

采用 XLP 技术的全功能低引脚数单片机

说明

PIC16(L)F18324/18344 单片机具有模拟外设、独立于内核的外设和通信外设，结合超低功耗（eXtreme Low Power, XLP）技术，适合各种通用和低功耗应用。外设引脚选择（Peripheral Pin Select, PPS）功能支持在使用数字外设（CLC、CWG、CCP、PWM 和通信）时进行引脚映射，提高了应用设计灵活性。

内核特性

- C 编译器优化的 RISC 架构
- 仅有 48 条指令
- 工作速度：
 - DC – 32 MHz 时钟输入
 - 最小指令周期为 125 ns
- 中断功能
- 16 级深硬件堆栈
- 最多 4 个 8 位定时器
- 最多 3 个 16 位定时器
- 低电流上电复位（Power-on Reset, POR）
- 上电延时定时器（Power-up Timer, PWRT）
- 欠压复位（Brown-out Reset, BOR）
- 低功耗 BOR（Low-Power BOR, LPBOR）选项
- 扩展型看门狗定时器（Watchdog Timer, WDT），采用专用片上振荡器，确保运行可靠
- 可编程代码保护

存储器

- 7 KB 闪存程序存储器
- 512B 数据 SRAM 存储器
- 256B EEPROM
- 直接、间接和相对寻址模式

工作特性

- 工作电压范围：
 - 1.8V 至 3.6V（PIC16LF18324/18344）
 - 2.3V 至 5.5V（PIC16F18324/18344）
- 温度范围：
 - 工业级：-40°C 至 85°C
 - 扩展级：-40°C 至 125°C

超低功耗（XLP）特性

- 休眠模式：40 nA（1.8V 时，典型值）
- 看门狗定时器：250 nA（1.8V 时，典型值）
- 辅助振荡器：300 nA（32 kHz 时）
- 工作电流：
 - 8 μ A（32 kHz、1.8V 时，典型值）：
 - 37 μ A/MHz（1.8V 时，典型值）

节能功能

- 空闲模式：能够使 CPU 内核进入休眠模式，而内部外设依靠系统时钟继续工作
- 打盹模式：能够使 CPU 内核工作速度慢于内部外设所使用的系统时钟
- 休眠模式：最低功耗
- 外设模块禁止（Peripheral Module Disable, PMD）：外设电源禁止硬件模块以将未使用外设的功耗降至最低

数字外设

- 可配置逻辑单元（Configurable Logic Cell, CLC）：
 - 4 个 CLC
 - 集成组合和顺序逻辑
- 互补波形发生器（Complementary Waveform Generator, CWG）：
 - 2 个 CWG
 - 上升沿和下降沿死区控制
 - 全桥、半桥和单通道驱动
 - 多个信号源
- 捕捉/比较/PWM（Capture/Compare/PWM, CCP）模块：
 - 4 个 CCP
 - 捕捉/比较模式：16 位分辨率
 - PWM 模式：10 位分辨率
- 脉宽调制器（Pulse-Width Modulator, PWM）：
 - 两个 10 位 PWM
- 数控振荡器（Numerically Controlled Oscillator, NCO）：
 - 高精度线性频率发生器（50% 占空比），源输入时钟的 0.0001% 步长
 - 输入时钟：0 Hz < F_{NCO} < 32 MHz
 - 分辨率： $F_{NCO}/2^{20}$
- 串行通信：
 - EUSART
 - 兼容 RS-232、RS-485 和 LIN
 - 自动波特率检测，检测到启动位时自动唤醒
 - 主同步串行端口（Master Synchronous Serial Port, MSSP）
 - SPI
 - 兼容 I²C、SMBus 和 PMBus™
- 数据信号调制器（Data Signal Modulator, DSM）：
 - 使用数字数据对载波信号进行调制，以生成定制的载波同步输出波形

PIC16(L)F18324/18344

- 最多 18 个 I/O 引脚：
 - 可单独编程的上拉
 - 压摆率控制
 - 带边沿选择功能的电平变化中断
 - 输入电平选择控制（ST 或 TTL）
 - 数字漏极开路使能
- 外设引脚选择（PPS）：
 - 数字外设的 I/O 引脚重映射
- 定时器模块：
 - **Timer0:**
 - 8/16 位定时器/计数器
 - 同步或异步操作
 - 可编程预分频器/后分频器
 - 捕捉/比较功能的时基
 - 带门控控制的 **Timer1/3/5:**
 - 16 位定时器/计数器
 - 可编程内部或外部时钟源
 - 多个门控源
 - 多个门控模式
 - 捕捉/比较功能的时基
 - **Timer2/4/6:**
 - 8 位定时器
 - 可编程预分频器/后分频器
 - PWM 功能的时基

模拟外设

- 10 位模数转换器（Analog-to-Digital Converter, ADC）：
 - 17 个外部通道
 - 可在休眠模式下进行转换
- 比较器：
 - 2 个比较器
 - 同相输入端施加固定参考电压
 - 比较器输出可从外部访问
- 5 位数模转换器（Digital-to-Analog Converter, DAC）：
 - 5 位分辨率，轨到轨
 - 正参考电压选择
 - 未经缓冲的 I/O 引脚输出
 - 内部连接至 ADC 和比较器
- 参考电压：
 - 固定参考电压：1.024V、2.048V 和 4.096V 输出电压

灵活的振荡器结构

- 高精度内部振荡器：
 - 可通过软件选择频率范围：最高 32 MHz
 - 4 MHz 标称校准点的精度为 $\pm 2\%$
- 4x PLL 用于外部时钟源
- 31 kHz 内部低功耗振荡器（LFINTOSC）
- 32 kHz 外部低功耗晶振（SOSC）
- 外部振荡器模块具有：
 - 3 种晶振/谐振器模式，频率最高为 20 MHz
 - 3 种外部时钟模式，频率最高为 32 MHz
 - 故障保护时钟监视器
 - 检测时钟源故障
 - 振荡器起振定时器（Oscillator Start-up Timer, OST）
 - 确保晶振源稳定性

PIC16(L)F18324/18344

PIC16(L)F183XX 系列类型

器件	数据手册索引	闪存程序存储器 (字)	闪存程序存储器 (KB)	数据存储器 (字节)	数据SRAM (字节)	I/O ⁽²⁾	10位ADC (通道)	5位DAC	高速比较器	CWG	参考时钟	定时器 (8/16位)	CCP	10位PWM	NCO	EUSART	I ² C/SPI	GLC	DSM	PPS	XLP	PMD	空闲和打盹 调试 ⁽¹⁾	
PIC16(L)F18313	(1)	2048	3.5	256	256	6	5	1	1	1	1	2/1	2	2	1	1	1/1	2	1	有	有	有	有	I
PIC16(L)F18323	(1)	2048	3.5	256	256	12	11	1	2	1	1	2/1	2	2	1	1	1/1	2	1	有	有	有	有	I
PIC16(L)F18324	(2)	4096	7	256	512	12	11	1	2	2	1	4/3	4	2	1	1	1/1	4	1	有	有	有	有	I
PIC16(L)F18324	(3)	8192	14	256	1024	12	11	1	2	2	1	4/3	4	2	1	1	2/2	4	1	有	有	有	有	I
PIC16(L)F18326	(4)	16384	28	256	2048	12	15	1	2	2	1	4/3	4	2	1	1	2/2	4	1	有	有	有	有	I
PIC16(L)F18344	(2)	4096	7	256	512	18	17	1	2	2	1	4/3	4	2	1	1	1/1	4	1	有	有	有	有	I
PIC16(L)F18344	(3)	8192	14	256	1024	18	17	1	2	2	1	4/3	4	2	1	1	2/2	4	1	有	有	有	有	I
PIC16(L)F18346	(4)	16384	28	256	2048	18	21	1	2	2	1	4/3	4	2	1	1	2/2	4	1	有	有	有	有	I

注 1: 调试方法: (I)——集成在芯片上;
2: 其中一个引脚仅用作输入。

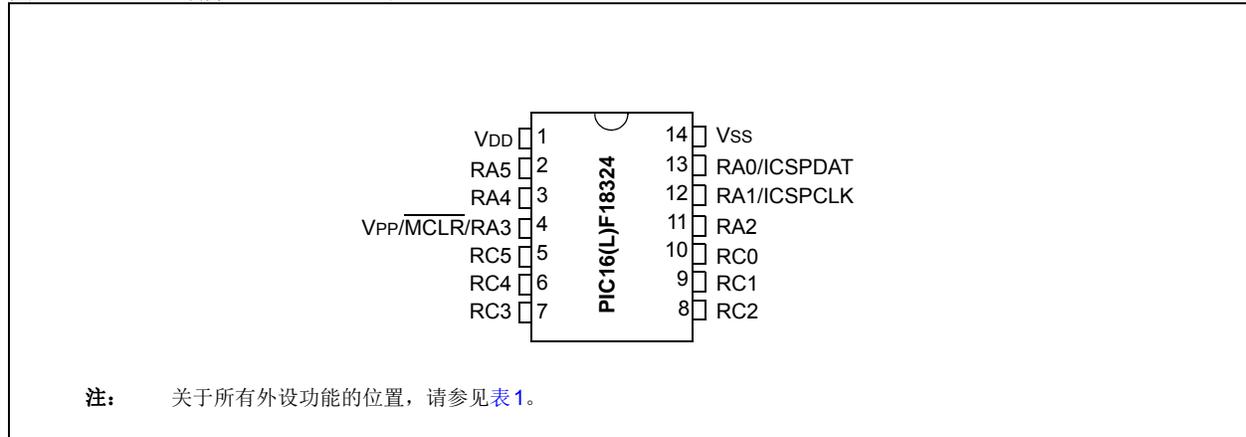
数据手册索引: (本文档仅介绍未用阴影表示的器件。)

- 1: DS40001799 PIC16(L)F18313/18323 数据手册, 采用 XLP 技术的全功能低引脚数单片机
- 2: DS40001800 PIC16(L)F18324/18344 数据手册, 采用 XLP 技术的全功能低引脚数单片机
- 3: DS40001795 PIC16(L)F18325/18345 数据手册, 采用 XLP 技术的全功能低引脚数单片机
- 4: DS40001839 PIC16(L)F18326/18346 数据手册, 采用 XLP 技术的全功能低引脚数单片机

注: 关于其他小型封装的供货和标识信息, 请访问 <http://www.microchip.com/packaging> 或联系当地的销售办事处。

引脚图

图1: 14引脚PDIP、SOIC和TSSOP



PIC16(L)F18324/18344

图2: 16引脚UQFN (4x4)

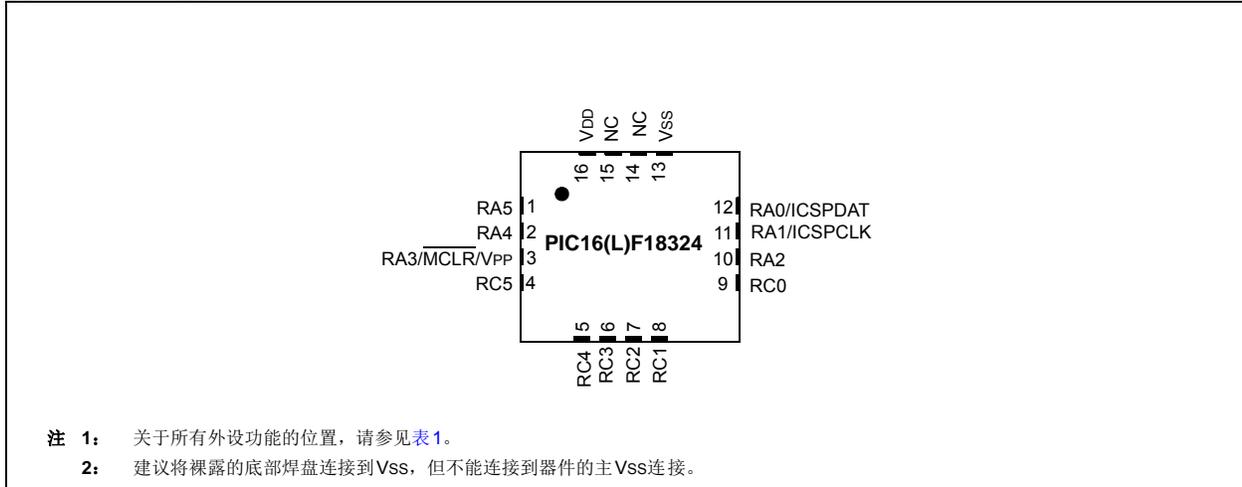


图3: 20引脚PDIP、SOIC和SSOP

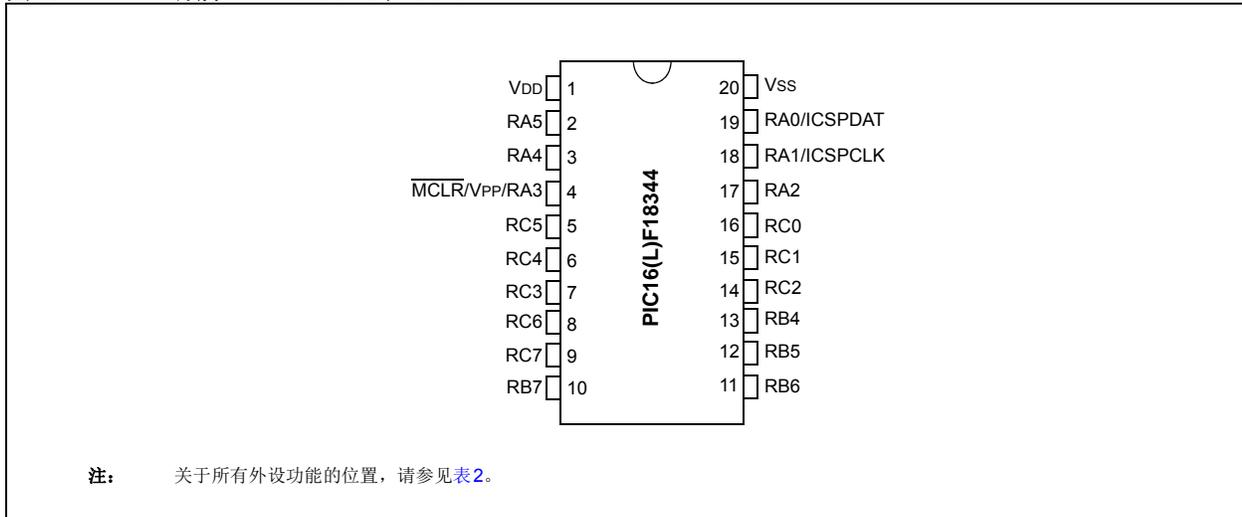
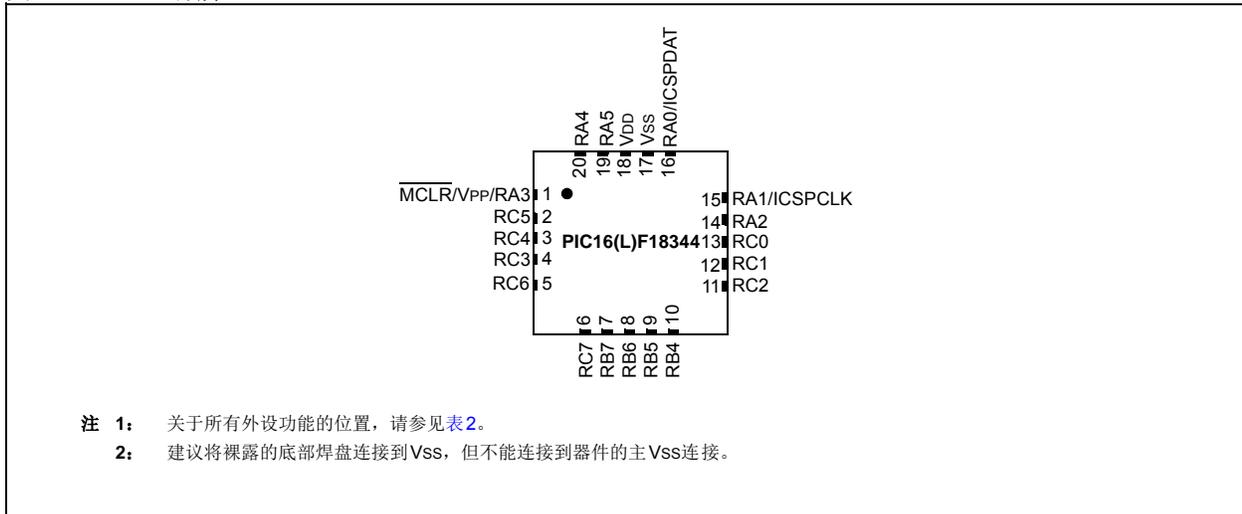


图4: 20引脚UQFN (4x4)



引脚分配表

表1: 14/16引脚分配表 (PIC16(L)F18324)

I/O ⁽²⁾	14引脚 PDIP/SOIC/TSSOP	16引脚 UQFN	ADC	参考电压	比较器	NCO	DAC	DSM	定时器	CCP	PWM	CWG	MSSP	EUSART	CLC	CLKR	中断	上拉	基本功能
RA0	13	12	ANA0	—	C1IN0+	—	DAC1OUT	—	—	—	—	—	—	—	—	—	IO	有	ICDDAT/ ICSPDAT
RA1	12	11	ANA1	VREF+	C1IN0- C2IN0-	—	DAC1REF+	—	—	—	—	—	—	—	—	—	IO	有	ICDCLK/ ICSPCLK
RA2	11	10	ANA2	VREF-	—	—	DAC1REF-	—	T0CKI ⁽¹⁾	CCP3 ⁽¹⁾	—	CWG1IN ⁽¹⁾ CWG2IN ⁽¹⁾	—	—	—	—	INT ⁽¹⁾ IO	有	—
RA3	4	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IO	有	$\overline{\text{MCLR}}$ V _{PP}
RA4	3	2	ANA4	—	—	—	—	—	T1G ⁽¹⁾ SOSCO	—	—	—	—	—	—	—	IO	有	CLKOUT OSC2
RA5	2	1	ANA5	—	—	—	—	—	T1CKI ⁽¹⁾ SOSCIN SOSCI	—	—	—	—	—	CLCIN3 ⁽¹⁾	—	IO	有	CLKIN OSC1
RC0	10	9	ANC0	—	C2IN0+	—	—	—	T5CKI ⁽¹⁾	—	—	—	SCK1 ⁽¹⁾ SCL1 ^(1,3,4)	—	—	—	IO	有	—
RC1	9	8	ANC1	—	C1IN1- C2IN1-	—	—	—	—	CCP4 ⁽¹⁾	—	—	SD1 ⁽¹⁾ SDA1 ^(1,3,4)	—	CLCIN2 ⁽¹⁾	—	IO	有	—
RC2	8	7	ANC2	—	C1IN2- C2IN2-	—	—	MDCIN1 ⁽¹⁾	—	—	—	—	—	—	—	—	IO	有	—
RC3	7	6	ANC3	—	C1IN3- C2IN3-	—	—	MDMIN ⁽¹⁾	T5G ⁽¹⁾	CCP2 ⁽¹⁾	—	—	$\overline{\text{SS}}1(1)$	—	CLCIN0 ⁽¹⁾	—	IO	有	—
RC4	6	5	ANC4	—	—	—	—	—	T3G ⁽¹⁾	—	—	—	—	—	CLCIN1 ⁽¹⁾	—	IO	有	—
RC5	5	4	ANC5	—	—	—	—	MDCIN2 ⁽¹⁾	T3CKI ⁽¹⁾	CCP1 ⁽¹⁾	—	—	—	RX ⁽¹⁾	—	—	IO	有	—
VDD	1	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	14	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VSS

- 注 1: 默认外设输入。通过PPS输入选择寄存器, 可以将输入移至任何其他引脚。
 2: 所有引脚均默认输出至端口数据锁存器。通过PPS输出选择寄存器, 可以选择任意引脚作为数字外设输出。
 3: 这些外设功能是双向的。输出引脚选择必须与输入引脚选择相同。
 4: 这些引脚配置为I²C逻辑电平; 时钟和数据信号可分配给任何这些引脚。分配给其他引脚(例如, RA5)可将操作, 但逻辑电平将是标准TTL/ST, 如INLVL寄存器所选。

表1: 14/16引脚分配表 (PIC16(L)F18324) (续)

I/O/I ⁽²⁾	14引脚 PDIP/SOIC/TSSOP	16引脚 UQFN	ADC	参考电压	比较器	NCO	DAC	DSM	定时器	CCP	PWM	CWG	MSSP	EUSART	CLC	CLKR	中断	上拉	基本功能
OUT ⁽²⁾	—	—	—	—	C1OUT	NCO1	—	DSM	TMR0	CCP1	PWM5	CWG1A CWG2A	SDA1 ⁽³⁾ SDA2 ⁽³⁾	CK	CLC1OUT	CLKR	—	—	—
	—	—	—	—	C2OUT	—	—	—	—	CCP2	PWM6	CWG1B CWG2B	SCL1 ⁽³⁾ SCL2 ⁽³⁾	DT	CLC2OUT	—	—	—	—
	—	—	—	—	—	—	—	—	—	CCP3	—	CWG1C CWG2C	SDO1 SDO2	TX	CLC3OUT	—	—	—	—
	—	—	—	—	—	—	—	—	—	CCP4	—	CWG1D CWG2D	SCK1 SCK2	—	CLC4OUT	—	—	—	—

- 注 1: 默认外设输入。通过PPS输入选择寄存器, 可以将输入移至任何其他引脚。
- 2: 所有引脚均默认输出至端口数据锁存器。通过PPS输出选择寄存器, 可以选择任意引脚作为数字外设输出。
- 3: 这些外设功能是双向的。输出引脚选择必须与输入引脚选择相同。
- 4: 这些引脚配置为I²C逻辑电平; 时钟和数据信号可分配给任何这些引脚。分配给其他引脚 (例如, RA5) 将可操作, 但逻辑电平将是标准TTL/ST, 如INLVL寄存器所选。

表2: 20引脚分配表 (PIC16(L)F18344)

I/O ⁽²⁾	20引脚PDIP/SOIC/SSOP	20引脚UQFN	ADC	参考电压	比较器	NCO	DAC	DSM	定时器	CCP	PWM	CWG	MSSP	EUSART	CLC	CLKR	中断	上拉	基本功能
RA0	19	16	ANA0	—	C1IN0+	—	DAC1OUT	—	—	—	—	—	—	—	—	—	IO	有	ICDDAT ICSPDAT
RA1	18	15	ANA1	VREF+	C1IN0- C2IN0-	—	DAC1REF+	—	—	—	—	—	—	—	—	—	IO	有	ICDCLK ICSPCLK
RA2	17	14	ANA2	VREF-	—	—	DAC1REF-	—	T0CKI ⁽¹⁾	CCP3 ⁽¹⁾	—	CWG1IN ⁽¹⁾ CWG2IN ⁽¹⁾	—	—	CLCIN0 ⁽¹⁾	—	IO INT ⁽¹⁾	有	—
RA3	4	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IO	有	MCLR VPP
RA4	3	20	ANA4	—	—	—	—	—	T1G ⁽¹⁾ T3G ⁽¹⁾ T5G ⁽¹⁾ SOSCO	CCP4 ⁽¹⁾	—	—	—	—	—	—	IO	有	CLKOUT OSC2
RA5	2	19	ANA5	—	—	—	—	—	T1CKI ⁽¹⁾ T3CKI ⁽¹⁾ T5CKI ⁽¹⁾ SOSCIN SOSCI	—	—	—	—	—	—	—	IO	有	CLKIN OSC1
RB4	13	10	ANB4	—	—	—	—	—	—	—	—	—	SDI1 ⁽¹⁾ SDA1 ^(1,3,4)	—	CLCIN2 ⁽¹⁾	—	IO	有	—
RB5	12	9	ANB5	—	—	—	—	—	—	—	—	—	—	RX ⁽¹⁾	CLCIN3 ⁽¹⁾	—	IO	有	—
RB6	11	8	ANB6	—	—	—	—	—	—	—	—	—	SCK1 ⁽¹⁾ SCL1 ^(1,3,4)	—	—	—	IO	有	—
RB7	10	7	ANB7	—	—	—	—	—	—	—	—	—	—	—	—	—	IO	有	—
RC0	16	13	ANC0	—	C2IN0+	—	—	—	—	—	—	—	—	—	—	—	IO	有	—
RC1	15	12	ANC1	—	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	—	—	—	IO	有	—
RC2	14	11	ANC2	—	C1IN2- C2IN2-	—	—	MDCIN1 ⁽¹⁾	—	—	—	—	—	—	—	—	IO	有	—

注 1: 默认外设输入。通过PPS输入选择寄存器, 可以将输入移至任何其他引脚。

2: 所有引脚均默认输出至端口数据锁存器。通过PPS输出选择寄存器, 可以选择任意引脚作为数字外设输出。

3: 这些外设功能是双向的。输出引脚选择必须与输入引脚选择相同。

4: 这些引脚配置为I²C逻辑电平; 时钟和数据信号可分配给任何这些引脚。分配给其他引脚 (例如, RA5) 将可操作, 但逻辑电平将是标准TTL/ST, 如INLVL寄存器所选。

表2: 20引脚分配表 (PIC16(L)F18344) (续)

I/O ⁽²⁾	PPS/IO/PI/DI/SSOP 20引脚	IO/PI/DI 20引脚	ADC	参考电压	比较器	NCO	DAC	DSM	定时器	CCP	PWM	CWG	MSSP	EUSART	CLC	CLKR	中断	移位	基本功能
RC3	7	4	ANC3	—	C1IN3- C2IN3-	—	—	MDMIN ⁽¹⁾	—	CCP2 ⁽¹⁾	—	—	—	—	CLCIN1 ⁽¹⁾	—	IOC	有	—
RC4	6	3	ANC4	—	—	—	—	—	—	—	—	—	—	—	—	—	IOC	有	—
RC5	5	2	ANC5	—	—	—	—	MDCIN2 ⁽¹⁾	—	CCP1 ⁽¹⁾	—	—	—	—	—	—	IOC	有	—
RC6	8	5	ANC6	—	—	—	—	—	—	—	—	—	SS1 ⁽¹⁾	—	—	—	IOC	有	—
RC7	9	6	ANC7	—	—	—	—	—	—	—	—	—	—	—	—	—	IOC	有	—
VDD	1	18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	20	17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VSS
OUT ⁽²⁾	—	—	—	—	C1OUT	NCO1	—	DSM	TMR0	CCP1	PWM5	CWG1A CWG2A	SDO1	DT	CLC1OUT	CLKR	—	—	—
	—	—	—	—	C2OUT	—	—	—	—	CCP2	PWM6	CWG1B CWG2B	SCK1	CK	CLC2OUT	—	—	—	—
	—	—	—	—	—	—	—	—	—	CCP3	—	CWG1C CWG2C	SCL1 ⁽³⁾	TX	CLC3OUT	—	—	—	—
	—	—	—	—	—	—	—	—	—	CCP4	—	CWG1D CWG2D	SDA1 ⁽³⁾	—	CLC4OUT	—	—	—	—

注 1: 默认外设输入。通过PPS输入选择寄存器, 可以将输入移至任何其他引脚。

2: 所有引脚均默认输出至端口数据锁存器。通过PPS输出选择寄存器, 可以选择任意引脚作为数字外设输出。

3: 这些外设功能是双向的。输出引脚选择必须与输入引脚选择相同。

4: 这些引脚配置为I²C逻辑电平; 时钟和数据信号可分配给任何这些引脚。分配给其他引脚 (例如, RA5) 将可操作, 但逻辑电平将是标准TTL/ST, 如INLVL寄存器所选。

PIC16(L)F18324/18344

目录

1.0	器件概述	11
3.0	增强型中档CPU	23
4.0	存储器构成	25
5.0	器件配置	61
6.0	复位	68
7.0	振荡器模块	76
8.0	中断	95
9.0	节能工作模式	111
10.0	看门狗定时器 (WDT)	117
11.0	非易失性存储器 (NVM) 控制	121
12.0	I/O 端口	138
13.0	外设引脚选择 (PPS) 模块	158
14.0	外设模块禁止	164
15.0	电平变化中断	170
16.0	固定参考电压 (FVR)	177
17.0	温度指示器模块	180
18.0	比较器模块	182
19.0	脉宽调制 (PWM)	191
20.0	互补波形发生器 (CWG) 模块	197
21.0	可配置逻辑单元 (CLC)	219
22.0	模数转换器 (ADC) 模块	234
23.0	数控振荡器 (NCO1)	248
24.0	5位数字模转换器 (DAC1) 模块	259
25.0	数据信号调制器 (DSM) 模块	263
26.0	Timer0 模块	274
27.0	带门控的 Timer1/3/5 模块	281
28.0	Timer2/4/6 模块	294
29.0	捕捉/比较/PWM 模块	299
30.0	主同步串行端口 (MSSP1) 模块	311
31.0	增强型通用同步/异步收发器 (EUSART1)	364
32.0	参考时钟输出模块	389
33.0	在线串行编程 (ICSP™)	392
34.0	指令集汇总	394
35.0	电气规范	408
36.0	直流和交流特性图表	438
37.0	开发支持	459
38.0	封装信息	463
	Microchip 网站	486
	变更通知客户服务	486
	客户支持	486
	产品标识体系	487

致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的需求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 CTRC@microchip.com。我们期待您的反馈。

最新数据手册

欲获得本数据手册的最新版本，请访问我公司网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中紧跟数字串后的字母是版本号，例如：DS30000000A_CN 是文档的 A 版本。

勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站 <http://www.microchip.com>
- 当地 Microchip 销售办事处（见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本（包括文献编号）。

客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 www.microchip.com 上注册。

PIC16(L)F18324/18344

1.0 器件概述

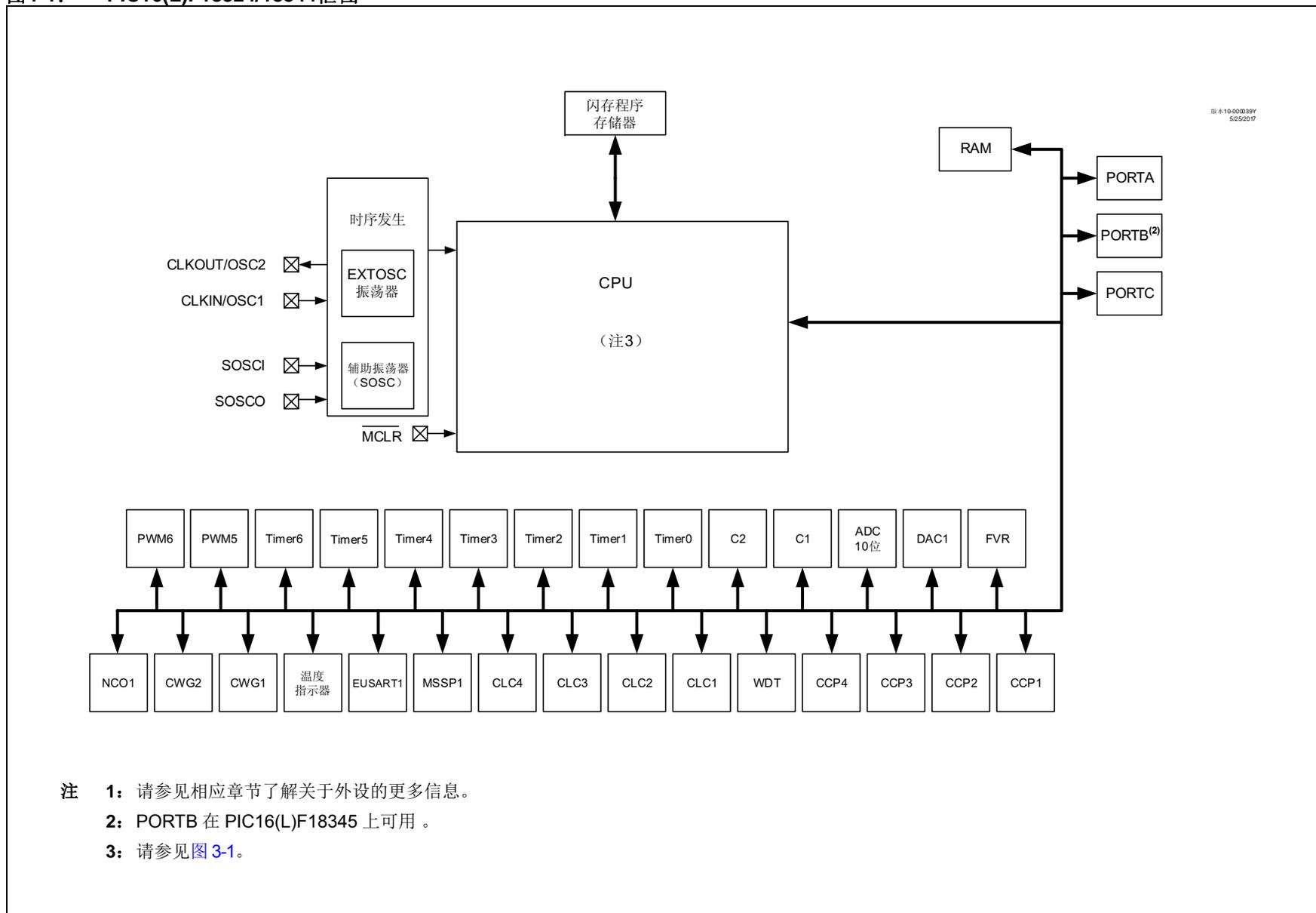
本数据手册介绍了PIC16(L)F18324/18344器件。PIC16(L)F18324提供14引脚PDIP、SOIC、TSSOP和16引脚UQFN封装。PIC16(L)F18344提供20引脚PDIP、SOIC、SSOP和UQFN封装。有关更多封装信息，请参见第38.0节“封装信息”。图1-1给出了PIC16(L)F18324/18344器件的框图。表1-2给出了引脚说明。

每款器件的可用外设，请参见表1-1。

表1-1: 器件外设汇总

外设	PIC16(L)F18324	PIC16(L)F18344
模数转换器 (ADC)	•	•
温度指示器	•	•
数模转换器 (DAC)		
	DAC1	• •
固定参考电压 (Fixed Voltage Reference, FVR)		
	ADCFVR	• •
	CDAFVR	• •
数字信号调制器 (Data Signal Modulator, DSM)		
	DSM1	• •
数控振荡器 (NCO)		
	NCO1	• •
捕捉/比较/PWM (CCP) 模块		
	CCP1	• •
	CCP2	• •
	CCP3	• •
	CCP4	• •
比较器		
	C1	• •
	C2	• •
互补波形发生器 (CWG)		
	CWG1	• •
	CWG2	• •
可配置逻辑单元 (CLC)		
	CLC1	• •
	CLC2	• •
	CLC3	• •
	CLC4	• •
增强型通用同步/异步收发器 (EUSART)		
	EUSART1	• •
主同步串行端口 (MSSP)		
	MSSP1	• •
脉宽调制器 (PWM)		
	PWM5	• •
	PWM6	• •
定时器 (TMR)		
	TMR0	• •
	TMR1	• •
	TMR2	• •
	TMR3	• •
	TMR4	• •
	TMR5	• •
	TMR6	• •

图 1-1: PIC16(L)F18324/18344框图



- 注 1: 请参见相应章节了解关于外设的更多信息。
 注 2: PORTB 在 PIC16(L)F18345 上可用。
 注 3: 请参见图 3-1。

PIC16(L)F18324/18344

表1-2: PIC16(L)F18324引脚说明

名称	功能	输入类型	输出类型	说明
RA0/ANA0/C1IN0+/DAC1OUT/ ICDDAT/ICSPDAT	RA0	TTL/ST	CMOS	通用 I/O。
	ANA0	AN	—	ADC通道A0输入。
	C1IN0+	AN	—	比较器C1的同相输入。
	DAC1OUT	—	AN	数模转换器输出。
	ICDDAT	TTL/ST	—	在线调试数据 I/O。
	ICSPDAT	TTL/ST	—	ICSP™ 数据 I/O。
RA1/ANA1/VREF+/C1IN0-/ C2IN0-/DAC1REF+ /ICDCLK/ ICSPCLK	RA1	TTL/ST	CMOS	通用 I/O。
	ANA1	AN	—	ADC通道A1输入。
	VREF+	AN	—	ADC正参考电压输入。
	C1IN0-	AN	—	比较器C1的反相输入。
	C2IN0-	AN	—	比较器C2的反相输入。
	DAC1REF+	AN	—	数模转换器正参考输入。
	ICDCLK	TTL/ST	—	在线调试时钟 I/O。
ICSPCLK	TTL/ST	—	ICSP 时钟 I/O。	
RA2/ANA2/VREF- /DAC1REF-/ T0CKI ⁽¹⁾ / CCP3 ⁽¹⁾ /CWG1IN ⁽¹⁾ / CWG2IN ⁽¹⁾ /INT ⁽¹⁾	RA2	TTL/ST	CMOS	通用 I/O。
	ANA2	AN	—	ADC通道A2输入。
	VREF-	AN	—	ADC负参考电压输入。
	DAC1REF-	AN	—	数模转换器负参考输入。
	T0CKI	TTL/ST	—	TMR0时钟输入。
	CCP3	TTL/ST	—	捕捉/比较/PWM 3输入。
	CWG1IN	TTL/ST	—	互补波形发生器1的输入。
	CWG2IN	TTL/ST	—	互补波形发生器2的输入。
INT	TTL/ST	—	外部中断输入。	
RA3/MCLR/VPP	RA3	TTL/ST	CMOS	通用 I/O。
	MCLR	TTL/ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
RA4/ANA4/T1G ⁽¹⁾ / SOSC0/ CLKOUT/OSC2	RA4	TTL/ST	CMOS	通用 I/O。
	ANA4	AN	—	ADC通道A4输入。
	T1G	ST	—	TMR1门控输入。
	SOSC0	—	XTAL	辅助振荡器连接。
	CLKOUT	—	CMOS	Fosc/4 输出。
OSC2	—	XTAL	晶振/谐振器 (LP、XT和HS模式)。	
RA5/ANA5/T1CKI ⁽¹⁾ / SOSCIN/ SOSCI/ CLCIN3 ⁽¹⁾ /CLKIN/ OSC1	RA5	TTL/ST	CMOS	通用 I/O。
	ANA5	AN	—	ADC通道A5输入。
	T1CKI	TTL/ST	—	TMR1时钟输入。
	SOSCIN	TTL/ST	—	辅助振荡器输入连接。
	SOSCI	XTAL	—	辅助振荡器连接。
	CLCIN3	TTL/ST	—	可配置逻辑单元3输入。
	CLKIN	TTL/ST	—	外部时钟输入。
OSC1	XTAL	—	晶振/谐振器 (LP、XT和HS模式)。	

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平施密特触发器输入 I²C = I²C 电平施密特触发器输入
HV = 高电压 XTAL = 晶振电平

- 注 1: 默认外设输入。通过PPS输入选择寄存器, 可以将输入移至任何其他引脚。请参见寄存器13-1。
注 2: 所有引脚均默认输出至端口数据锁存器。通过PPS输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见寄存器13-2。
注 3: 这些I²C功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F18324/18344

表1-2: PIC16(L)F18324引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RC0/ANC0/C2IN0+/ T5CKI ⁽¹⁾ / SCK1 ⁽¹⁾ / SCL1 ^(1,3)	RC0	TTL/ST	CMOS	通用 I/O。
	ANC0	AN	—	ADC 通道 C0 输入。
	C2IN0+	AN	—	比较器 C2 的同相输入。
	T5CKI	TTL/ST	—	TMR5 时钟输入。
	SCK1	TTL/ST	—	SPI 时钟 1。
	SCL1	I ² C	OD	I ² C 时钟 1。
RC1/ANC1/C1IN1-/C2IN1-/ CCP4 ⁽¹⁾ /SDI1 ⁽¹⁾ / SDA1 ^(1,3) / CLCIN2 ⁽¹⁾	RC1	TTL/ST	CMOS	通用 I/O。
	ANC1	AN	—	ADC 通道 C1 输入。
	C1IN1-	AN	—	比较器 C1 的反相输入。
	C2IN1-	AN	—	比较器 C2 的反相输入。
	CCP4	TTL/ST	—	捕捉/比较/PWM 4 输入。
	SDI1	TTL/ST	—	SPI 数据输入 1。
	SDA1	I ² C	OD	I ² C 数据 1。
	CLCIN2	TTL/ST	—	可配置逻辑单元 2 输入。
RC2/ANC2/C1IN2-/C2IN2-/ MDCIN1 ⁽¹⁾	RC2	TTL/ST	CMOS	通用 I/O。
	ANC2	AN	—	ADC 通道 C2 输入。
	C1IN2-	AN	—	比较器 C1 的反相输入。
	C2IN2-	AN	—	比较器 C2 的反相输入。
	MDCIN1	TTL/ST	—	模块化载波输入 1。
RC3/ANC3/C1IN3-/C2IN3-/ MDMIN ⁽¹⁾ /T5G ⁽¹⁾ / CCP2 ⁽¹⁾ / SS1 ⁽¹⁾ /CLCIN0 ⁽¹⁾	RC3	TTL/ST	CMOS	通用 I/O。
	ANC3	AN	—	ADC 通道 C3 输入。
	C1IN3-	AN	—	比较器 C1 的反相输入。
	C2IN3-	AN	—	比较器 C2 的反相输入。
	MDMIN	TTL/ST	—	模块化源输入。
	T5G	TTL/ST	—	TMR5 门控输入。
	CCP2	TTL/ST	—	捕捉/比较/PWM 2 输入。
	SS1	TTL/ST	—	从选择 1 输入。
	CLCIN0	TTL/ST	—	可配置逻辑单元 0 输入。
RC4/ANC4/T3G ⁽¹⁾ / CLCIN1 ⁽¹⁾	RC4	TTL/ST	CMOS	通用 I/O。
	ANC4	AN	—	ADC 通道 C4 输入。
	T3G	TTL/ST	—	TMR3 门控输入。
	CLCIN1	TTL/ST	—	可配置逻辑单元 1 输入。
RC5/ANC5/MDCIN2 ⁽¹⁾ / T3CKI ⁽¹⁾ /CCP1 ⁽¹⁾ /RX ⁽¹⁾	RC5	TTL/ST	CMOS	通用 I/O。
	ANC5	AN	—	ADC 通道 C5 输入。
	MDCIN2	TTL/ST	—	模块化载波输入 2。
	T3CKI	TTL/ST	—	TMR3 时钟输入。
	CCP1	TTL/ST	—	捕捉/比较/PWM 1 输入。
	RX	TTL/ST	—	EUSART 异步输入。
VDD	VDD	电源	—	EUSART 同步输入。
VSS	VSS	电源	—	正电源。

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平施密特触发器输入 I²C = I²C 电平施密特触发器输入
HV = 高电压 XTAL = 晶振电平

- 注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以将输入移至任何其他引脚。请参见 [寄存器 13-1](#)。
注 2: 所有引脚均默认输出至端口数据锁存器。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见 [寄存器 13-2](#)。
注 3: 这些 I²C 功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F18324/18344

表1-2: PIC16(L)F18324引脚说明 (续)

名称	功能	输入类型	输出类型	说明
OUT ⁽²⁾	C1	—	CMOS	比较器C1输出。
	C2	—	CMOS	比较器C2输出。
	NCO1	—	CMOS	数控振荡器输出。
	DSM	—	CMOS	数字信号调制器输出。
	TMR0	—	CMOS	TMR0时钟输出。
	CCP1	—	CMOS	捕捉/比较/PWM 1输出。
	CCP2	—	CMOS	捕捉/比较/PWM 2输出。
	CCP3	—	CMOS	捕捉/比较/PWM 3输出。
	CCP4	—	CMOS	捕捉/比较/PWM 4输出。
	PWM5	—	CMOS	脉宽调制器5的输出。
	PWM6	—	CMOS	脉宽调制器6的输出。
	CWG1A	—	CMOS	互补波形发生器1的输出A。
	CWG2A	—	CMOS	互补波形发生器2的输出A。
	CWG1B	—	CMOS	互补波形发生器1的输出B。
	CWG2B	—	CMOS	互补波形发生器2的输出B。
	CWG1C	—	CMOS	互补波形发生器1的输出C。
	CWG2C	—	CMOS	互补波形发生器2的输出C。
	CWG1D	—	CMOS	互补波形发生器1的输出D。
	CWG2D	—	CMOS	互补波形发生器2的输出D。
	SDA1 ⁽³⁾	I ² C	OD	I ² C数据输出。
	SCL1 ⁽³⁾	I ² C	OD	I ² C时钟输出。
	SDO1	—	CMOS	SPI1数据输出。
	SCK1	—	CMOS	SPI1时钟输出。
	TX/CK	—	CMOS	异步发送数据/同步时钟输出。
	DT	—	CMOS	EUSART同步数据输出。
	CLC1OUT	—	CMOS	可配置逻辑单元1源输出。
	CLC2OUT	—	CMOS	可配置逻辑单元2源输出。
	CLC3OUT	—	CMOS	可配置逻辑单元3源输出。
	CLC4OUT	—	CMOS	可配置逻辑单元4源输出。
	CLKR	—	CMOS	时钟参考输出。

图注: AN = 模拟输入或输出 CMOS = CMOS兼容输入或输出 OD = 漏极开路
TTL = TTL兼容输入 ST = CMOS电平施密特触发器输入 I²C = I²C电平施密特触发器输入
HV = 高电压 XTAL = 晶振电平

- 注 1:** 默认外设输入。通过PPS输入选择寄存器，可以将输入移至任何其他引脚。请参见寄存器13-1。
2: 所有引脚均默认输出至端口数据锁存器。通过PPS输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见寄存器13-2。
3: 这些I²C功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F18324/18344

表1-3: PIC16(L)F18344引脚说明

名称	功能	输入类型	输出类型	说明
RA0/ANA0/C1IN0+/DAC1OUT/ ICDDAT/ICSPDAT	RA0	TTL/ST	CMOS	通用 I/O。
	ANA0	AN	—	ADC 通道 A0 输入。
	C1IN0+	AN	—	比较器 C1 的同相输入。
	DAC1OUT	—	AN	数模转换器输出。
	ICDDAT	TTL/ST	—	在线调试数据 I/O。
	ICSPDAT	TTL/ST	—	ICSP 数据 I/O。
RA1/ANA1/VREF+/C1IN0-/ C2IN0-/ DAC1REF+/ ICDCLK/ ICSPCLK	RA1	TTL/ST	CMOS	通用 I/O。
	ANA1	AN	—	ADC 通道 A1 输入。
	VREF+	AN	—	ADC 正参考电压输入。
	C1IN0-	AN	—	比较器 C1 的反相输入。
	C2IN0-	AN	—	比较器 C2 的反相输入。
	DAC1REF+	AN	—	数模转换器正参考输入。
	ICDCLK	TTL/ST	—	在线调试时钟 I/O。
ICSPCLK	TTL/ST	—	ICSP 时钟 I/O。	
RA2/ANA2/VREF-/ DAC1REF-/ T0CKI ⁽¹⁾ / CCP3 ⁽¹⁾ /CWG1IN ⁽¹⁾ / CWG2IN ⁽¹⁾ /CLCIN0 ⁽¹⁾ / INT ⁽¹⁾	RA2	TTL/ST	CMOS	通用 I/O。
	ANA2	AN	—	ADC 通道 A2 输入。
	VREF-	AN	—	ADC 负参考电压输入。
	DAC1REF-	AN	—	数模转换器负参考输入。
	T0CKI	TTL/ST	—	TMR0 时钟输入。
	CCP3	TTL/ST	—	捕捉/比较/PWM 3 输入。
	CWG1IN	TTL/ST	—	互补波形发生器 1 的输入。
	CWG2IN	TTL/ST	—	互补波形发生器 2 的输入。
CLCIN0	TTL/ST	—	可配置逻辑单元 0 输入。	
INT	TTL/ST	—	外部中断输入。	
RA3/MCLR/VPP	RA3	TTL/ST	CMOS	通用 I/O。
	MCLR	TTL/ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
RA4/ANA4/T1G ⁽¹⁾ /T3G ⁽¹⁾ /T5G ⁽¹⁾ / SOSCO/CCP4 ⁽¹⁾ /CLKOUT/ OSC2	RA4	TTL/ST	CMOS	通用 I/O。
	ANA4	AN	—	ADC 通道 A4 输入。
	T1G	TTL/ST	—	TMR1 门控输入。
	T3G	TTL/ST	—	TMR3 门控输入。
	T5G	TTL/ST	—	TMR5 门控输入。
	SOSCO	—	XTAL	辅助振荡器连接。
	CCP4	TTL/ST	—	捕捉/比较/PWM 4 输入。
	CLKOUT	—	CMOS	FOSC/4 输出。
OSC2	—	XTAL	晶振/谐振器 (LP、XT 和 HS 模式)。	

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平施密特触发器输入 I²C = I²C 电平施密特触发器输入
HV = 高电压 XTAL = 晶振电平

- 注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以将输入移至任何其他引脚。请参见寄存器 13-2。
2: 所有引脚均默认输出至端口数据锁存器。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见寄存器 13-2。
3: 这些 I²C 功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F18324/18344

表1-3: PIC16(L)F18344引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RA5/ANA5/T1CKI ⁽¹⁾ / T3CKI ⁽¹⁾ / T5CKI ⁽¹⁾ / SOSCIN/SOSCI/ CLKIN/OSC1	RA5	TTL/ST	CMOS	通用 I/O。
	ANA5	AN	—	ADC 通道 A5 输入。
	T1CKI	TTL/ST	—	TMR1 时钟输入。
	T3CKI	TTL/ST	—	TMR3 时钟输入。
	T5CKI	TTL/ST	—	TMR5 时钟输入。
	SOSCIN	TTL/ST	—	辅助振荡器输入连接。
	SOSCI	XTAL	—	辅助振荡器连接。
	CLKIN	TTL/ST	—	外部时钟输入。
OSC1	XTAL	—	晶振/谐振器 (LP、XT 和 HS 模式)。	
RB4/ANB4/SDI1 ⁽¹⁾ / SDA1 ^(1,3) / CLCIN2 ⁽¹⁾	RB4	TTL/ST	CMOS	通用 I/O。
	ANB4	AN	—	ADC 通道 B4 输入。
	SDI1	TTL/ST	—	SPI 数据输入 1。
	SDA1	I ² C	OD	I ² C 数据 1。
	CLCIN2	TTL/ST	—	可配置逻辑单元 2 输入。
RB5/ANB5/RX ⁽¹⁾ /CLCIN3 ⁽¹⁾	RB5	TTL/ST	—	通用 I/O。
	ANB5	AN	—	ADC 通道 B5 输入。
	RX	TTL/ST	—	EUSART 异步输入。
	CLCIN3	TTL/ST	—	可配置逻辑单元 3 输入。
RB6/ANB6/SCK1 ⁽¹⁾ / SCL1 ^(1,3)	RB6	TTL/ST	CMOS	通用 I/O。
	ANB6	AN	—	ADC 通道 B6 输入。
	SCK1	TTL/ST	—	SPI 时钟 1。
	SCL1	I ² C	OD	I ² C 时钟 1。
RB7/ANB7	RB7	TTL/ST	CMOS	通用 I/O。
	ANB7	AN	—	ADC 通道 B7 输入。
RC0/ANC0/C2IN0+	RC0	TTL/ST	CMOS	通用 I/O。
	ANC0	AN	—	ADC 通道 C0 输入。
	C2IN0+	AN	—	比较器 C2 的同相输入。
RC1/ANC1/C1IN1-/C2IN1-	RC1	TTL/ST	CMOS	通用 I/O。
	ANC1	AN	—	ADC 通道 C1 输入。
	C1IN1-	AN	—	比较器 C1 的反相输入。
	C2IN1-	AN	—	比较器 C2 的反相输入。
RC2/ANC2/C1IN2-/C2IN2-/ MDCIN1 ⁽¹⁾	RC2	TTL/ST	CMOS	通用 I/O。
	ANC2	AN	—	ADC 通道 C2 输入。
	C1IN2-	AN	—	比较器 C1 的反相输入。
	C2IN2-	AN	—	比较器 C2 的反相输入。
	MDCIN1	TTL/ST	—	模块化载波输入 1。

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平施密特触发器输入 I²C = I²C 电平施密特触发器输入
HV = 高电压 XTAL = 晶振电平

- 注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以将输入移至任何其他引脚。请参见寄存器 13-2。
2: 所有引脚均默认输出至端口数据锁存器。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见寄存器 13-2。
3: 这些 I²C 功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F18324/18344

表1-3: PIC16(L)F18344引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RC3/ANC3/C1IN3-/C2IN3-/MDMIN ⁽¹⁾ /CCP2 ⁽¹⁾ /CLCIN1 ⁽¹⁾	RC3	TTL/ST	CMOS	通用 I/O。
	ANC3	AN	—	ADC 通道 C3 输入。
	C1IN3-	AN	—	比较器 C1 的反相输入。
	C2IN3-	AN	—	比较器 C2 的反相输入。
	MDMIN	TTL/ST	—	模块化源输入。
	CCP2	TTL/ST	—	捕捉/比较/PWM 2 输入。
	CLCIN1	TTL/ST	—	可配置逻辑单元 1 输入。
RC4/ANC4	RC4	TTL/ST	CMOS	通用 I/O。
	ANC4	AN	—	ADC 通道 C4 输入。
RC5/ANC5/MDCIN2 ⁽¹⁾ /CCP1 ⁽¹⁾	RC5	TTL/ST	CMOS	通用 I/O。
	ANC5	AN	—	ADC 通道 C5 输入。
	MDCIN2	TTL/ST	—	模块化载波输入 2。
	CCP1	TTL/ST	—	捕捉/比较/PWM 1 输入。
RC6/ANC6/SS1 ⁽¹⁾	RC6	TTL/ST	CMOS	通用 I/O。
	ANC6	AN	—	ADC 通道 C6 输入。
	SS1	TTL/ST	—	从选择 1 输入。
RC7/ANC7	RC7	TTL/ST	CMOS	通用 I/O。
	ANC7	AN	—	ADC 通道 C7 输入。
VDD	VDD	电源	—	正电源。
Vss	Vss	电源	—	参考地。

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平施密特触发器输入 I²C = I²C 电平施密特触发器输入
HV = 高电压 XTAL = 晶振电平

- 注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以将输入移至任何其他引脚。请参见寄存器 13-2。
2: 所有引脚均默认输出至端口数据锁存器。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见寄存器 13-2。
3: 这些 I²C 功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F18324/18344

表1-3: PIC16(L)F18344引脚说明 (续)

名称	功能	输入类型	输出类型	说明
OUT ⁽²⁾	C1	—	CMOS	比较器C1输出。
	C2	—	CMOS	比较器C2输出。
	NCO1	—	CMOS	数控振荡器输出。
	DSM	—	CMOS	数字信号调制器输出。
	TMR0	—	CMOS	Timer0时钟输出。
	CCP1	—	CMOS	捕捉/比较/PWM 1输出。
	CCP2	—	CMOS	捕捉/比较/PWM 2输出。
	CCP3	—	CMOS	捕捉/比较/PWM 3输出。
	CCP4	—	CMOS	捕捉/比较/PWM 4输出。
	PWM5	—	CMOS	脉宽调制器5的输出。
	PWM6	—	CMOS	脉宽调制器6的输出。
	CWG1A	—	CMOS	互补波形发生器1的输出A。
	CWG2A	—	CMOS	互补波形发生器2的输出A。
	CWG1B	—	CMOS	互补波形发生器1的输出B。
	CWG2B	—	CMOS	互补波形发生器2的输出B。
	CWG1C	—	CMOS	互补波形发生器1的输出C。
	CWG2C	—	CMOS	互补波形发生器2的输出C。
	CWG1D	—	CMOS	互补波形发生器1的输出D。
	CWG2D	—	CMOS	互补波形发生器2的输出D。
	SDA1 ⁽³⁾	I ² C	OD	I ² C数据输出。
	SCL1 ⁽³⁾	I ² C	OD	I ² C时钟输出。
	SDO1	—	CMOS	SPI1数据输出。
	SCK1	—	CMOS	SPI1时钟输出。
	TX/CK	—	CMOS	异步发送数据/同步时钟输出。
	DT	—	CMOS	EUSART同步数据输出。
	CLC1OUT	—	CMOS	可配置逻辑单元1源输出。
	CLC2OUT	—	CMOS	可配置逻辑单元2源输出。
	CLC3OUT	—	CMOS	可配置逻辑单元3源输出。
	CLC4OUT	—	CMOS	可配置逻辑单元4源输出。
	CLKR	—	CMOS	时钟参考输出。

图注: AN = 模拟输入或输出 CMOS = CMOS兼容输入或输出 OD = 漏极开路
TTL = TTL兼容输入 ST = CMOS电平施密特触发器输入 I²C = I²C电平施密特触发器输入
HV = 高电压 XTAL = 晶振电平

- 注 1:** 默认外设输入。通过PPS输入选择寄存器，可以将输入移至任何其他引脚。请参见寄存器13-2。
2: 所有引脚均默认输出至端口数据锁存器。通过PPS输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见寄存器13-2。
3: 这些I²C功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F18324/18344系列

2.0 PIC16(L)F183XX单片机入门指南

2.1 基本连接要求

在开始使用PIC16(L)F183XX系列8位单片机进行开发之前，需要注意最基本的器件引脚连接要求。

以下引脚必须始终连接：

- 所有VDD和VSS引脚（见第2.2节“电源引脚”）
- MCLR引脚（配置为进行外部操作时）（见第2.3节“主复位（MCLR）引脚”）

如果最终应用中使用以下引脚，也必须进行连接：

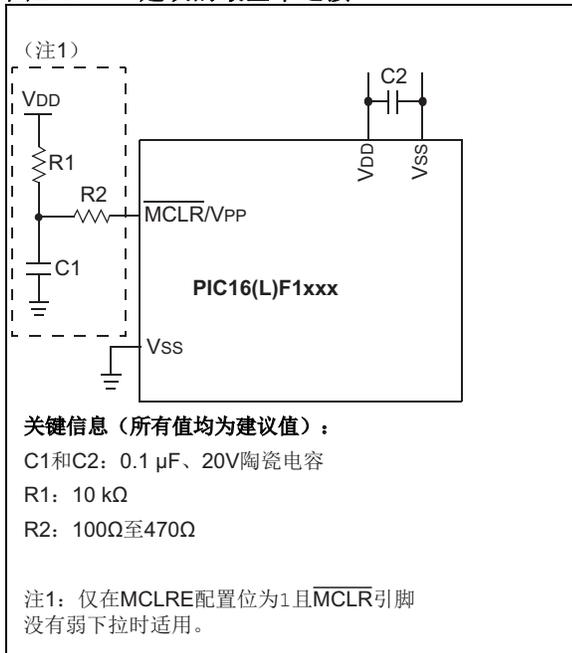
- 用于在线串行编程（In-Circuit Serial Programming™，ICSP™）和调试目的的ICSPCLK/ICSPDAT引脚（见第2.4节“ICSP™引脚”）
- OSC1和OSC2引脚（使用外部振荡器源时）（见第2.5节“外部振荡器引脚”）

此外，可能还需要以下引脚：

- 实现模拟模块的外部参考电压时，需要使用VREF+/VREF-引脚

必需的最基本连接如图2-1所示。

图2-1： 建议的最基本连接



2.2 电源引脚

2.2.1 去耦电容

需要在每对电源引脚（VDD和VSS）上使用去耦电容。所有VDD和VSS引脚都必须连接。任何引脚都不得悬空。

使用去耦电容时，需要考虑以下标准：

- **电容的类型和电容值：**建议使用0.1 μ F（100 nF）、10-20V电容。该电容应为低ESR器件，谐振频率为200 MHz或更高。建议使用陶瓷电容。
- **在印刷电路板上的放置：**去耦电容应尽可能靠近引脚。建议将电容与器件放置在电路板的同一层。如果空间受限，可以使用过孔将电容放置在PCB的另一层，但请确保从引脚到电容的走线长度不超过0.25英寸（6毫米）。
- **高频噪声处理：**如果电路板会遇到高频噪声（频率超过数十MHz），应为上述去耦电容并联一个陶瓷电容。这个陶瓷电容的电容值可以介于0.001 μ F和0.01 μ F之间。请将第二个电容放置在靠近每个主去耦电容的位置。在高速电路设计中，应考虑尽可能靠近电源和接地引脚放置这一对电容（例如，0.1 μ F与0.001 μ F的电容并联）。
- **最大程度提高性能：**对于从电源电路开始的电路板布线，需要将电源和返回走线先连接到去耦电容，然后再与器件引脚连接。这可以确保去耦电容是电源链中的第一个元件。同等重要的是尽可能减小电容和电源引脚之间的走线长度，从而降低PCB走线电感。

2.2.2 大电容

对于电源走线长度超出6英寸的电路板，建议对集成电路（包括单片机）使用大电容来提供本地电源。大电容的电容值应根据连接电源与器件的走线电阻和应用中器件消耗的最大电流确定。也就是说，选择的大电容需要满足器件的可接受电压骤降要求。典型值的范围为4.7 μ F至47 μ F。

2.3 主复位 ($\overline{\text{MCLR}}$) 引脚

MCLR引脚提供三种特定的器件功能:

- 器件复位 ($\text{MCLRE} = 1$ 时)
- 数字输入引脚 ($\text{MCLRE} = 0$ 时)
- 器件编程和调试

如果最终应用中不需要编程和调试,则将MCLRE配置位置1并将引脚用作数字输入,或者将MCLRE配置位清零并将引脚悬空以使用内部弱上拉。添加其他元件有助于防止应用因电压骤降引起的误复位而受到影响,这是十分有益的。图2-1给出了典型配置。根据应用的不同要求,可以实现其他电路设计。

在编程和调试过程中,必须考虑到引脚上可能会增加的电阻和电容。器件编程器和调试器会驱动MCLR引脚。因此,特定电压(V_{IH} 和 V_{IL})和快速信号跳变一定不能受到不利影响。所以在编程和调试操作期间,编程器MCLR/VPP输出应直接连接到该引脚,以使R1将电容C1与MCLR引脚隔离。

任何与MCLR引脚相关的元件都应放置在距离引脚0.25英寸(6毫米)以内的位置。

2.4 ICSP™ 引脚

ICSPCLK/ICSPDAT引脚用于在线串行编程(ICSP)和调试目的。建议尽可能减小ICSP连接器与器件ICSP引脚之间的走线长度。如果ICSP连接器会遇到ESD事件,则建议添加一个串联电阻,电阻值为几十欧姆,不要超出100Ω。

建议不要在ICSPCLK和ICSPDAT引脚上连接上拉电阻、串联二极管和电容,因为它们会影响编程器/调试器与器件的通信。如果应用需要此类分立元件,则在编程期间应通过应用和器件引脚之间的电阻将它们与编程器隔离,或者将它们从电路中移除。或者,请参见相应器件闪存编程规范中的交流/直流特性与时序要求信息,了解关于容性负载限制、引脚输入高电压(V_{IH})和输入低电压(V_{IL})要求的信息。

要进行器件仿真,请确保编程到器件中的“通信通道选择”(即ICSPCLK/ICSPDAT引脚)符合与Microchip调试器/仿真器工具的ICSP物理连接。

有关可用的Microchip开发工具连接要求的更多信息,请参见第37.0节“开发支持”。

PIC16(L)F18324/18344系列

2.5 外部振荡器引脚

许多单片机都有至少两个振荡器可供选择：高频主振荡器和低频辅助振荡器（有关详细信息，请参见第7.0节“振荡器模块”）。

振荡器电路与器件应放置在电路板的同一层。请将振荡器电路放置在靠近相应振荡器引脚的位置，使电路元件与引脚之间的距离不超过0.5英寸（12毫米）。负载电容应靠近振荡器本身，位于电路板的同一层。

请在振荡器电路周围使用接地覆铜区，以将其与周围电路隔离。接地覆铜区应与MCU地直接连接。不要在接地覆铜区内安排任何信号走线或电源走线。此外，如果使用双面电路板，请避免在电路板上晶振所在位置的背面有任何走线。

图2-2给出了布线建议。直插式封装可通过完全包围振荡器引脚的单面布线来处理。采用紧密排列的封装时，并非始终能够完全包围引脚和元件。合理的解决方案是将各个保护部分与镜像接地层相连。在所有情况下，保护走线都必须返回到地。

规划应用的布线和I/O分配时，需确保相邻端口引脚以及靠近振荡器的其他信号都是良好的（即，没有高频噪声、短时上升和下降噪声以及其他类似噪声）。

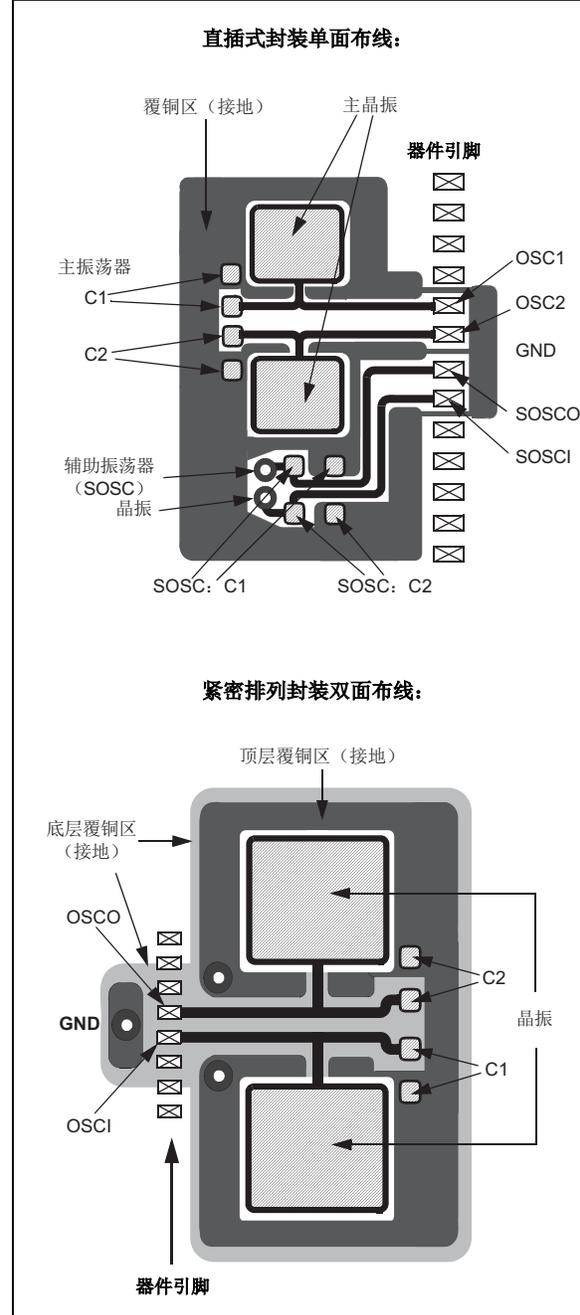
有关振荡器电路的更多信息和设计指南，请参见以下Microchip应用笔记，可从公司网站（www.microchip.com）下载：

- AN826, “Crystal Oscillator Basics and Crystal Selection for *rfPIC™* and *PICmicro®* Devices”
- AN849, “Basic *PICmicro®* Oscillator Design”
- AN943, “Practical *PICmicro®* Oscillator Analysis and Design”
- AN949, “Making Your Oscillator Work”

2.6 未用I/O

未用I/O引脚应配置为输出，并驱动为逻辑低电平状态。或者，在Vss和未用引脚之间连接一个1kΩ至10kΩ的电阻，并将输出驱动为逻辑低电平。

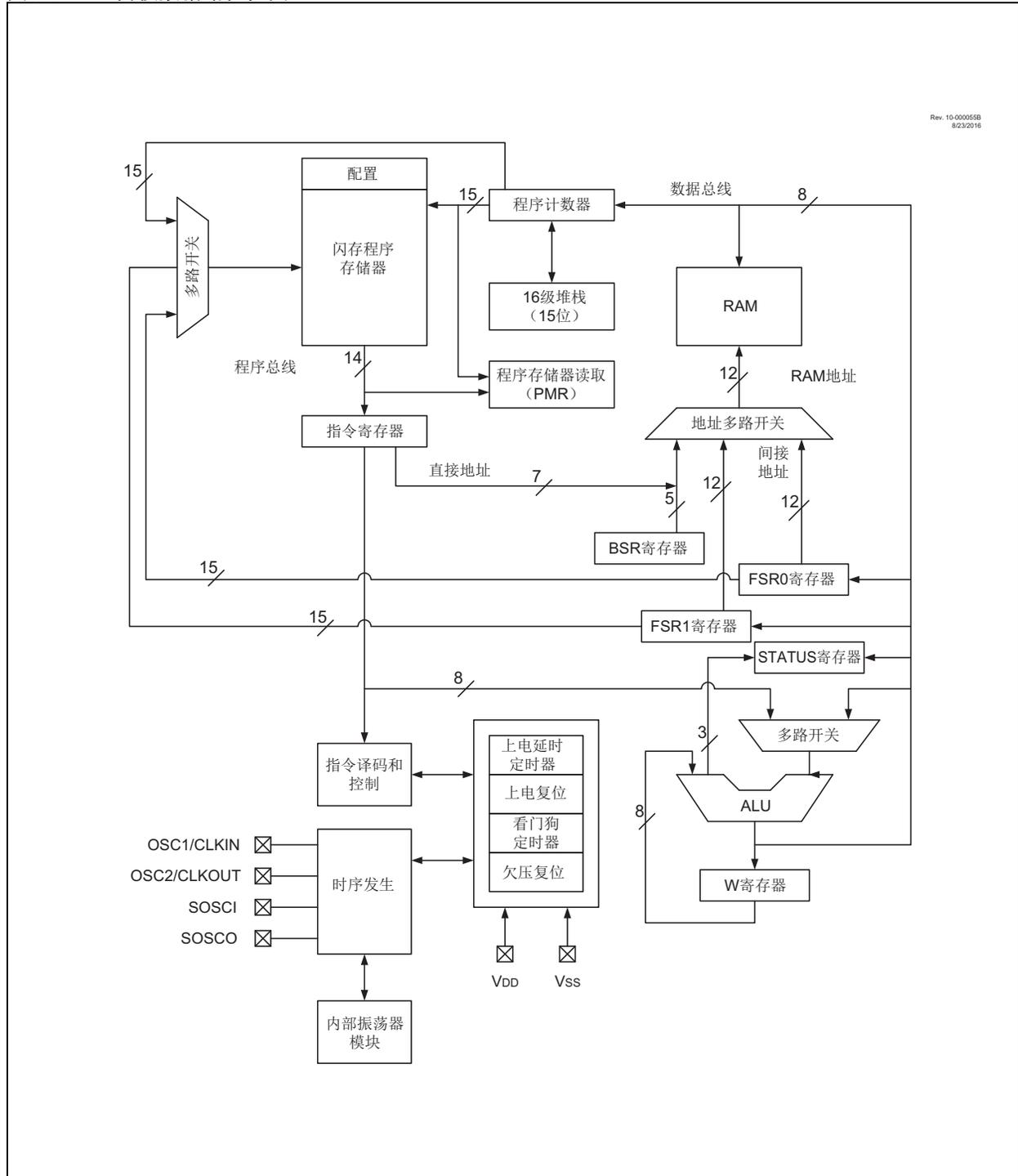
图2-2： 振荡器电路的建议布线方式



3.0 增强型中档CPU

本系列器件包含了增强型中档8位CPU内核。CPU具有48条指令。中断功能包含了自动现场保护功能。硬件堆栈为16级深，具有上溢和下溢复位功能。器件提供了直接寻址、间接寻址和相对寻址模式。用户可以通过两个文件选择寄存器（File Select Register, FSR）来读取程序和数据存储器。

图3-1: 内核数据路径框图



PIC16(L)F18324/18344

3.1 自动中断现场保护

在中断期间，器件会自动将一些寄存器保存到影子寄存器中，并在从中断返回时恢复这些寄存器。这可以节省堆栈空间和用户代码。更多信息，请参见[第8.5节“自动现场保护”](#)。

3.2 带有上溢和下溢的16级堆栈

这些器件具有15位宽、16字深的硬件堆栈存储器。在发生堆栈上溢或下溢时，PCON0寄存器中的相应位（STKOVF或STKUNF）会置1，如果使能堆栈上溢/下溢复位，则会导致软件复位。更多详细信息，请参见[第4.4节“堆栈”](#)。

3.3 文件选择寄存器

有两个16位文件选择寄存器（FSR）。FSR可以访问所有文件寄存器、程序存储器和数据EEPROM，支持对于所有存储器使用一个数据指针。当FSR指向程序存储器时，使用INDF的指令需要一个额外的指令周期来取数据。通用存储器现在也可以进行线性寻址，支持访问大于80字节的连续数据。此外，还有一些支持FSR的新指令。更多详细信息，请参见[第4.5节“间接寻址”](#)。

3.4 指令集

增强型中档CPU具有48条指令，用于支持CPU的特性。更多详细信息，请参见[第34.0节“指令集汇总”](#)。

4.0 存储器构成

这些器件包含以下存储器类型：

- 程序存储器
 - 配置字
 - 器件ID
 - 版本ID
 - 用户ID
 - 闪存程序存储器
- 数据存储器
 - 内核寄存器
 - 特殊功能寄存器
 - 通用RAM
 - 公共RAM
- 数据EEPROM

4.1 程序存储器构成

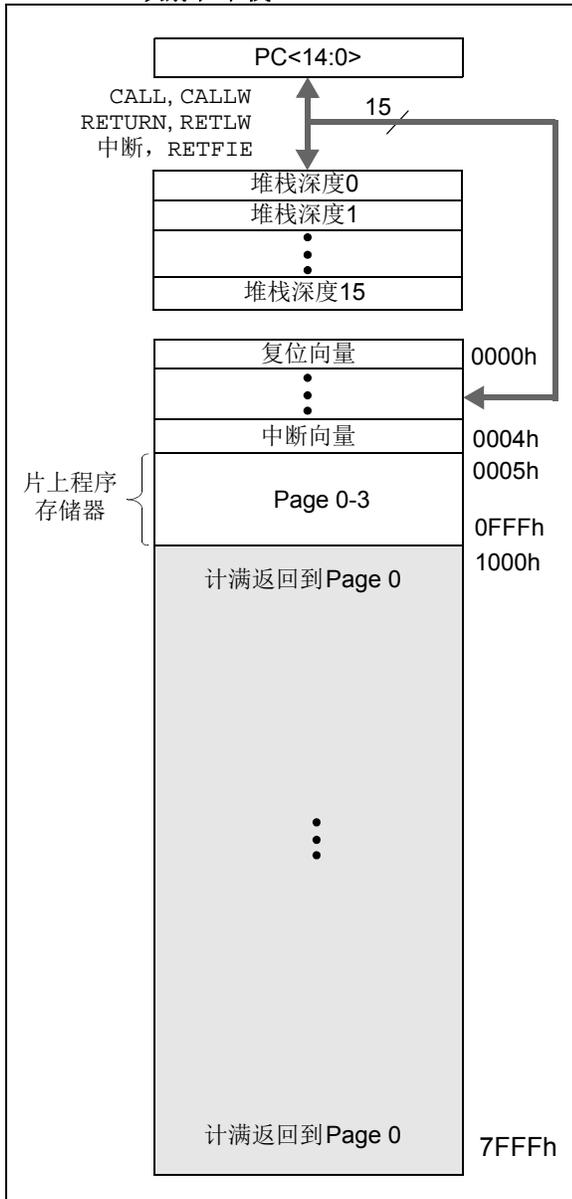
增强型中档内核具有一个15位程序计数器，能够寻址32K x 14的程序存储空间。表4-1列出了已实现的存储器大小。访问超出上述边界的存储单元，将返回到已实现的存储空间内。复位向量地址为0000h，中断向量地址为0004h（见图4-1）。

表4-1： 器件存储器容量和地址

器件	程序存储器容量（字）	程序存储器的最后一个地址
PIC16(L)F18324/18344	4096	0FFFh

PIC16(L)F18324/18344

图4-1: PIC16(L)F18324/18344的程序存储器映射和堆栈



4.1.1 将程序存储器作为数据进行读取

可以通过三种方法访问程序存储器中的常量。第一种方法是使用RETLW指令表。第二种方法是设置某个FSR，使之指向程序存储器。第三种方法是使用NVMCON寄存器访问程序存储器。

4.1.1.1 RETLW指令

RETLW指令可用于访问常量表。创建这种表的推荐方法如例4-1所示。

例4-1: RETLW指令

```
constants
    BRW                ;Add Index in W to
                       ;program counter to
                       ;select data

    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3

my_function
    ;... LOTS OF CODE...
    MOVLW    DATA_INDEX
    call constants
    ;... THE CONSTANT IS IN W
```

通过BRW指令，可以非常简单地实现这种表。如果代码必须保持对于前几代单片机的可移植性，则必须使用计算GOTO方法，因为BRW指令在某些器件（例如，PIC16F6XX、PIC16F7XX、PIC16F8XX和PIC16F9XX器件）中不可用。

4.1.1.2 通过FSR间接读取

通过将FSRxH寄存器的bit 7置1，并读取匹配的INDFx寄存器，可以将程序存储器作为数据进行访问。MOVIW指令会将所寻址字的低8位放入W寄存器。对程序存储器的写操作不能通过INDF寄存器执行。通过FSR读取程序存储器的指令需要一个额外的指令周期才能完成操作。例4-2给出了通过FSR读取程序存储器的代码。

如果某个标号指向程序存储器中的存储单元，HIGH伪指令会将bit 7置1。

例4-2: 通过FSR访问程序存储器

```
constants
    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;... LOTS OF CODE...
    MOVLW LOW constants
    MOVWF FSR1L
    MOVLW HIGH constants
    MOVWF FSR1H
    MOVIW 0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

4.1.1.3 NVMREG访问

NVMREG接口允许对可通过FSR访问的所有存储单元、用户ID存储单元和EEPROM进行读/写访问。NVMREG接口还提供对器件ID、版本ID和配置数据的只读访问。更多信息，请参见第11.4节“NVMREG访问”。

4.2 数据存储器构成

数据存储器划分为32个存储区，每个存储区有128字节。每个存储区都包含（图4-2）：

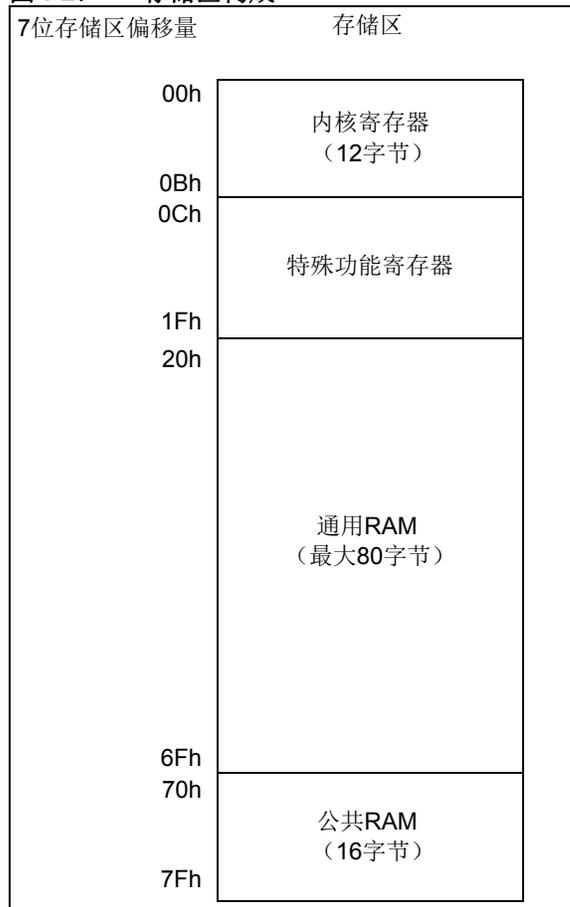
- 12个内核寄存器
- 特殊功能寄存器（Special Function Register, SFR）
- 最多80字节的通用RAM（General Purpose RAM, GPR）
- 16字节的公共RAM

4.2.1 存储区选择

工作存储区的选择通过向存储区选择寄存器（Bank Select Register, BSR）写入存储区编号来进行。未实现的存储单元将读为0。所有数据存储器可以直接访问（通过使用文件寄存器的指令），也可以通过两个文件选择寄存器（FSR）间接访问。更多信息，请参见第4.5节“间接寻址”。

数据存储器使用12位地址。地址的高5位用于定义存储区地址，低7位用于选择该存储区中的寄存器/RAM。

图4-2: 存储区构成



4.2.2 内核寄存器

内核寄存器包括直接影响器件基本操作的寄存器。内核寄存器占用每个数据存储区的前12个地址（地址x00h/x80h至x0Bh/x8Bh）。表4-2列出了这些寄存器。详细信息，请参见表4-4。

表4-2: 内核寄存器

地址	BANKx
x00h或x80h	INDF0
x01h或x81h	INDF1
x02h或x82h	PCL
x03h或x83h	STATUS
x04h或x84h	FSR0L
x05h或x85h	FSR0H
x06h或x86h	FSR1L
x07h或x87h	FSR1H
x08h或x88h	BSR
x09h或x89h	WREG
x0Ah或x8Ah	PCLATH
x0Bh或x8Bh	INTCON

PIC16(L)F18324/18344

4.2.2.1 STATUS 寄存器

STATUS 寄存器如寄存器 4-1 所示，包括：

- ALU 的算术运算状态
- 复位状态

与任何其他寄存器一样，STATUS 寄存器可作为任何指令的目标寄存器。如果一条影响 Z、DC 或 C 位的指令以 STATUS 寄存器作为目标寄存器，那么对这三个位的写操作将被禁止。这些位根据器件逻辑被置 1 或清零。而且， \overline{TO} 和 \overline{PD} 位均不可写。因此，当执行一条将 STATUS 寄存器作为目标寄存器的指令时，运行结果可能会与预想的不同。

例如，CLRF STATUS 将会清零高 3 位，并将 Z 位置 1。这将使 STATUS 寄存器中的值成为 000u u1uu（其中 u = 不变）。

因此，建议仅使用 BCF、BSF、SWAPF 和 MOVWF 指令来改变 STATUS 寄存器的值，因为这些指令不会影响任何状态位。关于其他不影响任何状态位的指令，请参见第 34.0 节“指令集汇总”。

注 1： 在减法运算中，C 和 DC 位分别作为借位位和半借位位。

寄存器 4-1: STATUS: 状态寄存器

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	\overline{TO}	\overline{PD}	Z	DC ⁽¹⁾	C ⁽¹⁾
bit 7							bit 0

图注：

R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

- bit 7-5 **未实现：** 读为 0
- bit 4 **\overline{TO} ：** 超时位
1 = 在上电或执行 CLRWDW 指令或 SLEEP 指令后
0 = 发生了 WDT 超时
- bit 3 **\overline{PD} ：** 掉电位
1 = 在上电或执行 CLRWDW 指令后
0 = 执行 SLEEP 指令
- bit 2 **Z：** 全零标志位
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
- bit 1 **DC：** 半进位/半借位位 (ADDWF、ADDLW、SUBLW 和 SUBWF 指令) ⁽¹⁾
1 = 结果的第 4 个低位发生了进位
0 = 结果的第 4 个低位未发生进位
- bit 0 **C：** 进位/借位位 ⁽¹⁾ (ADDWF、ADDLW、SUBLW 和 SUBWF 指令) ⁽¹⁾
1 = 结果的最高有效位发生了进位
0 = 结果的最高有效位未发生进位

注 1： 对于借位，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来执行的。对于移位指令 (RRF 和 RLF)，此位中将装入源寄存器的最高位或最低位。

4.2.3 特殊功能寄存器

特殊功能寄存器是由应用对器件中外设功能所需操作进行控制的寄存器。特殊功能寄存器占用每个数据存储区中内核寄存器之后的20字节（地址x0Ch/x8Ch至x1Fh/x9Fh），但存储区27、28和29（PPS和CLC寄存器）除外。本数据手册的相应外设章节中介绍了与外设操作相关的寄存器。

4.2.4 通用RAM

每个数据存储区中有最大80字节的GPR。通用RAM可以通过FSR以非分区方式访问。这可以简化对大存储器结构的访问。更多信息，请参见第4.5.2节“线性数据存储器”。

4.2.5 公共RAM

有16字节的公共RAM可以从所有存储区中进行访问。

4.2.6 器件存储器映射

表4-4给出了PIC16(L)F18324/18344的存储器映射。

表4-3: 特殊功能寄存器汇总, BANK 0-31 (所有存储区) (1)

存储区 偏移量	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR和 BOR时的值	所有其他 复位时的值	
所有存储区												
000h	INDF0	使用FSR0H/FSR0L的内容对数据存储器（非物理寄存器）进行寻址来寻址此存储单元								xxxx xxxx	xxxx xxxx	
001h	INDF1	使用FSR1H/FSR1L的内容对数据存储器（非物理寄存器）进行寻址来寻址此存储单元								xxxx xxxx	xxxx xxxx	
002h	PCL	程序计数器（Program Counter, PC）的最低有效字节								0000 0000	0000 0000	
003h	STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	---1 1000	---q quuu	
004h	FSR0L	间接数据存储器地址0低字节指针								0000 0000	uuuu uuuu	
005h	FSR0H	间接数据存储器地址0高字节指针								0000 0000	0000 0000	
006h	FSR1L	间接数据存储器地址1低字节指针								0000 0000	uuuu uuuu	
007h	FSR1H	间接数据存储器地址1高字节指针								0000 0000	0000 0000	
008h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
009h	WREG	工作寄存器								0000 0000	uuuu uuuu	
00Ah	PCLATH	—	程序计数器高7位的写缓冲区								-000 0000	-000 0000
00Bh	INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	00-- ---1	00-- ---1	

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 可从任何存储区访问这些寄存器。

表4-4: 特殊功能寄存器汇总 (BANK 0-31)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 0														
CPU 内核寄存器; 请参见表4-2了解详情														
00Ch	PORTA			—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu	
00Dh	PORTB	X	—	未实现									—	—
		—	X	RB7	RB6	RB5	RB4	—	—	—	—	—	xxxx ----	uuuu ----
00Eh	PORTC	X	—	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xxxx	--uu uuuu	
		—	X	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu	
00Fh	—	—	—	未实现									—	—
010h	PIR0			—	—	TMR0IF	IOCIF	—	—	—	INTF	--00 ---0	--00 ---0	
011h	PIR1			TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
012h	PIR2			TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	0000 0000	0000 0000	
013h	PIR3			OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	0000 0000	0000 0000	
014h	PIR4			CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	0000 0000	0000 0000	
015h	TMR0L			TMR0L<7:0>								xxxx xxxx	xxxx xxxx	
016h	TMR0H			TMR0H<7:0>								1111 1111	1111 1111	
017h	TOCON0			T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				0-00 0000	0-00 0000	
018h	TOCON1			T0CS<2:0>			T0ASYNC	T0CKPS<3:0>				0000 0000	0000 0000	
019h	TMR1L			TMR1L<7:0>								xxxx xxxx	uuuu uuuu	
01Ah	TMR1H			TMR1H<7:0>								xxxx xxxx	uuuu uuuu	
01Bh	T1CON			TMR1CS<1:0>		T1CKPS<1:0>		T1SOSC	T1SYNC	—	TMR1ON	0000 00-0	uuuu uu-u	
01Ch	T1GCON			TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		0000 0x00	uuuu uxuu	
01Dh	TMR2			TMR2<7:0>								0000 0000	0000 0000	
01Eh	PR2			PR2<7:0>								1111 1111	1111 1111	
01Fh	T2CON			—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000	

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 1														
CPU 内核寄存器; 请参见表4-2了解详情														
08Ch	TRISA			—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	--11 -111	--11 -111	
08Dh	TRISB	X	—	未实现									—	—
		—	X	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----	
08Eh	TRISC	X	—	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111	
		—	X	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111	
08Fh	—	—	—	未实现									—	—
090h	PIE0			—	—	TMR0IE	IOIE	—	—	—	INTE	--00 ---0	--00 ---0	
091h	PIE1			TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
092h	PIE2			TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	0000 0000	0000 0000	
093h	PIE3			OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	0000 0000	0000 0000	
094h	PIE4			CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	0000 0000	0000 0000	
095h	—	—	—	未实现									—	—
096h	—	—	—	未实现									—	—
097h	WDTCON			—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110	
098h	—	—	—	未实现									—	—
099h	—	—	—	未实现									—	—
09Ah	—	—	—	未实现									—	—
09Bh	ADRESL			ADRESL<7:0>								xxxx xxxx	uuuu uuuu	
09Ch	ADRESH			ADRESH<7:0>								xxxx xxxx	uuuu uuuu	
09Dh	ADCON0			CHS<5:0>						GO/DONE	ADON	0000 0000	0000 0000	
09Eh	ADCON1			ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		0000 -000	0000 -000	
09Fh	ADACT			—	—	—	ADACT<4:0>						---0 0000	---0 0000

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 2														
CPU 内核寄存器; 请参见表4-2了解详情														
10Ch	LATA			—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx	--uu -uuu	
10Dh	LATB	X	—	未实现									—	—
		—	X	LATB7	LATB6	LATB5	LATB4	—	—	—	—	xxxx ----	uuuu ----	
10Eh	LATC	X	—	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	--xx xxxx	--uu uuuu	
		—	X	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	uuuu uuuu	
10Fh	—	—	—	未实现									—	—
110h	—	—	—	未实现									—	—
111h	CM1CON0			C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC	00-0 -100	00-0 -100	
112h	CM1CON1			C1INTP	C1INTN	C1PCH<2:0>			C1NCH<2:0>			0000 0000	0000 0000	
113h	CM2CON0			C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC	00-0 -100	00-0 -100	
114h	CM2CON1			C2INTP	C2INTN	C2PCH<2:0>			C2MCH<2:0>			0000 0000	0000 0000	
115h	CMOUT			—	—	—	—	—	—	MC2OUT	MC1OUT	---- --00	---- --00	
116h	BORCON			SBOREN	—	—	—	—	—	—	BORRDY	1--- ---q	u--- ---u	
117h	FVRCON			FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		0q00 0000	0q00 0000	
118h	DACCON0			DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS	0-0- 00-0	0-0- 00-0	
119h	DACCON1			—	—	—	DAC1R<4:0>					---0 0000	---0 0000	
11Ah至11Fh	—	—	—	未实现									—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 3														
CPU 内核寄存器; 请参见表4-2了解详情														
18Ch	ANSELA			—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	--xx -xxx	--uu -uuu	
18Dh	ANSELB	X	—	未实现									—	—
		—	—	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	xxxx ----	uuuu ----	
18Eh	ANSELC	—	—	—	—	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	--xx xxxx	--uu uuuu	
		—	X	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	xxxx xxxx	uuuu uuuu	
18Fh	—	—	—	未实现									—	—
190h	—	—	—	未实现									—	—
191h	—	—	—	未实现									—	—
192h	—	—	—	未实现									—	—
193h	—	—	—	未实现									—	—
194h	—	—	—	未实现									—	—
195h	—	—	—	未实现									—	—
196h	—	—	—	未实现									—	—
197h	VREGCON ⁽¹⁾			—	—	—	—	—	—	VREGPM	保留	---- --01	---- --01	
198h	—	—	—	未实现									—	—
199h	RC1REG			RC1REG<7:0>									0000 0000	0000 0000
19Ah	TX1REG			TX1REG<7:0>									0000 0000	0000 0000
19Bh	SP1BRGL			SP1BRG<7:0>									0000 0000	0000 0000
19Ch	SP1BRGH			SP1BRG<15:8>									0000 0000	0000 0000
19Dh	RC1STA			SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x	
19Eh	TX1STA			CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TMRT	TX9D	0000 0010	0000 0010	
19Fh	BAUD1CON			ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00	

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 4														
CPU 内核寄存器; 请参见表4-2了解详情														
20Ch	WPUA			—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--00 0000	--00 0000	
20Dh	WPUB	X	—	未实现									—	—
		—	X	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	0000 ----	0000 ----	
20Eh	WPUC	X	—	—	—	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	--00 0000	--00 0000	
		—	X	WPUC7	WPUC6	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	0000 0000	0000 0000	
20Fh	—	—	—	未实现									—	—
210h	—	—	—	未实现									—	—
211h	SSP1BUF			SSP1BUF<7:0>								xxxx xxxx	uuuu uuuu	
212h	SSP1ADD			SSP1ADD<7:0>								0000 0000	0000 0000	
213h	SSP1MSK			SSP1MSK<7:0>								1111 1111	1111 1111	
214h	SSP1STAT			SMP	CKE	D \bar{A}	P	S	R \bar{W}	UA	BF	0000 0000	0000 0000	
215h	SSP1CON1			WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000	0000 0000	
216h	SSP1CON2			GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000	
217h	SSP1CON3			ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000	
218h至21Fh	—	—	—	未实现									—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值		
Bank 5															
CPU 内核寄存器; 请参见表4-2了解详情															
28Ch	ODCONA			—	—	ODCA5	ODCA4	—	ODCA2	ODCA1	ODCA0	--00 -000	--00 -000		
28Dh	ODCONB	X	—	未实现									—	—	
		—	X	ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—	0000 ----	0000 ----		
28Eh	ODCONC	X	—	—	—	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	--00 0000	--00 0000		
		—	X	ODCC7	ODCC6	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	0000 0000	0000 0000		
28Fh	—	—	—	未实现									—	—	
290h	—	—	—	未实现									—	—	
291h	CCPR1L			CCPR1<7:0>									xxxx xxxx	xxxx xxxx	
292h	CCPR1H			CCPR1<15:8>									xxxx xxxx	xxxx xxxx	
293h	CCP1CON			CCP1EN	—	CCP1OUT	CCP1FMT	CCP1MODE<3:0>				0-x0 0000	0-x0 0000		
294h	CCP1CAP			—	—	—	—	CCP1CTS<3:0>				---- 0000	---- xxxx		
295h	CCPR2L			CCPR2<7:0>									xxxx xxxx	xxxx xxxx	
296h	CCPR2H			CCPR2<15:8>									xxxx xxxx	xxxx xxxx	
297h	CCP2CON			CCP2EN	—	CCP2OUT	CCP2FMT	CCP2MODE<3:0>				0-x0 0000	0-x0 0000		
298h	CCP2CAP			—	—	—	—	CCP2CTS<3:0>				---- 0000	---- xxxx		
299h	—	—	—	未实现									—	—	
29Ah	—	—	—	未实现									—	—	
29Bh	—	—	—	未实现									—	—	
29Ch	—	—	—	未实现									—	—	
29Dh	—	—	—	未实现									—	—	
29Eh	—	—	—	未实现									—	—	
29Fh	CCPTMRS			C4TSEL<1:0>			C3TSEL<1:0>			C2TSEL<1:0>		C1TSEL<1:0>		0101 0101	0101 0101

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 6														
CPU 内核寄存器; 请参见表4-2了解详情														
30Ch	SLRCONA			—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	--11 -111	--11 -111	
30Dh	SLRCONB	X	—	未实现									—	—
		—	X	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	1111 ----	1111 ----	
30Eh	SLRCONC	X	—	—	—	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	--11 1111	--11 1111	
		—	X	SLRC7	SLRC6	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	1111 1111	
30Fh	—	—	—	未实现									—	—
310h	—	—	—	未实现									—	—
311h	CCPR3L			CCPR3<7:0>								xxxx xxxx	xxxx xxxx	
312h	CCPR3H			CCPR3<15:8>								xxxx xxxx	xxxx xxxx	
313h	CCP3CON			CCP3EN	—	CCP3OUT	CCP3FMT	CCP3MODE<3:0>			0-x0 0000	0-x0 0000		
314h	CCP3CAP			—	—	—	—	CCP3CTS<3:0>			---- 0000	---- xxxx		
315h	CCPR4L			CCPR4<7:0>								xxxx xxxx	xxxx xxxx	
316h	CCPR4H			CCPR4<15:8>								xxxx xxxx	xxxx xxxx	
317h	CCP4CON			CCP4EN	—	CCP4OUT	CCP4FMT	CCP4MODE<3:0>			0-x0 0000	0-x0 0000		
318h	CCP4CAP			—	—	—	—	CCP4CTS<3:0>			---- 0000	---- xxxx		
319h 至 31Fh	—	—	—	未实现									—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表 4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 7													
CPU 内核寄存器; 请参见表 4-2 了解详情													
38Ch	INLVLA			—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	--11 1111	--11 1111
38Dh	INVLVB	X	—	未实现								—	—
		—	X	INVLVB7	INVLVB6	INVLVB5	INVLVB4	—	—	—	—	1111 ----	1111 ----
38Eh	INLVLC	X	—	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	--11 1111	--11 1111
		—	X	INLVLC7	INLVLC6	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	1111 1111	1111 1111
38Fh	—	—	—	未实现								—	—
390h	—	—	—	未实现								—	—
391h	IOCAP			—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
392h	IOCAN			—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000	--00 0000
393h	IOCAF			—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
394h	IOCBP	X	—	未实现								—	—
		—	X	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	0000 ----
395h	IOCBN	X	—	未实现								—	—
		—	X	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	0000 ----
396h	IOCBF	X	—	未实现								—	—
		—	X	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	0000 ----
397h	IOCCP	X	—	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	--00 0000	--00 0000
		—	X	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
398h	IOCCN	X	—	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	--00 0000	--00 0000
		—	X	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
399h	IOCCF	X	—	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	--00 0000	--00 0000
		—	X	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为 0, r = 保留。阴影单元未实现, 读为 0。

注 1: 仅限 PIC16F18324/18344。

2: 寄存器可由用户和 ICD 调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 7													
CPU 内核寄存器; 请参见表4-2了解详情													
39Ah	CLKRCON			CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>			0--1 0000	0--1 0001
39Bh	—	—		未实现								—	—
39Ch	MDCON			MDEN	—	—	MDOPOL	MDOUT	—	—	MDBIT	0--0 0--0	0--0 0--0
39Dh	MDSRC			—	—	—	—		MDMS<3:0>			---- xxxxx	0--- uuuu
39Eh	MDCARH			—	MDCHPOL	MDCHSYNC	—		MDCH<3:0>			-xx- xxxxx	-uu- uuuu
39Fh	MDCARL			—	MDCLPOL	MDCLSYNC	—		MDCL<3:0>			-xx- xxxxx	-uu- uuuu

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 8													
CPU 内核寄存器; 请参见表4-2了解详情													
40Ch 至 410h	—	—	—	未实现								—	—
411h	TMR3L			TMR3L<7:0>								xxxx xxxx	uuuu uuuu
412h	TMR3H			TMR3H<7:0>								xxxx xxxx	uuuu uuuu
413h	T3CON			TMR3CS<1:0>		T3CKPS<1:0>		T3SOSC	$\overline{T3SYNC}$	—	TMR3ON	0000 00-0	uuuu uu-u
414h	T3GCON			TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ \overline{DONE}	T3GVAL	T3GSS<1:0>		0000 0x00	uuuu uxuu
415h	TMR4			TMR4<7:0>								0000 0000	0000 0000
416h	PR4			PR4<7:0>								1111 1111	1111 1111
417h	T4CON			—	T4OUTPS<3:0>			TMR4ON	T4CKPS<1:0>		-000 0000	-000 0000	
418h	TMR5L			TMR5L<7:0>								xxxx xxxx	uuuu uuuu
419h	TMR5H			TMR5H<7:0>								xxxx xxxx	uuuu uuuu
41Ah	T5CON			TMR5CS<1:0>		T5CKPS<1:0>		T5SOSC	$\overline{T5SYNC}$	—	TMR5ON	0000 00-0	uuuu uu-u
41Bh	T5GCON			TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/ \overline{DONE}	T5GVAL	T5GSS<1:0>		0000 0x00	uuuu uxuu
41Ch	TMR6			TMR6<7:0>								0000 0000	0000 0000
41Dh	PR6			PR6<7:0>								1111 1111	1111 1111
41Eh	T6CON			—	T6OUTPS<3:0>			TMR6ON	T6CKPS<1:0>		-000 0000	-000 0000	
41Fh	—	—	—	未实现								—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 9													
CPU 内核寄存器; 请参见表4-2了解详情													
48Ch 至 497h	—	—		未实现								—	—
498h	NCO1ACCL			NCO1ACC<7:0>								0000 0000	0000 0000
499h	NCO1ACCH			NCO1ACC<15:8>								0000 0000	0000 0000
49Ah	NCO1ACCU			—	—	—	—	NCO1ACC<19:16>				---- 0000	---- 0000
49Bh	NCO1INCL			NCO1INC<7:0>								0000 0001	0000 0001
49Ch	NCO1INCH			NCO1INC<15:8>								0000 0000	0000 0000
49Dh	NCO1INCUC			—	—	—	—	NCO1INC<19:16>				---- 0000	---- 0000
49Eh	NCO1CON			N1EN	—	N1OUT	N1POL	—	—	—	N1PFM	0-00 ---0	0-00 ---0
49Fh	NCO1CLK			N1PWS<2:0>			—	—	—	N1CKS<1:0>		000- --00	000- --00

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表 4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324 PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR时的值	所有其他 复位时的值
Bank 10-11												
CPU 内核寄存器; 请参见表 4-2 了解详情												
50Ch 至 51Fh	—	—	未实现								—	—
58Ch 至 59Fh	—	—	未实现								—	—
Bank 12												
60Ch	—	—	未实现								—	—
60Dh	—	—	未实现								—	—
60Eh	—	—	未实现								—	—
60Fh	—	—	未实现								—	—
610h	—	—	未实现								—	—
611h	—	—	未实现								—	—
612h	—	—	未实现								—	—
613h	—	—	未实现								—	—
614h	—	—	未实现								—	—
615h	—	—	未实现								—	—
616h	—	—	未实现								—	—
617h	PWM5DCL		PWM5DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
618h	PWM5DCH		PWM5DC<9:2>								xxxx xxxx	uuuu uuuu
619h	PWM5CON		PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—	0-00 ----	0-00 ----
61Ah	PWM6DCL		PWM6DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
61Bh	PWM6DCH		PWM6DC<9:2>								xxxx xxxx	uuuu uuuu
61Ch	PWM6CON		PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—	0-00 ----	0-00 ----
61Dh 至 61Eh	—	—	未实现								—	—
61Fh	PWMTMRS		—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		---- 0101	---- 0101

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为 0, r = 保留。阴影单元未实现, 读为 0。

注 1: 仅限 PIC16F18324/18344。

2: 寄存器可由用户和 ICD 调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 13													
CPU 内核寄存器; 请参见表4-2了解详情													
68Ch	—	—	—	未实现								—	—
68Dh	—	—	—	未实现								—	—
68Eh	—	—	—	未实现								—	—
68Fh	—	—	—	未实现								—	—
690h	—	—	—	未实现								—	—
691h	CWG1CLKCON			—	—	—	—	—	—	—	CS	---- --0	---- --0
692h	CWG1DAT			—	—	—	—	DAT<3:0>				---- 0000	---- 0000
693h	CWG1DBR			—	—	DBR<5:0>					--00 0000	--00 0000	
694h	CWG1DBF			—	—	DBF<5:0>					--00 0000	--00 0000	
695h	CWG1CON0			EN	LD	—	—	—	MODE<2:0>			00-- -000	00-- -000
696h	CWG1CON1			—	—	IN	—	POLD	POLC	POLB	POLA	--x- 0000	--x- 0000
697h	CWG1AS0			SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	0001 01--	0001 01--
698h	CWG1AS1			—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	---0 0000	---0 0000
699h	CWG1STR			OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	0000 0000	0000 0000
69Ah 至 69Fh	—	—	—	未实现								—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表 4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324 PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他 复位时的值
Bank 14												
CPU 内核寄存器; 请参见表 4-2 了解详情												
70Ch	—	—	未实现								—	—
70Dh	—	—	未实现								—	—
70Eh	—	—	未实现								—	—
70Fh	—	—	未实现								—	—
710h	—	—	未实现								—	—
711h	CWG2CLKCON		—	—	—	—	—	—	—	CS	---- --0	---- --0
712h	CWG2DAT		—	—	—	—	DAT<3:0>				---- 0000	---- 0000
713h	CWG2DBR		—	—	DBR<5:0>						--00 0000	--00 0000
714h	CWG2DBF		—	—	DBF<5:0>						--00 0000	--00 0000
715h	CWG2CON0		EN	LD	—	—	—	MODE<2:0>			00-- -000	00-- -000
716h	CWG2CON1		—	—	IN	—	POLD	POLC	POLB	POLA	--x- 0000	--x- 0000
717h	CWG2AS0		SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	0001 01--	0001 01--
718h	CWG2AS1		—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	---0 0000	---0 0000
719h	CWG2STR		OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	0000 0000	0000 0000
71Ah 至 71Fh	—	—	未实现								—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为 0, r = 保留。阴影单元未实现, 读为 0。

注 1: 仅限 PIC16F18324/18344。

2: 寄存器可由用户和 ICD 调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 15-16														
CPU 内核寄存器; 请参见表4-2了解详情														
78Ch至79Fh	—	—	—	未实现								—	—	
80Ch至81Fh	—	—	—	未实现								—	—	
Bank 17														
88Ch	—	—	—	未实现								—	—	
88Dh	—	—	—	未实现								—	—	
88Eh	—	—	—	未实现								—	—	
88Fh	—	—	—	未实现								—	—	
890h	—	—	—	未实现								—	—	
891h	NVMADRL			NVMADR<7:0>								0000 0000	0000 0000	
892h	NVMADRH			—	NVMADR<14:8>								1000 0000	1000 0000
893h	NVMDATL			NVMDAT<7:0>								0000 0000	0000 0000	
894h	NVMDATH			—	—	NVMDAT<13:8>						--00 0000	--00 0000	
895h	NVMCON1			—	NVMREGS	LWLO	FREE	WRERR	WREN	WR	RD	-000 x000	-000 q000	
896h	NVMCON2			NVMCON2								0000 0000	0000 0000	
897h	—	—	—	未实现								—	—	
898h	—	—	—	未实现								—	—	
899h	—	—	—	未实现								—	—	
89Ah	—	—	—	未实现								—	—	
89Bh	PCON0			STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	00-1 110q	qbb b-bb qbbu	
89Ch至89Fh	—	—	—	未实现								—	—	

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表 4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324 PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR时的值	所有其他 复位时的值
Bank 18												
CPU 内核寄存器; 请参见表 4-2 了解详情												
90Ch	—	—	未实现								—	—
90Dh	—	—	未实现								—	—
90Eh	—	—	未实现								—	—
90Fh	—	—	未实现								—	—
910h	—	—	未实现								—	—
911h	PMD0		SYSCMD	FVRMD	—	—	—	NVMMD	CLKRMD	IOCMD	00-- 000	00-- 000
912h	PMD1		NCOMD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD	0--- 000	0--- 000
913h	PMD2		—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	—	-00- --0-	-00- --0-
914h	PMD3		CWG2MD	CWG1MD	PWM6MD	PWM5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	-000 --00	-000 --00
915h	PMD4		—	—	UART1MD	—	—	MSSP2MD	MSSP1MD	—	--0- --0-	--0- --0-
916h	PMD5		—	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	DSMMD	---- 000	---- 000
917h	—	—	未实现								—	—
918h	CPUDOZE		IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			000- 000	000- 000
919h	OSCCON1		—	NOSC<2:0>			NDIV<3:0>				-qqq 0000	-qqq 0000
91Ah	OSCCON2		—	COSC<2:0>			CDIV<3:0>				-qqq 0000	-qqq 0000
91Bh	OSCCON3		CSWHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—	0000 0---	0000 0---
91Ch	OSCSTAT1		EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLL	qq-p qq-p	qq-p qq-p
91Dh	OSCCEN		EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—	00-0 00--	00-0 00--
91Eh	OSCTUNE		—	—	HFTUN<5:0>						--10 0000	--10 0000
91Fh	OSCFRQ		—	—	—	—	HFFRQ<3:0>				---- 0110	---- 0110

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为 0, r = 保留。阴影单元未实现, 读为 0。

注 1: 仅限 PIC16F18324/18344。

2: 寄存器可由用户和 ICD 调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 19-27													
CPU 内核寄存器; 请参见表4-2了解详情													
98Ch 至 9EFh	—	—	—	未实现								—	—
A0Ch 至 A6Fh	—	—	—	未实现								—	—
A8Ch 至 AEFh	—	—	—	未实现								—	—
B0Ch 至 B6Fh	—	—	—	未实现								—	—
B8Ch 至 BEFh	—	—	—	未实现								—	—
C0Ch 至 C6Fh	—	—	—	未实现								—	—
C8Ch 至 CEFh	—	—	—	未实现								—	—
D0Ch 至 D6Fh	—	—	—	未实现								—	—
D8Ch 至 DEFh	—	—	—	未实现								—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限 PIC16F18324/18344。

2: 寄存器可由用户和ICD 调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 28													
CPU 内核寄存器; 请参见表4-2了解详情													
E0Ch	—	—	—	—	—	—	—	—	—	—	—	—	—
E0Dh	—	—	—	—	—	—	—	—	—	—	—	—	—
E0Eh	—	—	—	—	—	—	—	—	—	—	—	—	—
E0Fh	PPSLOCK			—	—	—	—	—	—	—	PPSLOCKED	---- --0	---- --0
E10h	INTPPS			—	—	—	—	—	—	—	INTPPS<4:0>	---0 0010	---u uuuu
E11h	T0CKIPPS			—	—	—	—	—	—	—	T0CKIPPS<4:0>	---0 0010	---u uuuu
E12h	T1CKIPPS			—	—	—	—	—	—	—	T1CKIPPS<4:0>	---0 0101	---u uuuu
E13h	T1GPPS			—	—	—	—	—	—	—	T1GPPS<4:0>	---0 0100	---u uuuu
E14h	CCP1PPS			—	—	—	—	—	—	—	CCP1PPS<4:0>	---1 0011	---u uuuu
E15h	CCP2PPS			—	—	—	—	—	—	—	CCP2PPS<4:0>	---1 0101	---u uuuu
E16h	CCP3PPS			—	—	—	—	—	—	—	CCP3PPS<4:0>	---0 0010	---u uuuu
E17h	CCP4PPS	X	—	—	—	—	—	—	—	—	CCP4PPS<4:0>	---1 0001	---u uuuu
		—	X	—	—	—	—	—	—	—	CCP4PPS<4:0>	---0 0100	---u uuuu
E18h	CWG1PPS			—	—	—	—	—	—	—	CWG1PPS<4:0>	---0 0010	---u uuuu
E19h	CWG2PPS			—	—	—	—	—	—	—	CWG2PPS<4:0>	---0 0010	---u uuuu
E1Ah	MDCIN1PPS			—	—	—	—	—	—	—	MDCIN1PPS<4:0>	---1 0010	---u uuuu
E1Bh	MDCIN2PPS			—	—	—	—	—	—	—	MDCIN2PPS<4:0>	---1 0101	---u uuuu
E1Ch	MDMINPPS			—	—	—	—	—	—	—	MDMINPPS<4:0>	---1 0011	---u uuuu
E1Dh至 E20h	—	—	—	—	—	—	—	—	—	—	—	—	—

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 28													
CPU 内核寄存器; 请参见表4-2了解详情													
E21h	SSP1DATPPS	X	—	—	—	—	—	—	—	—	—	---1 0001	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 1100	---u uuuu
E22h	SSP1SSPPS	X	—	—	—	—	—	—	—	—	—	---1 0011	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---1 0100	---u uuuu
E23h	—	—	—	未实现							—	—	
E24h	RXPPS	X	—	—	—	—	—	—	—	—	—	---1 0101	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 1101	---u uuuu
E25h	TXPPS	X	—	—	—	—	—	—	—	—	—	---1 0100	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 1111	---u uuuu
E26h	—	—	—	未实现							—	—	
E27h	—	—	—	未实现							—	—	
E28h	CLCIN0PPS	X	—	—	—	—	—	—	—	—	—	---1 0011	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 0010	---u uuuu
E29h	CLCIN1PPS	X	—	—	—	—	—	—	—	—	—	---0 0100	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---1 0011	---u uuuu
E2Ah	CLCIN2PPS	X	—	—	—	—	—	—	—	—	—	---1 0001	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 1100	---u uuuu
E2Bh	CLCIN3PPS	X	—	—	—	—	—	—	—	—	—	---0 0101	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 1101	---u uuuu
E2Ch	T3CKIPPS	X	—	—	—	—	—	—	—	—	—	---1 0001	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 0101	---u uuuu
E2Dh	T3GPPS	X	—	—	—	—	—	—	—	—	—	---1 0001	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---1 0100	---u uuuu
E2Eh	T5CKIPPS	X	—	—	—	—	—	—	—	—	—	---1 0001	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---0 0101	---u uuuu
E2Fh	T5GPPS	X	—	—	—	—	—	—	—	—	—	---1 0001	---u uuuu
		—	X	—	—	—	—	—	—	—	—	---1 0100	---u uuuu

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表 4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 29													
CPU 内核寄存器; 请参见表 4-2 了解详情													
E8Dh	—	—	—	—				未实现				—	—
E8Eh	—	—	—	—				未实现				—	—
E8Fh	—	—	—	—				未实现				—	—
E90h	RA0PPS	—	—	—	—	—	RA0PPS<4:0>				---0 0000	---u uuuu	
E91h	RA1PPS	—	—	—	—	—	RA1PPS<4:0>				---0 0000	---u uuuu	
E92h	RA2PPS	—	—	—	—	—	RA2PPS<4:0>				---0 0000	---u uuuu	
E93h	—	—	—	—				未实现				—	—
E94h	RA4PPS	—	—	—	—	—	RA4PPS<4:0>				---0 0000	---u uuuu	
E95h	RA5PPS	—	—	—	—	—	RA5PPS<4:0>				---0 0000	---u uuuu	
E96h	—	—	—	—				未实现				—	—
E97h	—	—	—	—				未实现				—	—
E98h	—	—	—	—				未实现				—	—
E99h	—	—	—	—				未实现				—	—
E9Ah	—	—	—	—				未实现				—	—
E9Bh	—	—	—	—				未实现				—	—
E9Ch	RB4PPS	X	—	—				未实现				—	—
		—	X	—	—	—	RB4PPS<4:0>				---0 0000	---u uuuu	
E9Dh	RB5PPS	X	—	—				未实现				—	—
		—	X	—	—	—	RB5PPS<4:0>				---0 0000	---u uuuu	
E9Eh	RB6PPS	X	—	—				未实现				—	—
		—	X	—	—	—	RB6PPS<4:0>				---0 0000	---u uuuu	
E9Fh	RB7PPS	X	—	—				未实现				—	—
		—	X	—	—	—	RB7PPS<4:0>				---0 0000	---u uuuu	

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为 0, r = 保留。阴影单元未实现, 读为 0。

注 1: 仅限 PIC16F18324/18344。

2: 寄存器可由用户和 ICD 调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 29														
CPU 内核寄存器; 请参见表4-2了解详情														
EA0h	RC0PPS			—	—	—						RC0PPS<4:0>	---0 0000	---u uuuu
EA1h	RC1PPS			—	—	—						RC1PPS<4:0>	---0 0000	---u uuuu
EA2h	RC2PPS			—	—	—						RC2PPS<4:0>	---0 0000	---u uuuu
EA3h	RC3PPS			—	—	—						RC3PPS<4:0>	---0 0000	---u uuuu
EA4h	RC4PPS			—	—	—						RC4PPS<4:0>	---0 0000	---u uuuu
EA5h	RC5PPS			—	—	—						RC5PPS<4:0>	---0 0000	---u uuuu
EA6h	RC6PPS	X	—	未实现								—	—	
		—	X	—	—	—						RC6PPS<4:0>	---0 0000	---u uuuu
EA7h	RC7PPS	X	—	未实现								—	—	
		—	X	—	—	—						RC7PPS<4:0>	---0 0000	---u uuuu

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。
注 1: 仅限PIC16F18324/18344。
2: 寄存器可由用户和ICD调试器访问。

表 4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324 PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他 复位时的值
Bank 30												
CPU 内核寄存器; 请参见表 4-2 了解详情												
F0Ch	—	—	未实现								—	—
F0Dh	—	—	未实现								—	—
F0Eh	—	—	未实现								—	—
F0Fh	CLCDATA		—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	---- 0000	---- 0000
F10h	CLC1CON		LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			0-00 0000	0-00 0000
F11h	CLC1POL		LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	0--- xxxxx	0--- uuuu
F12h	CLC1SEL0		—	—	LC1D1S<5:0>					--xx xxxxx	--uu uuuu	
F13h	CLC1SEL1		—	—	LC1D2S<5:0>					--xx xxxxx	--uu uuuu	
F14h	CLC1SEL2		—	—	LC1D3S<5:0>					--xx xxxxx	--uu uuuu	
F15h	CLC1SEL3		—	—	LC1D4S<5:0>					--xx xxxxx	--uu uuuu	
F16h	CLC1GLS0		LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	xxxx xxxx	uuuu uuuu
F17h	CLC1GLS1		LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	xxxx xxxx	uuuu uuuu
F18h	CLC1GLS2		LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	xxxx xxxx	uuuu uuuu
F19h	CLC1GLS3		LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	xxxx xxxx	uuuu uuuu
F1Ah	CLC2CON		LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			0-00 0000	0-00 0000
F1Bh	CLC2POL		LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	0--- xxxxx	0--- uuuu
F1Ch	CLC2SEL0		—	—	LC2D1S<5:0>					--xx xxxxx	--uu uuuu	
F1Dh	CLC2SEL1		—	—	LC2D2S<5:0>					--xx xxxxx	--uu uuuu	
F1Eh	CLC2SEL2		—	—	LC2D3S<5:0>					--xx xxxxx	--uu uuuu	
F1Fh	CLC2SEL3		—	—	LC2D4S<5:0>					--xx xxxxx	--uu uuuu	

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为 0, r = 保留。阴影单元未实现, 读为 0。

注 1: 仅限 PIC16F18324/18344。

2: 寄存器可由用户和 ICD 调试器访问。

表4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324 PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR时的值	所有其他 复位时的值
Bank 30												
CPU 内核寄存器; 请参见表4-2了解详情												
F20h	CLC2GLS0		LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	xxxx xxxx	uuuu uuuu
F21h	CLC2GLS1		LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	xxxx xxxx	uuuu uuuu
F22h	CLC2GLS2		LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	xxxx xxxx	uuuu uuuu
F23h	CLC2GLS3		LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	xxxx xxxx	uuuu uuuu
F24h	CLC3CON		LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			0-00 0000	0-00 0000
F25h	CLC3POL		LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	0--- xxxx	0--- uuuu
F26h	CLC3SEL0		—	—	LC3D1S<5:0>						--xx xxxx	--uu uuuu
F27h	CLC3SEL1		—	—	LC3D2S<5:0>						--xx xxxx	--uu uuuu
F28h	CLC3SEL2		—	—	LC3D3S<5:0>						--xx xxxx	--uu uuuu
F29h	CLC3SEL3		—	—	LC3D4S<5:0>						--xx xxxx	--uu uuuu
F2Ah	CLC3GLS0		LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	xxxx xxxx	uuuu uuuu
F2Bh	CLC3GLS1		LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	xxxx xxxx	uuuu uuuu
F2Ch	CLC3GLS2		LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	xxxx xxxx	uuuu uuuu
F2Dh	CLC3GLS3		LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	xxxx xxxx	uuuu uuuu
F2Eh	CLC4CON		LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			0-00 0000	0-00 0000
F2Fh	CLC4POL		LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	0--- xxxx	0--- uuuu
F30h	CLC4SEL0		—	—	LC4D1S<5:0>						--xx xxxx	--uu uuuu
F31h	CLC4SEL1		—	—	LC4D2S<5:0>						--xx xxxx	--uu uuuu
F32h	CLC4SEL2		—	—	LC4D3S<5:0>						--xx xxxx	--uu uuuu
F33h	CLC4SEL3		—	—	LC4D4S<5:0>						--xx xxxx	--uu uuuu
F34h	CLC4GLS0		LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	xxxx xxxx	uuuu uuuu
F35h	CLC4GLS1		LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	xxxx xxxx	uuuu uuuu
F36h	CLC4GLS2		LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	xxxx xxxx	uuuu uuuu
F37h	CLC4GLS3		LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	xxxx xxxx	uuuu uuuu

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为0, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限PIC16F18324/18344。

2: 寄存器可由用户和ICD调试器访问。

表 4-4: 特殊功能寄存器汇总 (BANK 0-31) (续)

地址	名称	PIC16(L)F18324	PIC16(L)F18344	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 31 —— 除非另外声明, 否则只能从调试执行程序进行访问														
CPU 内核寄存器; 请参见表 4-2 了解详情														
F8Ch 至 FE3h	—	—	—	未实现								—	—	
FE4h ⁽²⁾	STATUS_SHAD			—	—	—	—	—	Z	DC	C	---- -xxx	---- -uuu	
FE5h ⁽²⁾	WREG_SHAD			工作寄存器的常规 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FE6h ⁽²⁾	BSR_SHAD			—	—	—	存储区选择寄存器的常规 (非 ICD) 影子寄存器					---x xxxx	---u uuuu	
FE7h ⁽²⁾	PCLATH_SHAD			—	程序计数器锁存器高字节寄存器的常规 (非 ICD) 影子寄存器								-xxx xxxx	-uuu uuuu
FE8h ⁽²⁾	FSR0L_SHAD			间接数据存储器地址 0 低字节指针寄存器的常规 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FE9h ⁽²⁾	FSR0H_SHAD			间接数据存储器地址 0 高字节指针寄存器的常规 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FEAh ⁽²⁾	FSR1L_SHAD			间接数据存储器地址 1 低字节指针寄存器的常规 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FEBh ⁽²⁾	FSR1H_SHAD			间接数据存储器地址 1 高字节指针寄存器的常规 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FECh	—	—		未实现								—	—	
FEDh ⁽²⁾	STKPTR			—	—	—	当前堆栈指针					---x xxxx	---1 1111	
FEEh ⁽²⁾	TOSL			栈顶低字节								xxxx xxxx	xxxx xxxx	
FEFh ⁽²⁾	TOSH			—	栈顶高字节								-xxx xxxx	-xxx xxxx

图注: x = 未知, u = 不变, q = 取决于具体条件, - = 未实现, 读为 0, r = 保留。阴影单元未实现, 读为 0。

注 1: 仅限 PIC16F18324/18344。

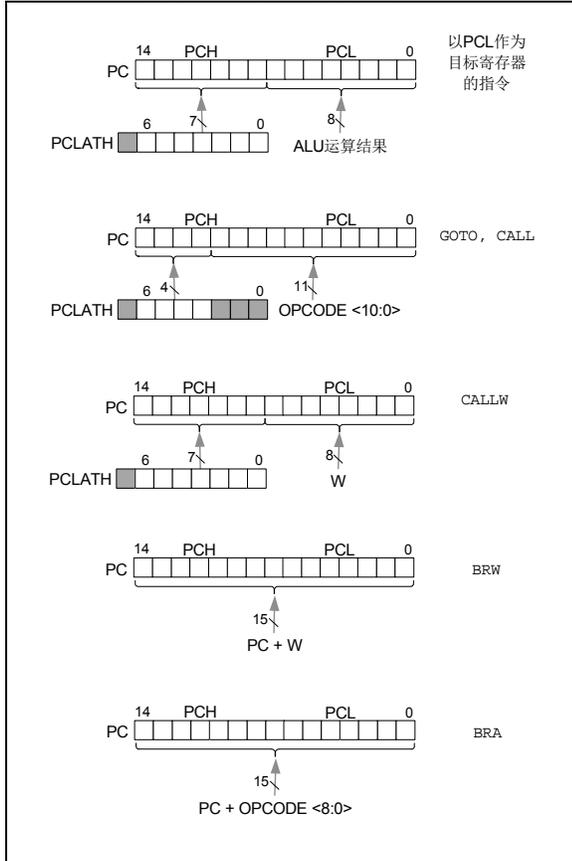
2: 寄存器可由用户和 ICD 调试器访问。

PIC16(L)F18324/18344

4.3 PCL和PCLATH

程序计数器（PC）为15位宽。其低字节来自可读写的PCL寄存器，高字节（PC<14:8>）来自PCLATH，不能直接读写。任何复位都将清零PC。图4-3显示了装载PC值的5种情形。

图4-3: 不同情形下PC的装载



4.3.1 修改PCL

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

4.3.2 计算GOTO

计算GOTO是通过向程序计数器加一个偏移量（ADDWF PCL）来实现的。当使用计算GOTO方法执行表读操作时，应注意表地址是否跨越了PCL存储器边界（每个存储块为256字节）。请参见应用笔记AN556“Implementing a Table Read”（DS00556）。

4.3.3 计算函数调用

利用计算函数CALL，程序可以维护一些函数表，并提供另一种执行状态机或查找表的方式。当使用计算函数CALL执行表读操作时，应注意表地址是否跨越了PCL存储器边界（每个存储块为256字节）。

如果使用CALL指令，PCH<2:0>和PCL寄存器中将装入CALL指令的操作数。PCH<6:3>中将装入PCLATH<6:3>。

CALLW指令通过将PCLATH和W组合构成目标地址来支持计算调用。计算CALLW通过向W寄存器中装入所需地址并执行CALLW来实现。PCL寄存器中装入W的值，PCH中装入PCLATH的值。

4.3.4 转移

转移指令会将一个偏移量与PC相加。这使得可以实现可重定位代码和跨越页边界的代码。存在两种转移形式：BRW和BRA。在两种形式中，PC都会发生递增，以便取下一条指令。使用任一转移指令时，都可以跨越PCL存储器边界。

如果使用BRW，则向W寄存器中装入所需的无符号地址，然后执行BRW。整个PC中将装入地址PC + 1 + W。

如果使用BRA，则将BRA指令操作数的9位有符号立即数值（k）加到PC。由于PC将发生递增以便取下一条指令，新地址将为PC + 1 + k。

4.4 堆栈

所有器件都具有16级x15位宽的硬件堆栈（见图4-4至图4-7）。堆栈既不占用程序存储空间，也不占用数据存储空间。当执行CALL或CALLW指令，或者中断导致程序转移时，PC值将被压入堆栈。而在执行RETURN、RETLW或RETFIE指令时，将从堆栈中弹出PC值。PCLATH不受压栈或出栈操作的影响。

如果STVREN位被编程为0（配置字），堆栈将作为循环缓冲区工作，并且在发生堆栈上溢或下溢时不会引起复位。这意味着在压栈16次后，第17次压入堆栈的值将会覆盖第一次压栈时所保存的值，而第18次压入堆栈的值将覆盖第二次压栈时所保存的值，依此类推。无论是否使能了复位，STKOVF和STKUNF标志位都将在上溢/下溢时置1。

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

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

4.4.1 访问堆栈

可通过TOSH、TOSL和STKPTR寄存器访问堆栈。STKPTR是堆栈指针的当前值。TOSH:TOSL寄存器对指向栈顶。两个寄存器都是可读写的。由于PC的大小为15位，所以TOS拆分为TOSH和TOSL。要访问堆栈，可以调整STKPTR的值（它会决定TOSH:TOSL位置），然后读/写TOSH:TOSL。STKPTR的宽度为5位，以允许检测上溢和下溢。

注： 在允许中断的情况下，在修改STKPTR时需要小心。

在正常程序操作期间，CALL、CALLW和中断会使STKPTR递增，而RETLW、RETURN和RETFIE会使STKPTR递减。可以随时读取STKPTR来查看剩余堆栈深度。STKPTR总是指向堆栈中当前使用的位置。因此，CALL或CALLW会先递增STKPTR，然后再将PC写入堆栈，而返回操作则会先写入PC，然后再递减STKPTR。

关于访问堆栈的示例，请参见图4-4至图4-7。

PIC16(L)F18324/18344

图4-4: 访问堆栈示例1

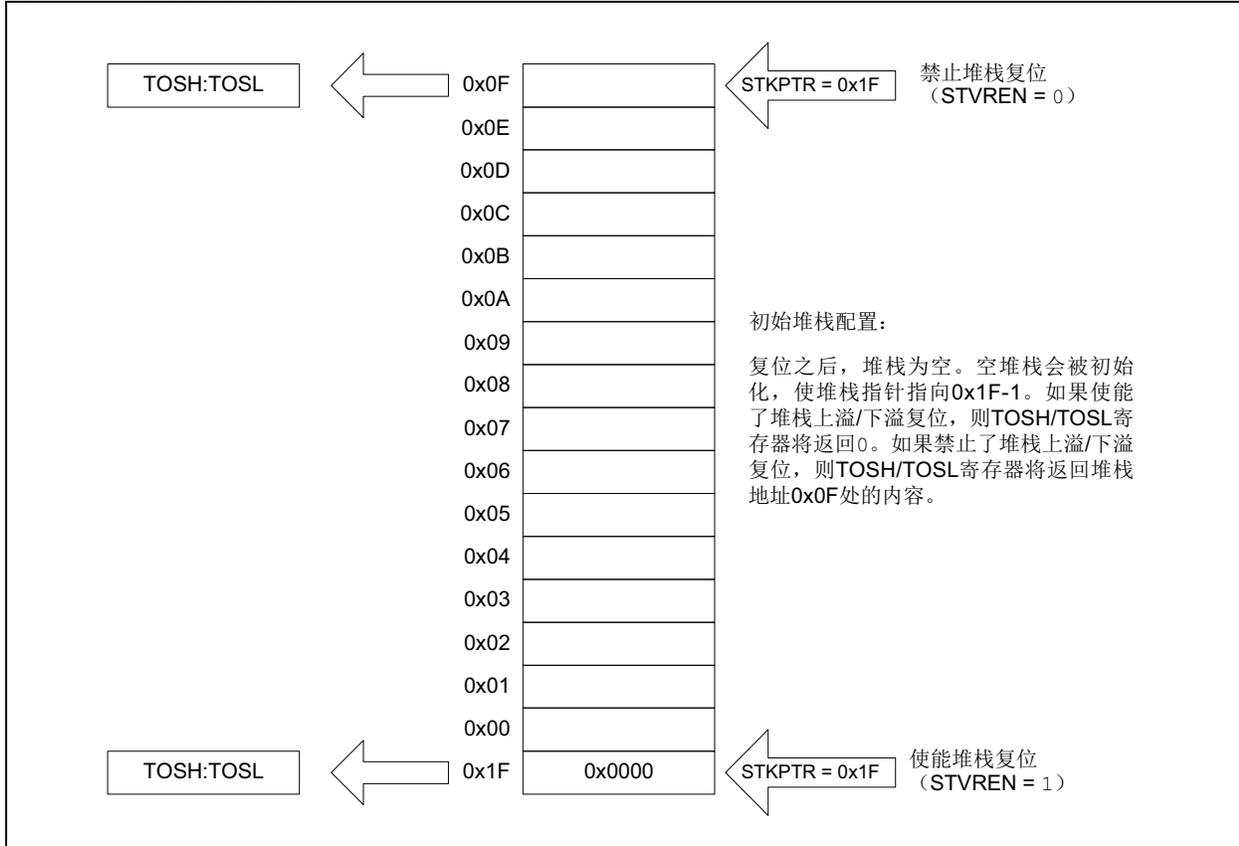


图4-5: 访问堆栈示例2

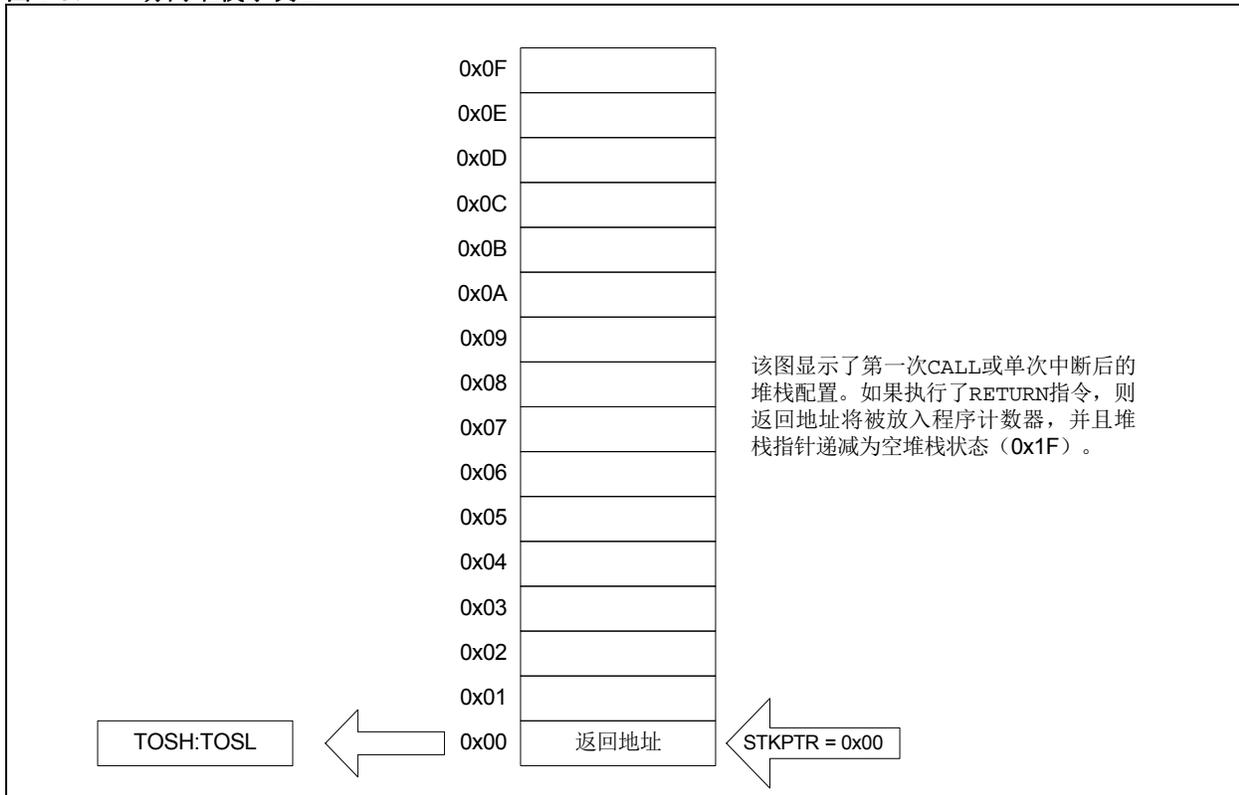


图 4-6: 访问堆栈示例 3

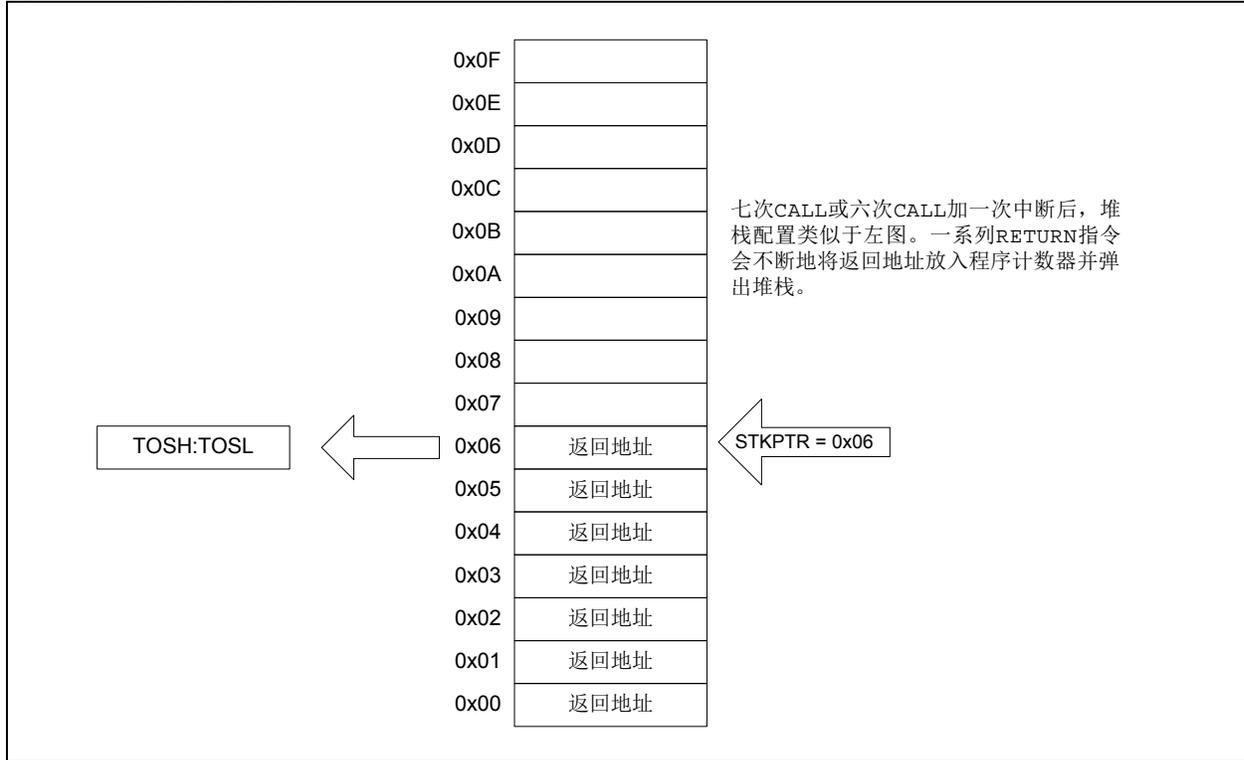
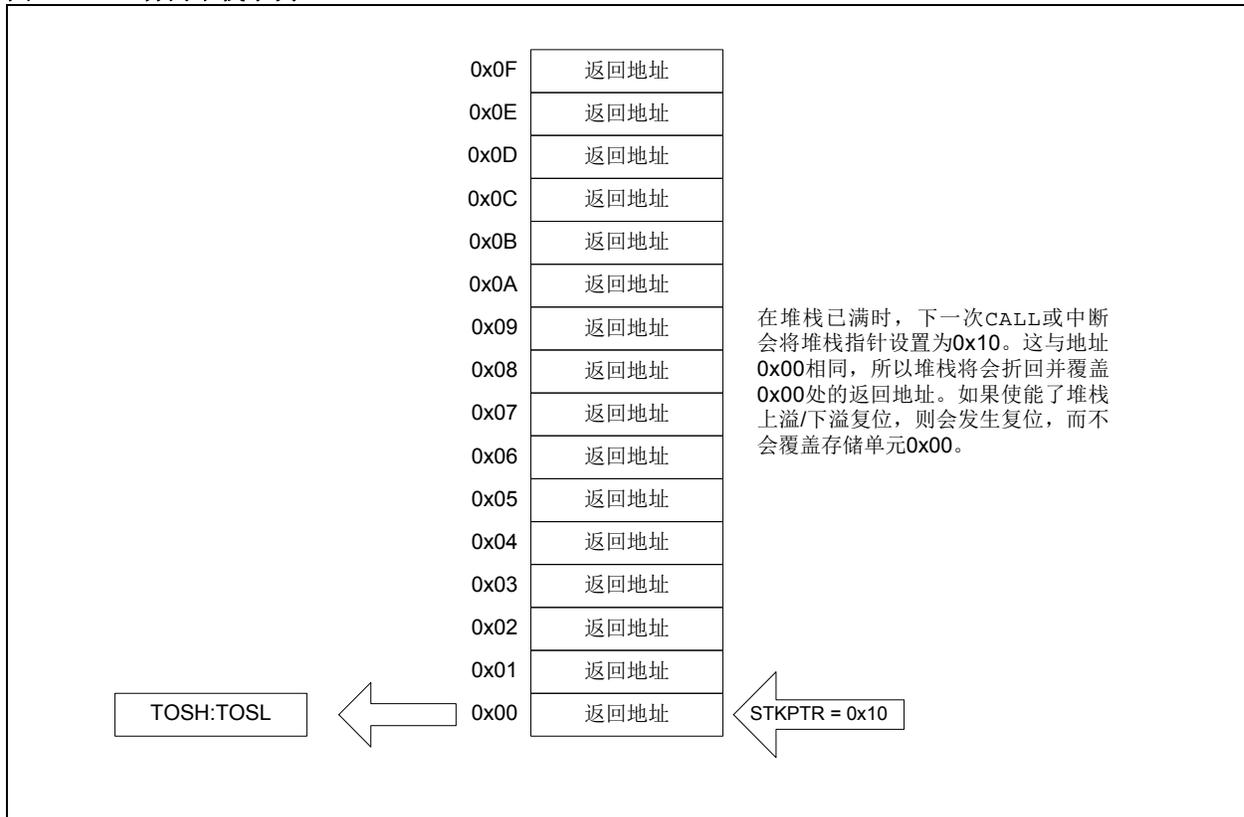


图 4-7: 访问堆栈示例 4



PIC16(L)F18324/18344

4.5 间接寻址

INDFn 寄存器不是物理寄存器。访问INDFn 寄存器的所有指令实际上访问的是由文件选择寄存器（FSR）指定的地址处的寄存器。如果FSRn 地址指定了两个INDFn 寄存器中的一个，则读操作将返回 0，写操作将不会发生（虽然状态位可能会受影响）。FSRn 寄存器值由FSRnH 和FSRnL 对构成。

FSR 寄存器构成一个 16 位地址，支持 65536 个存储单元的寻址空间。这些存储单元分为 4 个存储器区域：

- 传统/分区数据存储器
- 线性数据存储器
- 闪存程序存储器
- EEPROM

4.5.1 传统/分区数据存储器

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

图 4-8: 间接寻址

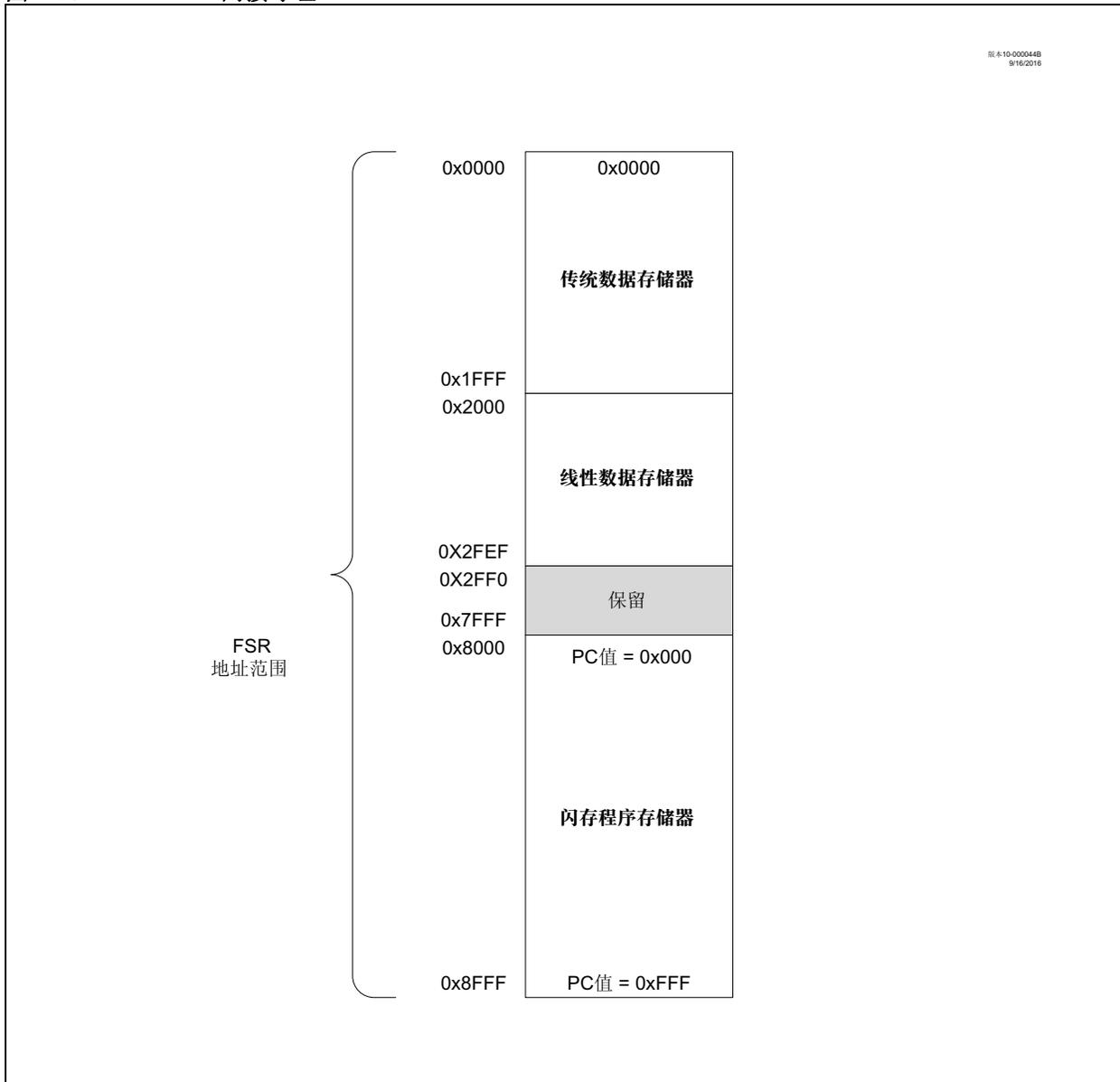
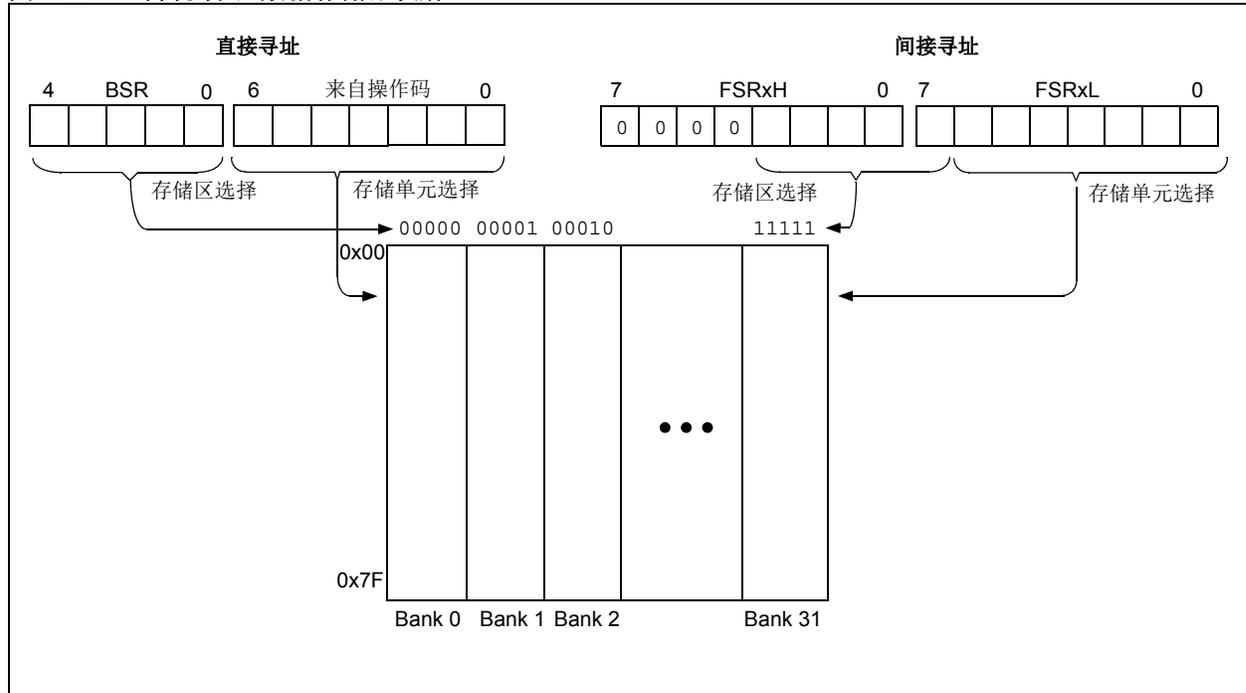


图4-9: 传统/分区数据存储器映射



PIC16(L)F18324/18344

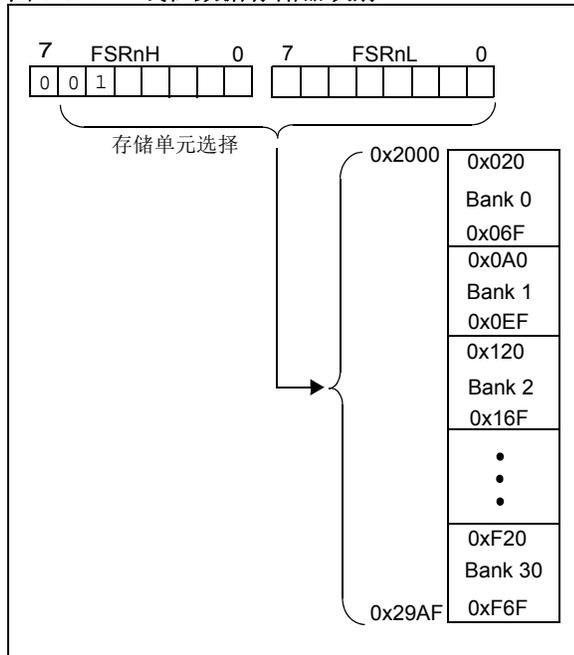
4.5.2 线性数据存储器

线性数据存储器是从FSR地址0x2000至FSR地址0x29AF的区域。该区域是一个虚拟区域，它指向所有存储区中80字节的GPR存储块。

未实现的存储器将读为0x00。通过使用线性数据存储器区域，可以支持大于80字节的缓冲区，因为在FSR递增至超过一个存储区时，将会直接转至下一个存储区的GPR存储器。

16字节的公共存储器不包含在线性数据存储器区域中。

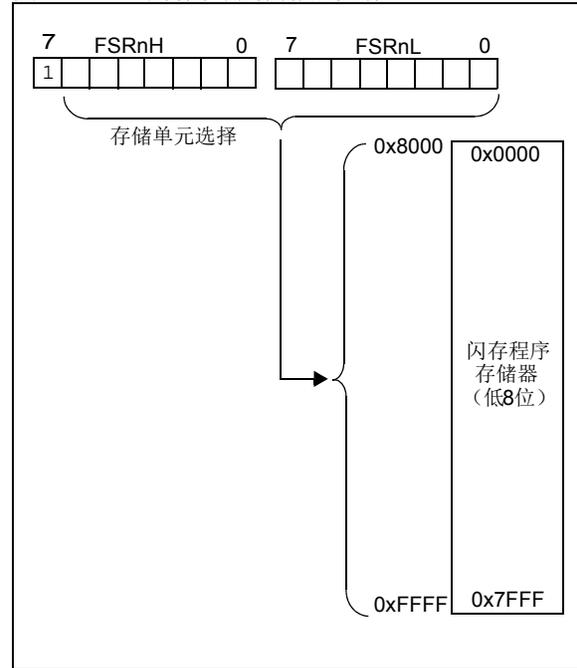
图4-10: 线性数据存储器映射



4.5.3 闪存程序存储器

为了方便地访问常量数据，整个闪存程序存储器都映射到FSR地址空间的上半部分。当FSRnH的MSB置1时，低15位为程序存储器中将通过INDF访问的地址。对于每个存储单元，只有低8位可通过INDF访问。对闪存程序存储器的写操作无法通过FSR/INDF接口实现。对于通过FSR/INDF接口访问闪存程序存储器的所有指令，都需要一个额外的指令周期才能完成操作。

图4-11: 闪存程序存储器映射



4.5.4 数据EEPROM存储器

可通过NVMCON寄存器接口读取或写入EEPROM存储器（见第11.2节“数据EEPROM”）。但是，要想更轻松访问EEPROM，也可使用间接寻址通过FSR对EEPROM的内容进行只读访问。当FSR的MSB（例如：FSRxH）设置为0x70时，低8位地址值（在FSRxL中）用于确定可（通过INDF寄存器）读取的EEPROM存储单元。

换句话说，EEPROM地址范围0x00-0xFF映射到0x7000-0x70FF之间的FSR地址空间。对EEPROM的写操作无法通过FSR/INDF接口实现。通过FSR/INDF接口读取EEPROM时，将需要一个额外的指令周期才能完成操作。

5.0 器件配置

器件配置功能由配置字、代码保护和器件ID组成。

5.1 配置字

有几个配置字位可用于选择不同的振荡器和存储器保护选项。这些位实现为位于8007h的配置字1，位于8008h的配置字2、位于8009h的配置字3和位于800Ah的配置字4。

注： 配置字中的 $\overline{\text{DEBUG}}$ 位由器件开发工具（包括调试器和编程器）自动管理。对于正常器件工作，该位应保持为1。

PIC16(L)F18324/18344

5.2 寄存器定义：配置字

寄存器5-1： 配置字1： 振荡器

R/P-1	U-1	R/P-1	U-1	U-1	R/P-1
FCMEN	—	CSWEN	—	—	CLKOUTEN
bit 13					bit 8

U-1	R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1
—	RSTOSC2	RSTOSC1	RSTOSC0	—	FEXTOSC2	FEXTOSC1	FEXTOSC0
bit 7							bit 0

图注：

R = 可读位 P = 可编程位 U = 未实现位，读为1
 0 = 清零 1 = 置1 n = 空白时的值

- bit 13 **FCMEN**: 故障保护时钟监视器使能位
 1 = ON 使能FSCM定时器
 0 = OFF 禁止FSCM定时器
- bit 12 **未实现**: 读为1
- bit 11 **CSWEN**: 时钟切换使能位
 1 = ON 允许写入NOSC和NDIV
 0 = OFF 无法通过用户软件更改NOSC和NDIV位
- bit 10-9 **未实现**: 读为1
- bit 8 **CLKOUTEN**: 时钟输出使能位
如果FEXTOSC = EC、HS、HT或LP，该位被忽略；否则：
 1 = OFF 禁止CLKOUT功能；OSC2为I/O或振荡器功能
 0 = ON 使能CLKOUT功能；OSC2为Fosc/4时钟
- bit 7 **未实现**: 读为1
- bit 6-4 **RSTOSC<2:0>**: COSC的上电默认值位
 该值是COSC的复位默认值，并选择用户软件首次使用的振荡器
 111 = EXT1X EXTOSC根据FEXTOSC<2:0>位操作
 110 = HFINT1 HFINTOSC (1 MHz)
 101 = 保留
 100 = LFINT LFINTOSC
 011 = SOSC SOSC (32.768 kHz)
 010 = 保留
 001 = EXT4X 采用4x PLL的EXTOSC；EXTOSC根据FEXTOSC<2:0>位操作
 000 = HFINT32 HFINTOSC (32 MHz)
- bit 3 **未实现**: 读为1
- bit 2-0 **FEXTOSC<2:0>**: FEXTOSC外部振荡器模式选择位
 111 = ECH EC (外部时钟)，高于8 MHz
 110 = ECM EC (外部时钟)，100 kHz至8 MHz
 101 = ECL EC (外部时钟)，低于100 kHz
 100 = OFF 未使能振荡器
 011 = 未实现
 010 = HS HS (晶振)，高于4 MHz
 001 = XT HT (晶振)，100 kHz至4 MHz
 000 = LP LP (晶振)，优化为32.768 kHz

寄存器 5-2: 配置字 2: 监控位

R/P-1	R/P-1	R/P-1	U-1	R/P-1	U-1
DEBUG	STVREN	PPS1WAY	—	BORV	—
bit 13			bit 8		

R/P-1	R/P-1	R/P-1	U-1	R/P-1	R/P-1	R/P-1	R/P-1
BOREN1	BOREN0	LPBOREN ⁽³⁾	—	WDTE1	WDTE0	PWRTE	MCLRE
bit 7							bit 0

图注:

R = 可读位
0 = 清零

P = 可编程位
1 = 置1

U = 未实现位, 读为1
n = 空白时的值

- bit 13 **DEBUG:** 调试器使能位⁽¹⁾
1 = OFF 禁止后台调试器; ICSPCLK 和 ICSPDAT 是通用 I/O 引脚
0 = ON 使能后台调试器; ICSPCLK 和 ICSPDAT 专用于调试器
- bit 12 **STVREN:** 堆栈上溢/下溢复位使能位
1 = ON 堆栈上溢或下溢将导致复位
0 = OFF 堆栈上溢或下溢不会导致复位
- bit 11 **PPS1WAY:** PPSLOCK 一次置1使能位
1 = ON PPSLOCK 位只可清零和置1一次; 一个清零/置1周期后 PPS 寄存器将保持锁定
0 = OFF PPSLOCK 位可重复清零和置1 (需要解锁序列)
- bit 10 **未实现:** 读为1
- bit 9 **BORV:** 欠压复位电压选择位⁽²⁾
1 = LOW 在 LF 器件上欠压复位电压 (VBOR) 设置为 1.9V, 而在 F 器件上设置为 2.45V
0 = HIGH 欠压复位电压 (VBOR) 设置为 2.7V
对于工作在 16 MHz 或高于 16 MHz 的情况, 建议使用更高电压设置。
- bit 8 **未实现:** 读为1
- bit 7-6 **BOREN<1:0>:** 欠压复位使能位
使能时, 欠压复位电压 (VBOR) 通过 BORV 位设置
11 = ON 使能欠压复位; 忽略 SBOREN 位
10 = SLEEP 欠压复位在运行时使能, 在休眠时禁止; 忽略 SBOREN 位
01 = SBOREN 根据 SBOREN 使能欠压复位
00 = OFF 禁止欠压复位
- bit 5 **LPBOREN:** 低功耗 BOR 使能位⁽³⁾
1 = OFF 禁止 ULPBOR
0 = ON 使能 ULPBOR
- bit 4 **未实现:** 读为1
- bit 3-2 **WDTE<1:0>:** 看门狗定时器使能位
11 = ON 使能 WDT; 忽略 SWDTEN
10 = SLEEP WDT 在运行时使能, 在休眠/空闲时禁止; 忽略 SWDTEN
01 = SWDTEN WDT 由 WDTCON 寄存器的 SWDTEN 位控制
00 = OFF 禁止 WDT; 忽略 SWDTEN
- bit 1 **PWRTE:** 上电延时定时器使能位
1 = OFF 禁止 PWRT
0 = ON 使能 PWRT
- bit 0 **MCLRE:** 主复位 (MCLR) 使能位
如果 LVP = 1:
RA3 引脚功能为 MCLR。
如果 LVP = 0:
1 = ON MCLR 引脚功能为 MCLR。
0 = OFF MCLR 引脚功能为端口定义功能。

- 注 1: 配置字寄存器中的 DEBUG 位由器件开发工具 (包括调试器和编程器) 自动管理。对于正常器件工作, 该位应保持为 1。
2: 关于具体跳变点电压, 请参见 VBOR 参数。
3: 仅限 PIC16LF18324/18344 器件。

PIC16(L)F18324/18344

寄存器 5-3: 配置字 3: 存储器

R/P-1	U-1	U-1	U-1	U-1	U-1
LVP ⁽¹⁾	—	—	—	—	—
bit 13					bit 8

U-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
—	—	—	—	—	—	WRT1	WRT0
bit 7						bit 0	

图注:

R = 可读位

P = 可编程位

U = 未实现位, 读为 1

0 = 清零

1 = 置 1

n = 空白时或批量擦除后的值

bit 13

LVP: 低电压编程使能位⁽¹⁾

1 = ON 使能低电压编程。 $\overline{\text{MCLR/VPP}}$ 引脚功能是 $\overline{\text{MCLR}}$ 。MCLRRE 配置位被忽略。

0 = OFF $\overline{\text{MCLR/VPP}}$ 上的 HV 必须用于编程。

bit 12-2

未实现: 读为 1

bit 1-0

WRT<1:0>: 用户 NVM 自写保护位

11 = OFF 关闭写保护

10 = BOOT 0000h 至 01FFh 被写保护, 0200h 至 1FFFh 可被修改

01 = HALF 0000h 至 0FFFh 被写保护, 1000h 至 1FFFh 可被修改

00 = ALL 0000h 至 1FFFh 被写保护, 没有地址可被修改

WRT 只适用于器件的自写功能; 通过 ICSP 进行写操作不受写保护。

注 1: 当通过 LVP 进入编程模式时, 不能将 LVP 位编程为 0。

PIC16(L)F18324/18344

寄存器 5-4: 配置字 4: 代码保护

U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—
bit 13					bit 8

U-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
—	—	—	—	—	—	$\overline{\text{CPD}}$	$\overline{\text{CP}}$
bit 7						bit 0	

图注:

R = 可读位
0 = 清零

P = 可编程位
1 = 置 1

U = 未实现位, 读为 1
n = 空白时或批量擦除后的值

bit 13-2 **未实现:** 读为 1

bit 1 **$\overline{\text{CPD}}$:** 数据 EEPROM 存储器代码保护位
1 = OFF 禁止数据 EEPROM 代码保护
0 = ON 使能数据 EEPROM 代码保护

bit 0 **$\overline{\text{CP}}$:** 程序存储器代码保护位
1 = OFF 禁止程序存储器代码保护
0 = ON 使能程序存储器代码保护

PIC16(L)F18324/18344

5.3 代码保护

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

5.3.1 程序存储器保护

整个程序存储空间都通过配置字中的 $\overline{\text{CP}}$ 位来防止外部读写操作。当 $\overline{\text{CP}} = 0$ 时，将禁止对程序存储器的外部读写操作，读取时将返回全0。无论保护位的设置如何，CPU都可以继续读取程序存储器。对程序存储器的自写操作则取决于写保护设置。更多信息，请参见第5.4节“写保护”。

5.3.2 数据存储器保护

整个数据EEPROM都通过配置字中的 $\overline{\text{CPD}}$ 位来防止外部读写操作。当 $\overline{\text{CPD}} = 0$ 时，将禁止对EEPROM存储器的外部读写操作，读取时将返回全0。无论保护位的设置如何，CPU都可以继续读写EEPROM存储器。

5.4 写保护

通过写保护，可以防止器件发生意外的自写操作。在保护应用程序（如自举程序软件）的同时，可以允许对程序存储器的其他区域进行修改。

配置字中的WRT<1:0>位定义受保护的程序存储块的大小。

5.5 用户ID

有4个存储单元（8000h-8003h）被指定为ID存储单元，供用户存储校验和其他代码标识号。在正常执行期间，这些存储单元是可读写的。关于访问这些存储单元的更多信息，请参见第11.4.7节“NVMREG EEPROM、用户ID、器件ID和配置字访问”。关于校验和计算的更多信息，请参见“PIC16(L)F183XX Memory Programming Specification”（DS40001738）。

5.6 器件ID和版本ID

14位器件ID字位于8006h，14位版本ID位于8005h。这些存储单元是只读的，不能擦除或修改。关于访问这些存储单元的更多信息，请参见第11.4.7节“NVMREG EEPROM、用户ID、器件ID和配置字访问”。

开发工具（如器件编程器和调试器）可用于读取器件ID和版本ID。

5.7 寄存器定义：器件和版本

寄存器 5-5: **DEVID: 器件ID寄存器**

R	R	R	R	R	R	R	R
DEV<13:8>							
bit 13				bit 8			

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7				bit 0			

图注:

R = 可读位

1 = 置1

0 = 清零

bit 13-0 **DEV<13:0>**: 器件ID位

器件	DEV<13:0>值
PIC16F18324	11 0000 0011 1010 (303Ah)
PIC16LF18324	11 0000 0011 1100 (303Ch)
PIC16F18344	11 0000 0011 1011 (303Bh)
PIC16LF18344	11 0000 0011 1101 (303Dh)

寄存器 5-6: **REVID: 版本ID寄存器**

R-1	R-0	R	R	R	R	R	R
REV<13:8>							
bit 13				bit 8			

R	R	R	R	R	R	R	R
REV<7:0>							
bit 7				bit 0			

图注:

R = 可读位

1 = 置1

0 = 清零

bit 13-0 **REV<13:0>**: 版本ID位

注: 版本ID寄存器的最高两位将始终读为10。

PIC16(L)F18324/18344

6.0 复位

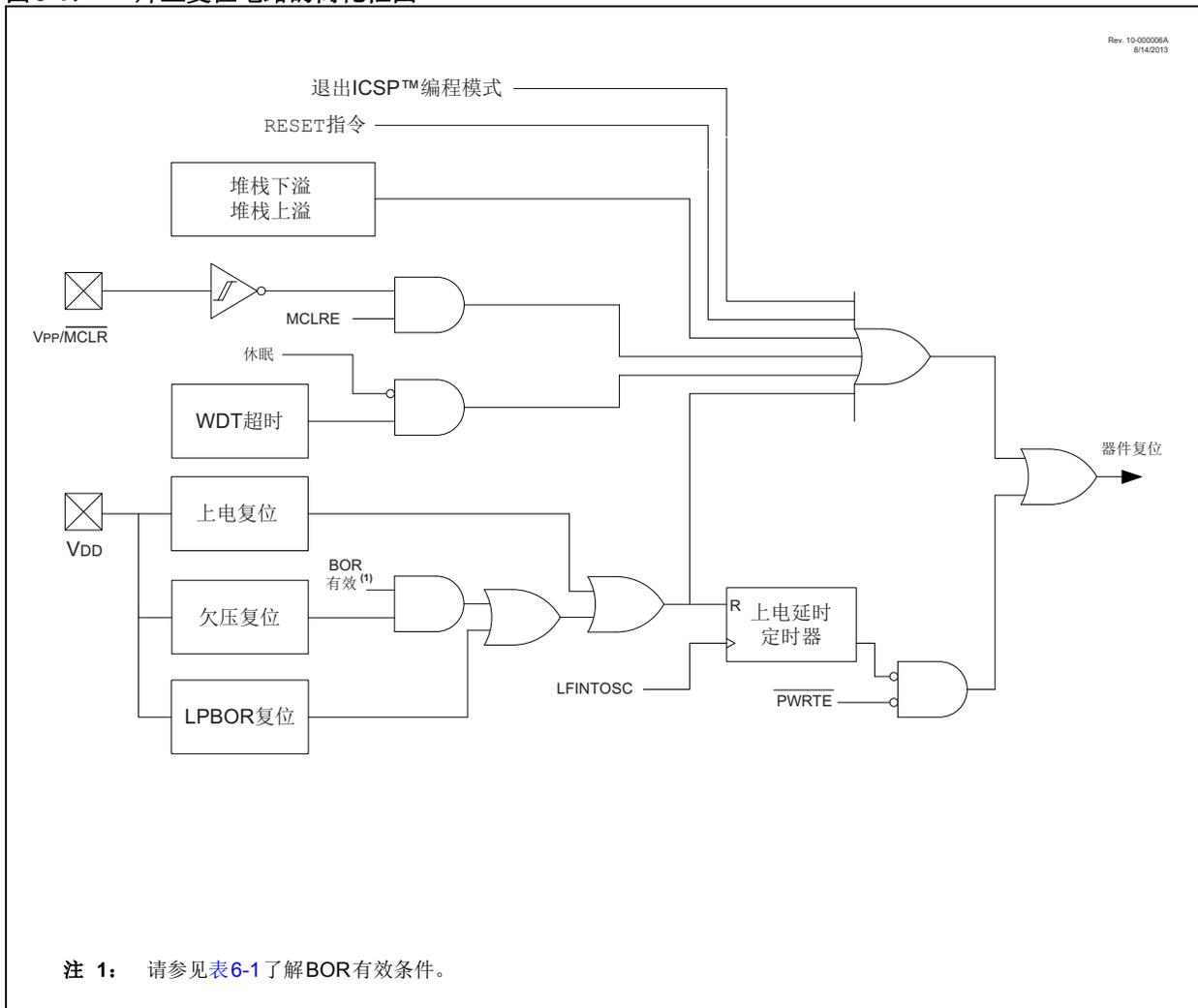
该器件有多种复位方式：

- 上电复位 (POR)
- 欠压复位 (BOR)
- 低功耗欠压复位 (Low-Power Brown-Out Reset, LPBOR)
- MCLR复位
- WDT复位
- RESET指令
- 堆栈上溢
- 堆栈下溢
- 编程模式退出

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

图6-1给出了片上复位电路的简化框图。

图6-1： 片上复位电路的简化框图



6.1 上电复位 (POR)

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

6.2 欠压复位 (BOR)

当VDD低于可选的最低电压时，BOR电路会将器件保持在复位状态。在POR和BOR之间，可在整个电压范围内对器件的执行进行保护。

欠压复位模块具有4种工作模式，它们由配置字中的BOREN<1:0>位控制。这4种工作模式是：

- BOR总是使能
- BOR在休眠模式下关闭
- BOR通过软件进行控制
- BOR总是禁止

更多信息，请参见表6-1。

对配置字中的BORV位进行配置来选择欠压复位电压。

VDD噪声抑制滤波器可以防止BOR在发生小事件时产生触发。如果VDD降至VBOR以下且持续时间超过参数TBORDC，则器件将复位，并且PCON0寄存器的BOR位将清零，指示发生了欠压复位。更多信息，请参见图6-2。

表6-1: BOR工作模式

BOREN<1:0>	SBOREN	器件模式	BOR 模式	在以下情况下的指令执行： POR释放或从休眠模式唤醒
11	X	X	有效	在“POR释放”和“从休眠模式唤醒”的特定情况下，启动时没有任何延时。在CPU准备好执行指令之前，BOR就绪标志会置1 (BORRDY = 1)，这是因为BOR电路通过BOREN<1:0>位被强制开启。
10	X	唤醒	有效	等待BOR释放 (BORRDY = 1)
		休眠	禁止	唤醒时忽略BOR
01	1	X	有效	在“POR释放”和“从休眠模式唤醒”的特定情况下，启动时没有任何延时。在CPU准备好执行指令之前，BOR就绪标志会置1 (BORRDY = 1)，这是因为BOR电路通过BOREN<1:0>位被强制开启。
	0	X	禁止	立即开始 (BORRDY = x)
00	X	X	禁止	

PIC16(L)F18324/18344

6.2.1 BOR总是使能

当配置字的BOREN位编程为11时，BOR将总是使能。器件启动会被延迟，直到BOR就绪，且VDD高于BOR阈值为止。

BOR保护在休眠期间有效。BOR不会延迟从休眠中唤醒。

6.2.2 BOR在休眠模式下关闭

当配置字的BOREN位编程为10时，除非处于休眠模式，否则BOR将使能。器件启动会被延迟，直到BOR就绪，且VDD高于BOR阈值为止。

BOR保护在休眠期间无效，但器件唤醒会被延迟，直到BOR可确定VDD高于BOR阈值。器件唤醒会被延迟，直到BOR就绪为止。

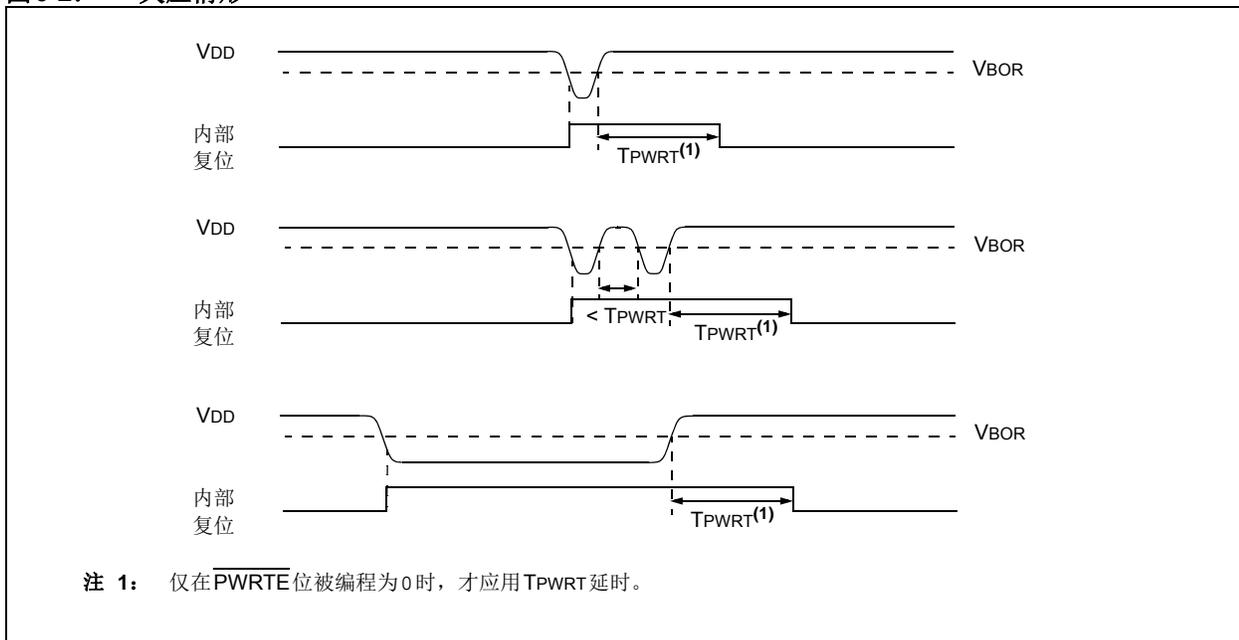
6.2.3 通过软件对BOR进行控制

当配置字的BOREN位编程为01时，BOR将通过BORCON寄存器的SBOREN位进行控制。仅当SBOREN位由软件清零且器件从非POR/BOR复位事件启动时，器件从休眠模式唤醒不会受BOR就绪条件或VDD电压影响而延迟。

BOR保护会在BOR电路就绪时立即开始。BOR电路的状态在BORCON寄存器的BORRDY位中反映。

BOR保护在休眠期间不变。

图6-2: 欠压情形



6.2.4 BOR总是关闭

当配置字2的BOREN位编程为00时，BOR将总是禁止。在该配置中，将SWBOREN位置1对BOR操作没有影响。

6.3 低功耗欠压复位 (LPBOR) (仅限 PIC16LF18324/18344 器件)

低功耗欠压复位 (LPBOR) 电路为 PIC16LF18324/18344 器件提供备用的欠压保护。当 V_{DD} 降至 LPBOR 阈值以下时，器件保持复位状态。发生这种情况时，PCON0 寄存器的 BOR 位将清零，指示发生了欠压复位。当 BOR 或 LPBOR 电路检测到 BOR 条件时，BOR 位将清零。无论是否使能了 BOR，均可使用 LPBOR 功能。

如果在使能了 BOR 时使用 LPBOR，则 LPBOR 可在 BOR 电路未成功检测到 BOR 条件时用作辅助保护电路。此外，如果在除休眠模式 (BOREN<1:0> = 10) 以外的情况下使能 BOR，则当 V_{DD} 低于 LPBOR 阈值时，LPBOR 电路将使器件保持复位状态，并且还将重新激活 POR。(关于 LPBOR 复位电压，请参见表 35-11)。

如果在未使能 BOR 时使用 LPBOR，则 LPBOR 电路将提供一个复位跳变点，这有助于降低电流消耗。

6.3.1 使能 LPBOR

LPBOR 由配置字的 $\overline{\text{LPBOR}}$ 位控制。在器件被擦除后，LPBOR 模块默认设置为禁止。

6.3.1.1 LPBOR 模块输出

LPBOR 模块的输出是一个用于指示是否要将复位置为有效的信号。该信号与 BOR 模块的复位信号进行逻辑或，用以提供通用 BOR 信号，并送至 PCON0 寄存器和电源控制模块。

6.4 $\overline{\text{MCLR}}$

$\overline{\text{MCLR}}$ 是可将器件复位的可选外部输入。 $\overline{\text{MCLR}}$ 功能由配置字的 MCLRE 位和 LVP 位控制 (表 6-2)。

表 6-2: $\overline{\text{MCLR}}$ 配置

MCLRE	LVP	$\overline{\text{MCLR}}$
0	0	禁止
1	0	使能
x	1	使能

6.4.1 $\overline{\text{MCLR}}$ 使能

当使能 $\overline{\text{MCLR}}$ 并且引脚保持低电平时，器件将保持在复位状态。 $\overline{\text{MCLR}}$ 引脚通过内部弱上拉连接到 V_{DD}。

器件在 $\overline{\text{MCLR}}$ 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。

注： 复位不会将 $\overline{\text{MCLR}}$ 引脚驱动为低电平。

6.4.2 $\overline{\text{MCLR}}$ 禁止

当 $\overline{\text{MCLR}}$ 被禁止时，引脚将用作通用输入，内部弱上拉由软件控制。更多信息，请参见第 12.2 节“PORTA 寄存器”。

6.5 看门狗定时器 (WDT) 复位

如果固件未在超时周期内发出 CLRWDT 指令，看门狗定时器会产生复位。 $\overline{\text{STATUS}}$ 寄存器中的 TO 和 PD 位以及 PCON0 寄存器中的 RWDT 位会发生改变，指示 WDT 复位。更多信息，请参见第 10.0 节“看门狗定时器 (WDT)”。

6.6 RESET 指令

RESET 指令会引起器件复位。PCON0 寄存器中的 $\overline{\text{RI}}$ 位将设置为 0。关于发生 RESET 指令之后的默认条件，请参见表 6-4。

6.7 堆栈上溢/下溢复位

器件可以在堆栈上溢或下溢时复位。PCON 寄存器的 STKOVF 或 STKUNF 位用于指示复位条件。这两种复位通过将配置字中的 STVREN 位置 1 来使能。更多信息，请参见第 4.4 节“堆栈”。

6.8 编程模式退出

在退出编程模式时，器件的行为与刚刚发生器件复位时的情况相同。

6.9 上电延时定时器

上电延时定时器在 POR 或欠压复位时提供一个 64 ms 标称值的延时。

只要 PWRT 处于活动状态，器件就保持在复位状态。PWRT 延时使 V_{DD} 有额外的时间上升到所需的电压。可通过清零配置字中的 PWRT 位使能上电延时定时器。

上电延时定时器会在 POR 和 BOR 释放之后启动。

更多信息，请参见应用笔记 AN607，“Power-up Trouble Shooting” (DS00607)。

PIC16(L)F18324/18344

6.10 启动序列

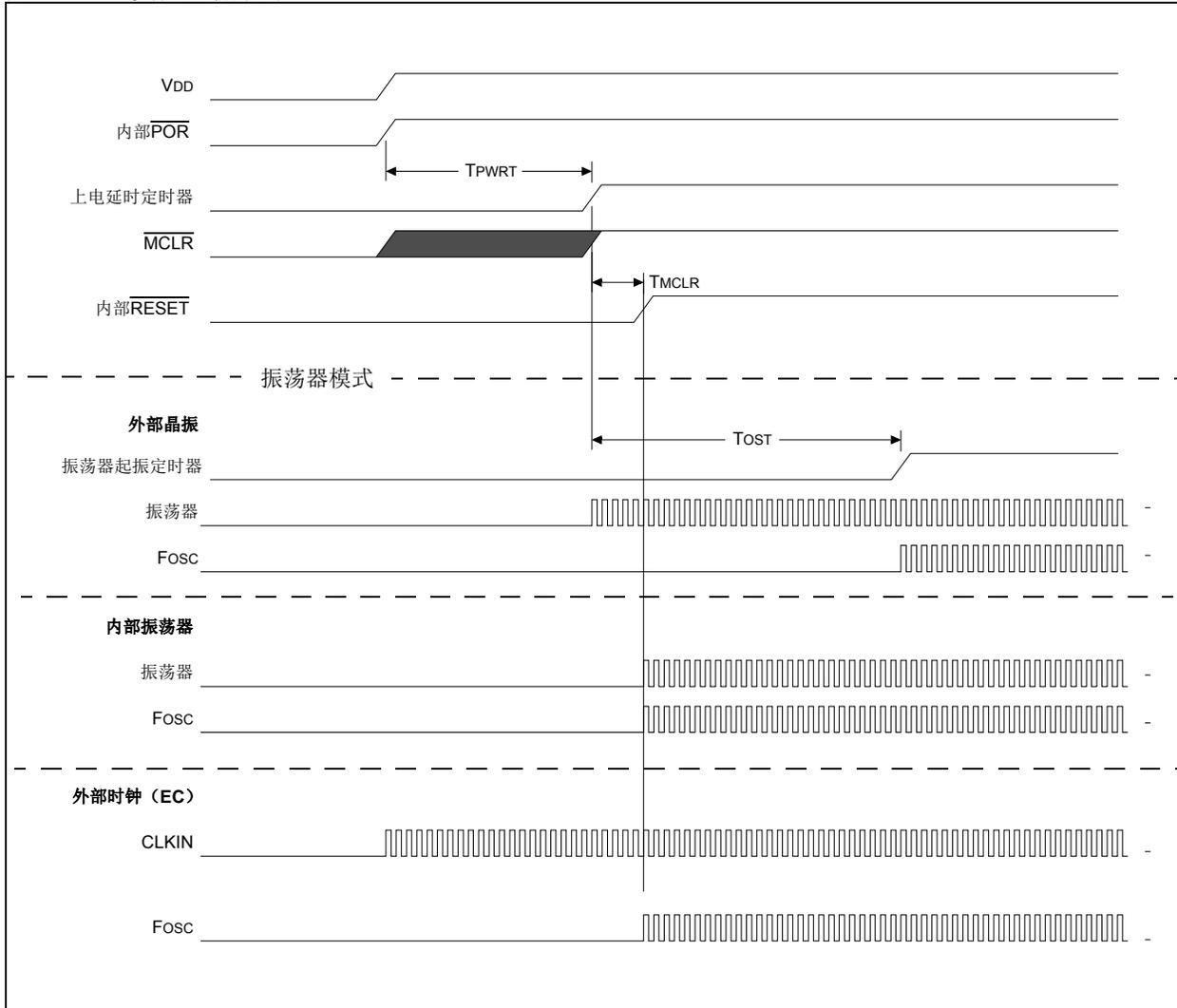
在POR或BOR释放时，只有先发生以下事件，器件才会开始执行：

1. 上电延时定时器运行完毕（如果使能）。
2. $\overline{\text{MCLR}}$ 必须被释放（如果使能）。
3. 振荡器起振定时器运行完毕（如果对于振荡器源需要）。

总延时取决于振荡器配置和上电延时定时器配置。更多信息，请参见第7.0节“振荡器模块”。

上电延时定时器和振荡器起振定时器的运行与 $\overline{\text{MCLR}}$ 复位无关。如果 $\overline{\text{MCLR}}$ 保持低电平的时间足够长，上电延时定时器将会延时结束。当 $\overline{\text{MCLR}}$ 变为高电平时，器件将在10个 F_{osc} 周期之后开始执行代码（见图6-3）。这对于测试或同步多个并行工作的器件来说是非常有用的。

图6-3: 复位启动序列



6.11 确定复位原因

在发生任何复位时，STATUS和PCON0寄存器中会有多个位发生更新，以指示复位的原因。表6-3和表6-4列出了这些寄存器的复位条件。

表6-3: 复位状态位及其含义

STKOVF	STKUNF	RWDT	RMCLR	RI	POR	BOR	TO	PD	条件
0	0	1	1	1	0	x	1	1	上电复位
0	0	1	1	1	0	x	0	x	非法的， $\overline{\text{POR}}$ 时 $\overline{\text{TO}}$ 被置1
0	0	1	1	1	0	x	x	0	非法的， $\overline{\text{POR}}$ 时 $\overline{\text{PD}}$ 被置1
0	0	u	1	1	u	0	1	1	欠压复位
u	u	0	u	u	u	u	0	u	WDT复位
u	u	u	u	u	u	u	0	0	被WDT从休眠模式唤醒
u	u	u	u	u	u	u	1	0	被中断从休眠模式唤醒
u	u	u	0	u	u	u	u	u	正常工作期间的 $\overline{\text{MCLR}}$ 复位
u	u	u	0	u	u	u	1	0	休眠期间的 $\overline{\text{MCLR}}$ 复位
u	u	u	u	0	u	u	u	u	执行了RESET指令
1	u	u	u	u	u	u	u	u	堆栈上溢复位 (STVREN = 1)
u	1	u	u	u	u	u	u	u	堆栈下溢复位 (STVREN = 1)

表6-4: 特殊寄存器的复位条件

条件	程序计数器	STATUS寄存器	PCON0寄存器
上电复位	0000h	---1 1000	00-- 110x
正常工作期间的 $\overline{\text{MCLR}}$ 复位	0000h	---u uuuu	uu-- 0uuu
休眠期间的 $\overline{\text{MCLR}}$ 复位	0000h	---1 0uuu	uu-- 0uuu
WDT复位	0000h	---0 uuuu	uu-0 uuuu
被WDT从休眠模式唤醒	PC + 1	---0 0uuu	uu-u uuuu
欠压复位	0000h	---1 1000	00-1 11u0
被中断从休眠模式唤醒	PC + 1 ⁽¹⁾	---1 0uuu	uu-u uuuu
执行了RESET指令	0000h	---u uuuu	uu-u u0uu
堆栈上溢复位 (STVREN = 1)	0000h	---u uuuu	1u-u uuuu
堆栈下溢复位 (STVREN = 1)	0000h	---u uuuu	u1-u uuuu

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) 被置1时, 返回地址被压入堆栈, 并且在执行PC + 1后, 将中断向量 (0004h) 装入PC。

PIC16(L)F18324/18344

寄存器 6-1: **BORCON: 欠压复位控制寄存器**

R/W-1/u	R/W-0-0	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN ⁽¹⁾	保留	—	—	—	—	—	BORRDY
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7

SBOREN: 软件欠压复位使能位⁽¹⁾

如果配置字中的BOREN <1:0> ≠ 01:

SBOREN可读/写, 但对BOR没有任何作用。

如果配置字中的BOREN <1:0> = 01:

1 = 使能BOR

0 = 禁止BOR

bit 6

保留。该位必须保持为0

bit 5-1

未实现: 读为0

bit 0

BORRDY: 欠压复位电路就绪状态位

1 = 欠压复位电路有效

0 = 欠压复位电路无效

注 1: BOREN<1:0> 位于配置字中。

6.12 电源控制 (PCON0) 寄存器

电源控制 (PCON0) 寄存器包含区分以下各种复位的标志位:

- 上电复位 ($\overline{\text{POR}}$)
- 欠压复位 ($\overline{\text{BOR}}$)
- RESET 指令复位 ($\overline{\text{RI}}$)
- MCLR 复位 ($\overline{\text{RMCLR}}$)
- 看门狗定时器复位 ($\overline{\text{RWD\overline{T}}}$)
- 堆栈下溢复位 (STKUNF)
- 堆栈上溢复位 (STKOVF)

PCON0 寄存器中的各位如寄存器 6-2 所示。

硬件将在复位过程中改变相应的寄存器位; 如果复位不是由相应条件引起的, 对应位保持不变 (表 6-4)。

在重启后, 软件应将相应位复位为无效状态 (硬件不会复位相应位)。

软件还可将 PCON0 寄存器中的任何位设置为有效状态, 这样可测试用户代码, 但不会产生任何复位操作。

6.13 寄存器定义：电源控制

寄存器 6-2: **PCON0: 电源控制寄存器 0**

R/W/HS-0/q	R/W/HS-0/q	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	\overline{RWDT}	\overline{RMCLR}	\overline{RI}	\overline{POR}	BOR
bit 7							bit 0

图注:

HC = 硬件清零位	HS = 硬件置 1 位
R = 可读位	W = 可写位
u = 不变	U = 未实现位, 读为 0
1 = 置 1	x = 未知
	-m/n = POR 和 BOR 时的值/所有其他复位时的值
	q = 值取决于具体条件

bit 7	STKOVF: 堆栈上溢标志位 1 = 发生了堆栈上溢 0 = 未发生堆栈上溢或已由固件清零
bit 6	STKUNF: 堆栈下溢标志位 1 = 发生了堆栈下溢 0 = 未发生堆栈下溢或已由固件清零
bit 5	未实现: 读为 0
bit 4	\overline{RWDT}: 看门狗定时器复位标志位 1 = 未发生看门狗定时器复位或由固件置 1 0 = 发生了看门狗定时器复位 (由硬件清零)
bit 3	\overline{RMCLR}: \overline{MCLR} 复位标志位 1 = 未发生 \overline{MCLR} 复位或由固件置 1 0 = 发生了 \overline{MCLR} 复位 (由硬件清零)
bit 2	\overline{RI}: RESET 指令标志位 1 = 未执行 RESET 指令或由固件置 1 0 = 执行了 RESET 指令 (由硬件清零)
bit 1	\overline{POR}: 上电复位状态位 1 = 未发生上电复位 0 = 发生了上电复位 (发生上电复位后必须用软件置 1)
bit 0	BOR: 欠压复位状态位 1 = 未发生欠压复位 0 = 发生了欠压复位 (发生上电复位或欠压复位后必须用软件置 1)

表 6-5: 与复位相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	74
PCON0	STKOVF	STKUNF	—	\overline{RWDT}	\overline{RMCLR}	\overline{RI}	\overline{POR}	BOR	75
STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	28
WDTCON	—	—	WDTPS<4:0>					SWDTEN	119

图注: — = 未实现位, 读为 0。复位不使用阴影单元。

PIC16(L)F18324/18344

7.0 振荡器模块

7.1 概述

振荡器模块具有多种时钟源和选择特性，从而使其应用非常广泛，同时最大程度地发挥性能并降低功耗。[图7-1](#)给出了振荡器模块的框图。

时钟源可由外部振荡器、石英晶体谐振器和陶瓷谐振器提供。此外，系统时钟源可由两个内部振荡器之一和PLL电路提供，并通过软件来选择速度。其他时钟特性包括：

- 可通过软件选择外部或内部时钟源作为系统时钟源。
- 故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM），用来检测外部时钟源（LP、XT、HS、ECH、ECM和ECL）故障并自动切换到内部振荡器。
- 振荡器起振定时器（OST），可确保晶振源的稳定性。

配置字1的各RSTOSC位决定在器件复位（包括最初上电时）时使用的振荡器类型。

内部时钟模式LFINTOSC、HFINTOSC（设置为1 MHz）或HFINTOSC（设置为32 MHz）可通过各RSTOSC位设置。

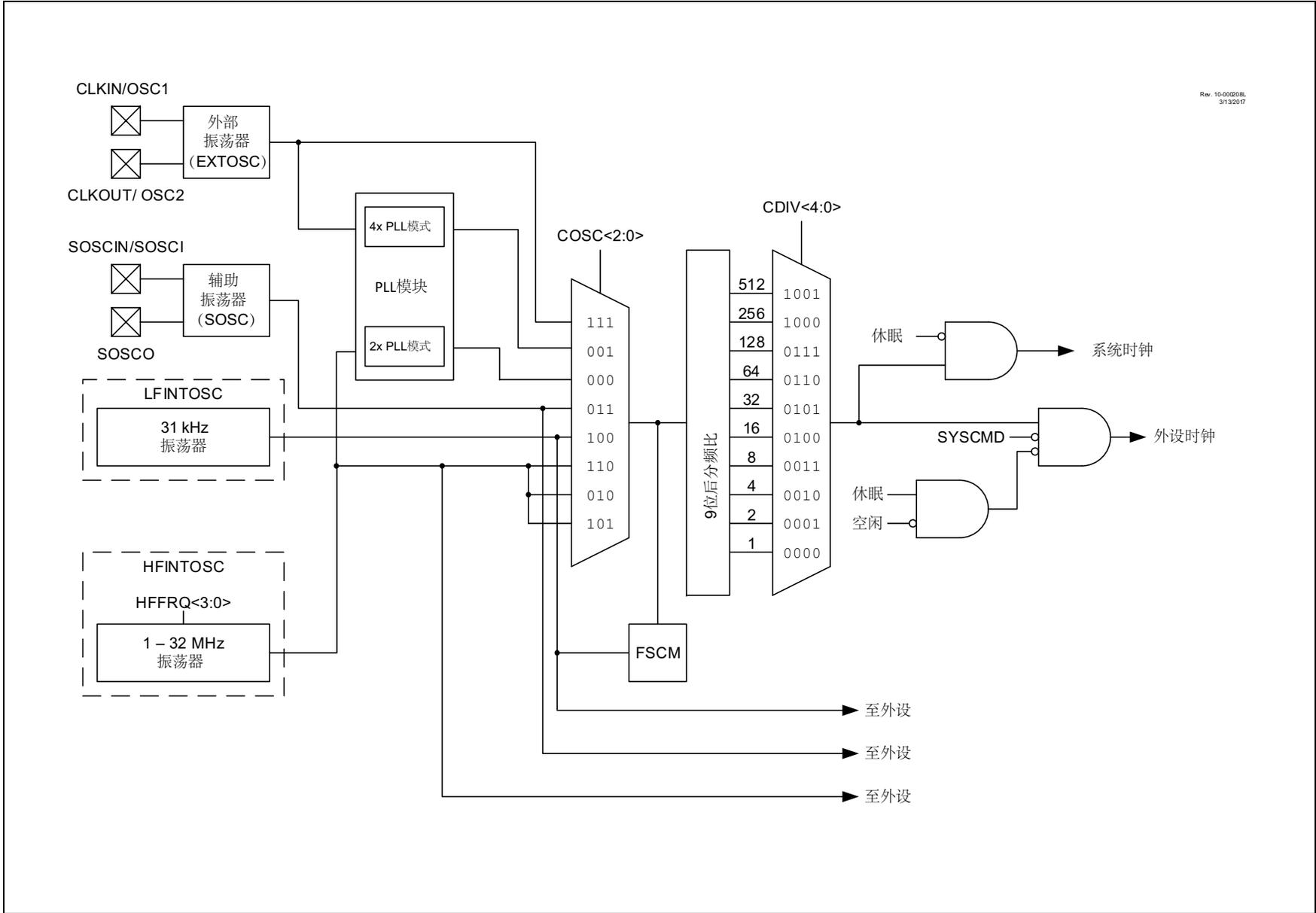
如果选择外部时钟源，必须结合使用配置字1的FEXTOSC位和RSTOSC位来选择外部时钟模式。

可通过设置配置字1的FEXTOSC<2:0>位将外部振荡器模块配置为以下时钟模式之一：

1. ECL——外部时钟低功耗模式（ ≤ 100 kHz）
2. ECM——外部时钟中等功耗模式（ ≤ 8 MHz）
3. ECH——外部时钟高功耗模式（ ≤ 32 MHz）
4. LP——32 kHz低功耗晶振模式
5. Xt——中等增益晶振或陶瓷谐振器振荡器模式（100 kHz至4 MHz）
6. HS——高增益晶振或陶瓷谐振器模式（高于4 MHz）

ECH、ECM和ECL时钟模式依靠外部逻辑电平信号作为器件时钟源。LP、XT和HS时钟模式要求器件在外部连接一个晶振或谐振器。每种模式都针对不同频率范围而优化。INTOSC内部振荡器模块可以产生低频和高频时钟源，分别用LFINTOSC和HFINTOSC表示。（见内部振荡器模块，[图7-1](#)）。

图7-1: PIC® MCU时钟源的简化框图



PIC16(L)F18324/18344

7.2 时钟源类型

时钟源可分为外部时钟源和内部时钟源。

外部时钟源依靠外部电路工作。例如：振荡器模块（ECH、ECM和ECL模式）、石英晶振或陶瓷谐振器（LP、XT和HS模式）。

还有一个辅助振荡器模块，针对32.768 kHz外部时钟源进行优化，可用作备用时钟源。

有两个内部振荡器模块：

- HFINTOSC
- LFINTOSC

HFINTOSC可产生1 MHz-16 MHz的时钟频率。LFINTOSC可产生31 kHz的时钟频率。

有一个PLL可供外部振荡器使用。更多详细信息，请参见第7.2.1.4节“4x PLL”。

此外，还有一个PLL可供HFINTOSC在某些特定频率下使用。更多详细信息，请参见第7.2.2.2节“2x PLL”。

7.2.1 外部时钟源

通过执行以下操作之一，可以使用外部时钟源作为器件系统时钟：

- 编程配置字中的RSTOSC<2:0>位，选择在器件复位时用作默认系统时钟的外部时钟源。
- 写入OSCCON1寄存器中的NOSC<2:0>和INDIV<3:0>位以切换系统时钟源。

更多信息，请参见第7.3节“时钟切换”。

7.2.1.1 EC模式

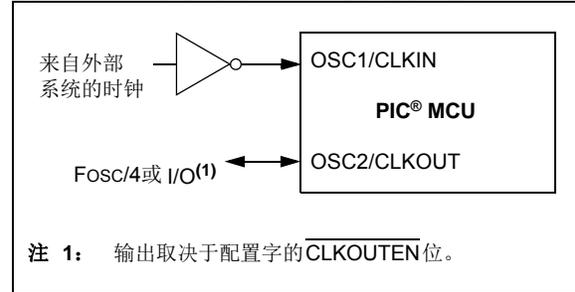
外部时钟（EC）模式允许外部产生的逻辑电平信号作为系统时钟源。工作在该模式下时，外部时钟源连接到CLKIN输入。OSC2/CLKOUT可用作通用I/O或CLKOUT。图7-2给出了EC模式的引脚连接图。

EC模式具有三种功耗模式，可通过配置字进行选择：

- ECH——高功耗，≤ 32 MHz
- ECM——中等功耗，≤ 8 MHz
- ECL——低功耗，≤ 0.1 MHz

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PIC® 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7-2: 外部时钟（EC）模式的工作原理



7.2.1.2 LP、XT和HS模式

LP、XT和HS模式支持使用连接到OSC1和OSC2的石英晶振或陶瓷谐振器（图7-3）。这三种模式选择内部反相放大器的低、中或高增益设置，以支持各种谐振器类型及速度。

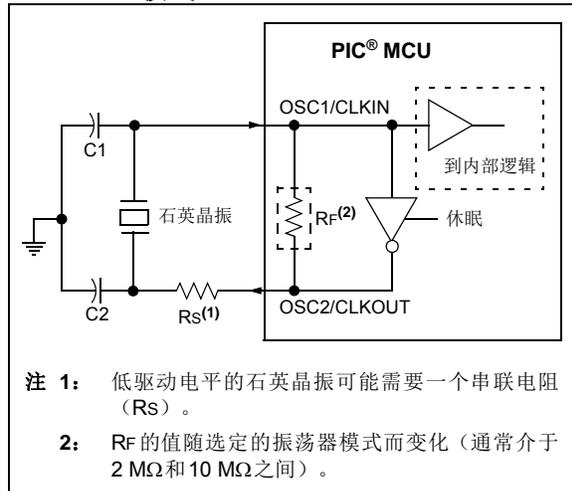
LP 振荡器模式选择内部反相放大器的最低增益设置。LP模式的电流消耗在三种模式中最小。该模式设计为用来驱动32.768 kHz的音叉（Tuning Fork）型晶振（时钟晶振），但工作频率最高可达100 kHz。

XT 振荡器模式选择内部反相放大器的中等增益设置。XT模式的电流消耗在三种模式中居中。该模式最适合驱动频率最高为4 MHz的晶振和谐振器。

HS 振荡器模式选择内部反相放大器的最高增益设置。HS模式的电流消耗在三种模式中最大。该模式最适合驱动需要工作频率最高为20 MHz的谐振器。

图7-3和图7-4分别显示了石英晶振和陶瓷谐振器的典型电路。

图7-3: 石英晶振的工作原理（LP、XT或HS模式）



注 1: 石英晶振的特性随类型、封装和制造商的不同而不同。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。

注 2: 应始终验证振荡器在应用要求的VDD和温度范围内的性能。

注 3: 如需振荡器设计帮助，请参见以下Microchip应用笔记：

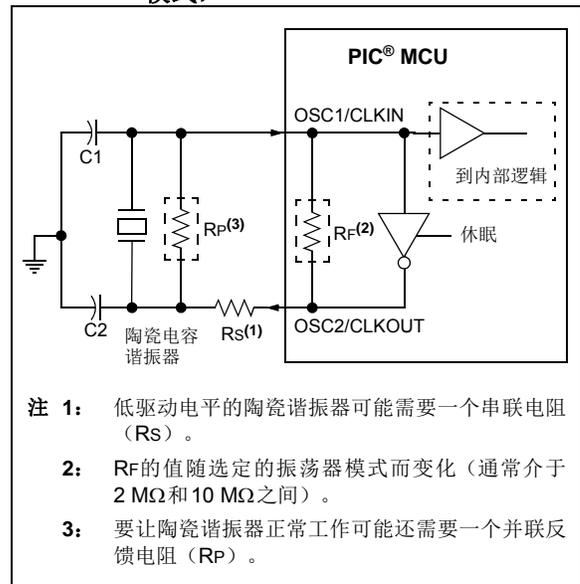
- AN826, “Crystal Oscillator Basics and Crystal Selection for *rPIC*[®] and *PIC*[®] Devices” (DS00826)

- AN849, “Basic *PIC*[®] Oscillator Design” (DS00849)

- AN943, “Practical *PIC*[®] Oscillator Analysis and Design” (DS00943)

- AN949, “Making Your Oscillator Work” (DS00949)

图7-4: 陶瓷谐振器的工作原理（XT或HS模式）



PIC16(L)F18324/18344

7.2.1.3 振荡器起振定时器（OST）

如果振荡器模块被配置为LP、XT或HS模式，则振荡器起振定时器（OST）对来自OSC1的振荡计数1024次。这发生在上电复位（POR）、欠压复位（BOR）或从休眠中唤醒后。OST确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经起振并为振荡器模块提供稳定的系统时钟。

7.2.1.4 4x PLL

振荡器模块包含了一个PLL，它可以与外部时钟源配合使用，用于提供系统时钟源。PLL的输入频率必须处于规范值范围内。请参见表35-9中的PLL时钟时序规范。

PLL可以通过两种方法之一使能：

1. 编程配置字1中的RSTOSC位以使能采用4x PLL的EXTOSC。
2. 写入OSCCON1寄存器中的NOSC<2:0>位以使能采用4x PLL的EXTOSC。

7.2.1.5 辅助振荡器

辅助振荡器是一个单独的振荡器模块，可用作备用系统时钟源。辅助振荡器针对32.768 kHz进行优化，可与连接到SOSCI和SOSCO器件引脚的外部晶振或连接到SOSCIN引脚的外部时钟源配合使用。辅助振荡器可在运行期间通过时钟切换来选择。更多信息，请参见第7.3节“时钟切换”。

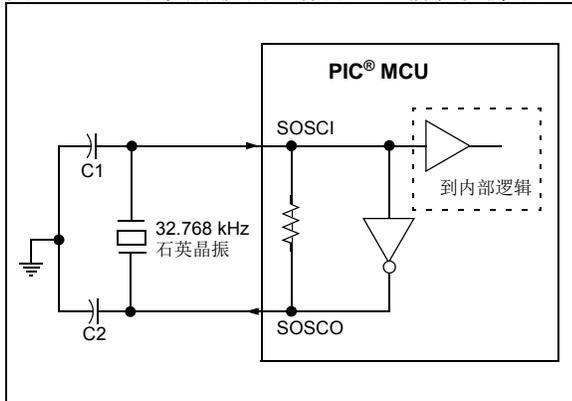
注 1: 石英晶振的特性随类型、封装和制造商的不同而不同。要了解规格说明和推荐应用，应查阅制造商提供的数据手册。

2: 应始终验证振荡器在应用要求的VDD和温度范围内的性能。

3: 如需振荡器设计帮助，请参见以下Microchip应用笔记：

- AN826, “Crystal Oscillator Basics and Crystal Selection for *rfPIC*[®] and *PIC*[®] Devices” (DS00826)
- AN849, “Basic *PICmicro*[®] Oscillator Design” (DS00849)
- AN943, “Practical *PICmicro*[®] Oscillator Analysis and Design” (DS00943)
- AN949, “Making Your Oscillator Work” (DS00949)
- TB097, “Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS” (DS91097)
- AN1288, “Design Practices for Low-Power External Oscillators” (DS01288)

图7-5: 石英晶振的工作原理（辅助振荡器）



7.2.2 内部时钟源

通过执行以下操作之一，可以将器件配置为使用内部振荡器模块作为系统时钟：

- 编程配置字中的 RSTOSC<2:0> 位，选择在器件复位时用作默认系统时钟的 INTOSC 时钟源。
- 在运行时写入 OSCCON1 寄存器中的 NOSC<2:0> 位，将系统时钟源切换为内部振荡器。更多信息，请参见第 7.3 节“时钟切换”。

OSC2/CLKOUT 引脚的功能由配置字中的 $\overline{\text{CLKOUTEN}}$ 位决定。

内部振荡器模块有两个独立的振荡器，可产生两个内部系统时钟源。

1. HFINTOSC（高频内部振荡器）出厂时已校准，工作频率最高为 32 MHz。
2. LFINTOSC（低频内部振荡器）出厂时已校准，工作频率为 31 kHz。

7.2.2.1 HFINTOSC

高频内部振荡器（HFINTOSC）是一个高精度数字控制内部时钟源，可产生最高 32 MHz 的稳定时钟。HFINTOSC 可通过以下方法之一来使能：

- 在器件上电或复位后，将配置字 1 中的 RSTOSC<2:0> 位编程为 110（1 MHz）或 000（32 MHz）以设置振荡器
- 在运行时，写入 OSCCON1 寄存器的 NOSC<2:0> 位

HFINTOSC 频率可通过 OSCFRQ 寄存器的 HFFRQ<2:0> 位的设置进行选择。

凭借 OSCCON1 寄存器的 NDIV<3:0> 位，可对所选时钟源的输出进行分频，分频比的范围为 1:1 至 1:512。

7.2.2.2 2x PLL

振荡器模块包含了一个 PLL，它可以与 HFINTOSC 时钟源配合使用，用于提供系统时钟源。PLL 的输入频率限制为 8 MHz、12 MHz 或 16 MHz，将分别产生 16 MHz、24 MHz 或 32 MHz 的系统时钟源。

PLL 可以通过两种方法之一使能：

1. 将配置字 1 中的 RSTOSC 位编程为 000 以使能 HFINTOSC（32 MHz）。此设置将 HFFRQ<2:0> 位配置为 110（16 MHz），并激活 2x PLL。
2. 向 OSCCON1 寄存器的 NOSC<2:0> 位写入 000 以使能 2x PLL，并将正确值写入 OSCFRQ 寄存器的 HFFRQ<3:0> 位来选择所需系统时钟频率。更多信息，请参见寄存器 7-6。

PIC16(L)F18324/18344

7.2.2.3 内部振荡器频率调整

HFINTOSC 和 LFINTOSC 内部振荡器在出厂时已校准。HFINTOSC 振荡器可以通过用软件写入 OSCTUNE 寄存器（寄存器 7-3）进行调整。OSCTUNE 寄存器不影响 LFINTOSC 频率。

OSCTUNE 寄存器的默认值为 00h。该值为 6 位的二进制补码。值为 1Fh 时，将调整为最高频率。值为 20h 时，将调整为最低频率。

当 OSCTUNE 寄存器被修改时，HFINTOSC 振荡器频率将开始改变为新频率。改变期间，代码将继续执行。不会明确指示是否已发生频率改变。

7.2.2.4 LFINTOSC

低频内部振荡器（LFINTOSC）在出厂时已校准为 31 kHz 内部时钟源。

LFINTOSC 是上电延时定时器（PWRT）、看门狗定时器（WDT）以及故障保护时钟监视器（FSCM）的时钟源。LFINTOSC 也可用作系统时钟，或用作某些外设的时钟或输入源。

可通过以下方法之一选择 LFINTOSC 作为时钟源：

- 编程配置字 1 的 RSTOSC<2:0> 位来使能 LFINTOSC。
- 写 OSCCON1 寄存器的 NOSC<2:0> 位。

7.2.2.5 振荡器状态和手动使能

每个振荡器的“就绪”状态显示在 OSCSTAT1 寄存器（寄存器 7-4）中。也可通过 OSCEN 寄存器（寄存器 7-5）手动使能振荡器。手动使能可验证 EXTOSC 或 SOSC 晶振的工作状态。这可通过使能所选振荡器，然后观察 OSCSTAT1 寄存器中振荡器的相应“就绪”状态位实现。

7.3 时钟切换

使用 OSCCON1 寄存器的新振荡器源 (NOSC) 位和新分频比选择请求选择 (NDIV) 位, 可通过软件在外部和内部时钟源之间切换系统时钟源。可以选择以下时钟源:

- 外部振荡器 (EXTOSC)
- 高频内部振荡器 (HFINTOSC)
- 低频内部振荡器 (LFINTOSC)
- 辅助振荡器 (SOSC)
- 采用 4x PLL 的 EXTOSC
- 采用 2x PLL 的 HFINTOSC

7.3.1 新振荡器源 (NOSC) 位和新分频比选择请求 (NDIV) 位

OSCCON1 寄存器的新振荡器源 (NOSC) 位和新分频比选择请求 (NDIV) 位选择用于 CPU 和外设的系统时钟源及其频率。

当将 NOSC<2:0> 和 NDIV<3:0> 的新值写入 OSCCON1 时, 当前选择的振荡器将在等待新时钟源指示其已稳定并就绪过程中继续作为系统时钟运行。在某些情况下, 新请求的时钟源可能已在使用并立即就绪。对于仅发生分频比更改的情况, 新时钟源和旧时钟源相同并立即就绪。器件在等待切换时可能进入休眠模式, 如第 7.3.3 节“时钟切换和休眠”所述。

当新振荡器就绪时, OSCCON3 寄存器的新振荡器就绪 (NOSCR) 位和 PIR3 寄存器的时钟切换中断标志 (CSWIF) 位置 1 (CSWIF = 1)。如果允许时钟切换中断 (CSWIE = 1), 则此时将产生中断。除了中断, 还可通过轮询 OSCCON3 的振荡器就绪 (ORDY) 位来确定振荡器何时已就绪。

如果 OSCCON3 寄存器的时钟切换暂停 (CSWHOLD) 位清零, 在新振荡器就绪 (NOSCR) 位置 1 时将发生振荡器切换, 并且将以新振荡器设置处理中断 (如果允许了中断的话)。

如果 CSWHOLD 置 1, 振荡器切换将暂停, 而器件执行将继续使用当前 (旧) 时钟源。当 NOSCR 位置 1 时, 软件应

- 设置 CSWHOLD = 0 以完成切换, 或
- 将 COSC 的值复制到 NOSC<2:0> 以放弃切换。

如果 DOZE 有效, 那么将在下一个时钟周期发生切换, 而无论 CPU 在该周期期间是否正在运行。

更改时钟后分频比而不更改时钟源 (如, 将 Fosc 从 1 MHz 更改为 2 MHz) 的处理方式与更改时钟源相同, 如前面所述。时钟源已经在工作, 因此切换相对较快。CSWHOLD 必须清零 (CSWHOLD = 0), 切换才能完成。

当前 COSC 和 CDIV 在 OSCCON2 寄存器中指示, 直至发生实际切换, 此时 OSCCON2 将更新且 ORDY 置 1。NOSCR 由硬件清零以指示切换完成。

7.3.2 PLL 输入切换

如上所述管理带 PLL 的时钟源和任何不带 PLL 的时钟源之间的切换。当 NOSC<2:0> 选择 PLL 时建立 PLL 的输入时钟源, 并通过 COSC 设置维持。

当 NOSC<2:0> 和 COSC 位选择带 PLL 的不同输入时钟源时, 系统继续使用 COSC 设置运行, 并且根据 NOSC<2:0> 位的值使能新振荡器源。当新振荡器就绪 (且 CSWHOLD = 0) 时, 系统操作在切换 PLL 输入时钟源且 PLL 获得锁定时暂停。

7.3.3 时钟切换和休眠

如果 OSCCON1 写入新值并且在切换完成之前将器件置于休眠模式, 那么切换将不会发生且器件进入休眠模式。

当器件从休眠模式唤醒且 CSWHOLD 位清零时, 器件唤醒时将采用“新”时钟, 且时钟切换中断标志位 (CSWIF) 将置 1。

当器件从休眠模式唤醒且 CSWHOLD 位置 1 时, 器件唤醒时将采用“旧”时钟并且再次请求新时钟。

PIC16(L)F18324/18344

图7-6: 时钟切换 (CSWHOLD = 0)

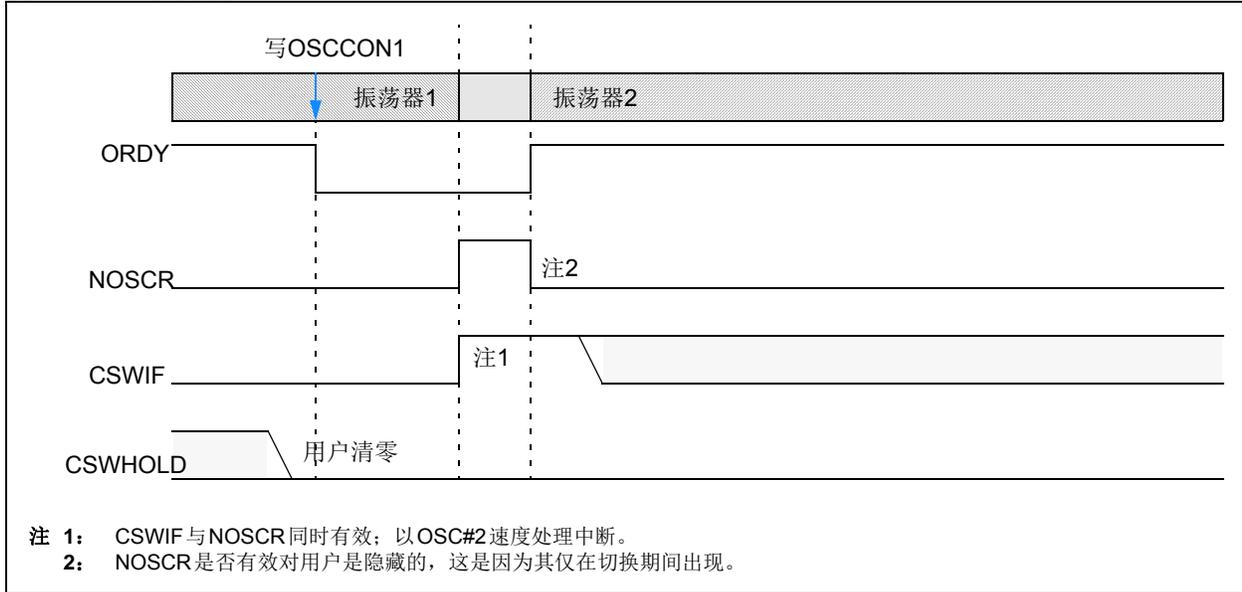


图7-7: 时钟切换 (CSWHOLD = 1)

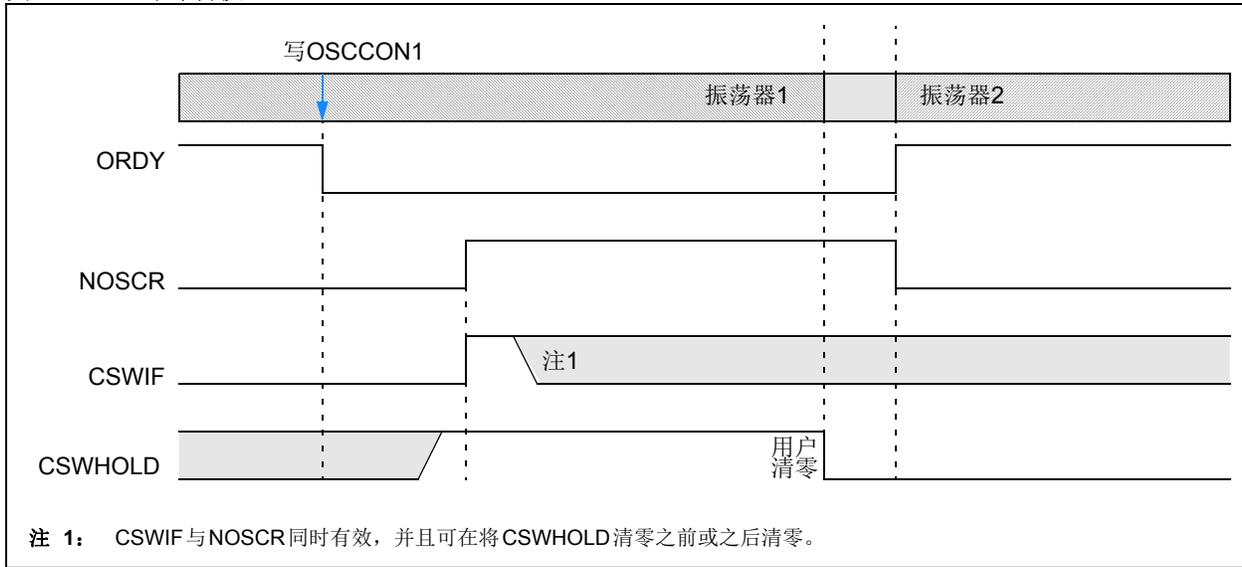
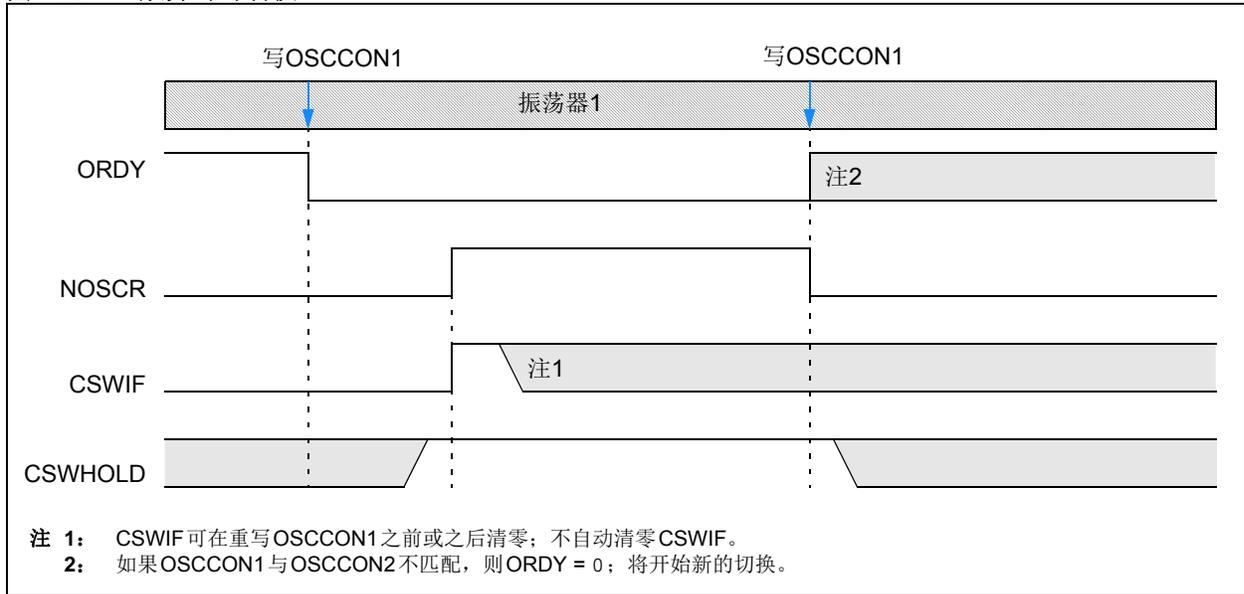


图7-8: 放弃时钟切换

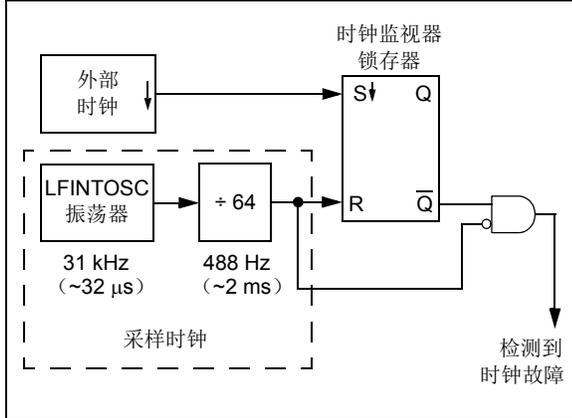


PIC16(L)F18324/18344

7.4 故障保护时钟监视器

故障保护时钟监视器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM通过将配置字中的FCMEN位置1来使能。FSCM可用于所有外部振荡器模式（LP、XT、HS、ECL、ECM、ECH 和辅助振荡器）。

图7-9: FSCM框图



7.4.1 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。通过对LFINTOSC时钟进行64分频得到采样时钟。请参见图7-9。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，外部时钟将锁存器置1。在采样时钟的每个上升沿，采样时钟将锁存器清零。如果已经经过采样时钟的整个半周期，但外部时钟仍未变为低电平，则会检测到故障。

7.4.2 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HFINTOSC（1 MHz时钟频率），并将PIR3寄存器的OSFIF标志位置1。如果PIE3寄存器的OSFIE位也置1，则此标志置1会产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续由内部时钟源提供，直到器件固件成功重启外部振荡器并切换回外部振荡器进行工作，这通过写OSCCON1寄存器的NOSC<2:0>和NDIV<3:0>位实现。

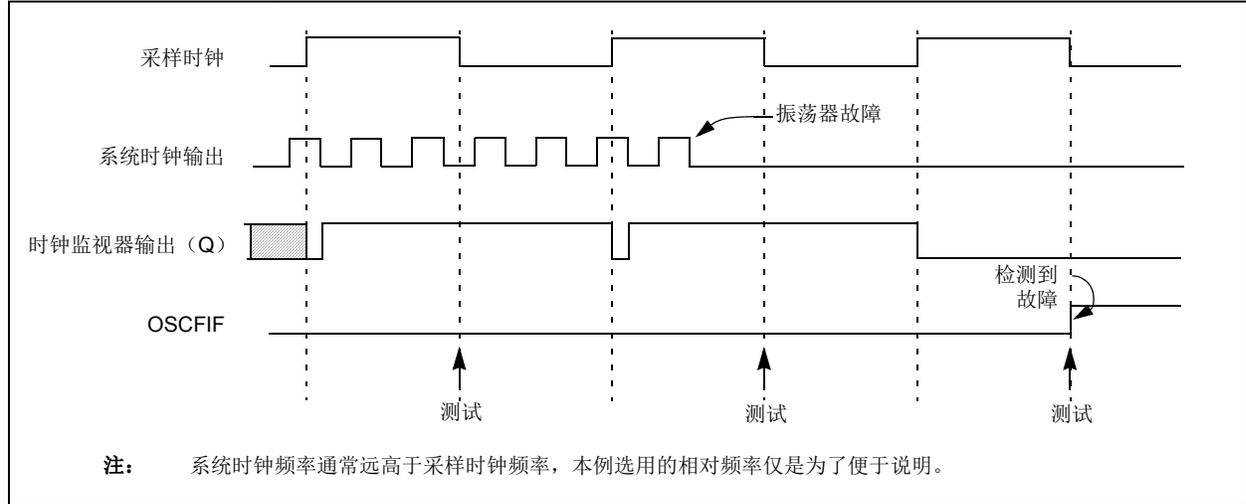
7.4.3 故障保护条件清除

在复位、执行SLEEP指令或更改OSCCON1寄存器的NOSC<2:0>和NDIV<3:0>位之后，故障保护条件被清除。当切换到外部振荡器或带PLL的外部振荡器时，重启OST。OST运行时，器件将依靠OSCCON1中选定的INTOSC继续工作。OST超时后，故障保护条件会在成功切换到外部时钟源之后被清除。在切换到外部时钟源之前，应先清零OSFIF位。如果故障保护条件仍然存在，硬件会再次将OSFIF标志置1。

7.4.4 复位或从休眠唤醒

FSCM设计为用于在振荡器起振定时器（OST）延时结束后检测振荡器故障。从休眠模式唤醒后以及任何类型的复位后使用OST。在EC时钟模式下不使用OST，所以可根据需要停止外部时钟信号。因此，当使用EC模式之一且OST运行时，器件总是在执行代码。

图7-10: FSCM时序图



PIC16(L)F18324/18344

7.5 寄存器定义：振荡器控制

寄存器 7-1: **OSCCON1: 振荡器控制寄存器 1**

U-0	R/W-f/f ⁽¹⁾	R/W-f/f ⁽¹⁾	R/W-f/f ⁽¹⁾	R/W-q/q ⁽⁴⁾	R/W-q/q ⁽⁴⁾	R/W-q/q ⁽⁴⁾	R/W-q/q ⁽⁴⁾
—	NOSC<2:0> ^(2,3)			NDIV<3:0> ^(2,3)			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	f = 由熔丝设置确定
q = 复位值由硬件确定		

bit 7 **未实现:** 读为0

bit 6-4 **NOSC<2:0>:** 新振荡器源请求位
设置按照表7-1请求振荡器源和PLL组合。
POR值 = RSTOSC (寄存器5-1) 的值。

bit 3-0 **NDIV<3:0>:** 新分频比选择请求位
设置按照表7-2确定后分频器新分频比。

注 1: 默认值 (f/f) 设置为等于RSTOSC配置位。

2: 如果将保留值写入NOSC (表7-1), 自动选择HFINTOSC作为时钟源。

3: 当CSWEN = 0时, 该寄存器为只读且不能更改为POR值以外的值。

4: 当RSTOSC = 110 (HFINTOSC 1 MHz) 时, NDIV位在复位时默认为0010; 对于所有其他NOSC设置, NVID位在复位时默认为0000。

寄存器 7-2: **OSCCON2: 振荡器控制寄存器 2**

U-0	R-q/q ⁽¹⁾						
—	COSC<2:0>			CDIV<3:0>			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	f = 由熔丝设置确定
q = 复位值由硬件确定		

bit 7 **未实现:** 读为0

bit 6-4 **COSC<2:0>:** 当前振荡器源选择位 (只读)
按照表7-1指示当前源振荡器和PLL组合。

bit 3-0 **CDIV<3:0>:** 当前分频比选择位 (只读)
按照表7-2指示后分频器当前分频比。

注 1: 复位值 (q/q) 将与NOSC<2:0>/NDIV<3:0>位匹配。

表7-1: NOSC/COSC位设置

NOSC<2:0> COSC<2:0>	时钟源
111	EXTOSC ⁽¹⁾
110	HFINTOSC (1 MHz)
101	保留
100	LFINTOSC
011	SOSC
010	保留
001	带4xPLL的EXTOSC ⁽¹⁾
000	带2x PLL的HFINTOSC (32 MHz)

注 1: EXTOSC通过配置字1 (寄存器5-1) 的 FEXTOSC位配置。

表7-2: NDIV/CDIV位设置

NDIV<3:0> CDIV<3:0>	时钟分频比
1111-1010	保留
1001	512
1000	256
0111	128
0110	64
0101	32
0100	16
0011	8
0010	4
0001	2
0000	1

寄存器 7-3: OSCCON3: 振荡器控制寄存器3

R/W/HC-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	U-0	U-0	U-0
CSWHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零 f = 由熔丝设置确定
q = 复位值由硬件确定 HC = 硬件清零位

- bit 7 **CSWHOLD:** 时钟切换暂停位
1 = 当NOSC选择的振荡器就绪时, 时钟切换将暂停 (通过中断)
0 = 当NOSC选择的振荡器就绪时, 时钟切换可能继续; 如果此时该位置1, NOSCR变为1, 将发生切换并产生中断。
- bit 6 **SOSCPWR:** 辅助振荡器功耗模式选择位
如果 SOSCBE = 0
1 = 辅助振荡器工作在高功耗模式下
0 = 辅助振荡器工作在低功耗模式下
如果 SOSCBE = 1
x = 该位被忽略
- bit 5 **SOSCBE:** 辅助振荡器旁路使能位
1 = 辅助振荡器SOSCI配置为外部时钟输入 (ST缓冲器); SOSC0未使用。
0 = 辅助振荡器配置为使用SOSC0和SOSCI引脚的晶振。
- bit 4 **ORDY:** 振荡器就绪位 (只读)
1 = OSCCON1 = OSCCON2; 当前系统时钟为NOSC所指定的时钟
0 = 正在进行时钟切换
- bit 3 **NOSCR:** 新振荡器就绪位 (只读)
1 = 正在进行时钟切换并且NOSC所选的振荡器指示“就绪”条件
0 = 未在进行时钟切换, 或者NOSC选择的振荡器尚未就绪
- bit 2-0 **未实现:** 读为0。

PIC16(L)F18324/18344

寄存器 7-4: **OSCSTAT1: 振荡器状态寄存器 1**

R-q/q	R-q/q	U-0	R-q/q	R-q/q	R-q/q	U-0	R-q/q
EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLL R
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	q = 复位值由硬件确定

- bit 7 **EXTOR:** EXTOSC (外部) 振荡器就绪
1 = 振荡器已就绪备用
0 = 振荡器未使能, 或尚未就绪备用。
- bit 6 **HFOR:** HFINTOSC 振荡器就绪
1 = 振荡器已就绪备用
0 = 振荡器未使能, 或尚未就绪备用。
- bit 5 **未实现:** 读为0
- bit 4 **LFOR:** LFINTOSC 振荡器就绪
1 = 振荡器已就绪备用
0 = 振荡器未使能, 或尚未就绪备用。
- bit 3 **SOR:** 辅助振荡器就绪
1 = 振荡器已就绪备用
0 = 振荡器未使能, 或尚未就绪备用。
- bit 2 **ADOR:** ADCRC 振荡器就绪
1 = 振荡器已就绪备用
0 = 振荡器未使能, 或尚未就绪备用。
- bit 1 **未实现:** 读为0
- bit 0 **PLL R:** PLL 就绪
1 = PLL 已就绪
0 = PLL 未使能, 所需输入源未就绪, 或PLL未就绪。

寄存器 7-5: OSCEN: 振荡器手动使能寄存器

R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EXTOEN	HFOEN	—	LFOEN	SOSCEN	ADOEN	—	—
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **EXTOEN:** 外部振荡器手动请求使能位
1 = 已明确使能EXTOSC, 具体操作由FEXTOSC指定
0 = EXTOSC可由另一个模块使能
- bit 6 **HFOEN:** HFINTOSC振荡器手动请求使能位
1 = 已明确使能HFINTOSC, 具体操作由OSCFRQ ([寄存器7-6](#)) 指定
0 = HFINTOSC可由另一个模块使能
- bit 5 **未实现:** 读为0
- bit 4 **LFOEN:** LFINTOSC (31 kHz) 振荡器手动请求使能位
1 = 已明确使能LFINTOSC
0 = LFINTOSC可由另一个模块使能
- bit 3 **SOSCEN:** 辅助振荡器手动请求使能位
1 = 已明确使能辅助振荡器
0 = 辅助振荡器可由另一个模块使能
- bit 2 **ADOEN:** ADOSC (600 kHz) 振荡器手动请求使能位
1 = 已明确使能ADOSC
0 = ADOSC可由另一个模块使能
- bit 1 **未实现:** 读为0
- bit 0 **未实现:** 读为0

PIC16(L)F18324/18344

寄存器 7-6: **OSCFRQ: HFINTOSC 频率选择寄存器**

U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	—	—	HFFRQ<3:0>			
bit 7				bit 0			

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置1 0 = 清零

bit 7-4 **未实现:** 读为0。

bit 3-0 **HFFRQ<3:0>:** HFINTOSC 频率选择位

HFFRQ<3:0>	标称频率 (MHz) (NOSC = 110)	2xPLL 频率 (MHz) (NOSC = 000)
0000	1	保留
0001	2	
0010	保留	
0011	4	
0100	8	16
0101	12	24
0110	16	32
0111	32	保留
1xxx	32	

PIC16(L)F18324/18344

寄存器 7-7: OSCTUNE: HFINTOSC调节寄存器

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	
—	—	HFTUN<5:0>						
bit 7								bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

f = 由熔丝设置确定

q = 复位值由硬件确定

bit 7-6 **未实现:** 读为0。

bit 5-0 **HFTUN<5:0>:** HFINTOSC 频率调节位

01 1111 = 最高频率

01 1110

•

•

•

00 0001

00 0000 = 中心频率。振荡器模块以校准的频率运行 (默认值)。

11 1111

•

•

•

10 0000 = 最低频率。

PIC16(L)F18324/18344

表7-3: 与时钟源相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
OSCCON1	—	NOSC<2:0>			NDIV<3:0>				88
OSCCON2	—	COSC<2:0>			CDIV<3:0>				88
OSCCON3	CWSHOLD	SOSCPWR	SOSCBE	ORDY	NOSCR	—	—	—	89
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR	90
OSCGEN	EXTOEN	HFOEN	—	LFOEN	SOSCGEN	ADOEN	—	—	91
OSCFRQ	—	—	—	—	HFFRQ<3:0>				92
OSCTUNE	—	—	HFTUN<5:0>						93

图注: — = 未实现位, 读为0。时钟源不使用阴影单元。

表7-4: 与时钟源相关的配置字汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	FCMEN	—	CSWEN	—	—	CLKOUTEN	62
	7:0	—	RSTOSC2	RSTOSC1	RSTOSC0	—	FEXTOSC2	FEXTOSC1	FEXTOSC0	

图注: — = 未实现位, 读为0。时钟源不使用阴影单元。

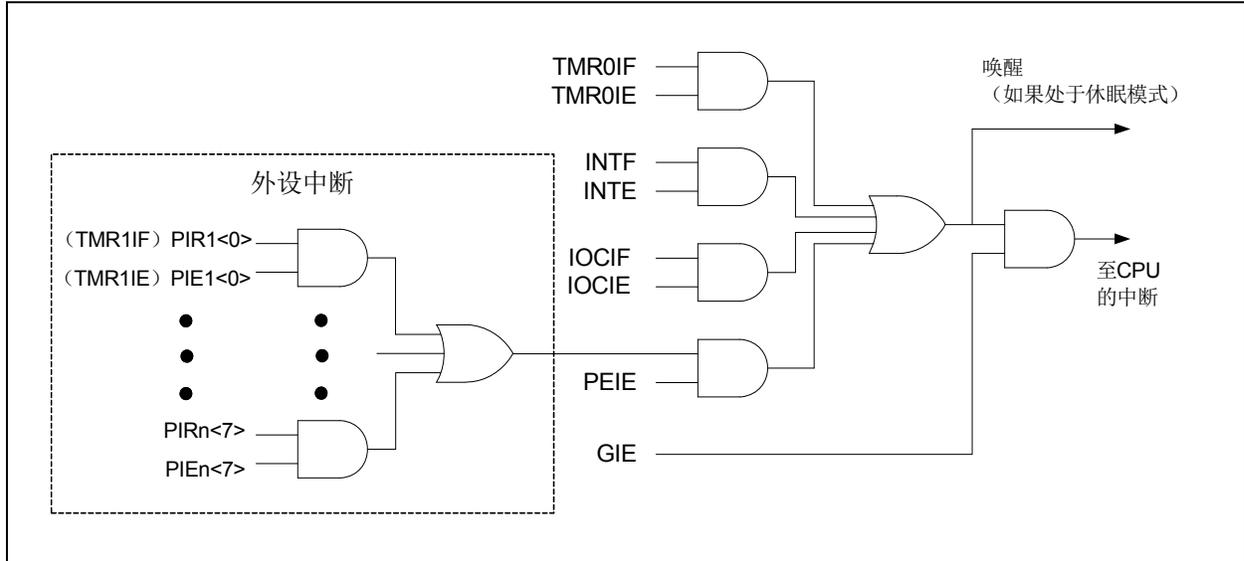
8.0 中断

通过中断功能，一些事件可以抢占正常的程序流。固件用于确定中断源，并执行相应的操作。有些中断可配置为将MCU从休眠模式唤醒。

许多外设都会产生中断。详情请参见相应章节。

图8-1给出了中断逻辑的框图。

图8-1： 中断逻辑



PIC16(L)F18324/18344

8.1 工作

任何器件复位时都会禁止中断。通过将以下位置1允许中断：

- INTCON寄存器的GIE位
- 特定中断事件的中断允许位（PIEx位）
- INTCON寄存器的PEIE位

PIR1、PIR2、PIR3和PIR4寄存器通过中断标志位记录各个中断。无论GIE、PEIE和各个中断允许位的状态如何，中断标志位都会在中断发生时置1。

当中断事件发生时，若GIE位置1，将发生以下事件：

- 清除当前预取的指令
- GIE位清零
- 程序计数器（PC）的当前值压入堆栈
- 自动将关键寄存器保存到影子寄存器中（见第8.5节“自动现场保护”）
- 将中断向量0004h装入PC

中断服务程序（Interrupt Service Routine, ISR）中的固件应通过查询中断标志位来确定中断源。退出ISR前必须清零中断标志位，以避免重复中断。由于GIE位清零，执行ISR期间发生的任何中断都会通过其中断标志位记录，但不会使处理器重定向到中断向量。

通过RETFIE指令退出ISR时，从堆栈弹出先前保存的地址、从影子寄存器恢复保存的现场数据并将GIE位置1。

关于特定中断操作的更多信息，请参见其外设章节。

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

8.2 中断延时

中断延时定义为从发生中断事件到开始执行中断向量处代码经过的时间。在指令周期的Q1期间对中断进行采样。因此，实际的中断延时取决于检测到中断时正在执行的指令。更多详细信息，请参见图8-2和图8-3。

图8-2: 中断延时

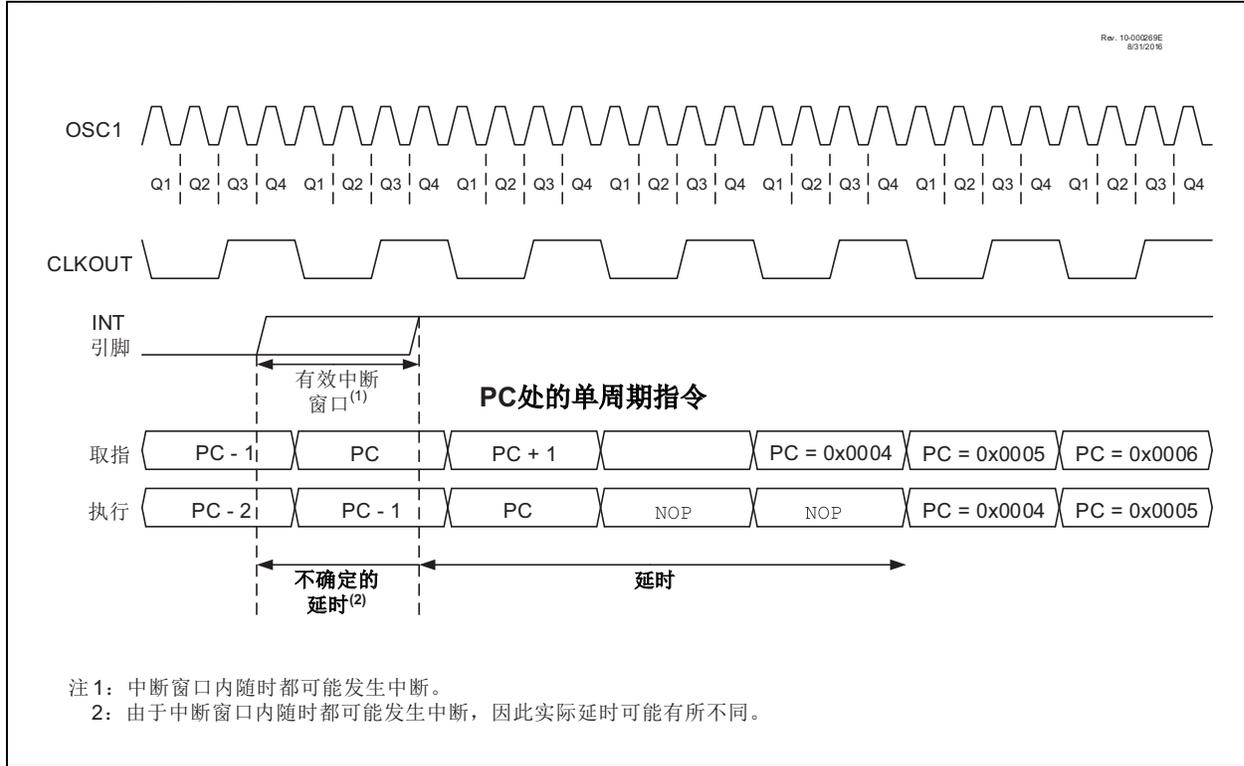
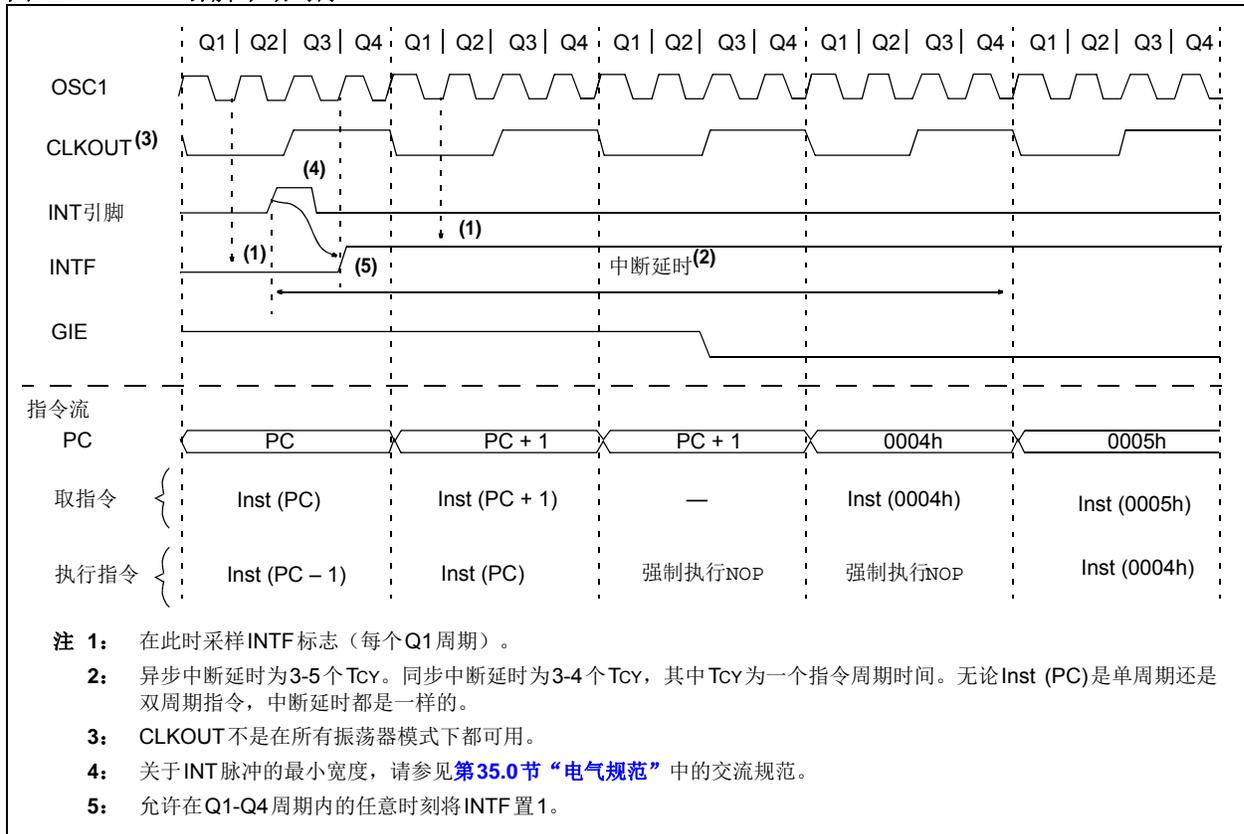


图8-3: INT引脚中断时序



PIC16(L)F18324/18344

8.3 休眠期间的中断

所有中断可用于将器件从休眠模式唤醒。要从休眠模式唤醒器件，外设必须能在没有系统时钟的情况下工作。进入休眠模式前，必须将相应中断源的中断允许位置1。

从休眠模式唤醒时，如果GIE位也置1，则处理器将跳转到中断向量。否则，处理器将继续执行SLEEP指令后的指令。紧接SLEEP指令后的指令总是会在跳转到ISR前执行。更多详细信息，请参见第9.0节“节能工作模式”。

8.4 INT引脚

INT引脚可用于产生异步边沿触发中断。可以通过将PIE0寄存器的INTE位置1来允许该中断。INTCON寄存器的INTEDG位确定中断在哪个边沿发生。INTEDG位置1时，上升沿将引起中断。INTEDG位清零时，下降沿将引起中断。PIR0寄存器的INTF位将在INT引脚上出现有效边沿时置1。如果GIE和INTE位也置1，则处理器会将程序执行重定向到中断向量。

8.5 自动现场保护

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

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

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

8.6 寄存器定义：中断控制

寄存器 8-1: **INTCON**: 中断控制寄存器

R/W/HS/HC-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	R-1/1
GIE	PEIE	—	—	—	—	—	INTEDG
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	
HS = 硬件置1位	HC = 硬件清零位	

- bit 7 **GIE**: 全局中断允许位
1 = 允许所有有效中断
0 = 禁止所有中断
- bit 6 **PEIE**: 外设中断允许位
1 = 允许所有有效外设中断
0 = 禁止所有外设中断
- bit 5-1 **未实现**: 读为0。
- bit 0 **INTEDG**: 中断边沿选择位
1 = INT引脚的上升沿触发中断
0 = INT引脚的下降沿触发中断

注: 要允许任何外设中断, INTCON 寄存器的 PEIE 位必须置1。

PIC16(L)F18324/18344

寄存器 8-2: **PIE0: 外设中断允许寄存器0**

U-0	U-0	R/W/HS-0/0	R/W-0/0	U-0	U-0	U-0	R/W/HS-0/0
—	—	TMR0IE	IOIE	—	—	—	INTE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1

bit 7-6 **未实现:** 读为0

bit 5 **TMROIE:** TMRO溢出中断允许位

1 = 允许TMRO中断

0 = 禁止TMRO中断

bit 4 **IOIE:** 电平变化中断允许位

1 = 允许IOC电平变化中断

0 = 禁止IOC电平变化中断

bit 3-1 **未实现:** 读为0

bit 0 **INTE:** INT外部中断标志位⁽¹⁾

1 = 允许INT外部中断

0 = 禁止INT外部中断

注 1: 外部中断INT引脚通过INTPPS (寄存器13-1) 选择。

注: 要允许任何外设中断, INTCON寄存器的PEIE位必须置1。

寄存器 8-3: PIE1: 外设中断允许寄存器1

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TMR1GIE | ADIE | RCIE | TXIE | SSP1IE | BCL1IE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **TMR1GIE:** Timer1 门控中断允许位
1 = 允许 Timer1 门控采集中断
0 = 禁止 Timer1 门控采集中断
- bit 6 **ADIE:** 模数转换器 (ADC) 中断允许位
1 = 允许 ADC 中断
0 = 禁止 ADC 中断
- bit 5 **RCIE:** EUSART 接收中断允许位
1 = 允许 EUSART 接收中断
0 = 禁止 EUSART 接收中断
- bit 4 **TXIE:** EUSART 发送中断允许位
1 = 允许 EUSART 发送中断
0 = 禁止 EUSART 发送中断
- bit 3 **SSP1IE:** 同步串行端口 (MSSP) 中断允许位
1 = 允许 MSSP 中断
0 = 禁止 MSSP 中断
- bit 2 **BCL1IE:** MSSP1 总线冲突中断允许位
1 = 允许 MSSP 总线冲突中断
0 = 禁止 MSSP 总线冲突中断
- bit 1 **TMR2IE:** TMR2 与 PR2 匹配中断允许位
1 = 允许 Timer2 与 PR2 匹配中断
0 = 禁止 Timer2 与 PR2 匹配中断
- bit 0 **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

注: 要允许任何外设中断, INTCON 寄存器的 PEIE 位必须置 1。

PIC16(L)F18324/18344

寄存器 8-4: **PIE2: 外设中断允许寄存器 2**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **TMR6IE:** TMR6与PR6匹配中断允许位
1 = 允许TMR6与PR6匹配中断
0 = 禁止TMR6与PR6匹配中断
- bit 6 **C2IE:** 比较器C2中断允许位
1 = 允许比较器C2中断
0 = 禁止比较器C2中断
- bit 5 **C1IE:** 比较器C1中断允许位
1 = 允许比较器C1中断
0 = 禁止比较器C1中断
- bit 4 **NVMIE:** NVM中断允许位
1 = 允许NVM任务完成中断
0 = 禁止NVM中断
- bit 3-2 **未实现:** 读为0
- bit 1 **TMR4IE:** TMR4与PR4匹配中断允许位
1 = 允许TMR4与PR4匹配中断
0 = 禁止TMR4与PR4匹配中断
- bit 0 **NCO1IE:** NCO中断允许位
1 = 允许NCO计满返回中断
0 = 禁止NCO计满返回中断

注: 要允许任何外设中断, INTCON寄存器的PEIE位必须置1。

寄存器 8-5: PIE3: 外设中断允许寄存器3

R/W-0/0							
OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE
bit 7							bit 0

图注:

R = 可读位
u = 不变
1 = 置1

W = 可写位
x = 未知
0 = 清零

U = 未实现位, 读为0
-n/n = POR和BOR时的值/所有其他复位时的值

- bit 7 **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
- bit 6 **CSWIE:** 时钟切换完成中断允许位
1 = 允许时钟切换模块中断
0 = 禁止时钟切换模块中断
- bit 5 **TMR3GIE:** Timer3 门控中断允许位
1 = 允许Timer3 门控中断
0 = 禁止Timer3 门控中断
- bit 4 **TMR3IE:** TMR3 溢出中断允许位
1 = 允许TMR3 溢出中断
0 = 禁止TMR3 溢出中断
- bit 3 **CLC4IE:** CLC4 中断标志位
1 = 允许CLC4 中断
0 = 禁止CLC4 中断
- bit 2 **CLC3IE:** CLC3 中断标志位
1 = 允许CLC3 中断
0 = 禁止CLC3 中断
- bit 1 **CLC2IE:** CLC2 中断允许位
1 = 允许CLC2 中断
0 = 禁止CLC2 中断
- bit 0 **CLC1IE:** CLC1 中断允许位
1 = 允许CLC1 中断
0 = 禁止CLC1 中断

注: 要允许任何外设中断, INTCON寄存器的PEIE位必须置1。

PIC16(L)F18324/18344

寄存器 8-6: PIE4: 外设中断允许寄存器4

R/W-0/0							
CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1

- bit 7 **CWG2IE:** CWG2 中断允许位
1 = 允许 CWG2 中断
0 = 禁止 CWG2 中断
- bit 6 **CWG1IE:** CWG1 中断允许位
1 = 允许 CWG1 中断
0 = 禁止 CWG1 中断
- bit 5 **TMR5GIE:** Timer5 门控中断允许位
1 = 允许 TMR5 门控中断
0 = 禁止 TMR5 门控中断
- bit 4 **TMR5IE:** TMR5 溢出中断允许位
1 = 允许 TMR5 溢出中断
0 = 禁止 TMR5 溢出中断
- bit 3 **CCP4IE:** CCP4 中断允许位
1 = 允许 CCP4 中断
0 = 禁止 CCP4 中断
- bit 2 **CCP3IE:** CCP3 中断允许位
1 = 允许 CCP3 中断
0 = 禁止 CCP3 中断
- bit 1 **CCP2IE:** CCP2 中断允许位
1 = 允许 CCP2 中断
0 = 禁止 CCP2 中断
- bit 0 **CCP1IE:** CCP1 中断允许位
1 = 允许 CCP1 中断
0 = 禁止 CCP1 中断

注: 要允许任何外设中断, INTCON 寄存器的 PEIE 位必须置1。

PIC16(L)F18324/18344

寄存器 8-7: PIR0: 外设中断请求寄存器0

U-0	U-0	R/W/HS-0/0	R-0	U-0	U-0	U-0	R/W/HS-0/0
—	—	TMR0IF	IOCF ⁽¹⁾	—	—	—	INTF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1

bit 7-6 **未实现:** 读为0

bit 5 **TMR0IF:** TMR0溢出中断标志位
1 = TMR0寄存器已发生溢出 (必须用软件清零)
0 = TMR0寄存器未溢出

bit 4 **IOCF:** 电平变化中断标志位 (只读)
1 = IOC模块检测到使能的边沿。某个IOCF位置1。
0 = IOC模块未检测到使能的边沿。没有IOCF位置1。
引脚通过IOCxP和IOCxN单独屏蔽。

bit 3-1 **未实现:** 读为0

bit 0 **INTF:** INT外部中断标志位⁽¹⁾
1 = 发生INT外部中断 (必须用软件清零)
0 = 未发生INT外部中断

注 1: IOCF位是所有IOCAF-IOCCF标志的逻辑或。因此, 要清零IOCF标志, 应用固件必须清零IOCAF-IOCCF寄存器的所有位。

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

PIC16(L)F18324/18344

寄存器 8-8: PIR1: 外设中断请求寄存器 1

R/W/HS-0/0	R/W/HS-0/0	R-0	R-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1

- bit 7 **TMR1GIF:** Timer1 门控中断标志位
1 = Timer1 门控已变为无效 (门控关闭)。
0 = Timer1 门控未变为无效。
- bit 6 **ADIF:** 模数转换器 (ADC) 中断标志位
1 = A/D 转换已完成
0 = A/D 转换未完成
- bit 5 **RCIF:** EUSART 接收中断标志位 (只读)
1 = EUSART1 接收缓冲区不为空
0 = EUSART1 接收缓冲区为空
- bit 4 **TXIF:** EUSART 发送中断标志位 (只读)
1 = EUSART1 发送缓冲区为空
0 = EUSART1 发送缓冲区不为空
- bit 3 **SSP1IF:** 同步串行端口 (MSSP) 中断标志位
1 = 发送/接收/总线条件已完成 (必须用软件清零)
0 = 等待发送/接收/总线条件发生
- bit 2 **BCL1IF:** MSSP 总线冲突中断标志位
1 = 检测到总线冲突 (必须用软件清零)
0 = 未检测到总线冲突
- bit 1 **TMR2IF:** Timer2 与 PR2 匹配中断标志位
1 = TMR2 与 PR2 发生匹配 (必须用软件清零)
0 = TMR2 与 PR2 未发生匹配
- bit 0 **TMR1IF:** Timer1 溢出中断标志位
1 = 发生了 TMR1 溢出 (必须用软件清零)
0 = 未发生 TMR1 溢出

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

寄存器 8-9: PIR2: 外设中断请求寄存器 2

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0
TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

HS = 硬件置 1

- bit 7 **TMR6IF:** TMR6 与 PR6 匹配中断标志位
 1 = TMR6 与 PR6 发生匹配 (必须用软件清零)
 0 = TMR6 与 PR6 未发生匹配
- bit 6 **C2IF:** 比较器 C2 中断标志位
 1 = 比较器 2 的中断置为有效
 0 = 比较器 2 的中断未置为有效
- bit 5 **C1IF:** 比较器 C1 中断标志位
 1 = 比较器 1 的中断置为有效
 0 = 比较器 1 的中断未置为有效
- bit 4 **NVMIF:** NVM 中断标志位
 1 = NVM 已完成编程任务
 0 = NVM 中断未置为有效
- bit 3-2 **未实现:** 读为 0。
- bit 1 **TMR4IF:** TMR4 与 PR4 匹配中断标志位
 1 = TMR4 与 PR4 发生匹配 (必须用软件清零)
 0 = TMR4 与 PR4 未发生匹配
- bit 0 **NCO1IF:** NCO 中断标志位
 1 = NCO 已计满返回。
 0 = NCO 中断未置为有效。

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

PIC16(L)F18324/18344

寄存器 8-10: PIR3: 外设中断请求寄存器 3

| R/W/HS-0/0 |
|------------|------------|------------|------------|------------|------------|------------|------------|
| OSFIF | CSWIF | TMR3GIF | TMR3IF | CLC4IF | CLC3IF | CLC2IF | CLC1IF |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1

- bit 7 **OSFIF:** 振荡器故障保护中断标志位
1 = 故障保护时钟监视器模块已检测到振荡器发生故障
0 = 外部振荡器正常工作
- bit 6 **CSWIF:** 时钟切换完成中断标志位
1 = 时钟切换模块已完成时钟切换; 新振荡器就绪
0 = 时钟切换模块未完成时钟切换
- bit 5 **TMR3GIF:** Timer3门控中断标志位
1 = TMR3门控已变为无效
0 = TMR3门控未变为无效
- bit 4 **TMR3IF:** TMR3溢出中断标志位
1 = 发生了TMR3溢出 (必须用软件清零)
0 = 未发生TMR3溢出
- bit 3 **CLC4IF:** CLC4中断标志位
1 = 已满足CLC4OUT中断条件
0 = 无CLC4中断
- bit 2 **CLC3IF:** CLC3中断标志位
1 = 已满足CLC3OUT中断条件
0 = 无CLC3中断
- bit 1 **CLC2IF:** CLC2中断标志位
1 = 已满足CLC2OUT中断条件
0 = 无CLC2中断
- bit 0 **CLC1IF:** CLC1中断标志位
1 = 已满足CLC1OUT中断条件
0 = 无CLC1中断

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

寄存器 8-11: PIR4: 外设中断请求寄存器4

R/W/HS-0/0							
CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1

bit 7

CWG2IF: CWG2中断标志位

1 = CWG2已进入关断状态
0 = CWG2正常工作, 或中断清除

bit 6

CWG1IF: CWG1中断标志位

1 = CWG1已进入关断状态
0 = CWG1正常工作, 或中断清除

bit 5

TMR5GIF: Timer5门控中断标志位

1 = TMR5门控已变为无效(门控关闭)。
0 = TMR5门控未变为无效。

bit 4

TMR5IF: Timer5溢出中断标志位

1 = 发生了TMR5溢出(必须用软件清零)
0 = 未发生TMR5溢出

bit 3

CCP4IF: CCP4中断标志位

值	CCPM模式		
	捕捉	比较	PWM
1	发生了捕捉(必须用软件清零)	发生了比较匹配(必须用软件清零)	出现了输出后沿(必须用软件清零)
0	未发生捕捉	未发生比较匹配	未出现输出后沿

bit 2

CCP3IF: CCP3中断标志位

值	CCPM模式		
	捕捉	比较	PWM
1	发生了捕捉(必须用软件清零)	发生了比较匹配(必须用软件清零)	出现了输出后沿(必须用软件清零)
0	未发生捕捉	未发生比较匹配	未出现输出后沿

PIC16(L)F18324/18344

寄存器 8-11: PIR4: 外设中断请求寄存器 4 (续)

bit 1

CCP2IF: CCP2 中断标志位

值	CCPM 模式		
	捕捉	比较	PWM
1	发生了捕捉 (必须用软件清零)	发生了比较匹配 (必须用软件清零)	出现了输出后沿 (必须用软件清零)
0	未发生捕捉	未发生比较匹配	未出现输出后沿

bit 0

CCP1IF: CCP1 中断标志位

值	CCPM 模式		
	捕捉	比较	PWM
1	发生了捕捉 (必须用软件清零)	发生了比较匹配 (必须用软件清零)	出现了输出后沿 (必须用软件清零)
0	未发生捕捉	未发生比较匹配	未出现输出后沿

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

表 8-1: 与中断相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	100
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
PIE2	TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	102
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	103
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	104
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	105
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
PIR2	TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF	107
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	108
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	109

图注: — = 未实现位, 读为 0。中断不使用阴影单元。

9.0 节能工作模式

掉电模式的目的是降低功耗。有三种掉电模式：打盹模式、空闲模式和休眠模式。

9.1 打盹模式

打盹模式通过减少CPU操作和程序存储器访问而不影响外设操作来支持节能。打盹模式与休眠模式的不同之处在于，系统振荡器继续运行，仅CPU和程序存储器受影响。通过消除CPU和存储器中的不必要操作来减少操作执行，从而节省功耗。

当打盹使能（DOZEN）位置1（DOZEN = 1）时，CPU每N个周期仅执行一个指令周期，这通过CPUDOZE寄存器的DOZE<2:0>位定义。例如，如果DOZE<2:0> = 100，则指令周期比是1:32。CPU和存储器执行一个指令周期，然后在31个指令周期内保持空闲。未使用的周期期间，外设继续以系统时钟速度运行。

9.1.1 打盹操作

图9-1给出了打盹操作。对于本例：

- 打盹使能（DOZEN）位置1（DOZEN = 1）
- DOZE<2:0> = 001（1:4）比
- 中断恢复（Recover-on-Interrupt, ROI）位置1（ROI = 1）

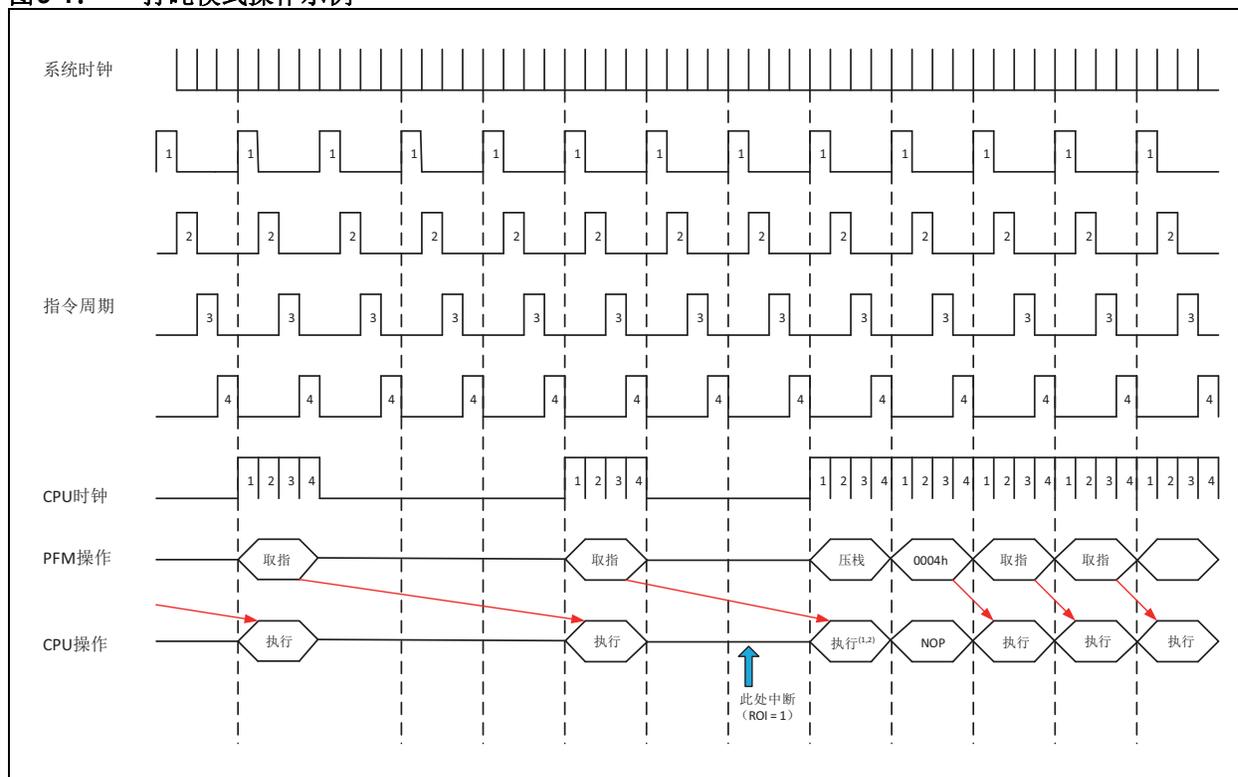
与正常操作一样，程序存储器为下一个指令周期取指。至外设的指令时钟继续运行。

9.1.2 打盹期间的中断

如果发生中断并且在中断时中断恢复（ROI）位清零（ROI = 0），中断服务程序（ISR）继续以DOZE<2:0>选择的速率执行。可通过DOZE<2:0>比延长中断延时。

如果发生中断并且在中断时ROI位置1（ROI = 1），DOZEN位清零且CPU以全速执行。执行预取指令，然后执行中断向量序列。在图9-1中，在打盹周期的第2个指令周期发生中断，并立即使CPU退出打盹模式。如果在执行RETFIE操作时退出打盹（Doze-On-Exit, DOE）位置1（DOE = 1），则DOZEN置1，并且CPU基于DOZE<2:0>比降速执行。

图9-1: 打盹模式操作示例



PIC16(L)F18324/18344

9.2 空闲模式

当空闲使能 (IDLEN) 位清零 (IDLEN = 0) 时, SLEEP 指令将器件置于完全休眠模式 (见第9.3节“休眠模式”)。当IDLEN置1 (IDLEN = 1) 时, SLEEP指令将器件置于空闲模式。在空闲模式下, CPU和存储器操作暂停, 但外设时钟继续运行。该模式与打盹模式类似, 但在空闲模式下CPU和程序存储器关闭。

注: 使用 Fosc 的外设将在空闲模式 (但不是休眠模式) 下继续运行。使用 HFINTOSC、LFINTOSC 或 SOSC 的外设将在空闲和休眠模式下继续运行。

注: 如果使能 CLKOUT (CLKOUT = 0, 配置字 1), 输出将在空闲模式下继续运行。

9.2.0.1 空闲与中断

发生中断时退出空闲模式 (即使 GIE = 0 也是如此), 但IDLEN不变。器件可通过执行SLEEP指令重新进入空闲模式。

如果同时使能中断恢复 (ROI = 1) 和打盹模式, 则发生中断时器件退出空闲模式并且CPU恢复全速执行。

9.2.0.2 空闲与WDT

在空闲模式下, WDT复位被禁止, 但会将器件唤醒。WDT唤醒不是中断, 因此ROI不适用。

注: WDT可使器件退出空闲模式, 与器件退出休眠模式的方式相同。DOZEN位不受影响。

9.3 休眠模式

通过执行SLEEP指令进入休眠模式, 而CPUDOZE寄存器的空闲使能 (IDLEN) 位清零 (IDLEN = 0)。如果在IDLEN位置1 (IDLEN = 1) 时执行SLEEP指令, CPU将进入空闲模式 (第9.3.3节“低功耗休眠模式”)。

在进入休眠模式时, 会存在以下条件:

1. WDT 之外的其他复位都不会受休眠模式影响; 如果在休眠期间使能WDT, 则WDT会清零, 但保持运行。
2. STATUS寄存器的 \overline{PD} 位清零。
3. STATUS寄存器的 \overline{TO} 位置1。
4. CPU和系统时钟被禁止。
5. 31 kHz LFINTOSC、HFINTOSC和SOSC将在任何外设请求其作为时钟源或者OSCEN寄存器中的HFOEN、LFOEN或SOSCEN位置1时保持使能状态。
6. 如果选择了专用ADCR振荡器, 则ADC不受影响。ADC时钟源不是ADCR时, 尽管ADON位仍保持置1, 但SLEEP指令会导致当前转换中止, ADC模块被关闭。
7. 仅当与I/O端口连接的外设均未激活时, I/O端口才会保持执行SLEEP指令之前的状态 (驱动为高电平、低电平或高阻态)。

关于休眠期间的外设操作的更多详细信息, 请参见各个章节。

要最大程度降低电流消耗, 应考虑以下条件:

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

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

可能拉电流的内部电路示例包括如DAC和FVR之类的模块。关于这些模块的更多信息, 请参见第24.0节“5位数模转换器 (DAC1) 模块”和第16.0节“固定参考电压 (FVR)”。

9.3.1 从休眠模式唤醒

发生以下任一事件将器件从休眠模式唤醒：

1. $\overline{\text{MCLR}}$ 引脚上的外部复位输入（如果使能）
2. BOR 复位（如果使能）。
3. POR 复位。
4. 看门狗定时器（如果使能）
5. 在休眠期间运行的外设所产生的中断（更多信息，请参见各个外设）。

前三个事件会导致器件复位。后两个事件视为继续执行程序。要确定是发生了器件复位还是唤醒事件，请参见第6.11节“确定复位原因”。

器件从休眠模式唤醒时，WDT 都将被清零，而与唤醒源无关。

9.3.2 使用中断唤醒

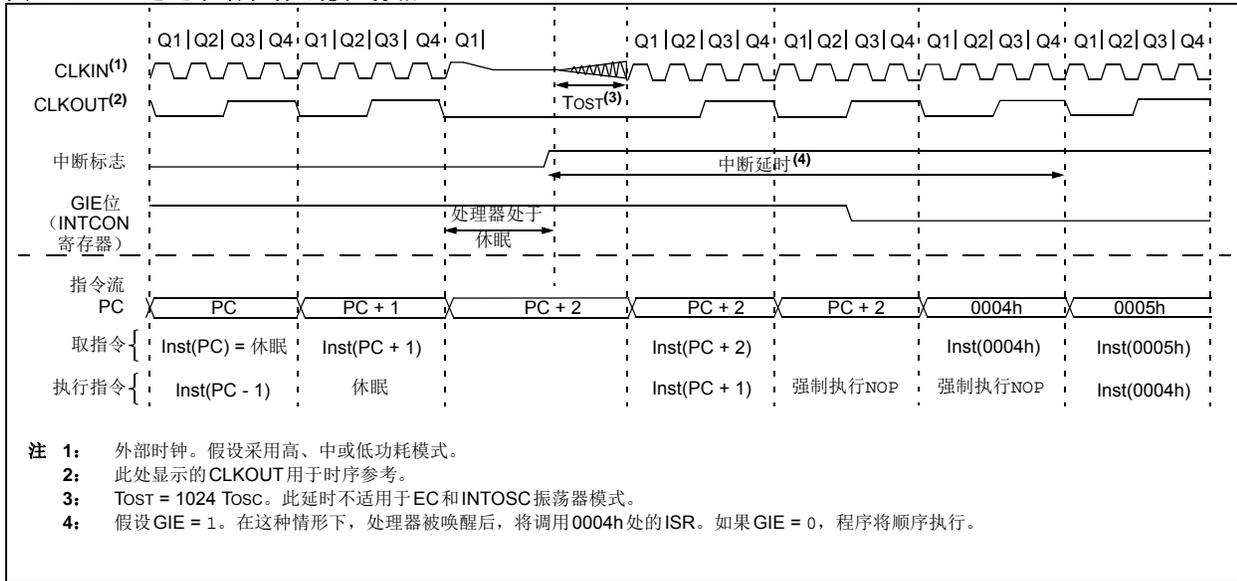
当禁止全局中断（GIE 被清零）时，并且任一中断源的中断允许位和中断标志位都置1，将会发生以下事件之一：

- 如果在执行 SLEEP 指令之前发生中断
 - SLEEP 指令将作为 NOP 指令执行
 - WDT 和 WDT 预分频器不会被清零
 - STATUS 寄存器的 $\overline{\text{TO}}$ 位不会被置1
 - STATUS 寄存器的 $\overline{\text{PD}}$ 位不会被清零
- 如果在执行 SLEEP 指令期间或之后发生中断
 - 将完整执行 SLEEP 指令
 - 器件将立即从休眠模式唤醒
 - WDT 和 WDT 预分频器将被清零
 - STATUS 寄存器的 $\overline{\text{TO}}$ 位将被置1
 - STATUS 寄存器的 $\overline{\text{PD}}$ 位将被清零

即使在执行 SLEEP 指令之前检查了标志位，这些标志位也有可能是在 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 $\overline{\text{PD}}$ 位。如果 $\overline{\text{PD}}$ 位置1，则说明 SLEEP 指令作为 NOP 指令执行了。

PIC16(L)F18324/18344

图9-2: 通过中断从休眠模式唤醒



9.3.3 低功耗休眠模式

PIC16F18324/18344 器件包含一个内部低压差 (Low Dropout, LDO) 稳压器，它让器件I/O引脚可以使用最高5.5V的电压工作，而内部器件逻辑可以使用较低的电压工作。在器件处于休眠模式时，LDO及其相关的参考电压电路必须保持活动状态。

PIC16F18324/18344 允许用户根据应用需求来优化休眠模式下的工作电流。

通过将VREGCON寄存器的VREGPM位置1，可以选择低功耗休眠模式。根据该位的配置，当器件处于休眠模式时，LDO和参考电压电路会被置为低功耗状态。

9.3.3.1 休眠电流与唤醒时间

在默认工作模式下，处于休眠模式时，LDO和参考电压电路会保持为正常配置。由于所有电路都保持活动状态，器件能够快速地退出休眠模式。在低功耗休眠模式下，从休眠模式中唤醒时，这些电路需要一个额外的延时，然后才会恢复为正常配置并稳定下来。

低功耗休眠模式对于需要长时间处于休眠模式的应用非常有益。正常模式对于需要快速地、频繁地从休眠模式中唤醒的应用非常有益。

9.3.3.2 休眠模式下的外设使用

选择低功耗休眠模式时，一些可以在休眠模式下工作的外设将无法正常工作。低功耗休眠模式旨在用于以下外设：

- 欠压复位 (BOR)
- 看门狗定时器 (WDT)
- 外部中断引脚/电平变化中断引脚
- Timer1 (带外部时钟源)

最终用户负责确定在进行VREGPM设置以确保休眠模式下的操作时其应用可接受的外设。

注： PIC16LF18324/18344 不具有可配置的低功耗休眠模式。PIC16LF18324/18344 是非稳压器件，它在休眠模式下总是处于最低功耗状态，并且没有唤醒时间延时。该器件的最大VDD和I/O电压低于PIC16F18324/18344。更多信息，请参见第35.0节“电气规范”。

9.4 寄存器定义：稳压器控制

寄存器9-1: VREGCON: 稳压器控制寄存器⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1
—	—	—	—	—	—	VREGPM	保留
bit 7						bit 0	

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-2 未实现: 读为0

bit 1 **VREGPM:** 稳压器功耗模式选择位
 1 = 休眠时使能低功耗休眠模式⁽²⁾; 休眠时消耗的电流最低, 唤醒速度较慢
 0 = 休眠时使能正常功耗休眠模式⁽²⁾; 休眠时消耗的电流较高, 唤醒速度较快

bit 0 保留: 读为1。保持该位置1。

注 1: 仅限PIC16F18324/18344。

2: 请参见第35.0节“电气规范”。

寄存器9-2: CPUDOZE: 打盹和空闲寄存器

R/W-0/u	R/W/HC/HS-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
IDLEN	DOZEN ^(1,2)	ROI	DOE	—	DOZE<2:0>		
bit 7						bit 0	

图注:

R = 可读位	W = 可写位	HS = 硬件置1位
u = 不变	x = 未知	U = 未实现位, 读为0
1 = 置1	0 = 清零	-n/n = POR和BOR时的值/所有其他复位时的值
		HS = 硬件清零位

bit 7 **IDLEN:** 空闲使能位
 1 = SLEEP指令禁止CPU时钟, 但不禁止外设时钟
 0 = SLEEP指令将器件置于完全休眠模式

bit 6 **DOZEN:** 打盹使能位^(1,2)
 1 = CPU根据DOZE设置执行指令周期
 0 = CPU执行所有指令周期(最快和最高功耗操作)

bit 5 **ROI:** 中断恢复位
 1 = 进入中断服务程序(ISR)使DOZEN位 = 0, CPU全速运行
 0 = 进入中断不改变DOZEN位状态

bit 4 **DOE:** 退出打盹位
 1 = 执行RETFIE使DOZEN = 1, CPU降速运行
 0 = RETFIE不改变DOZEN位状态

bit 3 未实现: 读为0

bit 2-0 **DOZE<2:0>:** CPU指令周期与外设指令周期比
 111 = 1:256
 110 = 1:128
 101 = 1:64
 100 = 1:32
 011 = 1:16
 010 = 1:8
 001 = 1:4
 000 = 1:2

注 1: 当ROI = 1或DOE = 1时, 通过进入和/或退出硬件中断改变DOZEN位状态。

2: 进入ICD将改写DOZEN, CPU返回到全速执行模式; 该位不受影响。

PIC16(L)F18324/18344

表9-1: 与掉电模式相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIE0	—	—	TMR0IE	IOCFIE	—	—	—	INTE	100
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
PIE2	TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	102
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	103
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	104
PIR0	—	—	TMR0IF	IOCFIF	—	—	—	INTF	105
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
PIR2	TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF	107
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	108
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	109
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	172
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	172
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	173
IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	173
IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	174
IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	174
IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	175
IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	175
IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	176
STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	28
VREGCON ⁽²⁾	—	—	—	—	—	—	VREGPM	—	115
CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE<2:0>			115
WDTCON	—	—	WDTPS<4:0>					SWDTEN	119

图注: — = 未实现位, 读为0。掉电模式下不使用阴影单元。

注 1: 仅限 PIC16(L)F18344。

2: 仅限 PIC16F18324/18344。

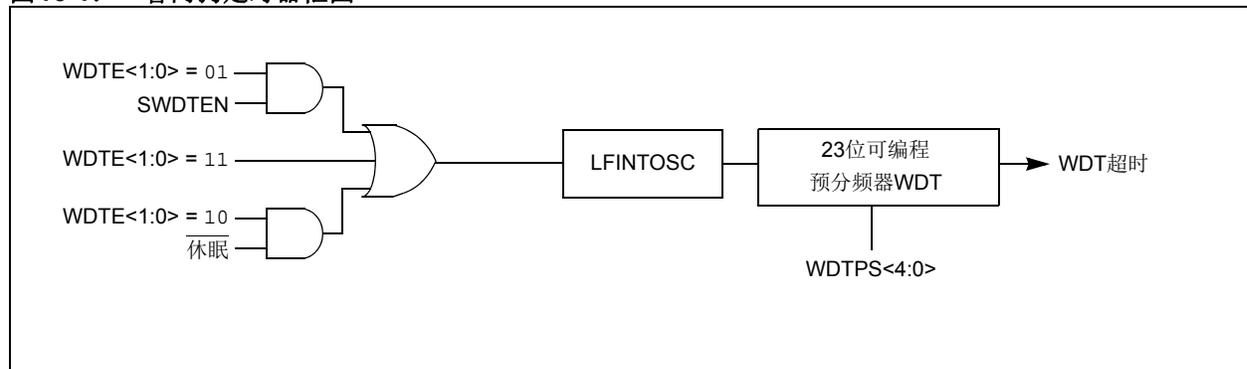
10.0 看门狗定时器 (WDT)

看门狗定时器是一个系统定时器，如果固件未在超时周期内发出 CLRWDT 指令，看门狗定时器会产生复位。看门狗定时器通常用于使系统从意外事件中恢复。

WDT 具有以下特性：

- 独立时钟源
- 多种工作模式
 - WDT 总是开启
 - WDT 在休眠模式下关闭
 - WDT 通过软件进行控制
 - WDT 总是关闭
- 超时周期可配置为从 1 ms 至 256s（标称值）
- 多种 WDT 清零条件
- 休眠期间的操作

图10-1： 看门狗定时器框图



10.1 独立时钟源

WDT以31 kHz LFINTOSC的内部振荡器作为其时基。本章中的时间间隔均基于1 ms的标称时间间隔。关于LFINTOSC规范，请参见表35-8。

10.2 WDT工作模式

看门狗定时器模块具有4种工作模式，这些工作模式由配置字中的WDTE<1:0>位控制。请参见表10-1。

10.2.1 WDT总是开启

当配置字的WDTE位设置为11时，WDT将总是使能。

WDT保护在休眠期间有效。

10.2.2 WDT在休眠模式下关闭

当配置字的WDTE位设置为10时，除非处于休眠模式，否则WDT将使能。

WDT保护在休眠期间无效。

10.2.3 WDT通过软件进行控制

当配置字的WDTE位设置为01时，WDT将通过WDTCON寄存器的SWDTEN位进行控制。

WDT保护在休眠期间不变。更多详细信息，请参见表10-1。

10.2.4 WDT总是关闭

当WDTE位设置为00时，WDT被禁止，WDTCON的SWDTEN位被忽略。

表10-1: WDT工作模式

WDTE<1:0>	SWDTEN	器件模式	WDT模式
11	X	X	有效
10	X	唤醒	有效
		休眠	禁止
01	1	X	有效
	0		禁止
00	X	X	禁止

表10-2: WDT清零条件

条件	WDT
WDTE = 00	清零并禁止
WDTE = 01且SWDTEN = 0	
因复位而退出休眠 + 系统时钟 = XT、HS或LP	清零，直到OST延时结束
因复位而退出休眠 + 系统时钟 = HFINTOSC、LFINTOSC、EC或SOSC	
因中断而退出休眠	清零
进入休眠	
CLRWDT命令	
振荡器故障（见第7.4节“故障保护时钟监视器”）	
系统复位	
任意时钟切换或分频比更改（见第7.3节“时钟切换”）	不受影响

10.3 超时周期

WDTCON寄存器的WDTPS<4:0>位用于设置从1 ms至256s（标称值）的超时周期。在复位之后，默认的超时周期为2秒。

10.4 清零WDT

当发生以下任何条件时，WDT被清零：

- 任何复位
- 执行了CLRWDT指令
- 器件进入休眠模式
- 器件由于中断从休眠模式唤醒
- 振荡器故障
- WDT被禁止
- 振荡器起振定时器（OST）正在运行

更多信息，请参见表10-2。

10.5 休眠期间的操作

当器件进入休眠模式时，WDT会被清零。如果使能WDT在休眠期间工作，WDT会继续计数。

当器件退出休眠模式时，WDT会被再次清零。WDT保持清零，直到OST（如果使能）延时结束为止。关于OST的更多信息，请参见第7.0节“振荡器模块”。

在器件处于休眠模式的情况下发生WDT超时，不会产生复位。器件将会唤醒并继续工作。STATUS寄存器中的TO和PD位会发生变化，指示发生的事件。更多信息，请参见STATUS寄存器（寄存器4-1）。

10.6 寄存器定义：看门狗控制

寄存器 10-1: WDTCON: 看门狗定时器控制寄存器

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0> ⁽¹⁾					SWDTEN
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-m/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 **未实现:** 读为0

bit 5-1 **WDTPS<4:0>:** 看门狗定时器周期选择位⁽¹⁾

位值 = 预分频比

11111 = 保留。产生最小的时间间隔 (1:32)

·

·

·

10011 = 保留。产生最小的时间间隔 (1:32)

10010 = 1:8388608 (2^{23}) (时间间隔标称值为256s)

10001 = 1:4194304 (2^{22}) (时间间隔标称值为128s)

10000 = 1:2097152 (2^{21}) (时间间隔标称值为64s)

01111 = 1:1048576 (2^{20}) (时间间隔标称值为32s)

01110 = 1:524288 (2^{19}) (时间间隔标称值为16s)

01101 = 1:262144 (2^{18}) (时间间隔标称值为8s)

01100 = 1:131072 (2^{17}) (时间间隔标称值为4s)

01011 = 1:65536 (时间间隔标称值为2s) (复位值)

01010 = 1:32768 (时间间隔标称值为1s)

01001 = 1:16384 (时间间隔标称值为512 ms)

01000 = 1:8192 (时间间隔标称值为256 ms)

00111 = 1:4096 (时间间隔标称值为128 ms)

00110 = 1:2048 (时间间隔标称值为64 ms)

00101 = 1:1024 (时间间隔标称值为32 ms)

00100 = 1:512 (时间间隔标称值为16 ms)

00011 = 1:256 (时间间隔标称值为8 ms)

00010 = 1:128 (时间间隔标称值为4 ms)

00001 = 1:64 (时间间隔标称值为2 ms)

00000 = 1:32 (时间间隔标称值为1 ms)

bit 0 **SWDTEN:** 看门狗定时器软件使能/禁止位

如果 WDTE<1:0> = 1x:

该位被忽略。

如果 WDTE<1:0> = 01:

1 = WDT 开启

0 = WDT 关闭

如果 WDTE<1:0> = 00:

该位被忽略。

注 1: 时间均为近似值。WDT 时间基于 31 kHz LFINTOSC。

表10-3: 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	寄存器所在页
STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	28
WDTCON	—	—	WDTPS<4:0>					SWDTEN	119

图注: x = 未知, u = 不变, — = 未实现位, 读为0。看门狗定时器不使用阴影单元。

表10-4: 与看门狗定时器相关的配置字汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5s	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG2	13:8	—	—	\overline{DEBUG}	STVREN	PPS1WAY	—	BORV	—	63
	7:0	BOREN1	BOREN0	$\overline{LPBOREN}$	—	WDTE1	WDTE0	\overline{PWRTEN}	MCLRE	

图注: — = 未实现位, 读为0。看门狗定时器不使用阴影单元。

11.0 非易失性存储器 (NVM) 控制

NVM分为两种类型：闪存程序存储器和数据EEPROM。

NVM可使用FSR和INDF寄存器访问或通过NVMREG寄存器接口进行访问。

写入时间由片上定时器控制。写入/擦除电压是由片上电荷泵产生的，此电荷泵在器件的工作电压范围内工作。

NVM可通过两种方式进行保护：代码保护或写保护。

代码保护（配置字4中的 $\overline{\text{CP}}$ 和 $\overline{\text{CPD}}$ 位）会禁止通过外部器件编程器对闪存程序存储器和EEPROM进行读写访问。代码保护不会影响自写和擦除功能。代码保护只能通过器件编程器对器件执行批量擦除操作，从而清除所有非易失性存储器、配置位和用户ID而复位。

写保护会禁止对由配置字3的WRT<1:0>位所定义的部分或全部闪存程序存储器进行自写和擦除操作。写保护不会影响器件编程器对器件进行读、写或擦除操作。

11.1 闪存程序存储器

闪存程序存储器由作为用户存储器的8192个14位字以及其他用于存储用户ID信息、配置字和中断向量的字组成。闪存程序存储器为以下内容提供存储单元：

- 用户程序指令
- 用户定义的数据

可通过以下方式读和/或写闪存程序存储器：

- CPU取指（只读）
- FSR/INDF间接访问（只读）
（第11.3节“FSR和INDF访问”）
- NVMREG访问（第11.4节“NVMREG访问”）
- 外部器件编程器

读操作返回存储器的单个字。对行完成写和擦除操作时，行大小在表11-1中定义。闪存程序存储器将擦除为逻辑1并编程为逻辑0。

表11-1: 各款器件闪存构成

器件	行擦除 (字)	写锁存 (字)
PIC16(L)F18324	32	32
PIC16(L)F18344		

要进行擦除和编程操作，了解闪存程序存储器结构非常重要。闪存程序存储器按行进行处理。每一行都包含32个14位程序存储字。行是可以通过用户软件擦除的最小大小。

可对一行的全部或部分内容进行编程。要写入程序存储器行的数据将写入14位宽的数据写锁存器中。用户不能直接访问这些锁存器，但可以通过连续写入NVMDATH:NVMDATL寄存器对来装入数据。

注： 如果只修改先前已编程行的一部分内容，则必须在擦除之前先读取整行内容，并保存到RAM或该行的写锁存器中。然后，可以将新数据和已保存数据写入写锁存器，以对闪存程序存储器行进行再编程。对于任何未经过编程的存储单元，则无需先擦除行即可写入。这种情况下，不需要保存并重新写入其他先前已编程的存储单元。

11.1.1 程序存储器电压

在整个VDD范围内的正常工作期间，闪存程序存储器都是可读写的。

11.1.1.1 外部编程

程序存储器单元和控制逻辑支持最小器件工作电压进行写和批量擦除操作。

11.1.1.2 自编程

程序存储器单元和控制逻辑支持在整个VDD范围内进行写和行擦除操作。进行自编程时，不支持批量擦除。

PIC16(L)F18324/18344

11.2 数据EEPROM

数据EEPROM由256字节的用户数据存储单元组成。EEPROM为8位用户定义的数据提供存储单元。

EEPROM可通过以下方式读和/或写：

- FSR/INDF间接访问（第11.3节“FSR和INDF访问”）
- NVMREG访问（第11.4节“NVMREG访问”）
- 外部器件编程器

与必须按行写入的闪存程序存储器不同，EEPROM可逐字节写入。

11.3 FSR和INDF访问

FSR和INDF寄存器允许间接访问闪存程序存储器或EEPROM。

11.3.1 FSR读

MOVIW指令或读取INDF操作将目标地址装入FSR寄存器，从而读取闪存程序存储器或EEPROM中的数据。读NVM操作需要一个指令周期。CPU操作在读操作期间暂停，并在完成之后立即恢复。读操作返回存储器的单个字。当FSR的MSB（例如：FSR_{xH}）设置为0x70时，低8位地址值（在FSR_{xL}中）用于确定可（通过INDF寄存器）读取的EEPROM存储单元。换句话说，EEPROM地址范围0x00-0xFF映射到0x7000-0x70FF之间的FSR地址空间。对EEPROM的写操作无法通过FSR/INDF接口实现。

11.3.2 FSR写

PIC16(L)F18324/18344器件中不支持通过FSR寄存器写/擦除NVM（例如MOVWI指令）。

11.4 NVMREG 访问

NVMREG 接口允许对可通过FSR访问的所有存储单元以及用户ID存储单元和EEPROM进行读/写访问，而只能对器件标识、版本和配置数据进行读访问。

当器件被代码保护时，阻止通过NVMREG接口读、写或擦除NVM。

11.4.1 NVMREG 读操作

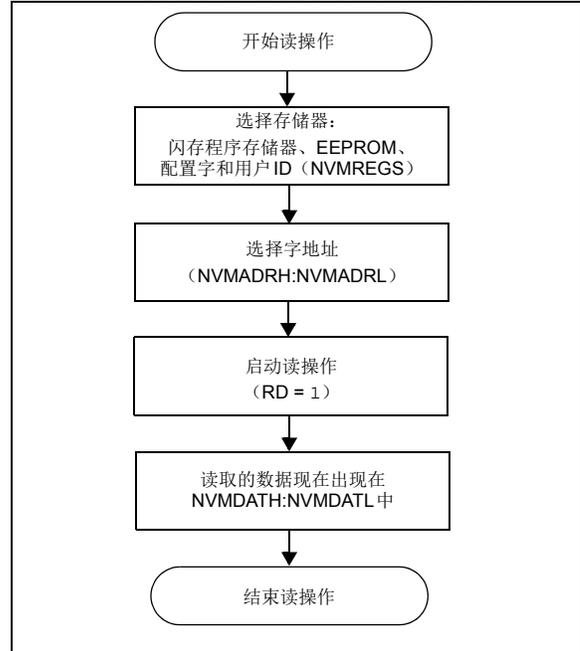
要使用NVMREG接口读NVM存储单元，用户必须：

1. 如果用户打算访问闪存程序存储器存储单元，则清零NVMCON1寄存器的NVMREGS位，或者如果用户打算访问用户ID、配置或EEPROM存储单元，则将NVMREGS置1。
2. 将所需地址写入NVMADRH:NVMADRL寄存器对（表11-2）。
3. 将NVMCON1寄存器的RD位置1启动读操作。

将读控制位置1后，CPU操作在读操作期间暂停，并在完成操作后立即恢复。在紧接着的下一个周期，数据出现在NVMDATH:NVMDATL寄存器对中；因此，可在随后的指令中读取为两个字节。

NVMDATH:NVMDATL寄存器对将保存该值直到用户进行另一次读操作或写入新值为止。完成时，RD位由硬件清零。

图11-1: 闪存程序存储器读操作流程



例11-1: 读闪存程序存储器

```

* This code block will read 1 word of program
* memory at the memory address:
  PROG_ADDR_HI: PROG_ADDR_LO
* data will be returned in the variables:
*  PROG_DATA_HI, PROG_DATA_LO

  BANKSEL  NVMADRL          ; Select Bank for NVMCON registers
  MOVLW   PROG_ADDR_LO     ;
  MOVWF   NVMADRL          ; Store LSB of address
  MOVLW   PROG_ADDR_HI     ;
  MOVWF   NVMADRH          ; Store MSB of address

  BCF     NVMCON1,NVMREGS  ; Do not select Configuration Space
  BSF     NVMCON1,RD       ; Initiate read

  MOVF    NVMDATL,W        ; Get LSB of word
  MOVWF   PROG_DATA_LO     ; Store in user location
  MOVF    NVMDATH,W        ; Get MSB of word
  MOVWF   PROG_DATA_HI     ; Store in user location
    
```

PIC16(L)F18324/18344

11.4.2 NVM解锁序列

解锁序列是一种用于保护NVM免于发生意外自写编程或擦除的机制。只有在无中断情况下执行并完成序列时，才能成功地完成以下操作之一：

- 闪存程序存储器擦除
- 向闪存程序存储器写锁存器装入数据
- 将闪存程序存储器写锁存器内容写入闪存程序存储器
- 将闪存程序存储器写锁存器内容写入用户ID
- 写入EEPROM

解锁序列由以下步骤组成且必须按如下顺序完成：

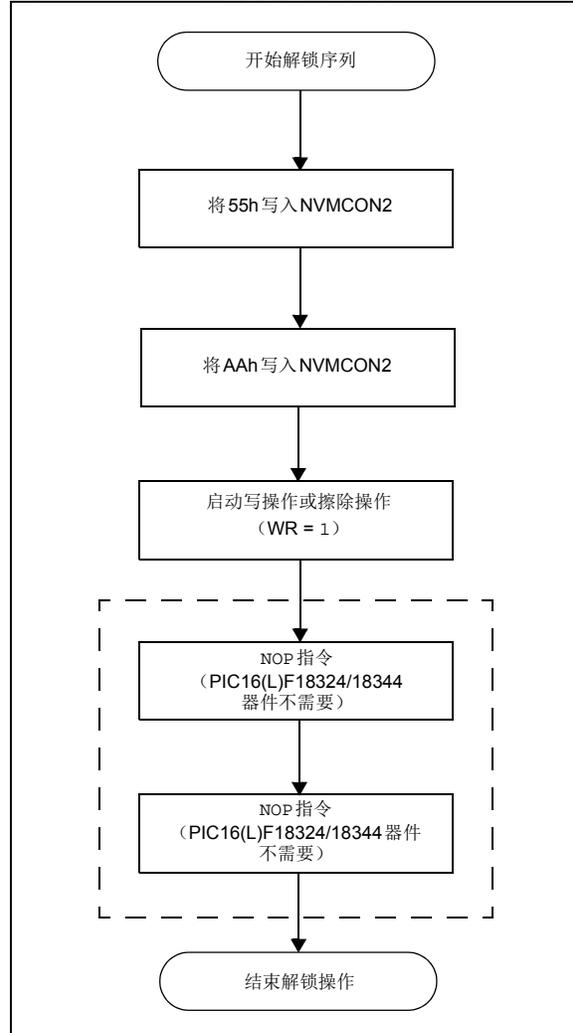
- 将55h写入NVMCON2
- 将AAh写入NVMCON2
- 将NVMCON1中的WR位置1

在WR位置1后，处理器会暂停内部操作，直到操作完成为止，然后再继续执行下一条指令。

注： 先前器件将WR位置1后需要两条NOP指令，但PIC16(L)F18324/18344器件不需要。请参见图11-2。

由于在执行解锁序列的过程中不能发生中断，所以在执行解锁序列之前应先禁止全局中断，然后在完成解锁序列之后重新允许中断。

图11-2: NVM解锁序列流程图



例11-2: NVM解锁序列

```
BANKSEL      NVMCON1
BSF          NVMCON1,WREN      ; Enable write/erase
MOVLW       55h                ; Load 55h
BCF         INTCON,GIE        ; Recommended so sequence is not interrupted

MOVWF       NVMCON2           ; Step 1: Load 55h into NVMCON2
MOVLW       AAh                ; Step 2: Load W with AAh
MOVWF       NVMCON2           ; Step 3: Load AAh into NVMCON2
BSF         NVMCON1,WR        ; Step 4: Set WR bit to begin write/erase
BSF         INTCON,GIE        ; Re-enable interrupts
```

注 1： 解锁序列开始于写NVMCON2；必须按所示的精确周期顺序执行步骤1-4。

2： 所示代码仅用作说明；任何具有所指示作用的指令均可使用。

11.4.3 NVMREG写入EEPROM

写入EEPROM通过以下步骤完成:

1. 将NVMCON1寄存器的NVMREGS和WREN位置1。
2. 将所需地址(地址+7000h)写入NVMADRH:NVMADRL寄存器对(表11-2)。
3. 按第11.4.2节“NVM解锁序列”所述执行解锁序列。

使用NVMDATA写单个EEPROM字节。操作包括该字节的隐式擦除周期(不必将FREE位置1),并且需要许多指令周期完成。CPU继续执行与之并行进行,并且当完成时,WR由硬件清零,NVMIF置1,并且当NVMIE也置1时将发生中断。软件必须查询WR位来确定写操作何时完成,或等待中断发生。WREN将保持不变。

开始EEPROM写操作后,清零WR位将不起作用;该操作将运行到完成。

11.4.4 NVMREG擦除闪存程序存储器

闪存程序存储器每次只能擦除一行。在启动对闪存程序存储器进行写操作时,并不会发生自动擦除操作。

要擦除闪存程序存储器行:

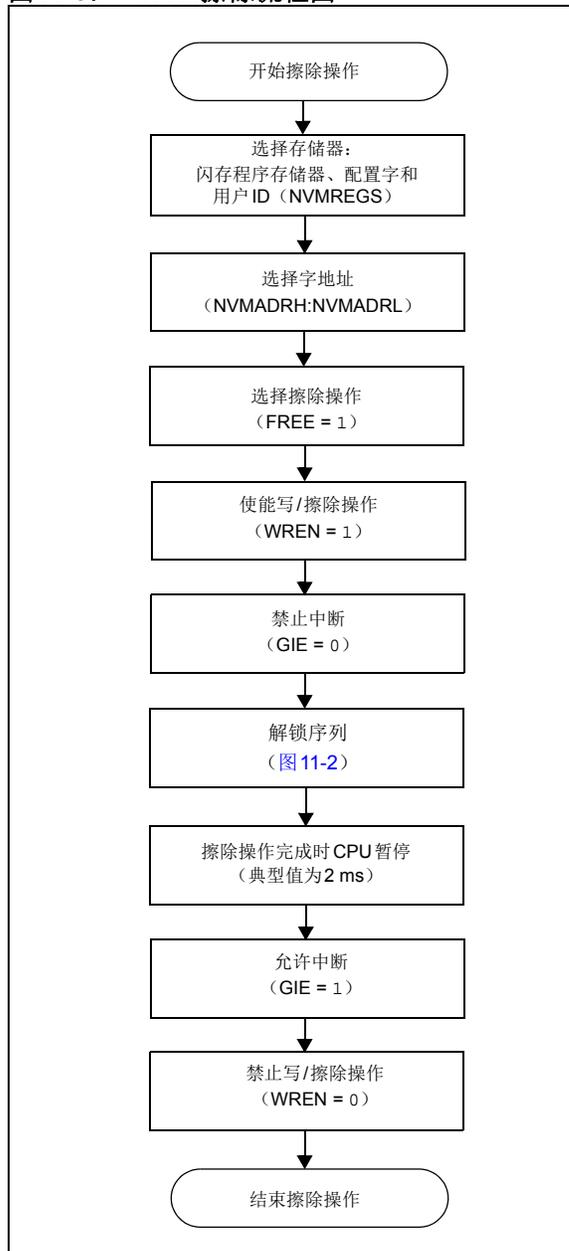
1. 清零NVMCON1寄存器的NVMREGS位以擦除闪存程序存储器存储单元或者将NVMREGS位置1以擦除用户ID单元。
2. 将所需地址写入NVMADRH:NVMADRL寄存器对(表11-2)。
3. 将NVMCON1寄存器的FREE和WREN位置1。
4. 按第11.4.2节“NVM解锁序列”所述执行解锁序列。

如果闪存程序存储器地址受写保护,则清零WR位且不会执行擦除操作。

擦除闪存程序存储器时,CPU操作暂停,并在操作完成时恢复。操作完成时,NVMIF置1,并且在NVMIE位也置1时将发生中断。

写锁存器数据不受擦除操作影响,且WREN将保持不变。

图11-3: NVM擦除流程图



PIC16(L)F18324/18344

例 11-3: 擦除闪存程序存储器的一行

```

; This sample row erase routine assumes the following:
; 1.A valid address within the erase row is loaded in variables ADDRH:ADDRL
; 2.ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)

BANKSEL      NVMADRL
MOVF         ADDRL,W
MOVWF       NVMADRL           ; Load lower 8 bits of erase address boundary
MOVF        ADDRH,W
MOVWF       NVMADRH         ; Load upper 6 bits of erase address boundary
BCF         NVMCON1,NVMREGS  ; Choose Program Flash Memory area
BSF         NVMCON1,FREE     ; Specify an erase operation
BSF         NVMCON1,WREN     ; Enable writes
BCF         INTCON,GIE       ; Disable interrupts during unlock sequence

; -----REQUIRED UNLOCK SEQUENCE:-----

MOVLW      55h               ; Load 55h to get ready for unlock sequence
MOVWF     NVMCON2           ; First step is to load 55h into NVMCON2
MOVLW     AAh               ; Second step is to load AAh into W
MOVWF     NVMCON2           ; Third step is to load AAh into NVMCON2
BSF      NVMCON1,WR         ; Final step is to set WR bit

; -----

BSF      INTCON,GIE         ; Re-enable interrupts, erase is complete
BCF      NVMCON1,WREN      ; Disable writes
    
```

表 11-2: NVM构成和访问信息

主值			NVMREG 访问			FSR 访问			
存储器功能	程序计数器 (PC), ICSP™ 地址	存储器类型	NVMREGS 位 (NVMCON1)	NVMADR <14:0>	允许的操作	FSR 地址	FSR 编程地址		
复位向量	0000h	闪存程序存储器	0	0000h	读写	8000h	只读		
用户存储器	0001h		0	0001h		8001h			
	0003h		0	0003h		8003h			
INT 向量	0004h		0	0004h		8004h			
用户存储器	0005h		0	0005h		8005h			
	17FFh	0	17FFh	FFFFh					
用户 ID	无 PC 地址	闪存程序存储器	1	0000h	读	无访问			
保留		—	—	0004h	—				
版本 ID		闪存程序存储器	1	0005h	读				
器件 ID			1	0006h					
CONFIG1			1	0007h					
CONFIG2			1	0008h					
CONFIG3			1	0009h					
CONFIG4			000Ah						
用户存储器		EEPROM	1	7000h	读			7000h	只读
				70FFh	写			70FFh	

11.4.5 NVMREG写入闪存程序存储器

要编程程序存储器，请执行以下步骤：

1. 将要编程的行的地址装入NVMADRH:NVMADRL。
2. 向每个写锁存器中装入数据。
3. 启动编程操作。
4. 重复步骤1至3，直到写入所有数据为止。

在写入程序存储器之前，要写入的字必须已擦除或先前未写入。程序存储器每次只能擦除一行。在启动写操作时，并不会发生自动擦除操作。

程序存储器每次可以写入一个或多个字。每次可以写入的最多字数等于写锁存器的数量。更多详细信息，请参见图11-4（对带32个写锁存器的程序存储器进行行写操作）。

写锁存器将对齐到由NVMADRH:NVMADR高10位（NVMADRH<6:0>:NVMADR<7:5>）定义的闪存行地址边界处，NVMADR的低5位（NVMADR<4:0>）将决定要装入的写锁存器。写操作不会跨越这些边界。在程序存储器写操作完成时，写锁存器中的数据会复位为包含0x3FFF。

要装入写锁存器并对程序存储器的一行进行编程，需要完成以下步骤。这些步骤分为两个部分。首先，在LWLO = 1的情况下，使用解锁序列将来自NVMDATH:NVMDATL的数据装入每个写锁存器。当要装入写锁存器的最后一个字就绪时，清零LWLO位并执行解锁序列。这将启动编程操作，将所有锁存器内容写入闪存程序存储器。

注： 要向写锁存器装入数据或启动闪存编程操作，需要执行一个特殊的解锁序列。如果在执行解锁序列的过程中发生中断，则不会启动对锁存器或程序存储器的写操作。

1. 将NVMCON1寄存器的WREN位置1。
2. 将NVMCON1寄存器的NVMREGS位清零。
3. 将NVMCON1寄存器的LWLO位置1。当NVMCON1寄存器的LWLO位为1时，写操作之后只会向写锁存器装入数据，而不会启动对闪存程序存储器的写操作。
4. 将要写入的存储单元的地址装入NVMADRH:NVMADRL寄存器对。
5. 将要写入的程序存储器数据装入NVMDATH:NVMDATL寄存器对。
6. 执行解锁序列（第11.4.2节“NVM解锁序列”）。此时，将数据装入写锁存器。
7. 递增NVMADRH:NVMADRL寄存器对，使之指向下一个存储单元。
8. 重复步骤5至步骤7，直到除了最后一个写锁存器之外的所有写锁存器中都装入数据为止。
9. 将NVMCON1寄存器的LWLO位清零。当NVMCON1寄存器的LWLO位为0时，写序列会启动对闪存程序存储器的写操作。
10. 将要写入的程序存储器数据装入NVMDATH:NVMDATL寄存器对。
11. 执行解锁序列（第11.4.2节“NVM解锁序列”）。整个程序存储器锁存器的内容现在会被写入闪存程序存储器中。

注： 在每个写操作或擦除操作完成时，程序存储器写锁存器将复位为空白状态（0x3FFF）。因此，不需要向所有程序存储器写锁存器中装入数据。未装入的锁存器将保持空白状态。

例11-4给出了一个完整写序列的示例。初始地址装入NVMADRH:NVMADRL寄存器对；数据使用间接寻址方式装入。

图11-4: 对带32个写锁寄存器的闪存程序存储器 (PFM) 进行块写操作

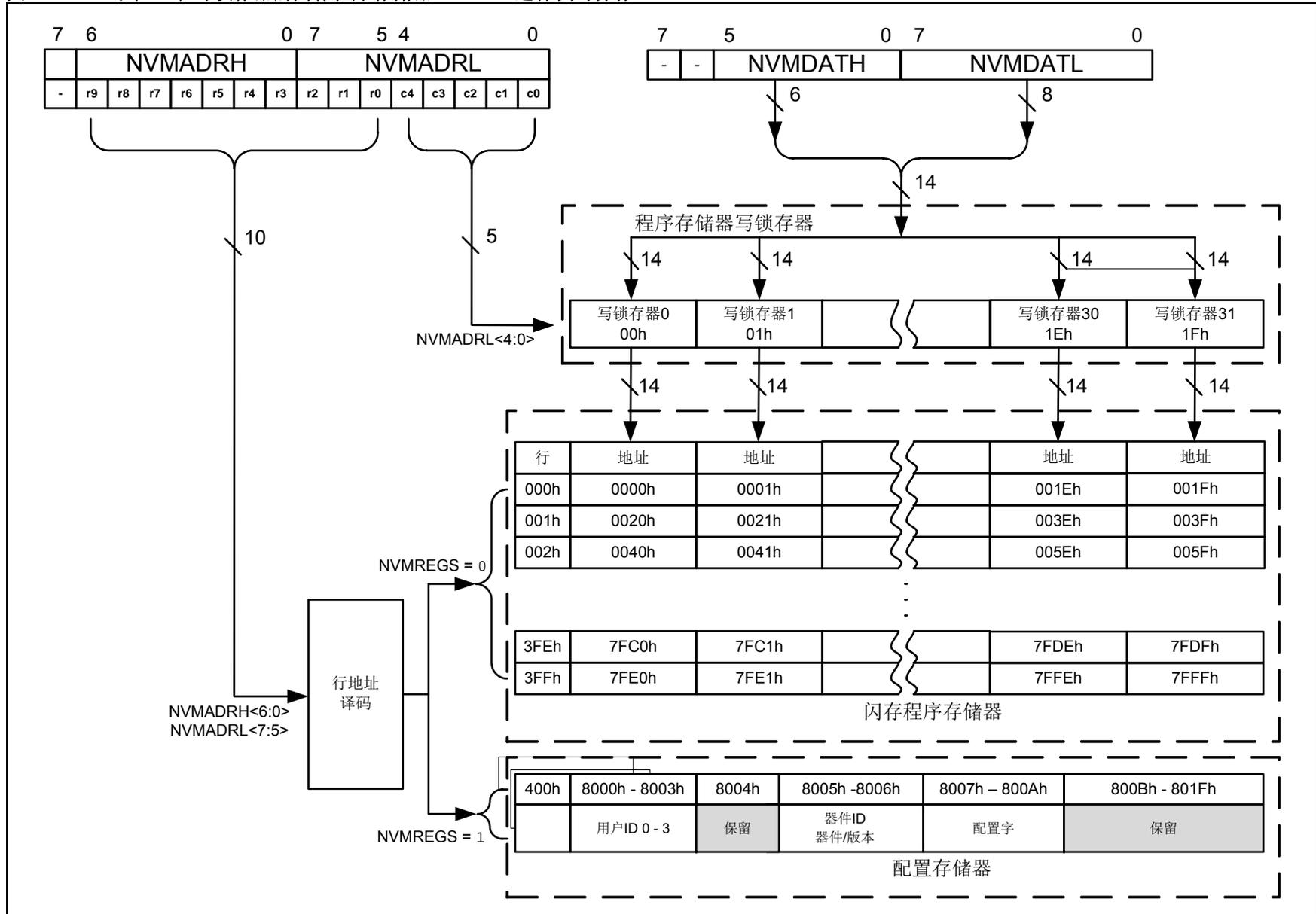
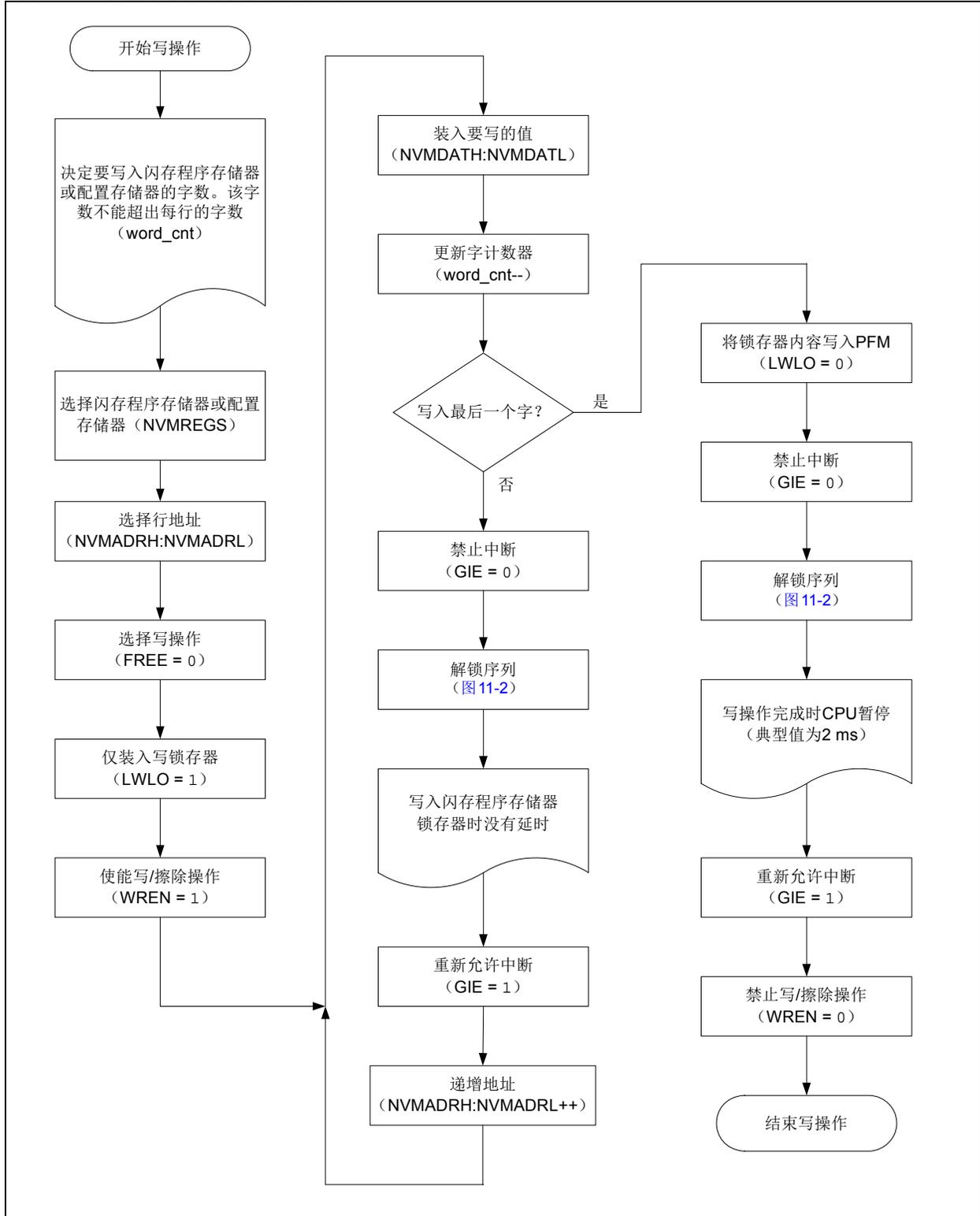


图11-5: 闪存程序存储器写流程图



PIC16(L)F18324/18344

例11-4: 写入闪存程序存储器

```
; This write routine assumes the following:
; 1. 32 words of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

    BANKSEL    NVMADRH
    MOVF       ADDRH,W
    MOVWF     NVMADRH           ; Load initial address
    MOVF       ADDRHL,W
    MOVWF     NVMADRL
    MOVLW     LOW DATA_ADDR    ; Load initial data address
    MOVWF     FSR0L
    MOVLW     HIGH DATA_ADDR
    MOVWF     FSR0H
    BCF       NVMCON1,NVMREGS   ; Set Program Flash Memory as write location
    BSF       NVMCON1,WREN      ; Enable writes
    BSF       NVMCON1,LWLO      ; Load only write latches

LOOP
    MOVIW     FSR0++
    MOVWF     NVMDATL           ; Load first data byte
    MOVIW     FSR0++
    MOVWF     NVMDATH           ; Load second data byte

    MOVF       NVMADRL,W
    XORLW     0x1F              ; Check if lower bits of address are 00000
    ANDLW     0x1F              ; and if on last of 32 addresses
    BTFSC     STATUS,Z          ; Last of 32 words?
    GOTO      START_WRITE      ; If so, go write latches into memory

    CALL      UNLOCK_SEQ        ; If not, go load latch
    INCF      NVMADRL,F         ; Increment address
    GOTO      LOOP

START_WRITE
    BCF       NVMCON1,LWLO      ; Latch writes complete, now write memory
    CALL      UNLOCK_SEQ        ; Perform required unlock sequence
    BCF       NVMCON1,LWLO      ; Disable writes

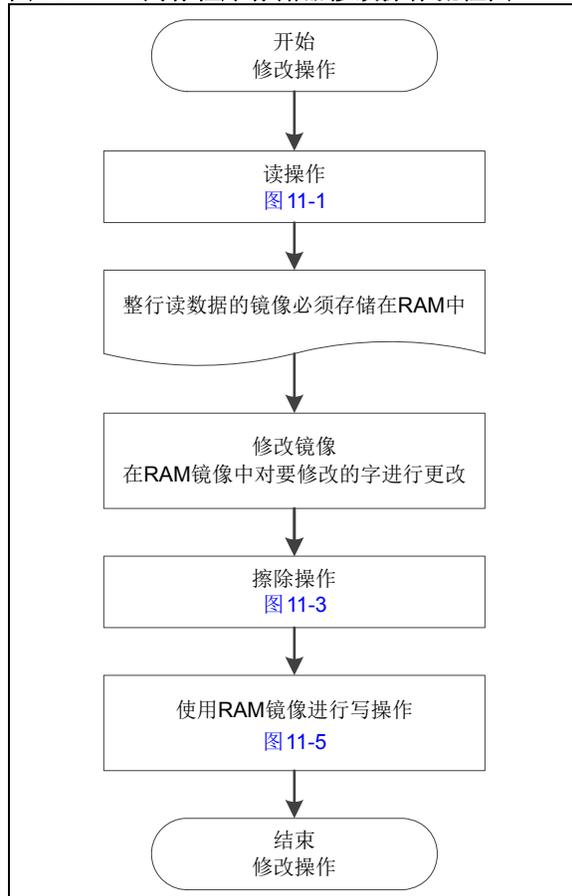
UNLOCK_SEQ
    MOVLW     55h
    BCF       INTCON,GIE        ; Disable interrupts
    MOVWF     NVMCON2           ; Begin unlock sequence
    MOVLW     AAh
    MOVWF     NVMCON2
    BSF       NVMCON1,WR
    BSF       INTCON,GIE        ; Unlock sequence complete, re-enable interrupts
    return
```

11.4.6 修改闪存程序存储器

当要修改程序存储器中某行的已有数据，并且又必须保留该行中的其他数据时，必须先读取数据并将数据保存到RAM镜像中。要修改程序存储器，请执行以下步骤：

1. 装入要修改的行的起始地址。
2. 将行中的已有数据读取到RAM镜像中。
3. 修改RAM镜像，使之包含要写入程序存储器的新数据。
4. 装入要重新写入的行的起始地址。
5. 擦除程序存储器行。
6. 将数据从RAM镜像装入写锁存器中。
7. 启动编程操作。

图 11-6: 闪存程序存储器修改操作流程图



PIC16(L)F18324/18344

11.4.7 NVMREG EEPROM、用户ID、器件ID和配置字访问

当NVMCON1寄存器中的NVMREGS = 1时，用户可以访问EEPROM、用户ID、器件ID/版本ID和配置字，而不是访问闪存程序存储器。这是在PC<15> = 1时指向的区域，但并不是所有地址都可以访问。可能存在不同的读写访问权限。请参见表11-3。

对表11-3列出的参数以外的地址进行读访问时，NVMDATH:NVMDATL寄存器对将被清零，读回0。

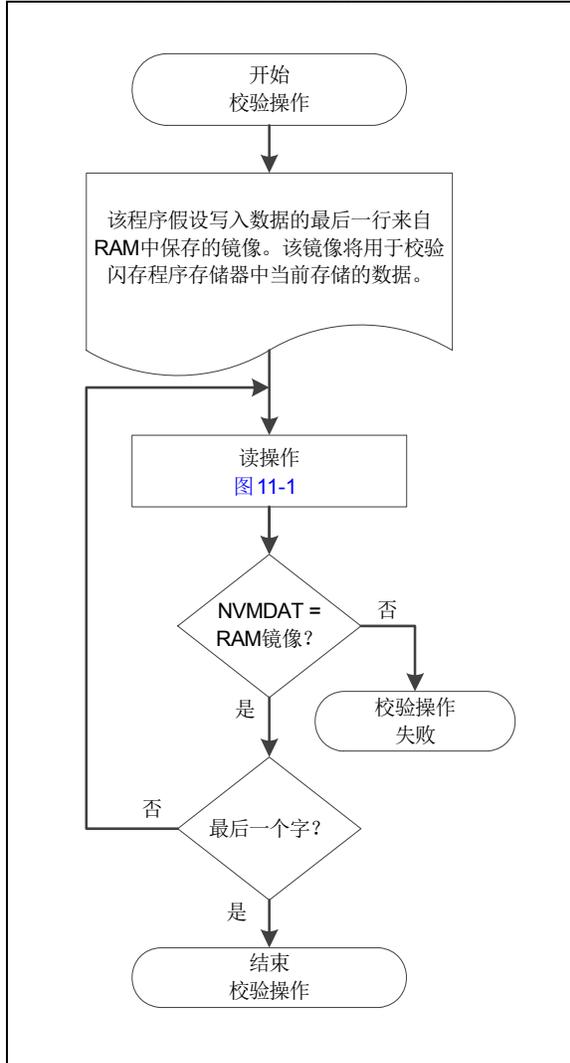
表11-3: EEPROM、用户ID、器件/版本ID和配置字访问 (NVMREGS = 1)

地址	功能	读访问	写访问
8000h-8003h	用户ID	是	是
8005h-8006h	器件ID/版本ID	是	否
8007h-800Ah	配置字1-4	是	否
F000h-F0FFh	EEPROM	是	是

11.4.8 写校验

校验程序存储器写入数据是否与预期值一致是一种良好的编程习惯。由于程序存储器以整行形式存储，因此所存储的程序存储器内容将在最后一次写操作完成之后与RAM中存储的预期数据进行比较。

图11-7: 闪存程序存储器校验操作流程



PIC16(L)F18324/18344

11.4.9 WRERR位

WRERR位可用于确定是否发生写错误。

如果发生以下其中一种情况，WRERR将置1：

- 如果在NVMADRH:NMVADRL指向写保护地址时WR置1
- 当正在进行自写操作时发生复位
- 解锁序列中断

WRERR位通常由硬件置1，但可由用户因测试目的而置1。WRERR置1后，必须用软件清零。

表11-4: WR = 1时的闪存程序存储器操作

自由	LWLO	WR = 1时的闪存程序存储器操作	备注
0	0	将写锁存器数据写入闪存程序存储器行。请参见第11.4.4节“NVMREG擦除闪存程序存储器”	<ul style="list-style-type: none">• 如果使能WP，则WR清零且WRERR置1• 写锁存器复位为3FFh• 忽略NVMDATH:NVMDATL
0	1	将NVMDATH:NVMDATL的内容复制到对应于NVMADR LSB的写锁存器。请参见第11.4.4节“NVMREG擦除闪存程序存储器”	<ul style="list-style-type: none">• 忽略写保护• 未发生存储器访问
1	x	擦除NVMADRH:NMVADRL存储单元的32字行。请参见第11.4.3节“NVMREG写入EEPROM”	<ul style="list-style-type: none">• 如果使能WP，则WR清零且WRERR置1• 擦除所有32字• 忽略NVMDATH:NVMDATL

11.5 寄存器定义：闪存程序存储器控制

寄存器 11-1: NVMDATL: 非易失性存储器数据低字节寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NVMDAT<7:0>							
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
 u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
 1 = 置1 0 = 清零

bit 7-0 **NVMDAT<7:0>**: 程序存储器低8位的读/写值

寄存器 11-2: NVMDATH: 非易失性存储器数据高字节寄存器

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—		NVMDAT<13:8>					
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
 u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
 1 = 置1 0 = 清零

bit 7-6 **未实现**: 读为0

bit 5-0 **NVMDAT<13:8>**: 程序存储器高6位的读/写值⁽¹⁾

注 1: 当写入EEPROM时, 该字节被忽略。

寄存器 11-3: NVMDRDL: 非易失性存储器地址低字节寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NVMDR<7:0>							
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
 u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
 1 = 置1 0 = 清零

bit 7-0 **NVMDR<7:0>**: 指定程序存储器地址的低8位

寄存器 11-4: NVMDRHL: 非易失性存储器地址高字节寄存器

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	NVMDR<14:8>						
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
 u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
 1 = 置1 0 = 清零

bit 7 **未实现**: 读为1

bit 6-0 **NVMDR<14:8>**: 指定程序存储器地址的高7位

PIC16(L)F18324/18344

寄存器 11-5: NVMCON1: 非易失性存储器控制1寄存器

U-0	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HS-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—	NVMREGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

图注:	q = 复位值由硬件确定	HS = 硬件置1位
R = 可读位	W = 可写位	U = 未实现位, 读为0
S = 只可置1位	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	HC = 硬件清零位

- bit 7 **未实现:** 读为0
- bit 6 **NVMREGS:** 配置选择位
1 = 访问EEPROM、配置、用户ID和器件ID寄存器
0 = 访问闪存程序存储器
- bit 5 **LWLO:** 仅装载写锁存器位
当FREE = 0时:
1 = 下一条WR命令更新行内该字的写锁存器; 不启动存储器操作。
0 = 下一条WR命令写数据或执行擦除操作。
其他情况: 该位被忽略。
- bit 4 **FREE:** 闪存程序存储器擦除使能位
当NVMREGS:NVMADR指向闪存程序存储器的存储单元时:
1 = 在下一条WR命令时执行擦除操作; 擦除包含所指示地址的行(为全1)以为写操作做准备
0 = 已正常完成所有擦除操作
- bit 3 **WRERR:** 编程/擦除错误标志位(1,2,3)
1 = 写操作被复位中断(中断的解锁序列), 或在NVMADR指向写保护地址时WR被写入1。
0 = 编程或擦除操作正常完成
- bit 2 **WREN:** 编程/擦除使能位
1 = 允许编程/擦除周期
0 = 禁止对程序闪存的编程/擦除操作
- bit 1 **WR:** 写控制位(4,5,6)
当NVMREG:NVMADR指向EEPROM存储单元时:
1 = 启动相应EEPROM存储单元的擦除/编程周期
0 = 对NVM的编程/擦除操作已完成并且变为无效
当NVMREG:NVMADR指向闪存程序存储器的存储单元时:
1 = 启动表11-5所指示的操作
0 = 对NVM的编程/擦除操作已完成并且变为无效
其他情况: 该位被忽略。
- bit 0 **RD:** 读控制位(7)
1 = 启动地址 = NVMADR1的读操作并将数据装入NVMDAT。读操作需要一个指令周期并且该位在操作完成时清零。用软件只能将该位置1(不能清零)。
0 = NVM读操作已完成并且变为无效。

- 注 1:** WR = 1时该位可能发生变化(在EEPROM写操作期间, 该位可能为0或1)。
2: 该位必须由软件清零; 硬件不能将该位清零。
3: 该位可由软件写入1以实现测试序列。
4: 只能紧接着第11.4.2节“NVM解锁序列”的解锁序列将该位置1。
5: 操作是自计时的, 并且当操作完成时WR位由硬件清零。
6: 启动写操作之后, 将该位设置为0将不起作用。
7: 读EEPROM仅装入NVMDATL<7:0>(寄存器11-1)。

PIC16(L)F18324/18344

寄存器 11-6: NVMCON2: 非易失性存储器控制2寄存器

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
NVMCON2							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
S = 只可置1位	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 **NVMCON2<7:0>**: 闪存解锁模式位
 要对写操作进行解锁, 必须先写入55h, 接着写入AAh, 然后再将NVMCON1寄存器的WR位置1。写入该寄存器的值用于对写操作进行解锁。

表11-5: 与非易失性存储器 (NVM) 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	寄存器所在页
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR2	TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF	107
PIE2	TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	102
NVMCON1	—	NVMREGS	LWLO	FREE	WRERR	WREN	WR	RD	136
NVMCON2	NVMCON2								137
NVMADRL	NVMADR<7:0>								135
NVMADRH	— ⁽¹⁾	NVMADR<14:8>							135
NVMDATL	NVMDAT<7:0>								135
NVMDATH	—	—	NVMDAT<13:8>						135

图注: — = 未实现位, 读为0。NVM不使用阴影单元。

注 1: 未实现, 读为1。

表11-6: 与非易失性存储器 (NVM) 相关的配置字汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG3	13:8	—	—	LVP	—	—	—	—	—	64
	7:0	—	—	—	—	—	—	WRT<1:0>		
CONFIG4	13:8	—	—	—	—	—	—	—	—	65
	7:0	—	—	—	—	—	—	CPD	CP	

图注: — = 未实现位, 读为0。NVM不使用阴影单元。

PIC16(L)F18324/18344

12.0 I/O 端口

表12-1: 每款器件可用的端口

器件	PORTA	PORTB	PORTC
PIC16(L)F18324	•		•
PIC16(L)F18344	•	•	•

每个端口都有用于控制其操作的10个标准寄存器。这些寄存器是:

- PORTx 寄存器 (读取器件引脚的电平)
- LATx 寄存器 (输出锁存器)
- TRISx 寄存器 (数据方向)
- ANSELx 寄存器 (模拟选择)
- WPUx 寄存器 (弱上拉)
- INLVLx (输入电平控制)
- SLRCONx 寄存器 (压摆率)
- ODCONx 寄存器 (漏极开路)

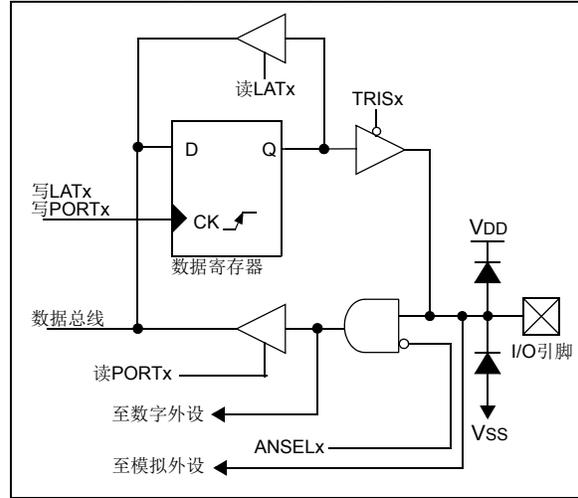
大多数端口引脚与器件模拟外设和数字外设共用功能。通常,当使能某个端口引脚上的外设时,该引脚将不能用作通用输出,但仍然可以对该引脚进行读操作。

数据锁存器(LATx寄存器)在对I/O引脚驱动值进行读-修改-写操作时非常有用。

对LATx寄存器的写操作与写入相应PORTx寄存器的效果相同。读取LATx寄存器时,将会读取I/O端口锁存器中保存的值,而读取PORTx寄存器时,将会读取实际的I/O引脚值。

支持模拟输入的端口具有相关的ANSELx寄存器。当某个ANSEL位置1时,与该位相关的数字输入缓冲器将被禁止。禁止输入缓冲器可以防止该引脚上介于逻辑高电平和低电平之间的模拟信号电压在逻辑输入电路中产生过大的电流。图12-1给出了通用I/O端口的简化模型,没有给出与其他外设的接口。

图12-1: 通用I/O端口的工作原理



12.1 I/O 优先级

在复位之后,每个引脚均默认设为端口数据锁存器。其他功能通过外设引脚选择逻辑选择。更多信息,请参见第13.0节“外设引脚选择(PPS)模块”。

外设引脚选择列表中未列出模拟输入功能,例如ADC和比较器输入。这些输入在使用ANSELx寄存器将I/O引脚设置为模拟模式时有效。当引脚处于模拟模式时,数字输出功能可以继续控制引脚。

当使能模拟输出时,模拟输出优先于数字输出且强制数字输出驱动器为高阻态。

12.2 PORTA 寄存器

12.2.1 数据寄存器

PORTA是一个6位宽的双向端口。对应的数据方向寄存器是TRISA（寄存器12-2）。将TRISA某位置1（=1）时，会将PORTA的相应引脚设为输入（即，禁止输出驱动器）。将TRISA某位清零（=0）时，会将PORTA的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选定的引脚）。RA3是个例外，仅可作为输入引脚，其TRIS位总是读为1。例12-1显示了如何初始化PORTA。

读PORTA寄存器（寄存器12-1）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读-修改-写操作。因此，对端口的写操作意味着先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器（LATA）。

端口数据锁存器LATA（寄存器12-3）保存输出端口数据，并包含写入LATA或PORTA的最新值。

例12-1： 初始化PORTA

```

; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL PORTA      ;
CLRFB PORTA        ;Clear PORTA
BANKSEL LATA        ;Data Latch
CLRFB LATA          ;
BANKSEL ANSELA     ;
CLRFB ANSELA       ;digital I/O
BANKSEL TRISA      ;
MOVLW B'00111000' ;Set RA<5:3> as inputs
MOVWF TRISA        ;and set RA<2:0> as
                  ;outputs
    
```

12.2.2 方向控制

TRISA寄存器（寄存器12-2）用于控制PORTA引脚输出驱动器，即使它们被用作模拟输入。当引脚用于模拟输入时，用户应确保TRISA寄存器中的相应位保持置1。配置为模拟输入的I/O引脚总是读为0。

12.2.3 漏极开路控制

ODCONA寄存器（寄存器12-6）用于控制端口的漏极开路功能。每个引脚的漏极开路操作可以独立进行选择。当ODCONA位置1时，相应的端口输出会变为只能灌入电流的漏极开路驱动器。当ODCONA位清零时，相应的端口输出引脚是能够拉出和灌入电流的标准推挽驱动器。

注： 当将引脚用于I²C功能时，不需要设置漏极开路控制；I²C模块控制该引脚，使引脚漏极开路。

12.2.4 压摆率控制

SLRCONA寄存器（寄存器12-7）用于控制每个端口引脚的压摆率选项。每个端口引脚的压摆率控制可以独立进行选择。当SLRCONA位置1时，相应端口引脚驱动器的压摆率会受到限制。当SLRCONA位清零时，相应端口引脚驱动器的压摆率将为最大可能值。

12.2.5 输入阈值控制

INLVLA寄存器（寄存器12-8）用于控制每个可用PORTA输入引脚的输入电压阈值。用户可以选择施密特触发器CMOS阈值或TTL兼容阈值。输入阈值对于确定PORTA寄存器的读取值很重要，同时它也是发生电平变化中断的电压（如果使能该功能）。关于阈值电压的更多信息，请参见表35-4。

注： 如果要更改所选择的输入阈值，则应先禁止所有外设模块再执行该操作。在模块处于活动状态时更改阈值电压，可能会意外产生与输入引脚相关联的电平变化，不论该引脚上的实际电压如何。

PIC16(L)F18324/18344

12.2.6 模拟控制

ANSELA 寄存器（寄存器 12-4）用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSELA 位设置为高电平，将使引脚上的所有数字读操作都读为 0，并允许引脚上的模拟功能正常工作。

ANSELA 位的状态不会影响数字输出功能。TRIS 清零且 ANSEL 置 1 的引脚将仍作为数字输出工作，但输入模式将变为模拟。当在受影响的端口上执行读-修改-写指令时，引脚行为可能与预期不符。

注： 在发生复位之后，ANSELA 位默认设为模拟模式。要将任意引脚用作数字通用输入或外设输入，必须通过用户软件将相应的 ANSEL 位初始化为 0。

12.2.7 弱上拉控制

WPUA 寄存器（寄存器 12-5）用于控制每个端口引脚的各个弱上拉功能。

PORTA 引脚 RA3 包括 $\overline{\text{MCLR}}/\text{VPP}$ 输入。 $\overline{\text{MCLR}}$ 输入可使器件复位，也可以通过配置字 2 的 MCLRE 位来禁止。弱上拉功能位于 RA3 端口引脚上。当使能 MCLR（MCLRE = 1）或 WPUA3 位置 1 时，使能弱上拉。当禁止 MCLR 或 WPUA3 位清零时，禁止弱上拉。

12.2.8 PORTA 功能和输出优先级

每个 PORTA 引脚都与其他功能复用。

在复位之后，每个引脚均默认设为 PORT 锁存器数据。其他输出功能通过外设引脚选择逻辑来选择。更多信息，请参见第 13.0 节“外设引脚选择 (PPS) 模块”。

外设引脚选择列表中未列出模拟输入功能，例如 ADC 和比较器输入。当引脚处于模拟模式时，数字输出功能可以继续控制引脚。

12.3 寄存器定义: PORTA

寄存器 12-1: **PORTA: PORTA 寄存器**

U-0	U-0	R/W-x/u	R/W-x/u	R-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	RA5	RA4	RA3 ⁽²⁾	RA2	RA1	RA0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 **未实现:** 读为0

bit 5-0 **RA<5:0>: PORTA I/O 值位⁽¹⁾**

1 = 端口引脚电压 ≥ V_{IH}

0 = 端口引脚电压 ≤ V_{IL}

注 1: 写入PORTA时, 实际上会写入相应的LATA寄存器。读取PORTA寄存器时, 将返回实际的I/O引脚值。

2: RA3位为只读位且在MCLRE = 1 (使能主复位) 时读为1。

寄存器 12-2: **TRISA: PORTA 三态寄存器**

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 **未实现:** 读为0

bit 5-4 **TRISA<5:4>: PORTA 三态控制位**

1 = PORTA 引脚被配置为输入 (三态)

0 = PORTA 引脚被配置为输出

bit 3 **未实现:** 读为1

bit 2-0 **TRISA<2:0>: PORTA 三态控制位**

1 = PORTA 引脚被配置为输入 (三态)

0 = PORTA 引脚被配置为输出

PIC16(L)F18324/18344

寄存器 12-3: LATA: PORTA 数据锁存器寄存器

U-0	U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

- bit 7-6 **未实现:** 读为0
- bit 5-4 **LATA<5:4>:** RC<5:4> 输出锁存值位⁽¹⁾
- bit 3 **未实现:** 读为0
- bit 2-0 **LATA<2:0>:** RC<2:0> 输出锁存值位⁽¹⁾

注 1: 写入PORTA时, 实际上会写入相应的LATA寄存器。读取PORTA寄存器时, 将返回实际的I/O引脚值。

寄存器 12-4: ANSELA: PORTA 模拟选择寄存器

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

- bit 7-6 **未实现:** 读为0
- bit 5-4 **ANSA<5:4>:** 将RA<5:4> 引脚选择为模拟或数字功能
1 = 模拟输入。引脚被配置为模拟输入⁽¹⁾。数字输入缓冲器被禁止。
0 = 数字I/O。引脚被配置为端口或数字特殊功能。
- bit 3 **未实现:** 读为0
- bit 2-0 **ANSA<2:0>:** 将RA<2:0> 引脚选择为模拟或数字功能
1 = 模拟输入。引脚被配置为模拟输入⁽¹⁾。数字输入缓冲器被禁止。
0 = 数字I/O。引脚被配置为端口或数字特殊功能。

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

寄存器 12-5: WPUA: 弱上拉PORTA寄存器

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	WPUA5	WPUA4	WPUA3 ⁽¹⁾	WPUA2	WPUA1	WPUA0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 未实现: 读为0

bit 5-0 **WPUA<5:0>**: 弱上拉寄存器位⁽²⁾
 1 = 使能上拉
 0 = 禁止上拉

注 1: 如果MCLRE = 1, 则RA3的弱上拉始终使能; WPUA3位不受影响。

2: 如果引脚被配置为输出, 则禁止弱上拉, 除非引脚也被配置为漏极开路。当引脚被配置为漏极开路时, 当输出值为高电平时使能上拉, 当输出值位低电平时禁止上拉。

寄存器 12-6: ODCONA: PORTA漏极开路控制寄存器

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ODCA5	ODCA4	—	ODCA2	ODCA1	ODCA0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 未实现: 读为0

bit 5-4 **ODCA<5:4>**: PORTA漏极开路使能位
 对于各个RA<5:4>引脚
 1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)
 0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

bit 3 未实现: 读为0

bit 2-0 **ODCA<2:0>**: PORTA漏极开路使能位
 对于各个RA<2:0>引脚
 1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)
 0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

PIC16(L)F18324/18344

寄存器 12-7: SLRCONA: PORTA 压摆率控制寄存器

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-4 **SLRA<5:4>**: PORTA 压摆率使能位

对于各个RA<5:4> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率为最大值

bit 3 未实现: 读为0

bit 2-0 **SLRA<2:0>**: PORTA 压摆率使能位

对于各个RA<2:0> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率为最大值

寄存器 12-8: INLVLA: PORTA 输入电平控制寄存器

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-0 **INLVLA<5:0>**: PORTA 输入电平选择位

对于各个RA<5:0> 引脚

1 = 对于端口读操作和电平变化中断, 使用ST输入

0 = 对于端口读操作和电平变化中断, 使用TTL输入

表12-2: 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	寄存器所在页
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	141
TRISA	—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	141
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	142
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	143
ODCONA	—	—	ODCA5	ODCA4	—	ODCA2	ODCA1	ODCA0	143
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	144
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	144

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

表12-3: 与PORTA相关的配置字汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG2	13:8	—	—	$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	—	BORV	—	63
	7:0	BOREN1	BOREN0	$\overline{\text{LPBOREN}}$	—	WDTE1	WDTE0	$\overline{\text{PWRTÉ}}$	MCLRE	

图注: — = 未实现位, 读为0。PORTA不使用阴影单元。

PIC16(L)F18324/18344

12.4 PORTB 寄存器（仅限 PIC16(L)F18344）

12.4.1 数据寄存器

PORTB 是一个 4 位宽的双向端口，仅在 PIC16(L)F18344 器件中可用。对应的数据方向寄存器是 TRISB（寄存器 12-10）。将 TRISB 某位置 1（= 1）时，会将 PORTB 的相应引脚设为输入（即，使相应的输出驱动器呈高阻态）。将 TRISB 某位清零（= 0）时，会将 PORTB 的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选定的引脚）。例 12-1 显示了如何初始化 I/O 端口。

读 PORTB 寄存器（寄存器 12-9）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读-修改-写操作。因此，对端口的写操作意味着先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器（LATB）。

端口数据锁存器 LATB（寄存器 12-11）存放输出端口数据，包含写入 LATB 或 PORTB 的最新值。

12.4.2 方向控制

TRISB 寄存器（寄存器 12-10）用于控制 PORTB 引脚输出驱动器，即使它们被用作模拟输入。当引脚用于模拟输入时，用户应确保 TRISB 寄存器中的相应位保持置 1。配置为模拟输入的 I/O 引脚总是读为 0。

12.4.3 输入阈值控制

INLVLB 寄存器（寄存器 12-16）用于控制每个可用 PORTB 输入引脚的输入电压阈值。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。输入阈值对于确定 PORTB 寄存器的读取值很重要，同时它也是发生电平变化中断的电压（如果使能该功能）。关于阈值电压的更多信息，请参见表 35-4。

注： 如果要更改所选择的输入阈值，则应先禁止所有外设模块再执行该操作。在模块处于活动状态时更改阈值电压，可能会意外产生与输入引脚相关联的电平变化，不论该引脚上的实际电压如何。

12.4.4 漏极开路控制

ODCONB 寄存器（寄存器 12-14）用于控制端口的漏极开路功能。每个引脚的漏极开路操作可以独立进行选择。当 ODCONB 位置 1 时，相应的端口输出会变为只能灌入电流的漏极开路驱动器。当 ODCONB 位清零时，相应的端口输出引脚是能够拉出和灌入电流的标准推挽驱动器。

注： 当将引脚用于 I²C 功能时，不需要设置漏极开路控制；I²C 模块控制该引脚，使引脚漏极开路。

12.4.5 压摆率控制

SLRCONB 寄存器（寄存器 12-15）用于控制每个端口引脚的压摆率选项。每个端口引脚的压摆率控制可以独立进行选择。当 SLRCONB 位置 1 时，相应端口引脚驱动器的压摆率会受到限制。当 SLRCONB 位清零时，相应端口引脚驱动器的压摆率将为最大可能值。

12.4.6 模拟控制

ANSELB 寄存器（寄存器 12-12）用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSELB 位设置为高电平，将使引脚上的所有数字读操作都读为 0，并允许引脚上的模拟功能正常工作。

ANSELB 位的状态不会影响数字输出功能。TRIS 清零且 ANSELB 置 1 的引脚仍将仍作为数字输出工作，但输入模式将变为模拟。当在受影响的端口上执行读-修改-写指令时，引脚行为可能与预期不符。

注： 在发生复位之后，ANSELB 位默认设为模拟模式。要将任意引脚用作数字通用输入或外设输入，必须通过用户软件将相应的 ANSEL 位初始化为 0。

12.4.7 弱上拉控制

WPUB 寄存器（寄存器 12-13）用于控制每个端口引脚的弱上拉功能。

12.4.8 PORTB 功能和输出优先级

在复位之后，每个引脚均默认设为端口锁存器数据。其他输出功能通过外设引脚选择逻辑来选择。更多信息，请参见第 13.0 节“外设引脚选择（PPS）模块”。

外设引脚选择列表中未列出模拟输入功能，例如 ADC 和比较器输入。当引脚处于模拟模式时，数字输出功能可以继续控制引脚。

12.5 寄存器定义: PORTB

寄存器 12-9: PORTB: PORTB寄存器

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
bit 7				bit 0			

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-4 **RB<7:4>: PORTB I/O 值位⁽¹⁾**

- 1 = 端口引脚电压 $\geq V_{IH}$
- 0 = 端口引脚电压 $\leq V_{IL}$

bit 3-0 **未实现: 读为0**

注 1: 写入PORTB时, 实际上会写入相应的LATB寄存器。读取PORTB寄存器时, 将返回实际的I/O引脚值。

寄存器 12-10: TRISB: PORTB三态寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
bit 7				bit 0			

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-4 **TRISB<7:4>: PORTB I/O 三态控制位**

- 0 = 使能PORTB输出驱动器
- 1 = 禁止PORTB输出驱动器

bit 3-0 **未实现: 读为0**

PIC16(L)F18324/18344

寄存器 12-11: LATB: PORTB数据锁存器寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **LATB<7:4>**: RB<5:4>输出锁存值位⁽¹⁾

bit 3-0 **未实现**: 读为0

注 1: 写入LATB等效于写入相应的PORTB寄存器。读取LATB寄存器将返回寄存器值, 而不是I/O引脚值。

寄存器 12-12: ANSELB: PORTB模拟选择寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **ANSB<7:4>**: 选择为模拟或数字功能

0 = 数字I/O。引脚被配置为端口或数字特殊功能。

1 = 模拟输入。引脚被配置为模拟输入⁽¹⁾。数字输入缓冲器被禁止。

bit 3-0 **未实现**: 读为0

注 1: 设置ANSB[n] = 1将禁止数字输入电路。弱上拉(如果可用)不受影响。用户必须将相应的TRIS位设置为输入模式, 以允许从外部控制引脚电压。

PIC16(L)F18324/18344

寄存器 12-13: WPUB: 弱上拉PORTB寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

bit 7-4 **WPUB<7:4>**: 弱上拉寄存器位

1 = 使能弱上拉
0 = 禁止弱上拉

bit 3-0 **未实现**: 读为0

注: 如果引脚被配置为输出, 则禁止弱上拉, 除非引脚也被配置为漏极开路。当引脚被配置为漏极开路时, 当输出值为高电平时使能上拉, 当输出值位低电平时禁止上拉。

寄存器 12-14: ODCONB: PORTB漏极开路控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

bit 7-4 **ODCB<7:4>**: PORTB漏极开路配置位

对于各个RB<7:4>引脚:
1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)
0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

bit 3-0 **未实现**: 读为0

PIC16(L)F18324/18344

寄存器 12-15: SLRCONB: PORTB 压摆率控制寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **SLRB<7:4>**: 各个RB<7:4>引脚的PORTB压摆率控制
0 = 禁止压摆率
1 = 使能压摆率

bit 3-0 **未实现**: 读为0

寄存器 12-16: INLVLB: PORTB 输入电平控制寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **INLVLB<7:4>**: 各个RB<7:4>引脚的PORTB输入电平选择
0 = 对于端口读操作, 使用TTL输入
1 = 对于端口读操作, 使用ST输入

bit 3-0 **未实现**: 读为0

表12-4: 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	寄存器所在页
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	147
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	148
ANSELB	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	149
ODCONB	ODCB7	ODCB6	ODCB5	ODCB4	—	—	—	—	149
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	150
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	150

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

PIC16(L)F18324/18344

12.6 PORTC 寄存器

12.6.1 数据寄存器

PORTC 是一个 6 位宽 (PIC16(L)F18324) 或 8 位宽 (PIC16(L)F18344) 的双向端口。对应的数据方向寄存器是 TRISC (寄存器 12-18)。将 TRISC 某位置 1 (= 1) 时, 会将 PORTC 的相应引脚设为输入 (即, 使相应的输出驱动器呈高阻态)。将 TRISC 某位清零 (= 0) 时, 会将 PORTC 的相应引脚设为输出 (即, 使能输出驱动器并将输出锁存器中的内容输出到选定的引脚)。例 12-1 显示了如何初始化 I/O 端口。

读 PORTC 寄存器 (寄存器 12-17) 将读出相应引脚的状态, 而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读-修改-写操作。因此, 对端口的写操作意味着先读端口引脚电平状态, 然后修改这个值, 最后再写入该端口的数据锁存器 (LATC)。

端口数据锁存器 LATC (寄存器 12-19) 存放输出端口数据, 包含写入 LATC 或 PORTC 的最新值。

12.6.2 方向控制

TRISC 寄存器 (寄存器 12-18) 用于控制 PORTC 引脚输出驱动器, 即使它们被用作模拟输入。当引脚用于模拟输入时, 用户应确保 TRISC 寄存器中的相应位保持置 1。配置为模拟输入的 I/O 引脚总是读为 0。

12.6.3 输入阈值控制

INLVLC 寄存器 (寄存器 12-24) 用于控制每个可用 PORTC 输入引脚的输入阈值电压。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。输入阈值对于确定 PORTC 寄存器的读取值很重要, 同时它也是发生电平变化中断的电压 (如果使能该功能)。关于阈值电压的更多信息, 请参见表 35-4。

注: 如果要更改所选择的输入阈值, 则应先禁止所有外设模块再执行该操作。在模块处于活动状态时更改阈值电压, 可能会意外产生与输入引脚相关联的电平变化, 不论该引脚上的实际电压如何。

12.6.4 漏极开路控制

ODCONC 寄存器 (寄存器 12-22) 用于控制端口的漏极开路功能。每个引脚的漏极开路操作可以独立进行选择。当 ODCONC 位置 1 时, 相应的端口输出会变为只能灌入电流的漏极开路驱动器。当 ODCONC 位清零时, 相应的端口输出引脚是能够拉出和灌入电流的标准推挽驱动器。

注: 当将引脚用于 I²C 功能时, 不需要设置漏极开路控制; I²C 模块控制该引脚, 使引脚漏极开路。

12.6.5 压摆率控制

SLRCONC 寄存器 (寄存器 12-23) 用于控制每个端口引脚的压摆率选项。每个端口引脚的压摆率控制可以独立进行选择。当 SLRCONC 位置 1 时, 相应端口引脚驱动器的压摆率会受到限制。当 SLRCONC 位清零时, 相应端口引脚驱动器的压摆率将为最大可能值。

12.6.6 模拟控制

ANSEL 寄存器 (寄存器 12-20) 用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSEL 位设置为高电平, 将使引脚上的所有数字读操作都读为 0, 并允许引脚上的模拟功能正常工作。

ANSEL 位的状态不会影响数字输出功能。TRIS 清零且 ANSEL 置 1 的引脚将仍作为数字输出工作, 但输入模式将变为模拟。当在受影响的端口上执行读-修改-写指令时, 引脚行为可能与预期不符。

注: 在发生复位之后, ANSEL 位默认设为模拟模式。要将任意引脚用作数字通用输入或外设输入, 必须通过用户软件将相应的 ANSEL 位初始化为 0。

12.6.7 弱上拉控制

WPUC 寄存器 (寄存器 12-21) 用于控制每个端口引脚的弱上拉。

12.6.8 PORTC 功能和输出优先级

在复位之后, 每个引脚均默认设为端口锁存器数据。其他功能通过外设引脚选择逻辑选择。更多信息, 请参见第 13.0 节“外设引脚选择 (PPS) 模块”。

外设引脚选择列表中未列出模拟输出功能, 例如 ADC 和比较器输入。当引脚处于模拟模式时, 数字输出功能可以继续控制引脚。

12.7 寄存器定义: PORTC

寄存器 12-17: PORTC: PORTC 寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 **RC<7:6>**: PORTC I/O 值位^(1,2)

1 = 端口引脚电压 $\geq V_{IH}$

0 = 端口引脚电压 $\leq V_{IL}$

bit 5-0 **RC<5:0>**: PORTC 通用 I/O 引脚位⁽²⁾

1 = 端口引脚电压 $\geq V_{IH}$

0 = 端口引脚电压 $\leq V_{IL}$

注 1: 仅限 PIC16(L)F18344; 否则读为 0。

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

PIC16(L)F18324/18344

寄存器 12-18: TRISC: PORTC三态寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7-6 **TRISC<7:6>**: PORTC三态控制位⁽¹⁾
 1 = PORTC引脚配置为输入 (三态)
 0 = PORTC引脚配置为输出
- bit 5-0 **TRISC<5:0>**: PORTC三态控制位
 1 = PORTC引脚配置为输入 (三态)
 0 = PORTC引脚配置为输出

注 1: 仅限PIC16(L)F18344; 否则读为0。

寄存器 12-19: LATC: PORTC数据锁存器寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7-6 **LATC<7:6>**: PORTC输出锁存值位⁽¹⁾
- bit 5-0 **LATC<5:0>**: PORTC输出锁存值位

注 1: 仅限PIC16(L)F18344; 否则读为0。

PIC16(L)F18324/18344

寄存器12-20: ANSELC: PORTC模拟选择寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7-6 **ANSC<7:6>**: 将RC<7:6>引脚分别选择为模拟或数字功能⁽¹⁾
 0 = 数字I/O。引脚被配置为端口或数字特殊功能。
 1 = 模拟输入。引脚被配置为模拟输入⁽²⁾。数字输入缓冲器被禁止。
- bit 5-0 **ANSC<5:0>**: 将RC<5:0>引脚分别选择为模拟或数字功能
 0 = 数字I/O。引脚被配置为端口或数字特殊功能。
 1 = 模拟输入。引脚被配置为模拟输入⁽²⁾。数字输入缓冲器被禁止。

注 1: 仅限PIC16(L)F18344; 否则读为0。

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

寄存器12-21: WPUC: 弱上拉PORTC寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7-6 **WPUC<7:6>⁽¹⁾**: 弱上拉寄存器位⁽²⁾
 1 = 使能上拉
 0 = 禁止上拉
- bit 5-0 **WPUC<5:0>**: 弱上拉寄存器位⁽²⁾
 1 = 使能上拉
 0 = 禁止上拉

注 1: 仅限PIC16(L)F18344; 否则读为0。

2: 如果引脚被配置为输出, 则禁止弱上拉, 除非引脚也被配置为漏极开路。当引脚被配置为漏极开路时, 当输出值为高电平时使能上拉, 当输出值低电平时禁止上拉。

PIC16(L)F18324/18344

寄存器 12-22: ODCONC: PORTC漏极开路控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ODCC7 ⁽¹⁾	ODCC6 ⁽¹⁾	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 ODCC<7:6>: PORTC漏极开路使能位⁽¹⁾

对于各个RC<7:6>引脚

1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)

0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

bit 5-0 ODCC<5:0>: PORTC漏极开路使能位

对于各个RC<5:0>引脚

1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)

0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

注 1: 仅限PIC16(L)F18344; 否则读为0。

寄存器 12-23: SLRCONC: PORTC压摆率控制寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 SLRC<7:6>: PORTC压摆率使能位⁽¹⁾

对于各个RC<7:6>引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率为最大值

bit 5-0 SLRC<5:0>: PORTC压摆率使能位

对于各个RC<5:0>引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率为最大值

注 1: 仅限PIC16(L)F18344; 否则读为0。

PIC16(L)F18324/18344

寄存器 12-24: INLVLC: PORTC输入电平控制寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 **INLVLC<7:6>**: PORTC输入电平选择位⁽¹⁾
 对于各个RC<7:6>引脚
 1 = 对于端口读操作和电平变化中断, 使用ST输入
 0 = 对于端口读操作和电平变化中断, 使用TTL输入

bit 5-0 **INLVLC<5:0>**: PORTC输入电平选择位
 对于各个RC<5:0>引脚
 1 = 对于端口读操作和电平变化中断, 使用ST输入
 0 = 对于端口读操作和电平变化中断, 使用TTL输入

注 1: 仅限PIC16(L)F18344; 否则读为0。

表 12-5: 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	寄存器所在页
PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	153
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
LATC	LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	154
ANSEL	ANSEL7 ⁽¹⁾	ANSEL6 ⁽¹⁾	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0	155
WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	155
ODCONC	ODCC7 ⁽¹⁾	ODCC6 ⁽¹⁾	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	156
SLRCONC	SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	156
INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	157

图注: — = 未实现位, 读为0。PORTC不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

PIC16(L)F18324/18344

13.0 外设引脚选择 (PPS) 模块

外设引脚选择 (PPS) 模块用于将外设输入和输出与器件 I/O 引脚连接。选择范围仅包含数字信号。所有模拟输入和输出均固定连接至它们所分配的引脚。输入和输出选择是独立的，如图 13-1 所示。

13.1 PPS 输入

每个外设均具有一个用于选择外设输入的 PPS 寄存器。输入包括器件引脚。

虽然每个外设均具有自己的 PPS 输入选择寄存器，但每个外设的选择是相同的，如寄存器 13-1 所示。

注： 寄存器名称中的符号“xxx”是外设标识符的占位符。例如，CLC1PPS。

13.2 PPS 输出

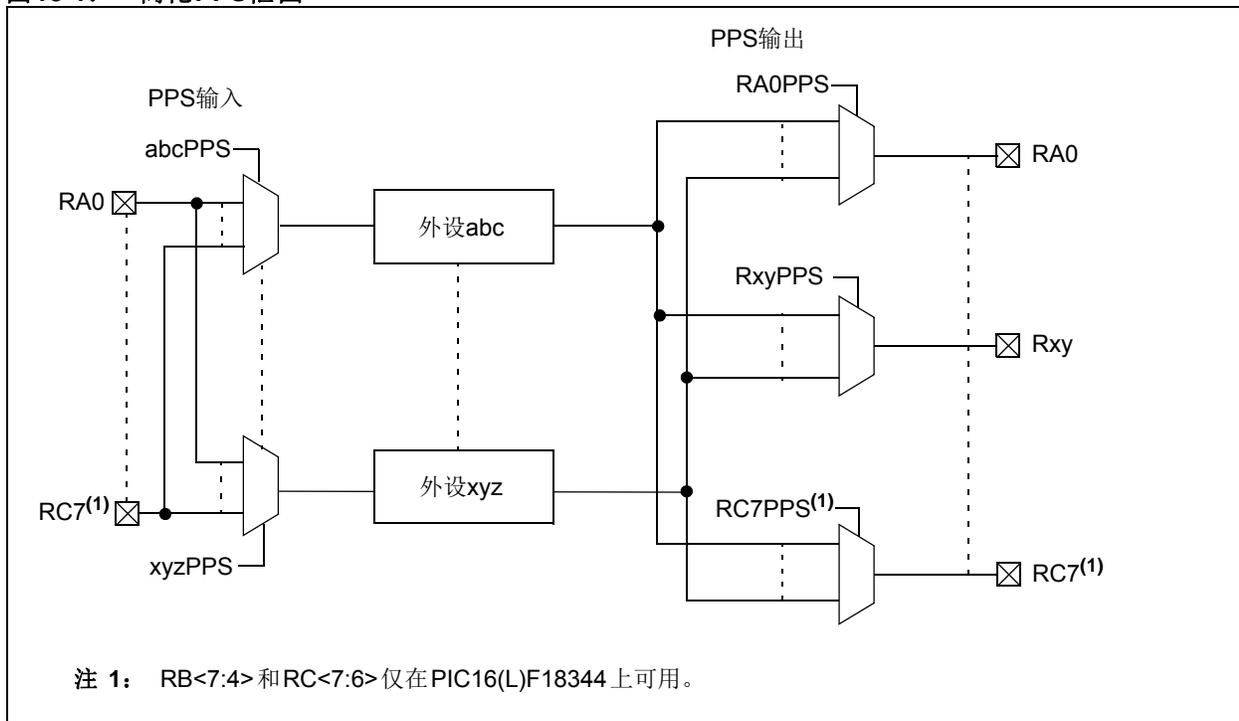
每个 I/O 引脚均具有一个用于选择引脚输出源的 PPS 寄存器。除了少数例外情况，与该引脚相关的端口 TRIS 控制将保持对引脚输出驱动器的控制权。作为外设操作的一部分，控制引脚输出驱动器的外设将根据需要改写 TRIS 控制。这些外设包括：

- EUSART1 (同步操作)
- MSSP (I²C)

虽然每个引脚均具有自己的 PPS 外设选择寄存器，但每个引脚的选择是相同的，如寄存器 13-2 所示。

注： 符号“Rxy”是引脚标识符的占位符。例如，RA0PPS。

图 13-1: 简化 PPS 框图



13.3 双向引脚

对于在单个引脚上具有双向信号的外设，在进行PPS选择时必须使PPS输入和PPS输出选择同一引脚。具有双向信号的外设包括：

- EUSART1（同步操作）
- MSSP（I²C）

注： I²C默认输入引脚与I²C和SMBus兼容，并且是PIC16(L)F18324/18344上具有这种兼容性的仅有引脚。时钟和数据信号可输送到任何引脚，但是不具有I²C兼容性的引脚将以标准TTL/ST逻辑电平（由INVLV寄存器选择）工作。

13.4 PPSLOCKED位

PPS包含了一种模式，在该模式下可以锁定所有输入和输出选择，以防止意外的更改。PPS选择通过将PPSLOCK寄存器的PPSLOCKED位置1来进行锁定。置1和清零该位需要一个特殊序列作为额外的预防措施，以防止意外的更改。例13-1给出了置1和清零PPSLOCKED位的示例。

例 13-1： PPS锁定/解锁序列

```
; suspend interrupts
    bcf    INTCON,GIE
; BANKSEL PPSLOCK    ; set bank
; required sequence, next 5 instructions
    movlw 0x55
    movwf PPSLOCK
    movlw 0xAA
    movwf PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
    bsf    PPSLOCK,PPSLOCKED
; restore interrupts
    bsf    INTCON,GIE
```

13.5 PPS1WAY位

通过将配置字2的PPS1WAY位置1，可以锁定PPS。

当PPS1WAY位置1时，PPSLOCK寄存器的PPSLOCKED位只能在器件复位之后清零和置1一次。PPS寄存器完成配置后，用户软件会将PPSLOCKED位置1，以防对PPS寄存器执行任何其他写操作。无论PPS1WAY或PPSLOCKED的设置如何，均可随时读取PPS寄存器。

当 PPS1WAY 位清零时，PPSLOCK 寄存器的 PPSLOCKED 位可在代码执行过程中清零和置1多次，但每次修改PPS寄存器时都需要执行PPS锁定/解锁序列。

13.6 休眠期间的操作

PPS输入和输出选择不受休眠影响。

13.7 复位的影响

器件上电复位（POR）会将所有PPS输入和输出选择清除为其默认值，并将PPSLOCK寄存器的PPSLOCKED位清零。所有其他复位会将选择保留不变。引脚分配表1和表2给出了默认的输入选择。

PIC16(L)F18324/18344

13.8 寄存器定义：PPS输入选择

寄存器 13-1: **xxxPPS**: 外设xxx输入选择

U-0	U-0	U-0	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u
—	—	—	xxxPPS<4:0>				
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于外设

bit 7-5

未实现: 读为0

bit 4-0

xxxPPS<4:0>: 外设xxx输入选择位
11xxx = 保留。不要使用。

10111 = 外设输入为RC7⁽¹⁾

10110 = 外设输入为RC6⁽¹⁾

10101 = 外设输入为RC5

10100 = 外设输入为RC4

10011 = 外设输入为RC3

10010 = 外设输入为RC2

10001 = 外设输入为RC1

10000 = 外设输入为RC0

...

01111 = 外设输入为RB7⁽¹⁾

01110 = 外设输入为RB6⁽¹⁾

01101 = 外设输入为RB5⁽¹⁾

01100 = 外设输入为RB4⁽¹⁾

...

0011x = 保留。不要使用。

00101 = 外设输入为RA5

00100 = 外设输入为RA4

00011 = 外设输入为RA3

00010 = 外设输入为RA2

00001 = 外设输入为RA1

00000 = 外设输入为RA0

注 1: 仅限PIC16(L)F18344。

寄存器 13-2: RxyPPS: 引脚Rxy输出源选择寄存器

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	
—	—	—	RxyPPS<4:0>					
bit 7								bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-5 **未实现:** 读为0

bit 4-0 **RxyPPS<4:0>:** 引脚Rxy输出源选择位

- 11111 = Rxy源为DSM
- 11110 = Rxy源为CLKR
- 11101 = Rxy源为NCO1
- 11100 = Rxy源为TMR0
- 11011 = 保留
- 11010 = 保留
- 11001 = Rxy源为SDO1/SDA1
- 11000 = Rxy源为SCK1/SCL1⁽¹⁾
- 10111 = Rxy源为C2
- 10110 = Rxy源为C1
- 10101 = Rxy源为DT⁽¹⁾
- 10100 = Rxy源为TX/CK⁽¹⁾
- 10011 = Rxy源为CWG2D⁽¹⁾
- 10010 = Rxy源为CWG2C⁽¹⁾
- 10001 = Rxy源为CWG2B⁽¹⁾
- 10000 = Rxy源为CWG2A⁽¹⁾
- 01111 = Rxy源为CCP4
- 01110 = Rxy源为CCP3
- 01101 = Rxy源为CCP2
- 01100 = Rxy源为CCP1
- 01011 = Rxy源为CWG1D⁽¹⁾
- 01010 = Rxy源为CWG1C⁽¹⁾
- 01001 = Rxy源为CWG1B⁽¹⁾
- 01000 = Rxy源为CWG1A⁽¹⁾
- 00111 = Rxy源为CLC4OUT
- 00110 = Rxy源为CLC3OUT
- 00101 = Rxy源为CLC2OUT
- 00100 = Rxy源为CLC1OUT
- 00011 = Rxy源为PWM6
- 00010 = Rxy源为PWM5
- 00001 = 保留
- 00000 = 保留

注 1: TRIS控制由外设根据需要改写。

PIC16(L)F18324/18344

寄存器 13-3: PPSLOCK: PPS 锁定寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PPSLOCKED
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置1

0 = 清零

bit 7-1 **未实现:** 读为0

bit 0 **PPSLOCKED:** PPS 锁定位

1 = PPS 已锁定。不能更改 PPS 选择。

0 = PPS 未锁定。可以更改 PPS 选择。

表13-1: 与 PPS 模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	162
INTPPS	—	—	—	INTPPS<4:0>					160
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					160
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					160
T1GPPS	—	—	—	T1GPPS<4:0>					160
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					160
T3GPPS	—	—	—	T3GPPS<4:0>					160
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					160
T5GPPS	—	—	—	T5GPPS<4:0>					160
CCP1PPS	—	—	—	CCP1PPS<4:0>					160
CCP2PPS	—	—	—	CCP2PPS<4:0>					160
CCP3PPS	—	—	—	CCP3PPS<4:0>					160
CCP4PPS	—	—	—	CCP4PPS<4:0>					160
CWG1PPS	—	—	—	CWG1PPS<4:0>					160
CWG2PPS	—	—	—	CWG2PPS<4:0>					160
MDCIN1PPS	—	—	—	MDCIN1PPS<4:0>					160
MDCIN2PPS	—	—	—	MDCIN2PPS<4:0>					160
MDMINPPS	—	—	—	MDMINPPS<4:0>					160
SSP1CLKPPS	—	—	—	SSP1CLKPPS<4:0>					160
SSP1DATPPS	—	—	—	SSP1DATPPS<4:0>					160
SSP1SSPPS	—	—	—	SSP1SSPPS<4:0>					160
RXPPS	—	—	—	RXPPS<4:0>					161
TXPPS	—	—	—	TXPPS<4:0>					160
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					160
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					160

图注: — = 未实现, 读为0。PPS 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F18344。

表13-1: 与PPS模块相关的寄存器汇总 (续)

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					160
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					160
RA0PPS	—	—	—	RA0PPS<4:0>					161
RA1PPS	—	—	—	RA1PPS<4:0>					161
RA2PPS	—	—	—	RA2PPS<4:0>					161
RA4PPS	—	—	—	RA4PPS<4:0>					161
RA5PPS	—	—	—	RA5PPS<4:0>					161
RB4PPS ⁽¹⁾	—	—	—	RB4PPS<4:0>					161
RB5PPS ⁽¹⁾	—	—	—	RB5PPS<4:0>					161
RB6PPS ⁽¹⁾	—	—	—	RB6PPS<4:0>					161
RB7PPS ⁽¹⁾	—	—	—	RB7PPS<4:0>					161
RC0PPS	—	—	—	RC0PPS<4:0>					161
RC1PPS	—	—	—	RC1PPS<4:0>					161
RC2PPS	—	—	—	RC2PPS<4:0>					161
RC3PPS	—	—	—	RC3PPS<4:0>					161
RC4PPS	—	—	—	RC4PPS<4:0>					161
RC5PPS	—	—	—	RC5PPS<4:0>					161
RC6PPS ⁽¹⁾	—	—	—	RC6PPS<4:0>					161
RC7PPS ⁽¹⁾	—	—	—	RC7PPS<4:0>					161

图注: — = 未实现, 读为0。PPS模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

PIC16(L)F18324/18344

14.0 外设模块禁止

PIC16(L)F18324/18344 具备禁止所选模块的功能，可将这些模块置于功耗尽可能低的模式。

由于传统原因，所有模块在器件复位后默认设为开启状态。

14.1 禁止模块

禁止模块有以下影响：

- 模块的所有时钟和控制输入暂停；不存在任何逻辑电平变化，模块无法正常工作。
- 模块保持在复位状态。
 - 禁止写入 SFR
 - 读操作返回 00h
- 禁止模拟输出；数字输出读为 0

14.2 使能模块

当寄存器位清零时，模块重新使能并处于复位状态；SFR 数据将反映 POR 复位值。

根据模块的不同，可能需要最多一个完整指令周期，模块才能变为活动状态。重新使能模块后，至少在一个指令周期内不要与模块有任何交互（例如，写寄存器）。

14.3 禁止模块

当禁止一个模块时，所有相关输入选择寄存器（Input Selection Register, ISM）也被禁止。

14.4 系统时钟禁止

SYSCMD（PMD0，寄存器 14-1）置 1 时，将禁止外设的系统时钟（Fosc）分布网络。并非所有的外设都使用 SYSCMD，所以不是所有外设都会受影响。请参见具体外设说明以了解其是否会受到该位的影响。

寄存器 14-1: PMD0: PMD 控制寄存器 0

R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
SYSCMD	FVRMD	—	—	—	NVMMD	CLKRMD	IOCMD
7							0

图注：

R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7 **SYSCMD**: 禁止外设系统时钟网络
请参见第 14.4 节“系统时钟禁止”中的说明。

- 1 = 禁止系统时钟网络（又称为 FOSC）
- 0 = 使能系统时钟网络

bit 6 **FVRMD**: 禁止固定参考电压 FVR 位

- 1 = 禁止 FVR 模块
- 0 = 使能 FVR 模块

bit 5-3 **未实现**: 读为 0

bit 2 **NVMMD**: NVM 模块禁止位⁽¹⁾

- 1 = 禁止读写数据 EEPROM；不能写入 NVMCON 寄存器；FSR 访问 EEPROM 将返回零。
- 0 = 使能 NVM 模块

bit 1 **CLKRMD**: 禁止参考时钟 CLKR 位

- 1 = 禁止 CLKR 模块
- 0 = 使能 CLKR 模块

bit 0 **IOCMD**: 禁止所有端口的电平变化中断位

- 1 = 禁止 IOC 模块
- 0 = 使能 IOC 模块

注 1: 使能 NVM 时，访问数据之前可能需要最长 1 μs 的延时。

寄存器 14-2: PMD1: PMD控制寄存器1

R/W-0/0							
NCOMD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

- bit 7 **NCOMD:** 禁止数控振荡器位
1 = 禁止NCO1模块
0 = 使能NCO1模块
- bit 6 **TMR6MD:** 禁止定时器TMR6位
1 = 禁止TMR6模块
0 = 使能TMR6模块
- bit 5 **TMR5MD:** 禁止定时器TMR5位
1 = 禁止TMR5模块
0 = 使能TMR5模块
- bit 4 **TMR4MD:** 禁止定时器TMR4位
1 = 禁止TMR4模块
0 = 使能TMR4模块
- bit 3 **TMR3MD:** 禁止定时器TMR3位
1 = 禁止TMR3模块
0 = 使能TMR3模块
- bit 2 **TMR2MD:** 禁止定时器TMR2位
1 = 禁止TMR2模块
0 = 使能TMR2模块
- bit 1 **TMR1MD:** 禁止定时器TMR1位
1 = 禁止TMR1模块
0 = 使能TMR1模块
- bit 0 **TMR0MD:** 禁止定时器TMR0位
1 = 禁止TMR0模块
0 = 使能TMR0模块

PIC16(L)F18324/18344

寄存器 14-3: **PMD2: PMD控制寄存器2**

U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	U-0
—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7 **未实现:** 读为0

bit 6 **DACMD:** 禁止DAC位
1 = 禁止DAC模块
0 = 使能DAC模块

bit 5 **ADCMD:** 禁止ADC位
1 = 禁止ADC模块
0 = 使能ADC模块

bit 4-3 **未实现:** 读为0

bit 2 **CMP2MD:** 禁止比较器C2位
1 = 禁止比较器C2模块
0 = 使能比较器C2模块

bit 1 **CMP1MD:** 禁止比较器C1位
1 = 禁止比较器C1模块
0 = 使能比较器C1模块

bit 0 **未实现:** 读为0

寄存器 14-4: PMD3: PMD控制寄存器3

R/W-0/0							
CWG2MD	CWG1MD	PWM6MD	PWM5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	q = 值取决于具体条件

- bit 7 **CWG2MD:** 禁止CWG2位
1 = 禁止CWG2模块
0 = 使能CWG2模块
- bit 6 **CWG1MD:** 禁止CWG1位
1 = 禁止CWG1模块
0 = 使能CWG1模块
- bit 5 **PWM6MD:** 禁止PWM6位
1 = 禁止PWM6模块
0 = 使能PWM6模块
- bit 4 **PWM5MD:** 禁止PWM5位
1 = 禁止PWM5模块
0 = 使能PWM5模块
- bit 3 **CCP4MD:** 禁止CCP4位
1 = 禁止CCP4模块
0 = 使能CCP4模块
- bit 2 **CCP3MD:** 禁止CCP3位
1 = 禁止CCP3模块
0 = 使能CCP3模块
- bit 1 **CCP2MD:** 禁止CCP2位
1 = 禁止CCP2模块
0 = 使能CCP2模块
- bit 0 **CCP1MD:** 禁止CCP1位
1 = 禁止CCP1模块
0 = 使能CCP1模块

PIC16(L)F18324/18344

寄存器 14-5: PMD4: PMD控制寄存器4

U-0	U-0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	U-0
—	—	UART1MD	—	—	—	MSSP1MD	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7-6 **未实现:** 读为0

bit 5 **UART1MD:** 禁止EUSART1位
1 = 禁止EUSART1模块
0 = 使能EUSART1模块

bit 4-2 **未实现:** 读为0

bit 1 **MSSP1MD:** 禁止MSSP1位
1 = 禁止MSSP1模块
0 = 使能MSSP1模块

bit 0 **未实现:** 读为0

PIC16(L)F18324/18344

寄存器 14-6: PMD5: PMD控制寄存器5

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	DSMMD
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

- bit 7-5 **未实现:** 读为0
- bit 4 **CLC4MD:** 禁止CLC4位
1 = 禁止CLC4模块
0 = 使能CLC4模块
- bit 3 **CLC3MD:** 禁止CLC3位
1 = 禁止CLC3模块
0 = 使能CLC3模块
- bit 2 **CLC2MD:** 禁止CLC2位
1 = 禁止CLC2模块
0 = 使能CLC2模块
- bit 1 **CLC1MD:** 禁止CLC1位
1 = 禁止CLC1模块
0 = 使能CLC1模块
- bit 0 **DSMMD:** 禁止数据信号调制器位
1 = 禁止DSM模块
0 = 使能DSM模块

PIC16(L)F18324/18344

15.0 电平变化中断

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

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

图 15-1 给出了 IOC 模块的框图。

15.1 使能模块

要允许各个引脚产生中断，PIE0 寄存器的 IOCIE 位必须置 1。如果 IOCIE 位被禁止，在引脚上仍然会发生边沿检测，但不会产生中断。

15.2 独立的引脚配置

对于每个引脚，都提供了上升沿检测器和下降沿检测器。要允许引脚检测上升沿，需要将 IOCxP 寄存器的相关位置 1。要允许引脚检测下降沿，需要将 IOCxN 寄存器的相关位置 1。

通过同时将 IOCxP 和 IOCxN 寄存器中的相关位置 1，一个引脚可以配置为同时检测上升沿和下降沿。

15.3 中断标志

位于 IOCxF 寄存器中的位是对应于每个端口的电平变化中断引脚的状态标志。如果在正确使能的引脚上检测到期望的边沿，则对应于该引脚的状态标志会置 1，并且如果 IOCIE 位也置 1，则还会产生中断。PIR0 寄存器的 IOCIF 位反映所有 IOCxF 位的状态。

15.3.1 清零中断标志

各个状态标志（IOCxF 寄存器位）可以通过将其复位为零的方式清零。如果在该清零操作期间检测到另一个边沿，则无论实际写入的值如何，相关的状态标志都会在序列结束时置 1。

为了确保在清零标志时不会丢失任何已检测到的边沿，应当仅执行可屏蔽已知更改位的逻辑与操作。以下序列是一个说明应执行何种操作的示例。

例 15-1： 清零中断标志（以 PORTA 为例）

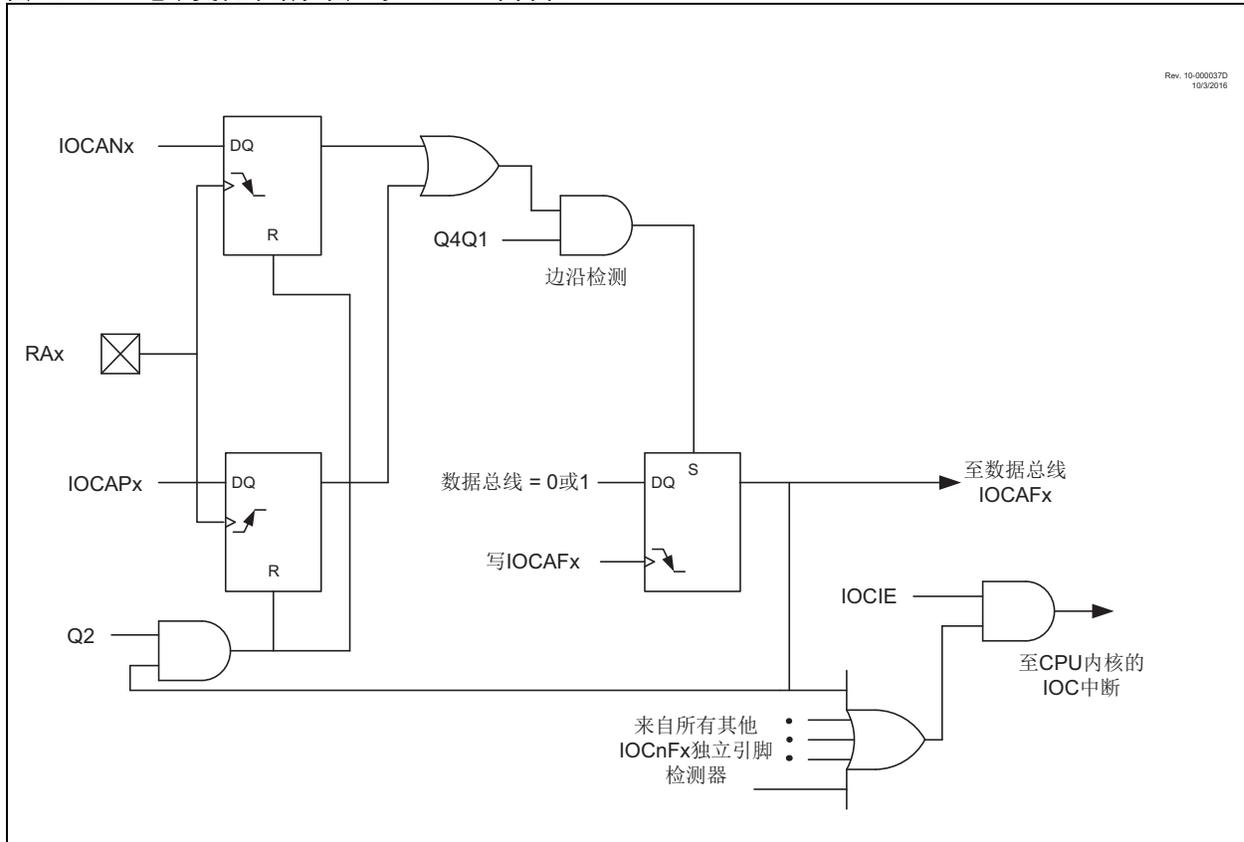
```
MOVLW  0xff
XORWF  IOCAF, W
ANDWF  IOCAF, F
```

15.4 休眠模式下的操作

如果 IOCIE 位置 1，电平变化中断事件会将器件从休眠模式唤醒。

如果在处于休眠模式时检测到边沿，则在退出休眠模式执行第一条指令之前，会先更新受影响的 IOCxF 寄存器。

图15-1: 电平变化中断框图 (以PORTA为例)



PIC16(L)F18324/18344

15.5 寄存器定义：电平变化中断控制

寄存器 15-1: **IOCAP**: 电平变化中断PORTA正边沿寄存器

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-0 **IOCAP<5:0>**: 电平变化中断PORTA正边沿允许位

1 = 在引脚上对于正向边沿允许电平变化中断。IOCAF_x位和IOCIF标志将在检测到边沿时置1。

0 = 禁止关联引脚的电平变化中断

寄存器 15-2: **IOCAN**: 电平变化中断PORTA负边沿寄存器

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-0 **IOCAN<5:0>**: 电平变化中断PORTA负边沿允许位

1 = 在引脚上对于负向边沿允许电平变化中断。IOCAF_x位和IOCIF标志将在检测到边沿时置1。

0 = 禁止关联引脚的电平变化中断

PIC16(L)F18324/18344

寄存器 15-3: IOCAF: 电平变化中断PORTA标志寄存器

U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	HS = 硬件置1位

bit 7-6 未实现: 读为0

bit 5-0 **IOCAF<5:0>**: 电平变化中断PORTA标志位

1 = 在关联引脚上检测到使能的电平变化。

在IOCAPx = 1, 并在RAX上检测到上升沿时置1, 或者在IOCANx = 1, 并在RAX上检测到下降沿时置1。

0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

寄存器 15-4: IOCBP: 电平变化中断PORTB正边沿寄存器⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-4 **IOCBP<7:4>**: 电平变化中断PORTB正边沿允许位

1 = 在引脚上对于正向边沿允许电平变化中断。IOCAFx位和IOCIF标志将在检测到边沿时置1。

0 = 禁止关联引脚的电平变化中断

bit 3-0 未实现: 读为0

注 1: 仅限PIC16(L)F18344。

PIC16(L)F18324/18344

寄存器 15-5: IOCBN: 电平变化中断PORTB负边沿寄存器⁽¹⁾

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4

IOCBN<7:4>: 电平变化中断PORTB负边沿允许位

1 = 在引脚上对于负向边沿允许电平变化中断。IOCAF_x位和IOCIF标志将在检测到边沿时置1。

0 = 禁止关联引脚的电平变化中断

bit 3-0

未实现: 读为0

注 1: 仅限PIC16(L)F18344。

寄存器 15-6: IOCBF: 电平变化中断PORTB标志寄存器⁽¹⁾

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1位

bit 7-4

IOCBF<7:4>: 电平变化中断PORTB标志位

1 = 在关联引脚上检测到使能的电平变化

在IOCBP_x = 1, 并在RB_x上检测到上升沿时置1, 或者在IOCBN_x = 1, 并在RB_x上检测到下降沿时置1。

0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

bit 3-0

未实现: 读为0

注 1: 仅限PIC16(L)F18344。

PIC16(L)F18324/18344

寄存器 15-7: IOCCP: 电平变化中断PORTC正边沿寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 **IOCCP<7:6>**: 电平变化中断PORTC正边沿允许位⁽¹⁾

1 = 在引脚上对于正向边沿允许电平变化中断。IOCCF_x位和IOCIF标志将在检测到边沿时置1。
0 = 禁止关联引脚的电平变化中断。

bit 5-0 **IOCCP<5:0>**: 电平变化中断PORTC正边沿允许位

1 = 在引脚上对于正向边沿允许电平变化中断。IOCCF_x位和IOCIF标志将在检测到边沿时置1。
0 = 禁止关联引脚的电平变化中断

注 1: 仅限PIC16(L)F18344。

寄存器 15-8: IOCCN: 电平变化中断PORTC负边沿寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 **IOCCN<7:6>**: 电平变化中断PORTC负边沿允许位⁽¹⁾

1 = 在引脚上对于负向边沿允许电平变化中断。IOCCF_x位和IOCIF标志将在检测到边沿时置1。
0 = 禁止关联引脚的电平变化中断

bit 5-0 **IOCCN<5:0>**: 电平变化中断PORTC负边沿允许位

1 = 在引脚上对于负向边沿允许电平变化中断。IOCCF_x位和IOCIF标志将在检测到边沿时置1。
0 = 禁止关联引脚的电平变化中断

注 1: 仅限PIC16(L)F18344。

PIC16(L)F18324/18344

寄存器 15-9: IOCCF: 电平变化中断PORTC标志寄存器

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

HS = 硬件置1位

bit 7-6

IOCCF<7:6>: 电平变化中断PORTC标志位

1 = 在关联引脚上检测到使能的电平变化。

在IOCCPx = 1, 并在RCx上检测到上升沿时置1, 或者在IOCCNx = 1, 并在RCx上检测到下降沿时置1。

0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

bit 5-0

IOCCF<5:0>: 电平变化中断PORTC标志位

1 = 在关联引脚上检测到使能的电平变化。

在IOCCPx = 1, 并在RCx上检测到上升沿时置1, 或者在IOCCNx = 1, 并在RCx上检测到下降沿时置1。

0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

注 1: 仅限PIC16(L)F18344。

表 15-1: 与电平变化中断相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	100
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	172
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	172
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	173
IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	173
IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	174
IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	174
IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	175
IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	175
IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	176

图注: — = 未实现位, 读为0。电平变化中断不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

16.0 固定参考电压 (FVR)

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

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

FVR 可以通过将 FVRCON 寄存器的 FVREN 位置 1 来使能。

注： 固定参考电压输出不能超出 V_{DD} 。

16.1 独立的增益放大器

送到 ADC、比较器和 DAC 的 FVR 输出会经过两个独立的可编程增益放大器。每个放大器的增益都可以设定为 1x、2x 或 4x，从而产生三种可能电压。

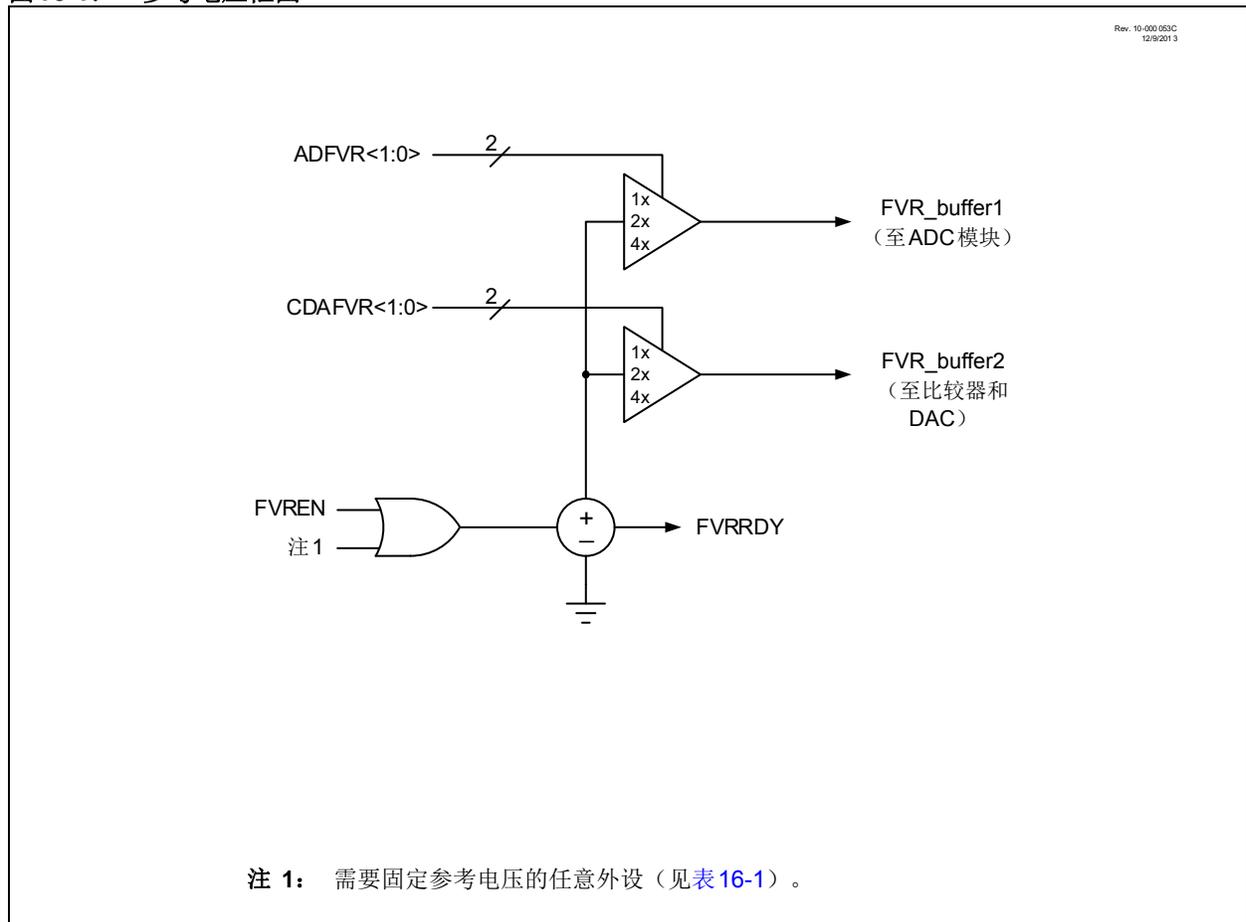
FVRCON 寄存器的 ADFVR<1:0> 位用于使能和配置送到 ADC 模块的参考电压的增益放大器设置。更多信息，请参见第 22.0 节“模数转换器 (ADC) 模块”。

可使用 FVRCON 寄存器的 CDADFVR<1:0> 位来为提供给 DAC 和比较器模块的参考电压使能和配置增益放大器设置。更多信息，请参见第 24.0 节“5 位数模转换器 (DAC1) 模块”和第 18.0 节“比较器模块”。

16.2 FVR 稳定周期

当固定参考电压模块使能时，参考电压和放大器电路需要一段时间才能达到稳定。在电路稳定下来、可供使用时，FVRCON 寄存器的 FVRRDY 位将会置 1。FVR 启动时间请参见表 35-16。

图 16-1: 参考电压框图



PIC16(L)F18324/18344

16.3 寄存器定义：FVR控制

寄存器 16-1: **FVRCON**: 固定参考电压控制寄存器

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN	FVRRDY ⁽¹⁾	TSEN ⁽³⁾	TSRNG ⁽³⁾	CDAFVR<1:0>		ADFVR<1:0>	
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	q = 值取决于具体条件

- bit 7 **FVREN**: 固定参考电压使能位
1 = 使能固定参考电压
0 = 禁止固定参考电压
- bit 6 **FVRRDY**: 固定参考电压就绪标志位⁽¹⁾
1 = 固定参考电压输出就绪备用
0 = 固定参考电压输出未就绪或未使能
- bit 5 **TSEN**: 温度指示器使能位⁽³⁾
1 = 使能温度指示器
0 = 禁止温度指示器
- bit 4 **TSRNG**: 温度指示器范围选择位⁽³⁾
1 = $V_{OUT} = V_{DD} - 4V_T$ (高电压范围)
0 = $V_{OUT} = V_{DD} - 2V_T$ (低电压范围)
- bit 3-2 **CDAFVR<1:0>**: 比较器FVR缓冲器增益选择位
11 = 比较器FVR缓冲器增益为4x (4.096V)⁽²⁾
10 = 比较器FVR缓冲器增益为2x (2.048V)⁽²⁾
01 = 比较器FVR缓冲器增益为1x (1.024V)
00 = 比较器FVR缓冲器关闭
- bit 1-0 **ADFVR<1:0>**: ADC FVR缓冲器增益选择位
11 = ADC FVR缓冲器增益为4x (4.096V)⁽²⁾
10 = ADC FVR缓冲器增益为2x (2.048V)⁽²⁾
01 = ADC FVR缓冲器增益为1x (1.024V)
00 = ADC FVR缓冲器关闭

- 注 1: FVRRDY 始终为1。
2: 固定参考电压输出不能超过V_{DD}。
3: 更多信息, 请参见第17.0节“温度指示器模块”。

表16-1: 与固定参考电压相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		178
ADCON0	CHS<5:0>						GO/DONE	ADON	242
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		243
CMxCON1	CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>			189
DAC1CON0	DAC1EN	—	DAC1OE	—	DAC1PPS<1:0>		—	DAC1NSS	261

图注: 固定参考电压不使用阴影单元。

PIC16(L)F18324/18344

17.0 温度指示器模块

本系列器件配备了用于测量管芯工作温度的温度电路。电路的工作温度范围介于-40°C和+85°C之间。其输出是与器件温度成比例的电压。温度指示器的输出在内部与器件ADC连接。

电路可以用作温度阈值检测器，也可以用作更精确的温度指示器，这取决于所执行的校准级别。执行单点校准时，电路可以指示邻近该点的温度。执行双点校准时，电路可以更精确地检测整个温度范围。关于校准过程的更多详细信息，请参见应用笔记AN2092，“*Using the Temperature Indicator Module*”（DS00002092）。

17.1 电路工作原理

图17-1给出了温度电路的简化框图。与温度成比例的电压输出通过测量多个硅结的正向压降而得到。

公式17-1描述了温度指示器的输出特性。

公式17-1: V_{OUT} 范围

高电压范围: $V_{OUT} = V_{DD} - 4V_T$

低电压范围: $V_{OUT} = V_{DD} - 2V_T$

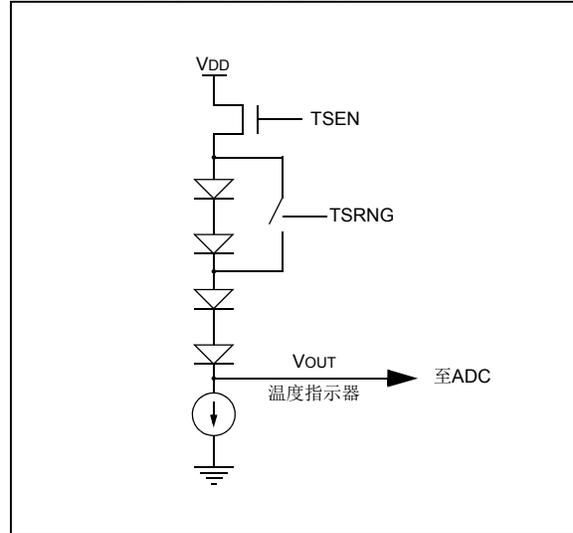
温度检测电路与固定参考电压（FVR）模块集成在一起。更多信息，请参见第16.0节“固定参考电压（FVR）”。

可以通过将FVRCON寄存器的TSEN位置1来使能该电路。在禁止时，电路不会消耗任何电流。

电路可以工作于高电压范围或低电压范围。高电压范围的选择方式是将FVRCON寄存器的TSRNG位置1，从而可提供较宽的输出电压。这可以在整个温度量程中提供更高的分辨率。该电压范围需要较高的偏置电压才能工作，所以需要较高的VDD。

低电压范围的选择方式是将FVRCON寄存器的TSRNG位清零。低电压范围产生的压降较小，所以只需较低的VDD电压就可以让电路工作。低电压范围旨在用于进行低电压操作。

图17-1: 温度指示器电路图



17.2 最小工作电压VDD

当温度电路工作于低电压范围时，器件可以在规范范围内的任意工作电压下工作。

当温度电路工作于高电压范围时，器件工作电压VDD必须足够高，以确保正确地偏置温度电路。

表17-1给出了建议的最小VDD与范围设置。

表17-1: 建议的VDD与范围

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

17.3 温度输出

电路的输出使用内部模数转换器测量。提供了一路通道用于温度电路输出。详细信息，请参见第22.0节“模数转换器（ADC）模块”。

17.4 ADC采集时间

为了确保精确的温度测量，用户必须在ADC输入多路开关连接到温度指示器输出之后至少等待200 μ s，然后再执行转换。此外，用户必须在温度指示器输出的连续两次转换之间等待200 μ s。

表17-2: 与温度指示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	寄存器所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		178

图注： 温度指示器模块不使用阴影单元。

PIC16(L)F18324/18344

18.0 比较器模块

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

- 可编程输入选择
 - 可选择参考电压
- 可编程输出极性
- 输出上升沿/下降沿中断
- 从休眠模式唤醒
- CWG自动关断源

18.1 比较器概述

图18-1所示为单比较器以及模拟输入电压与数字输出之间的关系。当VIN+上的模拟电压小于VIN-上的模拟电压时，比较器输出为数字低电平。当VIN+上的模拟电压大于VIN-上的模拟电压时，比较器输出为数字高电平。

表18-1列出了该器件可用的比较器。

表18-1: 可用的比较器

器件	C1	C2
PIC16(L)F18324	•	•
PIC16(L)F18344	•	•

图18-1: 单个比较器

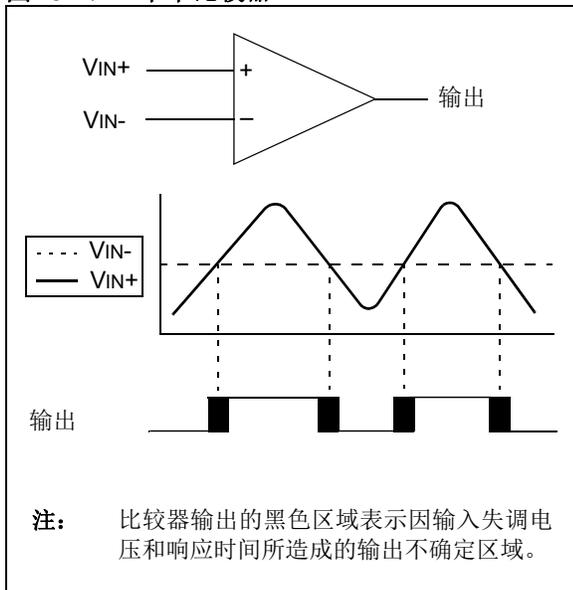
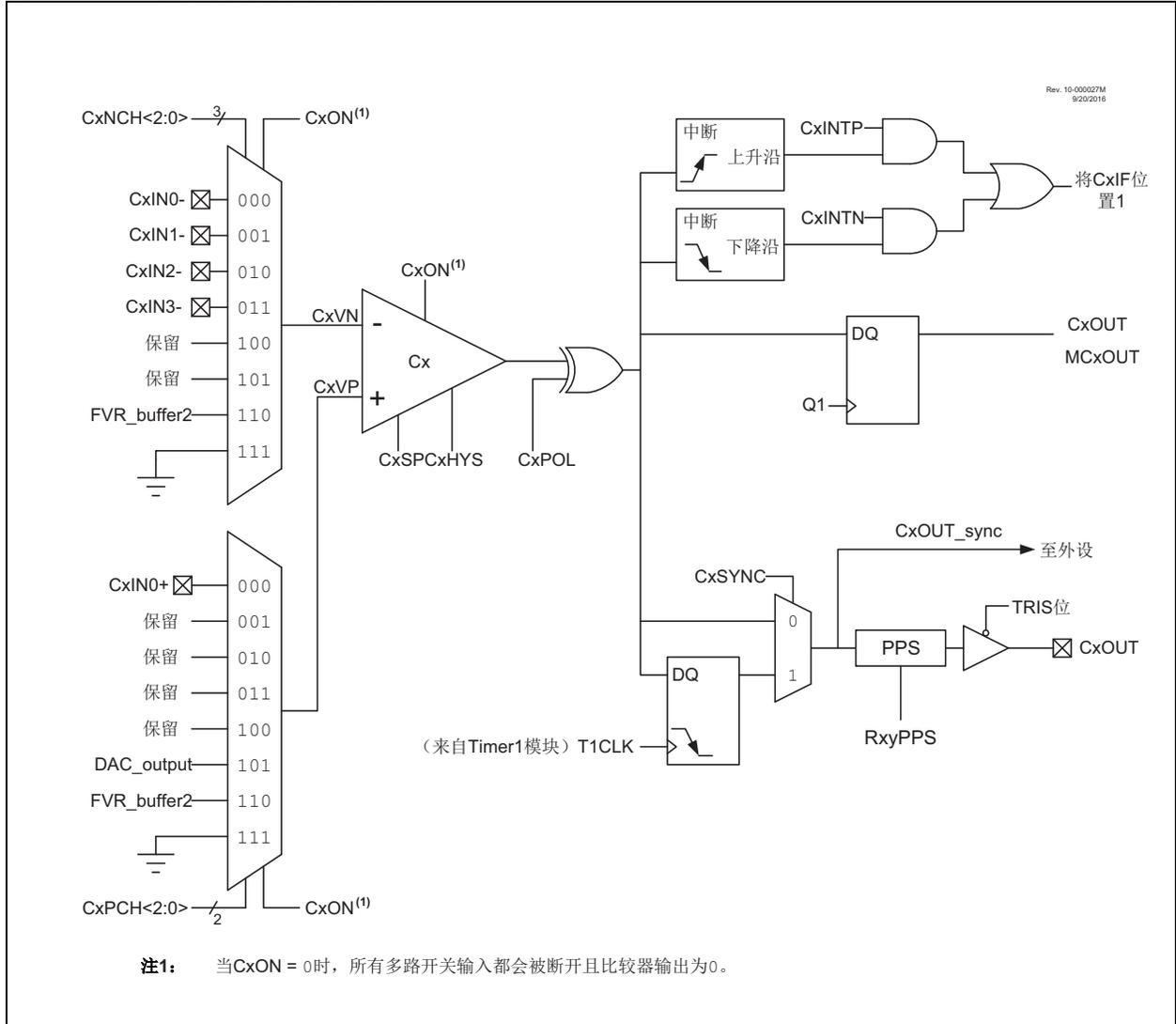


图18-2: 比较器模块的简化框图



PIC16(L)F18324/18344

18.2 比较器控制

每个比较器都具有2个控制寄存器：CMxCON0和CMxCON1。

CMxCON0寄存器（见寄存器18-1）包含以下控制和状态位：

- 使能
- 输出
- 输出极性
- 滞后使能
- Timer1输出同步

CMxCON1寄存器（见寄存器18-2）包含以下控制位：

- 正/负边沿中断允许
- 同相输入通道选择
- 反相输入通道选择

18.2.1 比较器使能

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

18.2.2 比较器输出

可以通过读CMxCON0寄存器的CxOUT位或CMOUT寄存器的MCxOUT位监视比较器的输出。

比较器输出还可以通过RxyPPS寄存器（寄存器13-2）输送到外部引脚。要使能引脚作为输出，必须清零相应的TRIS位。

注 1： 比较器的内部输出在每个指令周期被锁存。除非另外指定，否则不锁存外部输出。

18.2.3 比较器输出极性

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

表18-2给出了输出状态与输入条件的关系（包括极性控制）。

表18-2： 比较器输出状态与输入条件

输入条件	CxPOL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

18.3 比较器滞后

通过在每个比较器的输入引脚上施加不同的正向和负向增长阈值电压，可以为整体操作提供滞后功能。滞后功能通过将CMxCON0寄存器的CxHYS位置1来使能。

更多信息，请参见表35-14的比较器规范。

18.4 Timer1 门控操作

比较器操作产生的输出可以用作Timer1的门控源。更多信息，请参见第27.5节“Timer1 门控”。该功能可用于对模拟事件的持续时间或间隔时间进行计时。

建议将比较器输出与Timer1进行同步。这可以确保在比较器输出发生变化时，Timer1不会递增。

18.4.1 比较器输出同步

通过将CMxCON0寄存器的CxSYNC位置1，可以使比较器的输出与Timer1保持同步。

使能比较器的输出时，比较器的输出在Timer1时钟源的下降沿锁存。这使得定时器/计数器与CxOUT位同步，从而不会造成软件对时序含混不清。更多信息，请参见比较器框图（图18-2）和Timer1框图（图27-1）。

18.5 比较器中断

当上升沿或下降沿检测器检测到每个比较器的输出值发生变化时，将产生中断。

当触发任一边沿检测器时，如果它关联的允许位已置1（CMxCON1寄存器的CxINTP和/或CxINTN位），则相应的中断标志位（PIR2寄存器的CxIF位）会置1。

要允许中断，必须将以下位置1：

- CMxCON0寄存器的CxON位
- PIE2寄存器的CxIE位
- CMxCON1寄存器的CxINTP位（对于上升沿检测）
- CMxCON1寄存器的CxINTN位（对于下降沿检测）
- INTCON寄存器的PEIE和GIE位

相关的中断标志位（PIR2寄存器的CxIF位）必须用软件清零。如果在清零该标志时检测到另一个边沿，则标志仍然会在序列结束时置1。

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

18.6 比较器同相输入选择

通过配置CMxCON1寄存器的CxPCH<2:0>位，将内部参考电压或模拟引脚连接到比较器的同相输入：

- CxIN0+ 模拟引脚
- DAC输出
- FVR（固定参考电压）
- Vss（地）

关于固定参考电压模块的更多信息，请参见第16.0节“固定参考电压（FVR）”。

关于DAC输入信号的更多信息，请参见第24.0节“5位数模转换器（DAC1）模块”。

每当禁止比较器（CxON = 0）时，所有比较器输入都会被禁止。

18.7 比较器反相输入选择

CMxCON1寄存器的CxNCH<2:0>位指示模拟输入引脚、内部参考电压或模拟地连接到比较器的反相输入。

- CxIN- 引脚
- FVR（固定参考电压）
- 模拟地

一些反相输入选择与运算放大器输出功能共用一个引脚。同时使能这两个功能会将运算放大器输出定向到比较器反相输入。

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

18.8 比较器响应时间

在改变输入源或选择新的参考电压后的一段时间内，比较器的输出状态是不确定的。这段时间被称为响应时间。比较器的响应时间不同于参考电压的稳定时间。因此，在确定比较器输入改变的总响应时间时，必须考虑这两个时间。更多详细信息，请参见表35-14中的比较器和参考电压规范。

PIC16(L)F18324/18344

18.9 模拟输入连接注意事项

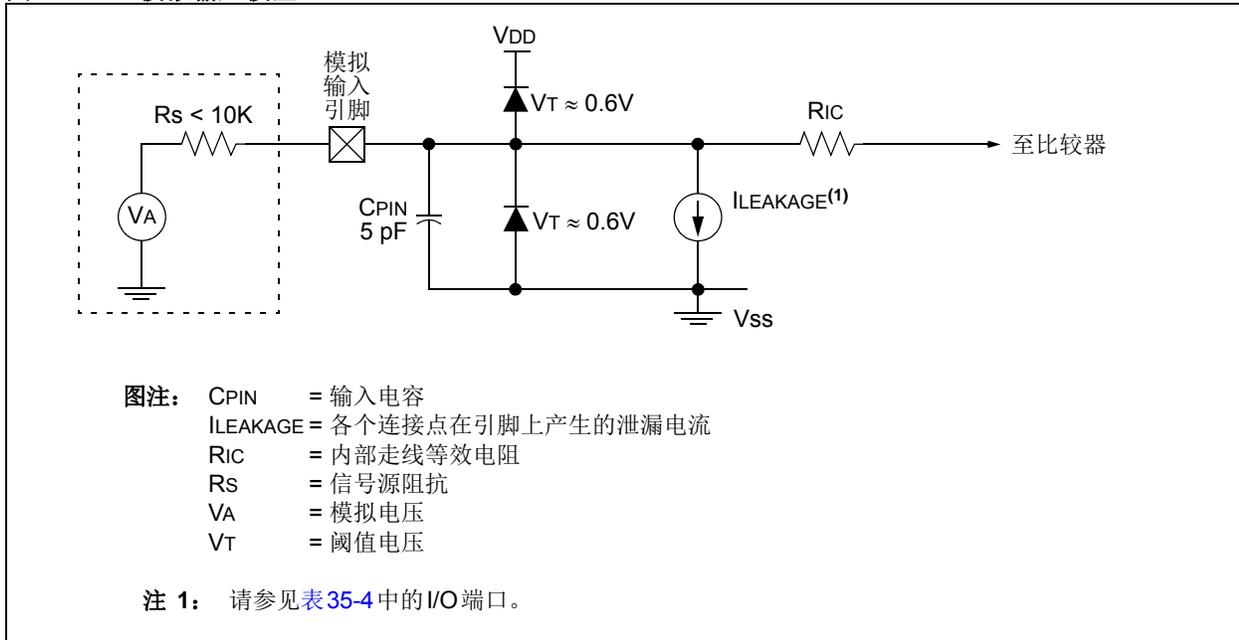
模拟输入的简化电路如图18-3所示。由于模拟输入引脚与数字输入共用连接，它们与VDD和VSS之间连有反向偏置的ESD保护二极管。因此，模拟输入的值必须在VSS和VDD之间。如果输入电压与这一范围偏离的绝对值超过0.6V，就可能发生一个二极管正向导通，从而可能导致锁死发生。

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

注 1: 当读端口寄存器时，所有配置为模拟输入的引脚都读为0。配置为数字输入的引脚将基于相应的TTL或ST输入缓冲器电平提供输入。

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

图18-3: 模拟输入模型



18.10 CWG 自动关断源

比较器模块的输出可用作CWG模块的自动关断源。当比较器的输出有效且相应的ASx \bar{E} 使能时，CWG操作立即暂停（第20.7.1.2节“外部输入源关断”）。

18.11 休眠模式下的操作

比较器模块可以在休眠模式下工作。比较器时钟源基于Timer1时钟源。如果Timer1时钟源为系统时钟（Fosc）或指令时钟（Fosc/4），Timer1将不会在休眠期间工作，并且同步比较器输出将无法运行。

比较器中断可以将器件从休眠模式唤醒。PIE2寄存器的CxIE位必须置1才能允许比较器中断。

PIC16(L)F18324/18344

18.12 寄存器定义：比较器控制

寄存器 18-1: **CMxCON0: 比较器Cx控制寄存器0**

R/W-0/0	R-0/0	U-0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **CxON:** 比较器使能位
1 = 使能比较器
0 = 禁止比较器, 不消耗有功功率
- bit 6 **CxOUT:** 比较器输出位
如果 CxPOL = 1 (极性反相):
1 = CxVP < CxVN
0 = CxVP > CxVN
如果 CxPOL = 0 (极性不反相):
1 = CxVP > CxVN
0 = CxVP < CxVN
- bit 5 **未实现:** 读为0
- bit 4 **CxPOL:** 比较器输出极性选择位
1 = 比较器输出反相
0 = 比较器输出不反相
- bit 3 **未实现:** 读为0。
- bit 2 **CxSP:** 比较器速度/功耗选择位
1 = 比较器工作在正常功耗、高速模式下
0 = 保留。(不要使用)
- bit 1 **CxHYS:** 比较器滞后使能位
1 = 使能比较器滞后
0 = 禁止比较器滞后
- bit 0 **CxSYNC:** 比较器输出同步模式位
1 = 送到Timer1和I/O引脚的比较器输出与Timer1时钟源的变化同步。输出在Timer1时钟源的下降沿进行更新。
0 = 送到Timer1和I/O引脚的比较器输出是异步的

寄存器 18-2: CMxCON1: 比较器Cx控制寄存器1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>		
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **CxINTP:** 比较器正向边沿中断允许位
 1 = 在 CxOUT 位的正向边沿, CxIF 中断标志将置 1
 0 = 在 CxOUT 位的正向边沿, CxIF 中断标志不会置 1
- bit 6 **CxINTN:** 比较器负向边沿中断允许位
 1 = 在 CxOUT 位的负向边沿, CxIF 中断标志将置 1
 0 = 在 CxOUT 位的负向边沿, CxIF 中断标志不会置 1
- bit 5-3 **CxPCH<2:0>:** 比较器同相输入通道选择位
 111 = CxVP 连接到 Vss
 110 = CxVP 连接到 FVR 缓冲器 2
 101 = CxVP 连接到 DAC 输出
 100 = CxVP 未连接
 011 = CxVP 未连接
 010 = CxVP 未连接
 001 = CxVP 未连接
 000 = CxVP 连接到 CxIN0+ 引脚
- bit 2-0 **CxNCH<2:0>:** 比较器反相输入通道选择位
 111 = CxVN 连接到 Vss
 110 = CxVN 连接到 FVR 缓冲器 2
 101 = CxVN 未连接
 100 = CxVN 未连接
 011 = CxVN 连接到 CxIN3- 引脚
 010 = CxVN 连接到 CxIN2- 引脚
 001 = CxVN 连接到 CxIN1- 引脚
 000 = CxVN 连接到 CxIN0- 引脚

PIC16(L)F18324/18344

寄存器 18-3: CMOUT: 比较器输出寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
bit 7						bit 0	

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

bit 7-2 未实现: 读为0
bit 1 **MC2OUT**: C2OUT的镜像副本位
bit 0 **MC1OUT**: C1OUT的镜像副本位

表 18-3: 与比较器模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TRISA	—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	141
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
CMxCON0	CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC	188
CMxCON1	CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>			189
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	190
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		178
DACCON0	DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS	261
DACCON1	—	—	—	DAC1R<4:0>					262
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BLC2IE	TMR4IE	NCO1IE	102
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BLC2IF	TMR4IF	NCO1IF	107
CLCINxPPS	—	—	—	CLCINxPPS<4:0>					160
MDMINPPS	—	—	—	MDMINPPS<4:0>					160
T1GPPS	—	—	—	T1GPPS<4:0>					160
CWGxAS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	216

图注: — = 未实现位, 读为0。比较器模块不使用阴影单元。

注 1: 仅限 PIC16(L)F18344。

19.0 脉宽调制 (PWM)

PWMx 模块产生不同频率和占空比的脉宽调制 (PWM) 信号。

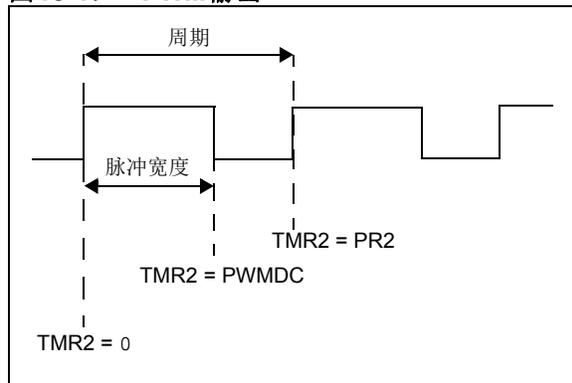
除了 CCP 模块，PIC16(L)F18324/18344 器件还包含 2 个 PWM 模块。

脉宽调制 (PWM) 是一种通过在完全开启和完全关闭状态之间进行快速切换而为负载供电的方案。PWM 信号类似于方波，信号的高电平部分视为开启状态 (脉冲宽度)，信号的低电平部分视为关闭状态。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，施加的功率就越低；占空比越高，施加的功率就越高。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 分辨率定义为可以在单个 PWM 周期中出现的最大步数。分辨率越高，就可以越精确地控制脉宽时间，从而更精确地控制施加在负载上的功率。

图 19-1 给出了 PWM 信号的典型波形。

图 19-1: PWM 输出



19.1 标准 PWM 模式

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

- TMR2、TMR4 或 TMR6 寄存器
- PR2、PR4 或 PR6 寄存器
- PWMxCON 寄存器
- PWMxDCH 寄存器
- PWMxDCL 寄存器

图 29-2 给出了 PWM 操作的简化框图。

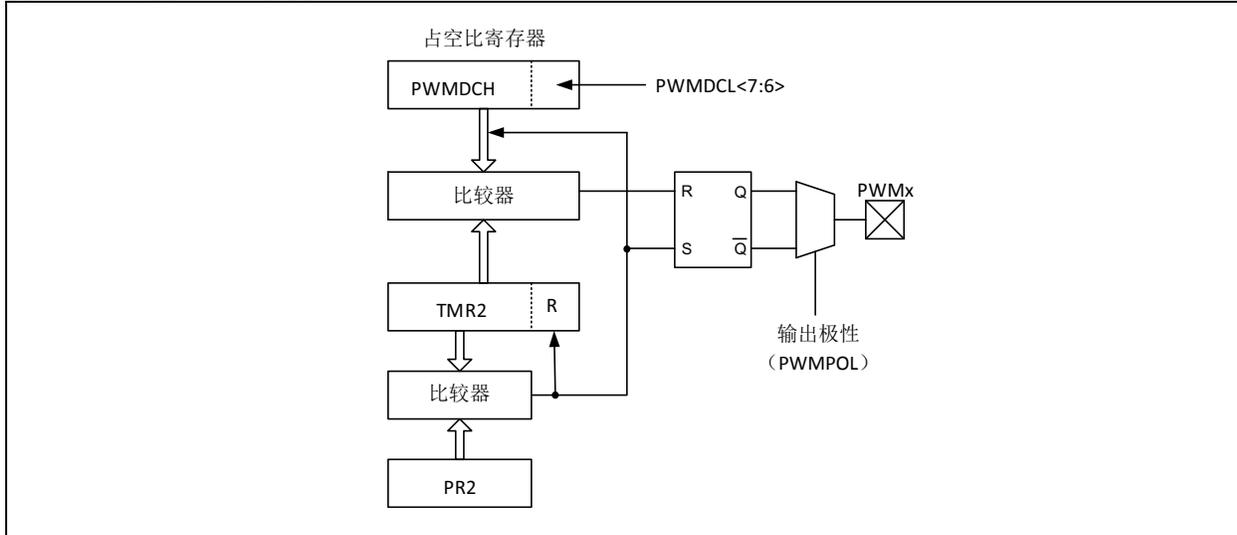
如果 PWMPOL = 0，输出的默认状态为 0。如果 PWMPOL = 1，默认状态为 1。如果 PWMEN = 0，输出将为默认状态。

注： 要使能 PWMx 引脚上的 PWM 输出，必须清零相应的 TRIS 位。

注： 为简便起见，相关公式和文本，请参见 TMR2 和 PR2。相同的公式和文本适用于 TMR4/6 和 PR4/6。定时器源可在寄存器 19-4 中选择。有关 TMR2/4/6 的更多信息，请参见第 28.0 节“Timer2/4/6 模块”

PIC16(L)F18324/18344

图19-2: 简化PWM框图



19.1.1 PWM周期

PWM输出的周期和脉冲宽度，请参见图19-1。PWM的频率是周期的倒数（1/周期）。

PWM周期可以通过写PR2寄存器来指定。使用以下公式计算PWM周期：

公式19-1: PWM周期

$$PWM周期 = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (TMR2 \text{ 预分频值})$$

注： $T_{osc} = 1/F_{osc}$

当TMR2中的值与PR2中的值相等时，在下一个递增周期将发生以下3个事件：

- TMR2被清零
- PWMx引脚被置1（例外情况：如果PWM占空比 = 0%，引脚将不会被置1。）
- 锁存PWMxDC中的PWM脉冲宽度。

注： 如果脉冲宽度值比周期长，则指定的PWM引脚将保持不变。

19.1.2 PWM占空比

可通过将一个10位值写入PWMxDC寄存器来指定PWM占空比。PWMxDC包含高8位，而PWMxDCL寄存器的bit <7:6>包含低2位。

PWMDC寄存器是双重缓冲的，可随时进行更新。这种双重缓冲极其重要，可以避免在PWM工作过程中产生毛刺。当TMR2 = PR2时，新值生效。请注意，PWMDC是左对齐的。

8位定时器TMR2寄存器与2位内部系统时钟（FOSC）或预分频器的2位一起构成10位时基。如果Timer2预分频比设置为1:1，则使用系统时钟。

公式19-2用于计算PWM脉冲宽度。

公式19-3用于计算PWM占空比。

公式19-2: 脉冲宽度

$$脉冲宽度 = (PWMxDC) \cdot T_{osc} \cdot (TMR2 \text{ 预分频值})$$

公式19-3: 占空比

$$占空比 = \frac{PWMxDC}{4(PR2 + 1)}$$

19.1.3 PWM分辨率

PWM分辨率以位数来表示，定义在单个PWM周期中可实现的最大离散步阶数。分辨率决定给定周期的可用占空比数。例如，10位分辨率将可得到1024个离散的步阶，而8位分辨率将可得到256个离散的步阶。

当PR2为255时，最大PWM分辨率为10位。分辨率是PR2寄存器值的函数，如公式19-4所示。

公式19-4: PWM分辨率

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

注： 如果脉冲宽度值比周期长，则指定的PWM引脚将保持不变。

19.1.4 休眠模式下的操作

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

19.1.5 改变系统时钟频率

PWM频率是由系统时钟频率得到的。系统时钟频率的任何改变将导致PWM频率的改变。更多详细信息，请参见第7.0节“振荡器模块”。

19.1.6 复位的影响

任何复位都将强制所有端口为输入模式，并强制PWMx寄存器为其复位状态。

19.1.7 设置PWM操作

当将模块配置为使用PWMx输出时，应遵循以下步骤：

1. 通过将相关的TRIS位置1，禁止PWMx引脚输出驱动器。
2. 通过配置PWMxCON寄存器的PWMxPOL位，配置PWM输出极性。
3. 将PWM周期值（由公式19-1确定）装入PR2寄存器。
4. 将PWM占空比值（由公式19-2确定）装入PWMxDCH寄存器和PWMxDCL寄存器的bit <7:6>。
5. 配置并启动Timer2：
 - 清零PIR1寄存器的TMR2IF中断标志位。
 - 通过配置T2CON寄存器的T2CKPS位来选择Timer2预分频值。
 - 通过将T2CON寄存器的TMR2ON位置1来使能Timer2。
6. 等待至TMR2IF置1。
7. 当TMR2IF标志位置1时：
 - 清零相关的TRIS位来使能输出驱动器。
 - 通过配置RxyPPS寄存器，将信号输送到所需引脚。
 - 通过将PWMxCON寄存器的PWMxEN位置1来使能PWMx模块。

为在第一个PWM输出时发送完整的占空比和周期，必须按给出的顺序执行上述步骤。如果以完整的PWM信号开始不是很重要，则可在第二步中通过将PWMxCON寄存器的PWMxEN位置1来使能PWM模块。

PIC16(L)F18324/18344

19.2 寄存器定义：PWM控制

寄存器 19-1: PWMxCON: PWM控制寄存器

R/W-0/0	U-0	R-0	R/W-0/0	U-0	U-0	U-0	U-0
PWMxEN	—	PWMxOUT	PWMxPOL	—	—	—	—
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

- bit 7 **PWMxEN:** PWM模块使能位
1 = 使能PWM模块
0 = 禁止PWM模块
- bit 6 **未实现:** 读为0
- bit 5 **PWMxOUT:** 读取该位时的PWM模块输出电平。
- bit 4 **PWMxPOL:** PWMx输出极性选择位
1 = PWM输出为低电平有效
0 = PWM输出为高电平有效
- bit 3-0 **未实现:** 读为0

寄存器 19-2: PWMxDCH: PWM占空比高位

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PWMxDC<9:2>							
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

- bit 7-0 **PWMxDC<9:2>:** PWM占空比高8位
这些位是PWM占空比的高8位。低2位位于PWMxDCL寄存器中。

寄存器 19-3: PWMxDCL: PWM占空比低位

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
PWMxDC<1:0>		—	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

- bit 7-6 **PWMxDC<1:0>:** PWM占空比低2位
这两位是PWM占空比的低2位。高8位位于PWMxDCH寄存器中。
- bit 5-0 **未实现:** 读为0

PIC16(L)F18324/18344

表19-1: PWM频率和分辨率示例 (Fosc = 20 MHz)

PWM 频率	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频值	16	4	1	1	1	1
PR2 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率 (位)	10	10	10	8	7	6.6

表19-2: PWM频率和分辨率示例 (Fosc = 8 MHz)

PWM 频率	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频值	16	4	1	1	1	1
PR2 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率 (位)	10	10	10	8	7	6.6

寄存器 19-4: PWMTMRS: PWM定时器控制寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>	
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **未实现:** 读为0

bit 3-2 **P6TSEL<1:0>:** PWM6 模式定时器选择

- 00 = 保留
- 01 = PWM6 基于 TMR2
- 10 = PWM6 基于 TMR4
- 11 = PWM6 基于 TMR6

bit 1-0 **P5TSEL<1:0>:** PWM5 模式定时器选择

- 00 = 保留
- 01 = PWM5 基于 TMR2
- 10 = PWM5 基于 TMR4
- 11 = PWM5 基于 TMR6

PIC16(L)F18324/18344

表19-3: 与PWMx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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
PWM5CON	PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—	194
PWM5DCH	PWM5DC<9:2>								194
PWM5DCL	PWM5DC<1:0>		—	—	—	—	—	—	194
PWM6CON	PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—	194
PWM6DCH	PWM6DC<9:2>								194
PWM6DCL	PWM6DC<1:0>		—	—	—	—	—	—	194
PWMTMRS	—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		195
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
PIR2	TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF	107
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
PIE2	TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	102
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		296
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		290
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		290
TMR2	TMR2<7:0>								297
TMR4	TMR4<7:0>								297
TMR6	TMR6<7:0>								297
PR2	PR2<7:0>								297
PR4	PR4<7:0>								297
PR6	PR6<7:0>								297
CWGxDAT	—	—	—	—	DAT<3:0>				213
CLCxSEly	—	—	LCxDyS<5:0>						227
MDSRC	—	—	—	—	MDMS<3:0>				270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272

图注: — = 未实现位, 读为0。PWM模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

20.0 互补波形发生器 (CWG) 模块

互补波形发生器 (CWGx) 可从选择的输入源产生带死区延时的互补波形。

CWGx 模块具有以下特性:

- 可选死区时钟源控制
- 可选输入源
- 输出使能控制
- 输出极性控制
- 使用独立的6位上升沿和下降沿死区计数器进行死区控制
- 自动关断控制:
 - 可选关断源
 - 自动重启使能
 - 自动关断引脚改写控制

20.1 基本操作

CWG 基于所选输入源产生两个输出波形。

每路输出由关到开的跳变可能会因另一路输出由开到关的跳变而受到延时, 因而会立即产生延时, 在延时时间内两路输出都不进行驱动。这被称为死区, [第20.6节“死区控制”](#) 对此进行了介绍。

可能需要防止电路发生故障、反馈事件太晚送达或根本不送达的可能性。在这种情况下, 必须在故障条件造成损坏之前终止有效驱动。这称为自动关断, [第20.7节“自动关断控制”](#) 将对此进行介绍。

20.2 工作模式

CWGx 模块可以在6种不同模式下工作, 这些模式由 CWGxCON0 寄存器的 MODE<2:0> 位指定:

- 半桥模式
- 推挽模式
- 异步转向模式
- 同步转向模式
- 全桥模式, 正向
- 全桥模式, 反向

所有模式均接受单脉冲数据输入, 并且提供最多4个输出, 如以下小节所述。

所有模式均包含自动关断控制, 如[第20.11节“寄存器定义: CWG控制”](#)所述。

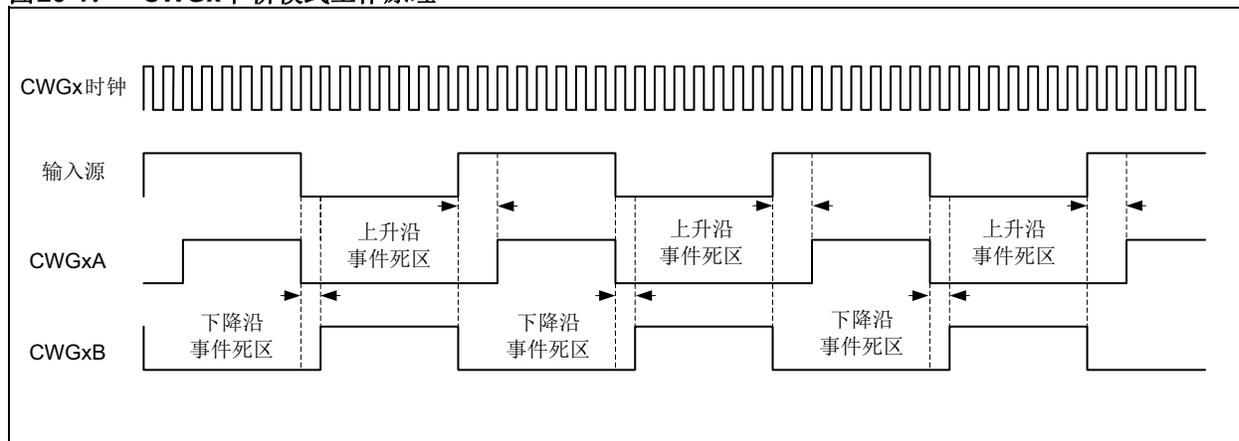
注: 除全桥模式 (如[第20.2.4节“全桥模式”](#)中所述) 外, 应仅在 EN = 0 (寄存器20-1) 时进行模式更改。

20.2.1 半桥模式

在半桥模式下, 将以输入的真值和反相形式生成两个输出信号, 如[图20-1](#)所示。在两个输出之间插入不重叠 (死区) 时间, 如[第20.6节“死区控制”](#)所述。转向模式不用于半桥模式。

未用输出 CWGxC 和 CWGxD 驱动类似的信号, 它们的极性分别由 POLC 和 POLD 位独立控制。

图20-1: CWGx半桥模式工作原理



PIC16(L)F18324/18344

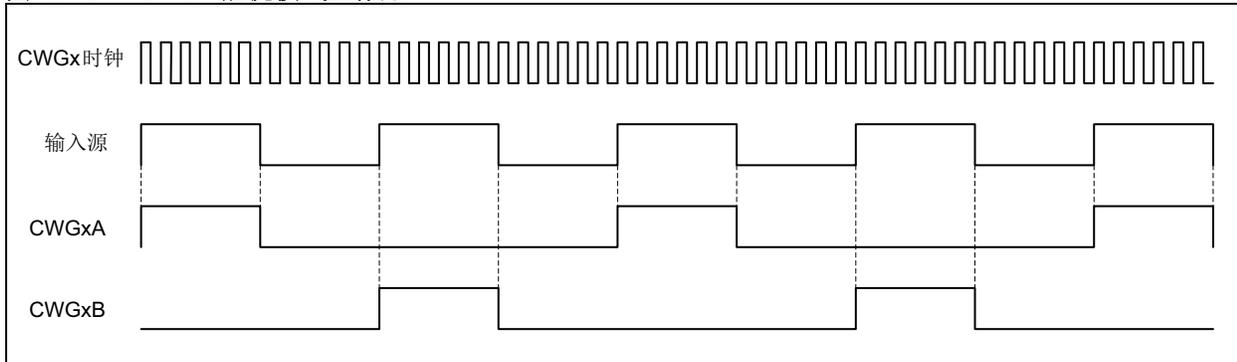
20.2.2 推挽模式

在推挽模式下，将生成两个输出信号，它们为输入的交替副本，如图20-2所示。这种交替可以产生驱动一些基于变压器的电源设计所需的推挽效应。死区控制和转向模式均不用于推挽模式。

推挽排序器在每次EN = 0或发生自动关断事件时复位。该排序器由第一个输入脉冲提供时钟，第一个输出出现在CWGxA上。

未用输出CWGxC和CWGxD分别驱动CWGxA和CWGxB的副本，但它们的极性分别由POLC和POLD位控制。

图20-2: CWGx推挽模式工作原理



20.2.3 转向模式

在同步和异步转向模式下，可以将调制输入信号转向到4个CWG输出任意组合并在未用于PWM输出的所有输出上送出固定值。每个输出具有独立的极性、转向和关断选项。死区控制不用于任一转向模式。

当STRy = 0（寄存器20-5）时，相应引脚保持在SDATy（寄存器20-5）定义的电平。当STRy = 1时，引脚由调制输入信号驱动。

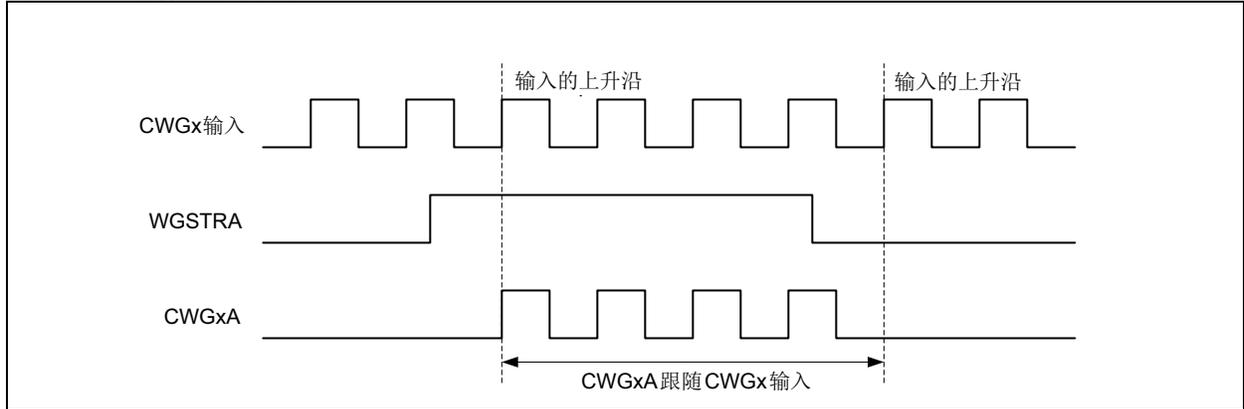
POLy位（寄存器20-2）仅在STRy = 1时控制信号极性。CWG自动关断工作原理也适用于转向模式，如第20.11节“寄存器定义：CWG控制”中所述。

注： 仅STRy位同步；DATy（数据）位不同步。

20.2.3.1 同步转向模式

在同步转向模式下 ($MODE\langle 2:0 \rangle$ 位 = 001, 寄存器20-1), 对转向选择寄存器的更改将在调制数据输入的下一个上升沿生效 (图20-3)。在同步转向模式下, 输出将始终产生完整波形。

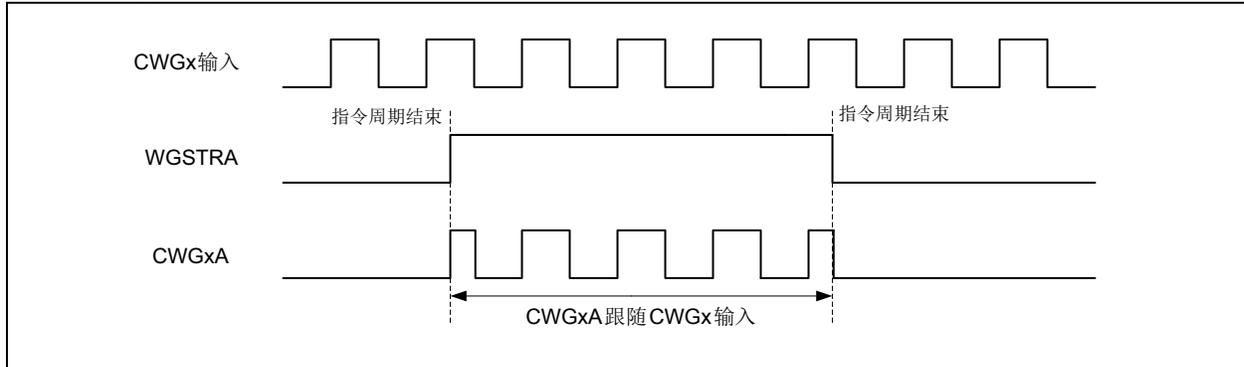
图20-3: 同步转向示例 ($MODE\langle 2:0 \rangle = 001$)



20.2.3.2 异步转向模式

在异步模式下 ($MODE\langle 2:0 \rangle$ 位 = 000, 寄存器20-1), 转向在写 $WGxSTR$ 的指令周期结束时生效。在异步转向模式下, 输出信号可能是不完整波形 (寄存器20-4)。此操作在用户固件需要立即除去该输出引脚的信号时非常有用。

图20-4: 异步转向示例 ($MODE\langle 2:0 \rangle = 000$)



20.2.3.3 启动注意事项

应用硬件必须在 CWG 输出引脚上使用适当的外部上拉和/或下拉电阻。这是必需的, 因为所有 I/O 引脚在复位时强制为高阻抗。

$POLy$ 位 (寄存器20-2) 允许用户选择输出信号是高电平有效还是低电平有效。

PIC16(L)F18324/18344

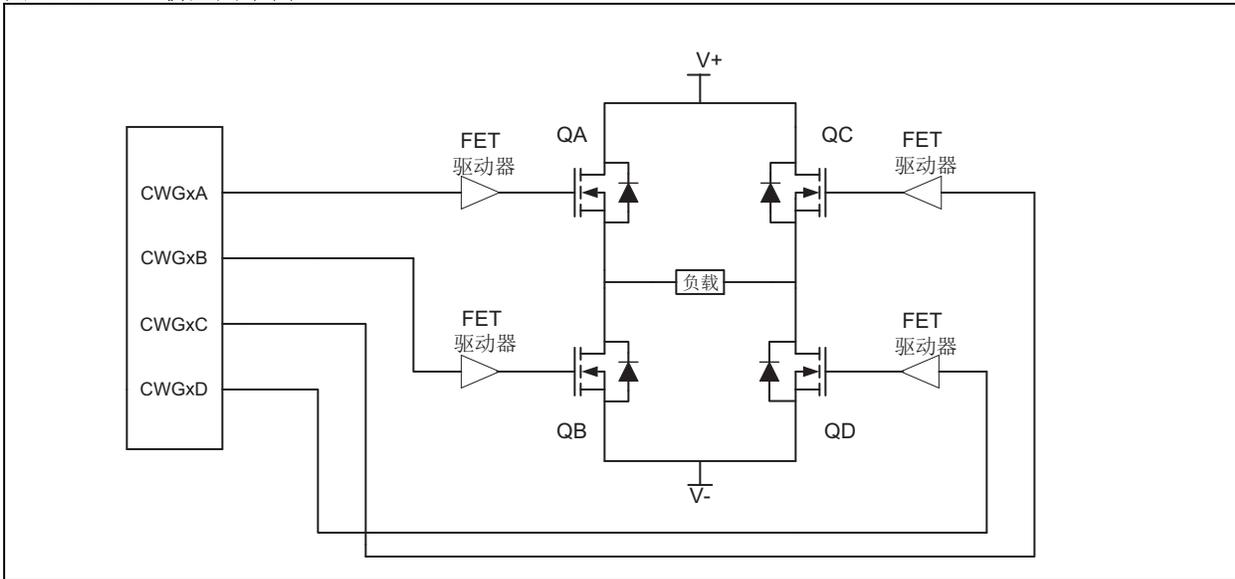
20.2.4 全桥模式

在正向和反向全桥模式下，3个输出驱动静态值，第4个输出则通过数据输入进行调制。第20.2.3节“转向模式”和第20.6节“死区控制”对死区控制进行了介绍。转向模式不与任一全桥模式配合使用。

模式选择可以在正向和反向之间切换（更改MODE<2:0>），而不清零EN。

当按图20-5所示连接时，输出适合全桥电机驱动器。每个CWG输出信号具有独立极性控制，因此该电路适合高电平有效和低电平有效驱动器。

图20-5： 全桥应用示例



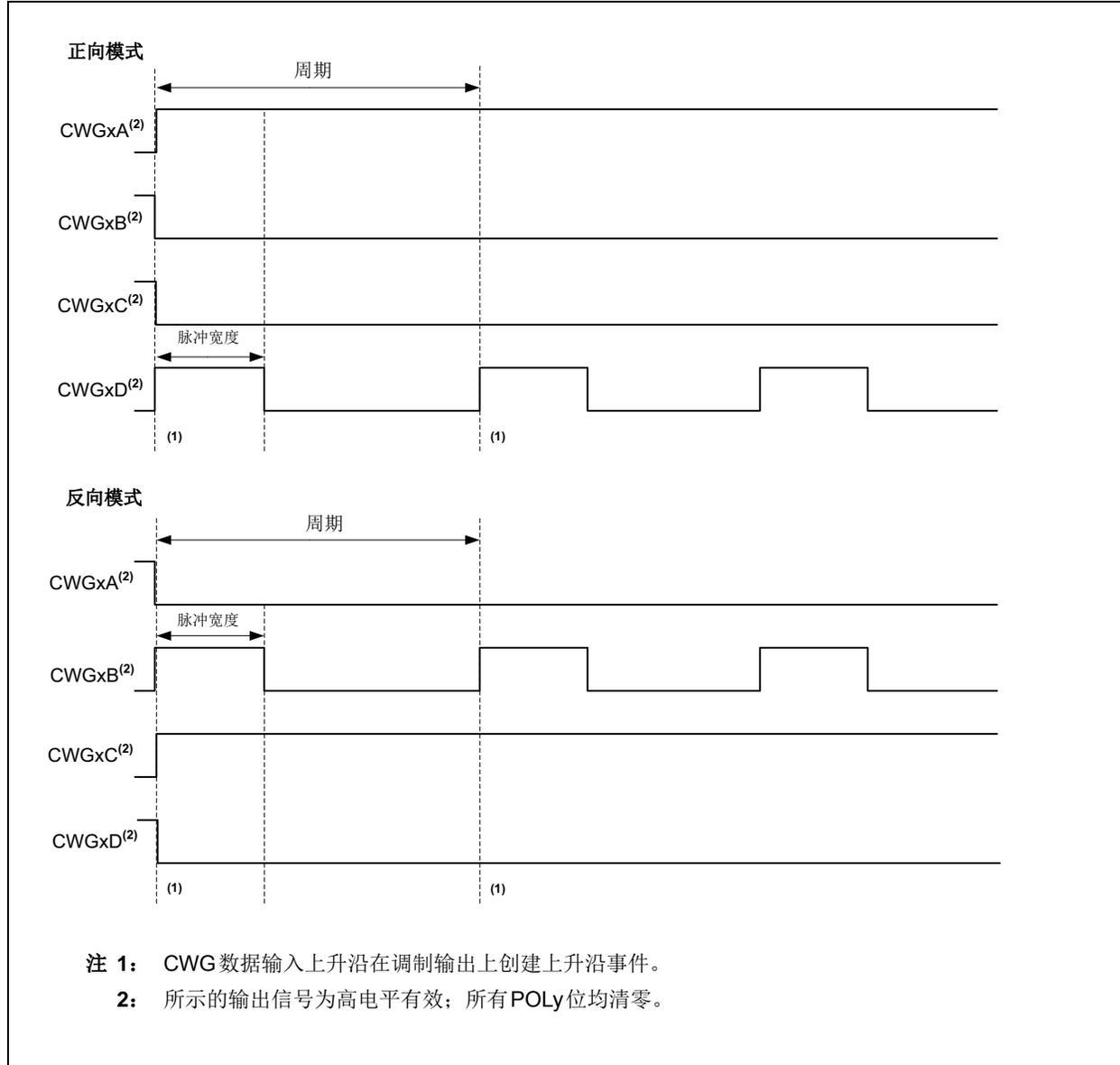
20.2.4.1 全桥正向模式

在全桥正向模式下 (MODE<2:0> = 010)，CWGxA驱动为其有效状态且对CWGxD进行调制，而CWGxB和CWGxC驱动为其无效状态，如图20-6顶部所示。

20.2.4.2 全桥反向模式

在全桥反向模式下 (MODE<2:0> = 011)，CWGxC驱动为其有效状态且对CWGxB进行调制，而CWGxA和CWGxD驱动为其无效状态，如图20-6底部所示。

图20-6: 全桥输出示例



20.2.4.3 全桥模式下的方向更改

在全桥模式下，更改MODE<2:0>控制正向/反向方向。对MODE<2:0>做出更改时，会在调制输入的下一个上升沿更改为新方向。

可用软件通过改变WGxCON0寄存器的MODE<2:0>位来启动方向更改。具体序列如图20-7所示。

- 相关的有效输出 CWGxA 和无效输出 CWGxC 切换为以相反的方向驱动。
- 先前调制输出 CWGxD 切换为无效状态，而先前无效输出 CWGxB 开始调制。
- 在经过方向切换死区后，恢复 CWG 调制。

PIC16(L)F18324/18344

20.2.4.4 全桥模式下的死区延时

当满足以下任一条件时，死区延时很重要：

1. 当数据输入的占空比接近或等于100%时，CWG输出的方向发生改变，或者
2. 功率开关（包括功率器件和驱动电路）的关断时间大于导通时间。

仅可在方向发生改变时插入死区延时，并且仅调制输出受到影响。静态配置输出（CWGxA和CWGxC）不提供死区，并且基本上同时开关。

图20-7给出了占空比接近100%时，CWG输出从正向变为反向的示例。在此示例中，在 t_1 时刻CWGxA和CWGxD输出变为无效，而CWGxC输出变为有效。由于功率器件的关断时间比导通时间长，直通电流可能在时间段“ t ”内流过功率器件QC和QD。当CWG方向从反向变为正向时，功率器件QA和QB上也会发生同样的现象。

如果应用需要在高占空比时更改CWG方向，则有两种方法可以避免出现直通电流：

1. 在更改方向前的一个周期减小CWG的占空比。
2. 使用可使开关元件的关断速度比导通速度更快的开关驱动器。

图20-7： 占空比接近100%时 PWM方向更改的示例

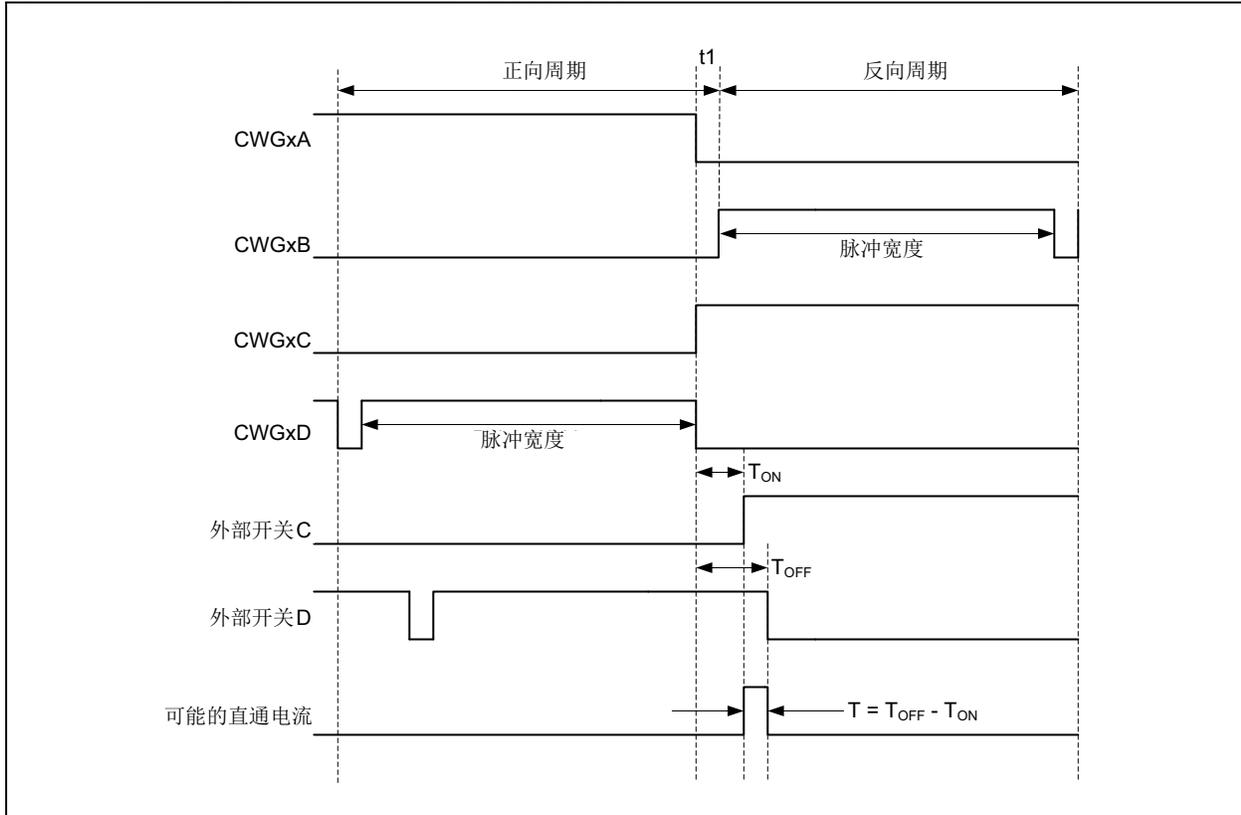
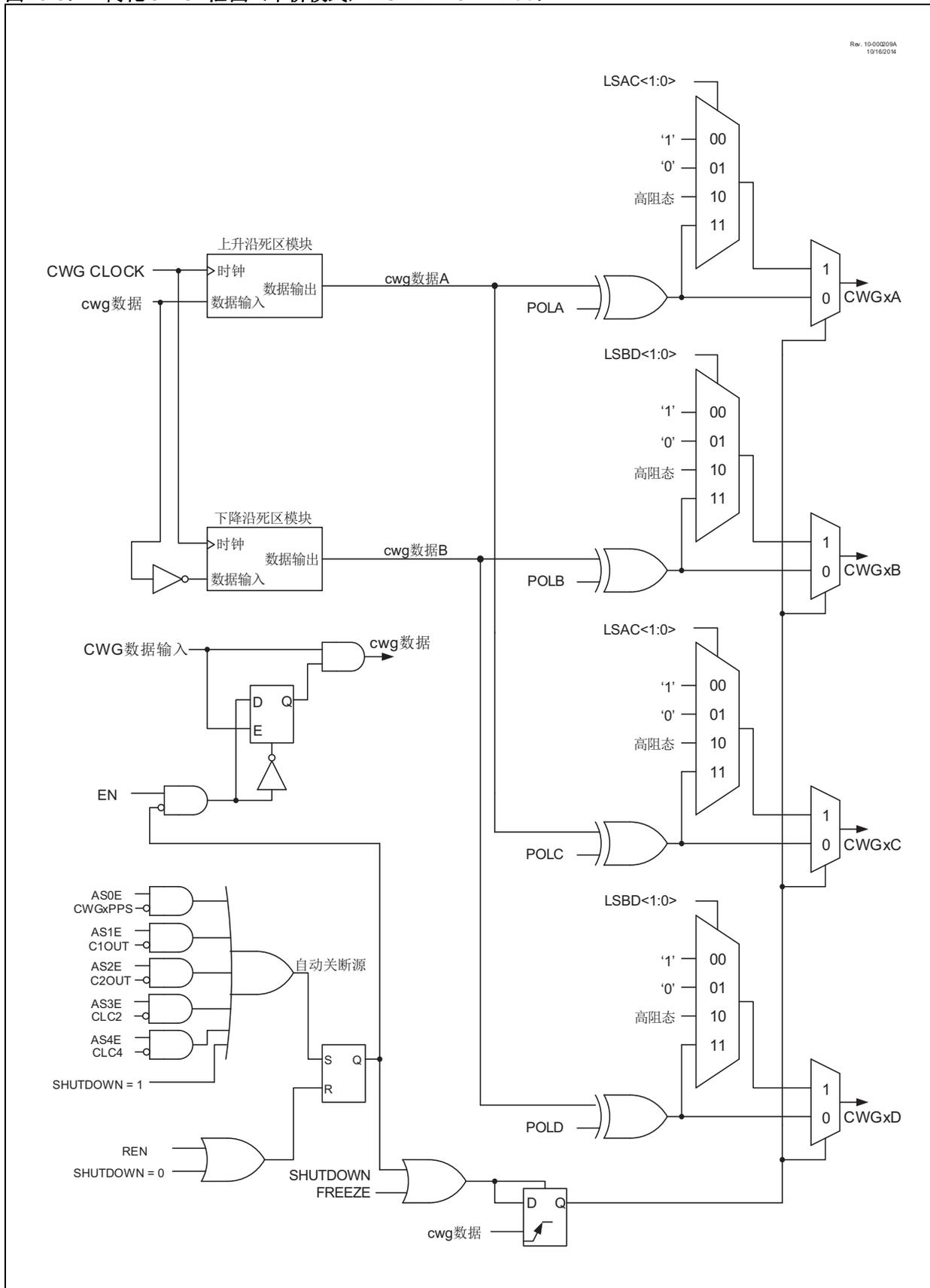


图 20-8: 简化CWGx框图 (半桥模式, MODE<2:0> = 100)



PIC16(L)F18324/18344

图20-9: 简化CWG框图 (推挽模式, MODE <2:0> = 101)

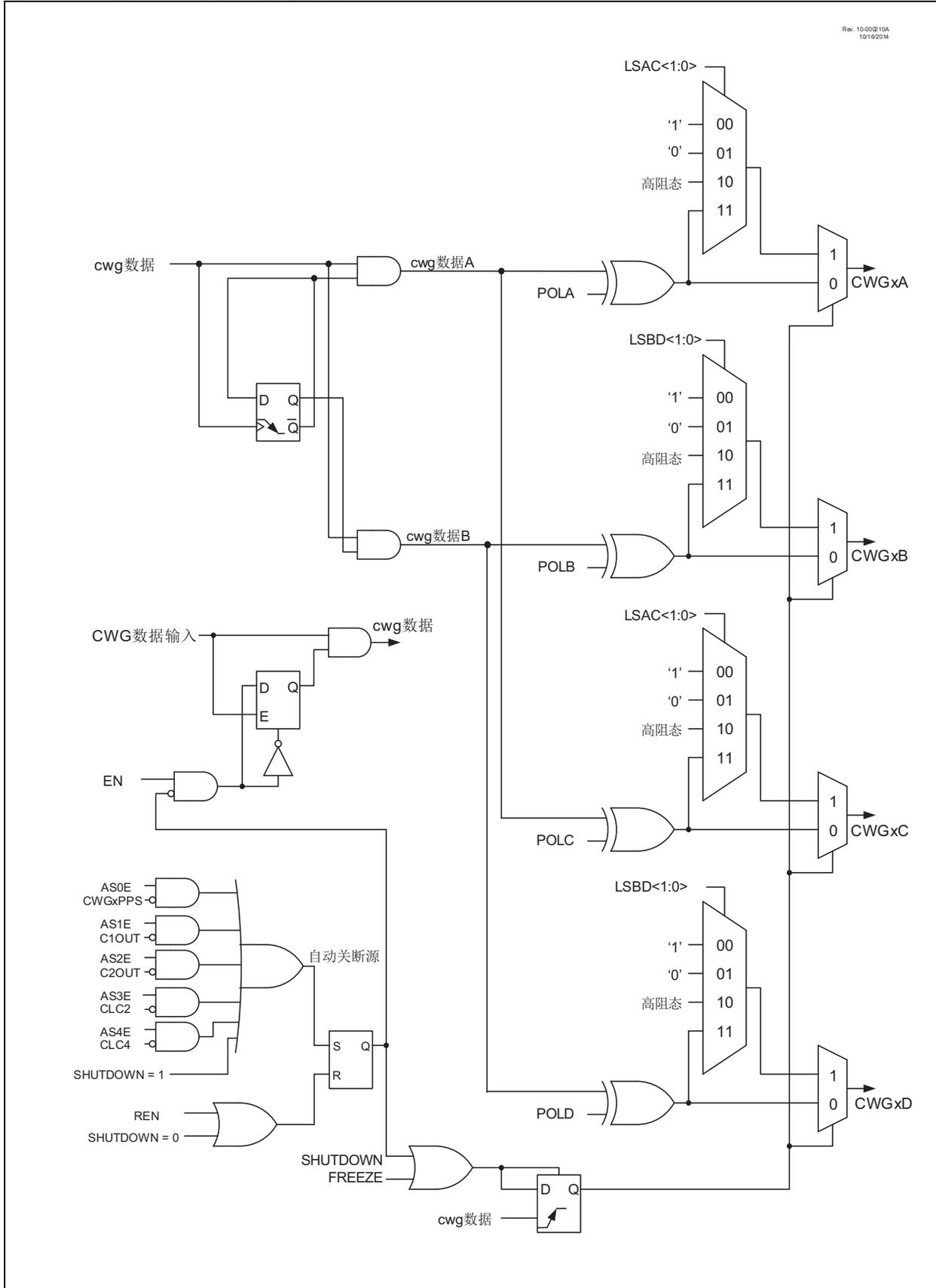
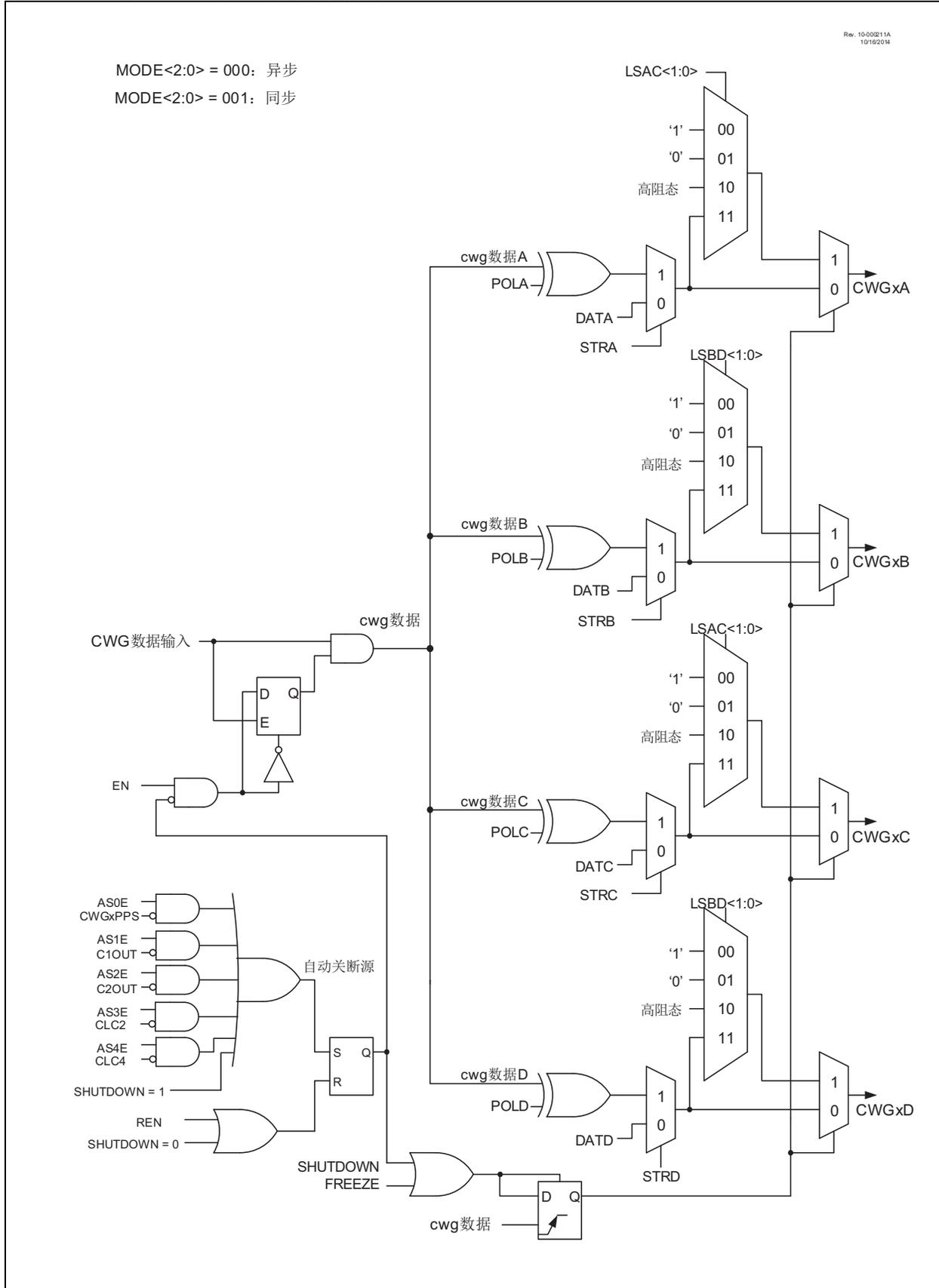
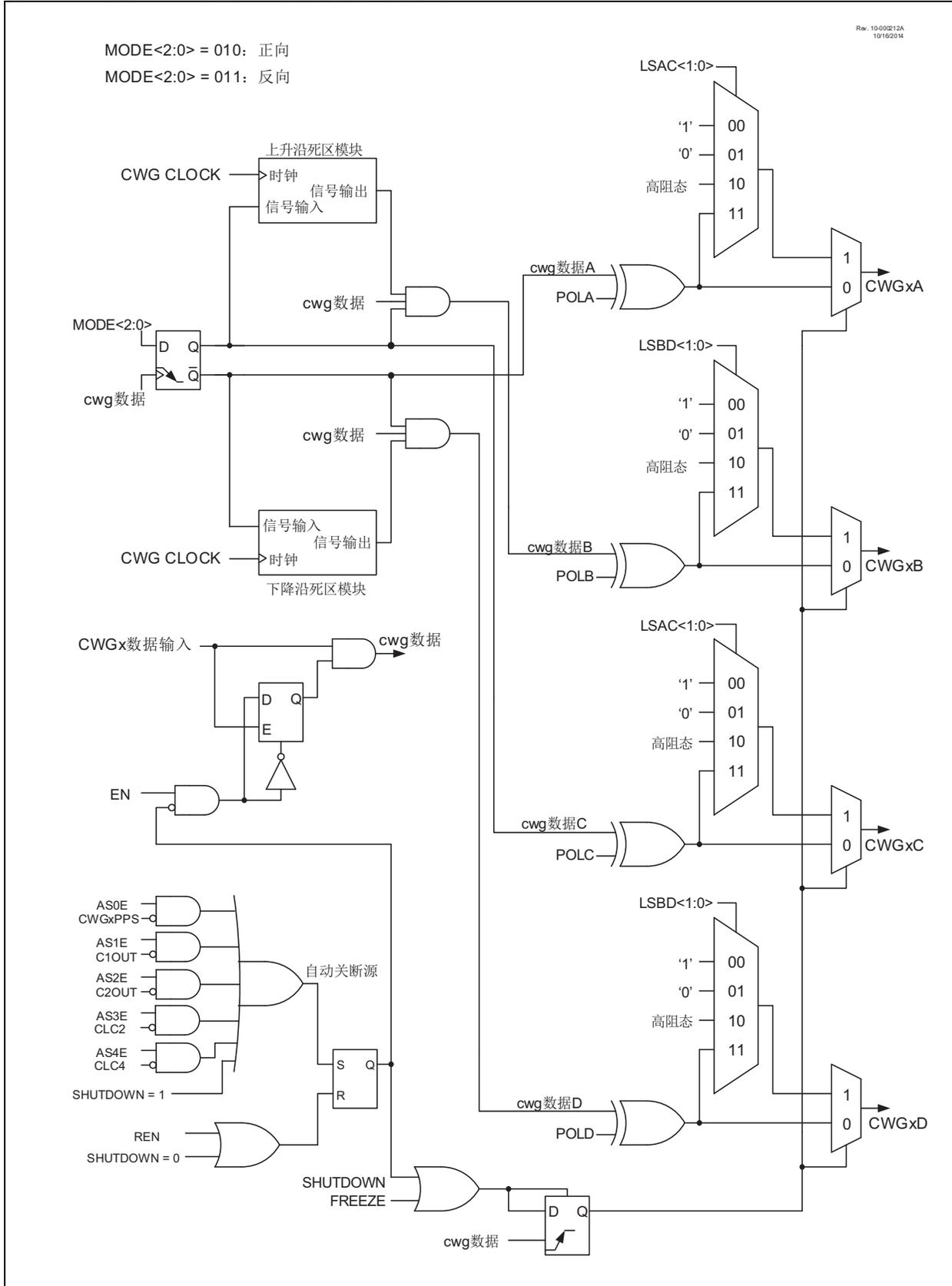


图20-10: 简化CWG框图 (输出转向模式)



PIC16(L)F18324/18344

图20-11: 简化CWG框图（正向和反向全桥模式）



20.3 时钟源

时钟源用于驱动死区时序电路。CWGx模块允许选择以下时钟源：

- Fosc（系统时钟）
- HFINTOSC（仅限 16 MHz）

选择HFINTOSC时，HFINTOSC将在休眠期间保持运行。因此，需要死区的CWG模式可在休眠模式下运行，前提是CWG数据输入也在休眠期间有效。时钟源使用CWGxCLKCON寄存器（寄存器20-3）的CS位进行选择。

20.4 可选输入源

CWG基于表20-1中的输入源来生成输出波形。

表20-1： 可选输入源

源外设	信号名
CWGxPPS	CWG PPS输入连接
C1OUT	比较器1输出
C2OUT	比较器2输出
CCP1	捕捉/比较/PWM输出
CCP2	捕捉/比较/PWM输出
CCP3	捕捉/比较/PWM输出
CCP4	捕捉/比较/PWM输出
PWM5	PWM5输出
PWM6	PWM6输出
NCO1	数控振荡器（NCO）输出
CLC1	可配置逻辑单元1输出
CLC2	可配置逻辑单元2输出
CLC3	可配置逻辑单元3输出
CLC4	可配置逻辑单元4输出

使用CWGxDAT寄存器（寄存器20-4）中的DAT<3:0>位选择输入源。

20.5 输出控制

使能CWG模块后，立即配置互补驱动，清零所有输出驱动。

20.5.1 CWGx输出

每个CWG输出可通过RxyPPS寄存器输送到外设引脚选择（PPS）输出（见第13.0节“外设引脚选择（PPS）模块”）。

20.5.2 极性控制

每个CWG输出的极性可以单独进行选择。当输出极性位置1时，相应的输出为低电平有效。清零输出极性位时，相应输出将配置为高电平有效。但是，极性不会影响改写电平。输出极性使用CWGxCON1寄存器的POLY位进行选择。

20.6 死区控制

死区控制用于提供不重叠的输出信号，以防止功率开关中产生直通电流。CWGx模块包含两个6位死区计数器。这些计数器可装入用于确定输入源的上升沿或下降沿上启动的死区长度的值。死区控制可用于半桥模式或全桥模式。

上升沿死区延时由上升沿死区计数寄存器（寄存器20-8，CWGxDBR）确定，而下降沿死区延时由下降沿死区计数寄存器（寄存器20-9，CWGxDBF）确定。死区持续时间的建立方式是对CWG时钟周期进行计数，从0开始一直计数至上升沿或下降沿死区计数器寄存器中装入的值。死区计数器在CWG时钟源的每个上升沿递增。

20.6.1 上升沿和反向死区

在半桥模式下，上升沿死区在CWG数据输入的上升沿之后延迟CWGxA输出的导通时间。在全桥模式下，仅在方向从正向模式变化为反向模式时才插入反向死区延时，并且仅调制输出CWGxB受到影响。

CWGxDBR寄存器用于确定输入源信号上升沿死区时间间隔的持续时间。该持续时间为0至64个CWG时钟周期。

死区总是在输入源信号的边沿启动。计数为0表示不存在死区。

如果输入源信号在完成死区计数前反转极性，则相应输出上将没有输出信号。

CWGxDBR寄存器值是双重缓冲的。当EN = 0（寄存器20-1）时，会在写入CWGxDBR时装入缓冲区。当EN = 1时，在LD位（寄存器20-1）置1后将在数据输入的下一个下降沿之后的上升沿装入缓冲区。

PIC16(L)F18324/18344

20.6.2 下降沿和正向死区

在半桥模式下，下降沿死区在CWGx数据输入的下降沿延迟CWGxB输出的导通时间。在全桥模式下，仅在方向从反向模式变化为正向模式时才插入正向死区延时，并且仅调制输出CWGxD受到影响。

CWGxDBF寄存器用于确定输入源信号下降沿死区时间间隔的持续时间。该持续时间为0至64个CWG时钟周期。

死区总是在输入源信号的边沿启动。计数为0表示不存在死区。

如果输入源信号在完成死区计数前反转极性，则相应输出上将没有输出信号。

CWGxDBF寄存器值是双重缓冲的。当EN = 0（寄存器20-1）时，会在写入CWGxDBF时装入缓冲区。当EN = 1时，在LD位（寄存器20-1）置1后将在数据输入的第一个下降沿之后的上升沿装入缓冲区。

20.6.3 死区抖动

CWG输入数据信号可能与CWG输入时钟异步，因此每个周期观察到的死区可能出现某些抖动。最大抖动等于1个CWG时钟周期。关于详细信息和示例，请参见公式20-1。

公式20-1: 死区延时计算

$$T_{\text{DEAD-BAND_MIN}} = \frac{1}{F_{\text{CWG_CLOCK}}} \cdot \text{DBx} < 4:0 >$$

$$T_{\text{DEAD-BAND_MAX}} = \frac{1}{F_{\text{CWG_CLOCK}}} \cdot \text{DBx} < 4:0 > + 1$$

$$T_{\text{JITTER}} = T_{\text{DEAD-BAND_MAX}} - T_{\text{DEAD-BAND_MIN}}$$

$$T_{\text{JITTER}} = \frac{1}{F_{\text{CWG_CLOCK}}}$$

$$T_{\text{DEAD-BAND_MAX}} = T_{\text{DEAD-BAND_MIN}} + T_{\text{JITTER}}$$

示例:

$$\text{DBR} < 4:0 > = 0x0A = 10$$

$$F_{\text{CWG_CLOCK}} = 8 \text{ MHz}$$

$$T_{\text{JITTER}} = \frac{1}{8 \text{ MHz}}$$

$$T_{\text{DEAD-BAND_MIN}} = 125 \text{ ns} \cdot 10 = 1.25 \text{ } \mu\text{s}$$

$$T_{\text{DEAD-BAND_MAX}} = 1.25 \text{ } \mu\text{s} + 0.125 \text{ } \mu\text{s} = 1.37 \text{ } \mu\text{s}$$

20.7 自动关断控制

自动关断是一种使用特定改写信号立即改写CWG输出电平，从而安全关断电路的方法。关断状态可以自动清除，也可以一直保持，直到用软件清除。

20.7.1 关断

关断状态可以通过以下两种方法之一进入：

- 软件生成
- 外部输入

SHUTDOWN位指示何时存在关断条件。该位可由软件或硬件置1或清零。

20.7.1.1 由软件产生的关断

将CWGxAS0寄存器的SHUTDOWN位置1可以强制CWG进入关断状态。

在禁止自动重启时，只要SHUTDOWN位置1，就会一直保持关断状态。

在使能自动重启时，SHUTDOWN位会自动清零，并在发生下一个上升沿事件时继续工作。

20.7.1.2 外部输入源关断

可选择任意自动关断外部输入来暂停CWG操作。这些输入源通过CWGxAS1寄存器（寄存器20-7）的ASxE位单独使能。当选定的任意输入变为有效（引脚为低电平有效）时，CWG输出会立即变为LSBD<1:0>和LSAC<1:0>位选定的改写电平，无任何软件延时（第20.7.1.3节“引脚改写电平”）。可以选择以下任一外部输入源来产生关断条件：

- 比较器C1
- 比较器C2
- CLC2
- CWGxPPS

注： 关断输入是电平敏感的，而不是边沿敏感的。只要关断输入电平仍然存在，除非禁止自动关断，否则无法清除关断状态。

20.7.1.3 引脚改写电平

在发生自动关断事件期间驱动到CWG输出的电平通过CWGxAS0寄存器（寄存器20-6）的LSBD<1:0>和LSAC<1:0>位进行控制。LSBD<1:0>位控制CWGxB/D输出电平，而LSAC<1:0>位控制CWGxA/C输出电平。

20.7.1.4 自动关断中断

当发生自动关断事件时，由软件或硬件将SHUTDOWN置1，PIR4寄存器（寄存器8-11）的CWGxIF标志位置1。

20.8 自动关断重启

在发生自动关断事件之后，可以使用两种方法来恢复工作：

- 软件控制
- 自动重启

在任一情况下，在进行重启之前必须清除关断源。即，必须除去关断条件，或者必须清零相应的WGASxE位。

20.8.1 软件控制重启

当CWGxASD0寄存器的REN位清零（REN = 0）时，在自动关断事件之后，必须用软件重启CWGx模块。

一旦除去所有自动关断源后，软件必须清零SHUTDOWN。SHUTDOWN清零后，CWG模块将在CWG数据输入的第一个上升沿继续工作。

注： 如果自动关断条件仍然存在，无法用软件清零SHUTDOWN位。

20.8.2 自动重启

当CWGxASD0寄存器的REN位置1（REN = 1）时，CWGx模块将从关断状态中自动重启。

一旦除去所有自动关断条件后，硬件将自动清零SHUTDOWN。SHUTDOWN清零后，CWG模块将在CWG数据输入的第一个上升沿继续工作。

注： 如果自动关断条件仍然存在，无法用软件清零SHUTDOWN位。

PIC16(L)F18324/18344

20.9 休眠期间的操作

CWGx模块将在休眠期间工作，前提是输入源保持活动状态。

如果选择HFINTOSC作为模块时钟源，死区生成将保持活动状态。这会直接影响休眠模式的电流。

20.10 配置CWG

1. 确保对应于CWG输出的TRIS控制位置1，从而将所有输出配置为输入，确保输出在设置期间无效。外部硬件应确保引脚电平保持为安全电平。
2. 清零EN位（如果尚未清零）。
3. 配置CWGxCON0寄存器的MODE<2:0>位来设置输出工作模式。
4. 配置CWGxCON1寄存器的POLy位来设置输出极性。
5. 配置CWGxDAT寄存器的DAT<3:0>位来选择数据输入源。
6. 如果选择转向模式，配置STRy位以在CWG输出上选择所需的输出。
7. 配置CWGxAS0寄存器的LSBD<1:0>和LSAC<1:0>位来选择自动关断输出改写状态（即使不使用自动关断这也是必要的，因为启动将从关断状态开始）。
8. 如果需要自动重启，则将CWGxAS0的REN位置1。
9. 如果需要自动关断，配置CWGxAS1寄存器的ASxE位来选择关断源。
10. 使用CWGxDBR和CWGxDBF寄存器设置所需的上升沿和下降沿死区时间。
11. 在CWGxCLKCON寄存器中选择时钟源。
12. 将EN位置1以使能模块。
13. 清零对应于CWG输出的TRIS位以将其设置为输出。
14. 如果要使用自动重启，则将REN位置1，SHUTDOWN位将会自动清零。否则，由软件清零SHUTDOWN位来启动CWG。

20.11 寄存器定义: CWG控制

寄存器 20-1: CWGxCON0: CWGx控制寄存器0

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD ⁽¹⁾	—	—	—	MODE<2:0>		
bit 7						bit 0	

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	HS/HC = 由硬件置1/清零位

- bit 7 **EN:** CWGx使能位
 1 = 使能CWGx
 0 = 禁止CWGx
- bit 6 **LD:** CWG装入缓冲区位⁽¹⁾
 1 = 在紧接着该位置1后第一个下降沿的CWG数据上升沿装入死区计数缓冲区。
 0 = 缓冲区保持不变
- bit 5-3 **未实现:** 读为0
- bit 2-0 **MODE<2:0>:** CWGx模式位
 111 = 保留
 110 = 保留
 101 = CWG输出在推挽模式下工作
 100 = CWG输出在半桥模式下工作
 011 = CWG输出在反向全桥模式下工作
 010 = CWG输出在正向全桥模式下工作
 001 = CWG输出在同步转向模式下工作
 000 = CWG输出在异步转向模式下工作

注 1: 该位只能在EN = 1后置1, 不能在EN置1的同一周期中置1。

PIC16(L)F18324/18344

寄存器 20-2: CWGxCON1: CWGx控制寄存器 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	q = 值取决于具体条件

- bit 7-6 **未实现:** 读为0
- bit 5 **IN:** CWGx数据输入信号 (只读)
- bit 4 **未实现:** 读为0
- bit 3 **POLD:** WGxD输出极性位
1 = 信号输出极性翻转
0 = 信号输出为正常极性
- bit 2 **POLC:** WGxC输出极性位
1 = 信号输出极性翻转
0 = 信号输出为正常极性
- bit 1 **POLB:** WGxB输出极性位
1 = 信号输出极性翻转
0 = 信号输出为正常极性
- bit 0 **POLA:** WGxA输出极性位
1 = 信号输出极性翻转
0 = 信号输出为正常极性

寄存器 20-3: CWGxCLKCON: CWGx时钟输入选择寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	CS
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	q = 值取决于具体条件

- bit 7-1 **未实现:** 读为0
- bit 0 **CS:** CWG时钟源选择位

CS	时钟源
0	Fosc
1	HFINTOSC (在休眠期间继续工作)

PIC16(L)F18324/18344

寄存器 20-4: CWGxDAT: CWGx数据输入选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	DAT<3:0>			
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7-4 **未实现:** 读为0

bit 3-0 **DAT<3:0>:** CWG 数据输入选择位

DAT	数据源
0000	CWGxPPS
0001	C1OUT
0010	C2OUT
0011	CCP1
0100	CCP2
0101	CCP3
0110	CCP4
0111	PWM5
1000	PWM6
1001	NCO1
1010	CLC1
1011	CLC2
1100	CLC3
1101	CLC4
1110	保留
1111	保留

PIC16(L)F18324/18344

寄存器 20-5: **CWGxSTR⁽¹⁾**: CWG转向控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OVRD	OVRC	OVRB	OVRA	STRD ⁽²⁾	STRC ⁽²⁾	STRB ⁽²⁾	STRA ⁽²⁾
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

- bit 7 **OVRD**: 转向数据D位
- bit 6 **OVRC**: 转向数据C位
- bit 5 **OVRB**: 转向数据B位
- bit 4 **OVRA**: 转向数据A位
- bit 3 **STRD**: 转向使能位D⁽²⁾
1 = CWGxD输出具有CWGx数据输入波形, 其极性由POLD位控制
0 = CWGxD输出指定为OVRD位的值
- bit 2 **STRC**: 转向使能位C⁽²⁾
1 = CWGxC输出具有CWGx数据输入波形, 其极性由POLC位控制
0 = CWGxC输出指定为OVRC位的值
- bit 1 **STRB**: 转向使能位B⁽²⁾
1 = CWGxB输出具有CWGx数据输入波形, 其极性由POLB位控制
0 = CWGxB输出指定为OVRB位的值
- bit 0 **STRA**: 转向使能位A⁽²⁾
1 = CWGxA输出具有CWGx数据输入波形, 其极性由POLA位控制
0 = CWGxA输出指定为OVRA位的值

注 1: 该寄存器中的位仅在MODE<2:0> = 00x (寄存器20-1, 转向模式) 时适用。

2: MODE<2:0> = 001时, 该位是双重缓冲的。

寄存器 20-6: CWGxAS0: CWG 自动关断控制寄存器 0

R/W/HS/SC-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	U-0	U-0
SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7 **SHUTDOWN:** 自动关断事件状态位^(1,2)

1 = 自动关断状态有效
0 = 未发生自动关断事件

bit 6 **REN:** 自动重启使能位

1 = 使能自动重启
0 = 禁止自动重启

bit 5-4 **LSBD<1:0>:** CWGxB 和 CWGxD 自动关断状态控制位

11 = 发生自动关断事件时, 将逻辑1放置在 CWGxB/D 上。
10 = 发生自动关断事件时, 将逻辑0放置在 CWGxB/D 上。
01 = 发生自动关断事件时, CWGxB/D 上的引脚处于三态。
00 = 发生自动关断事件时, 在所需的死区时间间隔之后, 将引脚的无效状态 (包括极性) 放置在 CWGxB/D 上。

bit 3-2 **LSAC<1:0>:** CWGxA 和 CWGxC 自动关断状态控制位

11 = 发生自动关断事件时, 将逻辑1放置在 CWGxA/C 上。
10 = 发生自动关断事件时, 将逻辑0放置在 CWGxA/C 上。
01 = 发生自动关断事件时, CWGxA/C 上的引脚处于三态。
00 = 发生自动关断事件时, 在所需的死区时间间隔之后, 将引脚的无效状态 (包括极性) 放置在 CWGxA/C 上。

bit 1-0 **未实现:** 读为0

注 1: 在 EN = 0 (寄存器 20-1) 时, 可以写入该位, 将输出置为关断配置。

2: 输出将一直保持在自动关断状态, 直到该位清零后出现 CWG 数据输入的下一个上升沿为止。

PIC16(L)F18324/18344

寄存器 20-7: CWGxAS1: CWG 自动关断控制寄存器 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	q = 值取决于具体条件

- bit 7-5 **未实现:** 读为0
- bit 4 **AS4E:** CWG 自动关断源4 (CLC4) 使能位
1 = 使能CLC4的自动关断
0 = 禁止CLC4的自动关断
- bit 3 **AS3E:** CWG 自动关断源3 (CLC2) 使能位
1 = 使能CLC2的自动关断
0 = 禁止CLC2的自动关断
- bit 2 **AS2E:** CWG 自动关断源2 (C2) 使能位
1 = 使能比较器2的自动关断
0 = 禁止比较器2的自动关断
- bit 1 **AS1E:** CWG 自动关断源1 (C1) 使能位
1 = 使能比较器1的自动关断
0 = 禁止比较器1的自动关断
- bit 0 **AS0E:** CWG 自动关断源0 (CWGxPPS) 使能位
1 = 使能CWGxPPS的自动关断
0 = 禁止CWGxPPS的自动关断

寄存器 20-8: CWGxDBR: CWGx上升沿死区计数寄存器

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	DBR<5:0>					
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	q = 值取决于具体条件

- bit 7-6 **未实现:** 读为0
- bit 5-0 **DBR<5:0>:** CWG 上升沿触发死区计数位
11 1111 = 63-64个CWG时钟周期
11 1110 = 62-63个CWG时钟周期
.
.
.
00 0010 = 2-3个CWG时钟周期
00 0001 = 1-2个CWG时钟周期
00 0000 = 0个CWG时钟周期。死区生成被旁路。

PIC16(L)F18324/18344

寄存器 20-9: CWGxDBF: CWGx下降沿死区计数寄存器

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	DBF<5:0>					
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7-6

未实现: 读为0

bit 5-0

DBF<5:0>: CWG下降沿触发死区计数位

11 1111 = 63-64个CWG时钟周期

11 1110 = 62-63个CWG时钟周期

.

.

.

00 0010 = 2-3个CWG时钟周期

00 0001 = 1-2个CWG时钟周期

00 0000 = 0个CWG时钟周期。死区生成被旁路。

PIC16(L)F18324/18344

表20-2: 与CWGx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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	109
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	104
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			211
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	212
CWG1CLKCON	—	—	—	—	—	—	—	CS	212
CWG1DAT	—	—	—	—	DAT<3:0>				213
CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	214
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	215
CWG1AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	216
CWG1DBR	—	—	DBR<5:0>						216
CWG1DBF	—	—	DBF<5:0>						217
CWG1PPS	—	—	—	CWG1PPS<4:0>					160
CWG2CON0	EN	LD	—	—	—	MODE<2:0>			211
CWG2CON1	—	—	IN	—	POLD	POLC	POLB	POLA	212
CWG2CLKCON	—	—	—	—	—	—	—	CS	212
CWG2DAT	—	—	—	—	DAT<3:0>				213
CWG2STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	214
CWG2AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	215
CWG2AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	216
CWG2DBR	—	—	DBR<5:0>						216
CWG2DBF	—	—	DBF<5:0>						217
CWG2PPS	—	—	—	CWG2PPS<4:0>					160

图注: — = 未实现位, 读为 0。CWGx 模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为 0。

21.0 可配置逻辑单元 (CLC)

可配置逻辑单元 (CLCx) 提供可超越软件执行速度限制而工作的可编程逻辑。该逻辑单元最多可接收 36 个输入信号，并通过使用可配置门将 36 个输入缩减为 4 条驱动 8 种可选单输出逻辑功能之一的逻辑线。

输入源是以下信号源的组合：

- I/O 引脚
- 内部时钟
- 外设
- 寄存器位

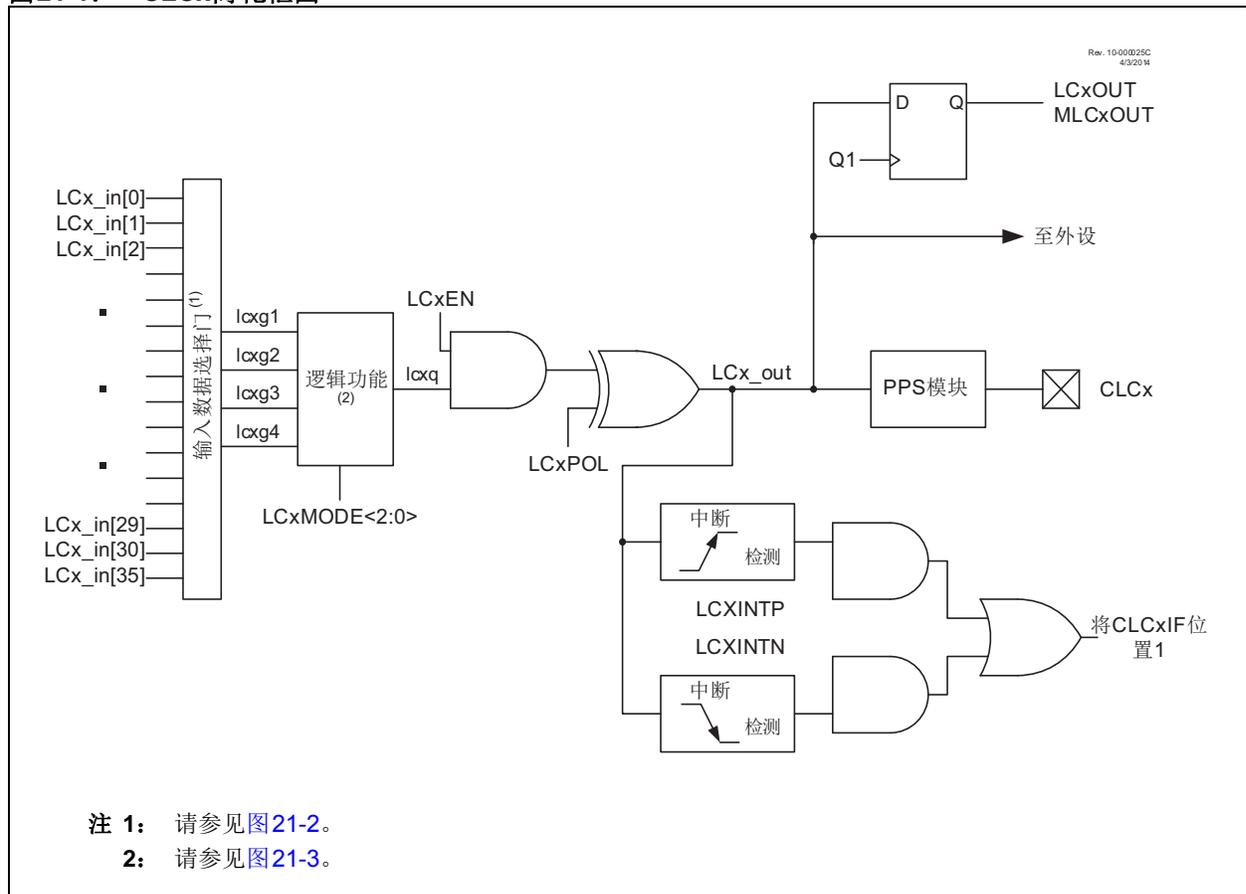
可将输出内部连接到外设和输出引脚。

关于说明通过 CLCx 的信号流的简化框图，请参见图 21-1。

可能的配置包括：

- 组合逻辑
 - AND
 - NAND
 - AND-OR
 - AND-OR-INVERT
 - OR-XOR
 - OR-XNOR
- 锁存器
 - S-R
 - 带置 1 和复位功能的时钟控制 D 型锁存器
 - 带置 1 和复位功能的透明 D 型锁存器
 - 带复位功能的时钟控制 J-K 型锁存器

图 21-1: CLCx 简化框图



PIC16(L)F18324/18344

21.1 CLCx 设置

CLCx 模块的编程通过配置逻辑信号流中的 4 级来实现。这 4 级为：

- 数据选择
- 数据门
- 逻辑功能选择
- 输出极性

每级都可在运行时通过写入相应的 CLCx 特殊功能寄存器来进行设置。这具有支持在程序执行期间运行时执行逻辑重新配置的额外优点。

21.1.1 数据选择

有 36 个信号可用作可配置逻辑的输入。

数据选择通过图 21-2 左侧所示的 4 个多路开关来进行。图中的数据输入使用 LCx_in 信号名称来标识。

表 21-1 列出了每个 CLC 模块的输入编号与实际信号的关联。标有“LCxDyS<5:0>值”的列给出了选定数据输入的多路开关选择代码。LCxDyS 是用于标识特定输入多路开关的缩写：LCxD1S<5:0> 至 LCxD4S<5:0>。

数据输入使用 CLCxSEL0 至 CLCxSEL3 寄存器（寄存器 21-3 至寄存器 21-6）进行选择。

注： 数据选择在上电时是未定义的。

表 21-1: CLCx 数据输入选择

LCxDyS<5:0>值	CLCx 输入源
100011 [35]	TMR6/PR6 匹配
100010 [34]	TMR5 溢出
100001 [33]	TMR4/PR4 匹配
100000 [32]	TMR3 溢出
111111 [31]	Fosc
111110 [30]	HFINTOSC
111101 [29]	LFINTOSC
111100 [28]	ADCRC
110111 [27]	IOCIF 中断标志位
110110 [26]	TMR2/PR2 匹配
110101 [25]	TMR1 溢出
110100 [24]	TMR0 溢出
101111 [23]	EUSART1 (DT) 输出
101110 [22]	EUSART1 (TX/CK) 输出
101101 [21]	保留
101100 [20]	保留
100111 [19]	SDA1
100110 [18]	SCL1
100011 [17]	PWM6 输出
100010 [16]	PWM5 输出
011111 [15]	CCP4 输出
011110 [14]	CCP3 输出
011101 [13]	CCP2 输出
011100 [12]	CCP1 输出
010111 [11]	CLKR 输出
010110 [10]	DSM 输出
010011 [9]	C2 输出
010010 [8]	C1 输出
001111 [7]	CLC4 输出
001110 [6]	CLC3 输出
001101 [5]	CLC2 输出
001100 [4]	CLC1 输出
000111 [3]	CLCIN3PPS
000110 [2]	CLCIN2PPS
000101 [1]	CLCIN1PPS
000100 [0]	CLCIN0PPS

21.1.2 输入数据选择门

来自输入多路开关的输出将通过数据门级转送到所需的逻辑功能输入。每个数据门可以转送由4个选定输入组成的任意组合。

可将门配置为将每个输入信号转换为反相或同相数据。每个门的输出可以先进行反相，然后再进入逻辑功能级。

门实际上是一个1至4的输入AND/NAND/OR/NOR门。如果将每个输入和输出进行反相，则该门的作用是对所有已使能数据输入进行逻辑或。如果输入和输出不进行反相，则该门的作用是对所有已使能输入进行逻辑与。

表21-2总结了可以通过使用门逻辑选择位和门极性位在门1中获得的基本逻辑。该表列出了具有4个输入变量的逻辑，但每个门可以配置为使用少于4个输入。如果未选择任何输入，则输出将为0或1，具体取决于门输出极性位。

表21-2: 数据门逻辑

CLCxGLSy	LCxGyPOL	门逻辑
0x55	1	4输入AND
0x55	0	4输入NAND
0xAA	1	4输入NOR
0xAA	0	4输入OR
0x00	0	逻辑0
0x00	1	逻辑1

用户可以（但建议不要）同时选择同一输入的正负值。如果这么做，则无论其他输入如何，门的输出都将为0，但可能会出现逻辑毛刺（瞬变引起的脉冲）。如果通道的输出必须为0或1，则建议的方法是将所有门位设置为0，并使用门极性位来设置所需的电平。

数据门控使用如下门逻辑选择寄存器进行配置：

- 门1: CLCxGLS0 (寄存器21-7)
- 门2: CLCxGLS1 (寄存器21-8)
- 门3: CLCxGLS2 (寄存器21-9)
- 门4: CLCxGLS3 (寄存器21-10)

寄存器编号后缀不同于门编号，这是因为该模块的其他形式在同一寄存器中具有多种门选择。

图21-2右侧给出了数据门的图示。其中仅详细说明了一个门。其余三个门使用相同的配置，只是数据使能对应于该门的使能信号。

21.1.3 逻辑功能

有8种可用的逻辑功能，包括：

- AND-OR
- OR-XOR
- AND
- S-R型锁存器
- 带置1和复位功能的D型触发器
- 带复位功能的D型触发器
- 带复位功能的J-K型触发器
- 带置1和复位功能的透明锁存器

这些逻辑功能如图21-3所示。每种逻辑功能具有4个输入和1个输出。4个输入是上一级的4个数据门输出。输出送到反相级，接着送到其他外设和输出引脚，然后回到CLCx。

21.1.4 输出极性

可配置逻辑单元中的最后一级是输出极性。将CLCxPOL寄存器的LCxPOL位置1时，来自逻辑级的输出信号会进行反相。如果在允许中断时改变极性会导致输出结果的变化而发生中断。

PIC16(L)F18324/18344

21.2 CLCx 中断

如果相应的中断允许位置1，则在CLCx的输出值改变时，将会产生中断。因此，每个CLC中都具有一个上升沿检测器和一个下降沿检测器。

触发其中一个边沿检测器，且其相关的中断允许位置1时，相关PIR3寄存器的CLCxIF位会置1。LCxINTP位用于允许上升沿中断，LCxINTN位用于允许下降沿中断。它们都位于CLCxCON寄存器中。

要完全允许中断，需要将以下位置1：

- PIE3寄存器的CLCXIE位
- CLCxCON寄存器的LCxINTP位（对于上升沿检测）
- CLCxCON寄存器的LCxINTN位（对于下降沿检测）
- INTCON寄存器的PEIE和GIE位

作为中断服务程序的一部分，必须用软件将PIR3寄存器的CLCxIF位清零。如果在清零该标志时检测到另一个边沿，则标志仍然会在序列结束时置1。

21.3 输出镜像副本

所有LCxCON输出位的镜像副本包含在CLCDATA寄存器中。读取该寄存器将同时采样所有CLC的输出。这可以防止由于测试或读取各个CLCxCON寄存器中的LCxOUT位而导致任何时序差错。

21.4 复位的影响

发生复位后，CLCxCON寄存器会清零。所有其他选择和门值保持不变。

21.5 休眠期间的操作

CLC模块独立于系统时钟工作，只要选定的输入源保持活动状态，它就会继续在休眠期间运行。

如果使能了CLC模块，并且选择HFINTOSC作为输入源，则无论所选择的系统时钟源如何，HFINTOSC都会在休眠期间保持活动状态。

即，如果在CLC使能时，同时选择HFINTOSC作为系统时钟和CLC输入源，则在休眠期间CPU会进入空闲状态，而CLC会继续工作，并且HFINTOSC将保持活动状态。

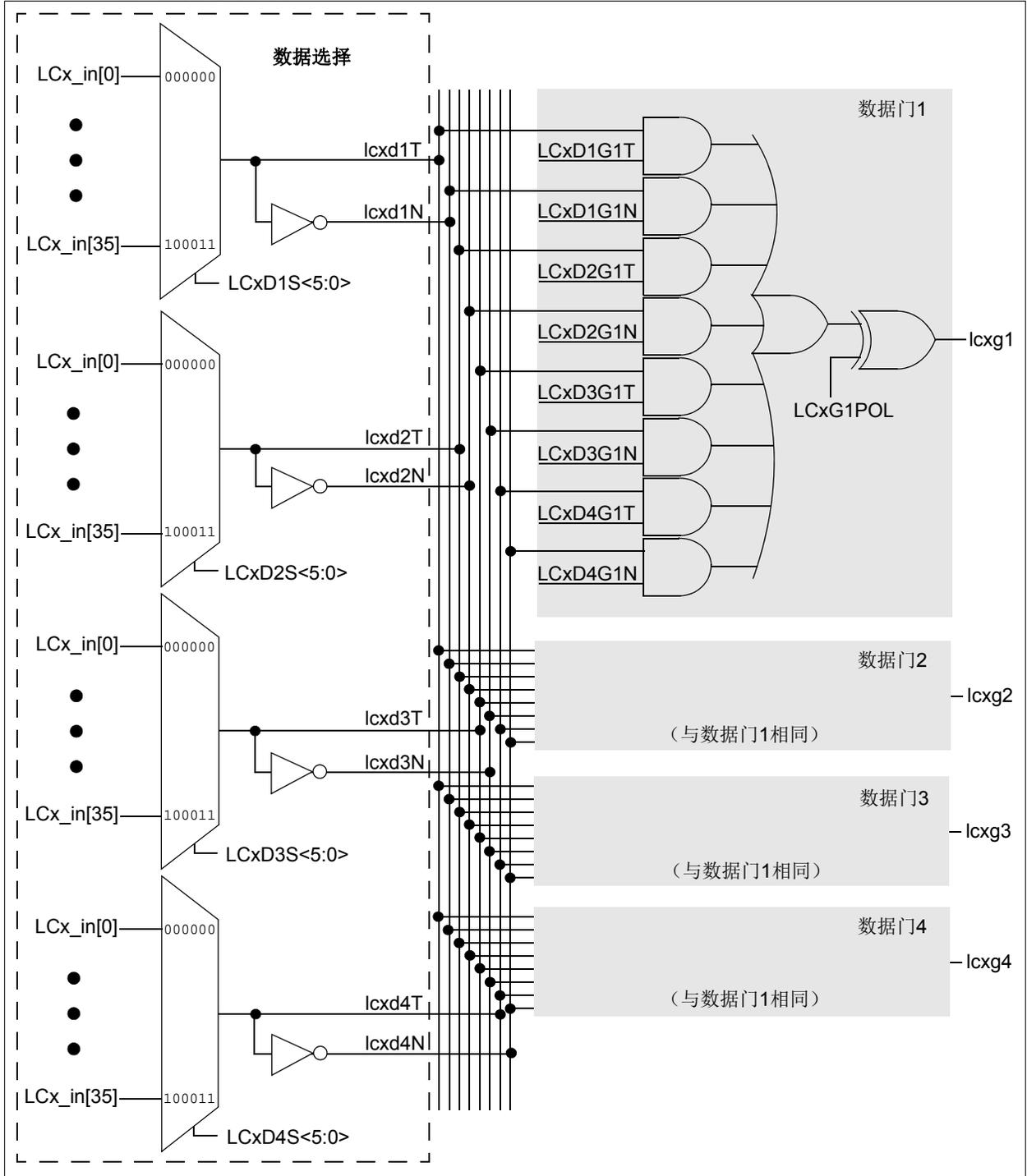
这会直接影响休眠模式的电流。

21.6 CLCx 设置步骤

在设置CLCx时，应遵循以下步骤：

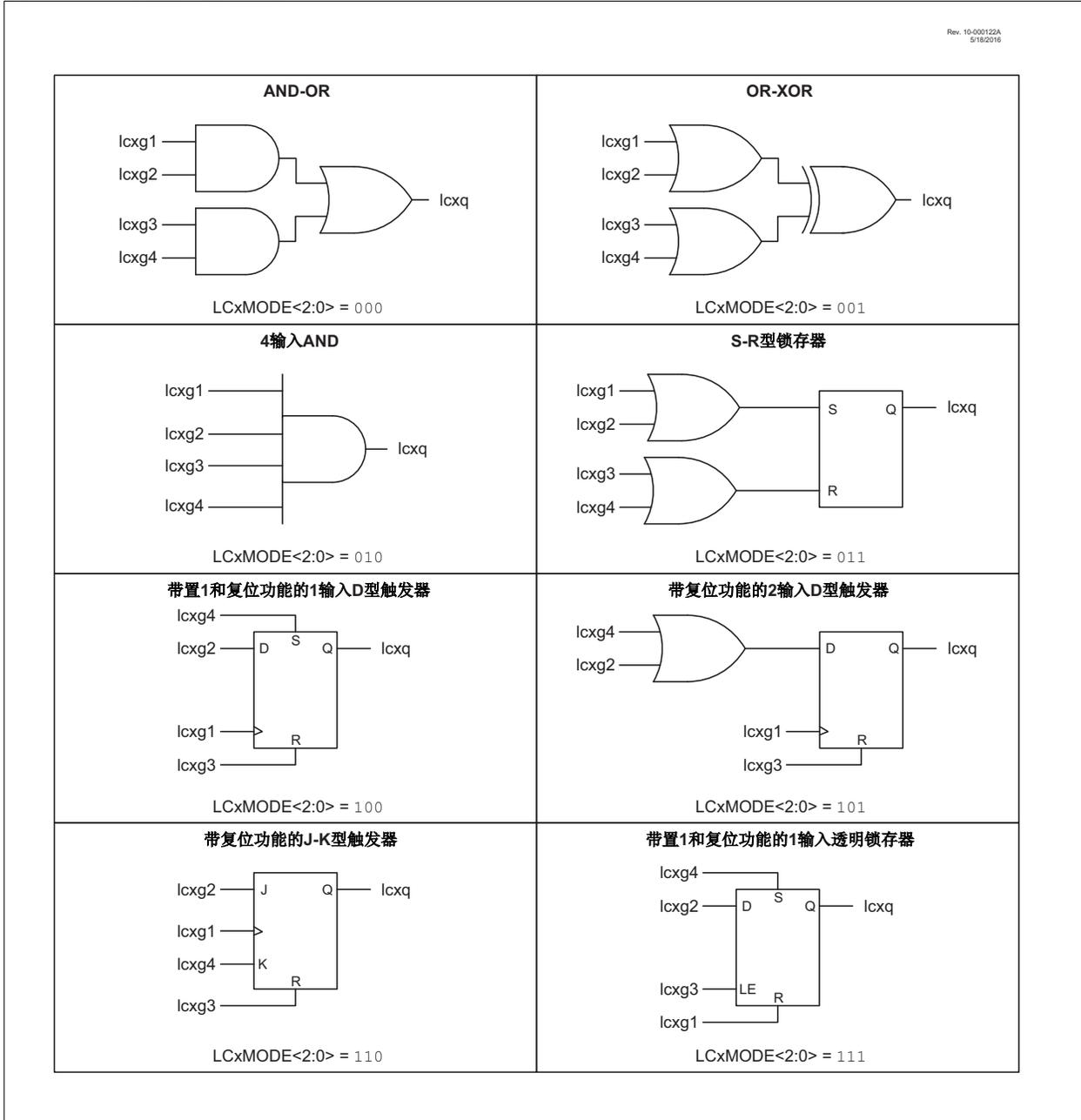
- 通过清零LCxEN位来禁止CLCx。
- 使用CLCxSEL0至CLCxSEL3寄存器选择所需的输入（见表21-1）。
- 清零所有相关的ANSEL位。
- 将与外部CLC输入相关的所有TRIS位置1。
- 使用CLCxGLS0、CLCxGLS1、CLCxGLS2和CLCxGLS3寄存器使能通过4个门的可选输入。
- 使用CLCxPOL寄存器的LCxGyPOL位选择门输出极性。
- 使用CLCxCON寄存器的LCxMODE<2:0>位选择所需的逻辑功能。
- 使用CLCxPOL寄存器的LCxPOL位选择所需的逻辑输出极性。（该步骤可与前面的门输出极性步骤结合）。
- 如果驱动某个器件引脚，则设置所需的引脚PPS控制寄存器，并另外清零对应于该输出的TRIS位。
- 如果需要中断，则配置以下位：
 - 上升沿事件时，将CLCxCON寄存器中的LCxINTP位置1。
 - 下降沿事件时，将CLCxCON寄存器中的LCxINTN位置1。
 - 将PIE3寄存器的CLCXIE位置1。
 - 将INTCON寄存器的GIE和PEIE位置1。
- 通过将CLCxCON寄存器的LCxEN位置1来使能CLCx。

图 21-2: 输入数据选择和门



PIC16(L)F18324/18344

图21-3: 可编程逻辑功能



21.7 寄存器定义：CLC控制

寄存器 21-1: CLCxCON: 可配置逻辑单元控制寄存器

R/W-0/0	U-0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	—	LCxOUT	LCxINTP	LCxINTN	LCxMODE<2:0>		
bit 7						bit 0	

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **LCxEN:** 可配置逻辑单元使能位
1 = 使能可配置逻辑单元, 并混合输入信号
0 = 禁止可配置逻辑单元, 并输出逻辑0
- bit 6 **未实现:** 读为0
- bit 5 **LCxOUT:** 可配置逻辑单元数据输出位
只读: 经过LCPOL之后的逻辑单元输出数据; 从CLCxOUT采样。
- bit 4 **LCxINTP:** 可配置逻辑单元上升边沿中断允许位
1 = CLCxIF将在CLCxOUT上出现上升沿时置1
0 = CLCxIF不会置1
- bit 3 **LCxINTN:** 可配置逻辑单元下降边沿中断允许位
1 = CLCxIF将在CLCxOUT上出现下降沿时置1
0 = CLCxIF不会置1
- bit 2-0 **LCxMODE<2:0>:** 可配置逻辑单元功能模式位
111 = 单元是带置1和复位功能的1输入透明锁存器
110 = 单元是带复位功能的J-K型触发器
101 = 单元是带复位功能的2输入D型触发器
100 = 单元是带置1和复位功能的1输入D型触发器
011 = 单元是S-R型锁存器
010 = 单元是4输入AND逻辑
001 = 单元是OR-XOR逻辑
000 = 单元是AND-OR逻辑

PIC16(L)F18324/18344

寄存器 21-2: **CLCxPOL: 信号极性控制寄存器**

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **LCxPOL:** CLCxOUT 输出极性控制位
1 = 逻辑单元的输出反相
0 = 逻辑单元的输出不反相
- bit 6-4 **未实现:** 读为0
- bit 3 **LCxG4POL:** 门3输出极性控制位
1 = 门3的输出在施加到逻辑单元时反相
0 = 门3的输出不反相
- bit 2 **LCxG3POL:** 门2输出极性控制位
1 = 门2的输出在施加到逻辑单元时反相
0 = 门2的输出不反相
- bit 1 **LCxG2POL:** 门1输出极性控制位
1 = 门1的输出在施加到逻辑单元时反相
0 = 门1的输出不反相
- bit 0 **LCxG1POL:** 门0输出极性控制位
1 = 门0的输出在施加到逻辑单元时反相
0 = 门0的输出不反相

PIC16(L)F18324/18344

寄存器 21-3: CLCxSEL0: 通用CLCx数据0选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD1S<5:0>					
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 未实现: 读为0

bit 5-0 **LCxD1S<5:0>**: CLCx数据1输入选择位
请参见表21-1。

寄存器 21-4: CLCxSEL1: 通用CLCx数据1选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD2S<5:0>					
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 未实现: 读为0

bit 5-0 **LCxD2S<5:0>**: CLCx数据2输入选择位
请参见表21-1。

寄存器 21-5: CLCxSEL2: 通用CLCx数据2选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD3S<5:0>					
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 未实现: 读为0

bit 5-0 **LCxD3S<5:0>**: CLCx数据3输入选择位
请参见表21-1。

PIC16(L)F18324/18344

寄存器 21-6: CLCxSEL3: 通用CLCx数据3选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD4S<5:0>					
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

bit 7-6 **未实现:** 读为0
bit 5-0 **LCxD4S<5:0>:** CLCx数据4输入选择位
请参见表21-1。

寄存器 21-7: CLCxGLS0: 门0逻辑选择寄存器

R/W-x/u							
LCxG1D4T	LCxG1D4N	LCxG1D3T	LCxG1D3N	LCxG1D2T	LCxG1D2N	LCxG1D1T	LCxG1D1N
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

bit 7 **LCxG1D4T:** 门0数据4真值(不反相)位
1 = CLCIN3 (不反相) 通过门输入到CLCx门0
0 = CLCIN3 (不反相) 不通过门输入到CLCx门0

bit 6 **LCxG1D4N:** 门0数据4取反(反相)位
1 = CLCIN3 (反相) 通过门输入到CLCx门0
0 = CLCIN3 (反相) 不通过门输入到CLCx门0

bit 5 **LCxG1D3T:** 门0数据3真值(不反相)位
1 = CLCIN2 (不反相) 通过门输入到CLCx门0
0 = CLCIN2 (不反相) 不通过门输入到CLCx门0

bit 4 **LCxG1D3N:** 门0数据3取反(反相)位
1 = CLCIN2 (反相) 通过门输入到CLCx门0
0 = CLCIN2 (反相) 不通过门输入到CLCx门0

bit 3 **LCxG1D2T:** 门0数据2真值(不反相)位
1 = CLCIN1 (不反相) 通过门输入到CLCx门0
0 = CLCIN1 (不反相) 不通过门输入到CLCx门0

bit 2 **LCxG1D2N:** 门0数据2取反(反相)位
1 = CLCIN1 (反相) 通过门输入到CLCx门0
0 = CLCIN1 (反相) 不通过门输入到CLCx门0

bit 1 **LCxG1D1T:** 门0数据1真值(不反相)位
1 = CLCIN0 (不反相) 通过门输入到CLCx门0
0 = CLCIN0 (不反相) 不通过门输入到CLCx门0

bit 0 **LCxG1D1N:** 门0数据1取反(反相)位
1 = CLCIN0 (反相) 通过门输入到CLCx门0
0 = CLCIN0 (反相) 不通过门输入到CLCx门0

PIC16(L)F18324/18344

寄存器 21-8: CLCxGLS1: 门1逻辑选择寄存器

| R/W-x/u |
|----------|----------|----------|----------|----------|----------|----------|----------|
| LCxG2D4T | LCxG2D4N | LCxG2D3T | LCxG2D3N | LCxG2D2T | LCxG2D2N | LCxG2D1T | LCxG2D1N |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位
u = 不变
1 = 置1

W = 可写位
x = 未知
0 = 清零

U = 未实现位, 读为0
-n/n = POR和BOR时的值/所有其他复位时的值

- bit 7 **LCxG2D4T:** 门1数据4真值（不反相）位
1 = CLCIN3（不反相）通过门输入到CLCx门1
0 = CLCIN3（不反相）不通过门输入到CLCx门1
- bit 6 **LCxG2D4N:** 门1数据4取反（反相）位
1 = CLCIN3（反相）通过门输入到CLCx门1
0 = CLCIN3（反相）不通过门输入到CLCx门1
- bit 5 **LCxG2D3T:** 门1数据3真值（不反相）位
1 = CLCIN2（不反相）通过门输入到CLCx门1
0 = CLCIN2（不反相）不通过门输入到CLCx门1
- bit 4 **LCxG2D3N:** 门1数据3取反（反相）位
1 = CLCIN2（反相）通过门输入到CLCx门1
0 = CLCIN2（反相）不通过门输入到CLCx门1
- bit 3 **LCxG2D2T:** 门1数据2真值（不反相）位
1 = CLCIN1（不反相）通过门输入到CLCx门1
0 = CLCIN1（不反相）不通过门输入到CLCx门1
- bit 2 **LCxG2D2N:** 门1数据2取反（反相）位
1 = CLCIN1（反相）通过门输入到CLCx门1
0 = CLCIN1（反相）不通过门输入到CLCx门1
- bit 1 **LCxG2D1T:** 门1数据1真值（不反相）位
1 = CLCIN0（不反相）通过门输入到CLCx门1
0 = CLCIN0（不反相）不通过门输入到CLCx门1
- bit 0 **LCxG2D1N:** 门1数据1取反（反相）位
1 = CLCIN0（反相）通过门输入到CLCx门1
0 = CLCIN0（反相）不通过门输入到CLCx门1

PIC16(L)F18324/18344

寄存器 21-9: CLCxGLS2: 门2逻辑选择寄存器

| R/W-x/u |
|----------|----------|----------|----------|----------|----------|----------|----------|
| LCxG3D4T | LCxG3D4N | LCxG3D3T | LCxG3D3N | LCxG3D2T | LCxG3D2N | LCxG3D1T | LCxG3D1N |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **LCxG3D4T:** 门2数据4真值(不反相)位
 1 = CLCIN3 (不反相) 通过门输入到CLCx门2
 0 = CLCIN3 (不反相) 不通过门输入到CLCx门2
- bit 6 **LCxG3D4N:** 门2数据4取反(反相)位
 1 = CLCIN3 (反相) 通过门输入到CLCx门2
 0 = CLCIN3 (反相) 不通过门输入到CLCx门2
- bit 5 **LCxG3D3T:** 门2数据3真值(不反相)位
 1 = CLCIN2 (不反相) 通过门输入到CLCx门2
 0 = CLCIN2 (不反相) 不通过门输入到CLCx门2
- bit 4 **LCxG3D3N:** 门2数据3取反(反相)位
 1 = CLCIN2 (反相) 通过门输入到CLCx门2
 0 = CLCIN2 (反相) 不通过门输入到CLCx门2
- bit 3 **LCxG3D2T:** 门2数据2真值(不反相)位
 1 = CLCIN1 (不反相) 通过门输入到CLCx门2
 0 = CLCIN1 (不反相) 不通过门输入到CLCx门2
- bit 2 **LCxG3D2N:** 门2数据2取反(反相)位
 1 = CLCIN1 (反相) 通过门输入到CLCx门2
 0 = CLCIN1 (反相) 不通过门输入到CLCx门2
- bit 1 **LCxG3D1T:** 门2数据1真值(不反相)位
 1 = CLCIN0 (不反相) 通过门输入到CLCx门2
 0 = CLCIN0 (不反相) 不通过门输入到CLCx门2
- bit 0 **LCxG3D1N:** 门2数据1取反(反相)位
 1 = CLCIN0 (反相) 通过门输入到CLCx门2
 0 = CLCIN0 (反相) 不通过门输入到CLCx门2

PIC16(L)F18324/18344

寄存器 21-10: CLCxGLS3: 门3逻辑选择寄存器

| R/W-x/u |
|----------|----------|----------|----------|----------|----------|----------|----------|
| LCxG4D4T | LCxG4D4N | LCxG4D3T | LCxG4D3N | LCxG4D2T | LCxG4D2N | LCxG4D1T | LCxG4D1N |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位
u = 不变
1 = 置1

W = 可写位
x = 未知
0 = 清零

U = 未实现位, 读为0
-n/n = POR和BOR时的值/所有其他复位时的值

- bit 7 **LCxG4D4T:** 门3数据4真值（不反相）位
1 = CLCIN3（不反相）通过门输入到CLCx门3
0 = CLCIN3（不反相）不通过门输入到CLCx门3
- bit 6 **LCxG4D4N:** 门3数据4取反（反相）位
1 = CLCIN3（反相）通过门输入到CLCx门3
0 = CLCIN3（反相）不通过门输入到CLCx门3
- bit 5 **LCxG4D3T:** 门3数据3真值（不反相）位
1 = CLCIN2（不反相）通过门输入到CLCx门3
0 = CLCIN2（不反相）不通过门输入到CLCx门3
- bit 4 **LCxG4D3N:** 门3数据3取反（反相）位
1 = CLCIN2（反相）通过门输入到CLCx门3
0 = CLCIN2（反相）不通过门输入到CLCx门3
- bit 3 **LCxG4D2T:** 门3数据2真值（不反相）位
1 = CLCIN1（不反相）通过门输入到CLCx门3
0 = CLCIN1（不反相）不通过门输入到CLCx门3
- bit 2 **LCxG4D2N:** 门3数据2取反（反相）位
1 = CLCIN1（反相）通过门输入到CLCx门3
0 = CLCIN1（反相）不通过门输入到CLCx门3
- bit 1 **LCxG4D1T:** 门3数据1真值（不反相）位
1 = CLCIN0（不反相）通过门输入到CLCx门3
0 = CLCIN0（不反相）不通过门输入到CLCx门3
- bit 0 **LCxG4D1N:** 门3数据1取反（反相）位
1 = CLCIN0（反相）通过门输入到CLCx门3
0 = CLCIN0（反相）不通过门输入到CLCx门3

PIC16(L)F18324/18344

寄存器 21-11: CLCDATA: CLC数据输出

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7-4 未实现: 读为0
- bit 3 **MLC4OUT**: LC4OUT的镜像副本位
- bit 2 **MLC3OUT**: LC3OUT的镜像副本位
- bit 1 **MLC2OUT**: LC2OUT的镜像副本位
- bit 0 **MLC1OUT**: LC1OUT的镜像副本位

表21-3: 与CLCx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	寄存器所在页
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	108
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	103
CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			225
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	226
CLC1SEL0	—	—	LC1D1S<5:0>						227
CLC1SEL1	—	—	LC1D2S<5:0>						227
CLC1SEL2	—	—	LC1D3S<5:0>						227
CLC1SEL3	—	—	LC1D4S<5:0>						228
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	228
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	229
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	230
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	231
CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			225
CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	226
CLC2SEL0	—	—	LC2D1S<5:0>						227
CLC2SEL1	—	—	LC2D2S<5:0>						227
CLC2SEL2	—	—	LC2D3S<5:0>						227
CLC2SEL3	—	—	LC2D4S<5:0>						228

图注: — = 未实现, 读为0。CLC 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F18344。

PIC16(L)F18324/18344

表21-3: 与CLCx相关的寄存器汇总 (续)

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页	
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	228	
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	229	
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	230	
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	231	
CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			225	
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	226	
CLC3SEL0	—	—	LC3D1S<5:0>						227	
CLC3SEL1	—	—	LC3D2S<5:0>						227	
CLC3SEL2	—	—	LC3D3S<5:0>						227	
CLC3SEL3	—	—	LC3D4S<5:0>						228	
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	228	
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	229	
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	230	
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	231	
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			225	
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	226	
CLC4SEL0	—	—	LC4D1S<5:0>						227	
CLC4SEL1	—	—	LC4D2S<5:0>						227	
CLC4SEL2	—	—	LC4D3S<5:0>						227	
CLC4SEL3	—	—	LC4D4S<5:0>						228	
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	228	
CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	229	
CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	230	
CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	231	
CLCDATA	—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	232	
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>						160
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>						160
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>						160
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>						160
CLC1OUTPPS	—	—	—	CLC1OUTPPS<4:0>						160
CLC2OUTPPS	—	—	—	CLC2OUTPPS<4:0>						160
CLC3OUTPPS	—	—	—	CLC3OUTPPS<4:0>						160
CLC4OUTPPS	—	—	—	CLC4OUTPPS<4:0>						160

图注: — = 未实现, 读为 0。CLC 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F18344。

PIC16(L)F18324/18344

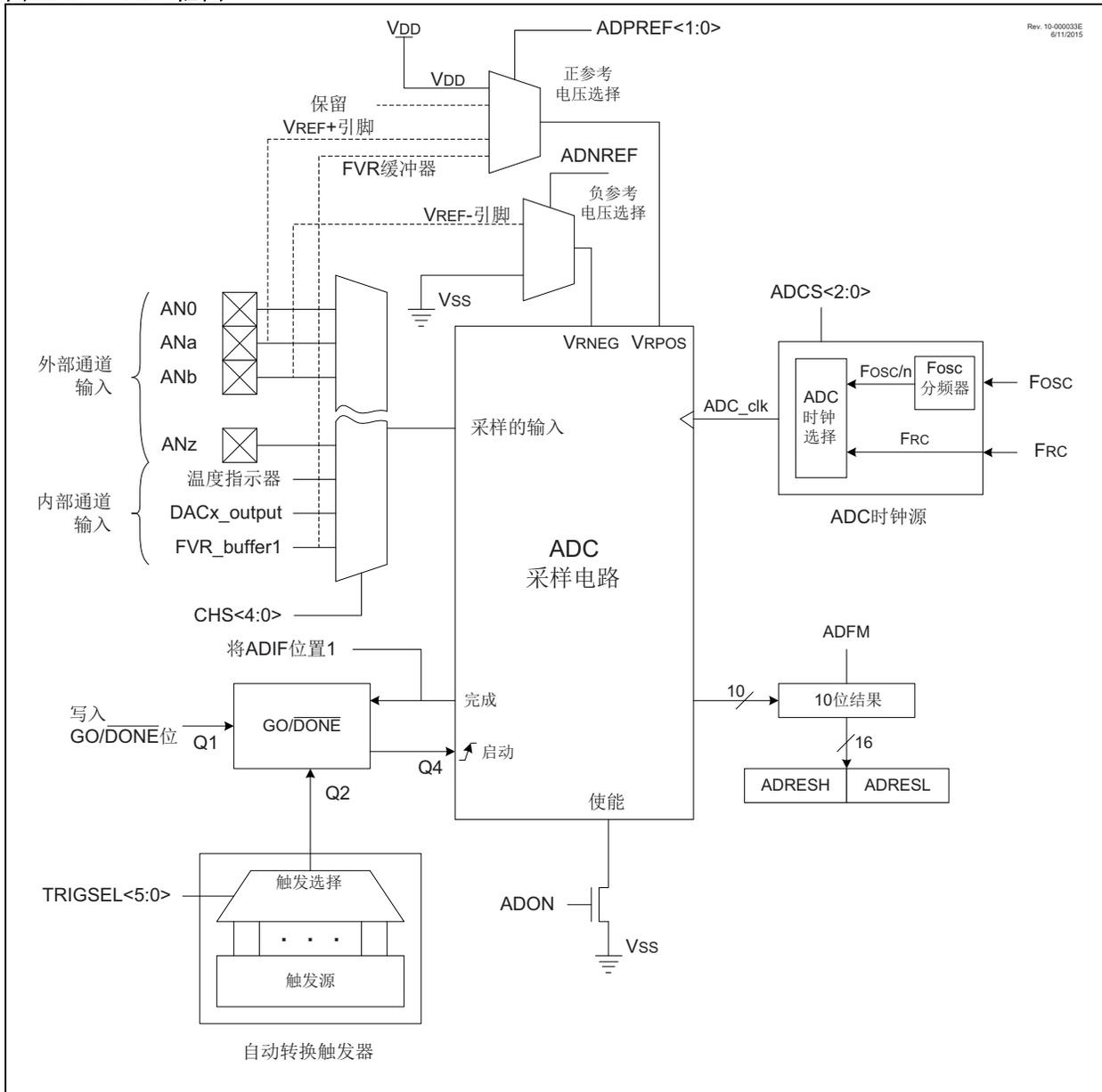
22.0 模数转换器 (ADC) 模块

模数转换器 (ADC) 可将模拟输入信号转换为信号的 10 位二进制表示。该模块使用模拟输入，这些输入通过多路开关连接到同一个采样和保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制结果，并将转换结果存储在 ADC 结果寄存器 (ADRESH:ADRESL 寄存器对) 中。图 22-1 给出了 ADC 的框图。

可通过软件选择内部产生的电压或外部提供的电压作为 ADC 参考电压。

ADC 可在转换完成时产生中断。该中断可用于将器件从休眠模式唤醒。

图 22-1: ADC 框图



22.1 ADC配置

配置和使用ADC时必须考虑以下功能：

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

22.1.1 端口配置

ADC可用于转换模拟信号和数字信号。转换模拟信号时，应通过设置相关的TRIS和ANSEL位将I/O引脚配置为模拟。更多信息，请参见第12.0节“[I/O端口](#)”。

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

22.1.2 通道选择

有多个通道选择可供使用：

- 5个PORTA引脚（RA0-RA2, RA4-RA5）
- 4个PORTB引脚（RB4-RB7, 仅限PIC16(L)F18344）
- 6个PORTC引脚（RC0-RC5, PIC16(L)F18324）
- 8个PORTC引脚（RC0-RC7, 仅限PIC16(L)F18344）
- 温度指示器
- DAC输出
- 固定参考电压（FVR）
- Vss（地）

ADCON0寄存器（[寄存器22-1](#)）的CHS<5:0>位决定与采样保持电路相连接的通道。

当改变通道时，在开始下一次转换前需要一段延时。更多信息，请参见第22.2节“[ADC工作原理](#)”。

注： 从高压ADC通道切换到低压通道时，建议先通过软件选择Vss通道，然后再切换到低压通道。如果ADC没有专用Vss输入通道，则可使用通过DAC输出通道的Vss选择（DAC1R<4:0> = b'00000'）。如果正在使用DAC，可将空闲输入通道连接到Vss，并且可使用该通道来代替DAC。

22.1.3 ADC参考电压

ADCON1寄存器的ADPREF<1:0>位控制正参考电压。正参考电压可以是：

- VREF+ 引脚
- VDD
- FVR 2.048V
- FVR 4.096V（在LF器件上不可用）

ADCON1寄存器的ADNREF位用于控制负参考电压。负参考电压可以是：

- VREF- 引脚
- Vss

关于固定参考电压的更多详细信息，请参见第16.0节“[固定参考电压（FVR）](#)”。

22.1.4 转换时钟

可通过软件设置ADCON1寄存器的ADCS<2:0>位来选择转换时钟源。有以下7种时钟可供选择：

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- ADCRC（专用RC振荡器）

完成一个位的转换所需的时间定义为TAD。一次完整的10位转换需要12个TAD周期，如[图22-2](#)所示。

为正确转换，必须满足相应的TAD规范。更多信息，请参见[表35-13](#)。[表22-1](#)给出了适当的ADC时钟选择的示例。

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

PIC16(L)F18324/18344

表22-1: ADC时钟周期 (TAD) 与器件工作频率关系表

ADC时钟周期 (TAD)		器件频率 (Fosc)					
ADC时钟源	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	62.5 ns ⁽²⁾	100 ns ⁽²⁾	125 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	2.0 μs
Fosc/4	100	125 ns ⁽²⁾	200 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	1.0 μs	4.0 μs
Fosc/8	001	0.5 μs ⁽²⁾	400 ns ⁽²⁾	0.5 μs ⁽²⁾	1.0 μs	2.0 μs	8.0 μs ⁽³⁾
Fosc/16	101	800 ns	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs ⁽³⁾
Fosc/32	010	1.0 μs	1.6 μs	2.0 μs	4.0 μs	8.0 μs ⁽³⁾	32.0 μs ⁽²⁾
Fosc/64	110	2.0 μs	3.2 μs	4.0 μs	8.0 μs ⁽³⁾	16.0 μs ⁽²⁾	64.0 μs ⁽²⁾
ADCRC	x11	1.0-6.0 μs ^(1,4)					

图注: 阴影单元表示超出了建议范围。

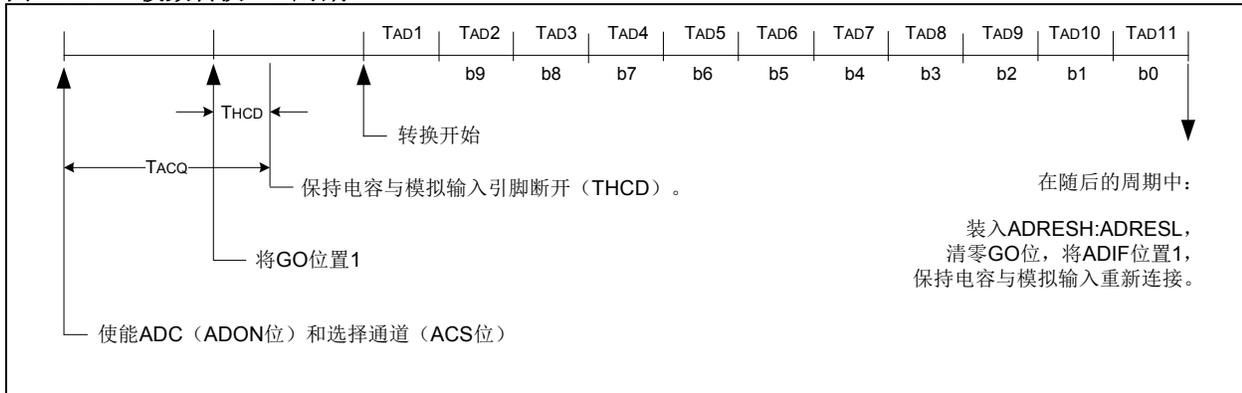
注 1: 关于 ADCRC 时钟源的典型 TAD 值, 请参见 TAD 参数。

2: 这些值违反了所需的 TAD 时间。

3: 超出了建议的 TAD 时间范围。

4: 通过系统时钟 Fosc 来产生 ADC 时钟时, 可以最大程度缩短 ADC 时钟周期 (TAD) 和 ADC 总转换时间。但是, 如果要在器件处于休眠模式时执行转换, 则必须使用 ADCRC 振荡器源。

图22-2: 模数转换 TAD 周期



22.1.5 中断

ADC 模块可在模数转换完成时产生中断。ADC 中断标志位是 PIR1 寄存器中的 ADIF 位。ADC 中断允许位是 PIE1 寄存器中的 ADIE 位。ADIF 位必须用软件清零。

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

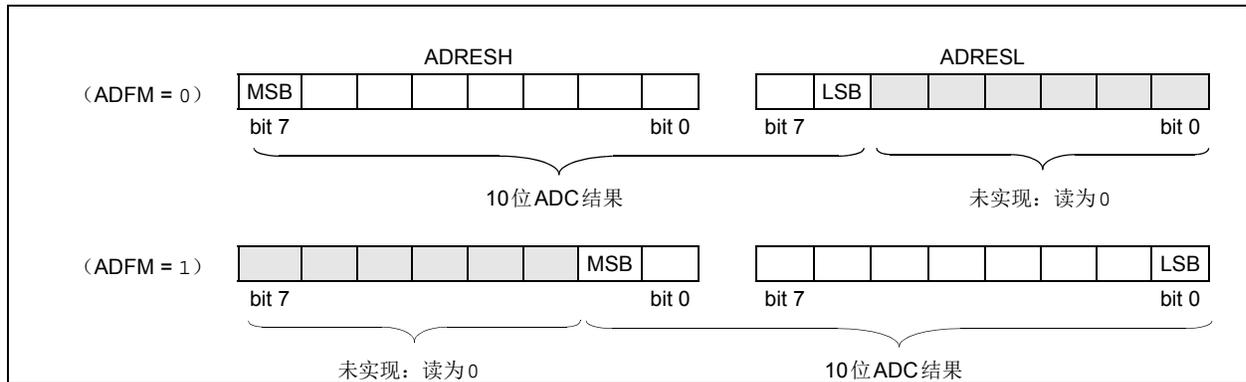
器件工作或休眠时都可产生该中断。如果器件处于休眠模式，该中断会唤醒器件。从休眠模式唤醒时，总是执行紧跟 SLEEP 指令后的下一条指令。如果用户试图从休眠模式唤醒并恢复主代码执行，则必须将 PIE1 寄存器的 ADIE 位和 INTCON 寄存器的 PEIE 位同时置 1，并将 INTCON 寄存器的 GIE 位清零。如果这 3 个位全部置 1，则执行将切换到中断服务程序。

22.1.6 结果格式

10 位 ADC 转换结果可以两种格式提供：左对齐或右对齐。ADCON1 寄存器的 ADFM 位控制输出格式。

图 22-3 给出了两种输出格式。

图 22-3: 10 位 ADC 转换结果格式



PIC16(L)F18324/18344

22.2 ADC工作原理

22.2.1 启动转换

要使能ADC模块，必须将ADCON0寄存器的ADON位设置为1。将ADCON0寄存器的GO/DONE位设置为1会启动模数转换。

注： 不应在启动ADC的同一条指令中将GO/DONE位置1。请参见第22.2.6节“ADC转换步骤”。

22.2.2 转换完成

转换完成时，ADC模块将：

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

22.2.3 终止转换

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

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

22.2.4 休眠期间的ADC操作

ADC模块可以在休眠模式下工作。这需要将ADC时钟源设置为ADCR选项。当选择ADCR振荡器源时，ADC需等待一个额外的指令周期后才能启动转换。这使得可以执行SLEEP指令，这将降低转换期间的系统噪声。如果允许了ADC中断，转换完成时器件将从休眠模式唤醒。如果禁止了ADC中断，尽管ADON位仍保持置1，但转换完成后ADC模块将关闭。

ADC时钟源不是ADCR时，尽管ADON位仍保持置1，但SLEEP指令会导致当前转换中止，ADC模块被关闭。

22.2.5 自动转换触发

自动转换触发器允许定期进行ADC测量而无需软件干预。当出现选定触发源的上升沿时，GO/DONE位由硬件置1。

自动转换触发源使用ADACT寄存器的ADACT<4:0>位进行选择。

关于自动转换源，请参见表22-2。

表22-2： ADC自动转换源

源外设	说明
TMR0	Timer0溢出条件
TMR1	Timer1溢出条件
TMR3	Timer3溢出条件
TMR5	Timer5溢出条件
TMR2	Timer2与PR2匹配
TMR4	Timer4与PR4匹配
TMR6	Timer6与PR6匹配
C1	比较器C1输出
C2	比较器C2输出
CLC1	CLC1输出
CLC2	CLC2输出
CLC3	CLC3输出
CLC4	CLC4输出
CCP1	CCP1输出
CCP2	CCP2输出
CCP3	CCP3输出
CCP4	CCP4输出

注： 当器件处于休眠模式时，ADC自动转换功能不可用。

22.2.6 ADC转换步骤

以下是用ADC执行模数转换的示例步骤：

1. 配置端口：
 - 禁止引脚输出驱动器（见TRIS寄存器）
 - 将引脚配置为模拟功能（见ANSEL寄存器）
2. 配置ADC模块：
 - 选择ADC转换时钟
 - 选择参考电压
 - 选择ADC输入通道
 - 开启ADC模块
3. 配置ADC中断（可选）：
 - 清零ADC中断标志
 - 允许ADC中断
 - 允许外设中断
 - 允许全局中断⁽¹⁾
4. 等待所需采集时间⁽²⁾。
5. 通过将GO/DONE位置1启动转换。
6. 通过以下方式之一等待ADC转换完成：
 - 查询GO/DONE位
 - 等待ADC中断
7. 读取ADC结果。
8. 清零ADC中断标志（如果已允许中断则需要此操作）。

注 1： 如果用户试图从休眠模式唤醒器件并恢复主代码执行，必须禁止全局中断。

2： 请参见第22.3节“ADC采集要求”。

例22-1： ADC转换

```

;This code block configures the ADC
;for polling, Vdd and Vss references, ADCRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion ;
are included.
;
BANKSEL    ADCON1    ;
MOVLW     B'11110000' ;Right justify, ADCRC
;oscillator
MOVWF     ADCON1    ;Vdd and Vss Vref
BANKSEL    TRISA     ;
BSF       TRISA,0   ;Set RA0 to input
BANKSEL    ANSEL     ;
BSF       ANSEL,0   ;Set RA0 to analog
BANKSEL    ADCON0    ;
MOVLW     B'00000001' ;Select channel AN0
MOVWF     ADCON0    ;Turn ADC On
CALL      SampleTime ;Acquisiton delay
BSF       ADCON0,ADGO ;Start conversion
BTFSC    ADCON0,ADGO ;Is conversion done?
GOTO     $-1        ;No, test again
BANKSEL    ADRESH    ;
MOVF     ADRESH,W   ;Read upper 2 bits
MOVWF    RESULTHI   ;store in GPR space
BANKSEL    ADRESL    ;
MOVF     ADRESL,W   ;Read lower 8 bits
MOVWF    RESULTLO   ;Store in GPR space
    
```

PIC16(L)F18324/18344

22.3 ADC采集要求

为了使ADC达到规定的精度，必须使充电保持电容（CHOLD）完全充电至输入通道的电压。模拟输入模型如图22-4所示。模拟信号源阻抗（Rs）和内部采样开关阻抗（Rss）直接影响电容CHOLD的充电时间。采样开关阻抗（Rss）随器件电压（VDD）的变化而变化，参见图22-4。模拟信号源的最大阻抗推荐值为10 kΩ。

采集时间可能随着源阻抗的降低而缩短。在选择（或改变）模拟输入通道后，必须在启动转换前完成ADC采集。可以使用公式22-1来计算最小采集时间。该公式假设误差为1/2 LSB（ADC转换需要1,024步）。1/2 LSB误差是ADC达到规定分辨率所能允许的最大误差。

公式22-1： 采集时间示例

假设： 温度 = 50°C， 外部阻抗为10 kΩ， VDD为5.0V

$$\begin{aligned}T_{ACQ} &= \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2 \mu s + T_C + [(\text{温度} - 25^\circ\text{C}) (0.05 \mu s / ^\circ\text{C})]\end{aligned}$$

Tc值可以用以下公式近似计算：

$$V_{APPLIED} \left(1 - \frac{1}{(2^{n+1}) - 1} \right) = V_{CHOLD} \quad ; [1] \text{充电到} V_{CHOLD} \text{ (1/2 lsb 误差范围)}$$

$$V_{APPLIED} \left(1 - e^{-\frac{T_C}{RC}} \right) = V_{CHOLD} \quad ; [2] \text{响应} V_{APPLIED} \text{ 充电到} V_{CHOLD}$$

$$V_{APPLIED} \left(1 - e^{-\frac{T_C}{RC}} \right) = V_{APPLIED} \left(1 - \frac{1}{(2^{n+1}) - 1} \right) \quad ; \text{合并[1]和[2]}$$

注： 其中n = ADC的位数。

求解Tc：

$$\begin{aligned}T_C &= -\text{CHOLD}(\text{RIC} + \text{Rss} + \text{Rs}) \ln(1/2047) \\ &= -10 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0004885) \\ &= 1.37 \mu s\end{aligned}$$

因此：

$$\begin{aligned}T_{ACQ} &= 2 \mu s + 892 \text{ ns} + [(50^\circ\text{C} - 25^\circ\text{C}) (0.05 \mu s / ^\circ\text{C})] \\ &= 4.62 \mu s\end{aligned}$$

注 1： 因为参考电压（VREF）自行抵消，因此它对该公式没有影响。

2： 充电保持电容（CHOLD）在每次转换后不会放电。

3： 模拟信号源的最大阻抗推荐值为10 kΩ。此要求是为了符合引脚泄漏电流规范。

图22-4: 模拟输入模型

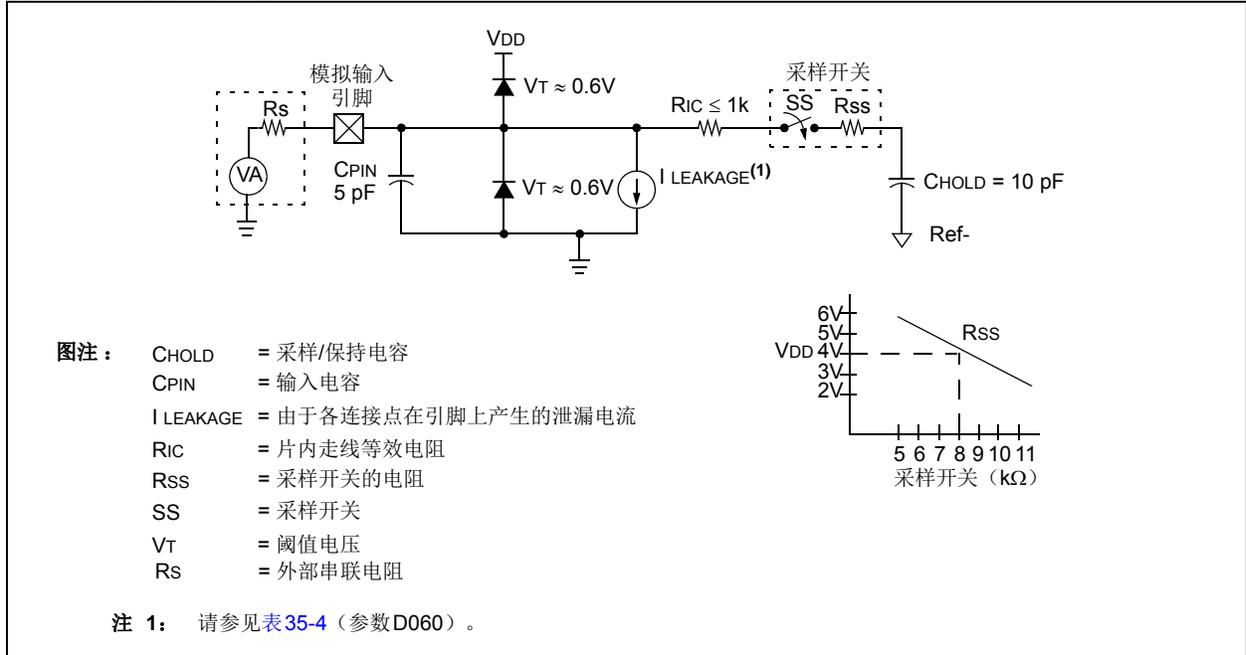
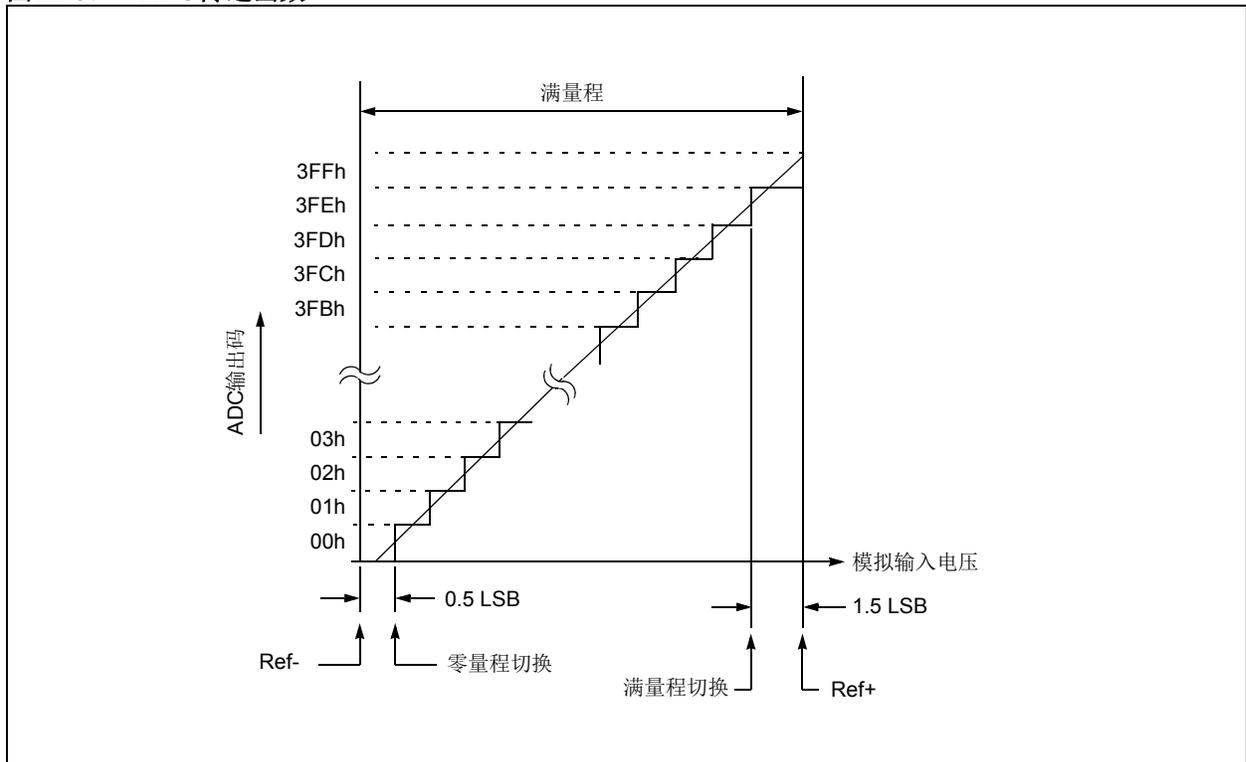


图22-5: ADC传递函数



PIC16(L)F18324/18344

22.4 寄存器定义：ADC控制

寄存器22-1: **ADCON0: ADC控制寄存器0**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CHS<5:0>						GO/DONE	ADON
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-2

CHS<5:0>: 模拟通道选择位

111111 = FVR (固定参考电压) ⁽²⁾

111110 = DAC1输出 ⁽¹⁾

111101 = 温度指示器 ⁽³⁾

111100 = V_{SS}

111011 = 保留。不连接任何通道

•

•

010111 = ANC7⁽⁴⁾

010110 = ANC6⁽⁴⁾

010101 = ANC5

010100 = ANC4

010011 = ANC3

010010 = ANC2

010001 = ANC1

010000 = ANC0

001111 = ANB7⁽⁴⁾

001110 = ANB6⁽⁴⁾

001101 = ANB5⁽⁴⁾

001100 = ANB4⁽⁴⁾

001011 = 保留。不连接任何通道

•

•

000101 = ANA5

000100 = ANA4

000011 = 保留。不连接任何通道

000010 = ANA2

000001 = ANA1

000000 = ANA0

bit 1

GO/DONE: ADC转换状态位

1 = ADC转换正在进行。将该位置1可启动ADC转换周期。

ADC转换完成时, 该位由硬件自动清零。

0 = ADC转换已完成/未进行

bit 0

ADON: ADC使能位

1 = 使能ADC

0 = 禁止ADC, 不消耗工作电流

注 1: 更多信息, 请参见第24.0节“5位数模转换器(DAC1)模块”。

2: 更多信息, 请参见第16.0节“固定参考电压(FVR)”。

3: 更多信息, 请参见第17.0节“温度指示器模块”。

4: 仅限PIC16(L)F18344。

PIC16(L)F18324/18344

寄存器 22-2: ADCON1: ADC控制寄存器 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>		—	ADNREF	ADPREF<1:0>		
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **ADFM:** ADC结果格式选择位
 1 = 右对齐。当装入转换结果时, ADRESH的高6位设置为0。
 0 = 左对齐。当装入转换结果时, ADRESL的低6位设置为0。
- bit 6-4 **ADCS<2:0>:** ADC转换时钟选择位
 111 = ADCRC (专用RC振荡器)
 110 = FOSC/64
 101 = FOSC/16
 100 = FOSC/4
 011 = ADCRC (专用RC振荡器)
 010 = FOSC/32
 001 = FOSC/8
 000 = FOSC/2
- bit 3 **未实现:** 读为0
- bit 2 **ADNREF:** A/D负参考电压配置位
 当ADON = 0时, 所有多路开关输入都会被断开。
 0 = VREF-连接到VSS
 1 = VREF-连接到外部VREF-
- bit 1-0 **ADPREF<1:0>:** ADC正参考电压配置位
 11 = VREF+连接到内部固定参考电压 (FVR) 模块⁽¹⁾
 10 = VREF+连接到外部VREF+引脚⁽¹⁾
 01 = 保留
 00 = VREF+连接到VDD

注 1: 当选择VREF+引脚作为正参考电压源时, 请注意存在最小电压规范值。有关详细信息, 请参见表35-13。

PIC16(L)F18324/18344

寄存器 22-3: ADACT: A/D自动转换触发

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	ADACT<4:0>				
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-5 未实现: 读为0

bit 4-0 **ADACT<4:0>**: 自动转换触发源选择位⁽¹⁾

10001	=	Timer5溢出 ⁽²⁾
10000	=	Timer3溢出 ⁽²⁾
1111	=	CCP4
1110	=	CCP3
1101	=	CCP2
1100	=	CCP1
1011	=	CLC4
1010	=	CLC3
1001	=	CLC2
1000	=	CLC1
0111	=	比较器C2
0110	=	比较器C1
0101	=	Timer2与PR2匹配
0100	=	Timer1溢出 ⁽²⁾
0011	=	Timer0溢出 ⁽²⁾
0010	=	Timer6与PR6匹配
0001	=	Timer4与PR4匹配
0000	=	不选择任何自动转换触发源

注 1: 这是所有触发源的上升沿敏感输入。

注 2: 触发对应于相应外设中断标志置1时。

PIC16(L)F18324/18344

寄存器 22-4: ADRESH: ADC结果寄存器高字节 (ADRESH) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 **ADRES<9:2>**: ADC结果寄存器位
10位转换结果的高8位

寄存器 22-5: ADRESL: ADC结果寄存器低字节 (ADRESL) ADFM = 0

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>		—	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 **ADRES<1:0>**: ADC结果寄存器位
10位转换结果的低2位

bit 5-0 **保留**: 不要使用。

PIC16(L)F18324/18344

寄存器 22-6: **ADRESH: ADC结果寄存器高字节 (ADRESH) ADFM = 1**

R/W-x/u	R/W-x/u						
—	—	—	—	—	—	ADRES<9:8>	
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-2 保留: 不要使用。

bit 1-0 **ADRES<9:8>**: ADC结果寄存器位
10位转换结果的高2位

寄存器 22-7: **ADRESL: ADC结果寄存器低字节 (ADRESL) ADFM = 1**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

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

PIC16(L)F18324/18344

表22-3: 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	寄存器所在页
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
ADCON0	CHS<5:0>						GO/DONE	ADON	242
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		243
ADACT	—	—	—	ADACT<4:0>					244
ADRESH	ADRESH<7:0>								245
ADRESL	ADRESL<7:0>								245
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		178
DAC1CON1	—	—	—	DAC1R<4:0>					262
OSCSTAT1	EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLOR	90

图注: — = 未实现, 读为0。ADC模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

23.0 数控振荡器（NCO1）模块

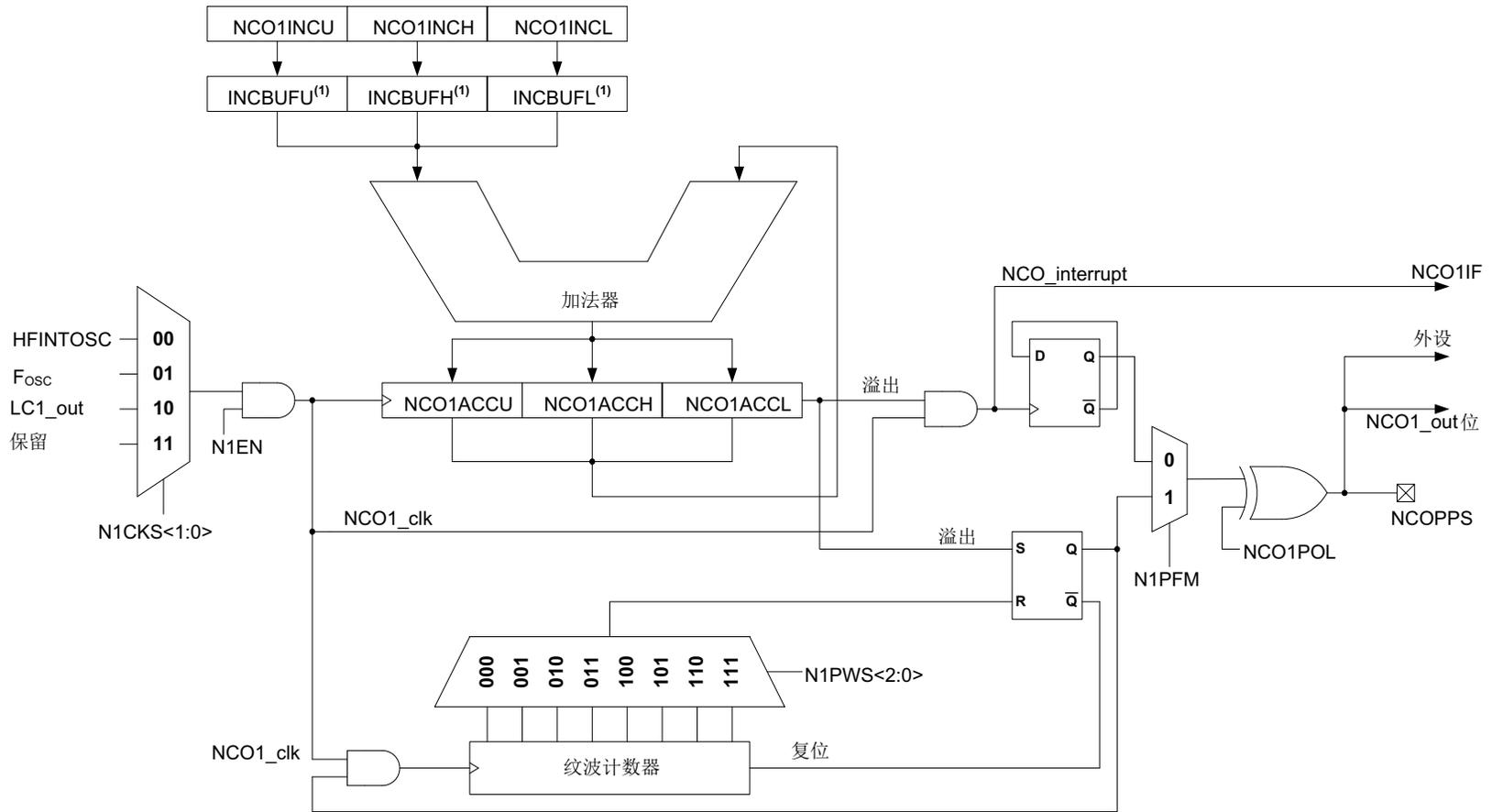
数控振荡器（NCO1）模块是一个定时器，它使用由于加上递增值而引起的溢出来对输入频率进行分频。加法运算相比于简单计数器驱动定时器的优势在于，输出频率分辨率不会随着分频器值的变化而变化。NCO1对于要求在固定占空比条件下确保频率精度和精细分辨率的应用最为有用。

NCO1的特性包括：

- 20位递增功能
- 固定占空比（Fixed Duty Cycle, FDC）模式
- 脉冲频率（Pulse Frequency, PF）模式
- 输出脉冲宽度控制
- 多个时钟输入源
- 输出极性控制
- 中断能力

图23-1给出了NCO1模块的简化框图。

图23-1: 数控振荡器模块简化框图



注 1: 递增寄存器是双重缓冲的, 以允许在没有先禁止NCO1模块的情况下更改值。用户不能访问这些缓冲区, 在此仅供参考。

PIC16(L)F18324/18344

23.1 NCO1工作原理

NCO1的工作方式是重复向累加器增加一个固定值。达到特定输入时钟速率时会进行加法运算。累加器会定期发生进位溢出，该进位为原始的NCO1输出（NCO_overflow）。这会有效降低输入时钟，降低的倍数为增加值与最大累加器值的比率。请参见公式23-1。

NCO1输出可以通过延长脉冲或翻转触发器进一步进行修正。修正后的NCO1输出随后会内部分配至其他外设，也可选择性地输出到引脚。累加器溢出也会产生中断（NCO_interrupt）。

NCO1周期以离散步阶进行变化，从而产生一个平均频率。

公式23-1: NCO1溢出频率

$$F_{OVERFLOW} = \frac{NCO1 \text{ 时钟频率} \times \text{递增值}}{2^{20}}$$

23.1.1 NCO1时钟源

可用于NCO1的时钟源包括：

- HFINTOSC
- Fosc
- LC1_out

NCO1时钟源通过配置NCO1CLK寄存器的N1CK<1:0>位进行选择。

23.1.2 累加器

累加器是一个20位寄存器。可通过以下三个寄存器对累加器进行读写访问：

- NCO1ACCL
- NCO1ACCH
- NCO1ACCU

23.1.3 加法器

NCO1加法器是一个全加器，它独立于系统时钟工作。先前结果与递增值的加法运算结果将在每个输入时钟的上升沿替换累加器值。

23.1.4 递增寄存器

递增值存储在三个寄存器中，以构成一个20位递增值。寄存器按从低字节到高字节的顺序排列为：

- NCO1INCL
- NCO1INCH
- NCO1INCU

当使能 NCO1 模块时，应首先写 NCO1INCU 和 NCO1INCH 寄存器，然后写 NCO1INCL 寄存器。NCO1INCL 寄存器写操作会在NCO1_clk信号的第二个上升沿同时启动装入递增缓冲寄存器。

这些寄存器是可读写的。递增寄存器是双重缓冲的，以允许在没有先禁止NCO1模块的情况下更改值。

当禁止NCO1模块时，在对递增寄存器执行写操作后，会立即装入递增缓冲区。

注： 用户不能访问递增缓冲寄存器。

23.2 固定占空比（FDC）模式

在固定占空比（FDC）模式下，每次累加器发生溢出（NCO_overflow）时，输出都会发生翻转。在递增值保持不变的情况下，这会提供固定频率、50%的占空比。可通过公式 23-2 计算 FDC 频率。FDC 频率是溢出频率的一半，因为产生一个 FDC 时钟周期需要两次溢出事件。更多信息，请参见图 23-2。

公式23-2: FDC频率

$$F_{fdc} = F_{overflow} / 2$$

可通过将 NCO1CON 寄存器中的 N1PFM 位清零来选择 FDC 模式。

23.3 脉冲频率（PF）模式

在脉冲频率（PF）模式下，每次累加器溢出（NCO_overflow）时，输出会在一个或多个时钟周期内有效。时钟周期结束时，输出返回到无效状态。这就产生了一个脉冲控制的输出。输出在紧跟溢出事件之后的时钟上升沿变为有效。更多信息，请参见图 23-2。

有效状态和无效状态的值取决于 NCO1CON 寄存器中的极性位 N1POL。

可通过将 NCO1CON 寄存器中的 N1PFM 位置 1 来选择 PF 模式。

23.3.1 输出脉冲宽度控制

在 PF 模式下工作时，输出有效状态的宽度会随多个时钟周期而变化。可使用 NCO1CLK 寄存器中的 N1PWS<2:0> 位来选择各种脉冲宽度。

当选定的脉冲宽度大于累加器溢出时间帧时，NCO1 的输出不翻转。

23.4 输出极性控制

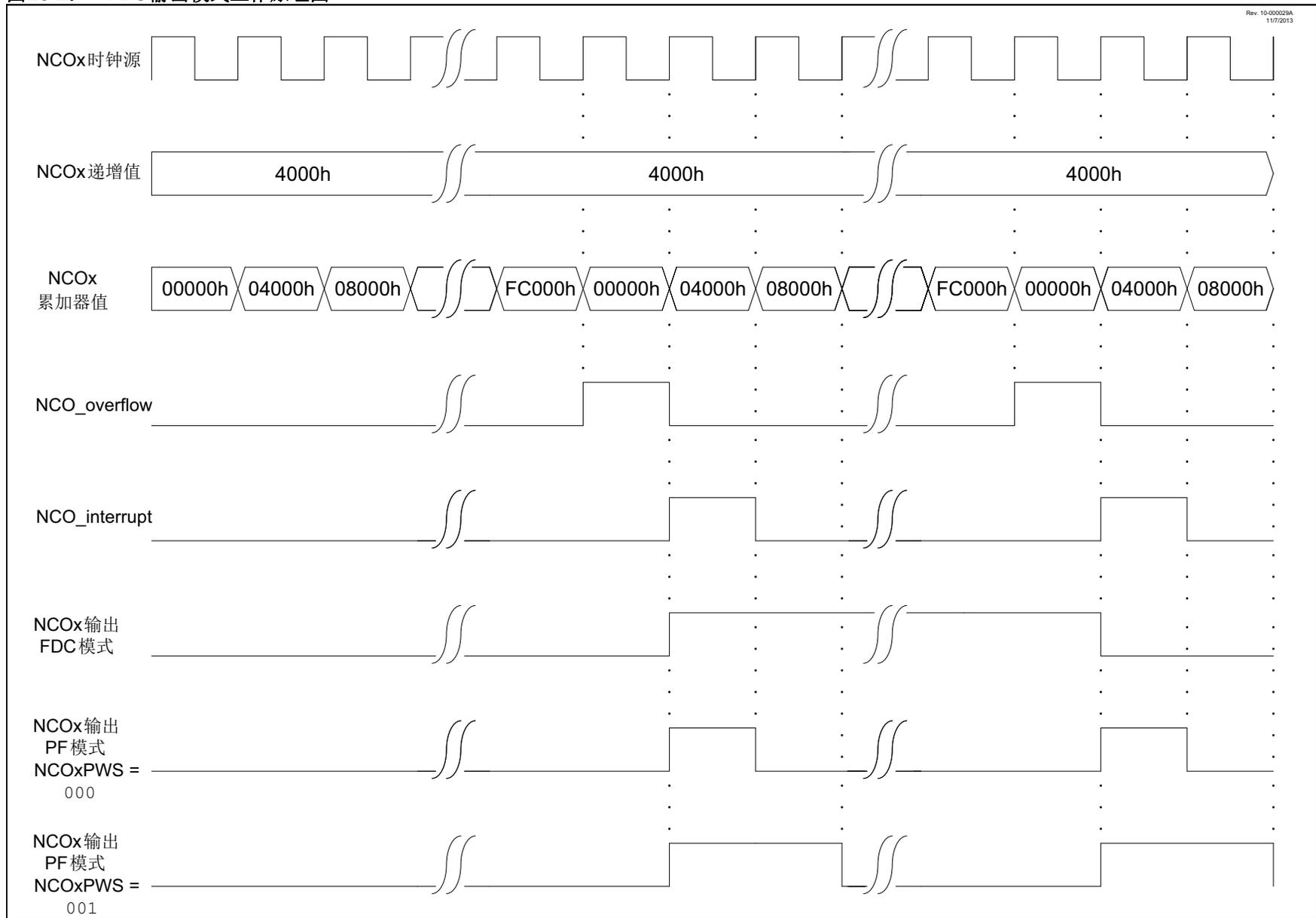
NCO1 模块中的最后一级是输出极性。NCO1CON 寄存器中的 N1POL 位用于选择输出极性。如果在允许中断时改变极性会导致输出结果的变化发生中断。

NCO1 输出可以在内部由源代码或其他外设使用。这可通过读取 NCO1CON 寄存器的 N1OUT（只读）位来实现。

NCO1 输出信号可用于下列外设：

- CWG

图 23-2: FDC 输出模式工作原理图



23.5 中断

当累加器溢出（NCO_overflow）时，PIR2寄存器的NCO1中断标志位NCO1IF置1。要允许此中断事件（NCO_interrupt），必须将以下各位置1：

- NCO1CON寄存器的N1EN位
- PIE2寄存器的NCO1IE位
- INTCON寄存器的PEIE位
- INTCON寄存器的GIE位

中断必须由软件通过在中断服务程序中将NCO1IF位清零来清除。

23.6 复位的影响

发生复位后，所有NCO1寄存器会清零。

23.7 休眠模式下的操作

NCO1模块独立于系统时钟工作，只要选定的时钟源保持活动状态，它就会继续在休眠期间运行。

如果使能了NCO1模块，并且选择HFINTOSC作为时钟源，则无论所选择的系统时钟源如何，HFINTOSC都会在休眠期间保持活动状态。

即，如果在NCO1使能时，同时选择HFINTOSC作为系统时钟和NCO1时钟源，则在休眠期间CPU会进入空闲状态，而NCO1会继续工作，并且HFINTOSC将保持活动状态。

这会直接影响休眠模式的电流。

PIC16(L)F18324/18344

23.8 NCO1控制寄存器

寄存器 23-1: NCO1CON: NCO1控制寄存器

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
N1EN	—	N1OUT	N1POL	—	—	—	N1PFM
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **N1EN:** NCO1使能位
1 = 使能NCO1模块
0 = 禁止NCO1模块
- bit 6 **未实现:** 读为0。
- bit 5 **N1OUT:** NCO1输出位
显示NCO1模块的当前输出值
- bit 4 **N1POL:** NCO1极性
1 = NCO1输出信号反相
0 = NCO1输出信号不反相
- bit 3-1 **未实现:** 读为0。
- bit 0 **N1PFM:** NCO1输出分频比模式
1 = NCO1工作在脉冲频率模式下
0 = NCO1工作在固定占空比模式下, 2分频

寄存器 23-2: NCO1CLK: NCO1输入时钟控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
N1PWS<2:0>			—	—	—	N1CKS<1:0>	
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-5 **N1PWS<2:0>**: NCO1输出脉冲宽度选择^(1, 2)
 000 = NCO1输出在1个输入时钟周期内有效
 001 = NCO1输出在2个输入时钟周期内有效
 010 = NCO1输出在4个输入时钟周期内有效
 011 = NCO1输出在8个输入时钟周期内有效
 100 = NCO1输出在16个输入时钟周期内有效
 101 = NCO1输出在32个输入时钟周期内有效
 110 = NCO1输出在64个输入时钟周期内有效
 111 = NCO1输出在128个输入时钟周期内有效

bit 4-2 **未实现**: 读为0

bit 1-0 **N1CKS<1:0>**: NCO1时钟源选择位
 00 = HFINTOSC (16 MHz)
 01 = Fosc
 10 = CLC1OUT
 11 = 保留

注 1: 仅在脉冲频率模式下工作时N1PWS才适用。

2: 如果NCO1脉冲宽度大于NCO1溢出周期, 则NCO1输出不翻转。

寄存器 23-3: NCO1ACCL: NCO1累加器寄存器——低字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCO1ACC<7:0>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 **NCO1ACC<7:0>**: NCO1累加器低字节

PIC16(L)F18324/18344

寄存器 23-4: NCO1ACCH: NCO1累加器寄存器——高字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCO1ACC<15:8>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **NCO1ACC<15:8>**: NCO1累加器高字节

寄存器 23-5: NCO1ACCU: NCO1累加器寄存器——最高字节⁽¹⁾

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—				NCO1ACC<19:16>			
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **未实现**: 读为0

bit 3-0 **NCO1ACC<19:16>**: NCO1累加器最高字节

注 1: 累加器跨越寄存器NCO1ACCU:NCO1ACCH:NCO1ACCL。24位保留, 但并未全部使用。该寄存器实时更新, 与CPU异步; 使用8位总线无法保证原子访问该24位空间。在模块正在工作时写该寄存器将产生不确定结果。

寄存器 23-6: NCO1INCL^(1,2): NCO1递增寄存器——低字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1
NCO1INC<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **NCO1INC<7:0>**: NCO1递增低字节

注 1: 逻辑递增跨越NCO1INCUNCO1INCH:NCO1INCL。

注 2: NCO1INC双重缓冲为INCBUF; 写入NCO1INCL后, INCBUF在写入NCO1INCL后的NCOCLK下一个下降沿更新; 应先写NCO1INCUNCO1INCH, 然后再写NCO1INCL。

寄存器 23-7: NCO1INCH⁽¹⁾: NCO1 递增寄存器——高字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCO1INC<15:8>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 **NCO1INC<15:8>**: NCO1 递增高字节

注 1: 逻辑递增跨越 NCO1INC_U:NCO1INCH:NCO1INCL。

寄存器 23-8: NCO1INC_U⁽¹⁾: NCO1 递增寄存器——最高字节

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCO1INC<19:16>			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-4 **未实现**: 读为0

bit 3-0 **NCO1INC<19:16>**: NCO1 递增最高字节

注 1: 逻辑递增跨越 NCO1INC_U:NCO1INCH:NCO1INCL。

PIC16(L)F18324/18344

表23-1: 与NCO1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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
PIR2	TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF	107
PIE2	TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	102
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
NCO1CON	N1EN	—	N1OUT	N1POL	—	—	—	N1PFM	254
NCO1CLK	N1PWS<2:0>			—	—	—	N1CKS<1:0>		255
NCO1ACCL	NCO1ACC <7:0>								255
NCO1ACCH	NCO1ACC <15:8>								256
NCO1ACCU	—	—	—	—	NCO1ACC <19:16>				256
NCO1INCL	NCO1INC<7:0>								256
NCO1INCH	NCO1INC<15:8>								257
NCO1INCU	—	—	—	—	NCO1INC<19:16>				257
CWG1DAT	—	—	—	—	DAT<3:0>				213
MDSRC	—	—	—	—	MDMS<3:0>				270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272
CCPxCAP	—	—	—	—	CCPxCTS<3:0>				307

图注: — = 未实现, 读为0。NCO1模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

24.0 5位数模转换器（DAC1）模块

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

DAC的输入可以连接到：

- 外部VREF引脚
- VDD供电电压
- 固定参考电压（FVR）

DAC的输出可以配置为向以下对象提供参考电压：

- 比较器的同相输入
- ADC输入通道
- DAC1OUT引脚

数模转换器（DAC）可以通过将DAC1CON0寄存器的DAC1EN位置1来使能。

24.1 输出电压选择

DAC具有32个输出电压可供选择。这32个电压通过DAC1CON1寄存器的DAC1R<4:0>位进行设置。

DAC输出电压由公式24-1确定：

公式24-1: DAC输出电压

$$V_{OUT} = \left((V_{SOURCE+}) - (V_{SOURCE-}) \times \frac{DAC1R\langle 4:0 \rangle}{2^5} \right) + (V_{SOURCE-})$$

$$V_{SOURCE+} = V_{DD} \text{ 或 } V_{REF+} \text{ 或 } FVR$$

$$V_{SOURCE-} = V_{SS} \text{ 或 } V_{REF-}$$

24.2 比例输出电压

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

表35-15中给出了梯形电阻网络中各个电阻的阻值。

24.3 DAC参考电压输出

通过将DAC1CON0寄存器的DAC1OE位置1，可以将DAC电压输出到DAC1OUT引脚。选择将DAC参考电压输出到DAC1OUT引脚会自动改写该引脚的数字输出缓冲器和数字输入阈值检测器功能，并禁止该引脚的弱上拉和恒流驱动功能。当DAC1OUT引脚已被配置为DAC参考电压输出时，读取该引脚将总是返回0。

由于电流驱动能力有限，因此必须在DAC参考电压输出引脚DAC1OUT上外接缓冲器。图24-2举例说明了这一缓冲技术。

PIC16(L)F18324/18344

图24-1: 数模转换器框图

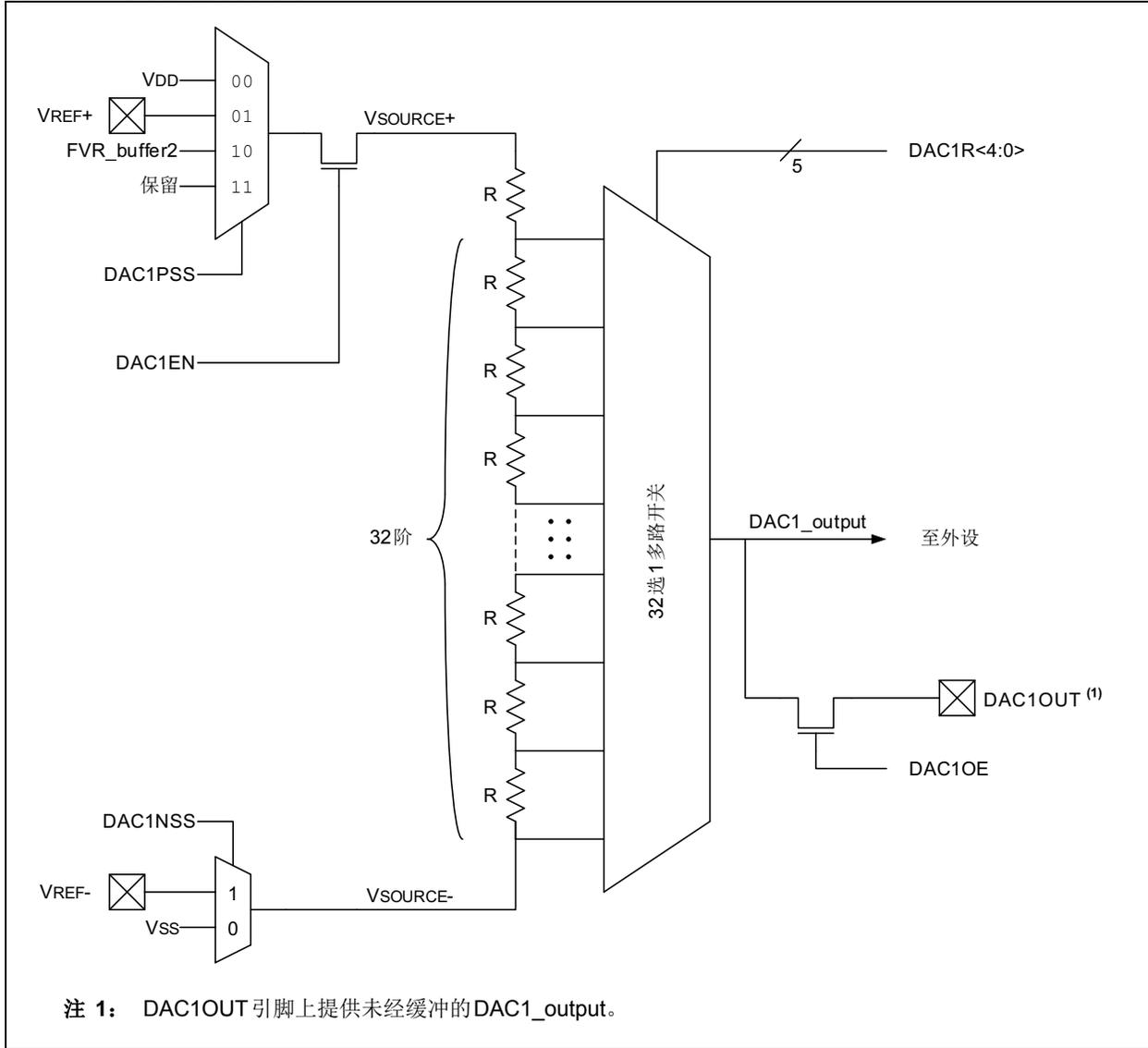
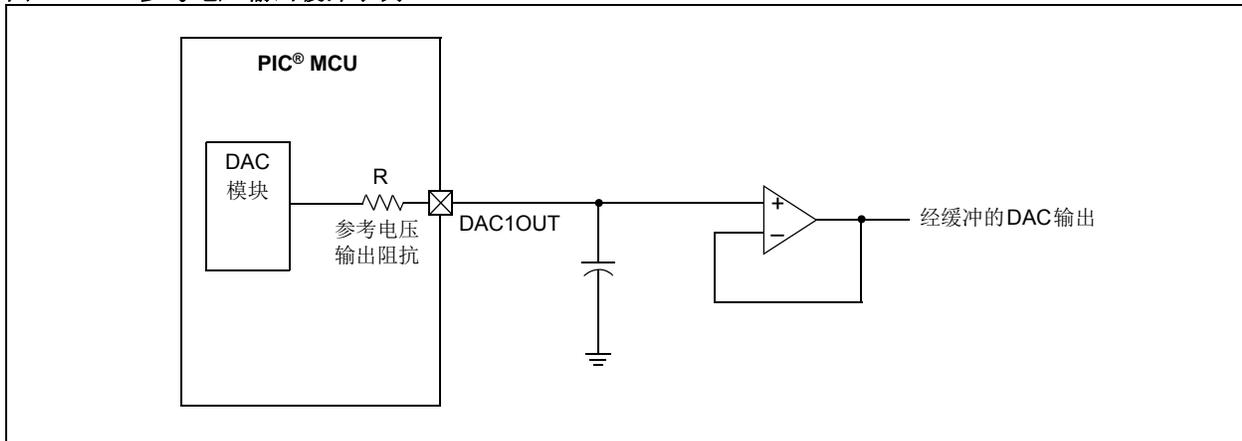


图24-2: 参考电压输出缓冲示例



24.4 休眠期间的操作

DAC在休眠期间继续工作。如果因中断或看门狗定时器超时将器件从休眠模式唤醒，DAC1CON0寄存器的内容将不受影响。

24.5 复位的影响

器件复位会产生以下影响：

- DAC被禁止。
- DAC输出电压从DAC1OUT引脚上被移除。
- DAC1R<4:0>范围选择位被清零。

24.6 寄存器定义：DAC控制

寄存器 24-1: DACCON0: 参考电压控制寄存器0

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **DAC1EN:** DAC1 使能位
1 = 使能DAC
0 = 禁止DAC
- bit 6 **未实现:** 读为0
- bit 5 **DAC1OE:** DAC1 电压输出1使能位
1 = DAC 电压从DAC1OUT 引脚输出
0 = DAC 电压从DAC1OUT 引脚断开
- bit 4 **未实现:** 读为0
- bit 3-2 **DAC1PSS<1:0>:** DAC1 正参考电压源选择位
11 = 保留，不要使用
10 = FVR 输出
01 = VREF+ 引脚
00 = VDD
- bit 1 **未实现:** 读为0
- bit 0 **DAC1NSS:** DAC1 负参考电压源选择位
1 = VREF- 引脚
0 = VSS

PIC16(L)F18324/18344

寄存器 24-2: DACCON1: 参考电压控制寄存器 1

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	DAC1R<4:0>				
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-5 **未实现:** 读为0

bit 4-0 **DAC1R<4:0>:** DAC1 电压输出选择位
 $V_{OUT} = (V_{SRC+} - V_{SRC-}) * (DAC1R<4:0>/32) + V_{SRC}$

表 24-1: 与 DAC1 模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
DACCON0	DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS	261
DACCON1	—	—	—	DAC1R<4:0>					262
CMxCON1	CxINTP	CxINTN	CxPCH<2:0>		CxNCH<2:0>				189
ADCON0	CHS<5:0>					GO/DONE	ADON		242

图注: — = 未实现位, 读为0。DAC1 模块不使用阴影单元。

25.0 数据信号调制器（DSM）模块

数据信号调制器（DSM）是一种外设，用户可以通过它将数据流（也称为调制器信号）与载波信号进行混合来产生调制输出。

载波和调制器信号均送到DSM模块，信号可以来自内部某个外设的输出，也可以通过某个输入引脚从外部提供。

调制输出信号的产生方式是：对载波和调制器信号执行逻辑与操作，然后送到MDOUT引脚上。

载波信号由两个不同的独立信号组成。高载波（CARH）信号和低载波（CARL）信号。在调制器（MOD）信号处于逻辑高电平状态期间，DSM会将高载波信号与调制器信号进行混合。在调制器信号处于逻辑低电平状态时，DSM会将低载波信号与调制器信号进行混合。

通过这种方法，DSM可以产生以下几种键控调制方案：

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

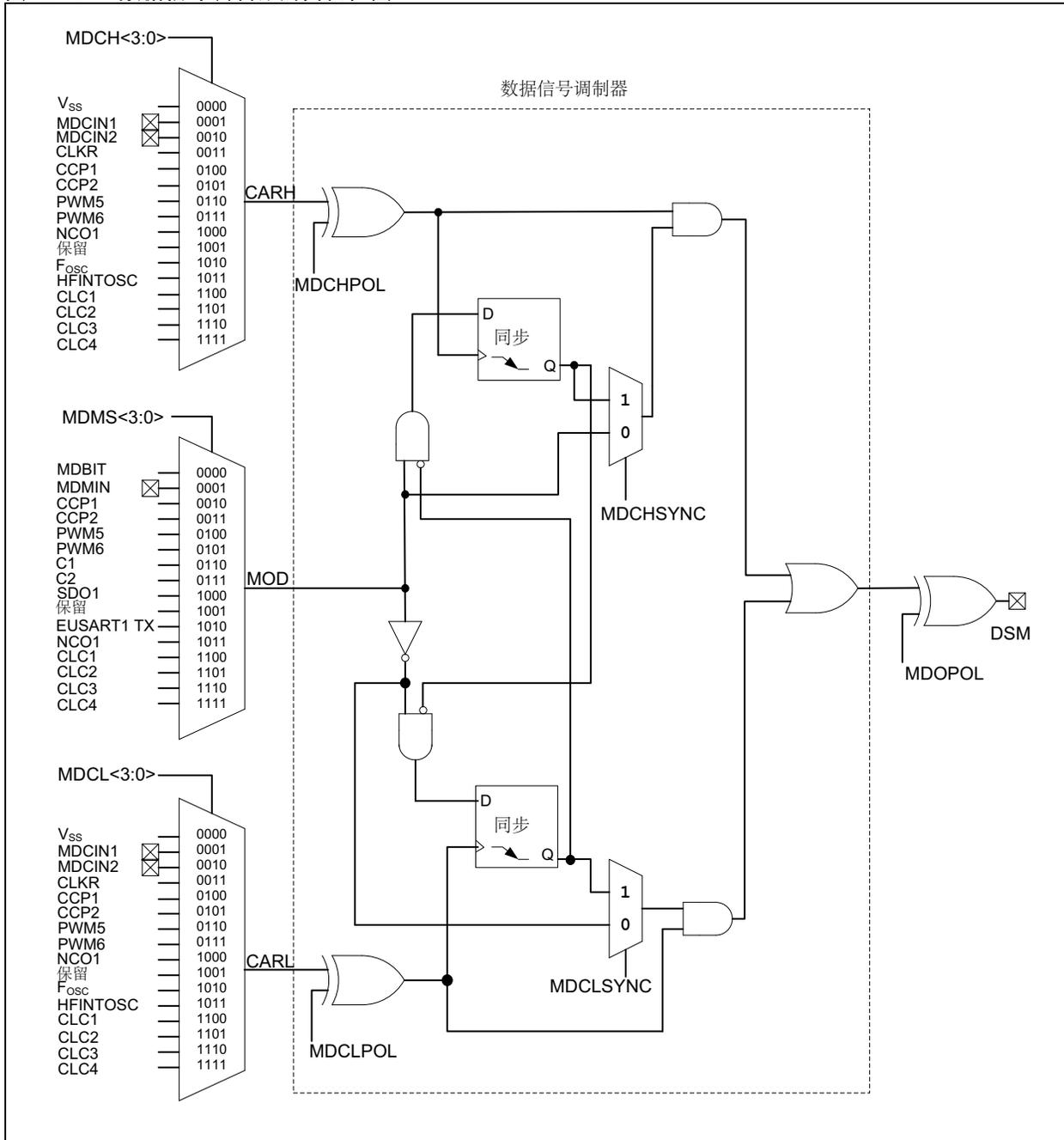
此外，DSM模块还提供了以下特性：

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

图25-1给出了数据信号调制器外设的简化框图。

PIC16(L)F18324/18344

图25-1: 数据信号调制器的简化框图



25.1 DSM工作原理

DSM模块通过将MDCON寄存器的MDEN位置1来使能。清零MDCON寄存器中的MDEN位时，将会通过自动将高载波信号和低载波信号切换至Vss信号源来禁止DSM模块。调制器信号源也会被切换至MDCON寄存器中的MDBIT位。这不仅可以确保DSM模块不工作，而且会使电流消耗降至最低。

当MDEN位清零且DSM模块被禁止时，由调制源、调制高载波信号和调制低载波信号控制寄存器保存的、用于选择高载波信号、低载波信号和调制器信号源的值不会受影响。在DSM不工作时，这些寄存器中的值会保持不变。当MDEN位置1且DSM模块再次使能并处于工作状态时，将会再次选择高载波信号、低载波信号和调制器信号的信号源。

不关闭DSM模块，也可以禁止调制输出信号。DSM模块将保持工作状态并继续混合信号，但输出值不会送到DSM引脚。在禁止输出期间，DSM引脚将保持低电平。可以通过清零MDCON寄存器中的MDEN位来禁止调制输出。

25.2 调制器信号源

调制器信号可以通过以下信号源提供：

- CCP1输出
- CCP2输出
- PWM5输出
- PWM6输出
- MSSP1 SDO1（仅SPI模式）
- 比较器C1输出
- 比较器C2输出
- EUSART1 TX输出
- MDMIN引脚上的外部信号
- NCO1输出
- CLC1输出
- CLC2输出
- CLC3输出
- CLC4输出
- MDCON寄存器中的MDBIT位

调制器信号通过配置MDSRC寄存器的MDMS<3:0>位来选择。

25.3 载波信号源

高载波信号和低载波信号可以通过以下信号源提供：

- CCP1输出
- CCP2输出
- PWM5输出
- PWM6输出
- NCO1输出
- Fosc（系统时钟）
- HFINTOSC
- CLC1输出
- CLC2输出
- CLC3输出
- CLC4输出
- CLKR
- MDCIN1引脚上的外部信号
- MDCIN2引脚上的外部信号
- Vss

高载波信号通过配置MDCARH寄存器的MDCH<3:0>位来选择。低载波信号通过配置MDCARL寄存器的MDCL<3:0>位来选择。

25.4 载波同步

当DSM在高载波信号源和低载波信号源之间切换时，调制输出信号中的载波数据可能会被截短。为了防止这种情况，可以将载波信号与调制器信号进行同步。当调制器信号转换偏离同步载波时，非同步载波源立即有效，而同步载波保持有效直至出现其下一个下降沿。当调制器信号转换回同步载波时，非同步载波立即禁止，而在同步载波有效之前调制器等待直至出现同步载波的下一个下降沿。

对于高载波信号源和低载波信号源，同步功能单独进行使能。高载波信号的同步通过将MDCARH寄存器的MDCHSYNC位置1来使能。低载波信号的同步通过将MDCARL寄存器的MDCLSYNC位置1来使能。

图25-1至图25-6给出了使用各种同步方法时的时序图。

PIC16(L)F18324/18344

图25-2: 开关键控 (OOK) 同步

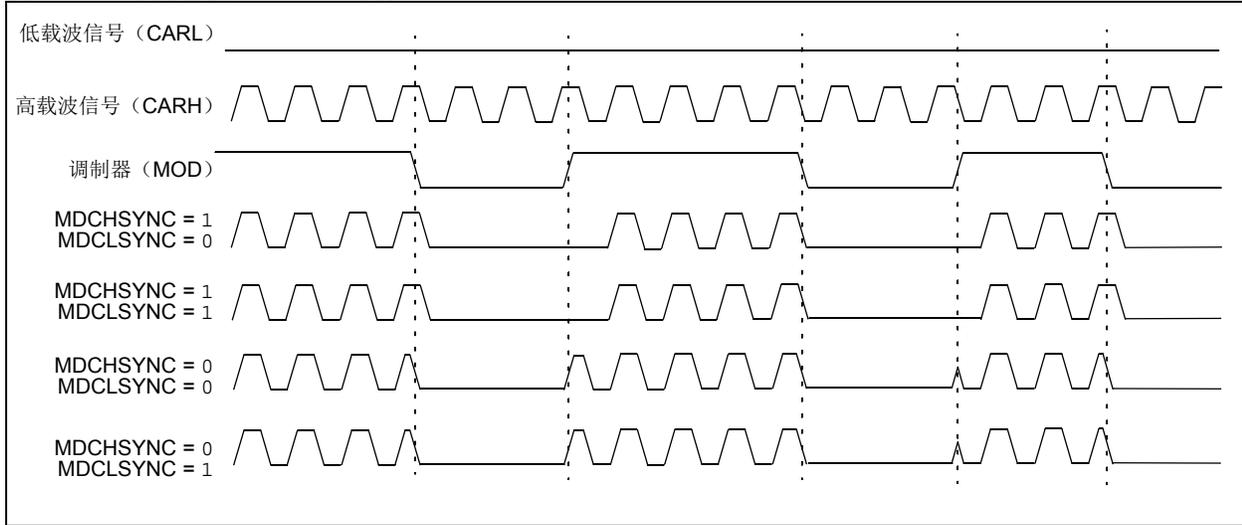


图25-3: 无同步 (MDCHSYNC = 0, MDCLSYNC = 0)

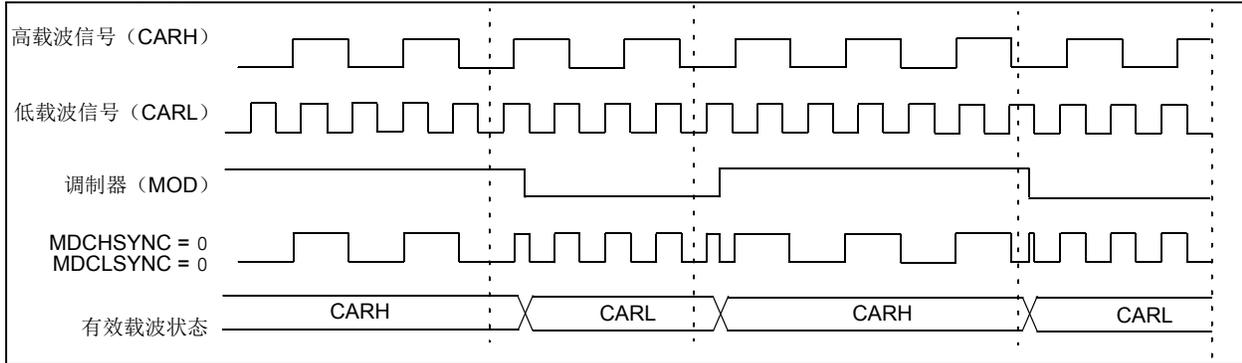


图25-4: 高载波信号同步 (MDCHSYNC = 1, MDCLSYNC = 0)

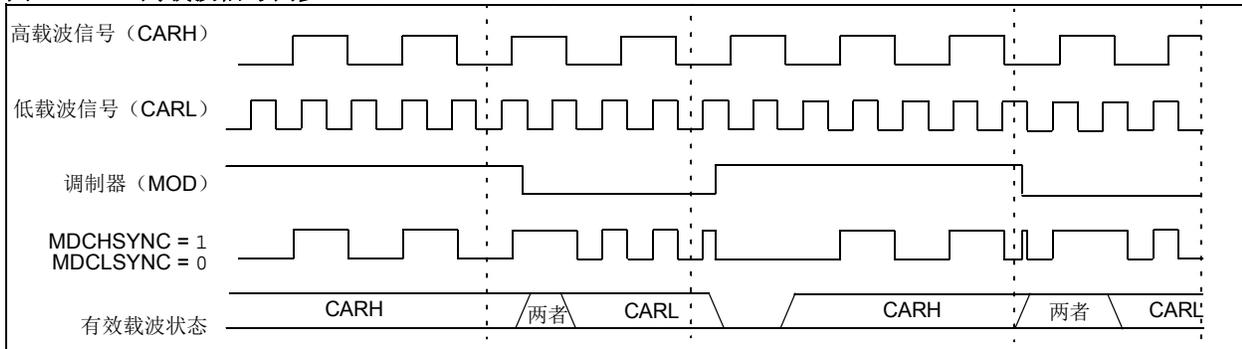


图25-5: 低载波信号同步 (MDCHSYNC = 0, MDCLSYNC = 1)

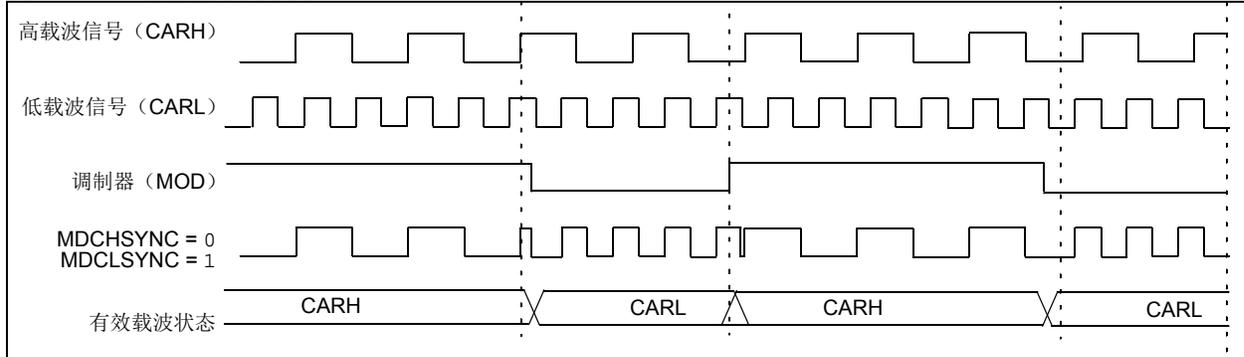
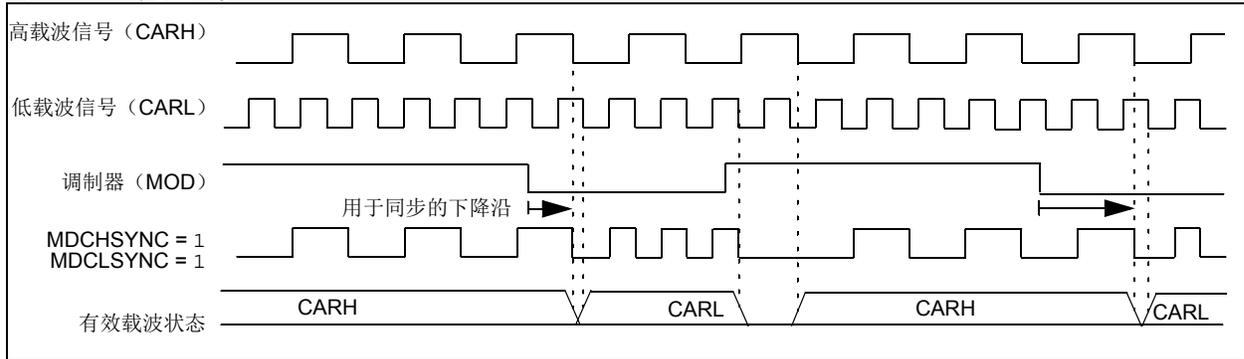


图25-6: 完全同步 (MDCHSYNC = 1, MDCLSYNC = 1)



PIC16(L)F18324/18344

25.5 载波源极性选择

高载波信号和低载波信号的任意选定输入源提供的信号都可以进行反相。对高载波信号源信号反相通过将MDCARH寄存器的MDCHPOL位置1来使能。对低载波信号源信号反相通过将MDCARL寄存器的MDCLPOL位置1来使能。

25.6 可编程调制器数据

用户可以选择MDCON寄存器的MDBIT位作为调制器信号的信号源。这使得用户能够设置用于调制的值。

25.7 调制输出极性

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

25.8 压摆率控制

可以禁止对输出端口引脚的压摆率限制。压摆率限制可以通过将与该引脚相关的SLRCON寄存器中的SLR位清零来移除。例如，清除RA5引脚的压摆率限制将需要清零SLRCONA寄存器的SLRA5位。

25.9 休眠模式下的操作

DSM模块不受休眠模式的影响。如果载波输入源和调制器输入源仍在休眠期间工作，则DSM在休眠期间仍然可以工作。

25.10 复位的影响

在发生任何器件复位时，都将禁止DSM模块。用户的固件负责在使能输出之前初始化该模块。寄存器会复位为其默认值。

25.11 寄存器定义：调制控制

寄存器 25-1: **MDCON**: 调制控制寄存器

R/W-0/0	U-0	U-0	R/W-0/0	R-0/0	U-0	U-0	R/W-0/0
MDEN	—	—	MDOPOL	MDOUT	—	—	MDBIT ⁽²⁾
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **MDEN**: 调制器模块使能位
 1 = 使能调制器模块, 且该模块混合输入信号
 0 = 禁止调制器模块, 且该模块没有输出
- bit 6-5 **未实现**: 读为0
- bit 4 **MDOPOL**: 调制器输出极性选择位
 1 = 调制器输出信号反相; 空闲时输出高电平
 0 = 调制器输出信号不反相; 空闲时输出低电平。
- bit 3 **MDOUT**: 调制器输出位
 指示调制器模块的当前输出值。⁽¹⁾
- bit 2-1 **未实现**: 读为0
- bit 0 **MDBIT**: 允许软件手动设置模块的调制源输入⁽²⁾

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

PIC16(L)F18324/18344

寄存器 25-2: MDSRC: 调制源控制寄存器

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	MDMS<3:0>			
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 未实现: 读为0

bit 3-0 MDMS<3:0>: 调制源选择位

1111 = CLC4输出

1110 = CLC3输出

1101 = CLC2输出

1100 = CLC1输出

1011 = NCO1输出

1010 = EUSART1 TX输出

1001 = 保留

1000 = MSSP1 SDO1输出

0111 = C2 (比较器2) 输出

0110 = C1 (比较器1) 输出

0101 = PWM6输出

0100 = PWM5输出

0011 = CCP2输出 (仅限PWM输出模式)

0010 = CCP1输出 (仅限PWM输出模式)

0001 = MDMINPPS

0000 = MDCON寄存器的MDBIT位是调制源

寄存器 25-3: MDCARH: 调制高载波信号控制寄存器

U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	MDCHPOL	MDCHSYNC	—	MDCH<3:0> ⁽¹⁾			
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7 **未实现:** 读为0

bit 6 **MDCHPOL:** 调制器高载波信号极性选择位

1 = 选定的高载波信号反相

0 = 选定的高载波信号不反相

bit 5 **MDCHSYNC:** 调制器高载波信号同步使能位

1 = 调制器先等待高载波信号上出现下降沿, 然后再切换为低载波信号

0 = 调制器输出不与高载波信号进行同步⁽¹⁾

bit 4 **未实现:** 读为0

bit 3-0 **MDCH<3:0>:** 调制器数据高载波信号选择位⁽¹⁾

1111 = CLC4输出

1110 = CLC3输出

1101 = CLC2输出

1100 = CLC1输出

1011 = HFINTOSC

1010 = Fosc

1001 = 保留。不连接任何通道

1000 = NCO1输出

0111 = PWM6输出

0110 = PWM5输出

0101 = CCP2输出 (仅限PWM输出模式)

0100 = CCP1输出 (仅限PWM输出模式)

0011 = 参考时钟模块信号 (CLKR)

0010 = MDCIN2PPS

0001 = MDCIN1PPS

0000 = Vss

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

PIC16(L)F18324/18344

寄存器 25-4: MDCARL: 调制低载波信号控制寄存器

U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	MDCLPOL	MDCLSYNC	—	MDCL<3:0> ⁽¹⁾			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **未实现:** 读为0
- bit 6 **MDCLPOL:** 调制器低载波信号极性选择位
1 = 选定的低载波信号反相
0 = 选定的低载波信号不反相
- bit 5 **MDCLSYNC:** 调制器低载波信号同步使能位
1 = 调制器先等待低载波信号上出现下降沿, 然后再切换为高载波信号
0 = 调制器输出不与低载波信号进行同步⁽¹⁾
- bit 4 **未实现:** 读为0
- bit 3-0 **MDCL<3:0>:** 调制器数据低载波信号选择位⁽¹⁾
1111 = CLC4输出
1110 = CLC3输出
1101 = CLC2输出
1100 = CLC1输出
1011 = HFINTOSC
1010 = Fosc
1001 = 保留。不连接任何通道
1000 = NCO1输出
0111 = PWM6输出
0110 = PWM5输出
0101 = CCP2输出 (仅限PWM输出模式)
0100 = CCP1输出 (仅限PWM输出模式)
0011 = 参考时钟模块信号 (CLKR)
0010 = MDCIN2PPS
0001 = MDCIN1PPS
0000 = Vss

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

表25-1: 与数据信号调制器模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	144
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	144
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
SLRCONB ⁽¹⁾	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	150
INVLVB ⁽¹⁾	INVLVB7	INVLVB6	INVLVB5	INVLVB4	—	—	—	—	150
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
SLRCONC	SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	156
INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	157
MDCON	MDEN	—	—	MDOPOL	MDOOUT	—	—	MDBIT	269
MDSRC	—	—	—	—	MDMS<3:0>			—	270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>			—	271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>			—	272
MDCIN1PPS	—	—	—	MDCIN1PPS<4:0>				—	160
MDCIN2PPS	—	—	—	MDCIN2PPS<4:0>				—	160
MDMINPPS	—	—	—	MDMINPPS<4:0>				—	160

图注: — = 未实现, 读为0。数据信号调制器模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现。读为1。

PIC16(L)F18324/18344

26.0 TIMER0模块

Timer0 模块是 8/16 位定时器/计数器，具有以下特性：

- 16 位定时器/计数器
- 带有可编程周期的 8 位定时器/计数器
- 同步或异步操作
- 可选择的时钟源
- 可编程预分频器（独立于看门狗定时器）
- 可编程后分频器
- 休眠模式下的操作
- 匹配或溢出时中断
- （通过 PPS）在 I/O 引脚上输出或输出至其他外设

26.1 Timer0 工作原理

Timer0 可用作 8 位定时器/计数器或 16 位定时器/计数器。模式通过 T0CON 寄存器的 T016BIT 位选择。

与 FOSC/4 时钟源一起使用时，模块用作定时器，并在每个指令周期递增。与任何其他时钟源一起使用时，模块可用作定时器或计数器，在外部时钟源的每个上升沿递增。

26.1.1 16 位模式

正常工作时，TMR0 在时钟源的上升沿递增。时钟输入上的 15 位预分频器提供几个预分频选项（见 T0CON1 寄存器中的预分频比控制位 T0CKPS<3:0>）。

26.1.1.1 Timer0 的 16 位读写模式

TMR0H 不是 Timer0 在 16 位模式下的实际高字节。它实际上是 Timer0 的实际高字节的缓冲版本，不能直接读写（见图 26-1）。在读 TMR0L 时使用 Timer0 高字节的内容更新 TMR0H。这样可以一次读取 Timer0 的全部 16 位，而无需验证读到的高字节和低字节的有效性（在高、低字节分两次连续读取的情况下，由于可能存在计满返回，因此需要验证读到字节的有效性）。

同样，写入 Timer0 的高字节也是通过 TMR0H 缓冲寄存器来操作的。在写入 TMR0L 的同时，使用 TMR0H 的内容更新 Timer0 的高字节。这样一次就可以完成 Timer0 全部 16 位的更新。

26.1.2 8 位模式

正常工作时，TMR0 在时钟源的上升沿递增。时钟输入上的 15 位预分频器提供几个预分频选项（见 T0CON1 寄存器中的预分频比控制位 T0CKPS<3:0>）。

在 8 位模式下，TMR0H 不再用作 Timer0 高字节，而是用作周期寄存器（Period Register, PR）。在每个时钟周期，TMR0L 的值都会与 TMR0H 的值进行比较。当两个值匹配时，发生以下事件：

- TMR0_out 变为高电平并保持一个预分频时钟周期
- 复位 TMR0L
- TMR0H 的内容复制到周期缓冲区

在 8 位模式下，TMR0L 和 TMR0H 寄存器均可直接读写。在任何器件复位时，TMR0L 寄存器都会清零，而 TMR0H 寄存器则初始化为 FFh。

发生以下事件时，预分频器和后分频器计数器均会清零：

- 对 TMR0L 寄存器进行写操作
- 对 T0CON0 或 T0CON1 寄存器进行写操作
- 任何器件复位——上电复位（POR）、MCLR 复位、看门狗定时器复位（WDTR）或欠压复位（BOR）

26.1.3 计数器模式

在计数器模式下，通常通过将 T0CON1 寄存器的 T0CKPS 位设置为 0000 来禁止预分频器。计数器在时钟输入（或预分频器输出，如果使用预分频器的话）的每个上升沿递增 1。

26.1.4 定时器模式

在定时器模式下，只要存在有效时钟信号且 T0CON1 寄存器（寄存器 26-4）的 T0CKPS 位设置为 0000，Timer0 模块将在每个指令周期递增。当增加预分频器时，定时器将以基于预分频值的速率递增。

26.1.5 异步模式

当 T0CON1 寄存器的 T0ASYNCR 位置 1（T0ASYNCR = 1）时，计数器在输入源（或预分频器的输出，如果使用的话）的每个上升沿递增。异步模式允许计数器在休眠模式期间继续运行，前提是时钟也在休眠期间继续运行。

26.1.6 同步模式

当T0CON1寄存器的T0ASYNC位清零（T0ASYNC = 0）时，计数器时钟与系统振荡器同步（Fosc/4）。当在同步模式下工作时，计数器时钟频率无法超过Fosc/4。

26.2 时钟源选择

T0CON1寄存器的T0CS<2:0>位用于选择Timer0的时钟源。寄存器26-4显示了时钟源选择。

26.2.1 内部时钟源

当选择内部时钟源时，Timer0作为定时器工作，并将在时钟源上升沿的整数倍（由Timer0预分频器决定）处递增。

26.2.2 外部时钟源

当选择外部时钟源时，Timer0模块可以作为定时器或计数器工作，Timer0将在外部时钟源上升沿的整数倍（由Timer0预分频器决定）处递增。

26.3 可编程预分频器

软件可编程的预分频器只能用于Timer0。Timer0有16个预分频比选项，范围从1:1到1:32768（2的次幂）。预分频值可通过T0CON1寄存器的T0CKPS<3:0>位进行选择。

预分频器不可直接读写。可通过写入TMR0L寄存器、T0CON0寄存器或T0CON1寄存器来清零预分频器寄存器。

26.4 可编程后分频器

软件可编程的后分频器（输出分频器）只能用于Timer0。Timer0有16个后分频比选项，范围从1:1至1:16。后分频值可通过T0CON0寄存器的T0OUTPS<3:0>位进行选择。

后分频器不可直接读写。可通过写入TMR0L寄存器、T0CON0寄存器或T0CON1寄存器来清零后分频器寄存器。

26.5 休眠期间的操作

当同步工作时，Timer0将暂停。当异步工作时，Timer0将继续递增并将器件从休眠模式唤醒（如果允许Timer0中断），前提是输入时钟源处于活动状态。

26.6 Timer0中断

Timer0中断标志位（TMR0IF）在发生以下任一条件时置1：

- 8位TMR0L与TMR0H值匹配
- 16位TMR0从FFFFh计满返回

当后分频比（T0OUTPS<3:0>）设置为1:1操作（无分频）时，T0IF标志位将在每次发生TMR0匹配或计满返回时置1。通常，TMR0IF标志位将在每发生T0OUTPS + 1次匹配或计满返回时置1。

如果允许Timer0中断（PIE0寄存器的TMR0IE位 = 1），CPU将发生中断且器件从休眠模式唤醒（更多详细信息，请参见第26.5节“休眠期间的操作”）。

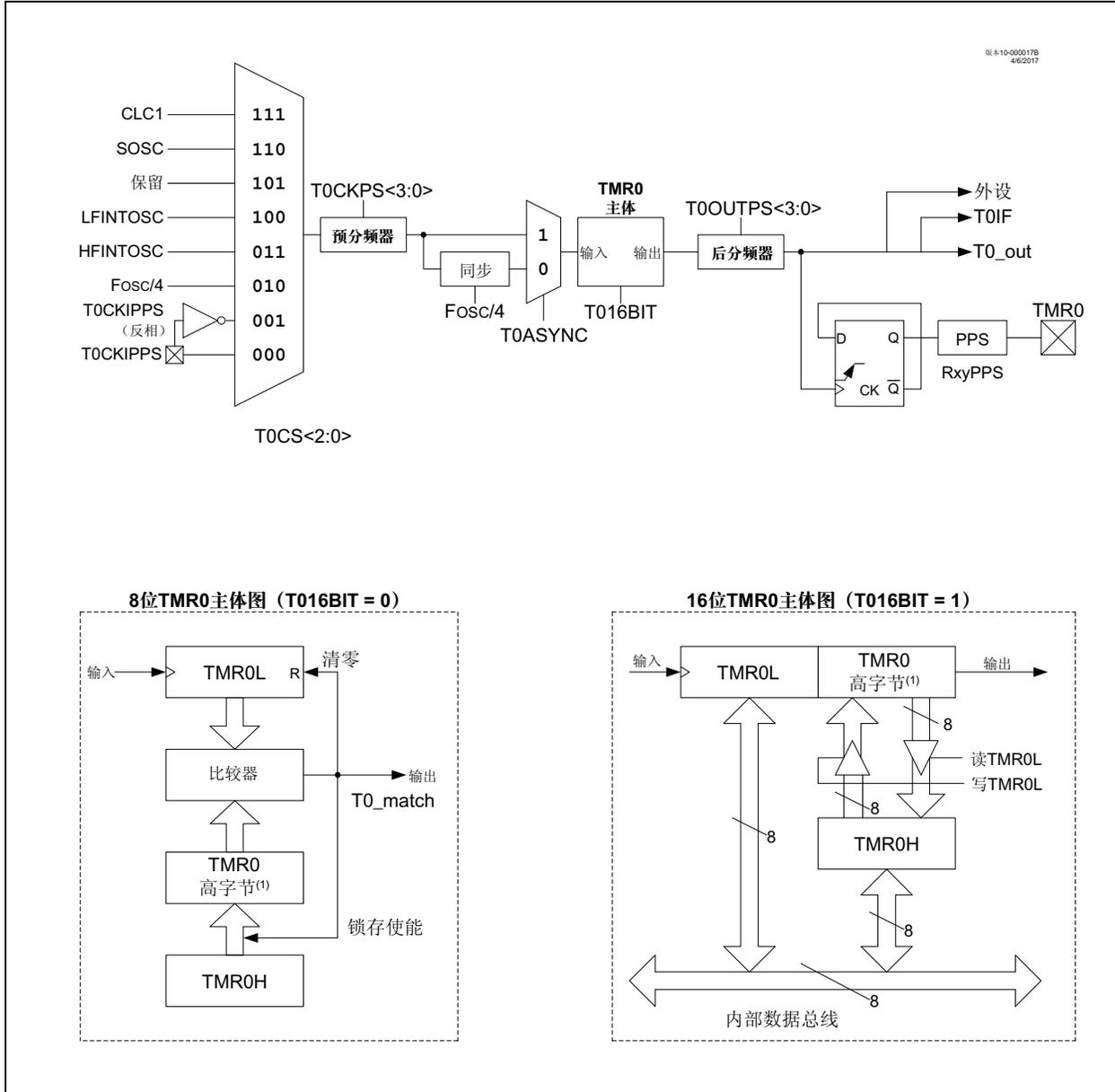
26.7 Timer0输出

可通过RxyPPS输出选择寄存器将Timer0输出输送到任一I/O引脚（有关详细信息，请参见第13.0节“外设引脚选择（PPS）模块”）。Timer0输出也可供其他外设使用，例如作为模数转换器的自动转换触发信号。最后，Timer0输出可由软件通过T0CON0寄存器（寄存器26-3）的Timer0输出位（T0OUT）监视。

当在8位模式下TMR0L值与TMR0H值发生匹配或在16位模式下TMR0计满返回时，TMR0_out将是一个后分频时钟周期。当发生匹配条件时，Timer0输出将在每发生T0OUTPS + 1次匹配时发生翻转。Timer0总周期内发生两次匹配事件，从而使输出的占空比为50%。

PIC16(L)F18324/18344

图 26-1: TMR0 框图



26.8 寄存器定义: Timer0 寄存器

寄存器 26-1: TMR0L: TIMER0 计数寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR0L<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **TMR0L<7:0>**: TMR0 计数器 bit 7..0

寄存器 26-2: TMR0H: TIMER0 周期寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TMR0H<7:0> 或 TMR0<15:8>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 当 T016BIT = 0 时
TMR0H<7:0>: TMR0 周期寄存器 bit 7..0
当 T016BIT = 1 时
TMR0<15:8>: TMR0 计数器 bit 15..8

PIC16(L)F18324/18344

寄存器26-3: T0CON0: TIMER0控制寄存器0

R/W-0/0	U-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>			
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为0
u = 不变 x = 未知 -n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1 0 = 清零

- bit 7 **T0EN:** TMR0使能位
1 = 模块已使能并且正在工作
0 = 模块已禁止并且处于最低功耗模式
- bit 6 **未实现:** 读为0
- bit 5 **T0OUT:** TMR0输出 (只读)
TMR0输出位
- bit 4 **T016BIT:** TMR0用作16位定时器选择位
1 = TMR0是16位定时器
0 = TMR0是8位定时器
- bit 3-0 **T0OUTPS<3:0>:** TMR0输出后分频比选择位
0000 = 1:1后分频比
0001 = 1:2后分频比
0010 = 1:3后分频比
0011 = 1:4后分频比
0100 = 1:5后分频比
0101 = 1:6后分频比
0110 = 1:7后分频比
0111 = 1:8后分频比
1000 = 1:9后分频比
1001 = 1:10后分频比
1010 = 1:11后分频比
1011 = 1:12后分频比
1100 = 1:13后分频比
1101 = 1:14后分频比
1110 = 1:15后分频比
1111 = 1:16后分频比

PIC16(L)F18324/18344

寄存器 26-4: T0CON1: TIMER0控制寄存器 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0CS<2:0>		T0ASYNC	T0CKPS<3:0>				
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-5 **T0CS<2:0>**: Timer0时钟源选择位

000 = T0CKIPPS (同相)

001 = T0CKIPPS (反相)

010 = Fosc/4

011 = HFINTOSC

100 = LFINTOSC

101 = 保留

110 = SOSC

111 = CLC1

bit 4 **T0ASYNC**: TMR0输入异步使能位

1 = TMR0计数器输入与系统时钟不同步

0 = TMR0计数器输入与Fosc/4同步

bit 3-0 **T0CKPS<3:0>**: 后分频比选择位

0000 = 1:1

0001 = 1:2

0010 = 1:4

0011 = 1:8

0100 = 1:16

0101 = 1:32

0110 = 1:64

0111 = 1:128

1000 = 1:256

1001 = 1:512

1010 = 1:1024

1011 = 1:2048

1100 = 1:4096

1101 = 1:8192

1110 = 1:16384

1111 = 1:32768

PIC16(L)F18324/18344

表 26-1: 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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
TMR0L	TMR0L<7:0>								277
TMR0H	TMR0H<7:0> 或 TMR0<15:8>								277
T0CON0	T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				278
T0CON1	T0CS<2:0>			T0ASYNC	T0CKPS<3:0>				279
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					160
TMR0PPS	—	—	—	TMR0PPS<4:0>					160
ADACT	—	—	—	ADACT<4:0>					244
CLCxSEly	—	—	LCxDyS<5:0>						227
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS<1:0>		291
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	105
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	100

图注: — = 未实现位, 读为0。Timer0模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

27.0 带门控的TIMER1/3/5模块

Timer1/3/5 模块是 16 位定时器/计数器，具有以下特性：

- 16 位定时器/计数器寄存器对（TMR1H:TMR1L）
- 可编程内部或外部时钟源
- 2 位预分频器
- 用于可选比较器同步的时钟源
- 多个 Timer1 门控（计数使能）源
- 溢出时产生中断
- 溢出触发唤醒（仅限外部时钟，异步模式）
- CCP 模块捕捉/比较功能的时基
- 自动转换触发（与 CCP 配合工作）
- 可选择的门控源极性
- 门控翻转模式
- 门控单脉冲模式
- 门控值状态
- 门控事件中断

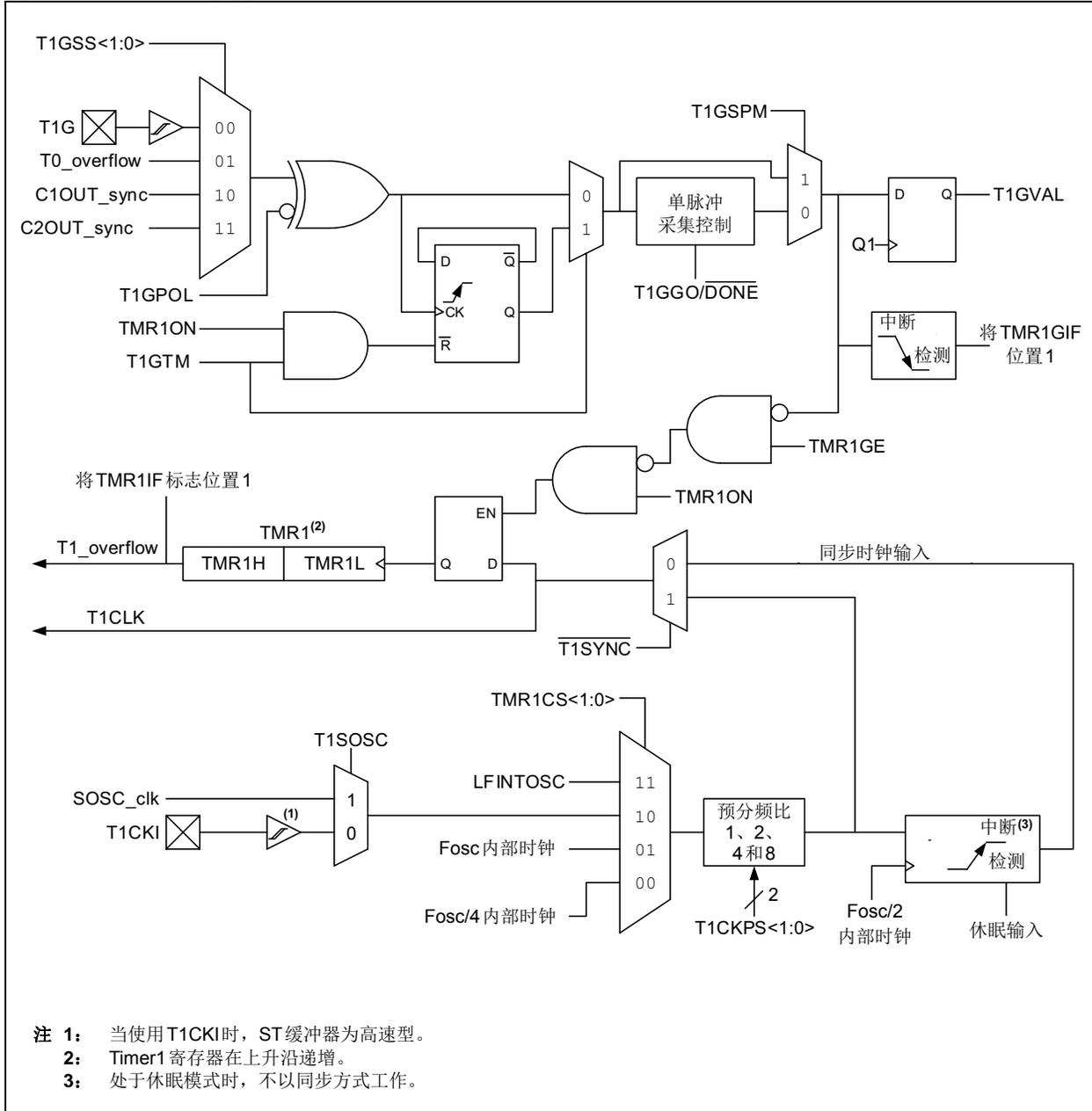
图27-1 给出了 Timer1 模块的框图。

注 1： 在具有多个定时器模块的器件中，要特别注意所使用的寄存器名称，这一点很重要。模块省略名之后的数字用于区分不同的模块。例如，T1CON和T3CON分别控制两个完全不同定时器模块相同的工作特性。

2： 在本章中，在所有工作模式下，通常指的 Timer1 模块都可以解释为同样适用于 Timerx 模块。在需要时，寄存器名称、模块信号、I/O 引脚和位名称可以使用通用标识符“x”（数字）来识别某个特定模块。

PIC16(L)F18324/18344

图27-1: TIMER1框图



27.1 Timer1 工作原理

Timer1 模块是 16 位递增定时器，可通过 TMR1H:TMR1L 寄存器对访问。写 TMR1H 或 TMR1L 会直接更新定时器。

该模块可与内部或外部时钟源配合使用，并且具有 Timer1 门控使能功能。Timer1 与 Timer1 门控使能配合使用时，定时器可对所关注两点间的时间间隔进行测量或者对这两点间的信号脉冲进行计数。不使用 Timer1 门控使能时，定时器仅测量时间间隔。

Timer1 分别通过配置 T1CON 和 T1GCON 寄存器中的 TMR1ON 和 TMR1GE 位使能。表 27-1 显示了 Timer1 使能选择。

表 27-1: TIMER1 使能选择

TMR1ON	TMR1GE	Timer1 工作状态
0	0	关闭
0	1	关闭
1	0	总是开启
1	1	计数使能

27.2 时钟源选择

T1CON 寄存器的 TMR1CS<1:0> 和 T1OSC 位用于选择 Timer1 的时钟源。表 27-2 显示了时钟源选择。TMR1H:TMR1L 寄存器对的递增频率将为时钟源的整数倍（取决于 Timer1 预分频器）。

当选择 Fosc 或 LFINTOSC 时钟源时，TMR1H:TMR1L 寄存器对将在每个时钟上升沿递增。如果在 Fosc 或 LFINTOSC 作为时钟源时读取 TMR1H:TMR1L 寄存器对，则将导致分辨率损失 2 个 LSB，这个问题可通过使用异步输入信号门控 Timer1 时钟输入来缓解（有关 Timer1 门控使能的更多信息，请参见第 26.5 节“休眠期间的操作”）。

当选择 Fosc/4 时钟源时，TMR1H:TMR1L 寄存器对将在每个指令周期递增（每 4 个 Fosc 脉冲一次）。

除内部时钟源外，Timer1 还具有专用的外部时钟输入引脚 T1CKI。T1CKI 可通过 T1CON 寄存器的 T1SYNC 位设为与系统时钟同步或异步运行。将 T1CKI 引脚用作时钟源时，TMR1H:TMR1L 寄存器对将在 T1CKI 时钟输入的上升沿递增。

注： 如果使用 Timer1 对事件进行计数，则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 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）。

表 27-2: 时钟源选择

TMR1CS<1:0>	时钟源
11	LFINTOSC
10	T1CKI 引脚上的外部时钟源
01	系统时钟 (Fosc)
00	指令时钟 (Fosc/4)

PIC16(L)F18324/18344

27.2.1 TIMER1（辅助）振荡器

引脚SOSCI（输入）和引脚SOSCO（放大器输出）之间内置了一个专用低功耗32.768 kHz振荡器电路。该内部电路设计与与外部32.768 kHz晶振结合使用。

可通过将T1CON寄存器的T1SOSC位置1来使能该振荡器电路。在休眠期间，振荡器将继续运行。

注： 振荡器在使用之前需要一定的起振和稳定时间。因此，应该将T1SOSC置1并在经过一段合适的延时后才能使用Timer1。适当的延时类似于OST延时，可通过在软件中清零TMR1IF位，然后将TMR1H:TMR1L寄存器对预设FC00h来实现。经过1024个时钟周期之后，TMR1IF标志将会被置1，指示振荡器正在运行且相当稳定。

27.3 Timer1 预分频器

Timer1有4个预分频比选项，允许对时钟输入进行1、2、4或8分频。T1CON寄存器的T1CKPS位控制预分频器计数器。对预分频器计数器不能直接进行读写操作；但是，通过写入TMR1H或TMR1L可将预分频器计数器清零。

27.4 异步模式下的Timer1操作

如果T1CON寄存器的控制位T1SYNC被置1，外部时钟输入就不同步。定时器异步于内部相位时钟进行递增计数。如果选择了外部时钟源，在休眠期间定时器将继续运行，并在溢出时产生中断以唤醒处理器。但是，用软件对定时器进行读/写操作时，要特别当心（见第27.4.1节“在异步模式下读写Timer1”）。

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

27.4.1 在异步模式下读写TIMER1

当定时器采用外部异步时钟运行时，对TMR1H或TMR1L的读操作将确保为有效读操作（由硬件实现）。但是，用户应该注意的是通过读两个8位值来读取16位定时器本身就会产生某些问题，这是因为定时器可能在两次读操作之间产生溢出。

对于写操作，建议用户直接停止定时器，然后写入所需的值。如果定时器寄存器正进行递增计数，对定时器寄存器进行写操作可能会导致写争用，这可能在TMR1H:TMR1L寄存器对中产生不可预测的值。

27.5 Timer1 门控

Timer1可配置为自由计数或用Timer1门控电路使能和禁止计数。这也称为Timer1门控使能。

Timer1门控也可由多个可选择源驱动。

27.5.1 TIMER1门控使能

通过将T1GCON寄存器的TMR1GE位置1使能Timer1门控使能模式。使用T1GCON寄存器的T1GPOL位来配置Timer1门控使能模式的极性。

使能Timer1门控使能模式时，Timer1将在Timer1时钟源的上升沿递增。禁止Timer1门控使能模式时，不会发生递增，Timer1将保持当前计数。时序详细信息请参见图27-3。

表27-3: TIMER1门控使能选择

T1CLK	T1GPOL	T1G	Timer1工作状态
↑	0	0	计数
↑	0	1	保持计数
↑	1	0	保持计数
↑	1	1	计数

27.5.2 TIMER1门控源选择

表27-4列出了Timer1门控源选择。源的选择由T1GCON寄存器的T1GSS位控制。所选输入的极性是可配置的。极性的选择由T1GCON寄存器的T1GPOL位控制。

表27-4: TIMER1门控源

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

27.5.2.1 T1G引脚门控操作

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

27.5.2.2 Timer0溢出门控操作

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

27.5.2.3 比较器C1门控操作

比较器1操作产生的输出可以选择作为Timer1的门控源。比较器1输出可以与Timer1时钟进行同步，也可以保持异步。更多信息，请参见第18.4.1节“比较器输出同步”。

27.5.2.4 比较器C2门控操作

比较器2操作产生的输出可以选择作为Timer1的门控源。比较器2输出可以与Timer1时钟进行同步，也可以保持异步。更多信息，请参见第18.4.1节“比较器输出同步”。

27.5.3 TIMER1门控翻转模式

使能Timer1门控翻转模式时，可测量Timer1门控信号的整个周期，而不是单脉冲的持续时间。

Timer1门控源经由一个触发器输送到Timer1，该触发器在信号的每个递增边沿改变状态。时序详细信息请参见图27-4。

Timer1门控翻转模式通过将T1GCON寄存器的T1GTM位置1使能。T1GTM位清零时，将清零触发器并保持清零。这对于控制测量哪个边沿是必需的。

注： 在使能翻转模式的同时改变门控极性，可能会导致不确定的操作。

27.5.4 TIMER1门控单脉冲模式

使能Timer1门控单脉冲模式时，可能会捕捉到一个单脉冲门控事件。Timer1门控单脉冲模式首先通过将T1GCON寄存器中的T1GSPM位置1来使能。接下来必须将T1GCON寄存器中的T1GGO/DONE位置1。Timer1将在Timer1门控信号的下一个递增边沿完全使能。在Timer1门控信号的下一个下降沿，将自动清零T1GGO/DONE位。不允许其他门控事件递增Timer1，直到T1GGO/DONE位再次由软件置1。时序详细信息请参见图27-5。

如果通过清零T1GCON寄存器的T1GSPM位来禁止单脉冲门控模式，则T1GGO/DONE位也会清零。

同时使能翻转模式和单脉冲模式将允许两种模式协同工作。这样就可以测量Timer1门控源的周期。时序详细信息请参见图27-6。

27.5.5 TIMER1门控值状态

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

PIC16(L)F18324/18344

27.6 Timer1 中断

Timer1 寄存器对 (TMR1H:TMR1L) 递增到 FFFFh, 然后计满返回到 0000h。当 Timer1 计满返回时, PIR1 寄存器的 Timer1 中断标志位将置 1。为允许计满返回时的中断, 必须将以下位置 1:

- T1CON 寄存器的 TMR1ON 位
- PIE1 寄存器的 TMR1IE 位
- INTCON 寄存器的 PEIE 位
- INTCON 寄存器的 GIE 位

在中断服务程序中将 TMR1IF 位清零清除中断。

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

27.7 休眠期间的 Timer1 操作

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

- 必须将 T1CON 寄存器的 TMR1ON 位置 1
- 必须将 PIE1 寄存器的 TMR1IE 位置 1
- 必须将 INTCON 寄存器的 PEIE 位置 1
- 必须将 T1CON 寄存器的 T1SYNC 位置 1
- 必须配置 T1CON 寄存器的 TMR1CS 位
- 必须配置 T1CON 寄存器的 T1SOSC 位

器件将在溢出时被唤醒并执行下一条指令。如果将 INTCON 寄存器的 GIE 位置 1, 器件将调用中断服务程序。

无论 T1SYNC 位的设置如何, 辅助振荡器都会在休眠模式下继续工作。

27.8 CCP 捕捉/比较时基

当工作在捕捉或比较模式下时, CCP 模块使用 TMR1H:TMR1L 寄存器对作为时基。

在捕捉模式下, 当发生配置的事件时, TMR1H:TMR1L 寄存器对中的值被复制到 CCPR1H:CCPR1L 寄存器对中。

在比较模式下, 当 CCPR1H:CCPR1L 寄存器对中的值与 TMR1H:TMR1L 寄存器对中的值相匹配时触发事件。该事件可以是自动转换触发信号。

更多信息, 请参见 [第29.0节“捕捉/比较/PWM模块”](#)。

27.9 CCP 自动转换触发信号

当任一 CCP 配置为触发自动转换时, 触发信号会将 TMR1H:TMR1L 寄存器对清零。该自动转换不会引起 Timer1 中断。CCP 模块仍可配置为产生 CCP 中断。

在该工作模式下, CCPR1H:CCPR1L 寄存器对变成 Timer1 的周期寄存器。

为了利用自动转换触发信号, Timer1 应进行同步, 并且应选择 $F_{osc}/4$ 作为时钟源。Timer1 的异步操作会导致错过自动转换触发信号。

如果对 TMR1H 或 TMR1L 的写操作与来自 CCP 的自动转换触发信号同时发生, 则写操作优先。

更多信息, 请参见 [第29.3.3节“自动转换触发”](#)。

图27-2: TIMER1递增边沿

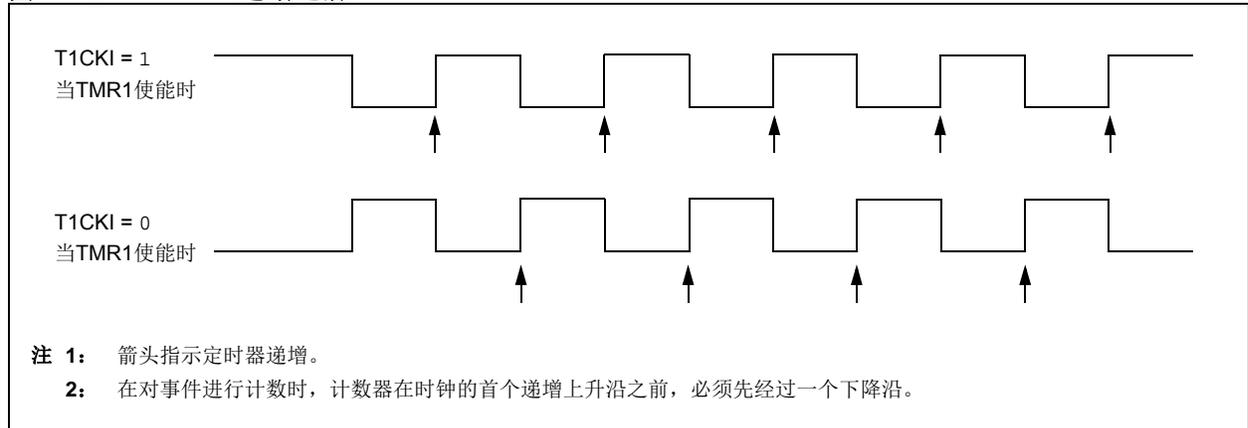
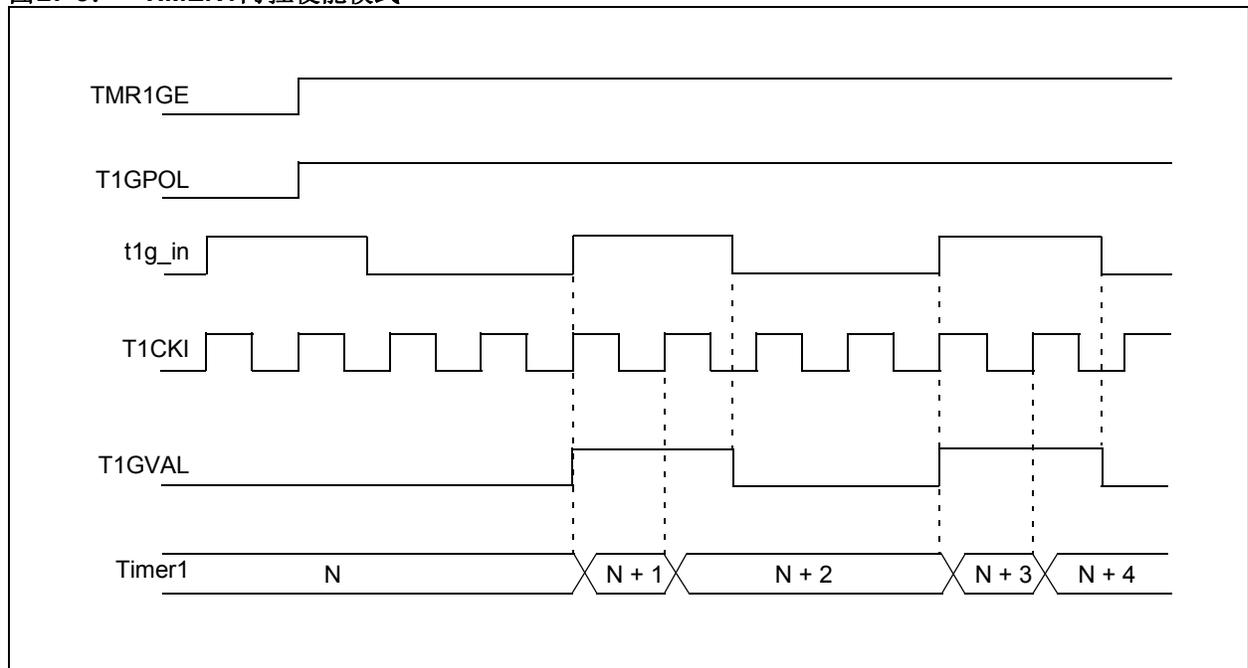


图27-3: TIMER1门控使能模式



PIC16(L)F18324/18344

图27-4: TIMER1门控翻转模式

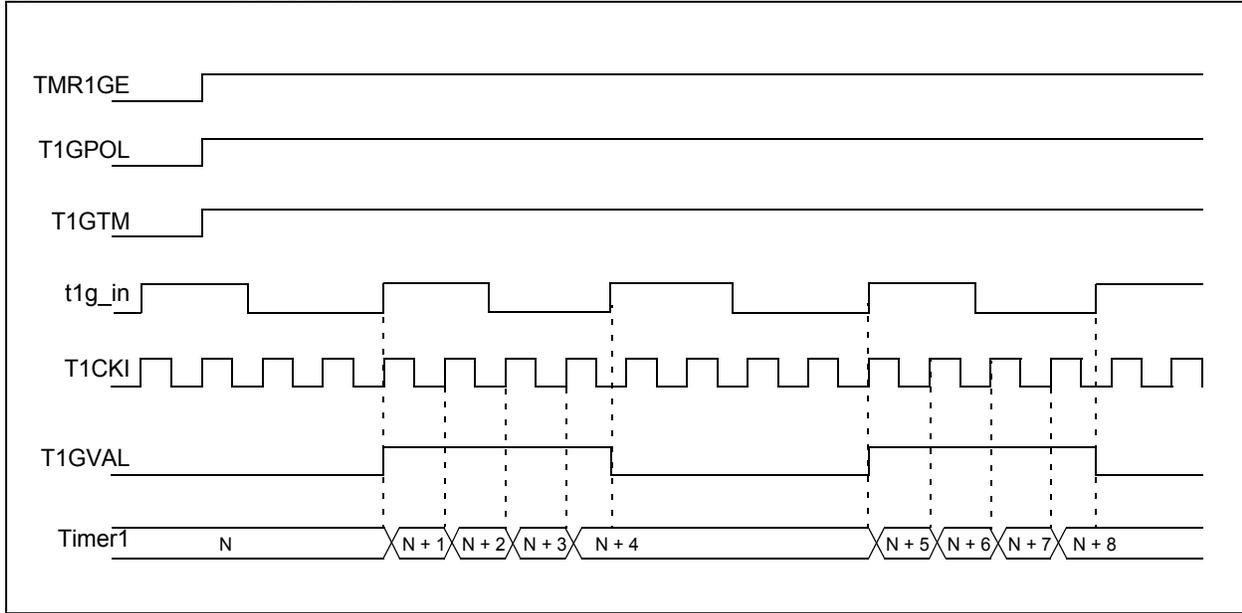


图27-5: TIMER1门控单脉冲模式

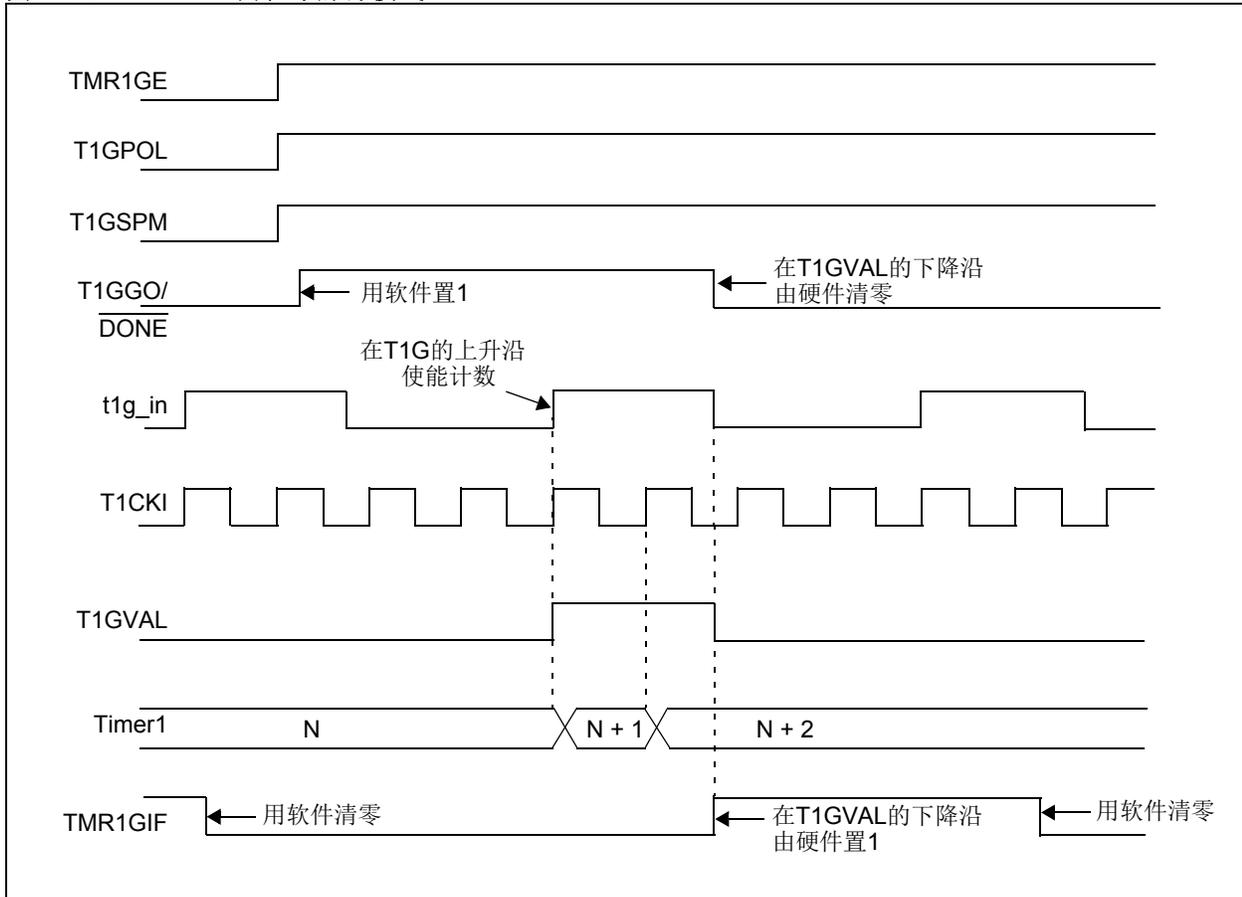
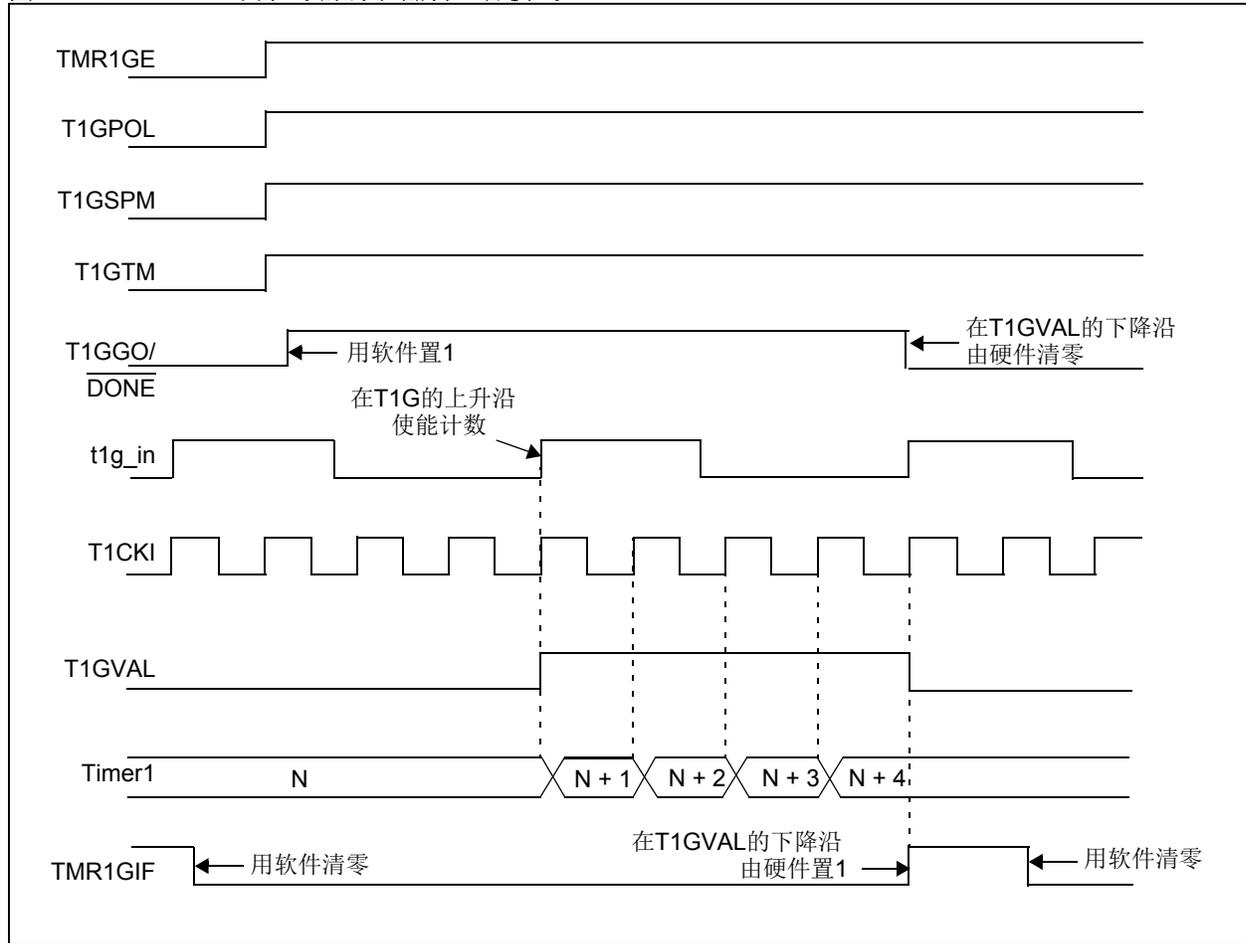


图27-6: TIMER1门控单脉冲和翻转组合模式



PIC16(L)F18324/18344

27.10 寄存器定义：Timer1/3/5控制

寄存器27-1: **TxCON⁽¹⁾**: TIMERx控制寄存器

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u
TMRxCS<1:0>	TxCKPS<1:0>	TxSOSC	$\overline{\text{TxSYNC}}$	—	TMRxON		
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-6 **TMRxCS<1:0>**: Timerx时钟源选择位

11 = Timerx时钟源为LFINTOSC

10 = Timerx时钟源为引脚或振荡器:

如果TxSOSC = 0:

来自TxCKIPPS引脚的外部时钟(上升沿触发计数)

如果TxSOSC = 1:

来自SOSC的时钟, TxSOSCI/TxSOSCO引脚上的晶振或SOSCIN输入

01 = Timerx时钟源为系统时钟(Fosc)

00 = Timerx时钟源为指令时钟(Fosc/4)

bit 5-4 **TxCKPS<1:0>**: Timerx输入时钟预分频比选择位

11 = 1:8 预分频比

10 = 1:4 预分频比

01 = 1:2 预分频比

00 = 1:1 预分频比

bit 3 **TxSOSC**: LP振荡器使能控制位

1 = 请求SOSC作为时钟源

0 = 使能TxCKI作为时钟源

bit 2 **TxSYNC**: Timerx同步控制位

$\text{TMRxCS}<1:0> = 1x$

1 = 不同步外部时钟输入

0 = 将外部时钟输入与系统时钟同步

$\text{TMRxCS}<1:0> = 0x$

该位被忽略。Timerx使用内部时钟且不执行任何其他同步。

bit 1 **未实现**: 读为0

bit 0 **TMRxON**: Timerx使能位

1 = 使能Timerx

0 = 停止Timerx并清零Timerx门控触发器

注 1: “x”表示“1”、“3”或“5”, 分别对应于Timer1/3/5寄存器。

寄存器 27-2: TxGCON⁽¹⁾: TIMERx 门控寄存器

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMRxGE	TxGPOL	TxGTM	TxGSPM	TxGGO/DONE	TxGVAL	TxGSS<1:0>	
bit 7						bit 0	

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HC = 硬件清零位

- bit 7 **TMRxGE:** Timerx 门控使能位
 如果 TMRxON = 0:
 该位被忽略
 如果 TMRxON = 1:
 1 = Timerx 计数由 Timerx 门控功能控制
 0 = Timerx 始终计数
- bit 6 **TxGPOL:** Timerx 门控极性位
 1 = Timerx 门控为高电平有效 (当门控信号为高电平时 Timerx 计数)
 0 = Timerx 门控为低电平有效 (当门控信号为低电平时 Timerx 计数)
- bit 5 **TxGTM:** Timerx 门控翻转模式位
 1 = 使能 Timerx 门控翻转模式
 0 = 禁止 Timerx 门控翻转模式并清除翻转触发器
 Timerx 门控触发器在每个上升沿翻转。
- bit 4 **TxGSPM:** Timerx 门控单脉冲模式位
 1 = 使能 Timerx 门控单脉冲模式, 并在使用该模式控制 Timerx 门控
 0 = 禁止 Timerx 门控单脉冲模式
- bit 3 **TxGGO/DONE:** Timerx 门控单脉冲采集状态位
 1 = Timerx 门控单脉冲采集就绪, 正在等待一个边沿
 0 = Timer1 门控单脉冲采集已经结束或尚未开始
 当 TxGSPM 位清零时, 该位自动清零
- bit 2 **TxGVAL:** Timerx 门控值状态位 (只读)
 指示提供给 TMRxH:TMRxL 的 Timerx 门控的当前状态 (在 Q1 锁存)
 不受 Timerx 门控使能 (TMRxGE) 的影响
- bit 1-0 **TxGSS<1:0>:** Timerx 门控源选择位
 11 = 比较器 2 的可选同步输出
 10 = 比较器 1 的可选同步输出
 01 = Timer0 溢出输出
 00 = Timerx 门控引脚

注 1: “x” 表示 “1”、“3” 或 “5”, 分别对应于 Timer1/3/5 寄存器。

PIC16(L)F18324/18344

寄存器 27-3: TMRxL⁽¹⁾: TIMERx低字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
TMRxL<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 TMRxL<7:0>: TMRx低字节位

注 1: “x”表示“1”、“3”或“5”, 分别对应于Timer1/3/5寄存器。

寄存器 27-4: TMRxH⁽¹⁾: TIMERx高字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
TMRxH<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 TMRxH<7:0>: TMRx高字节位

注 1: “x”表示“1”、“3”或“5”, 分别对应于Timer1/3/5寄存器。

PIC16(L)F18324/18344

表27-5: 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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
PIR3	OSFIF	CSWIF	TMR3GIF	TMR3IF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	108
PIE3	OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	103
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	109
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	104
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1SOSC	T1SYNC	—	TMR1ON	290
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS<1:0>		291
TMR1L	TMR1L<7:0>								292
TMR1H	TMR1H<7:0>								292
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					160
T1GPPS	—	—	—	T1GPPS<4:0>					160
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		T3SOSC	T3SYNC	—	TMR3ON	290
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/DONE	T3GVAL	T3GSS<1:0>		291
TMR3L	TMR3L<7:0>								292
TMR3H	TMR3H<7:0>								292
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					160
T3GPPS	—	—	—	T3GPPS<4:0>					160
T5CON	TMR5CS<1:0>		T5CKPS<1:0>		T5SOSC	T5SYNC	—	TMR5ON	290
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/DONE	T5GVAL	T5GSS<1:0>		291
TMR5L	TMR5L<7:0>								292
TMR5H	TMR5H<7:0>								292
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					160
T5GPPS	—	—	—	T5GPPS<4:0>					160
T0CON0	T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>				278
CMxCON0	CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC	188
CCPTMRS	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		309
CCPxCON	CCPxEN	—	CCPxOUT	CCPxFMT	CCPxMODE<3:0>				306
CLCxSELY	—	—	LCxDyS<5:0>						227
ADACT	—	—	—	ADACT<4:0>					244

图注: — = 未实现位, 读为0。Timer1模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

PIC16(L)F18324/18344

28.0 TIMER2/4/6模块

Timer2/4/6 模块是 8 位定时器，具有以下特性：

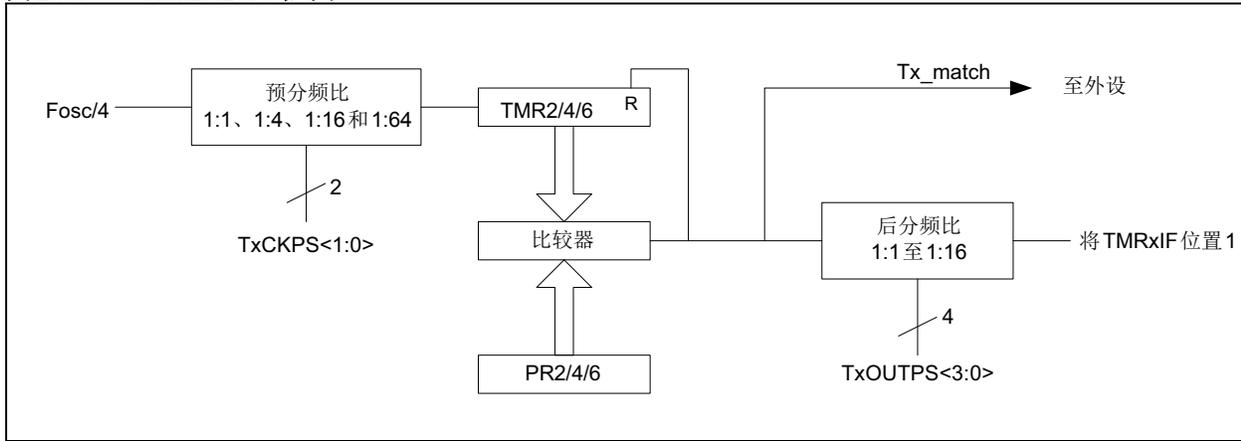
- 8 位定时器和周期寄存器（分别为 TMR2/4/6 和 PR2/4/6）
- 可读写（所有寄存器）
- 可软件编程的预分频比（1:1、1:4、1:16 和 1:64）
- 可软件编程的后分频比（1:1 至 1:16）
- TMR2/4/6 与 PR2/4/6 匹配时产生中断
- 可选择用作 MSSP1 模块的移位时钟

Timer2/4/6 框图请参见图 28-1。

注 1: 在具有多个定时器模块的器件中，要特别注意所使用的寄存器名称，这一点很重要。模块缩略名之后的数字用于区分不同的模块。例如，T2CON 和 T4CON 分别控制两个完全不同定时器模块相同的工作特性。

2: 在本章中，在所有工作模式下，通常指的 Timer2 模块都可以解释为同样适用于 Timerx 模块。在需要时，寄存器名称、模块信号、I/O 引脚和位名称可以使用通用标识符“x”（数字）来识别某个特定模块。

图 28-1: TIMER2/4/6 框图



28.1 Timer2工作原理

Timer2模块的时钟输入是系统指令时钟（FOSC/4）。

4位计数器/预分频器提供将时钟直接输入、4分频和16分频的预分频比选项。这些选项通过T2CON寄存器的预分频比控制位T2CKPS<1:0>进行选择。在每个时钟周期，TMR2的值都会与周期寄存器PR2中的值进行比较。如果两者匹配，则比较器生成一个匹配信号作为定时器输出。该信号也会将TMR2的值在下一个周期复位为00h，并驱动输出计数器/后分频器（见第28.2节“Timer2中断”）。

TMR2和PR2寄存器均可直接读写。在任何器件复位时，TMR2寄存器都会清零，而PR2寄存器则初始化为FFh。发生以下事件时，预分频器和后分频器计数器均会清零：

- 对TMR2寄存器进行写操作
- 对T2CON寄存器进行写操作
- 上电复位（POR）
- 欠压复位（BOR）
- MCLR复位
- 看门狗定时器（WDT）复位
- 堆栈上溢复位
- 堆栈下溢复位
- RESET指令

注： 写T2CON时TMR2不会清零。

28.2 Timer2中断

Timer2也可以产生可选的器件中断。Timer2输出信号（TMR2与PR2匹配时）为4位计数器/后分频器提供输入。该计数器产生TMR2匹配中断，对应的中断标志位为PIR1寄存器的TMR2IF位。可以通过将PIE1寄存器的TMR2匹配中断允许位TMR2IE置1来允许该中断。

可以通过T2CON寄存器的后分频比控制位T2OUTPS<3:0>在16个后分频比选项（从1:1至1:16）中选择其一。

28.3 Timer2输出

TMR2的未经分频的输出主要用于CCP模块，它用作CCP模块在PWM模式下工作时的时基。

可选择将Timer2作为MSSP1模块在SPI模式下工作时的移位时钟源。更多信息，请参见第30.0节“主同步串行端口（MSSP1）模块”

28.4 休眠期间的Timer2操作

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

PIC16(L)F18324/18344

28.5 寄存器定义：Timer2/4/6控制

寄存器 28-1: **TxCON⁽¹⁾**: TIMERx控制寄存器

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	TxOUTPS<3:0>				TMRxON	TxCKPS<1:0>	
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **未实现:** 读为0
- bit 6-3 **TxOUTPS<3:0>**: Timerx输出后分频比选择位
 - 1111 = 1:16后分频比
 - 1110 = 1:15后分频比
 - 1101 = 1:14后分频比
 - 1100 = 1:13后分频比
 - 1011 = 1:12后分频比
 - 1010 = 1:11后分频比
 - 1001 = 1:10后分频比
 - 1000 = 1:9后分频比
 - 0111 = 1:8后分频比
 - 0110 = 1:7后分频比
 - 0101 = 1:6后分频比
 - 0100 = 1:5后分频比
 - 0011 = 1:4后分频比
 - 0010 = 1:3后分频比
 - 0001 = 1:2后分频比
 - 0000 = 1:1后分频比
- bit 2 **TMRxON**: Timer2使能位
 - 1 = 使能Timerx
 - 0 = 禁止Timerx
- bit 1-0 **TxCKPS<1:0>**: Timerx时钟预分频比选择位
 - 11 = 预分频比为64
 - 10 = 预分频比为16
 - 01 = 预分频比为4
 - 00 = 预分频比为1

注 1: “x”指的是“2”、“4”或“6”，分别对应于Timer2/4/6寄存器。

寄存器 28-2: TMRx⁽¹⁾: TIMERx 计数寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMRx<7:0>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 **TMRx<7:0>**: TMRx 计数器 bit 7..0

注 1: “x” 表示 “2”、“4” 或 “6”，分别对应于 Timer2/4/6 寄存器。

寄存器 28-3: PRx: TIMERx 周期寄存器⁽¹⁾

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
PRx<7:0>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 **PRx<7:0>**: TMRx 计数器 bit 7..0

当 TMRx = PRx 时, 下一个时钟将复位计数器; 计数器周期为 (PRx+1)

注 1: “x” 表示 “2”、“4” 或 “6”，分别对应于 Timer2/4/6 寄存器。

PIC16(L)F18324/18344

表28-1: 与TIMER2/4/6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	寄存器所在页
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
PIR2	TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF	107
PIE2	TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	102
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		296
TMR2	TMR2<7:0>								297
PR2	PR2<7:0>								297
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		296
TMR4	TMR4<7:0>								297
PR4	PR4<7:0>								297
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		296
TMR6	TMR6<7:0>								297
PR6	PR6<7:0>								297
ADACT	—	—	—	ADACT<4:0>					244
PWMTMRS	—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		195
CLCxSELY	—	—	LCxDyS<5:0>						227

图注: — = 未实现位, 读为0。Timer2/4/6模块不使用阴影单元。

29.0 捕捉/比较/PWM模块

捕捉/比较/PWM模块是允许用户计时和控制不同事件，以及产生脉宽调制（PWM）信号的外设。在捕捉模式下，外设允许对事件的持续时间进行计时。当超过预先确定的时间时，比较模式允许用户触发一个外部事件。PWM模式可以产生不同频率和占空比的脉宽调制信号。

该器件系列包含四个标准捕捉/比较/PWM模块（CCP1、CCP2、CCP3和CCP4）。

所有CCP模块的捕捉和比较功能均相同。

29.1 CCP/PWM时钟选择

PIC16(L)F18324/18344器件允许每个CCP和PWM模块选择控制模块的定时器源。每个模块的选择是独立的。

由于最多存在3个具有自动重载功能的8位定时器（Timer2、Timer4和Timer6），CCP和PWM模块上的PWM模式可以使用这些定时器中的任意一个。

CCPTMRS寄存器用于选择要使用的定时器。

注 1: 在具有多个CCP模块的器件中，要特别注意所使用的寄存器名称，这一点很重要。模块缩略名之后的数字用于区分不同的模块。例如，CCP1CON和CCP2CON分别控制两个完全不同CCP模块相同的工作特性。

2: 在本章中，在所有工作模式下，通常指的CCP模块都可以解释为同样适用于CCPx模块。在需要时，寄存器名称、模块信号、I/O引脚和位名称可以使用通用标识符“x”（数字）来识别某个特定模块。

29.2 捕捉模式

捕捉模式使用16位Timer0或16位Timer1资源。当捕捉源上发生事件时，16位CCPRxH:CCPRxL寄存器对将分别捕捉和存储TMR0H:TMR0L或TMR1H:TMR1L寄存器对的16位值。这些事件定义如下，可通过CCPxCON寄存器的CCPxMODE<3:0>位进行配置：

- 每个下降沿
- 每个上升沿
- 每4个上升沿
- 每16个上升沿

进行捕捉时，PIR4寄存器的中断请求标志位CCPxIF被置1。该中断标志必须用软件清零。如果在CCPRxH和CCPRxL寄存器对中的值被读取之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖。

图29-1给出了捕捉操作的简化框图。

29.2.1 捕捉源

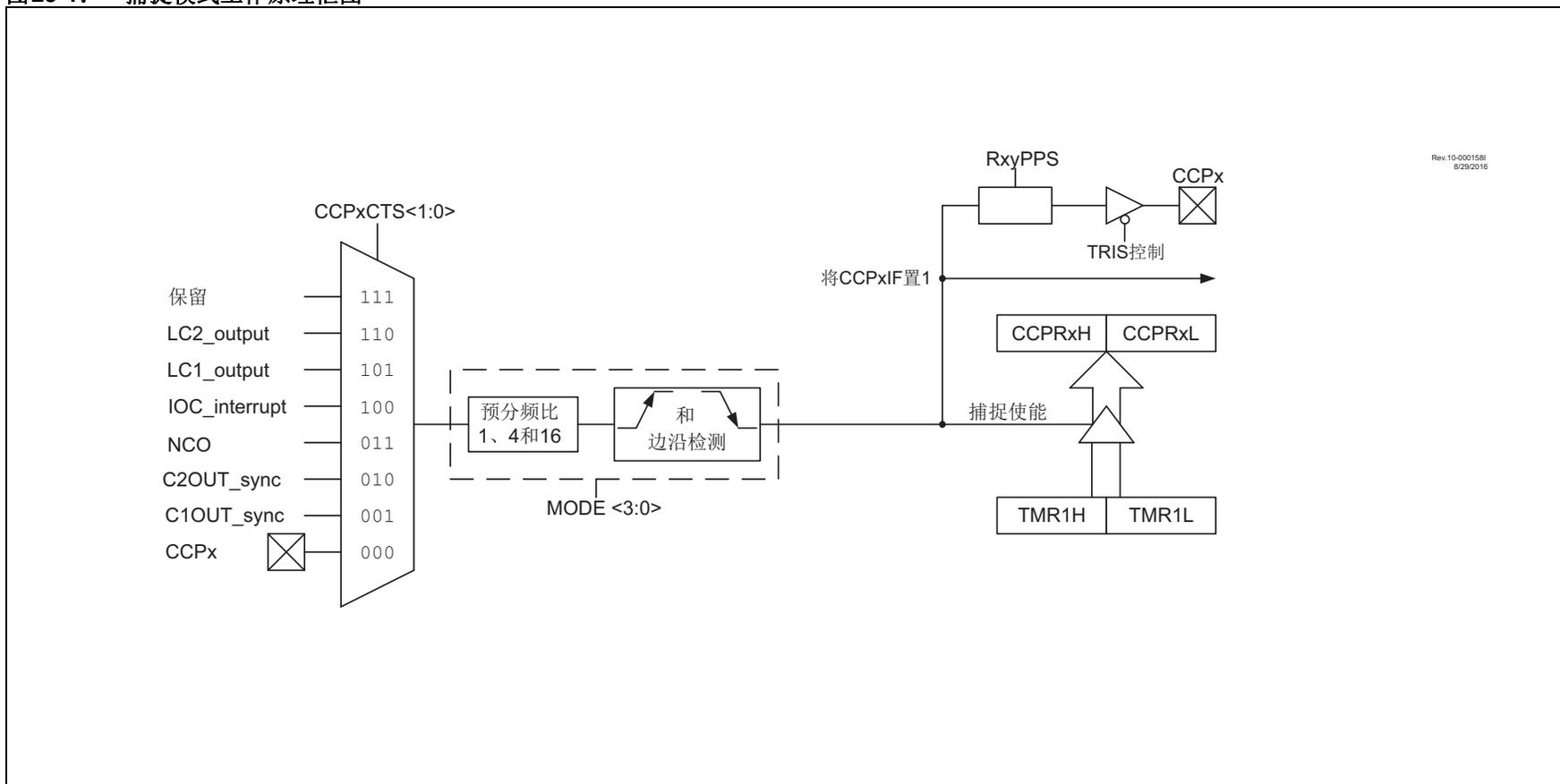
在捕捉模式下，应该通过将相应的TRIS控制位置1把CCPx引脚配置为输入引脚。

注: 如果CCPx引脚被配置为输出，则写端口将产生一次捕捉条件。

捕捉源通过配置CCPxCAP寄存器的CCPxCTS<3:0>位来选择。可以选择以下源：

- CCPxPPS输入
- C1_output
- C2_output
- NCO_output
- IOC_interrupt
- LC1_output
- LC2_output
- LC3_output
- LC4_output

图29-1: 捕捉模式工作原理框图



29.2.2 TIMER1/3/5模式资源

为使CCP模块使用捕捉特性，Timer1/3/5必须运行在定时器模式或同步计数器模式下。在异步计数器模式下，可能无法进行捕捉操作。

关于配置Timer1/3/5的更多信息，请参见第27.0节“带门控的Timer1/3/5模块”。

29.2.3 软件中断模式

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应保持PIE4寄存器的CCPxIE中断允许位清零以避免错误中断。此外，用户应在工作模式的任何改变之后清零PIR4寄存器的CCPxIF中断标志位。

注： 在捕捉模式下，Timer1/3/5时钟源不能由系统时钟（Fosc）提供。为在捕捉模式下识别CCPx引脚上的触发事件，Timer1/3/5时钟源必须由指令时钟（Fosc/4）或外部时钟源提供。

29.2.4 CCP预分频器

通过CCPxCON寄存器的CCPxMODE<3:0>位，可以指定4种预分频比设置。每当关闭CCP模块，或者CCP模块不在捕捉模式下时，预分频器计数器就会被清零。任何复位都会将预分频器计数器清零。

从一个捕捉预分频比切换到另一个捕捉预分频比不会清零预分频器，而且可能产生一次错误中断。为避免此意外操作，可在改变预分频比前通过清零CCPxCON寄存器来关闭模块。例29-1给出了执行此功能的代码。

例29-1： 切换捕捉预分频比

```
BANKSEL CCPxCON      ;Set Bank bits to point
                      ;to CCPxCON
CLRF    CCPxCON       ;Turn CCP module off
MOVLW  NEW_CAPT_PS   ;Load the W reg with
                      ;the new prescaler
MOVWF  CCPxCON       ;move value and CCP ON
MOVWF  CCPxCON       ;Load CCPxCON with this
                      ;value
```

29.2.5 休眠期间的捕捉操作

捕捉模式的正常工作依赖于Timer1/3/5模块。有两个时钟源选项可用于在捕捉模式下驱动Timer1/3/5模块。它可由指令时钟（Fosc/4）驱动，或由外部时钟源驱动。

当Timer1/3/5时钟源由Fosc/4提供时，Timer1/3/5不会在休眠期间递增。当器件被从休眠模式唤醒时，Timer1/3/5将从先前状态继续工作。

当Timer1/3/5通过外部时钟源提供时钟时，捕捉模式将在休眠模式期间继续工作。

PIC16(L)F18324/18344

29.3 比较模式

比较模式使用 16 位 Timer1/3/5 资源。CCPRxH:CCPRxL 寄存器对的 16 位值会不断与 TMR1/3/5H:TMR1/3/5L 寄存器对的 16 位值进行比较。当发生匹配时，将发生以下事件之一：

- 翻转 CCPx 输出
- 将 CCPx 输出置 1
- 将 CCPx 输出清零
- 产生自动转换触发信号
- 产生软件中断

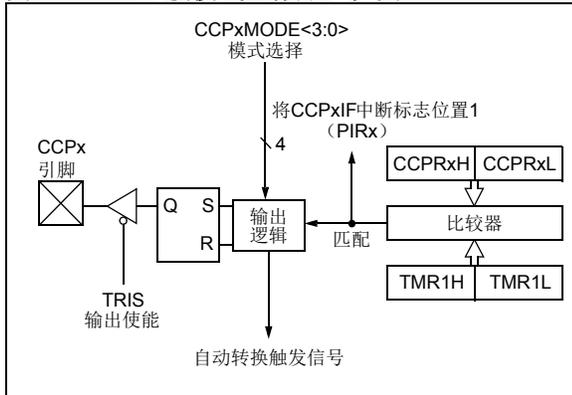
引脚的动作由 CCPxCON 寄存器的 CCPxMODE<3:0> 控制位的值决定。同时，中断标志位 CCPxIF 置 1，并且触发 ADC 转换（如果选择了的话）。

所有比较模式都能产生中断和触发 ADC 转换。

图 29-2 给出了比较操作的简化框图。

注： 当 CCP 配置为使用“发生匹配时输出电平翻转”的比较模式（CCPxMODE<3:0> 位 = 0010）且参考定时器的输入时钟预分频比设置为 1:1 以外的值时，CCP 的输出电平将翻转多次，最终稳定在逻辑电平 0。为避免出现这种情况，定时器输入时钟预分频比选择位必须设置为 1:1（TxCKPS = 00）。

图 29-2: 比较模式工作原理框图



29.3.1 CCPx 引脚配置

用户必须通过将相关的 TRIS 位清零并通过 RxyPPS 寄存器定义适当的输出引脚，将 CCPx 引脚配置为输出。更多详细信息，请参见第 13.0 节“外设引脚选择（PPS）模块”。

CCP 输出也可用作其他外设的输入。

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

29.3.2 TIMER1/3/5 模式资源

在比较模式下，Timer1/3/5 必须运行在定时器模式或同步模式下。在异步模式下，可能无法进行比较操作。

关于配置 Timer1/3/5 的更多信息，请参见第 27.0 节“带门控的 Timer1/3/5 模块”。

注： 在比较模式下，Timer1/3/5 时钟源不能由系统时钟（Fosc）提供。为在比较模式下识别 CCPx 引脚上的触发事件，Timer1/3/5 时钟源必须由指令时钟（Fosc/4）或外部时钟源提供。

29.3.3 自动转换触发

所有 CCPx 模式都将 CCP 中断标志位（CCPxIF）置 1。当发生匹配此标志位置 1 时，如果选择 CCP 模块作为转换触发源，将会触发自动转换。

更多信息，请参见第 22.2.5 节“自动转换触发”。

注： 通过在产生自动转换触发信号的时钟边沿和使定时器复位的时钟边沿之间更改 CCPRxH 和 CCPRxL 寄存器对的内容来移除匹配条件，可以避免复位发生。

29.3.4 休眠期间的比较操作

由于 Fosc 在休眠模式下关闭，比较模式在休眠模式下将不能正常工作，除非定时器正在运行。器件将在发生中断（如果允许）时唤醒。

29.3.5 比较中断

当 CCPRxH:CCPRxL 寄存器对和 TMR1/3/5H:TMR1/3/5L 寄存器对发生匹配时，CCPxIF 中断标志将置 1。如果器件处于休眠模式且允许了中断（CCPxIE = 1），则器件将唤醒（假定 Timer1 在休眠模式下工作）。

29.4 PWM概述

脉宽调制 (PWM) 是一种通过在完全开启和完全关闭状态之间进行快速切换而为负载供电的方案。PWM 信号类似于方波，信号的高电平部分视为开启状态，信号的低电平部分视为关闭状态。高电平部分（也称为脉宽）可以随时间而变，并以步为单位进行定义。施加的步数越多（这会增大脉宽），为负载提供的功率就越大。施加的步数降低时（这会缩短脉宽），提供的功率就越小。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 分辨率定义为可以在单个 PWM 周期中出现的最大步数。分辨率越高，就可以越精确地控制脉宽时间，从而更精确地控制施加在负载上的功率。

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，施加的功率就越低；占空比越高，施加的功率就越高。

图 29-3 给出了 PWM 信号的典型波形。

29.4.1 标准 PWM 操作

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

- PR2/4/6 寄存器
- T2/4/6CON 寄存器
- CCPRxL 寄存器
- CCPxCON 寄存器

图 29-4 给出了 PWM 操作的简化框图。

注： 要启用 CCPx 引脚上的 PWM 输出，必须清零相应的 TRIS 位。

图 29-3: CCP PWM 输出信号

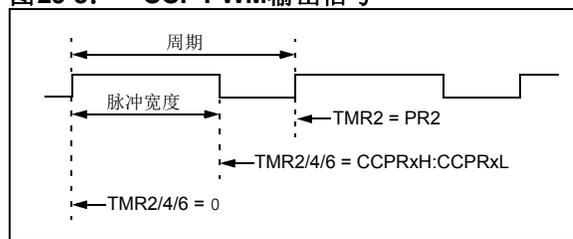
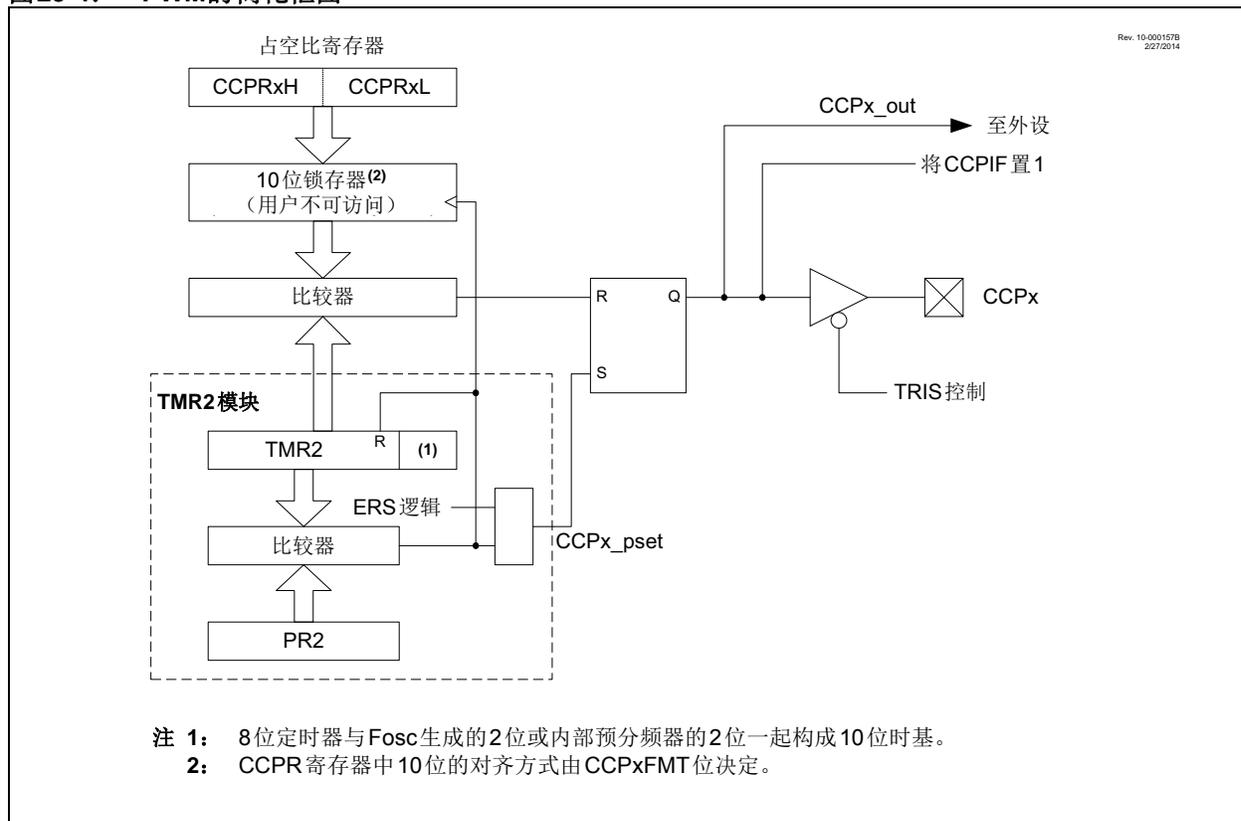


图 29-4: PWM 的简化框图



PIC16(L)F18324/18344

29.4.2 设置PWM操作

当将CCP模块配置为标准PWM操作时，可采用以下步骤：

1. 使用所需输出引脚 RxyPPS 控制来选择 CCPx 作为源，并通过将关联的 TRIS 位置 1 而禁止 CCPx 引脚输出驱动器。
2. 将 PWM 周期值装入 PR2 寄存器。
3. 通过将相应值装入 CCPxCON 寄存器，将 CCP 模块配置为 PWM 模式。
4. 将 PWM 占空比值装入 CCPRxL 寄存器和 CCPRxH 寄存器位并配置 CCPxCON 寄存器的 CCPxFMT 位来设置适当的寄存器对齐方式。
5. 配置并启动 Timer2/4/6。
 - 清零 PIR4 寄存器的 TMR2/4/6IF 中断标志位。请参见下面的“注”。
 - 用定时器预分频值配置 T2/4/6CON 寄存器的 T2/4/6CKPS 位。
 - 通过将 T2/4/6CON 寄存器的 TMR2/4/6ON 位置 1 使能定时器。
6. 使能 PWM 输出引脚：
 - 等待，直到定时器溢出且 PIR4 寄存器的 TMR2/4/6IF 位置 1。请参见下面的“注”。
 - 通过将相关的 TRIS 位清零，使能 CCPx 引脚输出驱动器。

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

29.4.3 TIMER2/4/6 定时器资源

PWM 标准模式使用 8 位 Timer2/4/6 定时器资源来指定 PWM 周期。

29.4.4 PWM 周期

PWM 周期可通过 Timer2/4/6 的 PRx 寄存器来指定。PWM 周期可由公式 29-1 计算。

公式 29-1: PWM 周期

$$PWM \text{ 周期} = [(PR2x) + 1] \cdot 4 \cdot TOSC \cdot (TMR2/4/6 \text{ 预分频值})$$

注： $TOSC = 1/FOSC$

当 TMR2/4/6 中的值与 PR2/4/6 中的值相等时，在下一个递增周期将发生以下 3 个事件：

- TMR2/4/6 被清零
- CCPx 引脚被置 1。（例外情况：如果 PWM 占空比 = 0%，引脚将不会被置 1。）
- PWM 占空比从 CCPRxL/H 寄存器对传送到 10 位缓冲区中。

注： 在确定 PWM 频率时不会用到定时器后分频比（见第 28.1 节“Timer2 工作原理”）。

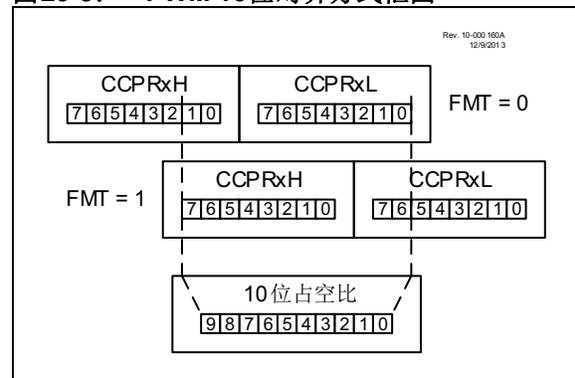
29.4.5 PWM 占空比

通过将 10 位值写入 CCPRxH:CCPRxL 寄存器对来指定 PWM 占空比。10 位值的对齐方式由 CCPxCON 寄存器的 CCPxFMT 位决定（见图 29-5）。CCPRxH:CCPRxL 寄存器对可以在任意时间写入，但是在 PR2/4/6 和 TMR2/4/6 发生匹配之前，占空比值不会被锁存到 10 位缓冲区中。

公式 29-2 用于计算 PWM 脉冲宽度。

公式 29-3 用于计算 PWM 占空比。

图 29-5: PWM 10 位对齐方式框图



公式29-2: 脉冲宽度

$$\text{脉冲宽度} = (\text{CCPRxH:CCPRxL 寄存器对}) \cdot T_{\text{OSC}} \cdot (\text{TMR2 预分频值})$$

公式29-3: 占空比

$$\text{占空比} = \frac{(\text{CCPRxH:CCPRxL 寄存器对})}{4(\text{PR2} + 1)}$$

CCPRxH:CCPRxL 寄存器对和一个2位的内部锁存器用于为PWM占空比提供双重缓冲。这种双重缓冲可以提供无毛刺PWM操作。

8位定时器TMR2/4/6寄存器与2位内部系统时钟（F_{OSC}）或预分频器的2位一起构成10位时基。如果Timer2/4/6预分频比设置为1:1，则使用系统时钟。

当10位时基与CCPRxH:CCPRxL寄存器相匹配时，CCPx引脚被清零（见图29-4）。

29.4.6 PWM分辨率

PWM分辨率以位数来表示，定义在单个PWM周期中可实现的最大离散步阶数。例如，10位分辨率将可得到1024个离散的步阶，而8位分辨率将可得到256个离散的步阶。

当PRx为255时，最大PWM分辨率为10位。分辨率是PRx寄存器值的函数，如公式29-4所示。

公式29-4: PWM分辨率

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

注： 如果脉冲宽度值比周期长，则指定的PWM引脚将保持不变。

表29-1: PWM频率和分辨率示例（F_{OSC} = 20 MHz）

PWM 频率	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频值	16	4	1	1	1	1
PRx 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率（位）	10	10	10	8	7	6.6

表29-2: PWM频率和分辨率示例（F_{OSC} = 8 MHz）

PWM 频率	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频值	16	4	1	1	1	1
PRx 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率（位）	8	8	8	6	5	5

29.4.7 休眠模式下的操作

在休眠模式下，TMR2/4/6寄存器将不会递增，模块状态也不会改变。如果CCPx引脚正在驱动一个值，则会继续驱动该值。当器件被唤醒时，TMR2/4/6将从先前状态继续。

29.4.8 改变系统时钟频率

PWM频率是由系统时钟频率得到的。系统时钟频率的任何改变将导致PWM频率的改变。更多详细信息，请参见第7.0节“振荡器模块”。

29.4.9 复位的影响

任何复位都将强制所有端口为输入模式，并强制CCPx寄存器为其复位状态。

PIC16(L)F18324/18344

29.5 寄存器定义：CCP控制

寄存器 29-1: CCPxCON: CCPx控制寄存器

R/W-0/0	U-0	R-x/x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPxEN	—	CCPxOUT	CCPxFMT	CCPxMODE<3:0>			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **CCPxEN:** CCP模块使能位
0 = 禁止CCP
1 = 使能CCP
- bit 6 **未实现:** 读为0
- bit 5 **CCPxOUT:** CCPx输出数据 (只读) 位
- bit 4 **CCPxFMT:** CCPW (脉宽) 对齐方式位
CCPxMODE = 捕捉模式
不使用
CCPxMODE = 比较模式
不使用
CCPxMODE = PWM模式
0 = 右对齐格式
1 = 左对齐格式
- bit 3-0 **CCPxMODE<3:0>:** CCPx模式选择位⁽¹⁾
1111 = PWM模式
1110 = 保留
1101 = 保留
1100 = 保留

1011 = 比较模式: 输出脉冲 (0-1-0); 清零TMR1/3/5
1010 = 比较模式: 输出脉冲 (0-1-0)
1001 = 比较模式: 发生比较匹配时清零输出。
1000 = 比较模式: 发生比较匹配时置1输出。

0111 = 捕捉模式: CCPx输入的每16个上升沿
0110 = 捕捉模式: CCPx输入的每4个上升沿
0101 = 捕捉模式: CCPx输入的每个上升沿
0100 = 捕捉模式: CCPx输入的每个下降沿

0011 = 捕捉模式: CCPx输入的每个边沿
0010 = 比较模式: 发生匹配时翻转输出
0001 = 比较模式: 发生匹配时翻转输出; TMR1/3/5清零
0000 = 捕捉/比较/PWM关闭 (复位CCPx模块)

注 1: 如果选择CCPx作为ADC触发源, 所有模式会将CCPxIF位置1且触发ADC转换。

PIC16(L)F18324/18344

寄存器 29-2: CCPxCAP: 捕捉输入选择寄存器

U-0	U-0	U-0	U-0	R/W-0/x	R/W-0/x	R/W-0/x	R/W-0/x
—	—	—	—	CCPxCTS<3:0>			
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4

未实现: 读为0

bit 3-0

CCPxCTS<3:0>: CCPx捕捉模式数据选择位

CTS<3:0>	CCP1CAP 捕捉输入	CCP2CAP 捕捉输入	CCP3CAP 捕捉输入	CCP4CAP 捕捉输入
0000	CCP1PPS	CCP2PPS	CCP3PPS	CCP4PPS
0001	C1OUT			
0010	C2OUT			
0011	NCO1			
0100	IOC_interrupt			
0101	LC1_output			
0110	LC2_output			
0111	LC3_output			
1000	LC4_output			
1001	保留			
...				
1111				

PIC16(L)F18324/18344

寄存器 29-3: CCPRxL 寄存器: CCPx 寄存器低字节

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRxL<7:0>:							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 CCPxMODE = 捕捉模式
CCPRxL<7:0>: TMR1/3/5L 的捕捉值
CCPxMODE = 比较模式
CCPRxL<7:0>: LSB 与 TMR1/3/5L 进行比较
 当 CCPxFMT = 0 时, CCPxMODE = PWM 模式
CCPRxL<7:0>: CCPW<7:0> —— 脉冲宽度低 8 位
 当 CCPxFMT = 1 时, CCPxMODE = PWM 模式
CCPRxL<7:6>: CCPW<1:0> —— 脉冲宽度低 2 位
CCPRxL<5:0>: 不使用。

寄存器 29-4: CCPRxH 寄存器: CCPx 寄存器高字节

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRxH<7:0>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置1	0 = 清零	

bit 7-0 CCPxMODE = 捕捉模式
CCPRxH<7:0>: TMR1/3/5H 的捕捉值
CCPxMODE = 比较模式
CCPRxH<7:0>: MSB 与 TMR1/3/5H 进行比较
 当 CCPxFMT = 0 时, CCPxMODE = PWM 模式
CCPRxH<7:2>: 不使用
CCPRxH<1:0>: CCPW<9:8> —— 脉冲宽度高 2 位
 当 CCPxFMT = 1 时, CCPxMODE = PWM 模式
CCPRxH<7:0>: CCPW<9:2> —— 脉冲宽度高 8 位

寄存器 29-5: CCPTMRS: CCP定时器控制寄存器

R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7-6 **C4TSEL<1:0>**: CCP4 捕捉、比较和PWM模式定时器选择位
表29-4给出了具体选择。
- bit 5-4 **C3TSEL<1:0>**: CCP3 捕捉、比较和PWM模式定时器选择位
表29-4给出了具体选择。
- bit 3-2 **C2TSEL<1:0>**: CCP2 捕捉、比较和PWM模式定时器选择位
表29-4给出了具体选择。
- bit 1-0 **C1TSEL<1:0>**: CCP1 捕捉、比较和PWM模式定时器选择位
表29-4给出了具体选择。

表29-3: 定时器选择

CxTSEL<1:0>	基于CCPxMODE<3:0>的工作模式	
	捕捉/比较	PWM
00	TMR0	TMR2
01	TMR1	
10	TMR3	TMR4
11	TMR5	TMR6

PIC16(L)F18324/18344

表29-4: 与CCPx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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	109
PIE4	CWG2IE	CWGIE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	104
CCPxCON	CCPxEN	—	CCPxOUT	CCPxFMT	CCPxMODE<3:0>				306
CCPxCAP	—	—	—	—	—	CCPxCTS<2:0>			307
CCPRxL	CCPRx<7:0>								308
CCPRxH	CCPRx<15:8>								308
CCPTMRS	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		309
CCP1PPS	—	—	—	CCP1PPS<4:0>					160
CCP2PPS	—	—	—	CCP2PPS<4:0>					160
CCP3PPS	—	—	—	CCP3PPS<4:0>					160
CCP4PPS	—	—	—	CCP4PPS<4:0>					160
ADACT	—	—	—	ADACT<4:0>					244
CLCxSEly	—	—	LCxDyS<5:0>					227	
CWGxDAT	—	—	—	—	DAT<3:0>				213
MDSRC	—	—	—	—	MDMS<3:0>				270
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272

图注: — = 未实现位, 读为0。CCP模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

30.0 主同步串行端口（MSSP1）模块

30.1 MSSP1 模块概述

主同步串行端口（MSSP1）模块是用于同其他外设或单片机进行通信的串行接口。这些外设可以是串行EEPROM、移位寄存器、显示驱动器和A/D转换器等。

MSSP1 模块有以下两种工作模式：

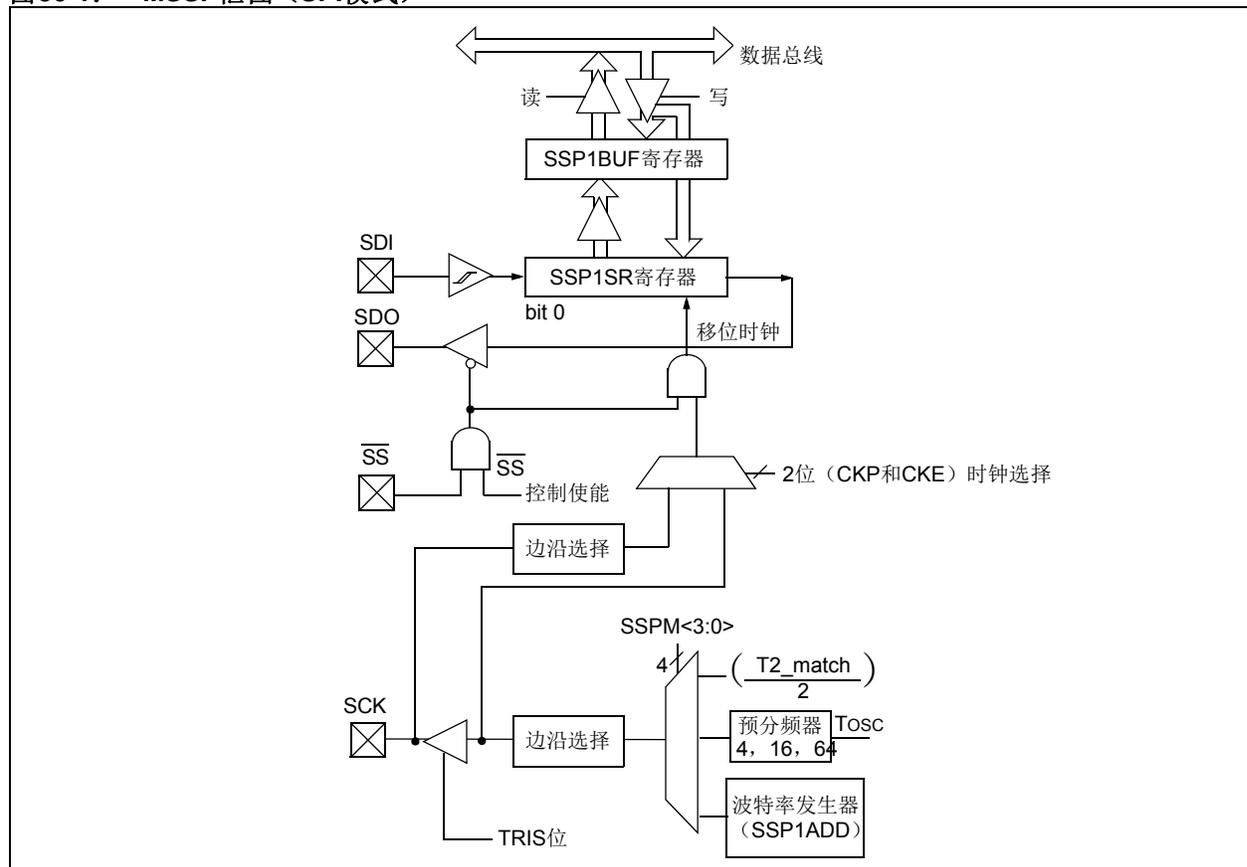
- 串行外设接口（Serial Peripheral Interface, SPI）
- I²C

SPI 接口支持以下模式和特性：

- 主模式
- 从模式
- 时钟极性
- 从选择同步（仅限从模式）
- 从器件的菊花链连接

图30-1 给出了SPI接口模块的框图。

图30-1: MSSP框图（SPI模式）



PIC16(L)F18324/18344

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
- 可选的SDA保持时间

图30-2给出了主模式下I²C接口模块的框图。图30-3给出了从模式下I²C接口模块的框图。

图30-2: MSSP框图 (I²C主模式)

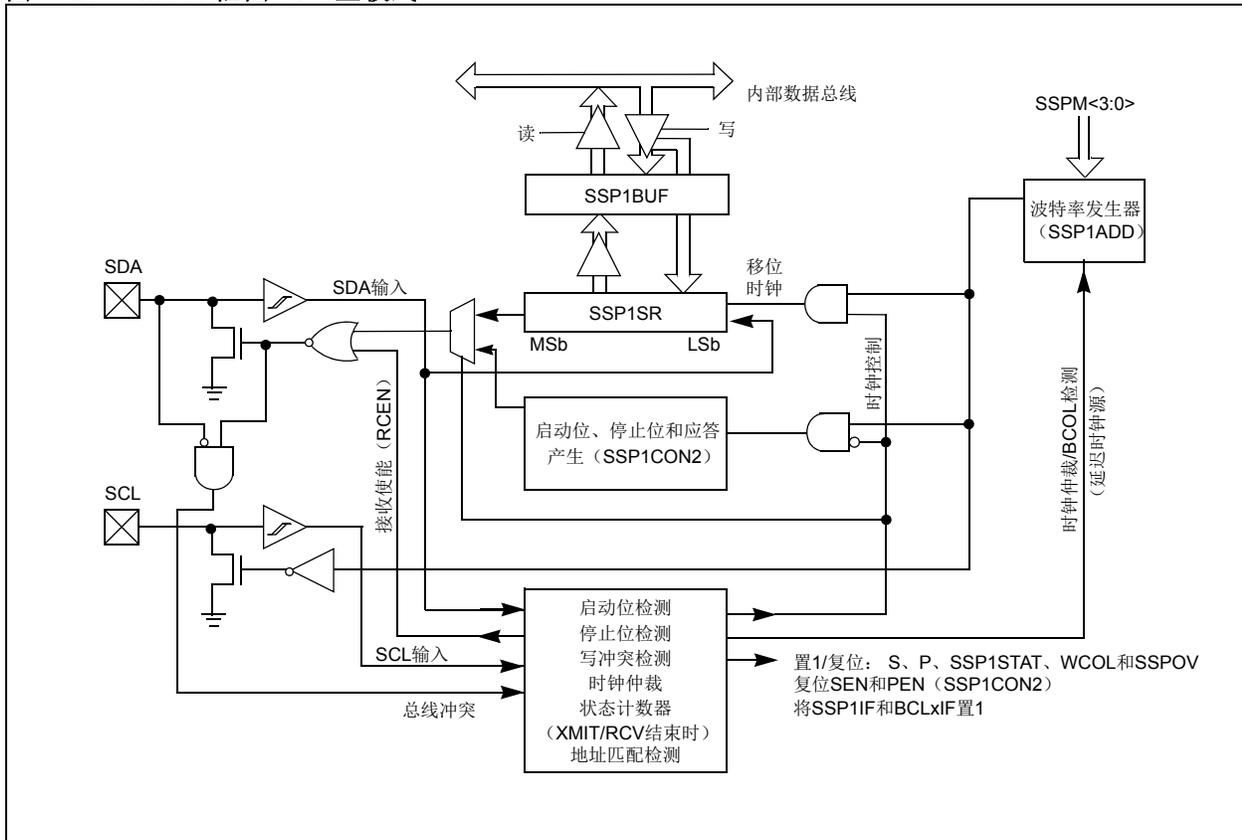
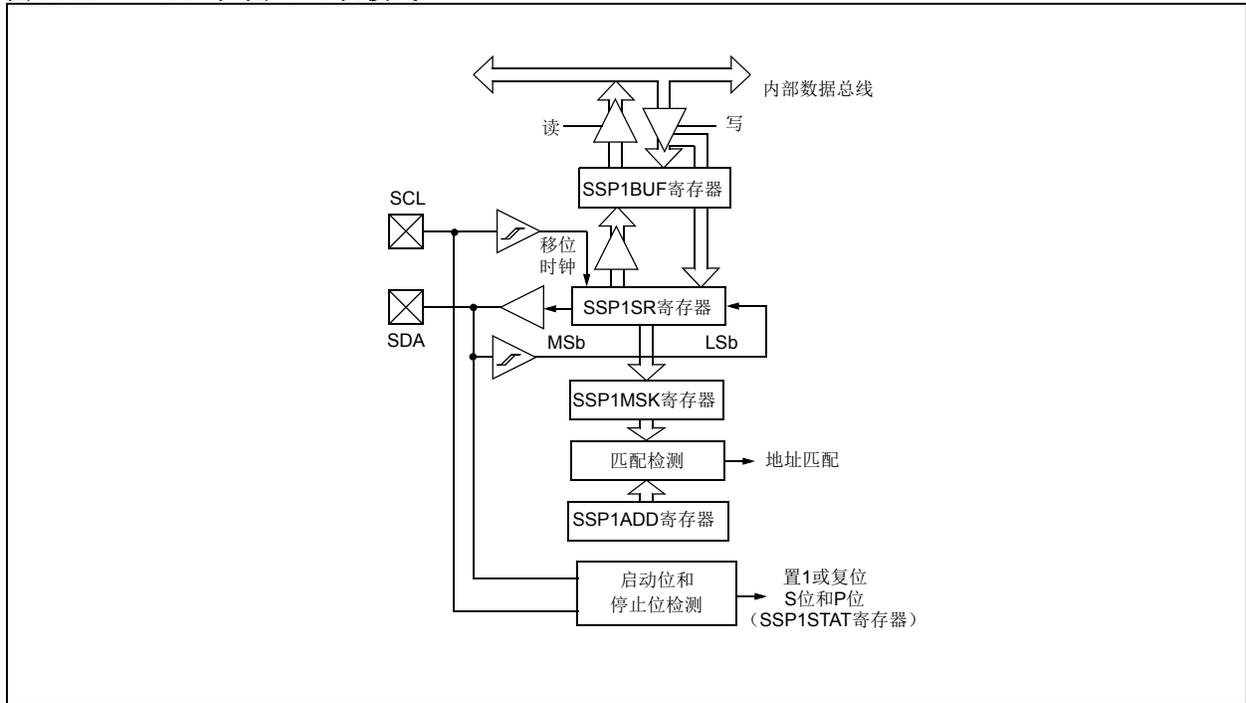


图 30-3: MSSP 框图 (I²C 从模式)



PIC16(L)F18324/18344

30.2 SPI模式概述

串行外设接口（SPI）总线是以全双工模式工作的同步串行数据通信总线。器件在由主器件启动通信的主/从器件环境中进行通信。从器件通过称为从选择的片选进行选择控制。

SPI总线规定了4种信号连接：

- 串行时钟（Serial Clock, SCK）
- 串行数据输出（Serial Data Out, SDO）
- 串行数据输入（Serial Data In, SDI）
- 从选择（Slave Select, SS）

图30-1给出了MSSP模块在SPI模式下工作时的框图。

SPI总线工作时使用单个主器件和一个或多个从器件。使用多个从器件时，可使用独立的从选择连接来寻址每个从器件。

图30-4给出了主器件和多个从器件之间的典型连接。

主器件每次仅选择一个从器件。大多数从器件都具有三态输出，所以在未选择它们时，它们的输出信号与总线断开。

数据发送涉及到两个移位寄存器，它们大小都为8位，一个在主器件中，一个在从器件中。数据总是每次移出一位，最高有效位（MSb）先移出。与此同时，新的最低有效位（LSb）会被移入同一寄存器。

图30-5给出了分别配置为主器件和从器件的两个处理器之间的典型连接。

数据在所设定的时钟边沿从两个移位寄存器移出，并在相反的时钟边沿锁存。

主器件通过它的SDO输出引脚发送信息，并由该引脚所连接的从器件SDI输入引脚接收。从器件通过它的SDO输出引脚发送信息，并由该引脚所连接的主器件SDI输入引脚接收。

要开始进行通信，主器件需要先送出时钟信号。主器件和从器件应配置为相同的时钟极性。

主器件会通过从它的移位寄存器中发送MSb而启动数据发送。从器件会从同一条线上读取该位，并将它保存到其移位寄存器的LSb中。

在每个SPI时钟周期中，会发生全双工数据传输。这意味着，在主器件从其移位寄存器中发送出MSb（在其SDO引脚上），从器件读取该位并将它保存为其移位寄存器的LSb的同时，从器件也会从其移位寄存器中发送出MSb（在其SDO引脚上），而主器件也会读取该位并将它保存为其移位寄存器的LSb。

在移出8位之后，主器件和从器件就交换了寄存器值。

如果需要交换更多数据，移位寄存器中会装入新数据，并重复该过程。

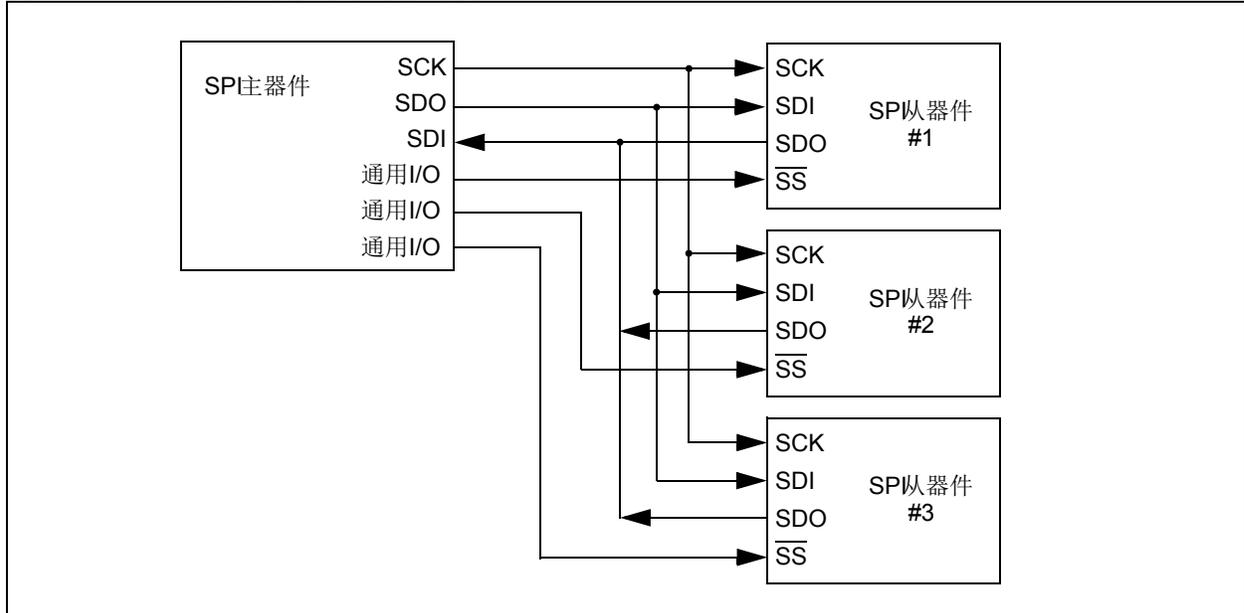
数据有意义还是无意义（无效数据），取决于应用软件。这就导致以下三种数据传输情形：

- 主器件发送有用数据，从器件发送无效数据。
- 主器件发送有用数据，从器件发送有用数据。
- 主器件发送无效数据，从器件发送有用数据。

必须在8的倍数个时钟周期中传输数据。在没有更多数据需要传输时，主器件会停止发送时钟信号，并取消选择从器件。

每个与总线连接、但未通过其从选择线选择的从器件，都必须忽略时钟和数据传输信号，并且不能发送自己的任何数据。

图30-4: SPI主器件和多个从器件连接



30.2.1 SPI模式寄存器

MSSP1 模块有5个寄存器用于SPI工作模式。它们包括:

- MSSP1 状态寄存器 (SSP1STAT)
- MSSP1 控制寄存器1 (SSP1CON1)
- MSSP1 控制寄存器3 (SSP1CON3)
- MSSP1 数据缓冲寄存器 (SSP1BUF)
- MSSP1 地址寄存器 (SSP1ADD)
- MSSP1 移位寄存器 (SSP1SR) (不可直接访问)

SSP1CON1 和 SSP1STAT 是在 SPI 模式下工作的控制寄存器和状态寄存器。SSP1CON1 寄存器是可读写的。SSP1STAT 的低6位是只读的。SSP1STAT 的高2位是可读写的。

在一种 SPI 主模式下, SSP1ADD 中可以装入在波特率发生器中使用的值。关于波特率发生器的更多信息, 请参见第30.7节“波特率发生器”。

SSP1SR 是用来将数据移入和移出的移位寄存器。SSP1BUF 用于间接访问 SSP1SR 寄存器。SSP1BUF 是缓冲寄存器, 可用于数据字节的写入或读出。

在接收操作中, SSP1SR 和 SSP1BUF 共同构成一个缓冲接收器。当 SSP1SR 接收到一个完整字节时, 字节会被传输到 SSP1BUF 中, 并且 SSP1IF 中断标志会置1。

在发送期间, SSP1BUF 是不可缓冲的。对 SSP1BUF 的写操作将同时写入 SSP1BUF 和 SSP1SR。

30.2.2 SPI工作模式

初始化 SPI 时需要指定几个选项。可以通过编程相应的控制位 (SSP1CON1<3:0> 和 SSP1STAT<7:6>) 来指定这些选项。这些控制位用于指定以下选项:

- 主模式 (SCK 作为时钟输出)
- 从模式 (SCK 作为时钟输入)
- 时钟极性 (SCK 的空闲状态)
- 数据输入采样阶段 (数据输出时间的中间或末尾)
- 时钟边沿 (在 SCK 的上升沿/下降沿输出数据)
- 时钟速率 (仅限主模式)
- 从选择模式 (仅限从模式)

要启用串口, SSP1CON1 寄存器的 SSP 使能位 SSPEN 必须置1。要复位或重新配置 SPI 模式, 先将 SSPEN 位清零, 重新初始化 SSP1CONx 寄存器, 然后再将 SSPEN 位置1。这会将 SDI、SDO、SCK 和 SS 引脚配置为串口引脚。要将引脚用作串口功能, 必须正确设置其中一些引脚的数据方向位 (在 TRIS 寄存器中):

- 必须将 SDI 对应的 TRIS 位置 1
- 必须将 SDO 对应的 TRIS 位清零
- 必须将 SCK (主模式) 对应的 TRIS 位清零
- 必须将 SCK (从模式) 对应的 TRIS 位置 1
- 必须将 SS 对应的 TRIS 位置 1

对于不需要的串口功能, 可通过将相应的数据方向 (TRIS) 寄存器编程为相反值来改写。

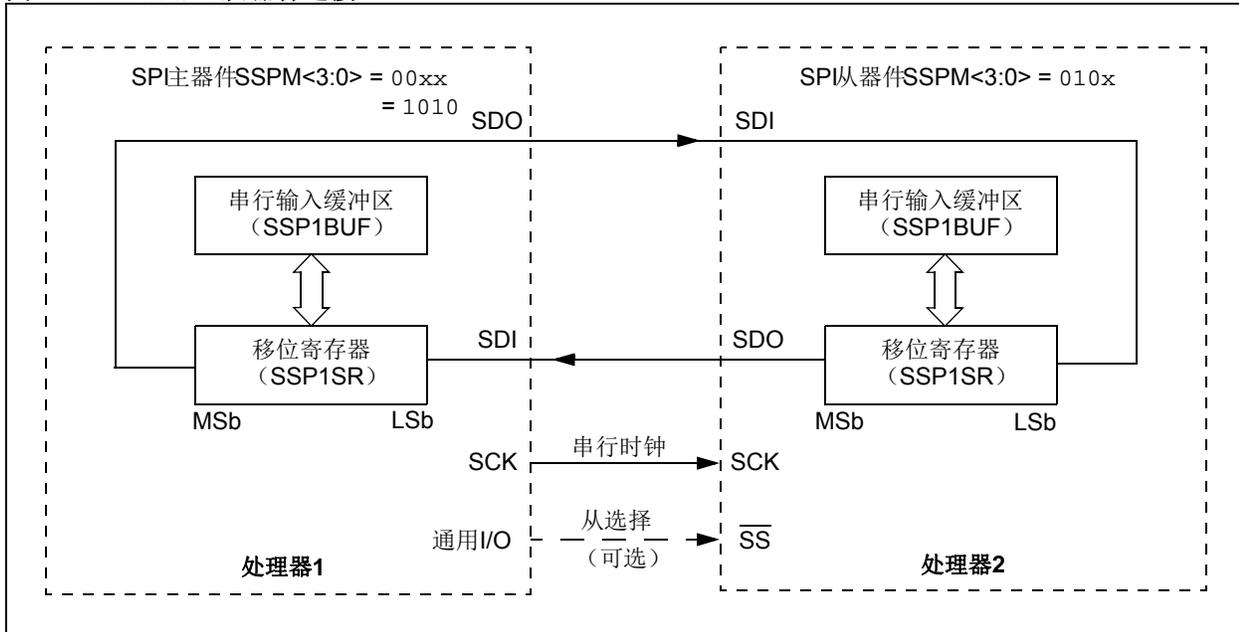
PIC16(L)F18324/18344

MSSP1 由一个发送/接收移位寄存器 (SSP1SR) 和一个缓冲寄存器 (SSP1BUF) 组成。SSP1SR 将数据移入/移出器件, 先移位 MSb。SSP1BUF 会保存先前写入 SSP1SR 的数据, 直到接收数据就绪为止。一旦 8 位数据接收完毕, 该字节就被移入 SSP1BUF 寄存器。然后, SSP1STAT 寄存器的缓冲区满检测位 BF 和中断标志位 SSP1IF 被置 1。在发送 / 接收数据期间对 SSP1BUF 寄存器的任何写操作都将被忽略, 同时 SSP1CON1 寄存器的写冲突检测位 WCOL 被置 1。用户软件必须将 WCOL 位清零, 才能使后续对 SSP1BUF 寄存器的写操作成功完成。

为确保应用软件能接收有效数据, 在下一个要发送的数据字节写入 SSP1BUF 之前, 读取 SSP1BUF 中现有的数据。SSP1STAT 寄存器的缓冲区满位 BF 用于指示何时 SSP1BUF 装入了接收到的数据 (发送完成)。SSP1BUF 中的数据被读取后, BF 位被清零。如果 SPI 仅作为一个发送器, 则不必理会该数据。一般来说, MSSP 中断用于确定发送/接收何时结束。如果不打算使用中断方法, 用软件查询的方法同样可确保不会发生写冲突。

SSP1SR 不能直接读写, 只能通过寻址 SSP1BUF 寄存器来进行访问。

图30-5: SPI主/从器件连接



30.2.3 SPI主模式

因为主器件控制SCK线，所以它可以在任意时刻启动数据传输。主器件根据软件协议确定从器件（图30-5中的处理器2）在何时广播数据。

在主模式下，数据一写入SSP1BUF寄存器就发送/接收。如果只打算将SPI作为接收器，则可以禁止SDO输出（将其编程设置为输入）。SSP1SR寄存器按所设定的时钟速率，对SDI引脚上的信号进行连续移入。每接收到一个字节，就将其装入SSP1BUF寄存器，就像接收到普通字节一样（中断和状态位相应置1）。

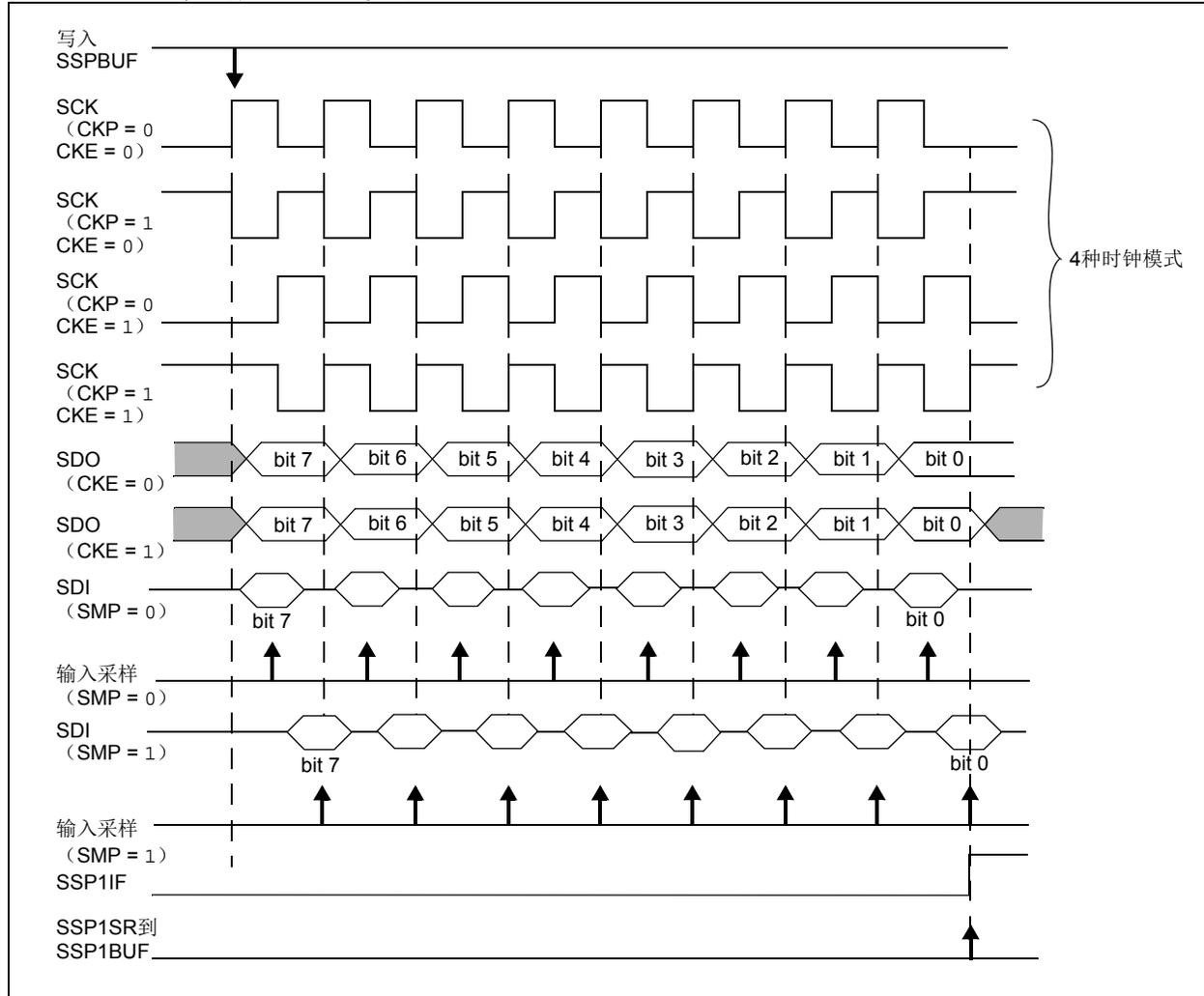
通过适当地设定SSP1CON1寄存器的CKP位和SSP1STAT寄存器的CKE位，可以选择时钟极性。图30-6、图30-8、图30-9和图30-10给出了SPI通信的波形图，其中MSb先发送。在主模式下，SPI时钟速率（比特率）可由用户编程为以下几种之一：

- $F_{osc}/4$ （或 T_{cy} ）
- $F_{osc}/16$ （或 $4 * T_{cy}$ ）
- $F_{osc}/64$ （或 $16 * T_{cy}$ ）
- Timer2输出/2
- $F_{osc}/(4 * (SSPADDD + 1))$

图30-6给出了主模式的波形图。

当CKE位置1时，SDO数据在SCK上出现时钟边沿前一直有效。图中所示为基于SMP位状态的输入采样的变化。图中给出了将接收到的数据装入SSP1BUF的时刻。

图30-6: SPI模式波形图（主模式）



PIC16(L)F18324/18344

30.2.4 SPI从模式

在从模式下，当SCK上出现外部时钟脉冲时发送和接收数据。当最后一位数据被锁存时，SSP1IF中断标志位置1。

在SPI从模式下使能该模块前，时钟线必须处于相应的空闲状态。时钟线可通过读SCK引脚来查看。空闲状态由SSP1CON1寄存器的CKP位决定。

在从模式下，外部时钟由SCK引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。

在休眠模式下，从器件仍可发送/接收数据。移位寄存器通过SCK引脚输入提供时钟，当接收到一个字节时，器件会产生中断。如果允许中断，器件会从休眠模式唤醒。

30.2.4.1 菊花链配置

SPI总线有时会采用菊花链配置进行连接。第一个从器件的输出与第二个从器件的输入连接，第二个从器件的输出与第三个从器件的输入连接，依此类推。最后一个从器件的输出与主器件的输入连接。在第二组时钟脉冲期间，每个从器件会送出在第一组时钟脉冲期间所接收数据的精确副本。整个链充当一个大的通信移位寄存器。菊花链功能只需要从主器件引出一条从选择线。

图30-7给出了在SPI模式下工作时典型菊花链连接的框图。

在菊花链配置中，从器件只需要总线上最近的一个字节。将SSP1CON3寄存器的BOEN位置1时，即使尚未读取前一个字节，也允许数据写入SSP1BUF寄存器。这使软件可以忽略不适用于它的数据。

30.2.5 从选择同步

从选择也可以用于对通信进行同步。从选择线会保持高电平，直到主器件准备好进行通信。当从选择线下拉为低电平时，从器件就知道新的数据发送正在启动。

如果从器件未能正确地接收到通信，它会在从选择线恢复为高电平状态、数据发送结束时发生复位。然后，从器件会在从选择线再次下拉为低电平时准备好接收新的发送数据。如果不使用从选择线，则会存在从器件最终与主器件脱离同步的风险。如果从器件丢失了某个位，则在之后的数据发送中，它将总是偏离一位。使用从选择线可以让从器件和主器件在每次发送开始时保持同步。

\overline{SS} 引脚允许器件工作于同步从模式。SPI必须处于从模式，并使能SS引脚控制（SSP1CON1<3:0>=0100）。

当 \overline{SS} 引脚为低电平时，使能数据的发送和接收，同时驱动SDO引脚。

当 \overline{SS} 引脚变为高电平时，即使是在字节的发送过程中，也不再驱动SDO引脚，而是将其变成悬空输出状态。根据具体应用，可能需要使用外部上拉/下拉电阻。

- | |
|--|
| <p>注 1: 当SPI处于从模式且使能\overline{SS}引脚控制（SSP1CON1<3:0>=0100）时，如果\overline{SS}引脚设置为VDD，SPI模块将会复位。</p> <p>2: 当SPI用于从模式且CKE置1时，用户必须使能\overline{SS}引脚控制。</p> <p>3: 工作于SPI从模式时，SSP1STAT寄存器的SMP位必须保持清零。</p> |
|--|

当SPI模块复位时，位计数器会被强制设为0。这可以通过将 \overline{SS} 引脚强制设为高电平或清零SSPEN位实现。

图30-7: SPI菊花链连接

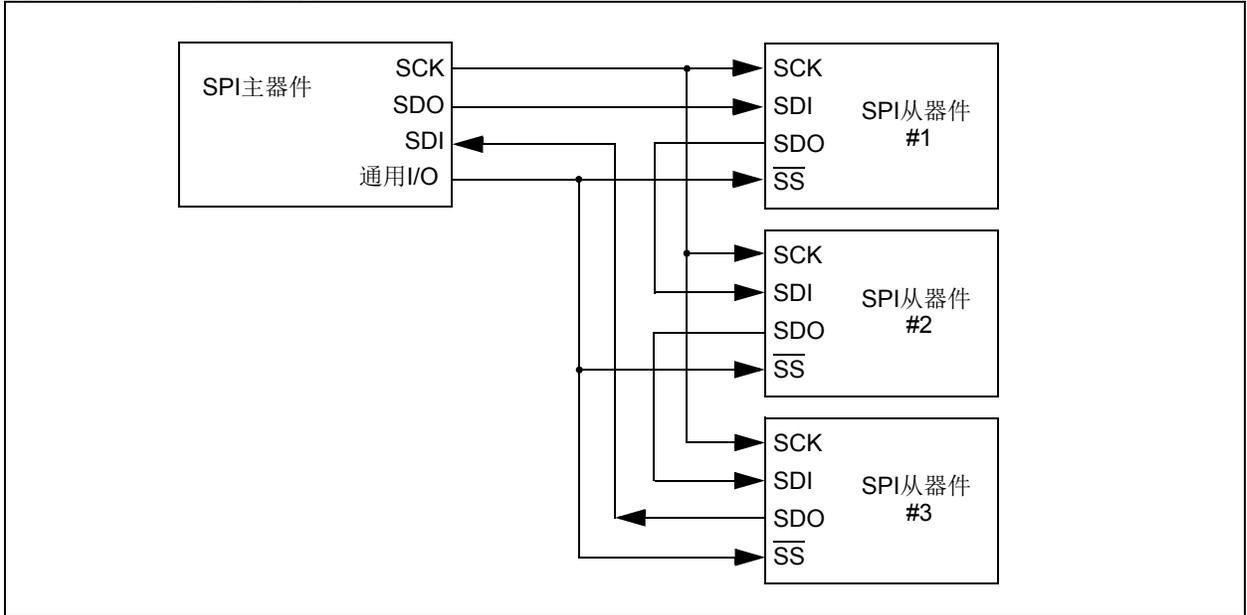
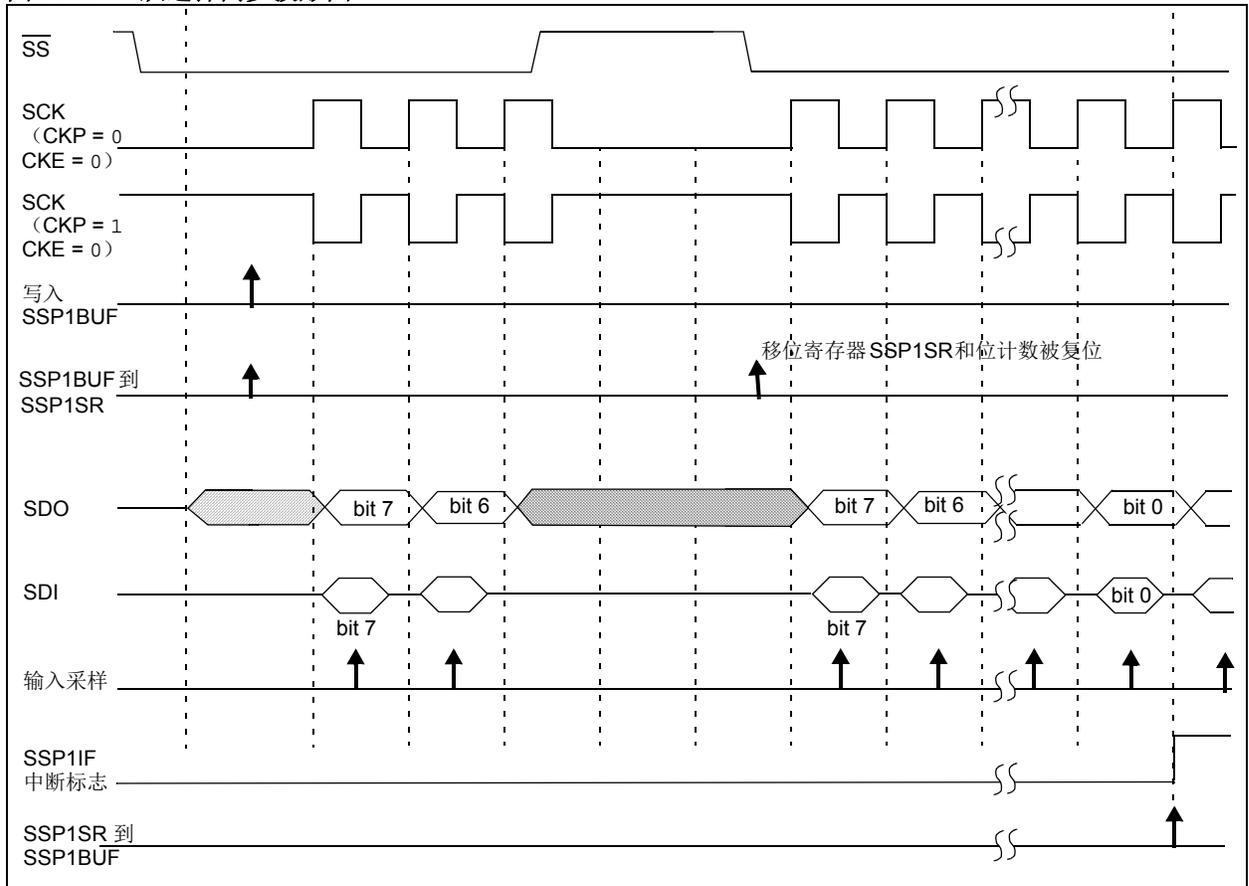


图30-8: 从选择同步波形图



PIC16(L)F18324/18344

图30-9: SPI模式波形图 (从模式, CKE = 0)

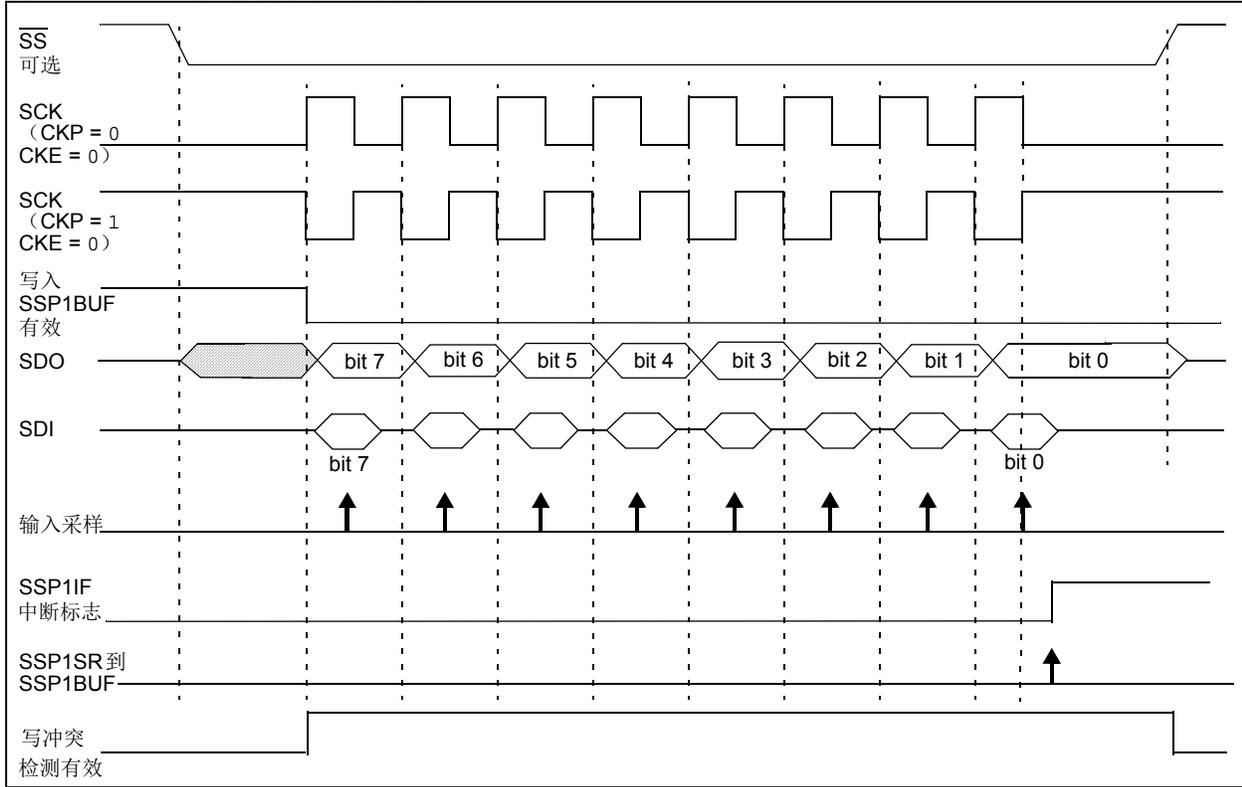
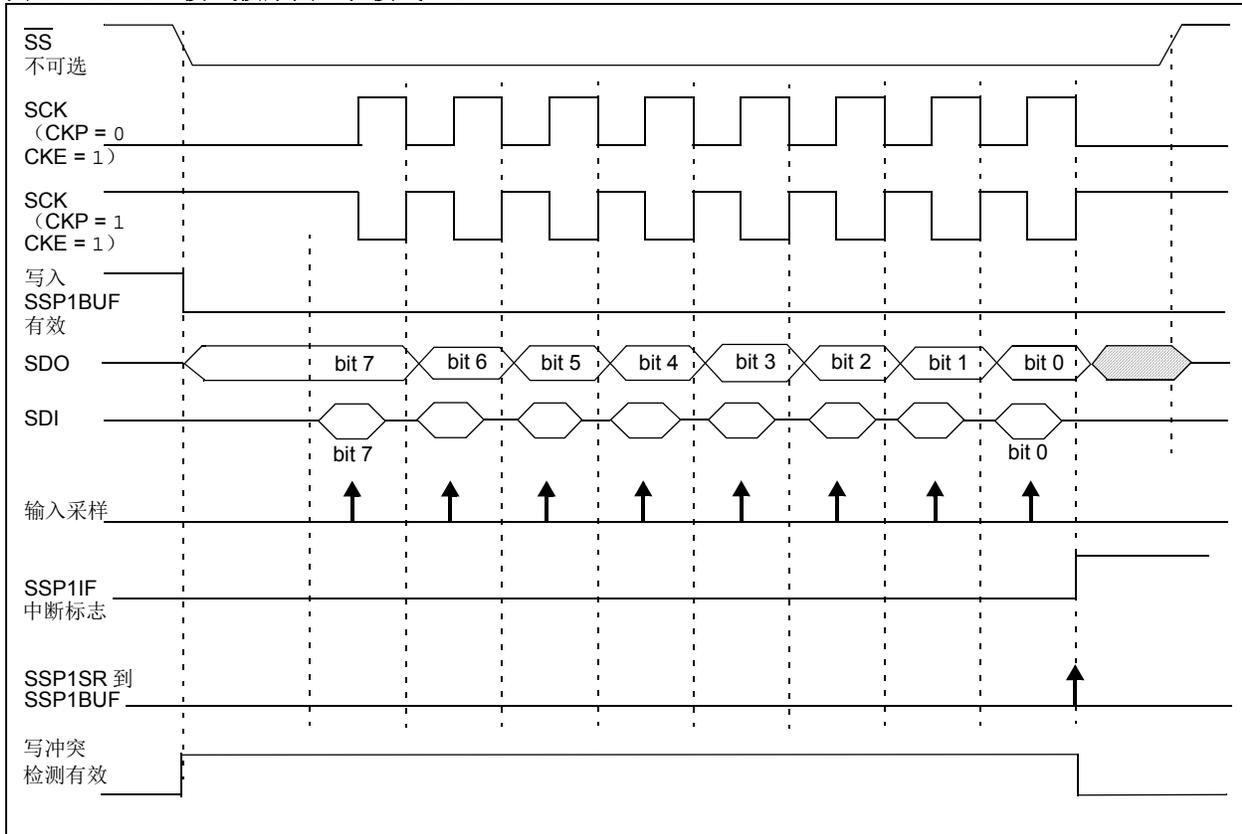


图30-10: SPI模式波形图 (从模式, CKE = 1)



30.2.6 休眠模式下的SPI操作

在SPI主模式下，当选择休眠模式时，所有模块时钟都暂停，并且在器件被唤醒前，发送/接收将保持在此暂停状态。器件返回到运行模式之后，模块将恢复发送和接收数据。

在SPI从模式下，SPI发送/接收移位寄存器与器件异步工作。这可使器件置于休眠模式，仍能将从器件移入SPI发送/接收移位寄存器。当接收到全部8位数据时，MSSP中断标志位将置1，并且如果允许中断的话，将唤醒器件。

30.3 I²C 模式概述

I²C总线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30-2和图30-3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，它们都需要使用连接到电源电压的上拉电阻。线路下拉到地时，视为逻辑0；线路保持悬空时，视为逻辑1。

图30-11给出了分别配置为主器件和从器件的两个处理器之间的典型连接。

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。

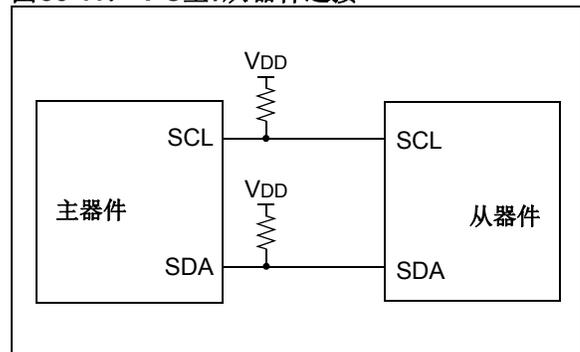
对于给定器件，有4种可能的工作模式：

- 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，主器件发送启动条件，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，主器件会继续向/从从器件发送或接收数据。

图30-11： I²C主/从器件连接



PIC16(L)F18324/18344

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**NACK**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，主器件可能在它处于接收模式时，发送一个重复启动条件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两次读操作或组合读写操作的组合报文。

30.3.1 时钟延长

在从器件尚未完成数据处理时，它可以通过时钟延长这一过程来延迟更多数据的传输。寻址到的从器件可以在接收或发送一位数据之后将**SCL**时钟线保持为低电平，指示它尚未准备好继续通信。与从器件进行通信的主器件将会尝试将**SCL**线拉为高电平，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。

30.3.2 仲裁

每个主器件都必须监视总线上是否出现启动位和停止位。如果器件检测到总线正忙，则在总线恢复为空闲状态之前，它无法开始新的报文。

但是，可能会有两个主器件尝试同时或几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。

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。

30.4 I²C模式操作

所有MSSP I²C通信都是针对字节的，并且会先移出MSb。6个SFR寄存器和2个中断标志用作模块与PIC[®]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。

30.4.1 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。

30.4.2 I²C术语的定义

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Philips I²C规范改写。

30.4.3 SDA和SCL引脚

选择任何I²C模式时，用户应通过将相应的TRIS位置1以将SCL和SDA引脚引脚设置为输入。

注： 通过PPS外设，可以选择任意器件引脚用于SDA和SCL功能。这些功能是双向的。SDA输入通过SSPDATPPS寄存器进行选择。SCL输入通过SSPCLKPPS寄存器进行选择。输出通过RxyPPS寄存器进行选择。用户需要负责确保在进行选择时，使每个功能的输入和输出处于同一引脚上。

30.4.4 SDA保持时间

SDA引脚的保持时间通过SSP1CON3寄存器的SDAHT位进行选择。保持时间是SDA在SCL的下降沿之后保持有效的时间。将SDAHT位置1可以选择最低300 ns的较长保持时间，这对于电容较大的总线会有帮助。

表30-1: I²C总线术语

术语	说明
发送器	将数据移出到总线上的器件。
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。
寻址到的从器件	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	从器件接收到的地址字节与存储在SSPADD中的值相匹配。
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。

PIC16(L)F18324/18344

30.4.5 启动条件

I²C 规范将启动条件定义为在 SCL 线为高电平时，SDA 线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 30-12 给出了启动和停止条件的波形图。

30.4.6 停止条件

停止条件定义为在 SCL 线为高电平时，SDA 线从低电平变为高电平状态。

注： 在停止条件生效之前，必须至少出现一个 SCL 低电平时间，因此，如果 SDA 线变为低电平然后再次变为高电平，而 SCL 线保持高电平，则只会检测到启动条件。

30.4.7 重复启动条件

重复启动条件在每次停止条件有效的时候有效。如果主器件希望在终止当前传输之后保持对总线的控制，主器件可以发出重复启动条件。重复启动对从器件产生的影响与启动条件相同，即复位所有从器件逻辑并使之准备接收一个地址。主器件可以寻址同一个或另一个从器件。图 30-13 给出了重复启动条件的波形图。

在 10 位寻址从模式下，要从寻址到的从器件中移出数据，主器件需要产生重复启动条件。从器件完全寻址（高地址字节和低地址字节均匹配）之后，主器件可以发出重复启动条件和 R/W 位置 1 的高地址字节。然后，从器件逻辑会保持时钟，并准备送出数据。

30.4.8 启动/停止条件中断屏蔽

SSP1CON3 寄存器的 SCIE 和 PCIE 位可以用于允许在通常不支持中断功能的从模式下产生中断。对于已允许启动和停止检测中断的从模式，这两位没有任何作用。

图 30-12: I²C 启动和停止条件

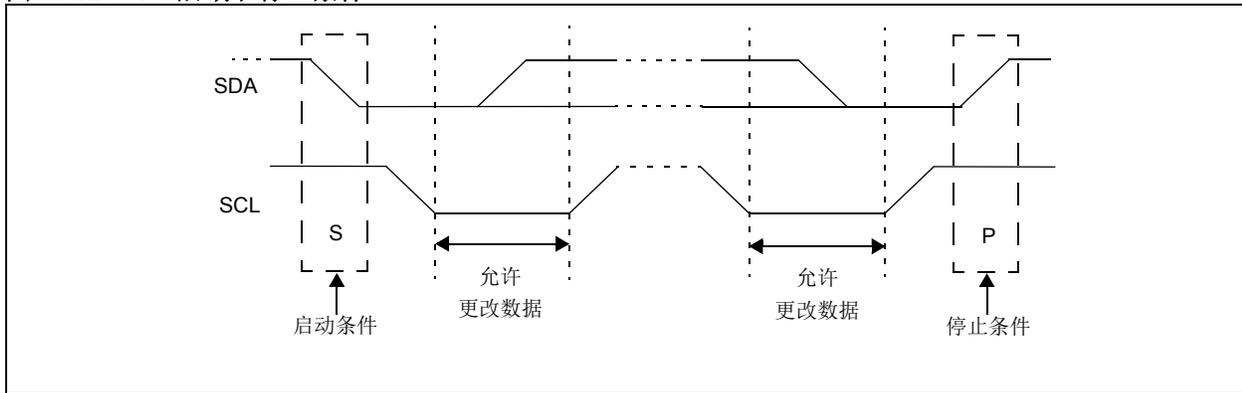
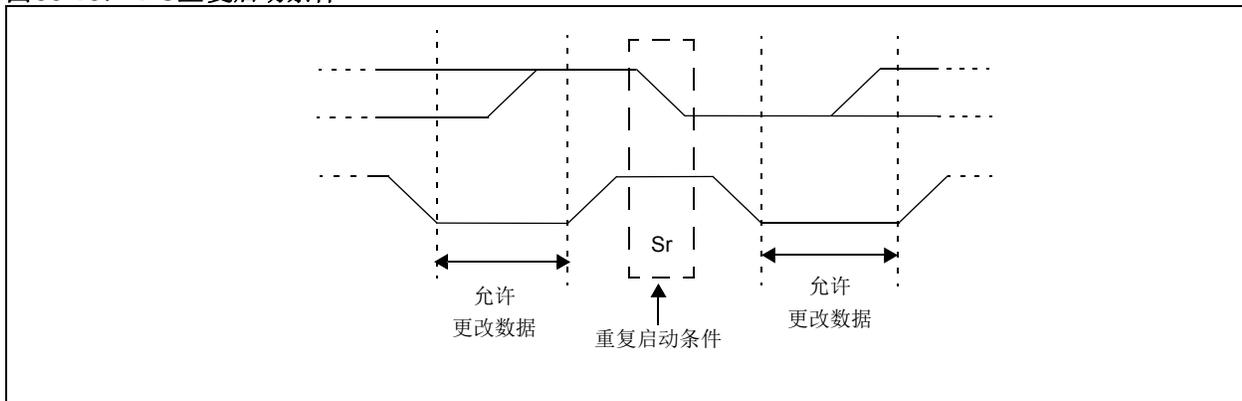


图 30-13: I²C 重复启动条件



30.4.9 应答序列

在I²C中，所有传输字节的第9个SCL脉冲都专门用作应答信号。它使接收器件可以通过将SDA线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。

$\overline{\text{ACK}}$ 的结果会被放入SSP1CON2寄存器的ACKSTAT位中。

当AHEN和DHEN位置1时，时钟被延长，从器件软件允许从器件有时间在回送发送器前改变 $\overline{\text{ACK}}$ 值。用户可以通过置1/清零SSP1CON2寄存器的ACKDT位来决定响应。

在一些条件下，从器件不会发送 $\overline{\text{ACK}}$ ：如果在接收到数据字节时，SSP1STAT寄存器的BF位或SSP1CON1寄存器的SSPOV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。

30.5 I²C从模式操作

MSSP从模式可以在4种模式之一下工作，这些模式通过SSP1CON1寄存器的SSPM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。

30.5.1 从模式地址

SSP1ADD寄存器(寄存器30-6)包含从模式地址。在启动或重复启动条件之后接收到的第一个字节将与该寄存器中的存储值进行比较。如果字节匹配，则值会被装入SSP1BUF寄存器，并产生中断。如果值不匹配，则模块会进入空闲状态，并且不会向软件指示是否发生了什么情况。

SSP掩码寄存器(寄存器30-5)会影响地址匹配过程。更多信息，请参见第30.5.9节“SSP掩码寄存器”。

30.5.1.1 I²C从器件7位寻址模式

在7位寻址模式下，在确定地址是否匹配时，所接收数据字节的LSb会被忽略。

30.5.1.2 I²C从器件10位寻址模式

在10位寻址模式下，接收到的第一个字节将与二进制值“1 1 1 1 0 A9 A8 0”进行比较。A9和A8是10位地址的最高2位，存储在SSP1ADD寄存器的bit 2和bit 1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。

PIC16(L)F18324/18344

30.5.2 从接收

当接收到的匹配地址字节的 $\overline{R/W}$ 位清零时，SSP1STAT寄存器的 $\overline{R/W}$ 位会清零。接收到的地址被装入SSP1BUF寄存器并产生应答信号。

当接收到的地址存在溢出条件时，将会产生无应答信号。溢出条件定义为SSP1STAT寄存器的BF位被置1或SSP1CON1寄存器的SSPOV位被置1。SSP1CON3寄存器的BOEN位会修改此操作。更多信息，请参见寄存器30-4。

每个传输的数据字节都会产生MSSP中断。标志位SSP1IF必须用软件清零。

当SSP1CON2寄存器的SEN位被置1时，SCL将在接收到每个字节后保持低电平（时钟延长）。必须通过将SSP1CON1寄存器的CKP位置1来释放时钟。

30.5.2.1 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。图30-14和图30-15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位会置1。
3. 接收到 $\overline{R/W}$ 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. 软件清零SSP1IF位。
11. 软件从SSP1BUF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。

30.5.2.2 AHEN和DHEN位置1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有时间决定是否应答（ACK）接收的地址或数据字节。

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30-16显示了同时使用地址和数据保持功能的模块。图30-17包含了SSP1CON2寄存器的SEN位置1时的操作。

1. SSP1STAT的S位置1；如果允许在检测到启动条件时产生中断，则SSP1IF位会置1。
2. $\overline{R/W}$ 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由硬件清零。
3. 从器件清零SSP1IF位。
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. 从器件软件通过将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位。

注： 即使不进行时钟延长，且BF已清零，SSP1IF仍然会在SCL的第9个下降沿之后置1。只有向主器件发送了NACK信号后，SSP1IF才不会置1。

11. 在所接收数据字节的第8个SCL下降沿之后，SSP1IF置1，CKP由硬件清零。
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SSP1STAT寄存器的P位才能检测停止条件。

图30-14: I²C从模式接收时序 (SEN = 0, AHEN = 0, DHEN = 0, 7位地址)

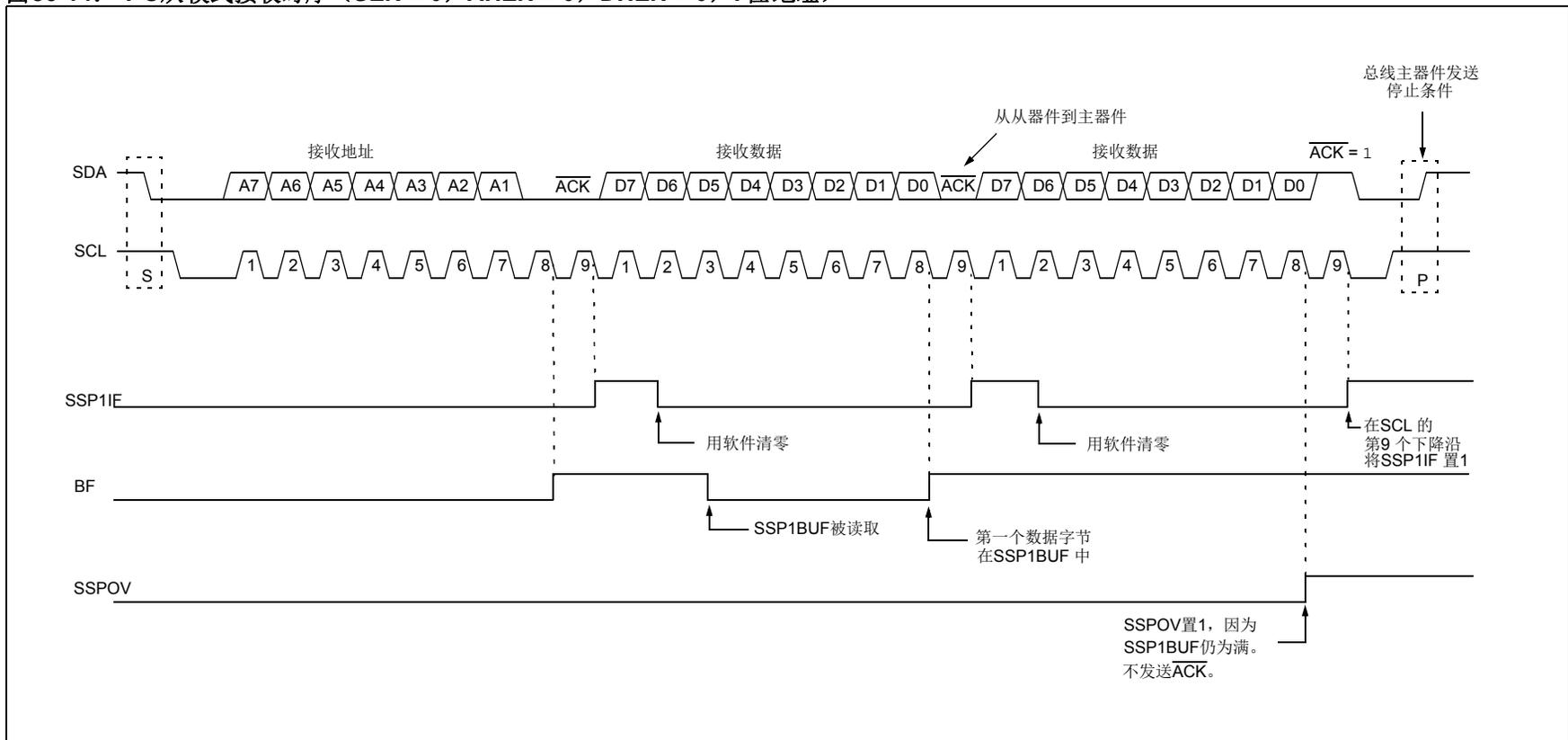


图30-15: I²C从模式接收时序 (SEN = 1, AHEN = 0, DHEN = 0, 7位地址)

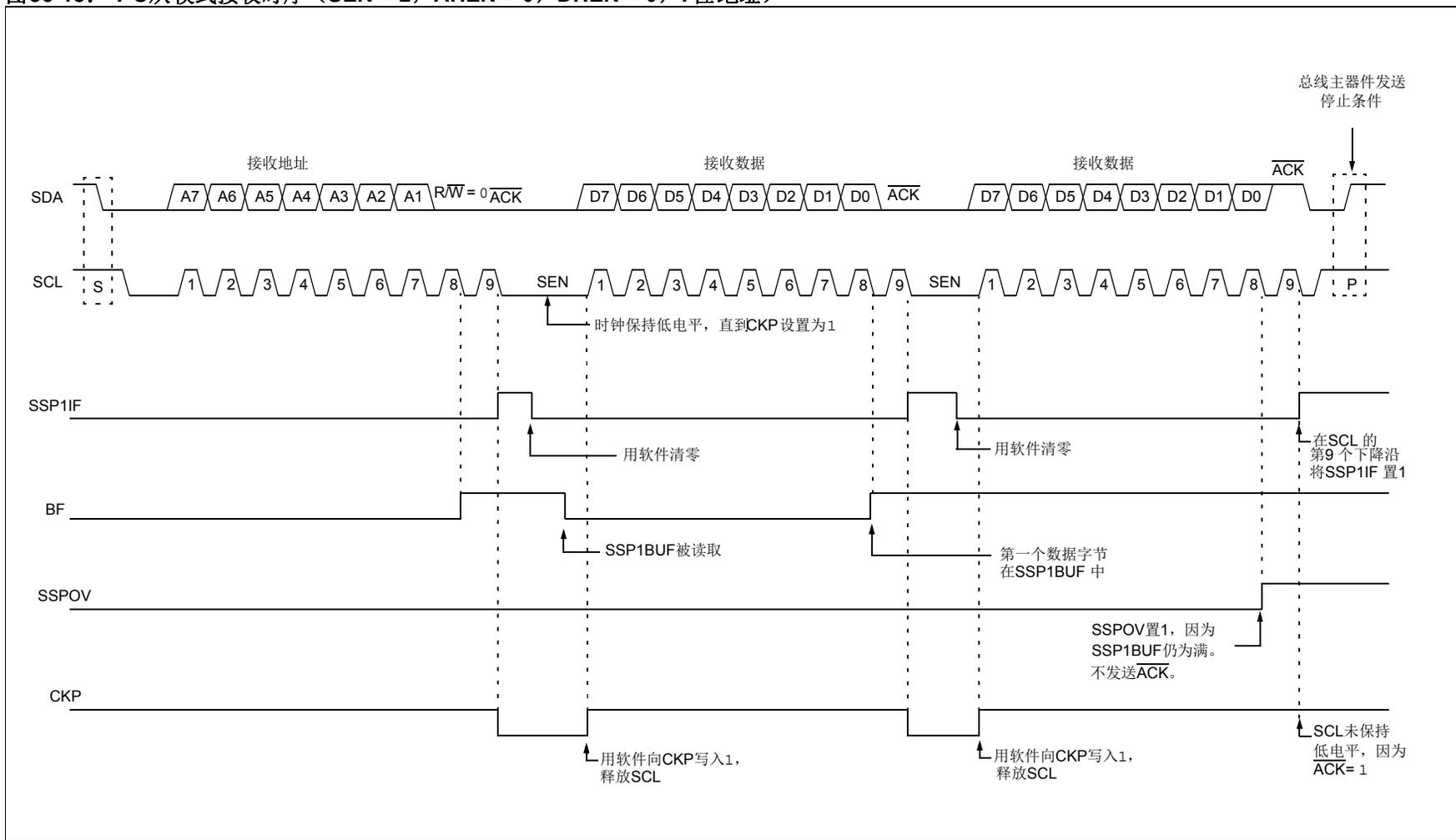


图30-16: I²C从模式接收时序 (SEN = 0, AHEN = 1, DHEN = 1, 7位地址)

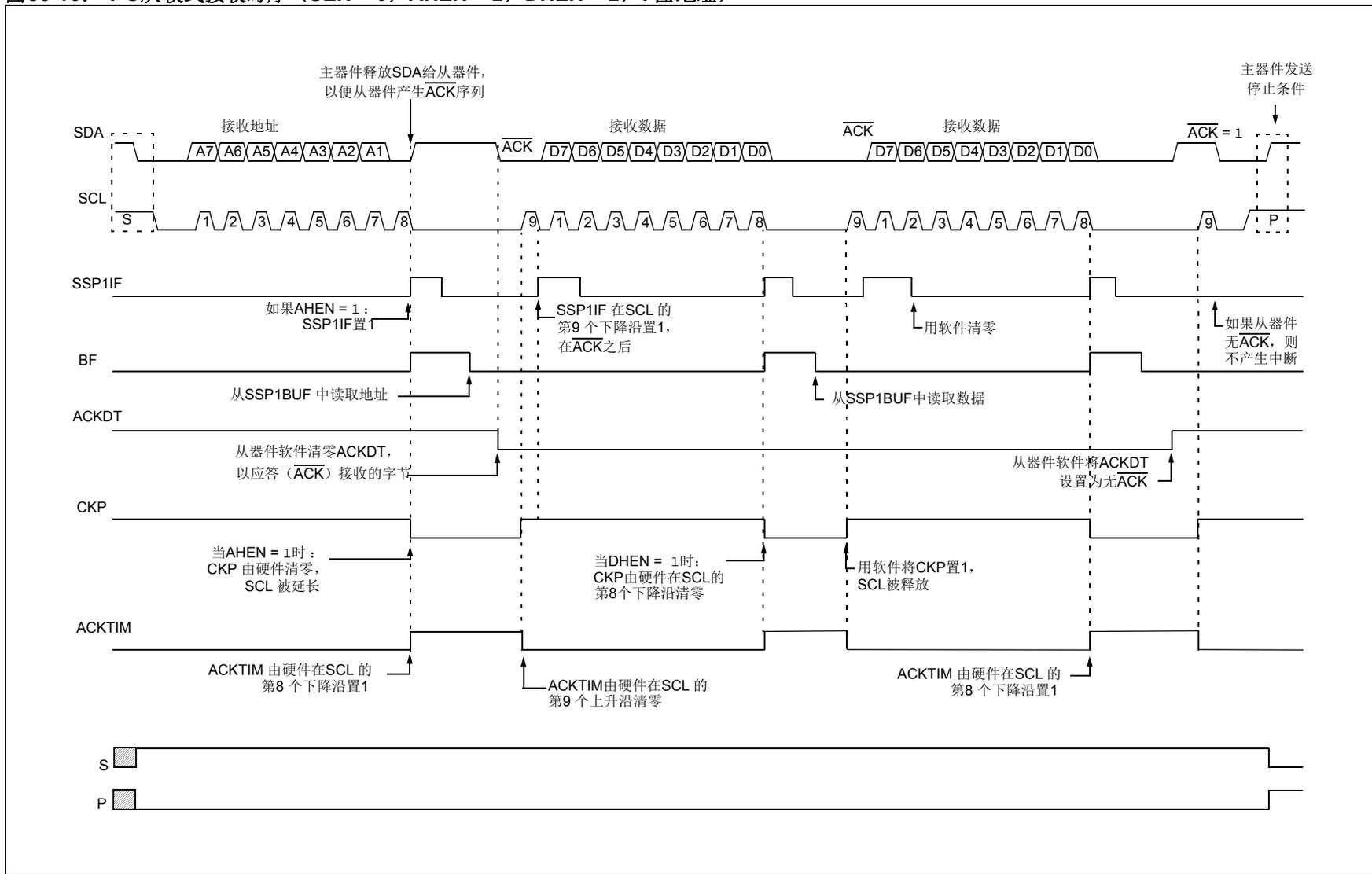
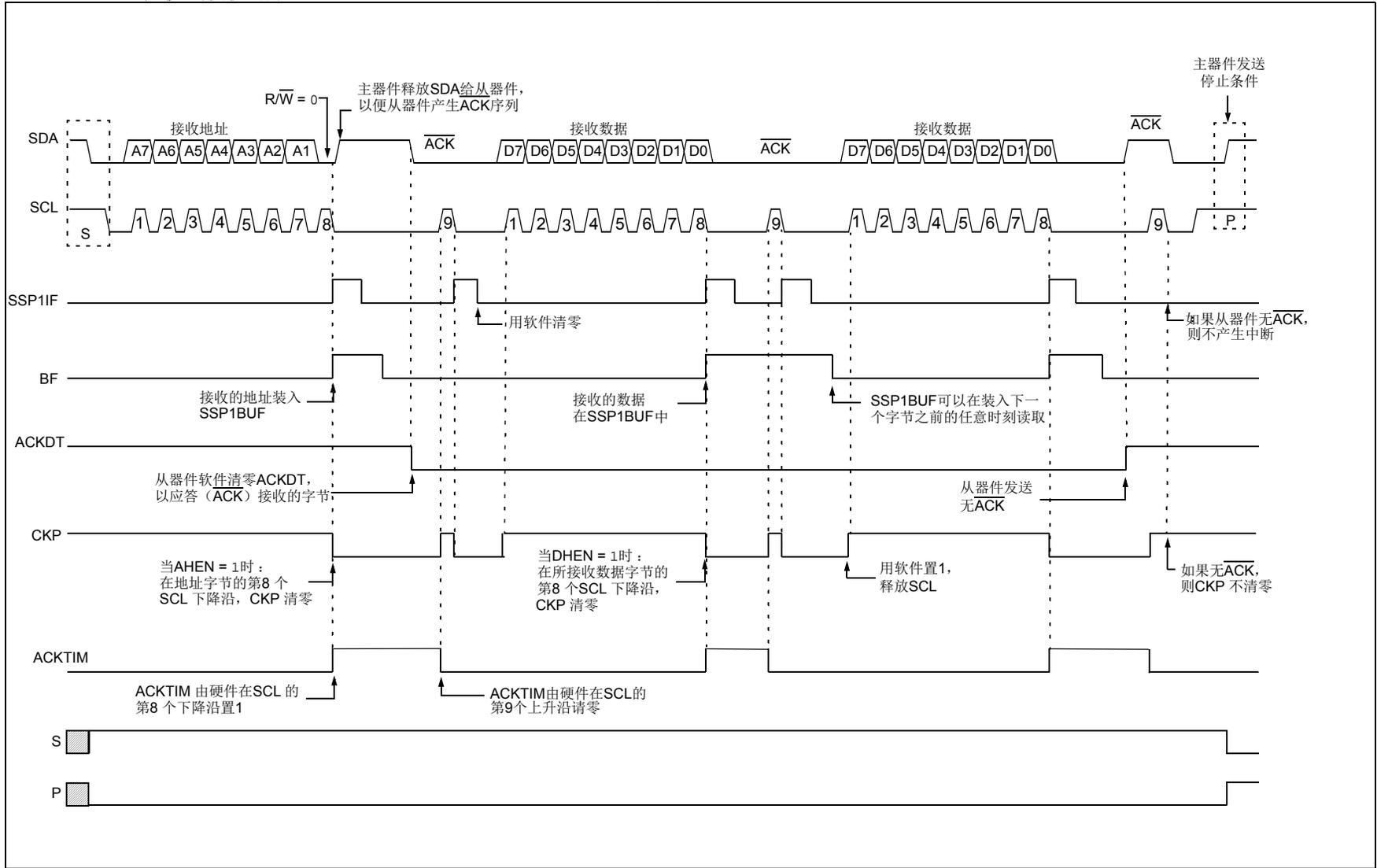


图30-17: I²C从模式接收时序 (SEN = 1, AHEN = 1, DHEN = 1, 7位地址)



30.5.3 从发送

当输入地址字节的 $\overline{R/W}$ 位置1并发生地址匹配时，SSP1STAT寄存器的 $\overline{R/W}$ 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 \overline{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，请参见第30.5.6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。

来自接收器的 \overline{ACK} 脉冲将在第9个SCL输入脉冲的上升沿锁存。该 \overline{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。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（ \overline{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MSSP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个时钟脉冲的下沿被置1。

30.5.3.1 从模式总线冲突

从器件接收到读请求，开始在SDA线上移出数据。如果检测到总线冲突，SSP1CON3寄存器的SBCDE位会置1，PIR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。

30.5.3.2 7位发送

主器件可以向从器件发送读请求，然后从从器件中移出数据。以下列表列出了在实现标准数据发送时，从器件软件需要执行的操作。图30-18可用作该列表的参考。

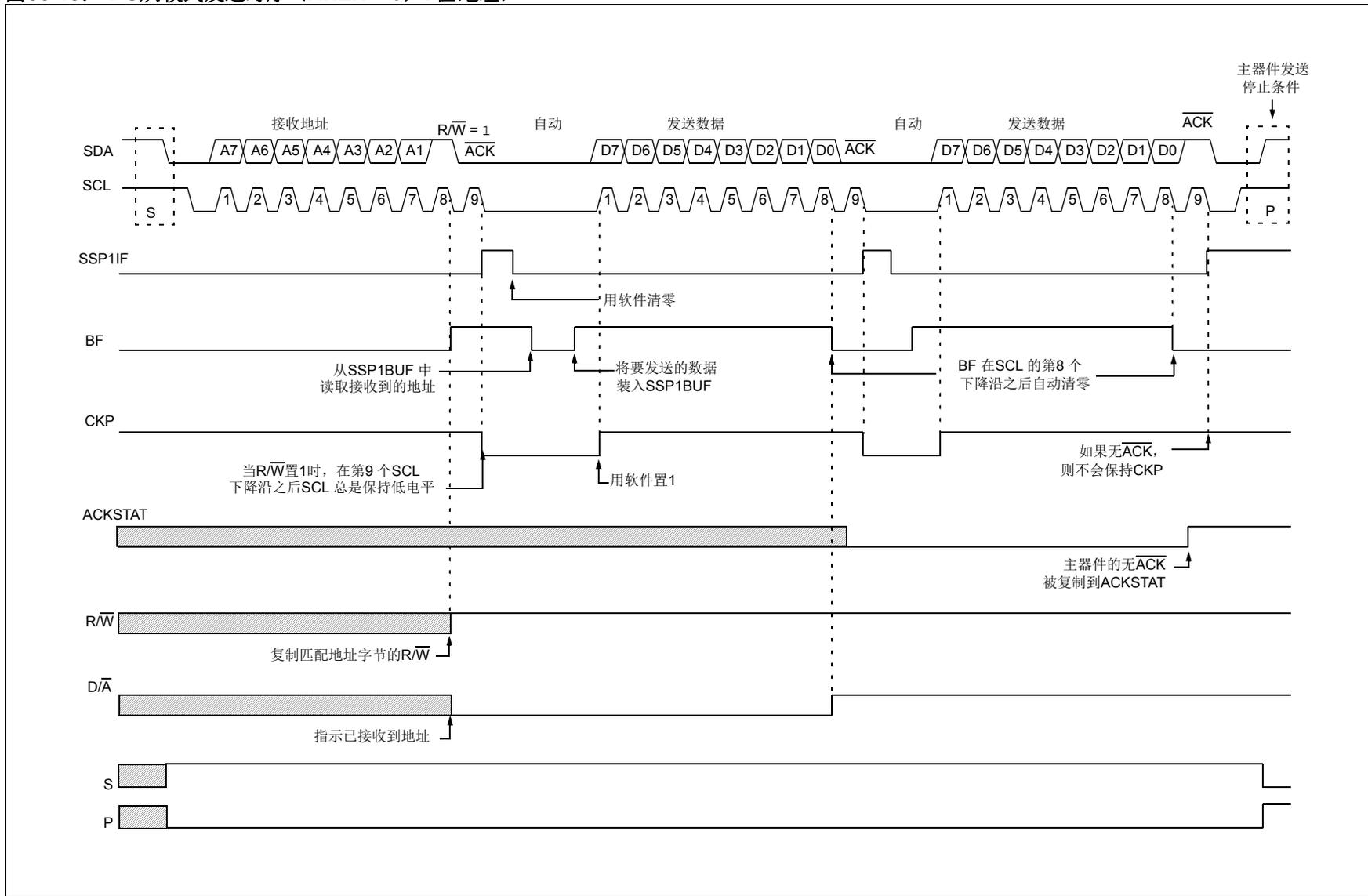
1. 主器件在SDA和SCL上发送一个启动条件。
2. SSP1STAT的S位置1；如果允许在检测到启动条件时产生中断，则SSP1IF位会置1。
3. 从器件接收到 $\overline{R/W}$ 位置1的匹配地址，将SSP1IF位置1。
4. 从器件硬件产生 \overline{ACK} ，并将SSP1IF置1。
5. 软件将SSP1IF位清零。
6. 软件从SSP1BUF中读取接收的地址，使BF清零。
7. $\overline{R/W}$ 置1，所以CKP会在 \overline{ACK} 之后由硬件自动清零。
8. 从器件软件将发送数据装入SSP1BUF。
9. CKP位由软件置1，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 \overline{ACK} 响应装入ACKSTAT位之后，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（ \overline{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 \overline{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30-18: I²C从模式发送时序 (AHEN = 0, 7位地址)



30.5.3.3 使能了地址保持的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30-19给出了在使能AHEN时7位地址从发送的标准波形图。

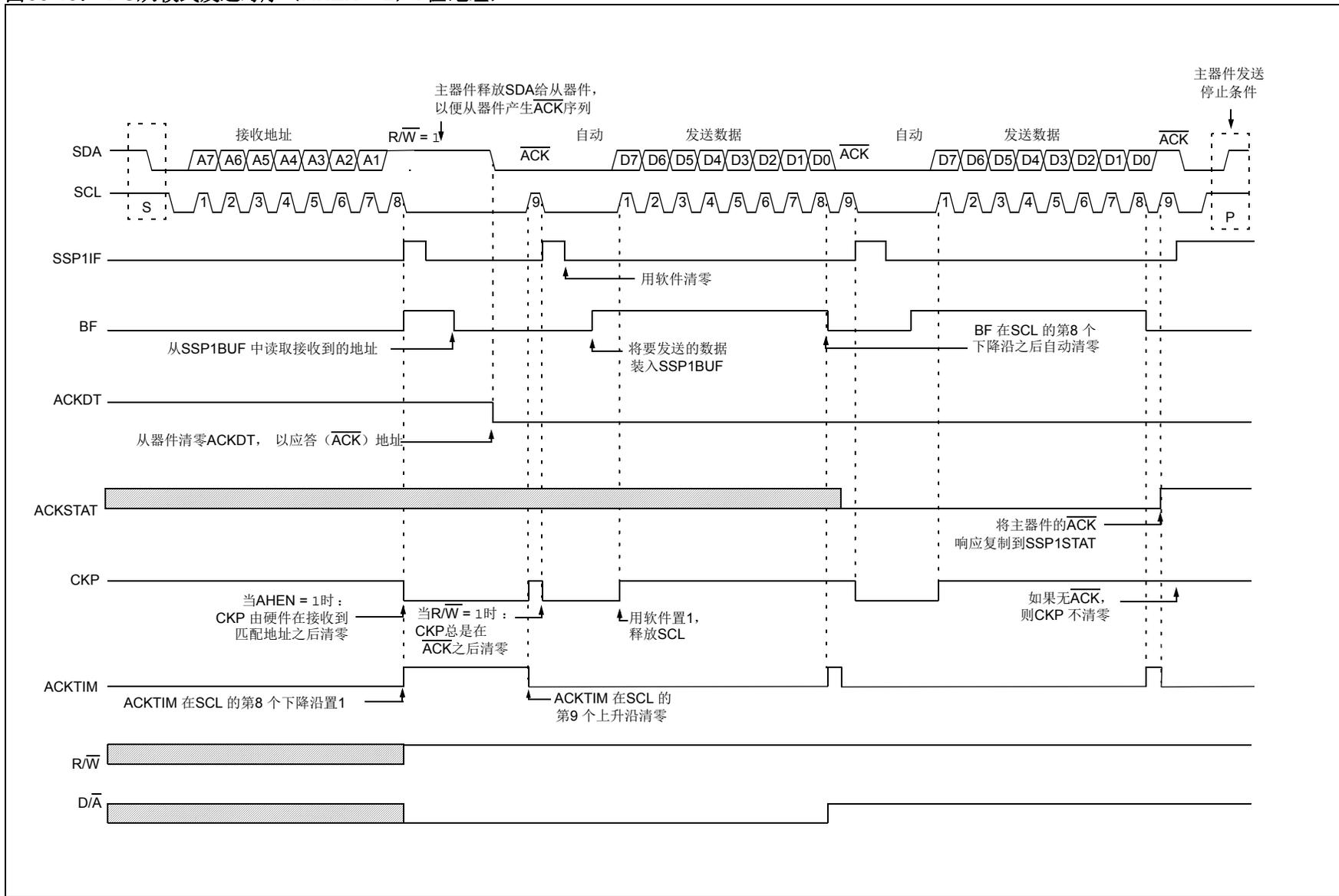
1. 总线启动时为空闲模式。
2. 主器件发送启动条件；SSP1STAT的S位置1；如果允许在检测到启动条件时产生中断，则SSP1IF会置1。
3. 主器件发送 $\overline{R/W}$ 位置1的匹配地址。在SCL线的第8个下降沿之后，CKP位由硬件清零，并产生SSP1IF中断。
4. 从软件清零SSP1IF。
5. 从器件软件读取SSP1CON3寄存器的 \overline{ACKTIM} 位，以及SSP1STAT寄存器的 $\overline{R/W}$ 和 $\overline{D/A}$ 位，以确定中断源。
6. 从器件从SSP1BUF寄存器中读取地址值，使BF位清零。
7. 从器件软件根据该信息确定它是产生 \overline{ACK} 还是产生无 \overline{ACK} ，并相应地设置SSP1CON2寄存器的ACKDT位。
8. 从器件软件将CKP位置1，释放SCL。
9. 主器件移入来自从器件的 \overline{ACK} 值。
10. 如果 $\overline{R/W}$ 位置1，则在 \overline{ACK} 之后，从器件硬件会自动将CKP位清零，将SSP1IF置1。
11. 从软件清零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 \overline{ACK} 之后，才能装入SSP1BUF。

13. 从器件软件将CKP位置1，释放时钟。
14. 主器件随着时钟从从器件中移出数据，并在第9个SCL脉冲发送 \overline{ACK} 值。
15. 从器件硬件将 \overline{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 \overline{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 \overline{ACK} ，以确保从器件释放SCL线来接收停止条件。

图30-19: I²C从模式发送时序 (AHEN = 1, 7位地址)



30.5.4 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30-20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中断，则SSP1IF会置1。
3. 主器件发送R \overline{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 \overline{ACK} ，SSP1IF置1。
5. 用软件清零SSP1IF位。
6. 软件从SSP1BUF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 \overline{ACK} 序列之后，才允许更新SSP1ADD寄存器。

9. 从器件发送 \overline{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位。
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 \overline{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位。
16. 从器件从SSP1BUF中读取接收的字节，使BF清零。
17. 如果SEN置1，从器件软件会将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。

30.5.5 带地址或数据保持的10位寻址

在AHEN或DHEN置1时，使用10位寻址的接收方式与7位模式相同。惟一的区别是需要使用UA位来更新SSP1ADD寄存器。所有功能（特别是在CKP位清零，SCL线保持低电平时）都是相同的。图30-21可以用作AHEN置1时10位寻址模式下从器件接收的参考图示。

图30-22给出了10位寻址模式下从发送器的标准波形图。

图 30-20: I²C 从模式接收时序 (SEN = 1, AHEN = 0, DHEN = 0, 10 位地址)

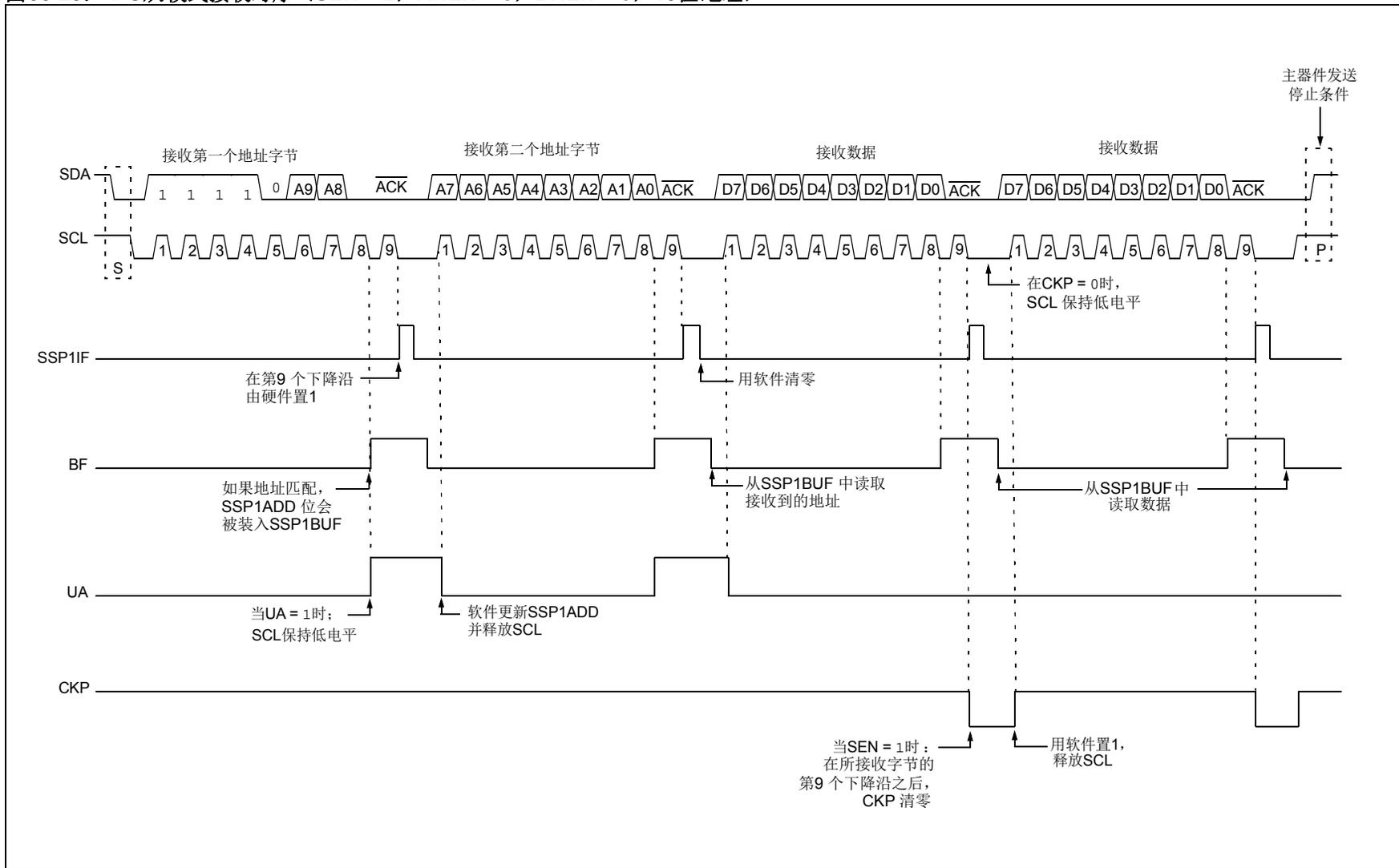


图 30-21: I²C 从模式接收时序 (SEN = 0, AHEN = 1, DHEN = 0, 10 位地址)

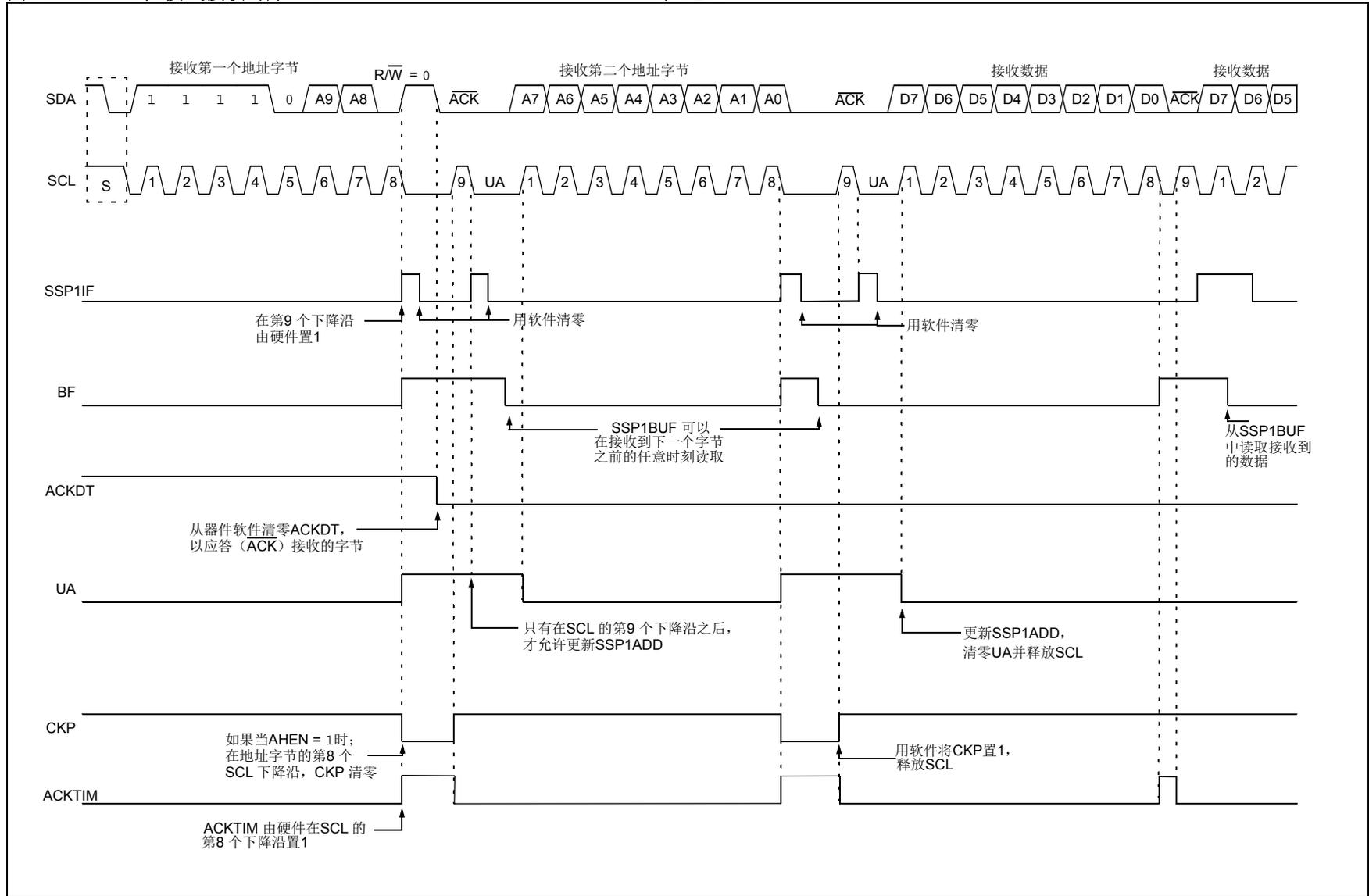
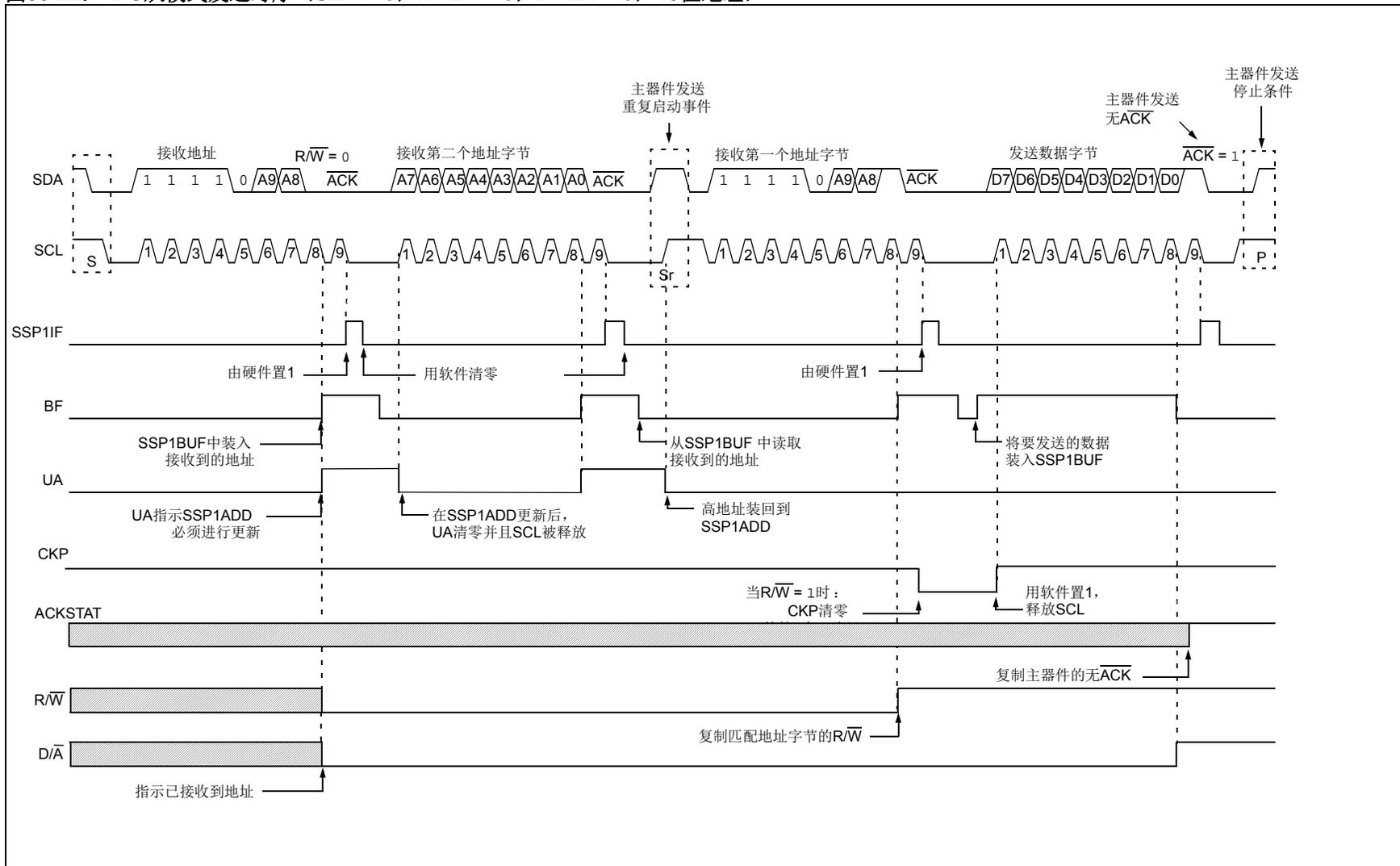


图30-22: I²C从模式发送时序 (SEN = 0, AHEN = 0, DHEN = 0, 10位地址)



30.5.6 时钟延长

当总线上的某个器件将SCL线保持为低电平而有效暂停通信时，就发生了时钟延长现象。从器件可以延长时钟，以便可以有更多时间来处理数据或准备响应主器件。时钟延长时并不关心主器件的工作，因为任何时候只需总线上主器件处于活动状态但是不传输数据就可以被认为是时钟延长。由从器件进行的任何时钟延长对于主器件软件都是不可见的，都由产生SCL的硬件进行处理。

SSP1CON1寄存器的CKP位用于控制时钟延长。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。

30.5.6.1 正常时钟延长

如果SSP1STAT的 $\overline{R\overline{W}}$ 位置1（读请求），则在 \overline{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 \overline{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。

30.5.6.2 10位寻址模式

在10位寻址模式下，当UA位置1时，时钟总是会被延长。这是无需清零CKP就会延长SCL的惟一情形。在写入SSP1ADD之后，SCL会立即被释放。

注： 如果第二个地址字节不匹配，先前版本的模块不会延长时钟。

30.5.6.3 字节无应答

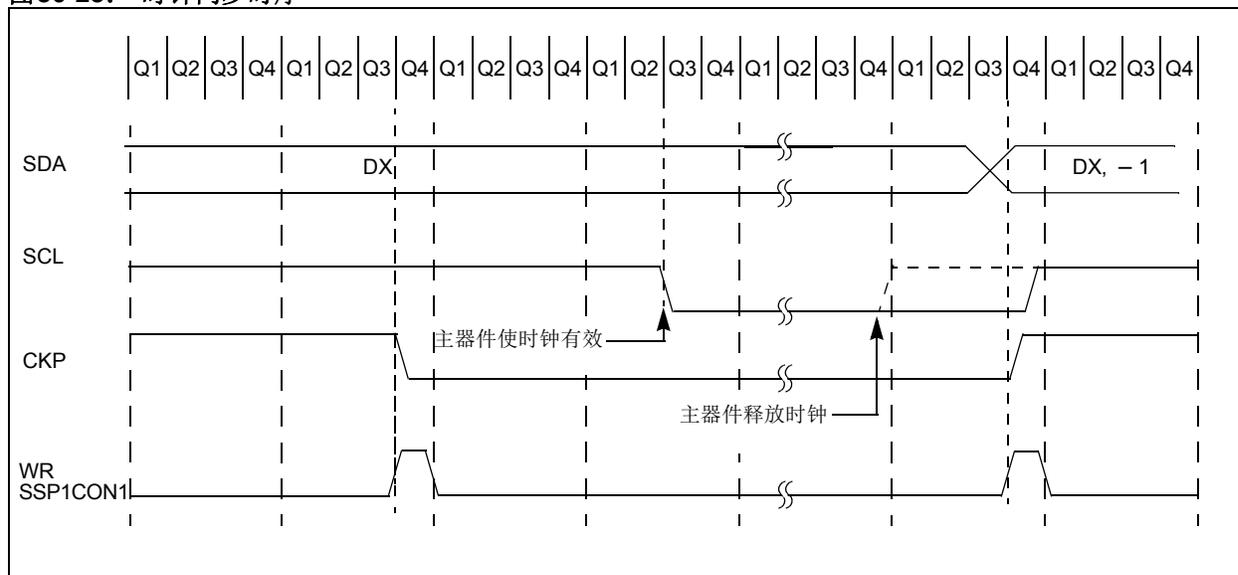
当SSP1CON3的AHEN位置1时，在所接收匹配地址字节的第8个SCL下降沿之后，硬件会将CKP清零。当SSP1CON3的DHEN位置1时，在所接收数据的第8个SCL下降沿之后，CKP会被清零。

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。

30.5.7 时钟同步和CKP位

每当CKP位清零时，模块就会等待SCL线变为低电平，然后保持低电平状态不变。但是，只有当已经采样到SCL输出为低电平时，清零CKP位才会将SCL输出置为低电平。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30-23）。

图30-23: 时钟同步时序



PIC16(L)F18324/18344

30.5.8 广播呼叫地址支持

在I²C总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有器件都应该发送一个应答信号来响应。

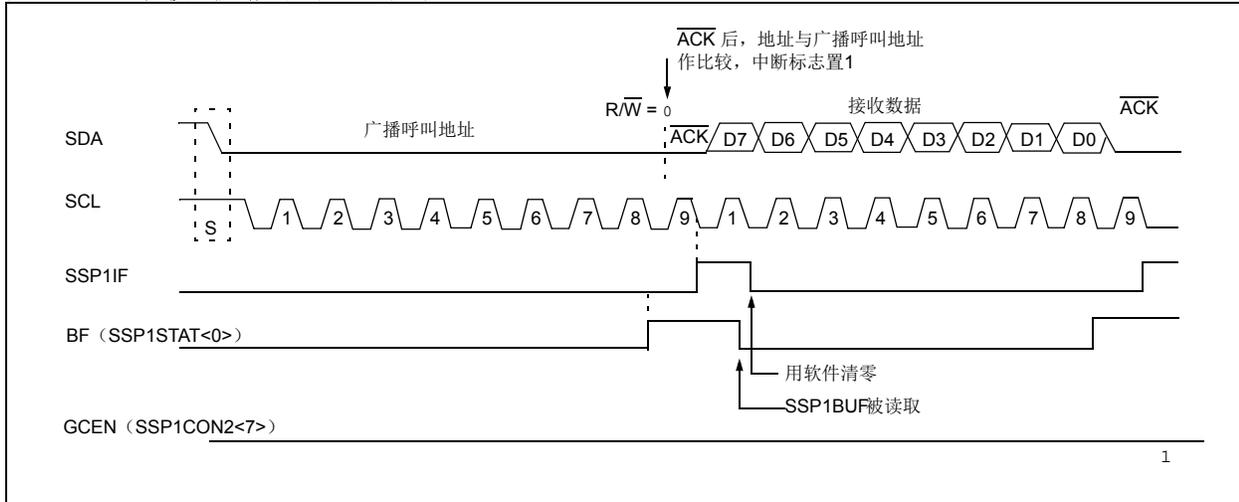
广播呼叫地址是I²C协议中的保留地址，定义为地址0x00。如果SSP1CON2寄存器的GCEN位置1，则无论SSP1ADD中存储的值如何，在接收到该地址时，从模块都会自动发送ACK。在从器件移入R/W位清零的全零地址之后，将会产生中断，从器件软件可以读取

SSP1BUF并进行响应。图30-24显示了广播呼叫接收序列。

在10位地址模式下，UA位不会在接收到广播呼叫地址时置1。从器件会准备接收作为数据的第二个字节，这与在7位模式下相同。

如果SSP1CON3寄存器的AHEN位置1，则与接收到任意其他地址时相同，从器件硬件会在SCL的第8个下降沿之后延长时钟。然后，从器件必须与正常情况下一样，设置它的ACKDT值，并释放时钟来继续进行通信。

图30-24: 从模式广播呼叫地址序列



30.5.9 SSP掩码寄存器

SSP掩码 (SSP1MSK) 寄存器 (寄存器30-5) 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SSP操作没有影响。

SSP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，SSP掩码没有影响。

30.6 I²C主模式

通过将SSP1CON1寄存器中的相应SSPM<3:0>位置1和清零，同时将SSPEN位置1，可以使能主模式。在主模式下，SDA和SCK引脚必须被配置为输入。当需要将引脚驱动为低电平时，MSSP外设硬件将改写输出驱动器的TRIS控制。

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时，可取得I²C总线的控制权，否则总线空闲。

在固件控制的主模式下，用户代码根据启动位和停止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唯一有效的电路。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SSP中断标志位SSP1IF置1（如果允许SSP中断，则产生中断）：

- 发生启动条件
- 发生停止条件
- 数据传输字节发送/接收
- 应答发送/接收
- 产生重复启动条件

注 1: 当配置为I²C主模式时，MSSP1模块不允许事件排队。例如，不允许用户在发出启动条件后，在启动条件结束前立即写SSP1BUF寄存器以启动传输。在这种情况下，将不会写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/停止检测并产生中断。

30.6.1 I²C主模式操作

主器件产生所有的串行时钟脉冲、启动条件和停止条件。以停止条件或重复启动条件结束传输过程。因为重复启动条件也是下一次串行传输的开始，因此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作为发送器件的从地址（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位从地址，后面跟随一个1来指示接收位。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。启动条件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30.7节“波特率发生器”。

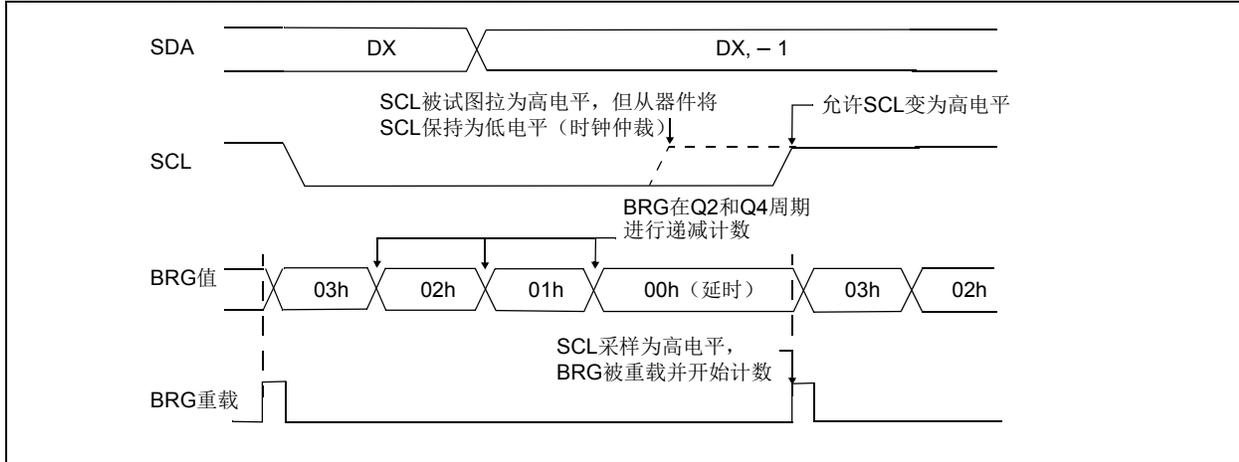
PIC16(L)F18324/18344

30.6.2 时钟仲裁

如果在任何接收、发送或重复启动/停止条件期间，主器件释放了SCL引脚（允许SCL悬空为高电平），就会发生时钟仲裁。当允许SCL引脚悬空为高电平时，波特率发生器（Baud Rate Generator, BRG）暂停计数，直

到SCL引脚被实际采样到高电平为止。当SCL引脚被采样到高电平时，波特率发生器重新装入SSP1ADD<7:0>的内容并开始计数。这可以确保在外部器件将时钟保持低电平时，SCL在至少一个BRG计满返回计数周期内总是保持高电平（图30-25）。

图30-25: 带有时钟仲裁的波特率发生器时序



30.6.3 WCOL状态标志

如果在启动、重复启动、停止、接收或发送序列过程中用户写SSP1BUF，则WCOL被置1，同时缓冲区内容不变（未发生写操作）。每当WCOL位置1时，它指示在模块不处于空闲状态时对SSP1BUF尝试了某个操作。

注： 由于不允许事件排队，在启动条件结束之前，不能写SSP1CON2的低5位。

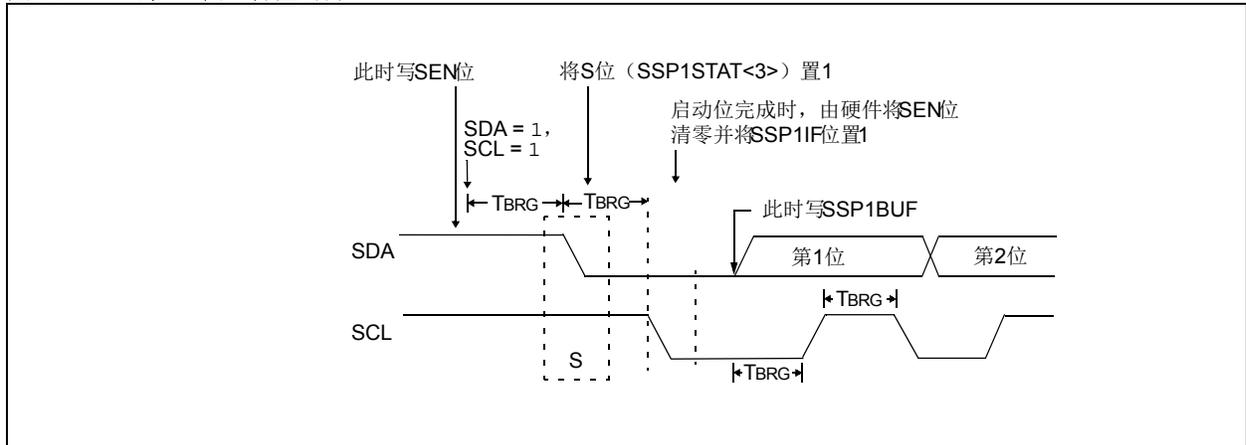
30.6.4 I²C主模式启动条件时序

要发起启动条件（图30-26），用户应将SSP1CON2寄存器的启动使能位SEN置1。当SDA和SCL引脚采样为高电平时，波特率发生器重新装入SSP1ADD<7:0>的内容并开始计数。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SSP1STAT寄存器的S位置1。随后波特率发生器重新装入SSP1ADD<7:0>的内容并恢复计数。当波特率发生器再次超时（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30-26: 第一个启动位时序



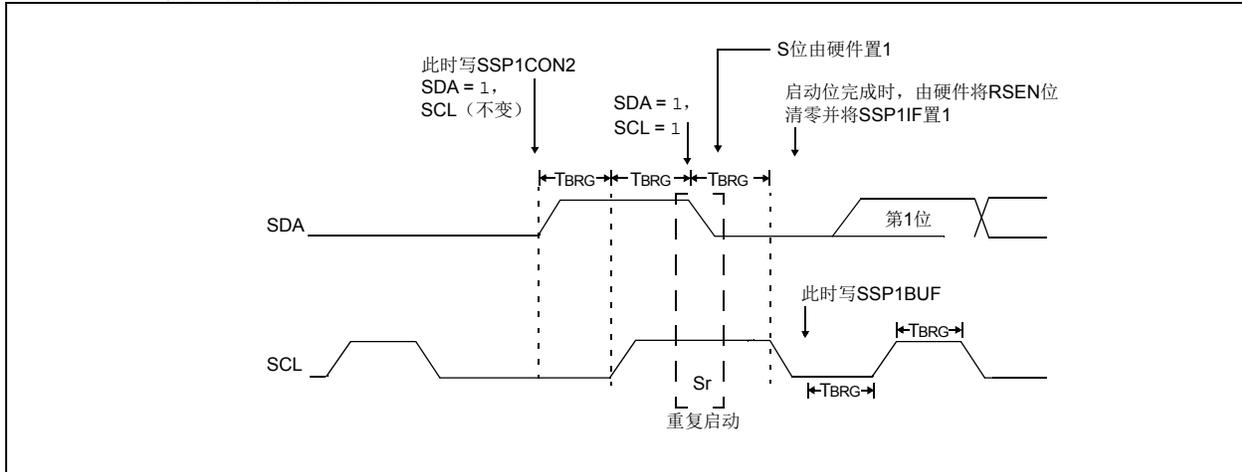
PIC16(L)F18324/18344

30.6.5 I²C主模式重复启动条件时序

当SSP1CON2寄存器的RSEN位设定为高电平，并且主器件状态机无效时，会产生重复启动条件（图30-27）。当RSEN位置1时，SCL引脚被拉为低电平。当SCL引脚被采样为低电平时，波特率发生器会装入值并开始计数。在一个波特率发生器计数周期（TBRG）内SDA引脚被释放（拉为高电平）。当波特率发生器超时，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位将被置1。SSP1IF位在波特率发生器超时之前不会被置1。

- 注 1:** 有任何其他事件在进行时，编程RSEN无效。
- 2:** 在重复启动条件期间，以下事件将会导致发生总线冲突：
- 当SCL由低电平变为高电平时，SDA被采样为低电平。
 - 在SDA被置为低电平之前，SCL变为低电平。这指示另一个主器件正试图发送一个数据1。

图30-27: 重复启动条件波形图



30.6.6 I²C主模式发送

发送一个数据字节、一个7位地址或一个10位地址的另一半都是通过简单地向SSP1BUF寄存器写入一个值来实现的。该操作将使缓冲区满标志位BF置1，并使波特率发生器开始计数和开始下一次发送。地址/数据的每一位将在SCL的下降沿置为有效之后移出到SDA引脚。在一个波特率发生器计满返回计数周期(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(图30-28)。

在写SSP1BUF之后，地址的每一位在SCL的下降沿被移出，直到所有7个地址位和R/W位都被移出。在第8个时钟的下降沿，主器件将释放SDA引脚，以允许从器件发出一个应答响应。在第9个时钟的下降沿，主器件通过采样SDA引脚来判断地址是否被从器件识别。ACK位的状态被装入SSP1CON2寄存器的ACKSTAT状态位。在发送地址的第9个时钟下降沿之后，SSP1IF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。

30.6.6.1 BF状态标志

在发送模式下，SSP1STAT寄存器的BF位在CPU写SSP1BUF时置1，在所有8位数据移出后清零。

30.6.6.2 WCOL状态标志

如果在发送过程中(即，SSP1SR仍在移出数据字节时)用户写SSP1BUF，则WCOL位被置1，同时缓冲区内内容不变(未发生写操作)。

在下次发送前WCOL必须用软件清零。

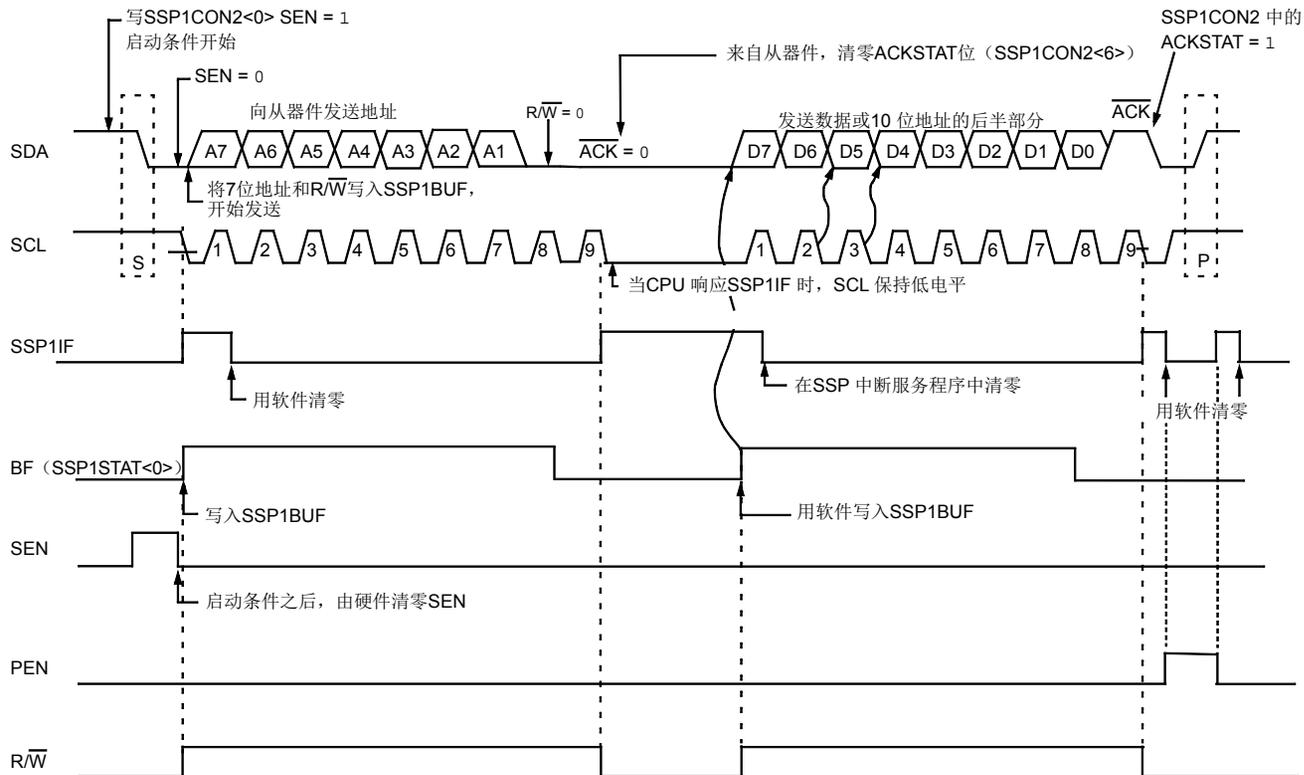
30.6.6.3 ACKSTAT状态标志

在发送模式下，当从器件已发送应答(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。

30.6.6.4 典型发送序列

1. 用户通过将SSP1CON2寄存器的SEN位置1来产生启动条件。
2. 启动条件完成时，SSP1IF由硬件置1。
3. SSP1IF由软件清零。
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. 从器件地址从SDA引脚移出，直到发送完所有8位地址。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. 在第9个时钟周期结束时，MSSP1模块通过将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来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30-28: I²C主模式波形图 (发送, 7位或10位地址)



30.6.7 I²C主模式接收

通过编程SSP1CON2寄存器的接收使能位RCEN，可以使能主模式接收（图30-29）。

注： 将RCEN位置1前，MSSP1模块必须处于空闲状态，否则对RCEN位置1将无效。

波特率发生器开始计数，每次计满返回时，SCL引脚的状态发生改变（由高变低或由低变高），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。

30.6.7.1 BF状态标志

接收过程中，把地址或数据字节从SSP1SR装入SSP1BUF时，BF位置1。在读SSP1BUF寄存器时将其清零。

30.6.7.2 SSPOV状态标志

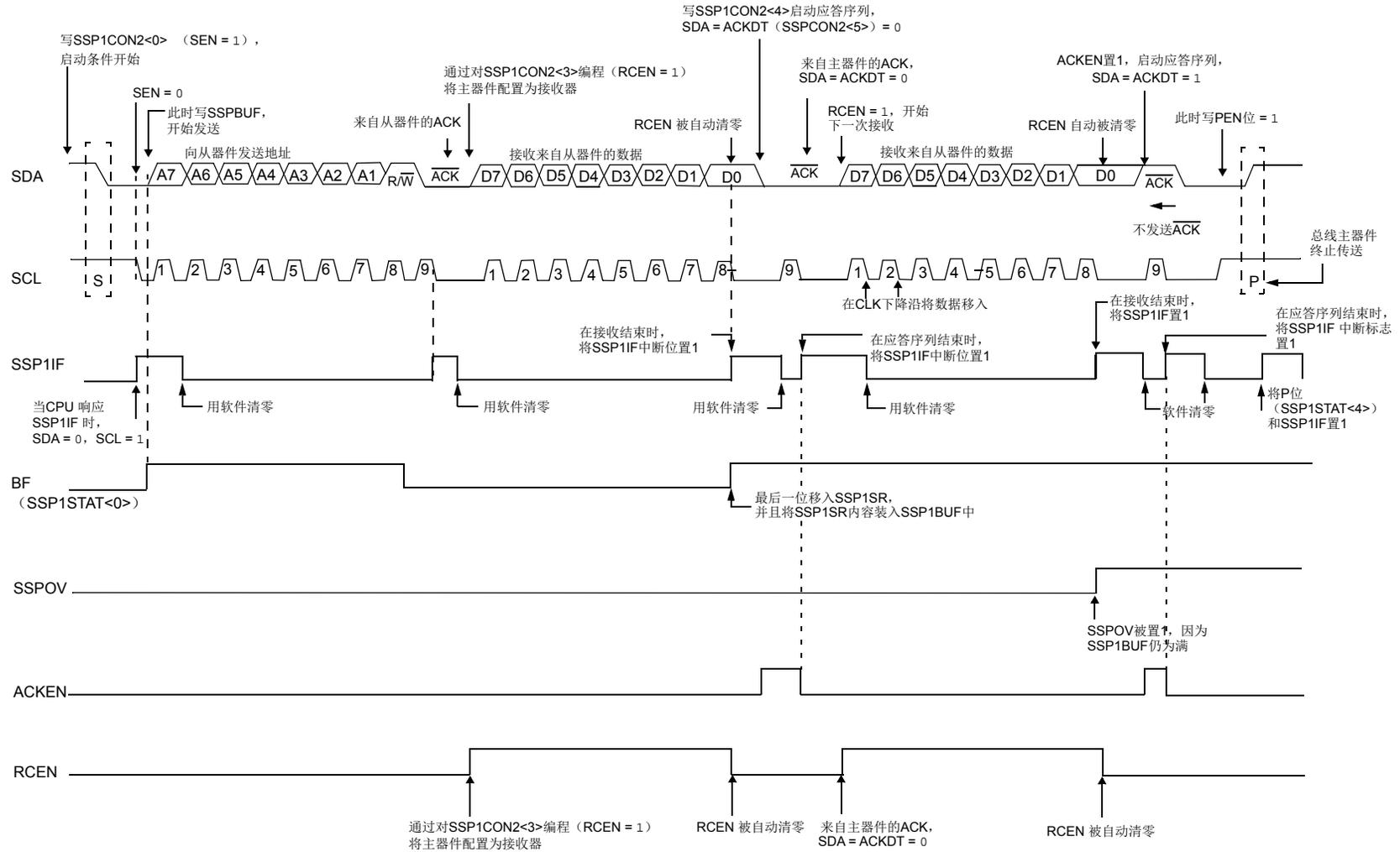
在接收操作中，当SSP1SR接收到8位且BF标志位已经在上一次接收中置1时，SSPOV位置1。

30.6.7.3 WCOL状态标志

如果在接收过程中（即，SSP1SR仍在移入数据字节时）用户写SSP1BUF，则WCOL位被置1，同时缓冲区内内容不变（未发生写操作）。

30.6.7.4 典型的接收序列：

1. 用户通过将SSP1CON2寄存器的SEN位置1来产生启动条件。
2. 启动条件完成时，SSP1IF由硬件置1。
3. SSP1IF由软件清零。
4. 用户将要发送的从器件地址写入SSP1BUF且R/W位置1。
5. 从器件地址从SDA引脚移出，直到发送完所有8位地址。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. 在第9个时钟周期结束时，MSSP1模块通过将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SSP1BUF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. 软件清零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30-29: I²C主模式波形图 (接收, 7位地址)

30.6.8 应答序列时序

将SSP1CON2寄存器的应答序列使能位ACKEN置1即可使能应答序列。当该位被置1时，SCL引脚被拉为低电平，应答数据位的内容输出到SDA引脚上。如果用户希望产生应答，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，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30-30）。

30.6.8.1 WCOL 状态标志

如果在应答序列进行过程中用户写SSP1BUF，则WCOL位被置1，同时缓冲区内容不变（未发生写操作）。

30.6.9 停止条件时序

如果将SSP1CON2寄存器的停止序列使能位PEN置1，则在接收/发送结束时，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。当波特率发生器发生超时，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引脚将被拉高。当SDA引脚采样为高电平且SCL也是高电平时，SSP1STAT寄存器的P位置1。另一个TBRG之后，PEN位被清零，同时SSP1IF位被置1（图30-31）。

30.6.9.1 WCOL 状态标志

如果在停止序列进行过程中用户写SSP1BUF，则WCOL位被置1，同时缓冲区内容不变（未发生写操作）。

图30-30: 应答序列波形图

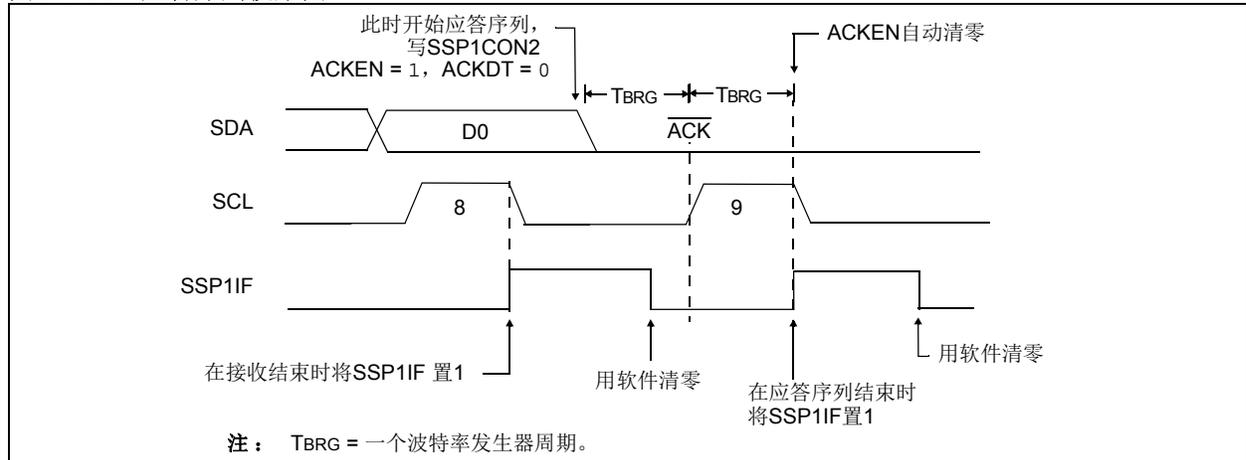
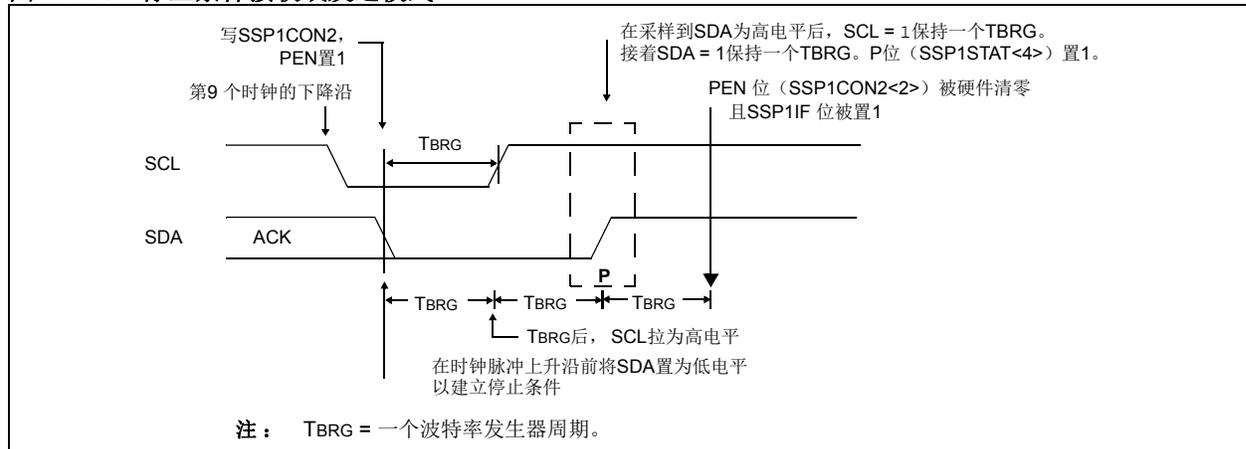


图30-31: 停止条件接收或发送模式



PIC16(L)F18324/18344

30.6.10 休眠模式下的操作

在休眠模式下，I²C 从模块能够接收地址或数据，并且在地址匹配或字节传输完成时，如果允许 MSSP 中断，会将处理器从休眠模式唤醒。

30.6.11 复位的影响

复位会禁止 MSSP1 模块并终止当前的数据传输。

30.6.12 多主器件模式

在多主器件模式下，在检测到启动和停止条件时将产生中断，这可用于判断总线是否空闲。停止 (P) 位和启动 (S) 位在复位或禁止 MSSP1 模块时清零。当 SSP1STAT 寄存器的 P 位置 1 时，可以取得 I²C 总线的控制权；否则，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

30.6.13 多主器件通信、总线冲突和总线仲裁

多主器件模式是通过总线仲裁来支持的。当主器件将地址/数据位输出到 SDA 引脚时，如果一个主器件在 SDA 引脚上输出 1（将 SDA 引脚悬空为高电平），而另一个主器件输出 0，就会发生总线仲裁。当 SCL 引脚悬空为高电平时，数据应是稳定的。如果 SDA 引脚上期望的数据是 1，而实际采样到的数据是 0，则发生了总线冲突。主器件会将总线冲突中断标志位 BCL1IF 置 1，并将 I²C 端口复位为空闲状态（图 30-32）。

如果在发送过程中发生总线冲突，则发送操作停止，BF 标志位被清零，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 I²C 总线空闲，用户可通过发出启动条件恢复通信。

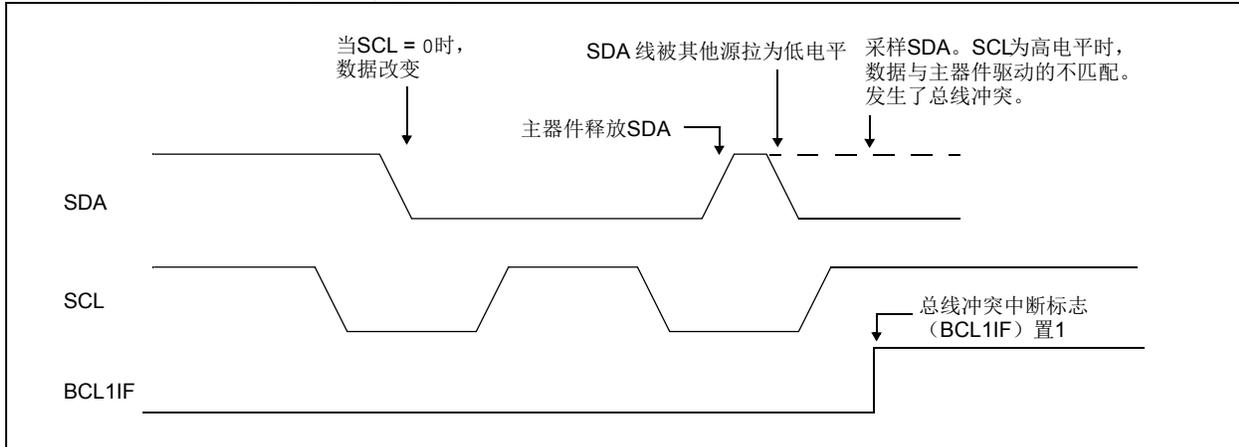
如果在启动、重复启动、停止或应答条件的进行过程中发生总线冲突，则这条件将被中止，SDA 和 SCL 线被置为无效，SSP1CON2 寄存器中的对应控制位清零。当执行总线冲突中断服务程序时，如果 I²C 总线空闲，用户可通过发出启动条件恢复通信。

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 寄存器的 P 位置 1 时，可以取得 I²C 总线的控制权；否则，总线处于空闲状态，S 位和 P 位都清零。

图 30-32: 发送和应答时的总线冲突时序



30.6.13.1 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

- 在启动条件开始时，SDA或SCL采样为低电平（图30-33）。
- SDA置为低电平之前，SCL采样为低电平（图30-34）。

在启动条件期间，SDA和SCL引脚都会被监视。

如果SDA引脚或SCL引脚已经是低电平，则发生以下所有事件：

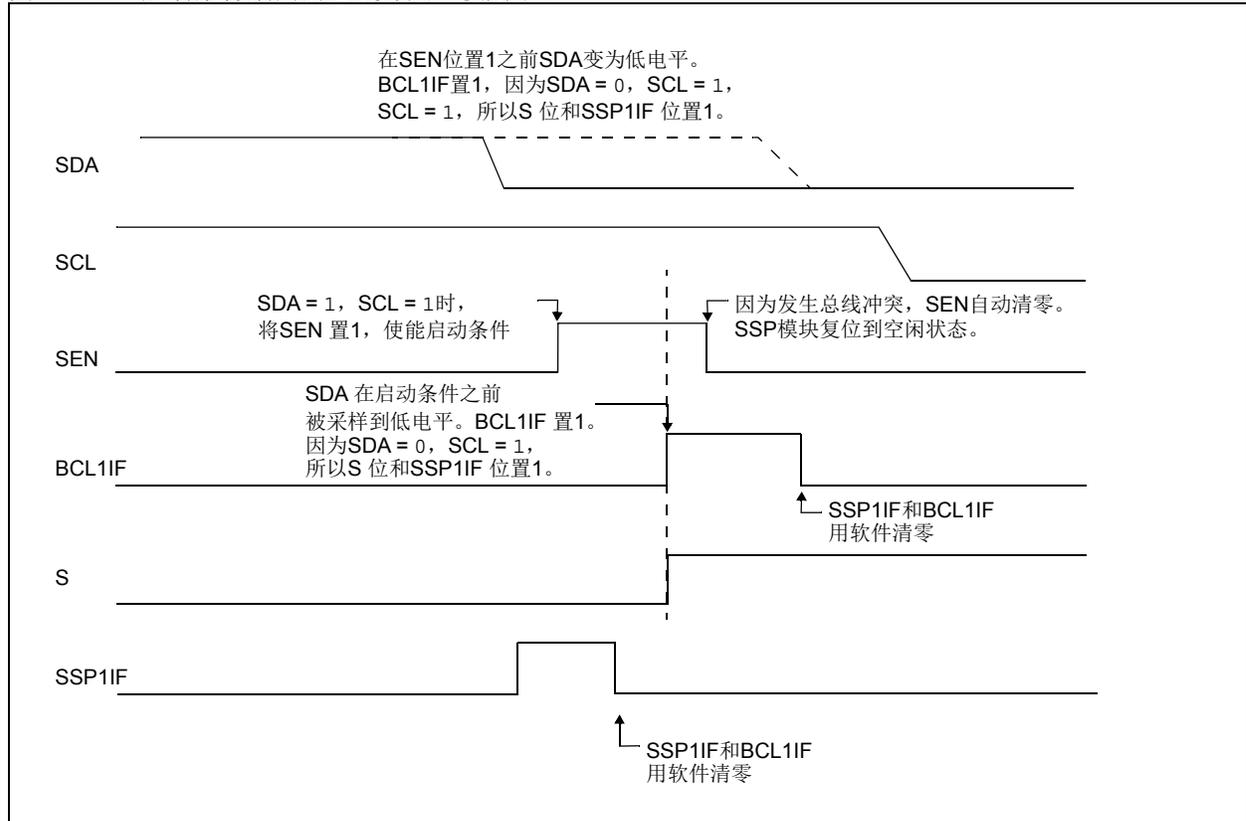
- 启动条件中止，
- BCL1IF标志置1，并且
- MSSP1模块复位为空闲状态（图30-33）。

启动条件从SDA和SCL引脚被置为无效时开始。当SDA引脚采样为高电平时，波特率发生器装入值并递减计数。如果在SDA为高电平时，SCL引脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图驱动一个数据1。

如果SDA引脚在该计数周期内采样为低电平，则BRG复位，且SDA线提前置为有效（图30-35）。但是，如果SDA引脚采样为1，则在BRG计数结束时该引脚将被置为低电平。接着，波特率发生器被重载并递减计数至0；在此期间，如果SCL引脚采样到0，则不会发生总线冲突。在BRG计数结束时，SCL引脚被置为低电平。

注： 在启动条件期间不会发生总线冲突，因为两个总线主器件不可能精确地在同一时刻发出启动条件。因此，一个主器件将总是先于另一个主器件将SDA置为有效。但是，上述情况不会引起总线冲突，因为两个主器件一定会对启动条件后的第一个地址进行仲裁。如果地址是相同的，必须继续对数据部分、重复启动条件或停止条件进行仲裁。

图30-33： 启动条件期间的总线冲突（仅用于SDA）



PIC16(L)F18324/18344

图30-34: 启动条件期间的总线冲突 (SCL = 0)

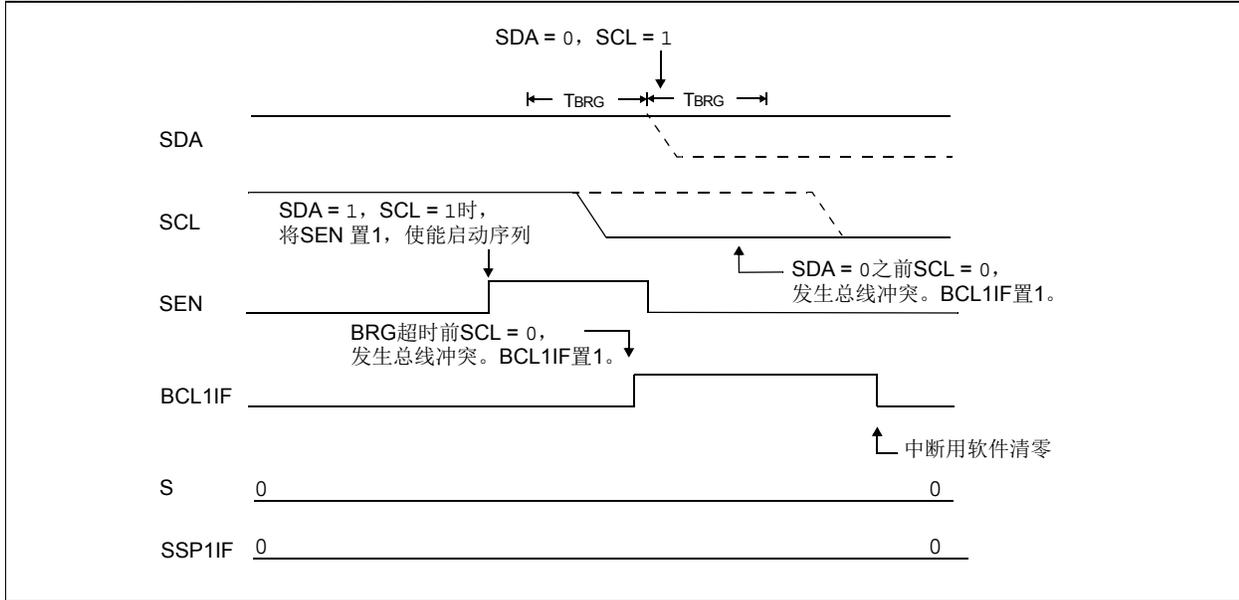
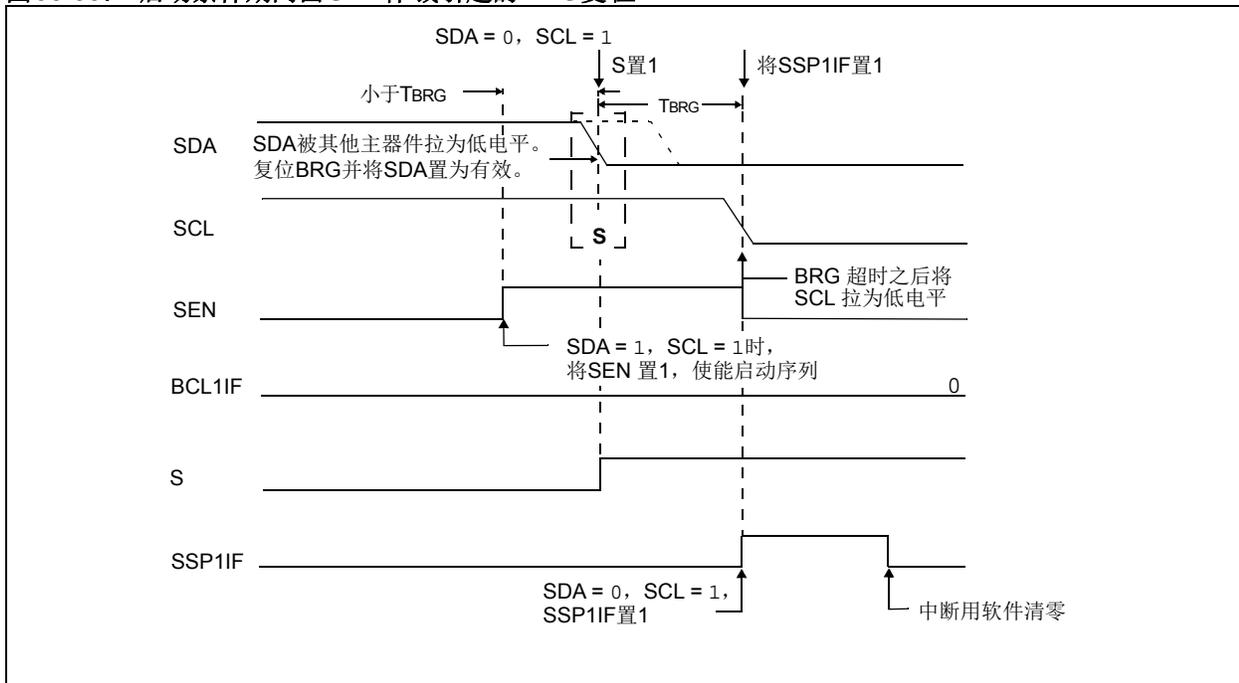


图30-35: 启动条件期间由SDA仲裁引起的BRG复位



30.6.13.2 重复启动条件期间的总线冲突

在重复启动条件期间，如果发生以下情况，则会发生总线冲突：

- 在SCL由低电平变为高电平期间，在SDA上采样到低电平（情形1）。
- 在SDA置为低电平之前，SCL变为低电平，表示另一个主器件正试图发送一个数据1（情形2）。

当用户释放SDA并允许该引脚悬空为高电平时，BRG装入SSP1ADD的值并递减计数至0。接着SCL引脚被置为无效，当SCL引脚采样到高电平时，对SDA引脚进行采样。

如果SDA为低电平，则已发生了总线冲突（即，另一个主器件正试图发送一个数据0，见图30-36）。如果SDA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。在这种情况下，另一个主器件在重复启动条件期间正在尝试发送数据1（见图30-37）。

如果在BRG超时结束时SCL和SDA都仍然是高电平，则SDA引脚被驱动为低电平，BRG被重载并开始计数。在计数结束时，不管SCL引脚的状态如何，SCL引脚都被驱动为低电平，重复启动条件结束。

图30-36： 重复启动条件期间的总线冲突（情形1）

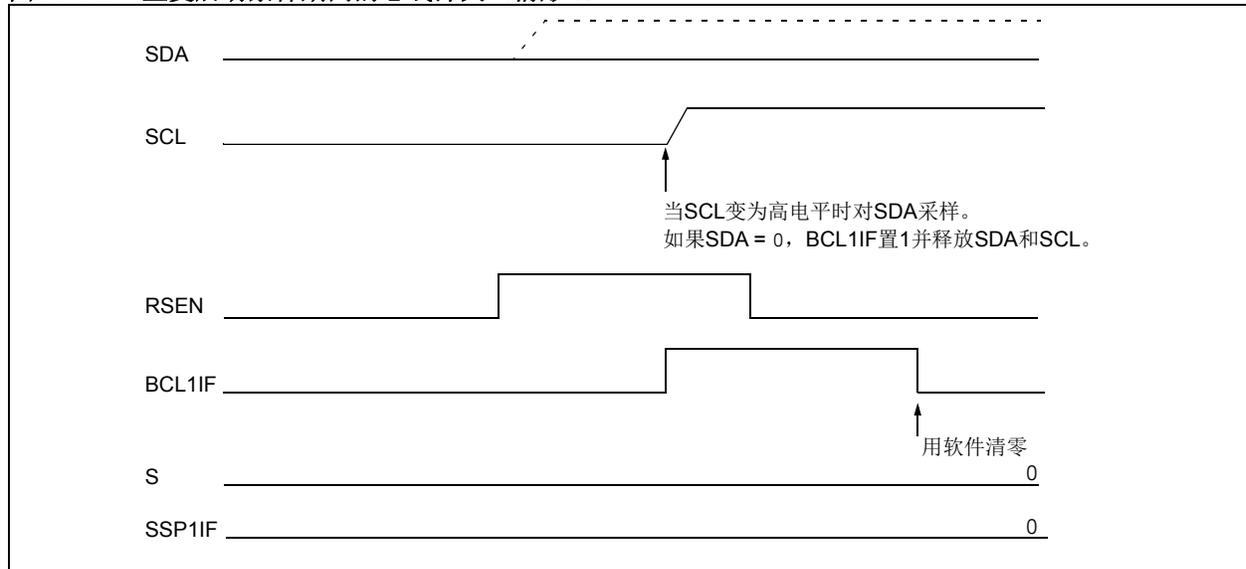
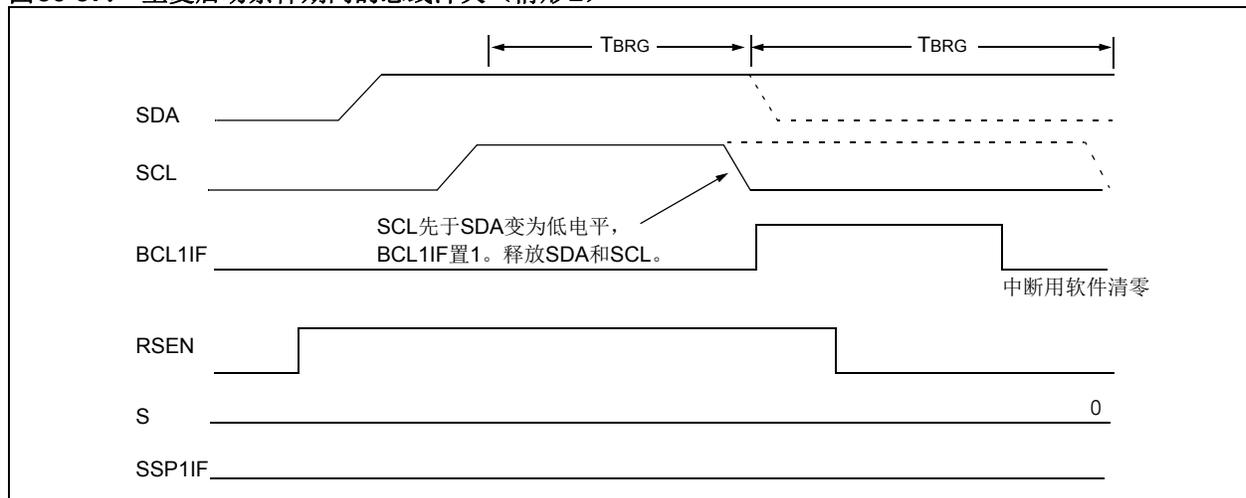


图30-37： 重复启动条件期间的总线冲突（情形2）



PIC16(L)F18324/18344

30.6.13.3 停止条件期间的总线冲突

以下事件会导致停止条件期间发生总线冲突：

- a) SDA引脚已被置为无效并允许悬空为高电平之后，SDA在BRG超时后被采样到低电平（情形1）。
- b) SCL引脚被置为无效之后，SCL在SDA变成高电平之前被采样到低电平（情形2）。

停止条件从SDA被置为低电平开始。当SDA采样为低电平时，允许SCL引脚悬空。当引脚被采样到高电平（时钟仲裁）时，波特率发生器装入SSP1ADD的值并递减计数至0。BRG超时后，SDA被采样。如果SDA采样为低电平，则已发生总线冲突。这是因为另一个主器件正在尝试驱动数据0（图30-38）。如果在允许SDA悬空为高电平前SCL引脚被采样到低电平，也会发生总线冲突。这是有另一个主器件正在尝试驱动数据0的另一种情况（图30-39）。

图30-38： 停止条件期间的总线冲突（情形1）

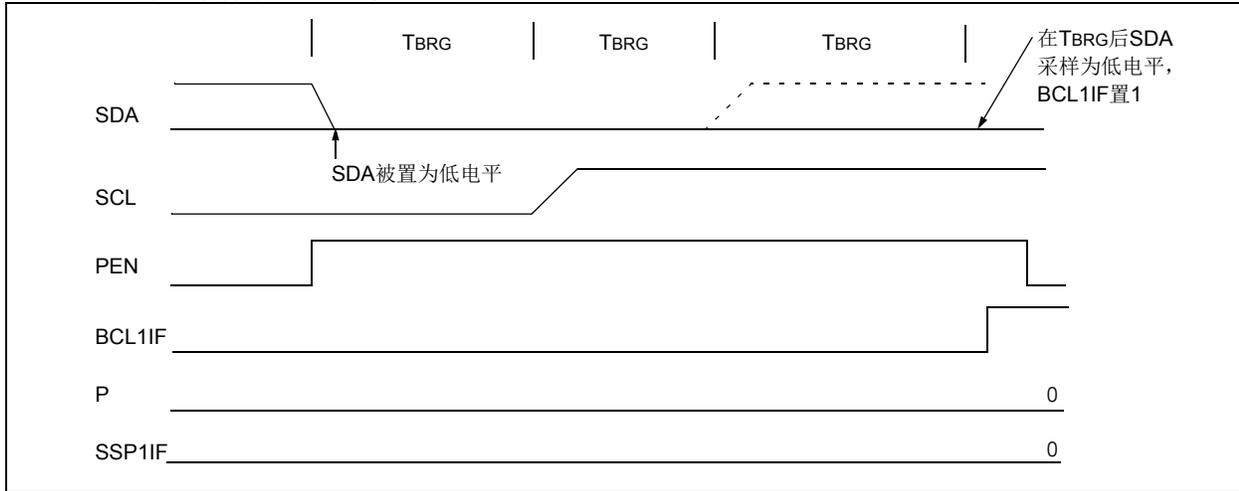
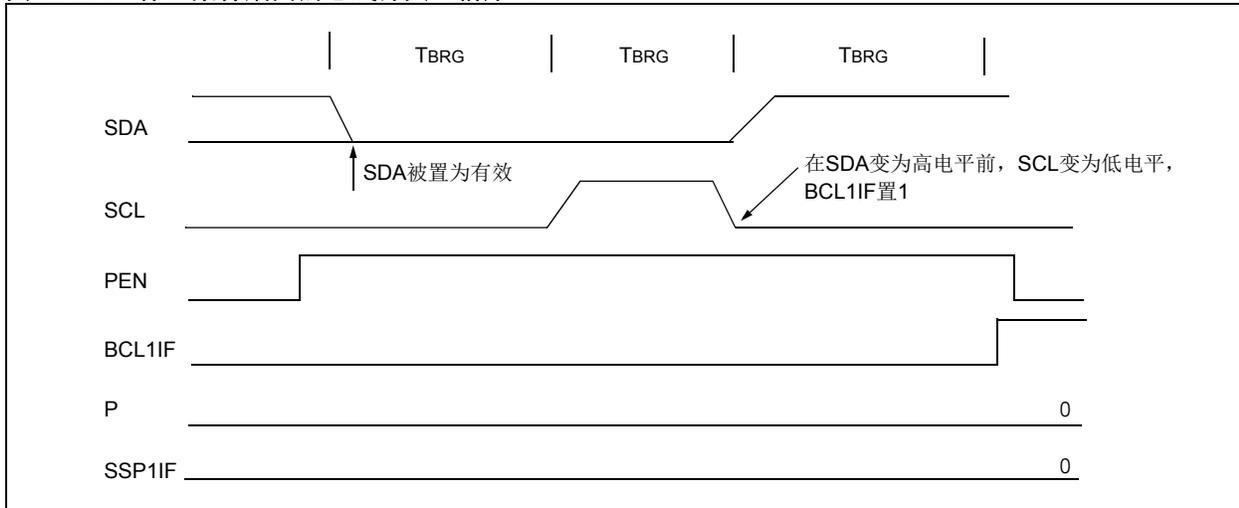


图30-39： 停止条件期间的总线冲突（情形2）



30.7 波特率发生器

MSSP1 模块具有一个波特率发生器，可用于在 I²C 和 SPI 主模式下产生时钟。波特率发生器 (BRG) 重载值放在 SSP1ADD 寄存器 (寄存器 30-6) 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。

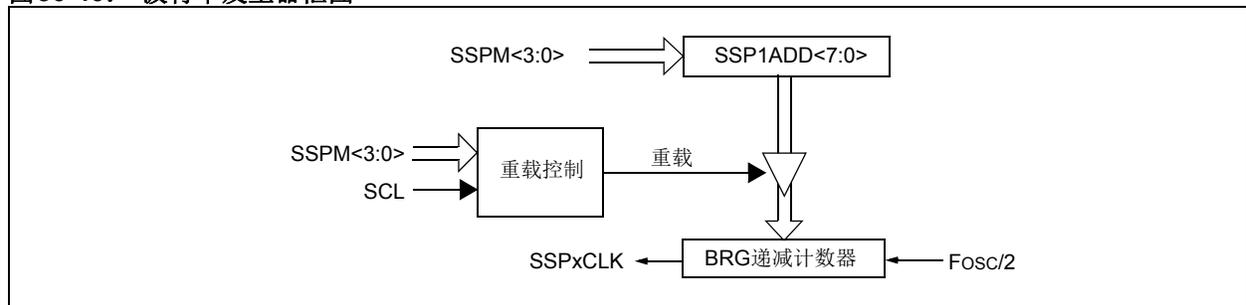
图 30-40 中的内部信号“重载”会触发将 SSP1ADD 值装入 BRG 计数器。对于模块时钟线的每次振荡，这会发生两次。指定重载信号何时置为有效的逻辑依赖于 MSSP1 当前的工作模式。

表 30-2 给出了基于指令周期以及 SSPADD 所装入 BRG 值的时钟速率。

公式 30-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

图 30-40: 波特率发生器框图



注: 在用作 I²C 的波特率发生器时，值 0x00、0x01 和 0x02 对于 SSP1ADD 是无效的。这是实现限制。

PIC16(L)F18324/18344

表30-2: 不同BRG值时的MSSP时钟速率

Fosc	Fcy	BRG 值	Fclock (两次BRG计满返回)
32 MHz	8 MHz	13h	400 kHz
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

注: 要确保所设计的系统支持IOL要求, 请参见表35-4中的I/O端口电气规范。

30.8 寄存器定义：MSSP控制

寄存器30-1: SSP1STAT: SSP1状态寄存器

R/W-0/0	R/W-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0	R/HS/HC-0/0
SMP	CKE ⁽¹⁾	D/Ā	P ⁽²⁾	S ⁽²⁾	R/W	UA	BF
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	HS/HC = 硬件置1/清零位

- bit 7 **SMP**: SPI数据输入采样位
SPI主模式:
 1 = 在数据输出时间的末端采样输入数据
 0 = 在数据输出时间的中间采样输入数据
SPI从模式:
 当SPI工作在从模式时, 必须将SMP清零
在I²C主模式或从模式下:
 1 = 标准速度模式下禁止压摆率控制 (100 kHz)
 0 = 高速模式下使能压摆率控制 (400 kHz)
- bit 6 **CKE**: SPI时钟边沿选择位 (仅限SPI模式) ⁽¹⁾
在SPI主模式或从模式下:
 1 = 时钟状态从有效转换到空闲时发送
 0 = 时钟状态从空闲转换到有效时发送
仅在I²C模式下:
 1 = 使能输入逻辑以使阈值符合SMBus规范
 0 = 禁止SMBus特定输入
- bit 5 **D/Ā**: 数据/地址位 (仅限I²C模式)
 1 = 指示上一个接收或发送的字节是数据
 0 = 指示上一个接收或发送的字节是地址
- bit 4 **P**: 停止位 ⁽²⁾
 (仅限I²C模式。在MSSP模块被禁止且SSPEN被清零时, 该位会被清零。)
 1 = 指示上次检测到停止位 (该位在复位时为0)
 0 = 上次未检测到停止位
- bit 3 **S**: 启动位 ⁽²⁾
 (仅限I²C模式。在MSSP模块被禁止且SSPEN被清零时, 该位会被清零。)
 1 = 指示上次检测到启动位 (该位在复位时为0)
 0 = 上次未检测到启动位
- bit 2 **R/W**: 读/写位信息 (仅限I²C模式)
 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进行逻辑或将指示MSSP是否处于空闲模式。
- bit 1 **UA**: 更新地址位 (仅限10位I²C模式)
 1 = 表示用户需要更新SSPADD寄存器中的地址
 0 = 不需要更新地址
- bit 0 **BF**: 缓冲区满状态位
接收 (SPI和I²C模式):
 1 = 接收完成, SSPBUF满
 0 = 接收未完成, SSPBUF空
发送 (仅限I²C模式):
 1 = 数据发送正在进行 (不包括ACK和停止位), SSPBUF满
 0 = 数据发送完成 (不包括ACK和停止位), SSPBUF空

注 1: 时钟极性状态由SSP1CON1寄存器的CKP位设置。

2: 该位在复位时以及SSPEN清零时清零。

PIC16(L)F18324/18344

寄存器 30-2: SSP1CON1: SSP1 控制寄存器 1

R/C/HS-0/0	R/C/HS-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WCOL	SSPOV ⁽¹⁾	SSPEN	CKP	SSPM<3:0>			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	HS = 硬件置1位 C = 用户清零位

- bit 7 **WCOL:** 写冲突检测位 (仅限发送模式)
 1 = 正在发送前一个字时, 又有数据写入SSPBUF寄存器 (必须用软件清零)
 0 = 未发生冲突
- bit 6 **SSPOV:** 接收溢出指示位⁽¹⁾
在SPI模式下:
 1 = SSPBUF寄存器仍存有前一数据时, 又接收到一个新的字节。如果发生溢出, SSPSR中的数据会丢失。溢出只会发生在从模式下发生。在从模式下, 即使只是发送数据, 用户也必须读SSPBUF, 以避免溢出位置1。在主模式下, 溢出位不会置1, 因为每次接收 (和发送) 新数据都是通过写入SSPBUF寄存器启动 (必须用软件清零)。
 0 = 无溢出
在I²C模式下:
 1 = SSPBUF寄存器仍存有前一字节时, 又接收到一个字节。在发送模式下, SSPOV是“无关位” (必须用软件清零)。
 0 = 无溢出
- bit 5 **SSPEN:** 同步串口使能位
 在两种模式下, 当使能时, 必须将下面这些引脚正确地配置为输入或输出
在SPI模式下:
 1 = 使能串口并将SCK、SDO、SDI和 \overline{SS} 配置为串口引脚源⁽²⁾
 0 = 禁止串口并将上述引脚配置为I/O端口引脚
在I²C模式下:
 1 = 使能串口并将SDA和SCL引脚配置为串口引脚源⁽³⁾
 0 = 禁止串口并将上述引脚配置为I/O端口引脚
- bit 4 **CKP:** 时钟极性选择位
在SPI模式下:
 1 = 时钟的空闲状态为高电平
 0 = 时钟的空闲状态为低电平
在I²C从模式下:
 SCL释放控制
 1 = 使能时钟
 0 = 保持时钟为低电平 (时钟延长) (用来确保数据建立时间。)
在I²C主模式下:
 在此模式下未使用

寄存器 30-2: SSP1CON1: SSP1 控制寄存器 1 (续)

bit 3-0

SSPM<3:0>: 同步串口模式选择位

1111 = I²C 从模式, 10 位地址, 并允许启动位和停止位中断

1110 = I²C 从模式, 7 位地址, 并允许启动位和停止位中断

1101 = 保留

1100 = 保留

1011 = I²C 固件控制的主模式 (从器件空闲)

1010 = SPI 主模式, 时钟 = $F_{osc}/(4 * (SSPADD+1))^{(5)}$

1001 = 保留

1000 = I²C 主模式, 时钟 = $F_{osc} / (4 * (SSPADD+1))^{(4)}$

0111 = I²C 从模式, 10 位地址

0110 = I²C 从模式, 7 位地址

0101 = SPI 从模式, 时钟 = SCK 引脚, 禁止 \overline{SS} 引脚控制, \overline{SS} 可用作 I/O 引脚

0100 = SPI 从模式, 时钟 = SCK 引脚, 使能 \overline{SS} 引脚控制

0011 = SPI 主模式, 时钟 = $T2_match/2$

0010 = SPI 主模式, 时钟 = $F_{osc}/64$

0001 = SPI 主模式, 时钟 = $F_{osc}/16$

0000 = SPI 主模式, 时钟 = $F_{osc}/4$

- 注 1:** 在主模式下, 溢出位不会置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSPBUF 寄存器启动。
- 2:** 当使能时, 必须将这些引脚正确地配置为输入或输出。使用 SSP1SSPPS、SSP1CLKPPS、SSP1DATPPS 和 RxyPPS 来选择引脚。
- 3:** 当使能时, 必须将 SDA 和 SCL 引脚配置为输入引脚。使用 SSP1CLKPPS、SSP1DATPPS 和 RxyPPS 来选择引脚。
- 4:** I²C 模式不支持 SSPADD 值为 0、1 或 2 的情况。
- 5:** 不支持 SSPADD 值为 0。而使用 SSPM = 0000。

PIC16(L)F18324/18344

寄存器 30-3: SSP1CON2: SSP1 控制寄存器 2 (仅限 I²C 模式) (1)

R/W-0/0	R/HS/HC-0	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HC = 硬件清零位 S = 用户置 1 位

- bit 7 **GCEN:** 广播呼叫使能位 (仅限 I²C 从模式)
 1 = 允许在 SSPSR 接收到广播呼叫地址 (0x00) 时产生中断
 0 = 禁止广播呼叫地址
- bit 6 **ACKSTAT:** 应答状态位 (仅限 I²C 模式)
 1 = 未接收到应答
 0 = 接收到应答
- bit 5 **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发送的值
 1 = 无应答
 0 = 应答
- bit 4 **ACKEN:** 应答序列使能位 (仅限 I²C 主模式)
在主接收模式下:
 1 = 在 SDA 和 SCL 引脚上发出应答序列, 并发送 ACKDT 数据位。由硬件自动清零。
 0 = 应答序列空闲
- bit 3 **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
- bit 2 **PEN:** 停止条件使能位 (仅限 I²C 主模式)
 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
- bit 1 **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
- bit 0 **SEN:** 启动条件使能/延长使能位
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 = 为从发送和从接收使能时钟延长
 0 = 禁止时钟延长

注 1: 对于 ACKEN、RCEN、PEN、RSEN 和 SEN 位: 如果 I²C 模块不处于空闲状态, 这些位可能不会置 1 (不支持并行操作), 并且可能也不会写入 SSPBUF。

寄存器 30-4: SSP1CON3: SSP1 控制寄存器 3

R-0/0	R/W-0/0						
ACKTIM ⁽³⁾	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **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
- bit 6 **PCIE:** 停止条件中断允许位 (仅限 I²C 模式)
 1 = 允许在检测到停止条件时产生中断
 0 = 禁止在检测到停止条件时产生中断 ⁽²⁾
- bit 5 **SCIE:** 启动条件中断允许位 (仅限 I²C 模式)
 1 = 允许在检测到启动或重复启动条件时产生中断
 0 = 禁止在检测到启动条件或重复启动条件时产生中断 ⁽²⁾
- bit 4 **BOEN:** 缓冲区改写使能位
在 SPI 从模式下: ⁽¹⁾
 1 = 每次移入一个新的数据字节, 便会更新 SSPBUF, 与 BF 位无关
 0 = 如果在 SSPSTAT 寄存器的 BF 位已置 1 的条件下接收到新字节, 则 SSPCON1 寄存器的 SSPOV 位置 1 且缓冲区不更新
在 I²C 主模式和 SPI 主模式下:
 该位被忽略。
在 I²C 从模式下:
 1 = 仅当 BF 位 = 0 时, 在接收到地址/数据字节时, 更新 SSPBUF 并产生 $\overline{\text{ACK}}$ 信号, 忽略 SSPOV 位的状态。
 0 = 仅当 SSPOV 位清零时更新 SSPBUF
- bit 3 **SDAHT:** SDA 保持时间选择位 (仅限 I²C 模式)
 1 = 在 SCL 的下降沿之后, 在 SDA 上最少有 300 ns 的保持时间
 0 = 在 SCL 的下降沿之后, 在 SDA 上最少有 100 ns 的保持时间
- bit 2 **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 PIR1 寄存器的 BCL1IF 位会置 1, 总线会变为空闲状态
 1 = 允许从模式总线冲突中断
 0 = 禁止从模式总线冲突中断
- bit 1 **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, SSPCON1 寄存器的 CKP 位将由硬件清零, SCL 将保持低电平。
 0 = 禁止地址保持
- bit 0 **DHEN:** 数据保持使能位 (仅限 I²C 从模式)
 1 = 在接收数据字节的 SCL 的第 8 个下降沿之后, 从硬件清零 SSPCON1 寄存器的 CKP 位且 SCL 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。当接收到新字节且 BF = 1 时, SSPOV 位仍然置 1, 但是硬件继续将最新字节写入 SSPBUF。

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。

PIC16(L)F18324/18344

寄存器 30-5: SSP1MSK: SSP1掩码寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SSP1MSK<7:0>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7-1 **SSP1MSK<7:1>**: 掩码位
 1 = 接收到的地址 bit n 与 SSPADD<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
- bit 0 **SSP1MSK<0>**: 用于 I²C 从模式、10 位地址的掩码位
I²C 从模式, 10 位地址 (SSPM<3:0> = 0111 或 1111):
 1 = 接收到的地址 bit 0 与 SSPADD<0> 相比较以检测 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):
 MSK0 位被忽略。

寄存器 30-6: SSP1ADD: SSP1地址和波特率寄存器 (I²C模式)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SSP1ADD<7:0>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

主模式:

- bit 7-0 **SSP1ADD<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——地址高字节:

- bit 7-3 **未使用**: 地址高字节不使用这 5 位。该寄存器中这几位的状态为“无关”。主器件发送的位模式由 I²C 规范确定, 必须等于 11110。但是, 这些位通过硬件进行比较, 不会受该寄存器中的值影响。
- bit 2-1 **SSP1ADD<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——地址低字节:

- bit 7-0 **SSP1ADD<7:0>**: 10 位地址的低 8 位

7位从模式:

- bit 7-1 **SSP1ADD<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”。

PIC16(L)F18324/18344

寄存器 30-7: SSP1BUF: SSP1 缓冲寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
SSP1BUF<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **SSP1BUF<7:0>**: MSSP 缓冲区位

表 30-3: 与 MSSP1 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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽³⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
INLVLA ⁽¹⁾	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	144
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽²⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
INLVLB ⁽²⁾	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	150
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INLVLC ⁽¹⁾	INLVLC7 ⁽²⁾	INLVLC6 ⁽²⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	157
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	357
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				358
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	360
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	361
SSP1MSK	SSP1MSK<7:0>								362
SSP1ADD	SSP1ADD<7:0>								362
SSP1BUF	SSP1BUF<7:0>								363
SSP1CLKPPS	—	—	—	SSP1CLKPPS<4:0>					160
SSP1DATPPS	—	—	—	SSP1DATPPS<4:0>					160
SSP1SSPPS	—	—	—	SSP1SSPPS<4:0>					160

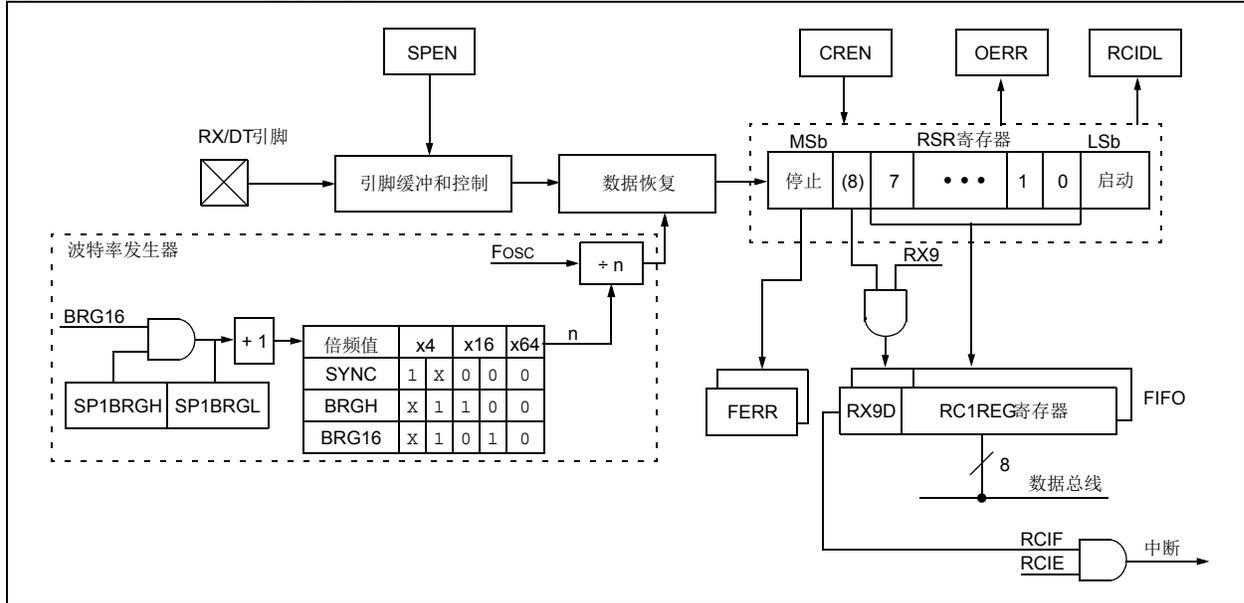
图注: — = 未实现位, 读为0。MSSP 模块不使用阴影单元。

注 1: 使用指定的 I²C 引脚时, 将忽略 INLVLCx 中的相关引脚值。

2: 仅限 PIC16(L)F18344。

3: 未实现, 读为1。

图31-2: EUSART1接收框图



EUSART1模块的操作由以下3个寄存器控制:

- 发送状态和控制寄存器 (TX1STA)
- 接收状态和控制寄存器 (RC1STA)
- 波特率控制寄存器 (BAUD1CON)

这些寄存器的详细信息请分别参见[寄存器31-1](#)、[寄存器31-2](#)和[寄存器31-3](#)。

RX和CK输入引脚分别使用RXPPS和CKPPS寄存器进行选择。TX、CK和DT输出引脚使用每个引脚的RxyPPS寄存器进行选择。由于在同步模式下RX输入与DT输出耦合在一起，所以在同步模式下工作时，用户需要负责为这两个功能选择同一引脚。EUSART1控制逻辑将自动控制数据方向驱动器。

PIC16(L)F18324/18344

31.1 EUSART1 异步模式

EUSART1 采用标准不归零 (non-return-to-zero, NRZ) 格式发送和接收数据。NRZ 实现为两种电平: VoH 标记状态 (Mark state) 代表 “1” 数据位, 而 VoL 空格状态 (Space state) 代表 “0” 数据位。NRZ 指的是连续发送的具有相同值的数据位保持在相应位的输出电平, 而不会在发送完每个位之后回到中间电平。NRZ 发送端口在标记状态下为空闲。每个字符发送包含 1 个启动位及随后的 8 个或 9 个数据位, 并始终由 1 个或多个停止位终止。启动位始终是一个空格, 停止位始终是标记。最常用的数据格式为 8 位。每个发送的位保持时间为 1/(波特率)。使用片上专用 8 位/16 位波特率发生器从系统振荡器产生标准波特率频率。波特率配置示例请参见表 31-3。

EUSART1 先发送和接收 LSb。EUSART1 的发送器和接收器在功能上是相互独立的, 但它们的数据格式和波特率相同。硬件不支持奇偶校验, 但可通过软件实现奇偶校验, 并将奇偶校验位作为第 9 个数据位存储。

31.1.1 EUSART1 异步发送器

图 31-1 给出了 EUSART1 发送器框图。发送器的核心是串行发送移位寄存器 (Transmit Shift Register, TSR), 该寄存器不可用软件直接访问。TSR 从发送缓冲区 (即 TX1REG 寄存器) 取得数据。

31.1.1.1 使能发送器

EUSART1 发送器可通过配置以下 3 个控制位使能为异步操作:

- TXEN = 1
- SYNC = 0
- SPEN = 1

假定所有其他 EUSART1 控制位均处于其默认状态。

将 TX1STA 寄存器的 TXEN 位置 1 可使能 EUSART1 的发送器电路。清零 TX1STA 寄存器的 SYNC 位可将 EUSART1 配置为异步操作。将 RC1STA 寄存器的 SPEN 位置 1 可使能 EUSART1 并自动将 TX/CK I/O 引脚配置为输出。如果 TX/CK 引脚与模拟外设共用, 则必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

注: TXEN 中断允许位置 1 时, TXIF 发送器中断标志位置 1。

31.1.1.2 发送数据

向 TX1REG 寄存器写入一个字符时启动发送。如果这是首字符, 或前一个字符被完全从 TSR 中送出, TX1REG 中的数据就立即被传送到 TSR 寄存器。如果 TSR 中仍保存前一个字符的全部或部分, 则新字符被保存在 TX1REG 中, 直到前一个字符的停止位被发送。之后, 在 TX1REG 中等待的字符在停止位发送后 1 个 Tcy 内被传送到 TSR 中。TX1REG 中的数据被传送到 TSR 后, 启动位、数据位和停止位序列的发送立即开始。

31.1.1.3 发送数据极性

可通过 BAUD1CON 寄存器的 SCKP 位来控制发送数据的极性。该位的默认状态为 0, 选择高电平有效发送空闲和数据位。将 SCKP 位设置为 1 将发送数据的极性取反, 从而选择低电平有效空闲和数据位。SCKP 位仅在异步模式下控制发送数据的极性。在同步模式下, SCKP 位有不同的功能。请参见第 31.4.1.2 节 “时钟极性”。

31.1.1.4 发送中断标志

只要 EUSART1 发送器被使能且 TX1REG 中没有等待发送的字符, PIR1 寄存器的 TXIF 中断标志位就被置 1。换句话说, 只有在 TSR 正在处理字符且 TX1REG 中还有一个排队等待发送的新字符时, TXIF 位才被清零。写入 TX1REG 时并不立即清零 TXIF 标志位。而是在之后的第二个指令周期将其清零。写入 TX1REG 后立即查询 TXIF 位将返回无效结果。TXIF 位是只读位, 不能用软件置 1 或清零。

将 PIE1 寄存器的 TXIE 中断允许位置 1 可允许 TXIF 中断。但是, 只要 TX1REG 为空, 无论 TXIE 中断允许位的状态如何, TXIF 标志位就会被置 1。

要在发送数据时使用中断, 应只在仍有数据要发送时才将 TXIE 位置 1。在将发送的最后一个字符写入 TX1REG 时应清零 TXIE 中断允许位。

31.1.1.5 TSR状态

TX1STA寄存器的TRMT位指示TSR寄存器的状态。该位是只读位。TSR寄存器为空时，TRMT位置1，而当一个字符从TX1REG传送到TSR寄存器中时，该位清零。TRMT位将保持清零，直到所有位移出TSR寄存器。该位不与任何中断逻辑关联，因此用户必须查询该位以确定TSR的状态。

注： TSR寄存器不映射到数据存储寄存器中，因此用户无法使用。

31.1.1.6 发送9位字符

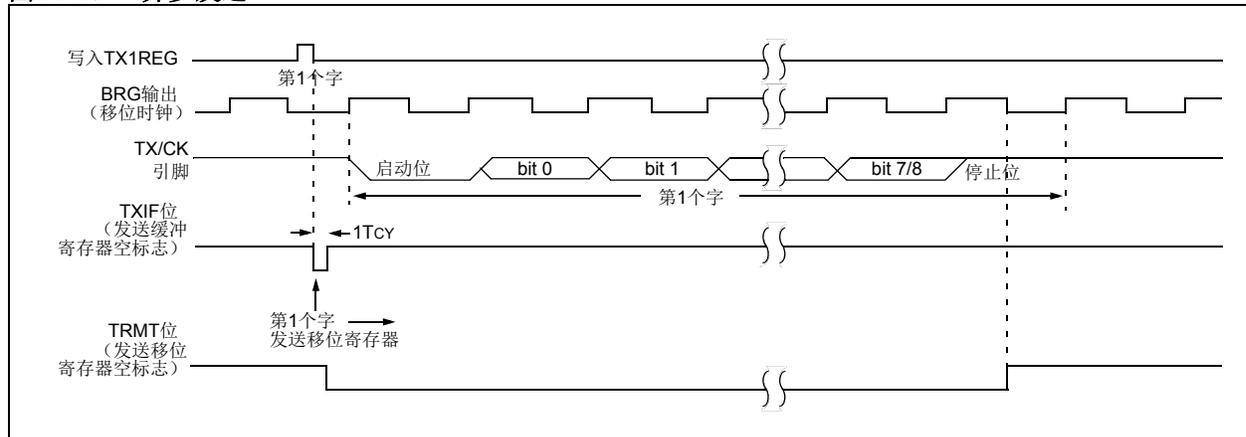
EUSART1支持9位字符发送。当TX1STA寄存器的TX9位置1时，EUSART1将在发送每个字符时移出9位。TX1STA寄存器的TX9D位是第9个数据位，也是最高有效位。发送9位数据时，TX9D数据位必须先于低8位写入TX1REG。写入TX1REG后，所有9个数据位将被立即传送到TSR移位寄存器。

有多个接收器时，可使用一种特殊的9位地址模式。关于地址模式的更多信息，请参见第31.1.2.7节“地址检测”。

31.1.1.7 异步发送设置

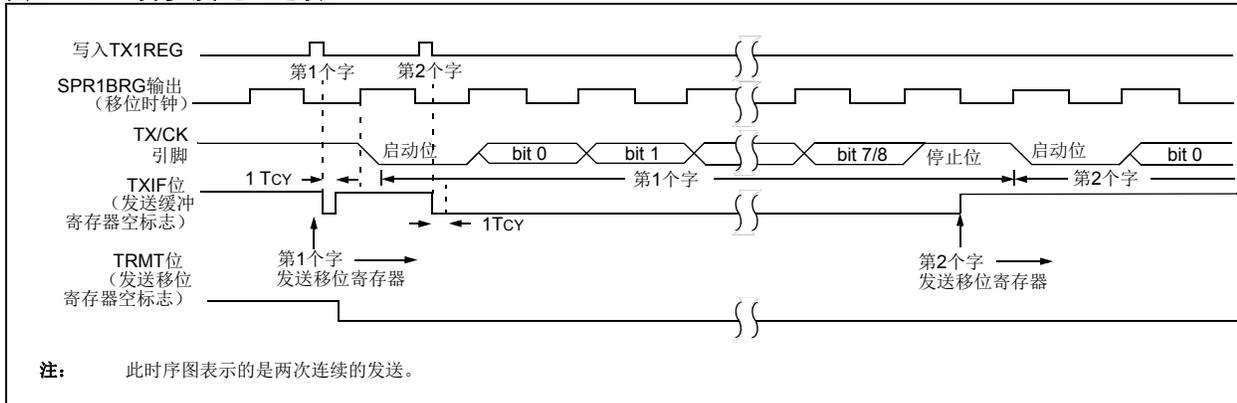
1. 初始化SP1BRGH和SP1BRGL寄存器对以及BRGH和BRG16位，以获得所需的波特率（见第31.3节“EUSART1波特率发生器（BRG）”）。
2. 通过清零SYNC位并将SPEN位置1，使能异步串口。
3. 如果需要9位发送，将TX9控制位置1。接收器置于地址检测模式时，第9个数据位置1表示8个低数据位为地址。
4. 如果需要将发送数据的极性取反，将SCKP位置1。
5. 将TXEN控制位置1使能发送。这将导致TXIF中断标志位置1。
6. 如果需要中断，将PIE1寄存器的TXIE中断允许位置1。如果INTCON寄存器的GIE和PEIE位也置1，则立即产生中断。
7. 如果选择了9位发送，应将第9位装入TX9D数据位。
8. 将8位数据装入TX1REG寄存器。这将启动发送。

图31-3： 异步发送



PIC16(L)F18324/18344

图31-4: 异步发送 (连续)



31.1.2 EUSART1异步接收器

异步模式通常用于RS-232系统中。图31-2给出了接收器框图。数据在RX/DT引脚上接收并驱动数据恢复模块。数据恢复模块实际上是一个高速移位器，工作频率为16倍波特率，而串行接收移位寄存器（Receive Shift Register, RSR）工作频率为比特率。所有8位或9位字符移入后被立即传送到双字符的先进先出（First-In-First-Out, FIFO）存储区中。软件开始处理EUSART1接收器前，FIFO缓冲区允许先接收两个完整字符和第三个字符的开始部分。FIFO和RSR寄存器不能直接用软件访问。通过RC1REG寄存器访问接收数据。

31.1.2.1 使能接收器

EUSART1接收器可通过配置以下3个控制位使能为异步操作：

- CREN = 1
- SYNC = 0
- SPEN = 1

假定所有其他EUSART1控制位均处于其默认状态。

将RC1STA寄存器的CREN位置1使能EUSART1的接收器电路。清零TX1STA寄存器的SYNC位可将EUSART1配置为异步操作。通过将RC1STA寄存器的SPEN位置1，可使能EUSART1。编程人员必须将相应的TRIS位置1，将RX/DT I/O引脚配置为输入。

注: 如果RX/DT功能位于模拟引脚上，则必须清零相应的ANSEL位使接收器工作。

31.1.2.2 接收数据

接收器的数据恢复电路在第一个位的下降沿启动字符接收。第一个位也称启动（Start）位，始终为零。数据恢复电路计数半个位的时间至启动位的中点并验证该位是否仍为零。如果该位非零，则数据恢复电路中字符接收，不产生错误，并恢复寻找启动位的下降沿。如果启动位被验证为零，则数据恢复电路计数一整个位时间至下个位的中点。该位被一个择多检测电路采样，其结果（0或1）被移入RSR。重复此过程直到所有数据位均被采样并移入RSR。最后一个位时间被测量且其电平被采样。它是停止位，总是为1。如果数据恢复电路在停止位处采样到0，则置1此字符的帧错误标志位，否则清零此字符的帧错误标志位。关于帧错误的更多信息，请参见第31.1.2.4节“接收帧错误”。

所有数据位和停止位被接收后，RSR中的字符就被立即传送到EUSART1接收FIFO，且PIR1寄存器的RCIF中断标志位被置1。读取RC1REG寄存器时，FIFO中顶部的字符被送出FIFO。

注: 如果接收FIFO溢出，在溢出条件被清除前不会接收更多字符。关于溢出错误的更多信息，请参见第31.1.2.5节“接收溢出错误”。

31.1.2.3 接收中断

只要EUSART1接收器被使能且接收FIFO中存在未被读取的字符，PIR1寄存器的RCIF中断标志位就会被置1。RCIF中断标志位是只读位，不能用软件置1或清零。

将以下所有位置1可允许RCIF中断：

- PIE1寄存器的中断允许位RCIE
- INTCON寄存器的外设中断允许位PEIE
- INTCON寄存器的全局中断允许位GIE

当FIFO中存在未被读取的字符时，无论中断允许位的状态如何，RCIF中断标志位均会被置1。

31.1.2.4 接收帧错误

接收FIFO缓冲区中的每个字符都有相应的帧错误状态位。帧错误表明在预期时间内未见到停止位。通过RC1STA寄存器的FERR位可访问帧错误状态。FERR位表示接收FIFO中顶部的未读字符的状态。因此，在读RC1REG之前必须先读FERR位。

FERR位是只读位，只适用于接收FIFO中顶部的未读字符。帧错误（FERR = 1）并不会禁止接收更多字符。此时不必将FERR位清零。从FIFO缓冲区读出下一个字符将使FIFO转到下一个字符和下一个相应的帧错误。

将RC1STA寄存器的SPEN位清零可复位EUSART1，这样就可将FERR位强制清零。将RC1STA寄存器的CREN位清零不影响FERR位。自身产生的帧错误不会产生中断。

注： 如果接收FIFO中的所有接收字符均有帧错误，重复读取RC1REG不会将FERR位清零。

31.1.2.5 接收溢出错误

接收FIFO缓冲区可容纳两个字符。在访问FIFO前接收到完整的第三个字符时会产生溢出错误。此时，RC1STA寄存器的OERR位置1。FIFO缓冲区中已有的字符可被读出，但溢出错误被清除前不能再接收其他字符。将RC1STA寄存器的CREN位清零或通过将RC1STA寄存器的SPEN位清零复位EUSART1，可清除该错误。

31.1.2.6 接收9位字符

EUSART1支持9位字符接收。当RC1STA寄存器的RX9位置1时，EUSART1将在接收每个字符时将9个位移入RSR。RC1STA寄存器的RX9D位是第9位，也是接收FIFO顶部未读字符的最高有效位。从接收FIFO缓冲区读取9位数据时，在读取RC1REG的低8位前必须先读取RX9D数据位。

31.1.2.7 地址检测

当多个接收器共用同一条传输线时（如在RS-485系统中），有一个特殊的地址检测模式可供使用。将RC1STA寄存器的ADDEN位置1可使能地址检测。

地址检测要求接收9位字符。使能地址检测时，只有第9个数据位置1的字符会被传送到接收FIFO缓冲区，从而将RCIF中断标志位置1。所有其他字符均被忽略。

接收到地址字符后，用户软件可判断地址是否与自身匹配。地址匹配时，发生下一个停止位前，用户软件必须通过清零ADDEN位禁止地址检测。当用户软件根据所使用的报文协议检测到报文的末尾时，软件将ADDEN位置1，将接收器重新置于地址检测模式。

PIC16(L)F18324/18344

31.1.2.8 异步接收设置

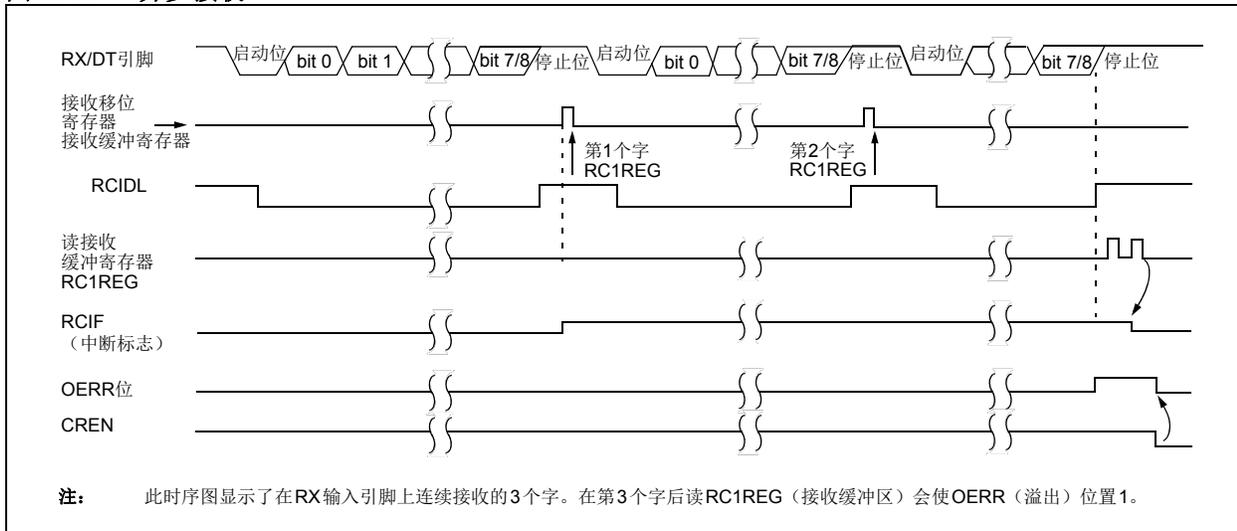
1. 初始化SP1BRGH:SP1BRGL寄存器对以及BRGH和BRG16位，获得所需的波特率（见第31.3节“EUSART1波特率发生器（BRG）”）。
2. 清零RX引脚的ANSEL位（如适用）。
3. 将SPEN位置1使能串口。SYNC位必须清零才能进行异步操作。
4. 如果需要中断，将PIE1寄存器的RCIE位以及INTCON寄存器的GIE和PEIE位置1。
5. 如果需要接收9位数据，将RX9位置1。
6. 将CREN位置1使能接收。
7. 当字符从RSR被移入接收缓冲区时，RCIF中断标志位将被置1。如果RCIE中断允许位也置1，则产生中断。
8. 读取RC1STA寄存器取得错误标志和第9个数据位（9位数据接收使能时）。
9. 读取RC1REG寄存器从接收缓冲区取得接收数据的低8位。
10. 发生溢出时，通过清零CREN接收器使能位清零OERR标志位。

31.1.2.9 9位地址检测模式设置

此模式通常用于RS-485系统中。设置使能地址检测的异步接收的步骤如下：

1. 初始化SP1BRGH:SP1BRGL寄存器对以及BRGH和BRG16位，获得所需的波特率（见第31.3节“EUSART1波特率发生器（BRG）”）。
2. 清零RX引脚的ANSEL位（如适用）。
3. 将SPEN位置1使能串口。SYNC位必须清零才能进行异步操作。
4. 如果需要中断，将PIE1寄存器的RCIE位以及INTCON寄存器的GIE和PEIE位置1。
5. 将RX9位置1使能9位接收。
6. 将ADDEN位置1使能地址检测。
7. 将CREN位置1使能接收。
8. 当第9位置1的字符从RSR被移入接收缓冲区时，RCIF中断标志位将被置1。如果RCIE中断允许位也置1，则产生中断。
9. 读取RC1STA寄存器取得错误标志。第9个数据位将始终置1。
10. 读取RC1REG寄存器从接收缓冲区取得接收数据的低8位。软件将判断此地址是否是器件地址。
11. 发生溢出时，通过清零CREN接收器使能位清零OERR标志位。
12. 如果器件被寻址，将ADDEN位清零以允许所有接收到的数据被送入接收缓冲区并产生中断。

图31-5： 异步接收



31.2 异步操作的时钟精度

内部振荡器模块输出（INTOSC）在出厂时做了校准。但是，VDD或温度变化时，INTOSC频率有可能漂移，这将直接影响异步波特率。有两种方法可用来调整波特率时钟，但它们都需要某种参考时钟源。

第一种（首选）方法使用OSCTUNE寄存器调整INTOSC输出。调整OSCTUNE寄存器的值可对系统时钟源的分辨率进行微调。更多信息，请参见第7.2.2.3节“内部振荡器频率调整”。

另一种方法调整波特率发生器的值。自动波特率检测可自动完成这种调整（见第31.3.1节“自动波特率检测”）。通过调整波特率发生器来补偿外设时钟频率的逐渐变化时，可能无法足够精细地调节分辨率。

31.3 EUSART1波特率发生器（BRG）

波特率发生器（BRG）是8位或16位定时器，专用于支持异步和同步EUSART1操作。默认情况下，BRG工作在8位模式下。将BAUD1CON寄存器的BRG16位置1可选择16位模式。

SP1BRGH和SP1BRGL寄存器对决定自由运行波特率定时器的周期。在异步模式下，波特率周期的倍频值由TX1STA寄存器的BRGH位和BAUD1CON寄存器的BRG16位决定。在同步模式下，BRGH位被忽略。

表31-1提供了确定波特率的公式。例31-1提供了确定波特率和波特率误差的计算示例。

为便于您使用，各种异步模式的典型波特率和误差值已经计算出来，如表31-3所示。使用高波特率（BRGH=1）或16位BRG（BRG16=1）有助于降低波特率误差。16位BRG模式用于在高速振荡器频率下实现低波特率。

将新值写入SP1BRGH和SP1BRGL寄存器对将导致BRG定时器复位（或清零）。这可以确保BRG无需等待定时器溢出就可以输出新的波特率。

如果系统时钟在有效的接收操作过程中被更改，可能会导致接收错误或数据丢失。为避免此问题，应检查RCIDL位的状态，以确保在改变系统时钟前接收操作处于空闲状态。

例31-1： 计算波特率误差

针对工作在异步模式下，Fosc = 16 MHz，目标波特率 = 9600且采用8位SP1BRG的器件：

$$\text{目标波特率} = \frac{F_{osc}}{64(\text{SP1BRGH}:\text{SP1BRGL} + 1)}$$

求解SP1BRGH:SP1BRGL：

$$X = \frac{F_{osc}}{\text{目标波特率}} - 1$$

$$= \frac{16000000}{9600} - 1$$

$$= [25.042] = 25$$

$$\text{计算波特率} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{误差} = \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

PIC16(L)F18324/18344

31.3.1 自动波特率检测

EUSART1 模块支持波特率自动检测和校准。

在自动波特率检测 (Auto-Baud Rate Detect, ABD) 模式下, 提供给 BRG 的时钟信号是反向的。BRG 并不为传入的 RX 信号提供时钟信号, 而是由 RX 信号为 BRG 定时。波特率发生器用于为接收的 55h (ASCII “U”) 定时, 55h 是 LIN 总线的同步字符。此字符的特殊之处在于它具有包括停止位边沿在内的 5 个上升沿。

通过将 BAUD1CON 寄存器的 ABDEN 位置 1, 可以启动自动波特率校准序列。当发生 ABD 序列时, EUSART1 状态机保持在空闲状态。在起始位之后, SPBRG 使用 BRG 计数器时钟在接收信号的第一个上升沿开始计数 (如图 31-6 所示)。在第 8 个位周期的末尾将在 RX 引脚上出现第 5 个上升沿。此时, 正确的 BRG 周期总数累计算值被保存在 SP1BRGH 和 SP1BRGL 寄存器对中, ABDEN 位被自动清零而 RCIF 中断标志被置 1。要清除 RCIF 中断, 需要读取 RC1REG 中的值。RC1REG 的内容应该被丢弃。在不使用 SP1BRGH 寄存器的模式下进行校准时, 用户可通过检查 SP1BRGH 寄存器的值是否为 00h 来验证 SP1BRGL 寄存器是否溢出。

BRG 自动波特率时钟由 BRG16 和 BRGH 位决定, 如表 31-1 所示。在 ABD 期间, SP1BRGH 和 SP1BRGL 寄存器都被用作 16 位计数器, 与 BRG16 位的设置无关。在校准波特率周期时, SP1BRGH 和 SP1BRGL 寄存器

的时钟频率为 BRG 基本时钟频率的 1/8。得到的字节测量结果为全速时的平均位时间。

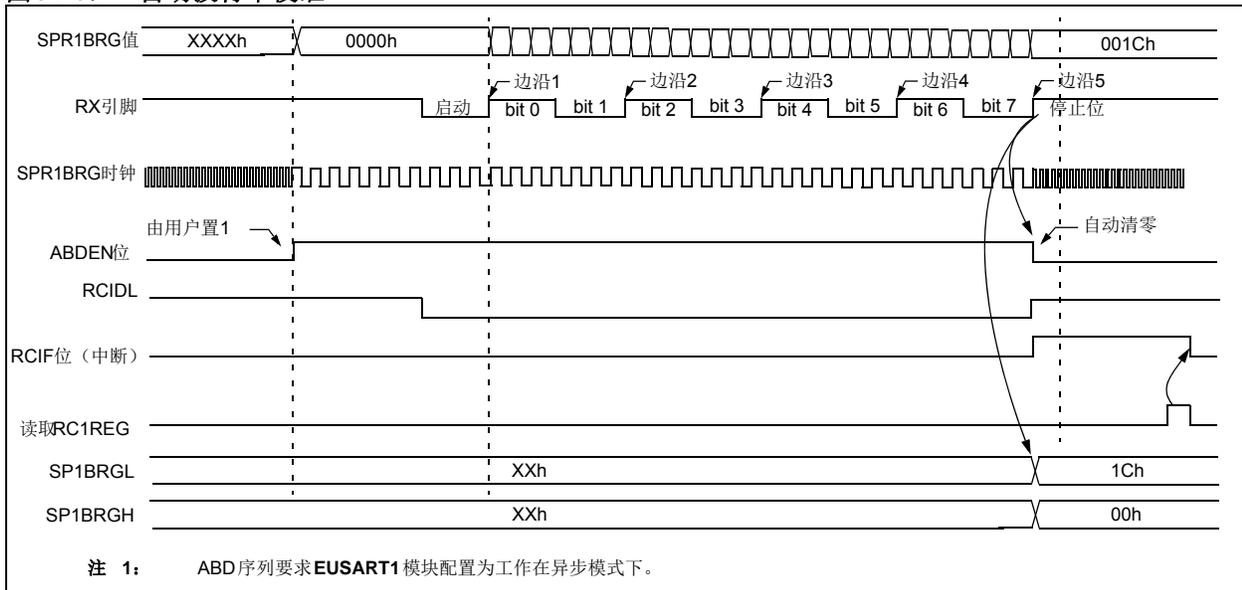
- 注 1:** 如果 WUE 位和 ABDEN 位都置 1, 自动波特率检测将从间隔字符之后的字节开始 (见第 31.3.3 节“接收到间隔字符时自动唤醒”)。
- 2:** 需要由用户来判断输入字符的波特率是否处于所选 BRG 时钟源范围内。可能无法实现某些振荡器频率和 EUSART1 波特率组合。
- 3:** 在自动波特率过程中, 自动波特率计数器从 1 开始计数。自动波特率序列完成后, 为了得到最准确的结果, 应从 SP1BRGH:SP1BRGL 寄存器对的值中减去 1。

表 31-1: BRG 计数器时钟速率

BRG16	BRGH	BRG 基本时钟	BRG ABD 时钟
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

注: 在 ABD 序列期间, SP1BRGL 和 SP1BRGH 寄存器都被用作 16 位计数器, 与 BRG16 的设置无关。

图 31-6: 自动波特率校准



31.3.2 自动波特率溢出

在自动波特率检测过程中，如果在RX引脚上检测到第5个上升沿之前波特率计数器溢出，则BAUD1CON寄存器的ABDOVF位将被置1。ABDOVF位指示计数器已超出SP1BRGH:SP1BRGL寄存器对的16位所能允许的最大计数值。溢出状态将RCIF标志位置1。计数器会继续计数，直到在RX引脚上检测到第5个上升沿为止。RCIDL位保持为假（0），直到第5个上升沿出现时RCIDL将置1。如果在发生溢出之后但在第5个上升沿之前读RC1REG，则第5个上升沿会再次将RCIF置1。

如果通过提早终止自动波特率过程来清除溢出条件，将无法正确检测到同步字符的第5个上升沿。如果在清零ABDEN位时尚未出现同步字符的任何下降沿，则这些下降沿会被错误地检测为启动位。建议通过以下步骤来清除溢出条件：

1. 通过读RC1REG来清零RCIF。
2. 如果RCIDL为零，则等待RCIF并重复步骤1
3. 清零ABDOVF位

31.3.3 接收到间隔字符时自动唤醒

在休眠模式下，EUSART1的所有时钟都会暂停。因此，波特率发生器处于不工作状态，不能正常进行字符接收。自动唤醒功能使控制器可被RX/DT线上的活动唤醒。该功能只在异步模式下可用。

自动唤醒功能可通过将BAUD1CON寄存器的WUE位置1来使能。一旦置1，RX/DT上的正常接收序列就被禁止，EUSART1保持在空闲状态，监视与CPU模式无关的唤醒事件。唤醒事件包含RX/DT线上电平由高至低的跳变。（这与同步间隔字符或LIN协议的唤醒信号字符的启动条件一致。）

EUSART1模块产生的RCIF中断与唤醒事件同时发生。在正常CPU工作模式下，与Q时钟同步产生中断（图31-7）；在休眠模式下，与Q时钟异步产生中断（图31-8）。通过读RC1REG寄存器可清除中断条件。

RX线在间隔字符末尾由低至高的跳变将自动清零WUE位。这向用户表明间隔事件结束。此时，EUSART1模块处于空闲模式，等待接收下一个字符。

31.3.3.1 特殊注意事项

间隔字符

在发生唤醒事件期间为了避免字符错误或字符碎片，唤醒字符必须为零。

唤醒被使能时，其工作状况与数据流的低电平时间无关。如果WUE位置1并接收到了有效的非零字符，则从启动位至第一个上升沿的低电平时间将被解读为唤醒事件。字符的其余位将作为碎片字符接收，后续字符有可能产生帧错误或溢出错误。

因此，发送的首字符必须为零。这必须持续10个或更长的位时间，对于LIN总线，建议持续13个位时间，而对于标准RS-232器件，可为任意个位时间。

振荡器起振时间

必须考虑振荡器起振时间，特别在使用起振时间较长的振荡器（即，LP、XT或HS/PLL模式）的应用中。同步间隔（或唤醒信号）字符必须足够长，并随后有一个足够长的间隔时间，以使所选的振荡器有足够的时间起振并在这段时间对EUSART1进行正确初始化。

WUE位

唤醒事件会通过将RCIF位置1产生一个接收中断。WUE位在RX/DT的上升沿由硬件清零。之后软件通过读取RC1REG寄存器并丢弃其内容将中断条件清除。

要确保不丢失实际数据，应在将WUE位置1前检查RCIDL位，验证没有接收操作在进行。如果未发生接收操作，可在进入休眠模式前将WUE位置1。

PIC16(L)F18324/18344

图31-7: 正常工作时的自动唤醒位 (WUE) 时序

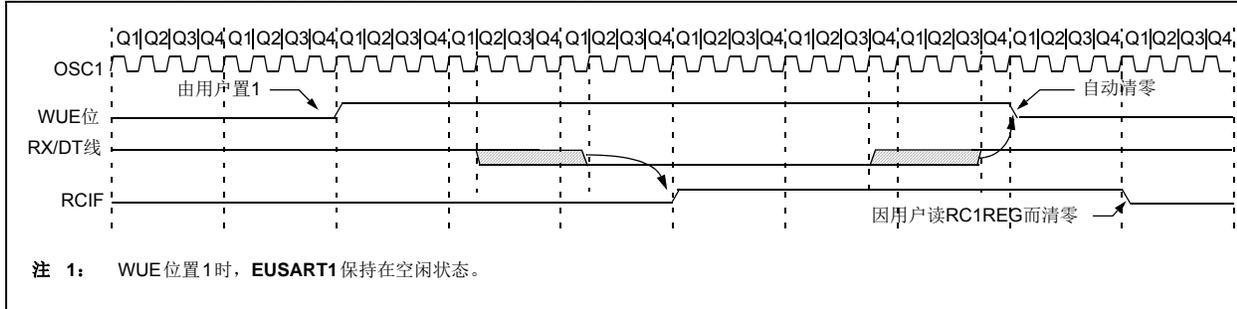
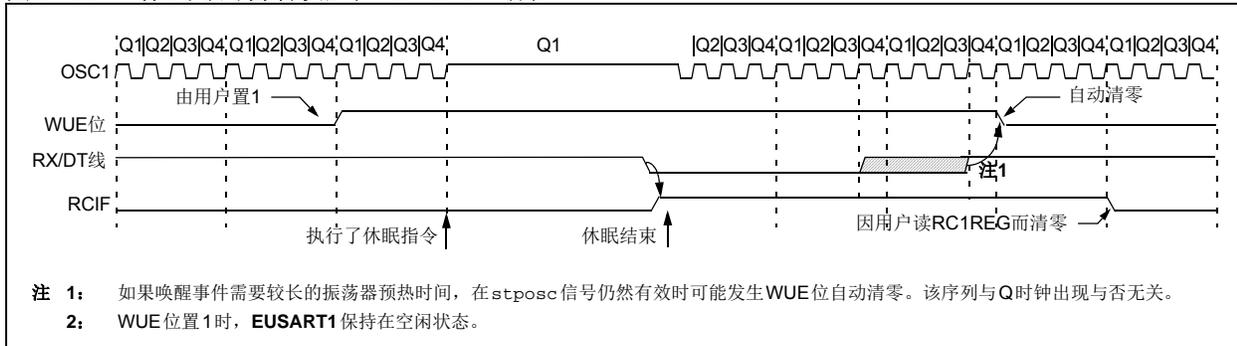


图31-8: 休眠时的自动唤醒位 (WUE) 时序



31.3.4 间隔字符序列

EUSART1 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。间隔字符包含 1 个启动位以及随后的 12 个 0 位和 1 个停止位。

要发送间隔字符, 应将 TX1STA 寄存器的 SENDB 和 TXEN 位置 1。随后对 TX1REG 执行写操作可启动间隔字符发送。写入 TX1REG 的数据值会被忽略并发送全 0。

在发送了相应的停止位后, 硬件会自动将 SENDB 位复位。这样用户可以在间隔字符 (在 LIN 规范中通常是同步字符) 后预先将下一个要发送字节装入发送 FIFO。

TX1STA 寄存器的 TRMT 位表明发送操作何时处于有效或空闲状态, 这与正常发送时相同。图 31-9 给出了发送间隔字符的时序。

31.3.4.1 间隔和同步发送序列

以下序列将启动报文帧头, 它由间隔字符和其后的自动波特率同步字节组成。这是 LIN 总线主器件的典型序列。

1. 将 EUSART1 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置 1 使能间隔序列。
3. 将无效字符装入 TX1REG, 启动发送 (该值会被忽略)。
4. 将 55h 写入 TX1REG, 以将同步字符装入发送 FIFO 缓冲区。
5. 发送间隔字符后, SENDB 位被硬件复位, 同步字符随后被发送。

当 TX1REG 为空时 (由 TXIF 指示), 下一个数据字节会写入 TX1REG。

31.3.5 接收间隔字符

增强型EUSART1模块接收间隔字符有两种方法。

第一种检测间隔字符的方法采用RC1STA寄存器的FERR位和RC1REG所指示的接收数据。假定波特率发生器已初始化为所需的波特率。

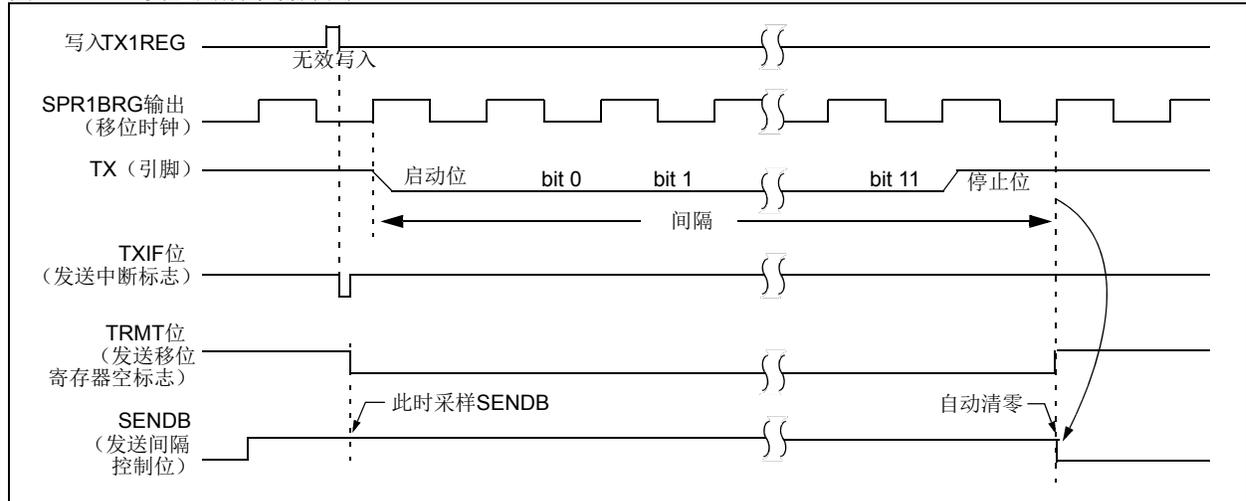
发生以下情况时，表明接收到间隔字符：

- RCIF位被置1
- FERR位被置1
- RC1REG = 00h

第二种方法采用第31.3.3节“接收到间隔字符时自动唤醒”中所述的自动唤醒功能。通过使能此功能，EUSART1将采样RX/DT上的下两次跳变，产生RCIF中断，并接收下一个数据字节并再产生一次中断。

请注意，在间隔字符后，用户通常希望使能自动波特率检测功能。采用这两种方法时，用户均可在EUSART1进入休眠模式前将BAUD1CON寄存器的ABDEN位置1。

图31-9： 发送间隔字符序列



PIC16(L)F18324/18344

31.4 EUSART1同步模式

同步串行通信通常用于具有一个主器件和一个或多个从器件的系统中。主器件包含生成波特率所需的电路，可将时钟提供给系统中的所有器件。从器件使用主时钟，可不再需要内部时钟生成电路。

同步模式下有两条信号线：一条双向数据线和一条时钟线。从器件使用主器件提供的外部时钟将串行数据移入或移出相应的接收和发送移位寄存器。由于数据线是双向的，同步操作只能是半双工的。半双工指主从器件能够接收和发送数据，但不能同时进行。EUSART1可作主器件，也可作为从器件。

同步发送时不使用启动位和停止位。

31.4.1 同步主模式

使用以下位将EUSART1配置为同步主操作：

- SYNC = 1
- CSRC = 1
- SREN = 0（用于发送）；SREN = 1（用于接收）
- CREN = 0（用于发送）；CREN = 1（用于接收）
- SPEN = 1

通过将TX1STA寄存器的SYNC位置1，可将器件配置为同步操作。通过将TX1STA寄存器的CSRC位置1，可将器件配置为主器件。通过将RC1STA寄存器的SREN和CREN位清零，可确保器件处于发送模式，否则器件将被配置为接收。通过将RC1STA寄存器的SPEN位置1，可使能EUSART1。

31.4.1.1 主时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为主器件的器件将时钟信号发送到TX/CK线上。EUSART1配置为同步发送或接收操作时，自动使能TX/CK引脚输出驱动器。串行数据位在时钟前沿改变，以确保其在每个时钟的后沿有效。为每个数据位产生一个时钟周期。数据位有多少，就产生多少个时钟周期。

31.4.1.2 时钟极性

为了与Microwire兼容，提供了时钟极性选项。时钟极性通过BAUD1CON寄存器的SCKP位进行选择。将SCKP位置1时，可将时钟空闲状态设置为高电平。当SCKP位置1时，数据在每个时钟的下降沿改变。将SCKP位清零时将时钟空闲状态设置为低电平。当SCKP位清零时，数据在每个时钟的上升沿改变。

31.4.1.3 同步主发送

从器件的RX/DT引脚发出数据。EUSART1配置为同步主发送操作时，RX/DT和TX/CK引脚的输出驱动器被自动使能。

向TX1REG寄存器写入一个字符时启动发送。如果TSR中仍保存前一个字符的全部或部分，则新字符被保存在TX1REG中，直到前一个字符的最后一位被发送。如果这是首字符，或前一个字符被完全从TSR中送出，TX1REG中的数据就立即被传送到TSR。字符发送在数据从TX1REG送入TSR后立即开始。

每个数据位在主时钟的时钟前沿改变，并在下一个时钟前沿到来前保持有效。

注： TSR寄存器不映射到数据存储中，因此用户无法使用。

31.4.1.4 同步主发送设置

1. 初始化SP1BRGH:SP1BRGL寄存器对以及BRGH和BRG16位，获得所需的波特率（见第31.3节“EUSART1波特率发生器（BRG）”）。
2. 将SYNC、SPEN和CSRC位置1，以使能同步主串口。
3. 将SREN和CREN位清零，以禁止接收模式。
4. 将TXEN位置1，以使能发送模式。
5. 如果需要9位发送，将TX9位置1。
6. 如果需要中断，将PIE1寄存器的TXIE位以及INTCON寄存器的GIE和PEIE位置1。
7. 如果选择了9位发送，应将第9位装入TX9D位。
8. 将数据装入TX1REG寄存器，启动发送。

图31-10: 同步发送

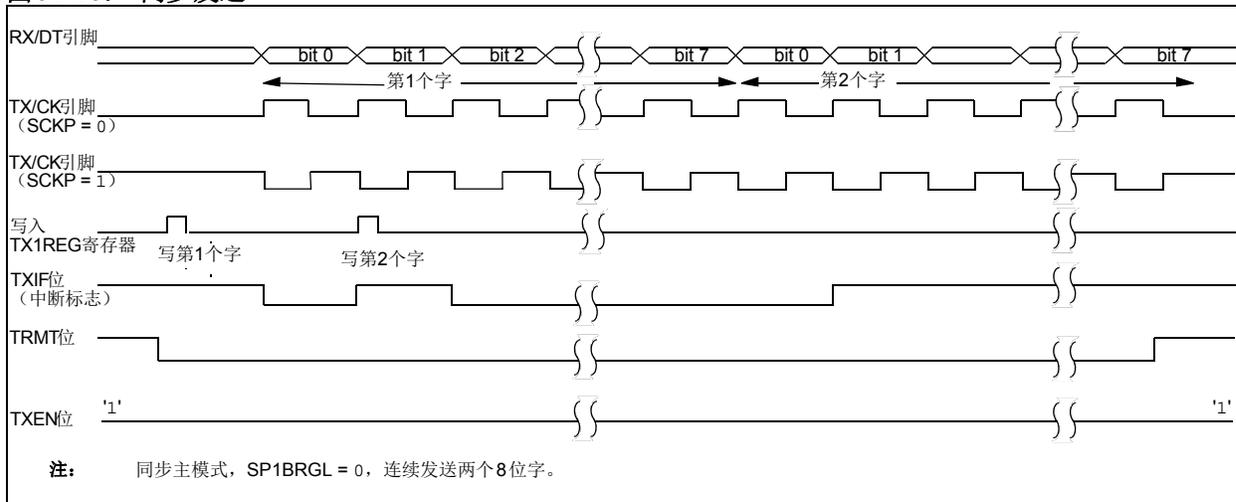
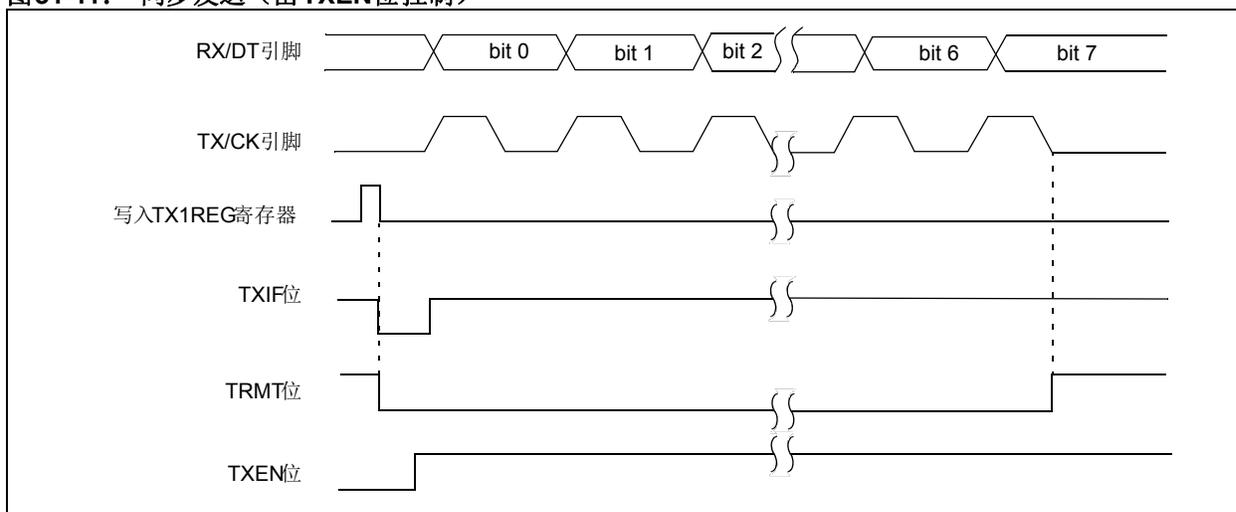


图31-11: 同步发送 (由TXEN位控制)



31.4.1.5 同步主接收

数据在RX/DT引脚上接收。将EUSART1配置为同步主接收操作时, 自动禁止RX/DT引脚输出驱动器。

在同步模式下, 可通过将单字符接收使能位 (RC1STA寄存器的SREN) 或连续接收使能位 (RC1STA寄存器的CREN) 置1使能接收。

SREN置1且CREN清零时, 一个字符中有多少数据位就产生多少个时钟周期。一个字符接收完成时SREN位被自动清零。CREN置1时, 将连续产生时钟直到CREN被清零。如果CREN在接收一个字符的过程中被清零, 则CK时钟立即停止, 接收到的部分字符被丢弃。如果SREN和CREN同时置1, 则首字符接收完成时SREN被清零, CREN优先。

要启动接收, 将SREN或CREN置1。在TX/CK时钟引脚的后沿对RX/DT引脚上的数据进行采样, 并移入接收移位寄存器 (RSR)。在完整的字符被接收到RSR时, RCIF位置1且该字符被自动送入两个字符的接收FIFO。接收FIFO中顶部字符的低8位在RC1REG中。只要接收FIFO中有未读字符, RCIF位就保持置1。

注: 如果RX/DT功能位于模拟引脚上, 则必须清零相应的ANSEL位使接收器工作。

PIC16(L)F18324/18344

31.4.1.6 从时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为从器件的器件在TX/CK线上接收时钟信号。将器件配置为同步从发送或接收操作时，自动禁止TX/CK引脚输出驱动器。串行数据位在时钟前沿改变，以确保其在每个时钟的后沿有效。每个时钟周期传送一个数据位。数据位有多少，就产生多少个接收时钟周期。

注： 如果将器件配置为从器件并且TX/CK功能位于模拟引脚上，则必须清零相应的ANSEL位。

31.4.1.7 接收溢出错误

接收FIFO缓冲区可容纳两个字符。在RC1REG被读取以访问FIFO前，接收到完整的第三个字符时会产生溢出错误。此时，RC1STA寄存器的OERR位置1。FIFO中的前一个数据不会被覆盖。FIFO缓冲区中的两个字符可被读出，但错误被清除前不能再接收其他字符。只有清除了溢出条件，才可将OERR位清零。如果SREN位置1且CREN清零时发生溢出错误，则读取RC1REG可清除错误。如果CREN位置1时发生溢出错误，则通过清零RC1STA寄存器的CREN位或清零SPEN位（该位清零将EUSART1复位），可以清除错误条件。

31.4.1.8 接收9位字符

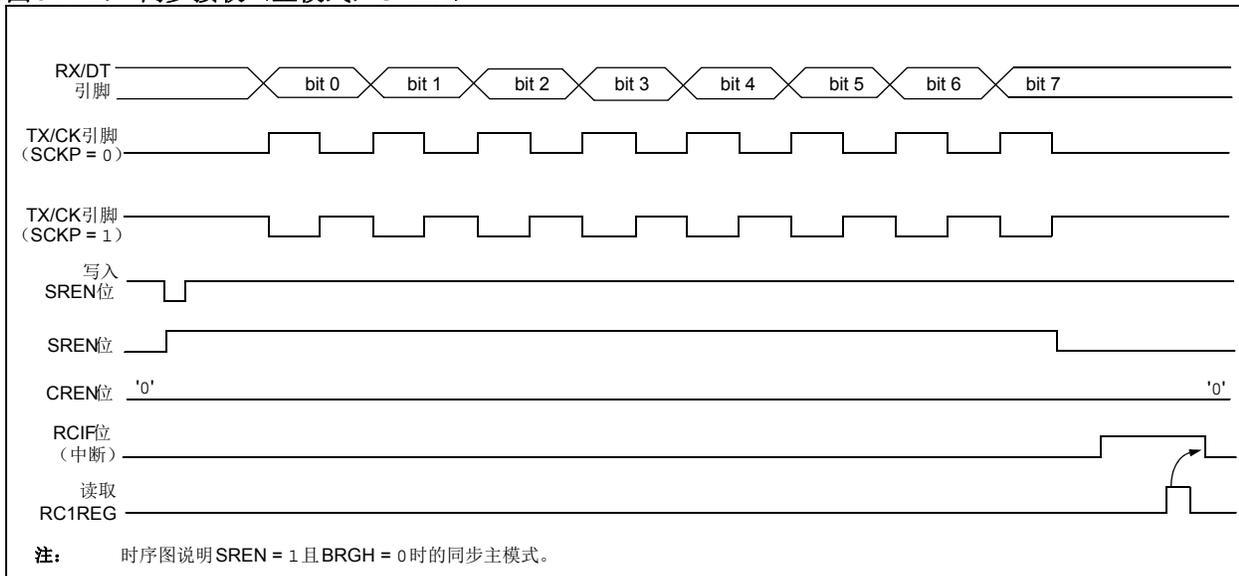
EUSART1支持9位字符接收。当RC1STA寄存器的RX9位置1时，EUSART1将在接收每个字符时将9个位移入RSR。RC1STA寄存器的RX9D位是第9位，也是接收

FIFO顶部未读字符的最高有效数据位。从接收FIFO缓冲区读取9位数据时，在读取RC1REG的低8位前必须先读取RX9D数据位。

31.4.1.9 同步主接收设置

1. 根据相应的波特率初始化SP1BRGH和SP1BRGL寄存器对。按需要将BRGH和BRG16位置1或清零，获得所需的波特率。
2. 清零RX引脚的ANSEL位（如适用）。
3. 将SYNC、SPEN和CSRC位置1，以使能同步主串口。
4. 确保将CREN和SREN位清零。
5. 如果需要中断，将PIE1寄存器的RCIE位以及INTCON寄存器的GIE和PEIE位置1。
6. 如果需要接收9位数据，将RX9位置1。
7. 将SREN位置1启动接收，或将CREN位置1使能连续接收。
8. 字符接收完成时中断标志位RCIF将被置1。如果中断允许位RCIE已置1，则产生中断。
9. 读取RC1STA寄存器取得第9位（如果已使能），并确定接收时是否发生了错误。
10. 读RC1REG寄存器来读取接收到的8位数据。
11. 如果发生了溢出错误，通过清零RC1STA寄存器的CREN位或清零SPEN位（该位清零将EUSART1复位），可以清除错误。

图31-12： 同步接收（主模式，SREN）



31.4.2 同步从模式

使用以下位将EUSART1配置为同步从操作:

- SYNC = 1
- CSRC = 0
- SREN = 0 (用于发送); SREN = 1 (用于接收)
- CREN = 0 (用于发送); CREN = 1 (用于接收)
- SPEN = 1

通过将TX1STA寄存器的SYNC位置1, 可将器件配置为同步操作。通过将TX1STA寄存器的CSRC位清零, 可将器件配置为从器件。通过将RC1STA寄存器的SREN和CREN位清零, 可确保器件处于发送模式, 否则器件将被配置为接收。通过将RC1STA寄存器的SPEN位置1, 可使能EUSART1。

31.4.2.1 EUSART1同步从发送

除了休眠模式以外, 同步主模式和从模式的工作是相同的(见第31.4.1.3节“同步主发送”)。

如果向TX1REG写入两个字, 然后执行SLEEP指令, 则会发生以下事件:

1. 第一个字符将立即传送到TSR寄存器并发送。
2. 第二个字将保留在TX1REG寄存器中。
3. TXIF位不会被置1。
4. 第一个字符移出TSR后, TX1REG寄存器会将第二个字符传送到TSR, 此时TXIF位将置1。
5. 如果PEIE和TXIE位均置1, 则发生中断将器件从休眠唤醒, 并执行下一条指令。如果GIE位也置1, 程序将调用中断服务程序。

31.4.2.2 同步从发送设置

1. 将SYNC和SPEN位置1并清零CSRC位。
2. 清零CK引脚的ANSEL位(如适用)。
3. 清零CREN和SREN位。
4. 如果需要中断, 将PIE1寄存器的TXIE位以及INTCON寄存器的GIE和PEIE位置1。
5. 如果需要9位发送, 将TX9位置1。
6. 将TXEN位置1, 以使能发送。
7. 如果选择了9位发送, 将最高有效位插入TX9D位。
8. 将低8位写入TX1REG寄存器, 启动发送。

31.4.2.3 EUSART1同步从接收

除下列各项外, 同步主模式和从模式的工作是相同的(第31.4.1.5节“同步主接收”):

- 休眠
- CREN位始终置1, 因此接收器从不空闲
- SREN位在从模式下为“无关位”

进入休眼前将CREN位置1, 可在休眠模式下接收一个字符。接收到数据字后, RSR寄存器会将数据传送到RC1REG寄存器。如果RCIE中断允许位置1, 产生的中断会将器件从休眠唤醒并执行下一条指令。如果GIE位也置1, 程序将跳转到中断向量。

31.4.2.4 同步从接收设置

1. 将SYNC和SPEN位置1并清零CSRC位。
2. 清零CK和DT引脚的ANSEL位(如适用)。
3. 如果需要中断, 将PIE1寄存器的RCIE位以及INTCON寄存器的GIE和PEIE位置1。
4. 如果需要接收9位数据, 将RX9位置1。
5. 将CREN位置1, 以使能接收。
6. 接收完成时RCIF位将被置1。如果RCIE位已置1, 则产生中断。
7. 如果使能了9位模式, 从RC1STA寄存器的RX9D位取出最高有效位。
8. 读取RC1REG寄存器, 从接收FIFO取出低8位。
9. 如果发生了溢出错误, 通过清零RC1STA寄存器的CREN位或清零SPEN位(该位清零将EUSART1复位), 可以清除错误。

31.5 休眠期间的EUSART1操作

EUSART1只有在同步从模式下，才会在休眠模式下保持工作状态。所有其他模式都需要系统时钟，因此在休眠模式下无法产生使发送或接收移位寄存器工作必需的信号。

同步从模式使用外部产生的时钟运行发送和接收移位寄存器。

31.5.1 休眠期间的同步接收

要在休眠模式下接收，进入休眠模式前必须满足以下所有条件：

- RC1STA和TX1STA控制寄存器必须配置为同步从接收（见第31.4.2.4节“同步从接收设置”）。
- 如果需要中断，将PIE1寄存器的RCIE位以及INTCON寄存器的GIE和PEIE位置1。
- 必须通过读RC1REG清零RCIF中断标志位，以卸载接收缓冲区中等待处理的任何字符。

进入休眠模式时，器件将准备好分别在RX/DT和TX/CK引脚上接收数据和时钟信号。数据字由外部器件随着时钟完全移入时，PIR1寄存器的RCIF中断标志位将置1，从而将处理器从休眠模式唤醒。

从休眠模式唤醒时，将执行SLEEP指令后紧跟的指令。如果INTCON寄存器的全局中断允许（GIE）位也置1，则调用地址004h处的中断服务程序。

31.5.2 休眠期间的同步发送

要在休眠模式下发送，进入休眠模式前必须满足以下所有条件：

- RC1STA和TX1STA控制寄存器必须配置为同步从发送（见第31.4.2.2节“同步从发送设置”）。
- 必须通过将输出数据写入TX1REG来清零TXIF中断标志位，从而填充TSR和发送缓冲区。
- 如果需要中断，将PIE1寄存器的TXIE位和INTCON寄存器的PEIE位置1。
- 必须将PIE1寄存器的TXIE中断允许位和INTCON寄存器的PEIE中断允许位置1。

进入休眠模式时，器件将在TX/CK引脚上接收时钟信号，在RX/DT引脚上发送数据。TSR中的数据字完全由外部器件随着时钟移出后，TX1REG中等待的字节将传输到TSR，TXIF标志位置1。从而将处理器从休眠模式唤醒。此时，TX1REG可用于接受另一个要发送的字符，从而将清零TXIF标志。

从休眠模式唤醒时，将执行SLEEP指令后紧跟的指令。如果全局中断允许（GIE）位也置1，则调用地址0004h处的中断服务程序。

31.6 寄存器定义：EUSART1控制

寄存器 31-1: TX1STA: 发送状态和控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **CSRC:** 时钟源选择位
异步模式:
 在此模式下不使用——值被忽略
同步模式:
 1 = 主模式 (时钟由内部BRG产生)
 0 = 从模式 (时钟来自外部时钟源)
- bit 6 **TX9:** 9位发送使能位
 1 = 选择9位发送
 0 = 选择8位发送
- bit 5 **TXEN:** 发送使能位⁽¹⁾
 1 = 使能发送
 0 = 禁止发送
- bit 4 **SYNC:** EUSART1 模式选择位
 1 = 同步模式
 0 = 异步模式
- bit 3 **SENDB:** 发送间隔字符位
异步模式:
 1 = 在下次发送时发送同步间隔字符——启动位, 后跟 12 个 0 位, 然后是停止位; 完成时由硬件清零
 0 = 禁止或已完成同步间隔字符的发送
同步模式:
 在此模式下不使用——值被忽略
- bit 2 **BRGH:** 高波特率选择位
异步模式:
 1 = 高速
 0 = 低速
同步模式:
 在此模式下不使用——值被忽略
- bit 1 **TRMT:** 发送移位寄存器状态位
 1 = TSR 空
 0 = TSR 满
- bit 0 **TX9D:** 发送数据的第9位
 可以是地址/数据位或奇偶校验位。

注 1: 在同步模式下, SREN/CREN可改写TXEN。

PIC16(L)F18324/18344

寄存器 31-2: RC1STA: 接收状态和控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-x/x
SPEN ⁽¹⁾	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7 **SPEN:** 串口使能位⁽¹⁾

1 = 使能串口

0 = 禁止串口 (保持在复位状态)

bit 6 **RX9:** 9位接收使能位

1 = 选择9位接收

0 = 选择8位接收

bit 5 **SREN:** 单字节接收使能位

异步模式:

在此模式下不使用——值被忽略

同步主模式:

1 = 使能单字节接收

0 = 禁止单字节接收

此位在接收完成后清零。

同步从模式:

在此模式下不使用——值被忽略

bit 4 **CREN:** 连续接收使能位

异步模式:

1 = 使能连续接收, 直到使能位CREN清零

0 = 禁止连续接收

同步模式:

1 = 使能连续接收, 直到使能位CREN清零 (CREN的优先级高于SREN)

0 = 禁止连续接收

bit 3 **ADDEN:** 地址检测使能位

9位异步模式 (RX9 = 1):

1 = 当接收缓冲区中的第9位置1时, 使能地址检测, 允许中断并装入接收缓冲区

0 = 禁止地址检测, 接收所有字节并且第9位可作为奇偶校验位

8位异步模式 (RX9 = 0):

在此模式下未使用——值被忽略

bit 2 **FERR:** 帧错误位

1 = 帧错误 (可以通过读RC1REG寄存器更新该位并接收下一个有效字节)

0 = 无帧错误

bit 1 **OERR:** 溢出错误位

1 = 溢出错误 (可以通过清零CREN位来清零该位)

0 = 无溢出错误

bit 0 **RX9D:** 接收数据的第9位

该位可以是地址/数据位或奇偶校验位, 并且必须由用户固件计算得到。

注 1: EUSART1 模块可自动将引脚的状态从三态更改为驱动为所需的状态。将TX/CK和RX/DT的相关TRIS位配置为1。

寄存器 31-3: BAUD1CON: 波特率控制寄存器

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7 **ABDOVF:** 自动波特率检测溢出位

异步模式:

- 1 = 自动波特率定时器溢出
- 0 = 自动波特率定时器未溢出

同步模式:

无关位

bit 6 **RCIDL:** 接收空闲标志位

异步模式:

- 1 = 接收器空闲
- 0 = 已接收到启动位且接收器正在接收

同步模式:

无关位

bit 5 **未实现:** 读为0

bit 4 **SCKP:** 时钟/发送极性选择位

异步模式:

- 1 = 发送 (TX) 的空闲状态为低电平
- 0 = 发送 (TX) 的空闲状态为高电平

同步模式:

- 1 = 时钟 (CK) 的空闲状态为高电平
- 0 = 时钟 (CK) 的空闲状态为低电平

bit 3 **BRG16:** 16位波特率发生器位

- 1 = 使用16位波特率发生器
- 0 = 使用8位波特率发生器

bit 2 **未实现:** 读为0

bit 1 **WUE:** 唤醒使能位

异步模式:

- 1 = EUSART将继续采样Rx引脚——在下降沿产生中断, 在下一个上升沿用硬件清零该位
- 0 = 不监视Rx引脚, 也不检测上升沿

同步模式:

在此模式下不使用——值被忽略

bit 0 **ABDEN:** 自动波特率检测使能位

异步模式:

- 1 = 使能在下一个字符传输过程中对波特率进行测量——需要接收到同步字段 (55h); 完成时由硬件清零
- 0 = 禁止波特率测量或测量已完成

0 = 禁止波特率测量或测量已完成

同步模式:

在此模式下不使用——值被忽略

PIC16(L)F18324/18344

寄存器 31-4: RC1REG⁽¹⁾: 接收数据寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RC1REG<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **RC1REG<7:0>**: 接收数据的低8位; 只读; 另请参见RX9D (寄存器31-2)

注 1: RC1REG (包括第9位) 为双重缓冲寄存器, 并且数据在接收到新数据时可用。

寄存器 31-5: TX1REG⁽¹⁾: 发送数据寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TX1REG<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **TX1REG<7:0>**: 发送数据的低8位; 只读; 另请参见TX9D (寄存器31-1)

注 1: TX1REG (包括第9位) 为双重缓冲寄存器, 可在开始移位前一个数据时写入。

寄存器 31-6: SP1BRGL⁽¹⁾: 波特率发生器寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SP1BRG<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **SP1BRG<7:0>**: 波特率发生器的低8位

注 1: 写SP1BRG将复位BRG计数器。

PIC16(L)F18324/18344

寄存器 31-7: SP1BRGH^(1, 2): 波特率发生器高字节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SP1BRG<15:8>							
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

bit 7 **SP1BRG<15:8>**: 波特率发生器的高8位

注 1: SP1BRGH值在所有模式下均被忽略, 除非BAUD1CON<BRG16>有效。

2: 写SP1BRGH将复位BRG计数器。

表31-2: 与EUSART1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	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	142
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	148
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	155
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	99
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	106
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	101
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	382
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	381
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	383
RC1REG	RC1REG<7:0>								384
TX1REG	TX1REG<7:0>								384
SP1BRGL	SP1BRG<7:0>								384
SP1BRGH	SP1BRG<15:8>								385
RXPPS	—	—	—	RXPPS<4:0>				160	
CLCxSELY	—	—	—	LCxDyS<4:0>				227	
MDSRC	—	—	—	—	MDMS<3:0>				270

图注: — = 未实现位, 读为0。EUSART1模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

PIC16(L)F18324/18344

表31-3: 波特率公式

配置位			BRG/EUSART1 模式	波特率公式
SYNC	BRG16	BRGH		
0	0	0	8位/异步	$F_{osc}/[64(n+1)]$
0	0	1	8位/异步	$F_{osc}/[16(n+1)]$
0	1	0	16位/异步	
0	1	1	16位/异步	$F_{osc}/[4(n+1)]$
1	0	x	8位/同步	
1	1	x	16位/同步	

图注: x = 无关位, n = SP1BRGH和SP1BRGL寄存器对的值。

表31-4: 异步模式下的波特率

波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	1221	1.73	255	1200	0.00	239	1200	0.00	143
2400	2404	0.16	207	2404	0.16	129	2400	0.00	119	2400	0.00	71
9600	9615	0.16	51	9470	-1.36	32	9600	0.00	29	9600	0.00	17
10417	10417	0.00	47	10417	0.00	29	10286	-1.26	27	10165	-2.42	16
19.2k	19.23k	0.16	25	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8
57.6k	55.55k	-3.55	3	—	—	—	57.60k	0.00	7	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

PIC16(L)F18324/18344

表31-4: 异步模式下的波特率 (续)

波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.64	2.12	16	113.64	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.63	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

PIC16(L)F18324/18344

表31-4: 异步模式下的波特率 (续)

波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)
300	300.0	0.00	26666	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215
1200	1200	0.00	6666	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303
2400	2400	0.01	3332	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151
9600	9604	0.04	832	9597	-0.03	520	9600	0.00	479	9600	0.00	287
10417	10417	0.00	767	10417	0.00	479	10425	0.08	441	10433	0.16	264
19.2k	19.18k	-0.08	416	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143
57.6k	57.55k	-0.08	138	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47
115.2k	115.9k	0.64	68	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23

波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)	实际波特率	% 误差	SPBRG 值 (十进制)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

32.0 参考时钟输出模块

参考时钟输出模块能够将时钟信号发送到参考时钟输出引脚（CLKR）。参考时钟输出还可用作其他外设的信号，如数据信号调制器（DSM）。

参考时钟输出模块具有以下特性：

- 系统时钟作为模块时钟源
- 可编程的时钟分频比
- 可选择的占空比

32.1 时钟源

参考时钟输出模块使用系统时钟（Fosc）作为时钟源。任何器件时钟切换将在时钟输出中反映。

32.1.1 时钟同步

参考时钟使能（CLKREN）位置1后，可确保模块在启动时无毛刺。

当禁止参考时钟输出时，输出信号将立即禁止。

当使能模块时，可更改时钟分频比和时钟占空比，但输出可能出现毛刺。为了避免可能的毛刺，时钟分频比和时钟占空比应仅在CLKREN清零时更改。

32.2 可编程时钟分频比

模块采用系统时钟输入并根据CLKRCON寄存器（[寄存器32-1](#)）的CLKRDIV<2:0>位对其进行分频。

可基于CLKRDIV<2:0>位进行以下配置：

- 基本Fosc值
- Fosc 2分频
- Fosc 4分频
- Fosc 8分频
- Fosc 16分频
- Fosc 32分频
- Fosc 64分频
- Fosc 128分频

时钟分频比值可在使能模块时更改；但是，为了防止对输出产生毛刺，只应在禁止模块（CLKREN = 0）时更改CLKRDIV<2:0>位。

32.3 可选占空比

CLKRCON寄存器的CLKRDC<1:0>位可以用来修改输出时钟的占空比。对于所有时钟速率，可选择25%、50%或75%占空比，但未经分频的基本Fosc值除外。

占空比可在使能模块时更改；但是，为了防止对输出产生毛刺，只应在禁止模块（CLKREN = 0）时更改CLKRDC<1:0>位。

注： CLKRDC1位复位为1。因此，默认占空比为50%。

32.4 休眠模式下的操作

参考时钟输出模块时钟基于系统时钟。当器件进入休眠模式时，模块输出将保持其当前状态。这将直接影响将参考时钟输出用作输入信号的外设。

PIC16(L)F18324/18344

图32-1: 参考时钟框图

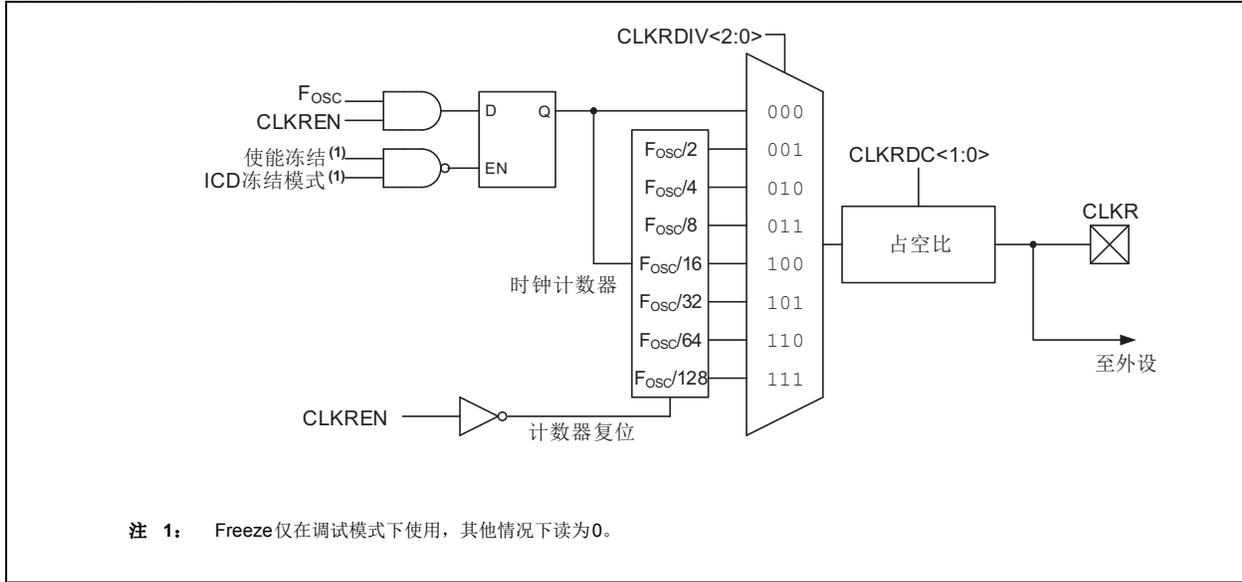
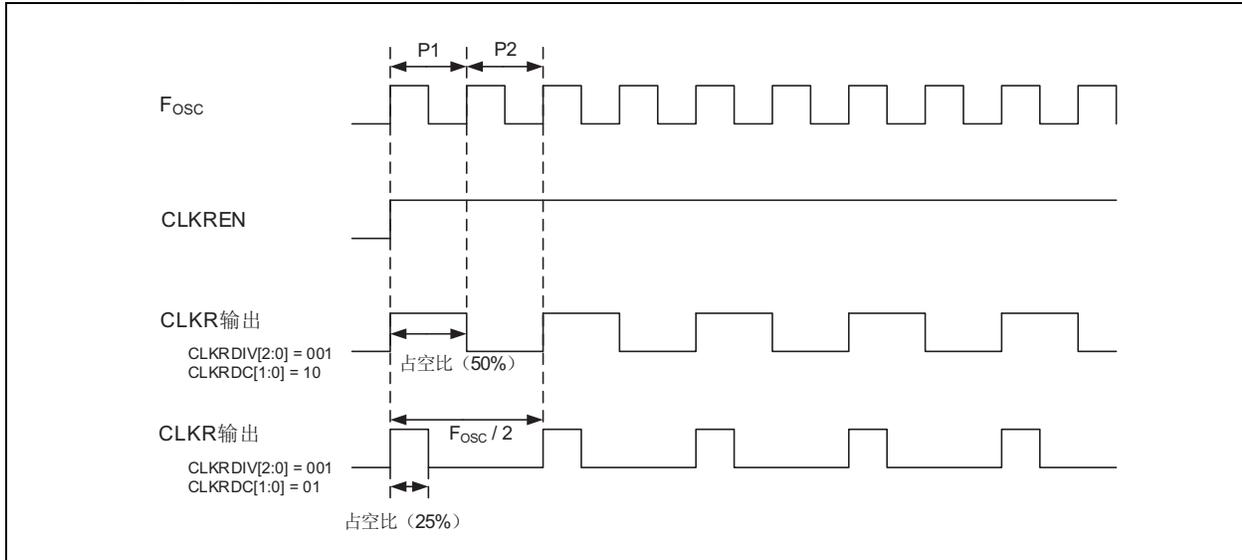


图32-2: 参考时钟时序



PIC16(L)F18324/18344

寄存器 32-1: CLKRCON: 参考时钟控制寄存器

R/W-0/0	U-0	U-0	R/W-1/1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CLKREN	—	—	CLKRDC<1:0>	CLKRDIV<2:0>			
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-n/n = POR和BOR时的值/所有其他复位时的值
1 = 置1	0 = 清零	

- bit 7 **CLKREN:** 参考时钟模块使能位
1 = 使能参考时钟模块
0 = 禁止参考时钟模块
- bit 6-5 **未实现:** 读为0
- bit 4-3 **CLKRDC<1:0>:** 参考时钟占空比位⁽¹⁾
11 = 时钟输出占空比为75%
10 = 时钟输出占空比为50%
01 = 时钟输出占空比为25%
00 = 时钟输出占空比为0%
- bit 2-0 **CLKRDIV<2:0>:** 参考时钟分频比位
111 = FOSC 128分频
110 = FOSC 64分频
101 = FOSC 32分频
100 = FOSC 16分频
011 = FOSC 8分频
010 = FOSC 4分频
001 = FOSC 2分频
000 = FOSC

注 1: 这些位对于为2或更大的参考时钟分频比值有效, 基本时钟无法再进一步分频。

表 32-1: 与参考时钟输出相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	141
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	147
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	154
CLKRCON	CLKREN	—	—	CLKRDC<1:0>	CLKRDIV<2:0>				225
CLCxSELY	—	—	LCxDyS<5:0>						227
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				271
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				272

图注: — = 未实现, 读为0。CLKR模块不使用阴影单元。

注 1: 仅限PIC16(L)F18344。

2: 未实现, 读为1。

PIC16(L)F18324/18344

33.0 在线串行编程 (ICSP™)

ICSP编程允许用户在生产电路板时使用未编程器件。编程可以在组装流程之后完成，从而可以使用最新版本的固件或者定制固件对器件编程。ICSP编程需要5个引脚：

- ICSPCLK
- ICSPDAT
- $\overline{\text{MCLR}}/\text{VPP}$
- VDD
- VSS

在编程/校验模式下，通过串行通信对程序存储器、数据EEPROM、用户ID和配置字进行编程。ICSPDAT引脚是用于传输串行数据的双向I/O，ICSPCLK引脚是时钟输入引脚。关于 ICSP 的更多信息，请参见“PIC16(L)F183XX Memory Programming Specification” (DS40001738)。

33.1 高电压编程模式

通过将ICSPCLK和ICSPDAT引脚保持为低电平，然后将 $\overline{\text{MCLR}}/\text{VPP}$ 上的电压升至 V_{IH}^{H} ，将器件置于高电压编程模式。

33.2 低电压编程模式

通过低电压编程模式，只需使用VDD就可以对PIC®闪存MCU进行编程，而无需使用高电压。当配置字的LVP位设置为1时，将会使能低电压ICSP编程模式。要禁止低电压ICSP模式，LVP位必须编程为0。只能使用高电压编程模式将LVP位再编程为0。

进入低电压编程模式需要执行以下步骤：

1. $\overline{\text{MCLR}}$ 电压设置为 V_{IL} 。
2. 在提供ICSPCLK时钟的同时，在ICSPDAT上送出32位密钥序列。

送完密钥序列后，在需要维持编程/校验模式的时间内，必须将 $\overline{\text{MCLR}}$ 保持为 V_{IL} 。

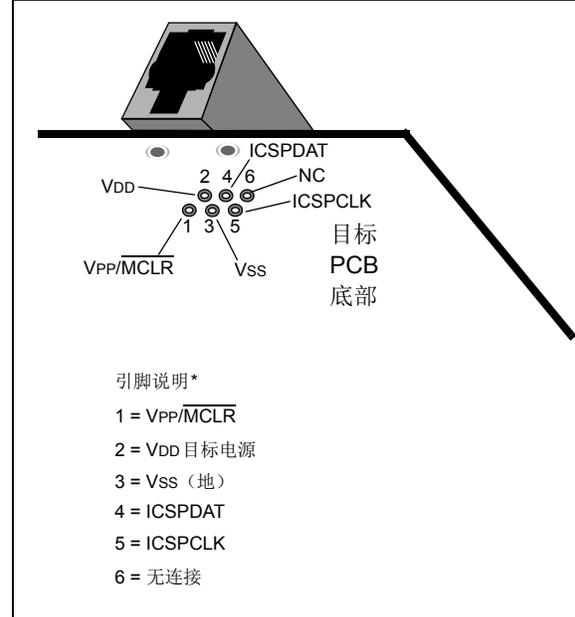
如果使能了低电压编程 (LVP = 1)，则 $\overline{\text{MCLR}}$ 复位功能会被自动使能，无法禁止。更多信息，请参见第6.4节“MCLR”。

LVP位只能通过使用高电压编程模式重新设定为0。

33.3 常用编程接口

通常，与目标器件的连接通过一个ICSP连接器完成。开发工具中常见的连接器是采用6P6C（6引脚，6连接器）配置的RJ-11。请参见图33-1。

图33-1： ICD RJ-11型连接器接口



另一种常用于PICKIT™编程器的连接器是间距为0.1英寸间距的标准6引脚插头。请参见图33-2。

关于其他接口建议，请在进行PCB设计之前参考具体的器件编程器手册。

建议使用隔离器件来隔离编程引脚与其他电路。隔离的类型高度依赖于具体应用，可能会包含诸如电阻、二极管甚至跳线之类的器件。更多信息，请参见图33-3。

图 33-2: PICkit™ 编程器型连接器接口

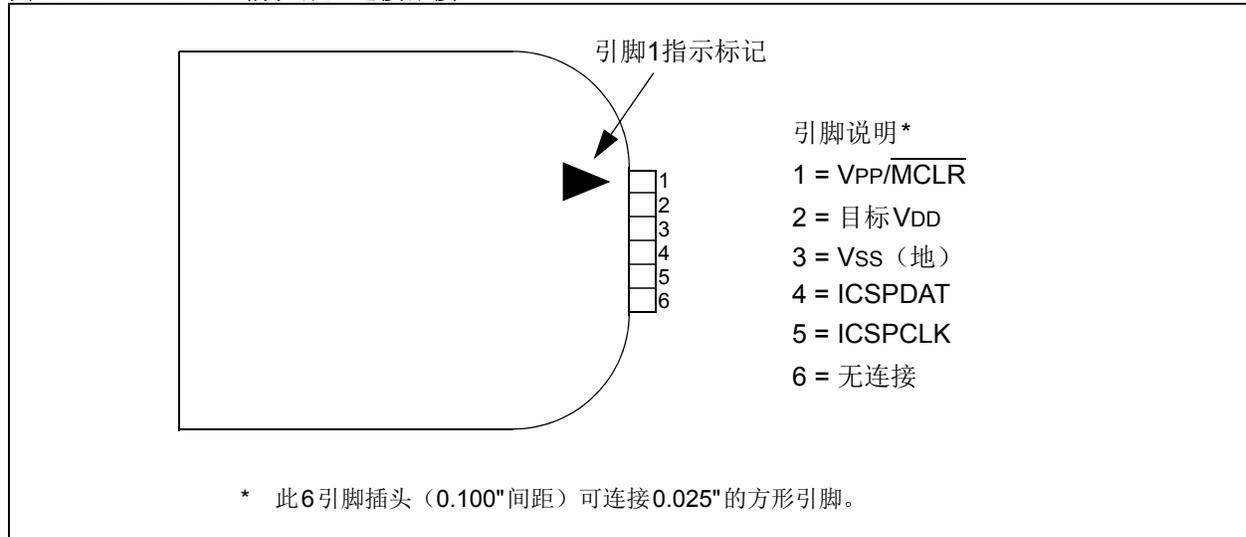
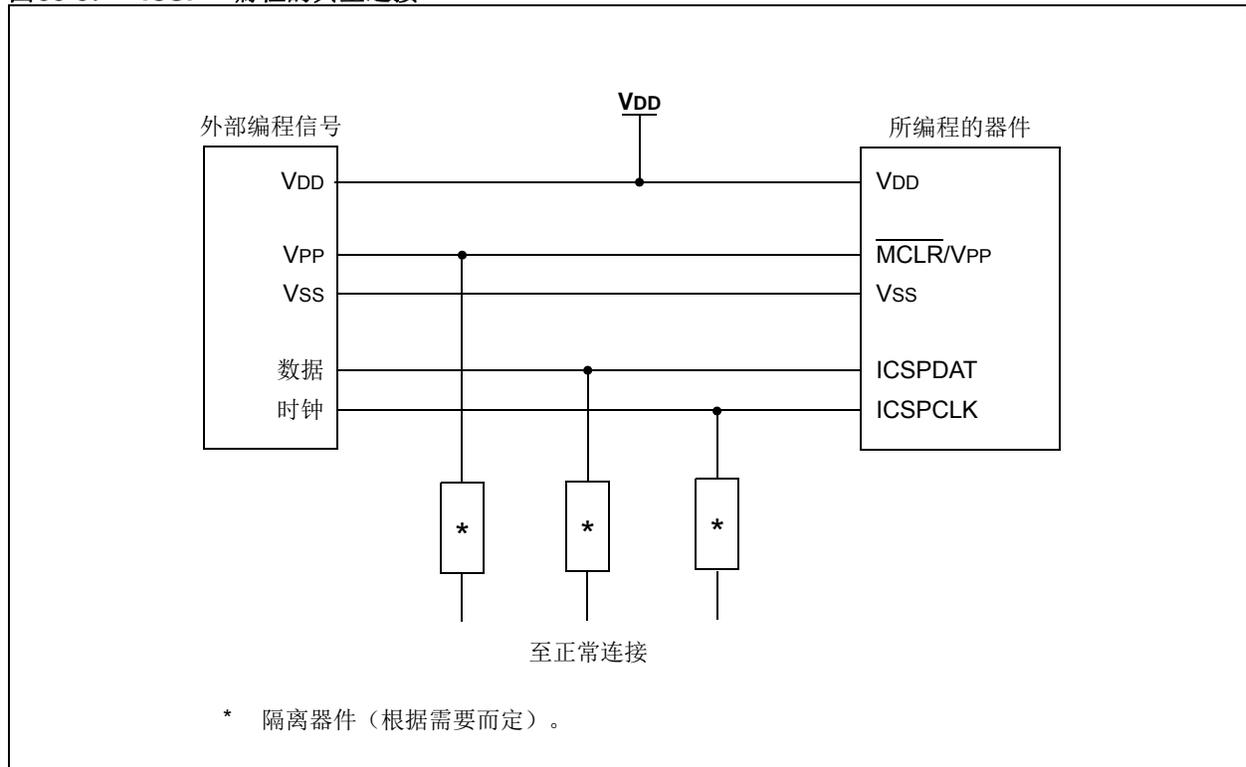


图 33-3: ICSP™ 编程的典型连接



PIC16(L)F18324/18344

34.0 指令集汇总

每条指令都是一个包含操作码和所有必需操作数的14位字。操作码可以分为三大类。

- 针对字节的操作类指令
- 针对位的操作类指令
- 立即数和控制操作类指令

立即数和控制类指令字格式最为丰富。

表34-3列出了MPASM™汇编器可识别的指令。

除了以下指令（可能需要2或3个周期），所有指令都在单个指令周期内执行：

- 子程序进入需要两个周期（CALL和CALLW）
- 中断返回或子程序返回指令需要两个周期（RETURN、RETLW和RETFIE）
- 程序转移指令需要两个周期（GOTO、BRA、BRW、BTFSS、BTFSC、DECFSZ和INCSFZ）
- 当任意指令引用某个间接文件寄存器，并且文件选择寄存器指向程序存储器时，将需要使用一个额外的指令周期。

一个指令周期包含4个振荡器周期；振荡器频率为4 MHz时，得到的标称指令执行速率为1 MHz。

所有指令示例均使用格式“0xhh”来表示一个十六进制数，其中“h”表示一个十六进制位。

34.1 读-修改-写操作

任何一条指定文件寄存器作为指令一部分的写指令都进行读-修改-写（Read-Modify-Write, R-M-W）操作。读寄存器、修改数据，并根据目标标识符“d”的状态，存储结果到工作（W）寄存器或原始文件寄存器（更多信息请参见表34-1）。即使指令写入该寄存器，还是会执行对寄存器的读操作。

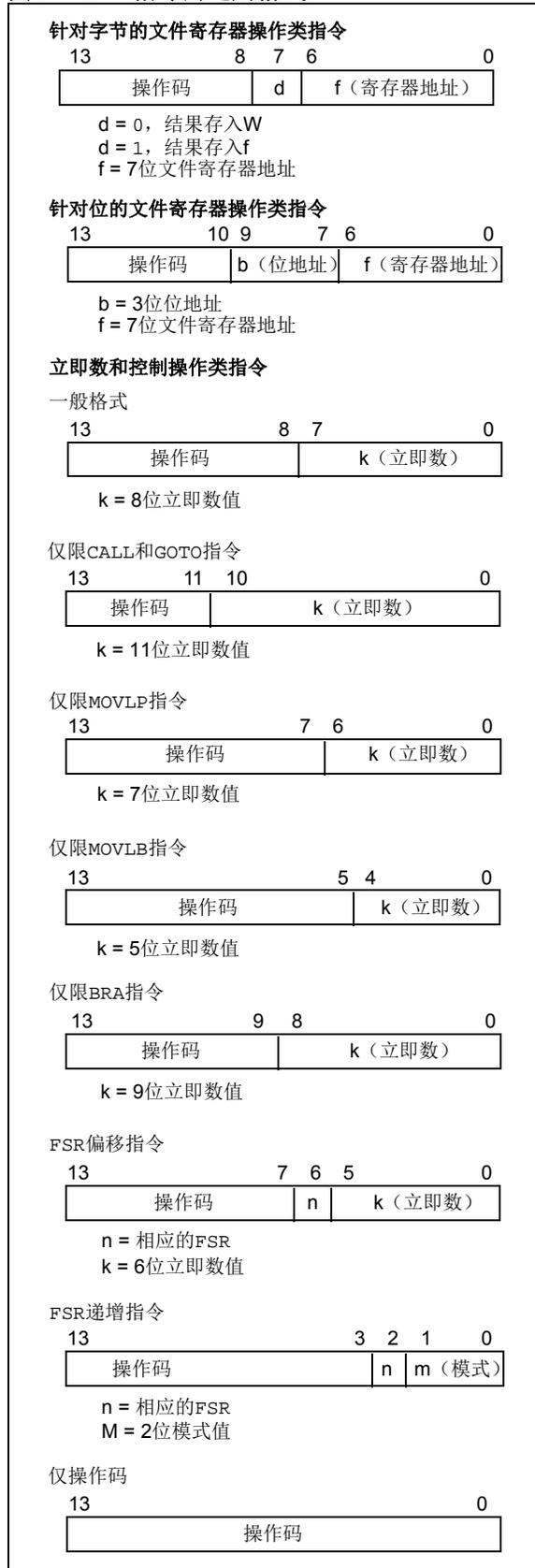
表34-1： 操作码字段说明

字段	说明
f	文件寄存器地址（0x00至0x7F）
W	工作寄存器（累加器）
b	8位文件寄存器内的位地址
k	立即数字段、常数或标号
x	无关位（= 0或1）。 汇编器将生成x = 0的代码。为了与所有的Microchip软件工具兼容，建议使用这种形式。
d	目标寄存器选择；d = 0：结果存入W， d = 1：结果存入文件寄存器f。
n	FSR或INDF编号。（0-1）
mm	预/后递增/递减模式选择

表34-2： 缩写说明

字段	说明
PC	程序计数器
TO	超时位
C	进位位
DC	半进位位
Z	全零标志位
PD	掉电位

图34-1: 指令的通用格式



PIC16(L)F18324/18344

表34-3: PIC16(L)F18324/18344指令集

助记符, 操作数	说明	周期数	14位操作码				受影响的状态	注	
			MSb	LSb					
针对字节的文件寄存器操作类指令									
ADDWF	f, d	W与f相加	1	00	0111	dfff	ffff	C、DC和Z	2
ADDWFC	f, d	W与f相加 (带进位)	1	11	1101	dfff	ffff	C、DC和Z	2
ANDWF	f, d	W与f作逻辑与运算	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	算术右移	1	11	0111	dfff	ffff	C和Z	2
LSLF	f, d	逻辑左移	1	11	0101	dfff	ffff	C和Z	2
LSRF	f, d	逻辑右移	1	11	0110	dfff	ffff	C和Z	2
CLRF	f	将f清零	1	00	0001	1fff	ffff	Z	2
CLRW	-	将W清零	1	00	0001	0000	00xx	Z	2
COMF	f, d	对f取反	1	00	1001	dfff	ffff	Z	2
DECF	f, d	f递减1	1	00	0011	dfff	ffff	Z	2
INCF	f, d	f递增1	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	W与f作逻辑或运算	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	传送f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	将W的内容传送到f	1	00	0000	1fff	ffff		2
RLF	f, d	f带进位循环左移	1	00	1101	dfff	ffff	C	2
RRF	f, d	f带进位循环右移	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	f减去W	1	00	0010	dfff	ffff	C、DC和Z	2
SUBWFB	f, d	f减去W (带借位)	1	11	1011	dfff	ffff	C、DC和Z	2
SWAPF	f, d	将f中的两个半字节进行交换	1	00	1110	dfff	ffff		2
XORWF	f, d	W与f作逻辑异或运算	1	00	0110	dfff	ffff	Z	2
针对字节的跳过操作类指令									
DECFSZ	f, d	f递减1, 为0则跳过	1 (2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	f递增1, 为0则跳过	1 (2)	00	1111	dfff	ffff		1, 2
针对位的文件寄存器操作类指令									
BCF	f, b	将f中的某位清零	1	01	00bb	bfff	ffff		2
BSF	f, b	将f中的某位置1	1	01	01bb	bfff	ffff		2
针对位的跳过操作类指令									
BTFSC	f, b	测试f中的某位, 为0则跳过	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	测试f中的某位, 为1则跳过	1 (2)	01	11bb	bfff	ffff		1, 2
立即数操作类指令									
ADDLW	k	立即数与W相加	1	11	1110	kkkk	kkkk	C、DC和Z	
ANDLW	k	立即数与W作逻辑与运算	1	11	1001	kkkk	kkkk	Z	
IORLW	k	立即数与W作逻辑或运算	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	将立即数送入BSR	1	00	0000	001k	kkkk		
MOVLP	k	将立即数送入PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	将立即数传送到W	1	11	0000	kkkk	kkkk		
SUBLW	k	立即数减去W	1	11	1100	kkkk	kkkk	C、DC和Z	
XORLW	k	立即数与W作逻辑异或运算	1	11	1010	kkkk	kkkk	Z	
控制操作类指令									
BRA	k	相对转移	2	11	001k	kkkk	kkkk		
BRW	-	使用W进行相对转移	2	00	0000	0000	1011		
CALL	k	调用子程序	2	10	0kkk	kkkk	kkkk		
CALLW	-	使用W调用子程序	2	00	0000	0000	1010		
GOTO	k	跳转到地址	2	10	1kkk	kkkk	kkkk		
RETFIE	k	从中断返回	2	00	0000	0000	1001		
RETLW	k	返回并将立即数送入W	2	11	0100	kkkk	kkkk		
RETURN	-	从子程序返回	2	00	0000	0000	1000		

注 1: 如果程序计数器 (PC) 被修改或条件测试结果为真, 则该指令需要两个周期。第二个周期执行一条NOP指令。

2: 如果该指令寻址的是INDF寄存器, 并且相应FSR的MSb置1, 则该指令将需要一个额外的指令周期。

3: MOVW和MOVWI指令的详细说明, 请参见第34.2节“指令说明”。

PIC16(L)F18324/18344

表 34-3: PIC16(L)F18324/18344 指令集 (续)

助记符, 操作数	说明	周期数	14位操作码				受影响的状态	注	
			MSb	LSb					
固有操作类指令									
CLRWDT	-	将看门狗定时器清零	1	00	0000	0110	0100	\overline{TO} 和 \overline{PD}	
NOP	-	空操作	1	00	0000	0000	0000		
RESET	-	软件器件复位	1	00	0000	0000	0001		
SLEEP	-	进入待机模式	1	00	0000	0110	0011	\overline{TO} 和 \overline{PD}	
TRIS	f	将W的内容装入TRIS寄存器	1	00	0000	0110	0fff		
针对C编译器优化的指令									
ADDFSR	n, k	将立即数k加到FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	将间接寄存器FSRn传送到W, 带有预/后递增/递减修改量mm	1	00	0000	0001	0nmm	Z	2, 3
	k[n]	将INDFn送入W, 间接变址寻址	1	11	1111	0nkk	kkkk	Z	2
MOVWI	n mm	将W传送到间接寄存器FSRn, 带有预/后递增/递减修改量mm	1	00	0000	0001	lnmm		2, 3
	k[n]	将W送入INDFn, 变址间接寻址	1	11	1111	lnkk	kkkk		2

- 注 1: 如果程序计数器 (PC) 被修改或条件测试结果为真, 则该指令需要两个周期。第二个周期执行一条NOP指令。
 注 2: 如果该指令寻址的是INDF寄存器, 并且相应FSR的MSb置1, 则该指令将需要一个额外的指令周期。
 注 3: MOVIW和MOVWI指令的详细说明, 请参见第34.2节“指令说明”。

PIC16(L)F18324/18344

34.2 指令说明

ADDFSR	将立即数加到FSRn
语法:	[标号] ADDFSR FSRn, k
操作数:	$-32 \leq k \leq 31$ $n \in [0, 1]$
操作:	$FSR(n) + k \rightarrow FSR(n)$
受影响的状态位:	无
说明:	将有符号6位立即数k加到FSRnH:FSRnL寄存器对的内容。 FSRn地址范围限制为0000h-FFFFh。地址超出该边界时，FSR会发生折回。

ADDLW	立即数与W相加
语法:	[标号] ADDLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) + k \rightarrow (W)$
受影响的状态位:	C、DC和Z
说明:	将W寄存器的内容与8位立即数k相加，结果存入W寄存器。

ADDWF	W与f相加
语法:	[标号] ADDWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) + (f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	C、DC和Z
说明:	将W寄存器的内容与寄存器f的内容相加。如果d为0，结果存入W寄存器。如果d为1，结果存回寄存器f。

ADDWFC	W与f相加（带进位）
语法:	[标号] ADDWFC f{,d}
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) + (f) + (C) \rightarrow \text{dest}$
受影响的状态位:	C、DC和Z
说明:	将W的内容、进位标志位与数据存储单元f的内容相加。如果d为0，结果放入W寄存器。如果d为1，结果放入数据存储单元f。

ANDLW	立即数与W作逻辑与运算
语法:	[标号] ANDLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) .AND.(k) \rightarrow (W)$
受影响的状态位:	Z
说明:	将W寄存器的内容与8位立即数k进行逻辑与运算。结果存入W寄存器。

ANDWF	W与f作逻辑与运算
语法:	[标号] ANDWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) .AND.(f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将W寄存器的内容与寄存器f的内容进行逻辑与运算。如果d为0，结果存入W寄存器。如果d为1，结果存回寄存器f。

ASRF	算术右移
语法:	[标号] ASRF f{,d}
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f<7>) \rightarrow \text{dest}<7>$ $(f<7:1>) \rightarrow \text{dest}<6:0>$ $(f<0>) \rightarrow C$
受影响的状态位:	C和Z
说明:	将寄存器f的内容连同进位标志位一起右移1位。MSb保持不变。如果d为0，结果放入W寄存器。如果d为1，结果存回寄存器f。



BCF 将f中的某位清零

语法: [标号] BCF f,b
 操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
 操作: $0 \rightarrow (f)$
 受影响的状态位: 无
 说明: 将寄存器f中的bit b清零。

BTFSC 测试f中的某位, 为0则跳过

语法: [标号] BTFSC f,b
 操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
 操作: 如果 $(f) = 0$, 则跳过
 受影响的状态位: 无
 说明: 如果寄存器f的bit b为1, 则执行下一条指令。
 如果寄存器f的bit b为0, 则丢弃下一条指令, 代之执行一条NOP指令, 使之成为一条双周期指令。

BRA 相对转移

语法: [标号] BRA label
 [标号] BRA \$+k
 操作数: $-256 \leq \text{label} - \text{PC} + 1 \leq 255$
 $-256 \leq k \leq 255$
 操作: $(\text{PC}) + 1 + k \rightarrow \text{PC}$
 受影响的状态位: 无
 说明: 将有符号9位立即数k与PC相加。由于PC将递增以便取出下一条指令, 所以新地址将为 $\text{PC} + 1 + k$ 。该指令为一条双周期指令。该转移的地址范围存在限制。

BTFSS 测试f中的某位, 为1则跳过

语法: [标号] BTFSS f,b
 操作数: $0 \leq f \leq 127$
 $0 \leq b < 7$
 操作: 如果 $(f) = 1$, 则跳过
 受影响的状态位: 无
 说明: 如果寄存器f的bit b为0, 则执行下一条指令。
 如果bit b为1, 则丢弃下一条指令, 代之执行一条NOP指令, 使之成为一条双周期指令。

BRW 使用W进行相对转移

语法: [标号] BRW
 操作数: 无
 操作: $(\text{PC}) + (\text{W}) \rightarrow \text{PC}$
 受影响的状态位: 无
 说明: 将W的内容(无符号)加到PC。由于PC将递增以便取出下一条指令, 所以新地址将为 $\text{PC} + 1 + (\text{W})$ 。该指令为一条双周期指令。

BSF 将f中的某位置1

语法: [标号] BSF f,b
 操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
 操作: $1 \rightarrow (f)$
 受影响的状态位: 无
 说明: 将寄存器f中的bit b置1。

PIC16(L)F18324/18344

CALL	调用子程序
语法:	[标号] CALL k
操作数:	$0 \leq k \leq 2047$
操作:	(PC) + 1 → TOS, k → PC<10:0>, (PCLATH<6:3>) → PC<14:11>
受影响的状态位:	无
说明:	调用子程序。首先, 将返回地址 (PC + 1) 压入堆栈。11位直接地址被装入PC的bit <10:0>。PC的最高4位从PCLATH装入。CALL是一条双周期指令。

CALLW	使用W调用子程序
语法:	[标号] CALLW
操作数:	无
操作:	(PC) + 1 → TOS, (W) → PC<7:0>, (PCLATH<6:0>) → PC<14:8>
受影响的状态位:	无
说明:	使用W调用子程序。首先, 将返回地址 (PC + 1) 压入返回堆栈。然后, W的内容被装入PC<7:0>, PCLATH的内容被装入PC<14:8>。CALLW是一条双周期指令。

CLRF	将f清零
语法:	[标号] CLRF f
操作数:	$0 \leq f \leq 127$
操作:	00h → (f) 1 → Z
受影响的状态位:	Z
说明:	寄存器f的内容被清零, 并且Z位被置1。

CLRW	将W清零
语法:	[标号] CLRW
操作数:	无
操作:	00h → (W) 1 → Z
受影响的状态位:	Z
说明:	W寄存器被清零。全零标志位 (Z) 被置1。

CLRWDT	将看门狗定时器清零
语法:	[标号] CLRWDT
操作数:	无
操作:	00h → WDT 0 → WDT分频器, 1 → \overline{TO} 1 → \overline{PD}
受影响的状态位:	\overline{TO} 和 \overline{PD}
说明:	CLRWDT指令复位看门狗定时器及其预分频器。状态位 \overline{TO} 和 \overline{PD} 均被置1。

COMF	对f取反
语法:	[标号] COMF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$\overline{(f)} \rightarrow$ (目标寄存器)
受影响的状态位:	Z
说明:	将寄存器f的内容取反。如果d为0, 结果存入W寄存器。如果d为1, 结果存回寄存器f。

DECF	F递减1
语法:	[标号] DECF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	(f) - 1 → (目标寄存器)
受影响的状态位:	Z
说明:	将寄存器f的内容递减1。如果d为0, 结果存入W寄存器。如果d为1, 结果存回寄存器f。

DECFSZ f 递减 1, 为 0 则跳过

语法: [标号] DECFSZ f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$

操作: $(f) - 1 \rightarrow$ (目标寄存器);
 如果结果 = 0, 则跳过

受影响的状态位: 无

说明: 将寄存器 f 的内容递减 1。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。
 如果结果为 1, 则执行下一条指令。如果结果为 0, 代之执行一条 NOP 指令, 使之成为一条双周期指令。

INCFSZ f 递增 1, 为 0 则跳过

语法: [标号] INCFSZ f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$

操作: $(f) + 1 \rightarrow$ (目标寄存器),
 如果结果 = 0, 则跳过

受影响的状态位: 无

说明: 将寄存器 f 的内容递增 1。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。
 如果结果为 1, 则执行下一条指令。如果结果为 0, 代之执行一条 NOP 指令, 使之成为一条双周期指令。

GOTO 无条件转移

语法: [标号] GOTO k
 操作数: $0 \leq k \leq 2047$

操作: $k \rightarrow PC<10:0>$
 $PCLATH<6:3> \rightarrow PC<14:11>$

受影响的状态位: 无

说明: GOTO 是一条无条件转移指令。11 位立即数值被装入 PC 的 bit <10:0>。PC 的最高 4 位从 PCLATH<4:3> 装入。GOTO 是一条双周期指令。

IORLW 立即数与 W 作逻辑或运算

语法: [标号] IORLW k
 操作数: $0 \leq k \leq 255$

操作: $(W) .OR. k \rightarrow (W)$

受影响的状态位: Z

说明: 将 W 寄存器的内容与 8 位立即数 k 进行逻辑或运算。结果存入 W 寄存器。

INCF F 递增 1

语法: [标号] INCF f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$

操作: $(f) + 1 \rightarrow$ (目标寄存器)

受影响的状态位: Z

说明: 将寄存器 f 的内容递增 1。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

IORWF W 与 f 作逻辑或运算

语法: [标号] IORWF f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$

操作: $(W) .OR. (f) \rightarrow$ (目标寄存器)

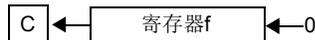
受影响的状态位: Z

说明: 将 W 寄存器的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

PIC16(L)F18324/18344

LSLF 逻辑左移

语法: [标号] LSLF f{,d}
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f<7>) \rightarrow C$
 $(f<6:0>) \rightarrow \text{dest}<7:1>$
 $0 \rightarrow \text{dest}<0>$
受影响的状态位: C和Z
说明: 将寄存器f的内容连同进位标志位一起左移1位。0移入LSb。如果d为0, 结果放入W寄存器。如果d为1, 结果存回寄存器f。



LSRF 逻辑右移

语法: [标号] LSRF f{,d}
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $0 \rightarrow \text{dest}<7>$
 $(f<7:1>) \rightarrow \text{dest}<6:0>$,
 $(f<0>) \rightarrow C$
受影响的状态位: C和Z
说明: 将寄存器f的内容连同进位标志位一起右移1位。0移入MSb。如果d为0, 结果放入W寄存器。如果d为1, 结果存回寄存器f。



MOVF 传送f

语法: [标号] MOVF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) \rightarrow (\text{目标寄存器})$
受影响的状态位: Z
说明: 根据d的状态, 将寄存器f的内容传送到目标寄存器。如果d = 0, 目标寄存器为W寄存器。如果d = 1, 目标寄存器为文件寄存器f本身。由于状态标志位Z受影响, 可用d = 1对文件寄存器内容进行检测。

指令字数: 1
指令周期数: 1
示例: MOVF FSR, 0

执行指令后
 $W = \text{FSR寄存器的值}$
 $Z = 1$

MOVIW 将INDFn的内容传送到W

语法: [标号] MOVIW ++FSRn
[标号] MOVIW --FSRn
[标号] MOVIW FSRn++
[标号] MOVIW FSRn--
[标号] MOVIW k[FSRn]

操作数: $n \in [0,1]$
 $mm \in [00,01, 10, 11]$
 $-32 \leq k \leq 31$

操作: $INDFn \rightarrow W$
有效地址通过以下方式确定

- FSR + 1 (预递增)
- FSR - 1 (预递减)
- FSR + k (相对偏移)

在传送之后, FSR值将为以下之一:

- FSR + 1 (全部递增)
- FSR - 1 (全部递减)
- 不变

受影响的状态位: Z

模式	语法	mm
预递增	++FSRn	00
预递减	--FSRn	01
后递增	FSRn++	10
后递减	FSRn--	11

说明: 该指令用于在W和一个间接寄存器(INDFn)之间传送数据。在该传送操作之前/之后, 将通过预/后递增/递减指针(FSRn)来更新指针。

注: INDFn寄存器不是物理寄存器。访问INDFn寄存器的所有指令实际上访问的是由FSRn指定的地址处的寄存器。

FSRn地址范围限制为0000h-FFFFh。地址递增/递减到超出这些边界时, 将导致它发生折回。

MOVLB 将立即数送入BSR

语法: [标号] MOVLB k

操作数: $0 \leq k \leq 31$

操作: $k \rightarrow BSR$

受影响的状态位: 无

说明: 将5位立即数k装入存储区选择寄存器(BSR)。

MOVLP 将立即数送入PCLATH

语法: [标号] MOVLP k

操作数: $0 \leq k \leq 127$

操作: $k \rightarrow PCLATH$

受影响的状态位: 无

说明: 将7位立即数k装入PCLATH寄存器。

MOVLW 将立即数传送到W

语法: [标号] MOVLW k

操作数: $0 \leq k \leq 255$

操作: $k \rightarrow (W)$

受影响的状态位: 无

说明: 将8位立即数k装入W寄存器。其余无关位均汇编为0。

指令字数: 1

指令周期数: 1

示例: `MOVLW 0x5A`
执行指令后
 $W = 0x5A$

MOVWF 将W的内容传送到f

语法: [标号] MOVWF f

操作数: $0 \leq f \leq 127$

操作: $(W) \rightarrow (f)$

受影响的状态位: 无

说明: 将W寄存器的数据传送到寄存器f。

指令字数: 1

指令周期数: 1

示例: `MOVWF OPTION_REG`
执行指令前
 $OPTION_REG = 0xFF$
 $W = 0x4F$
执行指令后
 $OPTION_REG = 0x4F$
 $W = 0x4F$

PIC16(L)F18324/18344

MOVWI 将W的内容传送到INDFn

语法:	[标号] MOVWI ++FSRn [标号] MOVWI --FSRn [标号] MOVWI FSRn++ [标号] MOVWI FSRn-- [标号] MOVWI k[FSRn]
操作数:	$n \in [0,1]$ $mm \in [00,01, 10, 11]$ $-32 \leq k \leq 31$
操作:	$W \rightarrow \text{INDFn}$ 有效地址通过以下方式确定 <ul style="list-style-type: none"> • FSR + 1 (预递增) • FSR - 1 (预递减) • FSR + k (相对偏移) 在传送之后, FSR值将为以下之一: <ul style="list-style-type: none"> • FSR + 1 (全部递增) • FSR - 1 (全部递减) 不变
受影响的状态位:	无

模式	语法	mm
预递增	++FSRn	00
预递减	--FSRn	01
后递增	FSRn++	10
后递减	FSRn--	11

说明: 该指令用于在W和一个间接寄存器 (INDFn) 之间传送数据。在该传送操作之前/之后, 将通过预/后递增/递减指针 (FSRn) 来更新指针。

注: INDFn 寄存器不是物理寄存器。访问 INDFn 寄存器的所有指令实际上访问的是由 FSRn 指定的地址处的寄存器。

FSRn 地址范围限制为 0000h-FFFFh。地址递增/递减到超出这些边界时, 将导致它发生折回。

对于 FSRn 的递增/递减操作不会影响任何状态位。

NOP 空操作

语法:	[标号] NOP
操作数:	无
操作:	空操作
受影响的状态位:	无
说明:	不执行任何操作。
指令字数:	1
指令周期数:	1
示例:	NOP

RESET 软件复位

语法:	[标号] RESET
操作数:	无
操作:	执行器件复位。复位PCON寄存器的 $\overline{\text{RI}}$ 标志。
受影响的状态位:	无
说明:	此指令可实现用软件执行硬件复位。

RETFIE 从中断返回

语法:	[标号] RETFIE k
操作数:	无
操作:	TOS \rightarrow PC, 1 \rightarrow GIE
受影响的状态位:	无
说明:	从中断返回。执行出栈操作, 将栈顶 (Top-of-Stack, TOS) 的内容装入 PC。通过将全局中断允许位 GIE (INTCON<7>) 置 1, 来允许中断。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	RETFIE
中断后	PC = TOS GIE = 1

RETLW	返回并将立即数送入W
语法:	[标号] RETLW k
操作数:	$0 \leq k \leq 255$
操作:	$k \rightarrow (W)$; $TOS \rightarrow PC$
受影响的状态位:	无
说明:	将8位立即数k装入W寄存器。将栈顶内容(返回地址)装入程序计数器。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	CALL TABLE;W contains table ;offset value ;W now has table value . . ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; . . . RETLW kn ; End of table
	TABLE
	执行指令前 W = 0x07 执行指令后 W = k8的值

RETURN	从子程序返回
语法:	[标号] RETURN
操作数:	无
操作:	$TOS \rightarrow PC$
受影响的状态位:	无
说明:	从子程序返回。执行出栈操作, 将栈顶(TOS)内容装入程序计数器。这是一条双周期指令。

RLF	f带进位循环左移
语法:	[标号] RLF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	参见如下说明
受影响的状态位:	C
说明:	将寄存器f的内容连同进位标志位一起循环左移1位。如果d为0, 结果存入W寄存器。如果d为1, 结果存回寄存器f。
	
指令字数:	1
指令周期数:	1
示例:	RLF REG1,0
	执行指令前 REG1 = 1110 0110 C = 0
	执行指令后 REG1 = 1110 0110 W = 1100 1100 C = 1

RRF	f带进位循环右移
语法:	[标号] RRF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	参见如下说明
受影响的状态位:	C
说明:	将寄存器f的内容连同进位标志位一起循环右移1位。如果d为0, 结果存入W寄存器。如果d为1, 结果存回寄存器f。
	

PIC16(L)F18324/18344

SLEEP 进入休眠模式

语法: [标号] SLEEP
操作数: 无
操作: 00h → WDT,
0 → WDT分频器,
1 → \overline{TO} ,
0 → PD
受影响的状态位: \overline{TO} 和PD
说明: 掉电状态位 \overline{PD} 被清零。超时状态位 \overline{TO} 被置1。看门狗定时器及其预分频器被清零。
更多信息, 请参见第9.3节“休眠模式”。

SUBLW 立即数减去W

语法: [标号] SUBLW k
操作数: $0 \leq k \leq 255$
操作: $k - (W) \rightarrow (W)$
受影响的状态位: C、DC和Z
说明: 用8位立即数k减去W寄存器的内容(通过二进制补码方式进行运算)。结果存入W寄存器。

C = 0	$W > k$
C = 1	$W \leq k$
DC = 0	$W\langle 3:0 \rangle > k\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq k\langle 3:0 \rangle$

SUBWF f减去W

语法: [标号] SUBWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) - (W) \rightarrow (\text{目标寄存器})$
受影响的状态位: C、DC和Z
说明: 用寄存器f的内容减去W寄存器的内容(通过二进制补码方式进行运算)。如果d为0, 结果存入W寄存器。如果d为1, 结果存回寄存器f。

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W\langle 3:0 \rangle > f\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq f\langle 3:0 \rangle$

SUBWFB f减去W(带借位)

语法: SUBWFB f{,d}
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) - (W) - (\overline{B}) \rightarrow \text{dest}$
受影响的状态位: C、DC和Z
说明: 用寄存器f的内容减去W的内容和借位标志(进位)(通过二进制补码方式进行运算)。如果d为0, 结果存入W。如果d为1, 结果存回寄存器f。

SWAPF 将f中的两个半字节进行交换

语法: [标号] SWAPF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f\langle 3:0 \rangle) \rightarrow (\text{目标寄存器}\langle 7:4 \rangle)$,
 $(f\langle 7:4 \rangle) \rightarrow (\text{目标寄存器}\langle 3:0 \rangle)$
受影响的状态位: 无
说明: 寄存器f的高半字节和低半字节相互交换。如果d为0, 结果存入W寄存器。如果d为1, 结果存入寄存器f。

TRIS 将W的内容装入TRIS寄存器

语法: [标号] TRIS f
操作数: $5 \leq f \leq 7$
操作: (W) → TRIS 寄存器 f
受影响的状态位: 无
说明: 将W寄存器的数据传送到TRIS寄存器。
 当f = 5时, 装入TRISA。
 当f = 6时, 装入TRISB。
 当f = 7时, 装入TRISC。

XORLW 立即数与W作逻辑异或运算

语法: [标号] XORLW k
操作数: $0 \leq k \leq 255$
操作: (W) .XOR. k → (W)
受影响的状态位: Z
说明: 将W寄存器的内容与8位立即数k进行逻辑异或运算。结果存入W寄存器。

XORWF W与f作逻辑异或运算

语法: [标号] XORWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: (W) .XOR.(f) → (目标寄存器)
受影响的状态位: Z
说明: 将W寄存器的内容与寄存器f的内容进行逻辑异或运算。如果d为0, 结果存入W寄存器。如果d为1, 结果存回寄存器f。

PIC16(L)F18324/18344

35.0 电气规范

35.1 绝对最大值^(†)

偏置时的环境温度	-40°C 至 +125°C
存储温度	-65°C 至 +150°C
引脚相对于V _{SS} 的电压	
V _{DD} 引脚	
PIC16F18324/18344	-0.3V 至 +6.5V
PIC16LF18324/18344	-0.3V 至 +4.0V
MCLR引脚	-0.3V 至 +9.0V
所有其他引脚	-0.3V 至 (V _{DD} + 0.3V)
最大电流	
V _{SS} 引脚 ⁽¹⁾	
-40°C ≤ T _A ≤ +85°C	250 mA
+85°C < T _A ≤ +125°C	85 mA
V _{DD} 引脚 ⁽¹⁾	
-40°C ≤ T _A ≤ +85°C	250 mA
+85°C < T _A ≤ +125°C	85 mA
任意 I/O 引脚	±50 mA
钳位电流, I _K (V _{PIN} < 0 或 V _{PIN} > V _{DD})	±20 mA
总功耗 ⁽²⁾	800 mW

注 1: 最大电流值要求 I/O 引脚上具有均匀的负载分布。最大电流可以通过器件封装功率耗散特性进行限制, 请参见 [表 35-3](#) 来计算器件规范值。

2: 功耗按如下公式计算:

$$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

†注: 如果器件工作条件超过上述“绝对最大值”, 可能对器件造成永久性损坏。上述值仅为运行条件极大值, 我们建议不要使器件在或超过该规范规定的条件下运行。器件长时间工作在最大值条件下, 其可靠性会受到影响。

35.2 标准工作条件

所有器件的标准工作条件定义如下：

工作电压 $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$

工作温度: $T_{A_MIN} \leq T_A \leq T_{A_MAX}$

V_{DD}——工作电源电压范围⁽¹⁾

PIC16LF18324/18344

V_{DDMIN} (F_{osc} ≤ 16 MHz) +1.8V

V_{DDMIN} (F_{osc} ≤ 32 MHz) +2.5V

V_{DDMAX} +3.6V

PIC16F18324/18344

V_{DDMIN} (F_{osc} ≤ 16 MHz) +2.3V

V_{DDMIN} (F_{osc} ≤ 32 MHz) +2.5V

V_{DDMAX} +5.5V

T_A——工作环境温度范围

工业级温度

T_{A_MIN} -40°C

T_{A_MAX} +85°C

扩展级温度

T_{A_MIN} -40°C

T_{A_MAX} +125°C

注 1: 请参见参数 D002，直流特性：供电电压。

PIC16(L)F18324/18344

图35-1: 电压—频率关系图, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$, 仅限 PIC16F18324/18344

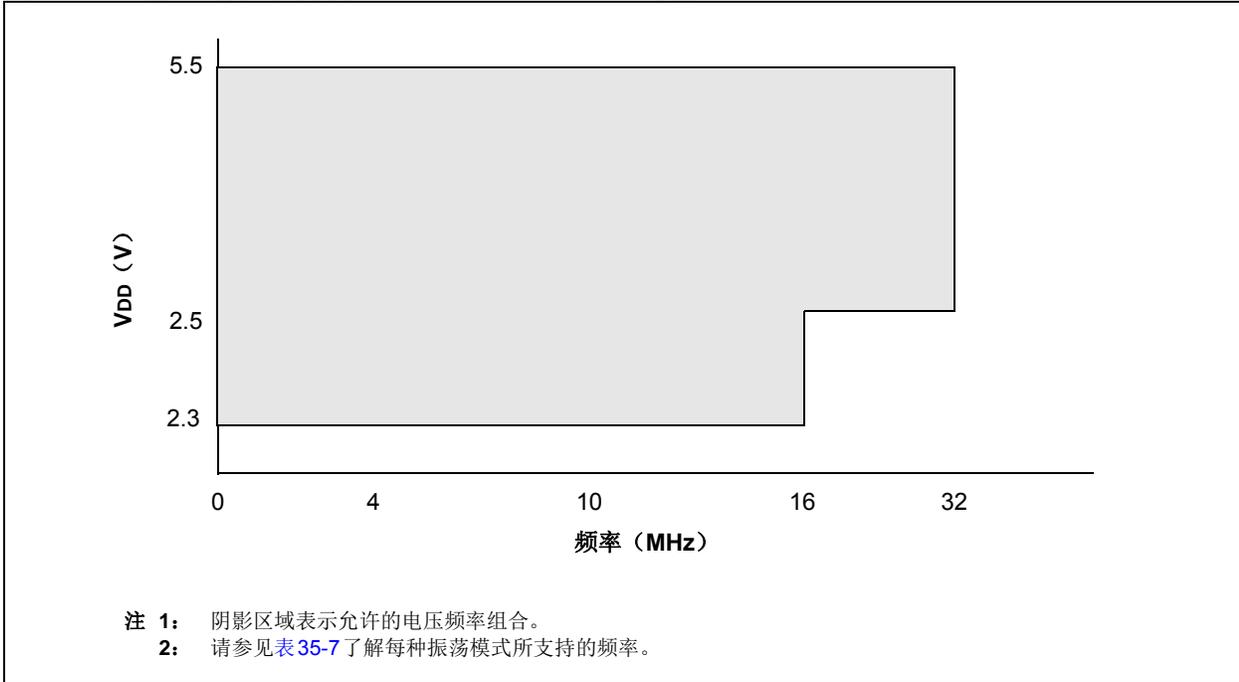
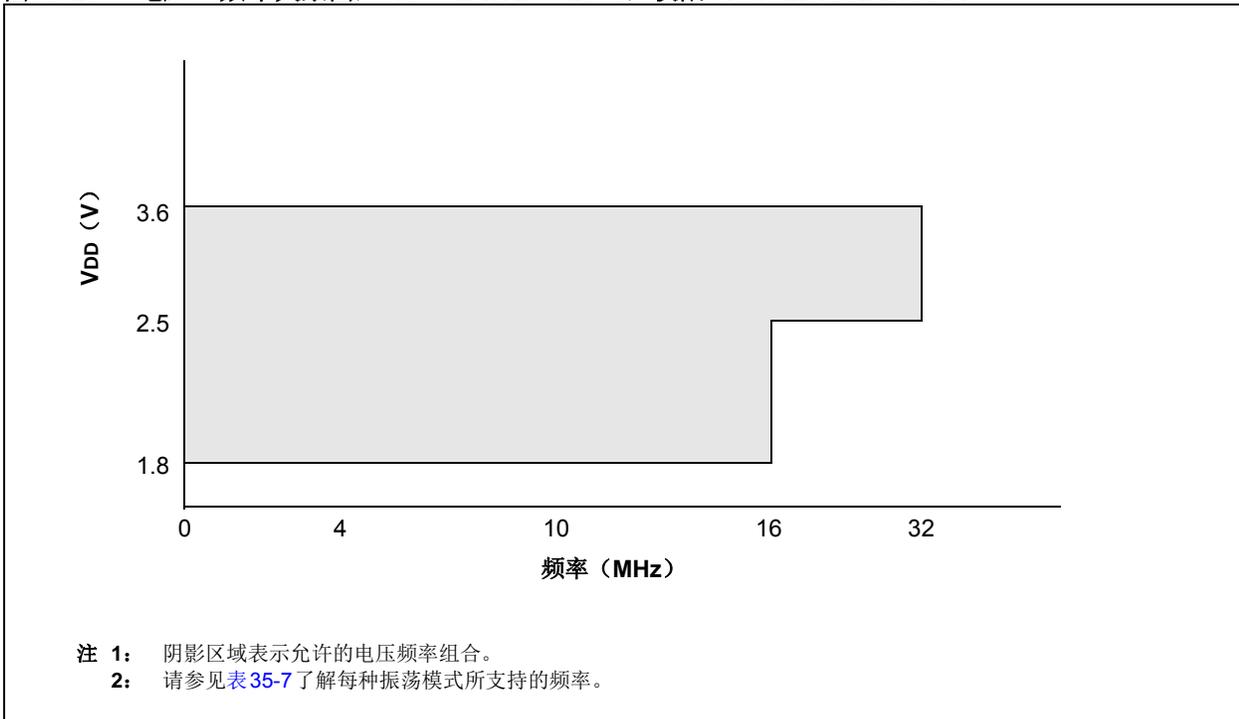


图35-2: 电压—频率关系图, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$, 仅限 PIC16LF18324/18344



35.3 直流特性

表 35-1: 电源电压

PIC16LF18324/18344		标准工作条件 (除非另外声明)					
PIC16F18324/18344		标准工作条件 (除非另外声明)					
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
电源电压							
D002	VDD		1.8	—	3.6	V	FOSC ≤ 16 MHz
			2.5	—	3.6	V	FOSC > 16 MHz
D002	VDD		2.3	—	5.5	V	FOSC ≤ 16 MHz:
			2.5	—	5.5	V	FOSC > 16 MHz
RAM 数据保持电压⁽¹⁾							
D003	VDR		1.5	—	—	V	器件处于休眠模式
D003	VDR		1.7	—	—	V	器件处于休眠模式
上电复位释放电压⁽²⁾							
D004	VPOR		—	1.6	—	V	禁止 BOR 和 LPBOR ⁽³⁾
D004	VPOR		—	1.6	—	V	禁止 BOR 和 LPBOR ⁽³⁾
上电复位重新激活电压⁽²⁾							
D005	VPORR		—	0.8	—	V	禁止 BOR 和 LPBOR ⁽³⁾
D005	VPORR		—	1.5	—	V	禁止 BOR 和 LPBOR ⁽³⁾
确保内部上电复位信号的 VDD 上升速率⁽²⁾							
D006	SVDD		0.05	—	—	V/ms	禁止 BOR 和 LPBOR ⁽³⁾
D006	SVDD		0.05	—	—	V/ms	禁止 BOR 和 LPBOR ⁽³⁾

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1: 这是在确保不丢失 RAM 数据的前提下，休眠模式下或器件复位期间 VDD 的下限值。

2: 请参见图 35-3。

3: 有关 BOR 和 LPBOR 跳变点的信息，请参见表 35-11。

PIC16(L)F18324/18344

图35-3: VDD缓慢上升时, POR和POR重新激活

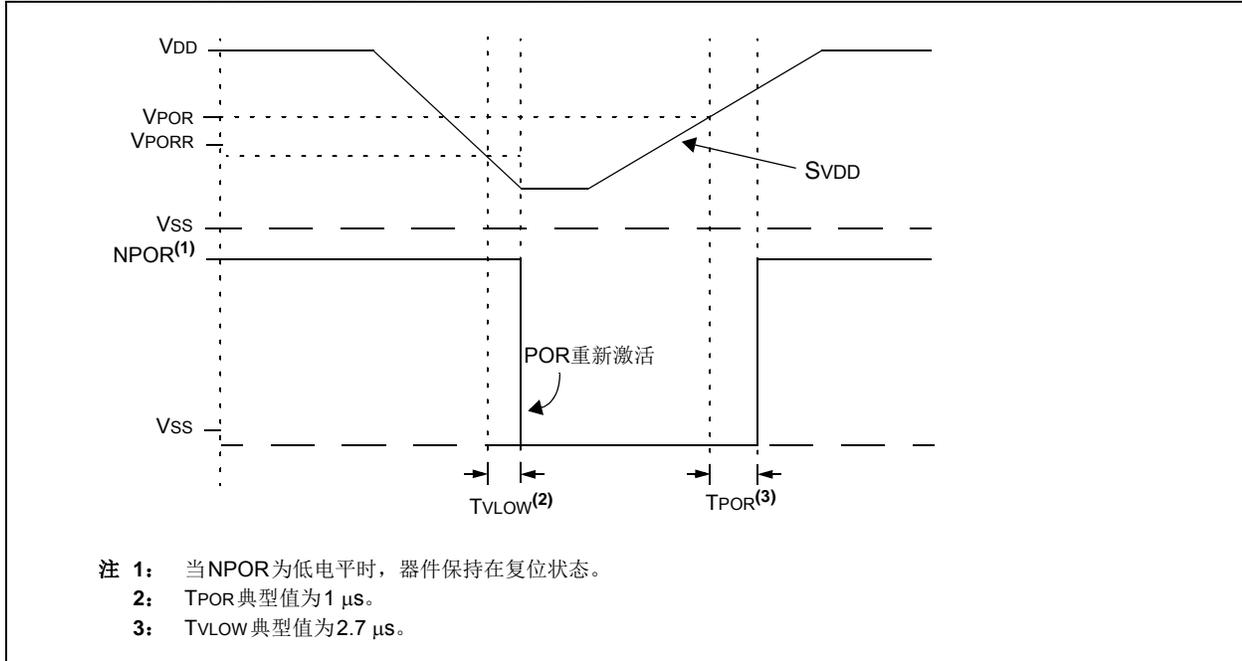


表 35-2: 电源电流 (IDD) (1,2)

PIC16LF18324/18344		标准工作条件 (除非另外声明)						
PIC16F18324/18344		标准工作条件 (除非另外声明)						
参数编号	符号	器件特性	最小值	典型值†	最大值	单位	条件	
							VDD	注
D100	IDDXT4	XT = 4 MHz	—	295	380	μA	3.0V	
D100	IDDXT4	XT = 4 MHz	—	306	400	μA	3.0V	
D101	IDDHFO16	HFINTOSC = 16 MHz	—	1.2	1.5	mA	3.0V	
D101	IDDHFO16	HFINTOSC = 16 MHz	—	1.3	1.6	mA	3.0V	
D102	IDDHFOPLL	HFINTOSC = 32 MHz	—	2.2	2.8	mA	3.0V	
D102	IDDHFOPLL	HFINTOSC = 32 MHz	—	2.3	2.9	mA	3.0V	
D103	IDDHSPLL32	HS+PLL = 32 MHz	—	2.1	2.5	mA	3.0V	
D103	IDDHSPLL32	HS+PLL = 32 MHz	—	2.2	2.6	mA	3.0V	
D104	IDDIDLE	空闲模式, HFINTOSC = 16 MHz	—	720	1000	μA	3.0V	
D104	IDDIDLE	空闲模式, HFINTOSC = 16 MHz	—	727	1100	μA	3.0V	
D105	IDDDOZE ⁽³⁾	打盹模式, HFINTOSC = 16 MHz, 打盹模式时钟分频比 = 16	—	777	—	μA	3.0V	
D105	IDDDOZE ⁽³⁾	打盹模式, HFINTOSC = 16 MHz, 打盹模式时钟分频比 = 16	—	784	—	μA	3.0V	

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 有效工作模式下, 所有IDD测量值的测试条件为: OSC1 = 外部方波, 轨到轨满幅; 所有I/O引脚均为三态, 上拉至VDD; MCLR = VDD; 禁止WDT。

2: 供电电流主要受工作电压和频率的影响。其他因素, 如I/O引脚负载和开关速率、振荡器类型、内部代码执行模式和温度, 也会对电流消耗产生影响。

3: $IDDDOZE = [IDDIDLE \cdot (N-1)/N] + IDDHFO16/N$, 其中N = 打盹模式时钟分频比 (见寄存器9-2)。

PIC16(L)F18324/18344

表35-3: 掉电电流 (IPD) (1,2,3)

PIC16LF18324/18344		标准工作条件 (除非另外声明)							
PIC16F18324/18344		标准工作条件 (除非另外声明) VREGPM = 1							
参数编号	符号	器件特性	最小值	典型值†	最大值 +85°C	最大值 +125°C	单位	条件	
								VDD	注
D200	IPD	基本IPD电流	—	0.03	2	5.2	μA	3.0V	
D200	IPD	基本IPD电流	—	0.3	2.4	5.6	μA	3.0V	
			—	12.8	22	27	μA	3.0V	VREGM = 0
D201	IPD_WDT	低频内部振荡器/ WDT	—	0.4	2.9	6	μA	3.0V	
D201	IPD_WDT	低频内部振荡器/ WDT	—	0.5	3.3	6.6	μA	3.0V	
D202	IPD_SOSC	辅助振荡器 (SOSC)	—	1.3	2.8	6	μA	3.0V	
D202	IPD_SOSC	辅助振荡器 (SOSC)	—	1.5	3.2	6.4	μA	3.0V	
D203	IPD_FVR	FVR	—	45	74	76	μA	3.0V	
D203	IPD_FVR	FVR	—	40	70	75	μA	3.0V	
D204	IPD_BOR	欠压复位 (BOR)	—	10.6	16	19	μA	3.0V	
D204	IPD_BOR	欠压复位 (BOR)	—	10.5	16.4	19.4	μA	3.0V	
D205	IPD_LPBOR	低功耗欠压复位 (LPBOR)	—	0.3	2.5	5.5	μA	3.0V	
D207	IPD_ADCA	ADC——不在进行转换	—	0.3	2	5.2	μA	3.0V	ADC不在进行转换 ⁽⁴⁾
D207	IPD_ADCA	ADC——不在进行转换	—	0.3	2.4	5.6	μA	3.0V	ADC不在进行转换 ⁽⁴⁾
D208	IPD_CMP	比较器	—	30	45	50	μA	3.0V	
D208	IPD_CMP	比较器	—	30	44	49	μA	3.0V	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 外设电流为基本IPD与该外设使能时所额外消耗的电流之和。可通过从该参数值中减去基本IDD或IPD电流, 以确定外设Δ电流。在计算总电流消耗时应使用最大值。

2: 在休眠模式下, 掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有I/O引脚处于高阻态并且连接到VSS时测得的。

3: 如果多个外设可用, 那么列出的所有外设电流均是基于每个外设的。

4: ADC时钟源为ADCR。C。

PIC16(L)F18324/18344

表35-4: I/O端口

直流特性		标准工作条件 (除非另外声明)					
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
D300 D301 D302 D303 D304 D305	V _{IL}	输入低电压					
		I/O 端口:					
		带 TTL 缓冲器	—	—	0.8	V	4.5V ≤ V _{DD} ≤ 5.5V
		带施密特触发器缓冲器	—	—	0.15 V _{DD}	V	1.8V ≤ V _{DD} ≤ 4.5V
		I ² C 电平	—	—	0.3 V _{DD}	V	
		SMBus 电平	—	—	0.8	V	2.7V ≤ V _{DD} ≤ 5.5V
D305		MCLR	—	—	0.2 V _{DD}	V	
D320 D321 D322 D323 D324 D325	V _{IH}	输入高电压					
		I/O 端口:					
		带 TTL 缓冲器	2.0	—	—	V	4.5V ≤ V _{DD} ≤ 5.5V
		带施密特触发器缓冲器	0.25 V _{DD} + 0.8	—	—	V	1.8V ≤ V _{DD} ≤ 4.5V
		I ² C 电平	0.7 V _{DD}	—	—	V	
		SMBus 电平	2.1	—	—	V	2.7V ≤ V _{DD} ≤ 5.5V
D325		MCLR	0.7 V _{DD}	—	—	V	
D340 D341 D342	I _{IL}	输入泄漏电流 ⁽²⁾					
		I/O 端口	—	± 5	± 125	nA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, 85°C
		MCLR ⁽²⁾	—	± 50	± 200	nA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, 85°C
D350	I _{PUR}	弱上拉电流					
			25	120	200	μA	V _{DD} = 3.0V, V _{PIN} = V _{SS}
D360	V _{OL}	输出低电压 ⁽⁴⁾					
		I/O 端口	—	—	0.6	V	I _{OL} = 10.0 mA, V _{DD} = 3.0V
D370	V _{OH}	输出高电压 ⁽⁴⁾					
		I/O 端口	V _{DD} - 0.7	—	—	V	I _{OH} = 6.0 mA, V _{DD} = 3.0V
D380	C _{IO}	所有 I/O 引脚					
			—	5	50	pF	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 负电流定义为引脚的拉电流。

2: MCLR 引脚上的泄漏电流主要取决于所施加的电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。

PIC16(L)F18324/18344

表 35-5: 存储器规范

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
高电压编程模式规范							
MEM01	V _{IHH}	进入编程模式的MCLR/VPP引脚电压	7.9	—	9	V	注2
MEM02	I _{PPGM}	处于编程模式期间MCLR/VPP引脚上的电流	—	—	—	uA	注2
编程模式规范							
MEM10	V _{BE}	用于批量擦除的VDD	—	2.7	—	V	
MEM11	I _{DDPGM}	编程时的供电电流	—	—	5	mA	
数据EEPROM存储器规范							
MEM20	E _D	数据EEPROM字节耐擦写能力	100k	—	—	E/W	-40°C ≤ T _A ≤ 85°C
MEM21	T _{D_RET}	特性保持时间	—	40	—	年	假设没有违反其他规范
MEM22	N _{D_REF}	刷新前的总擦除/写周期	—	—	100k	E/W	
MEM23	V _{D_RW}	用于读或擦除/写操作的VDD	V _{DDMIN}	—	V _{DDMAX}	V	
MEM24	T _{D_BEW}	字节擦除和写周期时间	—	4.0	5.0	ms	
闪存程序存储器规范							
MEM30	E _P	闪存存储器单元耐擦写能力	10k	—	—	E/W	-40°C ≤ T _A ≤ 85°C (注1)
MEM32	T _{P_RET}	特性保持时间	—	40	—	年	假设没有违反其他规范
MEM33	V _{P_RD}	用于读操作的VDD	V _{DDMIN}	—	V _{DDMAX}	V	
MEM34	V _{P_REW}	用于行擦除或写操作的VDD	V _{DDMIN}	—	V _{DDMAX}	V	
MEM35	T _{P_REW}	自定时行擦除或自定时写周期	—	2.0	2.5	ms	

† 除非另外声明，否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考，未经测试。

注 1: 闪存存储器单元耐擦写能力定义为：一个行擦除操作和一个自定时写操作。

2: 仅当禁止CONFIG[3].LVP时才需要。

表35-6: 温度特性

标准工作条件 (除非另外声明)					
参数编号	符号	特性	典型值	单位	条件
TH01	θ_{JA}	热阻 (结到环境)	70.0	°C/W	14 引脚PDIP封装
			95.3	°C/W	14 引脚SOIC封装
			100.0	°C/W	14 引脚TSSOP封装
			51.5	°C/W	16 引脚UQFN 4x4 mm封装
			62.2	°C/W	20 引脚PDIP封装
			87.3	°C/W	20 引脚SSOP封装
			77.7	°C/W	20 引脚SOIC封装
			43.0	°C/W	20 引脚UQFN 4x4 mm封装
TH02	θ_{JC}	热阻 (结到管壳)	32.75	°C/W	14 引脚PDIP封装
			31.0	°C/W	14 引脚SOIC封装
			24.4	°C/W	14 引脚TSSOP封装
			5.4	°C/W	16 引脚UQFN 4x4 mm封装
			27.5	°C/W	20 引脚PDIP封装
			31.1	°C/W	20 引脚SSOP封装
			23.1	°C/W	20 引脚SOIC封装
			5.3	°C/W	20 引脚UQFN 4x4 mm封装
TH03	TJMAX	最高结温	150	°C	
TH04	PD	功耗	0.800	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	内部功耗	—	W	PINTERNAL = IDD x VDD ⁽¹⁾
TH06	PI/O	I/O 功耗	—	W	PI/O = $\Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	PDER	降额功耗	—	W	PDER = PDMAX (TJ - TA)/ θ_{JA} ⁽²⁾

注 1: IDD为输出引脚上不驱动任何负载时使芯片独立运行的电流。

2: TA= 环境温度, TJ = 结温

PIC16(L)F18324/18344

35.4 交流特性

图35-4: 负载条件

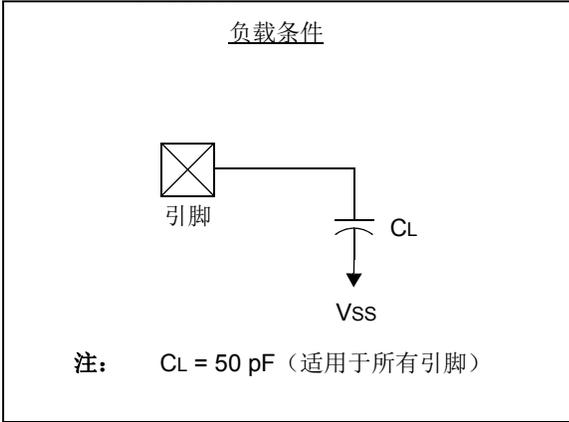


图35-5: 时钟时序

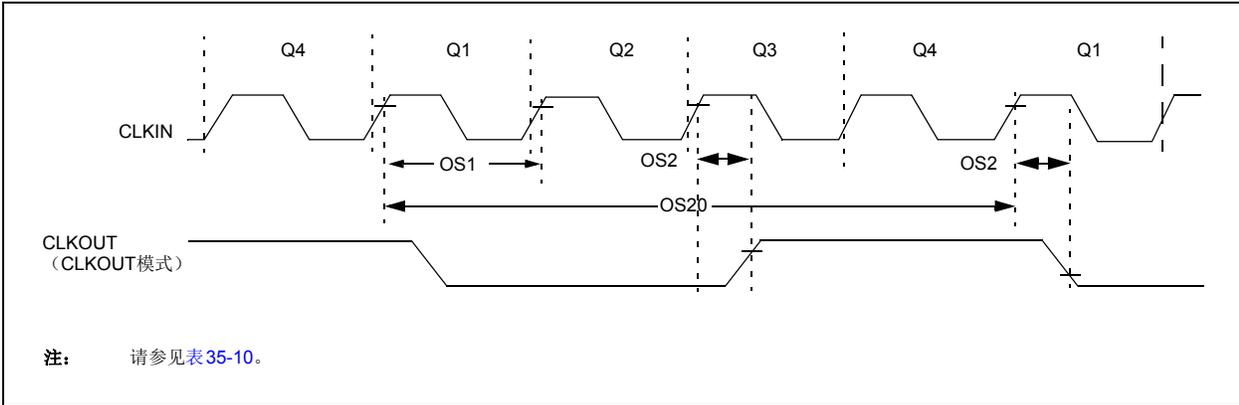


表35-7: 外部时钟/振荡器时序要求

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
ECL 振荡器							
OS1	FECL	时钟频率	—	—	100	kHz	
OS2	TECL_DC	时钟占空比	40	—	60	%	
ECM 振荡器							
OS3	FECM	时钟频率	—	—	8	MHz	注4
OS4	TECM_DC	时钟占空比	40	—	60	%	
ECH 振荡器							
OS5	FECH	时钟频率	—	—	32	MHz	
OS6	TECH_DC	时钟占空比	40	—	60	%	
LP 振荡器							
OS7	FLP	时钟频率	—	—	100	kHz	注4
XT 振荡器							
OS8	FXT	时钟频率	—	—	4	MHz	注4
HS 振荡器							
OS9	FHS	时钟频率	—	—	20	MHz	注4
系统时钟							
OS20	FOSC	系统时钟频率	—	—	32	MHz	(注2和注3)
OS21	Fcy	指令频率	—	FOSC/4	—	MHz	
OS22	Tcy	指令周期	125	1/Fcy	—	ns	

* 这些参数为特性值，但未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考，未经测试。

- 注 1:** 指令周期 (Tcy) 等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值，可能导致振荡器运行不稳定和/或导致电流消耗超出预期值。所有器件在测试“最小”值时，都在OSC1引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的“最大”周期时间限制为“DC”（无时钟）。
- 2:** 系统时钟频率 (Fosc) 通过“主时钟切换控制”选择，如第7.3节“时钟切换”所述。
- 3:** 系统时钟频率 (Fosc) 必须满足第35.2节“标准工作条件”中定义的电压要求。LP、XT和HS振荡器模式要求将一个适当的晶振或谐振器连接到器件。
- 4:** 当通过外部方波为器件提供时钟时，必须使用其中一个EC模式选择。

PIC16(L)F18324/18344

表35-8: 内部振荡器参数⁽¹⁾

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
OS20	FHFOSC	已校准精度的HFINTOSC频率	—	4 8 12 16 32	—	MHz	-40°C至125°C
OS21	FHFOSCLP	针对低功耗优化的HFINTOSC频率	0.93 1.86	1 2	1.07 2.14	MHz MHz	
OS23	FLFOSC	内部LFINTOSC频率	—	31	—	kHz	
OS24	THFOSCST	HFINTOSC从休眠模式唤醒的起振时间	—	11 50	20 —	μs μs	VREGPM = 0 VREGPM = 1
OS26	TLFOSCST	LFINTOSC从休眠模式唤醒的起振时间	—	0.2	—	ms	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 为了确保振荡器频率容差, 必须尽可能靠近器件、在VDD和VSS之间接去耦电容。建议并联0.1 μF和0.01 μF的电容。

2: 请参见图35-6。

图35-6: 器件VDD和温度范围内已校准精度的HFINTOSC频率的精度

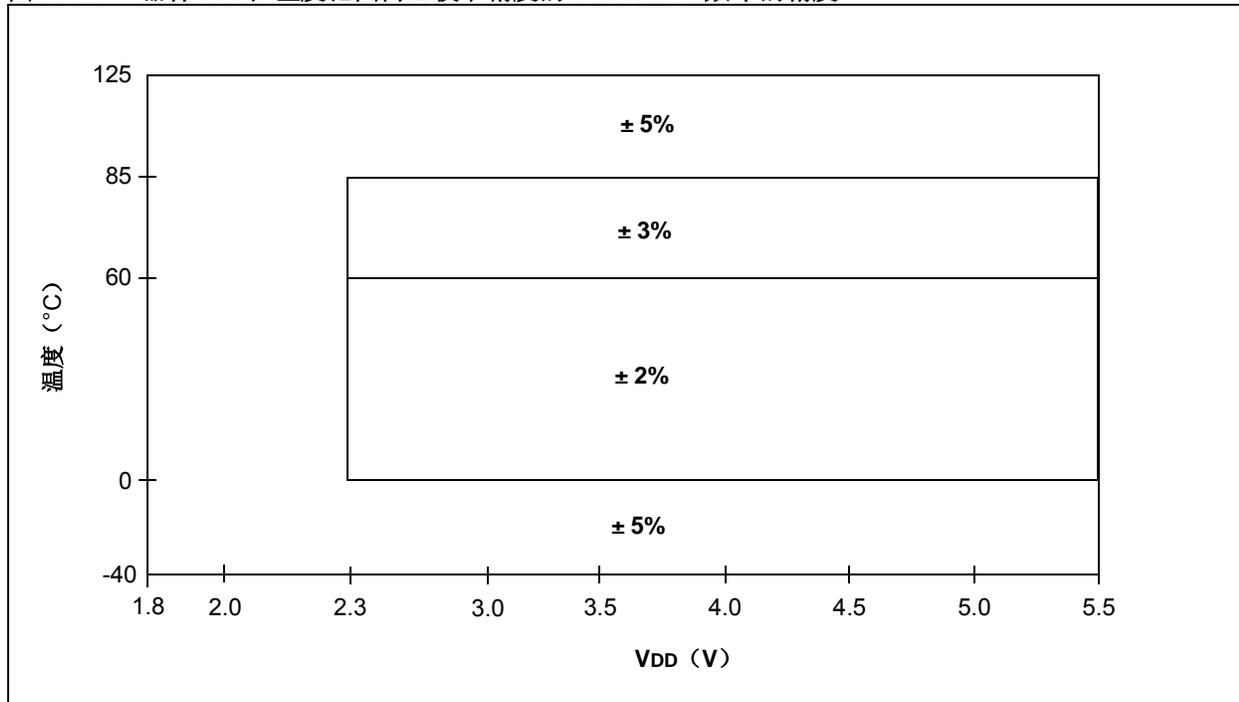


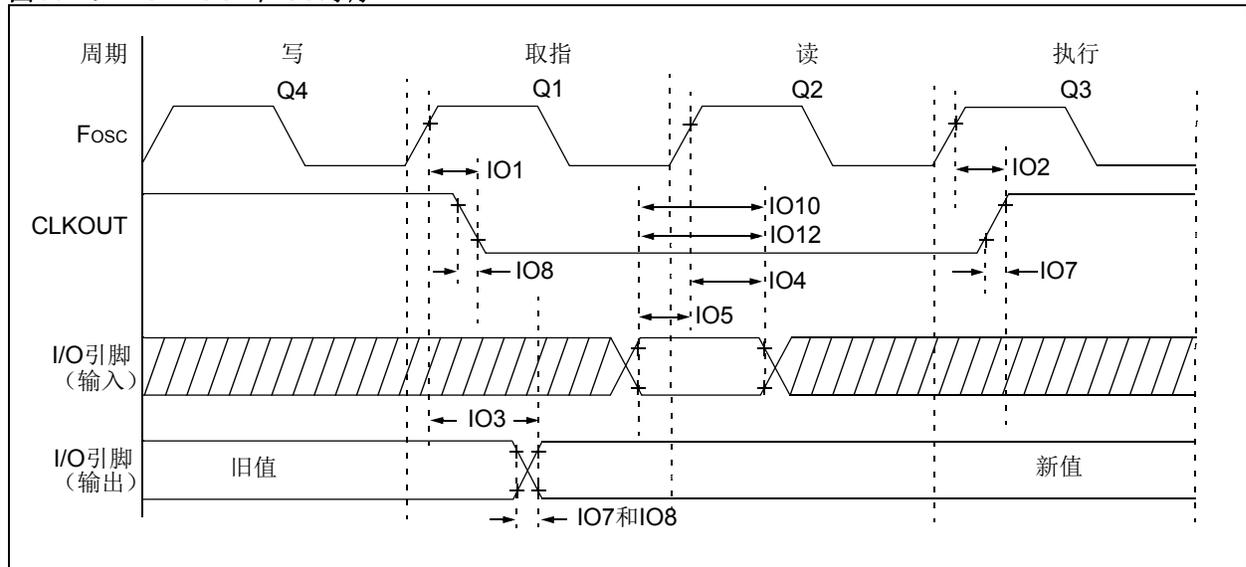
表35-9: PLL时钟时序规范

标准工作条件（除非另外声明）							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
PLL01	FPLLIN	PLL输入频率范围	4	—	8	MHz	
PLL02	FPLLOUT	PLL输出频率范围	16	—	32	MHz	
PLL03	TPLLST	PLL自起振的锁定时间	—	200	—	μs	
PLL04	FPLLJIT	PLL输出频率稳定性（抗抖动性）	-0.25	—	0.25	%	

* 这些参数为特性值，但未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为5V和25°C条件下的值。这些参数仅供设计参考，未经测试。

图35-7: CLKOUT和I/O时序



PIC16(L)F18324/18344

表35-10: CLKOUT和 I/O时序规范

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
IO1	TCLKOUTH	CLKOUT上升沿延时 (Fosc上升沿(Q1周期)到CLKOUT下降沿的时间)	—	—	70	ns	
IO2	TCLKOUTL	CLKOUT下降沿延时 (Fosc上升沿(Q3周期)到CLKOUT上升沿的时间)	—	—	72	ns	
IO3	Tio_VALID	端口输出有效时间 (Fosc上升沿(Q1周期)到端口输出有效的时间)	—	50	70	ns	
IO4	Tio_SETUP	端口输入建立时间 (Fosc上升沿(Q2周期)之前的建立时间)	20	—	—	ns	
IO5	Tio_HOLD	端口输入保持时间 (Fosc上升沿(Q2周期)之后的保持时间)	50	—	—	ns	
IO6	TIOR_SLREN	端口I/O上升时间, 使能压摆率	—	25	—	ns	VDD = 3.0V
IO7	TIOR_SLRDIS	端口I/O上升时间, 禁止压摆率	—	5	—	ns	VDD = 3.0V
IO8	TIOF_SLREN	端口I/O下降时间, 使能压摆率	—	25	—	ns	VDD = 3.0V
IO9	TIOF_SLRDIS	端口I/O下降时间, 禁止压摆率	—	5	—	ns	VDD = 3.0V
IO10	TINT	INT引脚触发中断的高电平时间或低电平时间	25	—	—	ns	
IO11	TIOC	电平变化中断触发中断的最短高电平或低电平时间	25	—	—	ns	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。

图35-8: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序

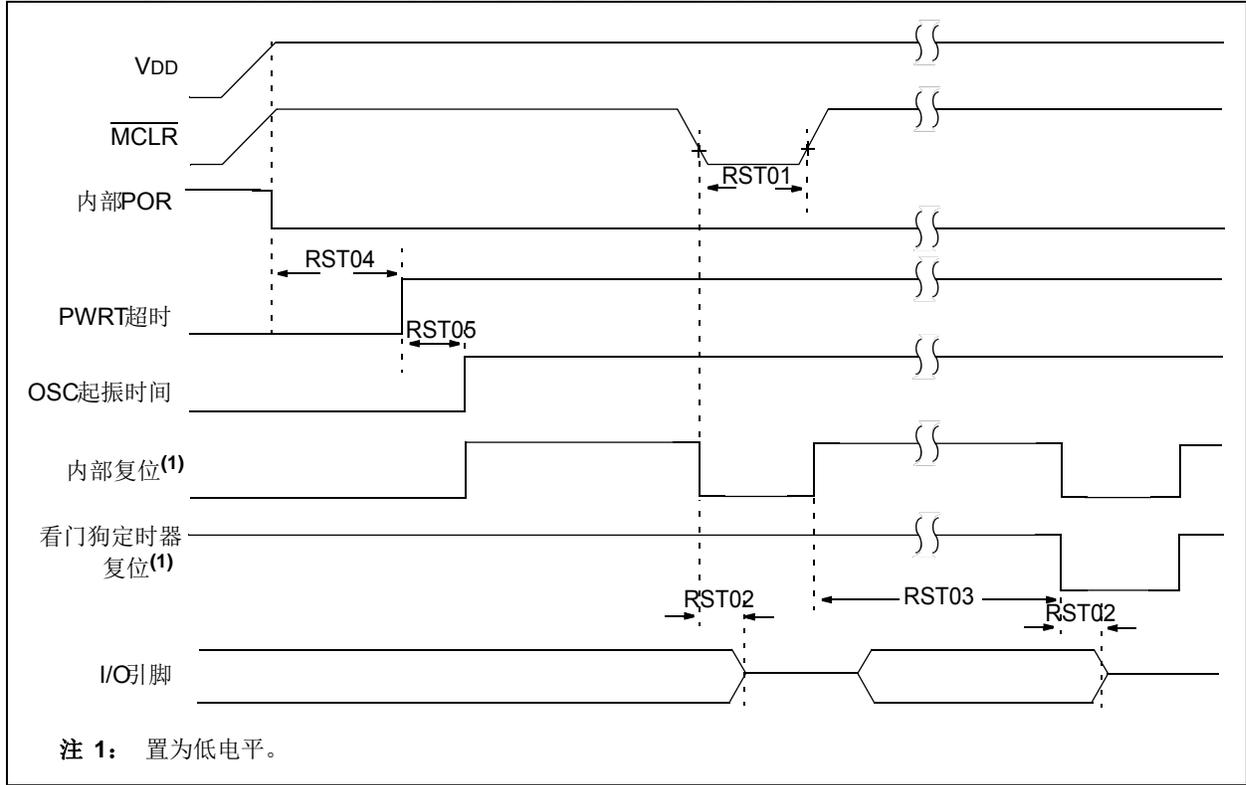
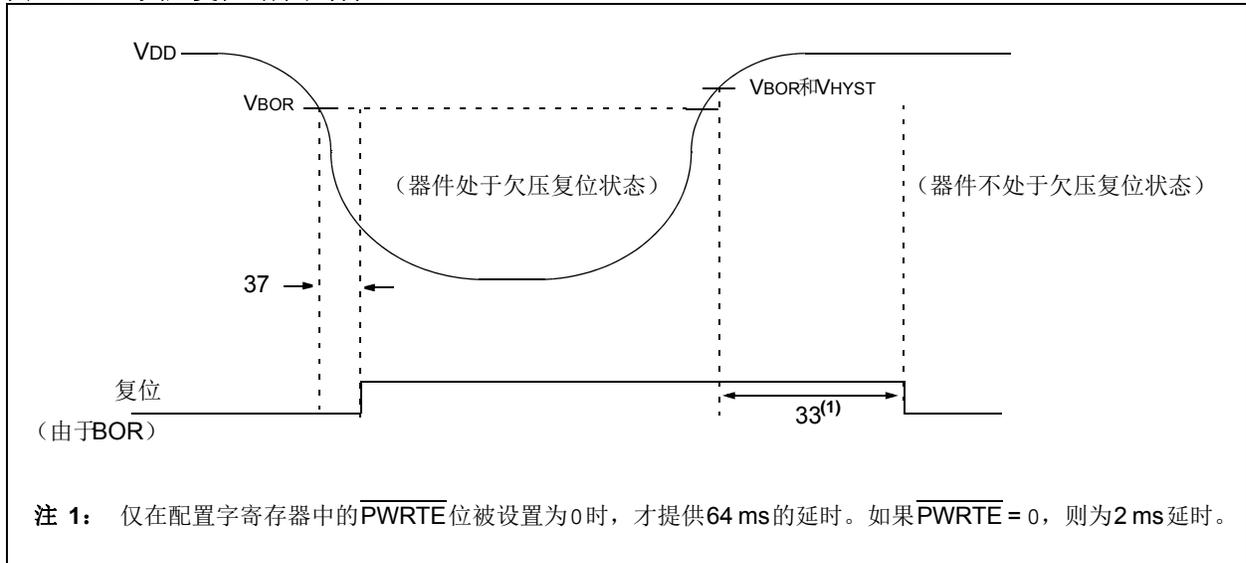


图35-9: 欠压复位时序和特性



PIC16(L)F18324/18344

表35-11： 复位、看门狗定时器、振荡器起振定时器、上电延时定时器、欠压复位和低功耗欠压复位规范

标准工作条件（除非另外声明）							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
RST01	TMCLR	确保复位的MCLR脉冲宽度（低电平）	2	—	—	μs	
RST02	TIOZ	自检测到复位起I/O处于高阻态的时间	—	—	2	μs	
RST03	TWDT	看门狗定时器超时周期	10	16	27	ms	16 ms 标称复位时间
RST04*	TPWRT	上电延时定时器周期	40	65	140	ms	
RST05	TOST	振荡器起振定时器周期 ^(1,2)	—	1024	—	TOSC	(注3)
RST06	VBOR	欠压复位电压 ⁽⁴⁾	2.55 2.30 1.80	2.70 2.45 1.90	2.85 2.60 2.10	V	BORV = 0 BORV = 1 (PIC16F18324/18344) BORV = 1 (PIC16LF18324/18344)
RST07	VBORHYS	欠压复位滞后电压	0	25	75	mV	
RST08	TBORDC	欠压复位响应时间	1	3	35	μs	
RST09	VLPBOR	低功耗欠压复位电压	1.8	2.1	2.5	V	PIC16LF18324/18344

* 这些参数为特性值，但未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考，未经测试。

注 1: 指令周期（Tcy）等于输入振荡器时钟周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值，可能导致振荡器运行不稳定和/或导致电流消耗超出预期值。所有器件在测试“最小”值时，都在OSC1引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的“最大”周期时间限制为“DC”（无时钟）。

2: 依设计而定。

3: 较慢时钟的周期。

4: 为了确保这些电压容差，必须尽可能靠近器件、在VDD和VSS之间接去耦电容。建议并联0.1 μF和0.01 μF的电容。

表35-12: 模数转换器 (ADC) 特性^(1,2)

标准工作条件 (除非另外声明) VDD = 3.0V, TA = 25°C							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
AD01	NR	分辨率	—	—	10	位	
AD02	EIL	积分误差	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD03	EDL	微分误差	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD04	E0FF	失调误差	—	0.5	±2.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD05	EGN	增益误差	—	±0.2	±2.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD06	VADREF	ADC参考电压 (ADREF+) ⁽³⁾	1.8	—	VDD	V	
AD07	VAIN	满量程	VSS	—	ADREF+	V	
AD08	ZAIN	模拟信号源的推荐阻抗	—	10	—	kΩ	
AD09	RVREF	ADC参考电压梯形阻抗	—	50	—	kΩ	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 总绝对误差为失调误差、增益误差和积分非线性 (INL) 误差的总和。

2: ADC转换结果不会因输入的增加而减小, 并且不会丢失编码。

PIC16(L)F18324/18344

表35-13: 模数转换器 (ADC) 转换时序规范^(1,2)

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
AD20	TAD	ADC时钟周期	1	—	9	us	使用Fosc作为ADC时钟源; ADCS != x11
AD21			1	2	6	us	使用ADCRC作为ADC时钟源; ADCS = x11
AD22	TCNV	转换时间	—	11	—	TAD	GO/DONE位置1到GO/DONE位清零的时间
AD23	TACQ	采集时间	—	2	—	us	
AD24	THCD	采样和保持电容断开时间	0.5	—	—	TAD	基于Fosc的时钟源
			0.5	—	—	TAD	基于ADCRC的时钟源

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

图35-10: ADC转换时序 (ADC时钟基于Fosc)

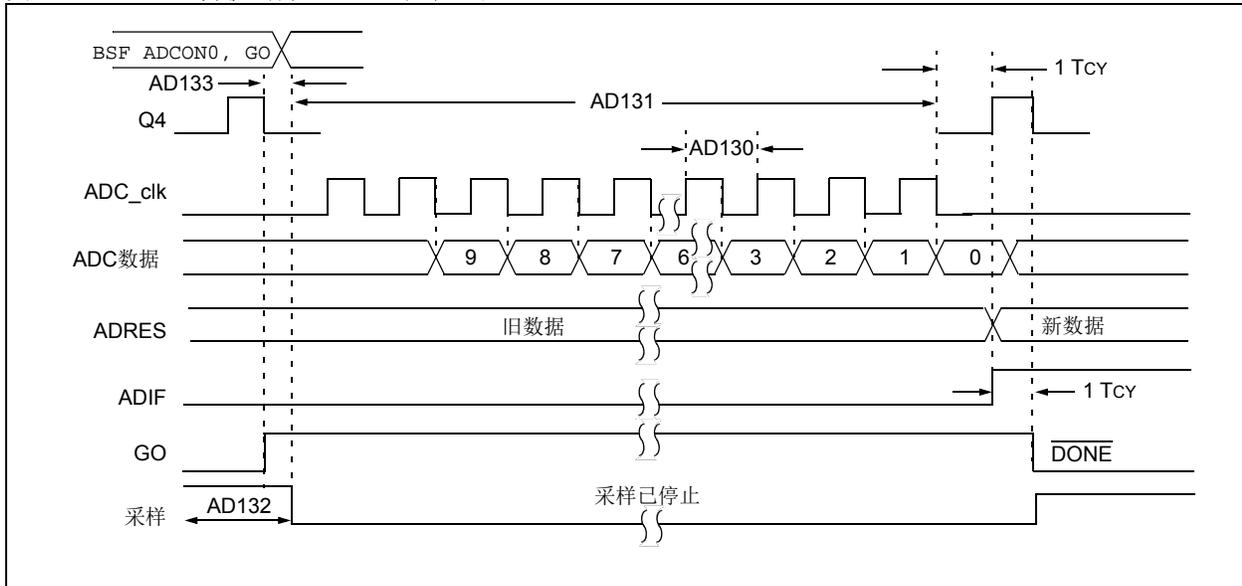
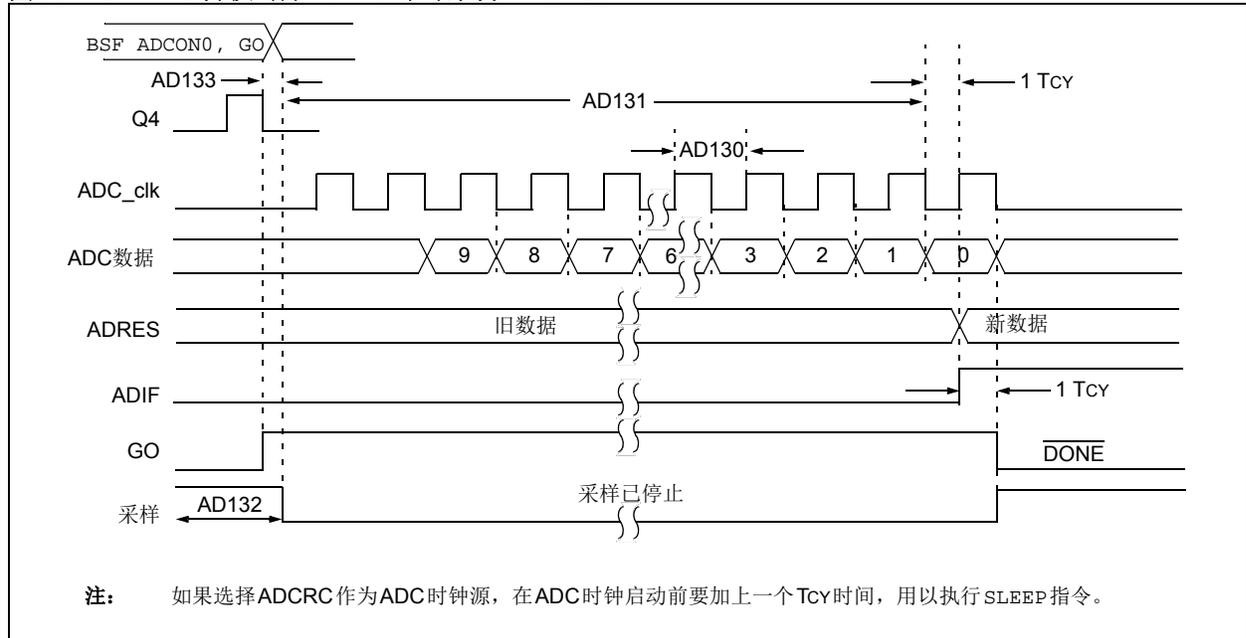


图35-11: ADC转换时序 (ADC时钟来自ADCRC)



PIC16(L)F18324/18344

表35-14: 比较器规范

标准工作条件（除非另外声明） VDD = 3.0V, TA = 25°C 关于工作特性, 请参见第36.0节“直流和交流特性图表”。							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
CM01	VI _{OFF}	输入失调电压	—	—	±50	mV	V _{ICM} = V _{DD} /2
CM02	V _{ICM}	输入共模电压	GND	—	V _{DD}	V	
CM03	CMRR	共模输入抑制比	—	50	—	dB	
CM04	CHYST	比较器滞后	15	25	35	mV	
CM05	T _{RESP} ⁽¹⁾	响应时间上升沿	—	300	600	ns	
		响应时间下降沿	—	220	500	ns	
CM06*	T _{M_{CV}2V_O} ⁽²⁾	模式改变到输出有效的时间	—	—	10	µs	

* 这些参数为特性值, 但未经测试。

注 1: 响应时间是在比较器的一个输入端电压为V_{DD}/2, 而另一个输入端从V_{SS}跳变到V_{DD}时测得的。

2: 模式改变包括改变任一控制寄存器值(包括模块使能)。

表35-15: 数模转换器(DAC)规范

标准工作条件（除非另外声明） VDD = 3.0V, TA = 25°C							
参数编号	符号	特性	最小值	典型值†	最大值	单位	备注
DSB01	V _{LSB}	步长	—	V _{DD} /32	—	V	
DSB01	V _{ACC}	绝对精度	—	—	±0.5	LSb	
DSB03*	R _{UNIT}	单位电阻值	—	6000	—	Ω	
DSB04*	T _{ST}	稳定时间 ⁽¹⁾	—	—	10	µs	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 稳定时间是在DACR<4:0>从00000跳变到01111时测得的。

表35-16: 固定参考电压(FVR)规范

标准工作条件（除非另外声明）							
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
FVR01	V _{FVR1}	1x增益(1.024V标称值)	-4	—	4	%	V _{DD} ≥ 2.5V, -40°C至85°C
FVR02	V _{FVR2}	2x增益(2.048V标称值)	-4	—	4	%	V _{DD} ≥ 2.5V, -40°C至85°C
FVR03	V _{FVR4}	4x增益(4.096V标称值)	-5	—	5	%	V _{DD} ≥ 4.75V, -40°C至85°C
FVR04	T _{FVRST}	FVR启动时间	—	35	—	µs	

图35-12: TIMER0和TIMER1外部时钟时序

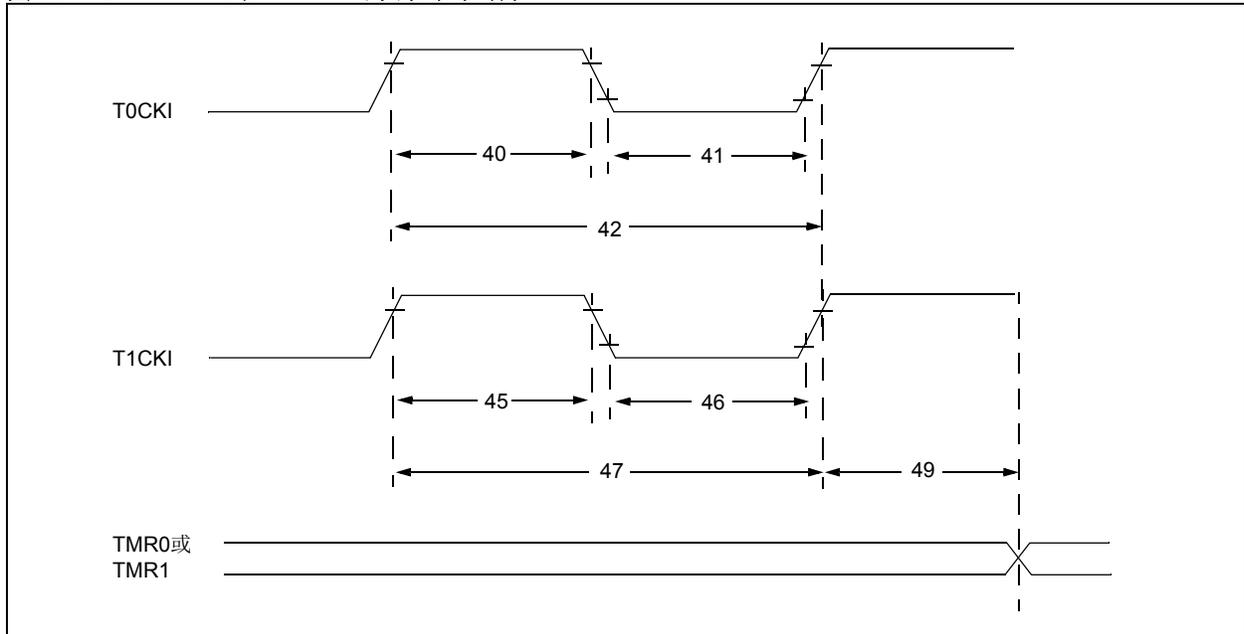


表35-17: TIMER0和TIMER1外部时钟要求

标准工作条件 (除非另外声明)								
参数编号	符号	特性		最小值	典型值†	最大值	单位	条件
40*	T _{T0H}	T0CKI 高电平脉冲宽度	无预分频器	0.5 T _{CY} + 20	—	—	ns	
			带预分频器	10	—	—	ns	
41*	T _{T0L}	T0CKI 低电平脉冲宽度	无预分频器	0.5 T _{CY} + 20	—	—	ns	
			带预分频器	10	—	—	ns	
42*	T _{T0P}	T0CKI 周期		取如下二者中较大值: 20 或 T _{CY} + 40 N	—	—	ns	N = 预分频值
45*	T _{T1H}	T1CKI 高电平时间	同步, 无预分频器	0.5 T _{CY} + 20	—	—	ns	
			同步, 带预分频器	15	—	—	ns	
			异步	30	—	—	ns	
46*	T _{T1L}	T1CKI 低电平时间	同步, 无预分频器	0.5 T _{CY} + 20	—	—	ns	
			同步, 带预分频器	15	—	—	ns	
			异步	30	—	—	ns	
47*	T _{T1P}	T1CKI 输入周期	同步	取如下二者中较大值: 30 或 T _{CY} + 40 N	—	—	ns	N = 预分频值
			异步	60	—	—	ns	
48	F _{T1}	辅助振荡器输入频率范围 (将 T1OSCEN 位置 1 使能振荡器)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	从外部时钟边沿到定时器递增的延时		2 T _{osc}	—	7 T _{osc}	—	同步模式下的定时器

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

PIC16(L)F18324/18344

图35-13: 捕捉/比较/PWM (CCP) 时序

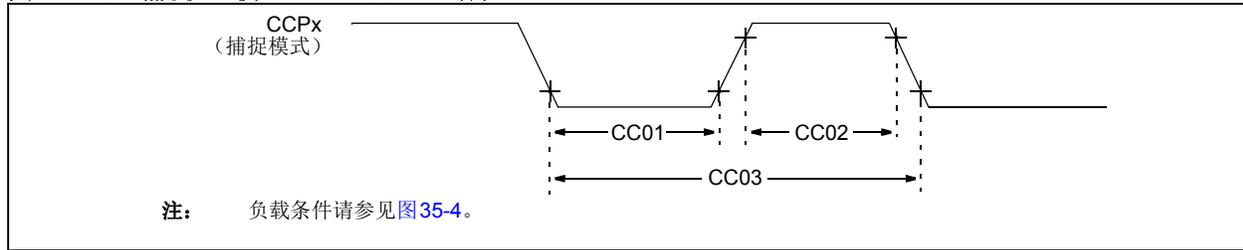


表35-18: 捕捉/比较/PWM (CCP) 特性

标准工作条件 (除非另外声明)								
参数编号	符号	特性		最小值	典型值†	最大值	单位	条件
CC01*	TccL	CCPx输入低电平时间	无预分频器	$0.5T_{CY} + 20$	—	—	ns	
			带预分频器	20	—	—	ns	
CC02*	TccH	CCPx输入高电平时间	无预分频器	$0.5T_{CY} + 20$	—	—	ns	
			带预分频器	20	—	—	ns	
CC03*	TccP	CCPx输入周期		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = 预分频值

* 这些参数为特性值，但未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考，未经测试。

图35-14: CLC传播时序

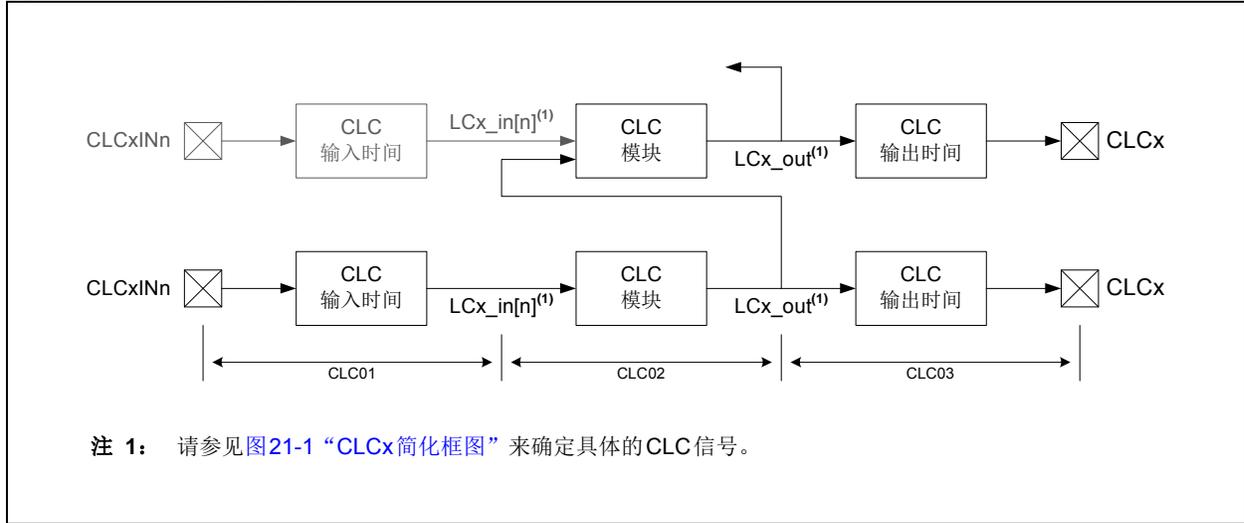


表35-19: 可配置逻辑单元 (CLC) 特性

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
CLC01*	TCLCIN	CLC输入时间	—	7	OS17	ns	(注1)
CLC02*	TCLC	CLC模块输入至输出的传播时间	—	24 12	—	ns ns	V _{DD} = 1.8V V _{DD} > 3.6V
CLC03*	TCLCOUT	CLC输出时间	上升时间	—	OS18	—	(注1)
			下降时间	—	OS19	—	(注1)
CLC04*	FCLCMAX	CLC最大开关频率	—	32	Fosc	MHz	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 关于OS17、OS18和OS19的上升和下降时间, 请参见表35-10。

PIC16(L)F18324/18344

图35-15: EUSART同步发送（主/从）时序

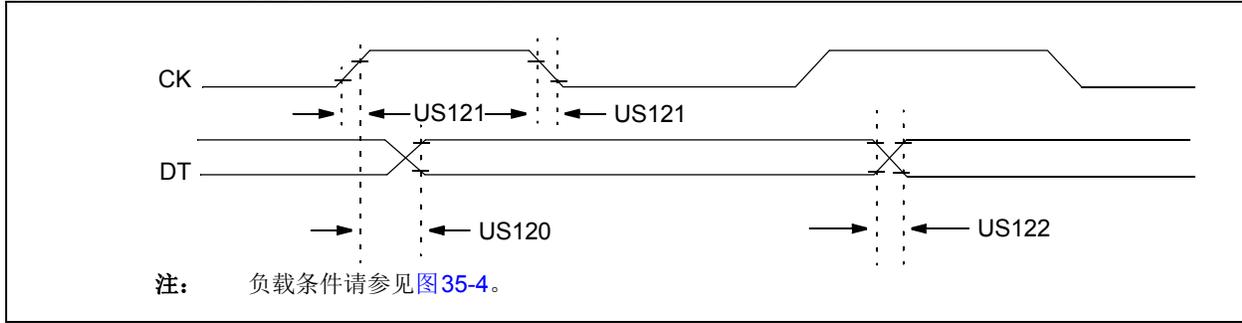


表35-20: EUSART同步发送特性

标准工作条件（除非另外声明）						
参数编号	符号	特性	最小值	最大值	单位	条件
US120	T _{CKH2DTV}	同步发送（主从模式） 时钟高电平到数据输出有效的时间	—	80	ns	3.0V ≤ V _{DD} ≤ 5.5V
			—	100	ns	1.8V ≤ V _{DD} ≤ 5.5V
US121	T _{CKRF}	时钟输出上升时间和下降时间 （主模式）	—	45	ns	3.0V ≤ V _{DD} ≤ 5.5V
			—	50	ns	1.8V ≤ V _{DD} ≤ 5.5V
US122	T _{DTRF}	数据输出上升时间和下降时间	—	45	ns	3.0V ≤ V _{DD} ≤ 5.5V
			—	50	ns	1.8V ≤ V _{DD} ≤ 5.5V

图35-16: EUSART同步接收（主/从）时序

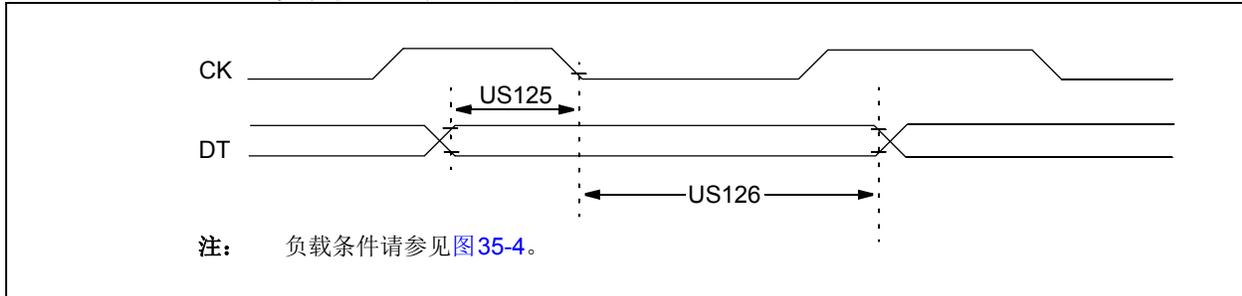


表35-21: EUSART同步接收特性

标准工作条件（除非另外声明）						
参数编号	符号	特性	最小值	最大值	单位	条件
US125	T _{DtV2CKL}	同步接收（主从模式） CK ↓ 前的数据建立时间（DT 保持时间）	10	—	ns	
US126	T _{CKL2DTL}	CK ↓ 之后的数据保持时间（DT 保持时间）	15	—	ns	

图35-17: SPI主模式时序 (CKE = 0, SMP = 0)

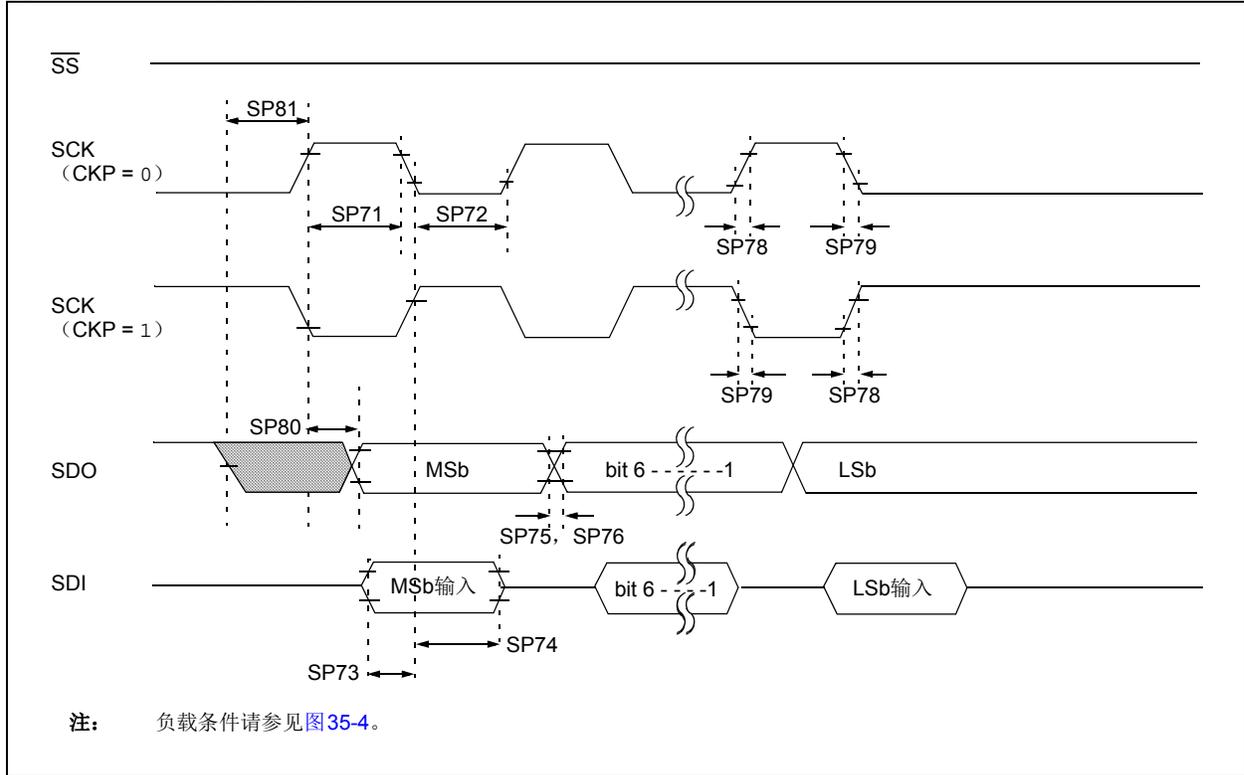
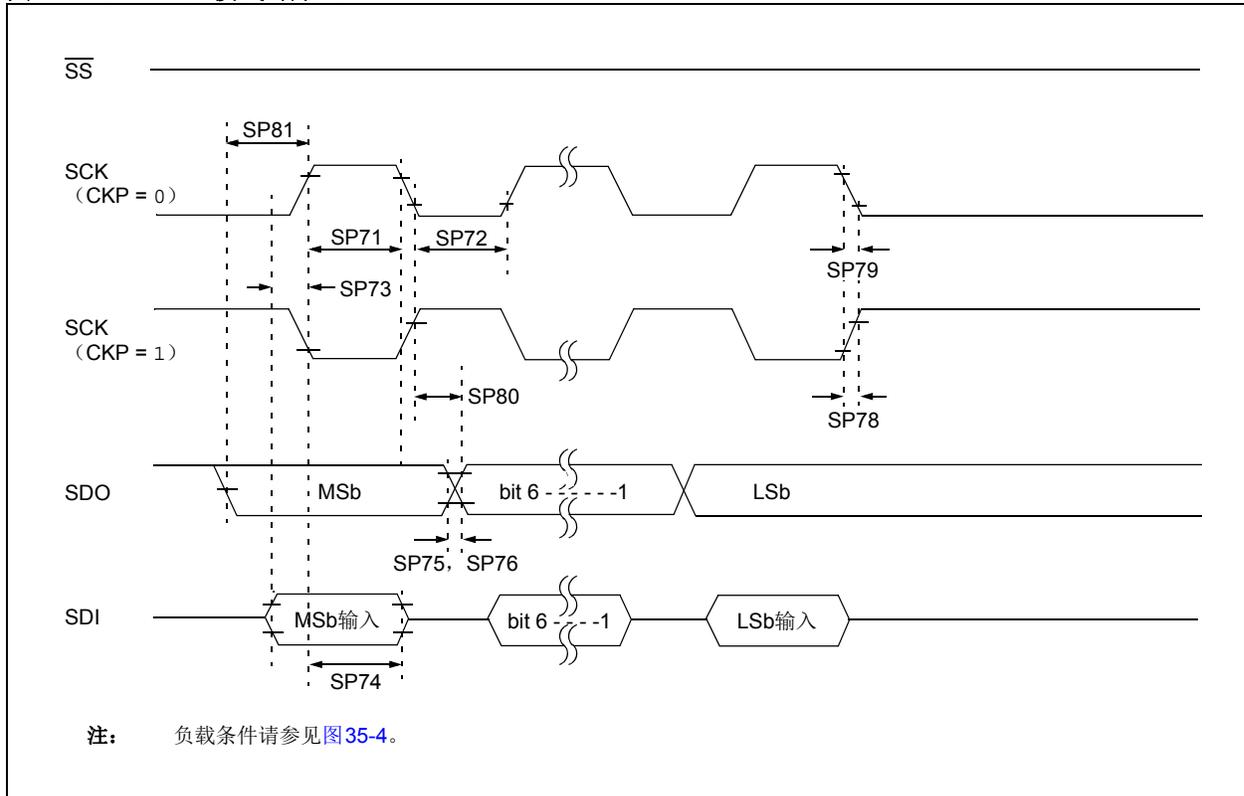


图35-18: SPI主模式时序 (CKE = 1, SMP = 1)



PIC16(L)F18324/18344

图35-19: SPI从模式时序 (CKE = 0)

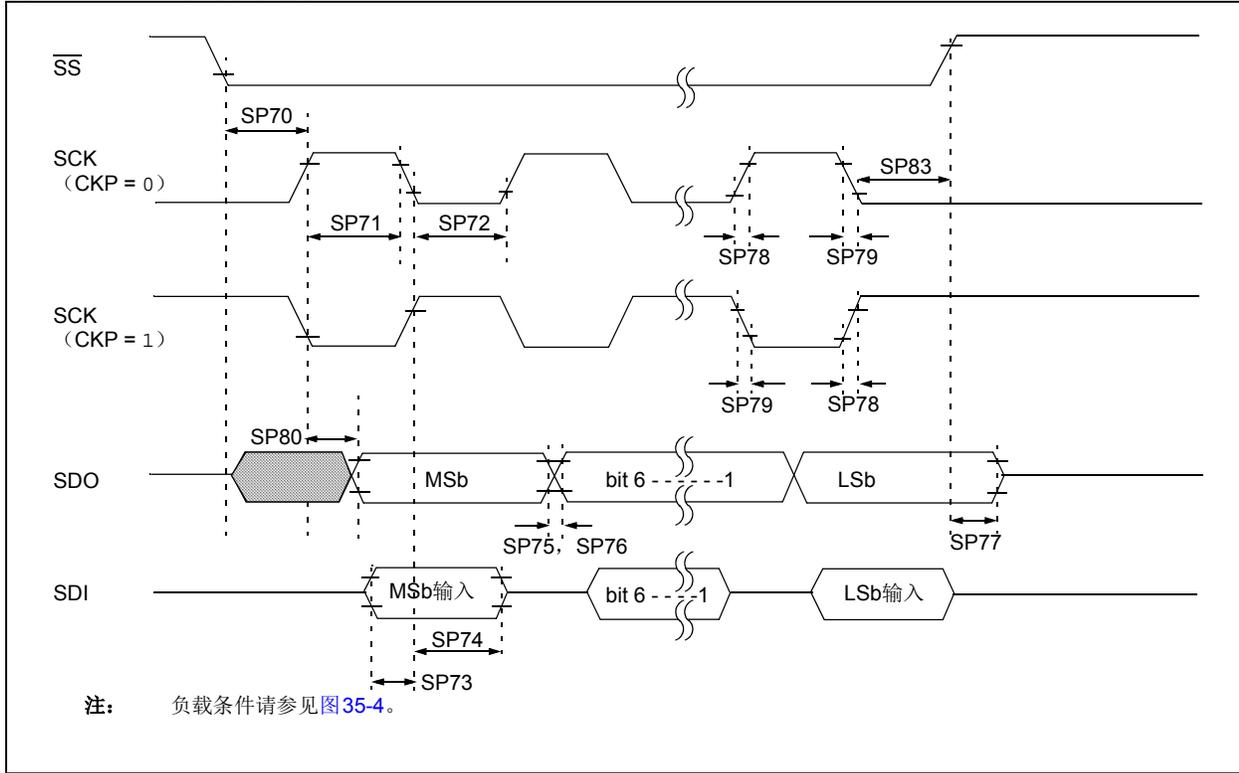


图35-20: SPI从模式时序 (CKE = 1)

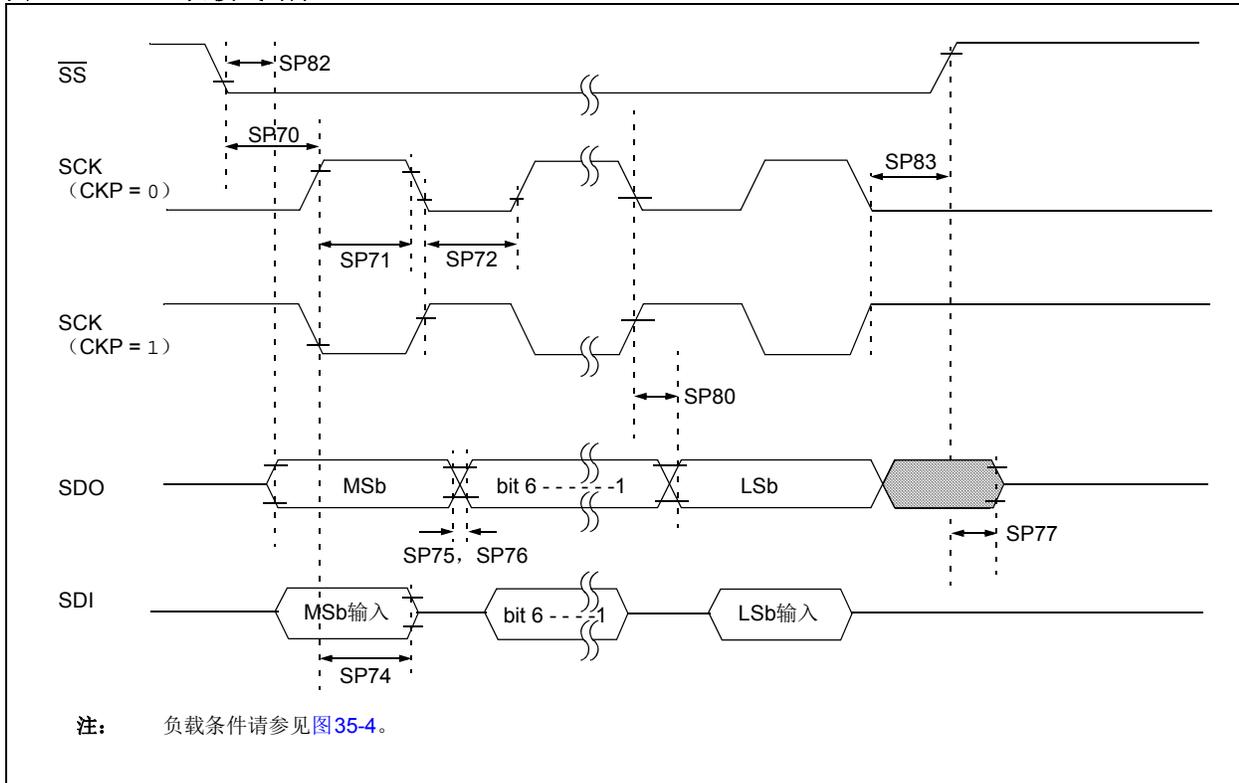


表35-22: SPI模式特性

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
SP70*	TssL2scH, TssL2scL	\overline{SS} ↓到SCK↓或SCK↑输入的时间	2.25*Tcy	—	—	ns	
SP71*	Tsch	SCK输入高电平时间 (从模式)	Tcy + 20	—	—	ns	
SP72*	Tscl	SCK输入低电平时间 (从模式)	Tcy + 20	—	—	ns	
SP73*	TdIV2scH, TdIV2scL	SDI数据输入到SCK边沿的建立时间	100	—	—	ns	
SP74*	Tsch2diL, Tscl2diL	SDI数据输入到SCK边沿的保持时间	100	—	—	ns	
SP75*	TdoR	SDO数据输出上升时间	—	10	25	ns	3.0V ≤ VDD ≤ 5.5V
			—	25	50	ns	1.8V ≤ VDD ≤ 5.5V
SP76*	TdoF	SDO数据输出下降时间	—	10	25	ns	
SP77*	TssH2doZ	\overline{SS} ↑到SDO输出高阻态的时间	10	—	50	ns	
SP78*	TscR	SCK输出上升时间 (主模式)	—	10	25	ns	3.0V ≤ VDD ≤ 5.5V
			—	25	50	ns	1.8V ≤ VDD ≤ 5.5V
SP79*	TscF	SCK输出下降时间 (主模式)	—	10	25	ns	
SP80*	Tsch2doV, Tscl2doV	SCK边沿之后SDO数据输出有效的时间	—	—	50	ns	3.0V ≤ VDD ≤ 5.5V
			—	—	145	ns	1.8V ≤ VDD ≤ 5.5V
SP81*	TdoV2scH, TdoV2scL	SDO数据输出建立到SCK边沿的时间	1 Tcy	—	—	ns	
SP82*	TssL2doV	\overline{SS} ↓边沿之后SDO数据输出有效的时间	—	—	50	ns	
SP83*	Tsch2ssH, Tscl2ssH	SCK边沿之后 \overline{SS} ↑有效的时间	1.5 Tcy + 40	—	—	ns	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅作为设计参考, 未经测试。

PIC16(L)F18324/18344

图35-21: I²C总线启动位/停止位时序

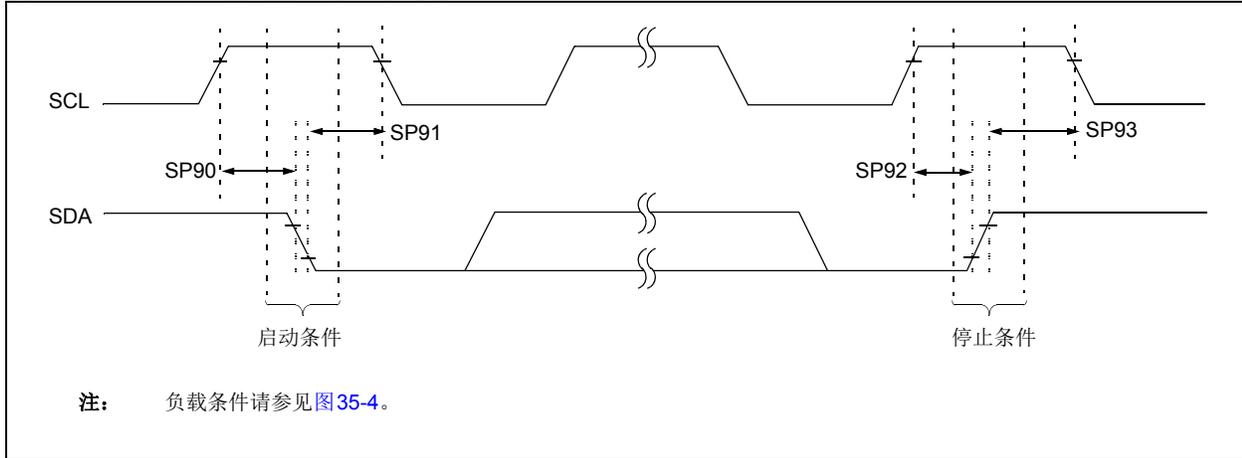


表35-23: I²C总线启动位/停止位特性

标准工作条件 (除非另外声明)								
参数编号	符号	特性	最小值	典型值	最大值	单位	条件	
SP90*	TSU:STA	启动条件 建立时间	100 kHz 模式	4700	—	—	ns	仅与重复启动条件相关
			400 kHz 模式	600	—	—		
SP91*	THD:STA	启动条件 保持时间	100 kHz 模式	4000	—	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz 模式	600	—	—		
SP92*	TSU:STO	停止条件 建立时间	100 kHz 模式	4700	—	—	ns	
			400 kHz 模式	600	—	—		
SP93	THD:STO	停止条件 保持时间	100 kHz 模式	4000	—	—	ns	
			400 kHz 模式	600	—	—		

* 这些参数为特性值, 但未经测试。

图35-22: I²C总线数据时序

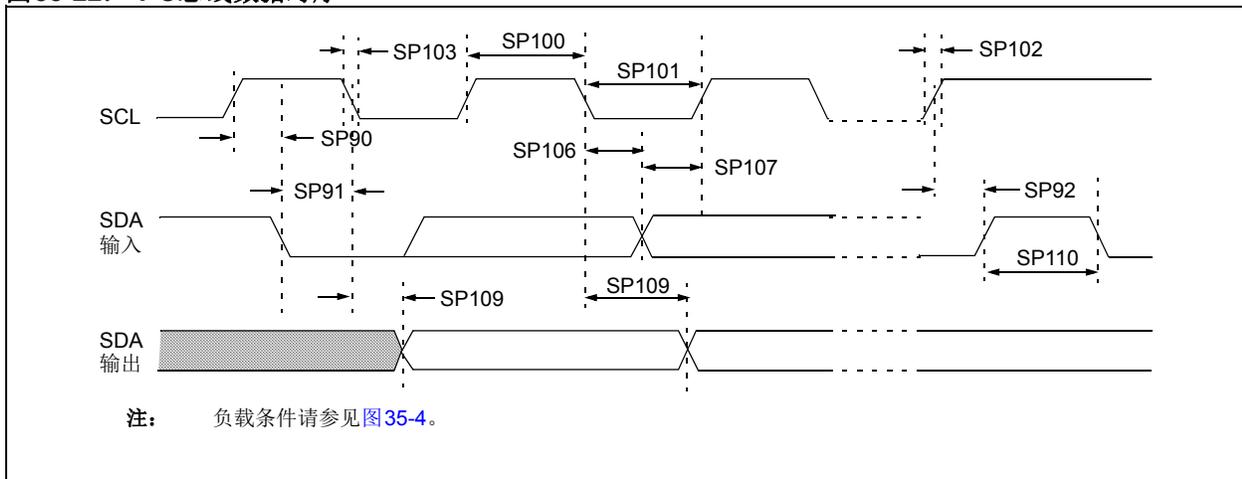


表35-24: I²C总线数据特性

标准工作条件（除非另外声明）							
参数编号	符号	特性	最小值	最大值	单位	条件	
SP100*	T _{HIGH}	时钟高电平时间	100 kHz 模式	4.0	—	μs	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	0.6	—	μs	器件工作频率不得低于 10 MHz
			SSP 模块	1.5T _{CY}	—		
SP101*	T _{LOW}	时钟低电平时间	100 kHz 模式	4.7	—	μs	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	1.3	—	μs	器件工作频率不得低于 10 MHz
			SSP 模块	1.5T _{CY}	—		
SP102*	T _R	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	20 + 0.1C _B	300	ns	C _B 值规定在 10-400 pF 之间
SP103*	T _F	SDA 和 SCL 下降时间	100 kHz 模式	—	250	ns	
			400 kHz 模式	20 + 0.1C _B	250	ns	C _B 值规定在 10-400 pF 之间
SP106*	T _{HD:DAT}	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
SP107*	T _{SU:DAT}	数据输入建立时间	100 kHz 模式	250	—	ns	(注2)
			400 kHz 模式	100	—	ns	
SP109*	T _{AA}	时钟有效到输出有效的时间	100 kHz 模式	—	3500	ns	(注1)
			400 kHz 模式	—	—	ns	
SP110*	T _{BUF}	总线空闲时间	100 kHz 模式	4.7	—	μs	在启动一个新的传输前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
SP111	C _B	总线容性负载	—	400	pF		

* 这些参数为特性值，但未经测试。

注 1: 为避免意外产生启动或停止条件，作为发送器的器件必须提供这个内部最小延时（最小值 300 ns）以补偿 SCL 下降沿的未定义区域。

2: 快速模式（400 kHz）的 I²C 总线器件也可在标准模式（100 kHz）的 I²C 总线系统中使用，但必须满足 T_{SU:DAT} ≥ 250 ns 的要求。如果器件没有延长 SCL 信号的低电平周期，则必然满足此条件。如果该器件延长了 SCL 信号的低电平时间，其下一个数据位必须输出到 SDA 线。在 SCL 线被释放前，根据标准模式 I²C 总线规范，T_{R max.} + T_{SU:DAT} = 1000 + 250 = 1250 ns。

PIC16(L)F18324/18344

36.0 直流和交流特性图表

本章中的图表供**设计参考**，**未经测试**。

一些图表中的数据可能**超出规定的工作范围**（例如，超出规定的V_{DD}范围）。这些数据**仅供参考**，必须保证器件只在规定的范围内工作。

除非另外声明，否则所有图形均同时适用于L和LF器件。

注： 以下图表为基于有限数量样本的统计结果，仅供参考。此处列出的特性未经测试，不做任何担保。一些图表中列出的数据可能超出规定的工作范围（例如，超出了规定的电源范围），因此不在担保范围内。

“典型值”代表25°C时分布的平均值。“Maximum”或“Max.”以及“Minimum”或“Min.”分别代表（平均值 + 3 σ ）或（平均值 - 3 σ ），其中 σ 是每个温度范围内的标准差。

PIC16(L)F18324/18344

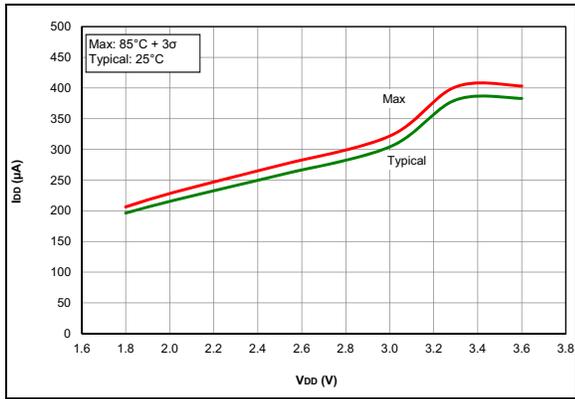


图36-1: I_{DD} , XT振荡器, 4 MHz, 仅限 PIC16LF18324/18344

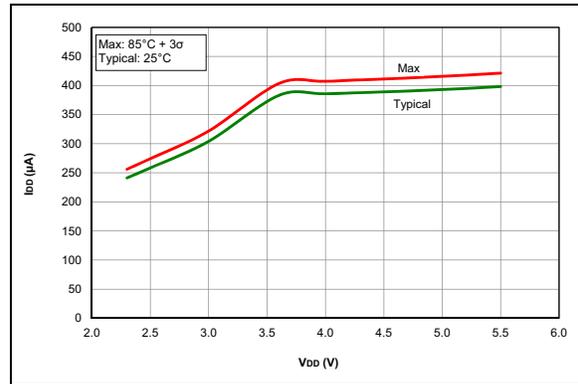


图36-2: I_{DD} , XT振荡器, 4 MHz, 仅限 PIC16F18324/18344

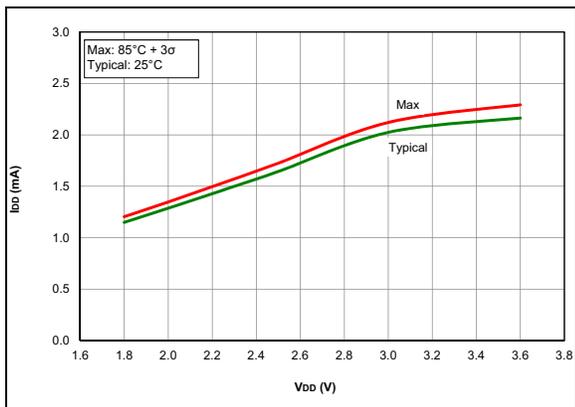


图36-3: I_{DD} , HS振荡器, 32 MHz, 仅限 PIC16LF18324/18344

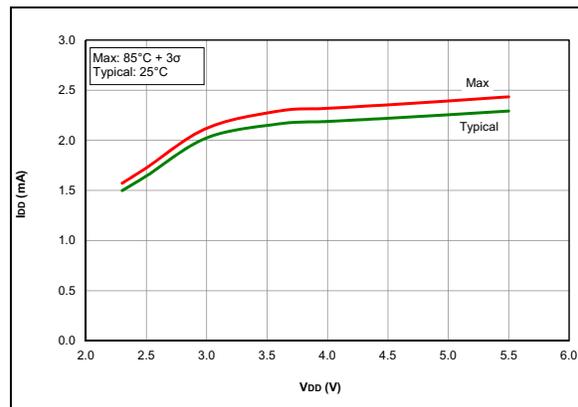


图36-4: I_{DD} , HS振荡器, 32 MHz, 仅限 PIC16F18324/18344

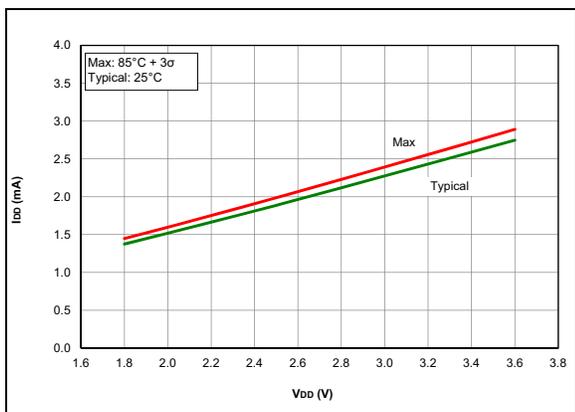


图36-5: I_{DD} , HFINTOSC模式, $F_{OSC} = 32$ MHz, 仅限 PIC16LF18324/18344

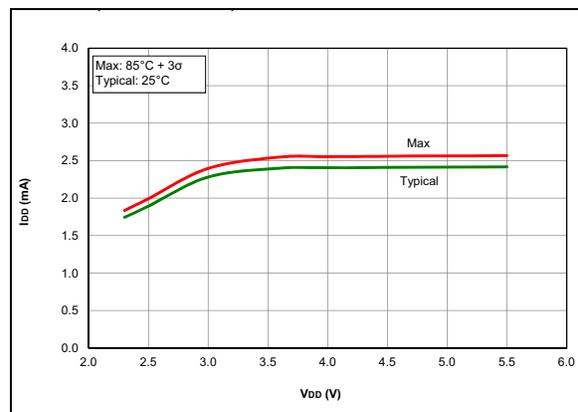


图36-6: I_{DD} , HFINTOSC模式, $F_{OSC} = 32$ MHz, 仅限 PIC16F18324/18344

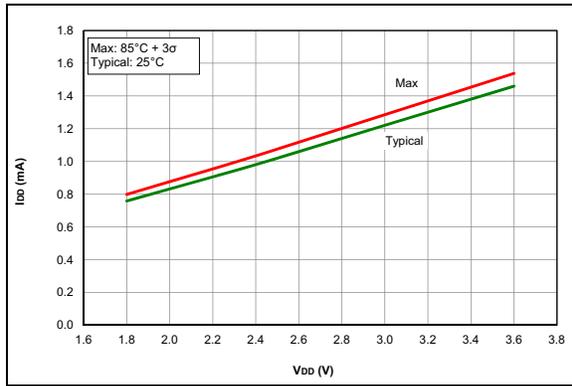


图36-7: I_{DD} , HFINTOSC模式,
FOSC = 16 MHz, 仅限PIC16LF18324/18344

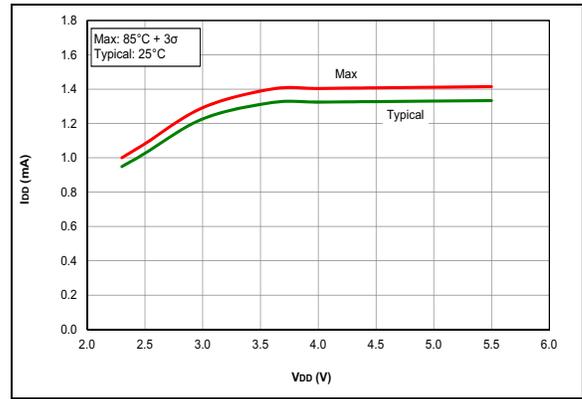


图36-8: I_{DD} , HFINTOSC模式,
FOSC = 16 MHz, 仅限PIC16F18324/18344

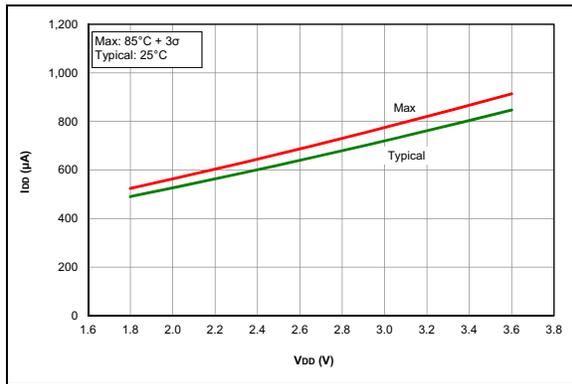


图36-9: I_{DD} , HFINTOSC空闲模式
FOSC = 16 MHz, 仅限PIC16LF18324/18344

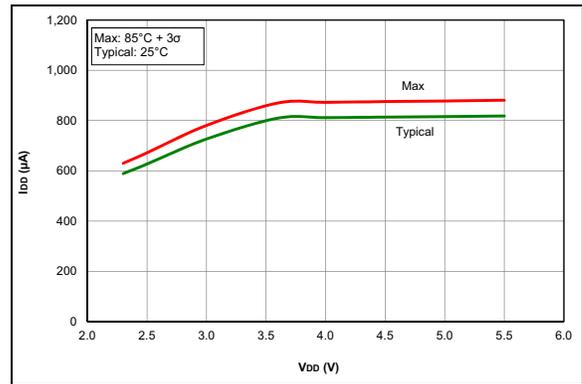


图36-10: I_{DD} , HFINTOSC空闲模式,
FOSC = 16 MHz, 仅限PIC16F18324/18344

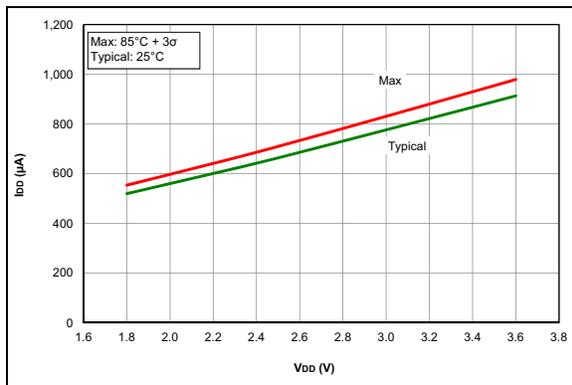


图36-11: I_{DD} , HFINTOSC打盹模式,
FOSC = 16 MHz, 仅限PIC16LF18324/18344

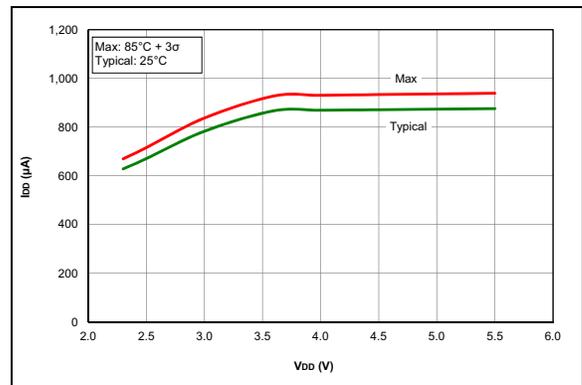


图36-12: I_{DD} , HFINTOSC打盹模式,
FOSC = 16 MHz, 仅限PIC16F18324/18344

PIC16(L)F18324/18344

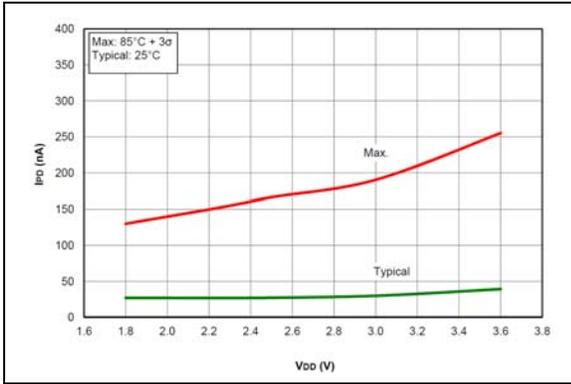


图36-13: 基本IPD, 低功耗休眠模式, 仅限 PIC16LF18324/18344

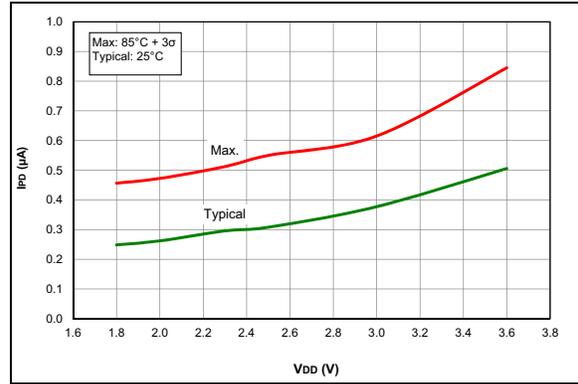


图36-14: IPD, 看门狗定时器 (WDT), 仅限 PIC16LF18324/18344

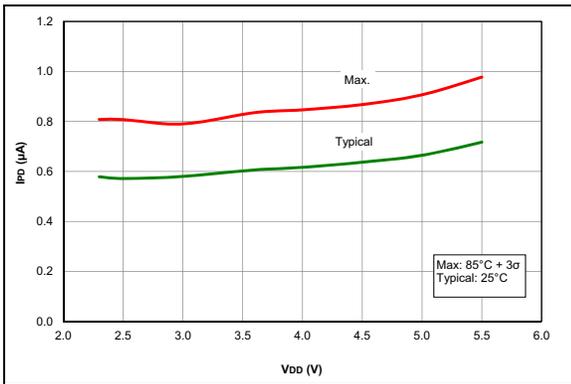


图36-15: IPD, 看门狗定时器 (WDT), 仅限 PIC16F18324/18344

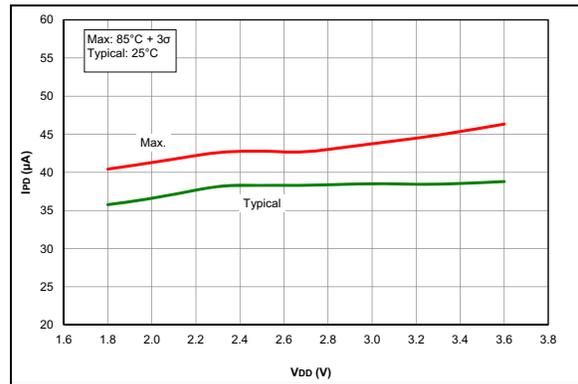


图36-16: IPD, 固定参考电压 (FVR), 仅限 PIC16LF18324/18344

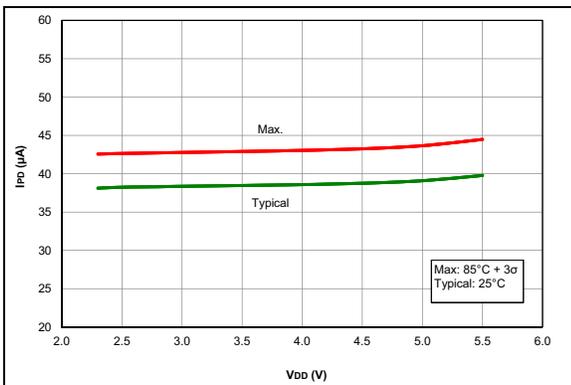


图36-17: IPD, 固定参考电压 (FVR), 仅限 PIC16F18324/18344

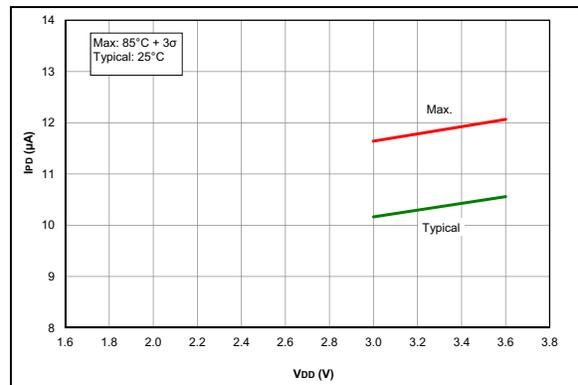


图36-18: IPD, 欠压复位 (BOR), BORV = 1, 仅限 PIC16LF18324/18344

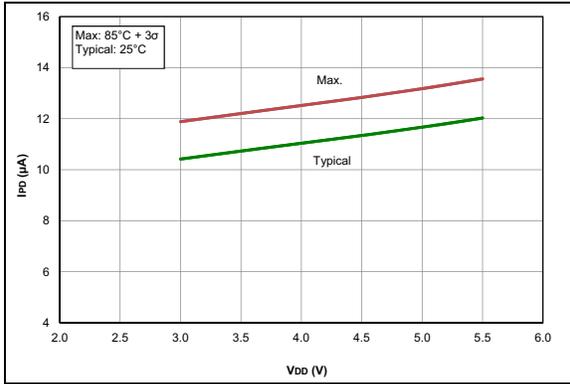


图36-19: IPD, 欠压复位 (BOR), BORV = 1, 仅限PIC16F18324/18344

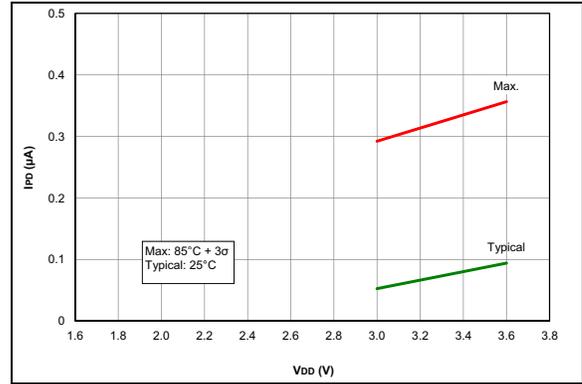


图36-20: IPD, 低功耗欠压复位 (LPBOR = 0), 仅限PIC16LF18324/18344

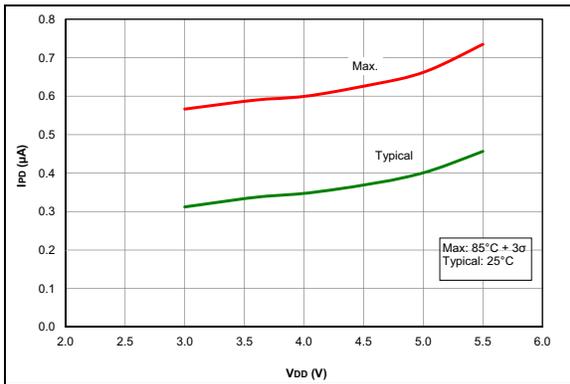


图36-21: IPD, 低功耗欠压复位 (LPBOR = 0), 仅限PIC16F18324/18344

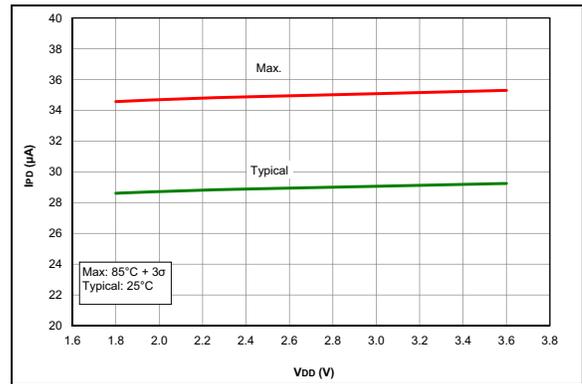


图36-22: IPD, 比较器, 仅限PIC16LF18324/18344

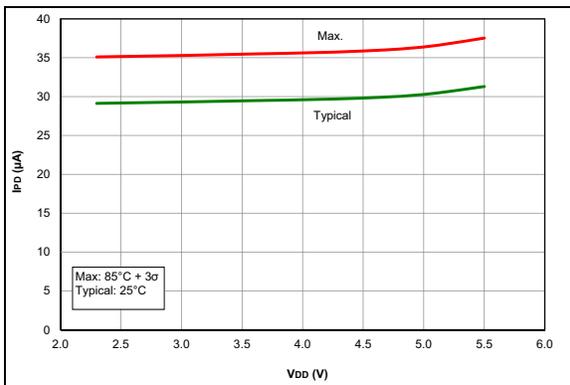


图36-23: IPD, 比较器, 仅限PIC16F18324/18344

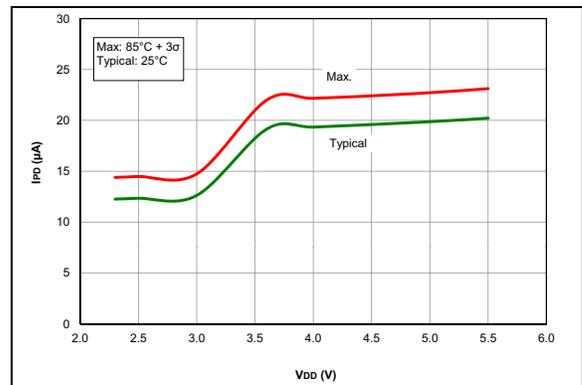


图36-24: 基本IPD, 01, 仅限PIC16F18324/18344

PIC16(L)F18324/18344

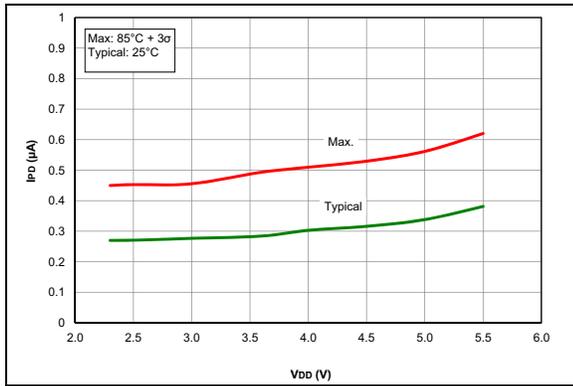


图36-25: 基本IPD, 11, 仅限 PIC16F18324/18344

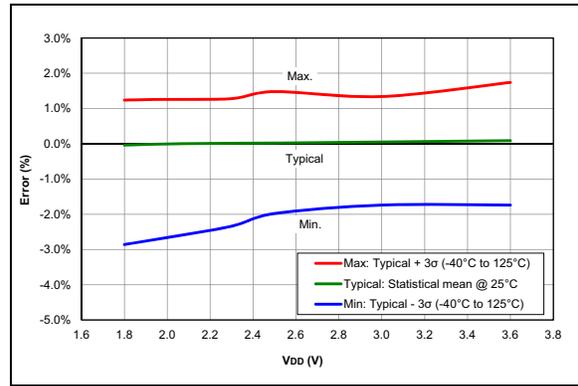


图36-26: HFINTOSC典型频率误差, 仅限 PIC16LF18324/18344

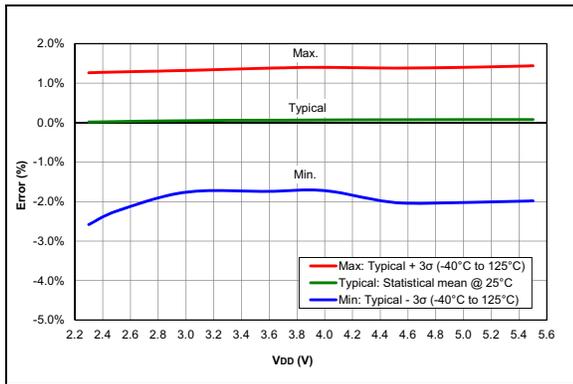


图36-27: HFINTOSC典型频率误差, 仅限 PIC16F18324/18344

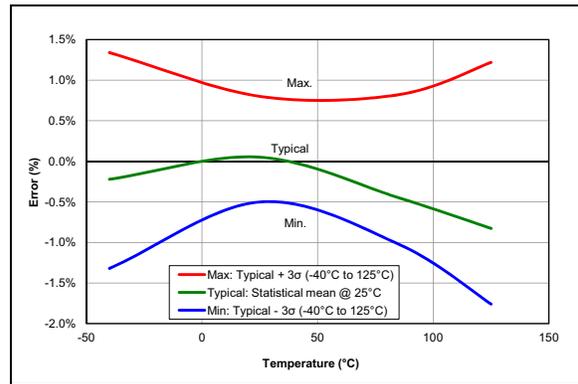


图36-28: HFINTOSC频率误差, VDD = 3V, 所有器件

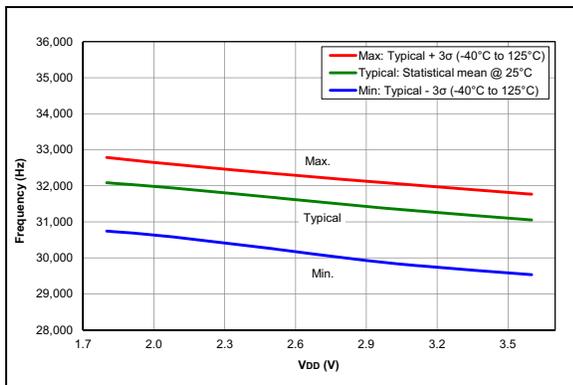


图36-29: LFINTOSC频率, 仅限 PIC16LF18324/18344

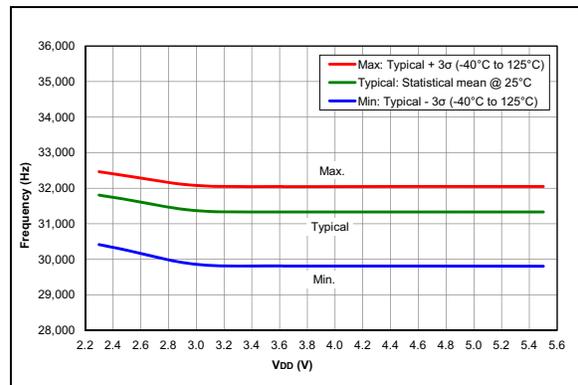


图36-30: LFINTOSC频率, 仅限 PIC16F18324/18344

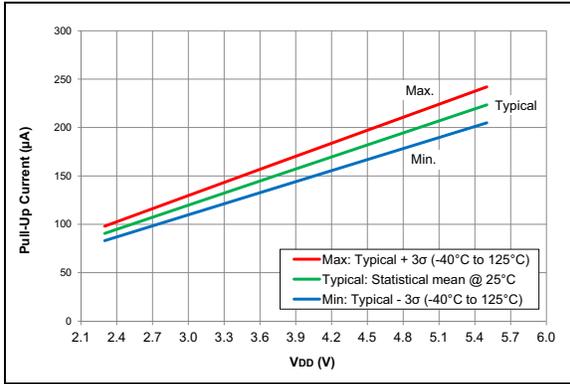


图36-31: 弱上拉电流, 仅限 PIC16F18324/18344

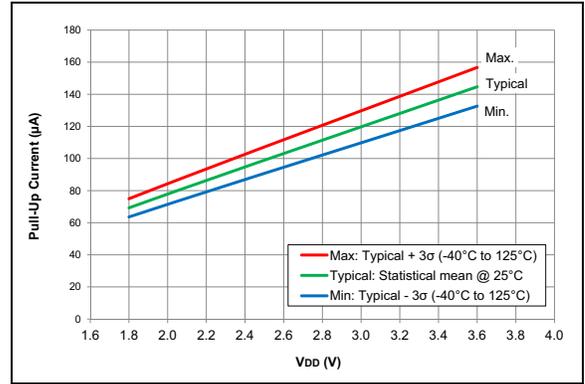


图36-32: 弱上拉电流, 仅限 PIC16LF18324/18344

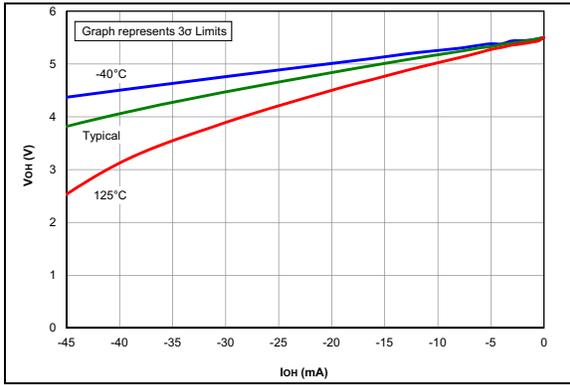


图36-33: 整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 5.5V$, 仅限PIC16F18324/18344

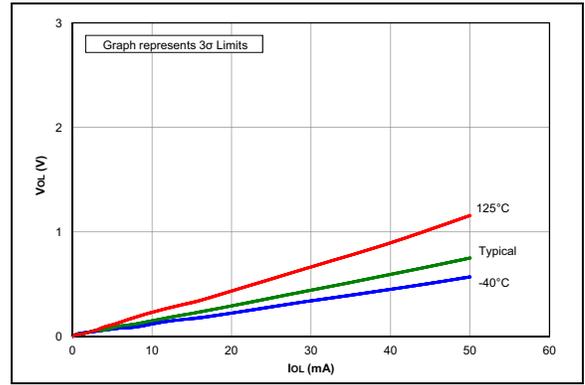


图36-34: 整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 5.5V$, 仅限PIC16F18324/18344

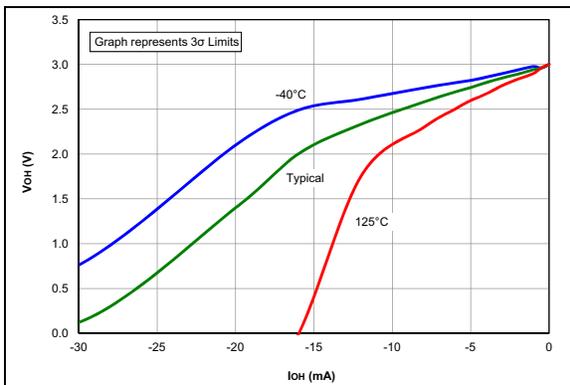


图36-35: 整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 3.0V$, 所有器件

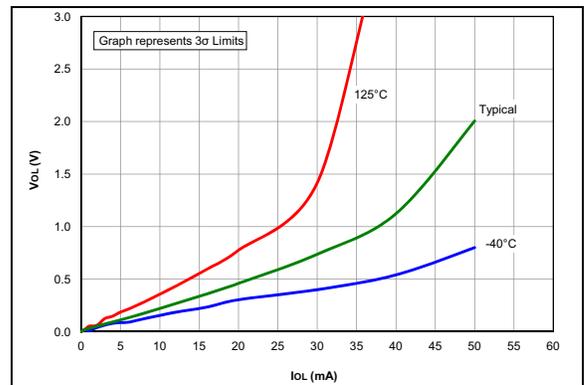


图36-36: 整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 3.0V$, 所有器件

PIC16(L)F18324/18344

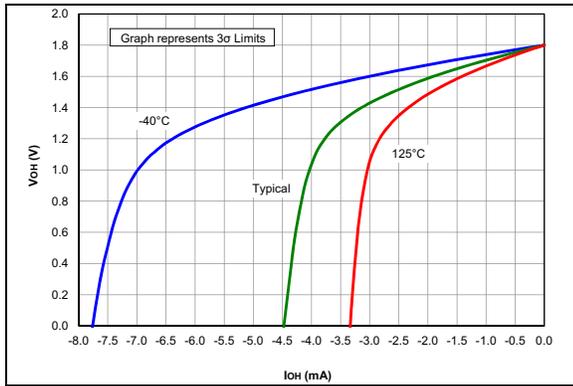


图36-37: 整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 1.8V$, 仅限PIC16LF18324/18344

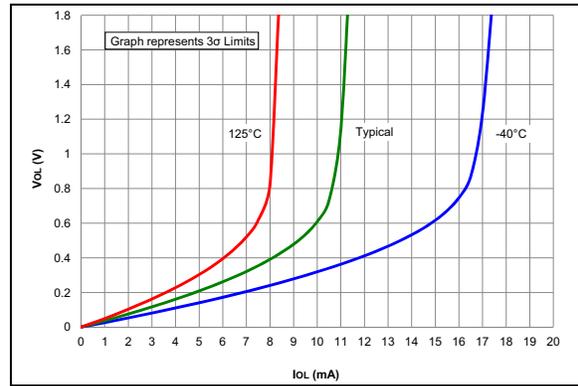


图36-38: 整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 1.8V$, 仅限PIC16LF18324/18344

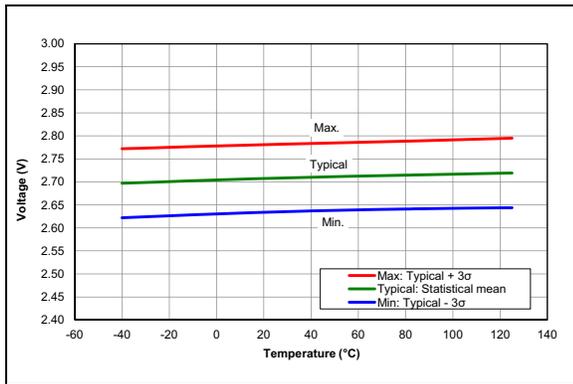


图36-39: 欠压复位电压, 高跳变点, ($BORV = 0$), 所有器件

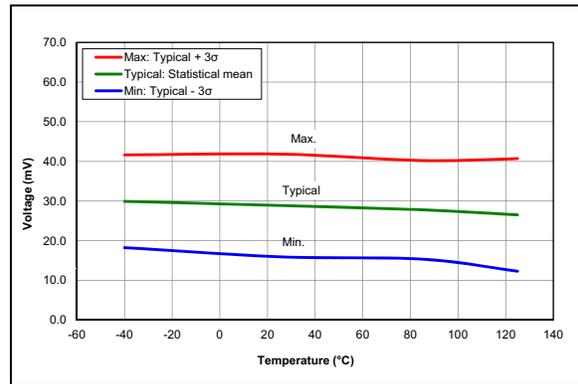


图36-40: 欠压复位滞后, 低跳变点, ($BORV = 0$), 所有器件

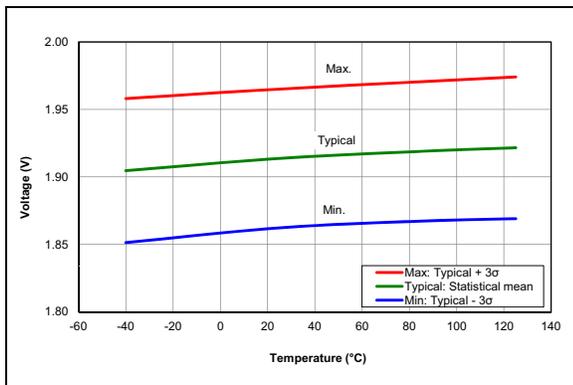


图36-41: 欠压复位电压, 跳变点, ($BORV = 1$), 仅限PIC16LF18324/18344

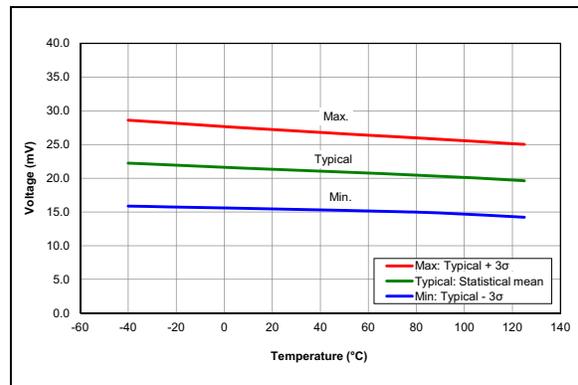


图36-42: 欠压复位滞后, 跳变点, ($BORV = 1$), 仅限PIC16LF18324/18344

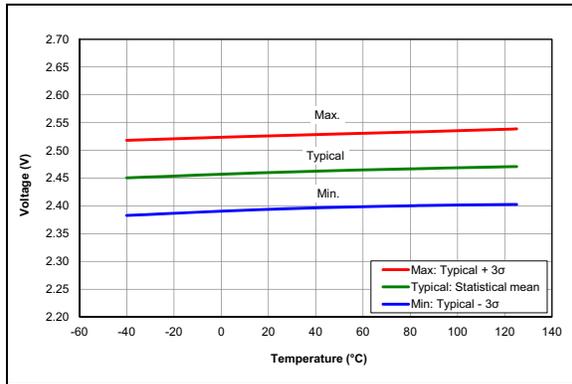


图36-43: 欠压复位电压, 跳变点, (BORV = 1), 仅限PIC16F18324/18344

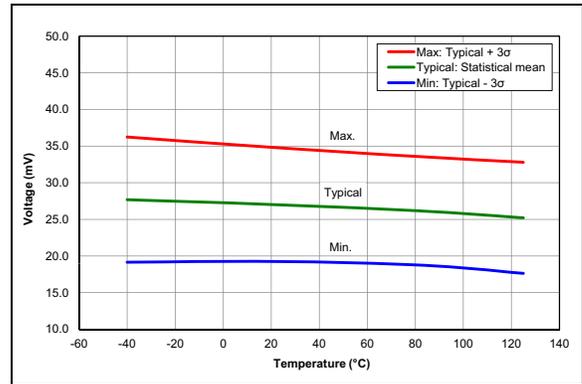


图36-44: 欠压复位滞后, 跳变点, (BORV = 1), 仅限PIC16F18324/18344

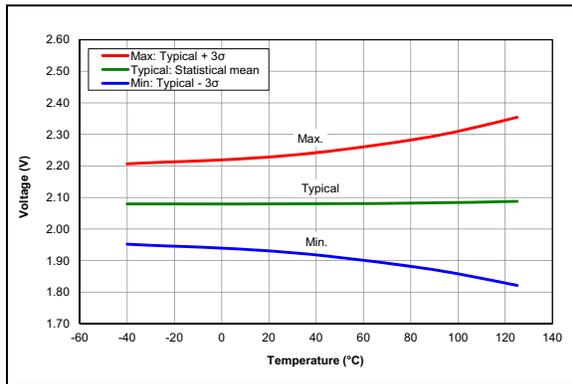


图36-45: LPBOR复位电压, 仅限PIC16LF18324/18344

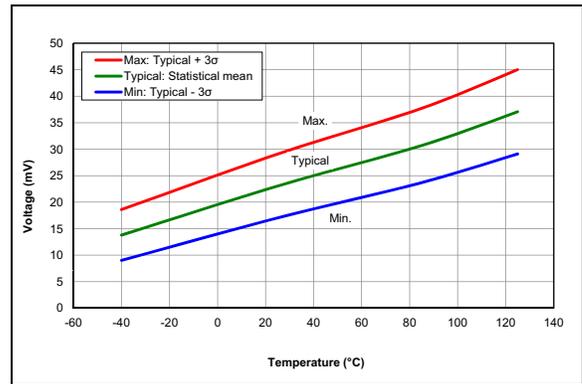


图36-46: LPBOR复位滞后, 仅限PIC16LF18324/18344

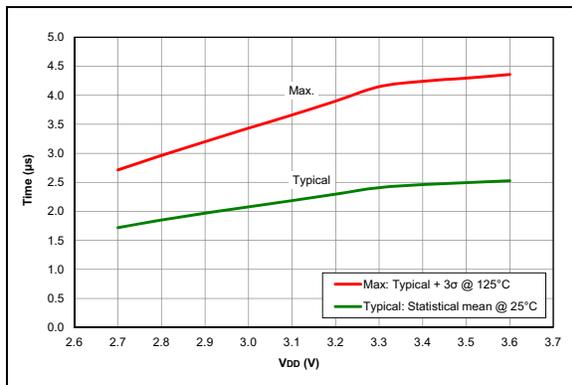


图36-47: BOR响应时间, 仅限PIC16LF18324/18344

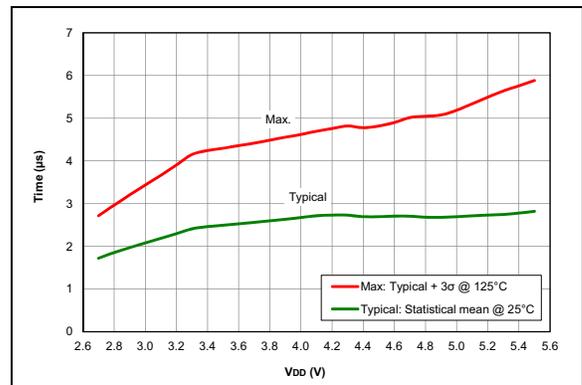


图36-48: BOR响应时间, 仅限PIC16F18324/18344

PIC16(L)F18324/18344

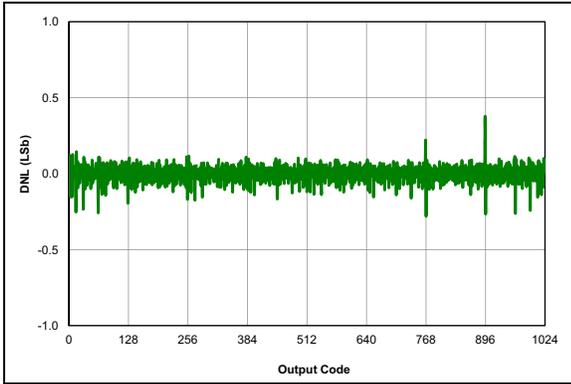


图36-49: ADC 10位模式, 单端DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 1 \mu s$, $25^{\circ}C$, 所有器件

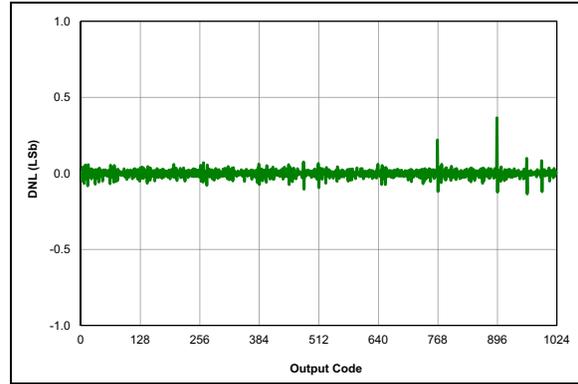


图36-50: ADC 10位模式, 单端DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 4 \mu s$, $25^{\circ}C$, 所有器件

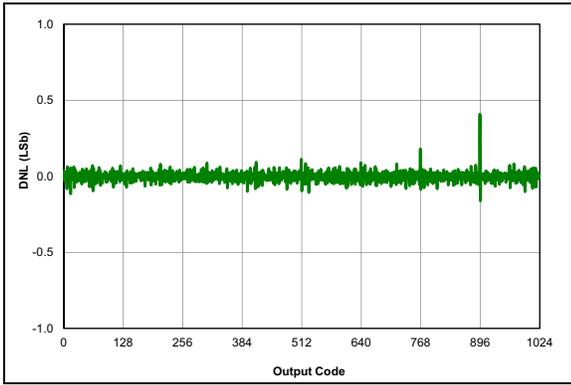


图36-51: ADC 10位模式, 单端DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 8 \mu s$, $25^{\circ}C$, 所有器件

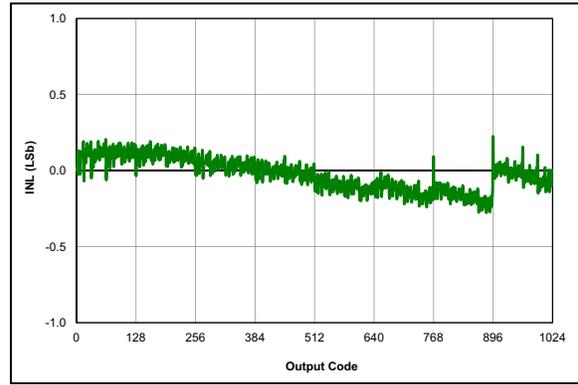


图36-52: ADC 10位模式, 单端INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 1 \mu s$, $25^{\circ}C$, 所有器件

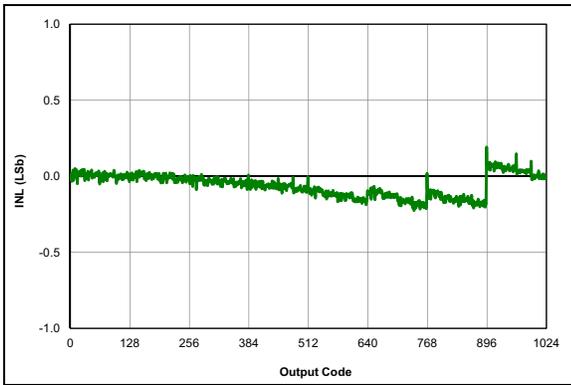


图36-53: ADC 10位模式, 单端INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 4 \mu s$, $25^{\circ}C$, 所有器件

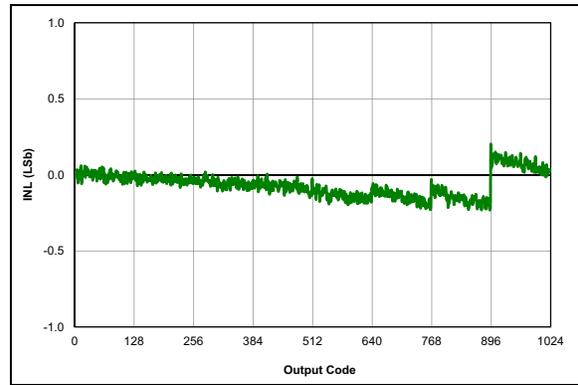


图36-54: ADC 10位模式, 单端INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $T_{AD} = 8 \mu s$, $25^{\circ}C$, 所有器件

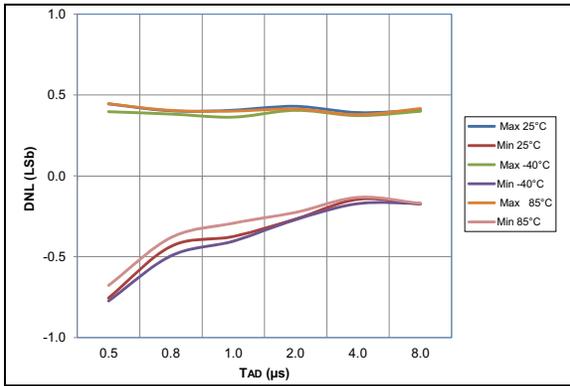


图36-55: ADC 10位模式, 单端DNL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, 所有器件

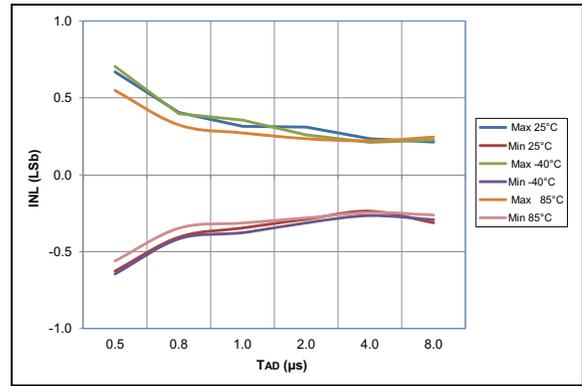


图36-56: ADC 10位模式, 单端INL, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, 所有器件

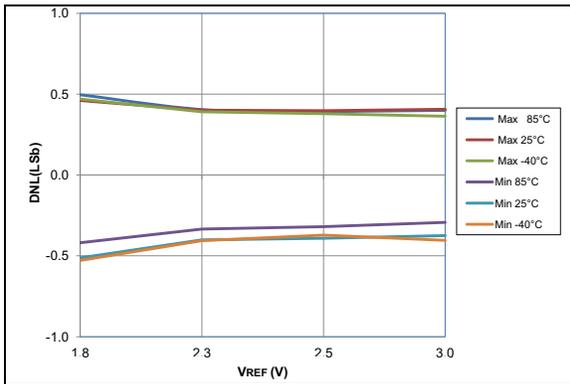


图36-57: ADC 10位模式, 单端DNL, $V_{DD} = 3.0V$, $T_{AD} = 1\mu s$, 所有器件

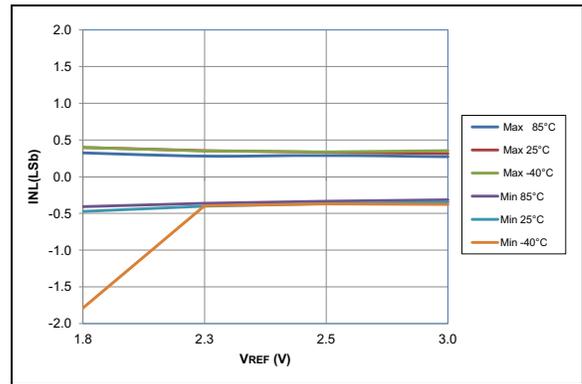


图36-58: ADC 10位模式, 单端INL, $V_{DD} = 3.0V$, $T_{AD} = 1\mu s$, 所有器件

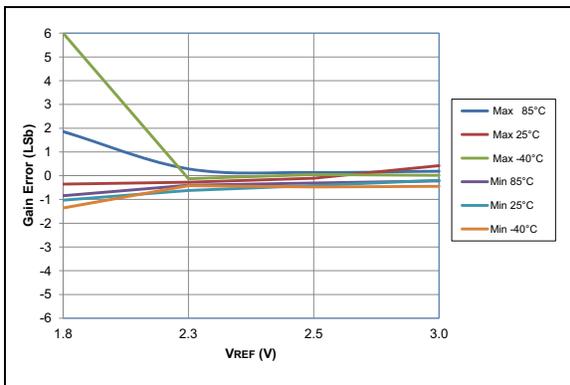


图36-59: ADC 10位模式, 单端增益误差, $V_{DD} = 3.0V$, $T_{AD} = 1\mu s$, 所有器件

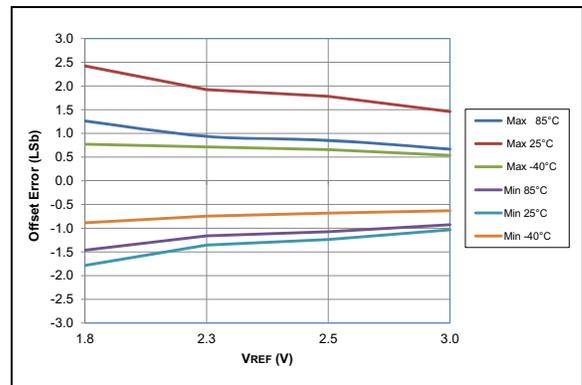


图36-60: ADC 10位模式, 单端失调误差, $V_{DD} = 3.0V$, $T_{AD} = 1\mu s$, 所有器件

PIC16(L)F18324/18344

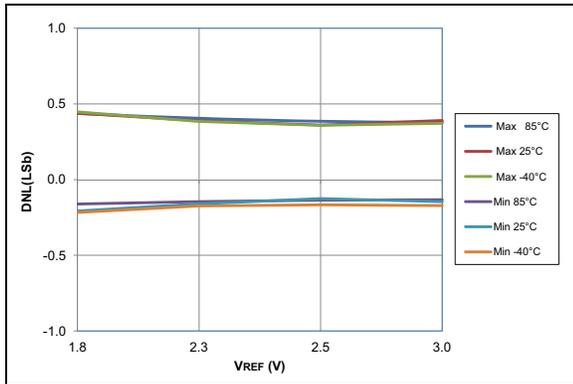


图36-61: ADC 10位模式, 单端DNL, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, 所有器件

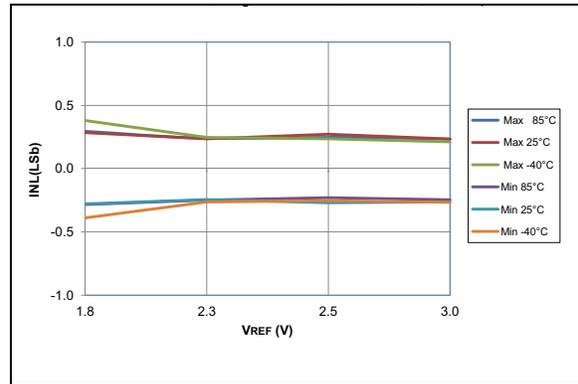


图36-62: ADC 10位模式, 单端INL, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, 所有器件

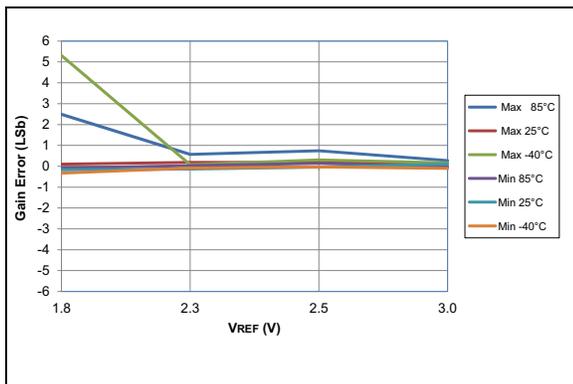


图36-63: ADC 10位模式, 单端增益误差, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, 所有器件

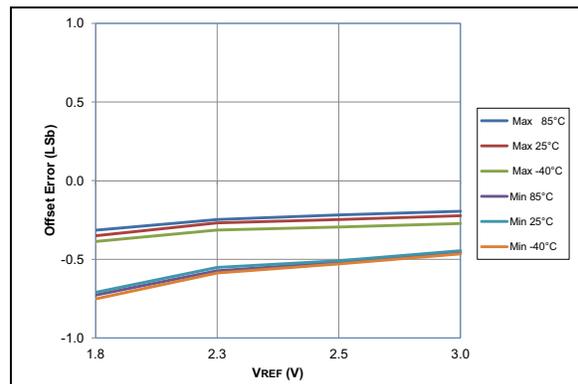


图36-64: ADC 10位模式, 单端失调误差, $V_{DD} = 3.0V$, $T_{AD} = 4 \mu s$, 所有器件

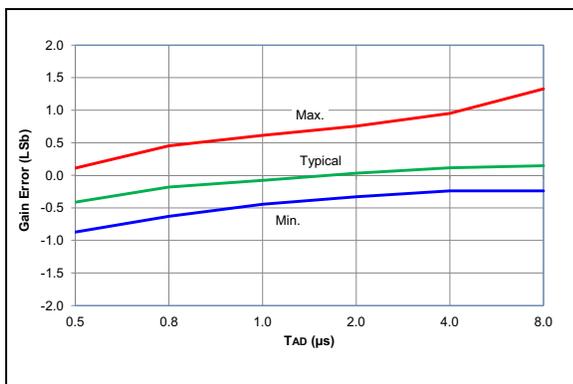


图36-65: ADC 10位模式, 单端增益误差, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $-40^{\circ}C$ 至 $85^{\circ}C$, 所有器件

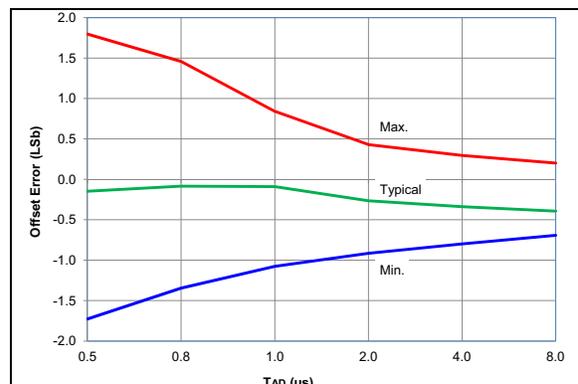


图36-66: ADC 10位模式, 单端失调误差, $V_{DD} = 3.0V$, $V_{REF} = 3.0V$, $-40^{\circ}C$ 至 $85^{\circ}C$, 所有器件

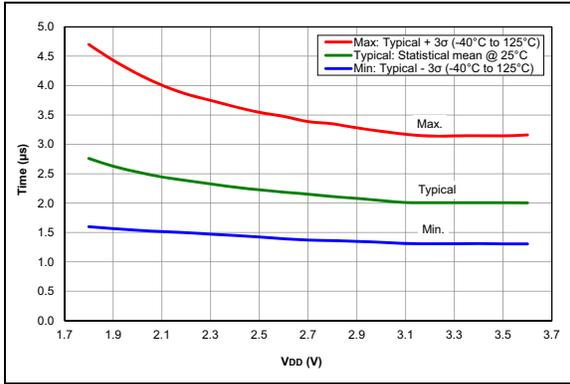


图36-67: ADC RC振荡器周期, 仅限 PIC16LF18324/18344

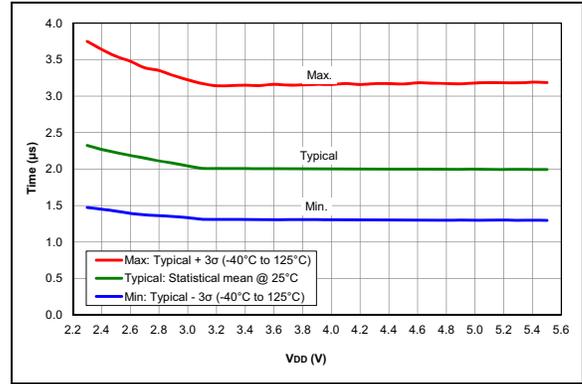


图36-68: ADC RC振荡器周期, 仅限 PIC16F18324/18344

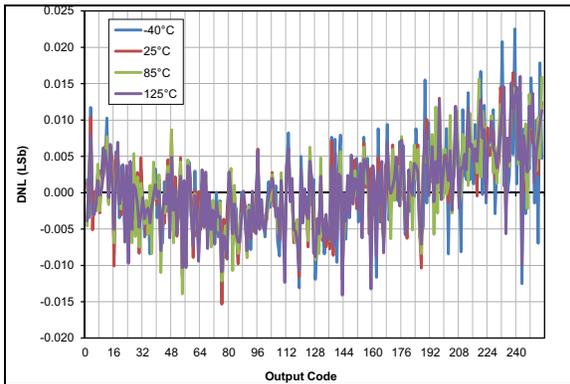


图36-69: 典型DAC DNL误差, $V_{DD} = 3.0V$, V_{REF} = 外部3V, 所有器件

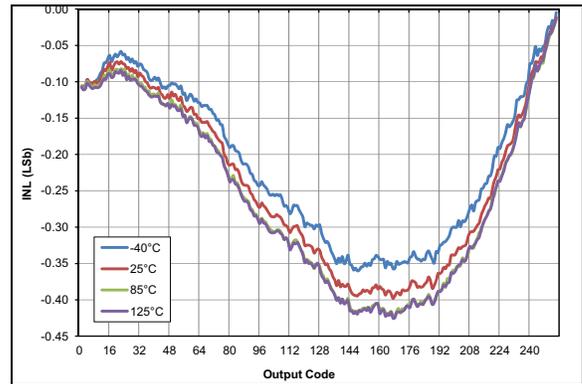


图36-70: 典型DAC INL误差, $V_{DD} = 3.0V$, V_{REF} = 外部3V, 所有器件

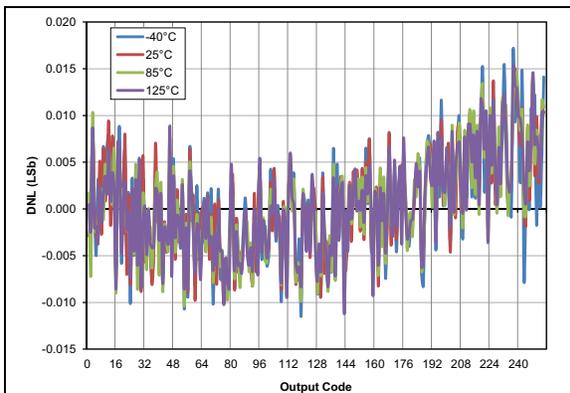


图36-71: 典型DAC DNL误差, $V_{DD} = 5.0V$, V_{REF} = 外部5V, 仅限PIC16F18324/18344

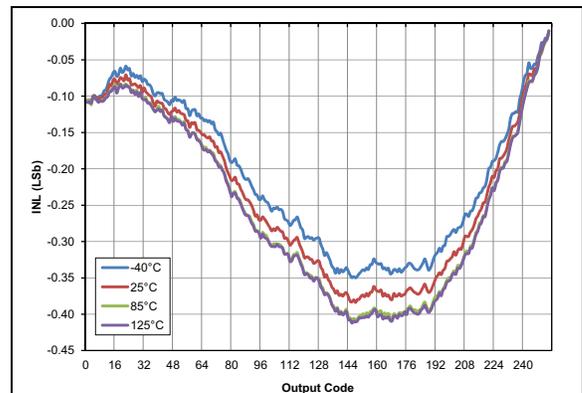


图36-72: 典型DAC INL误差, $V_{DD} = 5.0V$, V_{REF} = 外部5V, 仅限PIC16F18324/18344

PIC16(L)F18324/18344

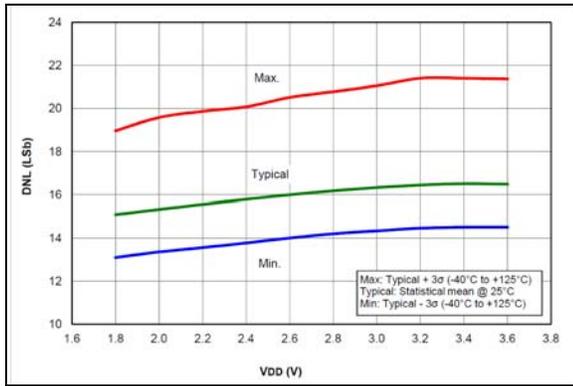


图36-73: DAC INL 误差, $V_{DD} = 3.0V$, 仅限 PIC16LF18324/18344

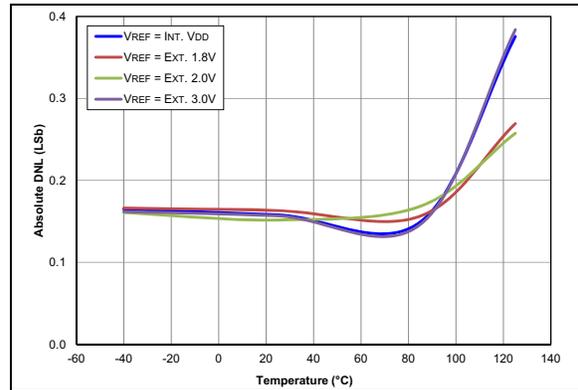


图36-74: DAC DNL 误差绝对值, $V_{DD} = 3.0V$, $V_{REF} = V_{DD}$, 所有器件

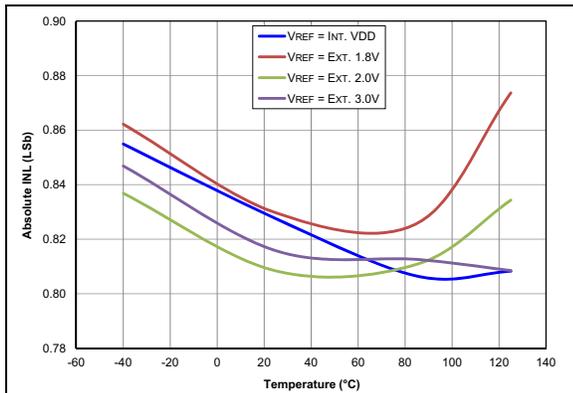


图36-75: DAC INL 误差绝对值, $V_{DD} = 3.0V$, $V_{REF} = V_{DD}$, 所有器件

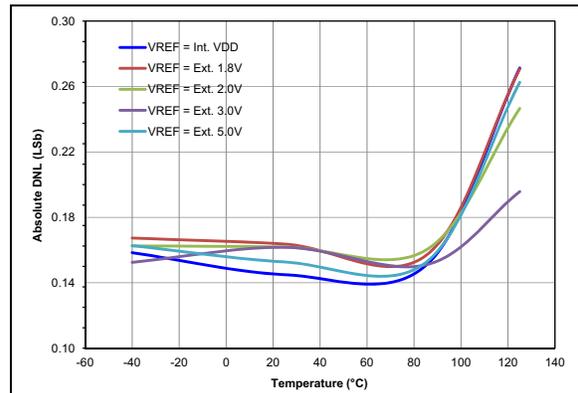


图36-76: DAC DNL 误差绝对值, $V_{DD} = 5.0V$, $V_{REF} = V_{DD}$, 仅限 PIC16F18324/18344

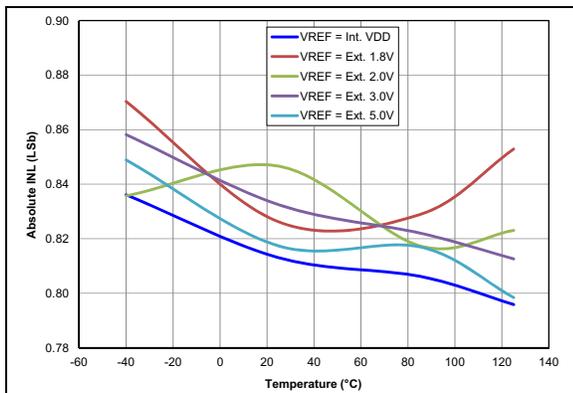


图36-77: DAC INL 误差绝对值, $V_{DD} = 5.0V$, $V_{REF} = V_{DD}$, 仅限 PIC16F18324/18344

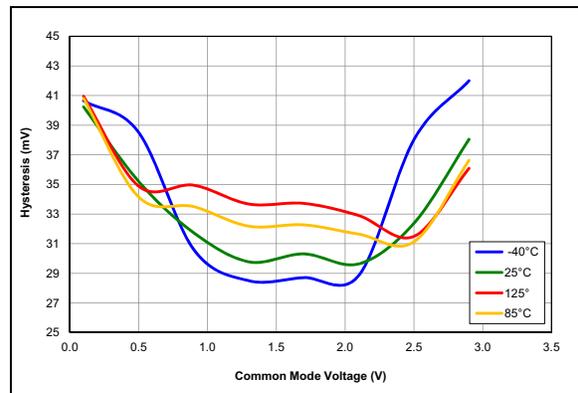


图36-78: 比较器滞后, 正常功耗模式 ($CxSP = 1$), $V_{DD} = 3.0V$, 典型测量值, 所有器件

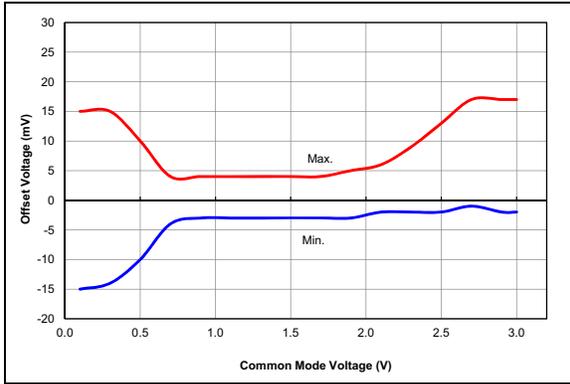


图36-79: 比较器失调电压, 正常功耗模式 (CxSP = 1), VDD = 3.0V, 25°C时典型测量值, 所有器件

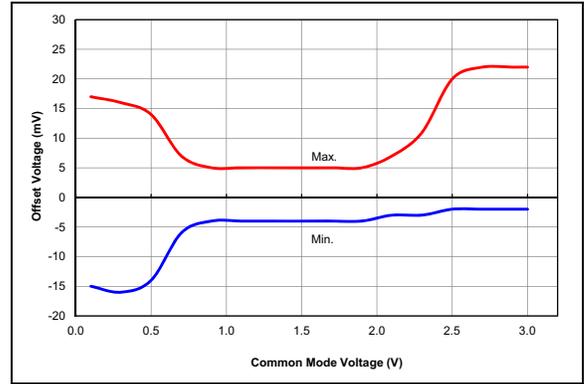


图36-80: 比较器失调电压, 正常功耗模式 (CxSP = 1), VDD = 3.0V, -40°C至125°C时典型测量值, 所有器件

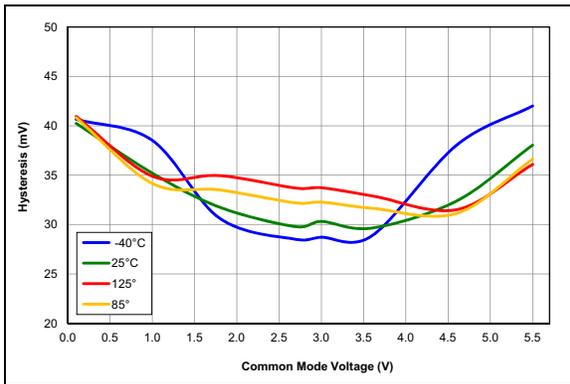


图36-81: 比较器滞后, 正常功耗模式 (CxSP = 1), VDD = 5.5V, 典型测量值, 仅限PIC16F18324/18344

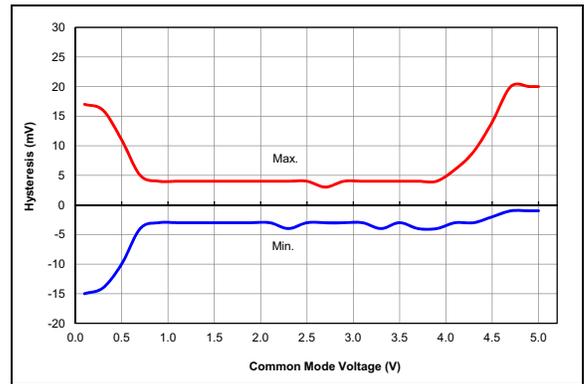


图36-82: 比较器失调电压, 正常功耗模式 (CxSP = 1), VDD = 5.0V, 25°C时典型测量值, 仅限PIC16F18324/18344

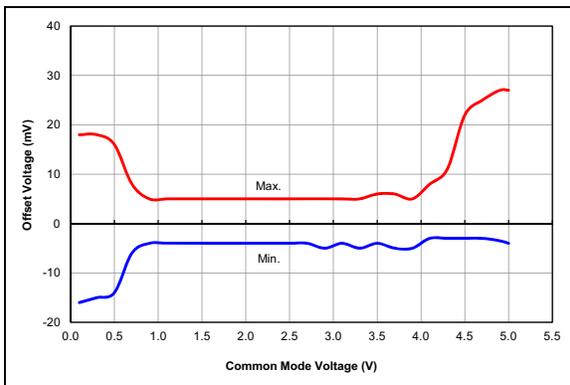


图36-83: 比较器失调电压, 正常功耗模式 (CxSP = 1), VDD = 5.5V, -40°C至125°C时典型测量值, 仅限PIC16F18324/18344

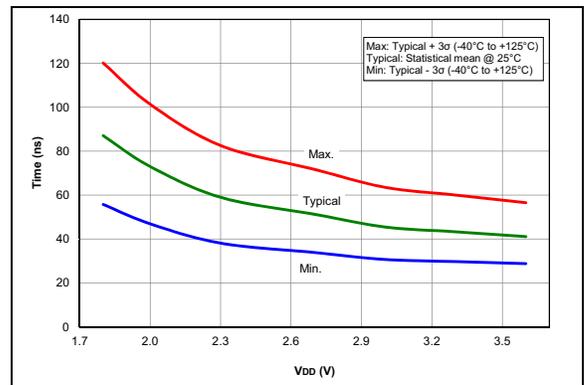


图36-84: 整个电压范围内的比较器响应时间, 正常功耗模式 (CxSP = 1), VDD = 5.5V, 典型测量值, 仅限PIC16LF18324/18344

PIC16(L)F18324/18344

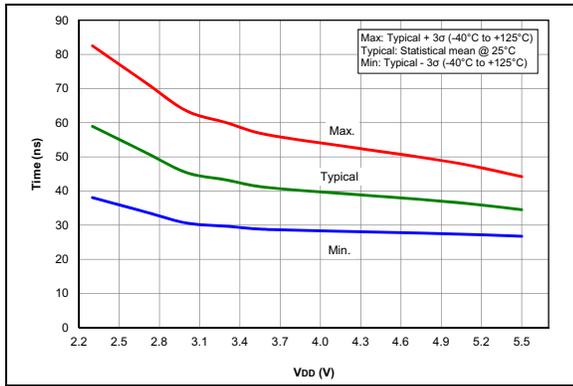


图36-85: 整个电压范围内的比较器响应时间, 正常功耗模式 ($CxSP = 1$), $V_{DD} = 5.5V$, 典型测量值, 仅限PIC16F18324/18344

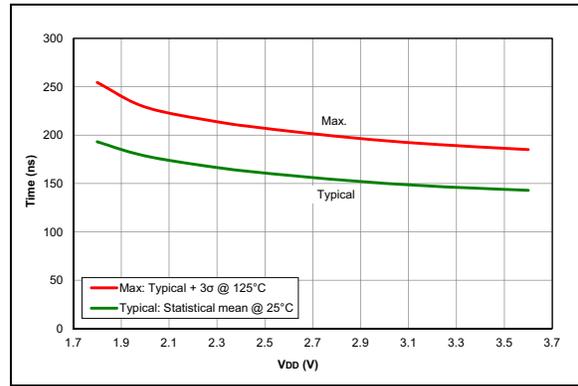


图36-86: 比较器响应时间下降沿, 仅限PIC16LF18324/18344

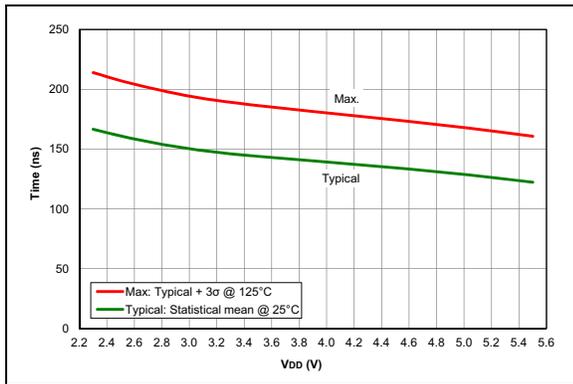


图36-87: 比较器响应时间下降沿, 仅限PIC16F18324/18344

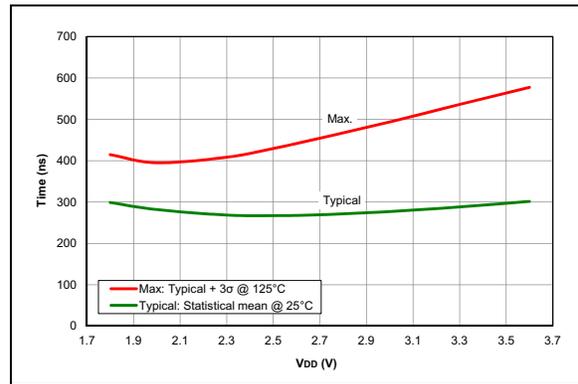


图36-88: 比较器响应时间上升沿, 仅限PIC16LF18324/18344

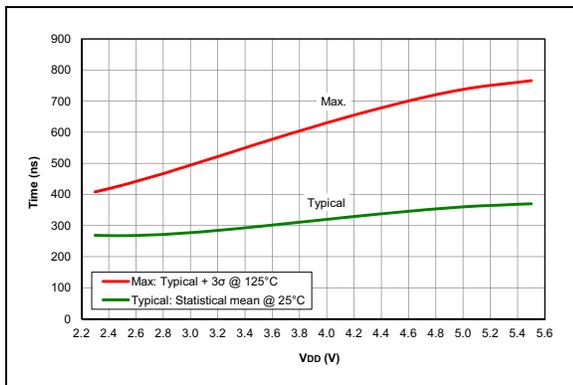


图36-89: 比较器响应时间上升沿, 仅限PIC16F18324/18344

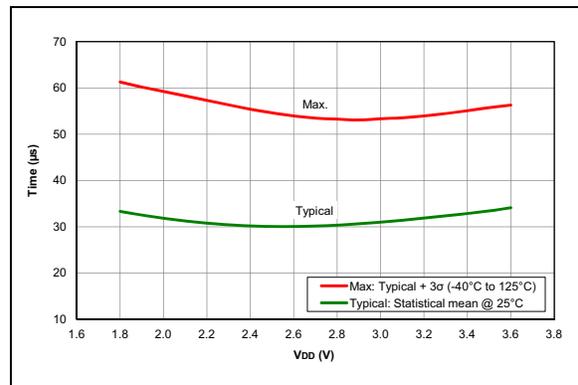


图36-90: 带隙就绪时间, 仅限PIC16LF18324/18344

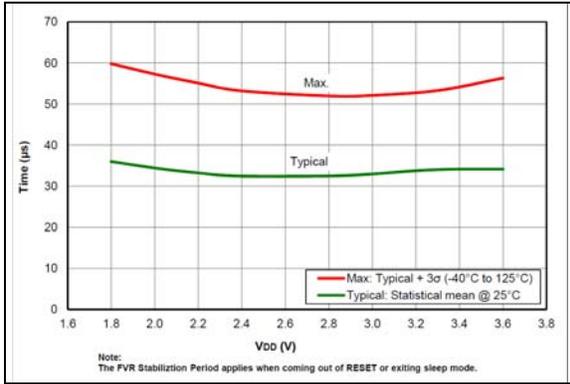


图36-91: FVR稳定周期, 仅限 PIC16LF18324/18344

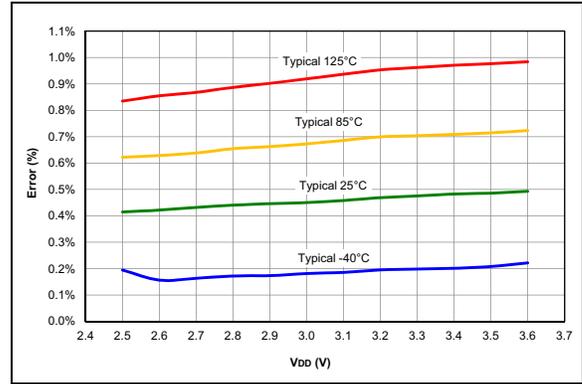


图36-92: 典型FVR电压 (1x), 仅限 PIC16LF18324/18344

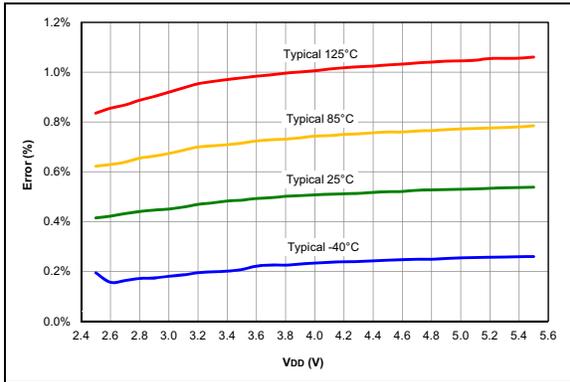


图36-93: FVR电压误差 (1x), 仅限 PIC16F18324/18344

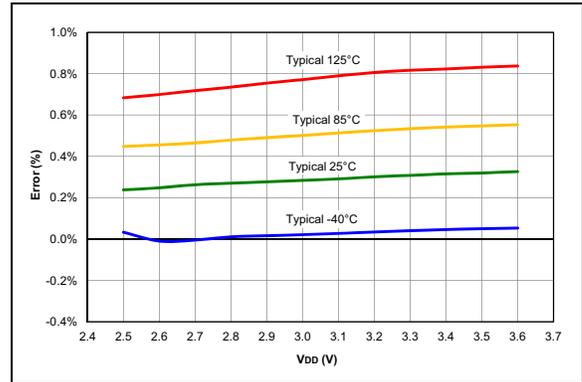


图36-94: FVR电压误差 (2x), 仅限 PIC16LF18324/18344

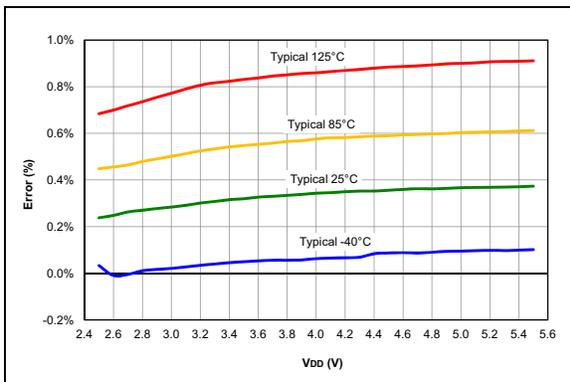


图36-95: FVR电压误差 (2x), 仅限 PIC16F18324/18344

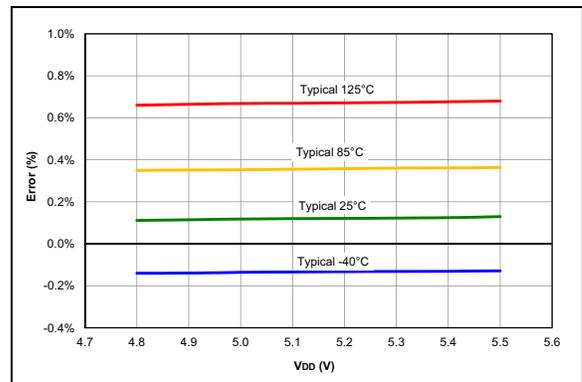


图36-96: FVR电压误差 (4x), 仅限 PIC16F18324/18344

PIC16(L)F18324/18344

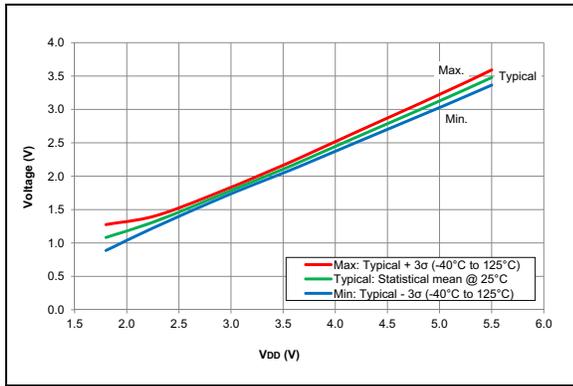


图36-97: 施密特触发器高电平电压值, 所有器件

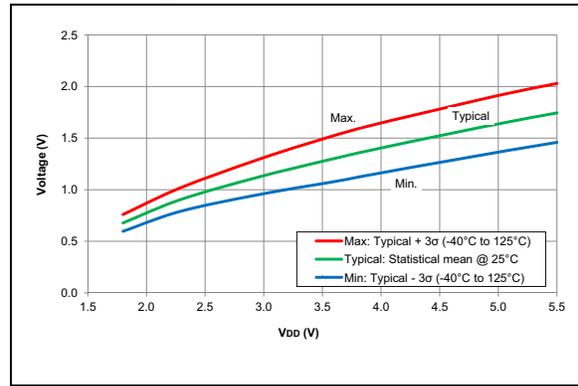


图36-98: 施密特触发器低电平电压值, 所有器件

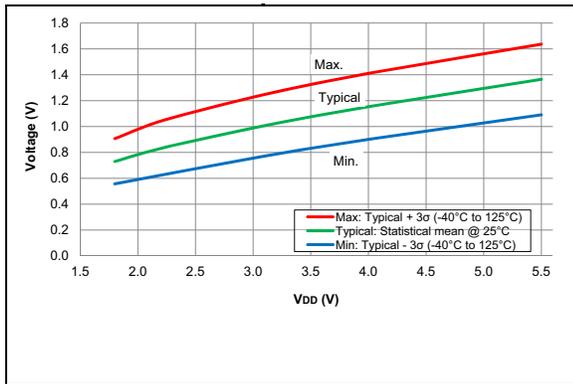


图36-99: TTL跳变阈值, 所有器件

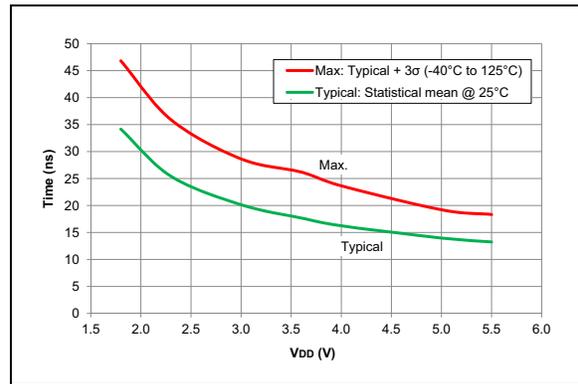


图36-100: 上升时间, 使能压摆率控制, 所有器件

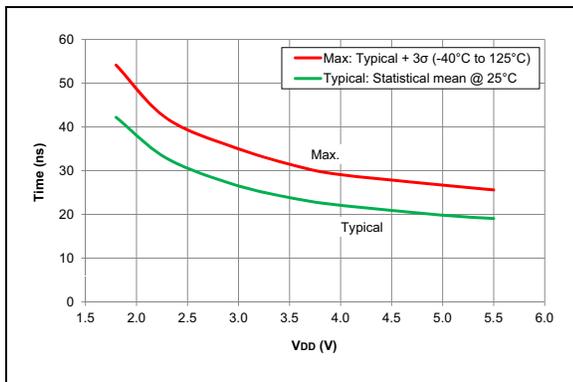


图36-101: 下降时间, 使能压摆率控制, 所有器件

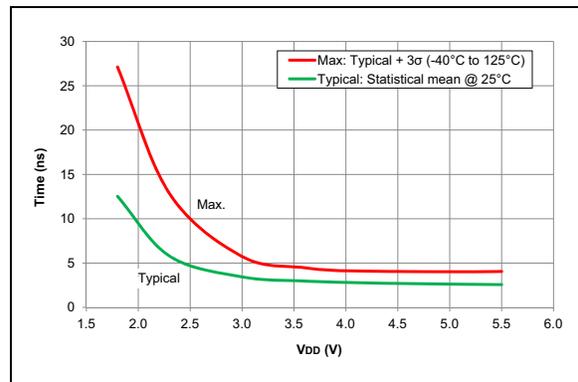


图36-102: 上升时间, 禁止压摆率控制, 所有器件

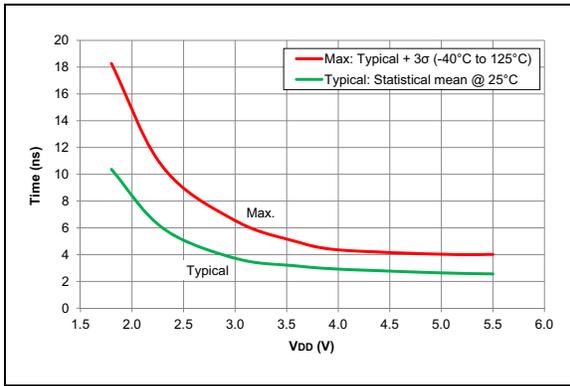


图36-103: 下降时间, 禁止压摆率控制, 所有器件

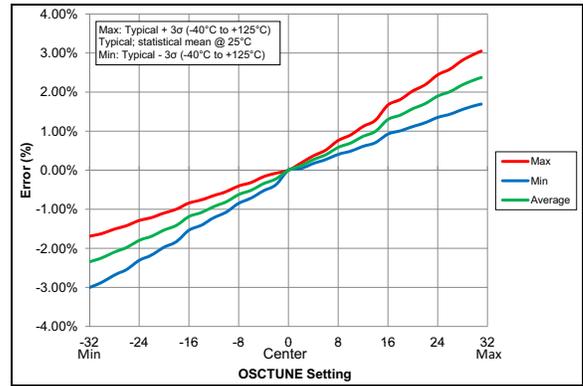


图36-104: OSCTUNE中心频率, 仅限PIC16LF18324/18344

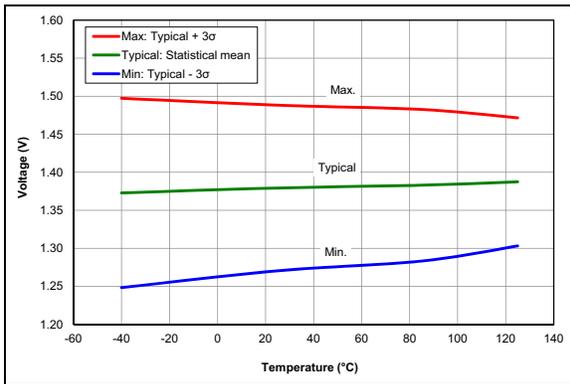


图36-105: POR释放电压, 所有器件

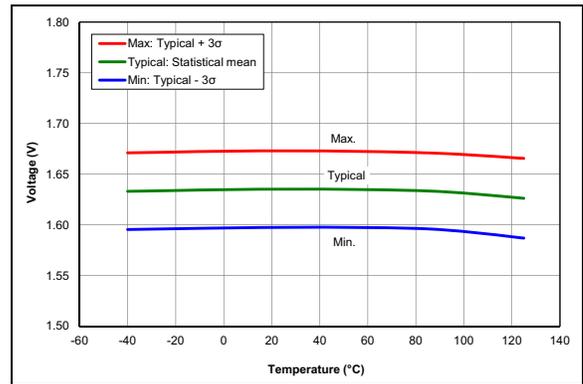


图36-106: POR重新激活电压, 正常功耗模式, 仅限PIC16F18324/18344

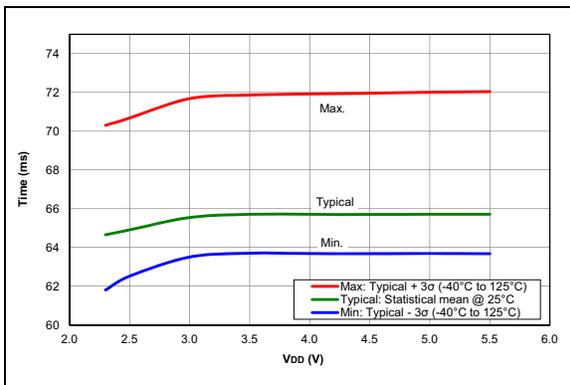


图36-107: PWRT周期, 仅限PIC16F18324/18344

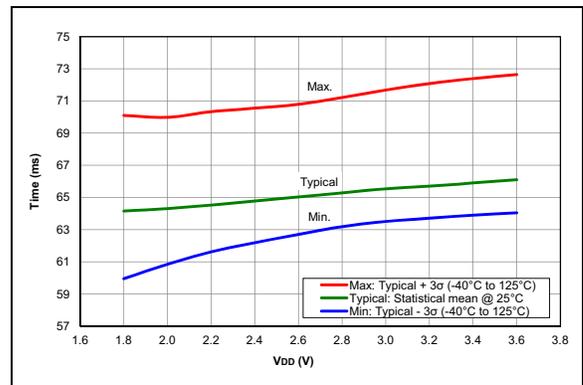


图36-108: PWRT周期, 仅限PIC16LF18324/18344

PIC16(L)F18324/18344

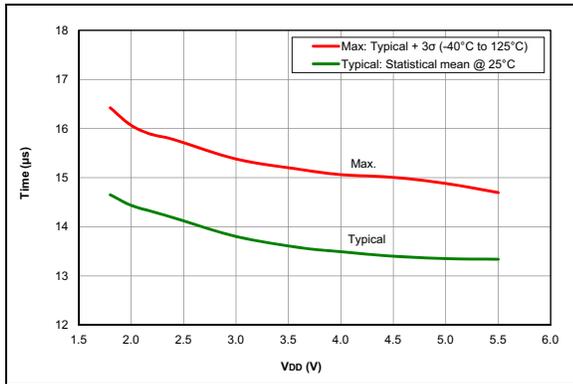


图36-109: 从休眠模式唤醒, VREGPM = 0, HFINTOSC = 4 MHz, 仅限PIC16F18324/18344

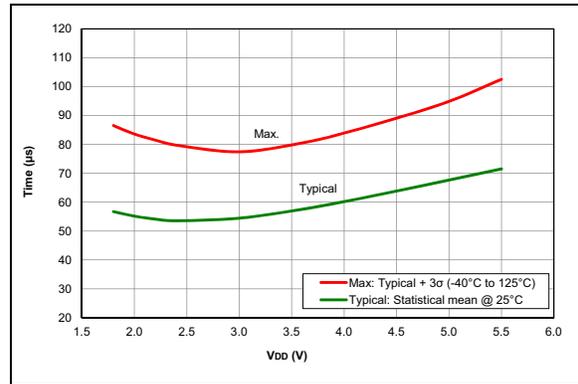


图36-110: ULP, 从休眠模式唤醒, VREGPM = 1, HFINTOSC = 4 MHz, 仅限PIC16F18324/18344

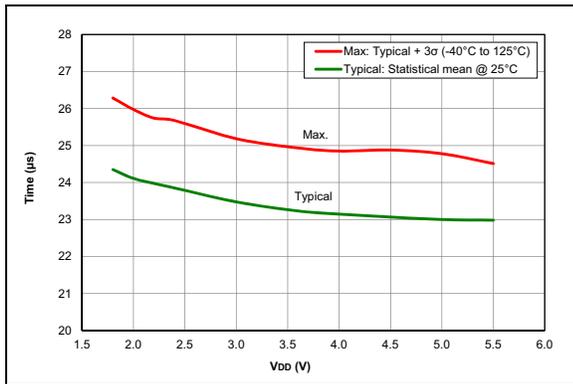


图36-111: 从休眠模式唤醒, VREGPM = 1, HFINTOSC = 16 MHz, 仅限PIC16F18324/18344

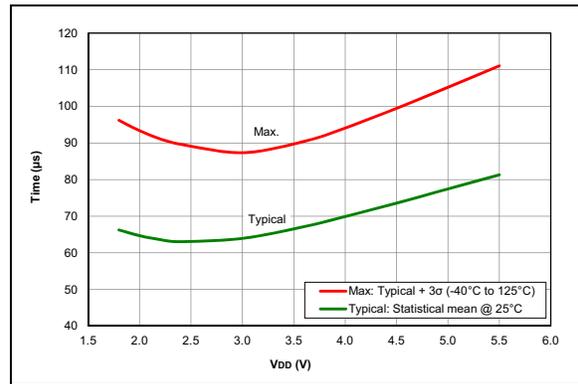


图36-112: ULP, 从休眠模式唤醒, VREGPM = 1, HFINTOSC = 16 MHz, 仅限PIC16F18324/18344

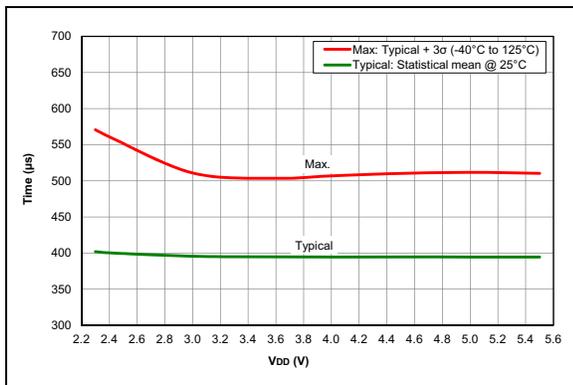


图36-113: 从休眠模式唤醒, VREGPM = 1, LFINTOSC, 仅限PIC16F18324/18344

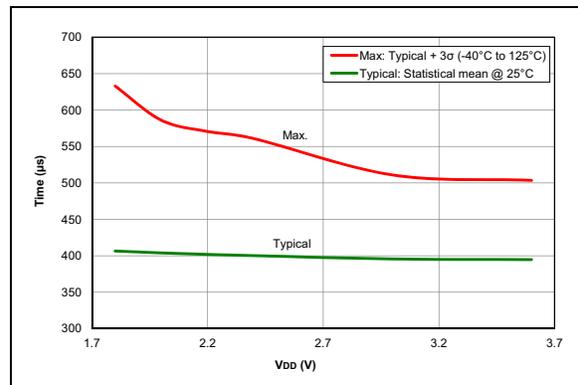


图36-114: 从休眠模式唤醒, LFINTOSC, 仅限PIC16LF18324/18344

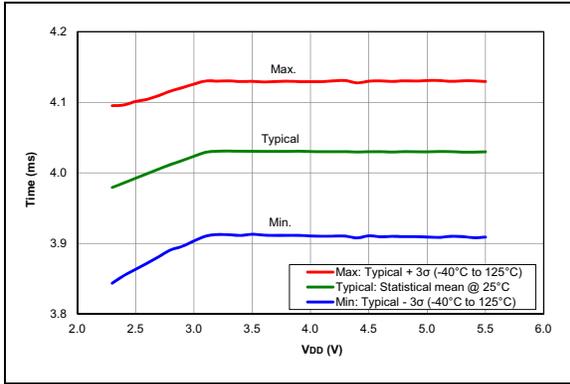


图36-115: WDT超时周期, 仅限
PIC16F18324/18344

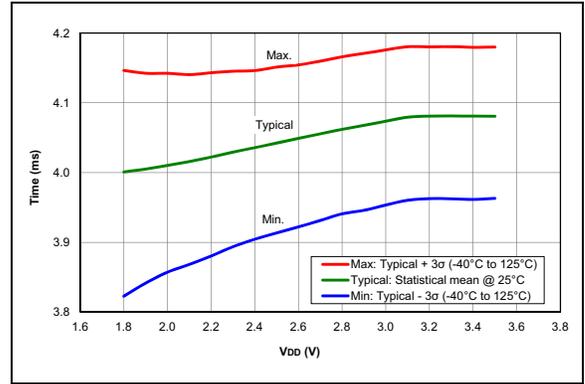


图36-116: WDT超时周期, 仅限
PIC16LF18324/18344

37.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机（MCU）和 dsPIC® 数字信号控制器（DSC）提供支持：

- 集成开发环境
 - MPLAB® X IDE 软件
 - MPLAB Xpress IDE 软件
 - Microchip 代码配置器（Microchip Code Configurator, MCC）
- 编译器 / 汇编器 / 链接器
 - MPLAB XC 编译器
 - MPASM™ 汇编器
 - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
 - 适用于各种器件系列的 MPLAB 汇编器 / 链接器 / 库管理器
- 模拟器
 - MPLAB X SIM 软件模拟器
- 仿真器
 - MPLAB REAL ICE™ 在线仿真器
- 在线调试器 / 编程器
 - MPLAB ICD 3
 - PICKit™ 3
- 器件编程器
 - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包
- 第三方开发工具

37.1 MPLAB X 集成开发环境软件

MPLAB X IDE 是适用于 Microchip 和第三方软硬件开发工具统一的通用图形用户界面，可以在 Windows®、Linux 和 Mac OS® X 上运行。MPLAB X IDE 是一款全新的 IDE，它基于 NetBeans IDE，包含许多免费的软件组件和插件，适用于高性能的应用程序开发和调试。通过这一无缝交互的用户界面，在不同工具之间的迁移以及从软件模拟器到硬件调试和编程工具的升级都变得极为简便。

MPLAB X IDE 具有完善的项目管理、可视化的调用图、可配置的观察窗口以及包含代码补全功能和上下文菜单的功能丰富编辑器，因此对于新用户来说非常灵活和友好。MPLAB X IDE 支持对多个项目使用多个工具和同时调试，因此也完全可以满足经验丰富用户的需求。

功能丰富的编辑器：

- 彩色高亮显示语法
- 智能代码补全功能，在输入代码时提供建议和提示
- 基于用户定义规则，代码自动格式化
- 即时解析

用户友好的可定制界面：

- 完全可定制界面：工具栏、工具栏图标、窗口和窗口放置等
- 调用图窗口

基于项目的工作空间：

- 多个项目
- 多个工具
- 多种配置
- 同时调试会话

文件历史和错误跟踪：

- 本地文件历史功能
- 内建对 Bugzilla 缺陷跟踪系统的支持

37.2 MPLAB XC 编译器

MPLAB XC编译器是适用于Microchip所有8位、16位和32位MCU以及DSC器件的完全ANSI C编译器。这些编译器提供强大的集成功能以及出色的代码优化功能，且易于使用。MPLAB XC编译器可在Windows、Linux或Mac OS X上运行。

为方便进行源代码级调试，编译器提供了已针对MPLAB X IDE优化的调试信息。

MPLAB XC编译器的免费版支持所有器件和命令，没有时间或存储容量限制，且为大多数应用程序提供了充分的代码优化。

MPLAB XC编译器包含汇编器、链接器和实用程序。汇编器生成可重定位目标文件，然后通过链接器将生成的可重定位目标文件与其他可重定位目标文件或归档文件归档或链接在一起，进而生成可执行文件。MPLAB XC编译器使用汇编器来生成目标文件。汇编器具有如下突出特性：

- 支持全部器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的伪指令集
- 灵活的宏语言
- 与MPLAB X IDE兼容

37.3 MPASM 汇编器

MPASM汇编器是全功能通用宏汇编器，适用于PIC10/12/16/18 MCU。

MPASM汇编器可生成用于MPLINK目标链接器的可重定位目标文件、Intel®标准HEX文件、详细描述存储器使用状况和符号参考的MAP文件、包含源代码行及生成机器码的绝对LST文件以及用于调试的COFF文件。

MPASM汇编器具有如下特性：

- 集成在MPLAB X IDE项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

37.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK目标链接器组合由MPASM汇编器生成的可重定位目标文件。通过使用链接器脚本中的伪指令，它还可链接预编译库中的可重定位目标文件。

MPLIB目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器/库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起来增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

37.5 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB汇编器为PIC24和PIC32 MCU以及dsPIC DSC器件从符号汇编语言生成可重定位机器码。MPLAB XC编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特性：

- 支持整个器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的指令集
- 与MPLAB X IDE兼容

37.6 MPLAB X SIM 软件模拟器

MPLAB X SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC DSC 进行模拟，可在 PC 主机环境下进行代码开发。对于任何给定的指令，都可以对数据区进行检查或修改，并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中，以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器。

MPLAB X SIM 软件模拟器完全支持使用 MPLAB XC 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

37.7 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件推出的新一代高速仿真器。结合 MPLAB X IDE 易于使用且功能强大的图形用户界面，该仿真器可对所有 8 位、16 位和 32 位 MCU 及 DSC 器件进行调试和编程。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器和新型抗噪声、高速低压差分信号 (LVDS) 互连电缆 (CAT5) 与目标板相连。

可通过 MPLAB X IDE 下载将来版本的固件，对该仿真器进行现场升级。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：全速仿真、运行时变量观察、跟踪分析、复杂断点、逻辑探针、耐用的探针接口及较长（长达 3 米）的互连电缆。

37.8 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 的闪存 DSC 和 MCU 器件。结合 MPLAB X IDE 功能强大但易于使用的图形用户界面，该调试器可对 PIC 闪存单片机和 dsPIC DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器和目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 连接器。

37.9 PICKit 3 在线调试器 / 编程器

结合 MPLAB X IDE 功能强大的图形用户界面，MPLAB PICKit 3 可对 PIC 闪存单片机和 dsPIC 数字信号控制器进行调试和编程，且价位较低。MPLAB PICKit 3 通过全速 USB 接口与设计工程师的 PC 相连，并利用 Microchip 调试连接器 (RJ-11) (与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容) 与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程 (In-Circuit Serial Programming™, ICSP™)。

PICKit 3 Debug Express 包括 PICKit 3、演示板和单片机、连接电缆和光盘 (内含用户指南、课程、教程、编译器和 MPLAB IDE 软件)。

37.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器 (128 x 64)，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC MCU 和 dsPIC DSC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡，用于文件存储及数据应用。

37.11 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVAL® 评估系统、 Σ - Δ ADC、流速传感器，等等。

同时还提供入门工具包，其中包含体验指定器件功能所需的所有软硬件。通常提供单个应用以及调试功能，都包含在一块电路板上。

有关演示、开发和评估工具包的完整列表，请访问 Microchip 网站 (www.microchip.com)。

37.12 第三方开发工具

Microchip 还提供一些来自第三方供应商的优秀开发工具。这些工具均经过精心挑选，功能独特，物有所值。

- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 ILogika® 等公司提供的嵌入式以太网解决方案

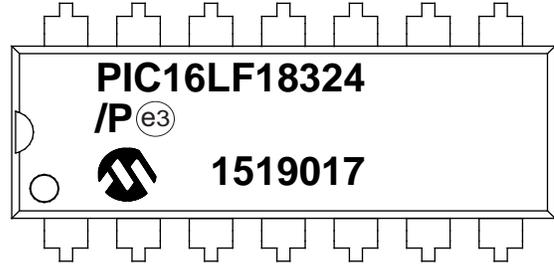
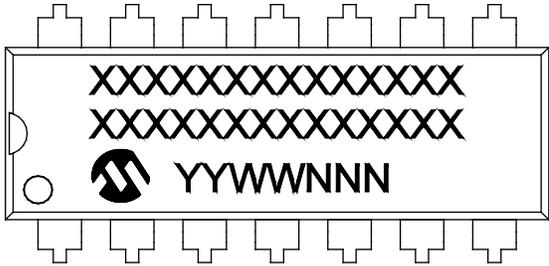
PIC16(L)F18324/18344

38.0 封装信息

38.1 封装标识信息

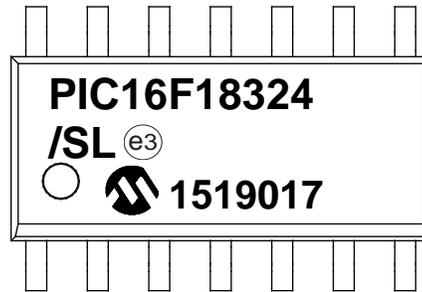
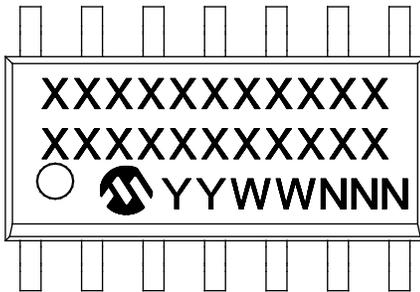
14引脚PDIP (300 mil)

示例



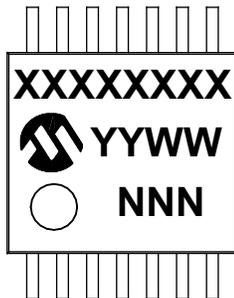
14引脚SOIC (3.90 mm)

示例



14引脚TSSOP (4.4 mm)

示例



图注:

XX...X	客户指定信息
Y	年份代码 (日历年的最后一位数字)
YY	年份代码 (日历年的最后两位数字)
WW	星期代码 (一月一日的星期代码为“01”)
NNN	以字母数字排序的追踪代码
^{e3}	雾锡 (Matte Tin, Sn) 的JEDEC无铅标志
*	表示无铅封装。JEDEC无铅标志 (^{e3}) 标示于此种封装的外包装上。

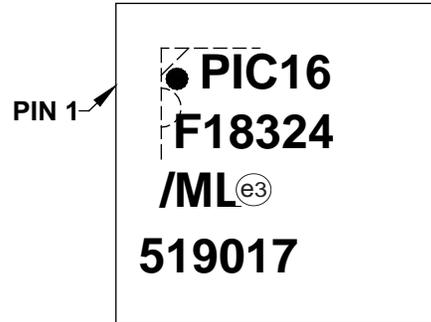
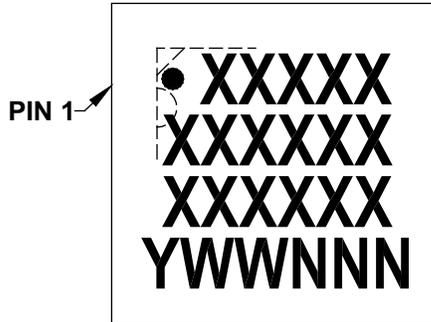
注: Microchip部件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户指定信息的字符数。

PIC16(L)F18324/18344

封装标识信息（续）

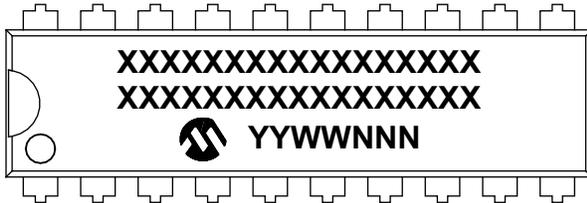
16引脚UQFN（4x4x0.5mm）

示例



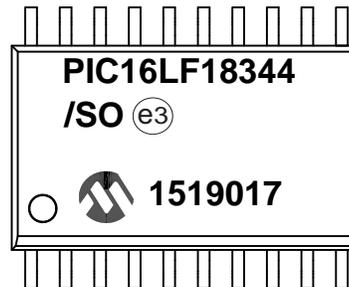
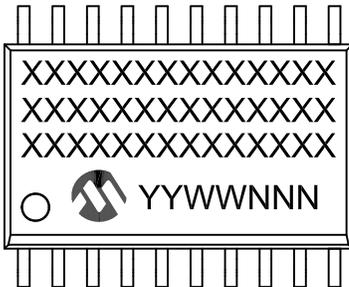
20引脚PDIP（300 mil）

示例



20引脚SOIC（7.50 mm）

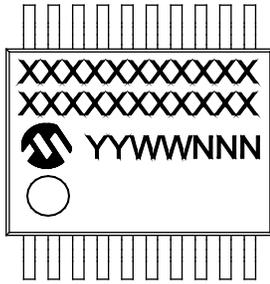
示例



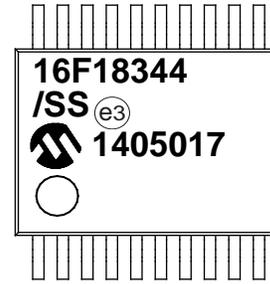
图注:	XX...X	客户指定信息
	Y	年份代码（日历年的最后一位数字）
	YY	年份代码（日历年的最后两位数字）
	WW	星期代码（一月一日的星期代码为“01”）
	NNN	以字母数字排序的追踪代码
	(e3)	雾锡（Matte Tin, Sn）的JEDEC无铅标志
	*	表示无铅封装。JEDEC无铅标志（(e3)）标示于此种封装的外包装上。
注:	Microchip部件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户指定信息的字符数。	

封装标识信息 (续)

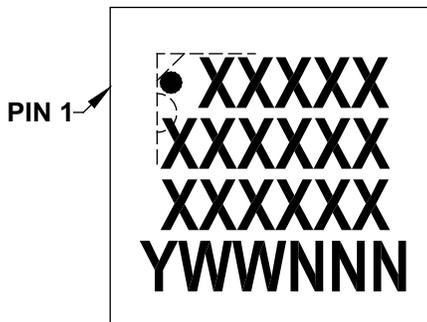
20引脚SSOP (5.30 mm)



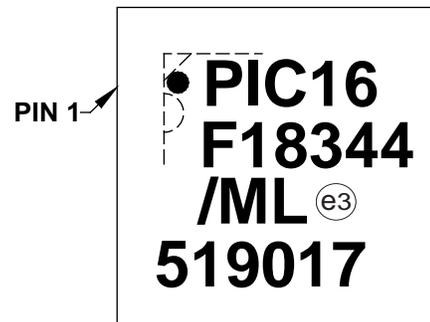
示例



20引脚UQFN (4x4x0.5 mm)



示例



图注:

- XX...X 客户指定信息
- Y 年份代码 (日历年的最后一位数字)
- YY 年份代码 (日历年的最后两位数字)
- WW 星期代码 (一月一日的星期代码为“01”)
- NNN 以字母数字排序的追踪代码
- (e3) 雾锡 (Matte Tin, Sn) 的 JEDEC 无铅标志
- * 表示无铅封装。JEDEC 无铅标志 ((e3)) 标示于此种封装的外包装上。

注: Microchip 部件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户指定信息的字符数。

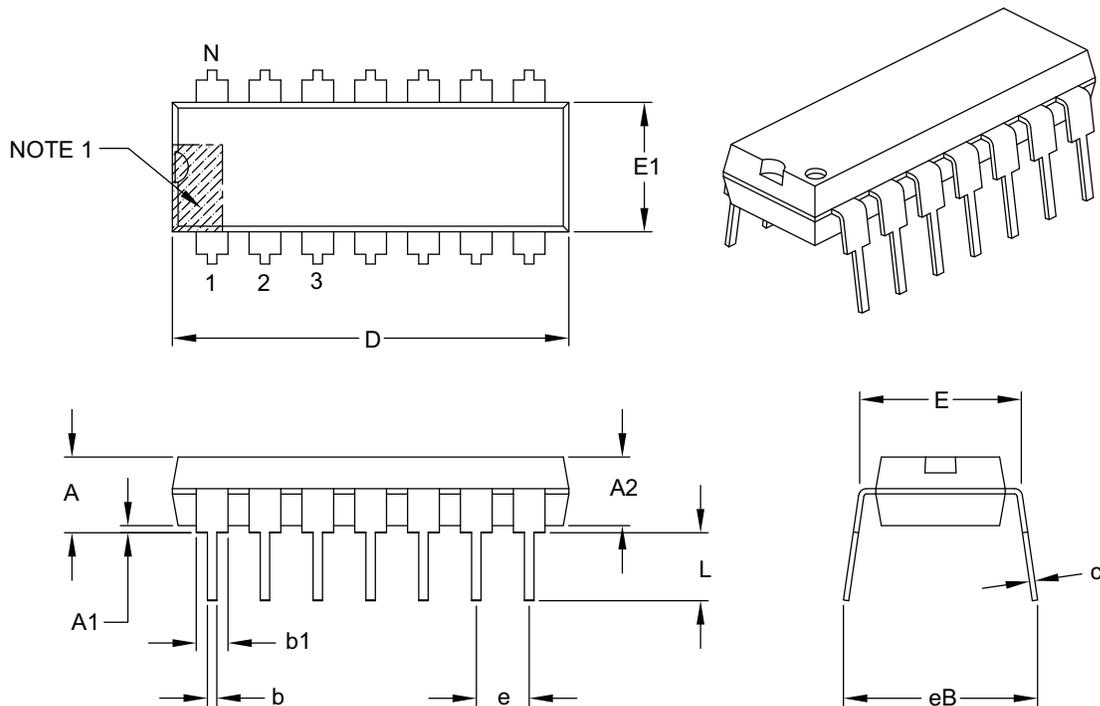
PIC16(L)F18324/18344

38.2 封装详细信息

以下部分将介绍各种封装的技术细节。

14引脚塑封双列直插式封装 (P) —— 主体 300 mil [PDIP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

- Pin 1 visual index feature may vary, but must be located with the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

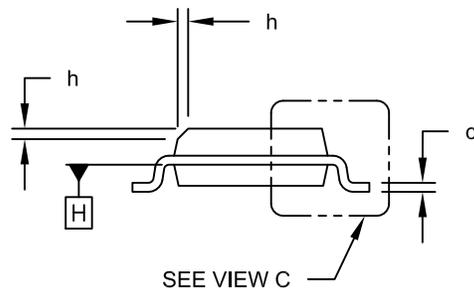
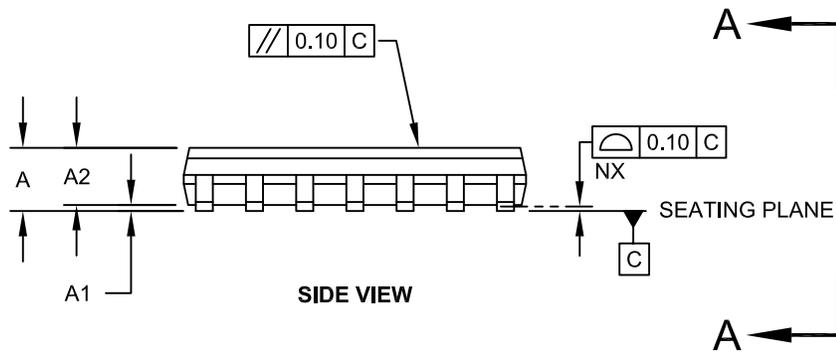
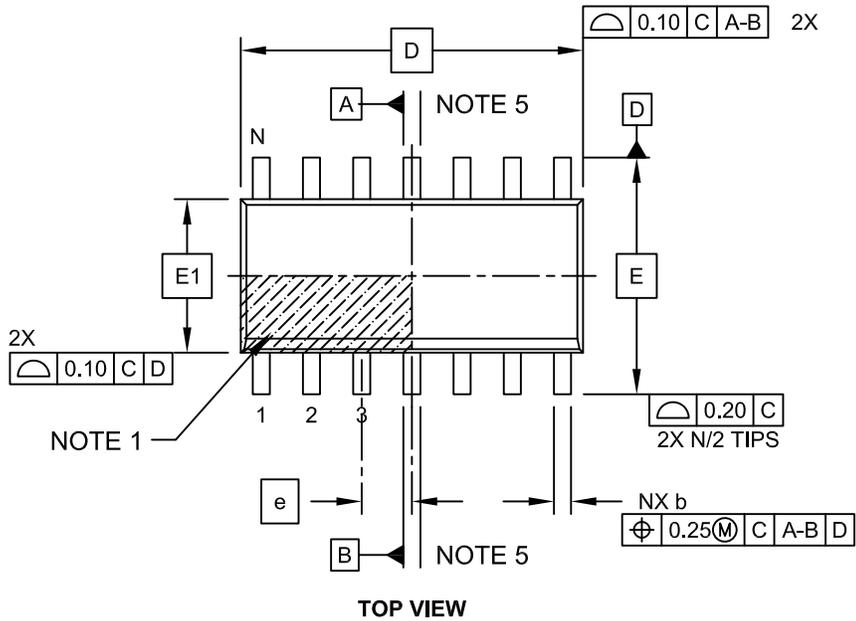
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

PIC16(L)F18324/18344

14 引脚塑封窄条小外形封装 (SL) —— 主体 3.90 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

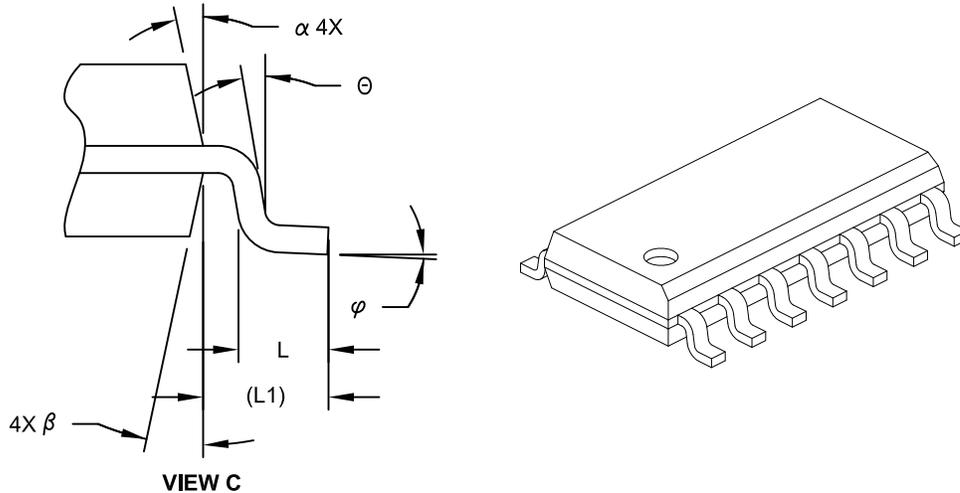


Microchip Technology Drawing No. C04-065C Sheet 1 of 2

PIC16(L)F18324/18344

14引脚塑封窄条小外形封装（SL）——主体3.90 mm [SOIC]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	8.65 BSC		
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1	1.04 REF		
Lead Angle	θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.10	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

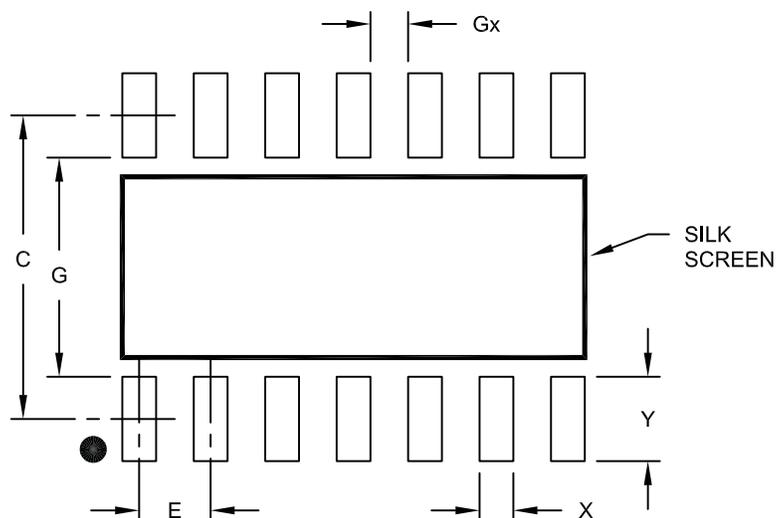
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2

14 引脚塑封窄条小外形封装 (SL) —— 主体 3.90 mm [SOIC]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		5.40	
Contact Pad Width	X			0.60
Contact Pad Length	Y			1.50
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	3.90		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

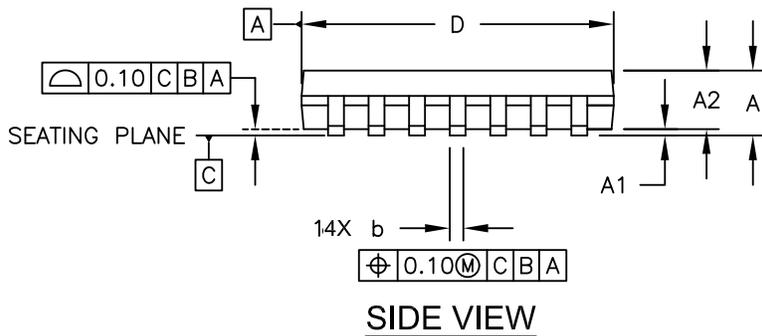
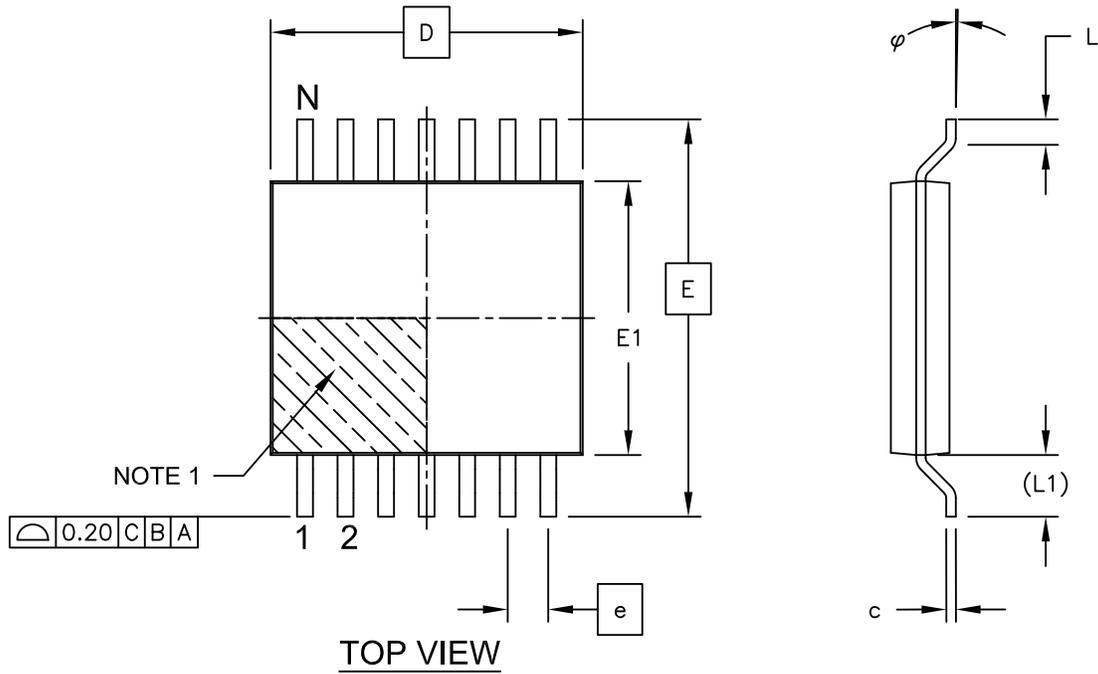
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

PIC16(L)F18324/18344

14引脚塑封薄型紧缩小外形封装（ST）——主体4.4 mm [TSSOP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。

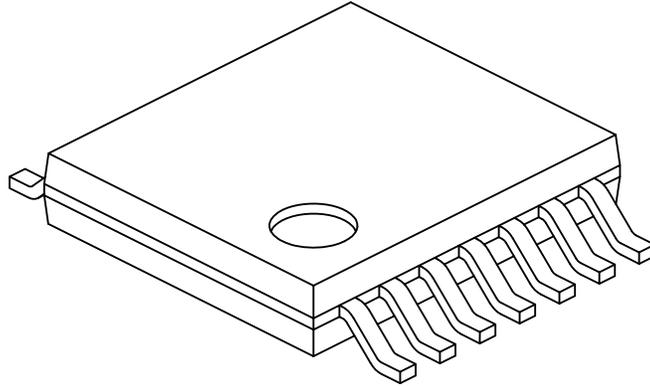


Microchip Technology Drawing C04-087C Sheet 1 of 2

PIC16(L)F18324/18344

14 引脚塑封薄型紧缩小外形封装 (ST) —— 主体 4.4 mm [TSSOP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	0.65 BSC		
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	-	0.15
Overall Width	E	6.40 BSC		
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	(L1)	1.00 REF		
Foot Angle	ϕ	0°	-	8°
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.19	-	0.30

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

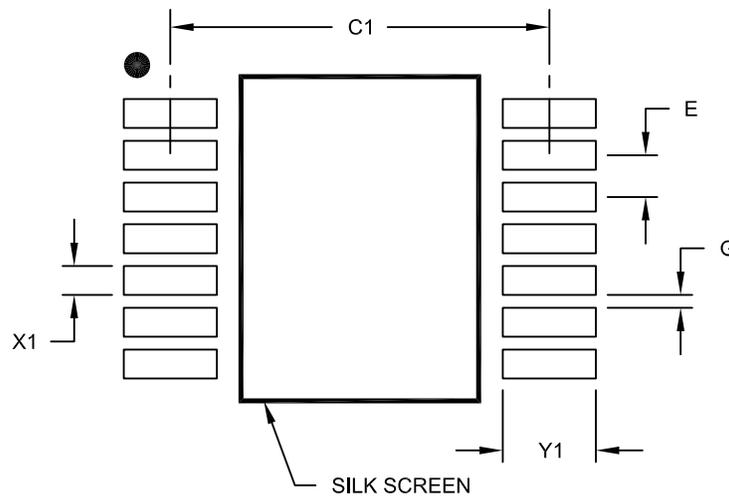
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

PIC16(L)F18324/18344

14 引脚塑封薄型紧缩小外形封装 (ST) —— 主体 4.4 mm [TSSOP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	C1		5.90	
Contact Pad Width (X14)	X1			0.45
Contact Pad Length (X14)	Y1			1.45
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

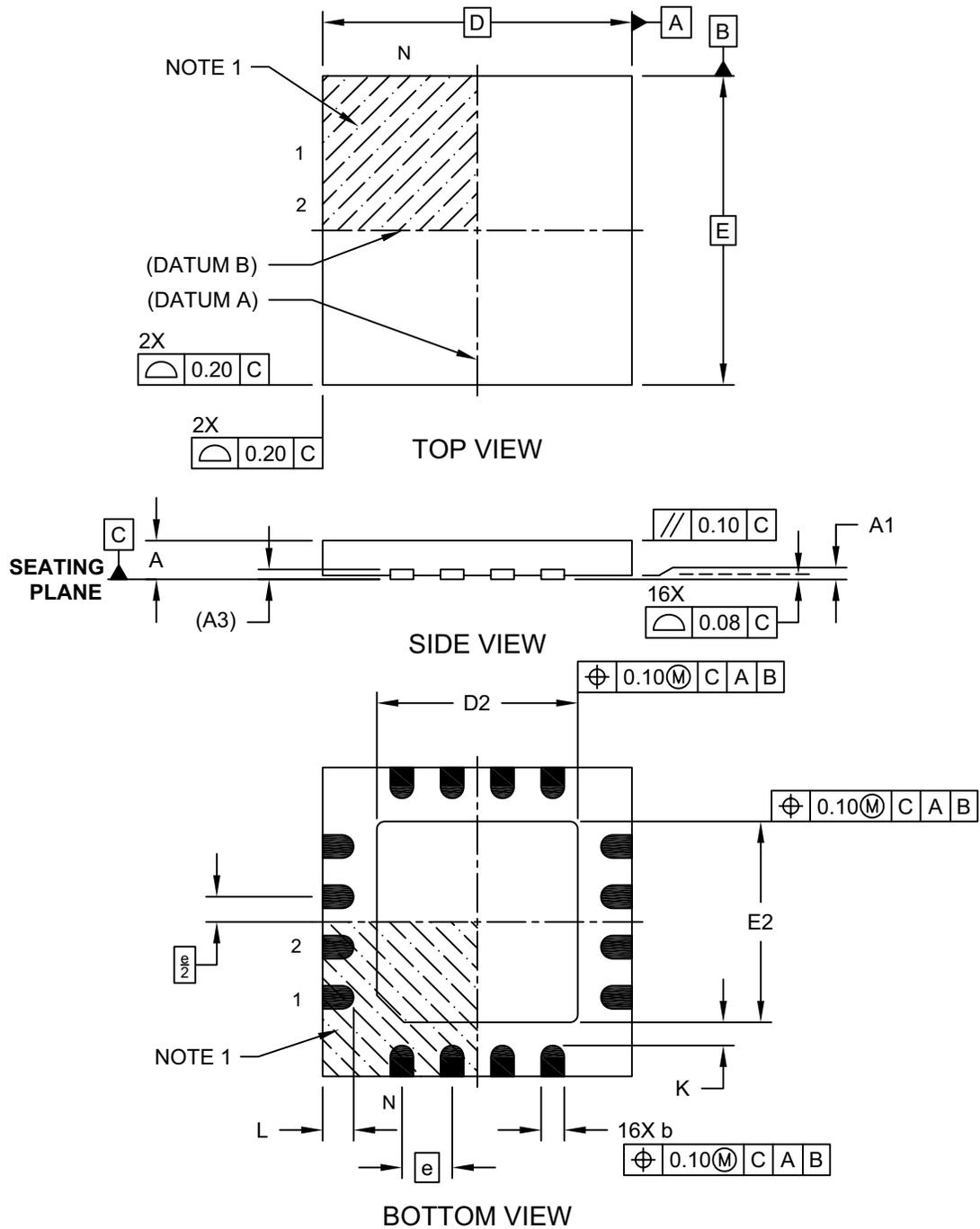
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

PIC16(L)F18324/18344

16 引脚塑封超薄正方扁平无脚封装 (JQ) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

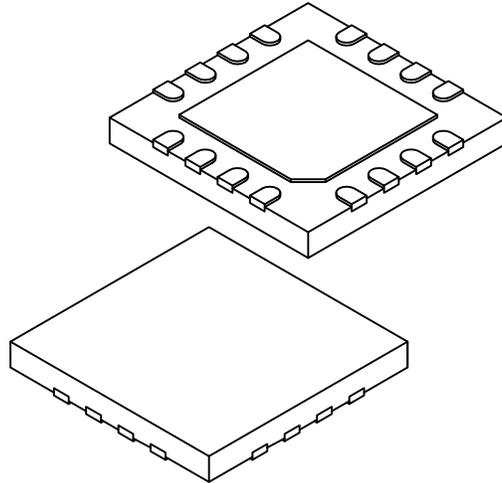


Microchip Technology Drawing C04-257A Sheet 1 of 2

PIC16(L)F18324/18344

16 引脚塑封超薄正方扁平无脚封装 (JQ) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	16		
Pitch	e	0.65 BSC		
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.127 REF		
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.50	2.60	2.70
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.50	2.60	2.70
Terminal Width	b	0.25	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

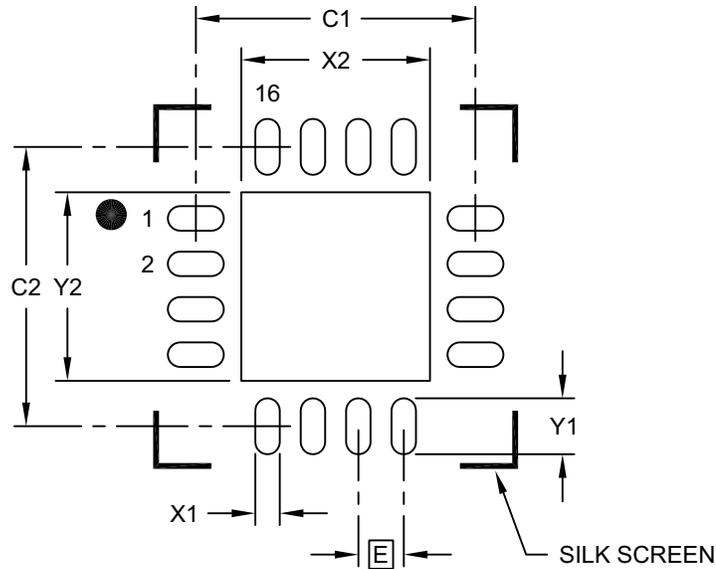
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-257A Sheet 2 of 2

PIC16(L)F18324/18344

16 引脚塑封超薄正方扁平无脚封装 (JQ) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	X2			2.70
Optional Center Pad Length	Y2			2.70
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X16)	X1			0.35
Contact Pad Length (X16)	Y1			0.80

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

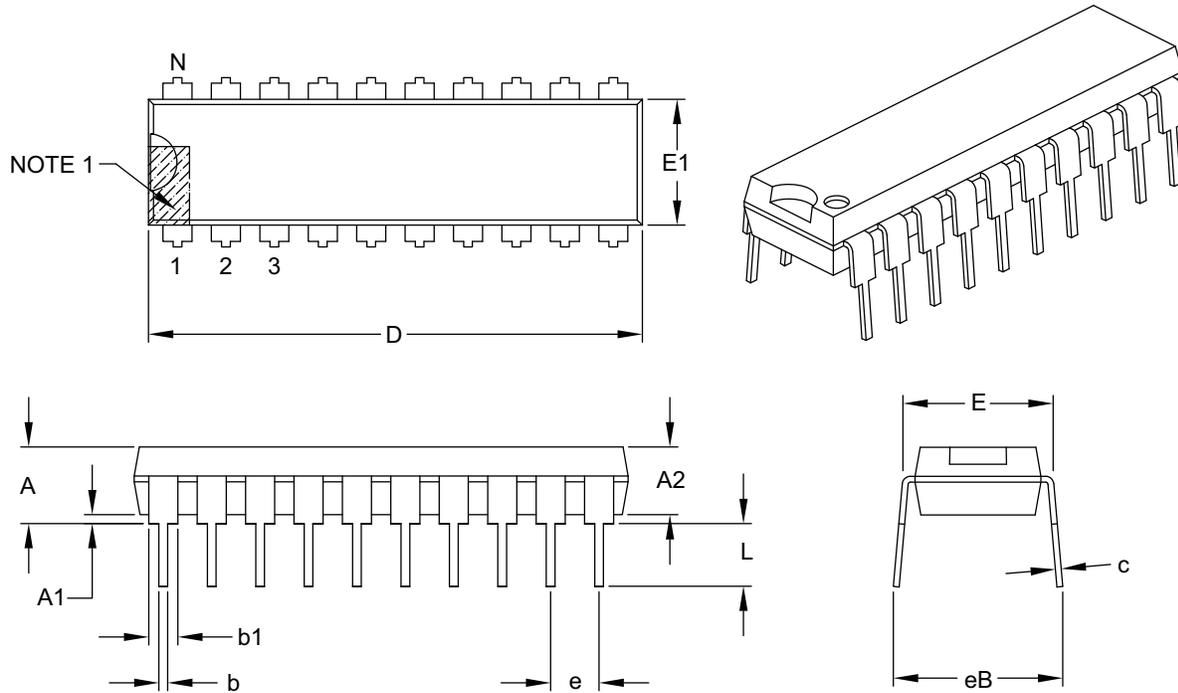
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2257A

PIC16(L)F18324/18344

20 引脚塑封双列直插式封装 (P) —— 主体 300 mil [PDIP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



		Units	INCHES		
Dimension Limits			MIN	NOM	MAX
Number of Pins	N		20		
Pitch	e		.100 BSC		
Top to Seating Plane	A	–	–	.210	
Molded Package Thickness	A2	.115	.130	.195	
Base to Seating Plane	A1	.015	–	–	
Shoulder to Shoulder Width	E	.300	.310	.325	
Molded Package Width	E1	.240	.250	.280	
Overall Length	D	.980	1.030	1.060	
Tip to Seating Plane	L	.115	.130	.150	
Lead Thickness	c	.008	.010	.015	
Upper Lead Width	b1	.045	.060	.070	
Lower Lead Width	b	.014	.018	.022	
Overall Row Spacing §	eB	–	–	.430	

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

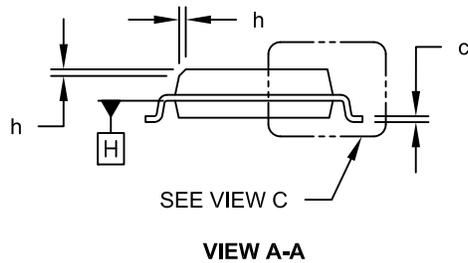
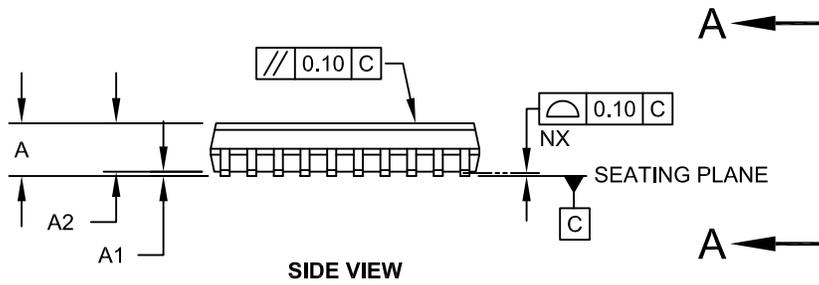
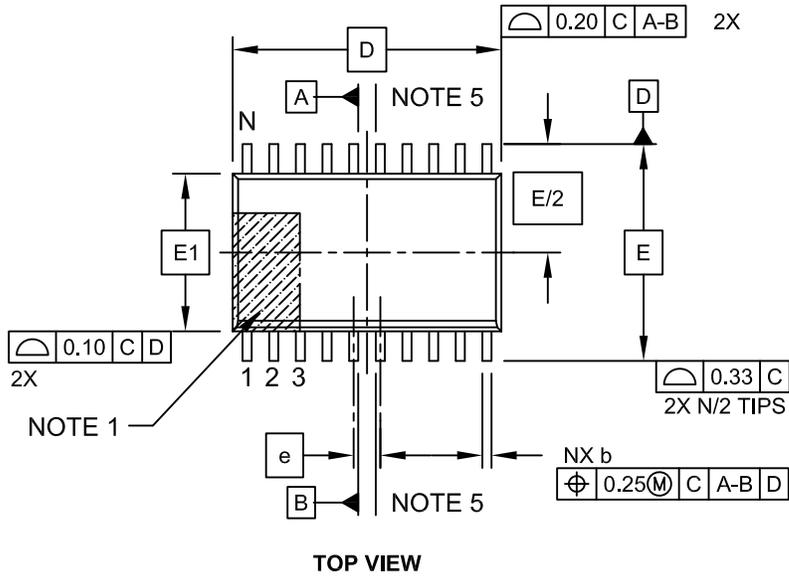
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B

PIC16(L)F18324/18344

20 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至<http://www.microchip.com/packaging> 查看 Microchip 封装规范。

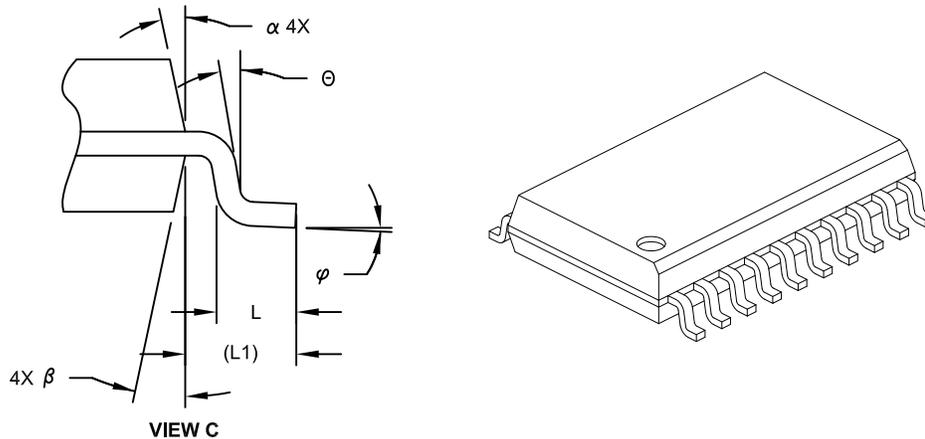


Microchip Technology Drawing C04-094C Sheet 1 of 2

PIC16(L)F18324/18344

20引脚塑封宽条小外形封装（SO）——主体7.50 mm [SOIC]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	12.80 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

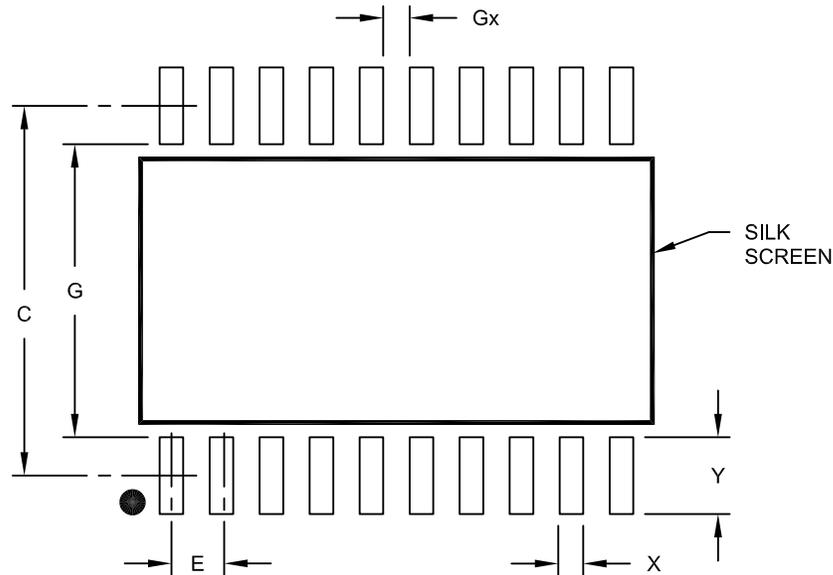
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

PIC16(L)F18324/18344

20 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
		MIN	NOM	MAX
Dimension Limits				
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

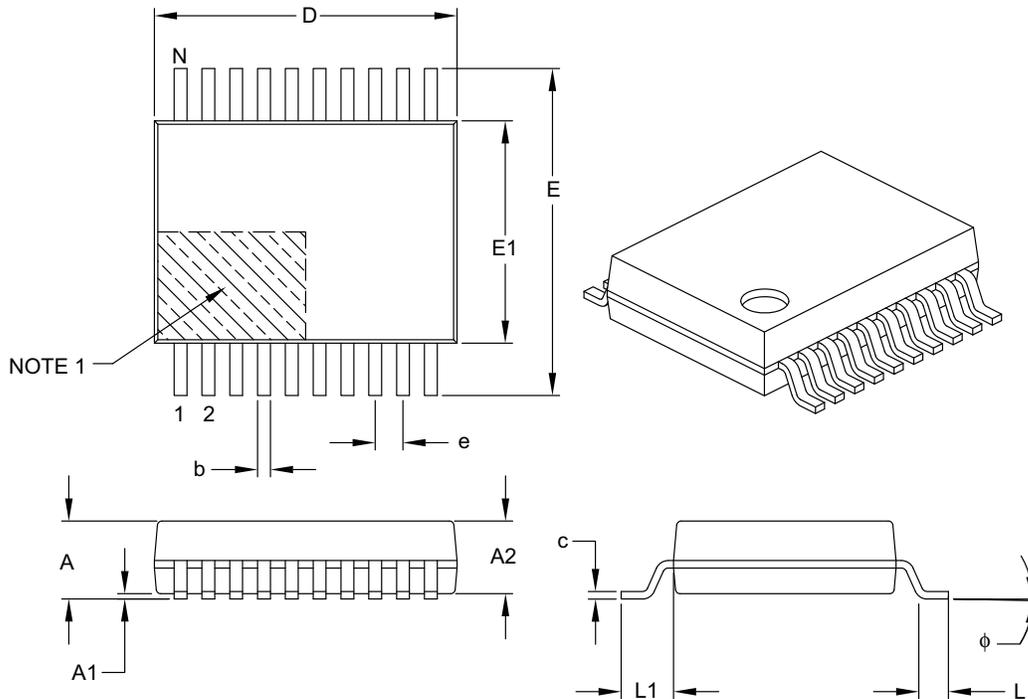
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

PIC16(L)F18324/18344

20 引脚塑封紧缩小外形封装 (SS) —— 主体 5.30 mm [SSOP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

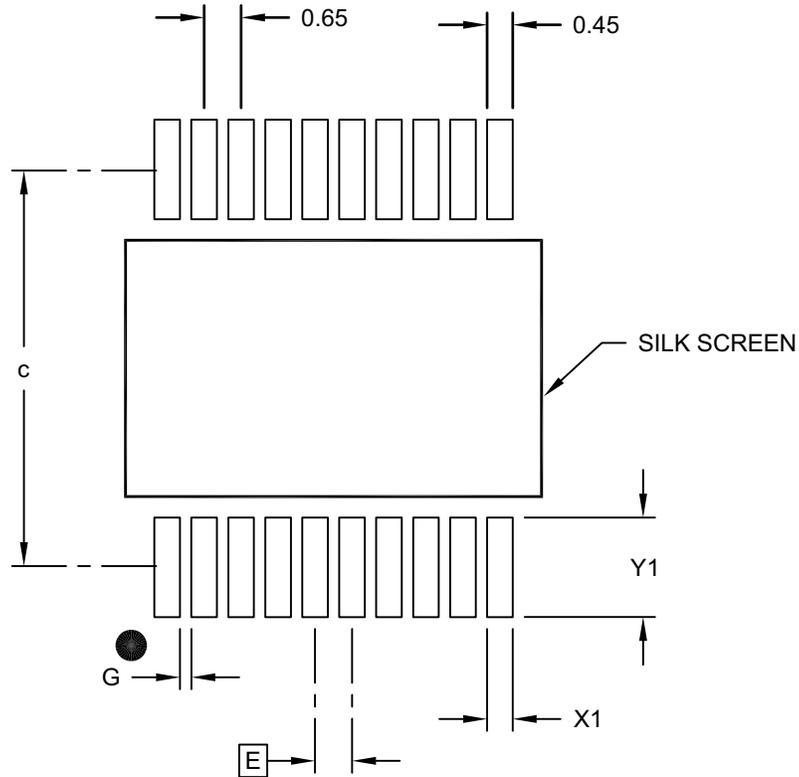
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

PIC16(L)F18324/18344

20 引脚塑封紧缩小外形封装 (SS) —— 主体 5.30 mm [SSOP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

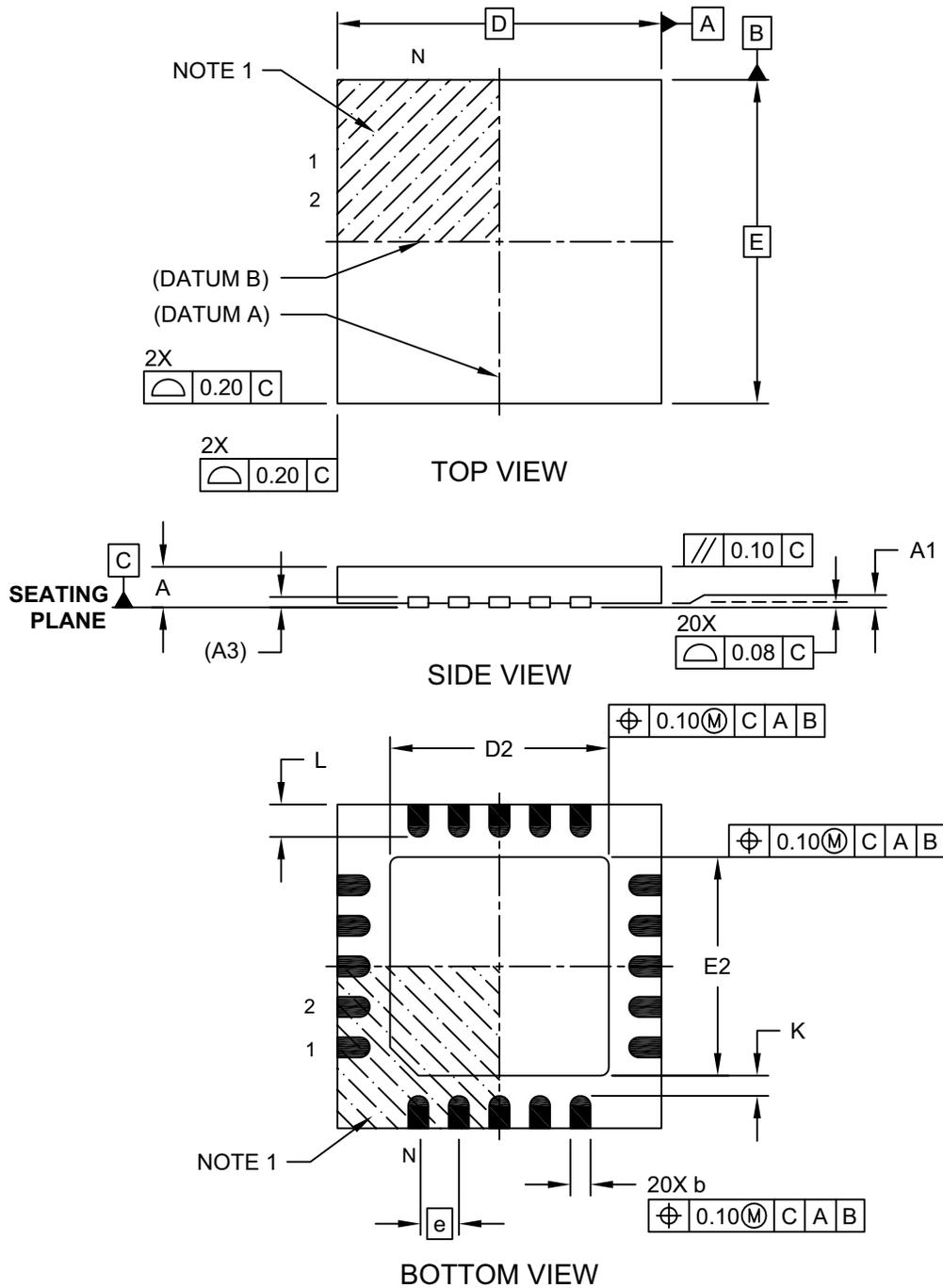
1. Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072B

PIC16(L)F18324/18344

20 引脚塑封超薄正方扁平无脚封装 (GZ) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

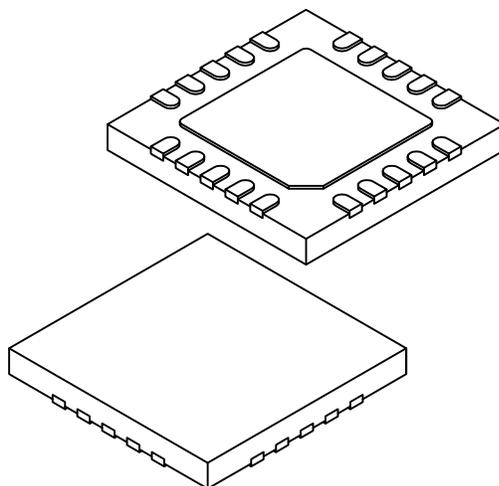


Microchip Technology Drawing C04-255A Sheet 1 of 2

PIC16(L)F18324/18344

20 引脚塑封超薄正方扁平无脚封装 (GZ) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Terminals	N		20		
Pitch	e		0.50 BSC		
Overall Height	A		0.45	0.50	0.55
Standoff	A1		0.00	0.02	0.05
Terminal Thickness	A3		0.127 REF		
Overall Width	E		4.00 BSC		
Exposed Pad Width	E2		2.60	2.70	2.80
Overall Length	D		4.00 BSC		
Exposed Pad Length	D2		2.60	2.70	2.80
Terminal Width	b		0.20	0.25	0.30
Terminal Length	L		0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

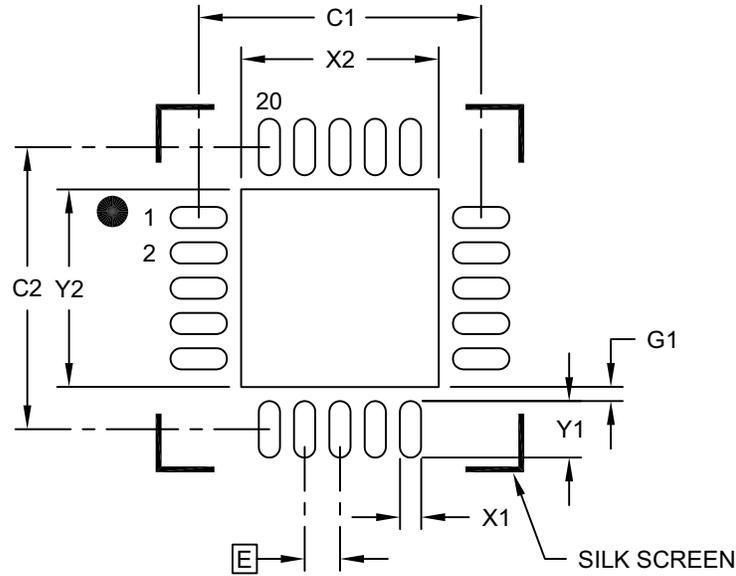
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-255A Sheet 2 of 2

PIC16(L)F18324/18344

20 引脚塑封超薄正方扁平无脚封装 (GZ) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	X2			2.80
Optional Center Pad Length	Y2			2.80
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X20)	X1			0.30
Contact Pad Length (X20)	Y1			0.80
Contact Pad to Center Pad (X20)	G1	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2255A

附录A： 版本历史

版本A（2015年07月）

本文档的初始版本。

版本B（2016年11月）

对表1-1、表1-2和表1-3进行了少量更新；新增了第二章“PIC16(L)F183XX单片机入门指南”；更新了图3-1；增加了第4.1.1.3节“NVMREG访问”和第4.2.1节“存储区选择”；更新了表6-1；更新了图7-1；更新了寄存器7-6；更新了图8-2；更新了寄存器9-1；增加了第10.2.4节“WDT始终关闭”；更新了表11-2；更新了例11-4；删除了例11-5；更新了图15-1；更新了图18-2；更新了图22-1；更新了图29-1；更新了图31-1。

版本C（2017年7月）

删除了“初稿”状态，更新了“电气规范”章节。

PIC16(L)F18324/18344

MICROCHIP网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息：

- **产品支持** —— 数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持** —— 常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务** —— 产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 www.microchip.com。在“支持”(Support)下，点击“变更通知客户(Customer Change Notification)”服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

器件	卷带式选项	温度范围	封装	定制编号	示例:	
	[X] ⁽¹⁾	-	X	/XX	XXX	
器件:	PIC16F18324、PIC16LF18324、PIC16F18344 和 PIC16LF18344。					
卷带式选项:	空白 = 标准封装 (料管或托盘封装) T = 卷带式 ⁽¹⁾					
温度范围:	I = -40°C 至 +85°C (工业级) E = -40°C 至 +125°C (扩展级)					
封装: ⁽²⁾	JQ = 16 引脚 UQFN (4x4) GZ = 20 引脚 UQFN (4x4) P = PDIP ST = TSSOP SL = 14 引脚 SOIC SO = 20 引脚 SOIC SS = SSOP					
定制编号:	QTP、SQTP、代码或特殊要求 (否则为空白)					
					注 1: 卷带式标识符仅出现在产品目录的部件编号描述中。该标识符用于订货目的，不会印刷在器件封装上。关于封装是否提供卷带式选项的信息，请咨询当地的 Microchip 销售办事处。 2: 可提供小型封装选项。关于小型封装的供货信息，请访问 www.microchip.com/packaging ，或联系当地的销售办事处。	

请注意以下有关 Microchip 器件代码保护功能的要点:

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信: 在正常使用的情况下, Microchip 系列产品是当今市场上同类产品中更安全的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下, 能访问您的软件或其他受版权保护的成果, 您有权依据该法案提起诉讼, 从而制止这种行为。

提供本文档的中文版本仅为为了便于理解。请勿忽视文档中包含的英文部分, 因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为为您提供便利, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保, 包括但不限于针对其使用情况、质量、性能、适用性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用, 一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时, 会维护和保障 Microchip 免于承担法律责任, 并加以赔偿。除非另外声明, 在 Microchip 知识产权保护下, 不得暗或以其他方式转让任何许可证。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器 and 模拟产品严格遵守公司的质量体系流程。此外, Microchip 在开发系统的设计和和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BeaconThings、BitCloud、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MedialB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、RightTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PureSilicon、QMatrix、RightTouch 徽标、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2016-2018, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-2745-2

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto
Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

澳大利亚 **Australia - Sydney**
Tel: 61-2-9868-6733

印度 **India - Bangalore**
Tel: 91-80-3090-4444

印度 **India - New Delhi**
Tel: 91-11-4160-8631

印度 **India - Pune**
Tel: 91-20-4121-0141

日本 **Japan - Osaka**
Tel: 81-6-6152-7160

日本 **Japan - Tokyo**
Tel: 81-3-6880-3770

韩国 **Korea - Daegu**
Tel: 82-53-744-4301

韩国 **Korea - Seoul**
Tel: 82-2-554-7200

马来西亚
Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

马来西亚 **Malaysia - Penang**
Tel: 60-4-227-8870

菲律宾 **Philippines - Manila**
Tel: 63-2-634-9065

新加坡 **Singapore**
Tel: 65-6334-8870

泰国 **Thailand - Bangkok**
Tel: 66-2-694-1351

越南 **Vietnam - Ho Chi Minh**
Tel: 84-28-5448-2100

欧洲

奥地利 **Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦
Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

芬兰 **Finland - Espoo**
Tel: 358-9-4520-820

法国 **France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Garching**
Tel: 49-8931-9700

德国 **Germany - Haan**
Tel: 49-2129-3766400

德国 **Germany - Heilbronn**
Tel: 49-7131-67-3636

德国 **Germany - Karlsruhe**
Tel: 49-721-625370

德国 **Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 **Germany - Rosenheim**
Tel: 49-8031-354-560

以色列 **Israel - Ra'anana**
Tel: 972-9-744-7705

意大利 **Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 **Italy - Padova**
Tel: 39-049-7625286

荷兰 **Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

挪威 **Norway - Trondheim**
Tel: 47-7289-7561

波兰 **Poland - Warsaw**
Tel: 48-22-3325737

罗马尼亚
Romania - Bucharest
Tel: 40-21-407-87-50

西班牙 **Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 **Sweden - Gothenberg**
Tel: 46-31-704-60-40

瑞典 **Sweden - Stockholm**
Tel: 46-8-5090-4654

英国 **UK - Wokingham**
Tel: 44-118-921-5800
Fax: 44-118-921-5820