YS83B08/YS84B08/YS84C12

数据手册（版本 V1.2）

8 位 MTP
CMOS 单片机

1.0 器件概述

♦ **存储器：**
MTP 空间：(4K+32)*14 位。可经受 1000 次写操作。
SRAM 空间：256+128(程序可选)字节。

♦ **8 级深硬件堆栈**

♦ **I/O 引脚配置**
- 26 个双向输入/输出 IO 口。
- PA、PB、PD 端口独立的可编程弱上拉。
- PC 端口独立的可编程弱上、下拉。
- PA、PB 端口引脚电平变化中断唤醒。
- 高灌（TYP：30mA）/高拉（TYP：20mA）电流能力(PA3 除外)，可直接驱动 LED。

♦ **定时器**
Timer0：带 8 位可编程预分频器的 8 位定时器/计数器。
Timer1：- 带有预分频器的 16 位定时器/计数器。
- 外部 Timer1 门控（计数使能）。
- 如果选择了 INTOSC 模式，或者在 LP 模式下可选择使用 OSC1 和 OSC2 作为 Timer1 的振荡器。

♦ **2 路 PWM 模块**
-2 路 10bit 高分辨率 PWM 输出。

♦ **中断源**
- Timer0、Timer1 中断。
- PWM 中断。
- PA、PB 电平变化中断。
- 触摸中断。
- I2C 中断。
- 两路外部中断（INT0、INT1）。
- AD 转换完成中断。

♦ **8 通道 12 位 ADC**
最多 8 个模拟输入通道。ADC 参考电压可用软件选择为内部或外部参考。当选择内部参考时，可通过配置寄存器选择片内 1.024V、2.048V 或者 VDD 作为参考电压。

♦ **16 通道 Touch KEY**
可通过软件单独对每路调节灵敏度。

♦ **双时钟系统**
外部高速时钟：高达 20MHz。
内部高速时钟：16MHz RC
Fcpu 支持 16Mhz（**VDD 需大于 4.5V**）、8MHz、4MHz、2MHZ、1MHZ、500KHz、250KHz。
内部低速时钟：RC 振荡器 31KHz。

♦ **高性能的 RISCCPU**
仅需学习 35 条指令。
除跳转指令外的所有指令都是单周期的。
直接、间接和相对寻址模式。

♦ **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 250kHz 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 LVR 低电压侦测选择(侦测电压有 1.9V、2.5V、3.8V 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）。
- 带上拉的主复位，可复用为输入引脚。
- 可编程代码保护。

♦ **低功耗特性**
待机电流：
- 电压为 5V 时，典型值 1uA（无触摸）。

工作电流：
- 频率为 1MHz、电压为 3.3V 时，典型值为 600uA。
- 频率为 31kHz、电压为 3.3V 时，典型值为 400uA。

♦ **通过两个引脚(PA0，PA1)实现在线串行编程和调试**

注：下表根据不同封装用“/”分隔列出资源数量

器件	ROM	RAM	I/O	12 位 A/D 转换器	Timer	PWM	Touch	封装
YS83B08	4096	256+128	14	—	2	—	11	SOP16
YS84B08	4096	256+128	14/18/22	2/4/8	2	1/2/2	9/9/12	SOP16/SOP20(SSOP20)/SOP24(SSOP24)
YS84C12	4096	256+128	22/26	6/8	2	2/2	12/16	SOP24(SSOP24)/SOP28(SSOP28)

目录

1.0 器件概述 1

1.1 系统结构图 5
1.2 封装脚位图 6
1.3 引脚说明 8

2.0 存储器构成 10

2.1 程序存储器构成 10
2.2 数据存储器构成 10
2.3 PCL 和 PCLATH 20
2.4 间接寻址、IAR 和 MSR 寄存器 21
2.5 看门狗定时器（WDT） 22

3.0 复位 25

3.1 概述 25
3.2 上电复位 26
3.3 MCLR 26
3.4 上电延时定时器（PWRT） 26
3.5 欠压复位 27
3.6 延时时序 28
3.7 电源控制（PSTA）寄存器 29

4.0 系统时钟 30

4.1 概述 30
4.2 振荡器控制 31
4.3 时钟源模式 31
4.4 外部时钟模式 32
4.5 内部时钟模式 34
4.6 时钟切换 36
4.7 掉电模式（休眠） 37
4.8 故障保护时钟监控器 38
4.9 内部快时钟调频功能 40

5.0 中断 41

5.1 中断特殊功能寄存器 43
5.2 外部中断 46
5.3 TIMER0 中断 46
5.4 PORTA、PORTB 电平变化中断 46
5.5 A/D 中断 46
5.6 中断的现场保护 46

6.0 I/O 端口 48

6.1 PORTA 和 CPIOA 寄存器 48
6.2 PORTB 和 CPIOB 寄存器 49
6.3 PORTC 和 CPIOC 寄存器 50
6.4 PORTD 和 CPIOD 寄存器 51

6.5 其它引脚功能 51
6.6 PORTA 引脚说明和引脚图 55
6.7 PORTB 引脚说明和引脚图 62
6.8 PORTC 引脚说明和引脚图 63
6.9 PORTD 引脚说明和引脚图 64

7.0 定时器 70

7.1 TIMER0 模块 70
7.2 带门控的 TIMER1 模块 72
7.3 PWM 模块 77

8.0 模拟数字转换器（ADC）模块 83

8.1 ADC 框图 83
8.2 ADC 的配置 84
8.3 ADC 的工作过程 86
8.4 ADC 寄存器定义 87

9.0 I2C 模块 91

9.1 I2C 概述 92
9.2 I2C 模式操作 94
9.3 I2C 从模式操作 96
9.4 I2C 主模式 111
9.5 波特率发生器 122
9.6 寄存器定义 123

10.0 电容触摸模块 128

10.1 原理说明 128
10.2 配置说明 128
10.3 电容感应数据区 128
10.4 寄存器说明 129

11.0 配置寄存器 134

11.1 CONFIG OPTION 134

12.0 指令表 136

13.0 电气特性 137

13.1 绝对极限参数 137
13.2 直流电气特性 138
13.3 交流电气特性 139

14.0 开发支持 140

14.1 仿真信息 140
14.2 烧录信息 141

15.0 封装信息 143

15.1 SOP16 143
15.2 SOP20 144

15.3 SSOP20 145
15.4 SOP 24 146
15.5 SSOP24 147
15.6 SOP 28 148
15.7 SSOP 28 149

16.0 汇春知识产权政策 150

16.1 专利权 150
16.2 著作权 150

1.1 系统结构图

配置
INT
13
程序计数器
数据总线
8
MTP
4K × 14
程序
存储器
8级深堆栈
（12位）
RAM
384字节
文件
寄存器
程序
总线
14
指令寄存器
RAM地址
9
地址MUX
直接寻址
7
8
间接寻址
FSR寄存器
状态寄存器
8
MUX
3
ALU
指令译码
和控制
上电延时
定时器
振荡器
起振定时器
上电
复位
看门狗
定时器
欠压
复位
8
W寄存器
时序发生
OSC1/CLKIN
OSC2/CLKOUT
内部
振荡电路
MCLR
VDD
VSS
PORTA
PA0
PA1
PA2
PA3
PA4
PA5
PORTB
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PORTC
PC0
PC1
PC2
PC3
PORTD
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
T1G
T1CK1
T0CK1
Timer0
Timer1
I2C
CS
PWM
模数转换器
VREF
AN0AN1AN2AN3
AN4 AN5AN6 AN7

1.2 封装脚位图

1.2.1 16 引脚图（SOP16）

YS83B08 SOP16

YS84B08 SOP16

注：PIN12 为 PA5 与 PD7（内部短接）共用引脚，编程时需特别注意信号处理

1.2.2 20 引脚图（SOP20、SSOP20）

YS84B08 SOP20/SSOP20

1.2.3 24 引脚图（SOP24、 SSOP24）

YS84B08 SOP24/SSOP24

YS84C12 SOP24/SSOP24

1.2.4 28 引脚图（SOP28、SSOP28）

YS84C12 SOP28/SSOP28

1.3 引脚说明

名称	功能	输入类型	输出类型	说明
PA0/SCK/ULPWU/SDA	PA0	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	ISCK	ST	—	串行编程和调试时钟输入，
	ULPWU	AN	—	超低功耗唤醒输入
	SDA	ST	NMOS	I2C数据
PA1/SDA/INT0	PA1	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	ISDA	ST/TTL	CMOS	串行编程和调试数据输入输出，
	INT0	ST	—	外部中断
PA2/T0CKI/SCL	PA2	ST	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	T0CKI	ST	—	Timer0时钟输入
	SCL	ST	CMOS	I2C时钟
PA3/$\overline{MCLR}$/VPP	PA3	TTL	Open Drain	具有可编程下拉和电平变化中断的PORTA I/O
	$\overline{MCLR}$	ST	—	带有内部上拉的主复位
	VPP	HV	—	编程电压(9.5V)
PA4/AN3/$\overline{T1G}$/OSC2/CLKOUT	PA4	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	$\overline{T1G}$	ST	—	Timer1门控（计数使能）
	OSC2	—	XTAL	晶振/谐振器
	CLKOUT	—	CMOS	Fosc/4输出
PA5/T1CKI/OSC1/CLKIN	PA5	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	T1CKI	ST	—	Timer1时钟
	OSC1	XTAL	—	晶振/谐振器
	CLKIN	ST	—	外部时钟输入/RC振荡器连接
PB0/CCH0	PB0	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH0	NSI	—	触摸按键通道0输入
PB1/CCH1	PB1	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH1	NSI	—	触摸按键通道1输入

PB2/CCH2	PB2	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH2	NSI	—	触摸按键通道2输入
PB3/CCH3	PB3	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH3	NSI	—	触摸按键通道3输入
PB4/CCH4	PB4	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH4	NSI	—	触摸按键通道4输入
PB5/CCH5	PB5	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH5	NSI	—	触摸按键通道5输入
PB6/CCH6	PB6	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH6	NSI	—	触摸按键通道6输入
PB7/CCH7	PB7	TTL	CMOS	具有可编程上拉和电平变化中断的PORTB I/O
	CCH7	NSI	—	触摸按键通道7输入
PC0/CCH8	PC0	TTL	CMOS	具有可编程上、下拉的PORTC I/O
	CCH8	NSI	—	触摸按键通道8输入
PC1/CCH9	PC1	TTL	CMOS	具有可编程上、下拉的PORTC I/O
	CCH9	NSI	—	触摸按键通道9输入
PC2/CCH10	PC2	TTL	CMOS	具有可编程上、下拉的PORTC I/O
	CCH10	NSI	—	触摸按键通道10输入
PC3/CCH11	PC3	TTL	CMOS	具有可编程上、下拉的PORTC I/O
	CCH11	NSI	—	触摸按键通道11输入
PD0/CCH12/AN0/VREF	PD0	TTL	CMOS	具有可编程上拉的PORTD I/O
	CCH12	NSI	—	触摸按键通道12输入
	AN0	AN	—	A/D通道0输入
	VREF	AN	—	A/D外部参考电压
PD1/CCH13/AN1	PD1	TTL	CMOS	具有可编程上拉的PORTD I/O
	CCH13	NSI	—	触摸按键通道13输入
	AN1	AN	—	A/D通道1输入
PD2/CCH14/AN2	PD2	TTL	CMOS	具有可编程上拉的PORTD I/O
	CCH14	NSI	—	触摸按键通道14输入
	AN2	AN	—	A/D通道2输入
PD3/CCH15/AN3	PD3	TTL	CMOS	具有可编程上拉的PORTD I/O
	CCH15	NSI	—	触摸按键通道15输入
	AN3	AN	—	A/D通道3输入
PD4/AN4	PD4	TTL	CMOS	具有可编程上拉的PORTD I/O
	AN4	AN	—	A/D通道4输入
PD5/AN5/PWM1	PD5	TTL	CMOS	具有可编程上拉的PORTD I/O
	AN5	AN	—	A/D通道5输入
	PWM1	—	CMOS	PWM1输出
PD6/AN6/INT1	PD6	TTL	CMOS	具有可编程上拉的PORTD I/O
	AN6	AN	—	A/D通道6输入
	INT1	ST	—	外部中断
PD7/AN7/PWM0	PD7	TTL	CMOS	具有可编程上拉的PORTD I/O
	AN7	AN	—	A/D通道7输入
	PWM0	—	CMOS	PWM0输出
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
ST = 带 CMOS 电平的施密特触发器输入　TTL = TLL 兼容输出　XTAL = 晶振
NSI= 无标准输入

2.0 存储器构成

2.1 程序存储器构成

YS83B08/YS84B08/YS84C12 在物理上实现了 4K×14（0000h-0FFFh）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 4K×14 存储空间。复位向量地址为 0000h，中断向量地址为 0004h。

图 2-1

2.1.1 复位向量（0000H）

具有一个字长的系统复位向量（0000H）。

- 上电复位
- 看门狗复位
- 外部复位
- 欠压复位

发生上述任一种复位后，程序将从 0000H 处重新开始执行，系统寄存器也都将恢复为默认值。根据 STATUS 寄存器中的 TF 和 PF 标志位的内容可以判断系统复位方式。

2.1.2 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。

2.2 数据存储器构成

数据存储器被分为 4 个存储区（bank），其中包含通用寄存器（GeneralPurposeRegister，GPR）和特殊

功能寄存器（SpecialFunctionRegister，SFR）特殊功能寄存器位于每个存储区开头的 32 个单元。通用寄存器位于 Bank0、Bank1、Bank2 的 20h-6Fh 中，70H-7FH 是公用存储区域， Bank1、Bank2 和 Bank3 中的寄存器单元 F0h-FFh 指向 Bank0 中的地址单元 70h-7Fh。Bank3 中地址 1A0h 到 1DFh 和扩展存储器中地址 3A0h 到 3DFh 的 128Byte 是触摸模块专用存储区域，在没有使用触摸模块时，也可以用作通用寄存器。所有其他的 RAM 均未实现，读取它们时将返回 0。

STATUS 寄存器的 RP1、RP0 位为存储区选择位。

00 —> 选定 Bank0

01 —> 选定 Bank1

10 —> 选定 Bank2

11 —> 选定 Bank3

2.2.1 通用文件寄存器

在 YS83B08/YS84B08/YS84C12 中通用寄存器是按 128x8 的形式实现的。可直接访问每个寄存器或通过文件选择寄存器（FileSelectRegister，MSR）间接访问每个存储器。

2.2.2 特殊功能寄存器

特殊功能寄存器是 CPU 和外设模块用来控制所需的器件操作的寄存器，这些寄存器都实现为静态 RAM。

特殊功能寄存器可分为两类：内核与外设。本章仅讲述与“内核”有关的特殊功能寄存器。那些与外设功能部件的操作有关的特殊功能寄存器将在相应的外设功能部件章节中讲述。

寄存器	地址	寄存器	地址	寄存器	地址	寄存器	地址
IAR	00h	IAR	80h	IAR	100h	IAR	180h
TMR0	01h	OPT_REG	81h		101h		181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
MSR	04h	MSR	84h	MSR	104h	MSR	184h
PORTA	05h	CPIOA	85h		105h		185h
PORTB	06h	CPIOB	86h		106h		186h
PORTC	07h	CPIOC	87h		107h		187h
PORTD	08h	CPIOD	88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTS	0Bh	INTS	8Bh	INTS	10Bh	INTS	18Bh
PIFB1	0Ch		8Ch		10Ch		18Ch
PIFB2	0Dh		8Dh		10Dh		18Dh
TMR1L	0Eh	PSTA	8Eh		10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh		10Fh		18Fh
T1STA	10h		90h		110h	CSCTL	190h
SSP1STAT	11h	ADINS	91h		111h	THCTL	191h
SSP1CON1	12h	IRCTRIML	92h	PRL0L	112h	PWM0L	192h
SSP1CON2	13h	IRCTRIMH	93h	PRL1L	113h	PWM1L	193h
SSP1CON3	14h		94h		114h	CHRLTH	194h
SSP1MSK	15h	PAPHR	95h		115h	CHRLTL	195h
SSP1ADD	16h	PBPHR	96h		116h	CHSELH	196h
SSP1BUF	17h	PCPHR	97h	T1PRLL	117h	CHSELL	197h
WDTCON	18h	PDPHR	98h	ET1PRL	118h	EPR0	198h
PIFB3	19h	PAINTR	99h		119h	EPR1	199h
PWMCON	1Ah	PBINTR	9Ah		11Ah	T1PRLH	19Ah
	1Bh		9Bh		11Bh	CSTMRH	19Bh
	1Ch		9Ch		11Ch	CSTMRL	19Ch
	1Dh		9Dh		11Dh	ADS0	19Dh
ADRESH	1Eh	ADRESL	9Eh	ADC0CN	11Eh	CSANA0	19Eh
	1Fh	ADC0TH	9Fh		11Fh	CSANA1	19Fh
通用寄存器 80字节	20h–6Fh	通用寄存器 80字节	A0h–EFh	通用寄存器 80字节	120h–16Fh	触摸专用寄存器 64字节	1A0h–1DFh
						禁止访问	1E0h–1EFh
公用RAM区 16字节	70h–7Fh	访问70h~7Fh	F0h–FFh	访问70h~7Fh	170h–17Fh	访问70h~7Fh	1F0h–1FFh
BANK 0		BANK 1		BANK 2		BANK 3	

未实现的数据存储单元，读为0。

未实现的数据存储单元，读为0。

图 2-2：数据存储映射

2.2.2.1 特殊功能寄存器位定义（BANK0）

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
BANK0										
00H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元（不是物理寄存器）								xxxx xxxx
01H	TMR0	TIMER0 模块寄存器								xxxx_xxxx
02H	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
03H	STATUS	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
04H	MSR	间接数据存储器地址指针								xxxx-xxxx
05H	PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--x0_x000
06H	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	0000_0000
07H	PORTC	—	—	—	—	PC3	PC2	PC1	PC0	----_0000
08H	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	0000_0000
09H	—	未实现								—
0AH	PCLATH	—	—	—	—	PC11	PC10	PC9	PC8	----_0000
0BH	INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
0CH	PIFB1	INT1E	ADCIE	OSFIE	TMR1IE	INT1F	ADCIF	OSFIF	TMR1IF	0000_0000
0DH	PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000_0000
0EH	TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx-xxxx
0FH	TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx-xxxx
10H	T1STA	T1GINV	TMR1GE	T1CKPS		T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000_0000
11H	SSP1STAT	—	—	D/A	P	S	R/W	UA	BF	0000_0000
12H	SSP1CON1	WCOL	SSP1OV	SSP1EN	CKP	SSP1M3	SSP1M2	SSP1M1	SSP1M0	0000_0000
13H	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000_0000
14H	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	—	SBCDE	AHEN	DHEN	0000_0000
15H	SSP1MSK	I2C 掩码								0000_0000
16H	SSP1ADD	波特率时钟分频与地址								0000_0000
17H	SSP1BUF	I2C 数据缓存								0000_0000
18H	WDTCON	—	—	—	WDTPS				SWDTEN	---0_1000
19H	PIFB3	—	—	—	BCL1IE	—	—	—	BCL1IF	---0_---0
1AH	PWMCON	—	PWMFS	PWM1OE	PWM0OE	PWM1EN	PWM0EN	TMR0EN	T1GSS	-000_0011
1BH	—	未实现								—
1CH	—	未实现								—
1DH	—	未实现								—
1EH	ADRESH	AD 结果寄存器								0000_0000
1FH	—	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，阴影 = 未实现

注：配置字寄存器中的 CFG_MCLRE 为 1 时 RA3 上拉使能。

2.2.2.2 特殊功能寄存器位定义（BANK1）

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
BANK1										
80H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元（不是物理寄存器）								xxxx xxxx
81H	OPT_REG	INT1EG	INT0EG	TCS	TCE	PSC	PS			1111_1111
82H	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
83H	STATUS	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
84H	MSR	间接数据存储器地址指针								xxxx-xxxx
85H	CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11_1111
86H	CPIOB	CPIOB7	CPIOB6	CPIOB5	CPIOB4	CPIOB3	CPIOB2	CPIOB1	CPIOB0	1111_1111
87H	CPIOC	—	—	—	—	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111_1111
88H	CPIOD	CPIOD7	CPIOD6	CPIOD5	CPIOD4	CPIOD3	CPIOD2	CPIOD1	CPIOD0	1111_1111
89H	—	未实现								—
8AH	PCLATH	—	—	—	—	PC11	PC10	PC9	PC8	----_0000
8BH	INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
8CH	—	未实现								—
8DH	—	未实现								—
8EH	PSTA	—	—	ULPWUE	SLVREN	$\overline{MCRL}$	IER	$\overline{POR}$	$\overline{LVR}$	--01_100q
8FH	OSCCON	—	IRCF			OSTS	HTS	LTS	SCS	-101_x001
90H	—	未实现								—
91H	ADINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111_1111
92H	IRCTRIML	内部快时钟校正寄存器低 8 位								0000_0000
93H	IRCTRIMH	—	—	IRCTEN	内部快时钟校正寄存器高 5 位					--00_0000
94H	—	未实现								—
95H	PAPHR	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0	--00_0000
96H	PBPHR	PBPHR7	PBPHR6	PBPHR5	PBPHR4	PBPHR3	PBPHR2	PBPHR1	PBPHR0	1111_1111
97H	PCPHR	PDC3	PDC2	PDC1	PDC0	PCPHR3	PCPHR2	PCPHR1	PCPHR0	0000_1111
98H	PDPHR	PDPHR7	PDPHR6	PDPHR5	PDPHR4	PDPHR3	PDPHR2	PDPHR1	PDPHR0	1111_1111
99H	PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	0000_0000
9AH	PBINTR	PBINTR7	PBINTR6	PBINTR5	PBINTR4	PBINTR3	PBINTR2	PBINTR1	PBINTR0	0000_0000
9BH	—	未实现								—
9CH	—	未实现								—
9DH	—	未实现								—
9EH	ADRESL	AD 结果寄存器								0000_0000
9FH	ADC0TH	ADC 零点校正寄存器								0000_0000

2.2.2.3 特殊功能寄存器位定义（BANK2）

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
BANK2										
100H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元（不是物理寄存器）								xxxx xxxx
101H	—	未实现								—
102H	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
103H	STATUS	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
104H	MSR	间接数据存储器地址指针								xxxx-xxxx
105H	—	未实现								—
106H	—	未实现								—
107H	—	未实现								—
108H	—	未实现								—
109H	—	未实现								—
10AH	PCLATH	—	—	—	—	PC11	PC10	PC9	PC8	----_000
10BH	INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
10CH	—	未实现								—
10DH	—	未实现								—
10EH	—	未实现								—
10FH	—	未实现								—
110H	—	未实现								—
111H	—	未实现								—
112H	PRL0L	PWM0 占空比重载寄存器低字节								0000_0000
113H	PRL1L	PWM1 占空比重载寄存器低字节								0000_0000
114H	—	未实现								—
115H	—	未实现								—
116H	—	未实现								—
117H	T1PRLL	PWM0/1 周期寄存器的低字节								0000_0000
118H	ET1PRL	PWM0/1 周期重载寄存器的低字节								0000_0000
119H	—	未实现								—
11AH	—	未实现								—
11BH	—	未实现								—
11CH	—	未实现								—
11DH	—	未实现								—
11EH	ADC0CN	ADCTHSEL	ADFM	SC1	SC0	ADC_EN	GO/$\overline{DONE}$	CM1	CM0	0000_0000
11FH	—	未实现								—

2.2.2.4 特殊功能寄存器位定义（BANK3）

地址	名称	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	默认值
BANK3										
180H	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元（不是物理寄存器）								xxxx xxxx
181H	—	未实现								—
182H	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000_0000
183H	STATUS	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
184H	MSR	间接数据存储器地址指针								xxxx-xxxx
185H	—	未实现								—
186H	—	未实现								—
187H	—	未实现								—
188H	—	未实现								—
189H	—	未实现								—
18AH	PCLATH	—	—	—	—	—	PC10	PC9	PC8	----_0000
18BH	INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
18CH	—	未实现								—
18DH	—	未实现								—
18EH	—	未实现								—
18FH	—	未实现								—
190H	CSCTL	CSEN	RAMEN	XRAM	CSBUSY	CSINTS	CSACU2	CSACU1	CSACU0	0000_0011
191H	THCTL	—	THGAP1	THGAP0	THMOD2	THMOD1	THMOD0	THDS1	THDS0	0010_1001
192H	PWM0L	PWM0 占空比比较寄存器低字节								0000_0000
193H	PWM1L	PWM1 占空比比较寄存器低字节								0000_0000
194H	CHRLTH	触摸通道结果高 8 位标志								0000_0000
195H	CHRLTL	触摸通道结果低 8 位标志								0000_0000
196H	CHSELH	触摸通道使能寄存器高 8 位								0000_0000
197H	CHSELL	触摸通道使能寄存器低 8 位								0000_0000
198H	EPR0	—	—	—	—	EPR0H(从动)		PWM0H		----_0000
199H	EPR1	—	—	—	—	EPR1H(从动)		PWM1H		----_0000
19AH	T1PRLH	—	—	—	—	ET1PRH (从动)		T1PRLH		----_0000
19BH	CSTMRH	触摸扫描周期计数器高 8 位								0000_0000
19CH	CSTMRL	触摸扫描周期计数器低 8 位								1111_1111
19DH	ADS0	CHNSEL<3:0>				REFSL<1:0>		—	VREFSEL	0000_01-1
19EH	CSANA0	—	BSTRM2	BSTRM1	BSTRM0	CMPTRM		VTSEL1	VTSEL0	0010_0111
19FH	CSANA1	—	—	—	DTTRM		CTTRM			---0_1011

2.2.2.5 STATUS 状态寄存器

状态（STATUS）寄存器包含：
- ALU 的算术运算状态
- 复位状态
- 数据存储器（SRAM）的存储区选择位

和其他寄存器一样，状态寄存器也可以作为任何指令的目标寄存器。如果一条影响 Z、HC 或 C 位的指令以状态寄存器作为目标寄存器，将禁止写这三位。根据器件逻辑，这些位会被置 1 或清零。此外，也不能写$\overline{TF}$和$\overline{PF}$位。因此，当执行一条把状态寄存器作为目标寄存器的指令后，状态寄存器的结果可能和预想的不一样。

例如，执行 CLRR STATUS 指令会清零该寄存器的高 3 位并将 Z 位置 1。从而使状态寄存器的值为“000uu1uu”（其中 u 表示不变）。

因此，建议仅使用 BCR、BSR、SWAPR 和 STWR 指令来改变状态寄存器，因为这些指令不影响任何状态位。欲知其他不会影响任何状态位的指令，请参见第 12.0 节“指令表”。

寄存器 03H/83H/183H：状态寄存器（STATUS）

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	**RP1**	**RP0**	**$\overline{TF}$**	**$\overline{PF}$**	**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 **IRP**：间接寻址最高位：
1 = 间接寻址第2、3页
0 = 间接寻址第0、1页

bit6 bit5 **RP1RP0**：寄存器存储区选择位（用于直接寻址）
00 = 直接寻址第0页
01 = 直接寻址第1页
10 = 直接寻址第2页
11 = 直接寻址第3页

bit4 **$\overline{TF}$**：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 **$\overline{PF}$**：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/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：借位的极性是相反的。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。

2.2.2.6 OPT_REG 选项寄存器

选项（OPT_REG）寄存器是可读写的寄存器，包含可对以下各项进行配置的各种配置位：

- Timer0/WDT 预分频器分配位
- 外部中断
- Timer0

> 注： 要为Timer0指定1:1的预分频比，应将OPT_REG寄存器的PSC位置1，以将预分频器分配给WDT。请参见第7.1.4节“软件可编程预分频器”。

寄存器 81H：选项寄存器（OPT_REG）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
INT1EG	**INT0EG**	**TCS**	**TCE**	**PSC**	**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 **INT1EG**：INT1中断边沿选择位
1 = 由PD6/INT1引脚的上升沿触发中断
0 = 由PD6/INT1引脚的下降沿触发中断

bit6 **INT0EG**：INT0中断边沿选择位
1 = 由PA1/INT0引脚的上升沿触发中断
0 = 由PA1/INT0引脚的下降沿触发中断

bit5 **TCS**：TMR0时钟源选择位
1 = PA2/T0CKI引脚上信号的跳变
0 = 内部指令周期时钟（FOSC/4）

bit4 **TCE**：TMR0时钟源边沿选择位
1 = 在PA2/T0CKI引脚电平发生由高到低的跳变时递增
0 = 在PA2/T0CKI引脚电平发生由低到高的跳变时递增

bit3 **PSC**：预分频器分配位
1 = 将预分频器分配给WDT
0 = 将预分频器分配给Timer0模块

bit2-0 **PS<2:0>**：预分频比选择位

位值	TMR0分频比	WDT分频比
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

2.2.2.7 PSTA 寄存器

电源控制（PSTA）寄存器包含区分以下复位的标志位：

- 上电复位（$\overline{\text{POR}}$）
- 欠压复位（$\overline{\text{LVR}}$）
- 看门狗定时器复位（WDT）
- 外部$\overline{\text{MCLR}}$复位

PSTA 寄存器也用于控制超低功耗唤醒和$\overline{\text{LVR}}$的软件使能。

寄存器 8EH：电源控制寄存器1（PSTA）

U-0	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-x
—	—	**ULPWUE**	**SLVREN**	**$\overline{\text{MCR}}$**	**IER**	**$\overline{\text{POR}}$**	**$\overline{\text{LVR}}$**
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 **ULPWUE**：超低功耗唤醒使能位
1 = 使能超低功耗唤醒
0 = 禁止超低功耗唤醒

bit4 **SLVREN**：软件欠压检测使能位(1)
1 = 使能欠压检测
0 = 禁止欠压检测

bit3 **$\overline{\text{MCR}}$**：外部复位状态位
1 = 上电复位值
0 = 发生了外部复位

bit2 **IER**：指令错误复位状态位
1 = 发生了指令错误复位
0 =上电复位值

bit1 **$\overline{\text{POR}}$**：上电复位状态位
1 = 未发生上电复位
0 = 发生了上电复位（必须在上电复位发生后用软件置1）

bit0 **$\overline{\text{LVR}}$**：欠压复位状态位
1 = 未发生欠压复位
0 = 已发生欠压复位（必须在发生欠压复位后用软件置1）

`注 1：当配置字寄存器中的LVREN<1:0> = 01时允许使用该位对LVR进行控制。

2.3 PCL 和 PCLATH

程序计数器（PC）为 12 位宽。其低 8 位来自可读写的 PCL 寄存器，高 4 位（PC<11:8>）来自 PCLATH，不能直接读写。只要发生复位，PC 就将被清零。图 2-3 显示了装载 PC 值的两种情形。图 2-3 中上方的例子说明在写 PCL（PCLATH<3:0>→PCH）时是如何装载 PC 的。

图 2-3：在不同情况下装载 PC

2.3.1 修改 PCL

执行任何以 PCL 寄存器为目标寄存器的指令将同时使程序计数器的 PC<11:8>位（PCH）被 PCLATH 寄存器的内容所取代。这样可通过将所需的高 4 位写入 PCLATH 寄存器来改变程序计数器的所有内容。当低 8 位写入 PCL 寄存器时，程序计数器的所有 12 位将变为 PCLATH 寄存器中所包含的值以及写入 PCL 寄存器中的值。

计算 LJUMP 指令是通过向程序计数器加入偏移量（ADDWR PCL）来实现的。通过修改 PCL 寄存器跳转到查找表或程序分支表（计算 LJUMP）时应特别谨慎。假定 PCLATH 设置为表的起始地址，如果表长度大于 255 条指令，或如果存储器地址的低 8 位在表的中间从 0xFF 计满返回到 0x00，那么在每次表起始地址与表内的目标地址之间发生计满返回时，PCLATH 必须均必须递增。

2.3.2 堆栈

YS83B08/YS84B08/YS84C12 器件具有 8 级深×12 位宽的硬件堆栈（见图 2-2 和图 2-4）。堆栈空间既不占用程序存储区空间，也不占用数据存储区空间，而且堆栈指针是不可读写的。当执行 CALL 指令或当中断导致程序跳转时，值 PC 将被压入（PUSH）堆栈。而在执行 RET、RTIW 或 RTFI 指令时，堆栈中的断点地址将从堆栈中弹出（POP）到 PC 中。PCLATH 不受 PUSH 或 POP 操作的影响。

堆栈的工作原理犹如循环缓冲区。这意味着当堆栈压栈 8 次后，第 9 次压栈的数值将会覆盖第一次压栈时所保存的数值，而第十次压栈数值将覆盖第二次压栈时保存的数值，以后依次类推。

注： 1：不存在指明堆栈是否上溢或下溢的状态标志位。

2：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CALL、RET、RTIW和RTFI指令，或源于指向中断向量地址。

寄存器 02H/82H/182H：程序计数器低字节控制寄存器（PCL）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PC7	**PC6**	**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<7:0>**：程序计数器低8位。

寄存器 0AH/8AH/18AH：程序计数器高字节控制寄存器（PCLATH）

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	**PC11**	**PC10**	**PC9**	**PC8**
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 **PC<11:8>**：程序计数器高4位。

2.4 间接寻址、IAR 和 MSR 寄存器

IAR 寄存器不是实际存在的寄存器，使用 IAR 寄存器寻址将产生间接寻址。

使用 IAR 寄存器可进行间接寻址。任何使用 IAR 寄存器的指令，实际上是对文件选择寄存器（MSR）所指向的数据进行存取。间接对 IAR 进行读操作将返回 00h。间接对 IAR 寄存器进行写操作将导致空操作（尽管可能会影响状态标志位）。通过将 8 位的 MSR 寄存器与 STATUS 寄存器的 IRP 位进行组合可得到一个有效的 9 位地址，如图 2-4 所示。

例 2-1 给出了一个使用间接寻址将 RAM 地址单元 20h-2Fh 清零的简单程序。

例 2-1：间接寻址

```
        ;功能：地址 20 开始的连续 16 个字节的 RAM 清 0
        LDWI    0X20    ;
        STWR    MSR     ;地址指针赋初值
NEXT:   CLRR    IAR     ;清除 MSR 值对应的寄存器值
        INCR    MSR,R   ;递增地址指针
        BTSS    MSR,4   ;完成没有？
        LJUMP   NEXT    ;没完成，继续
CONTINUE:               ;完成了
```

图 2-4：YS83B08/YS84B08/YS84C12 的直接/间接寻址

2.5 看门狗定时器（WDT）

WDT 具有以下特性：

- 工作于 LFINTOSC（31kHz）
- 包含 16 位预分频器
- 与 Timer0 共用 8 位预分频器
- 配置位和使用软件控制

WDT 在如表 2-1 所述的情况下清零。

2.5.1 WDT 振荡器

WDT 的时基来源于 31kHz LFINTOSC。OSCCON 寄存器的 LTS 位不反映 LFINTOSC 是否被使能。

在所有复位时 WDTCON 的值为“---01000”。这样标称时基为 17ms。

注：	当执行振荡器起振定时器（OST）时，WDT保持复位状态，因为OST使用WDT纹波计数器来执行振荡器延时计数。OST计数到期后，WDT将开始计数（如果使能）。

2.5.2 WDT 控制

WDTE 位在配置字寄存器中。该位置 1 时，WDT 连续运行。

配置字寄存器中的 WDTE 位置 1 时，WDTCON 寄存器中的 SWDTEN 位不起作用。如果 WDTE 清零，那么 SWDTEN 位可用于使能和禁止 WDT。该位置 1 使能 WDT，该位清零禁止 WDT。

看门狗定时器框图中（图 2-5）的 PSC 和 PS<2:0>位在 OPT_REG 寄存器中。更多信息请参见第 7.1 节“Timer0 模块”。

注 1：这是 **Timer0** 与 **WDT** 共用的预分频器。更多信息，请参见 **第7.0节 “Timer0 模块”**。

图 2–5：看门狗定时器框图

寄存器 18H：看门狗定时器控制寄存器（WDTCON）

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	**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

bit7-5　未实现：读为0

Bit4-1　**WDTPS<3:0>**：看门狗定时器周期选择位

位值 = 预分频
0000 = 1:32
0001 = 1:64
0010 = 1:128
0011 = 1:256
0100 = 1:512 （复位值）
0101 = 1:1024
0110 = 1:2048
0111 = 1:4096
1000 = 1:8192
1001 = 1:16384
1010 = 1:32768
1011 = 1:65536
1100 = 保留
1101 = 保留
1110 = 保留
1111 = 保留

Bit0　**SWDTEN**：软件使能或禁止看门狗定时器位(1)

1 = WDT 开启
0 = WDT 关闭（复位值）

注 1：如果配置字寄存器（CONFIG）的WDTE配置位=1，则WDT始终被使能，而与该控制位的状态无关。
如果配置字寄存器（CONFIG）的WDTE配置位=0，则可以使用该控制位开启/关闭WDT。

表 2-1：WDT 状态

条件	WDT
CLRWT 命令	清零
振荡器失效检测	
退出休眠 + 系统时钟 = T1OSC、EXTRC、NTRC 或 EXTCLK	
退出休眠 + 系统时钟 = XT、HS 或 LP	清零直到 OST 结束

表 2-2：与看门狗定时器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 LVR 时的值	所有其他复位值
WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 0000	xxxx xxxx
OPT_REG	—	—	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111	xxxx xxxx
CONFIG0	—	CPB	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0	—	—

图注： 看门狗定时器不使用阴影单元。

注 1： 关于配置字寄存器中所有位的操作，请参见寄存器 CCFG0。

3.0 复位

3.1 概述

YS83B08/YS84B08/YS84C12 有以下几种不同类型的复位：

a） 上电复位（POR）
b） 正常工作期间的 WDT 复位
c） 休眠期间的 WDT 复位
d） 休眠期间的$\overline{\text{MCLR}}$复位
e） 欠压复位（LVR）

有些寄存器不受任何复位的影响；在上电复位时它们的状态未知，而在其他复位时状态不变。大多数寄存器在以下复位时会复位到各自的“复位状态”：

- 上电复位
- $\overline{\text{MCLR}}$复位
- 休眠期间的$\overline{\text{MCLR}}$复位
- WDT 复位

WDT 唤醒不会导致寄存器像 WDT 复位那样复位，这是因为唤醒被视为恢复正常工作。TF 和 PF 位在不同的复位情形下会分别被置 1 或清零，如表 3-1 所示。软件可使用这些位判断复位的性质。

图 3-1 给出了片上复位电路的简化框图。$\overline{\text{MCLR}}$复位路径上有一个噪声滤波器，用来检测并滤除小脉冲。关于脉冲宽度规范，请参见第 13.0 节“电气特性”。

图 3-1：片上复位电路的简化框图

表 3-1：STATUS/PSTA 位及其含义

$\overline{\text{POR}}$	$\overline{\text{LVR}}$	$\overline{\text{TF}}$	$\overline{\text{PF}}$	条件
0	x	1	1	上电复位
u	0	1	1	欠压复位
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	正常工作期间的$\overline{\text{MCLR}}$复位
U	u	1	0	休眠期间的$\overline{\text{MCLR}}$复位

图注： u = 不变， x = 未知

3.2 上电复位

在 VDD 达到足以使器件正常工作的电平之前，片上上电复位电路将使器件保持在复位状态。要有效利用 POR，只要将$\overline{\text{MCLR}}$引脚通过一个电阻连接到 VDD 即可。需要一个最大上升时间才能达到 VDD。详见第 13.0 节“电气特性”。如果使能了欠压复位，那么该最大上升时间规范将不再适用。欠压复位电路将使器件保持在复位状态，直到 VDD 达到 VLVR（见第 3.5 节“欠压复位（LVR）)”。

注： 当VDD降低时，上电复位电路不会产生内部复位。要重新使能上电复位，VDD必须至少保持100us 的Vss电压。

当器件开始正常工作（退出复位状态）时，器件的工作参数（即电压、频率和温度等）必须得到满足，以确保其正常工作。如果不满足这些条件，那么器件必须保持在复位状态，直到满足工作条件为止。

3.3 MCLR

YS83B08/YS84B08/YS84C12 在$\overline{\text{MCLR}}$复位路径中有一个噪声滤波器。该滤波器检测并滤除小脉冲。

应注意，WDT 复位不会将$\overline{\text{MCLR}}$引脚驱动为低电平。

若施加在$\overline{\text{MCLR}}$引脚上的电压超出规范值，则在 ESD 事件发生期间可导致$\overline{\text{MCLR}}$复位且器件中流过超过规范值的过电流。因此，建议不要把$\overline{\text{MCLR}}$引脚直接连接到 VDD。建议使用图 3-3 给出的 RC 网络。

通过清零配置字寄存器中的$\overline{\text{MCLR}}$位，可使能内部$\overline{\text{MCLR}}$选项。当 MCLRE = 0 时，在内部产生芯片的复位信号。当 MCLRE = 1 时，PA3/$\overline{\text{MCLR}}$引脚变成外部复位输入。在这种模式下，PA3/$\overline{\text{MCLR}}$引脚具有到 VDD 的弱上拉功能。

图 3-3：建议的$\overline{\text{MCLR}}$电路

3.4 上电延时定时器（PWRT）

上电延时定时器仅在上电时（上电复位或欠压复位）提供一个 55ms（标称值）的固定延时。上电延时定时器采用 LFINTOSC 振荡器作为时钟源，工作频率为 31kHz。更多信息，请参见第 4.5 节“内部时钟模式”。

只要 PWRT 处于活动状态，芯片就保持在复位状态。配 PWRT 延时使 VDD 有足够的时间上升到所需的电平。置位$\overline{\text{PWRTE}}$可以禁止（如果置 1）或使能（如果清零或被编程）上电延时定时器。虽然不是必需的，但是在使能欠压复位时也应使能上电延时定时器。

由于以下原因不同芯片的上电延时定时器的延时也各不相同：

- VDD 差异
- 温度差异
- 制造工艺差异

详见直流参数（第 13.0 节“电气特性”）。

注：	在$\overline{\text{MCLR}}$引脚的低于VSS的电压尖峰，包括大于80mA的电流，可导致闭锁。因此，在$\overline{\text{MCLR}}$引脚上施加“低”电平时，应使用阻值在50-100Ω的串联电阻，而不是将该引脚直接拉到VSS。

3.5 欠压复位

配置字寄存器中的 LVREN0 和 LVREN1 位用于选择 4 种欠压复位模式中的一种。其中添加了两种允许使用软件或硬件对 LVR 的使能进行控制的模式。当 LVREN<1:0> = 01 时，可由 PSTA 寄存器的 SLVREN 位使能/禁止 LVR，从而能用软件对其进行控制。通过选择 LVREN<1:0>，可使欠压复位在休眠时被自动禁止，从而节约功耗；而在唤醒后被重新使能。在此模式下，SLVREN 位被禁止。关于配置字的定义，请参见寄存器 PSTA。

如果 VDD 下降到 VLVR 以下，且持续时间超过参数值（TLVR）（见第 13.0 节“电气特性”，欠压状况将使）器件复位。不管 VDD 的变化速率如何，上述情况都会发生。如果 VDD 低于 VLVR 的时间少于参数值（TLVR），则不一定会发生复位。

任何复位（上电复位、欠压复位或看门狗定时器复位等）都会使芯片保持复位状态，直到 VDD 上升到 VLVR 以上（见图 3-5）。如果使能了上电延时定时器，此时它将启动，并且会使器件保持复位状态的时间延长 64ms。

注：	配置字寄存器中的$\overline{\text{PWRTE}}$位用于使能上电延时定时器。

如果在上电延时定时器运行过程中，VDD 降低到 VLVR 以下，芯片将重新回到欠压复位状态并且上电延时定时器会恢复为初始状态。一旦 VDD 上升到 VLVR 以上，上电延时定时器将执行一段 64ms 的复位。

图 3-5：欠压情形

表 3-5：与欠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 LVR 时的值	所有其他复位值
PSTA	—	—	ULPWUE	SLVREN	MCR	IER	POR	LVR	0001 100x	xxxx xxxx
STATUS	IRP	PR1	PAGE	$\overline{\text{TF}}$	$\overline{\text{PF}}$	Z	HC	C	0001 1xxx	xxxx xxxx

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。 LVR 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。

3.6 延时时序

上电时的延时时序如下：

- 在 POR 延时结束后，施加一段 PWRT 延时。
- PWRT 超时后激活 OST。

总延时时间取决于振荡器配置和$\overline{\text{PWRTE}}$位的状态。例如，在 EC 模式且$\overline{\text{PWRTE}}$位被擦除（PWRT 禁止）的情况下，根本不会出现延时。图 3-6-1、3-6-2 和 3-6-3 分别给出了各种情形下的延时时序。当振荡器起振后，通过使能故障保护监控器，器件将以 INTOSC 作为时钟源来执行代码（第 4.9 节“故障保护时钟监控器”）。

由于延时是由上电复位脉冲触发的，因此如果$\overline{\text{MCLR}}$保持足够长时间的低电平，所有延时都将结束。将$\overline{\text{MCLR}}$电平拉高后，器件将立即开始执行代码（见图 3-6-2）。这对于测试或同步多个并行工作的 YS83B08/YS84B08/YS84C12 器件来说是非常有用的。

表 3-6-1 给出了一些特殊寄存器的复位条件，而表 3-2 给出了所有寄存器的复位条件。

图 3-6-1：上电时的超时时序（$\overline{\text{MCLR}}$延时）：情形 1

图 3-6-2：上电时的超时时序（$\overline{\text{MCLR}}$延时）：情形 2

图 3-6-3：上电时的超时时序（$\overline{MCLR}$ 及 VDD）

表 3-6-1：特殊寄存器的初始状态

条件	程序计数器	状态寄存器	PSTA 寄存器
上电复位	0000h	0001 1xxx	--01 --0x
正常工作期间的$\overline{MCLR}$复位	0000h	000u uuuu	--0u --uu
休眠期间的$\overline{MCLR}$复位	0000h	0001 0uuu	--0u --uu
WDT 复位	0000h	0000 uuuu	--0u --uu
WDT 唤醒	PC + 1	uuu0 0uuu	--uu --uu
欠压复位	0000h	0001 1uuu	--01 --u0
通过中断从休眠唤醒	PC + 1(1)	uuu1 0uuu	--uu --uu

图注： u = 不变， x = 未知， — = 未实现位，读为 0。

注 1： 当器件被中断唤醒且全局中断允许位 GIE 位置 1 时，执行 PC+1 后，PC 装入中断向是（0004h）。

表 3-6-2：各种情形下的超时

振荡器配置	上电		欠压复位		从休眠状态唤醒
	$\overline{PWRTE}$ = 0	$\overline{PWRTE}$ = 1	$\overline{PWRTE}$ = 0	$\overline{PWRTE}$ = 1	
XT，HS，LP	T_{PWRT} + 1024 * T_{OSC}	1024 * T_{OSC}	T_{PWRT} + 1024 * T_{OSC}	1024 * T_{OSC}	1024 * T_{OSC}
RC，EC，INTOSC	T_{PWRT}	—	T_{PWRT}	—	—

3.7 电源控制（PSTA）寄存器

电源控制寄存器 PSTA（寄存器 8EH）有两个状态位，用于指示上次发生的复位的类型。

bit0 是 LVR（欠压复位）标志位。LVR 在上电复位时未知。然后，用户必须将该位置 1，并在随后的复位发时检查 LVR 是否为 0，如果是，则表示已发生欠压复位。当禁止欠压复位电路（配置字寄存器中的 LVREN<1:0> = 00）时，LVR 状态位是“无关位”并且不一定预测得到。

bit1 是 POR（上电复位）标志位，在上电复位时值为 0，其他情况下不受影响。上电复位后，用户必须对该位写 1。发生后续复位后，如果 POR 为 0，则表示发生了上电复位（即 VDD 可能已经变为了低电平）。

更多信息，请参见第 6.5.5 节“超低功耗唤醒”和第 3.5 节“欠压复位（LVR）”。

4.0 系统时钟

4.1 概述

振荡器有多种时钟源和选择功能，从而使其应用非常广泛，并可最大限度地提高性能和降低功耗。图 4-1 给出了振荡器模块的框图。

时钟源可以配置为由外部振荡器、石英晶体谐振器、陶瓷谐振器以及阻容（RC）电路提供。此外，系统时钟源可以配置为由两个内部振荡器中的一个提供，并可以通过软件选择速度。其他时钟功能包括：

- 通过软件选择外部或内部系统时钟源（**注意：上电默认系统时钟为内部时钟**）。
- 故障保护时钟监视器（FSCM）旨在检测外部时钟源的故障（LP、XT、HS、EC 或 RC 模式）并自动切换到内部振荡器。

振荡器模块可配置为以下 8 种时钟模式之一。

1. EC——外部时钟，I/O 在 PA4 上，时钟从 PA5 输入。
2. LP——32kHz 低功耗晶振模式。
3. XT——中等增益晶振或陶瓷谐振振荡器模式。
4. HS——高增益晶振或陶瓷谐振器模式。
5. RC——外部阻容（RC），FOSC/4 输出到 OSC2/CLKOUT。
6. RC+IO——外部阻容，I/O 在 PA4 上。
7. IRC——内部振荡器，PA4 输出 1/4 系统时钟频率，PA5 为 I/O PIN。
8. IRC+IO——内部振荡器，PA4、PA5 作为 I/O PIN。

通过配置字寄存器（CONFIG）的 FOSC<2:0>位来配置时钟源模式。内部时钟可用两个内部振荡器产生。HFINTOSC 是经过校准的高频振荡器。LFINTOSC 是未经校准的低频振荡器。

图 4-1：MCU 时钟源框图

4.2 振荡器控制

振荡器控制（OSCCON）寄存器（图 4-1）控制系统时钟和频率选择等选项。OSCCON 寄存器包含以下位：

- 频率选择位（IRCF）
- 频率状态位（HTS 和 LTS）
- 系统时钟控制位（OSTS 和 SCS）

寄存器 8FH：振荡器控制寄存器（OSCCON）

U-0	R/W-1	R/W-0	R/W-1	R-1	R-0	R-0	R/W-1
—	**IRCF2**	**IRCF1**	**IRCF0**	**OSTS**(1)	**HTS**	**LTS**	**SCS**
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

bit6-4 **IRCF<2:0>**：内部振荡器频率选择位
000 = 31kHz
001 = 250kHz
010 = 500kHz
011 = 1MHz
100 = 2MHz
101 = 4MHz （缺省值）
110 = 8MHz
111 = 16MHz

bit3 **OSTS**：振荡器起振超时状态位(1)
1 = 器件运行在FOSC<2:0>定义的外部时钟之下
0 = 器件运行在内部振荡器之下（HFINTOSC或LFINTOSC）

bit2 **HTS**：HFINTOSC（高频——16MHz到250kHz）状态位
1 = HFINTOSC稳定
0 = HFINTOSC不稳定

bit1 **LTS**：LFINTOSC（低频——31kHz）状态位
1 = LFINTOSC稳定
0 = LFINTOSC不稳定

bit0 **SCS**：系统时钟选择位
1 = 内部振荡器用于系统时钟
0 = 时钟源由FOSC<2:0>决定

注 1：故障保护模式使能时，该位将复位为0。

4.3 时钟源模式

时钟源模式可分为外部和内部模式。

- 外部时钟模式依靠外部电路提供时钟源。例子有：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。振荡器模块有两个内部振荡器，一个是 16MHz 高频内部振荡器（HFINTOSC），另一个是 31kHz 低频内部振荡器（LFINTOSC）。

可通过 OSCCON 寄存器的系统时钟选择（SCS）位，在外部或内部时钟源之间选择系统时钟。（欲了解更多信息，请参见第 4.6 节“时钟切换”）。

4.4 外部时钟模式

4.4.1 振荡器起振定时器（OST）

如果振荡器模块配置为 LP、XT 或 HS 模式，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）之后以及上电延时定时器（PWRT）延时结束（如果配置了）时，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启动并向振荡器模块提供稳定的系统时钟信号。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 4-4-1 给出了振荡器延时的例子。

表 4-4-1：振荡器延时示例

切换自	切换到	频率	振荡器延时
休眠/POR	LFINTOSC HFINTOSC	31kHz 250kHz到16MHz	振荡器预热延时（T_{WARM}）
休眠/POR	EC，RC	DC—20MHz	双周期
LFINTOSC（31kHz）	EC，RC	DC—20MHz	每次一周期
休眠/POR	LP，XT，HS	32kHz到20MHz	1024个时钟周期（OST）
LFINTOSC（31kHz）	HFINTOSC	250kHz到16MHz	1us（近似值）

4.4.2 EC 模式

外部时钟（EC）模式允许外部产生的逻辑电平作为系统时钟源。工作在此模式下时，外部时钟源连接到 OSC1 输入，OSC2 引脚可用作通用 I/O。图 4-4-2 给出了 EC 模式的引脚连接。

当选取 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 YSPRING 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 4-4-2：外部时钟（EC）模式的工作原理

4.4.3 LP、XT 和 HS 模式

LP、XT 和 HS 模式支持连接到 OSC1 和 OSC2 的石英晶体谐振器或陶瓷谐振器的使用（图 4-4-3-1）。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。该模式设计仅用于驱动 32.768kHz 音叉（Tuning Fork）式晶振（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。该模式最适用于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。该模式最适用于驱动需要高驱动设定的谐振器。

图 4-4-3-1 和图 4-4-3-2 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：石英晶振的特性随类型、封装和制造商而变化。要了解规格说明和推荐应用，应查阅制造商提供的数据手册。

2：应始终验证振荡器在应用预期的VDD和温度范围内的性能。

图 4-4-3-1：石英晶体的工作原理（LP、XT 或 HS 模式）

图 4-4-3-2：陶瓷谐振器的工作原理（XT 或 HS 模式）

4.4.4 外部 RC 模式

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频率选择空间，且保持成本最低。有 RC 和 RCIO 两种模式。

在 RC 模式下，电路连接到 OSC1。RCOSC2/CLKOUT 输出 RC 振荡频率的 4 分频。该信号可用来为外部电路、同步、校准、测试或其他应用需求提供时钟。图 4-4-4 给出了外部 RC 模式的连接图。

图 4-4-4：外部 RC 模式

在 RCIO 模式下，RC 电路连接到 OSC1。OSC2 成为额外的通用 I/O 引脚。I/O 引脚成为 PORTA 的 bit4（PA4）。图 4-5-5 给出了 RCIO 模式的连接图。RC 振荡器频率是供电电压、电阻（R_{EXT}）和电容（C_{EXT}）值以及工作温度的函数。影响振荡器频率的其他因素有：

- 电压门限值变化
- 元件容差
- 不同封装的电容

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。

4.5 内部时钟模式

振荡器模块有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. HFINTOSC（高频内部振荡器）出厂时已校准，工作频率为 16MHz，精度范围为±1%。

2. LFINTOSC（低频内部振荡器）未经校准，工作频率为 31kHz。

通过软件对 OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>进行操作，可选择系统时钟速度。

可通过 OSCCON 寄存器的系统时钟选择（SCS）位，在外部或内部时钟源之间选择系统时钟（见第 4.6 节“时钟切换”）。

4.5.1 INTOSC 和 INTOSCIO 模式

当在配置字寄存器（CONFIG）中使用振荡器选择位 FOSC<2:0>设置器件时，在 INTOSC 和 INTOSCIO 模式下将内部振荡器配置为系统时钟源。在 INTOSC 模式下，OSC1/CLKIN 可用作通用 I/O。OSC2/CLKOUT 输出所选内部振荡器频率的 4 分频。CLKOUT 信号可用来为外部电路、同步、校准、测试或其他应用需求提供时钟。在 INTOSCIO 模式下，OSC1/CLKIN 和 OSC2/CLKOUT 引脚可用作通用 I/O。

4.5.2 HFINTOSC

高频内部振荡器（HFINTOSC）是出厂时已校准的 16MHz 内部时钟源。

HFINTOSC 的输出连接到后分频器和多路复用器（见图 4-1）。使用 OSCCON 寄存器的 IRCF<2:0>位，

可通过软件选择七个频率之一。更多信息，请参见第 4.5.4 节“频率选择位（IRCF）”。

将 OSCCON 寄存器的 ICRF<2:0>位设置为≠000 选择 16MHz 到 250kHz 之间的任一频率，可使能 HFINTOSC。OSCCON 寄存器的 HF 内部振荡器（HTS）位用于显示 HFINTOSC 是否稳定。

4.5.3 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准的 31kHz 内部时钟源。LFINTOSC 的输出连接到后分频器和多路复用器（见图 4-1）。通过软件对 OSCCON 寄存器的 IRCF<2:0>位进行操作，选取 31kHz。更多信息，请参见第 4.5.4 节“频率选择位（IRCF）”。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控器（FSCM）的时钟源。选取 31kHz（将 OSCCON 寄存器的 IRCF<2:0>位设置为 000）为系统时钟源（OSCCON 寄存器的位 SCS = 1），或者使能以下任一项时，LFINTOSC 将被使能：

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

OSCCON 寄存器的 LF 内部振荡器（LTS）位用于指示 LFINTOSC 是否稳定。

4.5.4 频率选择位（IRCF）

16MHz HFINTOSC 和 31kHz LFINTOSC 的输出连接到后分频器和多路复用器（见图 4-1）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>用于选择内部振荡器的频率输出。可通过软件选择以下 8 个频率之一：

- 16MHz
- 8MHz
- 4MHz （复位后的缺省值）
- 2MHz
- 1MHz
- 500kHz
- 250kHz
- 31kHz

注：	任何复位后，OSCCON寄存器的IRCF<2:0>位将被置为101且频率选择置为4MHz。用户可修改IRCF位来选择其他频率。

4.5.5 HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见图 4-5-5）。在这种情况下，OSCCON 寄存器的 IRCF 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

1. OSCCON 寄存器的 IRCF<2:0>位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低，时钟切换电路等待新时钟上升沿的到来。
5. 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 HTS 和 LTS 位按要求被更新。
6. 时钟切换完成。

更多信息请参见图 4-1。

如果选取的内部振荡器速度在 16MHz 到 250kHz 之间，选取新频率不存在启动延时。这是因为新旧频率都来自经过后分频器和多路复用器的 HFINTOSC。

启动延时规范在第 13.0 节“电气特性”中与振荡器相关的表格中。

图 4-5-5：内部振荡器切换时序

4.6 时钟切换

通过软件对 OSCCON 寄存器的系统时钟选择（SCS）位进行操作，可将系统时钟源在外部和内部时钟源之间切换。

4.6.1 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。

- OSCCON 寄存器的位 SCS = 0 时，系统时钟源由配置字寄存器（CONFIG）中 FOSC<2:0>位的配置决定。
- OSCCON 寄存器的位 SCS = 1 时，根据 OSCCON 寄存器的 IRCF<2:0>位所选的内部振荡器频率选取系统时钟源。复位后，OSCCON 寄存器的 SCS 总是被清零。

注：	任何自动时钟切换（可能产生自故障保护时钟监控器）都不更新OSCCON寄存器的SCS位。用户可监控OSCCON寄存器的OSTS位以确定当前的系统时钟源。

4.6.2 振荡器起振超时状态（OSTS）位

OSCCON 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，还是来自内部时钟源。外部时钟源由配置字寄存器（CONFIG）的 FOSC<2:0>定义。OSTS 还特别指明在 LP、XT 或 HS 模式下，振荡器起振定时器（OST）是否已超时。

4.7 掉电模式（休眠）

通过执行 SLEEP 指令可进入掉电模式。

如果使能看门狗定时器：

- WDT 将被清零并保持运行。
- 状态寄存器中的 PF 位被清零。
- TF 位被置 1。
- 关闭振荡器驱动器。
- I/O 端口保持执行 SLEEP 指令之前的状态（驱动为高电平、低电平或高阻状态）。

为使这种模式下的电流消耗降至最低，所有 I/O 引脚都应保持为 VDD 或 VSS，以确保没有外部电路从 I/O 引脚消耗电流。为了避免输入引脚悬空而引入开关电流，应在外部将高阻输入的 I/O 引脚拉为高电平或低电平。为使电流消耗降至最低，T0CKI 输入也应保持为 VDD 或 VSS。还应考虑 PORTA 片上上拉的影响。

$\overline{\text{MCLR}}$引脚必须为逻辑高电平。

注：	请注意，由于WDT超时产生的复位不会将$\overline{\text{MCLR}}$引脚驱动为低电平。

4.7.1 从休眠状态唤醒

由于在休眠期间没有片上时钟处于工作状态，下列外设中断可以将器件从休眠状态唤醒：

1. TMR1 中断。Timer1 必须用作异步计数器
2. PORTA、PORTB 电平变化中断
3. 来自 INT0 与 INT1 引脚的外部中断
4. 看门狗定时器唤醒（如果 WDT 使能）

$\overline{\text{MCLR}}$引脚上的有效信号会导致器件复位。其他事件被认为是程序执行的继续。状态寄存器中的 TF 和 PF 位用于确定器而在执行 SLEEP 件复位的原因。PF 位在上电时被置 1，执行指令时被清零。TF 位在发生 WDT 唤醒时被清零。

当执行 SLEEP 指令时，下一条指令（PC+1）将预先取出。如果希望通过中断事件唤醒器件，则必须将相应的中断允许位置 1（使能）。发生唤醒与 GIE 位的状态无关。如果 GIE 位被清零（禁止），器件将继续执行 SLEEP 指令之后的指令。如果 GIE 位被置 1（使能），器件执行 SLEEP 指令之后的指令，然后跳转到中断地址（0004h）处执行代码。如果不希望执行 SLEEP 指令之后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。

注：	如果禁止了全局中断（GIE被清零），但有任一中断源将其中断允许位以及相应的中断标志位置1，器件将立即从休眠状态唤醒。

器件从休眠状态唤醒时，WDT 都将被清零，而与唤醒原因无关。

4.7.2 使用中断唤醒

当禁止全局中断（GIE 被清零）时，并且有任一中断源将其中断允许位和中断标志位置 1，将会发生下列事件之一：

- 如果在执行 SLEEP 之前产生了中断，那么 SLEEP 指令将被作为一条 NOP 指令执行。因此，WDT 及其预分频器和后分频器（如果使能）将不会被清零，并且 TF 位将不会被置 1，同时 PF 位也不会被清零。

- 如果在执行 SLEEP 指令期间或之后产生了中断，那么器件将被立即从休眠状态唤醒。SLEEP 指令将在唤醒之前执行完毕。因此，WDT 及其预分频器和后分频器（如果使能）将被清零，并且 TF 位将被置 1，同时 PF 位也将被清零。

即使在执行 SLEEP 指令之前，检查到标志位为 0，它也可能在 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 PF 位。如果 PF 位置 1，则说明 SLEEP 指令被当作一条 NOP 指令执行了。

在执行 SLEEP 指令之前，必须先执行一条 CLRWT 指令，来确保将 WDT 清零。详情请参见图 4-8。

图 4-8：通过中断将器件从休眠状态唤醒

4.8 故障保护时钟监控器

故障保护时钟监控器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM 能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。FSCM 通过将配置字寄存器（CONFIG）中的 FCMEN 位置 1 来使能。FSCM 可用于所有外部振荡模式（LP、XT、HS、EC、RC 和 RCIO）。

图 4-9：FSCM 框图

4.8.1 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。LFINTOSC 除以 64，就产生了采样时钟。请参见图 4-9。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，锁存器被置 1。在采样时钟的每个上升沿，锁存器被清零。如果采样时钟的整个半周期流逝而主时钟依然未进入低电平，就检测到故障。

4.8.2 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 PIFB 寄存器的 OSFIF 标志位置 1。如果在 PIFB 寄存器的 OSFIE 位置 1 的同时将该标志位置 1，将产生中断。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。系统时钟将继续来自内部时钟源，直到器件固件成功重启外部振荡器并切换回外部操作。FSCM 所选的内部时钟源由 OSCCON 寄存器的 IRCF<2:0>位决定。这使内部振荡器可以在故障发生前就得以配置。

4.8.3 故障保护条件清除

复位、执行 SLEEP 指令或翻转 OSCCON 寄存器的 SCS 位后，故障保护条件被清除。OSCCON 寄存器的 SCS 位被修改后，OST 将重新启动。OST 运行时，器件继续从 OSCCON 中选定的 INTOSC 进行操作。OST 超时后，故障保护条件被清除，器件将从外部时钟源进行操作。必须先清除故障保护条件，才能清零 OSFIF 标志位。

4.8.4 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。OST 的使用场合为从休眠状态唤醒后以及任何类型的复位后。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 就处于激活状态。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的范围变化较大，在振荡器起振期间（即，从复位或休眠中退出时），故障保护电路不处于激活状态。经过一段适当的时间后，用户应检查OSCCON寄存器的OSTS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。

4.9 内部快时钟调频功能

当系统时钟为内部快时钟（IRC）且需要对频率进行调节时，可设置 IRC 频率调节使能位 IRCTEN 为 1，使能 IRC 校准功能。此时通过设置 **IRCTRIM**｛IRCTRIMH<4：0>，IRCTRIML<7:3>｝的值可对 IRC 频率进行粗调, 每个 LSB（步长）变化约为 50KHz；设置 **IRCFT**<2:0>值则可以微调，每个 LSB（步长）变化约为 6KHz。

IRCTRIM 值变化对应频率变化的规律如下：

注：

1、IRCTEN 为 0 时，对 IRC 的调节失效，此时 IRC 频率为出厂校准值；

2、频率的变化步长并非是绝对的；

3、由于 IRC 可能会存在小幅抖动，IRCFT 微调功能仅供参考；

寄存器92H：快时钟校正寄存器低8位 < IRCTRIML >

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
IRCTRIM<4：0>					IRCFT		

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

bit7-3 **IRCTRIM<4：0>**：IRC 频率调节位低 5 位

bit2-0 **IRCFT**：IRC 频率微调位

寄存器93H：快时钟校正寄存器高5位 < IRCTRIMH >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IRCTEN	IRCTRIM<9：5>				
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 未实现位

bit6 未实现位

bit5 **IRCTEN**：IRC 频率调节使能位
1 = IRC 频率调节使能
0 = IRC 频率调节禁止

bit4-0 **IRCTRIM<9：5>**：IRC频率调节位高5位

5.0 中断

YS83B08/YS84B08/YS84C12 有以下多种中断源：

- 外部中断 PA1/INT0、PD6/INT1
- Timer0、Timer1 溢出中断、PWM 中断
- PA、PB 电平变化中断
- A/D 中断
- 触摸中断、I2C 中断
- 时钟丢失中断

中断控制寄存器（INTS）和外设中断请求寄存器（PIFB1、PIFB2、PIFB3）在各自的标志位中记录各种中断请求。INTS 寄存器还包括各个中断允许位和全局中断允许位。

INTS 寄存器中的全局中断允许位 GIE 在置 1 时允许所有未屏蔽的中断，而在清零时禁止所有中断。可以通过 INTS 和 PIFB（1、2、3）寄存器中相应的允许位来禁止各个中断。复位时 GIE 被清零。

响应中断时，自动发生以下动作：

- GIE 被清零以禁止任何其他中断。
- 返回地址被压入堆栈。
- 在 PC 中装入 0004h。

执行“从中断返回”指令 RTFI 退出中断程序并将 GIE 位置 1，从而重新使能未屏蔽的中断。

INTS 寄存器包含以下中断标志位：

- INT0 引脚中断
- PORTA 电平变化中断
- Timer0 溢出中断

外设中断允许和标志位在 PIFB1、PIFB2、PIFB3 寄存器中。

对于外部中断事件，如 INT0、INT1 引脚或 PORTA、PORTB 电平变化中断，中断响应延时将为 3 到 4 个指令周期。确切的延时时间取决于发生中断事件的时间（见图 5-1-1）。对于单周期或双周期指令，中断响应延时完全相同。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

注： 1：各中断标志位的置1不受相应的中断屏蔽位或GIE位状态的影响。

2：当执行一条清零GIE位的指令后，任何等待在下一周期执行的中断都将被忽略。当GIE位被再次置1后，被忽略的中断仍会继续等待处理。

关于 Timer1、A/D 模块的更多信息，请参见相应的外设章节。

图 5-1-1：中断逻辑

注 1: 在此采样 **INTF** 标志（每个 **Q1** 周期）。
2: 异步中断响应延时为 3-4 个**TCY** 。同步中断响应延时为 3 个 **TCY**，其中 **TCY** 为一个指令周期，无论 **Inst (PC)** 是单周期还是双周期指令，中断响应延时都是相同的。
3: 只有在 **INTS** 和 **RC** 振荡器模式下 **CLKOUT** 才有效。
4: 关于 **INT** 脉冲的最小宽度，请参见 **第 13.0 节“电气特性”** 中的交流规范。
5: 允许在 **Q4-Q1** 周期内的任何时间将 **INTF** 置 **1** 。

图 5-1-2：INT 引脚中断时序

5.1 中断特殊功能寄存器

5.1.1 INTS 寄存器

INTS 寄存器是可读写的寄存器，包含 TMR0 寄存器溢出、PORTA 电平变化和外部 PA1/INT0 引脚中断的各种允许和标志位。

注： 当有中断条件产生时，不管相应的中断允许位或INTS寄存器的全局允许位GIE状态如何，中断标志位都将置1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

寄存器 0BH/8BH/10BH/18BH：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**	**TIS**	**INT0E**	**PAIE**	**TIF**	**INT0F**	**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 = 允许所有未屏蔽的中断
0 = 禁止所有中断

bit6 **PEIE**：外设中断允许位
1 = 允许所有未屏蔽的外设中断
0 = 禁止所有外设中断

bit5 **TIS**：Timer0溢出中断允许位
1 = 允许Timer0中断
0 = 禁止Timer0中断

bit4 **INT0E**：PA1/INT0外部中断允许位
1 = 允许PA1/INT0外部中断
0 = 禁止PA1/INT0外部中断

bit3 **PAIE**：PORTA电平变化中断允许位(1)
1 = 允许PORTA电平变化中断
0 = 禁止PORTA电平变化中断

bit2 **TIF**：Timer0溢出中断标志位(2)
1 = Timer0寄存器已经溢出（必须用软件清零）
0 = Timer0寄存器没有溢出

bit1 **INT0F**：PA1/INT0外部中断标志位
1 = 发生了PA1/INT0外部中断（必须用软件清零）
0 = 未发生PA1/INT0外部中断

bit0 **PAIF**：PORTA电平变化中断标志位(1)
1 = 至少一个PORTA<5:0>引脚的电平状态发生了改变（必须用软件清零）
0 = 没有一个PORTA<5:0>引脚的电平状态发生改变

注 1：必须同时使能PAINTR寄存器。
2：当TMR0计满回零时，TIF位置1。复位时TMR0的状态不变，它应该在清零TIF位之前被初始化。

5.1.2 外设中断允许寄存器

PIFB1、PIFB2、PIFB3 寄存器包含外设中断允许和标志位。

注： 当有中断条件产生时，不管相应的中断允许位或INTS寄存器的全局允许位GIE状态如何，中断标志位都将置1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

寄存器0CH：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
INT1E	**ADCIE**	**OSFIE**	**TMR1IE**	**INT1F**	**ADCIF**	**OSFIF**	**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 **INT1E**：PD6/INT1外部中断允许位
1 = 允许PD6/INT1外部中断
0 = 禁止PD6/INT1外部中断

bit6 **ADCIF**：ADC转换器中断允许位
1 = 允许A/D转换器中断
0 = 禁止A/D转换器中断

bit5 **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

bit4 **TMR1IE**：Timer1溢出中断允许位
1 = 允许Timer1溢出中断
0 = 禁止Timer1溢出中断

bit3 **INT1F**：PD6/INT1外部中断标志位
1 = 发生了PD6/INT1外部中断（必须用软件清零）
0 = 未发生PD6/INT1外部中断

bit2 **ADCIF**：ADC中断标志位
1 = A/D转换完成
0 = A/D转换尚未完成或尚未启动

bit1 **OSFIF**：振荡器故障中断标志位
1 = 系统振荡器发生故障，时钟输入切换为INTOSC（必须用软件清零）
0 = 系统时钟正常运行

bit0 **TMR1IF**：Timer1溢出中断标志位
1 = Timer1寄存器已经溢出（必须用软件清零）
0 = Timer1寄存器未溢出

寄存器0DH：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	R/W-0	R/W-0	R/W-0
SSP1IE	**CSIE**	**PBIE**	**PWMIE**	**SSP1IF**	**CSIF**	**PBIF**	**PWMIF**
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 **SSP1IE**：I2C中断允许位
1 = 允许I2C中断
0 = 禁止I2C中断

bit6 **CSIE**：触摸中断允许位
1 = 允许触摸中断
0 = 禁止触摸中断

bit5 **PBIE**：PORTB电平变化中断允许位
1 = 允许PORTB电平变化中断
0 = 禁止PORTB电平变化中断

bit4 **PWMIE**：PWM中断允许位
1 = 允许PWM中断
0 = 禁止PWM中断

bit3 **SSP1IF**：I2C中断标志位
1 = 发生了I2C中断（必须用软件清零）
0 = 未发生I2C中断

bit2 **CSIF**：触摸中断标志位
1 = 发生了触摸中断
0 = 未发生触摸中断

bit1 **PBIF**：PORTB电平变化中断标志位
1 = 至少一个PORTB<7:0>引脚的电平状态发生了改变（必须用软件清零）
0 = 没有一个PORTB<7:0>引脚的电平状态发生改变

bit0 **PWMIF**：PWM中断标志位
1 = 发生PWM中断（必须用软件清零）
0 = 未发生PWM中断

寄存器19H：外设中断请求寄存器3（PIFB3）

R-0	R-0	R-0	R/W-0	R-0	R-0	R-0	R/W-0
—	—	—	**BCL1IE**	—	—	—	**BCL1IF**
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 **未实现**

bit5 **未实现**

bit4 **BCL1IE**：I2C总线冲突中断允许位
1 = 允许I2C总线冲突中断
0 = 禁止I2C总线冲突中断

bit3 **未实现**

bit2 **未实现**

bit1 **未实现**

bit0 **BCL1IF**：I2C总线冲突中断标志位
1 = 发生了I2C总线冲突中断
0 = 未发生I2C总线冲突中断

表 5-1：与中断相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 LVR 时的值	所有其他复位值
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	xxxx xxxx
PAINTRA	—	—	PAINTRA5	PAINTRA4	PAINTRA3	PAINTRA2	PAINTRA1	PAINTRA0	--00 0000	xxxx xxxx
PBINTRB	PAINTRB7	PAINTRB6	PAINTRB5	PAINTRB4	PAINTRB3	PAINTRB2	PAINTRB1	PAINTRB0	0000 0000	xxxx xxxx
PIFB1	INT1IE	ADCIE	OSFIE	TMR1IE	INT1IF	ADCIF	OSFIF	TMRIF	0000 0000	xxxx xxxx
PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000 0000	xxxx xxxx
PIFB3	—	—	—	BCL1IE	—	—	—	BCL1IF	---0 ---0	xxxx xxxx

图注： x = 未知， u = 不变， — = 未实现（读为 0）， q = 取值视情况而定。中断模块不使用阴影单元。

5.2 外部中断

INT0(PA1)、INT1(PD6)引脚上的外部中断是边沿触发的；当 OPT_REG 寄存器的 INT1EG 与 INT0EG 位被置 1 时在上升沿触发，而当 INT1EG 与 INT0EG 位被清零时在下降沿触发。当 INT0(PA1)、INT1(PD6)引脚上出现有效边沿时，INTS 寄存器的 INT0F 位、PIFB1 寄存器的 INT1F 位置 1。可以通过将 INTS 寄存器的 INT0E、PIFB1 寄存器的 INT1E 控制位清零来禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INT0F、INT1F 位清零。如果 INT0E 与 INT1F 位在进入休眠状态前被置 1，则 INT0(PA1)、INT1(PD6)引脚上中断能将处理器从休眠状态唤醒。关于 INT0(PA1)、INT1(PD6)引脚上中断将处理器从休眠状态唤醒的时序，请参见图 4-8。

注： 必须对ADINS和CPIOA、CPIOD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总是读为0。

5.3 Timer0 中断

TMR0 寄存器溢出（FFh→ 00h）会将 INTS 寄存器的 T0IF 位置 1。可以通过置 1/清零 INTS 寄存器的 T0IE 位来使能/禁止该中断。关于 Timer0 模块的操作，请参见第 7.1 节 “Timer0 模块”。

5.4 PORTA、PORTB 电平变化中断

PORTA、PORTB 输入电平的变化会使的 PAIF、PBIF 位置 1。可以通过置 1/清零的 PAIE、PBIE 位来使能/禁止该中断。此外，可通过 PAINTAR、PAINTBR 寄存器对该端口的各个引脚进行配置。

注： 当读操作正在执行时发生了I/O引脚电平变化，则PAIF、PBIF中断标志位可能不会被置1。

5.5 A/D 中断

ADC 模块在模数转换完成时，PIFB1 寄存器中的 ADCIF 中断标志位均置 1。可以通过置 1/清零 PIFB1 寄存器中的 ADCIE 位来使能/禁止该中断。

5.6 中断的现场保护

在中断期间，仅将返回的 PC 值压入堆栈。通常情况下，用户可能希望在中断期间保存关键寄存器（例如，W 寄存器和状态寄存器）。这必须用软件实现。应将临时保存寄存器 W_TEMP 和 STATUS_TEMP 置于 GPR

的末 16 字节中（见图 2–2）。这 16 个单元是所有存储区共用的，无需分区。这样就简化了现场保护和恢复操作。例 5–7 中所示的代码可用于：

- 保存 W 寄存器
- 保存状态寄存器
- 执行 ISR 代码
- 恢复状态寄存器（和存储区选择位寄存器）
- 恢复 W 寄存器

注： YS83B08/YS84B08/YS84C12通常不需要保存PCLATH。但是，如果要在ISR和主程序中使用计算LJUMP，就必须在ISR中保存和恢复PCLATH。

例 5–7：将状态寄存器和 W 寄存器保存在 RAM 中

```
STWR     W_TEMP          ; 保存 W 寄存器到 W_TEMP
SWAPR    STATUS, W       ; 将 STATUS 寄存器高低四位交换后保存到 W（目的：不改变标志）
STWR     STATUS_TEMP     ; 保存 STATUS 寄存器内容到 STATUS_TEMP
                         ;
;(ISR)                   ; 中断服务程序
                         ;
SWAPR    STATUS_TEMP, W  ; 交换 STATUS_TEMP 高低四个字节
STWR     STATUS          ; 恢复标志状态寄存器
SWAPR    W_TEMP, R       ; 恢复 W 寄存器值（用交换指令的目的：不改变状态寄存器值）
SWAPR    W_TEMP, W
```

6.0 I/O 端口

YS83B08/YS84B08/YS84C12 共有 26 个通用 I/O 引脚。根据使能的外设不同，有些（或全部）引脚不能用作通用 I/O。通常使能了一个外设后，相关的引脚就不能用作通用 I/O 引脚了。

6.1 PORTA 和 CPIOA 寄存器

PORTA 是 6 位宽的双向端口。PORTA 对应的数据方向寄存器是 CPIOA（寄存器 85H）。将 CPIOA 位置 1（ = 1）可以使对应的 PORTA 引脚作为输入引脚（即禁止相应的输出驱动器）。将 CPIOA 位清零（ = 0）将使对应的 PORTA 引脚作为输出引脚（即使能输出驱动器并将输出锁存器的内容置于所选的引脚上）。PA3 是个例外，它是开漏极输出，且驱动能力很弱，最大只能 5ma 的灌电流。例 6-1 给出了初始化 PORTA 的方法。

读 PORTA 寄存器（寄存器 05H）将读取引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读－修改－写操作。因此，写一个端口就意味着读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。当 MCLRE = 1 时，PA3 读为 0。

即使在 PORTA 引脚被用作模拟输入的时候，CPIOA 寄存器仍然控制 PORTA 引脚的方向。在将它们用作模拟输入时，用户必须确保 CPIOA 寄存器中的位保持为置 1 状态。配置为模拟输入的 I/O 引脚始终读为 0。

注： 必须对ADINS寄存器进行初始化以将模拟通道配置为数字输入通道。配置为模拟输入的引脚读为0。

例 6-1：初始化 PORTA

```
BCR     STATUS,PAGE     ;Bank0
CLRR    PORTA           ;清除端口 A 输出缓冲器
BSR     STATUS,PAGE     ;Bank1
CLRR    ADINS           ; 所有模拟/数字复用端口设置为数字端口
LDWI    0CH             ;设置端口 A 的<3,2>位为输入，其他输出
STWR    CPIOA           ;
BCR     STATUS,PAGE     ;Bank0
```

寄存器 05H：PORTA寄存器（PORTA）

U-0	U-0	R/W-x	R/W-x	R/W-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>**：PORTA I/O引脚位
1 = PORTA引脚电平>V_{IH}
0 = PORTA引脚电平<V_{IL}

寄存器85H：PORTA三态寄存器（CPIOA）

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	**CPIOA5**	**CPIOA4**	**CPIOA3**	**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；
2：PA3仅可作开漏输出（Open Drain）；

6.2 PORTB 和 CPIOB 寄存器

PORTB 是 8 位宽的双向端口。引脚可以配置为 I/O 或作为触摸模拟输入。PORTB 对应的数据方向寄存器是 CPIOB（寄存器 86H）。将 CPIOB 位置 1（ =1）可以使对应的 PORTB 引脚作为输入引脚（即禁止相应的输出驱动器）。将 CPIOB 位清零（ = 0）将使对应的 PORTA 引脚作为输出引脚（即使能输出驱动器并将输出锁存器的内容置于所选的引脚上）。

读 PORTB 寄存器（寄存器 06H）将读取引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读－修改－写操作。因此，写一个端口就意味着读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。

即使在 PORTB 引脚被用作模拟输入的时候，CPIOB 寄存器仍然控制 PORTB 引脚的方向。在将它们用作模拟输入时，用户必须确保 CPIOB 寄存器中的位保持为置 1 状态。配置为模拟输入的 I/O 引脚始终读为 0。

寄存器06H：PORTB寄存器（PORTB）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PB7	**PB6**	**PB5**	**PB4**	**PB3**	**PB2**	**PB1**	**PB0**
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　　**PB<7:0>**：PORTB I/O引脚位
1 = PORTB引脚电平>V_{IH}
0 = PORTB引脚电平<V_{IL}

寄存器86H：PORTB三态寄存器（CPIOB）

R/W-1	R/W-1	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
CPIOB7	**CPIOB6**	**CPIOB5**	**CPIOB4**	**CPIOB3**	**CPIOB2**	**CPIOB1**	**CPIOB0**
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 **CPIOB<7:0>**：PORTB三态控制位
1 = PORTB引脚配置为输入（三态）
0 = PORTB引脚配置为输出

6.3 PORTC 和 CPIOC 寄存器

PORTC 是由 4 个双向引脚组成的通用 I/O。引脚可以配置为 I/O 或作为触摸模拟输入。

注： 必须对ADINS寄存器进行初始化以将模拟通道配置为数字输入通道。配置为模拟输入的引脚读为0。

例 6-4：初始化 PORTC

```
BCR     STATUS,PAGE     ;Bank 0
CLRR    PORTC           ;清输出缓冲区
BSR     STATUS,PAGE     ;Bank 1
CLRR    ADINS           ;设置端口为数字端口
LDWI    0Ch             ;设置 PC1，PC2 为输入，其他 PC 口输出
STWR    CPIOC
BCR     STATUS,PAGE     ;Bank 0
```

寄存器 07H：PORTC寄存器（PORTC）

U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	**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

Bit3-0 **PC<3:0>**：PORTCI/O引脚位
1 = PORTC引脚电平>VIH
0 = PORTC引脚电平<VIL

寄存器 87H：PORTC三态寄存器（CPIOC）

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	**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

Bit3-0 **CPIOC<3:0>**：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

6.4 PORTD 和 CPIOD 寄存器

PORTD 是由 8 个双向引脚组成的通用 I/O。引脚可以配置为 I/O 或作为触摸模拟输入。要了解各个功能的特定信息（ADC、Touch），请参见本数据手册的相应章节。

寄存器08H：PORTD寄存器（PORTD）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PD7	**PD6**	**PD5**	**PD4**	**PD3**	**PD2**	**PD1**	**PD0**
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 **PD<7:0>**：PORTD I/O引脚位
1 = PORTD引脚电平>V_{IH}
0 = PORTD引脚电平<V_{IL}

寄存器88H：CPIOD寄存器（CPIOD）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CPIOD7	**CPIOD6**	**CPIOD5**	**CPIOD4**	**CPIOD3**	**CPIOD2**	**CPIOD1**	**CPIOD0**
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 **CPIOD<7:0>**：PORTD 三态控制位
1 = PORTD引脚配置为输入（三态）
0 = PORTD引脚配置为输出

6.5 其它引脚功能

YS83B08/YS84B08/YS84C12 上的每一个 PORTA 引脚具有电平变化中断和弱上拉功能。PA0 具有超低功耗唤醒功能。下面三个小节将介绍这些功能。PORTB 引脚具有电平变化中断和弱上拉功能、PORTC 引脚具有弱上、下拉功能。PORTD 引脚具有弱上拉功能。

6.5.1 ADINS 寄存器

ADINS 寄存器用于将 I/O 引脚的输入模式配置为模拟。将相应的 ADINS 位置为高电平将使对该引脚的所有读操作结果为 0，并使该引脚的模拟功能正常进行。

ADINS 位的状态对数字输出功能没有影响。CPIO 清零且 ADINS 置 1 的引脚仍将作为数字输出工作，但其输出模式将为模拟。在对受影响端口执行读－修改－写指令时，这将导致意外的操作。

寄存器 91H：模拟选择寄存器（ADINS）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7	**ANS6**	**ANS5**	**ANS4**	**ANS3**	**ANS2**	**ANS1**	**ANS0**
PD7	**PD6**	**PD5**	**PD4**	**PD3**	**PD2**	**PD1**	**PD0**
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 **ANS<7:0>**：模拟选择位
1 =模拟输入。引脚被分配为模拟输入[1]。
0 =数字I/O。引脚被分配给端口或特殊功能。

注 1：将某引脚设置为模拟输入将自动禁止数字输入电路、弱上拉以及电平变化中断（如果有的话）。相应CPIO位必须设置为输入模式以允许对该引脚的电压进行外部控制。

6.5.2 弱上拉与下拉

PORTA、PORTB、PORTC、PORTD 引脚的某些位可配置内部弱上拉与下拉，控制位 PAPHRX、PBPHRX、PCPHRx、PDPHRX 使能或禁止每一个弱上拉。请参见寄存器 95H、寄存器 96H、寄存器 97H、和寄存器 98H。当将端口引脚配置为输出时，其弱上拉电路会自动切断。在 PA3 配置为$\overline{\text{MCLR}}$时，自动启动 PA3 的弱上拉功能。$\overline{\text{MCLR}}$上拉不受软件控制。

注意：PORTC 引脚复位过程中为默认输入上拉，可能会产生短暂的高电平（与复位时间有关），应用设计中要特别注意

寄存器 95H：弱上拉PORTA寄存器（PAPHR）

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	**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>**：PORTA弱上拉控制位
1 = 使能上拉
0 = 禁止上拉

注 1：如果引脚处于输出模式（CPIO = 0），则弱上拉器件被自动禁止。
2：在XT、HS和LP振荡模式下PAPHR<5:4>始终读为1。

寄存器96H：弱上拉PORTB寄存器（PBPHR）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PBPHR7	**PBPHR6**	**PBPHR5**	**PBPHR4**	**PBPHR3**	**PBPHR2**	**PBPHR1**	**PBPHR0**
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 **PBPHR <7:0>**：PORTB弱上拉控制位
1 = 使能上拉
0 = 禁止上拉

寄存器97H： PORTC弱上拉和弱下拉寄存器（PCPHR）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PDC3	**PDC2**	**PDC1**	**PDC0**	**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

bit7-4 **PDC <7:4>**：PORTC下拉控制位
1 = 使能下拉
0 = 禁止下拉

bit3-0 **PCPHR <3:0>**：PORTC上拉控制位
1 = 使能上拉
0 = 禁止上拉

寄存器98H：弱上拉PORTD寄存器（PDPHR）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDPHR7	**PDPHR6**	**PDPHR5**	**PDPHR4**	**PDPHR3**	**PDPHR2**	**PDPHR1**	**PDPHR0**
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 **PDPHR <7:0>**：PORTD上拉控制位
1 = 使能上拉
0 = 禁止上拉

6.5.3 电平变化中断

PORTA 和 PORTB 引脚的每一位均可分别配置为电平变化中断引脚。控制位 PAINTRx 与 PBINTRx 使能或禁止每个引脚的中断功能，在上电复位时禁止电平变化中断。

对于已允许电平变化中断的引脚，则将该引脚上的值同上一次读 PORTA、PORTB 时锁存的值进行比较。将上一次“不匹配”的输出一起作逻辑或运算，以便将 INTS、PIFB2 寄存器（寄存器 0BH 和寄存器 0DH）中 PORTA、PORTB 电平变化中断标志位（PAIF 和 PBIF）置 1。

该中断能唤醒休眠下的器件。用户在中断服务程序中通过以下方式清除中断：
a）对 PORTA、PORTB 进行读或写操作。这将结束引脚电平不匹配条件。
b）将标志位 PAIF、PBIF 清零。

电平不匹配条件会继续将 PAIF、PBIF 标志位置 1。而读 PORTA、PORTB 将结束不匹配条件并允许将 PAIF 和 PBIF 标志位清零。锁存器将保持最后一次读取的值不受$\overline{\text{MCLR}}$和欠压复位的影响。在这些复位之后，如果出现电平不匹配，PAIF 和 PBIF 标志位将继续被置 1。

注： 在执行任何PORTA、PORTB操作时如果I/O引脚的电平发生变化，则PAIF和PBIF中断标志位可能不会被置1。

寄存器99H：电平变化中断PORTA寄存器（PAINTR）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	**PAINTR5**	**PAINTR4**	**PAINTR3**	**PAINTR2**	**PAINTR1**	**PAINTR0**
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　　**PAINTR<5:0>**：电平变化中断PORTA控制位
1 = 允许电平变化中断
0 = 禁止电平变化中断

注　　1：必须允许全局中断允许（GIE）以使各中断被识别。
2：在XT、HS和LP振荡模式下PAINTR<5:4>始终读为1。

寄存器9AH：电平变化中断PORTB寄存器（PBINTR）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PBINTR7	**PBINTR6**	**PBINTR5**	**PBINTR4**	**PBINTR3**	**PBINTR2**	**PBINTR1**	**PBINTR0**
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　　**PBINTR <7:0>**：PORTB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

6.5.4 超低功耗唤醒

PA0 上的超低功耗唤醒允许电压缓慢跌落，从而可在不消耗额外电流的情况下，产生 PA0 电平变化中断。通过将 ULPWUE 位（PSTA<5>）置 1 选择该模式。这将产生一个很小的灌电流，可用于将 PA0 上的电容放电。

要使用该功能，PA0 引脚应先被配置为输出高电平以对电容充电，允许 PA0 的电平变化中断并且 PA0 要配置为输入引脚。将 ULPWUE 置 1 开始放电，执行 SLEEP 指令。当 PA0 上的电压下降到 VIL 后，器件将被唤醒并执行下一条指令。如果 INTS 寄存器的 GIE 位置 1，器件将调用中断服务程序（0004h）。更多信息，请参见第 6.5.4 节“电平变化中断”。

该功能提供了低功耗技术，可周期性地唤醒休眠下的器件，一般应用电路如图 6-5-1 所示。延时取决于 PA0 上 RC 电路的放电时间。要了解如何初始化超低功耗唤醒模块，请参见例 6-5-4。

串联电阻提供了 PA0 引脚的过电流保护功能，允许在软件中对延时进行校准（见图 6-6-1）。可用定时器测量电容的充放电时间。然后调整充电时间以提供所需的中断延时。该技术可补偿温度、电压和元件精度所带来的影响。超低功耗唤醒外设还可以配置为简单可编程低压检测设备或温度传感器。

例 6-5-4：超低功耗唤醒的初始化

```
BCR     STATUS,PAGE     ;Bank 0
BSR     PORTA,0         ;清 0 端口 A 的 PA0
BSF     STATUS,PAGE     ;Bank 1
BCR     ADINS,0         ;端口 A 的 PA0 设置为数字口
BCR     CPIOA,0         ;端口 A 的 PA0 设置为输出
LCALL   CapDelay        ; 延时
BSR     PSTA,ULPWUE     ;使能超低功耗唤醒功能
BSR     PAINTR,0        ;设置 PA0 口电平变化唤醒功能
BSR     CPIOA,0         ;设置 PA0 口输入
LDWI    B'10001000'     ;使能电平变化中断、全局中断
STWR    INTS            ;
SLEEP                   ;睡眠，等待唤醒
NOP
```

图 6-5-1 超低功耗唤醒应用电路

6.6 PORTA 引脚说明和引脚图

每个 PORTA 引脚都与其他功能复用, 其他复用功能请参见本数据手册中的相关章节。

6.6.1 PA0/ISCK/ULPWU/SDA

图 6-6-1 给出了此引脚的引脚图。PA0/ISCK/ULPWU/SDA 引脚可配置为下列功能之一：

- 通用 I/O
- 在线串行编程（In-CircuitSerialProgramming™）烧录和调试的时钟 SCK
- 超低功耗唤醒的模拟输入
- I2C 数据

数据总线
写
PAPHR
读
PAPHR
写
PORTA
写
CPIOA
读
CPIOA
读
PORTA
写
PAINTR
读
PAINTR
D
Q
CK
Q
电平变化中断
VDD
弱
I2C_SDO_OUT_EN
I2C_SDO
0
1
VDD
I/O 引脚
Vss
Vt
Vss
ULPWUE
Iulp
Q
D
EN
Q3
读 PORTA
至 I2C_SDI及ISCK

图 6-6-1：PA0 框图

6.6.2 PA1/ISDA/INT0

图 6-6-2 给出了此引脚的引脚图。PA1/ISDA/INT0 引脚可配置为下列功能之一：

- 通用 I/O
- 在线串行编程烧录和调试的数据端口 ISDA
- 外部中断 INT0

图 6-6-2：PA1 框图

6.6.3 PA2/T0CKI/SCL

图 6-6-3 给出了此引脚的引脚图。PA2/T0CKI/SCL 引脚可配置为下列功能之一：

- 通用 I/O
- TIMER0 的时钟输入
- I2C 时钟

图 6-6-3：PA2 框图

6.6.4 PA3/MCLR/VPP

图 6-6-4 给出了此引脚的引脚图。PA3/$\overline{MCLR}$/VPP 引脚可配置为下列功能之一：

- 输入引脚或开漏输出引脚；
- 带弱上拉的外部复位；
- 高压烧写引脚

图 6-6-4：PA3 框图

6.6.5 PA4/T1G/OSC2/CLKOUT

图 6-6-5 给出了此引脚的引脚图。PA4/T1G/OSC2/CLKOUT 引脚可配置为下列功能之一：

- 通用 I/O
- Timer1 门控（计数使能）
- 晶振/谐振器连接
- 时钟输出

图 6-6-5：PA4 框图

注：1、CLK 模式为 XT、HS、LP 和 LPTMR1，且 CLKOUT 使能。

2、具有 CLKOUT 选项。

6.6.6 PA5/T1CKI/OSC1/CLKIN

图 6-6-6 给出了此引脚的引脚图。PA5/T1CKI/OSC1/CLKIN 引脚可配置为下列功能之一：

- 通用 I/O
- TMR1 时钟输入
- 晶振/谐振器连接
- 时钟输入

图 6-6-6：PA5 框图

表 6-4：与 PORTA 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 LVR 时的值	所有其他复位值
ADINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	xxxx xxxx
PSTA	—	—	ULPWUE	SLVREN	MCLR	IER	$\overline{POR}$	$\overline{LVR}$	--01 --qq	xxxx xxxx
INTS	GIE	PE1E	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	--00 0000	xxxx xxxx
OPT_REG	INT1EDG	INT0EDG	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111	xxxx xxxx
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--x0 x000	xxxx xxxx
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	--11 1111	xxxx xxxx
PAPHR	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR 1	PAPHR0	--11 1111	xxxx xxxx

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

6.7 PORTB 引脚说明和引脚图

6.7.1 PBX/CCHX

图 6-7-1 给出了此引脚的引脚图。PBX/CCHX 引脚可配置为下列功能之一：

- 通用 I/O
- 触摸的模拟输入

图 6-7-1：PBX 框图

表 6-7：与 PORTB 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 LVR 时的值	所有其他复位值
INTS	GIE	PE1E	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000 0000	xxxx xxxx
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	0000 0000	xxxx xxxx
CPIOB	CPIOB7	CPIOB6	CPIOB5	CPIOB4	CPIOB3	CPIOB2	CPIOB1	CPIOB0	1111 1111	xxxx xxxx
PBPHR	PBPHR7	PBPHR6	PBPHR5	PBPHR4	PBPHR3	PBPHR2	PBPHR1	PBPHR0	1111 1111	xxxx xxxx
PBINTR	PBINTR7	PBINTR6	PBINTR5	PBINTR4	PBINTR3	PBINTR2	PBINTR1	PBINTR0	1111 1111	xxxx xxxx

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

6.8 PORTC 引脚说明和引脚图

图 6-8-1 所示给出了此引脚的引脚图。PCx/CHS，PC 引脚可以配置为下列功能之一：

- 通用 I/O
- 触摸的模拟输入

图 6-8-1：PCX 框图

表 6-8：与 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 LVR 时的值	所有其他复位值
INTS	GIE	PE1E	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000 0000	xxxx xxxx
PORTC	—	—	—	—	PC3	PC2	PC1	PC0	---- 0000	xxxx xxxx
CPIOC	—	—	—	—	CPIOC3	CPIOC2	CPIOC1	CPIOC0	---- 1111	xxxx xxxx
PCPHR	PDC3	PDC2	PDC1	PDC0	PCPHR3	PCPHR2	PCPHR1	PCPHR0	0000 1111	xxxx xxxx

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

6.9 PORTD 引脚说明和引脚图

6.9.1 PD0/CCH12/AN0/VREF

图 6-9-1 所示给出了此引脚的引脚图。PD0/AN0/CCH/VREF 引脚可配置为下列功能之一：

- 通用 I/O
- A/D 模拟输入
- A/D 参考电压输入
- 触摸模拟输入

图 6-9-1：PD0 框图

6.9.2 PD1~3/CCH13~15/AN1~3

图 6-9-2 所示给出了此引脚的引脚图。PD1~3/CCH13~15/AN1~3 引脚可配置为下列功能之一：

- 通用 I/O
- A/D 模拟输入
- A/D 参考电压输入
- 触摸模拟输入

图 6-9-2：PD1~3 框图

6.9.3 PD4/AN4

图 6-9-3 所示给出了此引脚的引脚图。PD0/AN4 引脚可配置为下列功能之一：

- 通用 I/O
- A/D 模拟输入

图 6-9-3：PD4 框图

6.9.4 PD5/AN5/PWM1

图 6-9-4 所示给出了此引脚的引脚图。PD0/AN5/PWM1 引脚可配置为下列功能之一：

- 通用 I/O
- A/D 模拟输入
- PWM1 输出

图 6-9-4：PD5 框图

6.9.5 PD6/AN6/INT1

图 6-9-5 所示给出了此引脚的引脚图。PD6/AN6/INT1 引脚可配置为下列功能之一：

- 通用 I/O
- A/D 模拟输入
- 外部中断 INT1

图 6-9-5：PD6 框图

6.9.6 PD7/AN7/PWM0

图 6-9-6 所示给出了此引脚的引脚图。PD7/AN7/PWM0 引脚可配置为下列功能之一：

- 通用 I/O
- A/D 模拟输入
- PWM0 输出

图 6-9-6：PD7 框图

表 6-9：与 PORTD 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 LVR 时的值	所有其他复位值
INTS	GIE	PE1E	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000 0000	xxxx xxxx
ADINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	xxxx xxxx
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	0000 0000	xxxx xxxx
CPIOD	CPIOD7	CPIOD6	CPIOD5	CPIOD4	CPIOD3	CPIOD2	CPIOD1	CPIOD0	1111 1111	xxxx xxxx
PDPHR	PDPHR7	PDPHR6	PDPHR5	PDPHR4	PDPHR3	PDPHR2	PDPHR1	PDPHR0	0000 1111	xxxx xxxx

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

7.0 定时器

7.1 Timer0 模块

Timer0 模块是 8 位定时器/计数器，具备以下特性：

- 8 位定时器/计数器寄存器（TMR0）
- 8 位预分频器（与看门狗定时器共用）
- 可编程内部或外部时钟源
- 可编程外部时钟边沿选择
- 溢出中断
- 定时器/计数器重载功能

图 7-1：Timer0 模块的框图

7.1.1 Timer0 的工作原理

Timer0 是一个 8 位并具有重载功能的定时器/计数器，读 TMR0，读的是计数器的值，写 TMR0，写的是重载寄存器里的值。

寄存器 01H：Timer0模块寄存器（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
TMR07	**TMR06**	**TMR05**	**TMR04**	**TMR03**	**TMR02**	**TMR01**	**TMR00**
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>**：8位定时器/计数器寄存器

7.1.2 8 位定时器模式

Timer0 的开关使能位 TMR0EN 见寄存器 PWMCON（寄存器 1AH），只有打开使能位，Timer0 才能工作。作为定时器使用时，Timer0 模块将在每个指令周期递增（无预分频器）。将 OPT_REG 寄存器（寄存器 81H）的 TCS 位清零选择定时器模式。

当 TMR0 被写入时，写入后将立即禁止 2 周期递增。

注：	在TMR0被写入时，为了计入两个指令周期的延时，可以调整写入TMR0寄存器的值。

7.1.3 8 位计数器模式

作为计数器使用时，Timer0 模块将在 T0CKI 引脚的每个上升沿递增。递增边沿由 OPT_REG 寄存器（寄存器 81H）的 TCE 位决定。将 OPT_REG 寄存器的 TCS 位置 1 选择计数器模式。

7.1.4 软件可编程预分频器

Timer0 或看门狗定时器（WDT）之一可使用一个软件可编程预分频器，但两者不能同时使用。预分频器的分配由 OPT_REG 寄存器的 PSC 位控制。要将预分频器分配给 Timer0，必须将 PSC 位清零。

Timer0 模块的预分频比有 8 种选项，从 1:2 至 1:256。预分频比可通过 OPT_REG 寄存器（寄存器 81H）的 PS<2:0>位进行选择。要使 Timer0 模块得到 1:1 的预分频比，必须将预分频器分配给 WDT 模块。

预分频器不可读写。当预分频器被分配给 Timer0 模块时，所有写入 TMR0 寄存器的指令均会将预分频器清零。当预分频器被分配给 WDT 时，一条 CLRWT 指令将同时清零预分频器和 WDT。

7.1.4.1 在 Timer0 和 WDT 模块间切换预分频器

由于预分频器可分配给 Timer0 或 WDT，因此在切换预分频比时可能产生不想要的器件复位。当把预分频器从 Timer0 切换到 WDT 模块时，必须执行例 7-1-4-1-1 所示的指令序列。

例 7-1-4-1-1：更改预分频器（TIMER0→WDT）

```
BCR     STATUS,PAGE     ;Bank 0
CLRWT                   ;清看门狗定时器
CLRR    TMR0            ;清 TMR0
BSR     STATUS,PAGE     ;Bank 1
LDWI    b'10001111'     ;禁止上拉，TMR0 分频给看门狗，分频比 1:128
TMODE                   ;
CLRWT                   ;清看门狗
BCR     STATUS,PAGE     ;Bank 0
```

当将预分频器从 WDT 切换到 Timer0 模块时，必须执行以下指令序列（见例 7-1-4-1-2）。

例 7-1-4-1-2：更改预分频器（WDT→TIMER0）

```
CLRWT                       ;清看门狗
BSR     STATUS,PAGE         ;Bank 1
LDWI    b'xxxx0xxx'         ;将分频器分配给 TMR0
TMODE                       ;
BCR     STATUS,PAGE         ;Bank 0
```

7.1.5 Timer0 中断

TMR0 寄存器从 FFh 溢出到 00h 时，Timer0 将产生中断。INTS 寄存器的 TIF 中断标志位将在每次 TMR0 寄存器溢出时被置 1，无论是否允许了 Timer0 中断。必须用软件将 TIF 位清零。Timer0 中断允许位是 INTS 寄存器的 TIS 位。

注： 由于定时器在休眠时被冻结，因此Timer0中断无法将处理器从休眠中唤醒。

7.1.6 Timer0 与外部时钟配合使用

Timer0 处理计数器模式时，T0CKI 输入和 Timer0 寄存器的同步是通过采样内部相位时钟的 Q2 和 Q4 周期实现的。因此，外部时钟源的高低周期必须满足第 13.0 节“电气特性”中所列的时序要求。

7.1.7 Timer0 重装载功能

Timer0 具有重装载功能，往 TMR0 寄存器内写入装载值后，Timer0 的计数值范围是 TMR0 的装载值~0XFF，记满后 Timer0 溢出中断标志位置 1，计数器重新装载 TMR0 数值

表 7-1-7：与 TIMER0 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 LVR 时的值	所有其他复位值
TMR0	Timer0 模块寄存器								xxxx xxxx	xxxx xxxx
INTS	GIE	PE1E	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
OPT_REG	INT1EG	INT0EG	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111	xxxx xxxx
PWMCON	—	PWMFS	PWM1OE	PWM0OE	PWM1EN	PWM0EN	TMR0EN	T1GSS	--00 0011	xxxx xxxx
CPIOA	—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0	--11 1111	xxxx xxxx

图注： x = 未知， u = 不变， — = 未实现（读为 0）。Timer0 不使用阴影单元。

7.2 带门控的 Timer1 模块

Timer1 模块是 16 位定时器/计数器，具备以下特性：

- 16 位定时器/计数器寄存器（TMR1H:TMR1L）
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选 LP 振荡器
- 同步或异步操作
- T1G 引脚的 Timer1 门控（计数使能）
- 溢出中断
- 溢出时唤醒（仅限外部时钟且异步模式时）

图 7-2：Timer1 模块的框图

7.2.1 Timer1 的工作原理

Timer1 模块是 16 位递增计数器，通过一对寄存器 TMR1H:TMR1L 访问。对 TMR1H 或 TMR1L 的写操作将直接更新计数器。

与内部时钟源配合使用时，该模块为定时器。与外部时钟源配合使用时，该模块可用作定时器或计数器。

如寄存器 10H 所示，Timer1 控制寄存器 T1STA 用于控制 Timer1 并选择 Timer1 模块的各种功能。

寄存器 10H：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	R/W-0	R/W-0
T1GINV	**TMR1GE**	**T1CKPS1**	**T1CKPS0**	**T1OSCEN**	**$\overline{\text{T1SYNC}}$**	**TMR1CS**	**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 **T1GINV**：Timer1门控翻转位[1]
1 = Timer1门控为高电平有效（门控为高电平时Timer1计数）
0 = Timer1门控为低电平有效（门控为低电平时Timer1计数）

bit6 **TMR1GE**：Timer1门控使能位[2]
如果TMR1ON = 0：
此位被忽略
如果TMR1ON = 1：
1 = Timer1在Timer1门控不活动时打开
0 = Timer1打开

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

bit3 **T1OSCEN**：LP振荡器使能控制位
如果无CLKOUT振荡器的INTOSC处于激活状态：
1 = LP振荡器被使能用于Timer1时钟
0 = LP振荡器关闭
否则：
此位被忽略

bit2 **T1SYNC**：Timer1外部时钟输入同步控制位
TMR1CS = 1：
1 = 不同步外部时钟输入
0 = 同步外部时钟输入
TMR1CS = 0：
此位被忽略。Timer1使用内部时钟

bit1 **TMR1CS**：Timer1时钟源选择位
1 = 来自T1CK1引脚（上升沿）的外部时钟
0 = 内部时钟（FOSC）

bit0 **TMR1ON**：Timer1打开位
1 = 使能Timer1
0 = 停止Timer1

注 1：无论门控源为何，T1GINV位均将翻转Timer1门控逻辑。
2：必须将TMR1GE位置1以使用T1G引脚。

Timer1 模块是 16 位定时器/计数器，TMR1H:TMR1L。

寄存器 0EH：16位Timer1低字节的保持寄存器（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
TMR1<7>	**TMR1<6>**	**TMR1<5>**	**TMR1<4>**	**TMR1<3>**	**TMR1<2>**	**TMR1<1>**	**TMR1<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 **TMR1 <7:0>**：16位Timer1定时器/计数器低字节寄存器

寄存器 0FH：16位Timer1高字节的保持寄存器（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
TMR1<15>	**TMR1<14>**	**TMR1<13>**	**TMR1<12>**	**TMR1<11>**	**TMR1<10>**	**TMR1<9>**	**TMR1<8>**
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 **TMR1 <15:8>**：16位Timer1定时器/计数器高字节寄存器

7.2.2 时钟源选择

T1STA 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS = 0 时，时钟源为 FOSC。TMR1CS = 1 时，时钟源由外部提供。

7.2.2.1 内部时钟源

选择内部时钟源时，TMR1H:TMR1L 这对寄存器将在 FOSC 的某个倍数递增，倍数由 Timer1 预分频器决定。

7.2.2.2 外部时钟源

选择外部时钟源时，Timer1 模块可作为定时器，也可作为计数器工作。

计数时，Timer1 在外部时钟输入 T1CKI 的上升沿递增。此外，计数器模式时钟可同步到单片机系统时钟，也可异步运行。

若需外部时钟振荡器(且单片机使用不带 CLKOUT 的 INTOSC)，则 Timer1 可使用 LP 振荡器作为时钟源。

注： 计数器模式下，发生以下任何一个或多个情况后，计数器在首个上升沿递增前，必须先经过一个下降沿：
- POR复位后使能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）。

请参见图7-2-2

注 1：箭头所指为计数器递增。
2：在计数器模式下，在时钟的首个递增上升沿之前必须先经过一个下降沿。

图 7-2-2：TIMER1 递增边沿

7.2.3 Timer1 的预分频器

Timer1 有四种预分频选项，可对时钟输入进行 1、2、4 或 8 分频。T1STA 寄存器的 T1CKPS 位控制预分频计数器。预分频计数器不可直接读写；但当发生对 TMR1H 或 TMR1L 的写操作时，预分频计数器被清零。

7.2.4 Timer1 振荡器

OSC1（输入）引脚与 OSC2（放大器输出）引脚之间内置有一个低功耗 32.768kHz 振荡器电路。T1STA 寄存器将的 T1OSCEN 控制位置 1 使能振荡器。休眠时振荡器继续工作。Timer1 振荡器与系统 LP 振荡器共用。这样，Timer1 就只能在主系统时钟来自内部振荡器或振荡器处于 LP 模式时使用此模式。用户必须提供软件延

时以确保振荡器正常起振。Timer1 振荡器被使能时，CPIOA5 和 CPIOA4 位置 1。PA5 和 PA4 位读为 0 且 CPIOA5 和 CPIOA4 位读为 1。

注： 振荡器在使用前需要一段起振和稳定时间。这样，应将T1OSCEN置1并在使能Timer1前应有一段适当的延时。

7.2.5 Timer1 工作于异步计数器模式

若 T1STA 寄存器的控制位$\overline{\text{T1SYNC}}$置 1，则外部时钟输入不同步。定时器与内部相位时钟异步递增。若选择了外部时钟源，则定时器将在休眠时继续运行，并可在溢出时产生中断，唤醒处理器。然而，读写定时器时应特别谨慎（见第 7.2.5.1 节“在异步计数器模式下读写 Timer1”）。

注： 从同步操作切换为异步操作时，有可能错过一次递增从异步操作切换为同步操作时，有可能多产生一次递增。

7.2.5.1 在异步计数器模式下读写 Timer1

定时器运行于外部异步时钟时，读取 TMR1H 或 TMR1L 将确保读操作有效（由硬件负责）。然而，应该注意的是，用两个 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溢出。

对于写操作，建议用户直接停止计数器，然后写入所期望的值。如果寄存器正进行递增计数，对定时器寄存器进行写操作，可能会导致写入竞争，从而可能在 TMR1H:TMR1L 这对寄存器中产生不可预测的值。

7.2.6 Timer1 门控

Timer1 门控源可软件配置为$\overline{\text{T1G}}$引脚, 这使器件可直接使用$\overline{\text{T1G}}$为外部事件定时。Timer1 门控源的选择请参见 PWMCON 寄存器（寄存器 1AH）。

注： T1STA寄存器的TMR1GE位必须置1，更多有关选择Timer1门控源请参见PWMCON寄存器（寄存器 1AH）。

使用 T1STA 寄存器的 T1GINV 位可翻转 Timer1 门控。这将配置 Timer1 以确保事件之间存在低电平有效或高电平有效的时间。

7.2.7 Timer1 中断

Timer1 的一对寄存器（TMR1H:TMR1L）递增至 FFFFh 后返回 0000h。Timer1 计满返回时，PIFB 寄存器的 Timer1 中断标志位被置 1。为确保计满返回时产生中断，您必须将以下位置 1：

- T1STA 寄存器的 TMR1ON 位
- PIFB 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 位清零将清除中断标志。

注： TMR1H:TTMR1L这对寄存器和TMR1IF位应在允许中断前清零。

7.2.8 Timer1 在休眠模式下的工作原理

只有在设定异步计数器模式时，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信号使计数器递增。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 PIFB1 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
- 必须将 T1STA 寄存器的 TMR1CS 位置 1
- 可将 T1STA 寄存器的 T1OSCEN 位置 1

溢出时器件将被唤醒并执行下一条指令。若 INTS 寄存器的的 GIE 位置 1，器件将调用中断服务程序（0004h）。

7.2.9 与 Timer1 相关的寄存器汇总

表 7-2-12：与 TIMER1 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 LVR 时的值	所有其他复位值
PWMCON	—	PWMFS	PWM1OE	PWM0OE	PWM1EN	PWM0EN	TMR0EN	T1GSS	0000 0011	xxxxxxxx
INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxxxxxx
PIFB1	INT1E	ADCIE	OSFIE	TMR1IE	INT1IF	ADCIF	OSFIF	TMR1IF	0000 0000	xxxxxxxx
TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx xxxx	xxxxxxxx
TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx xxxx	xxxxxxxx
T1STA	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	xxxxxxxx

图注： x = 未知， u = 不变， — = 未实现（读为 0）。Timer1 模块不使用阴影单元。

7.3 PWM 模块

脉宽调制（PWM）是通过一种完全开启和完全关闭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（也称脉宽）可随而变，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 7-3-3 给出了典型的 PWM 波形图。

7.3.1 PWM 的输出

PWM 模式将在 PD7、PD5 引脚产生频率相同、脉宽可调制的 2 路 PWM 信号。PWM 使能时，当{TMR1H,TMR1L} == {T1PRLH，T1PRLL}时，下一计数周期 Timer1 计数器清零。当{TMR1H,TMR1L} == 0 时，PWM 输出高电平，当{TMR1H,TMR1L} == {PWMXH，PWMXL}时，PWM 输出低电平。其占空比，周期和分辨率由以下寄存器决定：

- T1PRLL
- T1PRLH
- PWMXL
- PWMXH

图7-3-3 两路PWM输出

7.3.2 PWM 的周期

PWM 周期由 Timer1 的 T1PRLH、T1PRLL 寄存器来指定。ET1PRL 是周期重载从动寄存器的低 8 位，高两位是 T1PRLH[3:2]。PWM 周期计算公式：

PWM 周期 = [{T1PRLH,T1PRLL+1}] * Tosc * TMR1 预分频值。

当 TMR1 等于{T1PRLH,T1PRLL}时，下一次递增周期将发生以下三个事件：。

- TMR1 被清零。
- PWM 引脚被置 1（例外：若 PWM 占空比 = 0%，引脚不被置 1）。
- PWM 占空比重载寄存器值锁存到占空比寄存器；

通过对以下几个寄存器写入 10 位值可设定 PWM 占空比：PRLxH，PRLxL 为占空比重载寄存器，PWMxL、PWMxH 为占空比比较寄存器（只读）。脉冲宽度直到**周期完成时才将占空比重载寄存器的值锁存到占空比比较寄存器，**这样可以防止程序改变占空比时造成脉冲宽度跳变。用户**设定占空比只需写占空比重载寄存器**，而不是占空比比较寄存器；

脉冲宽度计算公式：

脉冲宽度 = [{PWMxH,PWMxL}+1] * Tosc * TMR1 预分频值。

占空比 = [{PWMxH,PWMxL}+1] /[{T1PRLH,T1PRLL}+1] 。

7.3.3 PWM 的分辨率

分辨率决定某个周期的有效占空比。例如，10 位分辨率有 1024 个分立的占空比，而 8 位分辨率则有 256 个分立的占空比。

[{T1PRLH,T1PRLL}] 为 1024 时产生 10 位的最大 PWM 分辨率。分辨率是[{T1PRLH,T1PRLL}]的函数。

PWM分辨率

$$分辨率 = \frac{Log([T1PRLH,T1PRLL] + 1)}{Log[2]} 位$$

表 7-3-3-1：PWM 频率和分辨率示例（Fosc = 16MHz）

PWM 频率	1.953kHz	3.906kHz	7.8125kHz	15.625kHz	62.5kHz	4000kHz
定时器预分频比（1、2、4、8）	8	4	2	1	1	1
[T1PRLH,T1PRLL]的值	0x03,0xFF	0x03,0xFF	0x03,0xFF	0x03,0x FF	0x00,0xFF	0x00,0x03
最大分辨率（位）	10	10	10	10	8	2

表 7-3-3-2：PWM 频率和分辨率示例（Fosc = 4MHz）

PWM 频率	0.488kHz	0.976kHz	1.953kHz	3.906kHz	15.625kHz	1000kHz
定时器预分频比（1、2、4、8）	8	4	2	1	1	1
[T1PRLH,T1PRLL]的值	0x03,0xFF	0x03,0xFF	0x03,0xFF	0x03,0xFF	0x00,0xFF	0x00,0x03
最大分辨率（位）	10	10	10	10	8	2

7.3.4 PWM 的高低频模式

每个 PWM 输出提供高频低频两种模式，高频模式下 PWM 输出噪声比较小，低频模式下 PWM 输出噪声比较大，高低频模式切换由 PWMCON 寄存器的 PWMFS 位控制。

7.3.5 PWM 的中断

PWM可产生定时中断。当TIMER1完成一个PWM周期时，可产生一个PWM中断，PIFB2的PWMIF中断标志位会置1。此时GIE与PEIE也为1，则会进入中断服务程序。

7.3.6 PWM 输出配置

以下是配置PWM输出的步骤示例：

1、端口配置：
配置PD5，PD7为输出口

2、配置周期：
配置TIMER1控制寄存器（T1STA），设定TIMER1为定时器模式，并设定预分频比
配置T1PRLH、T1PRLL（详见7.3.2 PWM的周期）

3、配置占空比：
配置占空比重载寄存器PRLxH，PRLxL

4、使能PWM：
设置PWM控制寄存器（PWMCON）,如果只要PWM

5、 使能TIMER1

置位TMR1ON，此时将会使PRLxH，PRLxL载入PWMxH，PWMxL

7.3.7 PWM 的寄存器定义

寄存器 1AH：PWM控制寄存器（PWMCON）

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
—	**PWMFS**	**PWM1OE**	**PWM0OE**	**PWM1EN**	**PWM0EN**	**TMR0EN**	**T1GSS**
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 **未实现位**

bit6 **PWMFS**：PWM高低频模式选择位
0 = PWM以低频模式输出
1 = PWM以高频模式输出
注：高频模式下比低频模式下PWM的驱动能力要小。

bit5 **PWM1OE:**PWM1引脚输出使能位
0 = PWM1禁止输出
1 = PWM1允许由PD5引脚输出

bit4 **PWM0OE**： PWM0引脚输出使能位
0 = PWM0 禁止输出
1 = PWM0 允许由 PD7 输出

bit3 **PWM1EN**： PWM1模块使能位
0 = PWM1 模块禁止
1 = PWM1 模块允许

bit2 **PWM0EN**：PWM0模块使能位
0 = PWM0 模块禁止
1 = PWM0 模块允许

bit1 **TMR0EN**：Timer0使能位
1 = Timer0打开
0 = Timer0关闭

bit0 **T1GSS**：TIMER1 门控源选择位。
1 = Timer1门控源为 T1G引脚（配置为数字输入）
0 = Timer1门控源为常开

寄存器 117H：PWM0/1周期寄存器的低字节（T1PRLL）

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
PWM0/1周期寄存器的低字节。							

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	**T1PRLL**：PWM0/1周期寄存器的低字节		

寄存器19AH：PWM0/1周期寄存器的高2位（T1PRLH）

R-0	R-0	R-0	R-0	R-0	R-0	R/W-0	R/W-0
bit7							bit0
—	—	—	—	**ET1PRH**		**T1PRLH**	

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 **未实现位**
bit3-2 **ET1PRH：** PWM0/1周期重载寄存器的高2位（只读位）
bit1-0 **T1PRLH：** PWM0/1周期寄存器的高2位。

寄存器118H：PWM0/1周期重载寄存器（ET1PRL）

R-0	R-0	R-0	R-0	R-0	R-0	R/W-0	R-0
bit7							bit0
ET1PRL							

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 **ET1PRL：** PWM0/1周期寄重载存器的低字节
注： 重载寄存器只能读，不能写。

寄存器 112H：PWM0占空比重载寄存器低字节（PRL0L）

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
PWM0占空比重载寄存器的低8位							

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

寄存器 192H：PWM0占空比比较寄存器低字节（PWM0L）

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-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 **PWM0L：** PWM0占空比比较寄存器低字节

寄存器 198H：PWM0占空比比较寄存器高2位与占空比重载寄存器高2位(EPR0)

R-0	R-0	R-0	R-0	R-0	R-0	R/W-0	R/W-0
bit7							bit0
				PWM0H		**PRL0H**	

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

bit3-2 **PWM0H：** PWM0 占空比比较寄存器高 2 位。
bit1-0 **PRL0H** ：PWM0占空比重载寄存器高2位。

寄存器 113H：PWM1占空比重载寄存器低字节（PRL1L）

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
PWM1占空比重载寄存器的低8位							

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

寄存器 193H：PWM1占空比比较寄存器低字节（PWM1L）

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-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　**PWM1L**：PWM1占空比比较寄存器低字节

寄存器 199H：PWM1占空比比较寄存器高2位与占空比重载寄存器高2位（EPR1）

R-0	R-0	R-0	R-0	R-0	R-0	R/W-0	R/W-0
bit7							bit0
				PWM1H		**PRL1H**	

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

bit3-2　**PWM1H**：PWM1 占空比比较寄存器高 2 位。
bit1-0　**PRL1H**：PWM1 占空比重载寄存器高 2 位。

表 7-3-6：与 PWM 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 LVR 时的值	所有其他复位值
PWMCON	—	PWMFS	PWM1OE	PWM0OE	PWM1EN	PWM0EN	TMR0EN	T1GSS	0000 0011	xxxx xxxx
INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000 0000	xxxx xxxx
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	0000 0000	xxxx xxxx
CPIOD	CPIOD7	CPIOD6	CPIOD5	CPIOD4	CPIOD3	CPIOD2	CPIOD1	CPIOD0	1111 1111	xxxx xxxx
PWM0L	PWM0 占空比寄存器低字节								0000 0000	xxxx xxxx
PWM1L	PWM1 占空比寄存器低字节								0000 0000	xxxx xxxx
T1PRLL	PWM0/1 周期寄存器低字节								0000 0000	xxxx xxxx
T1PRLH	—	—	—	—	ET1PRH		T1PRLH		0000 0000	xxxx xxxx
ET1PRL	PWM0/1 周期从动寄存器低字节								0000 0000	xxxx xxxx
PR0H	PWM0 占空比从动寄存器低字节								0000 0000	xxxx xxxx
PR1H	PWM1 占空比从动寄存器低字节								0000 0000	xxxx xxxx
PWM0H	—	—	—	—	EPR0H		PWM0H		0000 0000	xxxx xxxx
PWM1H	—	—	—	—	EPR1H		PWM1H		0000 0000	xxxx xxxx
TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx xxxx	xxxx xxxx
TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx xxxx	xxxx xxxx
T1STA	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	xxxx xxxx

图注：　x = 未知，　u = 不变，　— = 未实现（读为 0）。Timer1 模块不使用阴影单元。

8.0 模拟数字转换器（ADC）模块

模数转换器（Analog-to-digitalConverter，ADC）可将模拟输入信号转换为相应的 12 位二进制表征值。该器件采用多个模拟输入复用到一个采样保持电路。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 12 位二进制值，并将转换结果保存在 ADC 结果寄存器（ADRESL:ADRESH）中。为了保证转换结果的精度，器件内部设置了零点值校正寄存器。零点校正值是修正转换器转换模拟地的偏差值，在实际应用中，先进行对 VSS 采样 AD 值，并写入零点校准寄存器，器件内部硬件自动将 ADC 每次转换得到的值再减去零点校正值之后存入到 ADC 结果寄存器（ADRESL:ADRESH）中。零点校正值由 ADC0CN 寄存器的 ADCTHSEL 位来选择是 ADC0TH 寄存器，还是内部工厂校正值。

8.1 ADC 框图

8.2 ADC 的配置

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
- ADC 启动转换的方式
- ADC 中断控制
- 转换结果的对齐方式

8.2.1 端口配置

ADC 可用于模拟信号转换成数字信号。转换模拟信号时，应将相关的 CPIOD 和 ADINS 位置 1 将 I/O 引脚应配置为模拟输入功能。更多信息请参见相应的端口章节。

注： 如果定义为数字输入的引脚上存在模拟电压，可导致输入缓冲器传导过大的电流。

8.2.2 通道选择

ADS0 寄存器的 CHNSEL<3:0>位决定将哪个通道连接到采样保持电路。

改变通道时，开始下一次转换前需要一个延时。更多信息请参见第 8.3 节“ADC 的工作过程”。

8.2.3 ADC 参考电压选择

ADS0 寄存器的 REFSEL<1:0>位决定正电压参考的选择。正参考电压可以是：

- 选电压电压 VDD 做为 ADC 的参考。
- 选外部（来自 PD0）做为 ADC 的参考。
- 选内部 BANDGAP 的输出电压做为 ADC 的参考，这个电压可以通过配置寄存器选择 1.024V，2.048V。

负参考电压始终连接到参考地。

8.2.4 ADC 转换时钟

可用软件选择转换时钟源，通过设置 ADC0CN 寄存器的 SCx 位，有以下 4 种时钟选项：

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16

完成一位（bit）的转换时间定义为 T_{AD}。完成 12 位转换需要 17 个 T_{AD} 周期，如图 8-2-4 所示。

表 8-2-4 所示为正确选择 ADC 时钟的示例。

注： 任何系统时钟频率的变化均会改变ADC时钟频率，这将对ADC结果产生负面影响。

图 8-2-4：模数转换 Tad 周期

表 8-2-4：ADC 时钟周期（T_{AD}）—器件工作频率（Vdd≥3.0V）

ADC 时钟周期（T_{AD}）		器件频率（F_{OSC}）			
ADC 时钟源	**SCX<1:0>**	**16 MHz**	**8 MHz**	**4 MHz**	**1 MHz**
F_{OSC}/2	00	125 ns(1)	250 ns(1)	500 ns(1)	2.0 us
F_{OSC}/4	01	250 ns(1)	500 ns(1)	1.0 us	4.0 us
F_{OSC}/8	10	500 ns(2)	1.0 us	2.0 us	8.0 us(2)
F_{OSC}/16	11	1 us	2.0 us	4.0 us	16.0 us(2)

图注： 阴影单元超出了建议范围。

注 1：这些值违反了最小 T_{AD} 时间要求。

2：要加快转换，建议选择其他时钟源。

8.2.5 转换启动方式

有 3 种 A/D 转换启动方式，由 ADC0CN 中的 ADC0 转换启动方式位 CM[1:0]的状态决定采用哪一种方式。转换的触发源有：

- GO/$\overline{\text{DONE}}$写 1
- Timer0 溢出
- Timer1 溢出

注：ADC 转换进行中，即使新的转换启动条件有效，ADC 并不中断当前转换，硬件忽略新的转换启动。

8.2.6 ADC 模块中断

ADC 转换完成时，如果 ADC 中断 ADIE 使能并且全局中断标志 GIE 也使能的情况，将产生 ADC 中断，进入中断服务程序；在中断服务程序中必须软件清除中断标志 ADIF（清 0）。

8.2.7 数据对齐方式

ADC 转换结果可以设置右对齐或者左对齐，寄存器 ADC0CN 的 ADFM 位来决定。

8.3 ADC 的工作过程

8.3.1 启动转换

要使能 ADC 模块，必须将 ADS0 寄存器的 ADON 位置 1。将 ADS0 寄存器的 GO/$\overline{\text{DONE}}$位置 1 将启动模数转换。

注： 请参见第8.3.3节“A/D转换步骤”。

8.3.2 转换完成

转换完成时，ADC 模块将：

- 将 GO/$\overline{\text{DONE}}$位清零
- 将 ADIF 标志位置 1
- 用新的转换结果更新 ADRESH:ADRESL 寄存器

8.3.3 A/D 转换步骤

以下是使用 ADC 进行模数转换的步骤示例：

1. 配置端口：
   - 配置 IO 引脚为输入（见 CPIO 寄存器）
   - 将引脚配置为模拟口
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
   - 选择转换结果的格式
   - 打开 ADC 模块
3. 配置 ADC 中断（可选）：
   - 将 ADC 中断标志清零
   - 允许 ADC 中断
   - 允许外设中断
   - 允许全局中断(1)
4. 等待所需的采集时间(2)
5. 将 GO/$\overline{\text{DONE}}$ 置 1 启动转换（或等等待 TMR0、TMR1 溢出启动）
6. 通过以下情况之一等待 ADC 转换完成：
   - 查询 GO/$\overline{\text{DONE}}$ 位

- 等待 ADC 中断（允许中断时）

7. 读取 ADC 结果
8. 将 ADC 中断标志清零（必允许中断）。

例 8-3-4：A/D 转换

```
;-----------------------------------------------------------

BANK1
LDWI    00010000B
STWR    CPIOD
LDWI    00010000B          ;PD4 作为模拟输入（AN4）
STWR    ADINS              ;模拟口、数字口设定，1：模拟口，0：数字口

LDWI    01010001B
LCALL   GET_AD
LJUMP   $
;-----------------------------------------------------------
;ADC 转换子程序：
GET_AD:
BANK3
STWR    ADS0
BK3TOBK2
BSR     ADC_EN             ;开启 ADC 模块后需要延时
LDWI    20
LCALL   DELAY_US

BSR     GO                 ;启动 AD 转换
BTSC    BUSY
LJUMP   $-1                ;等待 AD 转换结束
BANK1
LDR     ADRESL,W
STWR    ADC_L
BK1TOBK0
LDR     ADRESH,W
STWR    ADC_H
BK0TOBK2
BCR     ADC_EN             ;关闭 ADC 转换模块，可以省电，
BK2TOBK0
GET_AD_EXIT:
RET
```

8.4 ADC 寄存器定义

8.4.1 ADC0CN

寄存器 11EH：A/D控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCTHSEL	**ADFM**	**SC1**	**SC0**	**ADC0EN**	**GO/$\overline{DONE}$**	**CM1**	**CM0**
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 **ADCTHSEL**：ADC零点校正值选择位
0 = ADC的零点校正值为内部工厂校正值
1 = ADC的零点校正值为ADC0TH

bit6 **ADFM**：ADC输出结果对齐方式选择位。
0 = 左对齐，低位补零
1 = 右对齐，高位补零

bit5-4 **SC<1:0>**：ADC时钟预分频选择位
00 = sys_clk 2分频
01 = sys_clk 4分频
10 = sys_clk 8分频
11 = sys_clk 16分频

bit3 **ADC_EN**：ADC使能位
0 = ADC禁止
1 = ADC使能

bit2 **GO/$\overline{\text{DONE}}$ (BUSY)**：ADC启动（忙）标志位
0 = 不忙
1 = 转换（忙）

bit1-0 **CM<1:0>**：ADC0启动转换模式选择，指定ADC启动转换源
0x = GO/$\overline{\text{DONE}}$写1。
10 = Timer0溢出
11 = Timer1溢出

8.4.2 ADRESH：ADC 结果高位寄存器

寄存器 1EH：ADC结果高位寄存器

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 ADC结果高字节寄存器位，
注：写数据动作使得ADRESH寄存器清零

8.4.3 ADRESL：ADC 结果低位寄存器

寄存器 9EH：ADC结果寄存器

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 ADC结果低字节寄存器位
注：写数据动作使得ADRESL寄存器清零

8.4.4 ADC0TH：ADC 零点值校正寄存器

寄存器9FH：ADC结果寄存器

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 ADC零点校正寄存器。

8.4.5 ADS0

寄存器 19DH：A/D控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R-1	R/W-1
CHNSEL3	**CHNSEL2**	**CHNSEL1**	**CHNSEL0**	**REFSL1**	**REFSL0**	—	**VREFSEL**
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

CHNSEL<3:0>: ADC模拟通道选择位

其他 = ADC 输入通道关断；

0001 = PD0 ADC 输入通道 0

0010 = PD1 ADC 输入通道 1

0011 = PD2 ADC 输入通道 2

0100 = PD3 ADC 输入通道 3

0101 = PD4 ADC 输入通道 4

0110 = PD5 ADC 输入通道 5

0111 = PD6 ADC 输入通道 6

1000 = PD7 ADC 输入通道 7

1110 = VSS ADC 输入电压为 VSS

1111 = VDD ADC 输入电压为 VDD/4

bit3-2 **REFSL<1:0>**：电压参考选择

00 = ADC 电压参考是 VDD pin。

01 = ADC 电压参考是 PD0/VREF pin。

1X = ADC 电压参考是内部电压参考。

bit1 未实现

bit0 **VREFSEL**：内部电压参考选择（1.024V/2.048V）

1 = 2.048V

0 = 1.024V

表 8-4：与 ADC 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 LVR 时的值	所有其他复位值
ADC0CN	ADCTHSEL	ADFM	SC1	SC0	ADCEN	GO/DONE	CM1	CM0	0000 0000	xxxx xxxx
ADINS	ANS7	ANS6	AN5	AN3	AN4	AN2	AN1	AN0	1111 1111	xxxx xxxx
ADS0	CHNSEL3	CHNSEL2	CHNSEL1	CHNSEL0	REFSL1	REFSL0	—	VREFSEL	0000 0100	xxxx xxxx
ADRESH	A/D 结果寄存器高字节								xxxx xxxx	xxxx xxxx
ADRESL	A/D 结果寄存器低字节								xxxx xxxx	xxxx xxxx
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	xxxx xxxx
PIFB1	INT1E	ADCIE	OSFIE	TMR1IE	INT1IF	ADCIF	OSFIF	TMRIF	0000 0000	xxxx xxxx
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	xxxx xxxx	xxxx xxxx
CPIOD	CPIOD7	CPIOD6	CPIOD5	CPIOD4	CPIOD3	CPIOD2	CPIOD1	CPIOD0	1111 1111	xxxx xxxx

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

9.0 I2C 模块

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

图9-1 给出了主模式下I2C 接口模块的框图。图9-2给出了从模式下I2C 接口模块的框图。

图 9-1 I2C 主模式接口

图 9-2　I2C 从模式接口

9.1 I2C 概述

I2C 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。

I2C 总线规定了两种信号连接：

- 串行时钟（SCL）
- 串行数据（SDA）

图9-3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9-3 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。I2C 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。

图9-3： I2C 主/ 从器件连接

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。

I2C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。

9.1.1 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。

9.1.2 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。

9.2 I2C 模式操作

所有 MSSP1 I2C 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 I2C 器件进行通信

9.2.1 字节格式

I2C 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。

9.2.2 I2C 术语定义

在 I2C 通信的描述中存在一些用语和术语，它们具有特定于 I2C 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。该表根据 Phillips I2C™ 规范改写。

表 9-1： 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/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/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。

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

9.2.3 启动和停止条件

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 9-4 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。

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。

图9-4： I2C 启动和停止条件

9.2.4 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。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。

图 9-5： I2C 重复启动条件

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。

9.2.5 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。

9.3 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。

9.3.1 从模式地址

SSP1ADD 寄存器包含从模式地址。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 掩码寄存器会影响地址匹配过程。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。

在 10 位寻址模式下，接收到的第一个字节将与二进制值“11110A9A80”进行比较。A9 和 A8 是 10 位地址的两个 MSB，存储在 SSP1ADD 寄存器的 bit2 和 bit1 中。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。

9.3.2 从模式接收

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。请参见（寄存器 11H）

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12H 、寄存器14H)。

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时请参见寄存器13H)，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。

9.3.2.1 7 位寻址接收

本节介绍在7 位寻址模式下，配置为I2C 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9-5和图9-6用直观的方式对此作了说明。

以下列出了实现I2C 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。

9.3.2.2 7 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9-7 显示了同时使用地址和数据保持功能的模块。图9-8 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。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9-6： I2C从模式， 7 位地址，接收（SEN = 0， AHEN = 0， DHEN = 0）

图9-7：I2C从模式， 7位地址，接收（SEN = 1， AHEN = 0， DHEN = 0）

图9-8： I2C 从模式， 7位地址，接收（SEN = 0， AHEN = 1， DHEN = 1）

图9-9： I2C 从模式， 7 位地址，接收（SEN = 1， AHEN = 1， DHEN = 1）

9.3.3 从模式发送

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9.3.4 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。

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，PIFB3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。

9.3.3.1 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 9-9 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 9-10： I2C 从模式， 7 位地址，发送（AHEN = 0）

9.3.3.2 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9-10 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 9-11： I2C 从模式， 7 位地址，发送（AHEN = 1）

9.3.3.2 从模式 10 位地址接收

本节介绍在 10 位寻址模式下，配置为 I2C 从器件的 MSSP1 模块的标准事件序列。

图 9-11 用直观的方式对此作了说明。

以下列出了实现I2C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。

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9-12 可以用作AHEN 置1 时10 位寻址模式下从器件的参考图示。图9-13给出了10位寻址模式下从发送器的标准波形图。

图 9-12： I2C 从模式， 10 位地址，接收（SEN = 1， AHEN = 0， DHEN = 0）

图 9-13： I2C 从模式， 10 位地址，接收（SEN = 0， AHEN = 1， DHEN = 0）

图9-14： I2C 从模式， 10 位地址，发送（SEN = 0， AHEN = 0， DHEN = 0）

9.3.4 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。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。

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。

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。

9.3.5 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 I2C 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 I2C 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 9-14）。

图 9-15： 时钟同步时序

9.3.6 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 9-15 显示了广播呼叫接收

图 9-16： 从模式广播呼叫地址序列

SSP1 掩码（SSP1MSK）寄存器（寄存器 15H）在 I2C 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。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。

9.4 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 I2C 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 I2C 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，在启动条件结束前，不允许用户发出启动条件以及立即写 SSP1BUF 寄存器以启动传输。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

9.4.1 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 I2C 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/W）位。在这种情况下，R/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/W 位。在这种情况下， R/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”。

9.4.2 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9-17）。

图9-17： 带有时钟仲裁的波特率发生器时序

9.4.3 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。

9.4.5 I2C 主模式启动条件时序

要发出启动条件，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（TBRG）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 SSP1STAT1 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（TBRG）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，I2C 模块复位到空闲状态。

2：Philips I2C 规范规定启动时不能发生总线冲突。

图 9-18： 第一个启动位时序

9.4.6 I2C 主模式重复启动条件时序

当SSP1CON2 寄存器的RSEN 位设置为一，并且主器件状态机不再有效时，会产生重复启动条件。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（TBRG）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TBRG 内采样为高电平。接下来，在一个TBRG 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。

9.4.7 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（TBRG）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TBRG 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。

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。

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。

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。

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。

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9-19： I2C 主模式波形图（发送， 7 位或10 位地址）

9.4.8 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。

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。

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。

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。

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）。

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9-20： I2C 主模式波形图（接收， 7 位地址）

9.4.9 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（TBRG）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 TBRG 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 9-20）。

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）。

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TBRG （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9-21）。

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）。

图9-21： 应答序列波形图

图9-22： 停止条件接收或发送模式

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。复位会禁止MSSP1 模块并终止当前的数据传输。 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

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I2C 端口复位到空闲状态（图9-22）。

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I2C 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。

图9-23： 发送和应答时的总线冲突时序

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9-23）。

b) SDA 被置为低电平之前， SCL 采样为低电平。

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

•MSSP1 模块复位为空闲状态（图9-23）。

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。

图9-24： 启动条件期间的总线冲突（SCL = 0）

图9-25： 启动条件期间由SDA 仲裁引起的BRG 复位

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。

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。

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9-25）。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9-26）。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9-26： 重复启动条件期间的总线冲突（情形**1**）

图9-27： 重复启动条件期间的总线冲突（情形2）

以下事件会导致停止条件期间发生总线冲突。

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。

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。

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9-27）。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9-28）

图9-28： 停止条件期间的总线冲突（情形1）

图9-29： 停止条件期间的总线冲突（情形2）

9.5 波特率发生器

MSSP1 模块具有一个波特率发生器，可用于在I2C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16H）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9-29 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9-2 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9-1

$$FCLOCK = \frac{FOSC}{(SSPxADD+1)(4)}$$

图 9-30

表 9-2

Fosc	Fcy	BRG 值	FCLOCK（两次 BRG 计满返回）
16MHZ	4MHZ	09h	400KHZ
16MHZ	4MHZ	0Ch	308KHZ
16MHZ	4MHZ	27h	100KHZ

9.6 寄存器定义

寄存器11H：I2C状态寄存器 < SSP1STAT >

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	**D/A**	**P**	**S**	**R/W**	**UA**	**BF**
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 **未实现**

bit5 **D/A**：数据/地址位
1 = 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
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
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/W**：读/ 写位信息
该位保存上一次地址匹配后的R/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 **U/A**：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
I2C 接收模式
1 = 接收完成， SSP1BUF 已满
0 = 接收未完成， SSP1BUF 为空
I2C 发送模式
1 = 数据发送正在进行（不包括 ACK 位和停止位）， SSP1BUF 已满
0 = 数据发送完成（不包括 ACK 位和停止位）， SSP1BUF 为空

寄存器12H：I2C控制寄存器1 < SSP1CON1 >

R/C-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	**SSP1OV**	**SSP1EN**	**CKP**	**SSP1M3**	**SSP1M2**	**SSP1M1**	**SSP1M0**
bit7							bit0

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

-n = POR时的值

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)
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 = 无溢出

bit5 **SSP1EN**：同步串口使能位
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
I2C从模式：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。）
I2C 主模式：此位未用

bit3-0 **SSP1M<3:0>**：同步串口模式选择位
0110 = I2C 从模式， 7 位地址
0111 = I2C 从模式， 10 位地址
1000 = I2C 主模式，时钟 = 16MHz / (4 * (SSP1ADD+1))(4)
1110 = I2C 从模式， 7 位地址，并允许启动位和停止位中断
1111 = I2C 从模式， 10 位地址，并允许启动位和停止位中断
其他值 保留

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 寄存器启动的。
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I2C 模式，不支持SSP1ADD 值0、1 或2。

寄存器13H：I2C控制寄存器2 < SSP1CON2 >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	**ACKSTAT**	**ACKDT**	**ACKEN**	**RCEN**	**PEN**	**RSEN**	**SEN**
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 **GCEN**：广播呼叫使能位（仅限 I2C 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I2C 模式）：
1 = 未接收到应答
0 = 接收到应答

bit5 **ACKDT**：应答数据位（仅限 I2C 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
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I2C 主模式）
1 = 使能I2C 接收模式
0 = 接收空闲

bit2 **PEN**：停止条件使能位（仅限I2C 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I2C 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**：缓冲区满状态位
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

寄存器14H：I2C控制寄存器3 < SSP1CON3 >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACKTIM	**PCIE**	**SCIE**	**BOEN**	—	**SBCDE**	**AHEN**	**DHEN**
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 **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
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
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
在主接收模式下该位被忽略
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 **未实现**

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PIFB3 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15H：I2C掩码寄存器 < SSP1MSK >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MSK7	**MSK6**	**MSK5**	**MSK4**	**MSK3**	**MSK2**	**MSK1**	**MSK0**
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

bit7-1 **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16H：I2C地址和波特率寄存器 < SSP1ADD >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD7	**ADD6**	**ADD5**	**ADD4**	**ADD3**	**ADD2**	**ADD1**	**ADD0**
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 **主模式：**
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：

bit 7-3 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。

bit 2-1 ADD<2:1>：10 位地址的高2 位

bit 0　　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：

bit 7-0　　ADD<7:0>：10 位地址的低 8 位

7 位从模式：

bit 7-1　　ADD<7:1>：7 位地址

bit 0　　未使用：在此模式下未使用。位状态为“ 无关”。

寄存器17H：I2C数据寄存器　< 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
SSP1BUF<7:0>							
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　　**SSP1BUF<7:0>**：I2C 发送/接收缓冲寄存器

主模式下：　发送数据缓冲寄存器

从模式下：　接收数据缓冲寄存器

表 9-6-1：与 I2C 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 LVR 时的值	所有其他复位值
INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000 0000	xxxx xxxx
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	0000 0000	xxxx xxxx
CPIOB	CPIOB7	CPIOB6	CPIOB5	CPIOB4	CPIOB3	CPIOB2	CPIOB1	CPIOB0	1111 1111	xxxx xxxx
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--00 -000	xxxx xxxx
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 -111	xxxx xxxx
SSP1STAT	—	—	D/A	P	S	R/W	UA	BF	0000 0000	xxxx xxxx
SSP1CON1	WCOL	SSP1OV	SSP1EN	CKP	SSP1M3	SSP1M2	SSP1M1	SSP1M0	0000 0000	xxxx xxxx
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	xxxx xxxx
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	—	SBCDE	AHEN	DHEN	0000 0000	xxxx xxxx
SSP1MSK	掩码寄存器								0000 0000	xxxx xxxx
SSP1ADD	波特率与地址寄存器								0000 0000	xxxx xxxx

图注：　x = 未知，　u = 不变，　— = 未实现（读为 0）。Timer1 模块不使用阴影单元。

10.0 电容触摸模块

对于YS83B08/YS84B08/YS84C12， 触摸模块最多可监视16个输入（CHSEL<15:0>）；要使用触摸功能，用户必须：

- 通过设置CHSELH跟CHSELL寄存器选择引脚为触摸功能。
- 将相应的CPIO位置1。
- 设置触摸相关参数（触摸灵敏度、初始化环境值、扫描时间等，详细说明见下文。）运行软件算法

10.1 原理说明

YS83B08/YS84B08/YS84C12 主要检测的是电容充电和放电的时间。首先，由一个 GPIO 对电容 Cx 进行充电；同时开启计时器进行计时；随着充电的进行，Cx 的电压中不断升高，最终它将会操作某个门限电压 V, 当其超过门限电压 V 后，GPIO 将会检测到这个事件，同时停止计时器并读出此时的数值。这样，就完成了一次充电计时过程，当手指接触到触摸按键时，Cx 将会变大，显然，充电时间也会变长。通过不断比较每次充电的时间，很自然地就能得知当前是否有按键被“按下”。

10.2 配置说明

要使用触摸功能，必须先配置触摸门限差值以及触摸门限初值，配置{CSTMRH、CSTMRL}设定帧频率，配置{CHSELH、CSSELL}选择触摸通道，配置 CSACU[2:0]选择通道扫描帧次数，配置 THCTL 选择工作模式，配置完成后打开触摸使能开关 CSEN，触摸模块开始工作。

10.3 电容感应数据区

下表列出每个通道电容感应的相关数据：

感应 通道编号	感应电容值 (16’b) L : H	门限值 (16’b) L : H	门限差 (12’b)L :H(bit3~0)	门限累加值 (20’b) L(bit7~4) : M : H
第 0 通道	0X1C1 : 0X1C0	0X1A1 : 0X1A0	0X3A1 : 0X3A0	0X3A0 : 0X3C1 : 0X3C0
第 1 通道	0X1C3 : 0X1C2	0X1A3 : 0X1A2	0X3A3 : 0X3A2	0X3A2 : 0X3C3 : 0X3C2
第 2 通道	0X1C5 : 0X1C4	0X1A5 : 0X1A4	0X3A5 : 0X3A4	0X3A4 : 0X3C5 : 0X3C4
第 3 通道	0X1C7 : 0X1C6	0X1A7 : 0X1A6	0X3A7 : 0X3A6	0X3A6 : 0X3C7 : 0X3C6
第 4 通道	0X1C9 : 0X1C8	0X1A9 : 0X1A8	0X3A9 : 0X3A8	0X3A8 : 0X3C9 : 0X3C8
第 5 通道	0X1CB : 0X1CA	0X1AB : 0X1AA	0X3AB : 0X3AA	0X3AA : 0X3CB : 0X3CA
第 6 通道	0X1CD : 0X1CC	0X1AD : 0X1AC	0X3AD : 0X3AC	0X3AC : 0X3CD : 0X3CC
第 7 通道	0X1CF : 0X1CE	0X1AF : 0X1AE	0X3AF : 0X3AE	0X3AE : 0X3CF : 0X3CE
第 8 通道	0X1D1 : 0X1D0	0X1B1 : 0X1B0	0X3B1 : 0X3B0	0X3B0 : 0X3D1 : 0X3D0

第 9 通道	0X1D3 : 0X1D2	0X1B3 : 0X1B2	0X3B3 : 0X3B2	0X3B2 : 0X3D3 : 0X3D2
第 10 通道	0X1D5 : 0X1D4	0X1B5 : 0X1B4	0X3B5 : 0X3B4	0X3B4 : 0X3D5 : 0X3D4
第 11 通道	0X1D7 : 0X1D6	0X1B7 : 0X1B6	0X3B7 : 0X3B6	0X3B6 : 0X3D7 : 0X3D6
第 12 通道	0X1D9 : 0X1D8	0X1B9 : 0X1B8	0X3B9 : 0X3B8	0X3B8 : 0X3D9 : 0X3D8
第 13 通道	0X1DB : 0X1DA	0X1BB : 0X1BA	0X3BB : 0X3BA	0X3BA : 0X3DB : 0X3DA
第 14 通道	0X1DD : 0X1DC	0X1BD : 0X1BC	0X3BD : 0X3BC	0X3BC : 0X3DD : 0X3DC
第 15 通道	0X1DF : 0X1DE	0X1BF : 0X1BE	0X3BF : 0X3BE	0X3BE : 0X3DF : 0X3DE

10.4 寄存器说明

寄存器190H：触摸控制存器 < CSCTL >

R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-1	R/W-1
CSEN	**RAMEN**	**XRAM**	**CSBUSY**	**CSINTS**	**CSACU2**	**CSACU1**	**CSACU0**
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 **CSEN**：触摸模块使能位
1 = 触摸模块打开
0 = 触摸模块关闭

bit6 **RAMEN**：CS RAM控制选择位
1 = CS RAM区域由MCU控制
0 = CS RAM区域由触摸模块控制（XRAM位无效）

bit5 **XRAM**：MCU 访问扩展区域 SRAM（RAMEN 为 1 有效）
1 = 访问扩展区域 (访问3A0H ~ 3DFH)
0 = 正常访问（访问1A0H ~ 1DFH）

bit4 **CSBUSY**：触摸扫描标志位
1 = 触摸扫描正在进行
0 = 触摸空闲

bit3 **CSINTS**：触摸产生方式选择
1 = 有触摸按键产生中断
0 = 每帧扫描完产生中断

bit2-0 **CSACU <2:0>**：通道每帧扫描次数
000 = 1
001 = 2
010 = 4
011 = 8 （缺省值）
100 = 16
101 = 32
110 = 64
111 = 128

寄存器191H：触摸门限自动更新寄存器 < THCTL>

R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1
THFIB	**THGAP1**	**THGAP0**	**THMOD2**	**THMOD1**	**THMOD0**	**THDS1**	**THDS0**
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 **未使用，请写0**

bit6-5 **THGAP<1: 0>**：门限抽取间隔帧数
00 = 3
01 = 7 （缺省值）
10 = 15
11 = 31

bit4-2 **THMOD<2:0>**：门限累加次数
000 = 2
001 = 4
010 = 8 （缺省值）
011 = 16
100 = 32
101 = 64
110 = 128
111 = 256

bit1-0 **THDS<1:0>**：在触摸按键释放后门限更新累加所需间隔的帧数
00 = 16
01 = 32 （缺省值）
10 = 64
11 = 128

寄存器194H：触摸通道结果标志高位寄存器 < CHRLTH >

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHRLT15	**CHRLT14**	**CHRLT13**	**CHRLT12**	**CHRLT11**	**CHRLT10**	**CHRLT9**	**CHRLT8**
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 **CHRLT<15:8>**：触摸通道结果标志
1 =对应通道有触摸
0 =对应通道没有触摸

寄存器195H：触摸通道结果标志低位寄存器 < CHRLTL >

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHRLT7	**CHRLT6**	**CHRLT5**	**CHRLT4**	**CHRLT3**	**CHRLT2**	**CHRLT1**	**CHRLT0**
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 **CHRLT<7:0>：**触摸通道结果标志
1 =对应通道有触摸
0 =对应通道没有触摸

寄存器196H：触摸通道掩码高位寄存器 < CHSELH >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHSEL15	**CHSEL14**	**CHSEL13**	**CHSEL12**	**CHSEL11**	**CHSEL10**	**CHSEL9**	**CHSEL8**
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 **CHSEL<15:8>：**允许/禁止该通道为触摸通道
1 = 对应通道配置成触摸通道
0 = 对应通道没有配置成触摸通道

寄存器197H：触摸通道掩码低位寄存器 < CHSELL >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHSEL7	**CHSEL6**	**CHSEL5**	**CHSEL4**	**CHSEL3**	**CHSEL2**	**CHSEL1**	**CHSEL0**
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 **CHSEL<7:0>：**允许/禁止该通道为触摸通道
1 = 对应通道配置成触摸通道
0 = 对应通道没有配置成触摸通道

寄存器19BH：触摸自动扫描定时器寄存器高8位 < CSTMRH >

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSTMR15	**CSTMR14**	**CSTMR13**	**CSTMR12**	**CSTMR11**	**CSTMR10**	**CSTMR9**	**CSTMR8**
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 **CSTMR<15:8>：**触摸扫描周期计数器高 8 位

寄存器19CH：触摸自动扫描定时器寄存器低8位 < CSTMRL >

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CSTMR7	**CSTMR6**	**CSTMR5**	**CSTMR4**	**CSTMR3**	**CSTMR2**	**CSTMR1**	**CSTMR0**
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 **CSTMR<7:0>：**触摸扫描周期计数器低 8 位

寄存器19EH：触摸灵敏度调节寄存器0 < CSANA0 >

U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
—	**BSTRM2**	**BSTRM1**	**BSTRM0**	**CMPTRM1**	**CMPTRM0**	**VTSEL1**	**VTSEL0**
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 未实现位

bit6-4 **BSTRM<2: 0>**：BIAS输出电流调节
000 = 3.5uA
001 = 3.75uA
010 = 4uA （缺省值）
011 = 4.25uA
100 = 4.5uA
101 = 5uA
110 = 5.5uA
111 = 6uA

bit3-2 **CMPTRM <1:0>**：触摸内部比较器消耗电流
00 = 6uA
01 = 12uA （缺省值）
10 = 18uA
11 = 24uA

bit1-0 **VTSEL <1:0>**：触摸内部比较高电平选择
00 = 0.65V
01 = 1.3V
10 = 1.9V
11 = 2.55V （缺省值）

寄存器19FH：触摸灵敏度调节寄存器1 < CSANA1 >

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-1
—	—	—	**DTTRM1**	**DTTRM0**	**CTTRM2**	**CTTRM1**	**CTTRM0**
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

Bit4-3 **DTTRM <1: 0>**：触摸放电时间调节
00 = 0.75us
01 = 1us （缺省值）
10 = 1.25us
11 = 1.5us

Bit2-0 **CTTRM <2:0>**：触摸充电电流调节，用于调节片外触摸电容量程，提高触摸精度
000 = 0.25uA (量程：0 ~ 256pF)
001 = 0.5uA (量程：0 ~ 128pF)
010 = 0.75uA (量程：0 ~ 96pF)
011 = 1uA (量程：0 ~ 64pF) （缺省值）
100 = 1.25uA (量程：0 ~ 56pF)
101 = 1.5uA (量程：0 ~ 48pF)
110 = 1.75uA (量程：0 ~ 40pF)
111 = 2uA (量程：0 ~ 32pF)

表 10-4-1：与触摸相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 LVR 时的值	所有其他复位值
INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PIFB2	SSP1IE	CSIE	PBIE	PWMIE	SSP1IF	CSIF	PBIF	PWMIF	0000 0000	xxxx xxxx
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	0000 0000	xxxx xxxx
CPIOB	CPIOB7	CPIOB6	CPIOB5	CPIOB4	CPIOB3	CPIOB2	CPIOB1	CPIOB0	1111 1111	xxxx xxxx
PORTC	—	—	—	—	PC3	PC2	PC1	PC0	---- 0000	xxxx xxxx
CPIOC	—	—	—	—	CPIOC3	CPIOC2	CPIOC1	CPIOC0	---- 1111	xxxx xxxx
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	0000 0000	xxxx xxxx
CPIOD	CPIOD7	CPIOD6	CPIOD5	CPIOD4	CPIOD3	CPIOD2	CPIOD1	CPIOD0	1111 1111	xxxx xxxx
ADINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	xxxx xxxx
CSCTL	CSEN	RAMEN	XRAM	CSBUSY	CSINTS	CSACU2	CSACU1	CSACU0	0000 0000	xxxx xxxx
THCTL	THFIB	THGAP1	THGAP0	THMOD2	THMOD1	THMOD0	THDS1	THDS0	0000 0000	xxxx xxxx
CHRLTH	CHRLT15	CHRLT14	CHRLT13	CHRLT12	CHRLT11	CHRLT10	CHRLT9	CHRLT8	0000 0000	xxxx xxxx
CHRLTL	CHRLT7	CHRLT6	CHRLT5	CHRLT4	CHRLT3	CHRLT2	CHRLT1	CHRLT0	0000 0000	xxxx xxxx
CHSELH	CHSEL15	CHSEL14	CHSEL13	CHSEL12	CHSEL11	CHSEL10	CHSEL9	CHSEL8	0000 0000	xxxx xxxx
CHSELL	CHSEL7	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	0000 0000	xxxx xxxx
CSTMRH	触摸扫描周期计数器高 8 位								0000 0000	xxxx xxxx
CSTMRL	触摸扫描周期计数器低 8 位								0000 0000	xxxx xxxx
CSANA0	—	BSTRM2	BSTRM1	BSTRM0	CMPTRM1	CMPTRM1	VTSEL1	VTSEL0	0000 0000	xxxx xxxx
CSANA1	—	—	—	DTTRM1	DTTRM0	CTTRM2	CTTRM1	CTTRM0	0000 0000	xxxx xxxx

图注： x = 未知， u = 不变， — = 未实现（读为 0）。

11.0 配置寄存器

配置寄存器是用来配置 MCU 的设置，其原始值存在 MTP 的某些位置，在 MCU 启动时从 MTP Boot 到配置寄存器中，配置寄存器位宽为 8 位，由于 Flash 的位宽为 14 位，Boot 时，只 Boot 低 8 位至配置寄存器。

11.1 Config Option

Config Option 用于给客户提供配置选项。

寄存器 801H：配置字寄存器高8位（Config1 Option）

—	RD_CTRL	LVDS1	LVDS0	FCMEN	IESO	LVREN1	LVREN0
bit15							bit8

寄存器 800H：配置字寄存器低8位（Config0 Option）

—	CPB	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit7							bit0

bit15 未实现：保留

bit14 **RD_CTRL**：输出模式时读端口控制
1：读数据端口返回的PIN上的值
0：读数据端口返回的锁存器上的值

bit13-12 **LVDS <1:0>**：低电压复位电压设定
00：1.9V
01：2.5V
10：3.8V
11：3.8V

bit11 **FCMEN**：系统时钟失效监测使能位
1：使能系统时钟失效监测，仅在系统时钟为外部晶振模式才起作用
0：禁止系统时钟失效监测

bit10 **IESO**：内、外时钟切换使能位，二速启动模式使能，
1：使能二速启动模式，仅在系统时钟为外部晶振模式才起作用
0：禁止二速启动模式

bit9-8 **LVREN <1:0>**：低电压复位选择(1)
11：低电压复位使能
10：MCU工作时，低电压复位开启；MCU睡眠时，低电压复位关闭
01：低电压复位由PCON之SLVREN位决定
00：禁止低电压复位

bit7 未实现：保留

bit6 **CPB**：代码保护位(3)
1：Flash 内容不保护
0：启动Flash内容保护，MCU能读，串口不能读

bit5 **MCLRE**：PA3/$\overline{\text{MCLR}}$引脚功能选择位(4)
1：PA3/$\overline{\text{MCLR}}$脚执行MCLR功能，是复位脚
0：PA3/$\overline{\text{MCLR}}$脚执行PA3功能，是数字输入引脚

bit4 **$\overline{\text{PWRTE}}$**：上电延时定时器使能位
1：PWRT禁止
0：PWRT使能

bit3 **WDTE**：看门狗定时器使能位
1：WDT使能，程序不能禁止
0：WDT禁止，但程序可通过设置WDTCON的SWDTEN位将WDT使能

bit2-0 **FOSC <2:0>**：振荡器选择位
111：INTOSCIO Oscillator模式，PA4为IO引脚，PA5为IO引脚
110：RCIO Oscillator模式，PA4为IO引脚，PA5时钟输入脚CLKIN
101：INTOSC Oscillator模式，PA4输出时钟CLKOUT，PA5为IO引脚
100：RC Oscillator模式，PA4输出时钟CLKOUT，PA5是时钟输入脚CLKIN
011：EC模式，PA4为IO引脚，PA5时钟输入脚CLKIN
010：HS Oscillator模式，PA4、PA5接高频晶振
001：XT Oscillator模式，PA4、PA5接晶振
000：LP Oscillator模式，PA4、PA5接低频晶振

注 1：使能欠压复位不会自动使能上电延时定时器。
2：在INTOSC或RC模式下当$\overline{\text{MCLR}}$有效时，内部时钟振荡器被禁止。

12.0 指令表

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
LCALL 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

13.0 电气特性

13.1 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.....120mA
流入 VDD 引脚的最大电流.....120mA
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

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。

图 13-1-1：YS83B08/YS84B08/YS84C12 外部振荡电压－频率图，-40℃ ≤ TA ≤ +85℃

图 13-1-2：器件 VDD 和工作温度范围内 IRC 偏移量（左边参数为偏移百分比，如 0.01 为 1%）

13.2 直流电气特性

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)	—	680	—	uA	3.3V	1M	WDT En LVR Dis IRC
			996				4M	
			1400				8M	
			800			5V	1M	
			1170				4M	
			1668				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		
		1.6	——	VDD		3V SCHMITT		
		1.6	——	VDD		5V TLL		
		3.0	——	VDD		5V SCHMITT		
IOL	输出灌电流	—	—	—	mA	VOL=0.7V	3V	
		—	18	—		VOL=0.3V		
		—	—	—		VOL=0.9V	5V	
		—	40	—		VOL=0.5V		

IOH	输出拉电流	—	—	—	mA	VOH=2.5V	3V
		—	7	—		VOH=2.7V	
		—	—	—		VOH=4.0V	5V
		—	15	—		VOH=4.5V	
VLVR	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%		MIDDLE	
		3.8 -20%	3.8	3.8 +20%		HIGH	
Rpu	上拉电阻	—	—	—	K	3V	
		—	31	—		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。

13.3 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
F_{OSC}	系统时钟	0	8M	20M	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
	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
TLVR	VDD 降低到 VLVR 以下持续时间	2	5	——	us	——

14.0 开发支持

14.1 仿真信息

14.1.1 软件：MDT-IDE

14.1.2 硬件：YS-Link

14.1.3 接口：

YS83B08/YS84B08/YS84C12 具备片内仿真功能，可通过串口模式仿真调试程序，串口接口如下：

YS83B08	YS84B08/YS84C12	YS-Link
VDD	VDD	VDD
GND	GND	GND
PA3	PA3	VPP
PB1	PA0	SCK
PB2	PA1	SDA

14.2 烧录信息

14.2.1 烧录软件：YSpringPro

14.2.2 烧录器：YS-Writer

烧录使用引脚：VDD、VSS、VPP、PA0、PA1、PA2 (Busy 信号)

注：YS83B08 引脚：VDD、GND、VPP、PB1、PB2、PA2(Busy 信号)

15.0 封装信息

15.1 SOP16

SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.70	3.90	4.10
A1	0.05	—	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	9.70	9.90	10.10	L1	1.05 BSC		
E	5.80	6.00	6.20	θ	0	—	8°

15.2 SOP20

SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	2.65	E1	7.40	7.50	7.60
A1	0.10	—	0.30	e	1.27BSC		
A2	2.25	2.30	2.35	h	0.25	—	0.75
A3	0.97	1.02	1.07	L	0.70	—	1.00
D	12.70	12.80	12.90	L1	1.40REF		
E	10.10	10.30	10.50	θ	0	—	8°
				b	0.35	—	0.43

15.3 SSOP20

SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	NOR
	(mm)				(mm)		
A	—	—	1.75	E1	3.80	3.90	4.00
A1	0.10	0.15	0.25	e	0.635BSC		
A2	1.30	1.40	1.50	c	0.20	—	0.24
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	8.55	8.65	8.75	L1	1.05REF		
h	0.30	—	0.50	c1	0.19	0.20	0.21
E	5.80	6.00	6.20	θ	0	—	8º
				b	0.23	—	0.31

15.4 SOP 24

SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	2.36	2.54	2.64	E1	7.40	7.50	7.60
A1	0.10	0.20	0.30	e	1.27BSC		
A2	2.26	2.30	2.35	h	0.25	—	0.75
A3	0.97	1.02	1.07	L	0.70	—	1.00
D	15.30	15.40	15.50	L1	1.40REF		
E	10.10	10.30	10.50	θ	0	—	8°
				b	0.39	—	047

15.5 SSOP24

SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	NOR
	(mm)				(mm)		
A	—	—	1.75	E1	3.80	3.90	4.00
A1	0.10	0.15	0.25	e	0.635BSC		
A2	1.30	1.40	1.50	c	0.20	—	0.24
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	8.55	8.65	8.75	L1	1.05REF		
h	0.30	—	0.50	c1	0.19	0.20	0.21
E	5.80	6.00	6.20	θ	0	—	8º
				b	0.23	—	0.31

15.6 SOP 28

SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	NOR
	(mm)				(mm)		
A	—	—	2.65	E1	7.40	7.50	7.60
A1	0.10	—	0.30	e	1.27BSC		
A2	2.25	2.30	2.35	c	0.25	—	0.29
A3	0.97	1.02	1.07	L	0.70	—	1.00
D	17.90	18.00	18.10	L1	1.40REF		
E	10.10	10.30	10.50	θ	0	—	8º
				b	0.39	—	0.47

15.7 SSOP 28

SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	NOR
	(mm)				(mm)		
A	—	—	1.75	E1	3.80	3.90	4.00
A1	0.05	—	0.225	e	0.635BSC		
A2	1.30	1.40	1.50	c	0.20	—	0.24
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	9.80	9.90	10.00	L1	1.05REF		
h	0.25	—	0.50	c1	0.19	0.20	0.21
E	5.80	6.00	6.20	θ	0	—	8º
				b	0.23	—	0.31

16.0 汇春知识产权政策

16.1 专利权

汇春公司在全球各地区已核准和申请中之专利权至少有 30 件以上，享有绝对之合法权益。与汇春公司 MCU 或其它产品有关的专利权并未被同意授权使用，任何经由不当手段侵害汇春公司专利权之公司、组织或个人，汇春将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨汇春公司因侵权行为所受之损失、或侵权者所得之不法利益。

16.2 著作权

Copyright 2017 by INC.

规格书中所出现的信息在出版当时相信是正确的，然而汇春对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，汇春不保证或不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。汇春产品不授权使用于救生、维生器件或系统中做为关键器件。汇春拥有不事先通知而修改产品的权利，对于最新的信息，请从我们官方网站中获取，网址：

http://www.yspringtech.com;