 中断。每次TMR0 寄存器上溢时都会将INTS 寄存器的TMR0IF 中断标志位置1，这与是否允许Timer0 中断无关。TMR0IF 位只能用软件清零。Timer0 中断允许位是INTS 寄存器的TMR0IE 位。

注：由于定时器在休眠状态下是停止的，所以Timer0中断无法将处理器从休眠状态唤醒。

20.1.5 8 位同步计数器模式

在8 位计数器模式下，T0CKI 引脚的递增边沿必须与指令时钟保持同步。同步可通过在指令时钟的Q2 和Q4周期对预分频器的输出进行采样实现。外部时钟源的高低电平周期必须满足第29.0节“ 电气特性” 中所示的时序要求。

20.1.6 休眠期间的操作

在处理器处于休眠模式时，Timer0 无法工作。在处理器处于休眠模式时， TMR0 寄存器的内容将保持不变。

20.2 寄存器说明

寄存器15H：定时器0数字寄存器（TMR0）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-0 **TMR0 <7:0>**：定时器0数字寄存器

寄存器95H：选项寄存器（OPTION）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WPUEN	**INTEDG**	**TMR0CS**	**TMR0SE**	**PSA**	**PS2**	**PS1**	**PS0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **WPUEN**：弱上拉使能位
1 = 禁止所有弱上拉
0 = 通过各个WPUx 锁存值使能弱上拉

bit6 **INTEDG**：中断边沿选择位
1 = PA2/INT 引脚的上升沿触发中断
0 = PA2/INT 引脚的下降沿触发中断

bit5 **TMR0CS**：Timer0 时钟源选择位
1 = PA4/T0CKI 引脚上的电平跳变
0 = 内部指令周期时钟（FOSC/4）

Bit4 **TMR0SE**：Timer0 时钟源边沿选择位
1 = T0CKI 引脚信号从高至低跳变时，递增计数
0 = T0CKI 引脚信号从低至高跳变时，递增计数

Bit3 **PSA**：预分频器分配位
1 = 预分频器未分配给Timer0 模块
0 = 预分频器分配给Timer0 模块

Bit2-0 **PS<2:0>**：预分频比选择位
000 = 预分频比1:2
001 = 预分频比1:4
010 = 预分频比1:8
011 = 预分频比1:16
100 = 预分频比1:32
101 = 预分频比1:64
110 = 预分频比1:128
111 = 预分频比1:256

表 20-1：与 TIMER0 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	0000 000x	0000 000q
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xx11 x111	xx11 x111
CPSCON0	CPSON	CPSRM	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	00xx 0000	00xx 0000
OPTION	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TMR0	TMER0 寄存器								xxxx xxxx	qqqq qqqq

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

21.0 带门控控制的 TIMER1 模块

Timer1 模块是16 位定时器/ 计数器，具有以下特性：

- 16 位定时器/ 计数器寄存器对（TMR1H:TMR1L）
- 可编程内部或外部时钟源
- 2 位预分频器
- 专用的32 kHz 振荡器电路
- 可选的同步比较器输出
- 多个Timer1 门控（计数使能）源
- 上溢时产生中断
- 上溢触发唤醒（仅限外部时钟，异步模式）
- 捕捉/ 比较功能的时基
- 特殊事件触发器（带CCP/ECCP）
- 可选择的门控源极性
- 门控翻转模式
- 门控单脉冲模式
- 门控值状态
- 门控事件中断

图21-1 给出了Timer1 模块的框图。

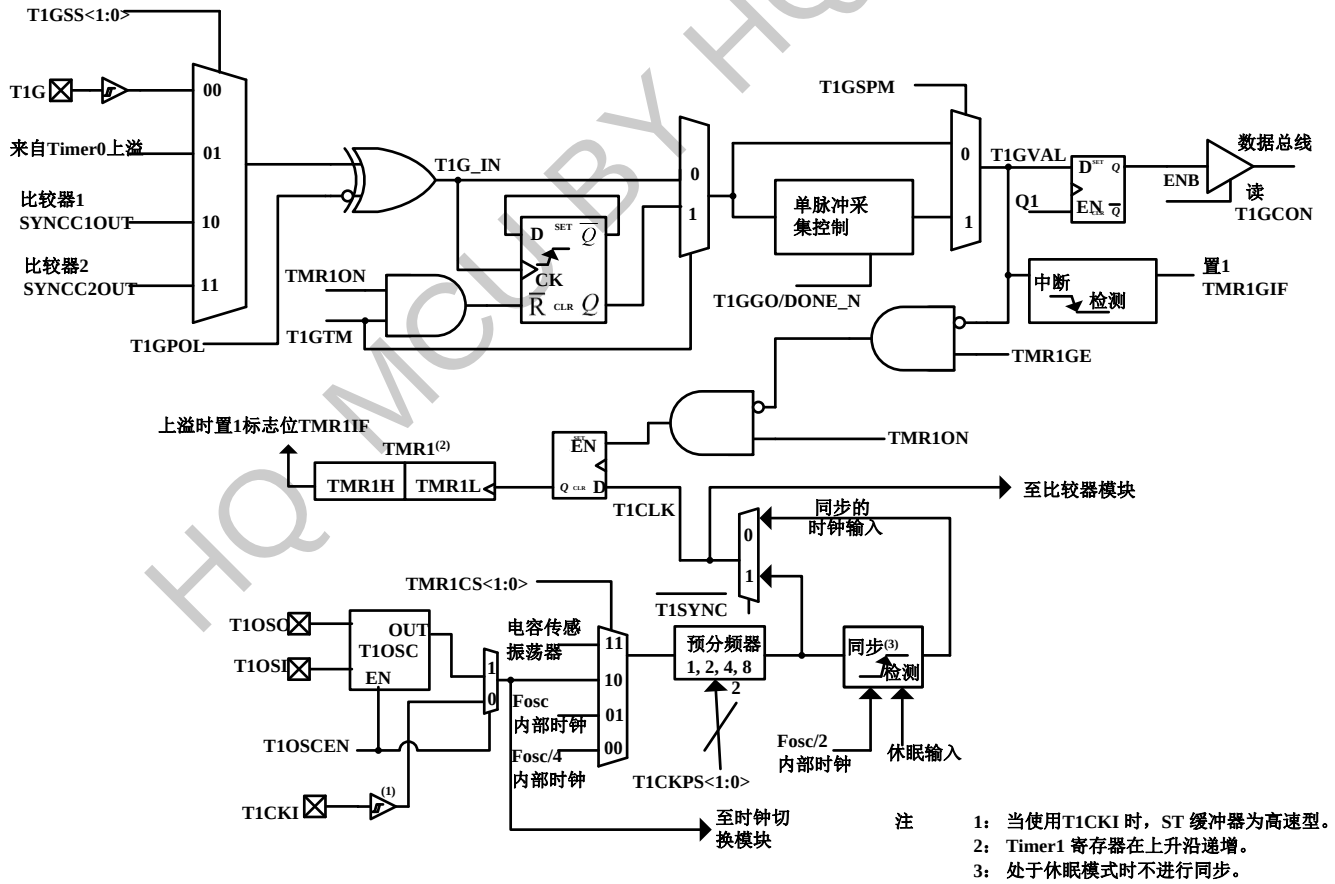


图21-1： TIMER1 框图

21.1 Timer1 工作原理

Timer1模块是16位递增计数器，可通过TMR1H:TMR1L寄存器对访问。写TMR1H或TMR1L会直接更新计数器。Timer1 与内部时钟源一起使用时，模块为定时器并在每个指令周期递增。与外部时钟源一起使用时，模块可用作定时器或计数器，在外部时钟源的每个选定边沿递增。Timer1 分别通过配置T1STA 和T1GCON 寄存器中的TMR1ON 和TMR1GE 位使能。表21-1 显示了Timer1使能选择。

表21-1： TIMER1 使能选择

TRM1ON	TMR1GE	Timer1 工作状态
0	0	关闭
0	1	关闭
1	0	总是开启
1	1	计数使能

21.2 时钟源选择

T1STA 寄存器的TMR1CS<1:0> 和T1OSCEN 位用于选择Timer1 的时钟源。表21-2 显示了时钟源选择。

表21-2： 时钟源选择

TMR1CS1	TMR1CS0	T1OSCEN	时钟源
0	1	x	系统时钟（FOSC）
0	0	x	指令时钟（FOSC/4）
1	1	x	电容传感振荡器
1	0	0	T1CKI 引脚上的外部时钟源
1	0	1	T1OSI/T1OSO 引脚上的振荡器电路

21.2.1 内部时钟源

当选择内部时钟源时，TMR1H:TMR1L 寄存器对的递增频率将为FOSC 的整数倍（取决于Timer1 预分频器）。选择FOSC 内部时钟源时，Timer1 寄存器的值将在每个指令时钟周期中递增4次。由于这个原因，在读取Timer1值时，分辨率将会出现2 LSB 的误差。为了利用Timer1的全部分辨率，必须使用异步输入信号来对Timer1 时钟输入进行门控。可以使用以下异步源：

- T1G 引脚上的异步事件用于进行Timer1 门控
- C1 或C2 比较器输入用于进行Timer1 门控

21.2.2 外部时钟源

当选择外部时钟源时，Timer1 模块可以作为定时器或计数器工作。Timer1 使能计数时，在外部时钟输入T1CKI 或电容传感振荡器信号的上升沿递增。这些外部时钟源既可以与单片机系统时钟同步，也可以异步运行。作为定时器采用时钟振荡器工作时，可以将外部32.768 kHz 晶振与专用内部振荡器电路一起使用。

注： 当中断条件发生时，无论相应中断允许位或全局中断允许位GIE （在INTS 寄存器中）的状态如何，中断标志位都将被置1。用户软件应在允许一个中断前，先将相应的中断标志位清零。

- POR 后使能Timer1
- 写入TMR1H 或TMR1L
- Timer1 被禁止
- T1CKI 为高电平时Timer1 被禁止（TMR1ON = 0），然后在T1CKI 为低电平时Timer1 被使能（TMR1ON = 1）

21.3 Timer1 预分频器

Timer1有4个预分频比选项，允许对时钟输入进行1、2、4 或8分频。T1STA寄存器的T1CKPS位控制预分频器计数器。对预分频器计数器不能直接进行读写操作；但是，通过写入TMR1H或TMR1L可将预分频器计数器清零。

21.4 Timer1 振荡器

在引脚T1OSI（输入）和T1OSO（放大器输出）之间接有一个内置专用低功耗32.768 kHz振荡器电路。该内部电路与一个外部32.768 kHz晶振联合使用。通过将T1STA寄存器的T1OSCEN位置1可使能振荡器电路。在休眠期间，振荡器将继续运行。

注： 振荡器在使用之前需要一定的起振和稳定时间。因此，T1OSCEN 应置1，且在使能Timer1 之前确保有适当的延时。

21.5 异步计数模式下 Timer1 的操作

如果T1STA寄存器的控制位T1SYNC 置1，外部时钟输入将不同步。定时器异步于内部相位时钟进行递增计数。如果选择了外部时钟源，在休眠期间定时器将继续运行，并在上溢时产生中断以唤醒处理器。但是，用软件对定时器进行读/ 写操作时，要特别当心（见第21.5.1 节“ 在异步计数器模式下读写Timer1”）。

注： 当从同步切换到异步操作时，可能会跳过一次递增。当从异步切换到同步操作时，可能会产生一次额外递增。

21.5.1 在异步计数器模式下读写 Timer1

当定时器采用外部异步时钟运行时，对TMR1H或TMR1L的读操作将确保为有效读操作（由硬件实现）。但是，应该注意的是，通过读两个8位值来读取16位定时器本身就会产生某些问题，这是因为定时器可能在两次读操作之间产生上溢。对于写操作，建议用户直接停止定时器，然后写入所需的值。如果定时器寄存器正进行递增计数，对定时器寄存器进行写操作，可能会导致写争用。这可能在TMR1H:TMR1L寄存器对中产生不可预测的值。

21.6 Timer1 门控

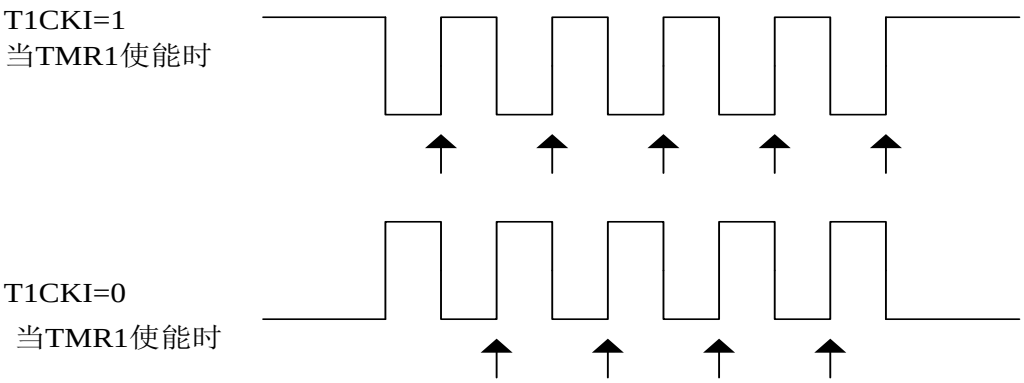
Timer1 可配置为自由计数或用Timer1 门控电路使能和禁止计数。这也称为Timer1 门控使能。Timer1 门控也可由多个可选择源驱动。

21.6.1 Timer1 门控使能

通过将T1GCON寄存器的TMR1GE位置1 使能Timer1门控使能模式。使用T1GCON寄存器的T1GPOL 位来配置Timer1 门控使能模式的极性。使能Timer1 门控使能模式时，Timer1 将在Timer1 时钟源的上升沿递增。禁止Timer1 门控使能模式时，不会发生递增，Timer1 将保持当前计数。时序详细信息请参见图21-3。

表21-3： TIMER1 门控使能选择

T1CLK	T1GPOL	T1G	Timer1 工作状态
↑	0	0	计数
↑	0	1	保存计数
↑	1	0	保存计数
↑	1	1	计数



注 **1**： 箭头指示计数器递增。
2： 在计数器模式下，计数器在首个上升沿递增之前，必须先经过一个下降沿。

图21-2： TIMER1 递增边沿

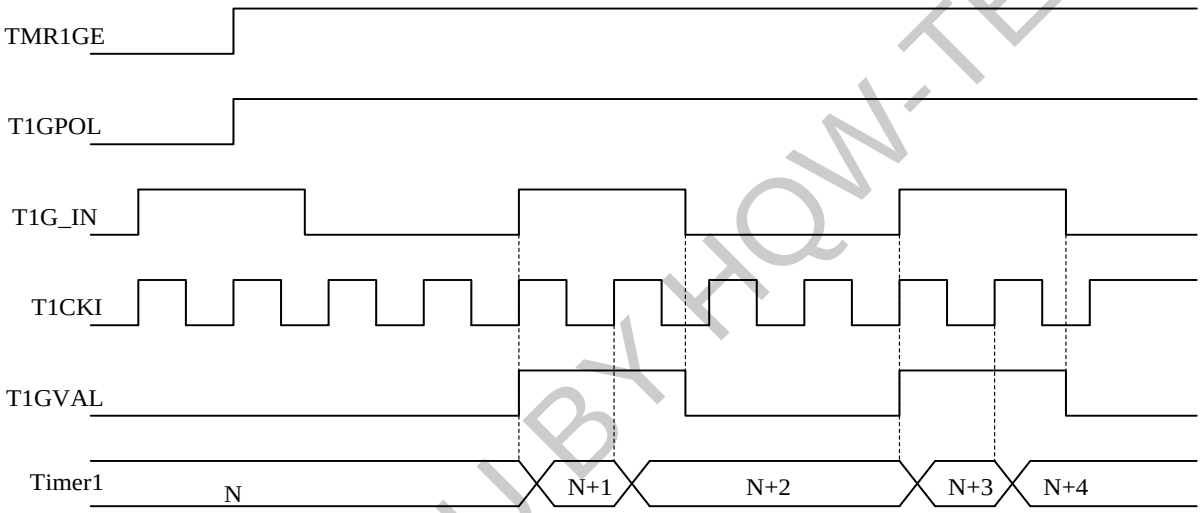


图21-3： TIMER1 门控使能模式

21.6.2Timer1 门控源选择

Timer1 门控源可从四种不同源之中选择。源的选择由T1GCON 寄存器的T1GSS 位控制。每个可用源的极性也是可选择的。极性的选择由T1GCON 寄存器的T1GPOL 位控制。

表21-4： TIMER1 门控源

T1GSS	Timer1 门控源
00	Timer1 门控引脚
01	Timer0 上溢（TMR0 从 FFh 递增到 00h）
10	比较器 1 的输出 SYNCC1OUT（可选择 Timer1 同步输出）
11	比较器 2 的输出 SYNCC2OUT（可选择 Timer1 同步输出）

21.6.2.1 T1G 门控操作

T1G 引脚是Timer1门控源之一。它可用于向Timer1门控电路提供外部源。

21.6.2.2 Timer0 上溢门控操作

Timer0 从FFh 递增到00h 时，将自动产生由低至高脉冲并在内部提供给Timer1 门控电路。

21.6.2.3 比较器 C1 门控操作

比较器1操作产生的输出可以选择作为Timer1 的门控控制源。比较器1输出（SYNCC1OUT）可以与Timer1时钟进行同步，也可以保持异步。

21.6.2.4 比较器 C2 门控操作

比较器2操作产生的输出可以选择作为Timer1 的门控控制源。比较器2输出（SYNCC2OUT）可以与Timer1时钟进行同步，也可以保持异步。

21.6.3 Timer1 门控翻转模式

使能Timer1 门控翻转模式时，可测量Timer1 门控信号整个周期的长度，而不是单电平脉冲的持续时间。Timer1 门控源经由一个单稳态触发器输送到Timer1，该单稳态触发器在信号的每个递增边沿改变状态。时序详细信息请参见图21-4。Timer1门控翻转模式通过将T1GCON寄存器的T1GTM位置1 使能。T1GTM 位清零时，将清除单稳态触发器并保持清零。这对于控制测量哪个边沿是必需的。

注：在使能翻转模式的同时改变门控极性，可能会导致不确定的操作。

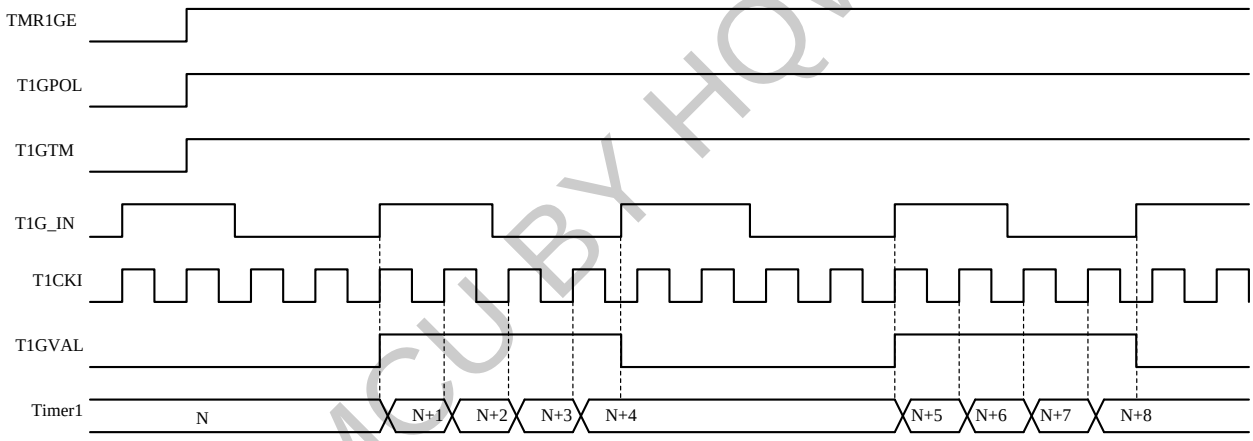


图21-4： TIMER1 门控翻转模式

21.6.4 Timer1 门控单脉冲模式

使能Timer1 门控单脉冲模式时，可能会捕捉到一个单脉冲门控事件。Timer1 门控单脉冲模式首先通过将T1GCON 寄存器中的T1GSPM 位置1 来使能。接下来必须将T1GCON 寄存器中的T1GGO/DONE 位置1。Timer1将在下一个递增边沿完全使能。在脉冲的下个后边沿，将自动清零T1GGO/DONE 位。不允许其他门控事件递增Timer1，直到T1GGO/DONE 位再次由软件置1。时序详细信息请参见图21-5。如果单脉冲门控模式通过清零T1GCON 寄存器的T1GSPM位来禁止，那么T1GGO/DONE位也会被清零。同时使能翻转模式和单脉冲模式将允许两部分协同工作。这样就可以测量Timer1 门控源的周期时间。时序详细信息请参见图21-6。

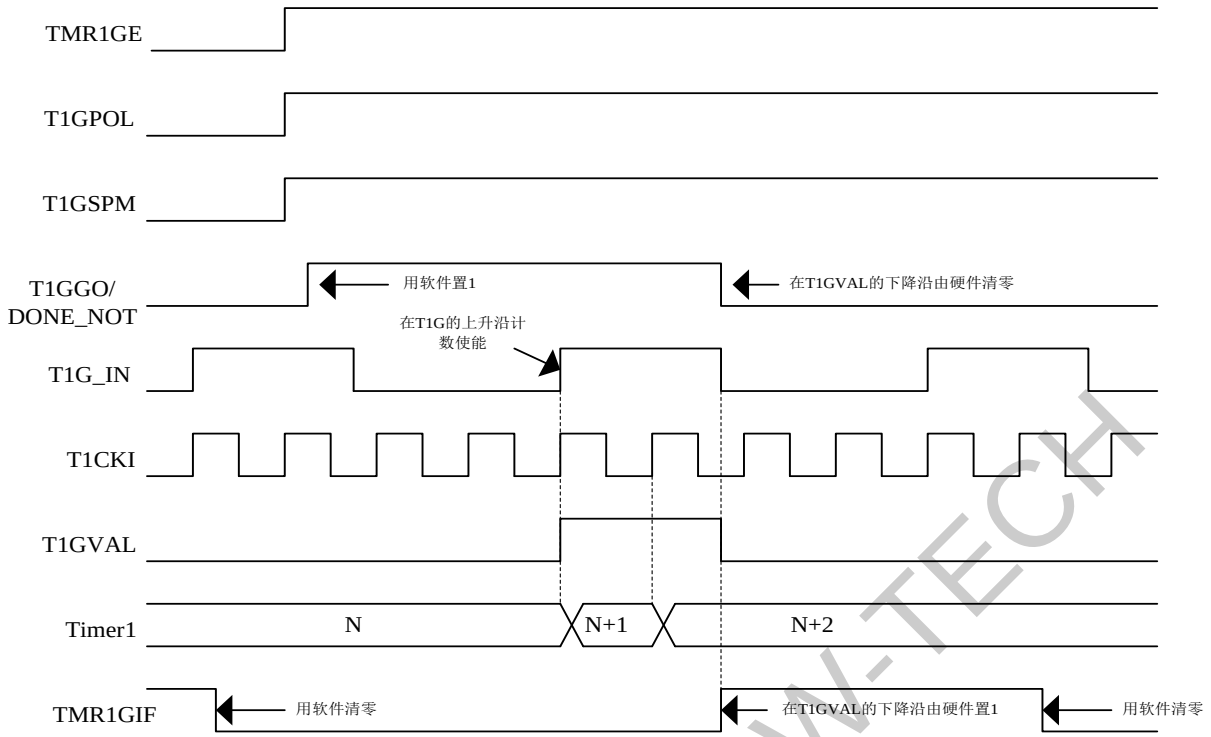


图21-5： TIMER1 门控单脉冲模式

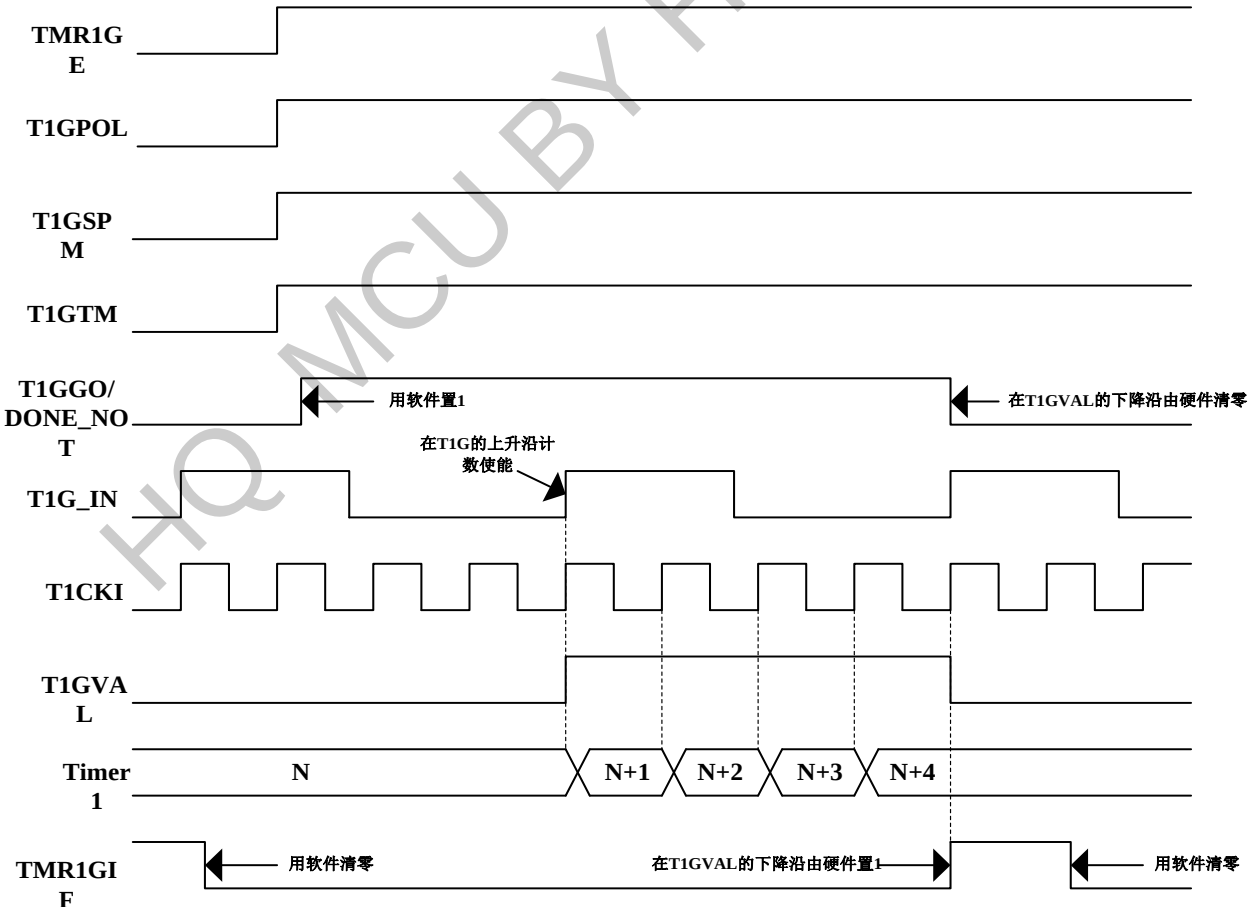


图21-6： TIMER1 门控单脉冲和翻转组合模式

21.6.5 Timer1 门控值状态

使用Timer1 门控值状态时，可读取门控控制值的最新电平。该值保存在T1GCON 寄存器的T1GVAL 位中。即使Timer1 门控未使能（TMR1GE 位清零），T1GVAL位也是有效的。

21.6.6 Timer1 门控事件中断

允许Timer1门控事件中断时，可在门控事件完成时产生一个中断。出现T1GVAL的下降沿时， PIFB1寄存器中的TMR1GIF标志位将置1。如果PIEB1寄存器中的TMR1GIE位置1，则会识别出一个中断。即使Timer1门控未使能（TMR1GE位清零），TMR1GIF标志位也能工作。

21.7 Timer1 中断

Timer1 寄存器对（TMR1H:TMR1L）递增到FFFFh，然后计满返回到0000h。当Timer1计满返回时，PIFB1寄存器的Timer1中断标志位将置1。为允许计满返回时的中断，必须将以下位置1：

- T1STA 寄存器的TMR1ON 位
- PIEB1 寄存器的TMR1IE 位
- INTS 寄存器的PEIE 位
- INTS 寄存器的GIE 位

在中断服务程序中将TMR1IF位清零将清除中断。

注：在允许中断前，应将TMR1H:TMR1L 寄存器对以及TMR1IF 位清零。

21.8 休眠期间的 Timer1 操作

只有在设置为异步计数器模式时，Timer1才能在休眠模式下工作。在该模式下，可使用外部晶振或时钟源使计数器递增计数。要设置定时器以唤醒器件：

- 必须将T1STA 寄存器的TMR1ON 位置1
- 必须将PIEB1 寄存器的TMR1IE 位置1
- 必须将INTS 寄存器的PEIE 位置1
- 必须将T1STA 寄存器的T1SYNC位置1
- 必须配置T1STA 寄存器的TMR1CS 位
- 必须配置T1STA 寄存器的T1OSCEN 位

器件将在上溢时被唤醒并执行下一条指令。如果将INTS 寄存器的GIE位置1，器件将调用中断服务程序。无论T1SYNC位的设置如何， Timer1振荡器都会在休眠模式下继续工作。

21.9 ECCP/CCP 捕捉/比较时基

当工作在捕捉或比较模式下时， CCP1 模块使用TMR1H:TMR1L 寄存器对作为时基。在捕捉模式下，当发生配置的事件时，TMR1H:TMR1L 寄存器对中的值被复制到CCPR1H:CCPR1L 寄存器对中。在比较模式下，当CCPR1H:CCPR1L 寄存器对中的值与TMR1H:TMR1L 寄存器对中的值相匹配时触发事件。该事件可以是特殊事件触发信号。

21.10 ECCP/CCP 特殊事件触发信号

当将任一CCP配置为触发特殊事件时，触发信号将清零TMR1H:TMR1L 寄存器对。该特殊事件不会引起Timer1 中断。CCP模块仍可配置为产生CCP 中断。在该工作模式下， CCPR1H:CCPR1L寄存器对变成Timer1的周期寄存器。为了利用特殊事件触发信号，Timer1应进行同步，并且应选择FOSC/4作为时钟源。Timer1的异步操作会导致错过特殊事件触发信号。如果对TMR1H或TMR1L的写操作和来自CCP的特殊事件触发信号同时发生，则写操作优先。

21.11 寄存器说明

寄存器18H：TIMER1控制寄存器（T1STA）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
TMR1CS1	**TMR1CS0**	**T1CKPS1**	**T1CKPS0**	**T1OSCEN**	**$\overline{\text{T1SYNC}}$**	**—**	**TMR1ON**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-6 **TMR1CS<1:0>**：Timer1 时钟源选择位
11 = Timer1 时钟源为电容传感振荡器（CAPOSC）
10 = Timer1 时钟源为引脚或振荡器：
如果T1OSCEN = 0：
来自T1CKI 引脚的外部时钟（上升沿触发计数）
如果T1OSCEN = 1：
T1OSI/T1OSO 引脚上的晶振
01 = Timer1 时钟源为系统时钟（FOSC）
00 = Timer1 时钟源为指令时钟（FOSC/4）

bit5-4 **T1CKPS<1:0>**：Timer1 输入时钟预分频比选择位
11 = 1:8 预分频比
10 = 1:4 预分频比
01 = 1:2 预分频比
00 = 1:1 预分频比

bit3 **T1OSCEN**：LP 振荡器使能控制位
1 = 使能专用的Timer1 振荡器电路
0 = 禁止专用的Timer1 振荡器电路

bit2 **$\overline{\text{T1SYNC}}$**：Timer1 外部时钟输入同步控制位
TMR1CS<1:0> = 1X
1 = 不同步外部时钟输入
0 = 将外部时钟输入与系统时钟同步（FOSC）
TMR1CS<1:0> = 0X
该位被忽略。

bit1 未实现

bit0 **TMR1ON**：Timer1 使能位
1 = 使能Timer1
0 = 停止Timer1
清零Timer1 门控单稳态触发器

寄存器19H：TIMER1门控控制寄存器1（T1GCON）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMR1GE	**T1GPOL**	**T1GTM**	**T1GSPM**	**T1GGO/$\overline{\text{DONE}}$**	**T1GVAL**	**T1GSS1**	**T1GSS0**
bit7							bit0

图注：
R = 可读位　W = 可写位　U = 未实现位，读为0
-n = POR时的值　1 = 置1　0 = 清零　x = 未知

bit7 **TMR1GE**：Timer1 门控使能位
如果TMR1ON = 0：
该位被忽略
如果TMR1ON = 1：
1 = Timer1 计数由Timer1 门控功能控制
0 = Timer1 计数与Timer1 门控功能无关

bit6 **T1GPOL**：Timer1 门控极性位
1 = Timer1 门控为高电平有效（当门控信号为高电平时Timer1 计数）
0 = Timer1 门控为低电平有效（当门控信号为低电平时Timer1 计数）

bit5 **T1GTM**：Timer1 门控翻转模式位
1 = 使能Timer1 门控翻转模式
0 = 禁止Timer1 门控翻转模式并清除触发器的输出
Timer1 门控单稳态触发器在每个上升沿翻转

bit4 **T1GSPM**：Timer1 门控单脉冲模式位
1 = 使能Timer1 门控单脉冲模式，控制Timer1 门控
0 = 禁止Timer1 门控单脉冲模式

bit3 **T1GGO/$\overline{\text{DONE}}$**：Timer1 门控单脉冲采集状态位
1 = Timer1 门控单脉冲采集就绪，正在等待一个边沿
0 = Timer1 门控单脉冲采集已经结束或尚未开始

bit2 **T1GVAL**：Timer1 门控当前状态位
指示可提供给TMR1H:TMR1L 的Timer1 门控信号的当前状态。
不受Timer1 门控使能（TMR1GE）的影响。

bit1-0 **T1GSS<1:0>**：Timer1 门控源选择位
00 = Timer1 门控引脚
01 = Timer0 上溢输出
10 = 比较器1 的可选同步输出（SYNCC1OUT）
11 = 比较器2 的可选同步输出（SYNCC2OUT）

寄存器16H：定时器1数字寄存器低字节（TMR1L）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
bit7							bit0

图注：
R= 可读位　W= 可写位　U= 未实现位，读为0
-n=POR时的值　1= 置1　0= 清零　x= 未知

bit7-0 **TMR1L <7:0>**：定时器1数字寄存器低字节

寄存器17H：定时器1数字寄存器高字节（TMR1H）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-0 **TMR1H <7:0>**：定时器1数字寄存器高字节

表 21-1：与 TMER1 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	xxx1 x111	xxx1 x111
INTS	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	0000 000x	0000 000q
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP11IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xx11 x111	xx11 x111
TMR1L	TIMER1 数据寄存器低字节								xxxx xxxx	qqqq qqqq
TMR1H	TIMER1 数据寄存器高字节								xxxx xxxx	qqqq qqqq
T1STA	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	0000 00x0	0000 00q0
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS1	T1GSS0	0000 0000	0000 0000

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

22.0 TIMER2 模块

Timer2 模块具有以下特性：

- 8 位定时器和周期寄存器（分别为TMR2 和PR2）
- 可读写（以上两个寄存器）
- 软件可编程的预分频器（分频比为1:1、1:4、1:16和1:64）
- 软件可编程的后分频器（分频比为1:1 至1:16）
- TMR2 与PR2 匹配时产生中断
- 可选择用作MSSP1 模块的移位时钟（仅限Timer2）

Timer2 框图请参见图22-1。

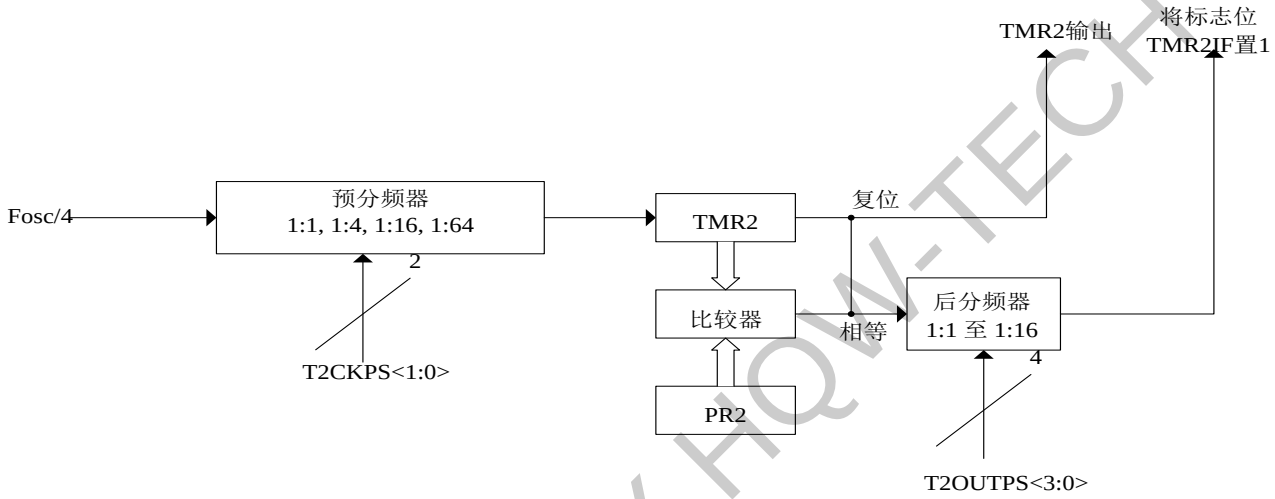


图22-1： TIMER2 框图

22.1 Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟（FOSC/4）。TMR2 会从00h 开始在每个时钟边沿递增。4 位计数器/ 预分频器提供了对时钟输入不分频、4 分频、16 分频和64 分频四个预分频选项。这些选项通过T2STA 寄存器的预分频控制位T2CKPS<1:0> 进行选择。在每个时钟周期，TMR2 的值都会与周期寄存器PR2 中的值进行比较。当两个值匹配时，由比较器产生匹配信号作为定时器的输出。该信号也会将TMR2 的值在下一个周期复位为00h，并驱动输出计数器/后分频器。TMR2 和PR2 寄存器均可直接读写。在任何器件复位时，TMR2 寄存器都会清零，而PR2 寄存器则初始化为FFh。发生以下事件时，预分频器和后分频器计数器均会清零：

- 对TMR2 寄存器进行写操作
- 对T2STA 寄存器进行写操作
- 上电复位（POR）
- 欠压复位（BOR）
- MCLR 复位
- 看门狗定时器（WDT）复位
- 堆栈上溢复位
- 堆栈下溢复位
- RESET 指令

注： 写 T2STA 时 TMR2 不会清零

22.2 Timer2 中断

Timer2 也可以产生可选的器件中断。Timer2 输出信号（TMR2 与PR2 匹配时）为4 位计数器/ 后分频器提供输入。该计数器产生TMR2 匹配中断，对应的中断标志位为PIFB1 寄存器的TMR2IF 位。可以通过将PIEB1 寄存器的TMR2匹配中断允许位TMR2IE置1来允许该中断。可以通过T2STA 寄存器的后分频比控T2OUTPS<3:0>在16 个后分频比选项（从1:1 至1:16）中选择其一。

22.3 Timer2 输出

TMR2 的未经分频的输出主要用于CCP1 模块，它用作CCP1 模块在PWM 模式下工作时的时基。还可选择将Timer2 用作MSSP1 模块在SPI 模式下工作时的移位时钟源。

22.4 休眠期间 Timer2 操作

在处理器处于休眠模式时，Timer2 定时器无法工作。在处理器处于休眠模式时，TMR2 和PR2 寄存器的内容将保持不变。

22.5 寄存器说明

寄存器1CH：TIMER2控制寄存器（T2STA）

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 未实现

Bit6-3 **T2OUTPS<3:0>**：Timer2 输出后分频比选择位
0000 = 1:1 后分频比
0001 = 1:2 后分频比
0010 = 1:3 后分频比
0011 = 1:4 后分频比
0100 = 1:5 后分频比
0101 = 1:6 后分频比
0110 = 1:7 后分频比
0111 = 1:8 后分频比
1000 = 1:9 后分频比
1001 = 1:10 后分频比
1010 = 1:11 后分频比
1011 = 1:12 后分频比
1100 = 1:13 后分频比
1101 = 1:14 后分频比
1110 = 1:15 后分频比
1111 = 1:16 后分频比

bit2 **TMR2ON**：Timer2 使能位

1 = 使能Timer2
0 = 关闭Timer2

Bit1-0 **T2CKPS<1:0>**：Timer2 时钟预分频比选择位
00 = 预分频比为1
01 = 预分频比为4
10 = 预分频比为16
11 = 预分频比为64

寄存器1AH：定时器2数字寄存器（TMR2）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-0 **TMR2 <7:0>**：定时器2数字寄存器

表 22-1：与 TMER2 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	0000 0000	0000 0000
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP11IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
TMR2	TIMER2 数据寄存器								0000 0000	0000 0000
T2STA	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	0000 0000	0000 0000

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

23.0 数据信号调制器

数据信号调制器（DSM）是一种外设，用户可以通过它将数据流（也称为调制器信号）与载波信号进行混合来产生调制输出。载波和调制器信号均送到DSM 模块，信号可以来自内部、某个外设的输出，也可以通过某个输入引脚从外部提供。调制输出信号的产生方式是：对载波和调制器信号执行逻辑与操作，然后送MDOUT引脚上。载波信号由两个不同的独立信号组成。载波高（CARH）信号和载波低（CARL）信号。在调制器（MOD）信号处于逻辑高电平状态期间，DSM会将载波高信号与调制器信号进行混合。在调制器信号处于逻辑低电平状态时， DSM 会将载波低信号与调制器信号进行混合。通过这种方法，DSM 可以产生以下几种键控调制方案：

- 频移键控（Frequency-Shift Keying， FSK）
- 相移键控（Phase-Shift Keying， PSK）
- 开关键控（On-Off Keying， OOK）

此外， DSM 模块还提供了以下特性：

- 载波同步
- 载波源极性选择
- 载波源引脚禁止
- 可编程调制器数据
- 调制器源引脚禁止
- 调制输出极性选择
- 压摆率控制

图23-1 给出了数据信号调制器外设的简化框图。

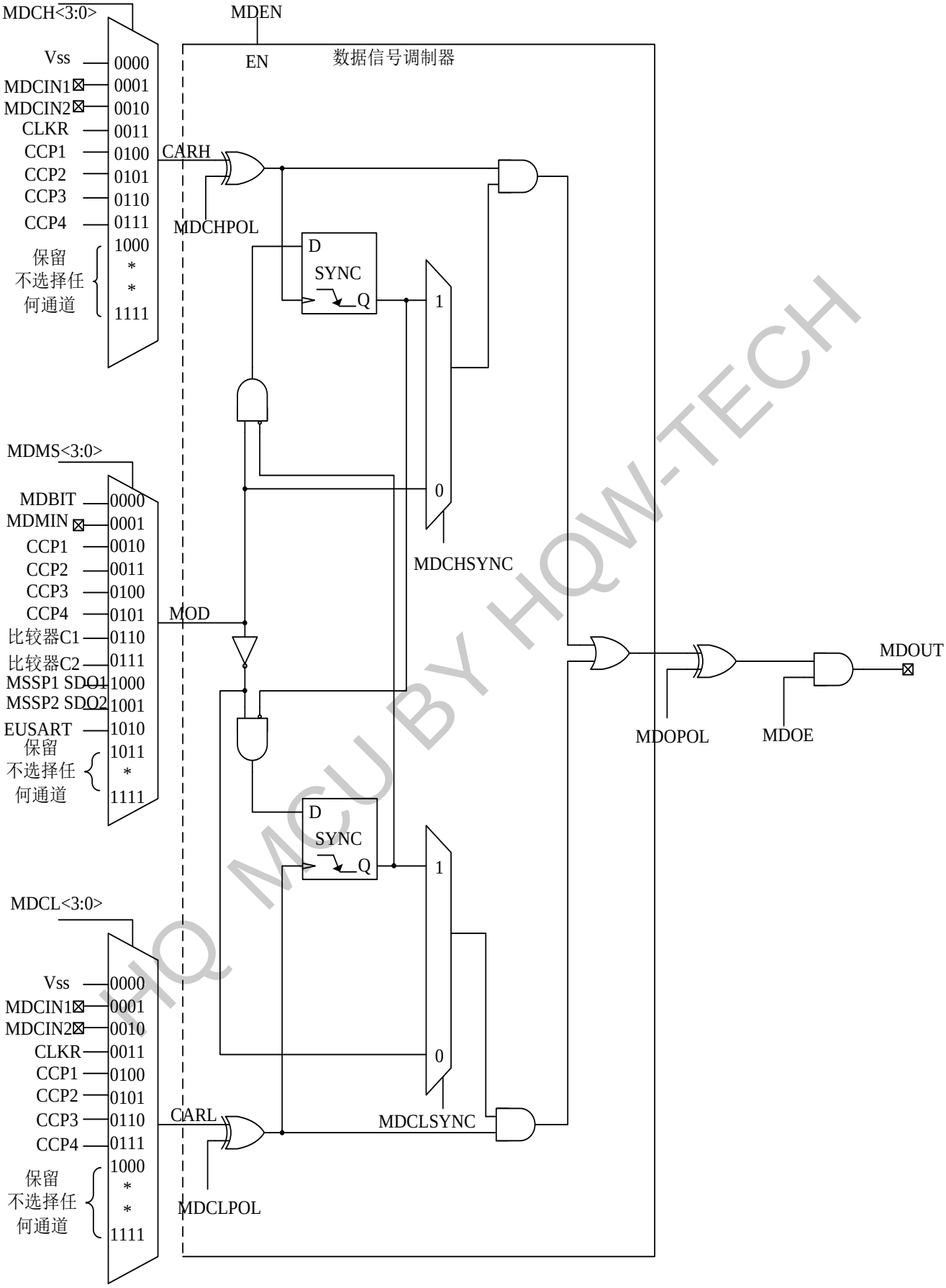


图23-1： 数据信号调制器的简化框图

23.1 DSM 操作

DSM 模块可以通过将MDCON 寄存器中的MDEN 位置1 来使能。清零MDCON 寄存器中的MDEN 位时，将会通过自动将载波高信号和载波低信号切换至VSS信号源来禁止DSM 模块。调制器信号源也会被切换至MDCON 寄存器中的MDBIT。这不仅可以确保DSM 模块不活动，而且会使电流消耗降至最低。

当MDEN 位清零且DSM 模块被禁止时，由调制源、调制载波高信号和调制载波低信号控制寄存器保存的、用于选择载波高信号、载波低信号和调制器信号源的值不会受影响。在DSM 不活动时，这些寄存器中的值会保持不变。当MDEN 位置1 且DSM 模块再次使能并处于活动状态时，将会再次选择载波高信号、载波低信号和调制器信号的信号源。用户可以在无需关闭DSM 模块的情况下禁止调制输出信号。DSM 模块将保持活动状态，继续对信号进行混合，但输出值不会发送到MDOUT 引脚上。在禁止输出期间，MDOUT 引脚将保持低电平。调制输出可以通过清零MDCON 寄存器中的MDOE 位来禁止。

23.2 调制器信号源

调制器信号可以通过以下信号源提供：

- CCP1 信号
- MSSP1 SDO1 信号（仅限SPI 模式）
- 比较器C1 信号
- 比较器C2 信号
- EUSART 发送信号
- MDMIN 引脚上的外部信号
- MDCON 寄存器中的MDBIT 位

调制器信号通过配置MDSRC寄存器中的MDMS <3:0>位来进行选择。

23.3 载波信号源

载波高信号和载波低信号可以通过以下信号源提供：

- CCP1 信号
- 参考时钟模块信号
- MDCIN1 引脚上的外部信号
- MDCIN2 引脚上的外部信号
- VSS

载波高信号通过配置MDCARH寄存器中的MDCH <3:0>位来进行选择。载波低信号通过配置MDCARL寄存器中的MDCL <3:0> 位来进行选择。

23.4 载波同步

当DSM 在载波高信号源和载波低信号源之间切换时，调制输出信号中的载波数据可能会被截短。为了防止这种情况，可以将载波信号与调制器信号进行同步。当使能同步时，DSM将允许在切换时进行混合的载波脉冲先变为低电平，然后再切换为另一个载波源。对于载波高信号源和载波低信号源，同步功能单独进行使能。载波高信号的同步可以通过将MDCARH寄存器中的MDCHSYNC 位置1 来使能。载波低信号的同步可以通过将MDCARL寄存器的MDCLSYNC 位置1 来使能。图23-2 至图23-6 给出了使用各种同步方法的时序图。

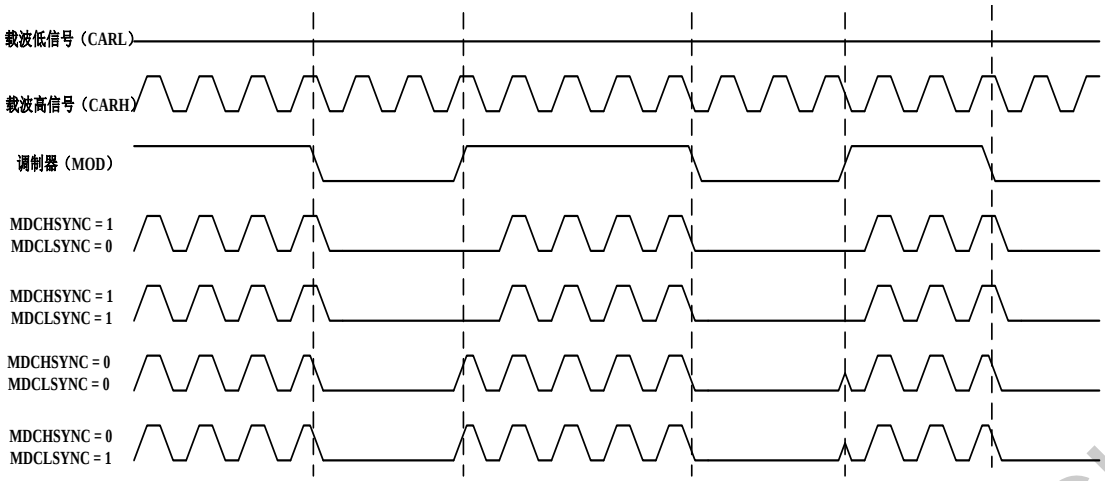


图23-2： 开关键控（OOK）同步

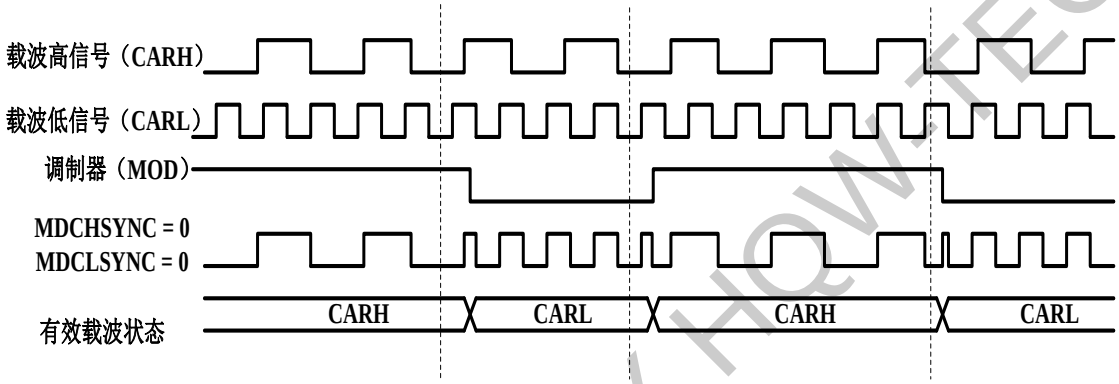


图23-3：无同步（MDSHSYNC = 0， MDCLSYNC = 0）

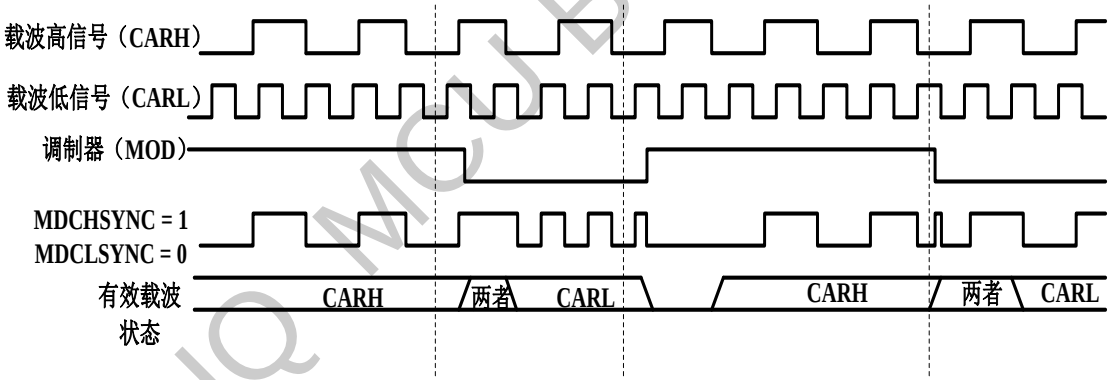


图23-4：载波高信号同步（MDSHSYNC = 1， MDCLSYNC = 0）

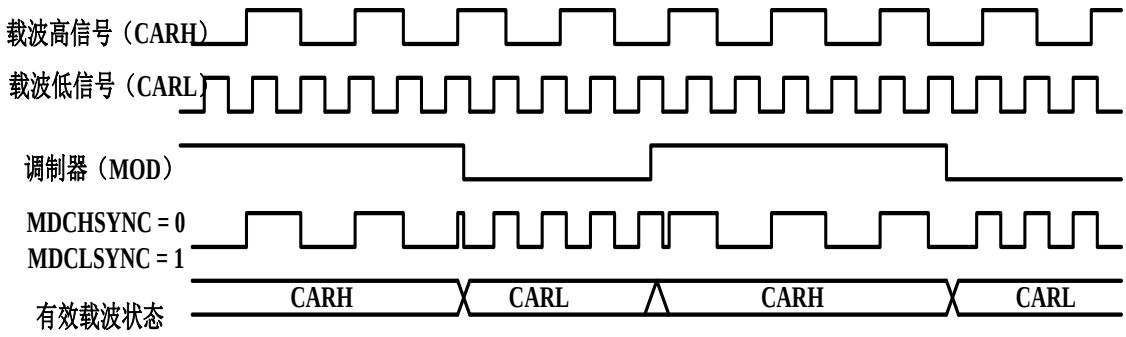


图 23-5：载波低信号同步（MDSHSYNC = 0， MDCLSYNC = 1

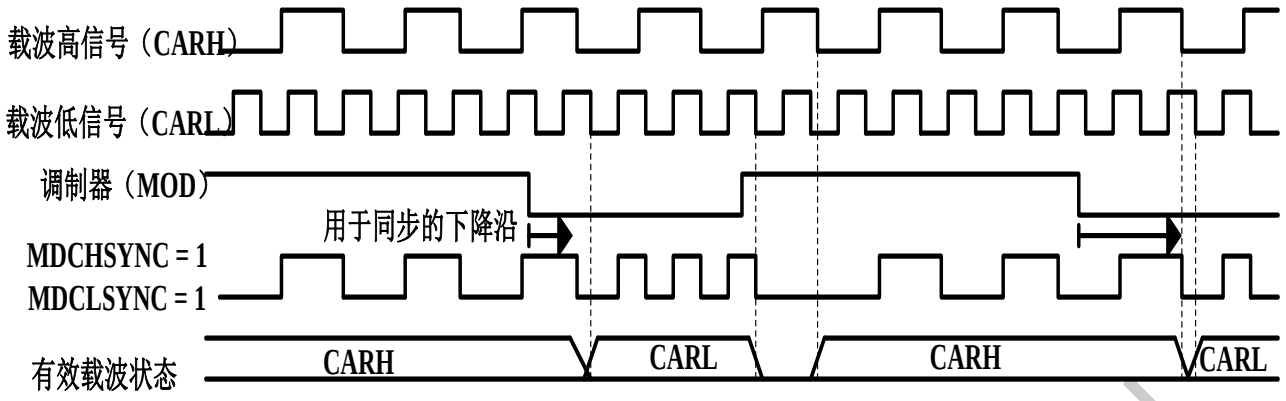


图 23-6：完全同步（MDSHSYNC = 1， MDCLSYNC = 1）

23.5 载波源极性选择

从任意选定输入源为载波高信号和载波低信号提供的信号都可以进行反相。对载波高信号源信号反相通过将MDCARH 寄存器的MDCHPOL 位置1 来使能。对载波低信号源信号反相通过将MDCARL寄存器MDCLPOL位置1来使能。

23.6 载波源引脚禁止

使能某些外设时，这些外设会维持对于它们相应输出引脚的控制。例如，当使能CCP1 模块时，CCP1 的输出将与CCP1 引脚连接。引脚的默认连接可以通过将MDCARH 寄存器中的MDCHODIS 位（对于载波高信号源）和MDCARL寄存器中的MDCLODIS位（对于载波低信号源）置1来禁止。

23.7 可编程调制器数据

用户可以选择MDCON 寄存器的MDBIT 作为调制器信号的信号源。这使用户可以设定用于调制的值。

23.8 调制器源引脚禁止

引脚的调制器信号源默认连接可以通过将MDSRC寄存器中的MDMSODIS 位置1 来禁止。

23.9 调制输出极性

送到MDOUT 引脚上的调制输出信号也可以进行反相。调制输出信号反相通过将MDCON 寄存器的MDOPOL位置1 来使能。

23.10 压摆率控制

用户可以禁止输出端口引脚上的压摆率限制。压摆率限制可以通过将MDCON 寄存器中的MDSLR 位清零而取消。

23.11 休眠模式下操作

DSM模块不受休眠模式的影响。如果载波和调制器输入源可在休眠期间继续工作，则DSM 也可以在休眠期间继续工作。

23.12 复位的影响

在发生任何器件复位时，数据信号调制器模块都会被禁止。用户的固件负责在使能输出之前初始化模块。寄存器会复位为它们的默认值。

23.13 寄存器说明

寄存器39CH：调制控制寄存器（MDCON）

R/W-0	R/W-0	R/W-1	R/W-0	R-0	U-0	U-0	R/W-0
MDEN	**MDOE**	**MDSLR**	**MDOPOL**	**MDOUT**	—	—	**MDBIT**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **MDEN**：调制器模块使能位
1 = 使能调制器模块，并对输入信号进行混合
0 = 禁止调制器模块，不产生任何输出

Bit6 **MDOE**：调制器模块引脚输出使能位
1 = 使能调制器引脚输出
0 = 禁止调制器引脚输出

Bit5 **MDEN**：MDOUT 引脚压摆率限制位
1 = 使能MDOUT 引脚压摆率限制
0 = 禁止MDOUT 引脚压摆率限制

Bit4 **MDOPOL**：调制器输出极性选择位
1 = 调制器输出信号反相
0 = 调制器输出信号不反相

Bit3 **MDOUT**：调制器输出位
指示调制器模块的当前输出值。(1)

bit2-1 **未实现：读为0**

Bit0 **MDBIT**：供软件用于手动设置模块的调制源输入(2)
1 = 调制器使用高载波信号源
0 = 调制器使用低载波信号源

注 **1、**调制输出频率可能会高于更新该寄存器位的时钟，与该时钟异步；位值对于速度较高的调制器或载波信号可能无效。
2、对于该操作，必须在MDSRC 寄存器中选择MDBIT 作为调制源。

寄存器39DH：调制源控制寄存器（MDSRC）

R/W-x	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
MDMSODIS	—	—	—	**MDMS3**	**MDMS2**	**MDMS1**	**MDMS0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **MDMSODIS**：调制源输出禁止位
1 = 禁止驱动外设输出引脚（通过MDMS<3:0> 选择）的输出信号
0 = 使能驱动外设输出引脚（通过MDMS<3:0> 选择）的输出信号

Bit6-4 未实现

Bit3-0 **MDMS<3:0>**：调制源选择位
1111 = 保留。不连接任何通道。
1110 = 保留。不连接任何通道。
1101 = 保留。不连接任何通道。

1100 = 保留。不连接任何通道。
1011 = 保留。不连接任何通道。
1010 = EUSART 发送输出
1001 = 保留。不连接任何通道。
1000 = MSSP1 SDO1 输出
0111 = 比较器2 的输出
0110 = 比较器1 的输出
0101 = 保留。不连接任何通道。
0100 = 保留。不连接任何通道。
0011 = 保留。不连接任何通道。
0010 = CCP1 输出（仅限PWM 输出模式）
0001 = MDMIN 端口引脚
0000 = MDCON 寄存器的MDBIT 位是调制源

注：如果载波未进行同步，则信号流中的载波脉宽可能会变窄，或者可能出现尖刺。

寄存器39FH：调制载波高信号控制寄存器（MDCARH）

R/W-x	R/W-x	R/W-x	U-0	R/W-x	R/W-x	R/W-x	R/W-x
MDCHODIS	**MDCHPOL**	**MDCHSYNC**	—	**MDCH3**	**MDCH2**	**MDCH1**	**MDCH0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **MDCHODIS**：调制器载波高信号输出禁止位
1 = 禁止驱动外设输出引脚（通过MDCH<3:0> 选择）的输出信号
0 = 使能驱动外设输出引脚（通过MDCH<3:0> 选择）的输出信号

bit6 **MDCHPOL**：调制器载波高信号极性选择位
1 = 选定的载波高信号反相
0 = 选定的载波高信号不反相

bit5 **MDCHSYNC**：调制器载波高信号同步使能位
1 = 调制器先等待载波高信号上出现下降沿，然后再切换为载波低信号
0 = 调制器输出不与载波高信号进行同步(1)

bit4 未实现：读为0

bit3-0 **MDCH<3:0>**：调制器数据载波高信号选择位(1)
1111 = 保留。不连接任何通道。
•
•
•
0101 = 保留。不连接任何通道。
0100 = CCP1 输出（仅限PWM 输出模式）
0011 = 参考时钟模块信号（CLKR）
0010 = MDCIN2 端口引脚
0001 = MDCIN1 端口引脚
0000 = VSS

注 1：如果载波未进行同步，则信号流中的载波脉宽可能会变窄，或者可能出现尖刺。

寄存器39EH：调制载波低信号控制寄存器（MDCARL）

R/W-x	R/W-x	R/W-x	U-0	R/W-x	R/W-x	R/W-x	R/W-x
MDCLODIS	**MDCLPOL**	**MDCLSYNC**	**—**	**MDCL3**	**MDCL2**	**MDCL1**	**MDCL0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **MDCLODIS**：调制器载波低信号输出禁止位
1 = 禁止驱动外设输出引脚（通过MDCARL 寄存器的MDCL<3:0> 选择）的输出信号
0 = 使能驱动外设输出引脚（通过MDCARL 寄存器的MDCL<3:0> 选择）的输出信号

bit6 **MDCLPOL**：调制器载波低信号极性选择位
1 = 选定的载波低信号反相
0 = 选定的载波低信号不反相

bit5 **MDCLSYNC**：调制器载波低信号同步使能位
1 = 调制器先等待载波低信号上出现下降沿，然后再切换为载波高信号
0 = 调制器输出不与载波低信号进行同步(1)

bit4 未实现：读为0

bit3-0 **MDCL<3:0>**：调制器数据载波低信号选择位(1)
1111 = 保留。不连接任何通道。
•
•
•
0101 = 保留。不连接任何通道
0100 = CCP1 输出（仅限PWM 输出模式）
0011 = 参考时钟模块信号
0010 = 保留。不连接任何通道
0001 = MDCIN1 端口输出
0000 = VSS

注：如果载波未进行同步，则信号流中的载波脉宽可能会变窄，或者可能出现尖刺。

表 23-1：与数据信号调制有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	xxx1 x1111	xxx1 x1111
ADINSC	—	—	—	—	AN7	AN6	AN5	AN4	xxxx 1111	xxxx 1111
MDCARH	MDCHODIS	MDCHPOL	MDCHSYNC	—	MDCH3	MDCH2	MDCH1	MDCH0	xxxx xxxx	qqqq qqqq
MDCARL	MDCHODIS	MDCLPOL	MDCLSYNC	—	MDCL3	MDCL2	MDCL1	MDCL0	xxxx xxxx	qqqq qqqq
MDCON	MDEN	MDOE	MDSLR	MDOPOL	MDOUT	—	—	MDBIT	0010 xxx0	0010 xxx0
MDSRC	MDMSODIS	—	—	—	MDMS3	MDMS2	MDMS1	MDMS0	x000 xxxx	q000 qqqq
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xxx1 x1111	xxx1 x1111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	xx11 1111	xx11 1111
PAPHR	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0	xx11 1111	xx11 1111
PCPHR	—	—	PCPHR5	PCPHR4	PCPHR3	PCPHR2	PCPHR1	PCPHR0	xx11 1111	xx11 1111

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

24.0 捕捉/比较/PWM 模块

捕捉/ 比较/PWM 模块是一个外设，允许用户计时和控制不同事件，以及产生脉宽调制（Pulse-Width Modulation，PWM）信号。在捕捉模式下，外设允许对事件的持续时间进行计时。当超过预先确定的时间时，比较模式允许用户触发一个外部事件。本器件中包含一个增强型捕捉/ 比较/PWM 模块（ECCP1）。全桥ECCP模块具有4个可用的I/O引脚，而半桥ECCP模块只有两个可用的I/O 引脚。

24.1 捕捉模式

捕捉模式使用16 位Timer1 资源。当CCP1 引脚上发生事件时， 16 位CCPR1H:CCPR1L 寄存器对会分别捕捉和存储TMR1H:TMR1L 寄存器对的16 位值。这些事件定义如下，可通过CCP1CON 寄存器的CCP1M<3:0> 位进行配置：

- 每个下降沿
- 每个上升沿
- 每4 个上升沿
- 每16 个上升沿

进行捕捉时，PIFB1 寄存器的中断请求标志位CCP1IF 被置1。该中断标志必须用软件清零。如果在CCPR1H 和CCPR1L 寄存器对中的值被读取之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖。

图24-1 给出了捕捉操作的简化框图。

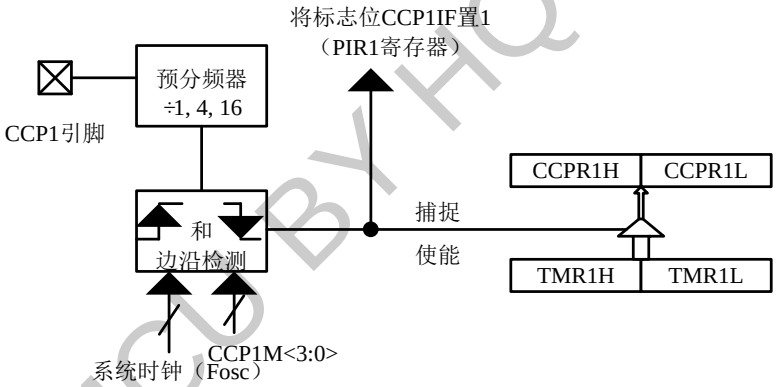


图 24-1：捕捉模式工作原理框图

24.1.1 CCP1 引脚配置

在捕捉模式下，应该通过将相关的CPIO 控制位置1，将CCP1 引脚配置为输入。此外，还可以通过使用APFCON 寄存器将CCP1 引脚功能转移到备用引脚上。

注：如果CCP1 引脚被配置为输出，则对端口执行一次写操作，会产生一次捕捉条件。

24.1.2 Timer1 模式资源

为使CCP1模块使用捕捉特性，Timer1必须运行在定时器模式或同步计数器模式下。在异步计数器模式下，可能无法进行捕捉操作。

24.1.3 软件中断模式

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应保持PIEB1寄存器的CCP1IE 中断允许位清零以避免错误中断。此外，用户应在工作模式的任何改变之后清零PIFB1寄存器的CCP1IF中断标志位。

注： 在捕捉模式下，Timer1时钟源不能由系统时钟（FOSC）提供。为在捕捉模式下识别CCP1引脚上的触发事件， Timer1时钟源必须由指令时钟（FOSC/4）或外部时钟源提供。

24.1.4 CCP1 预分频器

通过CCP1CON 寄存器的CCP1M<3:0> 位，可以指定4 种预分频比设置。每当关闭CCP1 模块，或者CCP1模块不在捕捉模式下时，预分频器计数器就会被清零。任何复位都会将预分频器计数器清零。从一个捕捉预分频比切换到另一个捕捉预分频比不会清零预分频器，而且可能产生一次错误中断。为避免此意外操作，可在改变预分频比之前通过清零CCP1CON 寄存器来关闭模块。例24-1给出了执行此功能的代码。

例 24-1： 改变捕捉预分频比

```
BANKSEL   CCP1CON       ;Set Bank bits to point
                        ;to CCP1CON
CLRR      CCP1CON       ;Turn CCP1 module off
LDWI      NEW_CAPT_PS   ;Load the W reg with
                        ;the new prescaler
                        ;move value and CCP1 ON
STWR      CCP1CON       ;Load CCP1CON with this
                        ;value
```

24.1.5 休眠期间的捕捉操作

捕捉模式能否正常工作取决于Timer1模块。有两个选项可用于在捕捉模式下驱动Timer1模块。它可由指令时钟（FOSC/4）驱动，或由外部时钟源驱动。当Timer1时钟源由FOSC/4提供时， Timer1将不会在休眠期间递增。当器件被从休眠状态唤醒时，Timer1将从先前状态继续。当Timer1通过外部时钟源提供时钟时，捕捉模式会在休眠模式期间继续工作。

24.1.6 备用引脚位置

该模块具有以下I/O 引脚：通过使用备用引脚功能寄存器APFCON 可将I/O 引脚转移到其他位置。

24.2 比较模式

比较模式使用16 位Timer1资源。CCPR1H:CCPR1L寄存器对的16位值会不断与TMR1H:TMR1L寄存器对的16 位值进行比较。当发生匹配时，将发生以下事件之一：

- 翻转CCP1输出
- 将CCP1输出置1
- 将CCP1输出清零
- 产生特殊事件触发信号
- 产生软件中断

引脚的动作由 CCP1CON寄存器的CCP1M<3:0>控制位的值决定。同时，中断标志位CCP1IF置1。所有比较模式都能产生中断。图24-2 给出了比较操作的简化框图。

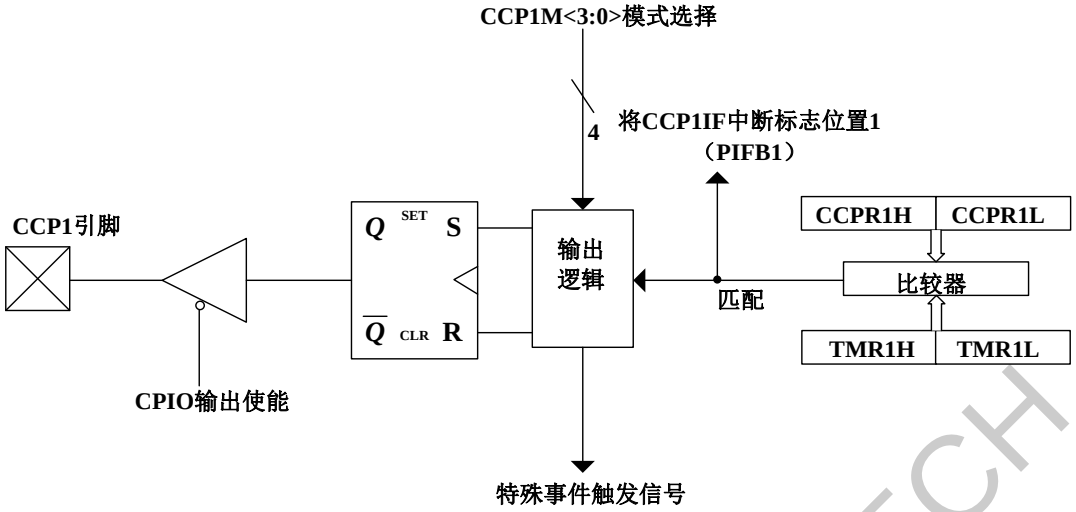


图 24-2： 比较模式工作原理框

24.2.1CCP1 引脚配置

用户必须通过将相关的CPIO 位清零，将CCP1 引脚配置为输出。此外，还可以通过使用APFCON 寄存器将CCP1 引脚功能转移到备用引脚上。

注：清零CCP1CON 寄存器会将CCP1 比较输出锁存器强制设为默认的低电平。这不是端口I/O 数据锁存器。

24.2.2 TIMER1 模式资源

在比较模式下，Timer1 必须运行在定时器模式或同步计数器模式下。在异步计数器模式下，可能无法进行比较操作。

注：在比较模式下，Timer1 时钟源不能由系统时钟（FOSC）提供。为在比较模式下识别CCP1 引脚上的触发事件， Timer1 时钟源必须由指令时钟（FOSC/4）或外部时钟源提供。

24.2.3 软件中断模式

当选择产生软件中断模式（CCP1M<3:0> = 1010）时，CCP1模块不会对CCP1引脚进行控制（见CCP1CON寄存器）。

24.2.4 特殊事件触发器

当选择特殊事件触发器模式（CCP1M<3:0> = 1011）时， CCP1 模块将进行以下操作：

- 复位Timer1
- 如果ADC 被使能，则启动ADC 转换

在该模式下，CCP1模块不会被CCP1引脚进行控制。一旦TMR1H和TMR1L寄存器对与CCPR1H和CCPR1L寄存器对之间发生匹配，便会发生CCP1的特殊事件触发输出。TMR1H和TMR1L寄存器对在Timer1 时钟的下一个上升沿到来之前不会复位。特殊事件触发输出也会启动一次A/D转换（如果A/D 模块被使能）。这使CCPR1H和CCPR1L寄存器对实际上作为Timer1 的16 位可编程周期寄存器。

注 1：CCP模块的特殊事件触发信号不会将PIFB1寄存器的中断标志位TMR1IF置1。
2：通过在产生特殊事件触发信号的时钟边沿和使Timer1 复位的时钟边沿之间更改CCPR1H和CCPR1L寄存器对的内容来移除匹配条件，可以避免复位发生。

24.2.5 休眠期间的比较操作

比较模式能否正常工作取决于系统时钟（FOSC）。由于FOSC 在休眠模式下关闭，比较模式在休眠期间将不能正常工作。

24.2.6 备用引脚

该模块具有以下I/O 引脚：通过使用备用引脚功能寄存器APFCON 可将I/O 引脚转移到其他位置。

24.3 PWM 概述

脉宽调制（PWM）是一种通过在完全开启和完全关闭状态之间进行快速切换而为负载供电的方案。PWM 信号类似于方波，信号的高电平部分视为开启状态，信号的低电平部分视为关闭状态。高电平部分（也称为脉宽）可以随时间而变，并以步幅为单位进行定义。施加的步幅数量越多（这会增大脉宽），为负载提供的电量就越多。施加的步幅数量降低时（这会缩短脉宽），提供的电量就会下降。PWM 周期定义为一个完整周期的持续时间，或者开启和关闭时间相加的总时间。PWM 分辨率定义可以在单个PWM 周期中出现的最大步幅数量。分辨率越高，就可以越精确地控制脉宽时间，从而更精确地控制在负载上的供电量。占空比这一术语描述开启时间与关闭时间之间以百分比形式表示的比例， 0% 代表完全关闭， 100% 代表完全开启。占空比越低，对应的供电量就越低；占空比越高，对应的供电量就越高。

图24-3 给出了PWM 信号的典型波形图。

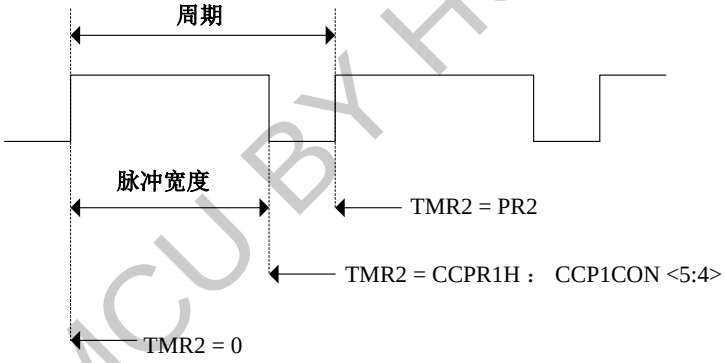


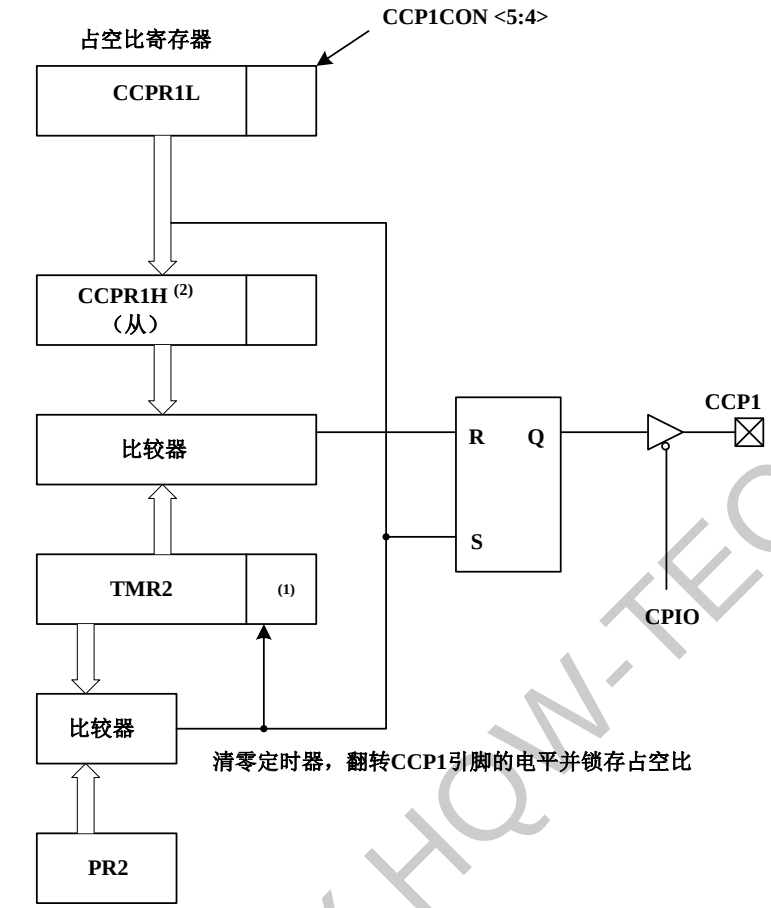
图 24-3： CCP1 PWM 输出信号

24.3.1 标准 PWM 操作

标准PWM 模式可以在CCP1 引脚上产生最高可达10位分辨率的脉宽调制（PWM）信号。周期、占空比和分辨率由以下寄存器控制：

- PR2寄存器
- T2STA寄存器
- CCPR1L 寄存器
- CCP1CON 寄存器

图24-4 给出了PWM 操作的简化框图。



注 1：8位定时器TMR2寄存器与2位内部系统时钟（Fosc）或预分频 器的2位一起构成10位时基。
2：在PWM模式下, CCPR1H是只读寄存器。

图 24-4： PWM 简化框图

注 1： 要使能CCP1 引脚上的PWM 输出，必须清零相应的CPIO 位。
2： 清零CCP1CON寄存器会放弃对CCP1引脚的控制。

24.3.2 设置 PWM 操作

当将CCP1 模块配置为标准PWM 操作时，可采用以下步骤：

1. 通过将相关的CPIO 位置1，禁止CCP1 引脚输出驱动器。
2. 将PWM 周期值装入PR2 寄存器。
3. 通过将相应值装入CCP1CON 寄存器，将CCP1模块配置为PWM 模式。
4. 将PWM 占空比值装入CCPR1L 寄存器和CCP1CON 寄存器的DC1B1 位。
5. 配置和启动Timer2：
   - 清零PIFB1 寄存器的TMR2IF 中断标志位。请参见下面的“ 注”。
   - 用定时器预分频值配置T2STA 寄存器的T2CKPS 位。
   - 通过将T2STA 寄存器的TMR2ON位置1，使能定时器。
6. 使能PWM 输出引脚：
   - 等待直到定时器上溢，PIFB1 寄存器的TMR2IF位置1。请参见下面的“ 注”。
   - 通过将相关的CPIO位清零，使能 CCP1引脚输出驱动器。

注： 为在第一个PWM 输出时发送完整的占空比和周期，设置过程必须包含上述步骤。如果在第一个输出时以完整的PWM 信号开始并非至关重要，那么可以忽略步骤6。

24.3.3 PWM 周期

PWM周期可通过Timer2的PR2寄存器来指定。PWM 可由公式24-1 计算。

公式 24-1： PWM 周期

PWM 周期 = [(PR2) + 1] · 4 · Tosc · (TMR2 与分频值)
注 1： Tosc = 1/FOSC

当TMR2 中的值等于PR2 中的值时，在下一个递增周期将发生以下3 个事件：

- TMR2被清零
- CCP1 引脚被置1。（例外情况：如果 PWM 占空比= 0%，引脚不会被置1。）
- PWM占空比从CCPR1L 锁存到CCPR1H。

注 在确定 PWM 频率时不会用到定时器后分频比。

24.3.4 PWM 占空比

通过将10 位值写入多个寄存器来指定PWM 占空比：CCPR1L寄存器和CCP1CON 寄存器的DC1B<1:0>位。CCPR1L包含高8位，而CCP1CON 寄存器的DC1B<1:0>位包含低2 位。可以在任意时刻写入CCPR1L 和CCP1CON 寄存器的DC1B<1:0> 位。在周期结束（即，PR2 和TMR2 寄存器发生匹配）之前，占空比值不会被锁存到CCPR1H 中。当使用PWM 时，CCPR1H 寄存器是只读的。

公式24-2 用于计算PWM 脉冲宽度。

公式24-3 用于计算PWM 占空比。

公式 24-2： 脉冲宽度

脉冲宽度 = (CCPR1L:CCP1CON <5:4>) · Tosc · (TMR2 预分频值)

公式 24-3： 占空比

$$\text{占空比} = \frac{(CCPRxL : CCPxCON < 5:4 >)}{4(PRx + 1)}$$

CCPR1H 寄存器和一个2 位的内部锁存器用于给PWM占空比提供双重缓冲。这种双重缓冲结构对避免在PWM 操作中产生毛刺非常重要。8 位定时器TMR2 寄存器与2 位内部系统时钟（FOSC）或预分频器的2 位一起构成10 位时基。如果Timer2 预分频比设置为1:1，则使用系统时钟。当10 位时基与CCPR1H 和2 位锁存值匹配时，CCP1引脚被清零（见图24-4）。

24.3.5 PWM 分辨率

分辨率决定在给定周期内的可用占空比数。例如，10 位分辨率将可得到1024 个不连续的占空比，而8 位分辨率将可得到256 个不连续的占空比。当PR2 为255 时，PWM 最大分辨率为10 位。分辨率是PR2 寄存器值的函数，如公式24-4 所示。

公式 24-4： PWM 分辨率

$$\text{分辨率} = \frac{\log[4(PR2+1)]}{\log(2)} \text{位}$$

表 24-5： PWM 频率和分辨率示例（FOSC = 8 MHz）

PWM 频率	1.22kHz	4.90kHz	19. 61kHz	76.92kHz	153.85kHz	200.0kHz
定时器预分频值（1、4 和 16）	16	4	1	1	1	1
PR2 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率（位）	8	8	8	6	5	5

24.3.6 休眠模式下的操作

在休眠模式下，TMR2 寄存器将不会递增，模块状态也不会改变。如果CCP1 引脚正在驱动一个值，则会继续驱动该值。当器件被唤醒时，TMR2将从先前状态继续。

24.3.7 改变系统时钟频率

PWM 频率是由系统时钟频率得到的。系统时钟频率的任何改变将导致PWM 频率的改变。

24.3.8 复位的影响

任何复位都将强制所有端口为输入模式，并强制CCP寄存器为其复位状态。

24.3.9 备用引脚位置

该模块具有以下I/O 引脚：通过使用备用引脚功能寄存器APFCON 可将I/O 引脚转移到其他位置。

24.4 PWM（增强型模式）

增强型PWM模式可以在最多4个不同的输出引脚上产生脉宽调制（PWM）信号，最高可达10位分辨率。周期、占空比和分辨率由以下寄存器控制：

- PR2寄存器
- T2STA寄存器
- CCPR1L寄存器
- CCP1CON寄存器

ECCP模式还另外具有以下PWM寄存器，这些寄存器控制自动关闭、自动重启、死区延时和PWM转向模式 ：

- CCP1AS寄存器
- PSTR1CON寄存器
- PWM1CON寄存器

增强型PWM模式可以产生以下4种PWM输出模式：

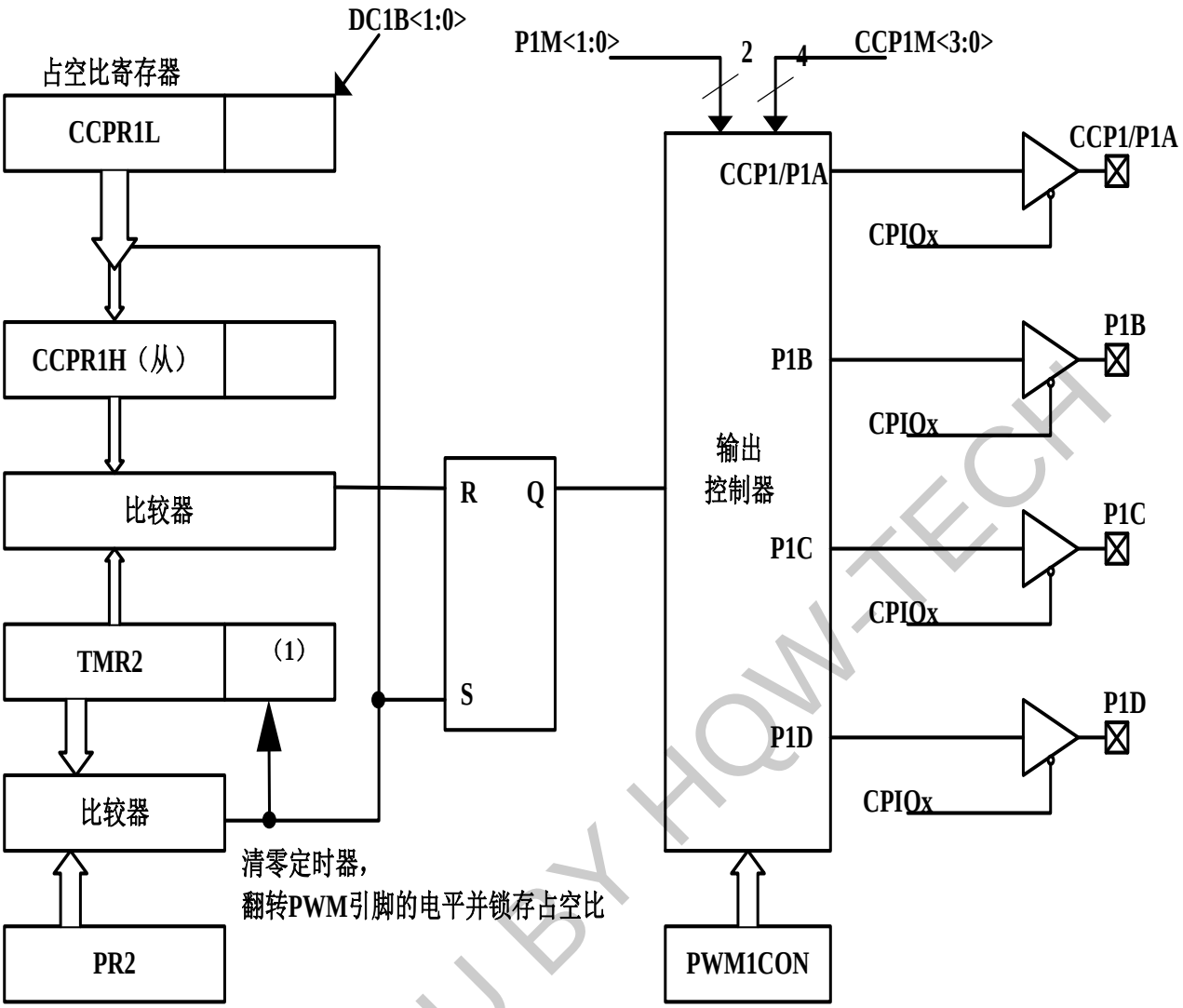
- 单PWM
- 半桥PWM
- 全桥PWM
- 带PWM转向模式的单PWM

要选择增强型PWM输出模式，必须适当配置CCP1CON寄存器的P1M 位。PWM 输出与I/O 引脚复用，指定为P1A、P1B、P1C 和P1D。PWM 引脚的极性是可配置的，通过适当设置CCP1CON寄存器中的CCP1M<3:0> 位进行选择。

图24-5 给出了增强型PWM 模块的简化框图的示例。

表24-9 列出了各种增强型PWM 模式的引脚分配。

注 1： 要使能CCP1 引脚上的PWM 输出，必须清零相应的CPIO 位。
2： 清零CCP1CON寄存器会放弃对CCP1引脚的控制。
3： 增强型PWM 模式下没有使用的任何引脚均具有备用引脚功能（如适用）。
4： 为防止在第一次使能PWM 时产生不完整的波形，ECCP 模块在产生PWM 信号前会等待，直到新的PWM 周期开始。



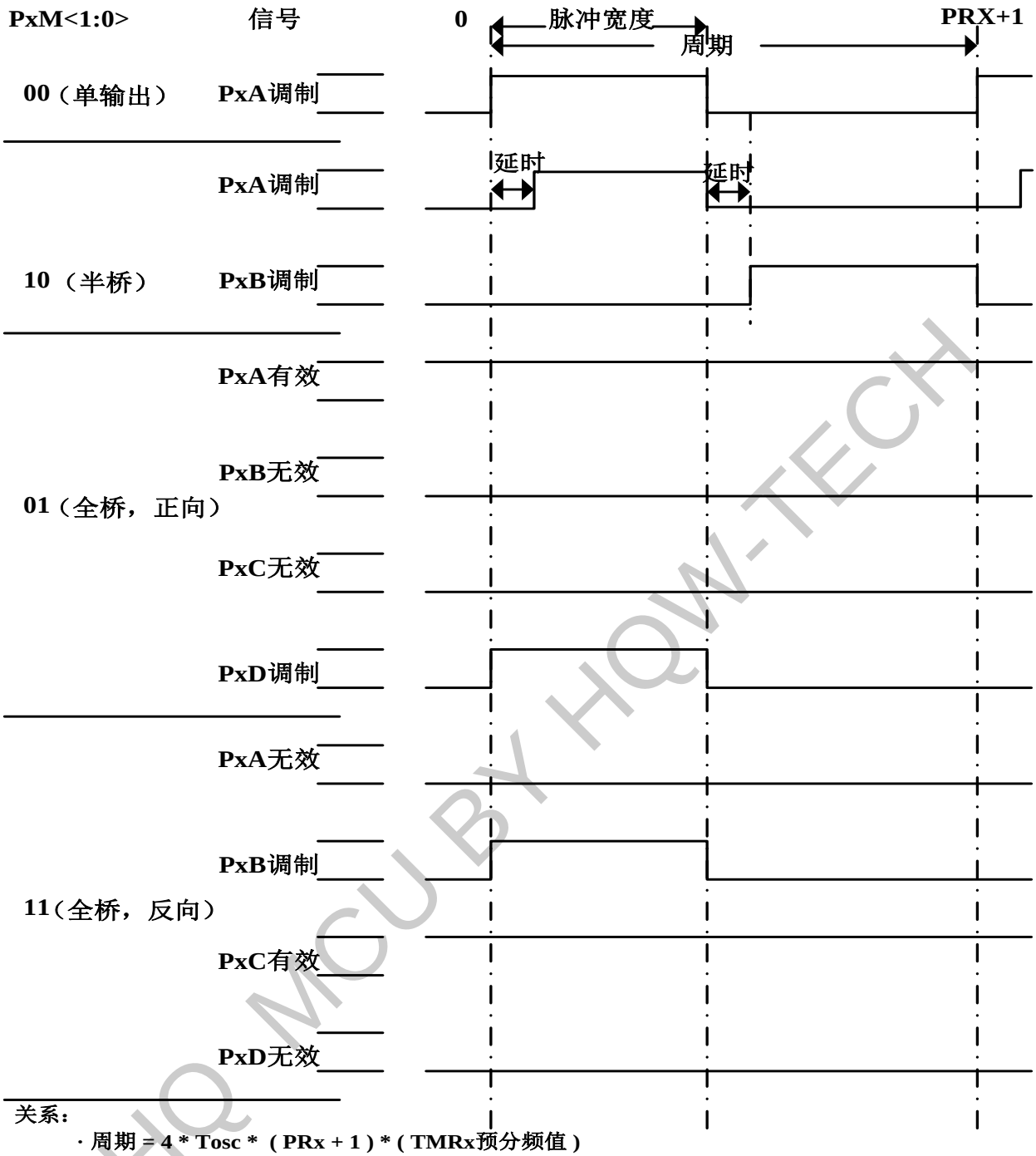
注 1: 8位定时器TMR1寄存器与2位内部Q时钟或预分频器的2位一起构成10位时基

图 24-5： 增强型PWM 模式的简化框图示例

表 24-9： 各种PWM 增强型模式的引脚分配示例

ECCP 模式	P1M<1:0>	CCP1/P1A	P1B	P1C	P1D
单	00	是(1)	是(1)	是(1)	是(1)
半桥	10	是	是	否	否
全桥，正向	01	是	是	是	是
全桥，反向	11	是	是	是	是

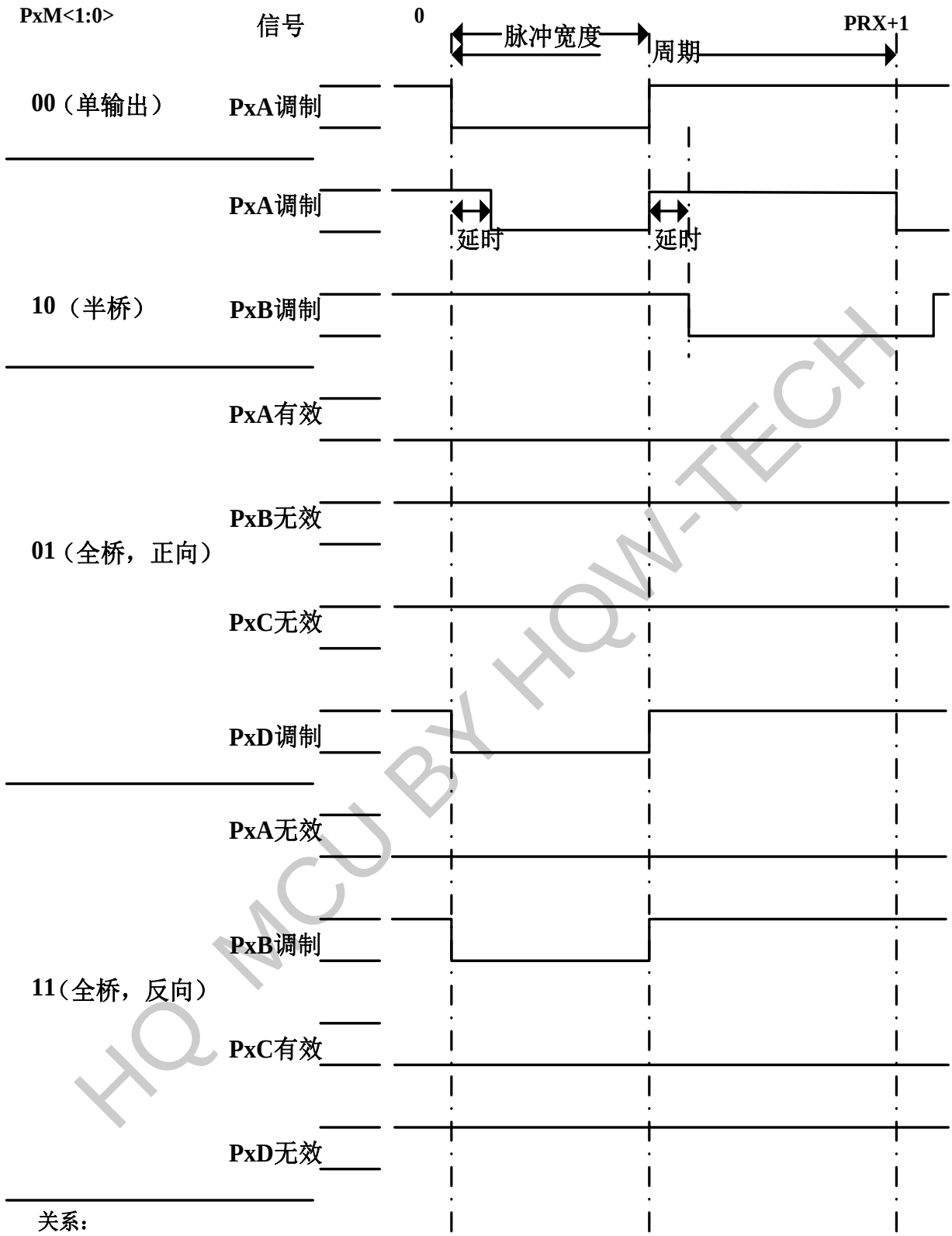
注 1：PWM 转向支持在单个模式下产生多个输出



关系：

- 周期 = 4 * Tosc * (PRx + 1) * (TMRx预分频值)
- 脉冲宽度 = Tosc * (CCPRxL<7:0> : CCPxCON<5:4>) * (TMRx预分频值)
- 延时 = 4 * Tosc * (PWMxCON<6:0>)

图 24-6： PWM （增强型模式）输出关系示例（高电平有效状态）



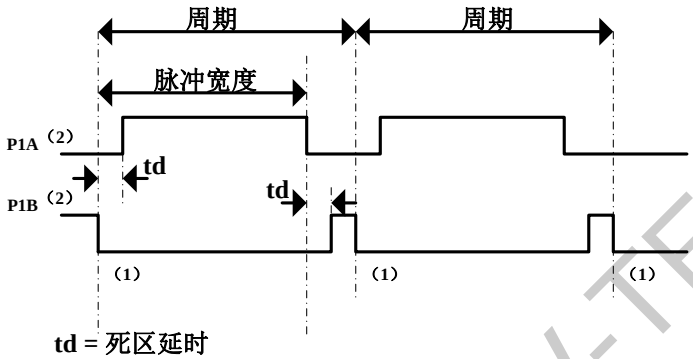
关系：

- 周期 = 4 * Tosc * (PRx + 1) * (TMRx预分频值)
- 脉冲宽度 = Tosc * (CCPRxL<7:0> : CCPxCON<5:4>) * (TMRx预分频值)
- 延时 = 4 * Tosc * (PWMxCON<6:0>)

24-7： 增强型PWM 输出关系示例 （低电平有效状态）

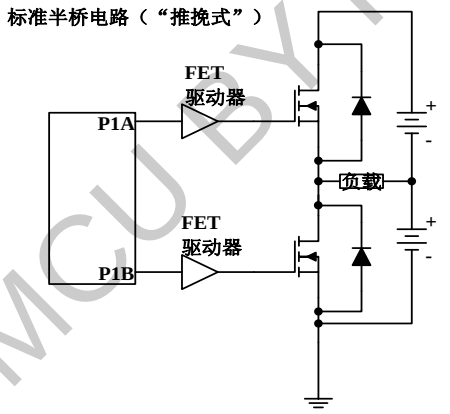
24.4.1 半桥模式

在半桥模式下，有两个引脚用作输出以驱动推挽式负载。CCP1/P1A 引脚输出PWM 输出信号，而P1B 引脚输出互补的PWM 输出信号（见图24-9）。这种模式可用于半桥应用（如图24-9 所示），或者用于全桥应用，在全桥应用中使用两个PWM 信号调制4 个功率开关。在半桥模式下，可编程死区延时可用来防止半桥功率器件中流过直通电流。PWM1CON 寄存器的PDC<6:0> 位的值在输出被驱动为有效之前设置指令周期数。如果这个值比占空比大，则在整个周期中相应的输出保持为无效。由于P1A 和P1B 输出与端口数据锁存器是复用的， 相关的CPIO位必须清零，从而将P1A和P1B配置为输出。



注 1：此时TMR2寄存器和PR2寄存器的值相等
2：图中输出信号为高电平有效

图 24-8： 半桥PWM 输出示例



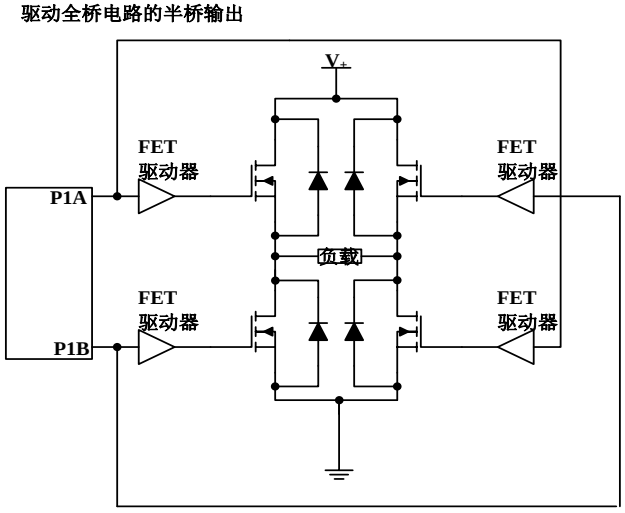


图 24-9：半桥应用示例

24.4.2 全桥模式

在全桥模式下，所有4 个引脚都用作输出。全桥应用的示例如图24-10 所示。在正向模式下，引脚CCP1/P1A 被驱动为有效状态，引脚P1D 被调制，而P1B 和P1C 将被驱动为无效状态，如图24-11 所示。在反向模式下，P1C 被驱动为有效状态，引脚P1B 被调制，而P1A 和P1D 将被驱动为无效状态，如图24-11 所示。P1A、P1B、P1C 和P1D 输出与端口数据锁存器复用。相关的CPIO 位必须清零，从而将P1A、P1B、P1C 和P1D 引脚配置为输出。

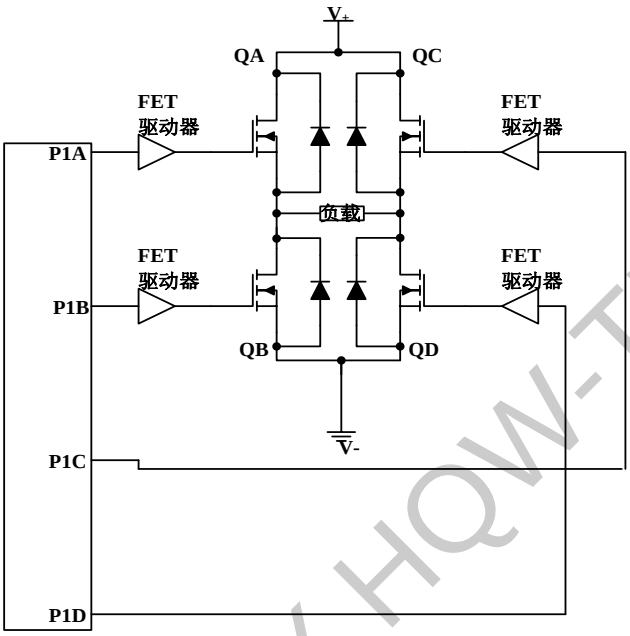


图 24-10：全桥应用示例

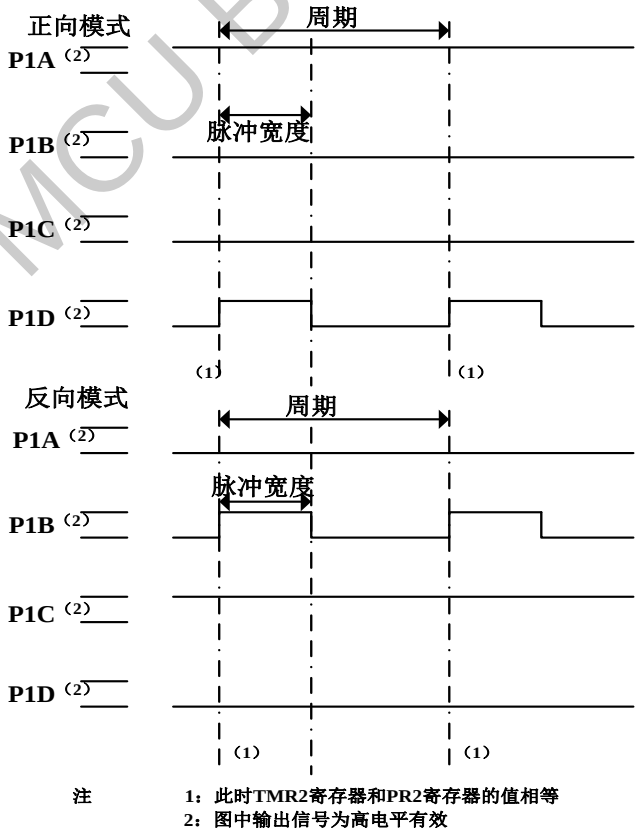


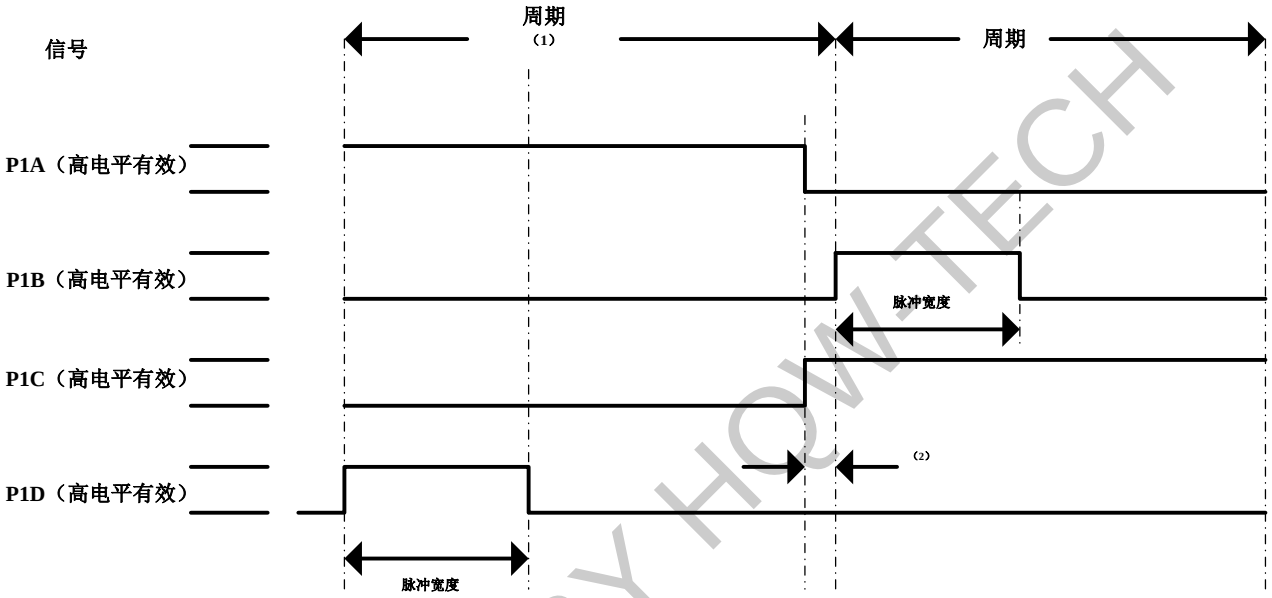
图 24-11：全桥PWM 输出示例

24.4.2.1 全桥模式下的方向改变

在全桥模式下，CCP1CON 寄存器的P1M1 位允许用户控制正/ 反向。当应用固件改变这个方向控制位时，模块将在下一个PWM 周期改用新的方向。通过改变CCP1CON 寄存器的P1M1 位，可以用软件改变方向。以下序列在当前PWM 周期结束前的4 个定时器周期时发生：

- 调制输出（P1B 和P1D）被置于无效状态。
- 相关的未调制输出（P1A 和P1C）被切换到以相反的方向驱动。
- PWM调制在下一个周期开始继续。

关于该序列的说明，请参见图24-12。



注 1：可在PWM周期的任意时刻，写CCP1CON寄存器的方向位P1M1.
2：当改变方向时，P1A和P1C信号在当前PWM周期结束前切换。此时被调制的P1B和P1D信号是无效的。该时间长 度为定时器计数值的4倍

图 24-12： PWM 方向改变的示例

全桥模式不提供死区延时。因为一次只有一个输出被制，所以一般不需要死区延时。有一种情况需要死区延时。这一情况发生在以下两个条件同时满足时：

1. 当输出的占空比达到或者接近100% 时， PWM输出方向改变。
2. 功率开关（包括功率器件和驱动电路）的关断时间比导通时间要长。

在图24-13所示的示例中，在占空比接近100%时， PWM方向从正向改变到反向。在这个示例中，在时间t1，输出P1A 和P1D 变为无效，而输出P1C 变为有效。因为功率器件的关断时间比导通时间要长，在“t” 时间内，功率器件QC 和QD 中可能流过直通电流（见 图24-10）。当PWM 方向从反向改变到正向时，功率器件 QA 和QB 也将出现相同的现象。如果应用中需要在高占空比时改变PWM 方向，避免直通电流可采用以下两种方法：

1. 在改变方向之前的一个PWM 周期降低PWM 占空比。
2. 使用开关驱动电路，使开关的关断时间比导通时间短。

也可能存在其他避免直通电流的方案。

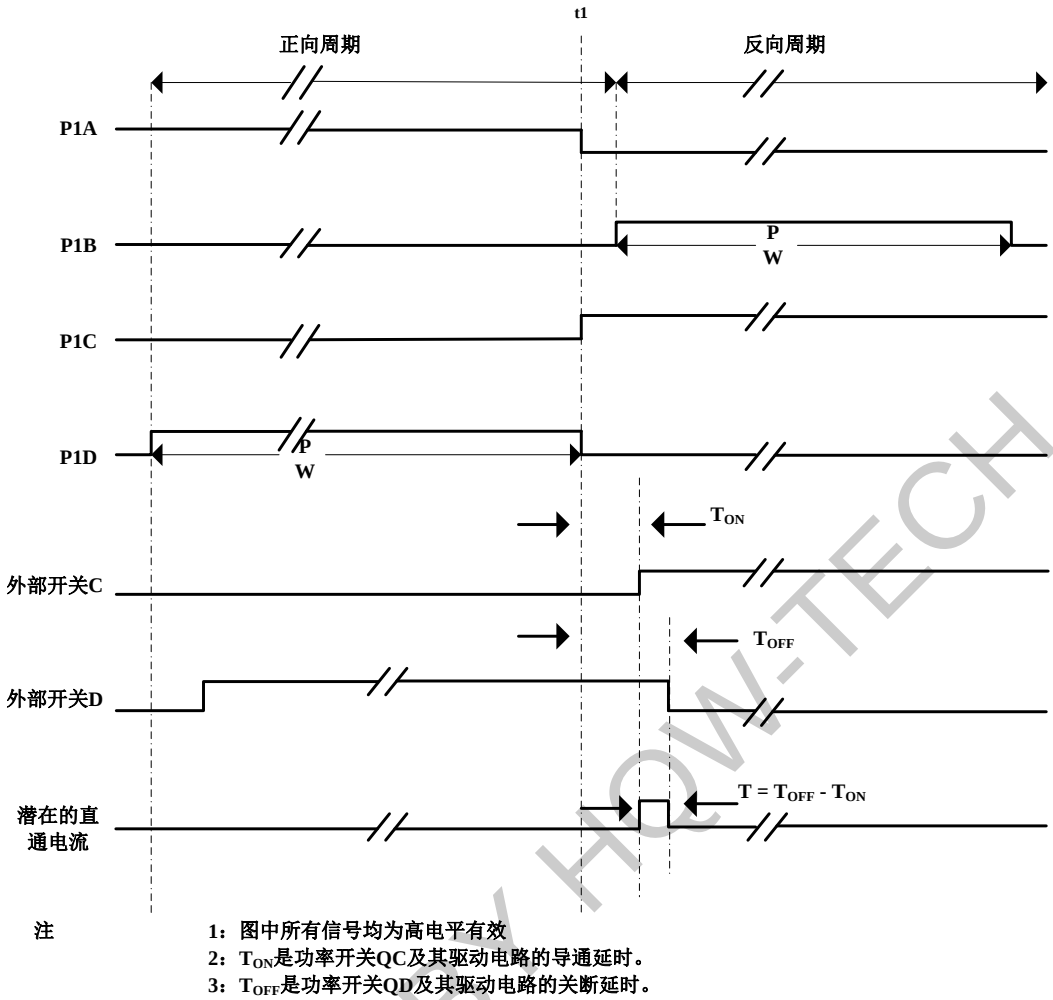


图 24-13： 在占空比接近100% 时改变PWM 方向的示例

24.4.3 增强型 PWM 自动关闭模式

PWM 模式支持自动关闭模式，当外部关闭事件发生时将禁止PWM 输出。自动关闭模式将PWM 输出引脚置于预先确定的状态。该模式用于防止PWM 破坏应用。通过使用CCP1AS 寄存器的CCP1AS<2:0> 位来选择自动关闭源。 关闭事件由以下条件产生：

• INT引脚上的逻辑0
• 比较器（C1）输出上的逻辑1

关闭条件由CCP1AS 寄存器的CCP1ASE （自动关闭事件状态）位指示。如果该位为0， PWM 引脚正常工作。如果该位为1， PWM 输出处于关闭状态。

当关闭事件发生时，会发生以下两件事：

CCP1ASE位被设置为1。CCP1ASE将保持置1 直到由固件清零或发生自动重启。

使能的PWM 引脚被异步置为其关闭状态。 PWM 输出引脚被分组为 [P1A/P1C] 和[P1B/P1D] 对。每对引脚的状态由CCP1AS 寄存器的PSS1AC 和PSS1BD 位决定。每对引脚可设置为以下3 种状态之一：

• 驱动逻辑1
• 驱动逻辑0
• 三态（高阻态）

注 1： 自动关闭条件是基于电平的信号，而不是基于边沿的信号。只要电平存在，自动关闭就将持续。
2： 当自动关闭条件持续时，禁止写CCP1ASE位。
3： 一旦自动关闭条件被移除并且发生PWM重启（通过固件或自动重启），PWM 信号将总是在下一个PWM 周期开始重启。
4： 在由比较器输出或INT 引脚事件引起的自动关闭事件之前，可由固件通过将CCP1AS 寄存器的CCP1ASE 位设置为1触发软件关闭。自动重启特性仅跟踪由比较器输出或INT 引脚事件引起的关闭的有效状态。如果此时使能了自动重启，它将立即清零该位，并且在下一个PWM 周期开始的时候重启ECCP 模块。

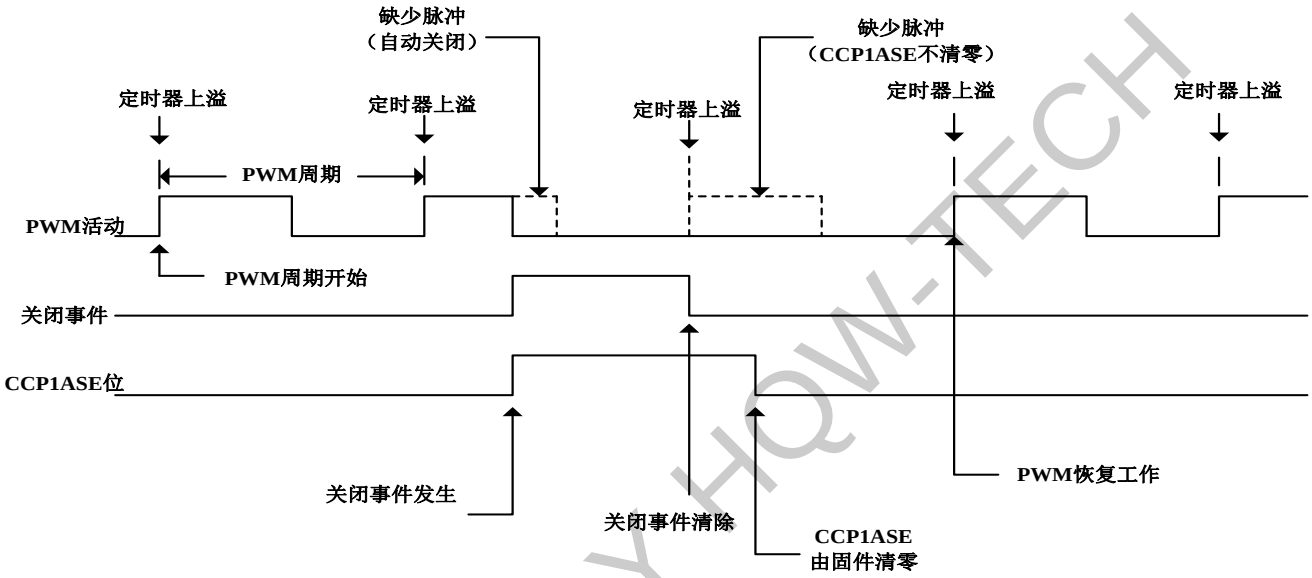


图 24-14： PWM 自动关闭，可由固件重启（P1RSEN = 0）

24.4.4 自动重启模式

一旦自动关闭条件被移除，增强型PWM 可被配置为自动重启PWM 信号。通过将PWM1CON 寄存器中的P1RSEN 位置1 使能自动重启。如果使能了自动重启，只要自动关闭条件有效，CCP1ASE 位将保持置1。当自动关闭条件被移除时，CCP1ASE 位将由硬件清零并继续正常工作。

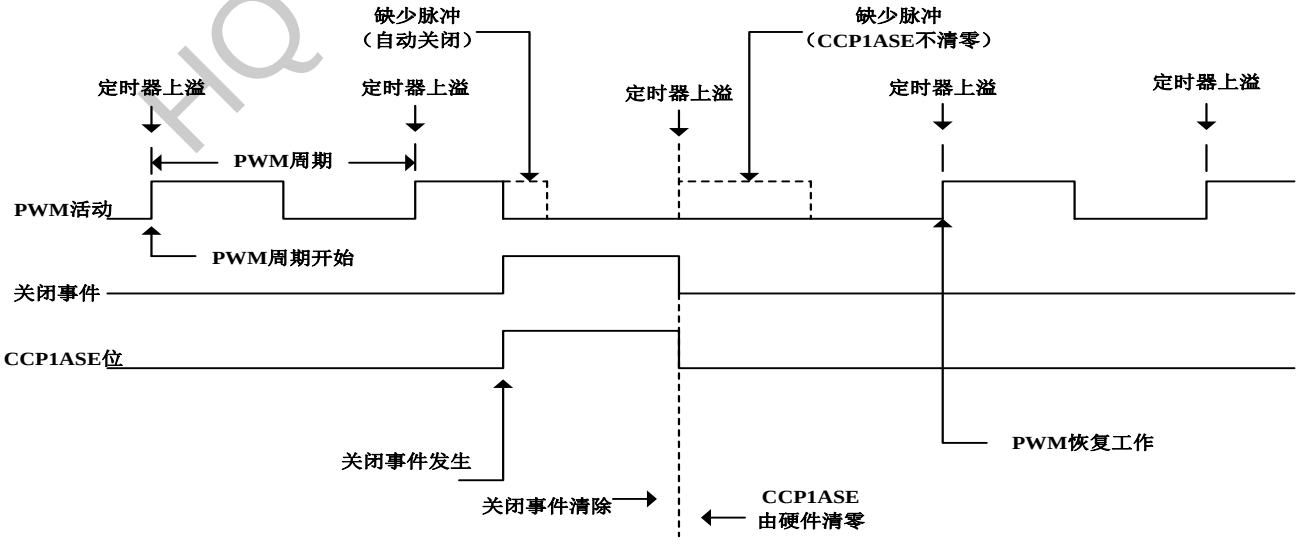
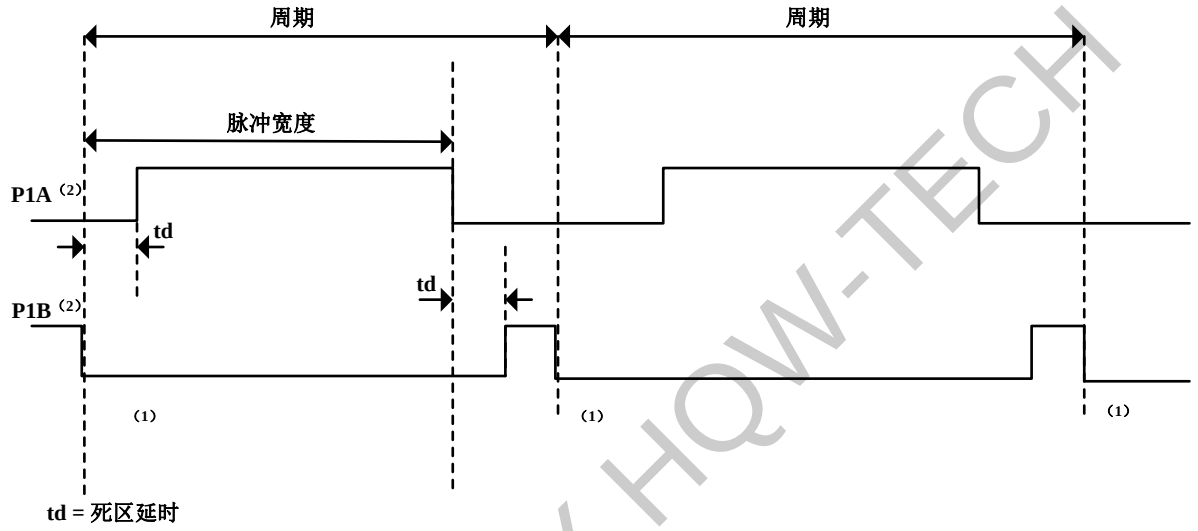


图 24-15： PWM 自动关闭，可自动重启（P1RSEN = 1）

24.4.5 可编程死区延时

在所有功率开关都以PWM 频率调制的半桥应用中，功率开关关断通常比导通需要更多的时间。如果上下两个功率开关同时开关（一个导通，另一个关断），那么在一段很短的时间里，两个开关可能同时导通，直到其中一个开关完全关断为止。在这短暂的时间中，两个功率开关中可能流过较高的电流（ 直通电流），使得该桥式供电电路短路。为避免开关过程中可能会出现的破坏性直通电流，通常需要延迟功率开关的导通，保证在另一个开关完全关断之后，再导通相应的功率开关。在半桥模式下，可采用数字可编程死区延时来避免出现破坏桥式功率开关的直通电流。在信号从无效状态切换到有效状态时发生延时。请参见图24-16。相关的PWM1CON寄存器的低7 位以单片机指令周期（TCY 或4 TOSC）为单位设置延时。



注 1：此时TMR2寄存器和PR2寄存器的值相等
2：图中输出信号为高电平有效

图 24-16： 半桥PWM 输出示例

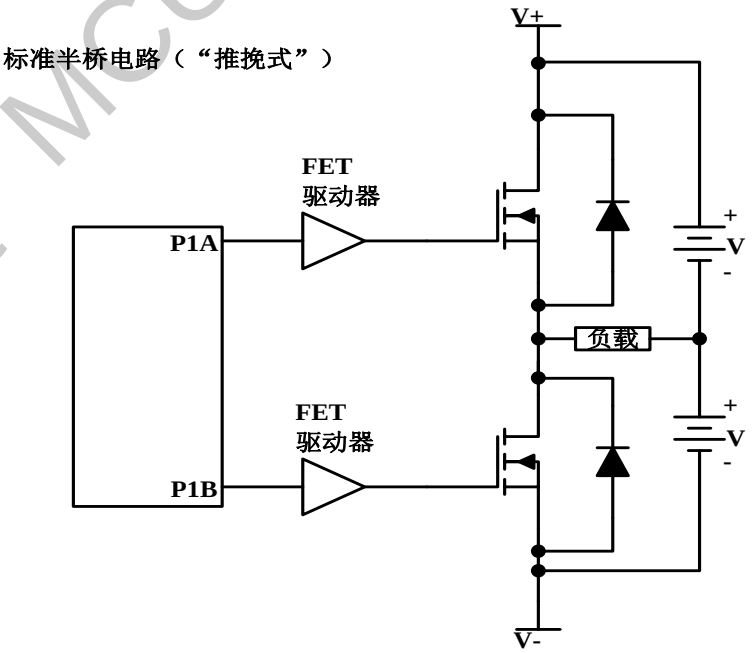


图24-17： 半桥应用示例

24.4.6PWM 转向模式

在单输出模式下，PWM转向允许任何PWM引脚为调制信号。此外，多个引脚上可以同时使用同一PWM 信号。一旦选择了单输出模式（CCP1CON 寄存器的CCP1M<3:2> = 11 且P1M<1:0> = 00），通过将PSTR1CON 寄存器的STR1 位置1，用户固件可将同一PWM信号加到1、2、3或4个输出引脚，如表24-9所示。

注：必须将相关的CPIO 位设置为输出（0）以使能引脚输出驱动器，从而在引脚上看到PWM 信号。

当PWM 转向模式有效时， CCP1CON 寄存器的CCP1M<1:0> 位决定输出引脚的极性。PWM自动关闭操作也适用PWM转向模式，如第24.4.3节“ 增强型PWM 自动关闭模式”中所述。自动关闭事件只对使能PWM 输出的引脚有影响。

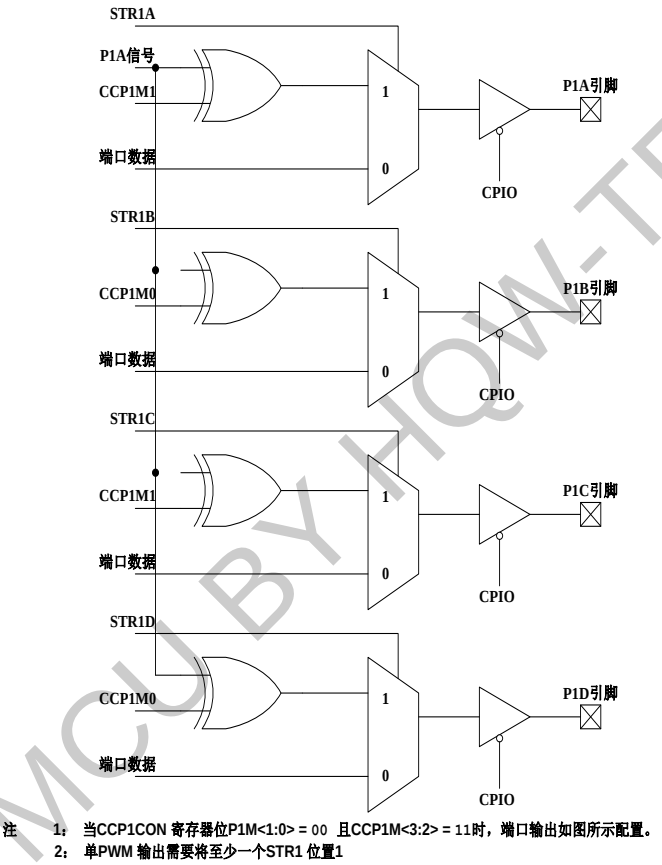


图 24-18： 转向简化框图

24.4.6.1 转向同步

当转向事件发生时，PSTR1CON 寄存器的STR1SYNC位向用户提供两种选择。当STR1SYNC 位为0 时，转向事件将发生在写PSTR1CON 寄存器指令结束时。在这种情况下，输出引脚的输出信号可能是一个不完整的PWM 波形。用户固件需要立即从引脚移除PWM 信号时，该操作非常有用。当STR1SYNC 位为1 时，在下一个PWM 周期开始将发生有效转向更新。此时，转向开/ 关PWM 输出将总是产生一个完整的PWM 波形。图24-19和图24-20 给出了根据STR1SYNC 设置的PWM 转向时序图。

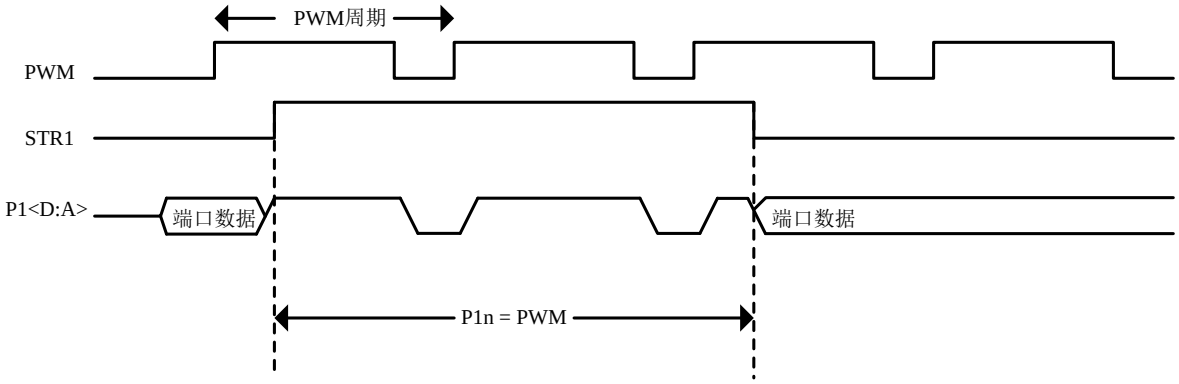


图 24-19： 指令结束时发生的转向事件的示例（STR1SYNC = 0）

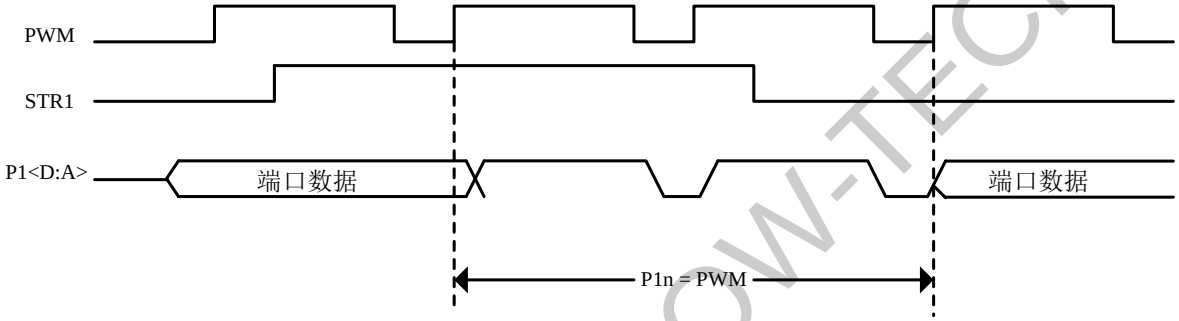


图 24-20： 指令开始时发生的转向事件的示例（STR1SYNC = 1）

24.4.7 启动注意事项

当使用任何PWM 模式时，应用硬件必须在PWM 输出引脚上外接适当的上拉和/ 或下拉电阻。通过CCP1CON 寄存器的CCP1M<1:0> 位，用户可以为每一对PWM 输出引脚（P1A/P1C 和P1B/P1D）选择PWM输出信号是高电平有效还是低电平有效。PWM输出极性必须在使能PWM 引脚输出驱动器之前选择。由于可能导致应用电路损坏，因此不建议在使能PWM引脚输出驱动器的同时修改极性配置。当PWM 模块初始化时，P1A、P1B、P1C 和P1D 输出锁存器可能不在正确的状态。 这样在使能增强型PWM模式的同时使能PWM 引脚输出驱动器，可能损坏应用电路。应首先将增强型PWM 模式配置为正确的输出模式并经过一个完整的PWM 周期之后，再使能PWM 引脚输出驱动器。当第二个PWM 周期开始时，PIFB1 寄存器的TMR2IF位置1表示一个完整的PWM周期结束了。

注：	当单片机退出复位状态时，所有I/O 引脚呈高阻态。外部电路必须保持功率开关器件处于截止状态，直到单片机将I/O 引脚驱动为适当的信号电平，或者激活PWM输出为止。

24.4.8 备用引脚位置

该模块具有以下I/O 引脚：通过使用备用引脚功能寄存器APFCON 可将I/O 引脚转移到其他位置。

24.5 寄存器说明

寄存器293H：CCP1控制寄存器（CCP1CON）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	**P1M0**	**DC1B1**	**DC1B0**	**CCP1M3**	**CCP1M2**	**CCP1M1**	**CCP1M0**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-6 **P1M1<1:0>**：增强型PWM 输出配置位

捕捉模式：

未使用

比较模式：

未使用

PWM 模式：

如果CCP1M<3:2> = 00、01 和10：

xx = P1A 配置为捕捉/ 比较输入； P1B、P1C 和P1D 配置为端口引脚

如果CCP1M<3:2> = 11：

00 = 单输出； P1A 被调制； P1B、P1C 和P1D 配置为端口引脚

01 = 全桥输出正向； P1D 被调制； P1A 有效； P1B 和P1C 无效

10 = 半桥输出； P1A 和P1B 被调制，带死区控制； P1C 和P1D 配置为端口引脚

11 = 全桥输出反相； P1B 被调制； P1C 有效； P1A 和P1D 无效

Bit5-4 **DC1B<1:0>**：PWM 占空比最低有效位

捕捉模式：

未使用

比较模式：

未使用

PWM 模式：

这两位是PWM 占空比的低2 位。高8 位在CCPR1L 中。

bit3-0 **CCP1M<3:0>**：ECCP1 模式选择位

0000 = 捕捉/ 比较/PWM 关闭（复位ECCP1 模块）

0001 = 保留

0010 = 比较模式：发生匹配时翻转输出

0011 = 保留

0100 = 捕捉模式：每个下降沿

0101 = 捕捉模式：每个上升沿

0110 = 捕捉模式： 每4 个上升沿

0111 = 捕捉模式： 每16 个上升沿

1000 = 比较模式：初始化ECCP1引脚为低电平；发生比较匹配时将输出置1（将 CCP1IF 置1）

1001 = 比较模式：初始化 ECCP1 引脚为高电平；发生比较匹配时将输出清零（将 CCP1IF 置1）

1010 = 比较模式： 仅产生软件中断； ECCP1 引脚恢复为I/O 状态

1011 = 比较模式： 特殊事件触发（CCP1复位定时器，将CCP1IF位置1，且如果使能了A/D模块，则启动A/D转换）

PWM 模式：

1100 = PWM 模式：P1A 和P1C 高电平有效； P1B 和P1D 高电平有效
1101 = PWM 模式：P1A 和P1C 高电平有效； P1B 和P1D 低电平有效
1110 = PWM 模式：P1A 和P1C 低电平有效； P1B 和P1D 高电平有效
1111 = PWM 模式：P1A 和P1C 低电平有效； P1B 和P1D 低电平有效

寄存器295H：CCP1自动关闭控制寄存器（CCP1AS）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCP1ASE	**CCP1AS2**	**CCP1AS1**	**CCP1AS0**	**PSS1AC1**	**PSS1AC0**	**PSS1BD1**	**PSS1BD0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **CCP1ASE**：CCP1 自动关闭事件状态位
1 = 发生了关闭事件； CCP1 输出为关闭状态
0 = CCP1 输出正常工作

Bit6-4 **CCP1AS<2:0>**：CCP1 自动关闭源选择位
000 = 禁止自动关闭
001 = 比较器C1 输出高电平(1)
010 = 比较器C2 输出高电平(1)
011 = 比较器C1 或 C2 输出高电平(1)
100 = INT 引脚电压为VIL
101 = INT 引脚电压为 VIL 或比较器C1 输出高电平(1)
110 = INT 引脚电压为 VIL 或比较器C2 输出高电平(1)
111 = INT 引脚电压为 VIL 或者比较器C1 或比较器C2 输出高电平(1)

Bit3-2 **PSS1AC<1:0>**：引脚P1A 和P1C 关闭状态控制位
00 = 驱动引脚P1A 和P1C 为0
01 = 驱动引脚P1A 和P1C 为1
1x = 引脚P1A 和P1C 为三态

Bit1-0 **PSS1BD<1:0>**：引脚P1B 和P1D 关闭状态控制位
00 = 驱动引脚P1B 和P1D 为0
01 = 驱动引脚P1B 和P1D 为1
1x = 引脚P1B 和P1D 为三态

注 **1**：如果C1SYNC 使能，则关闭由Timer1 延时。

寄存器294H：增强型PWM控制寄存器（PWM1CON）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1RSEN	**P1DC6**	**P1DC5**	**P1DC4**	**P1DC3**	**P1DC2**	**P1DC1**	**P1DC0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **P1RSEN**：PWM 重启使能位
1 = 自动关闭时，一旦关闭事件消失， CCP1ASE 位就自动清零； PWM 自动重启
0 = 自动关闭时， CCP1ASE 必须用软件清零以重启PWM

Bit6-0 **P1DC<6:0>**：PWM 延时计数位

P1DC1 = 在PWM信号应该转换为有效的预定时间和转换为有效的实际时间之间的FOSC/4（4 * TOSC）周期数

注 1：双速启动且选择LP、XT或HS作为振荡器模式，或者使能故障保护模式时，该位将复位为0。

寄存器296H：PWM转向控制寄存器（PSTR1CON）(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	**STR1SYNC**	**STR1D**	**STR1C**	**STR1B**	**STR1A**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7-5 **未实现：读为0**

Bit4 **STR1SYNC**：转向同步位
1 = 在下一个PWM 周期发生输出转向更新
0 = 在指令周期边界的开始发生输出转向更新

Bit3 **STRD**：转向使能位D
1 = P1D 引脚的PWM 波形极性受CCP1M<1:0> 控制
0 = P1D 引脚被分配为端口引脚

bit2 **STRC**：转向使能位C
1 = P1C 引脚的PWM 波形极性受CCP1M<1:0> 控制
0 = P1C 引脚被分配为端口引脚

bit1 **STRB**：转向使能位B
1 = P1B 引脚的PWM 波形极性受CCP1M<1:0> 控制
0 = P1B 引脚被分配为端口引脚

Bit0 **STRA**：转向使能位B
1 = P1A 引脚的PWM 波形极性受CCP1M<1:0> 控制
0 = P1A 引脚被分配为端口引脚

注 1： PWM 转向模式仅在CCP1CON 寄存器位CCP1M<3:2> = 11 且P1M<1:0> = 00 时可用。

寄存器1BH：PWM周期寄存器（PR2）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-0 **PR2<7:0>**：PWM 周期寄存器

寄存器291H：捕捉/ 比较/PWM 寄存器1 （LSB）寄存器（CCPR1L）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7-0 **CCPR1L <7:0>**：PWM占空比寄存器

寄存器292H：捕捉/ 比较/PWM（从动） 寄存器1 （MSB）（CCPR1H）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7-0 **CCPR1H <7:0>**：捕捉/ 比较/PWM（从动） 寄存器1 （MSB）

表 23-1：与 ECCP 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSC	—	—	—	—	AN7	AN6	AN5	AN4	xxxx 1111	xxxx 1111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	xx11 1111	xx11 1111
PCPHR	—	—	PCPHR5	PCPHR4	PCPHR3	PCPHR2	PCPHR1	PCPHR0	xx11 1111	xx11 1111
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCP1AS	CCP1ASE	CCP1AS2	CCP1AS1	CCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	0000 0000
CCPR1L	捕捉/比较/PWM 寄存器 1（LSB）								xxxx xxxx	xxxx xxxx
CCPR1H	捕捉/比较/PWM（从动）寄存器 1（MSB）								xxxx xxxx	xxxx xxxx
PWM1CON	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000	0000 0000
PSTR1CON	—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A	xx00 0001	xx00 0001

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

25.0 主同步串行口模块

25.1 主 SSP（MSSP1）模块概述

主同步串行端口（MSSP1）模块是用于同其他外设或单片机进行通信的串行接口。这些外设可以是串行EEPROM、移位寄存器、显示驱动器和A/D 转换器等。MSSP1 模块有以下两种工作模式：

- 串行外设接口（Serial Peripheral Interface，SPI）
- I2C

SPI 接口支持以下模式和特性：

- 主模式
- 从模式
- 时钟极性
- 从选择同步（仅限从模式）
- 从器件的菊花链连接

图25-1 给出了SPI 接口模块的框图。

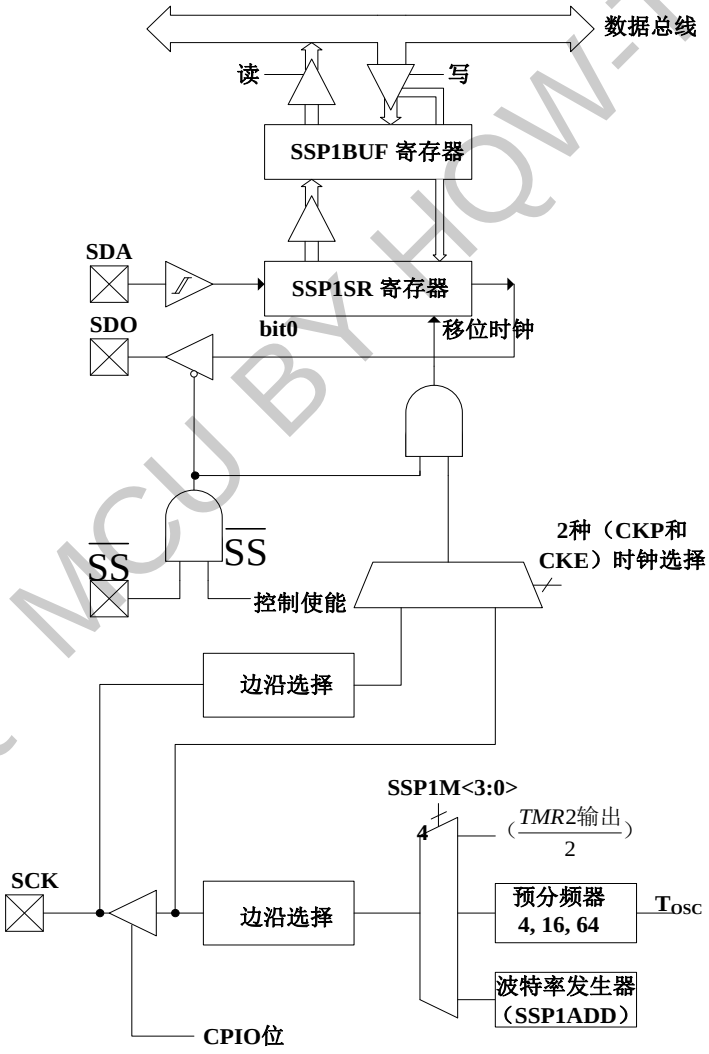


图25-1： MSSP1 框图（SPI 模式）

I²C 接口支持以下模式和特性：

- 主模式
- 从模式
- 字节无应答（从模式）
- 有限多主器件支持
- 7 位和 10 位寻址
- 启动和停止中断
- 中断屏蔽
- 时钟延长
- 总线冲突检测
- 广播呼叫地址匹配
- 地址掩码
- 地址保持模式和数据保持模式
- 可选的 SDA 保持时间

图 25-2 给出了主模式下 I²C 接口模块的框图。图 25-3 给出了从模式下 I²C 接口模块的框图。

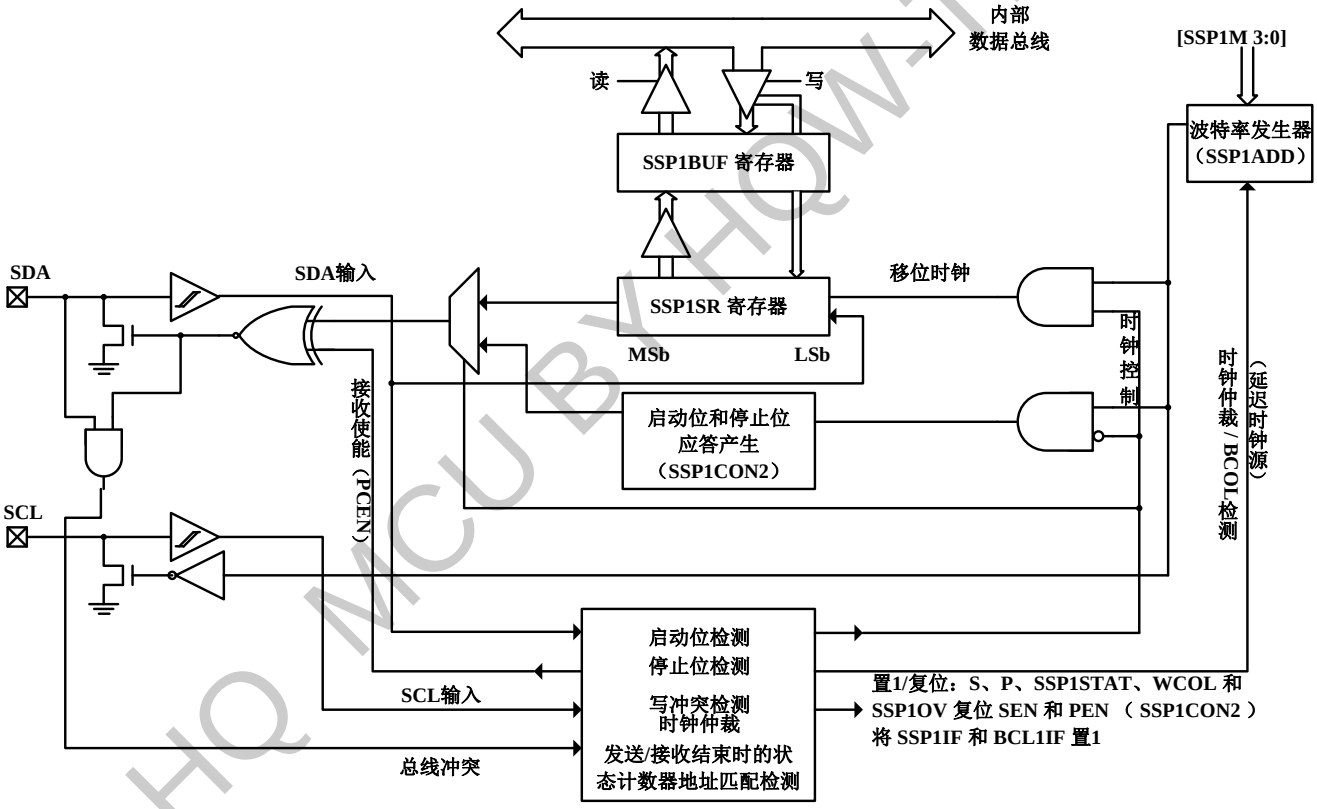


图25-2： MSSP1 框图（I2C主模式）

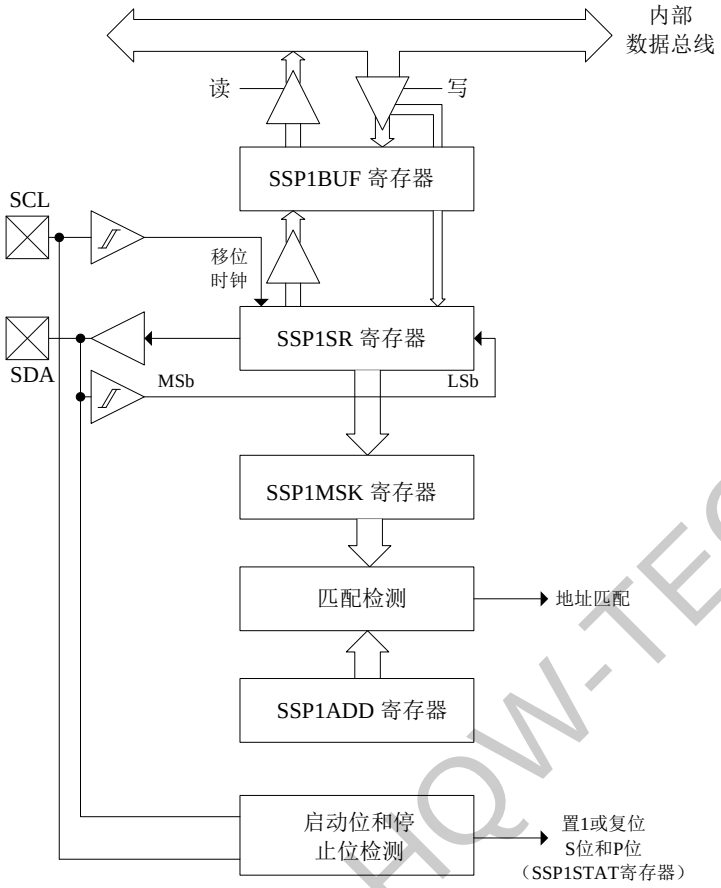


图25-3： MSSP1 框图（I2C从模式）

25.2 SPI 模式概述

串行外设接口（SPI）总线是以全双工模式工作的同步串行数据通信总线。器件在由主器件启动通信的主/从器件环境中进行通信。从器件通过称为从选择的片选进行控制。

SPI 总线规定了4 种信号连接：

- 串行时钟（SCK）
- 串行数据输出（SDO）
- 串行数据输入（SDI）
- 从选择（$\overline{SS}$）

图25-1给出了MSSP1模块在SPI模式下工作时的框图。SPI 总线工作时使用单个主器件和一个或多个从器件。使用多个从器件时，从主器件到每个从器件都需要独立的从选择连接。图25-4给出了主器件和多个从器件之间的典型连接。主器件每次仅选择一个从器件。大多数从器件都具有三态输出，所以在未选择它们时，它们的输出信号会看起来好像与总线断开。

数据发送涉及到两个移位寄存器，它们大小都为8 位，一个在主器件中，一个在从器件中。不论是对于主器件还是从器件，数据总是每次移出一位，最高有效位（MSb）先移出。与此同时，新的最低有效位（LSb）会被移入同一寄存器。图25-5给出了分别配置为主器件和从器件的两个处理器之间的典型连接。数据在所设定的时钟边沿从两个移位寄存器移出，并在相反的时钟边沿锁存。

主器件通过它的SDO 输出引脚上发送信息，并由该引脚所连接的从器件SDI 输入引脚接收。从器件通过它的SDO输出引脚上发送信息，并由该引脚所连接的主器件SDI 输入引脚接收。要开始进行通信，主器件需要先送出时钟信号。主器件和从器件应配置为相同的时钟极性。主器件会通过从它的移位寄存器中发送MSb 而启动数据发送。从器件会从同一条线上读取该位，并将它保存到其移位寄存器的LSb 单元中。在每个SPI 时钟周期中，会发生全双工数据发送。这意味着，在主器件从其移位寄存器中发送出MSb （在其SDO 引脚上），从

器件读取该位并将它保存为其移位寄存器的LSb 的同时，从器件也会从其移位寄存器中发送出MSb（在其SDO引脚上），而主器件也会读取该位并将它保存为其移位寄存器的LSb。在移出8 位之后，主器件和从器件就交换了寄存器值。如果需要交换更多数据，移位寄存器中会装入新数据，并重复该过程。

数据是否有意义（无效数据），取决于应用软件。这就导致以下三种数据传输情形：

- 主器件发送有用数据，从器件发送无效数据。
- 主器件发送有用数据，从器件发送有用数据。
- 主器件发送无效数据，从器件发送有用数据。

数据发送可能会需要不定数量的时钟周期。在没有更多数据需要发送时，主器件会停止发送时钟信号，并取消选择从器件。每个与总线连接、但未被通过其从选择线选择的从器件，都必须忽略时钟和数据发送信号，并且不能发送自己的任何数据。

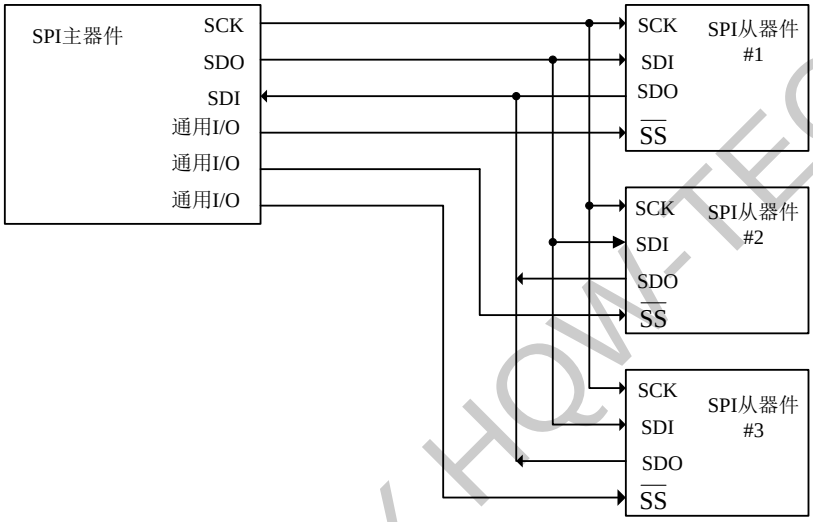


图25-4： SPI 主器件和多个从器件连接

25.2.1 SPI 模式寄存器

MSSP1 模块有6 个寄存器用于SPI 模式操作。这些寄存器是：

- MSSP1 状态寄存器（SSP1STAT）
- MSSP1 控制寄存器1 （SSP1CON1）
- MSSP1 控制寄存器3 （SSP1CON3）
- MSSP1 数据缓冲寄存器（SSP1BUF）
- MSSP1 地址寄存器（SSP1ADD）
- MSSP1 移位寄存器（SSP1SR）（不可直接访问）

SSP1CON1 和SSP1STAT 是SPI 模式操作下的控制寄存器和状态寄存器。SSP1CON1 寄存器是可读写的。SSP1STAT 的低6 位是只读的。SSP1STAT 的高2 位是可读写的。在一种SPI 主模式下，SSP1ADD中可以装入在波特率发生器中使用的值。关于波特率发生器的更多信息，请参见第25.7 节“ 波特率发生器”。SSP1SR是用来将数据移入和移出的移位寄存器。SSP1BUF 用于间接访问SSP1SR 寄存器。SSP1BUF是缓冲寄存器，可用于数据字节的写入或读出。在接收操作中，SSP1SR 和SSP1BUF 共同构成一个缓冲接收器。当SSP1SR接收到一个完整字节时，字节会被传输到SSP1BUF 中，并且SSP1IF 中断标志会置1。在发送期间， SSP1BUF不是可缓冲的。对SSP1BUF的写操作将同时写入SSP1BUF 和SSP1SR。

25.2.2 SPI 模式操作

初始化SPI 时需要指定几个选项。可以通过编程相应的控制位（SSP1CON1<5:0> 和SSP1STAT<7:6>）来指定这些选项。这些控制位用于指定以下选项：

- 主模式（SCK1 作为时钟输出）
- 从模式（SCK1 作为时钟输入）
- 时钟极性（SCK1 的空闲状态）
- 数据输入采样阶段（数据输出时间的中间或末尾）
- 时钟边沿（在SCK1 的上升沿/ 下降沿输出数据）
- 时钟速率（仅限主模式）
- 从选择模式（仅限从模式）

要使能串口， SSP1CON1 寄存器的SSP1 使能位SSP1EN 必须置1。要复位或重新配置SPI 模式，先将SSP1EN 位清零，重新初始化SSP1CONx 寄存器，然后再将SSP1EN 位置1。这会将SDI、SDO、SCK 和$\overline{SS}$引脚配置为串口引脚。要将上述引脚用于串口功能，必须正确设置其中一些引脚的数据方向位（在CPIO 寄存器中）：

- SDI 必须将相应的CPIO 位置1
- SDO 必须将相应的CPIO 位清零
- SCK（主模式）必须将相应的CPIO 位清零
- SCK（从模式）必须将相应的CPIO 位置1
- $\overline{SS}$ 必须将相应的CPIO 位置1

对于不需要的串口功能，可通过将相应的数据方向（CPIO）寄存器编程为相反值来改写。MSSP1 由一个发送/ 接收移位寄存器（SSP1SR）和一个缓冲寄存器（SSP1BUF）组成。SSP1SR 将数据移入/ 移出器件，先移位MSb。SSP1BUF 会一直保存先前写入SSP1SR 的数据，直到当前接收数据就绪为止。一旦8位数据接收完毕，该字节就被移入SSP1BUF寄存器。然后，SSP1STAT寄存器的缓冲区满检测位BF和中断标志位SSP1IF被置1。这种双重缓冲数据接收方式（SSP1BUF）允许在读取刚接收的数据之前就开始接收下一个字节。当SSP1BUF 寄存器正在发送/ 接收数据时，对它写入的任何数据都将被忽略，同时SSP1CON1 寄存器的写冲突检测位WCOL 被置1。用户软件必须将WCOL位清零才能使以后对SSP1BUF寄存器的写入成功完成。

为确保应用软件能接收有效数据，在下一个要发送的数据字节写入SSP1BUF 之前，读取SSP1BUF 中现有的数据。SSP1STAT寄存器的缓冲区满位BF 用于指示何时SSP1BUF 装入了接收到的数据（发送完成） 。SSP1BUF 中的数据被读取后，BF 位被清零。如果SPI仅作为一个发送器，则不必理会该数据。一般来说，MSSP1 中断用于检测发送/ 接收何时结束。如果不打算使用中断方法，用软件查询的方法同样可确保不会发生写冲突。

SSP1SR 不能直接读写，只能通过寻址 SSP1BUF 寄存器来进行访问。此外，SSP1STAT 寄存器用于指示各种状态条件。

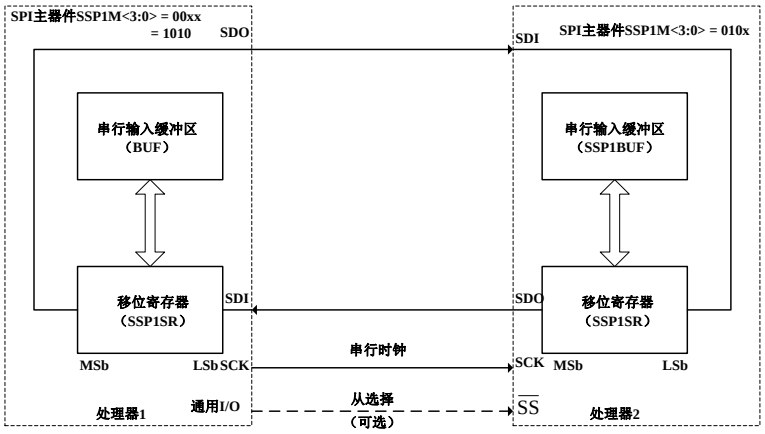


图25-5： SPI 主/从器件连接

25.2.3 SPI 主模式

因为主器件控制 SCK 线，所以它可以在任意时刻启动数据传输。主器件根据软件协议确定从器件（图 25-5 中的处理器 2）在何时广播数据。在主模式下，数据一写入 SSP1BUF 寄存器就发送/ 接收。如果只打算将 SPI 作为接收器，则可以禁止 SDO 输出（将其编程为输入）。SSP1SR 寄存器按所设定的时钟速率，对 SDI 引脚上的信号进行连续移入。每接收到一个字节，就将其装入 SSP1BUF 寄存器，就像接收到普通字节一样（中断和状态位相应置 1）。通过适当地设定 SSP1CON1 寄存器的 CKP 位和 SSP1STAT 寄存器的 CKE 位，可以选择时钟极性。图 25-6、图 25-8、图 25-9 和图 25-10 给出了 SPI 通信的波形图，其中 MSB 先发送。在主模式下，SPI 时钟速率（比特率）可由用户编程为以下几种之一：

- FOSC/4 （或TCY）
- FOSC/16 （或4 * TCY）
- FOSC/64 （或16 * TCY）
- Timer2 输出/2
- FOSC/(4 * (SSP1ADD + 1))

图25-6 给出了主模式的波形图。

当CKE 位置1 时， SDO 数据在SCK 上出现时钟边沿前一直有效。图中所示的输入采样的变化由SMP 位的状态反映。图中给出了将接收到的数据装入SSP1BUF的时刻。

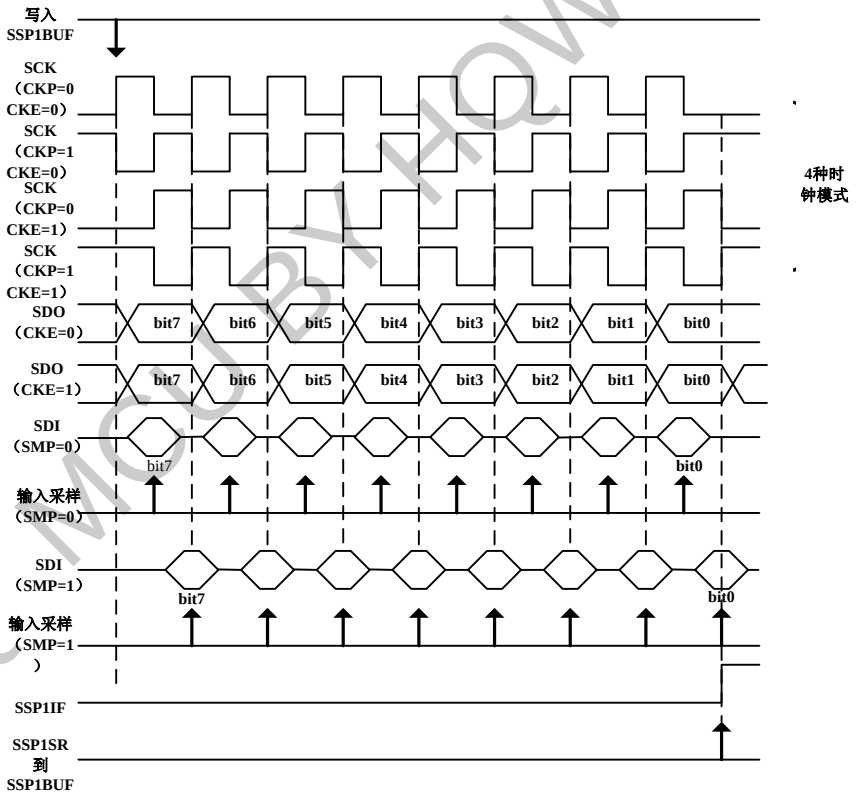


图25-6： SPI 模式波形图（主模式）

25.2.4 SPI 从模式

在从模式下，当SCK 上出现外部时钟脉冲时，发送和接收数据。锁存最后一位数据之后， SSP1IF 中断标志位会置1。在SPI 从模式下使能该模块前，时钟线必须处于相应的空闲状态。时钟线可通过读SCK 引脚来查看。空闲状态由SSP1CON1 寄存器的CKP 位决定。在从模式下，外部时钟由SCK 引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。在休眠模式下，从器件仍可发送/ 接收数据。移位寄存器通过SCK 引脚输入提供时钟，当接收到一个字节时，器件会产生中断。如果允许发生中断，器件会从休眠模式唤醒。

25.2.4.1 菊花链配置

SPI 总线有时会采用菊花链配置进行连接。第一个从器件的输出与第二个从器件的输入连接，第二个从器件的输出与第三个从器件的输入连接，如此类推。最后一个从器件的输出与主器件的输入连接。在第二组时钟脉冲期间，每个从器件会送出在第一组时钟脉冲期间所接收数据的精确副本。整个链充当一个很大的通信移位寄存器。菊花链功能只需要从主器件引出一条从选择线。图25-7 给出了在SPI 模式下工作时典型菊花链连接的框图。

在菊花链配置中，从器件只需要总线上最近的一个字节。将SSP1CON3 寄存器的BOEN 位置1 时，即使尚未读取前一个字节，也允许数据写入SSP1BUF 寄存器。这使软件可以忽略不适用于它的数据。

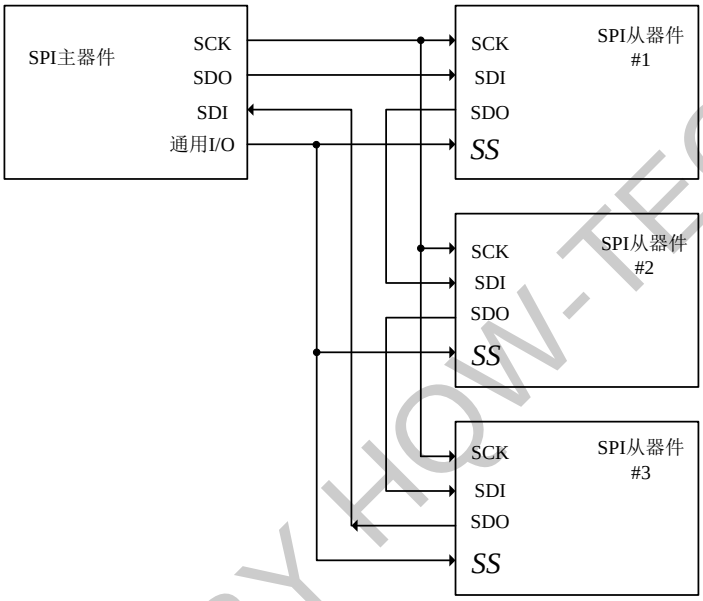


图25-7： SPI 菊花链连接

25.2.5 SPI 从选择同步

从选择也可以用于对通信进行同步。从选择线会一直保持高电平，直到主器件准备好进行通信。当从选择线下拉为低电平时，从器件就知道新的数据发送正在启动。如果从器件未能正确地接收到通信，它会在从选择线恢复为高电平状态、数据发送结束时发生复位。然后，从器件会在从选择线再次下拉为低电平时准备好接收新的发送数据。如果不使用从选择线，则会存在从器件最终与主器件脱离同步的风险。如果从器件丢失了某个位，则在之后的数据发送中，它将总是偏离一位。使用从选择线可以让从器件和主器件在每次发送开始时相互对齐。

SS引脚允许器件工作于同步从模式。SPI 必须处于从模式，并使能SS控制（SSP1CON1<3:0> = 0100）。当SS引脚为低电平时，使能数据的发送和接收，同时驱动SDO 引脚。当SS引脚变为高电平时，即使是在字节的发送过程中，也不再驱动SDO 引脚，而是将其变成悬空输出状态。根据具体应用，可能需要使用外部上拉/ 下拉电阻。

当SPI 模块复位时，位计数器被强制为0。这通过强制将$\overline{SS}$引脚拉为高电平或将SSP1EN 位清零来实现。

注	1：当SPI 处于从模式且使能SS 引脚控制（SSP1CON1<3:0> = 0100）时，如果SS引脚设置为VDD， SPI 模块将会复位。 2：当SPI 用于从模式且CKE 置1 时，用户必须使能SS引脚控制。 3：工作于SPI 从模式时，SSP1STAT 寄存器的SMP 位必须保持清零。

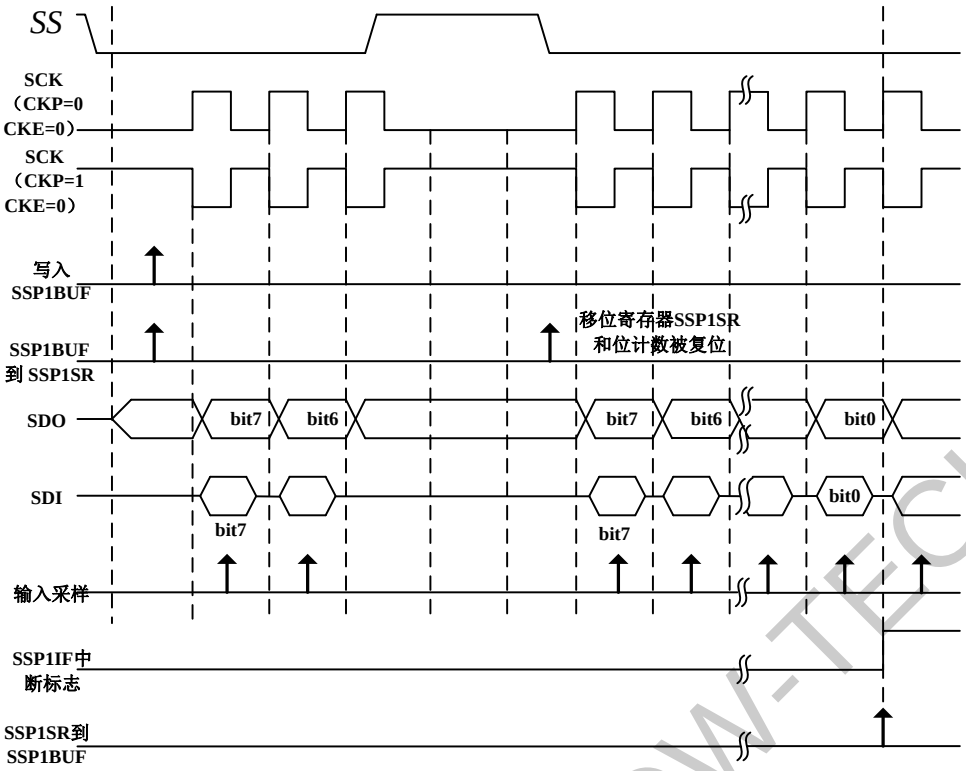


图25-8： 从选择同步波形图

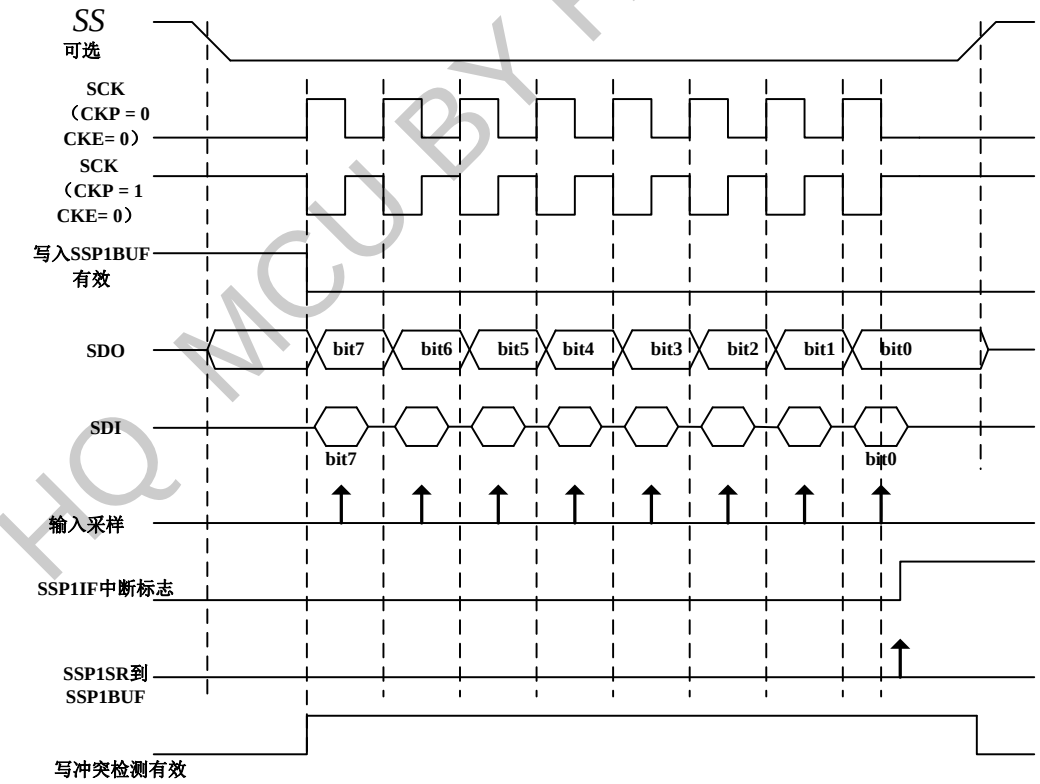


图25-9： SPI 模式波形图（从模式， CKE = 0）

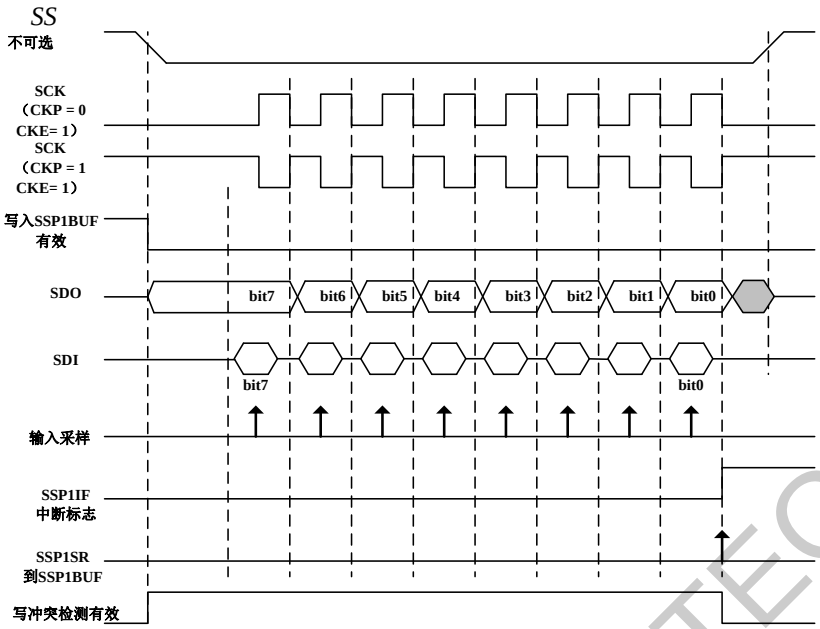


图25-10： SPI 模式波形图（从模式， CKE = 1）

25.2.6 休眠模式下的 SPI 操作

在SPI 主模式下，模块时钟速度与全功耗模式下的不同；处于休眠模式时，所有时钟都暂停。在MSSP1 时钟速度远高于系统时钟时，用户需要特别小心。在从模式下，当允许MSSP1 中断时，在主器件发送完数据时，MSSP1 中断会将控制器从休眠状态唤醒。如果不想从休眠模式退出，应该禁止MSSP1 中断。在SPI 主模式下，当选择休眠模式时，所有模块的时钟都将暂停，并且在器件被唤醒前，发送/ 接收将保持此暂停状态。器件返回到运行模式之后，模块将恢复发送和接收数据。在SPI 从模式下， SPI 发送/ 接收移位寄存器与器件异步工作。这可使器件置于休眠模式，仍能将数据移入SPI 发送/ 接收移位寄存器。当接收到全部8 位数据时，MSSP1 中断标志位将置1，并且如果允许中断的话，将唤醒器件。

25.3 I2C 模式概述

I^2C 是一种多主器件串行数据通信总线。器件在由主器件启动通信的主/ 从器件环境中进行通信。从器件通过寻址进行控制。I^2C 总线规定了两种信号连接：

- 串行时钟（SCL）
- 串行数据（SDA）

图25-2和图25-3 给出了MSSP1 模块在I2C模式下工作时的框图。SCL 和SDA 连接都是双向的漏极开路线路，它们都需要使用用于电源电压的上拉电阻。线路下拉为地电压时，信号视为逻辑0 ；线路保持悬空时，信号视为逻辑1。图25-11 给出了分别配置为主器件和从器件的两个处理器之间的典型连接。I^2C 总线工作时可以有一个或多个主器件，以及一个或多个从器件。对于给定器件，有4 种可能的工作模式：

- 主发送模式
（主器件向从器件发送数据）
- 主接收模式
（主器件从从器件接收数据）
- 从发送模式
（从器件向主器件发送数据）
- 从接收模式
（从器件从主器件接收数据）

要开始进行通信，主器件需要以主发送模式启动。主器件送出启动位，后面跟随它希望进行通信的从器件的地址字节。后面再跟随单个读/ 写位，该位决定主器件是向从器件发送数据还是从从器件接收数据。

如果总线上存在所请求的从器件，从器件会使用应答位（也称为ACK）进行响应。然后，主器件会以发送模式或接收模式继续通信，从器件则以互补模式（分别为接收模式或发送模式）继续通信。

启动位由SCL 线保持为高电平时SDA 线的由高至低跳变来指示。地址和数据字节随后送出，先发送最高有效位（MSb）。在主器件希望从从器件读取数据时，送出的读/ 写位为逻辑1，在主器件希望向从器件写入数据时，该位为逻辑0。

应答位（ACK）是低电平有效信号，它会将SDA 线保持为低电平，用于指示发送器，从器件已接收到发送数据，并已准备好接收更多数据。

数据位的跳变总是在SCL线保持低电平时执行。在SCL线保持高电平时发生的跳变用于指示启动位和停止位。如果主器件希望向从器件写入数据，则它会重复发送一个字节的数据，而从器件则在接收每个字节之后使用 ACK位进行响应。在该示例中，主器件处于主发送模式，从器件处于从接收模式。如果主器件希望从从器件读取数据，则它会从从器件重复接收一个字节的数据，并在接收每个字节之后使用 ACK位进行响应。在该示例中，主器件处于主接收模式，从器件处于从发送模式。

在传输最后一个数据字节之后，主器件可以通过发送停止位来结束数据发送。如果主器件处于接收模式，它会发送停止位来代替最后一个ACK位。停止位由SCL 线保持为高电平时SDA 线的由低至高跳变来指示。

在某些情况下，主器件可能希望维持对总线的控制，并重新启动另一次数据发送。如果是这样，主器件可以在它处于接收模式时，发送另一个启动位来代替停止位或最后一个ACK位。

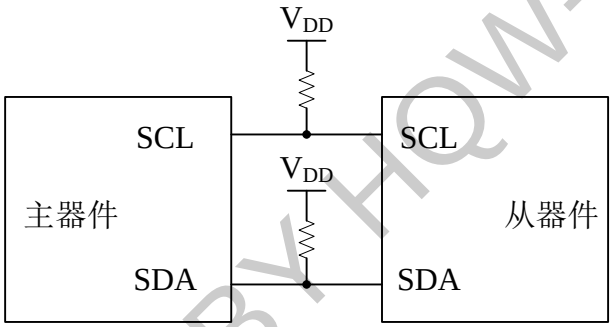


图25-11： I2C 主/ 从器件连接

I^2C 总线规定了三种报文协议：

- 主器件向从器件写数据的单一报文。
- 主器件从从器件读数据的单一报文。
- 主器件对一个或多个从器件启动至少两次写操作或读操作，或者读写操作组合的组合报文。

在一个器件发送逻辑1 （或将线路保留悬空），第二个器件发送逻辑0 （或将线路保持为低电平）时，第一个器件会检测到线路不为逻辑1。这种检测在用于SCL 线时，称为时钟延长。时钟延长为从器件提供了一种控制数据流的机制。这种检测在用于SDA 线时，称为仲裁。仲裁可以确保任意时刻只有一个主器件在进行通信。

25.3.1 时钟延长

在从器件尚未完成数据处理时，它可以通过时钟延长这一过程来延迟更多数据的传输。寻址到的从器件可以在接收或发送一位数据之后将SCL时钟线保持为低电平，指示它尚未准备好继续。与从器件进行通信的主器件将会尝试上拉SCL 线，以传输下一位数据，但它会检测到时钟线尚未被释放。由于SCL 连接是漏极开路，所以从器件可以一直将线路保持为低电平，直到它准备好继续通信为止。通过时钟延长，无法与发送器保持同速的接收器可以控制传入数据流。

25.3.2 仲裁

每个主器件都必须监视总线上是否出现启动位和停止位。如果器件检测到总线正忙，则在总线恢复为空闲状态之前，它无法开始新的报文。

但是，可能会有两个主器件尝试同时或近乎同时启动数据发送。发生这种情况时，将会开始仲裁过程。每个发送器会检查SDA 数据线的电平，并将它与自己期望的电平进行比较。发现两个电平不匹配的第一个发送器

会在仲裁中失败，必须停止在SDA 线上发送数据。

例如，如果一个发送器将SDA 线保持为逻辑1 （保留悬空），而第二个发送器将它保持为逻辑0 （下拉为低电平），则结果是SDA 线将为低电平。那么，第一个发送器会发现线路电平与期望电平不同，并断定有另一个发送器正在进行通信。

发现电平不同的第一个发送器将是仲裁失败的发送器，必须停止驱动SDA 线。如果该发送器同时也是主器件，则它还必须停止驱动SCL 线。然后，它可以在尝试重新启动数据发送之前监视线路上是否出现停止条件。与此同时，另一个未发现期望电平与SDA 线实际电平不同的器件将继续原来的数据发送。它可以无需进行任何复杂处理，因为到目前为止，发送条件与所期望的完全相同，没有其他发送器对报文产生干扰。

当主器件对多个从器件进行寻址时，也会对从发送模式进行仲裁，但这种情况较少见。如果有两个主器件在地址阶段向两个不同的从器件发送报文，则发送较小从器件地址的主器件总是会在仲裁中获胜。当两个主器件向同一从器件地址发送报文时，地址有时会指向多个从器件，仲裁过程必须继续进入到数据阶段。仲裁通常极少发生，但它是正确支持多主器件所必需的过程。

25.4 I2C 模式操作

所有MSSP1 I^2C 通信都是针对字节的，并且会先移出MSb。有6 个SFR 寄存器和2 个中断标志用作模块与单片机和用户软件的接口。模块通过两个引脚SDA 和SCL 来与其他外部I^2C 器件进行通信。

25.4.1 字节格式

I^2C 中的所有通信都采用9 位形式。从主器件向从器件（或者反之）发送一个字节之后，将会送回一个应答位。在SCL 线第8 个下降沿之后，在SDA 上输出数据的器件会将该引脚改为输入，并在下一个时钟脉冲读入应答值。时钟信号SCL由主器件提供。在SCL信号为低电平时，数据可以有效地更改，并且在时钟上升沿进行采样。在SCL线为高电平时，SDA线上的电平变化定义总线上的一些特殊条件，以下会对此进行说明。

25.4.2 I2C 术语的定义

在I^2C 通信的描述中存在一些用语和术语，它们具有特定于I^2C 的定义。下面定义了词语的用法，在本文档其他部分中，将不加说明地使用它们。

表 25-1：I2C 总线术语

术语	说明
发送器	将数据移送到总线上的器件。
接收器	从总线上移入数据的器件。
主器件	启动数据传输、产生时钟信号和终止数据传输的器件。
从器件	主器件寻址到的器件。
多主器件	有多个器件可以启动数据传输的总线。
仲裁	用于确保每次只有一个主器件控制总线的过程。仲裁获胜可以确保报文不会被损坏。
同步	用于将总线上两个或更多器件的时钟进行同步的过程。
空闲	没有任何主器件在控制总线，并且SDA和SCL线均为高电平。
有效	每当有一个或多个主器件在控制总线时。
可寻址的从器件	已接收到匹配地址、并且正在由主器件提供时钟的从器件
匹配地址	送入从器件中，并与SSP1ADD中的存储值匹配的地址字节
写请求	从器件接收到R/$\overline{W}$位清零的匹配地址，并已准备好移入数据。
读请求	主器件发送R/$\overline{W}$位置1的地址字节，表示要求从器件在时钟控制下将数据移出。从器件在接收到该地址字节后会立即移出所有数据字节，直到发生重复启动或停止条件。
时钟延长	总线上的器件通过将SCL保持为低电平来暂停通信的时间
总线冲突	每当模块进行输出并期望SDA线为高电平，却采样到SDA线为低电平时。

25.4.3 SDA 和 SCL 引脚

在SSP1EN 位置1 的情况下选择任意I²C 时， SCL 和SDA 引脚将会强制设为漏极开路。用户应通过将相应的CPIO 位置1 来将这些引脚设置为输入。

注： 在使能 I²C 模式时，数据将设为输出 0。

25.4.4 SDA 保持时间

SDA引脚的保持时间通过SSP1CON3寄存器的SDAHT位进行选择。保持时间是SDA在SCL的下降沿之后保持有效的时间。将SDAHT位置1可以选择最低300ns的较长保持时间，这对于电容较大的总线会有帮助。

25.4.5 启动条件

I²C 规范将启动条件定义为在SCL 线为高电平时，SDA从高电平变为低电平状态。启动条件总是由主器件产生，指示总线从空闲状态变为有效状态。图25-12 给出了启动和停止条件的波形图。
如果模块在将SDA 线置为低电平之前采样到SDA 线为低电平，则会在产生启动条件时发生总线冲突。这一点不符合I²C 规范，该规范规定不能在产生启动条件时发生总线冲突。

25.4.6 停止条件

停止条件定义为在SCL 线为高电平时， SDA 线从低电平变为高电平状态。

注： 在停止条件生效之前，必须至少出现一个SCL 低电平时间，因此，如果SDA 线变为低电平然后再次变为高电平，而SCL 线保持高电平，则只会检测到启动条件。

25.4.7 重复启动条件

重复启动条件在每次停止条件有效的时候有效。如果主器件希望在终止当前传输之后保持总线，主器件可以发出重复启动条件。重复启动对从器件产生的影响与启动条件相同，即复位所有从器件逻辑并使之准备接收一个地址。主器件可以寻址同一个或另一个从器件。图25-13给出了重复启动条件的波形图。

在10 位寻址从模式下，要从寻址到的从器件中移出数据，主器件需要产生重复启动条件。从器件完全寻址（高地址字节和低地址字节均匹配）之后，主器件可以发出重复启动条件和R/W位置1 的高地址字节。然后，从器件逻辑会保持时钟，并准备送出数据。在10 位模式下，与R/W清零的地址字节完全匹配后，前一次预先匹配标志会置1 并保持置1。在产生停止条件之前，R/W清零的高地址或高地址匹配都会失败。

25.4.8 启动/停止条件中断屏蔽

SSP1CON3寄存器的SCIE 和PCIE 位可以用于允许在通常不支持中断功能的从模式下产生中断。对于已允许启动和停止检测中断的从模式，这两位没有任何作用。

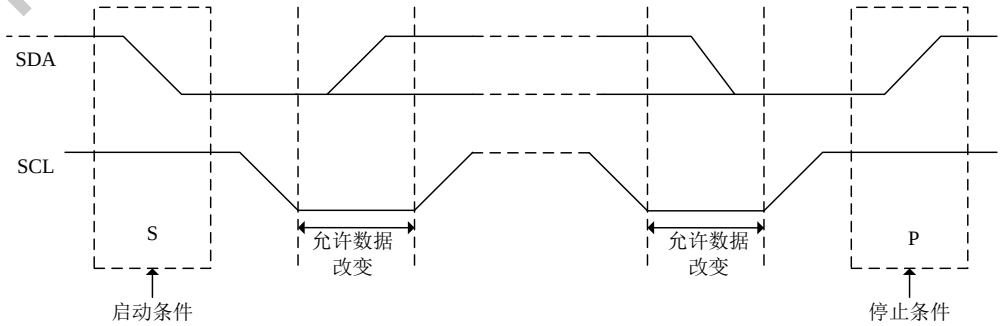


图25-12： I2C 启动和停止条件

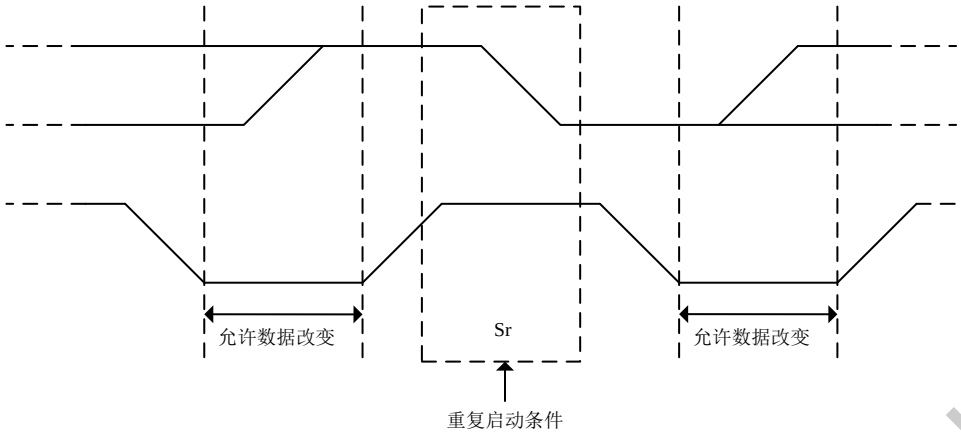


图25-13： I2C 重复启动条件

25.4.9 应答序列

在I²C 中，所有传输字节的第9 个SCL 脉冲都专门用作应答信号。它使接收器件可以通过将SDA 下拉为低电平来响应发送器。发送器在该时间内必须释放对线路的控制，以移入响应信号。应答（ACK）是低电平有效信号，它会将SDA 线下拉为低电平，用于指示发送器器件已接收到发送数据并已准备好接收更多数据。ACK的结果会被放入SSP1CON2 寄存器的ACKSTAT位中。当AHEN 和DHEN 位置1 时，从器件软件允许用户设置要回送到发送器的ACK值。用户可以通过置1/ 清零SSP1CON2 寄存器的ACKDT 位来决定响应。

如果SSP1CON3 寄存器的AHEN 和DHEN 位清零，从器件硬件会产生ACK响应。有一些条件下，从器件不会发送ACK。如果在接收到数据字节时， SSP1STAT 寄存器的BF 位或SSP1CON1寄存器的SSP1OV 位置1。对模块进行寻址时，在总线上的第8 个SCL 下降沿之后，SSP1CON3 寄存器的ACKTIM 位会置1。ACKTIM位指示有效总线的应答时间。ACKTIM 状态位仅在AHEN 位或DHEN 位使能时有效。

25.5 I2C 从模式操作

MSSP1 从模式可以在4 种模式下工作，这些模式通过SSP1CON1寄存器的SSP1M 位进行选择。这些模式可以分为7 位和10 位寻址模式。10 位寻址模式的工作方式与7 位寻址模式相同，只是在处理较大地址时需要一些额外的开销。带启动位和停止位中断的模式的工作方式与其他模式相同，只是在检测到启动、重复启动或停止条件时，另外会将SSP1IF 置1。

25.5.1 从模式地址

SSP1ADD 寄存器（寄存器212H）包含从模式地址。在启动或重复启动条件之后接收到的第一个字节将与该寄存器中的存储值进行比较。如果字节匹配，则值会被装入SSP1BUF 寄存器，并产生中断。如果值不匹配，则模块会进入空闲状态，并且不会向软件指示是否发生了什么情况。SSP 掩码寄存器（寄存器213H）会影响地址匹配过程。更多信息，请参见第25.5.9 节“SSP1 掩码寄存器”。

25.5.1.1 I2C 从器件 7 位寻址模式

在 7 位寻址模式下，在确定地址是否匹配时，所接收数据字节的 LSb 会被忽略。

25.5.1.2 I2C 从器件 10 位寻址模式

在应答高字节之后，UA 位会置1，SCL 会保持低电平，直到用户使用低地址更新SSP1ADD 为止。在低地址字节送入之后，全部8 位将与SSP1ADD 中的低地址值进行比较。即使地址不匹配， SSP1IF 和UA 也会置1，SCL 会保持低电平，直到SSP1ADD 发生更新可再次接收高字节为止。当SSP1ADD发生更新时， UA 位会被清零。这可以确保模块准备好在下一次通信时接收高地址字节。

在所有10 位寻址通信开始时，都需要以写请求方式进行高地址和低地址匹配。在寻址到从器件后，通过发出重复启动条件并随着时钟移入R/W位置1的高地址字节来启动数据发送。然后，从器件硬件将会应答读请求，并准备好随着时钟移出数据。这只有在从器件接收到完全匹配的高地址和低地址字节之后才有效。

25.5.2 从接收

当接收到的匹配地址字节的R/W位清零时，SSP1STAT寄存器的R/W位会清零。接收到的地址被装入SSP1BUF 寄存器并产生应答。当接收到的地址存在上溢条件时，将会产生无应答信号。上溢条件定义为SSP1STAT 寄存器的BF 位被置1，或SSP1CON1寄存器的SSP1OV位被置1。SSP1CON3寄存器的BOEN 位会修改该操作。更多信息，请参见寄存器217H。每个传输的数据字节都会产生MSSP1 中断。标志位SSP1IF 必须用软件清零。

当 SSP1CON2 寄存器的 SEN 位被置 1 时， SCL 将在接收到每个字节后保持低电平（时钟延长） 。必须通过将 SSP1CON1 寄存器的 CKP 位置 1 来释放时钟， 10 位模式下的特殊情况除外。更多详细信息，请参见第 25.2.3 节“SPI 主模式”。

25.5.2.1 7 位寻址接收

本节介绍在7 位寻址模式下，配置为I^2C 从器件的MSSP1 模块的标准事件序列。还描述了由硬件或软件作出的所有决定及其对接收的影响。图25-14和图25-15用直观的方式对此作了说明。

以下列出了实现I^2C 通信时通常必须完成的步骤。

1. 检测到启动位。
2. SSP1STAT 的S位置1 ；如果允许在检测到启动条件时产生中断，则SSP1IF 会置1。
3. 接收到R/W位清零的匹配地址。
4. 从器件通过将SDA 下拉为低电平而向主器件发送ACK，并将SSP1IF 位置1。
5. 用软件清零SSP1IF 位。
6. 软件从SSP1BUF 中读取接收的地址，使BF 标志清零。
7. 如果SEN = 1，从器件软件会通过将CKP 位置1来释放SCL 线。
8. 主器件送出数据字节。
9. 从器件通过将SDA 驱动为低电平而向主器件发送ACK，并将SSP1IF 位置1。
10. 用软件清零SSP1IF。
11. 软件从SSP1BUF中读取接收的字节，使BF清零。
12. 对于从主器件接收到的所有字节重复步骤8-12。
13. 主器件发送停止条件，将 SSP1STAT的P位置1，总线变为空闲状态。

25.5.2.2 使用 AHEN 和 DHEN 时的 7 位接收

在AHEN 和DHEN 置1 时，从器件接收的工作方式与不使用这些选项时的工作方式相同，只是在SCL 的第8个下降沿之后添加了额外的中断和时钟延长。这些额外中断允许从器件软件决定是否应答（ACK）接收的地址或数据字节，而不是由硬件决定。以下列表介绍了要对I2C 通信使用这些选项时，从器件软件需要执行的步骤。图25-16 显示了同时使用地址和数据保持功能的模块。图25-17 包含了SSP1CON2 寄存器的SEN 位置1 时的操作。

1. SSP1STAT 的S位置1 ；如果允许在检测到启动条件时产生中断，则SSP1IF 会置1。
2. R/W位清零的匹配地址随时钟移入。在SCL 的第8 个下降沿之后， SSP1IF 置1， CKP 清零。
3. 从器件清零SSP1IF。
4. 从器件可以查看SSP1CON3 寄存器的ACKTIM位，以确定SSP1IF是在ACK之前还是之后置1。
5. 从器件从SSP1BUF 中读取地址值，使BF 标志清零。
6. 从器件通过设置ACKDT 来设置要送到主器件的ACK值。
7. 从器件通过将CKP 置1 来释放时钟。
8. SSP1IF 会在ACK之后置1，不会在NACK 之后置1。
9. 如果SEN = 1，从器件硬件会在ACK之后延长时钟。
10. 从器件清零SSP1IF。

注：	即使不进行时钟延长，且BF已清零，SSP1IF仍然会在SCL 的第9 个下降沿之后置1。只有向主器件发送了NACK 信号后，SSP1IF 才不会置1。

11. 在所接收数据字节的第8 个SCL 下降沿之后，SSP1IF 置1， CKP 清零。
12. 从器件通过查看SSP1CON3 的ACKTIM 位来确定中断源。
13. 从器件从SSP1BUF 中读取接收的数据，使BF清零。
14. 对于接收的每个数据字节，重复步骤7-14。
15. 从器件发送ACK= 1 或主器件发送停止条件可结束通信。如果发送了停止条件且禁止了停止条件检测中断，则从器件只能通过查询SSTSTAT 寄存器的P 位才能知道停止条件。

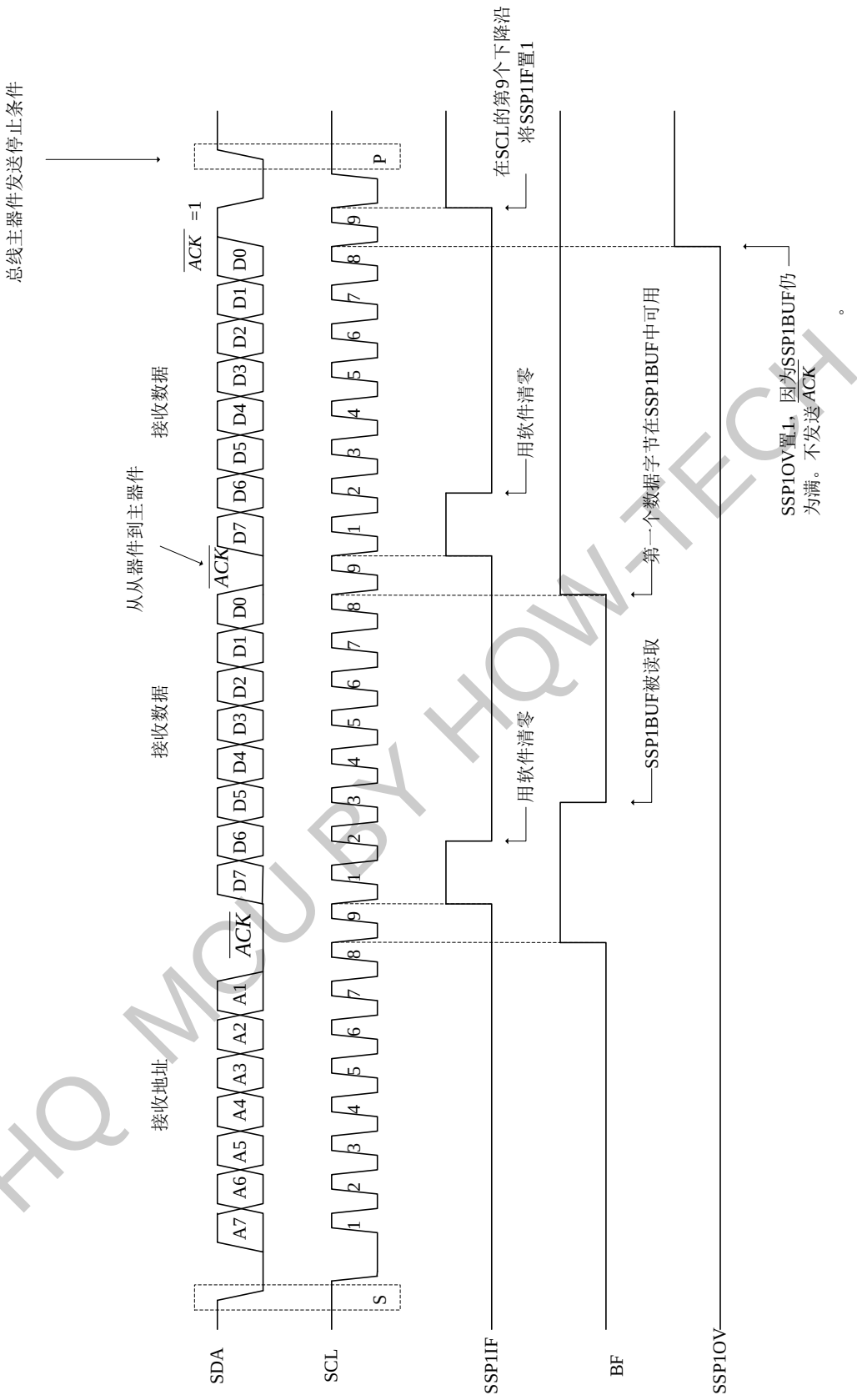


图25-14： I2C 从模式， 7 位地址，接收（SEN = 0， AHEN = 0， DHEN = 0）

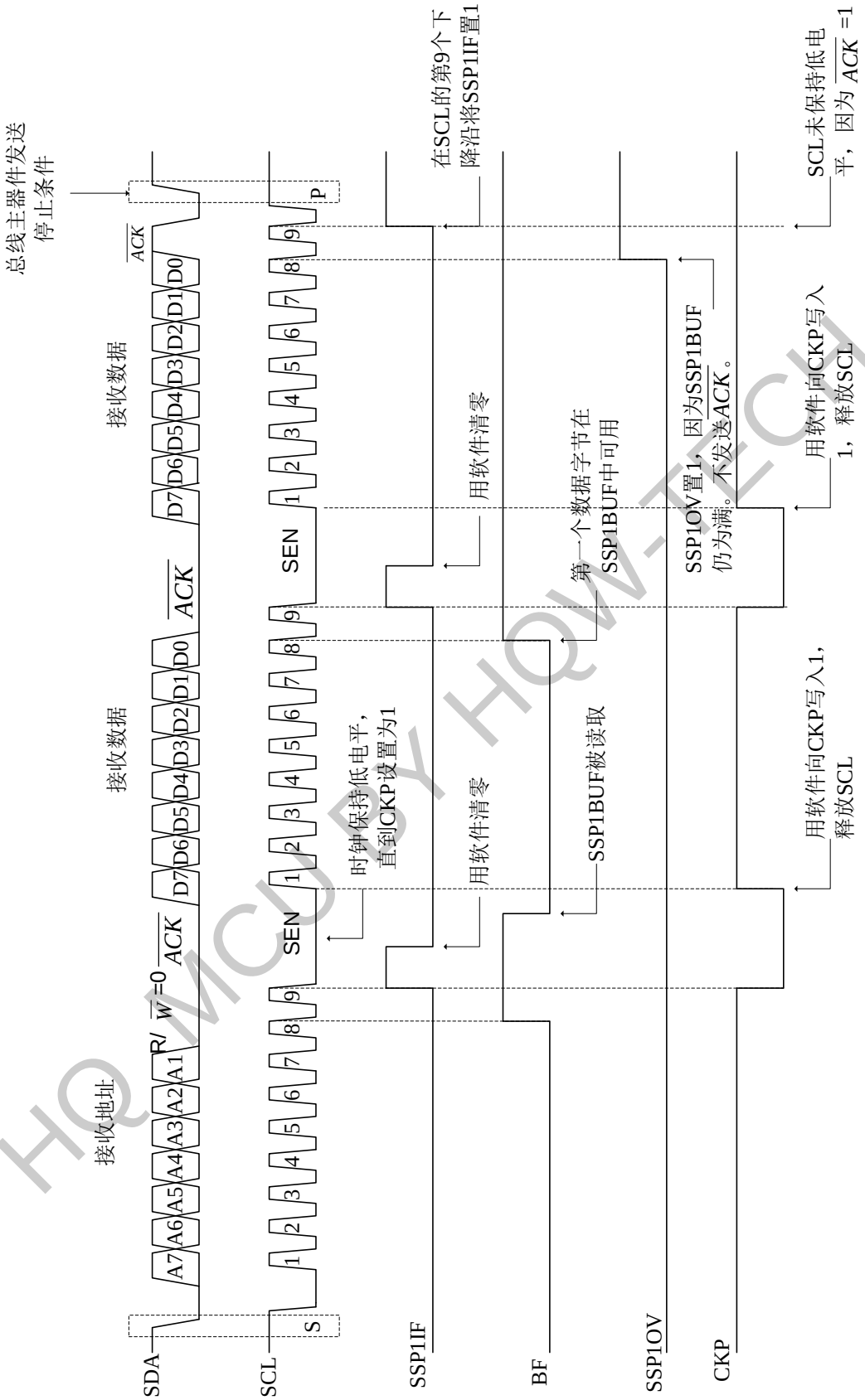


图25-15： I2C 从模式， 7 位地址，接收（SEN = 1， AHEN = 0， DHEN = 0）

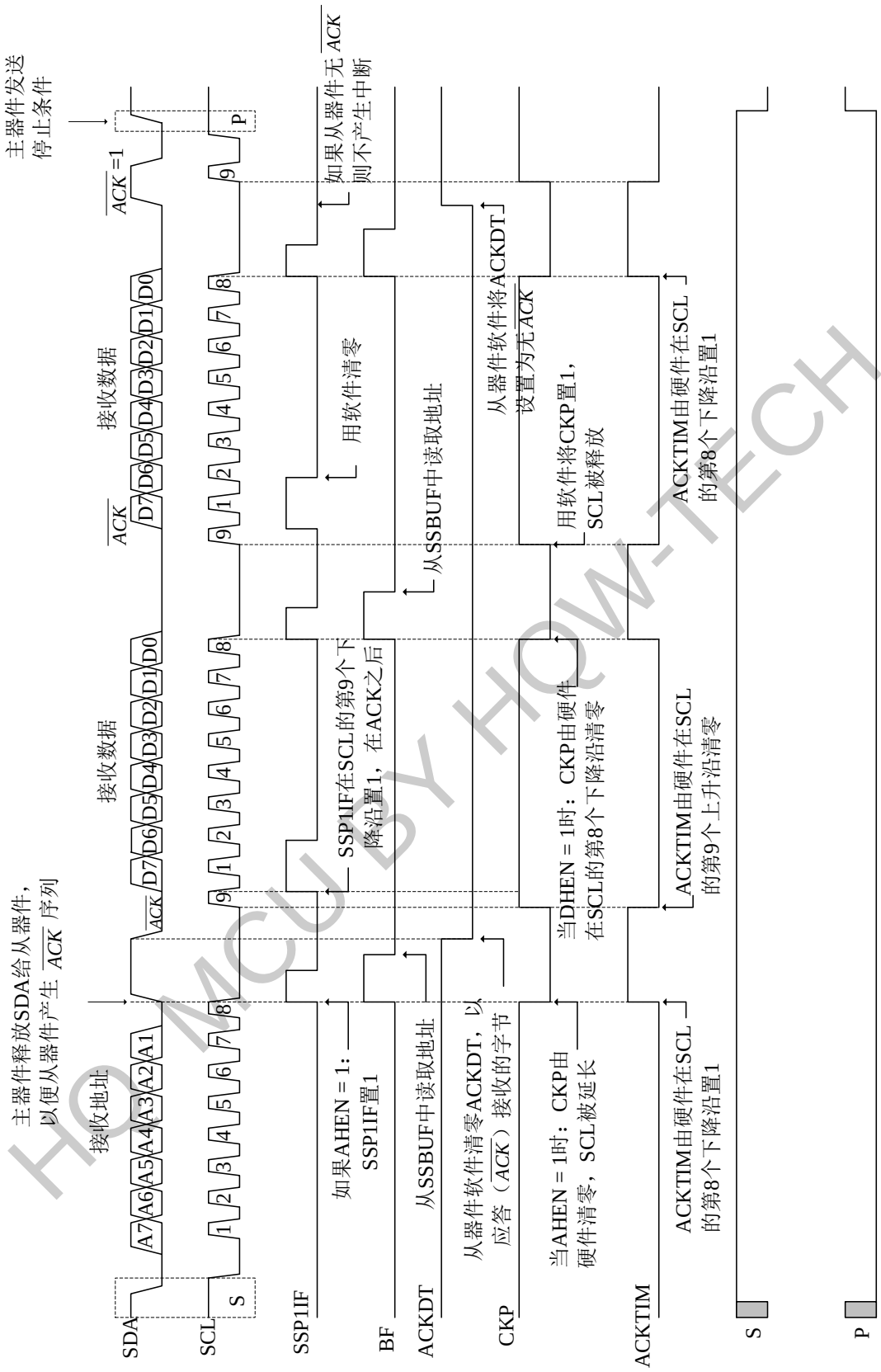


图25-16： I2C 从模式， 7 位地址，接收（SEN = 0， AHEN = 1， DHEN = 1）

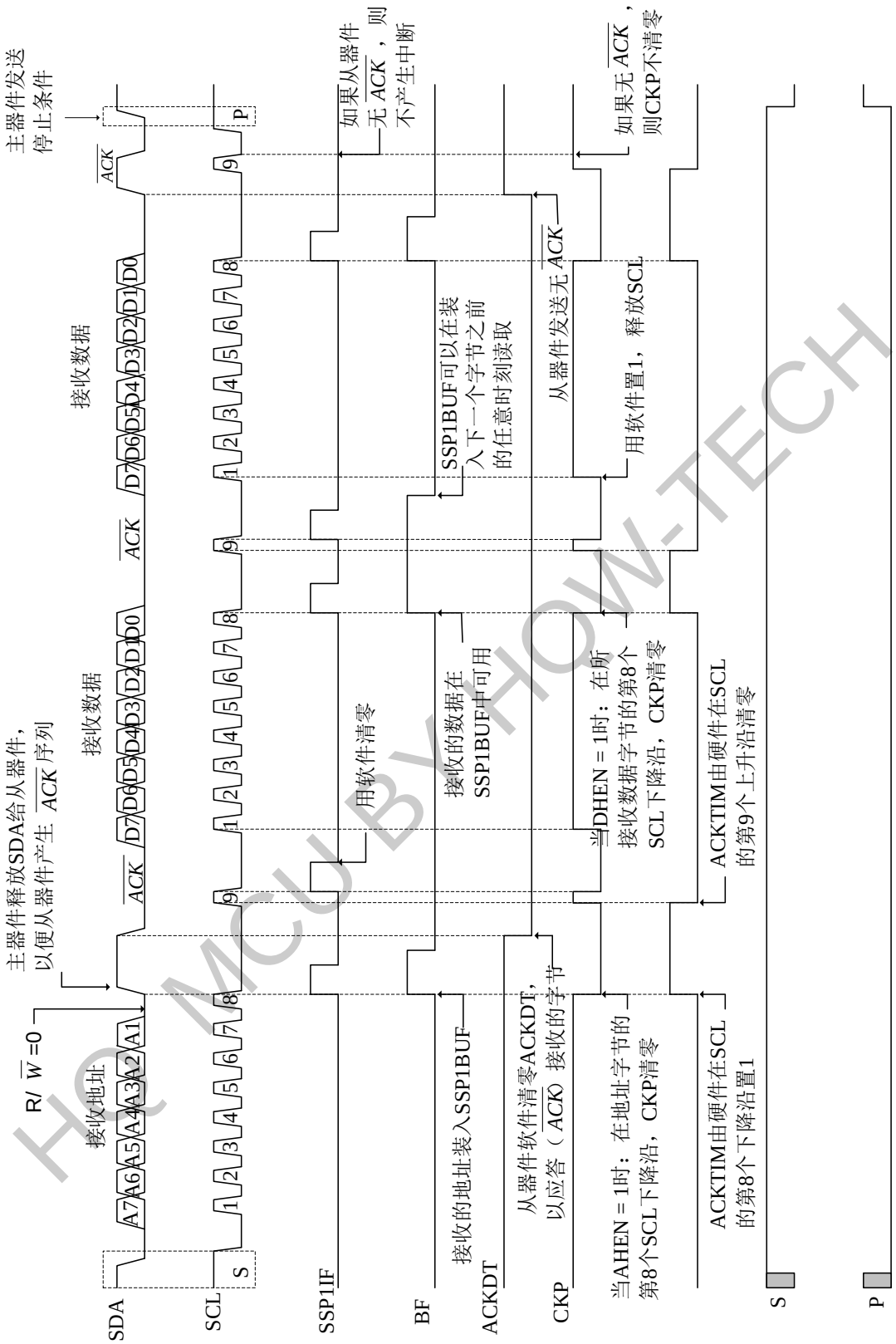


图 25-17： I2C 从模式， 7 位地址，接收（SEN = 1， AHEN = 1， DHEN = 1）

25.5.3 从发送

当输入地址字节的R/W位置1并发生地址匹配时，SSP1STAT寄存器的R/W位被置1。接收到的地址会被装入SSP1BUF寄存器，并且从器件会在第9个位发送ACK脉冲。

在ACK之后，从器件硬件会清零CKP 位，并且SCL 引脚保持低电平（更多详细信息，见第25.3.1 节“ 时钟延长”）。通过延长时钟，主器件只有在从器件准备好发送数据时，才发出另一个时钟脉冲。发送数据必须装入SSP1BUF 寄存器，此时该数据也会被装入SSP1SR 寄存器。然后，应通过将SSP1CON1寄存器的CKP 位置1 来释放SCL 引脚。8 个数据位在SCL 输入的下降沿被移出。这可确保在SCL 为高电平期间SDA 信号是有效的。

来自主接收器的ACK脉冲将在第9 个SCL 输入脉冲的上升沿锁存。该ACK值会被复制到SSP1CON2 寄存器的ACKSTAT 位中。如果ACKSTAT 置1（无ACK应答信号） ，那么表示数据传输已完成。这种情况下，在从器件锁存无ACK时，从器件会进入空闲状态，并等待出现另一个启动位。如果SDA 线为低电平（ACK），则必须将下一个要发送的数据装入SSP1BUF寄存器。同样，必须通过将CKP 位置1 来释放SCL 引脚。每个数据传输字节都会产生MSSP1 中断。SSP1IF 位必须用软件清零，SSP1STAT 寄存器用于确定字节的状态。SSP1IF 位在第9 个时钟脉冲的下降沿被置1。

25.5.3.1 从模式总线冲突

从器件接收到读请求，开始在SDA 线上移出数据。如果检测到总线冲突， SSP1CON3 寄存器的SBCDE位会置1，PIRx寄存器的BCL1IF位会置1。在检测到总线冲突时，从器件会变为空闲状态，等待再次被寻址。用户软件可以通过使用BCL1IF 位来处理从器件总线冲突。

25.5.3.2 7 位发送

主器件可以向从器件发送读请求，然后从从器件中移出数据。以下列表列出了在实现标准数据发送时，从器件软件需要执行的操作。图25-18可用作该列表的参考。

1. 主器件在SDA 和SCL 上发送一个启动条件。
2. SSP1STAT 的S位置1 ；如果允许在检测到启动条件时产生中断，则SSP1IF 会置1。
3. 从器件接收到R/W位置1 的匹配地址，并将SSP1IF 位置1。
4. 从器件硬件产生ACK，并将SSP1IF 置1。
5. 用户将SSP1IF 位清零。
6. 软件从SSP1BUF中读取接收的地址，使BF清零。
7. R/W置1，所以CKP 会在ACK之后自动清零。
8. 从器件软件将发送数据装入SSP1BUF。
9. CKP 位置1，释放SCL，使主器件可以从从器件中移出数据。
10. 来自主器件的ACK响应装入ACKSTAT 寄存器之后， SSP1IF 置1。
11. SSP1IF 位清零。
12. 从器件软件通过检查 ACKSTAT 位来确定主器件是否要移出更多数据

注	1：如果主器件应答（ACK），时钟将被延长。 2：ACKSTAT 是惟一一个在SCL 上升沿（第9 个）而不是下降沿发生更新的位。

13. 对于每个发送字节重复步骤9-13。
14. 如果主器件发送无ACK，则不会保持时钟，但SSP1IF 仍然会置1。
15. 主器件发送重复启动条件或停止条件。
16. 从器件不再被寻址。

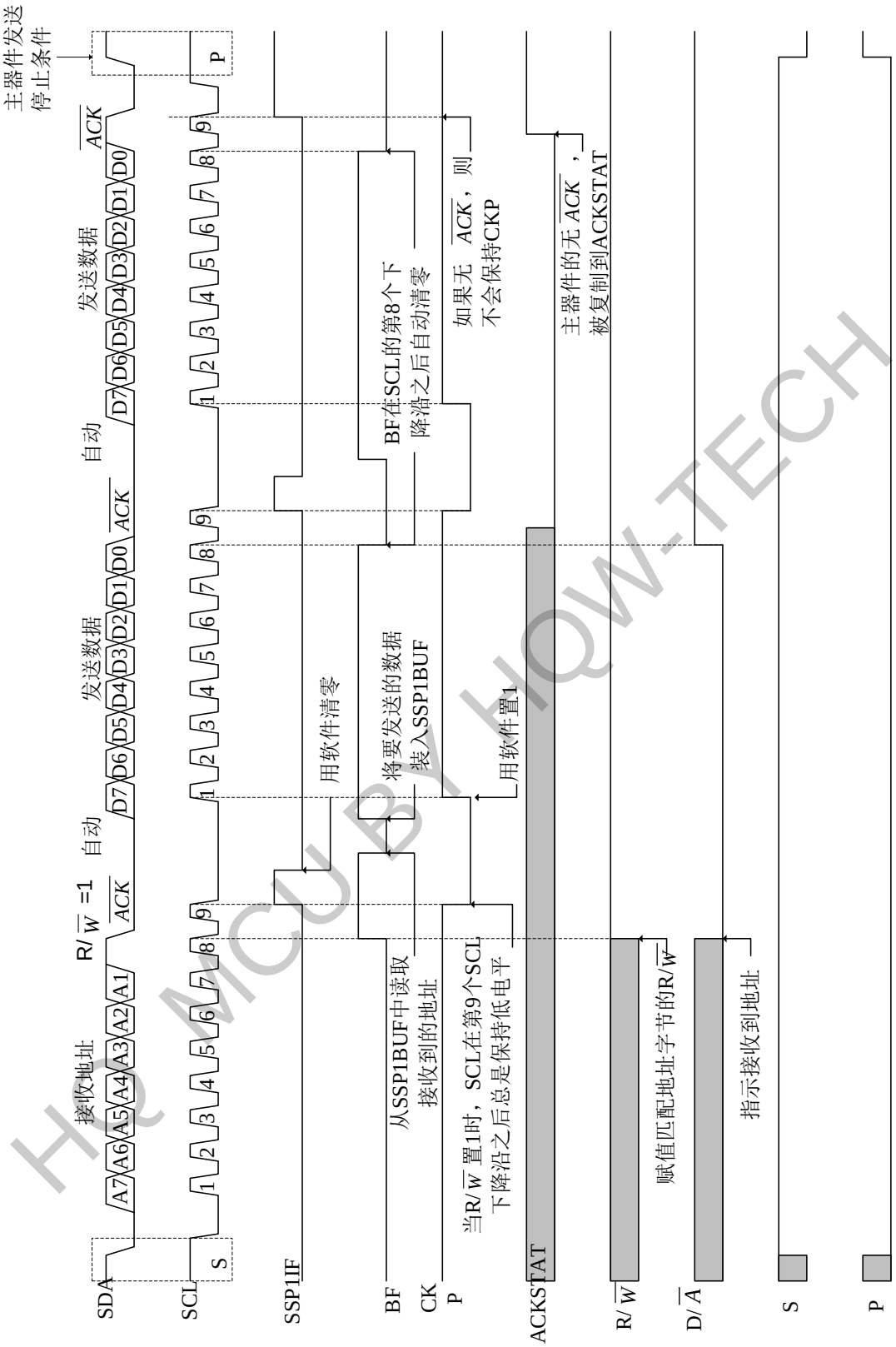


图 25-18： I2C 从模式， 7 位地址，发送（AHEN = 0）

25.5.3.3 使能地址保持时的 7 位发送

将SSP1CON3 寄存器的AHEN 位置1 时，器件会在所接收匹配地址的第8 个下降沿之后延长时钟和产生中断。在匹配地址送入之后，CKP 会清零，SSP1IF 中断标志会置1。图25-19 给出了在使能AHEN 时7 位地址从发送的标准波形图。

1. 总线启动时为空闲模式。
2. 主器件发送启动条件；SSP1STAT 的S位置1 ；如果允许在检测到启动条件时产生中断，则SSP1IF会置1。
3. 主器件发送R/W位置1 的匹配地址。在SCL 线的第8 个下降沿之后， CKP 位清零，并产生SSP1IF 中断。
4. 从器件软件清零SSP1IF。
5. 从器件软件读取SSP1CON3 寄存器的ACKTIM位，以及SSP1STAT寄存器的R/W和D/A 位，以确定中断源。
6. 从器件从SSP1BUF寄存器中读取地址值，使BF位清零。
7. 从器件软件根据该信息确定它是产生ACK还是产生无ACK，并相应地设置SSP1CON2寄存器的ACKDT位。
8. 从器件将CKP 位置1，释放SCL。
9. 主器件移入来自从器件的ACK值。
10. 如果R/W位置1，则在ACK之后，从器件硬件会自动将CKP 位清零，将SSP1IF 置1。
11. 从器件软件清零SSP1IF。
12. 从器件将要发送给主器件的值装入SSP1BUF，使BF 位置1。

注： 只有在ACK之后，才能装入SSP1BUF。

13. 从器件将CKP 位置1，释放时钟。
14. 主器件从从器件中移出数据，并在第9 个SCL 脉冲发送ACK值。
15. 从器件硬件将ACK值复制到SSP1CON2寄存器的ACKSTAT 位中。
16. 对于从从器件发送到主器件的每个字节重复步骤10-15。
17. 如果主器件发送无ACK，从器件会释放总线，让主器件可以发送停止条件和结束通信。

注： 主器件必须对于最后一个字节发送无ACK，以确保从器件释放SCL 线来接收停止条件。

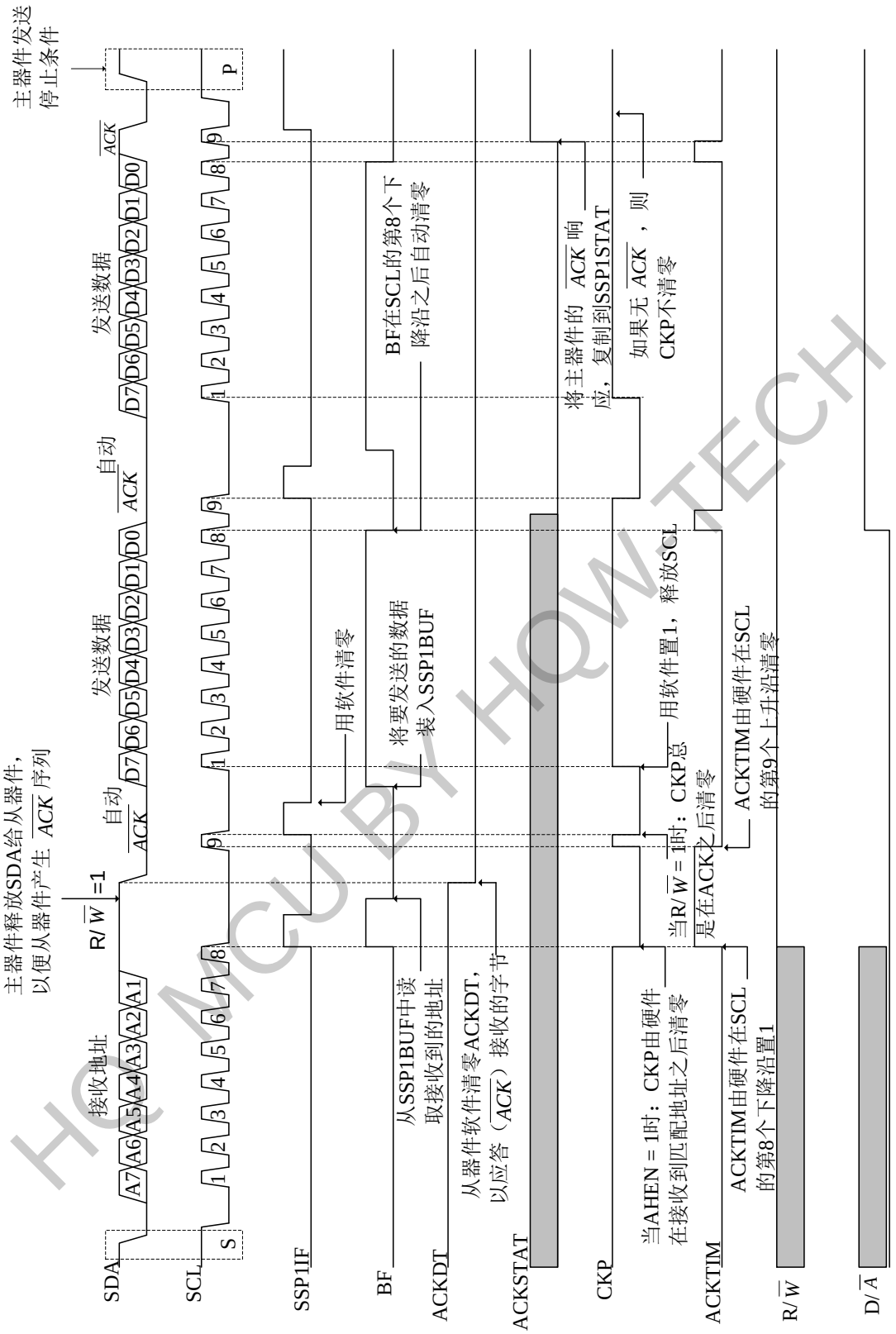


图 25-19： I2C 从模式， 7 位地址，发送（AHEN = 1）

25.5.4 从模式 10 地址接收

本节介绍在10 位寻址模式下，配置为I^2C 从器件的MSSP1 模块的标准事件序列。
图 25-20 用直观的方式对此作了说明。
以下列出了实现I^2C通信时从器件软件必须完成的步骤。

1. 总线启动时为空闲模式。
2. 主器件发送启动条件；SSP1STAT 的S位置1 ；如果允许在检测到启动条件时产生中断，则SSP1IF 会置1。
3. 主器件发送R/W位清零的匹配高地址；SSP1STAT寄存器的UA 位置1。
4. 从器件发送ACK， SSP1IF 置1。
5. 用软件清零SSP1IF 位。
6. 软件从SSP1BUF 中读取接收的地址，使BF 标志清零。
7. 从器件将低地址装入SSP1ADD，释放SCL。
8. 主器件向从器件发送匹配的低地址字节； UA 位置1。

注： 只有在ACK序列之后，才允许更新SSP1ADD寄存器。

9. 从器件发送ACK， SSP1IF 置1。

注： 如果低地址不匹配，SSP1IF和UA仍然会置1，从而让从器件软件可以将SSP1ADD 恢复为高地址。由于不匹配，BF 不会置1。CKP 不受影响。

10. 从器件清零SSP1IF。
11. 从器件从SSP1BUF 中读取接收的匹配地址，使BF 清零。
12. 从器件将高地址装入SSP1ADD。
13. 主器件随着时钟将数据字节移入从器件，并在第9 个SCL 脉冲随着时钟将ACK移出从器件；SSP1IF 置1。
14. 如果SSP1CON2 的SEN 位置1，CKP 会被硬件清零，时钟会被延长。
15. 从器件清零SSP1IF。
16. 从器件从SSP1BUF 中读取接收的字节，使BF清零。
17. 如果SEN 置1，从器件会将CKP 置1，以释放SCL。
18. 对于接收的每个字节重复步骤13-17。
19. 主器件发送停止条件来结束数据发送。

25.5.5 带地址或数据保持的 10 位寻址

在AHEN 或DHEN 置1 时，使用10 位寻址的接收方式与7 位模式相同。惟一的区别是需要使用UA 位来更新SSP1ADD 寄存器。所有功能（特别是在CKP 位清零，SCL 线保持低电平时）都是相同的。图25-21 可以用作AHEN 置1 时10 位寻址模式下从器件的参考图示。图25-22给出了10位寻址模式下从发送器的标准波形图。

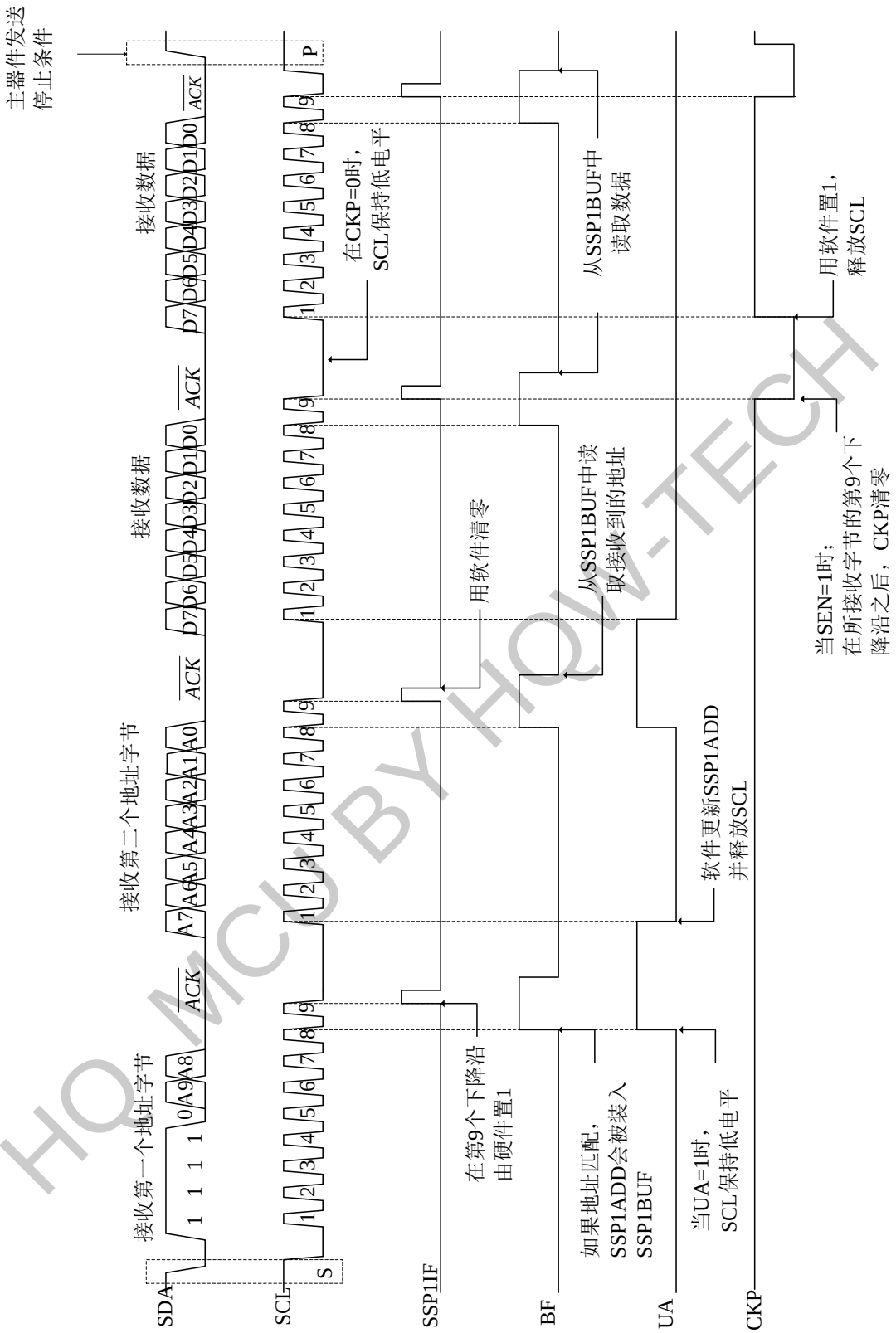


图 25-20： I2C 从模式， 10 位地址，接收（SEN = 1， AHEN = 0， DHEN = 0）

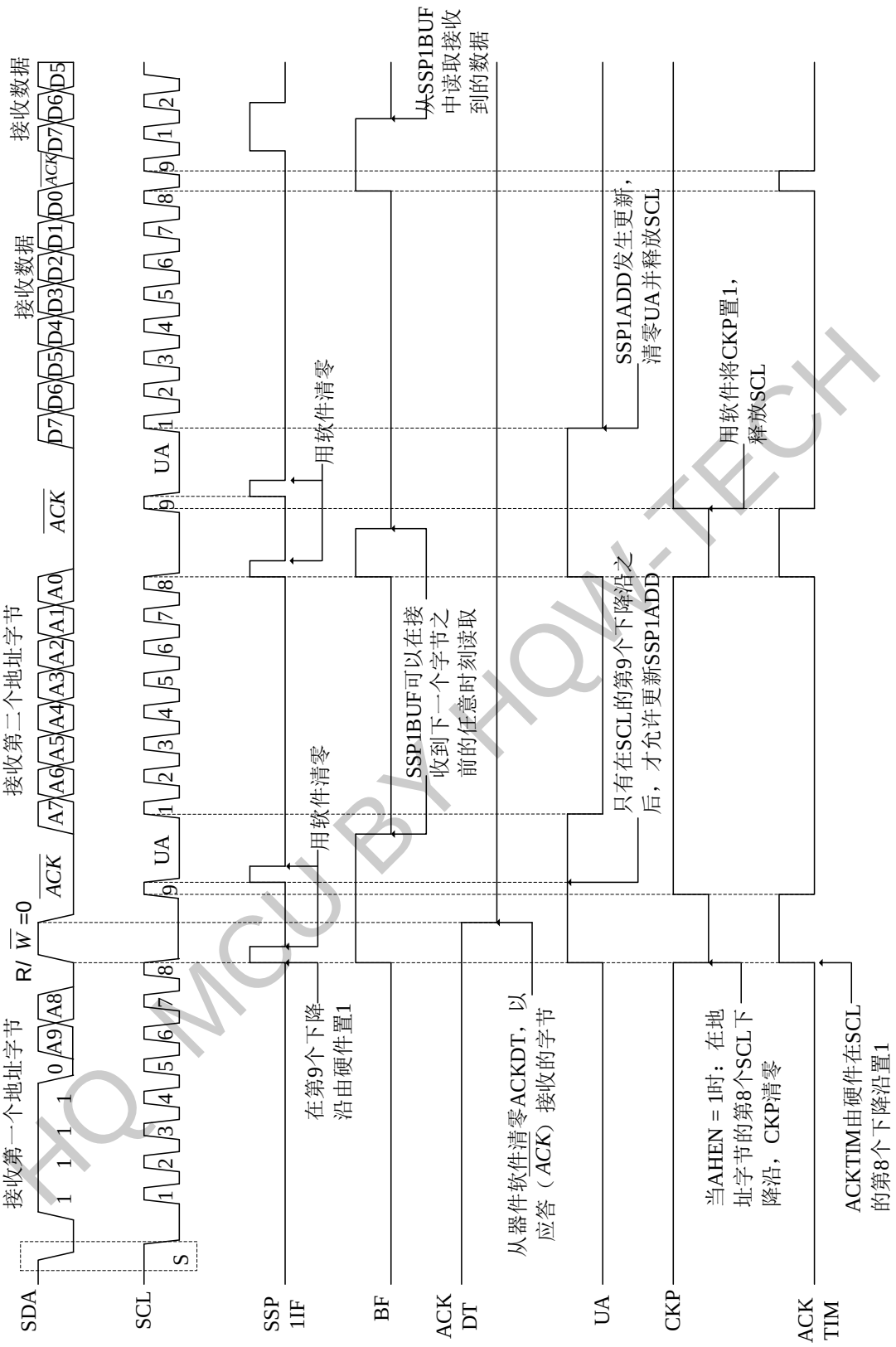


图 25-21： I2C 从模式， 10 位地址，接收（SEN = 0， AHEN = 1， DHEN = 0）

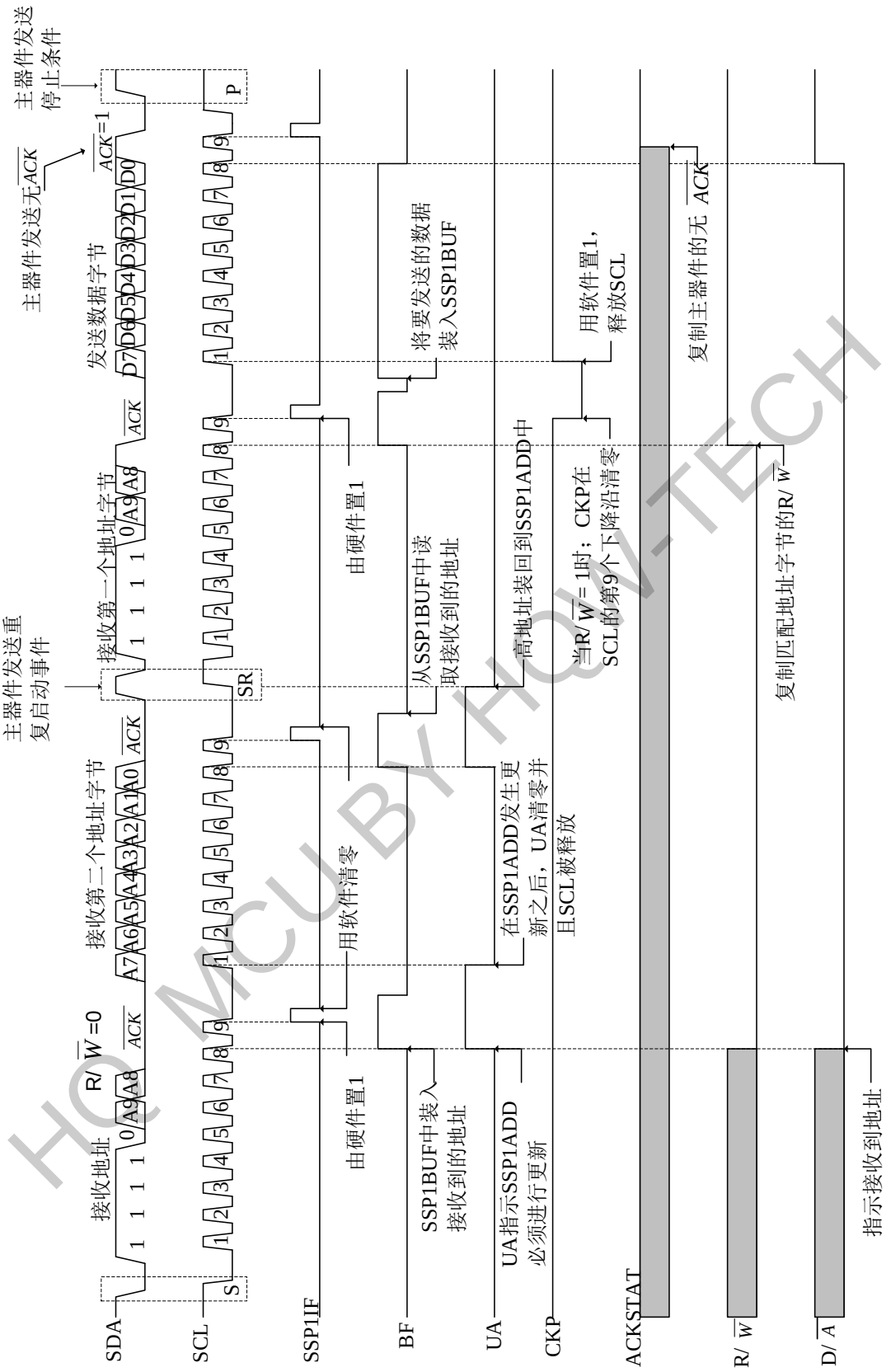


图 25-22： I2C 从模式， 10 位地址，发送（SEN = 0， AHEN = 0， DHEN = 0）

25.5.6 时钟延长

当总线上的某个器件将SCL线保持为低电平而有效暂停通信时，就发生了时钟延长现象。从器件可以延长时钟，以便可以有更多时间来处理数据或准备响应主器件。时钟延长时并不关心主器件的工作，因为任何时候只需总线上主器件处于活动状态，但是不传输数据就可以被认为是时钟延长。由从器件进行的任何时钟延长对于主器件软件都是不可见的，都由产生SCL的硬件进行处理。

SSP1CON1 寄存器的CKP 位用于在软件中控制时钟延长。每当CKP 位清零时，模块就会等待SCL 线变为低电平，然后保持低电平状态不变。将CKP 置1 将会释放SCL，允许继续进行通信。

25.5.6.1 正常时钟延长

如果SSP1STAT 的R/W位置1 （读请求），则在ACK之后，从器件硬件会清零CKP。这让从器件可以有时间使用要传输给主器件的数据更新SSP1BUF。如果SSP1CON2 的SEN 位置1，则在ACK序列之后，从器件将总是延长时钟。在从器件就绪之后，软件会将CKP置1，并继续进行通信。

注 1：BF 位对于是否延长时钟没有任何影响。这一点与模块的先前版本不同：如果在SCL的第9 个下降沿之前读取了SSP1BUF，先前版本将不会延长时钟，清零CKP。 2：如果在SCL 的第9 个下降沿之前装入SSP1BUF，则模块的先前版本不会为数据发送延长时钟。现在，对于读请求，总是会将该位清零。

25.5.6.2 10 位寻址模式

在10 位寻址模式下，当UA 位置1 时，时钟总是会被延长。这是无需清零CKP 就会延长SCL 的惟一情形。在写入SSP1ADD 之后， SCL 会立即被释放。

注： 如果第二个地址字节不匹配，先前版本的模块不会延长时钟。

25.5.6.3 字节无答应

当SSP1CON3 的AHEN 位置1 时，则在所接收匹配地址字节的第8 个SCL下降沿之后，硬件会将CKP清零。当SSP1CON3 的DHEN 位置1 时，则在所接收数据的第8 个SCL 下降沿之后， CKP 会被清零。通过在SCL 的第8 个下降沿之后延长时钟，从器件可以检查接收到的地址或数据，并确定是否要应答接收到的数据。

25.5.7 时钟同步和 CKP 位

每当 CKP 位清零时，模块就会等待 SCL 线变为低电平，然后保持低电平状态不变。但是，清零 CKP 位并不会将 SCL 输出置为低电平，只有在已经采样到 SCL 输出为低电平之后才会。因此，CKP 位不会将 SCL 线拉为低电平，除非外部 I^2C 主器件已将 SCL 线拉为低电平。SCL 输出将保持低电平，直到 CKP 位置 1 且 I^2C 总线上的所有其他器件已释放 SCL 为止。这可以确保对 CKP 位的写操作不会违反 SCL 的最短高电平时间要求（见图 25-23）。

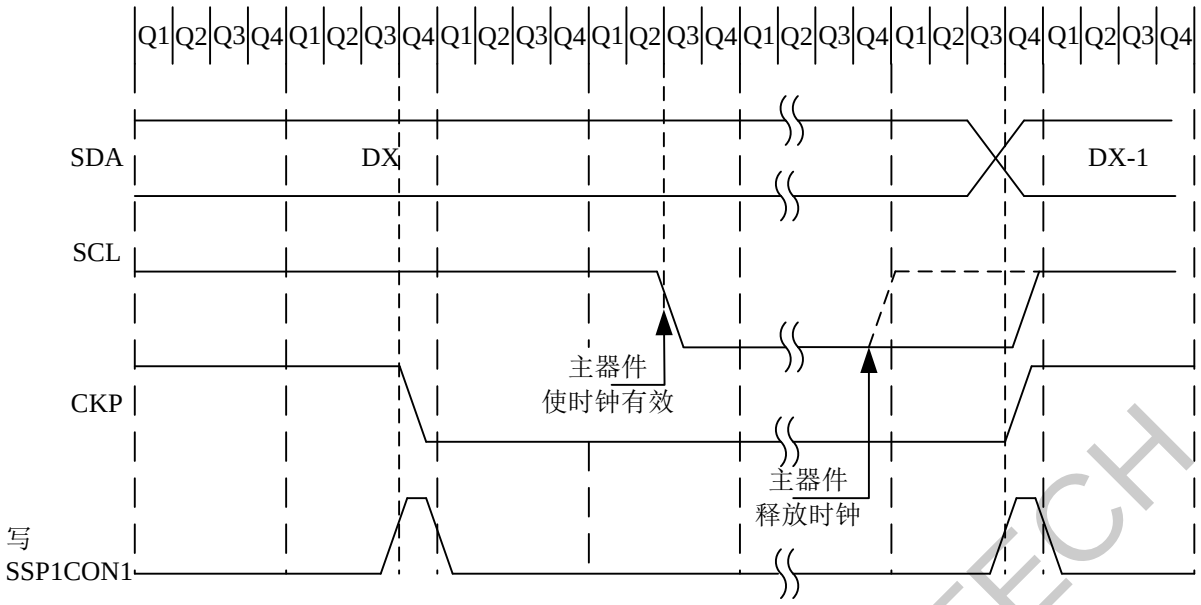


图25-23：时钟同步时序

25.5.8 广播呼叫地址支持

在I²C总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有器件都应该发送一个应答信号来响应。

广播呼叫地址是I²C协议中的保留地址，定义为地址0x00。如果SSP1CON2寄存器的GCEN位置1，则无论SSP1ADD中存储的值如何，在接收到该地址时，从模块都会自动发送ACK。在从器件移入R/W位清零的全零地址之后，将会产生中断，从器件软件可以读取SSP1BUF并进行响应。图25-24显示了广播呼叫接收序列。

在10 位地址模式下，UA 位不会在接收到广播呼叫地址时置1。从器件会准备接收作为数据的第二个字节，这与在7 位模式下相同。

如果SSP1CON3 寄存器的AHEN 位置1，则与接收到任意其他地址时相同，从器件硬件会在SCL 的第8个下降沿之后延长时钟。然后，从器件必须与正常情况下一样，设置它的ACKDT 值，并释放时钟来继续进行通信。

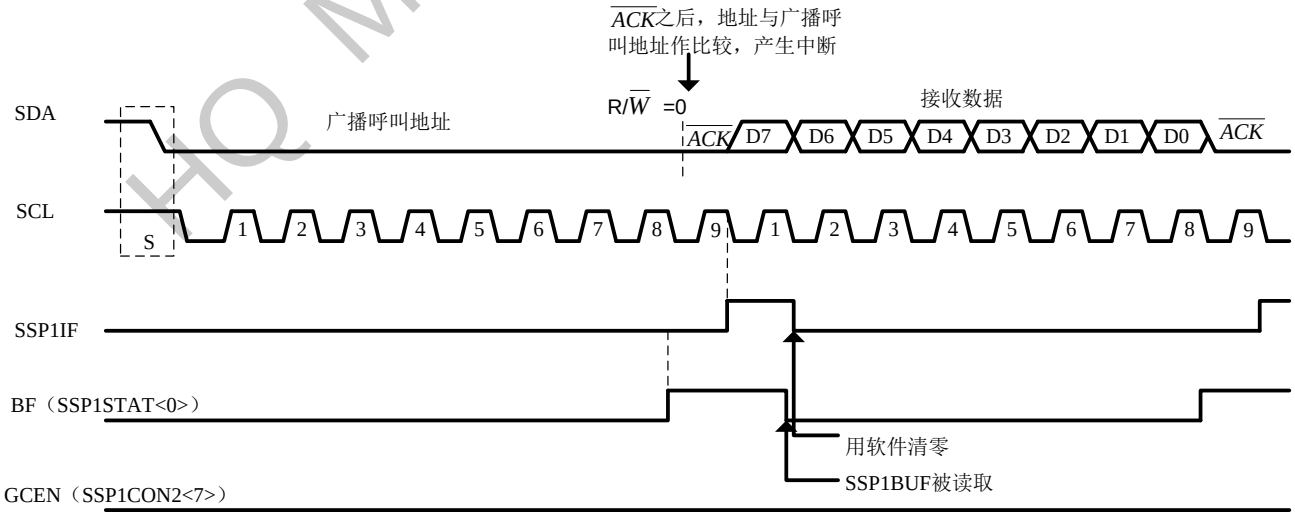


图25-24： 从模式广播呼叫地址序列

25.5.9 SSP1 掩码寄存器

SSP1 掩码（SSP1MSK）寄存器（寄存器213H）在I^2C 从模式下可用，用作地址比较操作期间SSP1SR 寄存器中保存的值的掩码。SSP1MSK 寄存器中的零（0）位可使接收地址中相应位变为“ 无关位”。发生任何复位条件时，该寄存器都会复位到全1 状态，因此，在写入掩码值之前对标准SSP1 操作没有影响。SSP1 掩码寄存器在以下期间保持有效：

• 7 位地址模式：A<7:1> 的地址比较。

• 10 位地址模式：仅针对A<7:0> 的地址比较。在接收地址的第一个（高）字节期间，SSP1 掩码没有影响。

25.6 I2C 主模式

通过将SSP1CON1 中的相应SSP1M 位置1 和清零，同时将SSP1EN 位置1，可以使能主模式。在主模式下，SCL和SDA线被设置为输入，由MSSP1硬件操作。通过在检测到启动和停止条件时产生中断来支持主操作模式。停止（P）位和启动（S）位在复位或禁止MSSP1模块时清零。当P 位置1 或总线空闲时，可取得I^2C 的控制权。

在固件控制的主模式下，用户代码根据启动位和停止位条件检测执行所有的I^2C 总线操作。在该模式下，启动和停止条件检测是惟一有效的电路。所有其他通信都通过用户软件直接操作SDA 和SCL 线来完成。以下事件会使SSP1 中断标志位SSP1IF 置1 （如果允许SSP1 中断，则产生中断）：

- 检测到启动条件
- 检测到停止条件
- 数据传输字节发送/ 接收
- 应答发送/ 接收
- 产生重复启动条件

注	1：当配置为I2C 主模式时，MSSP1 模块不允许事件排队。例如，不允许用户在发出启动条件，但启动条件尚未结束前，立即写SSP1BUF 寄存器。在这种情况下，将不会执行写SSP1BUF，WCOL 位将被置1，指示没有发生对SSP1BUF 的写操作。 2：处于主模式时，如果SEN/PEN 位清零，并且启动/ 停止条件完成，则会屏蔽启动/ 停止检测和产生中断。

25.6.1 I2C 主模式操作

主器件产生所有的串行时钟脉冲、启动条件和停止条件。以停止条件或重复启动条件结束传输过程。因为重复启动条件也是下一次串行传输的开始，因此I^2C 总线不会被释放。

在主发送器模式下，串行数据通过SDA 输出，而串行时钟由SCL输出。发送的第一个字节包括接收器件的从器件地址（7 位）和读/ 写（R/$\overline{W}$）位。在这种情况下，R/$\overline{W}$位将为逻辑0。一次发送8 位串行数据。每发送一个字节，都会接收到一个应答位。输出启动和停止条件指示串行传输的开始和结束。

在主接收模式下，发送的第一个字节包括发送器件的从器件地址（7 位）和R/$\overline{W}$位。在这种情况下， R/$\overline{W}$位将为逻辑1。因此，发送的第一个字节是一个7 位从器件地址，后跟1 指示接收位。串行数据通过SDA 接收，而串行时钟由SCL 输出。一次接收8 位串行数据。每接收到一个字节，都会发送一个应答位。启动和停止条件指示发送的开始和结束。波特率发生器用于设置从SCL输出的时钟频率。更多详细信息，请参见第25.7 节“ 波特率发生器”。

25.6.2 时钟仲裁

如果在任何接收、发送或重复启动/ 停止条件期间，主器件释放了SCL 引脚（允许SCL 悬空为高电平），就会发生时钟仲裁。当允许SCL 引脚悬空为高电平时，波特率发生器（Baud Rate Generator，BRG）暂停计数，直到SCL 引脚被实际采样到高电平为止。当SCL 引脚被采样到高电平时，波特率发生器重新装入SSP1ADD<7:0>的内容并开始计数。这可以确保在外部器件将时钟保持低电平时，SCL 在至少一个BRG 计满返回计数周期内总是保持高电平（图25-25）。

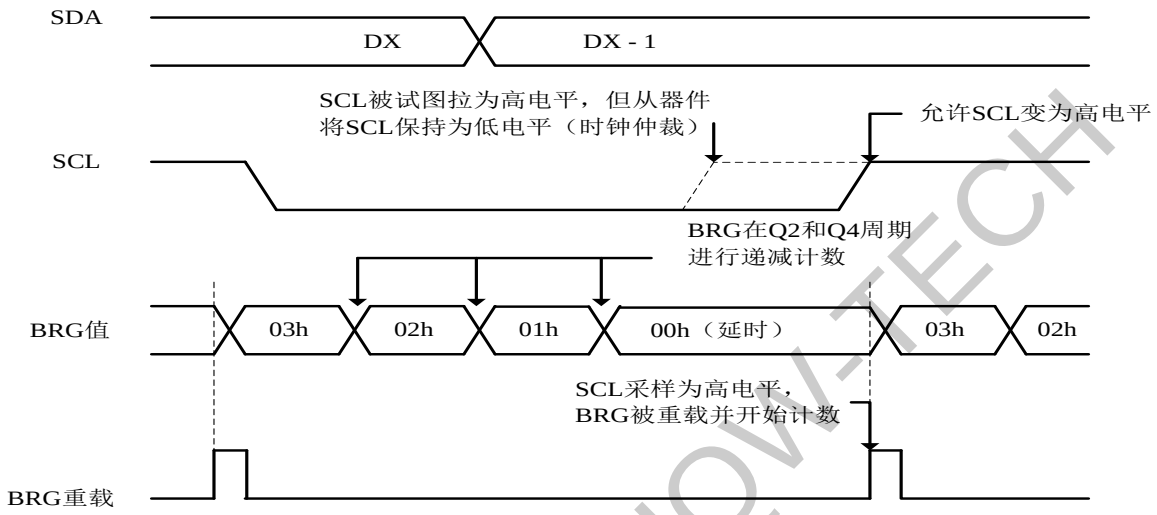


图 25-25： 带有时钟仲裁的波特率发生器时序

25.6.3 WCOL 状态标志

如果在启动、重复启动、停止、接收或发送序列过程中用户写 SSP1BUF，则 WCOL 被置 1，同时缓冲区内容不变（未发生写操作）。每当 WCOL 位置 1 时，它指示在模块不处于空闲状态时对 SSP1BUF 尝试了某个操作。

注： 由于不允许事件排队，在启动条件结束之前，不能写SSP1CON2 的低5 位。

25.6.4 I2C 主模式启动条件时序

要发出启动条件（图25-26），用户应将SSP1CON2寄存器的启动使能位SEN 置1。如果SDA 和SCL 引脚被采样为高电平，则波特率发生器会重新装入SSP1ADD<7:0>的内容并开始计数。如果波特率发生器超时（T_{BRG}）时，SCL 和SDA 都被采样为高电平，则SDA引脚被驱动为低电平。当SCL为高电平时，将SDA驱动为低电平将产生启动条件，并使SSP1STAT 寄存器的S 位置1。随后波特率发生器重新装入SSP1ADD<7:0>的内容并恢复计数。当波特率发生器再次超时（T_{BRG}）时，SSP1CON2 寄存器的SEN 位将自动被硬件清零；波特率发生器暂停工作， SDA 线保持低电平，启动条件结束。

注 1：如果在启动条件开始时，SDA 和SCL 引脚已经采样为低电平，或者在启动条件期间，SCL 线在SDA 线被驱动为低电平之前已经采样为低电平，则会发生总线冲突。总线冲突中断标志位BCL1IF置1，启动条件中止，I^2C 模块复位到空闲状态。 2：Philips I^2C™ 规范规定启动时不能发生总线冲突。

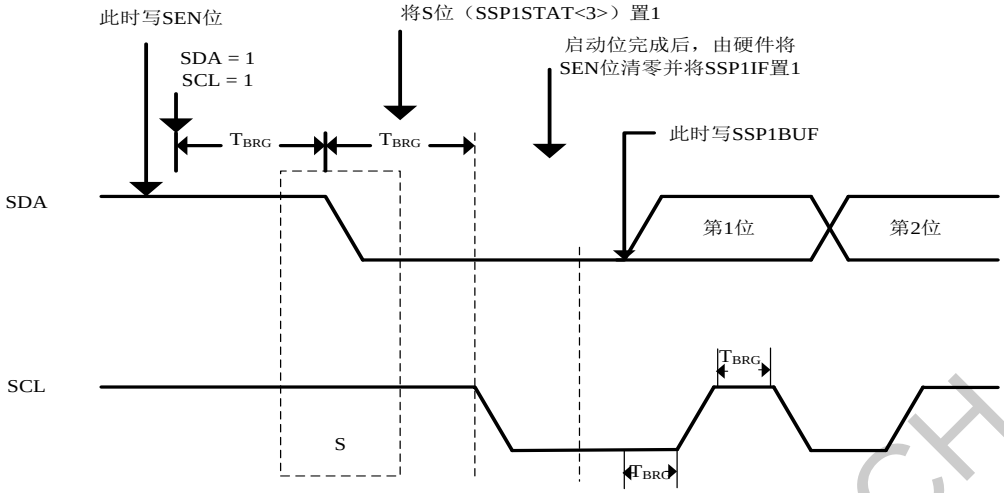


图25-26： 第一个启动位时序

25.6.5 I2C 主模式重复启动条件时序

当SSP1CON2 寄存器的RSEN 位设定为高电平，并且主器件状态机不再有效时，会产生重复启动条件（图25-27）。当RSEN 位置1 时，SCL 引脚被拉为低电平。当SCL引脚被采样为低电平时，波特率发生器会装入值并开始计数。在一个波特率发生器计数周期（T_{BRG}）内SDA 引脚被释放（拉为高电平）。当波特率发生器超时时，如果SDA 被采样为高电平， SCL 引脚将被置为无效（拉为高电平）。当SCL 被采样为高电平时，波特率发生器被重载并开始计数。SDA 和SCL 必须在一个T_{BRG} 内采样为高电平。接下来，在一个T_{BRG} 中，将SDA 引脚置为有效（SDA = 0），同时SCL 保持高电平。SCL 被置为低电平。随后SSP1CON2 寄存器的RSEN 位将自动清零，这次波特率发生器不会重载，SDA 引脚保持低电平。一旦在SDA 和SCL 引脚上检测到启动条件， SSP1STAT 寄存器的S 位就会被置1。SSP1IF 位在波特率发生器超时之前不会被置1。

注	1：	有任何其他事件在进行时，编程RSEN无效。
	2：	在重复启动条件期间，以下事件将会导致发生总线冲突：
		• 当SCL由低电平变为高电平时，SDA被采样为低电平。
		• 在SDA 被置为低电平之前，SCL 变为低电平。这指示另一个主器件正试图发送一个数据1。

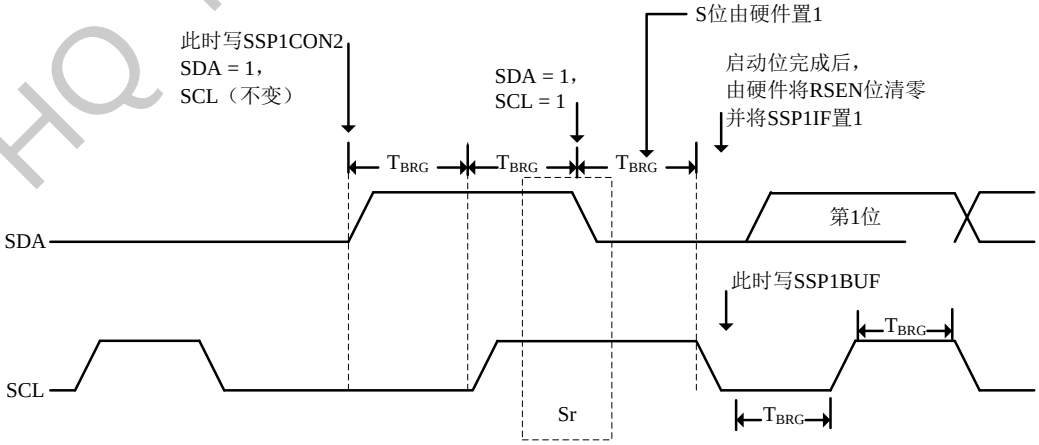


图25-27：重复启动条件波形图

25.6.6 I2C 主模式发送

发送一个数据字节、一个7 位地址或一个10 位地址的另一半都是通过简单地向SSP1BUF 寄存器写入一个值来实现的。该操作将使缓冲区满标志位BF 置1，并使波特率发生器开始计数和开始下一次发送。地址/ 数据的每一位将在SCL 的下降沿置为有效之后移出到SDA引脚。在一个波特率发生器计满返回计数周期（T_{BRG}）内， SCL 保持低电平。在SCL 被释放为高电平之前，数据应保持有效。当SCL 引脚释放为高电平时，它将在一个T_{BRG} 内保持高电平状态。在此期间以及SCL 的下一个下降沿之后的一段保持时间内，SDA 引脚上的数据必须保持稳定。在第8 位数据被移出（第8 个时钟的下降沿）之后， BF 标志被清零，同时主器件释放SDA。此时如果发生地址匹配或是数据被正确接收，被寻址的从器件将在第9 个位时间发出一个ACK位作为响应。ACK的状态在第9 个时钟的上升沿被写入ACKSTAT位。如果主器件接收到应答，应答状态位ACKSTAT 会被清零。如果未接收到应答，则该位被置1。在第9 个时钟之后，SSP1IF 位会置1，主时钟（波特率发生器）暂停，直到下一个数据字节装入SSP1BUF，SCL 保持低电平， SDA 保持不变（图25-28）。

在写SSP1BUF 之后，地址的每一位在SCL 的下降沿被移出，直到所有7 个地址位和R/W位都被移出。在第8个时钟的下降沿，主器件将释放SDA 引脚，以允许从器件发出一个应答响应。在第9 个时钟的下降沿，主器件通过采样SDA 引脚来判断地址是否被从器件识别。ACK位的状态被装入SSP1CON2 寄存器的ACKSTAT状态位。在发送地址的第9 个时钟下降沿之后，SSPIF置1，BF 标志清零，波特率发生器关闭直到发生下一次写SSP1BUF，且SCL 保持低电平，允许SDA 悬空。

25.6.6.1 BF 状态标志

在发送模式下， SSP1STAT 寄存器的BF 位在CPU 写SSP1BUF 时置1，在所有8 位数据移出后清零。

25.6.6.2 WCOL 状态标志

如果在发送过程中（即： SSP1SR 仍在移出数据字节时）用户写SSP1BUF，则WCOL 被置1，同时缓冲区内容不变（未发生写操作）。在下一次发送前WCOL 必须用软件清零。

25.6.6.3 ACKSTAT 状态标志

在发送模式下，当从器件发送应答（ACK= 0）时，SSP1CON2 寄存器的ACKSTAT 位被清零；当从器件没有应答（ACK= 1）时，该位被置1。从器件在识别出其地址（包括广播呼叫地址）或正确接收数据后，会发送一个应答。

25.6.6.4 典型的发送序列

1. 用户通过将SSP1CON2 寄存器的SEN 位置1，产生启动条件。
2. 在启动条件结束时，硬件将SSP1IF 置1。
3. SSP1IF 用软件清零。
4. 在进行任何其他操作前， MSSP1 模块将等待所需的启动时间。
5. 用户将从器件地址装入SSP1BUF 进行发送。
6. 器件地址从SDA引脚移出，直到发送完所有8 位地址数据。数据发送会在写入SSP1BUF 后立刻开始。
7. MSSP1 模块移入来自从器件的ACK位，并将它的值写入SSP1CON2 寄存器的ACKSTAT 位。
8. MSSP1模块在第9 个时钟周期结束时将SSP1IF位置1 产生中断。
9. 用户将8 位数据装入SSP1BUF。
10. 数据从 SDA 引脚移出，直到发送完所有 8 位数据。
11. MSSP1 模块移入来自从器件的ACK位，并将它的值写入SSP1CON2 寄存器的ACKSTAT 位。
12. 对于发送的所有数据字节重复步骤8-11。
13. 用户通过将SSP1CON2 寄存器的PEN 或RSEN位置1，产生停止或重复启动条件。停止/ 重复启动条件完成时产生中断。

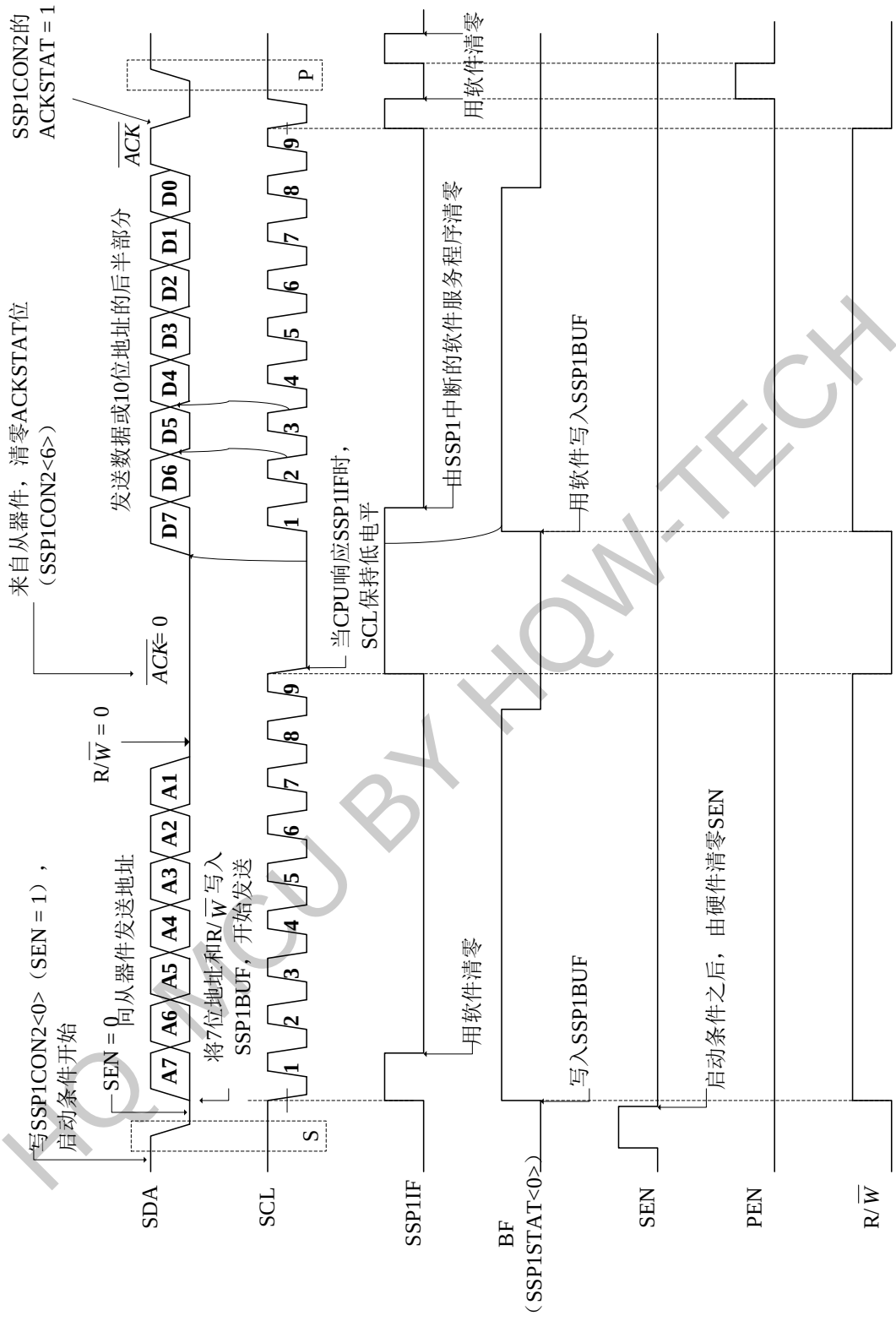


图25-28： I2C 主模式波形图（发送， 7 位或10 位地址）

25.6.7 I2C 主模式接收

通过编程SSP1CON2 寄存器的接收使能位RCEN 使能主模式接收（图25-29）。

注： 将RCEN 位置1 前， MSSP1 模块必须处于空闲状态，否则对RCEN位置1将无效。

波特率发生器开始计数，每次计满返回时，SCL 引脚的状态发生改变（由高变低/ 由低变高），数据被移入SSP1SR。在第8 个时钟的下降沿之后，接收使能标志自动清零， SSP1SR 的内容装入SSP1BUF， BF 标志位置1，SSP1IF 标志位置1，波特率发生器暂停计数，且SCL 保持为低电平。此时MSSP1 处于空闲状态，等待下一条命令。当CPU 读缓冲区时，BF 标志位会自动清零。通过将SSP1CON2 寄存器的应答序列使能位ACKEN 置1，用户可以在接收结束时发送应答位。

25.6.7.1 BF 状态标志

在接收操作中，将地址或数据字节从SSP1SR 装入SSP1BUF 时，BF 位被置1。在读SSP1BUF 寄存器时将其清零。

25.6.7.2 SSP1OV 状态标志

在接收操作中，当SSP1SR 接收到8 位数据且BF 标志位已经在上一次接收中置1 时，SSP1OV 位置1。

25.6.7.3 WCOL 状态标志

如果在接收过程中（即， SSP1SR 仍在移入数据字节时）用户写SSP1BUF，则WCOL 位被置1，同时缓冲区内容不变（未发生写操作）。

25.6.7.4 典型的接收序列

1. 用户通过将SSP1CON2 寄存器的SEN 位置1，产生启动条件。
2. 在启动条件结束时，硬件将SSP1IF 置1。
3. SSP1IF 用软件清零。
4. 用户将要发送的从器件地址写入SSP1BUF 且R/W位置1。
5. 器件地址从SDA引脚移出，直到发送完所有8 位地址数据。数据发送会在写入SSP1BUF 后立刻开始。
6. MSSP1 模块移入来自从器件的ACK位，并将它的值写入SSP1CON2 寄存器的ACKSTAT 位。
7. MSSP1模块在第9 个时钟周期结束时将SSP1IF位置1 产生中断。
8. 用户将SSP1CON2 寄存器的RCEN 位置1， 主器件从从器件移入一个字节。
9. 在SCL的第8个下降沿之后，SSP1IF和BF置1。
10. 主器件清零SSP1IF，并从SSP1UF 中读取接收到的字节，使BF 清零。
11. 主器件在SSP1CON2 寄存器的ACKDT 位中设置要发送给从器件的ACK值，并通过将ACKEN位置1 来发出ACK。
12. 主器件向从器件送出ACK，并且SSP1IF 置1。
13. 用户清零SSP1IF。
14. 对于从从器件接收到的每个字节重复步骤8-13。
15. 主器件通过发送无ACK或停止条件来结束通信。

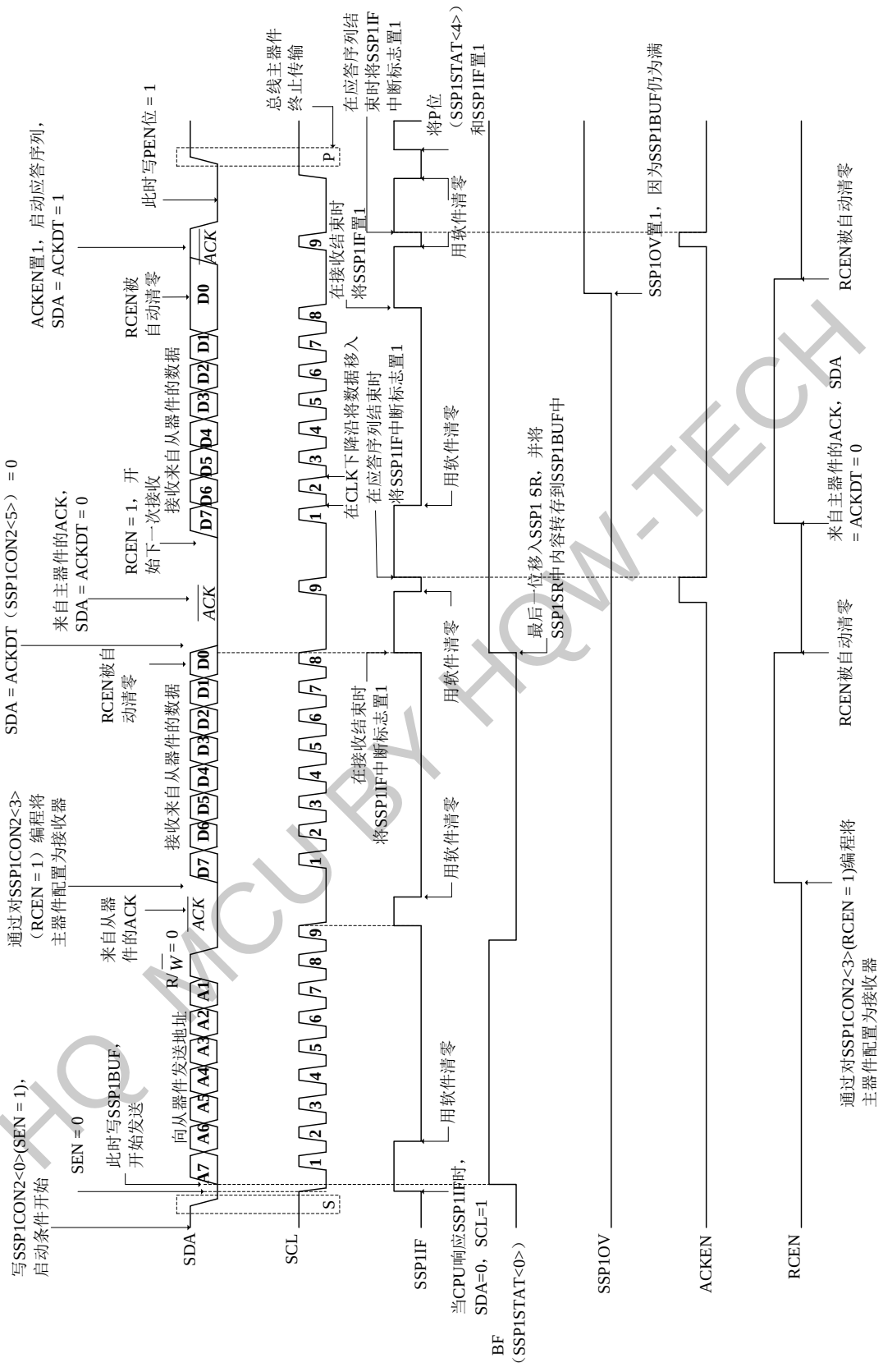


图25-29： I2C 主模式波形图（接收， 7 位地址）

25.6.8 应答序列时序

将SSP1CON2 寄存器的应答序列使能位ACKEN置1 即可使能应答序列。当该位被置1 时，SCL 引脚被拉为低电平，应答数据位的内容输出到SDA 引脚上。如果用户希望产生一个应答，则应将ACKDT 位清零。否则，用户应在应答序列开始前将ACKDT 位置1。然后波特率发生器进行一个计满返回周期（T_{BRG}）的计数，随后SCL引脚电平被置为无效（拉为高电平）。当SCL 引脚被采样为高电平（时钟仲裁）时，波特率发生器再进行一个T_{BRG} 周期的计数。然后SCL 引脚被拉为低电平。在这之后，ACKEN 位自动清零，波特率发生器关闭，MSSP1模块进入空闲模式（图25-30）。

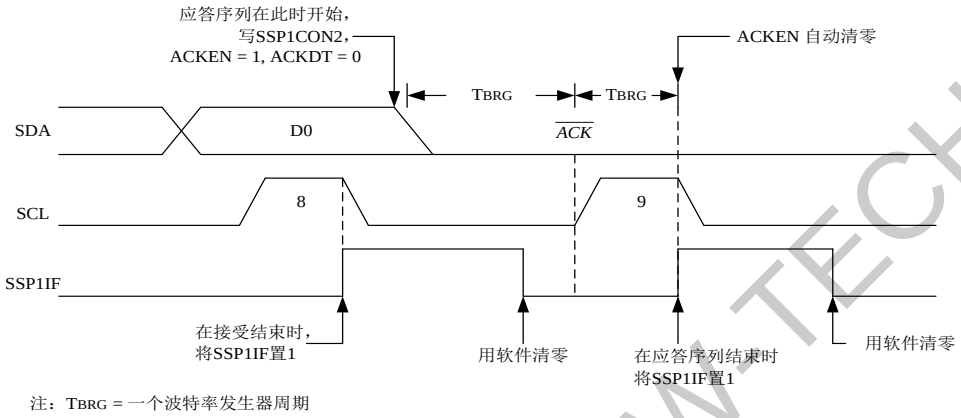


图25-30： 应答序列波形图

25.6.8.1 WCOL 状态标志

如果在应答序列进行过程中用户写SSP1BUF，则WCOL 被置1，同时缓冲区内容不变（未发生写操作）。

25.6.9 停止条件时序

如果将SSP1CON2 寄存器的停止序列使能位PEN置1，则在接收/ 发送结束后，SDA 引脚上将产生停止位。在接收/ 发送结束时，SCL 线在第9 个时钟的下降沿后保持低电平。当PEN 位置1 时，主器件将SDA 线置为低电平。当SDA 线被采样为低电平时，波特率发生器被重载并递减计数至0。当波特率发生器超时时，SCL 引脚被拉为高电平，在一个T_{BRG} （波特率发生器计满返回周期）之后， SDA 引脚将被置为无效。当SDA 引脚被采样为高电平且SCL 也是高电平时， SSP1STAT 寄存器的P 位被置1。另一个TBRG 之后，PEN 位被清零，同时SSP1IF 位被置1 （图25-31）。

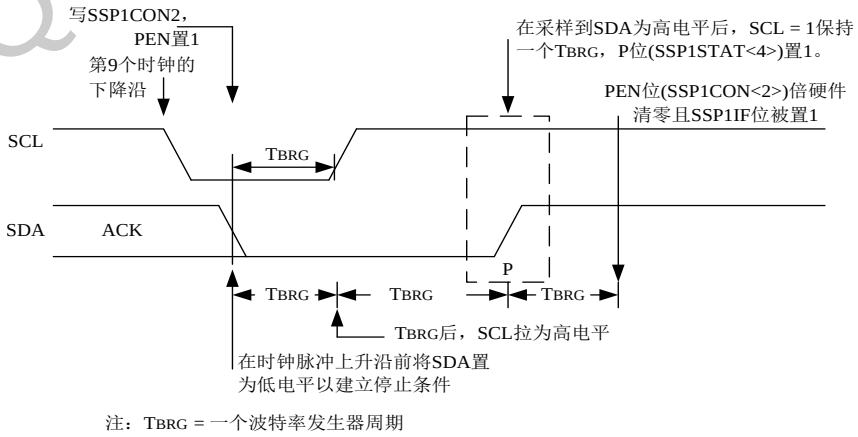


图25-31： 停止条件接收或发送模式

25.6.9.1 WCOL 状态标志

如果在停止序列进行过程中用户写SSP1BUF，则WCOL 位被置1，同时缓冲区内容不变（未发生写操作）。

25.6.10 休眠模式下的操作

在休眠模式下， I2C 从模块能够接收地址或数据，并且在地址匹配或字节传输完成时，如果允许MSSP1中断，会将处理器从休眠状态唤醒。

25.6.11 复位的影响

复位会禁止MSSP1 模块并终止当前的数据传输。

25.6.12 多主器件模式

在多主器件模式下，在检测到启动和停止条件时将产生中断，这可用于判断总线是否空闲。停止（P）位和启动（S）位在复位或禁止MSSP1 模块时清零。当SSP1STAT 寄存器的P 位置1 时，可以取得I2C 总线的控制权；或者，总线处于空闲状态，S位和P位都清零。当总线忙且允许SSP中断时，一旦发生停止条件便产生SSP 中断。在多主器件操作中，必须监视SDA 线来进行仲裁，以查看信号电平是否为期望的输出电平。此操作由硬件实现，其结果保存在BCL1IF 位中。

可能导致仲裁失败的情况是：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

25.6.13 多主器件通信、总线冲突和总线仲裁

多主器件模式是通过总线仲裁来支持的。当主器件将地址/ 数据位输出到SDA 引脚时，如果一个主器件在SDA引脚上输出1（将SDA 引脚悬空为高电平），而另一个主器件输出0，就会发生总线仲裁。当SCL 引脚悬空为高电平时，数据应是稳定的。如果SDA 引脚上期望的数据是1，而实际采样到的数据是0，则发生了总线冲突。主器件会将总线冲突中断标志BCL1IF 置1，并将I^2C 端口复位到空闲状态（图25-32）。

如果在发送过程中发生总线冲突，则发送操作停止，BF标志被清零， SDA 和SCL 线被置为无效，并且可写入SSP1BUF。当执行总线冲突中断服务程序时，如果I^2C总线空闲，用户可通过发出启动条件恢复通信。

如果在启动、重复启动、停止或应答条件过程中发生总线冲突，则条件被中止， SDA 和SCL 线被置为无效，SSP1CON2 寄存器中的相应控制位清零。当执行总线冲突中断服务程序时，如果I^2C 总线空闲，用户可通过发出启动条件恢复通信。主器件将继续监视SDA 和SCL 引脚。一旦出现停止条件， SSP1IF 位将被置1。

发生总线冲突时无论发送的进度如何，写入SSP1BUF都会从第一个数据位开始发送数据。在多主器件模式下，通过在检测到启动条件和停止条件时产生中断可以确定总线何时空闲。当SSP1STAT寄存器中的P 位置1时，可以获取I^2C 总线的控制权；或者，总线处于空闲状态， S 位和P 位都清零。

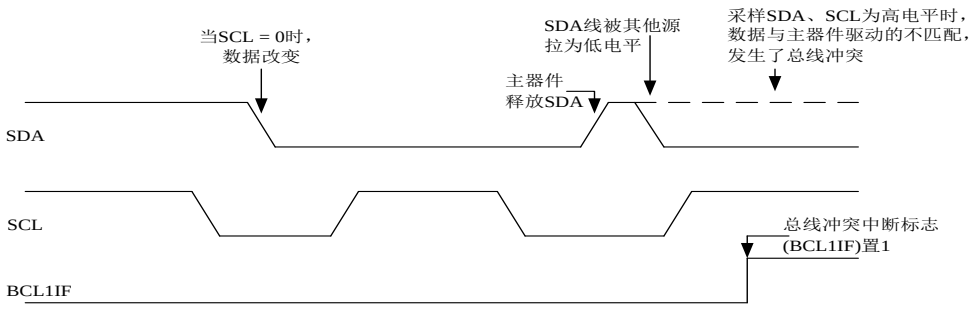


图25-32：发送和应答时的总线冲突时序

25.6.13.1 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

a) 在启动条件开始时，SDA 或SCL 被采样为低电平（图25-33）。

b) SDA 被置为低电平之前， SCL 采样为低电平（图25-34）。

在启动条件期间， SDA 和SCL 引脚都会被监视。

如果SDA 引脚或SCL 引脚已经是低电平，则发生以下所有事件：

- 中止启动条件，
- BCL1IF 标志置1，并且
- MSSP1 模块复位为空闲状态（图25-33）。

启动条件从SDA 和SCL 引脚被置为无效开始。当SDA引脚采样为高电平时，波特率发生器装入值并递减计数。如果在SDA 为高电平时，SCL 引脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图驱动一个数据1。

如果SDA 引脚在该计数周期内采样为低电平，则BRG复位，且SDA 线提前置为高电平（图25-35）。但是，如果SDA 引脚采样为1，则在BRG 计数结束时该引脚将被置为低电平。接着，波特率发生器被重载并递减计数至0 ；在此期间，如果SCL 引脚采样到0，则不会发生总线冲突。在BRG 计数结束时， SCL 引脚被置为低电平。

注：	在启动条件期间不会发生总线冲突，因为两个总线主器件不可能精确地在同一时刻发出启动条件。因此一个主器件将总是先于另一个主器件将SDA 置为有效。但是上述情况不会引起总线冲突，因为两个主器件一定会对启动条件后的第一个地址进行仲裁。如果地址是相同的，必须继续对数据部分、重复启动条件或停止条件进行仲裁。

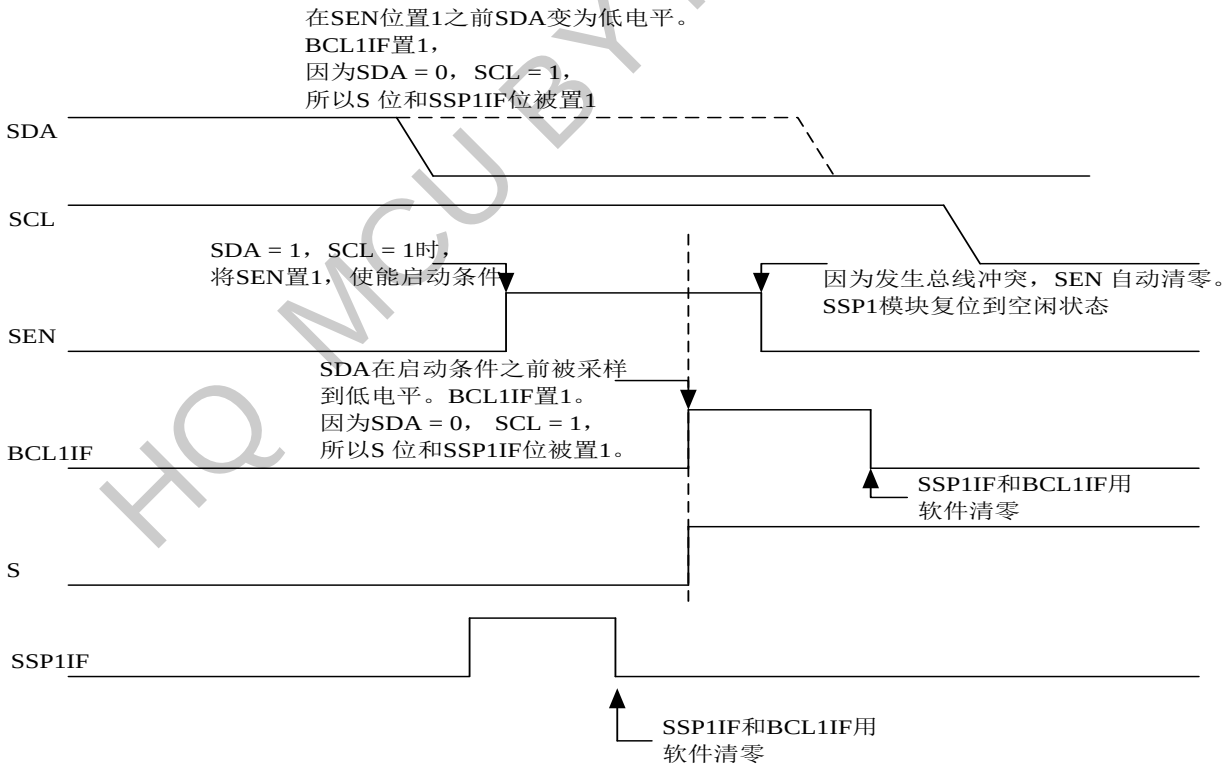


图25-33：启动条件期间的总线冲突（仅用于SDA）

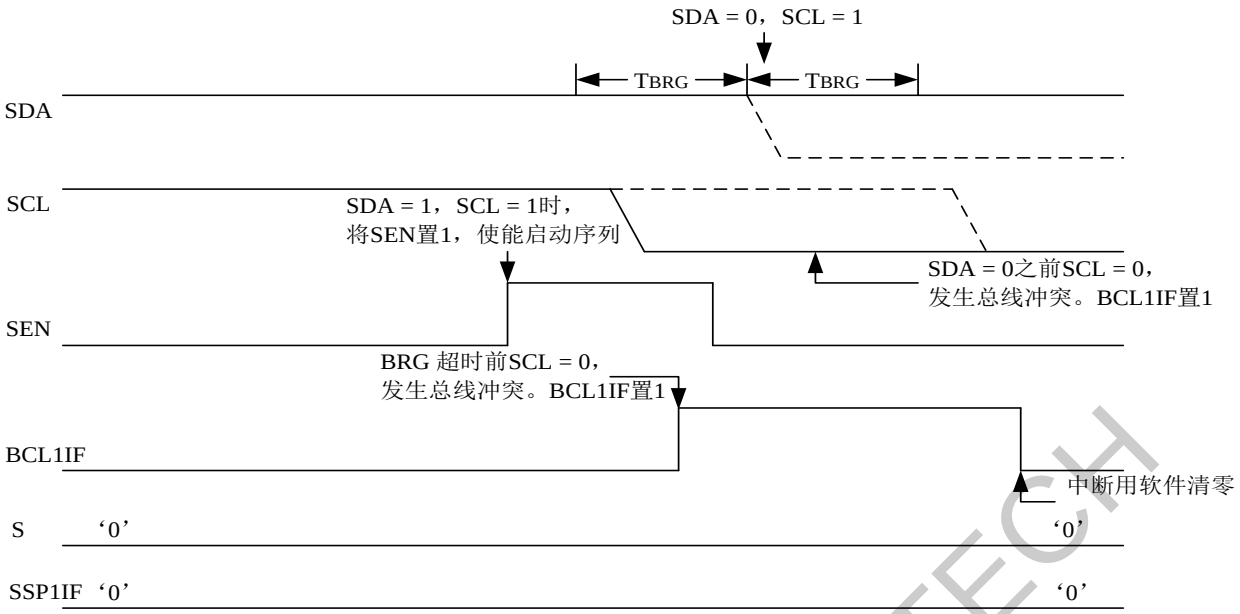


图25-34：启动条件期间的总线冲突（SCL = 0）

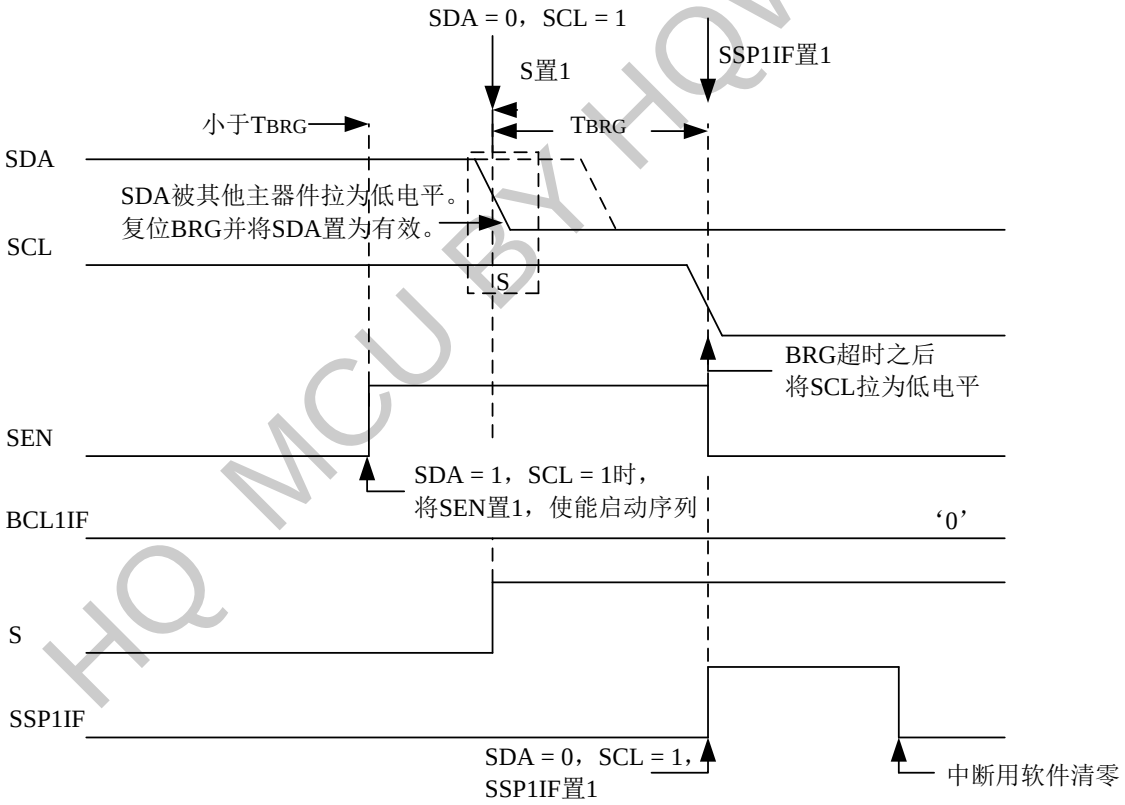


图25-35：启动条件期间由SDA 仲裁引起的BRG 复位

25.6.13.2 重复启动条件期间的总线冲突

在重复启动条件期间，如果发生以下情况，则会发生总线冲突：

a) 在SCL 由低电平变为高电平期间，在SDA 上采样到低电平（情形1）。

b) 在SDA 被置为低电平之前，SCL 变为低电平，表示另一个主器件正试图发送一个数据1（情形2）。

当用户释放SDA 并允许该引脚悬空为高电平时， BRG装入SSP1ADD 的值并递减计数至0。接着SCL 引脚被置为无效，当SCL 引脚采样到高电平时，对SDA 引脚进行采样。如果SDA 为低电平，则已发生了总线冲突（即，另一个主器件正试图发送一个数据0，见图25-36）。如果SDA 被采样到高电平，则BRG 被重载并开始计数。如果SDA 在BRG 超时之前从高电平变为低电平，则不会发生总线冲突，因为两个主器件不可能精确地在同一时刻将SDA 置为有效。如果SCL 在BRG 超时之前从高电平变为低电平，且SDA 尚未被置为有效，那么将发生总线冲突。在此情况下，另一个主器件在重复启动条件期间正试图发送一个数据1 （见图25-37）。

如果在BRG 超时结束时SCL 和SDA 都仍然是高电平，则SDA引脚被驱动为低电平，BRG被重载并开始计数。在计数结束时，不管SCL 引脚的状态如何， SCL 引脚都被驱动为低电平，重复启动条件结束。

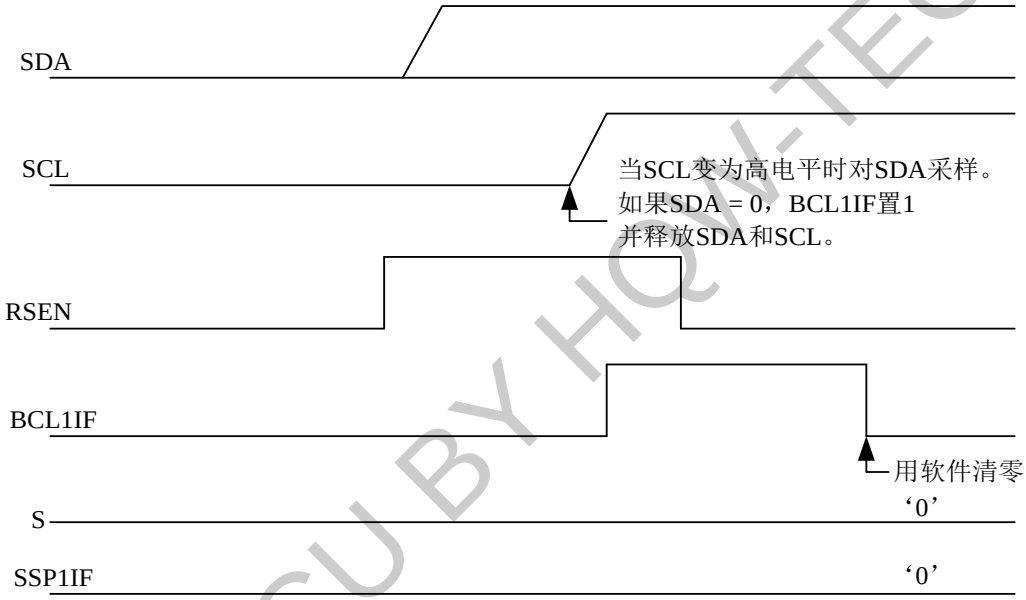


图25-36：重复启动条件期间的总线冲突（情形1）

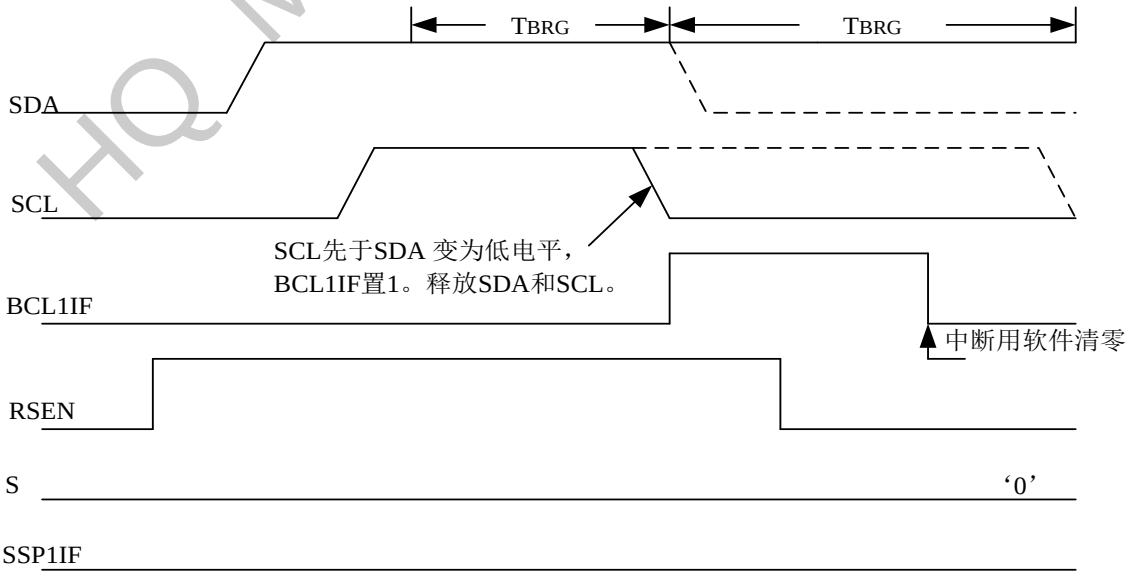


图25-37：重复启动条件期间的总线冲突（情行2）

25.6.13.3 停止条件期间的总线冲突

在停止条件期间，如果发生以下情况，则会发生总线冲突 ：

a) SDA 已被置为无效并允许悬空为高电平之后，SDA 在BRG 超时后被采样到低电平（情形1）。

b) SCL 引脚被置为无效之后，SCL 在SDA 变成高电平之前被采样到低电平（情形2）。

停止条件从SDA 被置为低电平开始。当SDA 采样为低电平时，允许SCL 引脚悬空。当引脚被采样到高电平（时钟仲裁）时，波特率发生器装入SSP1ADD 的值并递减计数至0。BRG 超时后， SDA 被采样。如果SDA采样为低电平，则已发生总线冲突。这是因为另一个主器件正试图发送一个数据0 （图25-38）。如果在允许SDA悬空为高电平前SCL 引脚被采样到低电平，也会发生总线冲突。这是另一个主器件正试图发送一个数据0的另外一种情况（图25-39）。

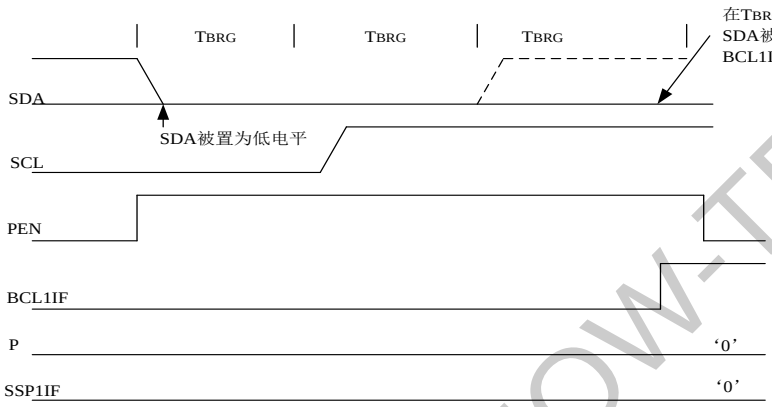


图25-38：停止条件期间的总线冲突（情形1）

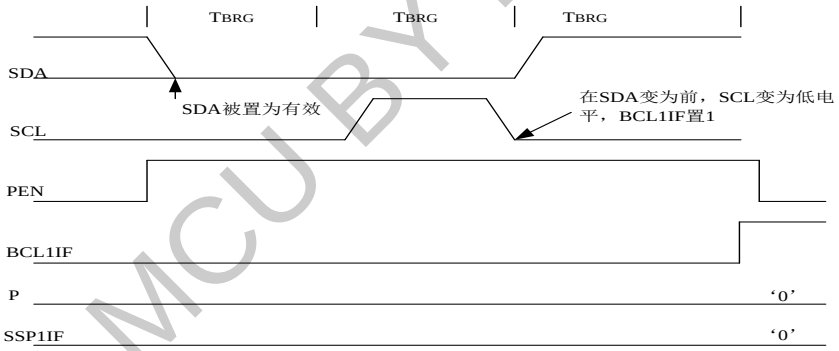


图25-39：停止条件期间的总线冲突（情形2）

25.7 波特率发生器

MSSP1 模块具有一个波特率发生器，可用于在I^2C 和SPI 主模式下产生时钟。波特率发生器（BRG）重载值放在SSP1ADD 寄存器（寄存器212H）中。当发生对SSP1BUF 的写操作时，波特率发生器将自动开始递减计数。在给定操作完成时，内部时钟会自动停止计数，并且时钟引脚将保持它的最后状态。图25-40 中的内部信号“ 重载” 会触发将SSP1ADD 值装入BRG 计数器。对于模块时钟线的每次振荡，这会发生两次。指定重载信号何时置为有效的逻辑依赖于MSSP1 当前的工作模式。表25-2 列出了不同的指令周期下的时钟速率以及装入SSP1ADD的BRG 值。

公式25-1：

$$FCLOCK = \frac{FOSC}{(SSPxADD + 1)(4)}$$

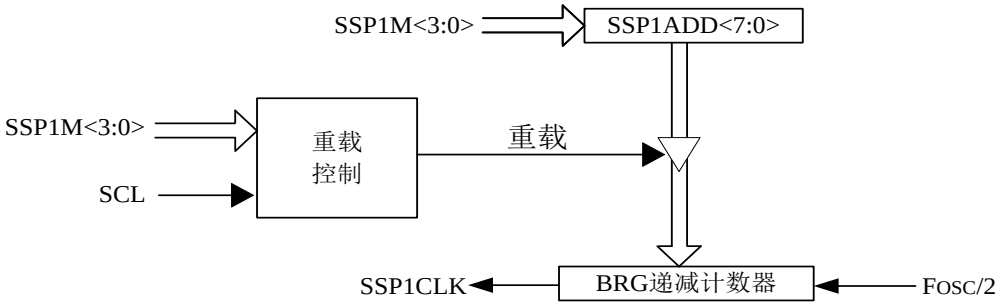


图 25-40：波特率发生器框图

注： 在用作I2C的波特率发生器时，值0x00、0x01和0x02 对于SSP1ADD 是无效的。这是实现限制。

表 25-2：使用 BRG 的 MSSP1 时钟速率

FOSC	F_{CY}	BRG 值	F_{CLOCK}（两次 BRG 计满返回）
16MHz	4MHz	09H	400kHz (1)
16MHz	4MHz	0CH	308kHz
16MHz	4MHz	27H	100kHz
4MHz	1MHz	09H	100kHz

注： 1： 虽然I2C 接口各方面都不符合400 kHz I2C 规范（该规范适用于大于100 kHz 的频率），但在需要较高频率的应用场合可以慎重使用。

25.8 寄存器说明

寄存器214H：SSP1状态寄存器（SSP1STAT）

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
SMP	**CKE**	**D/$\overline{A}$**	**P**	**S**	**R/$\overline{W}$**	**UA**	**BF**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	x= 未知
-n=POR时的值	1= 置1	0= 清零	u= 不变

bit7 **SMP**：SPI 数据输入采样位
SPI 主模式：
1 = 在数据输出时间的末端采样输入数据
0 = 在数据输出时间的中间采样输入数据
SPI 从模式：
当SPI 工作在从模式时，必须将SMP 清零
在I^2C 主模式或从模式下：
1 = 禁止标准速度模式下的压摆率控制（100 kHz 和1 MHz）
0 = 使能高速模式下的压摆率控制（400 kHz）

bit6 **CKE**：SPI 时钟边沿选择位（仅限SPI 模式）
在SPI 主模式或从模式下：
1 = 时钟状态从有效转换到空闲时发送
0 = 时钟状态从空闲转换到有效时发送
仅在I^2C模式下：

1 = 使能输入逻辑以使阈值符合SMBus 规范
0 = 禁止SMBus 特定输入

bit5 **D/$\overline{A}$**：数据/$\overline{地址}$位（仅限I²C 模式）
1 = 指示上一个接收或发送的字节是数据
0 = 指示上一个接收或发送的字节是地址

bit4 **P**：停止位
仅限I2C 模式。在MSSP1 模块被禁止且SSP1EN 被清零时，该位会被清零。）
1 = 指示上次检测到停止位（该位在复位时为0）
0 = 上次未检测到停止位

bit3 **S**：启动位
（仅限I2C 模式。在MSSP1 模块被禁止且SSP1EN 被清零时，该位会被清零。）
1 = 指示上次检测到启动位（该位在复位时为0）
0 = 上次未检测到启动位

bit2 **R/$\overline{W}$**：读/$\overline{写}$位信息（仅限I²C 模式）
该位保存上一次地址匹配后的R/$\overline{W}$位信息。该位仅在从地址匹配到出现下一个启动位、停止位或非$\overline{ACK}$位之间有效。
在I²C 从模式下：
1 = 读
0 = 写
在I²C 主模式下：
1 = 正在进行发送
0 = 未进行发送
将该位与SEN、RSEN、PEN、RCEN 或ACKEN 进行逻辑或运算将指示MSSP1 是否处于空闲模式。

bit1 **UA**：更新地址位（仅限10 位I²C 模式）
1 = 指示用户需要更新SSP1ADD 寄存器中的地址
0 = 不需要更新地址

bit0 **BF**：缓冲区满状态位
接收（SPI 和I²C 模式）：
1 = 接收完成， SSP1BUF 已满
0 = 接收未完成， SSP1BUF 为空
发送（仅限I²C 模式）：
1 = 数据发送正在进行（不包括$\overline{ACK}$位和停止位）， SSP1BUF 已满
0 = 数据发送完成（不包括$\overline{ACK}$位和停止位）， SSP1BUF 为空

寄存器215H：SSP1控制寄存器1（SSP1CON1）

R/C/HS-0/0	R/C/HS-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WCOL	**SSP1OV**	**SSP1EN**	**CKP**	**SSP1M<3:0>**
bit7				bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	u = 不变	HS = 硬件置1位
		0 = 清零	C = 未知

bit7 **WCOL**：写冲突检测位
主模式：
1 = 当I2C 不满足启动发送数据的条件时，试图向SSP1BUF 寄存器写入数据

0 = 未发生冲突

从模式：

1 = 正在发送前一个字时，又有数据写入SSP1BUF 寄存器（必须用软件清零）

0 = 未发生冲突

bit6 **SSP1OV：**接收上溢指示位(1)

在SPI 模式下：

1 = SSP1BUF 寄存器中仍保存前一数据时，又接收到一个新的字节。如果发生上溢， SSP1SR 中的数据会丢失。上溢只会在从模式下发生。在从模式下，即使只是发送数据，用户也必须读SSP1BUF，以避免将上溢位置1。在主模式下，上溢位不会被置1，因为每次接收（和发送）新数据都是通过写入SSP1BUF 寄存器启动的（必须用软件清零）。

0 = 无上溢

在I^2C 模式下：

1 = SSP1BUF寄存器中仍保存前一字节时，又接收到一个新的字节。在发送模式下，SSP1OV是“无关位”（必须用软件清零）。

0 = 无上溢

bit5 **SSP1EN：**同步串口使能位

在两种模式下，当使能时，必须将这些引脚正确地配置为输入或输出

在SPI 模式下：

1 = 使能串口并将SCK、SDO、SDI 和SS 配置为串口引脚源(2)

0 = 禁止串口并将上述引脚配置为I/O 端口引脚

在I^2C 模式下：

1 = 使能串口并将SDA 和SCL 引脚配置为串口引脚源(3)

0 = 禁止串口并将上述引脚配置为I/O 端口引脚

bit4 **CKP：**时钟极性选择位

在SPI 模式下：

1 = 时钟的空闲状态为高电平

0 = 时钟的空闲状态为低电平

在I^2C 从模式下：

SCL 释放控制

1 = 使能时钟

0 = 保持时钟为低电平（时钟延长）（用来确保数据建立时间。）

在I^2C 主模式下：

在此模式下未使用

bit3-0 **SSP1M<3:0>：**同步串口模式选择位

0000 = SPI 主模式，时钟 = FOSC/4

0001 = SPI 主模式，时钟 = FOSC/16

0010 = SPI 主模式，时钟 = FOSC/64

0011 = SPI 主模式，时钟 = TMR2 输出/2

0100 = SPI 从模式，时钟 = SCK 引脚，使能SS引脚控制

0101 = SPI 从模式，时钟 = SCK 引脚，禁止SS引脚控制，SS可用作I/O引脚

0110 = I^2C 从模式， 7 位地址

0111 = I^2C 从模式， 10 位地址

1000 = I^2C 主模式，时钟 = FOSC / (4 * (SSP1ADD+1))(4)

1001 = 保留

1010 = SPI 主模式，时钟 = FOSC/(4 * (SSP1ADD+1))(5)

1011 = I^2C 固件控制的主模式（从器件空闲）

1100 = 保留
1101 = 保留
1110 = I^2C 从模式， 7 位地址，并允许启动位和停止位中断
1111 = I^2C 从模式， 10 位地址，并允许启动位和停止位中断

注
1： 在主模式下，上溢位不会被置1，因为每次接收（和发送）新数据都是通过写入SSP1BUF 存器启动的。
2： 当使能时，必须将这些引脚正确地配置为输入或输出。
3： 当使能时，必须将SDA 和SCL 引脚配置为输入引脚。
4： 对于I^2C 模式，不支持SSP1ADD 值0、1 或2。
5： SSP1ADD 值为0 不支持。取而代之，使用SSP1M = 0000。

寄存器216H：SSP1控制寄存器2（SSP1CON2）

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0
GCEN	**ACKSTAT**	**ACKDT**	**ACKEN**	**RCEN**	**PEN**	**RSEN**	**SEN**
bit7							bit0

图注：		U = 未实现位，读为0	x = 未知
R = 可读位	W = 可写位	u = 不变	HS = 硬件置1位
-n = POR时的值	1 = 置1	0 = 清零	S = 用户置1位

bit7 **GCEN**：广播呼叫使能位（仅限I^2C 从模式）
1 = 当SSP1SR 接收到广播呼叫地址（0x00 或00h）时允许中断
0 = 禁止广播呼叫地址

bit6 **ACKSTAT**：应答状态位（仅限I^2C 模式）
1 = 未接收到应答
0 = 接收到应答

bit5 **ACKDT**：应答数据位（仅限I^2C 模式）
在接收模式下：
当用户在接收结束时发出一个应答序列时要发送的值
1 = 无应答
0 = 应答

bit4 **ACKEN**：应答序列使能位（仅限I^2C 主模式）
在主接收模式下：
1 = 在SDA 和SCL 引脚上发出应答序列，并发送ACKDT 数据位。由硬件自动清零。
0 = 应答序列空闲

bit3 **RCEN**：接收使能位（仅限I^2C 主模式）
1 = 使能I^2C 接收模式
0 = 接收空闲

bit2 **PEN**：停止条件使能位（仅限I^2C 主模式）
SCK 释放控制：
1 = 在SDA 和SCL 引脚上发出停止条件。由硬件自动清零。
0 = 停止条件空闲

bit1 **RSEN**：重复启动条件使能位（仅限I^2C 主模式）
1 = 在SDA 和SCL 引脚上发出重复启动条件。由硬件自动清零。
0 = 重复启动条件空闲

bit0 **SEN**：启动条件使能位（仅限I^2C 主模式）
在主模式下：

1 = 在SDA 和SCL 引脚上发出启动条件。由硬件自动清零。
0 = 启动条件空闲
在从模式下：
1 = 为从发送和从接收（已使能时钟延长）使能时钟延长
0 = 禁止时钟延长

注 1： 对于ACKEN、RCEN、PEN、RSEN 和SEN 位：如果I²C 模块不处于空闲模式，该位可能不会被置1（不支持并行操作），并且可能不会写入SSP1BUF （或禁止写入SSP1BUF）。

寄存器217H：SSP1控制寄存器3（SSP1CON3）

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACKTIM	**PCIE**	**SCIE**	**BOEN**	**SDAHT**	**SBCDE**	**AHEN**	**DHEN**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	u = 不变
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7 **ACKTIM**：应答时间状态位（仅限I²C 模式）(3)
1 = 指示I²C 总线处于应答序列中，在SCL 时钟的第8 个下降沿置1
0 = 不处于应答序列中，在SCL 时钟的第9 个上升沿清零

bit6 **PCIE**：停止条件中断允许位（仅限I²C 模式）
1 = 在检测到停止条件时允许中断
0 = 禁止检测到停止条件时的中断(2)

bit5 **SCIE**：启动条件中断允许位（仅限I²C 模式）
1 = 在检测到启动或重复启动条件时允许中断
0 = 禁止检测到启动条件时的中断(2)

bit4 **BOEN**：缓冲区改写使能位
在SPI 从模式下：(1)
1 = SSP1BUF 在每次新的数据字节移入时更新，并忽略BF 位
0 = 如果在接收到新字节时SSP1STAT 寄存器的BF 位已置1，则SSP1CON1 寄存器的SSP1OV位会置1，并且不会更新缓冲区
在I²C主模式和SPI主模式下：
该位被忽略。
在I2 C 从模式下：
1 = 仅当BF 位 = 0 时，在接收到地址/ 数据字节时，更新SSP1BUF 并产生ACK 信号，忽略SSP1OV 位的状态。
0 = 只有在SSP1OV 清零时才更新SSP1BUF

bit3 **SDAHT**：SDA 保持时间选择位（仅限I2C 模式）
1 = 在SCL 的下降沿之后，在SDA 上最少有300 ns 的保持时间
0 = 在SCL 的下降沿之后，在SDA 上最少有100 ns 的保持时间

bit2 **SBCDE**：从模式总线冲突检测使能位（仅限I2C 从模式）
如果在SCL 的上升沿，在模块输出高电平状态时采样到SDA 为低电平，则PIFB2 寄存器的BCL1IF 位会置1，总线会变为空闲状态
1 = 允许从器件总线冲突中断
0 = 禁止从器件总线冲突中断

bit1 **AHEN**：地址保持使能位（仅限I2C 从模式）
1 = 在所接收匹配地址字节的第8 个SCL 下降沿之后，SSP1CON1 寄存器的CKP 位将清零，

SCL 将保持低电平。

0 = 禁止地址保持

bit0 **DHEN:** 数据保持使能位（仅限I²C从模式）

1 = 在所接收数据字节的第8 个SCL 下降沿之后，从器件硬件清零SSP1CON1 寄存器的CKP 位，而SCL 则保持低电平。

0 = 禁止数据保持

注 1： 用于菊花链SPI 操作；使用户可以忽略除最后一个接收到的字节之外的所有字节。在接收到新字节且BF = 1 时， SSP1OV 仍然会置1，但硬件会继续将最新字节写入SSP1BUF。

2： 在启动和停止条件检测明确列为使能的从模式下，该位没有任何作用。

3： ACKTIM 状态位仅在AHEN 位或DHEN 位置1 时有效。

寄存器213H：SSP1掩码寄存器1（SSP1MSK）

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
MSK<7:0>							
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	u = 不变
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-1 **MSK<7:1>**：掩码位

1 = 接收到的地址bit n 与SSP1ADD<n> 相比较来检测I²C 模式下地址是否匹配

0 = 接收到的地址bit n 不用于检测I²C 模式下地址是否匹配

bit0 **MSK<0>**：用于I²C 从模式， 10 位地址的掩码位

I²C 从模式， 10 位地址（SSP1M<3:0> = 0111 或1111）：

1 = 接收到的地址bit 0 与SSP1ADD<0> 相比较来检测I²C 模式下地址是否匹配

0 = 接收到的地址bit 0 不用于检测I²C 模式下地址是否匹配I²C 从模式， 7 位地址，该位被忽略

寄存器212H：SSP1地址和波特率寄存器（I2C模式）（SSP1ADD）

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADD<7:0>							
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	u = 不变
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

主模式：

bit7-0 **ADD<7:0>**：波特率时钟分频比位

SCL 引脚时钟周期 = ((ADD<7:0> + 1) *4)/FOSC

10 位从模式—— 高位地址字节：

bit7-3 未使用：不使用高位地址字节。该寄存器的位状态为“无关”。主器件发送的位格式由I²C 规范确定，必须等于11110。但是，那些位通过硬件进行比较，并且不受该寄存器中的值影响。

bit2-1 **ADD<2:1>**：10 位地址的高2 位

bit0 未使用：在此模式下未使用。位状态为“无关”。

10 位从模式—— 低位地址字节：

bit7-0 **ADD<7:0>**：10 位地址的低8 位

7 位从模式：

bit7-1 **ADD<7:1>:** 7 位地址

bit0 未使用：在此模式下未使用。位状态为“无关”。

寄存器211H：移位寄存器SSP1SR的缓冲寄存器（SSP1BUF）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7-0 **SSP1BUF<7:0>**：移位寄存器SSP1SR的缓冲寄存器

表 25-3：与 SPI 操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	---1 -111	---1 -111
ADINSC	—	—	—	—	AN7	AN6	AN5	AN4	---- 1111	---- 1111
APFCON	RXDTSEL	SDOSEL	SSSEL	—	T1GSEL	TXCKSEL	P1BSEL	CCP1SEL	000- 0000	000- 0000
INTS	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000000x	0000000u
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	00000000	00000000
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	00000000	00000000
SSP1BUF	同步串行口接收缓冲/ 发送寄存器								xxxxxxxx	uuuuuuuu
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				00000000	00000000
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	00000000	00000000
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	00000000	00000000
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	--11 1111	--11 1111

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。 BOR 不使用阴影单元。

注 1： 其他（非上电）复位包括正常工作时的$\overline{MCLR}$复位和看门狗定时器复位。

表 25-4：与 I^2C 操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000000x	0000000u
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	00000000	00000000
PIEB2	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	—	00000---	00000---
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	00000000	00000000
PIFB2	OSFIF	C2IF	C1IF	EEIF	BCL1IF	—	—	—	00000---	00000---
SSP1ADD	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0	00000000	00000000
SSP1BUF	同步串行口接收缓冲/ 发送寄存器								xxxxxxxx	uuuuuuuu
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				00000000	00000000
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	00000000	00000000
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	00000000	00000000
SSP1MSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	1111 1111
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	00000000	00000000
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	--11 1111	--11 1111

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。 BOR 不使用阴影单元。

注 1： 其他（非上电）复位包括正常工作时的$\overline{MCLR}$复位和看门狗定时器复位。

26.0 增强型通用同步/异步收发器（EUSART）

增强型通用同步/异步收发器（EUSART）模块是一种串行I/O通信外设。它包含用来完成与器件程序执行无关的输入或输出串行数据传输所需的所有时钟发生器、移位寄存器和数据缓冲区等。EUSART也可称为串行通信接口（Serial Communications Interface，SCI），可配置为全双工异步系统或半双工同步系统。全双工模式可用来与外设系统通信，如CRT终端和个人计算机。半双工同步模式用于与外设器件通信，如A/D或D/A集成电路、串行EEPROM或其他单片机。这些器件通常不具备用以产生波特率的内部时钟，并需要由主同步器件提供外部时钟信号。

EUSART模块具备以下功能：

- 全双工异步收发
- 双字符输入缓冲区
- 单字符输出缓冲区
- 可编程8位或9位字符长度
- 9位模式下的地址检测
- 输入缓冲区溢出错误检测
- 接收字符帧错误检测
- 半双工同步主模式
- 半双工同步从模式
- 同步模式下的可编程时钟极性
- 休眠模式下的操作

EUSART模块还具备以下特性，使其成为局域互联网（Local Interconnect Network，LIN）总线系统的理想选择：

- 自动检测和波特率校准
- 接收到间隔字符时唤醒
- 13位间隔字符发送

EUSART发送器和接收器的框图如图26-1和图26-2所示。

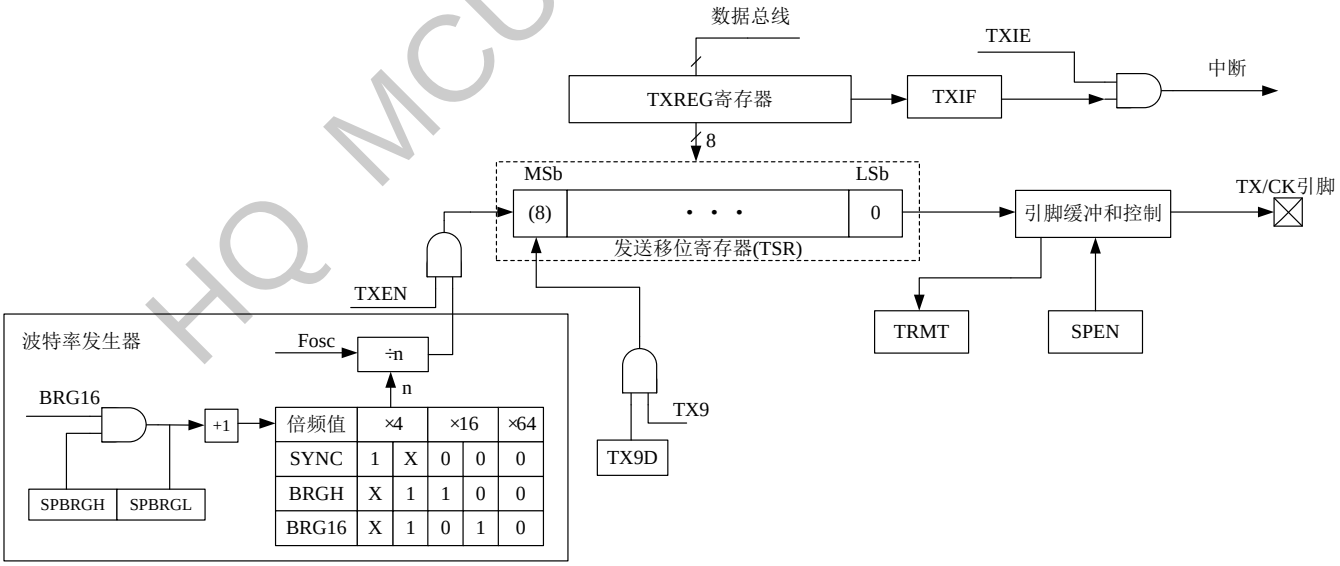


图26-1： EUSART 发送框图

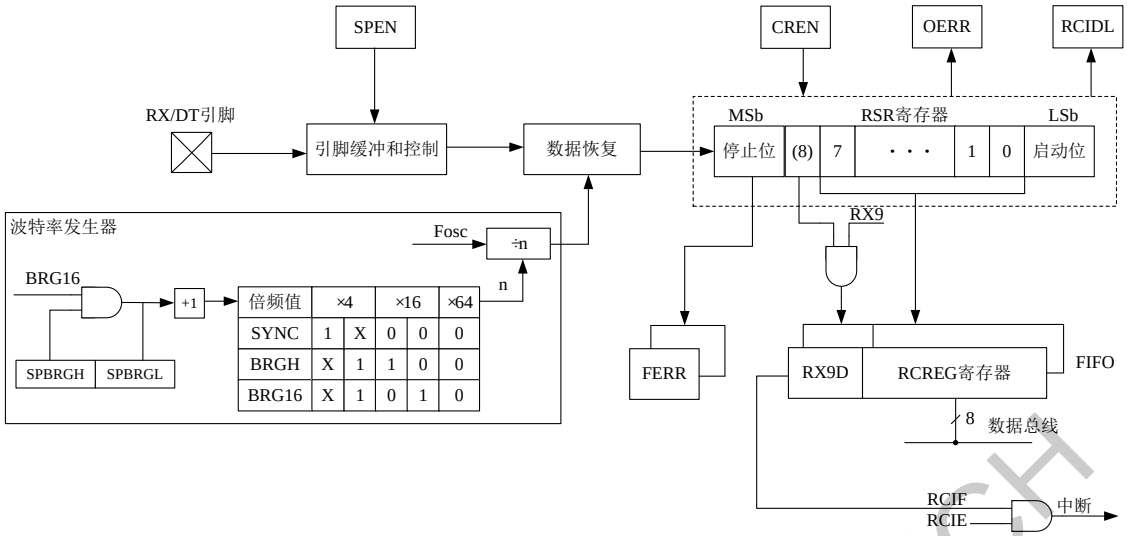


图 26-2： EUSART 接收框图

EUSART模块的操作由以下3个寄存器控制：

- 发送状态和控制寄存器（TXSTA）
- 接收状态和控制寄存器（RCSTA）
- 波特率控制寄存器（BAUDCON）

这些寄存器的详细信息请分别参见寄存器19EH、寄存器19DH 和寄存器19FH。

当未使能接收器或发送器部分时，对应的RX或TX引脚可用于通用输入和输出。

26.1 EUSART 异步模式

EUSART采用标准不归零（non-return-to-zero，NRZ）格式发送和接收数据。NRZ实现为两种电平：VOH标记状态（mark state）代表“1”数据位，而VOL空格状态（space state）代表“0”数据位。NRZ指的是连续发送具有相同值的数据位时，输出电平始终保持不变，而不会在发送完每个位之后回到中间电平。NRZ发送端口在标记状态空闲。每个字符发送包含1个启动位及随后的8个或9个数据位，并始终由1个或多个停止位终止。启动位始终是一个空格，停止位始终是标记。最常见的数据格式为8位。每个发送位持续时间为1/(波特率)。使用片上专用8位/16位波特率发生器从系统振荡器产生标准波特率频率。波特率配置示例请参见表26-2。

EUSART先发送和接收LSb。EUSART的发送器和接收器在功能上是相互独立的，但它们的数据格式和波特率相同。硬件不支持奇偶校验，但可通过软件实现并作为第9个数据位存储。

26.1.1 EUSART 异步发送器

图 26-1 给出了 EUSART 发送器框图。发送器的核心是串行发送移位寄存器（Transmit Shift Register，TSR），该寄存器不可用软件直接访问。TSR 从发送缓冲区（即 TXREG 寄存器）取得数据。

26.1.1.1 使能发送器

EUSART发送器可通过配置以下3个控制位使能为异步操作：

- TXEN = 1
- SYNC = 0
- SPEN = 1

假定所有其他EUSART控制位均处于其默认状态。

将 TXSTA 寄存器的 TXEN 位置 1 使能 EUSART 的发送器电路。清零 TXSTA 寄存器的 SYNC 位将 EUSART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1 可使能 EUSART，且自动将 TX/CK I/O 引脚配置为输出。如果 TX/CK 引脚与模拟外设共用，则模拟 I/O 功能必须通过清零相应的 ADINS 位禁止。

注	1：TXEN 中断允许位置1 时，TXIF 发送器中断标志位置1。

26.1.1.2 发送数据

向TXREG寄存器写入一个字符时启动发送。如果这是首字符，或前一个字符被完全从TSR中送出，TXREG中的数据就立即被传送到TSR寄存器。如果TSR中仍保存前一个字符的全部或部分，则新字符被保存在TXREG中，直到前一个字符的停止位被发送。之后，在TXREG中等待的字符在停止位发送后1 个T_{CY}内被传送到TSR中。TXREG中的数据被传送到TSR后，启动位、数据位和停止位的序列发送立即开始。

26.1.1.3 发送中断标志

只要EUSART发送器被使能且TXREG中没有等待发送的字符， PIFB1寄存器的TXIF中断标志位就被置1。换句话说，只有在TSR正在处理字符且TXREG中还有一个排队等待发送的新字符时，TXIF位才被清零。写入TXREG后并不立即清零TXIF标志位，而是在之后的第二个指令周期将其清零。写入TXREG后立即查询TXIF位将返回无效结果。TXIF位是只读的，不能用软件置1或清零。将PIEB1寄存器的TXIE中断允许位置1可允许TXIF中断。但是，只要TXREG为空，TXIF标志位就会被置1，无论TXIE中断允许位的状态如何。

要在发送数据时使用中断，应只在仍有数据要发送时才将 TXIE 位置 1。在将发送的最后一个字符写入 TXREG 后应清零 TXIE 中断允许位。

26.1.1.4 TSR 状态

TXSTA 寄存器的 TRMT 位指示 TSR 寄存器的状态。该位是只读位。TSR 寄存器为空时，TRMT 位置 1，而当一个字符从 TXREG 传送到 TSR 寄存器中时，该位清零。TRMT 位将保持清零，直到所有位移出 TSR 寄存器。该位不与任何中断逻辑有关，因此用户必须查询该位以确定 TSR 的状态。

注： TSR 寄存器不映射到数据存储器中，因此用户无法使用。

26.1.1.5 发送 9 位字符

EUSART支持9位字符发送。当TXSTA寄存器的TX9位置1时，EUSART将在发送每个字符时移出9位。TXSTA 寄存器的TX9D位是第9个数据位，也是最高有效位。发送9位数据时， TX9D数据位必须先于低8位写入TXREG。写入TXREG后，所有9个位将被立即传送到TSR移位寄存器中。有多个接收器时，可使用一种特殊的9位地址模式。关于地址模式的更多信息，请参见第26.1.2.7节“地址检测”。

26.1.1.6 异步发送设置

1. 初始化SPBRGH和SPBRGL寄存器对以及BRGH和BRG16位，获得所需的波特率（见第26.3节“EUSART波特率发生器（BRG）”）。
2. 通过清零SYNC位并将SPEN位置1，使能异步串口。
3. 如果需要9位发送，将TX9控制位置1。接收器置于地址检测模式时，第9个数据位置1表示低8个数据位为地址。
4. 将TXEN控制位置1使能发送。这将导致TXIF中断标志位置1。
5. 如果需要中断，将PIEB1寄存器的TXIE中断允许位置1。如果INTS寄存器的GIE和PEIE位也置1，则立即产生中断。
6. 如果选择了9位发送，应将第9位装入TX9D数据位。
7. 将 8 位数据装入 TXREG 寄存器。这将启动发送。

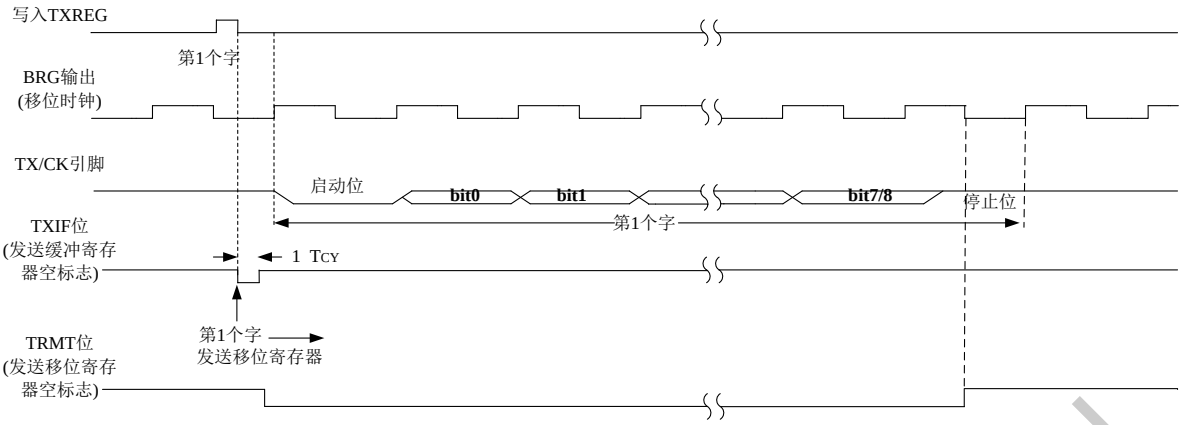
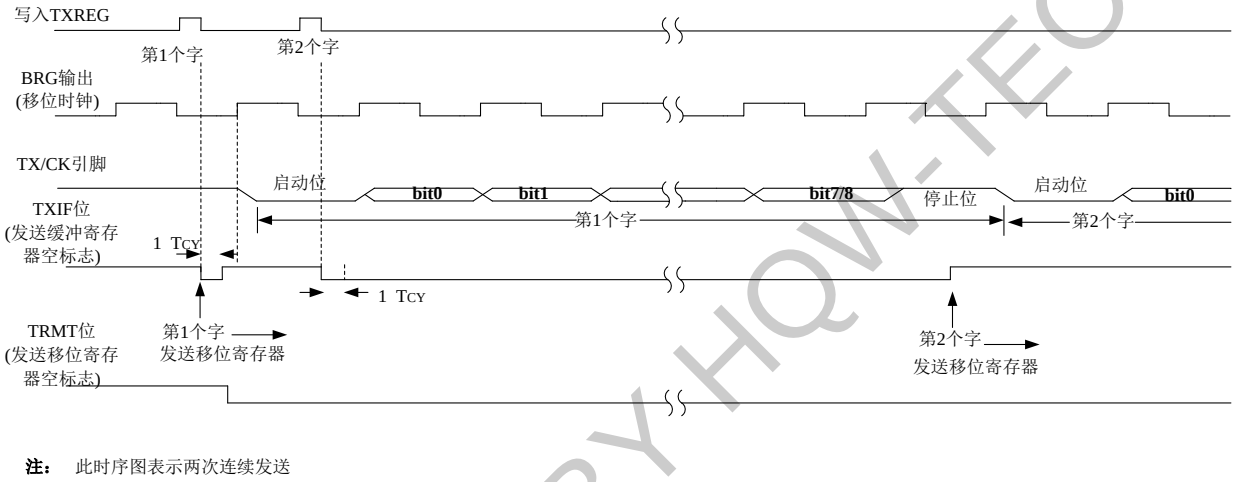


图 26-3： 异步发送



注： 此时序图表示两次连续发送

图 26-4：异步发送（背对背）

26.1.2 EUSART 异步接收器

异步模式通常用于RS-232系统中。图26-2给出了接收器框图。数据在RX/DT引脚上接收并驱动数据恢复模块。数据恢复模块实际上是一个高速移位器，工作频率为16倍波特率，而串行接收移位寄存器（Receive Shift Register，RSR）工作频率为比特率。所有8位或9位字符移入后被立即传送到双字符的先进先出（First-In-First-Out，FIFO）存储区中。FIFO缓冲区允许先接收两个完整字符和第三个字符的开始部分后，再启动软件服务EUSART 接收器。FIFO和RSR寄存器不能直接用软件访问。通过RCREG寄存器访问接收数据。

26.1.2.1 使能接收器

EUSART接收器可通过配置以下3个控制位使能为异步操作：

- CREN = 1
- SYNC = 0
- SPEN = 1

假定所有其他EUSART控制位均处于其默认状态。

将RCSTA寄存器的CREN位置1使能EUSART的接收器电路。清零TXSTA寄存器的SYNC位将EUSART配置为异步操作。将RCSTA寄存器的SPEN位置1可使能EUSART。程序员必须将相应的CPIO位置1，将RX/DT I/O引脚配置为输入。

注 1： 如果RX/DT 功能位于模拟引脚上，则必须清零相应的ADINS 位使接收器工作。

26.1.2.2 接收数据

接收器的数据恢复电路在第一位的下降沿启动字符接收。第一位也称启动（Start）位，始终为零。数据恢复电路计数传输半个位的时间至启动位的中点并验证该位是否仍为零。如果该位非零则数据恢复电路中止字符接收，不产生错误，并恢复寻找启动位的下降沿。如果启动位被验证为零，则数据恢复电路计数一整个位时间至下个位的中点。该位被一个择多检测电路采样，其结果（0 或 1）被移入 RSR。重复此过程直到所有数据位均被采样并移入 RSR。最后一个位时间被测量且其电平被采样。此为停止（Stop）位，始终为 1。如果数据恢复电路在停止位处采样到 0，则置 1 此字符的帧错误标志位，否则清零此字符的帧错误标志位。关于帧错误的更多信息，请参见第 26.1.2.4 节“ 接收帧错误”。所有数据位和停止位被接收后，RSR 中的字符就被立即传送到 EUSART 接收 FIFO，且 PIFB1 寄存器的 RCIF 中断标志位被置 1。读取 RCREG 寄存器时，FIFO 中顶部的字符被送出 FIFO。

如果接收FIFO溢出，在溢出条件被清除前不会接收更多字符。关于溢出错误的更多信息，请参见第26.1.2.5节“ 接收溢出错误”。

26.1.2.3 接收中断

只要EUSART 接收器被使能且接收FIFO中存在未被读取的字符，PIFB1寄存器的RCIF中断标志位就会被置1。RCIF中断标志位是只读位，不能用软件置1或清零。

将以下位置1可允许RCIF中断：

- PIEB1寄存器的RCIE中断允许位
- INTS寄存器的PEIE外设中断允许位
- INTS寄存器的GIE全局中断允许位

当FIFO中存在未被读取的字符时，无论中断允许位的状态如何， RCIF中断标志位均会被置1。

26.1.2.4 接收帧错误

接收FIFO缓冲区中的每个字符都有相应的帧错误状态位。帧错误表明在预期时间内未见到停止位。通过RCSTA寄存器的FERR位可访问帧错误状态。FERR位表示接收FIFO中顶部的未读字符的状态。因此，在读取RCREG前必须先读FERR位。FERR位是只读位，只用于接收FIFO中顶部的未读字符。帧错误（FERR= 1）并不会禁止接收更多字符。此时不必将FERR位清零。从FIFO缓冲区读出下一个字符将使FIFO进入下一个字符和下一个相应的帧错误。将RCSTA寄存器的SPEN位清零可复位EUSART，这样就可将FERR位强制清零。将RCSTA寄存器的CREN位清零不影响FERR位。自身产生的帧错误不会产生中断。

注： 如果接收FIFO中的所有接收字符均有帧错误，反复读取RCREG不会将FERR位清零。

26.1.2.5 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。在访问 FIFO 前接收到完整的第三个字符时会产生溢出错误。此时，RCSTA 寄存器的 OERR 位置 1。FIFO 缓冲区中已有的字符可被读出，但溢出错误被清除前不能再接收其他字符。将 RCSTA 寄存器的 CREN 位清零或通过将 RCSTA 寄存器的 SPEN 位清零复位 EUSART，可清除该错误。

26.1.2.6 接收 9 位字符

EUSART支持9位字符接收。当RCSTA寄存器的RX9位置1 时，EUSART将在接收每个字符时将9个位移入RSR。RCSTA寄存器的RX9D位是第9位，也是接收FIFO顶部未读字符的最高有效数据位。从接收FIFO缓冲区读取9位数据时，在读取RCREG的低8位前必须先读取RX9D数据位。

26.1.2.7 地址检测

当多个接收器共用同一条传输线时，有一个特殊的地址检测模式可供使用。将RCSTA寄存器的ADDEN位置1可使能地址检测。地址检测要求接收9位字符。使能地址检测时，只有第9个数据位置1的字符会被传送到接收FIFO缓冲区，并将RCIF中断标志位置1。所有其他字符均被忽略。接收到地址字符后，用户软件可判断地址是否与自身匹配。地址匹配时，发生下一个停止位前，用户软件必须通过清零ADDEN位禁止地址检测。当用户软件根据所使用的报文协议检测到报文的末尾时，软件将ADDEN位置1，将接收器重新置于地址检测模式。

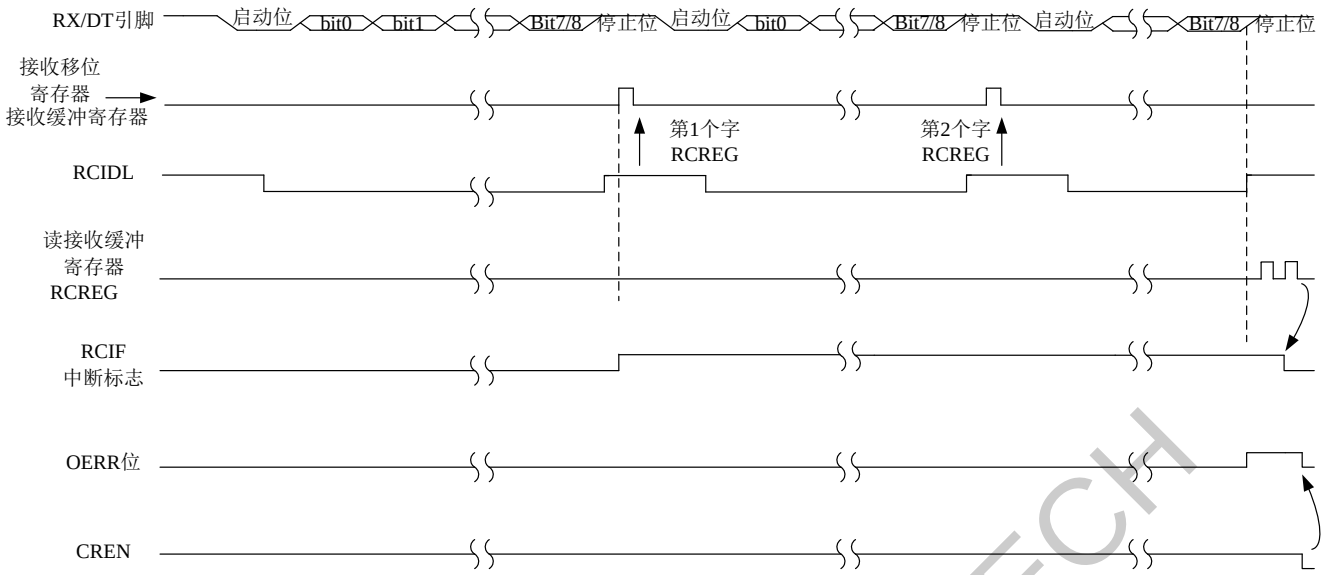
26.1.2.8 异步接收设置

1. 初始化SPBRGH和SPBRGL寄存器对以及BRGH和BRG16位，获得所需的波特率（见第26.3节“EUSART波特率发生器（BRG）”）。
2. 清零RX引脚的ADINS位（如适用）。
3. 将SPEN位置1使能串口。SYNC位必须清零才能进行异步操作。
4. 如果需要中断，将PIEB1寄存器的RCIE位以及INTS寄存器的GIE和PEIE位置1。
5. 如果需要接收9位数据，将RX9位置1。
6. 将CREN位置1使能接收。
7. 当字符从RSR被移入接收缓冲区时，RCIF中断标志位将被置1。如果RCIE中断允许位也置1，则产生中断。
8. 读取RCSTA寄存器取得错误标志和第9个数据位（9位数据接收使能时）。
9. 读取RCREG寄存器从接收缓冲区取得接收数据的低8位。
10. 发生溢出时，通过清零CREN接收器使能位清零OERR标志位。

26.1.2.9 9 位地址检测模式设置

设置使能地址检测的异步接收的步骤如下：

1. 初始化SPBRGH和SPBRGL寄存器对以及BRGH和BRG16位，获得所需的波特率（见第26.3节“EUSART波特率发生器（BRG）”）。
2. 清零RX引脚的ADINS位（如适用）。
3. 将SPEN位置1使能串口。SYNC位必须清零才能进行异步操作。
4. 如果需要中断，将PIEB1寄存器的RCIE位以及INTS寄存器的GIE和PEIE位置1。
5. 将RX9位置1使能9位接收。
6. 将ADDEN位置1使能地址检测。
7. 将CREN位置1使能接收。
8. 当第9位置1的字符从RSR被移入接收缓冲区时，RCIF中断标志位将被置1。如果RCIE中断允许位也置1，则产生中断。
9. 读取RCSTA寄存器取得错误标志。第9个数据位将始终置1。
10. 读取RCREG寄存器从接收缓冲区取得接收数据的低8位。软件将判断此地址是否是器件地址。
11. 发生溢出时，通过清零CREN接收器使能位清零OERR标志位。
12. 如果器件被寻址，将 ADDEN 位清零以允许所有接收到的数据被送入接收缓冲区并产生中断。



注： 此时序图显示了在RX输入引脚上顺序接收的3个字。在第3个字后读RCREG(接收缓冲区)会使OERR(溢出)位置1。

图 26-5： 异步接收

26.2 异步操作的时钟精度

内部振荡器模块输出（INTOSC）在出厂时做了校准。但是VDD或温度变化时，INTOSC频率有可能漂移，这将直接影响异步波特率可以通过调整波特率发生器的值来调整波特率时钟。自动波特率检测可自动完成这种调整（见第26.3.1节“自动波特率检测”）。通过调整波特率发生器来补偿外设时钟频率的逐渐变化时，可能无法足够细微地调节分辨率。

26.3 EUSART 波特率发生器（BRG）

波特率发生器（BRG）是8位或16位定时器，专用于支持异步和同步EUSART操作。默认情况下，BRG工作在8位模式下。将BAUDCON寄存器的BRG16位置1可选择16位模式。

SPBRGH和SPBRGL寄存器对决定自由运行波特率定时器的周期。在异步模式下，波特率周期的倍频值由TXSTA寄存器的BRGH位和BAUDCON寄存器的BRG16位决定。在同步模式下，BRGH位被忽略。表26-1提供了确定波特率的公式。例26-1提供了确定波特率和波特率误差的计算示例。

为便于使用，各种异步模式的典型波特率和误差值已经计算出来，如表26-1所示。使用高波特率（BRGH=1）或16位BRG（BRG16= 1）有助于降低波特率误差。16位BRG模式用于在高速振荡器频率下实现低波特率。将新值写入SPBRGH和SPBRGL寄存器对将导致BRG定时器复位（或清零）。这可以确保BRG无需等待定时器上溢就可以输出新的波特率。如果系统时钟在有效的接收操作过程中被更改，可能会导致接收错误或数据丢失。为避免此问题，应检查RCIDL位的状态，以确保在改变系统时钟前接收操作处于空闲状态。

例26-1： 计算波特率误差

针对工作在异步模式下、Fosc=16MHz、
目标波特率=9600 且采用 8 位 BRG 的器件：

$$目标波特率 = \frac{Fosc}{64([SPBRGH:SPBRGL]+1)}$$

求解 SPBRGH:SPBRGL:

$$X = \frac{\frac{Fosc}{目标波特率}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$计算波特率 = \frac{16000000}{64(25+1)}$$

$$= 9615$$

$$误差 = \frac{计算波特率 - 目标波特率}{目标波特率}$$

$$= \frac{(9615-9600)}{目标波特率} = 0.16\%$$

表26-1：波特率公式

配置位			BRG/EUSART 模式	波特率公式
SYNC	BRG16	BRGH		
0	0	0	8 位/异步	FOSC/[64(n+1)]
0	0	1	8 位/异步	FOSC/[16(n+1)]
0	1	0	16 位/异步	
0	1	1	16 位/异步	FOSC/[4(n+1)]
1	0	x	8 位/同步	
1	1	x	16 位/同步	

表26-2：异步模式下的波特率

波特率	SYNC = 0，BRGH = 0，BRG16 = 0								
	FOSC = 18.432MHz			FOSC = 11.0592MHz			FOSC = 8.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	—	—	—	—	—	—	—	—	—
1200	1200	0.00	239	1200	0.00	143	1202	0.16	103
2400	2400	0.00	119	2400	0.00	71	2404	0.16	51
9600	9600	0.00	29	9600	0.00	17	9615	0.16	12
10417	10286	-1.26	27	10165	-2.42	16	10417	0.00	11
19.2k	19.20k	0.00	14	19.20k	0.00	8	—	—	—
57.6k	57.60k	0.00	7	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0，BRGH = 0，BRG16 = 0								
	FOSC = 4.000MHz			FOSC = 3.6864MHz			FOSC = 1.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	25	2400	0.00	23	—	—	—
9600	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0，BRGH = 1，BRG16 = 0								
	FOSC = 18.432MHz			FOSC = 11.0592MHz			FOSC = 8.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	2404	0.16	207
9600	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.20k	0.00	59	19.20k	0.00	35	19.231	0.16	25
57.6k	57.60k	0.00	19	57.60k	0.00	11	55.55k	-3.55	8
115.2k	115.2k	0.00	9	115.2k	0.00	5	—	—	—

波特率	SYNC = 0，BRGH = 1，BRG16 = 0								
	FOSC = 4.000MHz			FOSC = 3.6864MHz			FOSC = 1.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	—	—	—	—	—	—	300	0.16	207
1200	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	25j
9600	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—

波特率	SYNC = 0，BRGH = 0，BRG16 = 1								
	FOSC = 18.432MHz			FOSC = 11.0592MHz			FOSC = 8.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	300.0	0.00	3839	300.0	0.00	2303	299.9	-0.02	1666
1200	1200	0.00	959	1200	0.00	575	1199	-0.08	416
2400	2400	0.00	479	2400	0.00	287	2404	0.16	207
9600	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.20k	0.00	59	19.20k	0.00	35	19.23k	0.16	25
57.6k	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	115.2k	0.00	9	115.2k	0.00	5	—	—	—

波特率	SYNC = 0，BRGH = 0，BRG16 =1								
	FOSC = 4.000MHz			FOSC = 3.6864MHz			FOSC = 1.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—

波特率	SYNC = 0，BRGH = 1，BRG16 = 1 或 SYNC = 1，BRG16 = 1								
	FOSC = 18.432MHz			FOSC = 11.0592MHz			FOSC = 8.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	300.0	0.00	15359	300.0	0.00	9215	300.0	0.00	6666
1200	1200	0.00	3839	1200	0.00	2303	1200	-0.02	1666
2400	2400	0.00	1919	2400	0.00	1151	2401	0.04	832
9600	9600	0.00	479	9600	0.00	287	9615	0.16	207
10417	10425	0.08	441	10433	0.16	264	10417	0	191
19.2k	19.20k	0.00	239	19.20k	0.00	143	19.23k	0.16	103
57.6k	57.60k	0.00	79	57.60k	0.00	47	57.14k	-0.79	34
115.2k	115.2k	0.00	39	115.2k	0.00	23	117.6k	2.12	16

波特率	SYNC = 0，BRGH = 1，BRG16 = 1 或 SYNC = 1，BRG16 = 1								
	FOSC = 4.000MHz			FOSC = 3.6864MHz			FOSC = 1.000MHz		
	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）	实际波特率	%误差	SPBRG值（十进制）
300	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

26.3.1 自动波特率检测

EUSART模块支持波特率自动检测和校准。在自动波特率检测（Auto-Baud Rate Detect，ABD）模式下，提供给BRG的时钟信号是反向的。BRG并不为传入的RX信号提供时钟信号，而是由RX信号为BRG定时。波特率发生器用于为接收的55h （ASCII“U”）定时，55h是LIN总线的同步字符。此字符的特殊之处在于它具有包括停止位边沿在内的5个上升沿。

将BAUDCON寄存器的ABDEN位置1 将启动自动波特率校验序列（图26-6）。当发生ABD序列时，EUSART状态机保持在空闲状态。在接收线的第一个上升沿（启动位之后），SPBRG使用BRG计数器时钟递增计数，如表26-3所示。在第8位周期的末尾将在RX引脚上出现第5个上升沿。此时，累计数据即正确的BRG 周期总数被保存在SPBRGH和SPBRGL寄存器对中，ABDEN位被自动清零而RCIF中断标志被置1。要清除RCIF中断，需要读取RCREG中的值。RCREG的内容应该被丢弃。在不使用SPBRGH寄存器的模式下进行校准时，用户可通过查询SPBRGH寄存器的值是否为00h来验证SPBRGL寄存器是否上溢。

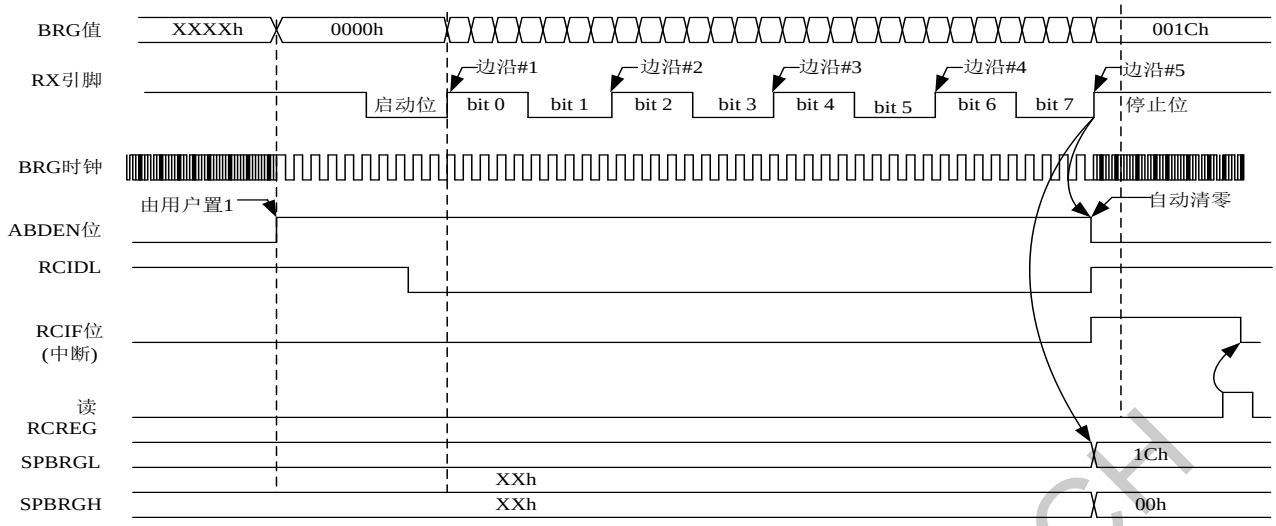
BRG自动波特率时钟由BRG16和BRGH位决定，如表26-3 所示。在ABD期间，SPBRGH和SPBRGL寄存器都被用作16位计数器，与BRG16位的设置无关。在校准波特率周期时，SPBRGH和SPBRGL寄存器的时钟频率为BRG基本时钟频率的1/8。得到的字节测量结果为全速时的平均位时间。

注 1： 如果WUE位和ABDEN位都置1，自动波特率检测将在间隔字符之后的字节开始（见第26.3.3 节“ 接收到间隔字符时自动唤醒”）。
2： 需要由用户来判断输入字符的波特率是否处于所选BRG时钟源范围内。可能无法实现某些振荡器频率和EUSART波特率组合。
3： 在自动波特率过程中，自动波特率计数器从1 开始计数。自动波特率序列完成后，为了得到最准确的结果，应从SPBRGH:SPBRGL寄存器对的值中减去1。

表26-3： BRG 计数器时钟速率

BRG16	BRGH	BRG 基本时钟	BRG ABD 时钟
0	0	FOSC/64	FOSC/512
0	1	FOSC/16	FOSC/128
1	0	FOSC/16	FOSC/128
1	1	FOSC/4	FOSC/32

注： 在 ABD 序列期间，SPBRGL 和 SPBRGH 寄存器都被用作 16 位计数器，与 BRG16 的设置无关。



注 1：ABD序列要求EUSART模块配置为工作在异步模式下

图26-6：自动波特率校准

26.3.2 自动波特率上溢

在自动波特率检测过程中，如果在RX 引脚上检测到第5 个上升沿之前波特率计数器上溢，则BAUDCON 寄存器的ABDOVF 位将被置1。ABDOVF 位指示计数器已超出SPBRGH:SPBRGL 寄存器对的16 位所能允许的最大计数值。在ABDOVF 置1 后，计数器将继续计数，直到在RX 引脚上检测到第5 个上升沿为止。一旦检测到第5 个RX 边沿，硬件会将RCIF 中断标志置1，并将BAUDCON 寄存器的ABDEN 位清零。可以通过读取RCREG 寄存器将RCIF 标志清零。BAUDCON 寄存器的ABDOVF 标志可以用软件直接清零。若要在RCIF 标志置1 前终止自动波特率进程，请先将ABDEN 位清零，然后将BAUDCON 寄存器的ABDOVF位清零。如果没有先将ABDEN 位清零，ABDOVF 位将保持置1 状态。

26.3.3 接收到间隔字符时自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于不工作状态，不能正常进行字符接收。自动唤醒功能使控制器可被RX/DT 线上的活动唤醒。该功能只在异步模式下可用。自动唤醒功能可通过将BAUDCON 寄存器的WUE位置1来使能。一旦置1，RX/DT 上的正常接收序列就被禁止，EUSART 保持在空闲状态，监视与CPU 模式无关的唤醒事件。唤醒事件包含RX/DT 线上电平由高至低的跳变。（这与同步间隔字符或LIN协议的唤醒信号字符的启动条件一致。）EUSART 模块产生的RCIF 中断与唤醒事件同步。在正常CPU工作模式下，中断产生与Q时钟同步（图26-7），而器件处于休眠模式时则异步产生（图26-8）。通过读RCREG寄存器可清除中断条件。RX线在间隔字符末尾由低至高的跳变将自动清零WUE位。这向用户表明间隔事件结束。此时，EUSART 模块处于空闲模式，等待接收下一个字符。

26.3.3.1 特殊注意事项

间隔字符

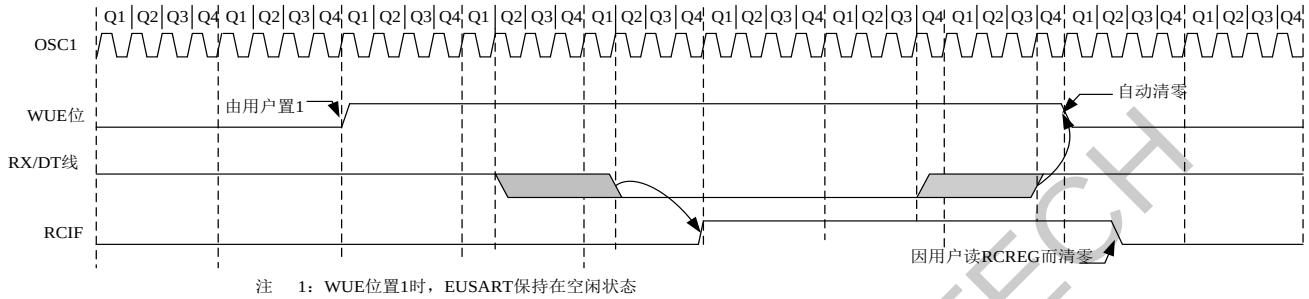
在发生唤醒事件期间为了避免字符错误或字符碎片，唤醒字符必须为全零。唤醒被使能时，其工作状况与数据流的低电平时间无关。如果WUE 位置1 并接收到了有效的非零字符，则从启动位至第一个上升沿的低电平时间将被解读为唤醒事件。字符的其余位将作为碎片字符接收，后续字符有可能产生帧错误或溢出错误。因此，发送的首字符必须为全0。这必须持续10 个或更长的位时间，对于LIN 总线，建议持续13 个位时间，而标准RS-232 器件，可为任意个位时间。

振荡器起振时间

必须考虑振荡器起振时间，特别在使用起振时间较长的振荡器（即，LP、XT 或HS模式）的应用中。同步间隔（或唤醒信号）字符必须足够长，并随后有一个足够长的间隔时间，以使所选的振荡器有足够的时间起振并在这段时间对EUSART 进行正确初始化。

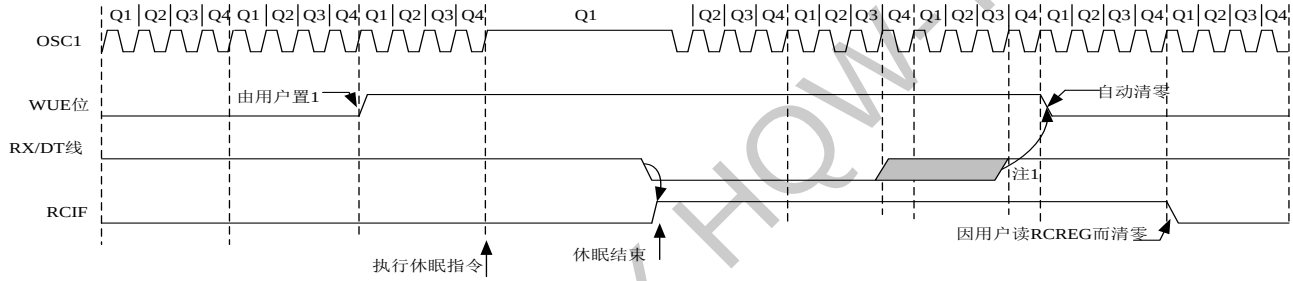
WUE 位

唤醒事件会通过将RCIF位置1产生一个接收中断。WUE位在RX/DT的上升沿由硬件清零。之后软件通过读取RCREG寄存器并丢弃其内容将中断条件清除。要确保不丢失实际数据，应在将WUE 位置1 前检查RCIDL 位，验证没有接收操作在进行。如果未发生接收操作，可在进入休眠模式前将WUE 位置1。



注 1：WUE位置1时，EUSART保持在空闲状态

图26-7：正常工作时的自动唤醒位（WUE）时序



注 1：如果唤醒事件需要较长的振荡器预热时间，在stposc信号仍然有效时间可发生WUE位自动清零。该时序与Q时钟出现与否五无关。
2：WUE位置1时，EUSART保持在空闲状态

图26-8：休眠时的自动唤醒位（WUE）时序

26.3.4 间隔字符序列

EUSART模块能够发送符合LIN 总线标准的特殊间隔字符序列。间隔字符包含1 个启动位以及随后的12个0位和1 个停止位。要发送间隔字符，应将TXSTA 寄存器的SENDB 和TXEN 位置1。随后对TXREG 执行写操作可启动间隔字符发送。写入TXREG的数据值会被忽略并发送全0。在发送了相应的停止位后，硬件会自动将SENDB 位复位。这样用户可以在间隔字符（在LIN 规范中通常是同步字符）后预先将下一个要发送字节装入发送FIFO。TXSTA 寄存器的TRMT位表明发送操作何时处于有效或空闲状态，这与正常发送时相同。图26-9 给出了发送间隔字符的时序。

26.3.4.1 间隔和同步发送序列

以下序列将启动报文帧头，它由间隔字符和其后的自动波特率同步字节组成。这是LIN 总线主器件的典型序列。

1. 将EUSART 配置为所需的模式。
2. 将TXEN 和SENDB 位置1 使能间隔序列。
3. 将无效字符装入TXREG，启动发送（该值会被忽略）。
4. 将“55h” 写入TXREG，以便将同步字符装入发送FIFO 缓冲区。
5. 发送间隔字符后， SENDB 位被硬件复位，同步字符随后被发送。

当TXREG 为空时（由TXIF 指出），下一个数据字节会写入TXREG。

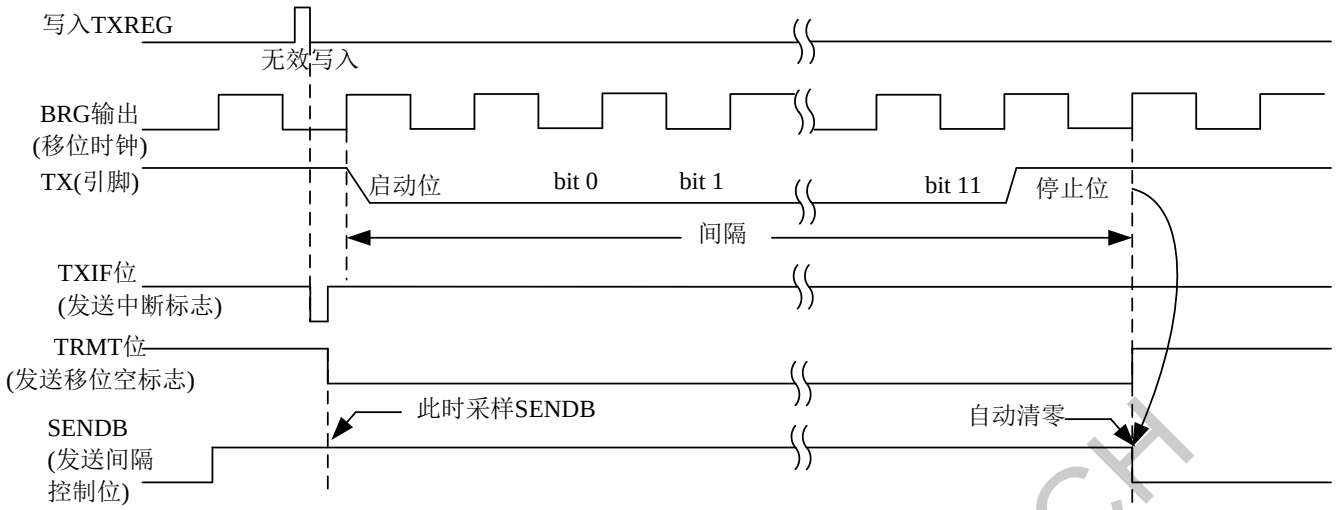


图26-9：发送间隔字符序列

26.3.5 接收间隔字符

增强型EUSART 模块接收间隔字符有两种方法。第一种检测间隔字符的方法采用RCSTA 寄存器的FERR 位和如RCREG 所指示的接收数据。假定波特率发生器已初始化为所需的波特率。

发生以下情况时，表明接收到间隔字符：

- RCIF 位被置1
- FERR 位被置1
- RCREG = 00h

第二种方法采用第26.3.3 节“ 接收到间隔字符时自动唤醒”中所述的自动唤醒功能。通过使能此功能，EUSART 将采样RX/DT 上的下两次跳变，产生RCIF 中断，并接收下一个数据字节并再产生一次中断。请注意，在间隔字符后，用户通常希望使能自动波特率检测功能。采用这两种方法时，用户均可在EUSART 进入休眠模式前将BAUDCON 寄存器的ABDEN 位置1。

26.4 EUSART 同步模式

同步串行通信通常用于具有一个主器件和一个或多个从器件的系统中。主器件包含生成波特率所需的电路，可将时钟提供给系统中的所有器件。从器件使用主时钟，可不再需要内部时钟生成电路。同步模式下有两条信号线：双向数据线和时钟线。从器件使用主器件提供的外部时钟将串行数据移入或移出相应的接收和发送移位寄存器。由于数据线是双向的，同步操作只能是半双工的。半双工指主从器件能够接收和发送数据，但不能同时进行。EUSART 可作为主器件，也可作为从器件。同步发送时不使用启动位和停止位。

26.4.1 同步主模式

使用以下位将EUSART 配置为同步主操作：

- SYNC = 1
- CSRC = 1
- SREN = 0 （用于发送）； SREN = 1 （用于接收）
- CREN = 0 （用于发送）； CREN = 1 （用于接收）
- SPEN = 1

将TXSTA 寄存器的SYNC 位置1 将器件配置为同步操作。将TXSTA 寄存器的CSRC 位置1 可将器件配置为主器件。将RCSTA 寄存器的SREN 和CREN 位清零可确保器件处于发送模式，否则器件将被配置为接收。将RCSTA 寄存器的SPEN 位置1 可使能EUSART。

26.4.1.1 主时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为主器件的器件将时钟信号发送到TX/CK 线上。EUSART 配置为同步发送或接收操作时，自动使能TX/CK 引脚输出驱动器。串行数据位在时钟前沿改变，以确保其在时钟的后沿有效。为每个数据位产生一个时钟周期。数据位有多少，就产生多少个时钟周期。

26.4.1.2 时钟极性

时钟极性通过BAUDCON 寄存器的SCKP 位选择。将SCKP 位置1将时钟空闲状态设置为高电平。SCKP 位置1 时，数据在每个时钟的下降沿改变。将SCKP 位清零将时钟空闲状态设置为低电平。SCKP 位清零时，数据在每个时钟的上升沿改变。

26.4.1.3 同步主发送

从器件的RX/DT 引脚输出数据。EUSART 配置为同步主发送操作时， RX/DT 和TX/CK 引脚的输出驱动器被自动使能。向TXREG 寄存器写入一个字符时启动发送。如果TSR中仍保存前一个字符的全部或部分，则新字符被保存在TXREG 中，直到前一个字符的最后一位被发送。如果这是首字符，或前一个字符被完全从TSR中送出，TXREG 中的数据就立即被传送到TSR。字符发送在数据从TXREG 送入TSR 后立即开始。每个数据位在主时钟的时钟前沿改变，并在下一个时钟前沿到来前保持有效。

注： TSR 寄存器并未映射到数据存储器中，因此用户无法使用。

26.4.1.4 同步主发送设置

1. 初始化SPBRGH 和SPBRGL 寄存器对以及BRGH 和BRG16 位，获得所需的波特率（见第26.3 节“EUSART 波特率发生器（BRG）”）。
2. 将SYNC、SPEN和CSRC位置1使能同步主串口。
3. 将SREN 和CREN 位清零禁止接收模式。
4. 将TXEN 位置1 使能发送模式。
5. 如果需要9 位发送，将TX9 位置1。
6. 如果需要中断，将PIEB1 寄存器的TXIE 位以及INTS 寄存器的GIE 和PEIE 位置1。
7. 如果选择了9位发送，应将第9位装入TX9D位。
8. 将数据装入TXREG 寄存器，启动发送。

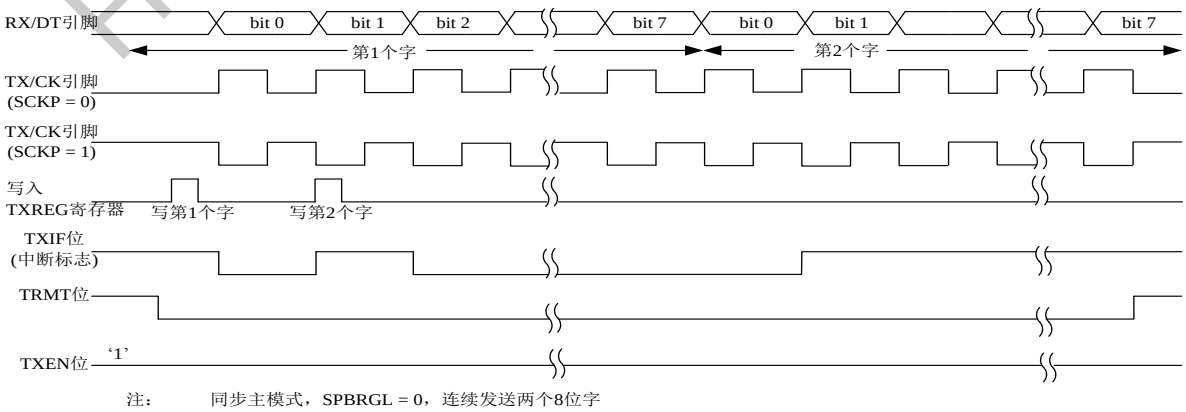


图26-10：同步发送

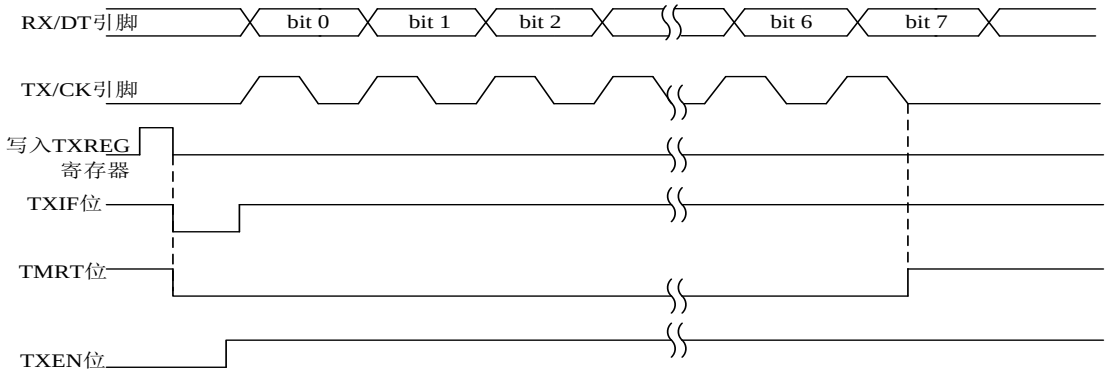


图26-11：同步发送（由TXEN 位控制）

26.4.1.5 同步主接收

数据在RX/DT 引脚上接收。将EUSART 配置为同步主接收操作时，自动禁止RX/DT 引脚输出驱动器。在同步模式下，可通过将单字节接收使能位（RCSTA寄存器的SREN）或连续接收使能位（RCSTA 寄存器的CREN）置1 使能接收。SREN 置1 且CREN 清零时，一个字符中有多少数据位就产生多少个时钟周期。一个字符接收完成后SREN 位被自动清零。CREN置1时，将连续产生时钟直到CREN被清零。如果CREN 在字符接收过程中被清零，则CK时钟立即停止，接收到的部分字符被丢弃。如果SREN和CREN 同时置1，则首字符接收完成时SREN 被清零， CREN 优先。要启动接收，将SREN 或CREN 置1。在TX/CK 时钟引脚的后沿对RX/DT引脚上的数据进行采样，并移入接收移位寄存器（RSR）。当完整的字符被接收进RSR后， RCIF 位置1 且该字符被自动送入两个字符的接收FIFO。接收FIFO 中顶部字符的低8 位在RCREG 中。只要接收FIFO 中有未读字符， RCIF 位就保持置1。

注：	如果RX/DT 功能位于模拟引脚上，则必须清零相应的ADINS 位使接收器工作。

26.4.1.6 从时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为从器件的器件在TX/CK 线上接收时钟信号。将器件配置为同步从器件发送或接收操作时，自动禁止TX/CK引脚输出驱动器。串行数据位在时钟前沿改变，以确保其在时钟的后沿有效。每个时钟周期传送一个数据位。数据位有多少，就产生多少个接收时钟周期。

注：	如果将器件配置为从器件并且TX/CK 功能位于模拟引脚上，则必须清零相应的ADINS 位。

26.4.1.7 接收溢出错误

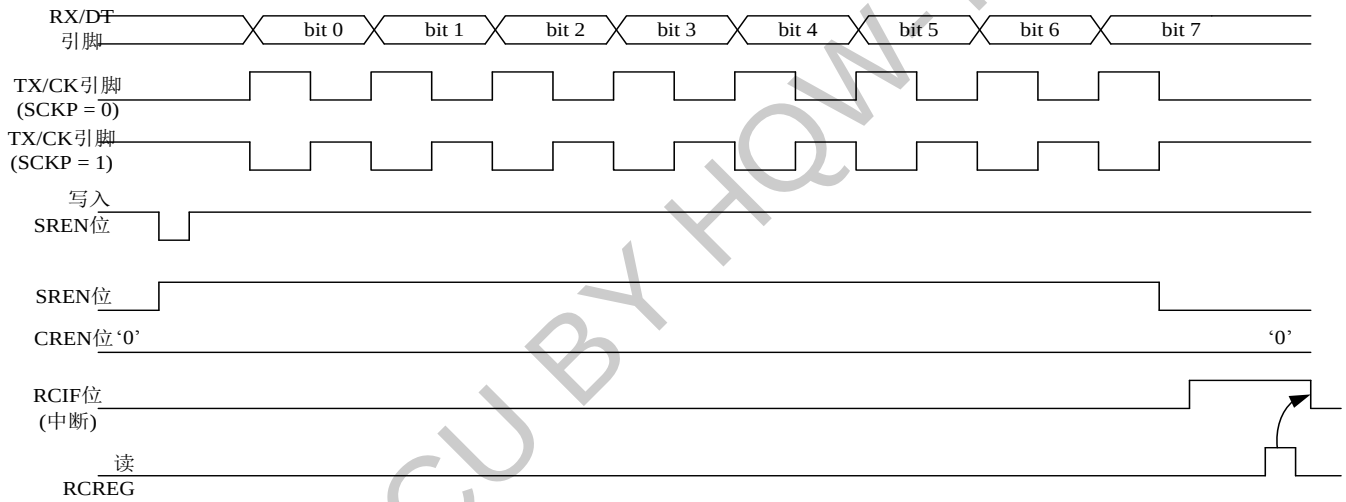
接收FIFO 缓冲区可容纳两个字符。在读RCREG 以访问FIFO 前，接收到完整的第三个字符时，则会产生溢出错误。此时，RCSTA 寄存器的OERR 位置1。FIFO中的前一个数据不会被覆盖。FIFO 缓冲区中的两个字符可被读出，但错误被清除前不能再接收其他字符。只有清除了溢出条件才可将OERR 位清零。如果SREN 位置1 且CREN 清零时发生溢出错误，则读取RCREG可清除错误。如果CREN 位置1 时发生溢出，则可通过清零RCSTA 寄存器的CREN位或清零可将使EUSART复位的SPEN 位清除错误条件。

26.4.1.8 接收 9 位字符

EUSART 支持9 位字符接收。当RCSTA 寄存器的RX9位置1 时，EUSART 将在接收每个字符时将9 个位移入RSR。RCSTA寄存器的RX9D 位是第9 位，也是接收FIFO 顶部未读字符的最高有效位。从接收FIFO 缓冲区读取9 位数据时，在读取RCREG 的低8 位前必须先读取RX9D 数据位。

26.4.1.9 同步主接收设置

1. 初始化SPBRGH:SPBRGL 寄存器对，获得所需的波特率。按需要将BRGH 和BRG16 位置1 或清零，获得所需的波特率。
2. 清零RX 引脚的ADINS 位（如适用）。
3. 将SYNC、SPEN和CSRC位置1使能同步主串口。
4. 确保将CREN 和SREN 位清零。
5. 如果需要中断，将PIEB1 寄存器的RCIE 位以及INTS 寄存器的GIE 和PEIE 位置1。
6. 如果需要接收9 位数据，将RX9 位置1。
7. 将SREN 位置1 启动接收，或将CREN 位置1使能连续接收。
8. 字符接收完成时中断标志位RCIF 将被置1。如果中断允许位RCIE 已置1，则产生中断。
9. 读取RCSTA寄存器取得第9 位（如果已使能），并确定接收时是否发生了错误。
10. 通过读取RCREG寄存器来读取接收到的8位数据。
11. 如果发生了溢出错误，可通过清零 RCSTA 寄存器的 CREN 位或清零可将 EUSART 复位的 SPEN 位清除错误。



注： 时序图说明SREN = 1且BRGH = 0时的同步主模式

图26-12：同步接收（主模式， SREN）

26.4.2 同步从模式

使用以下位将EUSART 配置为同步从操作：

- SYNC = 1
- CSRC = 0
- SREN = 0 （用于发送）； SREN = 1 （用于接收）
- CREN = 0 （用于发送）； CREN = 1 （用于接收）
- SPEN = 1

将TXSTA 寄存器的SYNC 位置1 将器件配置为同步操作。将TXSTA 寄存器的CSRC位清零将器件配置为从器件。将RCSTA寄存器的SREN 和CREN 位清零可确保器件处于发送模式，否则器件将被配置为接收模式。将RCSTA寄存器的SPEN 位置1 可使能EUSART。

26.4.2.1 EUSART 同步从发送

除了休眠模式以外，同步主模式和从模式的工作原理是相同的（见第26.4.1.3 节“ 同步主发送”）。
如果向TXREG 写入两个字，然后执行SLEEP 指令，则会发生以下事件：
1. 第一个字符将立即传送到TSR 寄存器并发送。
2. 第二个字将保留在TXREG 寄存器中。
3. TXIF 位不会被置1。
4. 第一个字符移出TSR 后，TXREG 寄存器会将第二个字符传送到TSR，此时TXIF 位将置1。
5. 如果PEIE 和TXIE 位均置1，则发生中断将器件从休眠唤醒，并执行下一条指令。如果GIE 位也置1，程序将调用中断服务程序。

26.4.2.2 同步从发送设置

1. 将SYNC 和SPEN 位置1 并清零CSRC 位。
2. 清零CK 引脚的ANSEL 位（如适用）。
3. 清零CREN 和SREN 位。
4. 如果需要中断，将PIEB1 寄存器的TXIE 位以及INTS 寄存器的GIE 和PEIE 位置1。
5. 如果需要9 位发送，将TX9 位置1。
6. 将TXEN 位置1 使能发送。
7. 如果选择了9位发送，将最高有效位插入TX9D位。
8. 将低8 位写入TXREG 寄存器，启动发送。

26.4.2.3 EUSART 同步从接收

除下列各项外，同步主模式和从模式的工作原理是相同的（第26.4.1.5 节“ 同步主接收”）：
• 休眠
• CREN 位始终置1，因此接收器从不空闲
• SREN 位在从模式下为“ 无关位”
进入休眠前，将CREN 位置1 可在休眠模式下接收一个字符。接收到该字后， RSR 寄存器将把数据发送到RCREG 寄存器。如果RCIE 中断允许位置1，产生的中断会将器件从休眠唤醒并执行下一条指令。如果GIE位也置1，程序将跳转到中断向量。

26.4.2.4 同步从接收设置

1. 将SYNC 和SPEN 位置1 并清零CSRC 位。
2. 清零CK 和DT 引脚的ADINS 位（如适用）。
3. 如果需要中断，将PIEB1 寄存器的RCIE 位以及INTS 寄存器的GIE 和PEIE 位置1。
4. 如果需要接收9 位数据，将RX9 位置1。
5. 将CREN 位置1 使能接收。
6. 接收完成时RCIF 位将被置1。如果RCIE 位已置1，则产生中断。
7. 如果使能了9 位模式，从RCSTA寄存器的RX9D位取出最高有效位。
8. 读取RCREG寄存器，从接收FIFO 取出低8 位。
9. 如果发生了溢出错误，可通过清零RCSTA寄存器的CREN 位或清零SPEN 位（该位将EUSART复位）清除错误。

26.5 休眠期间的 EUSART 操作

EUSART只有在同步从模式下，才会在休眠模式下保持工作状态。所有其他模式都需要系统时钟；因此，在休眠模式下无法产生运行发送或接收移位寄存器必需的信号。同步从模式使用外部产生的时钟运行发送和接收移位寄存器。

26.5.1 休眠期间的同步接收

要在休眠模式下接收，进入休眠模式前必须满足以下所有条件：

- RCSTA 和TXSTA 控制寄存器必须配置为同步从接收（见第26.4.2.4 节“ 同步从接收设置”）。
- 如果需要中断，将PIEB1 寄存器的RCIE 位以及INTS 寄存器的GIE 和PEIE 位置1。
- 必须通过读RCREG 清零RCIF 中断标志位，以卸载接收缓冲区中等待处理的任何字符。

进入休眠模式时，器件将准备好分别在RX/DT和TX/CK引脚上接收数据和时钟信号。数据字从外部器件随着时钟完全移入时，PIFB1 寄存器的RCIF 中断标志位将置1。从而将处理器从休眠模式唤醒。

从休眠状态唤醒时，将执行SLEEP指令后紧跟的指令。如果INTCON寄存器的全局中断允许（GIE）位也置1，将调用地址0004h处的中断服务程序。

26.5.2 休眠期间的同步发送

要在休眠模式下发送，进入休眠模式前必须满足以下所有条件：

- RCSTA 和TXSTA 控制寄存器必须配置为同步从发送（见第26.4.2.2 节“ 同步从发送设置”）。
- 必须通过将输出数据写入TXREG 来清零TXIF 中断标志位，从而填充TSR 和发送缓冲区。
- 如果需要中断，将PIEB1寄存器的TXIE位和INTS寄存器的PEIE 位置1。
- 必须将PIEB1寄存器的TXIE中断允许位和INTS寄存器的PEIE 中断允许位置1。

进入休眠模式时，器件将在TX/CK 引脚上接收时钟信号，在RX/DT 引脚上发送数据。TSR 中的数据字随着由外部器件提供的时钟完全移出后， TXREG 中等待的字节将传输到TSR，TXIF 标志位置1。从而将处理器从休眠模式唤醒。此时， TXREG 可接收其他字符进行发送，此操作将清零TXIF 标志位。

从休眠状态唤醒时，将执行 SLEEP 指令后紧跟的指令。如果全局中断允许（GIE）位也置 1，将调用地址 0004h 处的中断服务程序。

26.5.3 备用引脚位置

该模块具有以下I/O 引脚：通过使用备用引脚功能寄存器APFCON 可将I/O 引脚转移到其他位置。

26.6 寄存器说明

寄存器19EH：发送状态和控制寄存器（TXSTA）

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	**TX9**	**TXEN(1)**	**SYNC**	**SENDB**	**BRGH**	**TRMT**	**TX9D**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	x= 未知
-n=POR时的值	1= 置1	0= 清零	u= 不变

bit7 **CSRC**：时钟源选择位

异步模式：

无关位

同步模式：

1 = 主模式（时钟由内部BRG 产生）

0 = 从模式（时钟来自外部时钟源）

bit6 **TX9**：9 位发送使能位
1 = 选择9 位发送
0 = 选择8 位发送

bit5 **TXEN**：发送使能位(1)
1 = 使能发送
0 = 禁止发送

bit4 **SYNC**：EUSART 模式选择位
1 = 同步模式
0 = 异步模式

bit3 **SENDB**：发送间隔字符位
异步模式：
1 = 在下一次发送时发送同步间隔字符（完成后由硬件清零）
0 = 同步间隔字符发送完成
同步模式：
无关位

bit2 **BRGH**：高波特率选择位
异步模式：
1 = 高速
0 = 低速
同步模式：
在此模式下未使用

bit1 **TRMT**：发送移位寄存器状态位
1 = TSR 空
0 = TSR 满

bit0 **TX9D**: 发送数据的第9 位
可以是地址/ 数据位或奇偶校验位。

注 1： 在同步模式下， SREN/CREN 可改写TXEN。

寄存器19DH：接收状态和控制寄存器（RCSTA）

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-x/x
SPEN	**RX9**	**SREN**	**CREN**	**SDDEN**	**FERR**	**OERR**	**RX9D**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7 **SPEN**：串口使能位
1 = 使能串口（将RX/DT 和TX/CK 引脚配置为串口引脚）
0 = 禁止串口（保持在复位状态）

bit6 **RX9**：9 位接收使能位
1 = 选择9 位接收
0 = 选择8 位接收

bit5 **SREN**：单字节接收使能位
异步模式：
无关位
同步主模式：

1 = 使能单字节接收
0 = 禁止单字节接收
此位在接收完成后清零。
同步从模式
无关位

bit4 **CREN**：连续接收使能位
异步模式：
1 = 使能接收器
0 = 禁止接收器
同步模式：
1 = 使能连续接收，直到使能位CREN 清零（CREN 的优先级高于SREN）
0 = 禁止连续接收

bit3 **SDDEN**：地址检测使能位
9 位异步模式（RX9 = 1）：
1 = 当RSR<8> 置1 时，使能地址检测，允许中断并装入接收缓冲区
0 = 禁止地址检测，接收所有字节并且第9 位可作为奇偶校验位
8 位异步模式（RX9 = 0）：
无关位

bit2 **FERR**：帧错误位
1 = 帧错误（可以通过读RCREG 寄存器更新该位并接收下一个有效字节）
0 = 无帧错误

bit1 **OERR**：溢出错误位
1 = 溢出错误（可以通过清零CREN 位来清零该位）
0 = 无溢出错误

bit0 **RX9D:** 接收数据的第9 位
该位可以是地址/ 数据位或奇偶校验位，并且必须由用户固件计算得到。

寄存器19FH：波特率控制寄存器（BAUDCON）

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	**RCIDL**	—	**SCKP**	**BRG16**	—	**WUE**	**ABDEN**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7 **ABDOVF**：自动波特率检测上溢位
异步模式：
1 = 自动波特率定时器上溢
0 = 自动波特率定时器未上溢
同步模式：
无关位

bit6 **RCIDL**：接收空闲标志位
异步模式：
1 = 接收器空闲
0 = 已接收到启动位且接收器正在接收
同步模式：

无关位

bit5 未实现：读为0

bit4 **SCKP**：同步时钟极性选择位

异步模式：

1 = 发送反相数据至TX/CK 引脚

0 = 发送未反相数据至TX/CK 引脚

同步模式：

1 = 数据在时钟上升沿同步

0 = 数据在时钟下降沿同步

bit3 **BRG16**：16 位波特率发生器位

1 = 使用16 位波特率发生器

0 = 使用8 位波特率发生器

bit2 未实现：读为0

bit1 **WUE**：唤醒使能位

异步模式：

1 = 接收器正在等待下降沿。不会接收到任何字符，但RCIF将被置1。RCIF置1后WUE将被自动清零。

0 = 接收器正常工作

同步模式：

无关位

bit0 **ABDEN**：自动波特率检测使能位

异步模式：

1 = 使能自动波特率模式（完成自动波特率检测后清零）

0 = 禁止自动波特率模式

同步模式：

无关位

寄存器199H：EUSART接收数据寄存器（RCREG）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7-0 **RCREG<7:0>**：EUSART接收的8bit数据位

寄存器19AH：EUSART发送数据寄存器（TXREG）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7-0 **TXREG<7:0>**：EUSART要发送的8bit数据位

寄存器19BH：波特率BRG低8位计数寄存器（SPBRGL）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7-0 **SPBRGL<7:0>**：波特率BRG计数器低8位

寄存器19CH：波特率BRG高8位计数寄存器（SPBRGH）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	x = 未知
-n = POR时的值	1 = 置1	0 = 清零	u = 不变

bit7-0 **SPBRGH<7:0>**：波特率BRG计数器高8位

表 26-4：与 EUSART 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-00-00	01-00-00
INTS	GIE	PEIE	TMR0IE	INTE	PAIE	TMR0IF	INTF	PAIF	0000000x	0000000u
PIEB1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	00000000	00000000
PIFB1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	00000000	00000000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000000x	0000000x
SPBRGL	BRG<7:0>								00000000	00000000
SPBRGH	BRG<15:8>								00000000	00000000
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	--11 1111	--11 1111
TXREG	EUSART 发送数据寄存器								00000000	00000000
RCREG	EUSART 接收数据寄存器								00000000	00000000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	00000010	00000010

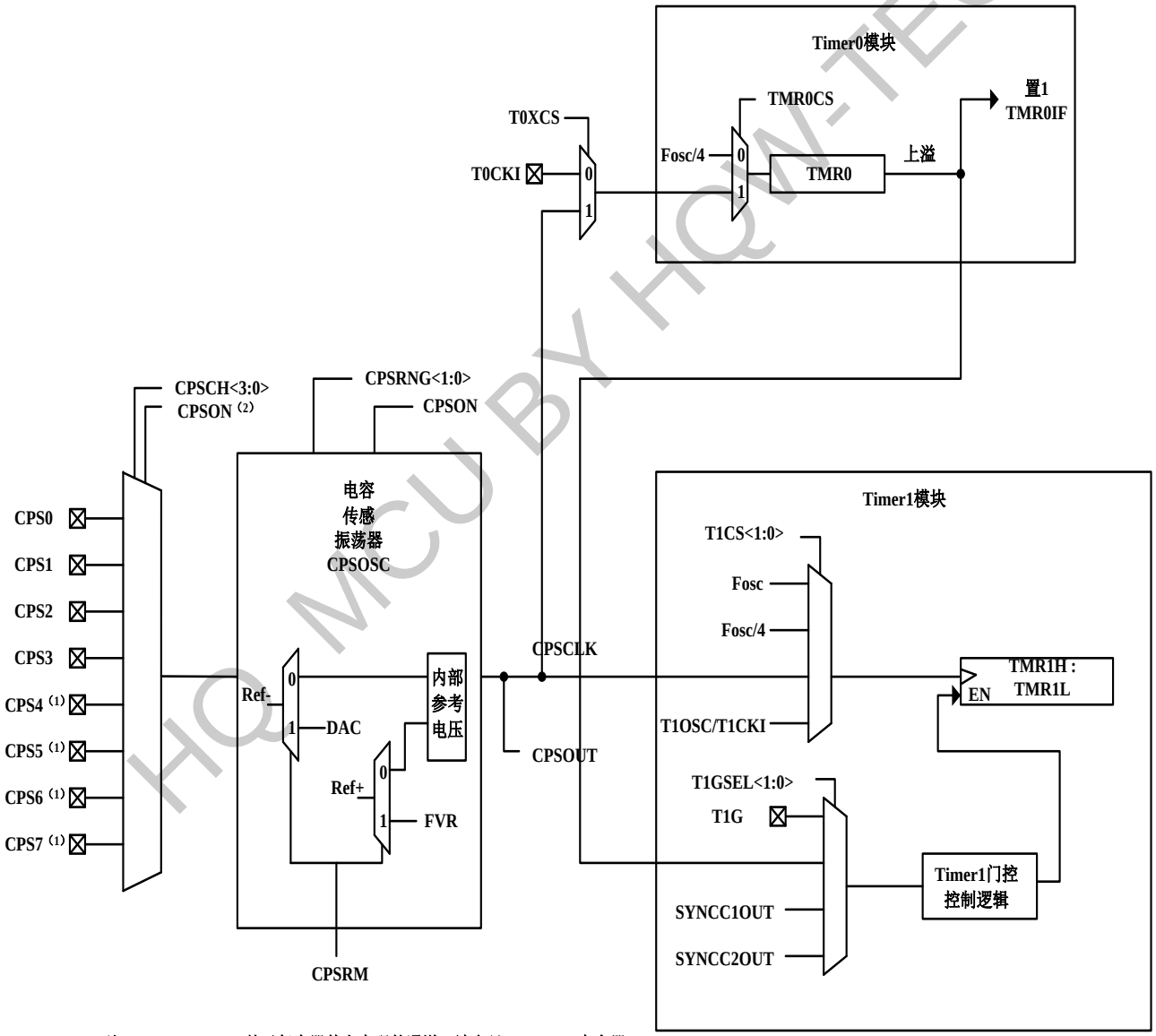
图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。 BOR 不使用阴影单元。

注 1： 其他（非上电）复位包括正常工作时的$\overline{MCLR}$复位和看门狗定时器复位。

27.0 电容传感（CPS）模块

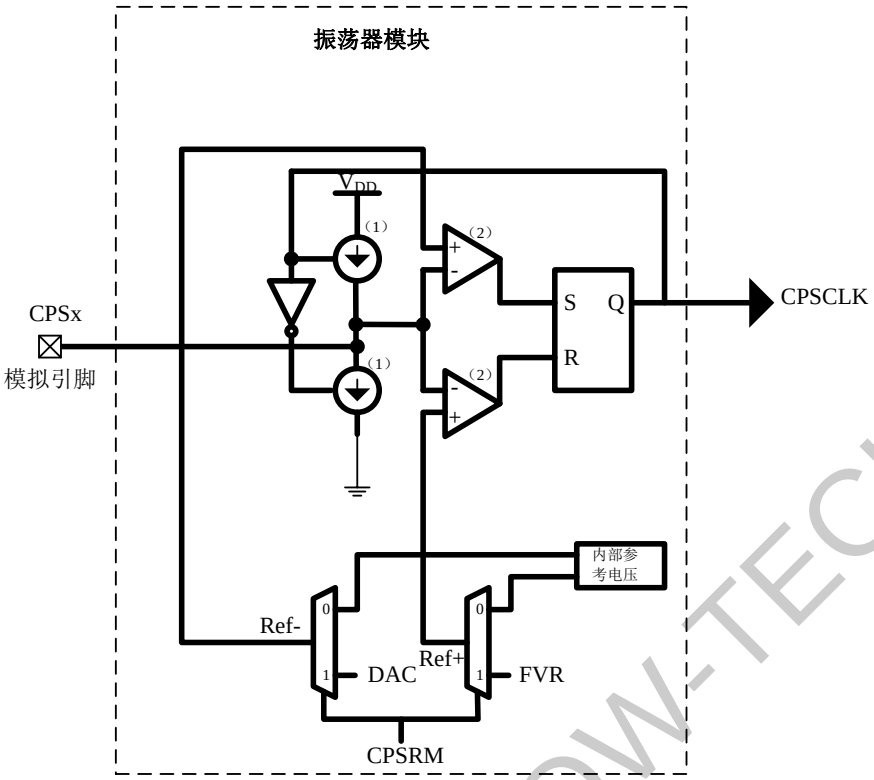
电容传感（CPS）模块无需机械接口即可与最终用户交互。在典型应用中， CPS 模块连接到印刷电路板（Printed Circuit Board， PCB）上的焊盘，焊盘与最终用户电气隔离。最终用户将手指放到PCB 焊盘上时，就加入了一个容性负载，引起CPS 模块中的频率漂移。CPS 模块需要软件和至少一个定时器资源，才能判断频率改变。该模块的主要特性包括：

- 用于监视多个输入的模拟MUX
- 电容传感振荡器
- 多种功耗模式
- 高功耗范围使用可变参考电压
- 多个定时器资源
- 软件控制
- 休眠期间的操作



注 1：关于每个器件上实现的通道，请参见CPSCON1寄存器。
2：如果COSON = 0，禁止电容传感，不选择任何通道

图 27-1：电容传感框图



注 1：图中未显示模块使能和功耗模式选择
2：比较器在噪声检测模式下保持有效

图 27-2：电容传感振荡器框图

27.1 模拟 MUX

MDT10F1822/1823最多可监视8 个输入（CPSCH<7:0>）。电容传感输入根据器件定义为CPS<7:0>。要判断是否发生频率改变，用户必须：

- 通过设置CPSCON1 寄存器的CPSCH 位选择适当的CPS 引脚。
- 将相应的ADINS 位置1。
- 将相应的CPIO 位置1。
- 运行软件算法

在模块使能时选择CPSx 引脚会导致电容传感振荡器位于CPSx 引脚。如果未能将相应的ADINS 和CPIO 位置1，会导致电容传感振荡器停止，从而使得频率读取错误。

27.2 电容传感振荡器

电容传感振荡器由恒定的拉电流和恒定的灌电流组成，产生三角波形。CPSCON0 寄存器的CPSOUT 位显示电容传感振荡器的状态，即它是灌电流还是拉电流。振荡器设计为驱动容性负载（单个PCB 焊盘），同时用作Timer0 或Timer1 的时钟源。振荡器有三种不同的电流设置，由CPSCON0 寄存器的CPSRNG<1:0> 定义。

不同的振荡器电流设置用于两种用途：

- 在固定时基下，最大化定时器的计数个数。
- 在频率发生改变时，最大化定时器的计数差值。

27.3 参考电压

电容传感振荡器使用参考电压来提供用于产生振荡的两个阈值电压。高阈值电压称为Ref+，低阈值电压称为Ref-。用户可以选择使用固定参考电压（这些参考电压位于电容传感振荡器内部），也可以使用可变参考电压（它们由固定参考电压（FVR）模块和数模转换器（DAC）模块提供）。使用固定参考电压时，VSS 电压决定低阈值电压（Ref-），而VDD 电压则决定高阈值电压（Ref+）。使用可变参考电压时，DAC 电压决定低阈值电压（Ref-），而FVR 电压则决定高阈值电压（Ref+）。使用这些参考源的优点是振荡频率可以在VDD 改变的情况下保持恒定。通过使用这些可变参考电压，可以获得不同的振荡频率。高参考电压降得越多，低参考电压升得越多，电容传感振荡器频率也就变得越高。参考电压之间的选择通过CPSCON0 寄存器的CPSRM位进行控制。该位置1 时将选择可变参考电压，该位清零时将选择固定参考电压。

27.4 功耗模式

电容传感振荡器可以工作于7 种不同的功耗模式。功耗模式分为两个范围：低功耗范围和高功耗范围。选择振荡器的低功耗范围时，将使用电容传感振荡器的内部固定参考电压。选择振荡器的高功耗范围时，将使用由FVR 和DAC 模块提供的可变参考电压。参考电压之间的选择通过CPSCON0 寄存器的CPSRM位进行控制。

在每个范围中有三种不同的功耗模式：低、中和高。电流消耗取决于所选择的范围和模式。在每个范围中选择功耗模式的实现方法是配置CPSCON0 寄存器中的CPSRNG <1:0> 位。关于适当的功耗模式选择，请参见表27-1。

最后剩下的一种模式是处于高功耗范围内的噪声检测模式。噪声检测模式的独特之处在于它会禁止模拟引脚上的灌电流和拉电流，但将振荡器电路的其余部分保持有效。这可以将模拟引脚上的振荡频率降为0，并且还可以极大地降低振荡器模块消耗的电流。当有噪声传入引脚时，振荡器将以由噪声决定的频率驱动。这会在比较器输出上产生可检测的信号，指示引脚上存在活动。图27-2 给出了与振荡器关联的电流源和比较器的更详细图示。

表 27-1： 功耗模式选择

CPSRM	范围	CPSRNG<1:0>	模式	标称电流
0	低	00	关闭	0.0μA
		01	低	0.1μA
		10	中	1.2μA
		11	高	18μA
1	高	00	噪声检测	0.0μA
		01	低	9μA
		10	中	30μA
		11	高	100μA

27.5 定时器资源

要测量电容传感振荡器的频率改变，需要固定时基。在固定时基期间，电容传感振荡器用作Timer0 或Timer1的时钟源。电容传感振荡器的频率等于定时器中的计数值除以固定时基周期。

27.6 固定时基

要测量电容传感振荡器的频率，需要固定时基。任何定时器资源或软件循环都可用于建立固定时基。产生固定时基的方法由最终用户决定。

注： 固定时基不能由讲电容传感器振荡器用作时钟源的定时器资源产生。

27.6.1 TIMER0

要选择Timer0 作为 CPS 模块的定时器资源：

- 将CPSCON0 寄存器的T0XCS 位置1。
- 将OPTION 寄存器的TMR0CS 位清零。

选择Timer0 作为定时器资源时，电容传感振荡器将作为Timer0 的时钟源。更多信息，请参见第20.0 节“Timer0 模块”。

27.6.2 TIMER1

要选择Timer1 作为CPS模块的定时器资源，将T1STA寄存器的TMR1CS<1:0> 设置为11。选择Timer1 作为定时器资源时，电容传感振荡器作为Timer1 的时钟源。由于Timer1 模块具有门控控制，用于频率测量的时基开发可以使用Timer0 上溢标志进行简化。建议将Timer0 上溢标志与Timer1 门控的翻转模式配合使用，用于开发CPS 模块软件部分所需的固定时基。

表 27-2： TIMER1 使能功能

TMR1ON	TMR1GE	Timer1 工作状态
0	0	关闭
0	1	关闭
1	0	开启
1	1	通过输入使能计数

27.7 软件控制

要判断电容传感振荡器的频率改变，需要CPS 模块的软件部分。这是通过以下步骤实现的：

- 设置固定时基以获取Timer0 或Timer1 上的计数。
- 确定电容传感振荡器的标称频率。
- 确定由于额外容性负载造成电容传感振荡器降低的频率。
- 设置频率阈值。

27.7.1 标称频率（无容性负载）

要确定电容传感振荡器的标称频率：

- 移除所选CPSx 引脚上的多余容性负载。
- 固定时基开始时将定时器资源清零。
- 固定时基结束时保存定时器资源中的值。

对于给定时基，定时器资源的值是电容传感振荡器的振荡次数。电容传感振荡器的频率等于定时器中的计数值除以固定时基周期。

27.7.2 降低的频率（额外的容性负载）

额外的容性负载会导致电容传感振荡器频率降低。要确定电容传感振荡器降低的频率：

- 在所选CPSx 引脚上添加典型的容性负载。
- 将相同的固定时基用作标称频率测量值。
- 固定时基开始时将定时器资源清零。
- 固定时基结束时保存定时器资源中的值。

定时器资源的值是带额外容性负载的电容传感振荡器的振荡次数。电容传感振荡器的频率等于定时器中的计数值除以固定时基周期。该频率应低于测量标称频率时获得的值。

27.7.3 频率阈值

频率阈值应置于电容传感振荡器标称频率值和降低的频率的中间。

27.8 休眠期间的操作

只要模块使能，电容传感振荡器就会持续运行，即使器件处于休眠状态亦然。为了让软件能判断是否发生频率改变，必须唤醒器件。但是，定时器资源采集计数时，无需唤醒器件。

注： Timer0 不能在休眠模式下工作，因此在休眠模式下不能用于电容传感测量。

27.9 寄存器说明

寄存器1EH：电容传感控制寄存器0（CPSCON0）

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R-0	R/W-0
CPSON	**CPSRM**	—	—	**CPSRNG1**	**CPSRNG0**	**CPSOUT**	**T0XCS**
bit7							bit0

图注：			
R= 可读位	W= 可写位	U= 未实现位，读为0	
-n=POR时的值	1= 置1	0= 清零	x= 未知

bit7 **CPSON**：CPS 模块使能位
1 = 使能CPS 模块
0 = 禁止CPS 模块

bit6 **CPSRM**：CPS 参考电压模式位
1 = CPS 模块处于高功耗范围。DAC 和FVR 提供振荡器参考电压。
0 = CPS 模块处于低功耗范围。使用内部振荡器参考电压。

bit5-4 **未实现：读为0**

bit3-2 **CPSRNG<1:0>**：电容传感电流范围位
如果CPSRM = 0 （低功耗范围）：
00 = 振荡器关闭
01 = 振荡器处于低功耗范围。充电/ 放电电流的标称值为0.1 μA
10 = 振荡器处于中等功耗范围。充电/ 放电电流的标称值为1.2 μA
11 = 振荡器处于高功耗范围。充电/ 放电电流的标称值为18 μA
如果CPSRM = 1 （高功耗范围）：
00 = 振荡器开启。噪声检测模式。不提供充电/ 放电电流。
01 = 振荡器处于低功耗范围。充电/ 放电电流的标称值为9 μA
10 = 振荡器处于中等功耗范围。充电/ 放电电流的标称值为30 μA
11 = 振荡器处于高功耗范围。充电/ 放电电流的标称值为100 μA

bit1 **CPSOUT**：电容传感振荡器状态位
1 = 振荡器在拉电流（流出引脚的电流）
0 = 振荡器在灌电流（流入引脚的电流）

bit0 **T0XCS**：Timer0 外部时钟源选择位
如果TMR0CS = 1：
T0XCS 位控制用哪个位于内核/Timer0 模块外部的时钟作为Timer0 的时钟源：
1 = Timer0 时钟源是电容传感振荡器
0 = Timer0 时钟源是T0CK 引脚

如果TMR0CS = 0：

Timer0 时钟源由内核/Timer0 模块控制，为FOSC/4

寄存器1FH：电容传感控制寄存器1（CPSCON1）

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	**CPSCH3**	**CPSCH2**	**CPSCH1**	**CPSCH0**
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit7-4 **未实现：读为0**

bit3-0 **CPSCH<3:0>**：电容传感通道选择位

如果CPSON = 0：

这些位为无关位。不选择任何通道。

如果CPSON = 1：

0000 = 通道0，（CPS0）
0001 = 通道1，（CPS1）
0010 = 通道2，（CPS2）
0011 = 通道3，（CPS3）
0100 = 通道4，（CPS4）
0101 = 通道5，（CPS5）
0110 = 通道6，（CPS6）
0111 = 通道7，（CPS7）
1000 = 保留。未使用。
•
•
•
1111 = 保留。未使用。

表 27-1：与 CPS 有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINSA	—	—	—	AN3	—	AN2	AN1	AN0	xxx1 x111	xxx1 x111
ADINSC	—	—	—	—	AN7	AN6	AN5	ANSC0	xxxx 1111	xxxx 1111
INTS	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	0000 000x	0000 000q
OPTION	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
CPSCON0	CPSON	CPSRM	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	00xx 0000	00xx 0000
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	00xx 0000	00xx 0000
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	xx11 x111	xx11 x111
CPIOC	—	—	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	xx11 1111	xx11 1111
T1STA	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	—	TMR1ON	0000 00x0	0000 00q0

图注： u = 不变， x = 未知， — = 未实现位， 读为 0， q = 取值视具体情况而定。

28.0 指令表

每条指令都是一个包含操作码和所有必需操作数的14 位字。操作码可以分为三大类。

- 面向字节的操作类指令
- 面向位的操作类指令
- 立即数和控制操作类指令

立即数和控制类指令字格式最为丰富。

表28-1 列出了汇编器可识别的指令。

除了以下指令（可能需要2 或3 个周期），所有指令都在单个指令周期内执行：

- 子程序指令需要两个周期（LCALL 和LCALLW）
- 中断或子程序返回指令需要两个周期（RET、RTIW 和RTFI）
- 程序跳转指令需要两个周期（LJUMP、BRA、BRW、BTSS、BTSC、DECRSZ 和INCSRZ）
- 当任意指令引用某个间接文件寄存器，并且文件选择寄存器指向程序存储器时，将需要使用一个额外的指令周期。

一个指令周期包含4个振荡器周期；振荡器频率为4 MHz时，得到的标称指令执行速率为1 MHz。

所有指令示例均使用格式“0xhh”来表示一个十六进制数，其中“h”表示一个十六进制数字。

表28-1： MDT10F1822/1823增强指令集

指令	功能	操作	影响状态位
NOP	空操作	无	无
CLRWT	清看门狗	0→WT	/TF, /PF
SLEEP	进入睡眠模式	0→WT, stop OSC	/TF, /PF
TMODE	将 W 寄存器值装入选项寄存器（81H）	W→81H	无
CPIO R	设置端口方向寄存器（1 输入，0 输出）	W→CPIO r	无
STWR R	将 W 寄存器内容送 R 寄存器	W→R	无
LDR R, t	读寄存器 R,结果保存在 R(t=1)或者 W(t=0)	R→t	Z
LDWI I	立即数送 W 寄存器	I→W	无
SWAPR R, t	交换寄存器 R 的高低四位，结果保存在 R(t=1)或者 W(t=0)中	[R(0~3) R(4~7)]→t	无
INCR R, t	递增寄存器 R,结果保存在 R(t=1)或者 W(t=0)中	R + 1→t	Z
INCRSZ R, t	递增寄存器 R,结果保存在 R(t=1)或者 W(t=0)中;如果结果等于 0 则跳过该指令接下来的指令	R + 1→t	无
ADDWR R, t	W 寄存器与 R 寄存器相加，结果保存在 R(t=1)或者 W(t=0)中	W + R→t	C, HC, Z
SUBWR R, t	R 寄存器减去 W 寄存器，结果保存在 R(t=1)或者 W(t=0)中	R A W→t (R+/W+1→t)	C, HC, Z
DECR R, t	递减寄存器 R，结果保存在 R(t=1)或者 W(t=0)中	R A 1→t	Z
DECRSZ R, t	递减寄存器 R，结果保存在 R(t=1)或者 W(t=0)中；如果结果等于 0 则跳过该指令接下来的指令	R A 1→t	无
ANDWR R, t	R 寄存器与 W 寄存器做“与”操作，结果保存在 R(t=1)或者 W(t=0)中	R ∩ W→t	Z
ANDWI I	W 寄存器与立即数 I 做“与”操作，结果保存到 W 寄存器中	I ∩ W→W	Z
IORWR R, t	R 寄存器与 W 寄存器做“或”操作，结果保存在 R(t=1)或者 W(t=0)中	R ∪ W→t	Z
IORWI I	W 寄存器与立即数 I 做“或”操作，结果保存在 R(t=1)或者 W(t=0)中	I ∪ W→W	Z
XORWR R, t	R 寄存器与 W 寄存器做“异或”操作，结果保存在 R(t=1)或者 W(t=0)中	R ⊕ W→t	Z

XORWI I	W 寄存器与立即数 I 做“异或”操作，结果保存在 R(t=1)或者 W(t=0)中	I ⊕ W→W	Z
COMR R, t	R 寄存器“取反”操作，结果保存在 R(t=1)或者 W(t=0)中	/R→t	Z
RRR R, t	R 寄存器循环“右移”操作，结果保存在 R(t=1)或者 W(t=0)中	R(n)→R(n-1), C→R(7), R(0)→C	C
RLR R, t	R 寄存器循环“左移”操作，结果保存在 R(t=1)或者 W(t=0)中	R(n)→r(n+1), C→R(0), R(7)→C	C
CLRW	W 寄存器清 0	0→W	Z
CLRR R	R 寄存器清 0	0→R	Z
BCR R, b	R 寄存器的第 b 位清 0	0→R(b)	无
BSR R, b	R 寄存器的第 b 位置 1	1→R(b)	无
BTSC R, b	如果 R 寄存器的第 b 位为 0，则跳过该指令接下来的指令	Skip if R(b)=0	无
BTSS R, b	如果 R 寄存器的第 b 位为 1，则跳过该指令接下来的指令	Skip if R(b)=1	无
LLCALL N	在整个 2K 区域内的调用指令	N→PC, PC+1→Stack	无
LJUMP N	在整个 2K 区域内的跳转指令	N→PC	无
RTIW I	带立即数从子程序返回	Stack→PC, I→W	无
ADDWI I	W 寄存器与立即数 I 相加，结果保存到 W 中	PC+1→PC,W+I→W	C,HC,Z
SUBWI I	立即数 I 减去 W 寄存器，结果保存到 W 寄存器中	I-W→W	C,HC,Z
RTFI	中断返回	Stack→PC,1→GIS	无
RET	从子程序返回	Stack→PC	无
ADDWRC R,t	W 与 F 相加（带进位）	W + R + C→t	C,HC,Z
ASRR R,t	算术右移(带符号位右移，符号位为 1 左边补 1，否则补 0)	R(n)→R(n-1),	C,Z
LSLR	逻辑左移（不考虑符号位，右边补 0）	R(n)→r(n+1)	C,Z
LSRR	逻辑右移（不考虑符号位，右边补 0）	R(n)→R(n-1)	C,Z
SUBWRB R,t	R 减去 W（带借位）	R -W - C→t	C,HC,Z
ADDWI I	立即数与 W 相加	W + I→W	C,HC,Z
LDLB I	将立即数传送到 BSR	I→BSR	
LDLP I	将立即数传送 PCLATH	I→PCLATH	
SUBWI I	立即数减去 W	I - W→W	C,HC,Z
BRA	相对跳转		
BRW	使用 W 进行相对跳转		
LCALLW	使用 W 调用子程序		
RST	软件器件复位		

注 1： 如果程序计数器（PC）被修改或条件测试结果为真，则该指令需要两个周期。第二个周期执行一条 NOP 指令。

2： 如果该指令寻址的是IAR 寄存器，并且相应MSR 的MSb 置1，则该指令将需要一个额外的指令周期。

29.0 电气特性

29.1 绝对极限参数

偏置电压下的环境温度.....-40°C 至+85°C
储存温度.....-65°C 至+150°C
VDD 引脚相对于 VSS 的电压.....-0.3V 至+6.5V
$\overline{MCLR}$引脚相对于 Vss 的电压.....-0.3V 至+9.5V
所有其他引脚相对于 VSS 的电压.....-0.3V 至(VDD+0.3V)
总功耗(1).....600mW
流出 VSS 引脚的最大电流.....95mA
流入 VDD 引脚的最大电流.....95mA
输入钳位电流，I_{IK}（V_I<0 或 V_I>VDD）.....±20mA
输出钳位电流，I_{OK}（V_o<0 或 V_o>VDD）.....±20mA
任一 I/O 引脚的最大输出灌电流.....25mA
任一 I/O 引脚的最大输出拉电流.....25mA
PORTA 和 PORTC（联合）最大灌电流.....90mA
PORTA 和 PORTC（联合）最大拉电流.....90mA

注 1：功耗计算公式为：$P_{DIS} = VDD \times \{ I_{DD} - \sum I_{OH} \} + \sum \{ (V_{DD} - V_{OH}) \times I_{OH} \} + \sum (V_{OL} \times I_{OL})$

注意： 如果运行条件超过了上述“绝对极限参数值”，即可能对器件造成永久性损坏。上述值仅为运行条件的极大值，我们不建议器件运行在该规范范围以外。器件长时间工作在绝对极限参数条件下，其稳定性可能受到影响。

29.2 直流电器特性

直流特性		标准工作条件 工作温度 -40℃≤Ta≤+85℃						
符号	特性	最小值	典型值 (1)	最大值	单位	条件		
VDD	电源电压	2.2		5.5	V			
VDR	RAM 数据保持电压 (2)	—	0.5*	—	V	器件处于休眠模式		
VPOR	Vdd 起始电压确保能够产生上电复位信号	—	Vss	—	V			
SVDD	Vdd 上升速率确保能够产生上电复位信号	0.05*	—	—	V/ms			
IDD	工作电流 (3)	—	420	—	uA	3.3V	1M	WDT En PED Dis IRC
			600				4M	
			810				8M	
			510			5V	1M	
			790				4M	
			1100				8M	
IPD	掉电流 (4)	—	1	—	uA	WDT Disable VDD=2.5V		
△IWDT	WDT 电流 (4)	—	0.5	—	uA	VDD=5V		
VIL	输入低电压	VSS	——	1.1	V	3V TLL		
		VSS	——	1.1		3V SCHMITT		
		VSS	——	1.6		5V TLL		
		VSS	——	1.6		5V SCHMITT		
VIH	输入高电压	1.1	——	VDD	V	3V TLL		
		1.9	——	VDD		3V SCHMITT		
		1.6	——	VDD		5V TLL		
		3.5	——	VDD		5V SCHMITT		
IOL	输出灌电流	—	24	—	mA	VOL=0.7V	3V	
		—	28	—		VOL=0.9V		
		—	35	—		VOL=0.7V	5V	
		—	42	—		VOL=0.9V		
IOH	输出拉电流	—	13	—	mA	VOH=1.7V	3V	
		—	10	—		VOH=2.1V		
		—	22	—		VOH=3.6V	5V	
		—	12	—		VOH=4.2V		
VBOR	低电压复位电压	1.9 -20%	1.9	1.9+20%	V	LOW		
		2.5 -20%	2.5	2.5 +20%		HIGH		
		—	—	—		—		
Rpu	上拉电阻	—	32	—	K	3V		
		—	18	—		5V		

注： “—”表示没有，或待定。
（1）典型栏中数据均为 25℃条件下值，此部分数据仅供参考。
（2）该电压是保证不丢失 RAM 数据的最小 VDD。
（3）工作电流主要随工作电压和频率而变化。其它因素，如总线负载、总线速率、内部代码执行模式和温度也会影响电流消耗。
（4）掉电电流是在器件休眠时，所有 I/O 引脚都处于高阻态并且连接到 Vdd 或 Vss 时测得。

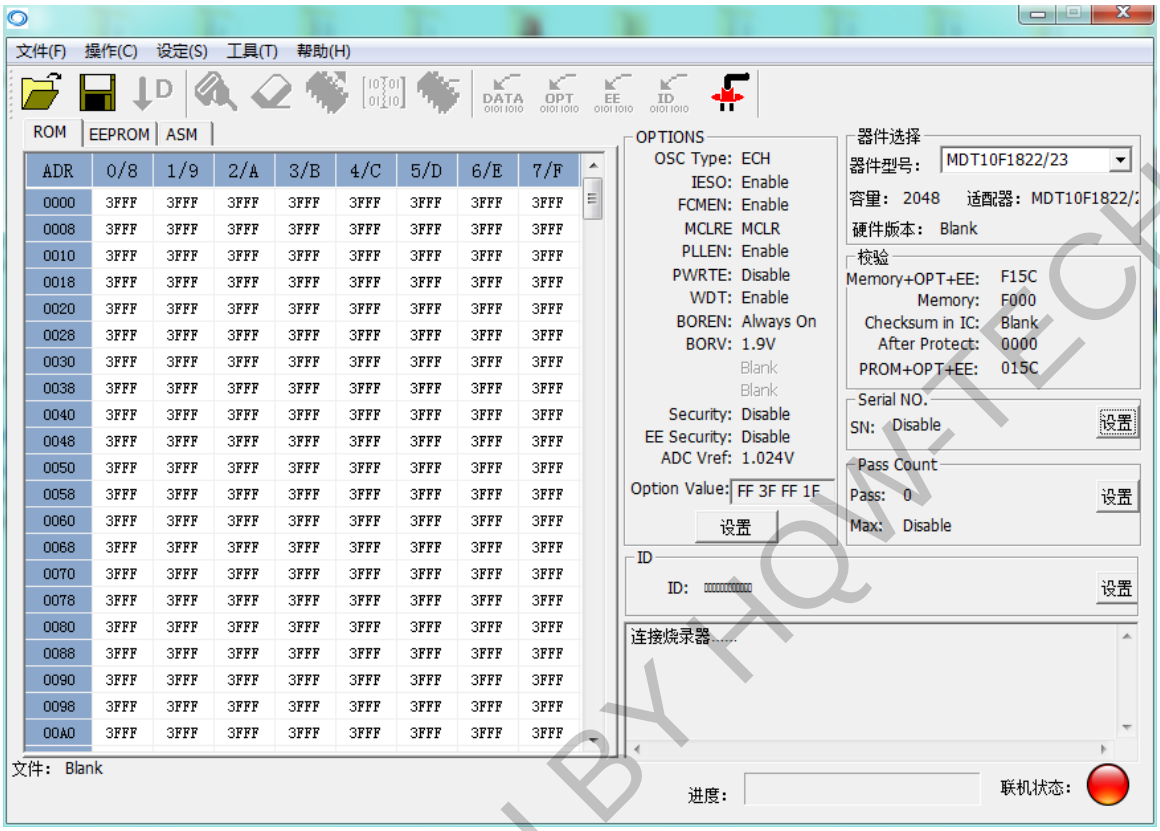
29.3 交流电气特性

交流特性		标准工作条件 工作温度-40℃≤Ta≤+85℃				
符号	特性	最小值	典型值 (1)	最大值	单位	条件
FOSC	系统时钟	0	8M	16M	Hz	5V
F_{HIRC}	内部高速时钟	——	4M	16M	Hz	5V
F_{LIRC}	内部低速时钟	——	31K	——	Hz	5V
T_{INT}	中断脉冲		2	3	Tins	指令周期
T_{SST}	系统启动时间 （上电复位）		13	20	ms	5V
	系统启动时间 （由 SLEEP 模式唤醒，Fsys 在 SLEEP 模式下关闭）		1024		Tsys	5V
						5V
	系统启动时间(由 SLEEP 模式唤醒,主要在 SLEEP 模式下开启）		1024		Tsys	5V
T_{RSTD}	系统复位延迟时间 （上电复位）		13	20	ms	5V
	系统复位时间 （WDT 正常复位）		77	84	ms	5V

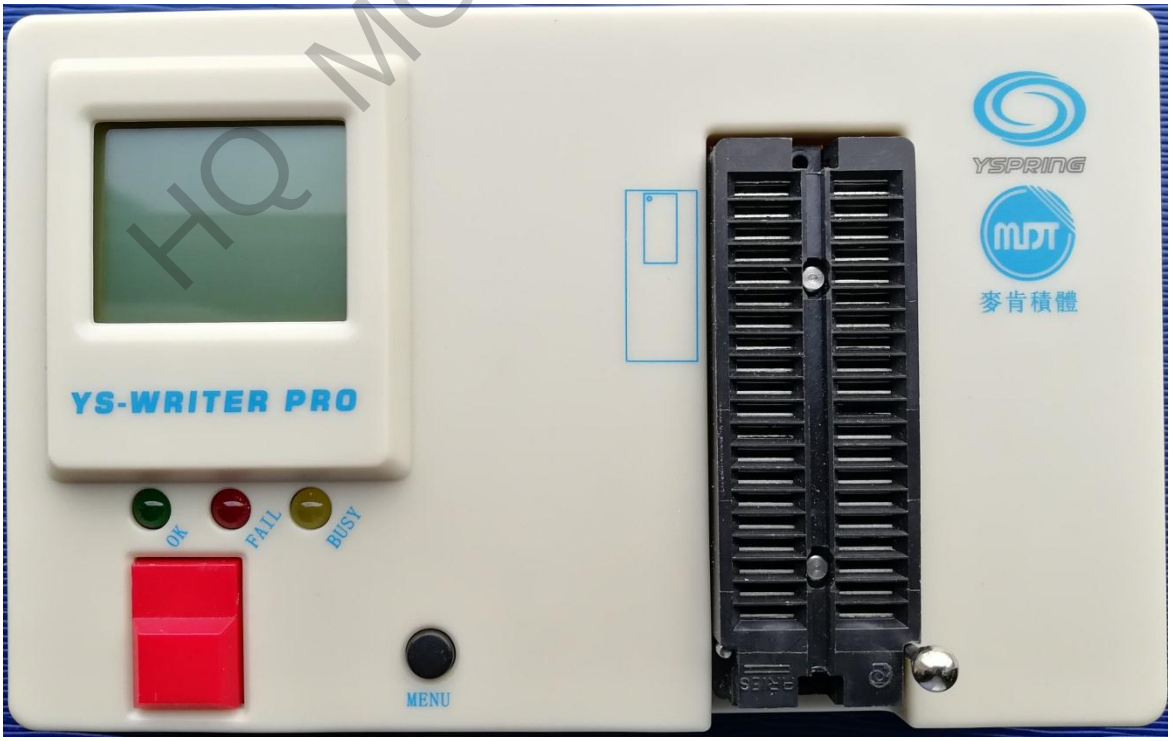
30.0 开发支持

30.1 烧录信息

30.1.1 烧录软件：YSpringPro



30.1.2 烧录器：YS-Writer

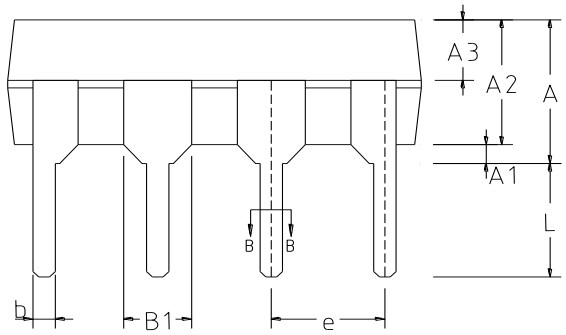


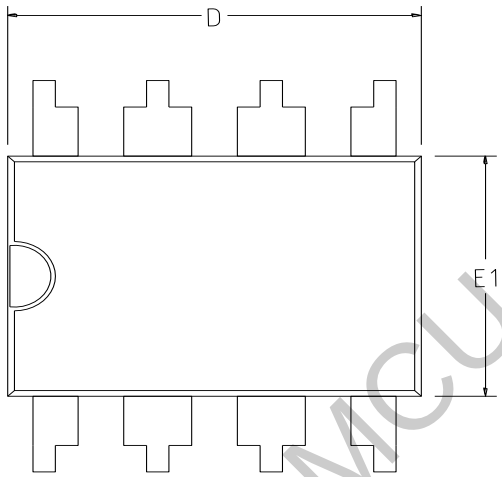
烧录使用引脚：VDD、VSS、VPP、PA0、PA1、PA2

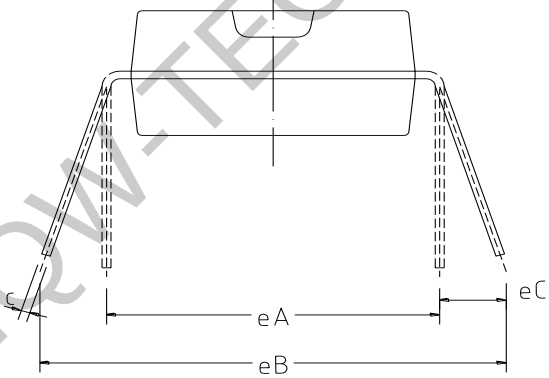
注：PA0 为烧录数据线，PA1 为烧录时钟线，PA2 为烧录 Busy 信号线

31.0 封装信息

31.1 P-DIP 8 PIN

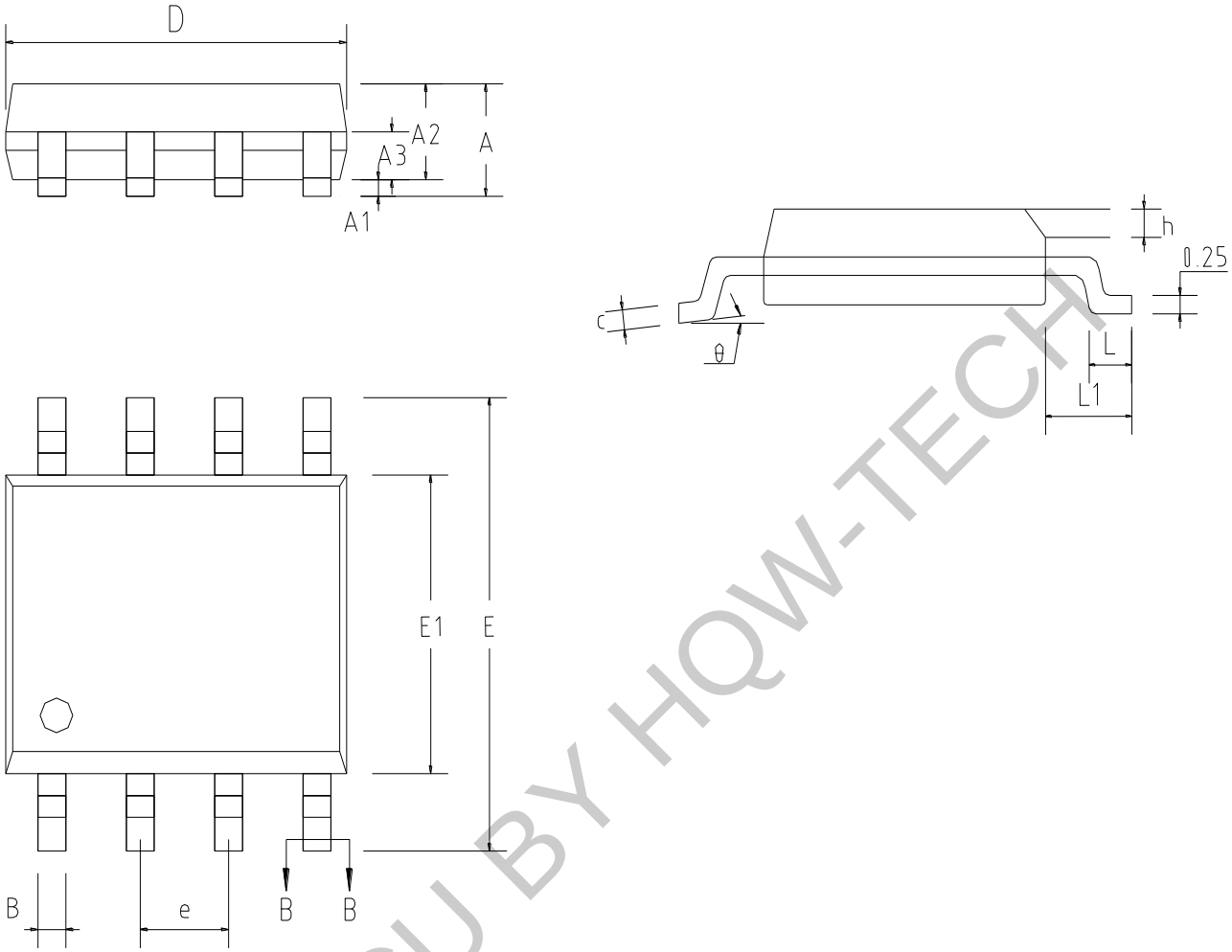






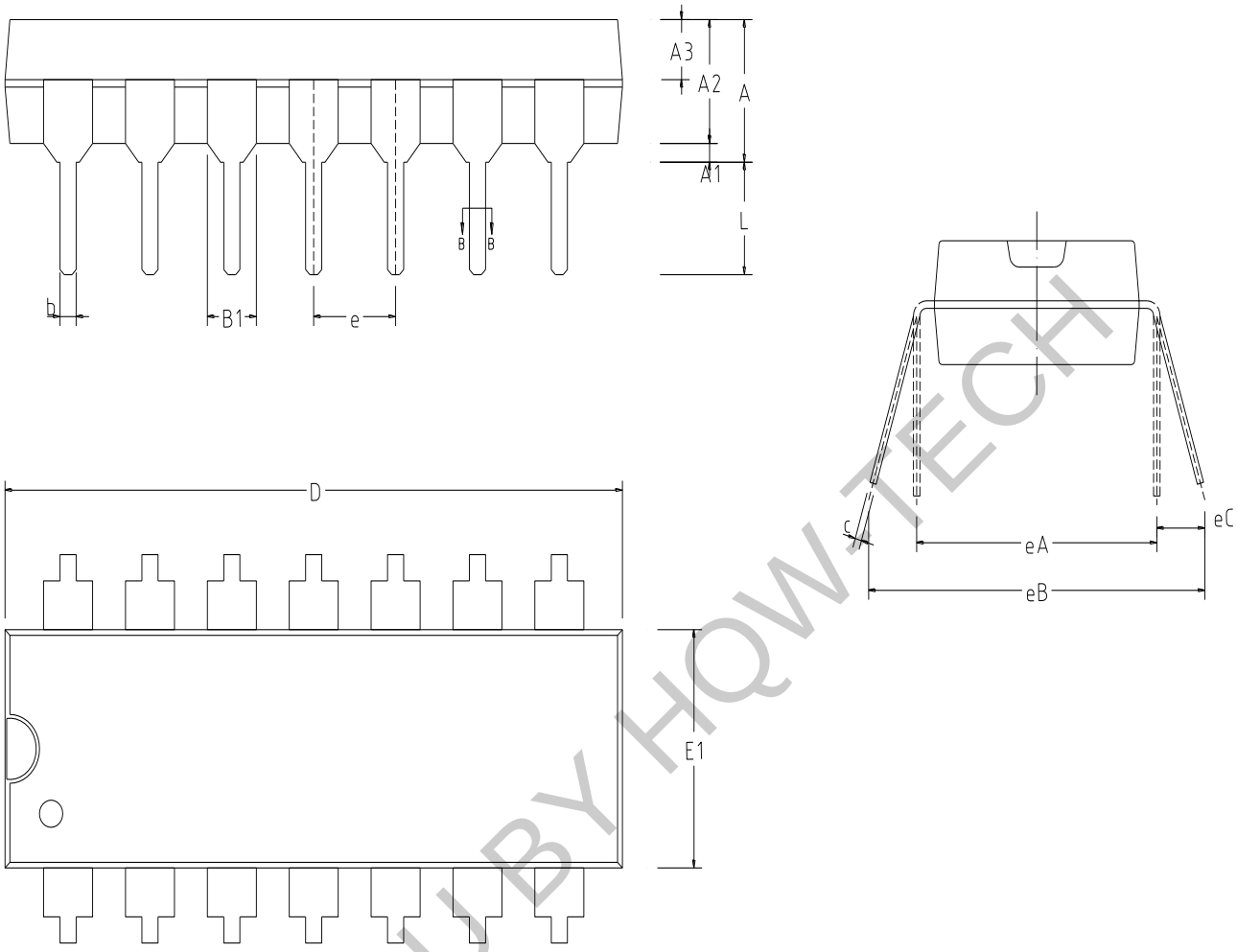
SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	3.60	3.80	4.00	E1	6.15	6.35	6.55
A1	0.51	—	—	e	2.54BSC		
A2	3.00	3.30	3.40	eA	7.62BSC		
A3	1.55	1.60	1.65	eB	7.62	—	9.30
B1	1.52BSC			eC	0	—	0.84
D	9.05	9.25	9.45	L	3.00	—	—
L/F 载体尺寸(mil)	80*80						

31.2 SOP 8 PIN



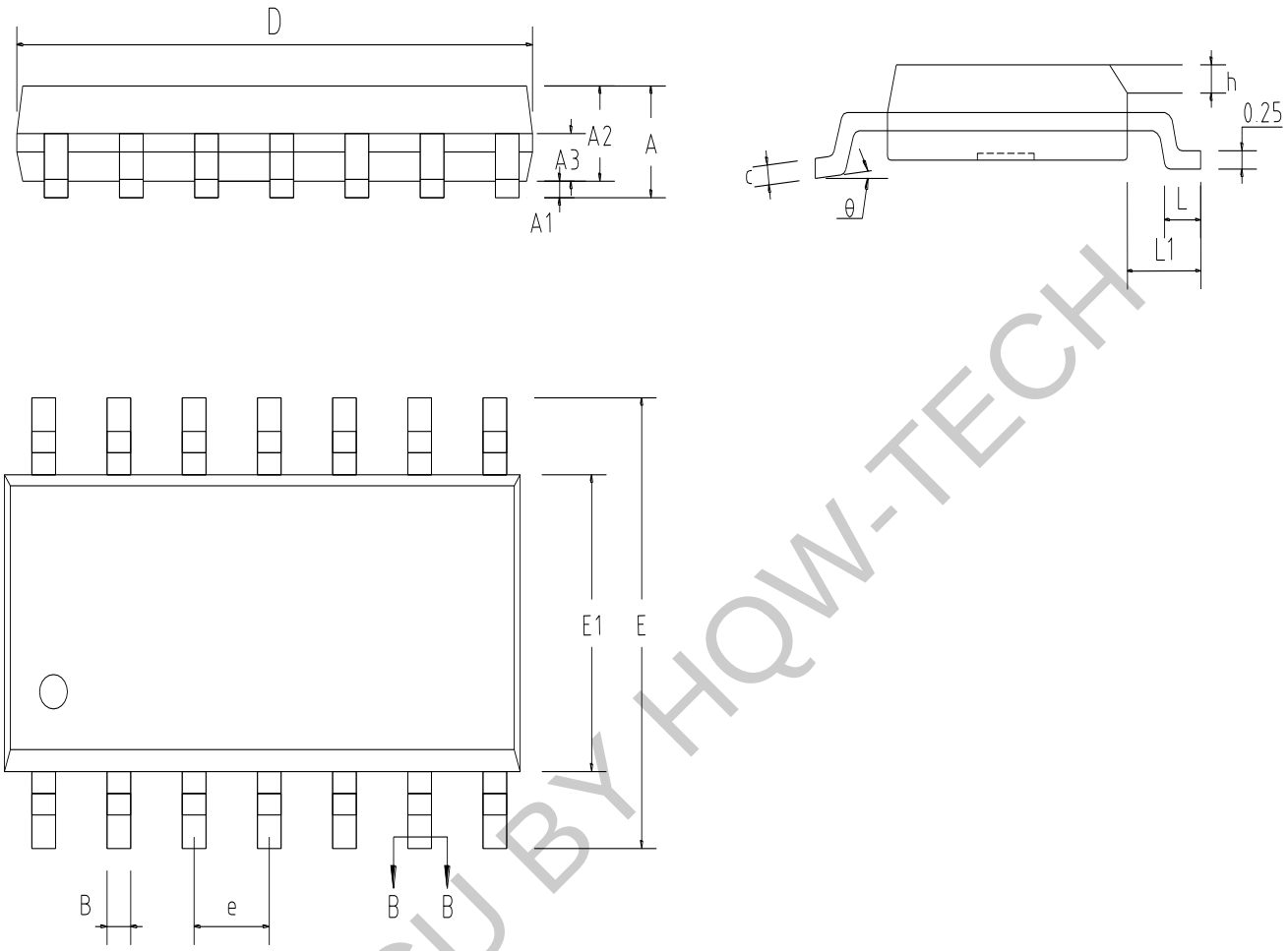
SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.70	3.90	4.10
A1	0.10	—	0.225	e	1.27BSC		
A2	1.30	1.40	1.50	h	0.25	—	0.50
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	4.70	4.90	5.10	L1	1.05 BSC		
E	5.80	6.00	6.20	θ	0	—	8°
L/F 载体尺寸(mil)	80*80	90*90	95*130	——			

31.3 P-DIP 14 PIN



SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	3.60	3.80	4.00	E1	6.15	6.35	6.55
A1	0.51	—	—	e	2.54BSC		
A2	3.20	3.30	3.40	eA	7.62BSC		
A3	1.47	1.52	1.57	eB	7.62	—	9.30
B1	1.52BSC			eC	0	—	0.84
D	18.90	19.10	19.30	L	3.00	—	—

31.4 SOP 14 PIN



SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.70	3.90	4.10
A1	0.10	—	0.225	e	1.27BSC		
A2	1.30	1.40	1.50	h	0.25	—	0.50
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	8.45	8.65	8.85	L1	1.05 BSC		
E	5.80	6.00	6.20	θ	0	—	8°
L/F 载体尺寸(mil)	70*70 90*110	98*150	100.4*210	——			

扫一扫 关注公众号

扫一扫 关注阿里旺铺

联系我们
深圳市华琦微科技有限公司
SHENZHEN HUAQIWEI TECHNOLOGY CO.,LTD
WEB:www.hqwtech.com
TEL: 0755-88603251
FAX: 0755-88609209
ADDRESS: 深圳市福田区深南中路嘉汇新城汇商中心2727