

14/20 引脚 8 位闪存单片机

说明

PIC16(L)F1764/5/8/9 单片机集成了智能模拟外设和数字外设，可满足各种功能和终端设备的需求。这些 14/20 引脚器件提供了 10 位 ADC、运放、过零检测、高电流 I/O、通信、外设引脚选择和其他关键外设等诸多特性，使其在追求设计灵活性的应用中极具吸引力。

内核特性

- C 编译器优化的 RISC 架构
- 仅有 49 条指令
- 工作速度：
 - DC – 32 MHz 时钟输入
 - 最小指令周期为 125 ns
- 中断能力
- 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
- 上电复位 (Power-on Reset, POR)
- 可配置上电延时定时器 (Power-up Timer, PWRT)
- 跳变点可选的欠压复位 (Brown-out Reset, BOR)
- 扩展型看门狗定时器 (Extended Watchdog Timer, EWDT)：
 - 低功耗 31 kHz WDT
 - 软件可选预分频比
 - 软件可选使能

存储器

- 最大 14 KB 的闪存程序存储器
- 最大 1024 字节的数据 RAM 存储器
- 直接、间接和相对寻址模式
- 高耐用性闪存 (High Endurance Flash, HEF)：
 - 128B 的非易失性数据存储
 - 100,000 次擦 / 写

工作特性

- 工作电压范围：
 - 1.8V 至 3.6V (PIC16LF1764/5/8/9)
 - 2.3V 至 5.5V (PIC16F1764/5/8/9)
- 温度范围：
 - 工业级：-40°C 至 +85°C
 - 扩展级：-40°C 至 +125°C

超低功耗 (eXtreme Low-Power, XLP) 特性

- 休眠模式：50 nA (1.8V 时, 典型值)
- 看门狗定时器：500 nA (1.8V 时, 典型值)
- 辅助振荡器：500 nA (32 kHz 时)
- 工作电流：
 - 8 μA (32 kHz、1.8V 时, 典型值)
 - 32 μA/MHz (1.8V 时, 典型值)
- 低功耗 BOR (Low-Power BOR, LPBOR)：
 - 休眠时为 200 nA

数字外设

- 可配置逻辑单元 (Configurable Logic Cell, CLC)：
 - 最多 3 个 CLC；最多 4 个选定输入
 - 集成组合和状态逻辑
- 最多两个互补输出发生器 (Complementary Output Generator, COG)：
 - 推挽、全桥和转向模式
- 最多两个捕捉 / 比较 / PWM (Capture/Compare/PWM, CCP) 模块
- 脉宽调制器 (Pulse-Width Modulator, PWM)：
 - 最多两个 10 位 PWM
 - 最多两个 16 位 PWM
- 外设引脚选择 (Peripheral Pin Select, PPS)：
 - 可将任意数字引脚配置为输出
- 串行通信：
 - 增强型 USART (EUSART)
 - 符合 SPI、I²C、RS-232、RS-485 和 LIN 规范
 - 自动波特率检测, 检测到启动位时自动唤醒
- 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
- 最多两个数据信号调制器 (Data Signal Modulator, DSM)

智能模拟外设

- 10 位模数转换器 (Analog-to-Digital Converter, ADC)：
 - 最多 12 路外部通道
 - 可在休眠模式下进行转换
- 最多两个运算放大器 (Operational Amplifier, OPA)：
 - 可选的内部和外部通道
- 最多 4 个快速比较器 (COMP)：
 - 最多 5 个外部反相输入
 - 最多 8 个外部同相输入
 - 同相输入端施加固定参考电压
 - 比较器输出可供外部访问
- 数模转换器 (Digital-to-Analog Converter, DAC)：
 - 最多两个 10 位分辨率 DAC
 - 最多两个 5 位分辨率 DAC

PIC16(L)F1764/5/8/9

智能模拟外设 (续)

- 参考电压：
 - 固定参考电压 (Fixed Voltage Reference, FVR)：1.024V、2.048V 和 4.096V 输出电压
 - 过零检测器 (Zero-Cross Detector, ZCD)：
 - 检测高压交流信号
 - 可编程斜坡发生器 (Programmable Ramp Generator, PRG)：
 - 斜率补偿
 - 斜坡发生
 - 高电流驱动 I/O：
 - 5V 下的电流能力为 100 mA
- 16 MHz 内部振荡器：
 - 校准后精度为 $\pm 1\%$
 - 可选择频率范围为 31 kHz 至 32 MHz
 - 31 kHz 低功耗内部振荡器
 - 4 倍频锁相环 (Phase-Locked Loop, PLL)：
 - 可用于频率最高为 32 MHz 的内部振荡器操作
 - 外部振荡器模块具有：
 - 3 种外部时钟模式，频率最高为 32 MHz

表 1：PIC16(L)F1764/5/8/9 系列类型

器件	数据手册索引	闪存程序存储器 (字 /KB)	高耐用性闪存 (B)	数据 SRAM (字节)	I/O 引脚 (2)	8/16 位定时器	比较器	10 位 ADC (通道)	5/10 位 DAC	CCP	10/16 位 PWM	COG	数据信号调制器	CLC	运放	过零检测	可编程斜坡发生器	高电流 I/O	外设引脚选择	EUSART	I ² C/SPI	调试 (1)
PIC16(L)F1764	(A)	4096/7	128	512	12	4/3	2	8	1/1	1	1/1	1	1	3	1	1	1	2	有	1	1	I/H
PIC16(L)F1765	(A)	8192/14	128	1024	12	4/3	2	8	1/1	1	1/1	1	1	3	1	1	1	2	有	1	1	I/H
PIC16(L)F1768	(A)	4096/7	128	512	18	4/3	4	12	2/2	2	2/2	2	2	3	2	1	2	2	有	1	1	I/H
PIC16(L)F1769	(A)	8192/14	128	1024	18	4/3	4	12	2/2	2	2/2	2	2	3	2	1	2	2	有	1	1	I/H

注 1：调试方法：(I)——集成在芯片上；(H)——通过 ICD 调试头；E——仿真产品。

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。)

A. DS40001775A_CN PIC16(L)F1764/5/8/9 数据手册，14/20 引脚 8 位闪存单片机。

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> 或联系您当地的销售办事处。

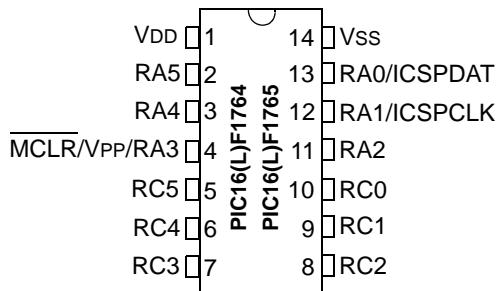
表 2：封装

封装	PDIP	SOIC	TSSOP	QFN	SSOP
PIC16(L)F1764	•	•	•	•	
PIC16(L)F1765	•	•	•	•	
PIC16(L)F1768	•	•		•	•
PIC16(L)F1769	•	•		•	•

注：引脚详细信息可能有所变动。

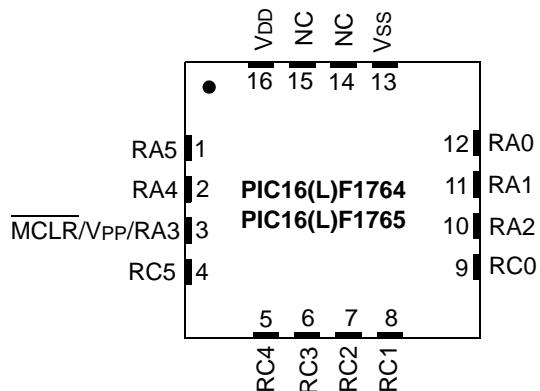
引脚图

图 1: 14 引脚 PDIP、SOIC 和 TSSOP



注：关于所有外设功能的位置，请参见表 3。

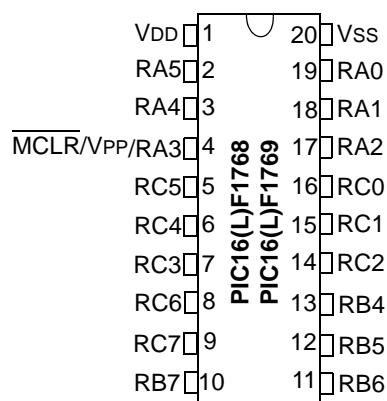
图 2: 16 引脚 QFN (4x4)



注：关于所有外设功能的位置，请参见表 3。

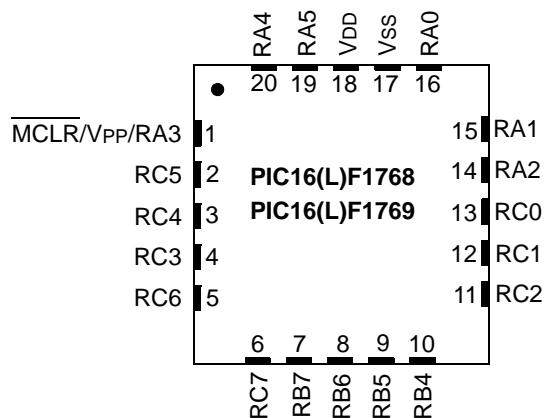
PIC16(L)F1764/5/8/9

图 3: 20 引脚 PDIP/SOIC/SSOP



注: 关于所有外设功能的位置, 请参见表 4。

图 4: 20 引脚 QFN (4x4)



注: 关于所有外设功能的位置, 请参见表 4。

引脚分配表

表 3: 14 引脚和 16 引脚分配表 (PIC16(L)F1764/5)

IO	14 引脚 PDIP/SOIC/TSSOP	16 引脚 QFN	ADC	参考电压	DAC	温度	片选	可编程寄存器	定时器	PWM	CCP	COG	CLC	波特率	USART	MSP	中置	上拉	高电流	基本功能	
RA0	13	12	AN0	VREF-DAC1REF-DAC3REF-	DAC1OUT1 DAC3OUT1	—	C1IN0+	—	—	—	—	—	—	—	—	IOC	有	—	ICSPDAT		
RA1	12	11	AN1	VREF+ DAC1REF+ DAC3REF+	—	—	C1IN0-C2IN0-	—	—	—	—	—	—	—	—	IOC	有	—	ICSPCLK		
RA2	11	10	AN2	—	—	—	—	ZCD	—	T0CKI ⁽¹⁾	—	—	COG1IN ⁽¹⁾	—	—	—	INT ⁽¹⁾ IOC	有	—	—	
RA3	4	3	—	—	—	—	—	—	—	T6CKI ⁽¹⁾	—	—	—	MD1CH ⁽¹⁾	—	—	IOC	有	—	VPP MCLR	
RA4	3	2	AN3	—	—	—	—	—	—	T1G ⁽¹⁾ SOSCO	—	—	—	—	MD1CL ⁽¹⁾	—	—	IOC	有	—	OSC2 CLKOUT
RA5	2	1	—	—	—	—	—	—	—	T1CKI ⁽¹⁾ T2CKI ⁽¹⁾ SOSCI	—	—	CLCIN3 ⁽¹⁾	MD1MOD ⁽¹⁾	—	—	IOC	有	—	OSC1 CLKIN	
RC0	10	9	AN4	—	—	OPA1IN+	C2IN0+	—	—	T5CKI ⁽¹⁾	—	—	—	—	—	SCL ⁽¹⁾ SCK ^(1,3)	IOC	有	—	—	
RC1	9	8	AN5	—	—	OPA1IN-	C1IN1-C2IN1-	—	—	T4CKI ⁽¹⁾	—	—	—	CLCIN2 ⁽¹⁾	—	—	SDI ⁽¹⁾ SDA ^(1,3)	IOC	有	—	—
RC2	8	7	AN6	—	—	OPA1OUT	C1IN2-C2IN2-	—	PRG1IN0	—	—	—	—	—	—	—	IOC	有	—	—	
RC3	7	6	AN7	—	—	—	C1IN3-C2IN3-	—	—	T5G ⁽¹⁾	—	—	CLCINO ⁽¹⁾	—	—	SS ⁽¹⁾	IOC	有	—	—	
RC4	6	5	—	—	—	—	—	—	PRG1R ⁽¹⁾	T3G ⁽¹⁾	—	—	CLCIN1 ⁽¹⁾	—	CK ⁽¹⁾	—	IOC	有	—	—	
RC5	5	4	—	—	—	—	—	—	PRG1F ⁽¹⁾	T3CKI ⁽¹⁾	—	CCP1 ⁽¹⁾	—	—	RX ^(1,3)	—	IOC	有	—	—	
VDD	1	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD	
Vss	14	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Vss	
OUT ⁽²⁾	—	—	—	—	—	—	C1OUT	—	—	PWM3	CCP1	COG1A	CLC1OUT	MD1OUT	DT ⁽³⁾	SDO	INT	—	—	—	
	—	—	—	—	—	—	C2OUT	—	—	PWM5	—	COG1B	CLC2OUT	—	TX	SDA ⁽³⁾	—	—	—	—	
	—	—	—	—	—	—	—	—	—	—	COG1C	CLC3OUT	—	CK	SCK	—	—	—	—		
	—	—	—	—	—	—	—	—	—	—	COG1D	—	—	—	SCL ⁽³⁾	—	—	—	—		

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 PORT 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。

表 4: 20 引脚分配表 (PIC16(L)F1768/9)

I/O	20 引脚 PDIP/SOIC/SSOP	20 引脚 QFN	参考电压	DAC	运放	比较器	过零检测	可编程 斜坡发生器	定时器	PWM	CCP	COG	CLC	调制器	EUSART	MSSP	中断	上拉	高电流	基本功能
	20 引脚 PDIP/SOIC/SSOP	20 引脚 QFN		ADC																
RA0	19	16	AN0	V _{REF} - DAC1REF- DAC2REF- DAC3REF- DAC4REF-	DAC1OUT1 DAC2OUT1 DAC3OUT1 DAC4OUT1	—	C1IN0+ C3IN0+	—	—	—	—	—	—	—	—	—	IOC	有	—	ICSPDAT
RA1	18	15	AN1	V _{REF} + DAC1REF+ DAC2REF+ DAC3REF+ DAC4REF+	—	—	C1IN0- C2IN0- C3IN0- C4IN0-	—	—	—	—	—	—	—	—	—	IOC	有	—	ICSPCLK
RA2	17	14	AN2	—	—	—	ZCD	—	T0CKI ⁽¹⁾	—	—	COG1IN ⁽¹⁾ COG2IN ⁽¹⁾	—	—	—	—	INT ⁽¹⁾ IOC	有	—	—
RA3	4	1	—	—	—	—	—	—	T6CKI ⁽⁴⁾	—	—	—	—	MD1CH ⁽¹⁾ MD2CH ⁽¹⁾	—	—	IOC	有	—	V _{PP} MCLR ICD
RA4	3	20	AN3	—	—	—	—	—	T1G ⁽¹⁾ SOSCO	—	—	—	—	MD1CL ⁽¹⁾ MD2CL ⁽¹⁾	—	—	IOC	有	—	OSC2 CLKOUT
RA5	2	19	—	—	—	—	—	—	T1CKI ⁽¹⁾ T2CKI ⁽¹⁾ SOSCI	—	—	—	CLCIN3 ⁽¹⁾	MD1MOD ⁽¹⁾ MD2MOD ⁽¹⁾	—	—	IOC	有	—	OSC1 CLKIN
RB4	13	10	AN10	—	—	OPA1IN0-	—	—	—	—	—	—	—	—	—	SDI ⁽¹⁾ SDA ^(1,3)	IOC	有	—	—
RB5	12	9	AN11	—	—	OPA1IN0+	—	—	—	—	—	—	—	—	RX ^(1,3)	—	IOC	有	—	—
RB6	11	8	—	—	—	—	C1IN1+ C3IN1+	—	—	—	—	—	—	—	SCL ⁽¹⁾ SCK ^(1,3)	IOC	有	—	—	—
RB7	10	7	—	—	—	—	C2IN1+ C4IN1+	—	—	—	—	—	—	—	CK ⁽¹⁾	—	IOC	有	—	—
RC0	16	13	AN4	—	—	—	C2IN0+ C4IN0+	—	—	T5CKI ⁽¹⁾	—	—	—	—	—	—	IOC	有	—	—
RC1	15	12	AN5	—	—	—	C1IN1- C2IN1- C3IN1- C4IN1-	—	T4CKI ⁽¹⁾	—	—	—	CLCIN2 ⁽¹⁾	—	—	—	IOC	有	—	—
RC2	14	11	AN6	—	—	OPA1OUT OPA2IN1- OPA2IN1+	C1IN2- C2IN2-	—	PRG1IN0 PRG2IN1	—	—	—	—	—	—	—	IOC	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 PORT 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。

表 4: 20 引脚分配表 (PIC16(L)F1768/9) (续)

I/O	20引脚PDI/PDI/SOIC/SSOP 20引脚QFN	参考电压	DAC	ADC	温度	时钟源	可编程斜坡发生器	PWM	CCP	COG	CLC	调制器	EUSART	MSSP	中断	上拉	高电流	基本功能		
RC3	7	4	AN7	—	—	OPA2OUT OPA1IN1- OPA1IN1+	C1IN3- C2IN3- C3IN3- C4IN3-	—	PRG2IN0 PRG1IN1	T5G ⁽¹⁾	—	CCP2 ⁽¹⁾	—	CLC1IN0 ⁽¹⁾	—	—	—	IOC	有	—
RC4	6	3	—	—	—	—	—	—	PRG1R ⁽¹⁾ PRG2R ⁽¹⁾	T3G ⁽¹⁾	—	—	—	CLC1IN1 ⁽¹⁾	—	—	—	IOC	有	有
RC5	5	2	—	—	—	—	—	—	PRG1F ⁽¹⁾ PRG2F ⁽¹⁾	T3CK ⁽¹⁾	—	CCP1 ⁽¹⁾	—	—	—	—	IOC	有	有	
RC6	8	5	AN8	—	—	OPA2IN0-	—	—	—	—	—	—	—	—	—	—	SS ⁽¹⁾	IOC	有	—
RC7	9	6	AN9	—	—	OPA2IN0+	—	—	—	—	—	—	—	—	—	—	—	IOC	有	—
VDD	1	18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
VSS	20	17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
OUT ⁽²⁾	—	—	—	—	—	C1OUT	—	—	—	PWM3	CCP1	COG1A	CLC1OUT	MD1OUT	DT ⁽³⁾	SDO	—	—	—	—
—	—	—	—	—	—	C2OUT	—	—	—	PWM4	CCP2	COG1B	CLC2OUT	MD2OUT	TX	SDA ⁽³⁾	—	—	—	—
—	—	—	—	—	—	C3OUT	—	—	—	PWM5	—	COG1C	CLC3OUT	—	CK	SCK	—	—	—	—
—	—	—	—	—	—	C4OUT	—	—	—	PWM6	—	COG1D	—	—	—	SCL ⁽³⁾	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	COG2A	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	COG2B	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	COG2C	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	COG2D	—	—	—	—	—	—	—	—	—

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 PORT 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。

PIC16(L)F1764/5/8/9

目录

1.0 器件概述	10
2.0 增强型中档 CPU	23
3.0 存储器构成	25
4.0 器件配置	64
5.0 振荡器模块（带故障保护时钟监视器）	71
6.0 复位	89
7.0 中断	97
8.0 掉电模式（休眠）	110
9.0 看门狗定时器（WDT）	114
10.0 闪存程序存储器控制	119
11.0 I/O 端口	136
12.0 外设引脚选择（PPS）模块	156
13.0 电平变化中断	164
14.0 固定参考电压（FVR）	171
15.0 温度指示器模块	174
16.0 模数转换器（ADC）模块	176
17.0 5 位数模转换器（DAC）模块	190
18.0 10 位数模转换器（DAC）模块	195
19.0 比较器模块	202
20.0 过零检测（ZCD）模块	212
21.0 Timer0 模块	217
22.0 带门控控制的 Timer1/3/5 模块	220
23.0 Timer2/4/6 模块	231
24.0 捕捉 / 比较 / PWM 模块	251
25.0 10 位脉宽调制（PWM）模块	265
26.0 16 位脉宽调制（PWM）模块	271
27.0 互补输出发生器（COG）模块	296
28.0 可配置逻辑单元（CLC）	331
29.0 运算放大器（OPA）模块	346
30.0 可编程斜坡发生器（PRG）模块	352
31.0 数据信号调制器（DSM）	365
32.0 主同步串行端口（MSSP）模块	375
33.0 增强型通用同步 / 异步收发器（EUSART）	428
34.0 在线串行编程（ICSP™）	459
35.0 指令集汇总	461
36.0 电气规范	475
开发支持	522
封装信息	526
附录 A: 版本历史	545
Microchip 网站	546
变更通知客户服务	546
客户支持	546
产品标识体系	547

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 <http://www.microchip.com> 上注册。

PIC16(L)F1764/5/8/9

1.0 器件概述

本数据手册介绍了 PIC16(L)F1764/5/8/9 器件。关于可用封装配置的信息，请参见表 2。

图 1-1 给出了 PIC16(L)F1764/5 器件的框图。图 1-2 给出了 PIC16(L)F1768/9 器件的框图。表 1-2 和表 1-3 给出了引脚说明。

关于每款器件的可用外设，请参见表 1-1。

表 1-1：器件外设汇总

	PIC16(L)F1764	PIC16(L)F1765	PIC16(L)F1768	PIC16(L)F1769
外设				
模数转换器 (ADC)	•	•	•	•
固定参考电压 (FVR)	•	•	•	•
过零检测 (ZCD)	•	•	•	•
温度指示器	•	•	•	•
互补输出发生器 (COG)				
COG1	•	•	•	•
COG2			•	•
可编程斜坡发生器 (PRG)				
PRG1	•	•	•	•
PRG2			•	•
10 位数模转换器 (DAC)				
DAC1	•	•	•	•
DAC2			•	•
5 位数模转换器 (DAC)				
DAC3	•	•	•	•
DAC4			•	•
捕捉 / 比较 /PWM (CCP/ECCP) 模块				
CCP1	•	•	•	•
CCP2			•	•
比较器				
C1	•	•	•	•
C2	•	•	•	•
C3			•	•
C4			•	•
可配置逻辑单元 (CLC)				
CLC1	•	•	•	•
CLC2	•	•	•	•
CLC3	•	•	•	•
数据信号调制器 (DSM)				
DSM1	•	•	•	•
DSM2			•	•

表 1-1：器件外设汇总

外设	PIC16(L)F1764	PIC16(L)F1765	PIC16(L)F1768	PIC16(L)F1769
增强型通用同步 / 异步收发器 (EUSART)				
EUSART	•	•	•	•
主同步串行端口				
MSSP	•	•	•	•
运放				
运放 1	•	•	•	•
运放 2			•	•
10 位脉宽调制器 (PWM)				
PWM3	•	•	•	•
PWM4			•	•
16 位脉宽调制器 (PWM)				
PWM5	•	•	•	•
PWM6			•	•
8 位定时器				
Timer0	•	•	•	•
Timer2	•	•	•	•
Timer4	•	•	•	•
Timer6	•	•	•	•
16 位定时器				
Timer1	•	•	•	•
Timer3	•	•	•	•
Timer5	•	•	•	•

1.1 寄存器和位命名约定

1.1.1 寄存器名称

当器件中存在相同外设的多个实例时，外设控制寄存器将以外设标识符、外设实例和控制标识符串接的形式表示。控制寄存器部分将仅显示所有寄存器名称的一个实例，在外设实例编号的位置使用“x”。当某种外设在器件中只有一个实例时，可能也会对该外设采用这种命名约定，以便与系列中包含多个实例的其他器件保持兼容。

1.1.2 位名称

位名称有两种形式：

- 短名称：位功能缩写
- 长名称：外设缩写 + 短名称

1.1.2.1 短位名称

短位名字是位功能的缩写。例如，一些外设使用 EN 位进行使能。寄存器中显示的位名称是短名称形式。

短位名称可用于在 C 程序中访问位。通过短名称访问位的一般形式为 *RegisterNamebits.ShortName*。例如，在 C 程序中，可以使用指令 COG1CON0bits.EN = 1 将 COG1CON0 寄存器中的使能位 EN 置 1。

通常不在汇编程序中使用短名称，因为不同外设可能在不同位位置使用相同名称。发生这种情况时，在包含文件生成期间，会在该短位名称的所有实例后追加一个下划线及该位所在寄存器的名称，以避免命名争用。

1.1.2.2 长位名称

长位名称的构造方法是在短名称前加上外设缩写前缀。前缀对于外设是唯一的，从而使每个长位名称也唯一。COG1 使能位的长位名称为 COG1 前缀 G1 后追加使能位短名称 EN，从而得到唯一位名称 G1EN。

在 C 程序和汇编程序中都可以使用长位名称。例如，在 C 程序中，可以使用 G1EN = 1 指令将 COG1CON0 使能位置 1。在汇编程序中，可以使用 BSF COG1CON0,G1EN 指令将该位置 1。

1.1.2.3 位域

位域是同一个寄存器中的两个或更多相邻位。位域仅遵从短位命名约定。例如，COG1CON0 寄存器的低 3 位包含模式控制位。该位域的短名称为 MD。不存在长位名称形式。只能在 C 程序中进行位域访问。以下示例给出了将 COG1 设置为推挽模式的 C 程序指令：

```
COG1CON0bits.MD = 0x5;
```

此外，还可以使用长位名称和短位名称来访问位域中的各个位。每个位的名称为位域名称后追加位域内的位位置编号。例如，最高有效模式位的短位名称为 MD2，长位名称为 G1MD2。以下两个示例给出了将 COG1 设置为推挽模式的汇编程序序列：

例 1：

```
MOVLW ~ (1<<G1MD1)
ANDWF COG1CON0,F
MOVLW 1<<G1MD2 | 1<<G1MD0
IORWF COG1CON0,F
```

例 2：

```
BSF COG1CON0,G1MD2
BCF COG1CON0,G1MD1
BSF COG1CON0,G1MD0
```

1.1.3 寄存器和位命名例外情况

1.1.3.1 状态、中断和镜像位

状态、中断允许、中断标志和镜像位包含在跨多个外设的寄存器中。在这些情况下，所显示的位名称是唯一的，因此没有前缀或短名称形式。

1.1.3.2 旧式外设

有一些外设并不严格遵从这些命名约定。一些外设已经存在多年，几乎在每款器件中都存在，这些外设属于例外情况。为了限制新约定对于遗留代码的负面影响，这些例外情况是必要的。对于遵从新约定的外设，寄存器部分会包含一个表，列出每个外设实例的长名称前缀。属于例外类别的外设不具有该表。这些外设包括但不限于以下外设：

- EUSART
- MSSP

PIC16(L)F1764/5/8/9

图 1-1: PIC16(L)F1764/5 框图

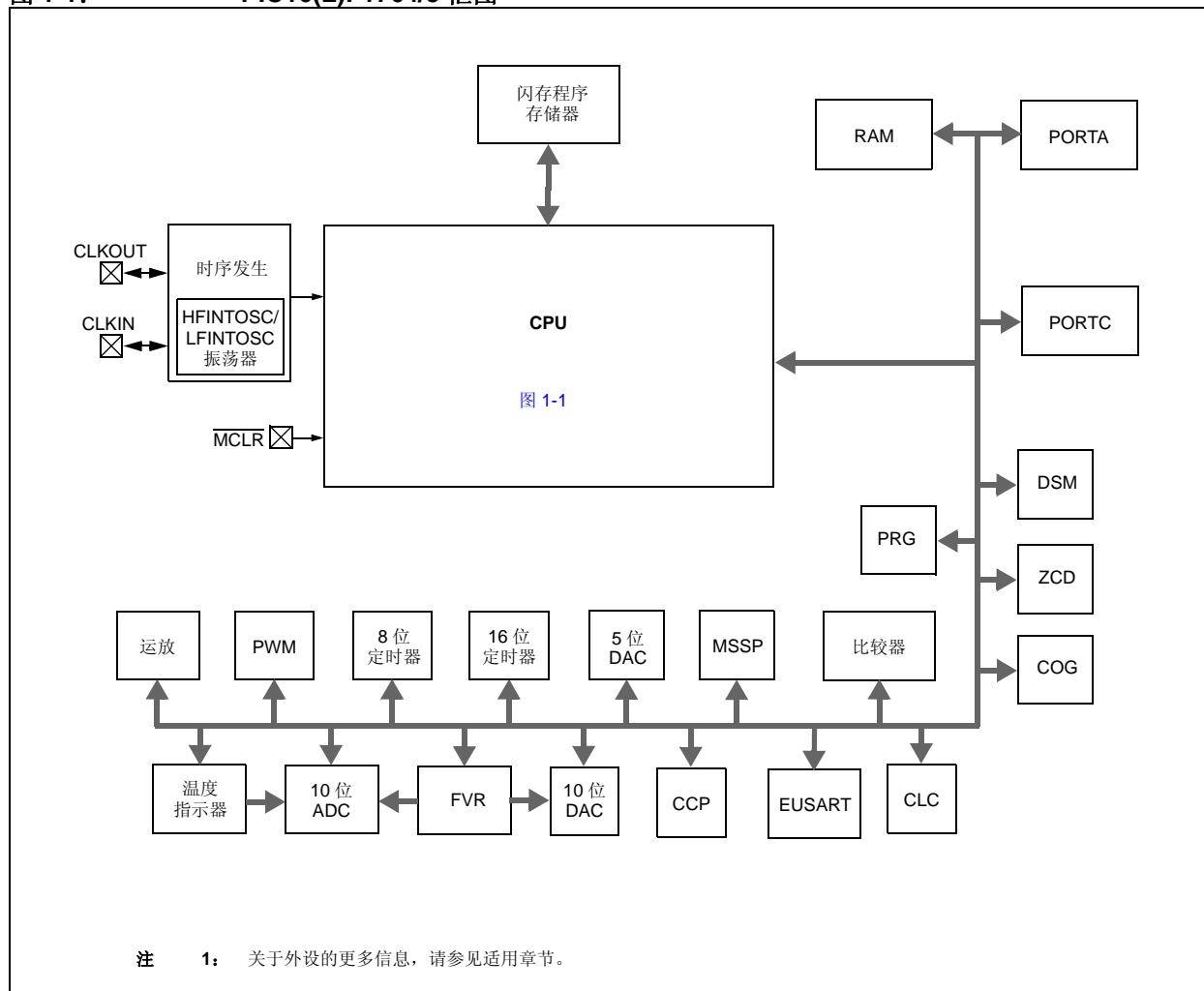
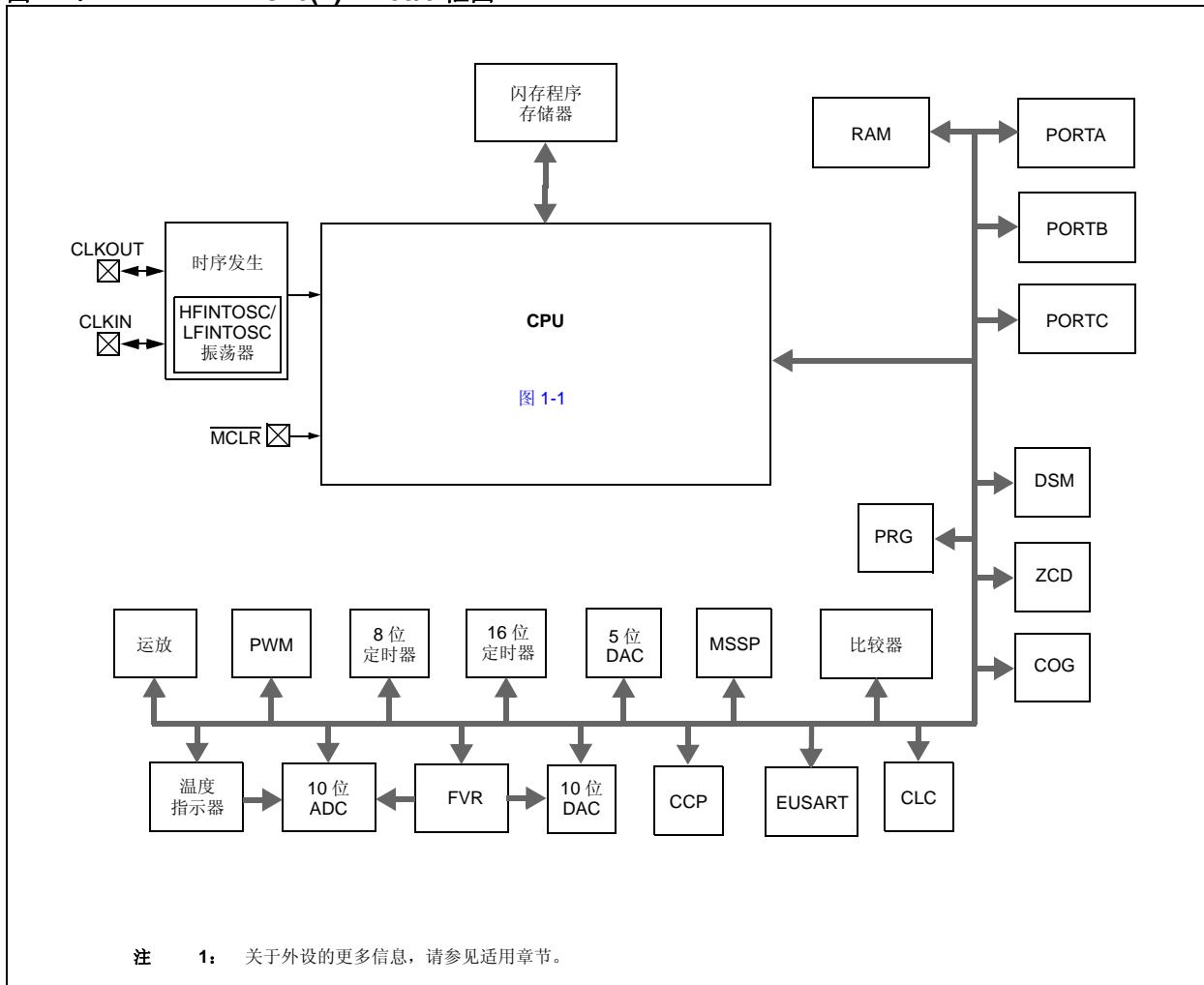


图 1-2: PIC16(L)F1768/9 框图



PIC16(L)F1764/5/8/9

表 1-2: PIC16(L)F1764/5 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN0+/VREF-/DAC1REF-/DAC3REF-/DAC1OUT1/DAC3OUT1/ICSPDAT	RA0	TTL/ST	CMOS	通用 I/O
	AN0	AN	—	ADC 通道 0 的输入
	C1IN0+	AN	—	比较器 C1 的同相输入
	VREF-	AN	—	ADC 负参考电压
	DAC1REF-	AN	—	DAC1 负参考电压
	DAC3REF-	AN	—	DAC3 负参考电压
	DAC1OUT1	—	AN	DAC1 电压输出
	DAC3OUT1	—	AN	DAC3 电压输出
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O
RA1/AN1/C1IN0-/C2IN0-/VREF+/DAC1REF+/DAC3REF+/ICSPCLK	RA1	TTL/ST	CMOS	通用 I/O
	AN1	AN	—	ADC 通道 1 的输入
	C1IN0-	AN	—	比较器 C1 的反相输入
	C2IN0-	AN	—	比较器 C2 的反相输入
	VREF+	AN	—	ADC 正参考电压
	DAC1REF+	AN	—	DAC1 正参考电压
	DAC3REF+	AN	—	DAC3 正参考电压
	ICSPCLK	ST	—	串行编程时钟
	RA2	TTL/ST	CMOS	通用 I/O
RA2/AN2/ZCD/T0CKI/COG1IN/INT	AN2	AN	—	ADC 通道 2 的输入
	ZCD	AN	—	过零检测输入
	T0CKI	TTL/ST	—	Timer0 时钟输入
	COG1IN ⁽¹⁾	TTL/ST	—	互补输出发生器 1 输入
	INT ⁽¹⁾	TTL/ST	—	中断输入
	RA3/T6CKI/MD1CH/MCLR/VPP	TTL/ST	CMOS	通用 I/O
RA3/T6CKI/MD1CH/MCLR/VPP	T6CKI ⁽¹⁾	TTL/ST	—	Timer6 时钟输入
	MD1CH ⁽¹⁾	TTL/ST	—	数据信号调制器 1 高载波输入
	MCLR	ST	—	主复位输入
	VPP	HV	—	编程使能
	RA4/AN3/SOSCO/T1G/MD1CL/OSC2/CLKOUT	TTL/ST	CMOS	通用 I/O
RA4/AN3/SOSCO/T1G/MD1CL/OSC2/CLKOUT	AN3	AN	—	ADC 通道 3 的输入
	SOSCO	—	XTAL	辅助振荡器连接
	T1G ⁽¹⁾	TTL/ST	—	Timer1 门控输入
	MD1CL ⁽¹⁾	TTL/ST	—	数据信号调制器 1 低载波输入
	OSC2	—	XTAL	晶振 / 谐振器 (LP、XT 和 HS 模式)
	CLKOUT	—	CMOS	Fosc/4 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

注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以选择备用引脚作为外设输入。

2: 所有引脚数字输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器, 可以选择备用输出作为外设数字输出。

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。

表 1-2: PIC16(L)F1764/5 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RA5/T1CKI/T2CKI/CLCIN3/ MD1MOD/SOSCI/OSC1/CLKIN	RA5	TTL/ST	CMOS	通用 I/O
	T1CKI ⁽¹⁾	TTL/ST	—	Timer1 时钟输入
	T2CKI ⁽¹⁾	TTL/ST	—	Timer2 时钟输入
	CLCIN3 ⁽¹⁾	TTL/ST	—	CLC 输入 3
	MD1MOD ⁽¹⁾	TTL/ST	—	数据信号调制器调制输入
	SOSCI	—	XTAL	辅助振荡器连接
	OSC1	XTAL	—	晶振 / 谐振器 (LP、XT 和 HS 模式)
	CLKIN	ST	—	外部时钟输入 (EC 模式)
RC0/AN4/OPA1IN+/C2IN0+/ T5CKI/SCL/SCK	RC0	TTL/ST	CMOS	通用 I/O
	AN4	AN	—	ADC 通道 4 的输入
	OPA1IN+	AN	—	运算放大器 1 的同相输入
	C2IN0+	AN	—	比较器 2 的同相输入
	T5CKI ⁽¹⁾	TTL/ST	—	Timer5 时钟输入
	SCL ^(1,3)	I ² C	—	I ² C 时钟输出
	SCK ⁽¹⁾	TTL/ST	—	SPI 时钟输入
RC1/AN5/OPA1IN-/C1IN1-/ C2IN1-/T4CKI/CLCIN2/SDI/SDA	RC1	TTL/ST	CMOS	通用 I/O
	AN5	AN	XTAL	ADC 通道 5 的输入
	OPA1IN-	AN	—	运算放大器 1 的反相输入
	C1IN1-	AN	—	比较器 1 的反相输入
	C2IN1-	AN	—	比较器 2 的反相输入
	T4CKI ⁽¹⁾	TTL/ST	—	Timer4 时钟输入
	CLCIN2 ⁽¹⁾	TTL/ST	—	CLC 输入 2
	SDI ⁽¹⁾	TTL/ST	—	SPI 数据输入
	SDA ⁽¹⁾	I ² C	—	I ² C 数据输出
RC2/AN6/OPA1OUT/C1IN2-/ C2IN2-/PRG1IN0	RC2	TTL/ST	CMOS	通用 I/O
	AN6	AN	—	ADC 通道 6 的输入
	OPA1OUT	—	AN	运算放大器 1 的输出
	C1IN2-	AN	—	比较器 1 的反相输入
	C2IN2-	AN	—	比较器 2 的反相输入
	PRG1IN0	AN	—	斜坡发生器 1 参考电压输入
RC3/AN7/C1IN3-/C2IN3-/T5G/ CLCIN0/SS	RC3	TTL/ST	CMOS	通用 I/O
	AN7	AN	—	ADC 通道 7 的输入
	C1IN3-	AN	—	比较器 1 的反相输入
	C2IN3-	AN	—	比较器 2 的反相输入
	T5G ⁽¹⁾	TTL/ST	—	Timer5 门控输入
	CLCIN0 ⁽¹⁾	TTL/ST	—	CLC 输入 0
	SS ⁽¹⁾	TTL/ST	—	SPI 从选择输入

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

注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以选择备用引脚作为外设输入。

2: 所有引脚数字输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器, 可以选择备用输出作为外设数字输出。

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。

PIC16(L)F1764/5/8/9

表 1-2: PIC16(L)F1764/5 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RC4/T3G/PRG1R/CLCIN1/CK	RC4	TTL/ST	CMOS	通用 I/O
	T3G ⁽¹⁾	TTL/ST	—	Timer3 门控输入
	PRG1R ⁽¹⁾	TTL/ST	—	斜坡发生器 set_rising 输入
	CLCIN1 ⁽¹⁾	TTL/ST	—	CLC 输入 1
	CK ⁽¹⁾	TTL/ST	—	EUSART 时钟输入
RC5/T3CKI/PRG1F/CCP1/RX	RC5	TTL/ST	CMOS	通用 I/O
	T3CKI ⁽¹⁾	TTL/ST	—	Timer3 时钟输入
	PRG1F ⁽¹⁾	TTL/ST	—	斜坡发生器 set_falling 输入
	CCP1 ⁽¹⁾	TTL/ST	—	CCP1 捕捉输入
	RX ^(1,3)	TTL/ST	—	EUSART 接收输入
VDD	VDD	电源	—	正电源
Vss	Vss	电源	—	参考地
OUT ⁽²⁾	C1OUT		CMOS	比较器 1 的输出
	C2OUT		CMOS	比较器 2 的输出
	CCP1		CMOS	比较 /PWM1 输出
	MD1OUT		CMOS	数据信号调制器 1 输出
	PWM3		CMOS	PWM3 输出
	PWM5		CMOS	PWM5 输出
	COG1A		CMOS	互补输出发生器输出 A
	COG1B		CMOS	互补输出发生器输出 B
	COG1C		CMOS	互补输出发生器输出 C
	COG1D		CMOS	互补输出发生器输出 D
	SDA ⁽³⁾	OD	I ² C	数据输出
	SCK	CMOS	SPI	时钟输出
	SCL ⁽³⁾	OD	I ² C	时钟输出
	SDO	CMOS	SPI	数据输出
	TX	CMOS	EUSART	异步发送数据输出
	CK	CMOS	EUSART	同步时钟输出
	DT ⁽³⁾	CMOS	EUSART	同步数据输出
	CLC1OUT		CMOS	可配置逻辑单元 1 输出
	CLC2OUT		CMOS	可配置逻辑单元 2 输出
	CLC3OUT		CMOS	可配置逻辑单元 3 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

注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以选择备用引脚作为外设输入。

2: 所有引脚数字输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器, 可以选择备用输出作为外设数字输出。

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。

表 1-3: PIC16(L)F1768/9 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN0+/C3IN0+/VREF-/DAC1REF-/DAC2REF-/DAC3REF-/DAC4REF-/DAC1OUT1/DAC2OUT1./DAC3OUT1/DAC4OUT1/ICSPDAT	RA0	TTL/ST	CMOS	通用 I/O
	AN0	AN	—	ADC 通道 0 的输入
	C1IN0+	AN	—	比较器 C1 的同相输入
	C3IN0+	AN	—	比较器 C3 的同相输入
	DAC1REF-	AN	—	DAC1 负参考电压
	DAC2REF-	AN	—	DAC2 负参考电压
	DAC3REF-	AN	—	DAC3 负参考电压
	DAC4REF-	AN	—	DAC4 负参考电压
	DAC1OUT1	—	AN	DAC1 电压输出
	DAC2OUT1	—	AN	DAC2 电压输出
	DAC3OUT1	—	AN	DAC3 电压输出
	DAC4OUT1	—	AN	DAC4 电压输出
	VREF-	AN	—	ADC 负参考电压
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O
RA1/AN1/C1IN0-/C2IN0-/C3IN0-/C4IN0-/VREF+/-DAC1REF+/DAC2REF+/DAC3REF+/DAC4REF+/ICSPCLK	RA1	TTL/ST	CMOS	通用 I/O
	AN1	AN	—	ADC 通道 1 的输入
	C1IN0-	AN	—	比较器 C1 的反相输入
	C2IN0-	AN	—	比较器 C2 的反相输入
	C3IN0-	AN	—	比较器 C3 的反相输入
	C4IN0-	AN	—	比较器 C4 的反相输入
	DAC1REF+	AN	—	DAC1 正参考电压
	DAC2REF+	AN	—	DAC2 正参考电压
	DAC3REF+	AN	—	DAC3 正参考电压
	DAC4REF+	AN	—	DAC4 正参考电压
	VREF+	AN	—	ADC 正参考电压
	ICSPCLK	ST	—	串行编程时钟
RA2/AN2/ZCD/T0CKI/COG1IN/COG2IN/INT	RA2	TTL/ST	CMOS	通用 I/O
	AN2	AN	—	ADC 通道 2 的输入
	ZCD	AN	—	过零检测输入
	T0CKI ⁽¹⁾	TTL/ST	—	Timer0 时钟输入
	COG1IN ⁽¹⁾	TTL/ST	—	互补输出发生器 1 输入
	COG2IN ⁽¹⁾	TTL/ST	—	互补输出发生器 2 输入
	INT ⁽¹⁾	TTL/ST	—	中断输入
RA3/T6CKI/MD1CH/MD2CH/MCLR/VPP	RA3	TTL/ST	CMOS	通用 I/O
	T6CKI ⁽¹⁾	TTL/ST	—	Timer6 时钟输入
	MD1CH ⁽¹⁾	TTL/ST	—	数据信号调制器 1 高载波输入
	MD2CH ⁽¹⁾	TTL/ST	—	数据信号调制器 2 高载波输入
	MCLR	ST	—	主复位输入
	VPP	HV	—	编程使能

图注: AN = 模拟输入或输出

TTL = TTL 兼容输入

HV = 高电压

CMOS = CMOS 兼容输入或输出

ST = CMOS 电平施密特触发器输入

XTAL = 晶振

OD = 漏极开路

I²C = I²C 电平施密特触发器输入

注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以选择备用引脚作为外设输入。

2: 所有引脚数字输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器, 可以选择备用输出作为外设数字输出。

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。

PIC16(L)F1764/5/8/9

表 1-3: PIC16(L)F1768/9 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RA4/AN3/SOSCO/T1G/ DSM1CL/DSM2CL/OSC2/ CLKOUT	RA4	TTL/ST	CMOS	通用 I/O
	AN3	AN	—	ADC 通道 3 的输入
	SOSCO	—	XTAL	辅助振荡器连接
	T1G ⁽¹⁾	TTL/ST	—	Timer1 门控输入
	DSM1CL ⁽¹⁾	TTL/ST	—	数据信号调制器 1 低载波输入
	DSM2CL ⁽¹⁾	TTL/ST	—	数据信号调制器 2 低载波输入
	OSC2	—	XTAL	晶振 / 谐振器 (LP、XT 和 HS 模式)
	CLKOUT	—	CMOS	Fosc/4 输出
RA5/T1CKI/T2CKI/CLCIN3/ DSM1MOD/DSM2MOD/ SOSCI/OSC1/CLKIN	RA5	TTL/ST	CMOS	通用 I/O
	T1CKI ⁽¹⁾	TTL/ST	—	Timer1 时钟输入
	T2CKI ⁽¹⁾	TTL/ST	—	Timer2 时钟输入
	CLCIN3 ⁽¹⁾	TTL/ST	—	CLC 输入 3
	DSM1MOD ⁽¹⁾	TTL/ST	—	数据信号调制器 1 调制输入
	DSM2MOD ⁽¹⁾	TTL/ST	—	数据信号调制器 2 调制输入
	SOSCI	XTAL	—	辅助振荡器连接
	OSC1	XTAL	—	晶振 / 谐振器 (LP、XT 和 HS 模式)
	CLKIN	ST	—	外部时钟输入 (EC 模式)
RB4/AN10/OPA1IN0-/SDI/SDA	RB4	TTL/ST	CMOS	通用 I/O
	AN10	AN	—	ADC 通道 10 的输入
	OPA1IN0-	AN	—	运算放大器 1 的反相输入
	SDI ⁽¹⁾	TTL/ST	—	SPI 数据输入
	SDA ^(1,3)	I ² C	—	I ² C 数据输出
RB5/AN11/OPA1IN0+/RX	RB5	TTL/ST	CMOS	通用 I/O
	AN11	AN	—	ADC 通道 11 的输入
	OPA1IN0+	AN	—	运算放大器 1 的同相输入
	RX ^(1,3)	TTL/ST	—	EUSART 接收输入
RB6/C1IN1+/C3IN1+/SCK/SCL	RB6	TTL/ST	CMOS	通用 I/O
	C1IN1+	AN	—	比较器 C1 的同相输入
	C3IN1+	AN	—	比较器 C3 的同相输入
	SCK ⁽¹⁾	TTL/ST	—	SPI 时钟输入
	SCL ^(1,3)	I ² C	—	I ² C 时钟输出
RB7/C2IN1+/C4IN1+/CK	RB7	TTL/ST	CMOS	通用 I/O
	C2IN1+	AN	—	比较器 C2 的同相输入
	C4IN1+	AN	—	比较器 C4 的同相输入
	CK ⁽¹⁾	TTL/ST	—	EUSART 时钟输入
RC0/AN4/C2IN0+/C4IN0+/ T5CKI	RC0	TTL/ST	CMOS	通用 I/O
	AN4	AN	—	ADC 通道 4 的输入
	C2IN0+	AN	—	比较器 C2 的同相输入
	C4IN0+	AN	—	比较器 C4 的同相输入
	T5CKI ⁽¹⁾	TTL/ST	—	Timer5 时钟输入

图注: AN = 模拟输入或输出
TTL = TTL 兼容输入
HV = 高电压

CMOS = CMOS 兼容输入或输出
ST = CMOS 电平施密特触发器输入
XTAL = 晶振

OD = 漏极开路
I²C = I²C 电平施密特触发器输入

- 注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以选择备用引脚作为外设输入。
 2: 所有引脚数字输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器, 可以选择备用输出作为外设数字输出。
 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。

表 1-3: PIC16(L)F1768/9 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RC1/AN5/C1IN1-/C2IN1-/C3IN1-/C4IN1-/T4CKI/CLCIN2	RC1	TTL/ST	CMOS	通用 I/O
	AN5	AN	XTAL	ADC 通道 5 的输入
	C1IN1-	AN	—	比较器 1 的反相输入
	C2IN1-	AN	—	比较器 2 的反相输入
	C3IN1-	AN	—	比较器 3 的反相输入
	C4IN1-	AN	—	比较器 4 的反相输入
	T4CKI ⁽¹⁾	TTL/ST	—	Timer4 时钟输入
	CLCIN2 ⁽¹⁾	TTL/ST	—	CLC 输入 2
RC2/AN6/OPA1OUT/OPA2IN1-/OPA2IN1+/C1IN2-/C2IN2-/PRG1IN0/PRG2IN1	RC2	TTL/ST	CMOS	通用 I/O
	AN6	AN	—	ADC 通道 6 的输入
	OPA1OUT	—	AN	运算放大器 1 的输出
	OPA2IN1-	AN	—	运算放大器 2 的反相输入
	OPA2IN1+	AN	—	运算放大器 2 的同相输入
	C1IN2-	AN	—	比较器 1 的反相输入
	C2IN2-	AN	—	比较器 2 的反相输入
	PRG1IN0	AN	—	斜坡发生器 1 参考电压输入
	PRG2IN1	AN	—	斜坡发生器 2 参考电压输入
	RC3	TTL/ST	CMOS	通用 I/O
RC3/AN7/OPA2OUT/OPA1IN1-/OPA1IN1+/C1IN3-/C2IN3-/C3IN3-/C4IN3-/PRG1IN1/PRG2IN0/T5G/CCP2/CLCIN0	AN7	AN	—	ADC 通道 7 的输入
	OPA2OUT	—	AN	运算放大器 2 的输出
	OPA1IN1-	AN	—	运算放大器 1 的反相输入
	OPA1IN1+	AN	—	运算放大器 1 的同相输入
	C1IN3-	AN	—	比较器 1 的反相输入
	C2IN3-	AN	—	比较器 2 的反相输入
	C3IN3-	AN	—	比较器 3 的反相输入
	C4IN3-	AN	—	比较器 4 的反相输入
	PRG1IN1	AN	—	斜坡发生器 1 参考电压输入
	PRG2IN0	AN	—	斜坡发生器 2 参考电压输入
	T5G ⁽¹⁾	TTL/ST	—	Timer5 门控输入
	CCP2 ⁽¹⁾	TTL/ST	—	CCP2 捕捉输入
	CLCIN0 ⁽¹⁾	TTL/ST	—	CLC 输入 0
	RC4	TTL/ST	CMOS	通用 I/O
RC4/T3G/PRG1R/PRG2R/CLCIN1	T3G ⁽¹⁾	TTL/ST	—	Timer3 门控输入
	PRG1R ⁽¹⁾	TTL/ST	—	斜坡发生器 1 的 set_rising 输入
	PRG2R ⁽¹⁾	TTL/ST	—	斜坡发生器 2 的 set_rising 输入
	CLCIN1 ⁽¹⁾	TTL/ST	—	CLC 输入 1
	RC5	TTL/ST	CMOS	通用 I/O
RC5/T3CKI/PRG1F/PRG2F/CCP1	T3CKI ⁽¹⁾	TTL/ST	—	Timer3 时钟输入
	PRG1F ⁽¹⁾	TTL/ST	—	斜坡发生器 1 的 set_falling 输入
	PRG2F ⁽¹⁾	TTL/ST	—	斜坡发生器 2 的 set_falling 输入
	CCP1 ⁽¹⁾	TTL/ST	—	CCP1 捕捉输入

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

注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以选择备用引脚作为外设输入。

2: 所有引脚数字输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器, 可以选择备用输出作为外设数字输出。

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。

PIC16(L)F1764/5/8/9

表 1-3: PIC16(L)F1768/9 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RC6/AN8/OPA2IN0-/SS	RC6	TTL/ST	CMOS	通用 I/O
	AN8	AN	—	ADC 通道 8 的输入
	OPA2IN0-	AN	—	运算放大器 2 的反相输入
	SS ⁽¹⁾	TTL/ST	—	SPI 从选择输入
RC7/AN9/OPA2IN0+	RC7	TTL/ST	CMOS	通用 I/O
	AN9	AN	—	ADC 通道 9 的输入
	OPA2IN0+	AN	—	运算放大器 2 的同相输入
VDD	VDD	电源	—	正电源
Vss	VSS	电源	—	参考地
OUT ⁽²⁾	C1OUT		CMOS	比较器 1 的输出
	C2OUT		CMOS	比较器 2 的输出
	C3OUT		CMOS	比较器 3 的输出
	C4OUT		CMOS	比较器 4 的输出
	CCP1		CMOS	比较 /PWM1 输出
	CCP2		CMOS	比较 /PWM2 输出
	MD1OUT		CMOS	数据信号调制器 1 输出
	MD2OUT		CMOS	数据信号调制器 2 输出
	PWM3		CMOS	PWM3 输出
	PWM4		CMOS	PWM4 输出
	PWM5		CMOS	PWM5 输出
	PWM6		CMOS	PWM6 输出
	COG1A		CMOS	互补输出发生器 1 输出 A
	COG1B		CMOS	互补输出发生器 1 输出 B
	COG1C		CMOS	互补输出发生器 1 输出 C
	COG1D		CMOS	互补输出发生器 1 输出 D
	COG2A		CMOS	互补输出发生器 2 输出 A
	COG2B		CMOS	互补输出发生器 2 输出 B
	COG2C		CMOS	互补输出发生器 2 输出 C
	COG2D		CMOS	互补输出发生器 2 输出 D
	SDA ⁽³⁾	OD	I ² C 数据输出	
	SCK	CMOS	SPI 时钟输出	
	SCL ⁽³⁾	OD	I ² C 时钟输出	
	SDO	CMOS	SPI 数据输出	
	TX	CMOS	EUSART 异步发送数据输出	
	CK	CMOS	EUSART 同步时钟输出	
	DT ⁽³⁾	CMOS	EUSART 同步数据输出	
	CLC1OUT		CMOS	可配置逻辑单元 1 输出
	CLC2OUT		CMOS	可配置逻辑单元 2 输出
	CLC3OUT		CMOS	可配置逻辑单元 3 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

注 1: 默认外设输入。通过 PPS 输入选择寄存器, 可以选择备用引脚作为外设输入。

2: 所有引脚数字输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器, 可以选择备用输出作为外设数字输出。

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。

1.2 外设连接矩阵

许多外设上的输入选择多路开关允许选择另一个外设的输出，从而使信号路径完全包含在器件内。虽然也可以通过PPS选择功能将外设输出送到引脚上，但这并不是必需的。[表1-4](#)列出了所有可能的外设间信号连接。要获取所需连接的多路开关选择代码，请参见相应的外设章节。

PIC16(L)F1764/5/8/9

表 1-4: 外设连接矩阵

外设输出	外设输入																								
	ADC 触发	COG 时钟	COG 上升沿 / 下降沿事件	COG 关断	10 位 DAC	5 位 DAC	PRG 模拟输入	PRG 上升沿 / 下降沿事件	比较器 +	比较器 -	CLC	DSM CH	DSM CL	DSM 模块	运放 +	运放 -	运放改写	10 位 PWM	16 位 PWM	CCP 捕捉	CCP 时钟	Timer2/4/6 时钟	Timer2/4/6 复位	Timer1/3/5 门控	Timer0 时钟
FVR			•	•	•			•						•	•										
ZCD										•						•						•			
PRG								•							•	•									
10 位 DAC					•	•									•	•									
5 位 DAC						•	•								•	•									
CCP	•	•							•	•	•	•	•				•						•		
比较器 (同步)	•							•		•							•			•			•	•	
比较器 (异步)			•	•												•									
CLC	•	•	•							•	•	•	•	•			•		•		•	•	•		
DSM																									
COG																	•								
EUSART TX/CK										•															
EUSART DT										•															
MSSP SCK/SCL										•															
MSSP SDO/SDA										•															
运放							•																		
10 位 PWM	•	•						•		•	•	•	•	•		•							•		
16 位 PWM	•	•						•		•	•	•	•	•		•							•		
Timer0 溢出	•									•													•		
Timer2 = T2PR				•						•								•				•	•		
Timer4 = T4PR				•						•								•				•	•		
Timer6 = T6PR				•						•								•				•	•		
Timer2 后分频器	•			•						•								•				•	•		
Timer4 后分频器	•			•						•								•				•	•		
Timer6 后分频器	•			•						•								•				•	•		
Timer1 溢出	•									•								•				•	•		
Timer3 溢出	•									•								•				•	•		
Timer5 溢出	•									•								•				•	•		
SOSC																			•				•		
Fosc/4	•																						•		
Fosc	•										•	•	•	•					•				•		
HFINTOSC	•										•	•	•	•					•				•		
LFINTOSC											•								•				•		
MFINTOSC																							•		
IOCIF											•								•	•					
PPS 输入引脚			•	•					•			•	•	•					•	•	•	•	•	•	

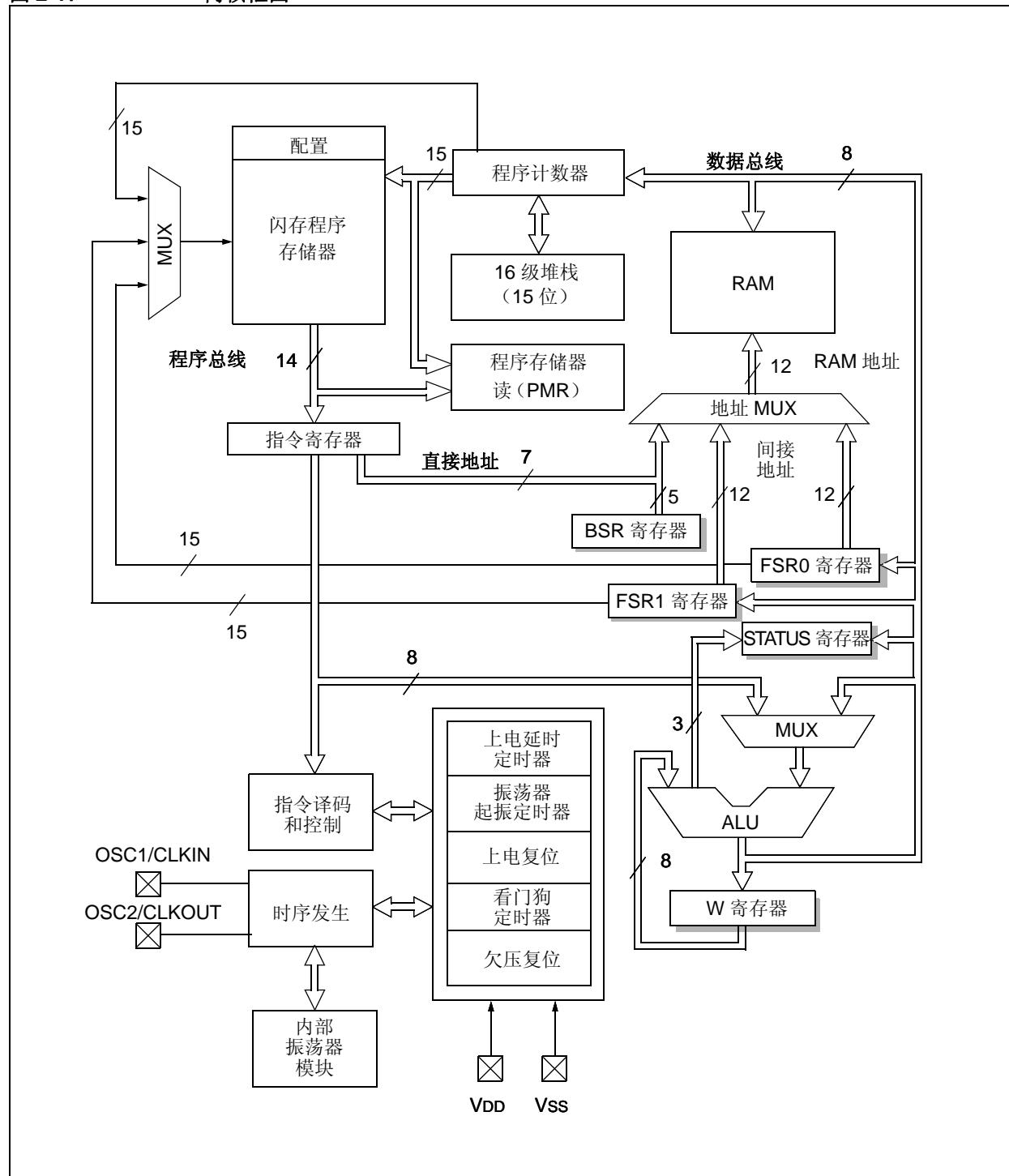
2.0 增强型中档 CPU

本器件系列包含了增强型中档 8 位 CPU 内核。CPU 具有 49 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。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

个文件选择寄存器（FSR）来读取程序和数据存储器。

- 自动中断现场保护
- 带有上溢和下溢的 16 级堆栈
- 文件选择寄存器
- 指令集

图 2-1：内核框图



2.1 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 7.5 节 “自动现场保护”](#)。

2.2 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，PCON 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 3.6 节 “堆栈”](#)。

2.3 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和程序存储器，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，用于取数据。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 3.7 节 “间接寻址”](#)。

2.4 指令集

增强型中档 CPU 具有 49 条指令，用于支持 CPU 的特性。更多详细信息，请参见[第 35.0 节 “指令集汇总”](#)。

3.0 存储器构成

这些器件包含以下类型的存储器：

- 程序存储器
 - 配置字
 - 器件 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

注 1: 第 10.0 节“**闪存程序存储器控制**”介绍了通过 PMCON 寄存器访问闪存的方法。

以下特性与程序存储器和数据存储器的访问和控制相关联：

- PCL 和 PCLATH
- 堆栈
- 间接寻址

表 3-1: 器件存储容量和地址

器件	程序存储空间（字）	程序存储器的最后一个地址	高耐用性闪存地址范围 ⁽¹⁾
PIC16(L)F1764	4,096	0FFFh	0F80h-0FFFh
PIC16(L)F1765	8,192	1FFFh	1F80h-1FFFh
PIC16(L)F1768	4,096	0FFFh	0F80h-0FFFh
PIC16(L)F1769	8,192	1FFFh	1F80h-1FFFh

注 1: 高耐用性闪存位于范围内每个地址的低字节。

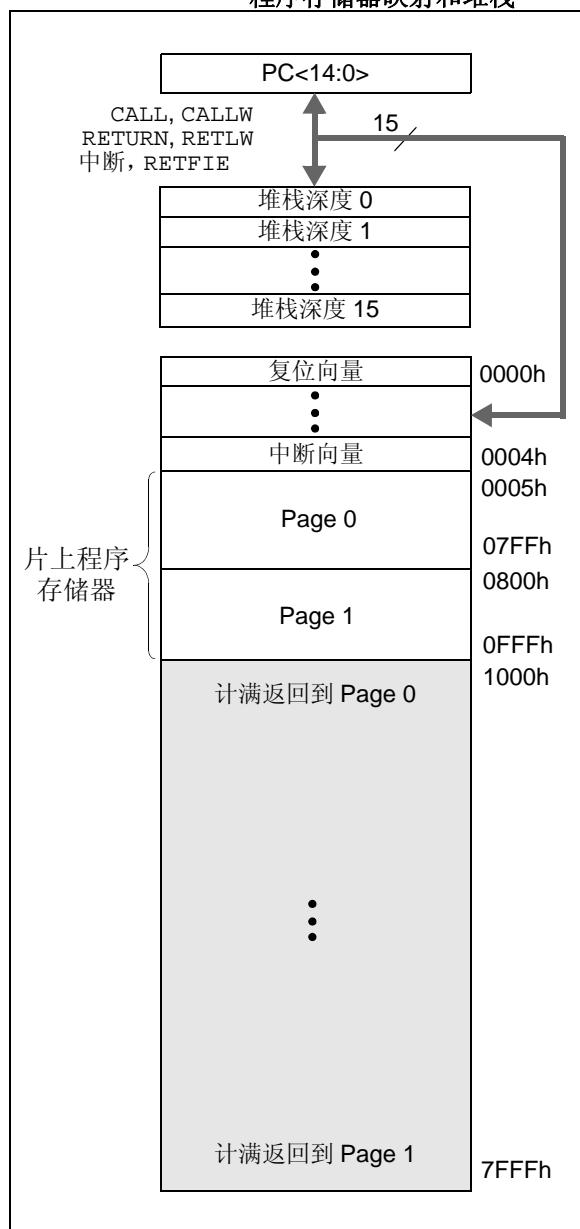
3.1 程序存储器构成

增强型中档内核具有 15 位程序计数器，能够寻址 32K x 14 的程序存储空间。表 3-1 列出了 PIC16(L)F1764/5/8/9 系列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。复位向量位于 0000h，而中断向量位于 0004h（见图 3-1）。

3.2 高耐用性闪存

这些器件具有一个 128 字节的高耐用性闪存程序存储器（Program Flash Memory, PFM）区域，用以代替数据 EEPROM。该区域特别适合于预期在最终产品的整个工作寿命期间频繁更新的非易失性数据存储。关于向 PFM 写入数据的更多信息，请参见第 10.2 节“**闪存程序存储器概述**”。关于使用 FSR 寄存器读取存储在 PFM 中的字节数据的更多信息，请参见第 3.2.1.2 节“**通过 FSR 间接读取**”。

图 3-1: PIC16(L)F1764/5/8/9 的程序存储器映射和堆栈



3.2.1 将程序存储器作为数据进行读取

有两种方法来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。

3.2.1.1 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 3-1 所示。

例 3-1: 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移植性，则 BRW 指令不可用，此时代码必须使用先前的表读方法。

3.2.1.2 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 访问程序存储器的指令需要一个额外的指令周期才能完成。例 3-2 给出了通过 FSR 访问程序存储器的代码。

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 3-2: 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

```

3.3 数据存储器构成

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

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

工作存储区的选择通过向存储区选择寄存器（Bank Select Register, BSR）写入存储区编号来进行。未实现的存储区将读为 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 3.7 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。

3.3.1 内核寄存器

内核寄存器包含会直接影响基本操作的寄存器。内核寄存器占用每个数据存储区的前 12 个地址（地址 x00h/x08h 至 x0Bh/x8Bh）。表 3-2 列出了这些寄存器。详细信息，请参见表 3-15。

表 3-2: 内核寄存器

地址	BANKx
x00h 或 80h	INDF0
x01h 或 81h	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

3.3.1.1 STATUS 寄存器

STATUS 寄存器如寄存器 3-1 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，TO 和 PD 位均为不可写的。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 置位 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 35.0 节“指令集汇总”。

注： 在减法运算中，C 和 DC 位分别作为借位和平借位位。

PIC16(L)F1764/5/8/9

3.4 寄存器定义：状态

寄存器 3-1： 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
—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	$\text{DC}^{(1)}$	$\text{C}^{(1)}$
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

$\overline{\text{TO}}$ ：超时位

1 = 在上电或执行 CLRWDT 指令或 SLEEP 指令后

0 = 发生了 WDT 超时

$\overline{\text{PD}}$ ：掉电位

1 = 在上电或执行 CLRWDT 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 $\overline{\text{借位}}$ ，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。

3.4.1 特殊功能寄存器

特殊功能寄存器（SFR）是由应用对器件中外设功能所需操作进行控制的寄存器。SFR 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）。本数据手册的相应外设章节中介绍了与每个外设操作相关的寄存器。

3.4.2 通用 RAM

每个数据存储区中有最大 80 字节的通用寄存器（GPR）。GPR 占据紧接在选定数据存储区的 SFR 之后的空间。选定存储区数量取决于器件中的可用 GPR 空间的总量。

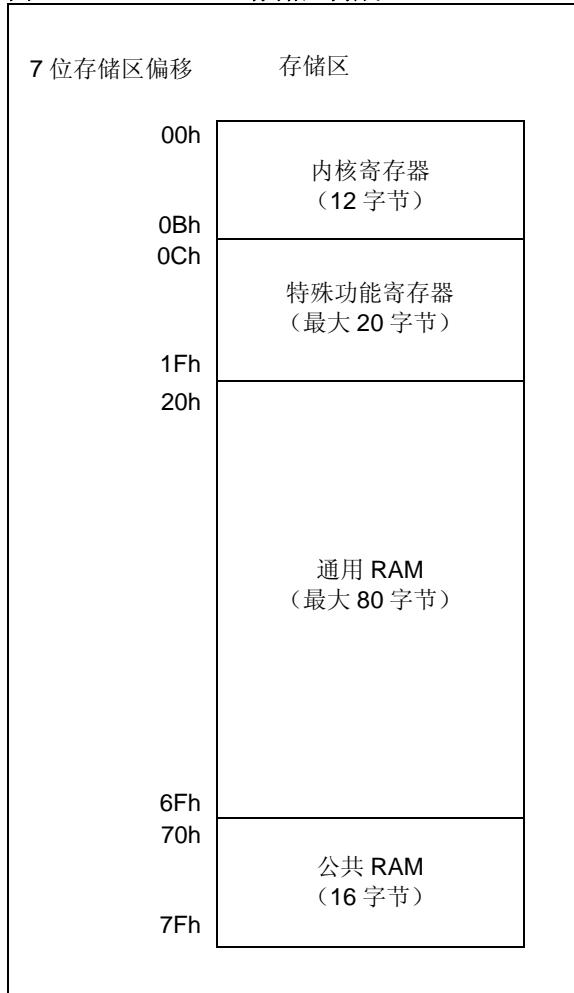
3.4.2.1 线性访问 GPR

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 3.7.2 节“线性数据存储器”](#)。

3.4.3 公共 RAM

有 16 字节的公共 RAM 可以从所有存储区中进行访问。

图 3-2: 存储区构成



3.4.4 器件存储器映射

[表 3-3](#) 至 [表 3-14](#) 给出了器件系列的存储器映射。

PIC16(L)F1764/5/8/9

表 3-3: PIC16(L)F1764 存储器映射 (BANK 0-7)

BANK 0	BANK 1	BANK 2	BANK 3	BANK 4	BANK 5	BANK 6	BANK 7
000h 内核寄存器 (表 3-2)	080h 内核寄存器 (表 3-2)	100h 内核寄存器 (表 3-2)	180h 内核寄存器 (表 3-2)	200h 内核寄存器 (表 3-2)	280h 内核寄存器 (表 3-2)	300h 内核寄存器 (表 3-2)	380h 内核寄存器 (表 3-2)
00Bh PORTA	08Bh TRISA	10Bh LATA	18Bh ANSELA	20Bh WPUA	28Bh ODCONA	30Bh SLRCONA	38Bh INLVLA
00Ch —	08Ch —	10Dh —	18Dh —	20Dh —	28Dh —	30Dh —	38Dh —
00Dh PORTC	08Eh TRISC	10Eh LATC	18Eh ANSEL0	20Eh WPUC	28Eh ODCONC	30Eh SLRCONC	38Eh INLVLC
00Eh —	08Fh —	10Fh CMOUT	18Fh —	20Fh —	28Fh —	30Fh —	38Fh —
010h —	090h CM1CON0	110h CM1CON1	190h —	210h —	290h —	310h —	390h —
011h PIR1	091h PIE1	111h CM1CON1	191h PMADRL	211h SSP1BUF	291h CCP1L	311h —	391h IOCAP
012h PIR2	092h PIE2	112h CM1NSEL	192h PMADRH	212h SSP1ADD	292h CCP1H	312h —	392h IOCAN
013h PIR3	093h PIE3	113h CM1PSEL	193h PMDATL	213h SSP1MSK	293h CCP1CON	313h —	393h IOCASF
014h PIR4	094h PIE4	114h CM2CON0	194h PMDATH	214h SSP1STAT	294h CCP1CAP	314h —	394h —
015h TMR0	095h OPTION_REG	115h CM2CON1	195h PMCON1	215h SSP1CON1	295h —	315h —	395h —
016h TMR1L	096h PCON	116h CM2NSEL	196h PMCON2	216h SSP1CON2	296h —	316h —	396h —
017h TMR1H	097h WDTCON	117h CM2PSEL	197h VREGCON ⁽¹⁾	217h SSP1CON3	297h —	317h —	397h IOCCP
018h T1CON	098h OSCTUNE	118h —	198h —	218h —	298h —	318h —	398h IOCCN
019h T1GCON	099h OSCCON	119h —	199h RC1REG	219h —	299h —	319h —	399h IOCCF
01Ah T2TMR	09Ah OSCSTAT	11Ah —	19Ah TX1REG	21Ah —	29Ah —	31Ah —	39Ah —
01Bh T2PR	09Bh ADRESL	11Bh —	19Bh SP1BRGL	21Bh —	29Bh —	31Bh —	39Bh MD1CON0
01Ch T2CON	09Ch ADRESH	11Ch —	19Ch SP1BRGH	21Ch —	29Ch —	31Ch —	39Ch MD1CON1
01Dh T2HLT	09Dh ADCON0	11Dh —	19Dh RC1STA	21Dh BORCON	29Dh —	31Dh —	39Dh MD1SRC
01Eh T2CLKCON	09Eh ADCON1	11Eh —	19Eh TX1STA	21Eh FVRCON	29Eh CCPTMRS	31Eh —	39Eh MD1CARL
01Fh T2RST	09Fh ADCON2	11Fh —	19Fh BAUD1CON	21Fh ZCD1CON	29Fh —	31Fh —	39Fh MD1CARH
020h 通用寄存器 80 字节	0A0h 通用寄存器 80 字节	120h 通用寄存器 80 字节	1A0h 通用寄存器 80 字节	220h 通用寄存器 80 字节	2A0h 通用寄存器 16 字节	320h 未实现 读为 0	3A0h 未实现 读为 0
06Fh 公共 RAM 70h – 7Fh	0EFh 快速操作存储区 70h – 7Fh	16Fh 快速操作存储区 70h – 7Fh	1EFh 快速操作存储区 70h – 7Fh	26Fh 快速操作存储区 70h – 7Fh	2EFh 快速操作存储区 70h – 7Fh	36Fh 快速操作存储区 70h – 7Fh	3EFh 快速操作存储区 70h – 7Fh
070h	0F0h	170h	1FFh	270h	2F0h	370h	3F0h
07Fh	OFFh	17Fh		27Fh	2FFh	37Fh	3FFh

图注: █ = 未实现的数据存储单元, 读为 0。

注 1: PIC16LF1764 上未实现。

表 3-4: PIC16LF1765 存储器映射 (BANK 0-7)

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	内核寄存器 (表 3-2)	080h	内核寄存器 (表 3-2)	100h	内核寄存器 (表 3-2)	180h	内核寄存器 (表 3-2)	200h	内核寄存器 (表 3-2)	280h	内核寄存器 (表 3-2)	300h	内核寄存器 (表 3-2)	380h	内核寄存器 (表 3-2)
00Bh	PORTA	08Bh	TRISA	10Bh	LATA	18Bh	ANSELA	20Bh	WPUA	28Bh	ODCONA	30Bh	SLRCONA	38Bh	INLVLA
00Ch	—	08Ch	—	10Dh	—	18Dh	—	20Dh	—	28Ch	—	30Ch	SLRCONA	38Ch	—
00Dh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Dh	ODCONC	30Dh	—	38Dh	INLVLC
00Fh	—	08Fh	—	10Fh	CMOUT	18Fh	—	20Fh	—	28Eh	—	30Eh	SLRCONC	38Eh	—
010h	—	090h	—	110h	CM1CON0	190h	—	210h	—	28Fh	—	30Fh	—	38Fh	—
011h	PIR1	091h	PIE1	111h	CM1CON1	191h	PMADRL	211h	SSP1BUF	290h	—	310h	—	390h	—
012h	PIR2	092h	PIE2	112h	CM1NSEL	192h	PMADRH	212h	SSP1ADD	291h	CCPR1L	311h	—	391h	IOCAP
013h	PIR3	093h	PIE3	113h	CM1PSEL	193h	PMDATL	213h	SSP1MSK	292h	CCPR1H	312h	—	392h	IOCAN
014h	PIR4	094h	PIE4	114h	CM2CON0	194h	PMDATH	214h	SSP1STAT	293h	CCP1CON	313h	—	393h	IOCAF
015h	TMR0	095h	OPTION_REG	115h	CM2CON1	195h	PMCON1	215h	SSP1CON	294h	CCP1CAP	314h	—	394h	—
016h	TMR1L	096h	PCON	116h	CM2NSEL	196h	PMCON2	216h	SSP1CON2	295h	—	315h	—	395h	—
017h	TMR1H	097h	WDTCON	117h	CM2PSEL	197h	VREGCON ⁽¹⁾	217h	SSP1CON3	296h	—	316h	—	396h	—
018h	T1CON	098h	OSCTUNE	118h	—	198h	—	218h	—	297h	—	317h	—	397h	IOCCP
019h	T1GCON	099h	OSCCON	119h	—	199h	RC1REG	219h	—	298h	—	318h	—	398h	IOCCN
01Ah	T2TMR	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	299h	—	319h	—	399h	IOCCF
01Bh	T2PR	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	—	29Ah	—	31Ah	—	39Ah	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SP1BRGH	21Ch	—	29Bh	—	31Bh	—	39Bh	MD1CON0
01Dh	T2HLT	09Dh	ADCON0	11Dh	—	19Dh	RC1STA	21Dh	BORCON	29Ch	—	31Ch	—	39Ch	MD1CON1
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	TX1STA	21Eh	FVRCON	29Dh	—	31Dh	—	39Dh	MD1SRC
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	BAUD1CON	21Fh	ZCD1CON	29Eh	CCPTMRS	31Eh	—	39Eh	MD1CARL
020h		0A0h		120h		1A0h		220h		2A0h		320h		3A0h	MD1CARH
06Fh	通用寄存器 80 字节	0EFh	通用寄存器 80 字节	16Fh	通用寄存器 80 字节	1EFh	通用寄存器 80 字节	26Fh	通用寄存器 80 字节	2EFh	通用寄存器 80 字节	36Fh	通用寄存器 80 字节	3EFh	通用寄存器 80 字节
070h	公共 RAM 70h – 7Fh	0F0h	快速操作存储区 70h – 7Fh	170h	快速操作存储区 70h – 7Fh	1FFh	快速操作存储区 70h – 7Fh	27Fh	快速操作存储区 70h – 7Fh	2F0h	快速操作存储区 70h – 7Fh	370h	快速操作存储区 70h – 7Fh	3F0h	快速操作存储区 70h – 7Fh
07Fh		0FFh		17Fh						2FFh		37Fh		3FFh	

图注: = 未实现的数据存储单元, 读为 0。

注 1: PIC16LF1765 上未实现。

PIC16(L)F1764/5/8/9

表 3-5: PIC16(L)F1768 存储器映射 (BANK 0-7)

BANK 0	BANK 1	BANK 2	BANK 3	BANK 4	BANK 5	BANK 6	BANK 7
000h 内核寄存器 (表 3-2)	080h 内核寄存器 (表 3-2)	100h 内核寄存器 (表 3-2)	180h 内核寄存器 (表 3-2)	200h 内核寄存器 (表 3-2)	280h 内核寄存器 (表 3-2)	300h 内核寄存器 (表 3-2)	380h 内核寄存器 (表 3-2)
00Bh	08Bh	10Bh	18Bh	20Bh	28Bh	30Bh SLRCONA	38Bh
00Ch PORTA	08Ch TRISA	10Ch LATB	18Ch ANSELA	20Ch WPUB	28Ch ODCONA	30Ch SLRCONB	38Ch INLVLA
00Dh PORTB	08Dh TRISB	10Dh LATC	18Dh ANSELB	20Dh WPUC	28Dh ODCONB	30Dh SLRCONC	38Dh INLVLB
00Eh PORTC	08Eh TRISC	10Eh CMOUT	18Eh ANSELC	20Eh —	28Eh ODCONC	30Eh SLRCONC	38Eh INLVLC
00Fh —	08Fh —	10Fh CM1CON0	18Fh —	20Fh —	28Fh —	30Fh —	38Fh —
010h —	090h —	110h CM1CON1	190h —	210h —	290h —	310h —	390h —
011h PIR1	091h PIE1	111h CM1CON1	191h PMADRL	211h SSP1BUF	291h CCPR1L	311h —	391h IOCAP
012h PIR2	092h PIE2	112h CM1NSEL	192h PMADRH	212h SSP1ADD	292h CCPR1H	312h —	392h OCAN
013h PIR3	093h PIE3	113h CM1PSEL	193h PMDATL	213h SSP1MSK	293h CCP1CON	313h —	393h OCAF
014h PIR4	094h PIE4	114h CM2CON0	194h PMDATH	214h SSP1STAT	294h CCP1CAP	314h —	394h OCBP
015h TMR0	095h OPTION_REG	115h CM2CON1	195h PMCON1	215h SSP1CON1	295h —	315h —	395h OCBN
016h TMR1L	096h PCON	116h CM2NSEL	196h PMCON2	216h SSP1CON2	296h —	316h —	396h OCBF
017h TMR1H	097h WDTCON	117h CM2PSEL	197h VREGCON ⁽¹⁾	217h SSP1CON3	297h —	317h —	397h OCCP
018h T1CON	098h OSCTUNE	118h CM3CON0	198h —	218h —	298h CCPR2L	318h —	398h OCCN
019h T1GCON	099h OSCCON	119h CM3CON1	199h RC1REG	219h —	299h CCPR2H	319h —	399h OCCF
01Ah T2TMR	09Ah OSCSTAT	11Ah CM3NSEL	19Ah TX1REG	21Ah —	29Ah CCP2CON	31Ah —	39Ah —
01Bh T2PR	09Bh ADRESL	11Bh CM3PSEL	19Bh SP1BRGL	21Bh —	29Bh CCP2CAP	31Bh MD2CON0	39Bh MD1CON0
01Ch T2CON	09Ch ADRESH	11Ch CM4CON0	19Ch SP1BRGH	21Ch —	29Ch —	31Ch MD2CON1	39Ch MD1CON1
01Dh T2HLT	09Dh ADCON0	11Dh CM4CON1	19Dh RC1STA	21Dh BORCON	29Dh —	31Dh MD2SRC	39Dh MD1SRC
01Eh T2CLKCON	09Eh ADCON1	11Eh CM4NSEL	19Eh TX1STA	21Eh FVRCON	29Eh CCPTMRS	31Eh MD2CARL	39Eh MD1CARL
01Fh T2RST	09Fh ADCON2	11Fh CM4PSEL	19Fh BAUD1CON	21Fh ZCD1CON	29Fh —	31Fh MD2CARH	39Fh MD1CARH
020h	0A0h	120h	1A0h	220h	2A0h	320h 通用寄存器 16 字节	3A0h
06Fh 070h 公共 RAM 70h - 7Fh	0EFh 0F0h 快速操作存储区 70h - 7Fh	16Fh 170h 快速操作存储区 70h - 7Fh	1EFh 1F0h 快速操作存储区 70h - 7Fh	26Fh 270h 快速操作存储区 70h - 7Fh	2EFh 2F0h 快速操作存储区 70h - 7Fh	32Fh 330h 未实现 读为 0	3EFh 3F0h 快速操作存储区 70h - 7Fh
07Fh	0FFh	17Fh	1FFh	27Fh	2FFh	36Fh 370h 快速操作存储区 70h - 7Fh	3FFh

图注: ■ = 未实现的数据存储单元, 读为 0。

注 1: PIC16LF1768 上未实现。

表 3-6: PIC16(L)F1769 存储器映射 (BANK 0-7)

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	内核寄存器 (表 3-2)	080h	内核寄存器 (表 3-2)	100h	内核寄存器 (表 3-2)	180h	内核寄存器 (表 3-2)	200h	内核寄存器 (表 3-2)	280h	内核寄存器 (表 3-2)	300h	内核寄存器 (表 3-2)	380h	内核寄存器 (表 3-2)
00Bh	PORTA	08Bh	TRISA	10Bh	LATA	18Bh	ANSELA	20Bh	WPUA	28Bh	ODCONA	30Bh	SLRCONA	38Bh	INLVLA
00Ch	PORTB	08Ch	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Ch	ODCONB	30Ch	SLRCONB	38Ch	INLVLB
00Dh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Dh	ODCONC	30Dh	SLRCONC	38Dh	INLVLC
00Eh	—	08Fh	—	10Fh	CMOUT	18Fh	—	20Fh	—	28Eh	—	30Eh	—	38Eh	—
010h	—	090h	—	110h	CM1CON0	190h	—	210h	—	290h	—	310h	—	38Fh	—
011h	PIR1	091h	PIE1	111h	CM1CON1	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	390h	—
012h	PIR2	092h	PIE2	112h	CM1NSEL	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	391h	IOCAP
013h	PIR3	093h	PIE3	113h	CM1PSEL	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	392h	IOCAN
014h	PIR4	094h	PIE4	114h	CM2CON0	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	393h	IOCAF
015h	TMR0	095h	OPTION_REG	115h	CM2CON1	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	394h	IOCBP
016h	TMR1L	096h	PCON	116h	CM2NSEL	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	395h	IOCBN
017h	TMR1H	097h	WDTCON	117h	CM2PSEL	197h	VREGCON ⁽¹⁾	217h	SSP1CON3	297h	—	317h	—	396h	IOCBF
018h	T1CON	098h	OSCTUNE	118h	CM3CON0	198h	—	218h	—	298h	CCPR2L	318h	—	397h	IOCCP
019h	T1GCON	099h	OSCCON	119h	CM3CON1	199h	RC1REG	219h	—	299h	CCPR2H	319h	—	398h	IOCCN
01Ah	T2TMR	09Ah	OSCSTAT	11Ah	CM3NSEL	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—	399h	IOCCF
01Bh	T2PR	09Bh	ADRESL	11Bh	CM3PSEL	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	MD2CON0	39Ah	—
01Ch	T2CON	09Ch	ADRESH	11Ch	CM4CON0	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	MD2CON1	39Bh	MD1CON0
01Dh	T2HLT	09Dh	ADCON0	11Dh	CM4CON1	19Dh	RC1STA	21Dh	BORCON	29Dh	—	31Dh	MD2SRC	39Ch	MD1CON1
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	CM4NSEL	19Eh	TX1STA	21Eh	FVRCON	29Eh	CCPTMRS	31Eh	MD2CARL	39Dh	MD1SRC
01Fh	T2RST	09Fh	ADCON2	11Fh	CM4PSEL	19Fh	BAUD1CON	21Fh	ZCD1CON	29Fh	—	31Fh	MD2CARH	39Eh	MD1CARL
020h		0A0h		120h		1A0h		220h		2A0h		320h		3A0h	通用寄存器 80 字节
06Fh	通用寄存器 80 字节	0EFh	通用寄存器 80 字节	16Fh	通用寄存器 80 字节	1EFh	通用寄存器 80 字节	26Fh	通用寄存器 80 字节	2EFh	通用寄存器 80 字节	36Fh	通用寄存器 80 字节	3EFh	通用寄存器 80 字节
070h	公共 RAM 70h – 7Fh	0F0h	快速操作存储区 70h – 7Fh	170h	快速操作存储区 70h – 7Fh	1FFh	快速操作存储区 70h – 7Fh	270h	快速操作存储区 70h – 7Fh	2F0h	快速操作存储区 70h – 7Fh	370h	快速操作存储区 70h – 7Fh	3F0h	快速操作存储区 70h – 7Fh
07Fh		0FFh		17Fh				27Fh		2FFh		37Fh		3FFh	

图注: █ = 未实现的数据存储单元, 读为 0。

注 1: PIC16LF1769 上未实现。

PIC16(L)F1764/5/8/9

表 3-7: PIC16(L)F1764 存储器映射, BANK 8-23

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	内核寄存器 (表 3-2)	480h	内核寄存器 (表 3-2)	500h	内核寄存器 (表 3-2)	580h	内核寄存器 (表 3-2)	600h	内核寄存器 (表 3-2)	680h	内核寄存器 (表 3-2)	700h	内核寄存器 (表 3-2)	780h	内核寄存器 (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Dh	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Eh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Fh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	610h	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACLD	611h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	612h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	613h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	—	593h	DAC1REFH	614h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	—	594h	—	615h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	—	595h	—	616h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	—	596h	—	617h	PWM3DCL	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	618h	PWM3DCH	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	619h	PWM3CON	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	—	61Ah	—	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	—	61Bh	—	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	—
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Ch	—	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	—
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Dh	—	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	—
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Eh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	—
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Fh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	—
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	620h	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	—
420h	未实现 读为 0	4A0h	未实现 读为 0	520h	未实现 读为 0	5A0h	未实现 读为 0	620h	未实现 读为 0	6A0h	未实现 读为 0	720h	未实现 读为 0	7A0h	未实现 读为 0
46Fh	—	4EFh	—	56Fh	—	5EFh	—	66Fh	—	6EFh	未实现 读为 0	76Fh	快速操作存储区 70h - 7Fh	7EFh	—
470h	快速操作存储区 70h - 7Fh	4F0h	快速操作存储区 70h - 7Fh	570h	快速操作存储区 70h - 7Fh	5FFh	快速操作存储区 70h - 7Fh	67Fh	快速操作存储区 70h - 7Fh	6FFh	快速操作存储区 70h - 7Fh	77Fh	快速操作存储区 70h - 7Fh	7FFh	快速操作存储区 70h - 7Fh
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	内核寄存器 (表 3-2)	880h	内核寄存器 (表 3-2)	900h	内核寄存器 (表 3-2)	980h	内核寄存器 (表 3-2)	A00h	内核寄存器 (表 3-2)	A80h	内核寄存器 (表 3-2)	B00h	内核寄存器 (表 3-2)	B80h	内核寄存器 (表 3-2)
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	未实现 读为 0	B0Bh	未实现 读为 0	B8Bh	未实现 读为 0
80Ch	未实现 读为 0	88Ch	未实现 读为 0	90Ch	未实现 读为 0	98Ch	未实现 读为 0	A0Ch	未实现 读为 0	A8Ch	未实现 读为 0	B0Ch	未实现 读为 0	B8Ch	未实现 读为 0
86Fh	—	8EFh	—	96Fh	—	9EFh	—	A6Fh	—	AEFh	未实现 读为 0	B6Fh	快速操作存储区 70h - 7Fh	BEFh	—
870h	快速操作存储区 70h - 7Fh	8F0h	快速操作存储区 70h - 7Fh	970h	快速操作存储区 70h - 7Fh	9FFh	快速操作存储区 70h - 7Fh	A7Fh	快速操作存储区 70h - 7Fh	AFFh	快速操作存储区 70h - 7Fh	B7Fh	快速操作存储区 70h - 7Fh	BFFh	快速操作存储区 70h - 7Fh
87Fh	—	8FFh	—	97Fh	—	9FFh	—	A7Fh	—	AFFh	—	B7Fh	—	BFFh	—

图注: = 未实现的数据存储单元, 读为 0。

表 3-8: PIC16(L)F1765 存储器映射, BANK 8-23

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	内核寄存器 (表 3-2)	480h	内核寄存器 (表 3-2)	500h	内核寄存器 (表 3-2)	580h	内核寄存器 (表 3-2)	600h	内核寄存器 (表 3-2)	680h	内核寄存器 (表 3-2)	700h	内核寄存器 (表 3-2)	780h	内核寄存器 (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Dh	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Eh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Fh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	610h	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACLD	611h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	612h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	613h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	—	593h	DAC1REFH	614h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	—	594h	—	615h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	—	595h	—	616h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	—	596h	—	617h	PWM3DCL	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	618h	PWM3DCH	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	619h	PWM3CON	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	—	61Ah	—	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	—	61Bh	—	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	—
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Ch	—	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	—
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Dh	—	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	—
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Eh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	—
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Fh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	—
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	620h	通用寄存器 48 字节	6A0h	COG1STR	71Fh	COG2STR	79Fh	—
420h	通用寄存器 80 字节	4A0h	通用寄存器 80 字节	520h	通用寄存器 80 字节	5A0h	通用寄存器 80 字节	64Fh	未实现 读为 0	6E0h	未实现 读为 0	720h	未实现 读为 0	7A0h	未实现 读为 0
46Fh	快速操作存储区 70h – 7Fh	4EFh	快速操作存储区 70h – 7Fh	56Fh	快速操作存储区 70h – 7Fh	550h	快速操作存储区 70h – 7Fh	66Fh	快速操作存储区 70h – 7Fh	6F0h	快速操作存储区 70h – 7Fh	76Fh	快速操作存储区 70h – 7Fh	7F0h	快速操作存储区 70h – 7Fh
47Fh	快速操作存储区 70h – 7Fh	4FFh	快速操作存储区 70h – 7Fh	57Fh	快速操作存储区 70h – 7Fh	5FFh	快速操作存储区 70h – 7Fh	67Fh	快速操作存储区 70h – 7Fh	6FFh	快速操作存储区 70h – 7Fh	77Fh	快速操作存储区 70h – 7Fh	7FFh	快速操作存储区 70h – 7Fh
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	内核寄存器 (表 3-2)	880h	内核寄存器 (表 3-2)	900h	内核寄存器 (表 3-2)	980h	内核寄存器 (表 3-2)	A00h	内核寄存器 (表 3-2)	A80h	内核寄存器 (表 3-2)	B00h	内核寄存器 (表 3-2)	B80h	内核寄存器 (表 3-2)
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	—	B0Bh	—	B8Bh	—
80Ch	未实现 读为 0	88Ch	未实现 读为 0	90Ch	未实现 读为 0	98Ch	未实现 读为 0	A0Ch	未实现 读为 0	A8Ch	未实现 读为 0	B0Ch	未实现 读为 0	B8Ch	未实现 读为 0
86Fh	快速操作存储区 70h – 7Fh	8EFh	快速操作存储区 70h – 7Fh	96Fh	快速操作存储区 70h – 7Fh	970h	快速操作存储区 70h – 7Fh	9EFh	快速操作存储区 70h – 7Fh	9F0h	快速操作存储区 70h – 7Fh	A6Fh	快速操作存储区 70h – 7Fh	B6Fh	快速操作存储区 70h – 7Fh
870h	快速操作存储区 70h – 7Fh	8F0h	快速操作存储区 70h – 7Fh	97Fh	快速操作存储区 70h – 7Fh	9FFh	快速操作存储区 70h – 7Fh	A7Fh	快速操作存储区 70h – 7Fh	AFFh	快速操作存储区 70h – 7Fh	B7Fh	快速操作存储区 70h – 7Fh	BFFh	快速操作存储区 70h – 7Fh
87Fh	快速操作存储区 70h – 7Fh	8FFh	快速操作存储区 70h – 7Fh	97Fh	快速操作存储区 70h – 7Fh	9FFh	快速操作存储区 70h – 7Fh	A7Fh	快速操作存储区 70h – 7Fh	AFFh	快速操作存储区 70h – 7Fh	B7Fh	快速操作存储区 70h – 7Fh	BFFh	快速操作存储区 70h – 7Fh

图注: = 未实现的数据存储单元, 读为 0。

PIC16(L)F1764/5/8/9

表 3-9: PIC16(L)F1768 存储器映射, BANK 8-23

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	内核寄存器 (表 3-2)	480h	内核寄存器 (表 3-2)	500h	内核寄存器 (表 3-2)	580h	内核寄存器 (表 3-2)	600h	内核寄存器 (表 3-2)	680h	内核寄存器 (表 3-2)	700h	内核寄存器 (表 3-2)	780h	内核寄存器 (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Dh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Eh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Fh	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	610h	DACLD	690h	COG1BLKF	710h	COG2BLKF	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DAC1CON0	611h	—	691h	COG1DBR	711h	COG2DBR	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1REFL	612h	—	692h	COG1DBF	712h	COG2DBF	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	792h	—
413h	T4TMR	493h	TMR3L	513h	OPA2NCHS	593h	DAC1REFH	614h	—	694h	COG1CON1	714h	COG2CON1	793h	—
414h	T4PR	494h	TMR3H	514h	OPA2PCHS	594h	DAC2CON0	615h	DAC2REFL	695h	COG1RIS0	715h	COG2RIS0	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	OPA2CON	595h	DAC2REFH	616h	—	696h	COG1RIS1	716h	COG2RIS1	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	OPA2ORS	596h	DAC3CON0	617h	PWM3DCL	697h	COG1RSIM0	717h	COG2RSIM0	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3REF	618h	PWM3DCH	698h	COG1RSIM1	718h	COG2RSIM1	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC4CON0	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	DAC4REF	61Ah	PWM4DCL	69Ah	COG1FIS1	71Ah	COG2FIS1	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	—	61Bh	PWM4DCH	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Ah	PRG2RTSS
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Ch	PWM4CON	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Bh	PRG2FTSS
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Ch	PRG2INS
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Dh	PRG2CON0
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Eh	PRG2CON1
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	620h	—	6A0h	未实现 读为 0	720h	未实现 读为 0	79Fh	PRG2CON2
420h	未实现 读为 0	4A0h	未实现 读为 0	520h	未实现 读为 0	5A0h	未实现 读为 0	620h	未实现 读为 0	6A0h	未实现 读为 0	720h	未实现 读为 0	7A0h	未实现 读为 0
46Fh	快速操作存储区 70h - 7Fh		4EFh	快速操作存储区 70h - 7Fh		56Fh	快速操作存储区 70h - 7Fh		5EFh	快速操作存储区 70h - 7Fh		66Fh	快速操作存储区 70h - 7Fh		
470h	快速操作存储区 70h - 7Fh		4F0h	快速操作存储区 70h - 7Fh		57Fh	快速操作存储区 70h - 7Fh		5F0h	快速操作存储区 70h - 7Fh		67Fh	快速操作存储区 70h - 7Fh		
47Fh	快速操作存储区 70h - 7Fh		4FFh	快速操作存储区 70h - 7Fh		5FFh	快速操作存储区 70h - 7Fh		6FFh	快速操作存储区 70h - 7Fh		77Fh	快速操作存储区 70h - 7Fh		
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	内核寄存器 (表 3-2)	880h	内核寄存器 (表 3-2)	900h	内核寄存器 (表 3-2)	980h	内核寄存器 (表 3-2)	A00h	内核寄存器 (表 3-2)	A80h	内核寄存器 (表 3-2)	B00h	内核寄存器 (表 3-2)	B80h	内核寄存器 (表 3-2)
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	—	B0Bh	—	B8Bh	—
80Ch	未实现 读为 0	88Ch	未实现 读为 0	90Ch	未实现 读为 0	98Ch	未实现 读为 0	A0Ch	未实现 读为 0	A8Ch	未实现 读为 0	B0Ch	未实现 读为 0	B8Ch	未实现 读为 0
86Fh	快速操作存储区 70h - 7Fh		8EFh	快速操作存储区 70h - 7Fh		96Fh	快速操作存储区 70h - 7Fh		9F0h	快速操作存储区 70h - 7Fh		A6Fh	快速操作存储区 70h - 7Fh		
870h	快速操作存储区 70h - 7Fh		8F0h	快速操作存储区 70h - 7Fh		97Fh	快速操作存储区 70h - 7Fh		9FFh	快速操作存储区 70h - 7Fh		A70h	快速操作存储区 70h - 7Fh		
87Fh	快速操作存储区 70h - 7Fh		8FFh	快速操作存储区 70h - 7Fh		97Fh	快速操作存储区 70h - 7Fh		9FFh	快速操作存储区 70h - 7Fh		A7Fh	快速操作存储区 70h - 7Fh		
图注:		■ = 未实现的数据存储单元, 读为 0。													

表 3-10: PIC16(L)F1769 存储器映射, BANK 8-23

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	内核寄存器 (表 3-2)	480h	内核寄存器 (表 3-2)	500h	内核寄存器 (表 3-2)	580h	内核寄存器 (表 3-2)	600h	内核寄存器 (表 3-2)	680h	内核寄存器 (表 3-2)	700h	内核寄存器 (表 3-2)	780h	内核寄存器 (表 3-2)
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	60Fh	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACLD	610h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	611h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	612h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	OPA2NCHS	593h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	OPA2PCHS	594h	DAC2CON0	614h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	OPA2CON	595h	DAC2REFL	615h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	OPA2ORS	596h	DAC2REFH	616h	—	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	617h	PWM3DCL	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	618h	PWM3DCH	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	DAC4CON0	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	DAC4REF	61Ah	PWM4DCL	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	PRG2RTSS
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Bh	PWM4DCH	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	PRG2FTSS
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Ch	PWM4CON	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	PRG2INS
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	PRG2CON0
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	PRG2CON1
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	PRG2CON2
420h	通用寄存器 80 字节	4A0h	通用寄存器 80 字节	520h	通用寄存器 80 字节	5A0h	通用寄存器 80 字节	620h	通用寄存器 48 字节	6A0h	未实现 读为 0	720h	未实现 读为 0	7A0h	未实现 读为 0
46Fh	快速操作存储区 70h – 7Fh	4EFh	快速操作存储区 70h – 7Fh	56Fh	快速操作存储区 70h – 7Fh	5F0h	快速操作存储区 70h – 7Fh	66Fh	快速操作存储区 70h – 7Fh	6F0h	未实现 读为 0	76Fh	快速操作存储区 70h – 7Fh	7F0h	快速操作存储区 70h – 7Fh
47Fh	快速操作存储区 70h – 7Fh	4FFh	快速操作存储区 70h – 7Fh	57Fh	快速操作存储区 70h – 7Fh	5FFh	快速操作存储区 70h – 7Fh	67Fh	快速操作存储区 70h – 7Fh	6FFh	快速操作存储区 70h – 7Fh	77Fh	快速操作存储区 70h – 7Fh	7FFh	快速操作存储区 70h – 7Fh
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	内核寄存器 (表 3-2)	880h	内核寄存器 (表 3-2)	900h	内核寄存器 (表 3-2)	980h	内核寄存器 (表 3-2)	A00h	内核寄存器 (表 3-2)	A80h	内核寄存器 (表 3-2)	B00h	内核寄存器 (表 3-2)	B80h	内核寄存器 (表 3-2)
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	—	B0Bh	—	B8Bh	—
80Ch	未实现 读为 0	88Ch	未实现 读为 0	90Ch	未实现 读为 0	98Ch	未实现 读为 0	A0Ch	未实现 读为 0	A8Ch	未实现 读为 0	B0Ch	未实现 读为 0	B8Ch	未实现 读为 0
86Fh	快速操作存储区 70h – 7Fh	8EFh	快速操作存储区 70h – 7Fh	96Fh	快速操作存储区 70h – 7Fh	9FFh	快速操作存储区 70h – 7Fh	A6Fh	快速操作存储区 70h – 7Fh	AEFh	快速操作存储区 70h – 7Fh	B6Fh	快速操作存储区 70h – 7Fh	BFFh	快速操作存储区 70h – 7Fh
870h	快速操作存储区 70h – 7Fh	8F0h	快速操作存储区 70h – 7Fh	970h	快速操作存储区 70h – 7Fh	9F0h	快速操作存储区 70h – 7Fh	A7Fh	快速操作存储区 70h – 7Fh	AF0h	快速操作存储区 70h – 7Fh	B7Fh	快速操作存储区 70h – 7Fh	BF0h	快速操作存储区 70h – 7Fh
87Fh	快速操作存储区 70h – 7Fh	8FFh	快速操作存储区 70h – 7Fh	97Fh	—	9FFh	—	A7Fh	—	AFFh	—	B7Fh	—	BFFh	—

图注:

= 未实现的数据存储单元, 读为 0。

PIC16(L)F1764/5/8/9

表 3-11: PIC16(L)F1764/5/8/9 存储器映射, BANK 24-31

BANK 24	BANK 25	BANK 26	BANK 27	BANK 28	BANK 29	BANK 30	BANK 31
C00h C0Bh C0Ch C0Dh C0Eh C0Fh C10h C11h C12h C13h C14h C15h C16h C17h C18h C19h C1Ah C1Bh C1Ch C1Dh C1Eh C1Fh C20h	C80h C8Bh C8Ch C8Dh C8Eh C8Fh C90h C91h C92h C93h C94h C95h C96h C97h C98h C99h C9Ah C9Bh C9Ch C9Dh C9Eh C9Fh CA0h	D00h D0Bh D0Ch D0Dh D0Eh D0Fh D10h D11h D12h D13h D14h D15h D16h D17h D18h D19h D1Ah D1Bh D1Ch D1Dh D1Eh D1Fh D20h	D80h D8Bh	E00h E0Bh	E80h E8Bh	F00h F0Bh	F80h F8Bh
内核寄存器 (表 3-2)	内核寄存器 (表 3-2)	内核寄存器 (表 3-2)	内核寄存器 (表 3-2)	内核寄存器 (表 3-2)	内核寄存器 (表 3-2)	内核寄存器 (表 3-2)	内核寄存器 (表 3-2)
—	—	—	—	—	—	—	—
未实现 读为 0	未实现 读为 0	未实现 读为 0	DEFh DF0h	E6Fh E70h	EEFh EF0h	F6Fh F70h	FEFh FF0h
快速操作存储区 70h – 7Fh	CEFh CF0h	快速操作存储区 70h – 7Fh	D6Fh D70h	快速操作存储区 70h – 7Fh	EEFh EF0h	快速操作存储区 70h – 7Fh	快速操作存储区 70h – 7Fh
CFFh	CFFh	D7Fh	DFFh	E7Fh	EFFh	F7Fh	FFFh
							请参见表 3-14 了 解寄存器映射详 细信息

图注: = 未实现的数据存储单元, 读为 0。

表 3-12: PIC16(L)F1764/5 存储器映射, BANK 27-30

	Bank 27	Bank 28	Bank 29	Bank 30
D8Ch	—	E0Ch	—	F0Ch
D8Dh	—	E0Dh	—	F0Dh
D8Eh	PWMEN	E0Eh	—	F0Eh
D8Fh	PWMLD	E0Fh	PPSLOCK	F0Fh CLC1DATA
D90h	PWMOUT	E10h	INTPPS	F10h CLC1CON
D91h	PWM5PHL	E11h	T0CKIPPS	F11h CLC1POL
D92h	PWM5PHH	E12h	T1CKIPPS	F12h CLC1SEL0
D93h	PWM5DCL	E13h	T1GPPS	F13h CLC1SEL1
D94h	PWM5DCH	E14h	CCP1PPS	F14h CLC1SEL2
D95h	PWM5PRL	E15h	—	F15h CLC1SEL3
D96h	PWM5PRH	E16h	COG1INPPS	F16h CLC1GLS0
D97h	PWM5OFL	E17h	—	F17h CLC1GLS1
D98h	PWM5OFH	E18h	—	F18h CLC1GLS2
D99h	PWM5TMRL	E19h	T2CKIPPS	F19h CLC1GLS3
D9Ah	PWM5TMRH	E1Ah	T3CKIPPS	F1Ah CLC2CON
D9Bh	PWM5CON	E1Bh	T3GPPS	F1Bh CLC2POL
D9Ch	PWM5INTE	E1Ch	T4CKIPPS	F1Ch CLC2SEL0
D9Dh	PWM5INTF	E1Dh	T5CKIPPS	F1Dh CLC2SEL1
D9Eh	PWM5CLKCON	E1Eh	T5GPPS	F1Eh CLC2SEL2
D9Fh	PWM5LDCON	E1Fh	T6CKIPPS	F1Fh CLC2SEL3
DA0h	PWM5OFCON	E20h	SSPCLKPPS	F20h CLC2GLS0
DA1h	—	E21h	SSPDATPPS	F21h CLC2GLS1
DA2h	—	E22h	SSPSSPPS	F22h CLC2GLS2
DA3h	—	E23h	—	F23h CLC2GLS3
DA4h	—	E24h	RXPPS	F24h CLC3CON
DA5h	—	E25h	CKPPS	F25h CLC3POL
DA6h	—	E26h	—	F26h CLC3SEL0
DA7h	—	E27h	—	F27h CLC3SEL1
DA8h	—	E28h	CLCIN0PPS	F28h CLC3SEL2
DA9h	—	E29h	CLCIN1PPS	F29h CLC3SEL3
DAAh	—	E2Ah	CLCIN2PPS	F2Ah CLC3GLS0
DABh	—	E2Bh	CLCIN3PPS	F2Bh CLC3GLS1
DACh	—	E2Ch	PRG1FPPS	F2Ch CLC3GLS2
DADh	—	E2Dh	PRG1RPPS	F2Dh CLC3GLS3
DAEh	—	E2Eh	—	F2Eh —
DAFh	—	E2Fh	—	F2Fh —
DB0h	—	E30h	MD1CHPPS	F30h —
DB1h	—	E31h	MD1CLPPS	F31h —
DB2h	—	E32h	MD1MODPPS	F32h —
DB3h	—	E33h	—	F33h —
DB4h	—	E34h	—	F34h —
DB5h	—	E35h	—	F35h —
DB6h	—	E36h	—	F36h —
DB7h	—	E37h	—	F37h —
DB8h	—	E38h	—	F38h —
DB9h	—	E39h	—	F39h —
DBAh	—	E3Ah	—	F3Ah —
DBBh	—	E3Bh	—	F3Bh —
DBCh	—	E3Ch	—	F3Ch —
DBDh	—	E3Dh	—	F3Dh —
DBEh	—	E3Eh	—	F3Eh —
DBFh	—	E3Fh	—	F3Fh —
DC0h	—	E40h	—	F40h —
DEFh	—	E6Fh	—	F6Fh —

图注:  = 未实现的数据存储单元, 读为 0。

PIC16(L)F1764/5/8/9

表 3-13: PIC16(L)F1768/9 存储器映射, BANK 27-30

Bank 27	Bank 28	Bank 29	Bank 30
D8Ch	—	E0Ch	—
D8Dh	—	E0Dh	—
D8Eh	PWMEN	E0Eh	—
D8Fh	PWMLD	E0Fh	PPSLOCK
D90h	PWMOUT	E10h	INTPPS
D91h	PWM5PHL	E11h	T0CKIPPS
D92h	PWM5PHH	E12h	T1CKIPPS
D93h	PWM5DCL	E13h	T1GPPS
D94h	PWM5DCH	E14h	CCP1PPS
D95h	PWM5PRL	E15h	CCP2PPS
D96h	PWM5PRH	E16h	COG1INPPS
D97h	PWM5OFL	E17h	COG2INPPS
D98h	PWM5OFH	E18h	—
D99h	PWM5TMRL	E19h	T2CKIPPS
D9Ah	PWM5TMRH	E1Ah	T3CKIPPS
D9Bh	PWM5CON	E1Bh	T3GPPS
D9Ch	PWM5INTE	E1Ch	T4CKIPPS
D9Dh	PWM5INTF	E1Dh	T5CKIPPS
D9Eh	PWM5CLKCON	E1Eh	T5GPPS
D9Fh	PWM5LDCON	E1Fh	T6CKIPPS
DA0h	PWM5OFCON	E20h	SSPCLKPPS
DA1h	PWM6PHL	E21h	SSPDATPPS
DA2h	PWM6PHH	E22h	SSPSSPPS
DA3h	PWM6DCL	E23h	—
DA4h	PWM6DCH	E24h	RXPPS
DA5h	PWM6PRL	E25h	CKPPS
DA6h	PWM6PRH	E26h	—
DA7h	PWM6OFL	E27h	—
DA8h	PWM6OFH	E28h	CLCIN0PPS
DA9h	PWM6TMRL	E29h	CLCIN1PPS
DAAh	PWM6TMRH	E2Ah	CLCIN2PPS
DABh	PWM6CON	E2Bh	CLCIN3PPS
DACh	PWM6INTE	E2Ch	PRG1FPPS
DADh	PWM6INTF	E2Dh	PRG1RPPS
DAEh	PWM6CLKCON	E2Eh	PRG2FPPS
DAFh	PWM6LDCON	E2Fh	PRG2RPPS
DB0h	PWM6OFCON	E30h	MD1CHPPS
DB1h	—	E31h	MD1CLPPS
DB2h	—	E32h	MD1MODPPS
DB3h	—	E33h	MD2CHPPS
DB4h	—	E34h	MD2CLPPS
DB5h	—	E35h	MD2MODPPS
DB6h	—	E36h	—
DB7h	—	E37h	—
DB8h	—	E38h	—
DB9h	—	E39h	—
DBAh	—	E3Ah	—
DBBh	—	E3Bh	—
DBC	—	E3Ch	—
DBDh	—	E3Dh	—
DBEh	—	E3Eh	—
DBFh	—	E3Fh	—
DC0h	—	E40h	—
DEFh	—	E6Fh	—
		EEFh	—
		EC0h	—
			F6Fh

图注: = 未实现的数据存储单元, 读为 0。

表 3-14: PIC16(L)F1764/5/8/9 存储器映射,
BANK 31

Bank 31	
F8Ch	未实现 读为 0
FE3h	
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEKh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

图注:  = 未实现的数据存储单元, 读为 0。

PIC16(L)F1764/5/8/9

3.4.5 内核功能寄存器汇总

表3-15列出了内核功能寄存器，可从任何存储区寻址这些寄存器。

表 3-15： 内核功能寄存器汇总⁽¹⁾

地址	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
Bank 0-31											
x00h 或 80h	INDF0	使用 FSR0H/FSR0L 的内容寻址该存储单元来寻址数据存储器 (不是物理寄存器)								xxxx xxxx	uuuu uuuu
x01h 或 81h	INDF1	使用 FSR1H/FSR1L 的内容寻址该存储单元来寻址数据存储器 (不是物理寄存器)								xxxx xxxx	uuuu uuuu
x02h 或 x82h	PCL	程序计数器 (Program Counter, PC) 的最低有效字节								0000 0000	0000 0000
x03h 或 x83h	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu
x04h 或 x84h	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu
x05h 或 x85h	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000
x06h 或 x86h	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu
x07h 或 x87h	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000
x08h 或 x88h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000
x09h 或 x89h	WREG	工作寄存器								0000 0000	uuuu uuuu
x0Ah 或 x8Ah	PCLATH	—	程序计数器高 7 位的写缓冲区							-000 0000	-000 0000
x0Bh 或 x8Bh	INTCON	GIE	PEIE	TMROIE	INTE	IOCIE	TMROIF	INTF	IOCIF	0000 0000	0000 0000

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

注 1： 可从任何存储区寻址这些寄存器。

表 3-16: 特殊功能寄存器汇总

地址	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		
Bank 0													
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu		
00Dh	PORTB ⁽²⁾	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ---	uuuu ---		
00Eh	PORTC	RC7 ⁽²⁾	RC6 ⁽²⁾	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu		
00Fh	—	未实现							—	—	—		
010h	—	未实现							—	—	—		
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000		
012h	PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽²⁾	C3IF ⁽²⁾	CCP2IF ⁽²⁾	000- 0000	000- 0000		
013h	PIR3	PWM6IF ⁽²⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽²⁾	CLC3IF	CLC2IF	CLC1IF	0000 0000	0000 0000		
014h	PIR4	—	—	TMR5GIF	TMR5IF	TMR3GIF	TMR3IF	TMR6IF	TRM4IF	--00 0000	--00 0000		
015h	TMR0	Timer0 模块寄存器							0000 0000	0000 0000	0000 0000		
016h	TMR1L	16 位 TMR1 寄存器最低有效字节的保持寄存器							xxxx xxxx	uuuu uuuu	uuuu uuuu		
017h	TMR1H	16 位 TMR1 寄存器最高有效字节的保持寄存器							xxxx xxxx	uuuu uuuu	uuuu uuuu		
018h	T1CON	CS<1:0>		CKPS<1:0>		OSCEN	SYNC	—	ON	0000 00-0	uuuu uu-u		
019h	T1GCON	GE	GPOL	GTM	GSPM	GO/DONE	GVAL	GSS<1:0>		0000 0x00	uuuu uxuu		
01Ah	T2TMR	8 位 TMR2 寄存器的保持寄存器							0000 0000	0000 0000	0000 0000		
01Bh	T2PR	TMR2 周期寄存器							1111 1111	1111 1111	1111 1111		
01Ch	T2CON	ON	CKPS<2:0>		OUTPS<3:0>				0000 0000	0000 0000	0000 0000		
01Dh	T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>				--0 0000	--0 0000	--0 0000		
01Eh	T2CLKCON	—	—	—	—	CS<3:0>		RSEL<3:0>		---- 0000	---- 0000		
01Fh	T2RST	—	—	—	—	RSEL<3:0>				---- 0000	---- 0000		
Bank 1													
08Ch	TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111		
08Dh	TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----		
08Eh	TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	1111 1111	1111 1111		
08Fh	—	未实现							—	—	—		
090h	—	未实现							—	—	—		
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000		
092h	PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽²⁾	C3IE ⁽²⁾	CCP2IE ⁽²⁾	000- 0000	000- 0000		
093h	PIE3	PWM6IE ⁽²⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽²⁾	CLC3IE	CLC2IE	CLC1IE	0000 0000	0000 0000		
094h	PIE4	—	—	TMR5GIE	TMR5IE	TMR3GIE	TMR3IE	TMR6IE	TRM4IE	--00 0000	--00 0000		
095h	OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>		1111 1111	1111 1111	1111 1111		
096h	PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	00-1 11qq	qq-q qqqu		
097h	WDTCON	—	—	WDTPS<4:0>				SWDTEN		--01 0110	--01 0110		
098h	OSCTUNE	—	—	TUN<5:0>				--00 0000		--00 0000	--00 0000		
099h	OSCCON	SPLLLEN	IRCF<3:0>		—	SCS<1:0>		0011 1-00	0011 1-00	0011 1-00	0011 1-00		
09Ah	OSCSTAT	SOSCR	PLLR	OSTS	HFIOPR	HFIOLF	MFIOFR	LFIOFR	HFIOFS	00q0 0q0q	qqqq qq0q		
09Bh	ADRESL	ADC 结果寄存器的低字节							xxxx xxxx	uuuu uuuu	uuuu uuuu		
09Ch	ADRESH	ADC 结果寄存器的高字节							xxxx xxxx	uuuu uuuu	uuuu uuuu		
09Dh	ADCON0	—	CHS<4:0>				GO/DONE	ADON	-000 0000	-000 0000	-000 0000		
09Eh	ADCON1	ADFM	ADCS<2:0>		—	ADNREF	ADPREF<1:0>		0000 -000	0000 -000	0000 -000		
09Fh	ADCON2	TRIGSEL<4:0>				—	—	—	0000 0---	0000 0---	0000 0---		

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

PIC16(L)F1764/5/8/9

表 3-16: 特殊功能寄存器汇总 (续)

地址	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			
Bank 2														
10Ch	LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx	--uu -uuu			
10Dh	LATB ⁽²⁾	LATB7	LATB6	LATB5	LATB4	—	—	—	—	xxxx ----	uuuu -----			
10Eh	LATC	LATC7 ⁽²⁾	LATC6 ⁽²⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	uuuu uuuu			
10Fh	CMOUT	—	—	—	—	MC4OUT ⁽²⁾	MC3OUT ⁽²⁾	MC2OUT	MC1OUT	---- --00	---- --00			
110h	CM1CON0	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	00-0 0100	00-0 0100			
111h	CM1CON1	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00			
112h	CM1NSEL	—	—	—	—	—	NCH<2:0>			---- -000	---- -000			
113h	CM1PSEL	—	—	—	—	PCH<2:0>				---- 0000	---- 0000			
114h	CM2CON0	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	00-0 0100	00-0 0100			
115h	CM2CON1	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00			
116h	CM2NSEL	—	—	—	—	—	NCH<2:0>			---- -000	---- -000			
117h	CM2PSEL	—	—	—	—	PCH<2:0>				---- 0000	---- 0000			
118h	CM3CON0 ⁽²⁾	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	00-0 0100	00-0 0100			
119h	CM3CON1 ⁽²⁾	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00			
11Ah	CM3NSEL ⁽²⁾	—	—	—	—	—	NCH<2:0>			---- -000	---- -000			
11Bh	CM3PSEL ⁽²⁾	—	—	—	—	PCH<2:0>				---- 0000	---- 0000			
11Ch	CM4CON0 ⁽²⁾	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	00-0 0100	00-0 0100			
11Dh	CM4CON1 ⁽²⁾	—	—	—	—	—	—	INTP	INTN	---- --00	---- --00			
11Eh	CM4NSEL ⁽²⁾	—	—	—	—	—	NCH<2:0>			---- -000	---- -000			
11Fh	CM4PSEL ⁽²⁾	—	—	—	—	PCH<2:0>				---- 0000	---- 0000			
Bank 3														
18Ch	ANSELA	—	—	—	ANS4	—	ANS2	ANS1	ANS0	--l 1111	--l 1111			
18Dh	ANSELB ⁽²⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	1111 ----	1111 -----			
18Eh	ANSELC	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	11-- 1111	11-- 1111			
18Fh	—	未实现								—	—			
190h	—	未实现								—	—			
191h	PMADRL	程序存储器地址寄存器的低字节								0000 0000	0000 0000			
192h	PMADRH	(1)	程序存储器地址寄存器的高字节								1000 0000			
193h	PMDATL	程序存储器读数据寄存器的低字节								xxxx xxxx	uuuu uuuu			
194h	PMDATH	—	—	程序存储器读数据寄存器的高字节										
195h	PMCON1	(1)	CFG5	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000			
196h	PMCON2	程序存储器控制寄存器 2								0000 0000	0000 0000			
197h	VREGCON ⁽⁴⁾	—	—	—	—	—	—	VREGPM	保留	---- --01	---- --01			
198h	—	未实现								—	—			
199h	RC1REG	EUSART 接收数据寄存器								0000 0000	0000 0000			
19Ah	TX1REG	EUSART 发送数据寄存器								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 0000	0000 0000			
19Eh	TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	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 = 值取决于具体条件, — = 未实现, 读为 0, r = 保留。
阴影单元未实现, 读为 0。

注 1: 未实现位, 读为 1。

2: 仅限 PIC16(L)F1768/9。

3: 仅限 PIC16(L)F1764/5。

4: PIC16LF1764/5/8/9 上未实现。

表 3-16: 特殊功能寄存器汇总 (续)

地址	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			
Bank 4														
20Ch	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--11 1111	--11 1111			
20Dh	WPUB ⁽²⁾	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----			
20Eh	WPUC	WPUC7 ⁽²⁾	WPUC6 ⁽²⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	1111 1111	1111 1111			
20Fh	—	未实现								—	—			
210h	—	未实现								—	—			
211h	SSP1BUF	同步串行端口接收缓冲区 / 发送寄存器								xxxx xxxx	uuuu uuuu			
212h	SSP1ADD	ADD<7:0>								0000 0000	0000 0000			
213h	SSP1MSK	MSK<7:0>								1111 1111	1111 1111			
214h	SSP1STAT	SMP	CKE	D/A	P	S	R/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 21Ch	—	未实现								—	—			
21Dh	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	10-- ---q	uu-- ---u			
21Eh	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		0q00 0000	0q00 0000			
21Fh	ZCD1CON	EN	—	OUT	POL	—	—	INTP	INTN	0-x0 --00	0-x0 --00			
Bank 5														
28Ch	ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	--00 -000	--00 -000			
28Dh	ODCONB ⁽²⁾	ODB7	ODB6	ODB5	ODB4	—	—	—	—	0000 ----	0000 ----			
28Eh	ODCONC	ODC7 ⁽²⁾	ODC6 ⁽²⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000 0000	0000 0000			
28Fh	—	未实现								—	—			
290h	—	未实现								—	—			
291h	CCPR1L	捕捉 / 比较 /PWM 寄存器 1 (LSB)								xxxx xxxx	uuuu uuuu			
292h	CCPR1H	捕捉 / 比较 /PWM 寄存器 1 (MSB)								xxxx xxxx	uuuu uuuu			
293h	CCP1CON	EN	—	OUT	FMT	MODE<3:0>				0-00 0000	0-00 0000			
294h	CCP1CAP	—	—	—	—	—	CTS<2:0>			---- -000	---- -000			
295h 297h	—	未实现								—	—			
298h	CCPR2L ⁽²⁾	捕捉 / 比较 /PWM 寄存器 2 (LSB)								xxxx xxxx	uuuu uuuu			
299h	CCPR2H ⁽²⁾	捕捉 / 比较 /PWM 寄存器 2 (MSB)								xxxx xxxx	uuuu uuuu			
29Ah	CCP2CON ⁽²⁾	EN	—	OUT	FMT	MODE<3:0>				0-00 0000	0-00 0000			
29Bh	CCP2CAP ⁽²⁾	—	—	—	—	—	CTS<2:0>			---- -000	---- -000			
29Ch 29Dh	—	未实现								—	—			
29Eh	CCPTMRS	P4TSEL<1:0> ⁽²⁾		P3TSEL<1:0>		C2TSEL<1:0> ⁽²⁾		C1TSEL<1:0>		0000 0000	0000 0000			
29Fh	—	未实现								—	—			

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

PIC16(L)F1764/5/8/9

表 3-16：特殊功能寄存器汇总（续）

地址	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
Bank 6											
30Ch	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	--11 1111	--11 1111
30Dh	SLRCONB ⁽²⁾	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	1111 ----	1111 ----
30Eh	SLRCONC	SLRC7 ⁽²⁾	SLRC6 ⁽²⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	1111 1111
30Fh — 31Ah	—	未实现								—	—
31Bh	MD2CON0 ⁽²⁾	EN	—	OUT	OPOL	—	—	—	BIT	0-00 ---0	0-00 ---0
31Ch	MD2CON1 ⁽²⁾	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	--00 --00	--00 --00
31Dh	MD2SRC ⁽²⁾	—	—	—	MS<4:0>					---0 0000	---0 0000
31Eh	MD2CARL ⁽²⁾	—	—	—	—	CL<3:0>			---- 0000	---- 0000	---- 0000
31Fh	MD2CARH ⁽²⁾	—	—	—	—	CH<3:0>			---- 0000	---- 0000	---- 0000
Bank 7											
38Ch	INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	--11 1111	--11 1111
38Dh	INLVLB ⁽²⁾	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	1111 ----	1111 ----
38Eh	INLVLC	INLVLC7 ⁽²⁾	INLVLC6 ⁽²⁾	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 ⁽²⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	0000 ----
395h	IOCBN ⁽²⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	0000 ----
396h	IOCBF ⁽²⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	0000 ----
397h	IOCCP	IOCCP7 ⁽²⁾	IOCCP6 ⁽²⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
398h	IOCCN	IOCCN7 ⁽²⁾	IOCCN6 ⁽²⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
399h	IOCCF	IOCCF7 ⁽²⁾	IOCCF6 ⁽²⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000
39Ah	—	未实现								—	—
39Bh	MD1CON0	EN	—	OUT	OPOL	—	—	—	BIT	0-00 ---0	0-00 ---0
39Ch	MD1CON1	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	--00 --00	--00 --00
39Dh	MD1SRC	—	—	—	MS<4:0>					---0 0000	---0 0000
39Eh	MD1CARL	—	—	—	—	CL<3:0>			---- 0000	---- 0000	---- 0000
39Fh	MD1CARH	—	—	—	—	CH<3:0>			---- 0000	---- 0000	---- 0000

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

- 注 1：未实现位，读为 1。
 2：仅限 PIC16(L)F1768/9。
 3：仅限 PIC16(L)F1764/5。
 4：PIC16LF1764/5/8/9 上未实现。

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
----	----	-------	-------	-------	-------	-------	-------	-------	-------	---------------	-----------

Bank 8

40Ch — 40Dh	—	未实现								—	—
40Eh	HIDRVC	—	—	HIDC5	HIDC4	—	—	—	—	--00 ----	--00 ----
40Fh — 412h	—	未实现								—	—
413h	T4TMR	8 位 TMR4 寄存器的保持寄存器								0000 0000	0000 0000
413h	T4PR	TMR4 周期寄存器								1111 1111	1111 1111
415h	T4CON	ON	CKPS<2:0>			OUTPS<3:0>					0000 0000
416h	T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>						---0 0000
417h	T4CLKCON	—	—	—	—	CS<3:0>					---- 0000
418h	T4RST	—	—	—	—	RSEL<3:0>					---- 0000
419h	—	未实现								—	—
41Ah	T6TMR	8 位 TMR4 寄存器的保持寄存器								0000 0000	0000 0000
41Bh	T6PR	TMR4 周期寄存器								1111 1111	1111 1111
41Ch	T6CON	ON	CKPS<2:0>			OUTPS<3:0>					0000 0000
41Dh	T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>						---0 0000
41Eh	T6CLKCON	—	—	—	—	CS<3:0>					---- 0000
41Fh	T6RST	—	—	—	—	RSEL<3:0>					---- 0000

Bank 9

48Ch 至 492h	—	未实现								—	—
493h	TMR3L	16 位 TMR1 寄存器最低有效字节的保持寄存器								xxxx xxxx	uuuu uuuu
494h	TMR3H	16 位 TMR1 寄存器最高有效字节的保持寄存器								xxxx xxxx	uuuu uuuu
495h	T3CON	CS<1:0>		CKPS<1:0>			OSCEN	SYNC	—	ON	0000 00-0
496h	T3GCON	GE	GPOl	GTM	GSPM	GGO/ DONE	GVAL	GSS<1:0>			0000 0x00
497h 至 499h	—	未实现								—	—
49Ah	TMR5L	16 位 TMR1 寄存器最低有效字节的保持寄存器								xxxx xxxx	uuuu uuuu
49Bh	TMR5H	16 位 TMR1 寄存器最高有效字节的保持寄存器								xxxx xxxx	uuuu uuuu
49Ch	T5CON	CS<1:0>		CKPS<1:0>			OSCEN	SYNC	—	ON	0000 00-0
49Dh	T5GCON	GE	GPOl	GTM	GSPM	GGO/ DONE	GVAL	GSS<1:0>			0000 0x00
49Eh 至 49Fh	—	未实现								—	—

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

PIC16(L)F1764/5/8/9

表 3-16：特殊功能寄存器汇总（续）

地址	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
Bank 10											
50Ch — 50Eh	—	未实现								—	—
50Fh	OPA1NCHS	—	—	—	—	NCH<3:0>			---- 0000	---- 0000	
510h	OPA1PCHS	—	—	—	—	PCH<3:0>			---- 0000	---- 0000	
511h	OPA1CON	EN	—	—	UG	—	ORPOL	ORM<1:0>	0--0 -000	0--0 -000	
512h	OPA1ORS	—	—	—		ORS<4:0>			---0 0000	---0 0000	
513h	OPA2NCHS ⁽²⁾	—	—	—	—	NCH<3:0>			---- 0000	---- 0000	
514h	OPA2PCHS ⁽²⁾	—	—	—	—	PCH<3:0>			---- 0000	---- 0000	
515h	OPA2CON ⁽²⁾	EN	—	—	UG	—	ORPOL	ORM<1:0>	0--0 -000	0--0 -000	
516h	OPA2ORS ⁽²⁾	—	—	—		ORS<4:0>			---0 0000	---0 0000	
517h — 51Fh	—	未实现							—	—	
Bank 11											
590h	DACLD	---	---	---	---	---	---	DAC2LD ⁽²⁾	DAC1LD	---- --00	---- --00
591h	DAC1CON0	EN	FM	OE1	---	PSS<1:0>	—	NSS<1:0>	000- 0000	000- 0000	
592h	DAC1REFL				REF<7:0>				00000 0000	0000 0000	
593h	DAC1REFH				REF<15:8>				00000 0000	0000 0000	
594h	DAC2CON0 ⁽²⁾	EN	FM	OE1	---	PSS<1:0>	—	NSS<1:0>	000- 0000	000- 0000	
595h	DAC2REFL ⁽²⁾				REF<7:0>				00000 0000	0000 0000	
596h	DAC2REFH ⁽²⁾				REF<15:8>				00000 0000	0000 0000	
597h	DAC3CON0	EN	---	OE1	---	PSS<1:0>	---	NSS	0-0- 00-0	0-00 00-0	
598h	DAC3REF	---	---	---		REF<4:0>			---0 0000	0000 0000	
599h	DAC4CON0 ⁽²⁾	EN	---	OE1	---	PSS<1:0>	---	NSS	0-0- 00-0	0-00 00-0	
59Ah	DAC4REF ⁽²⁾	---	---	---		REF<4:0>			---0 0000	0000 0000	
59Bh 至 59Fh	—	未实现							—	—	

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

注 1：未实现位，读为 1。

2：仅限 PIC16(L)F1768/9。

3：仅限 PIC16(L)F1764/5。

4：PIC16LF1764/5/8/9 上未实现。

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
Bank 12											
60Ch 至 616h	—	未实现								—	—
617h	PWM3DCL	DC<1:0>	—	—	—	—	—	—	—	xx-- -----	uu-- -----
618h	PWM3DCH		DC<9:2>							xxxx xxxx	uuuu uuuu
619h	PWM3CON	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
61Ah	PWM4DCL ⁽²⁾	DC<1:0>	—	—	—	—	—	—	—	00-- -----	uu-- -----
61Bh	PWM4DCH ⁽²⁾		DC<9:2>							0000 0000	uuuu uuuu
61Ch	PWM4CON ⁽²⁾	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
61Dh — 61Fh	—	未实现								—	—
Bank 13											
68Ch	—	未实现								—	—
68Dh	COG1PHR	—	—	COG 上升沿相位延时计数寄存器						--00 0000	--00 0000
68Eh	COG1PHF	—	—	COG 下降沿相位延时计数寄存器						--00 0000	--00 0000
68Fh	COG1BLKR	—	—	COG 上升沿消隐计数寄存器						--00 0000	--00 0000
690h	COG1BLKF	—	—	COG 下降沿消隐计数寄存器						--00 0000	--00 0000
691h	COG1DBR	—	—	COG 上升沿死区计数寄存器						--00 0000	--00 0000
692h	COG1DBF	—	—	COG 下降沿死区计数寄存器						--00 0000	--00 0000
693h	COG1CON0	EN	LD	—	CS<1:0>		MD<2:0>			00-0 0000	00-0 0000
694h	COG1CON1	RDBS	FDBS	—	—	POLD	POLC	POLB	POLA	00-- 0000	00-- 0000
695h	COG1RIS0	RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0	0000 0000	-000 0000
696h	COG1RIS1	—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8	--00 0000	-000 0000
697h	COG1RSIM0	RSIM7	RSIM6	RSIM5	RSIM4	RSIM3	RSIM2	RSIM1	RSIM0	0000 0000	-000 0000
698h	COG1RSIM1	—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8	--00 0000	-000 0000
699h	COG1FIS0	FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0	0000 0000	-000 0000
69Ah	COG1FIS1	—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8	--00 0000	-000 0000
69Bh	COG1FSIM0	FSIM7	FSIM6	FSIM5	FSIM4	FSIM3	FSIM2	FSIM1	FSIM0	0000 0000	-000 0000
69Ch	COG1FSIM1	—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8	--00 0000	-000 0000
69Dh	COG1ASD0	ASE	ARSEN	ASDBD<1:0>		ASDAC<1:0>	—	—	—	0001 01--	0001 01--
69Eh	COG1ASD1	AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	0000 0000	0000 0000
69Fh	COG1STR	SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA	0000 0000	0000 0000

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

PIC16(L)F1764/5/8/9

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
Bank 14											
70Ch	—	未实现								—	—
70Dh	COG2PHR ⁽²⁾	—	—	COG 上升沿相位延时计数寄存器						--00 0000	--00 0000
70Eh	COG2PHF ⁽²⁾	—	—	COG 下降沿相位延时计数寄存器						--00 0000	--00 0000
70Fh	COG2BLKR ⁽²⁾	—	—	COG 上升沿消隐计数寄存器						--00 0000	--00 0000
710h	COG2BLKF ⁽²⁾	—	—	COG 下降沿消隐计数寄存器						--00 0000	--00 0000
711h	COG2DBR ⁽²⁾	—	—	COG 上升沿死区计数寄存器						--00 0000	--00 0000
712h	COG2DBF ⁽²⁾	—	—	COG 下降沿死区计数寄存器						--00 0000	--00 0000
713h	COG2CON0 ⁽²⁾	EN	LD	—	CS<1:0>		MD<2:0>			00-0 0000	00-0 0000
714h	COG2CON1 ⁽²⁾	RDBS	FDBS	—	—	POLD	POLC	POLB	POLA	00-- 0000	00-- 0000
715h	COG2RIS0 ⁽²⁾	RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0	0000 0000	-000 0000
716h	COG2RIS1 ⁽²⁾	—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8	--00 0000	-000 0000
717h	COG2RSIM0 ⁽²⁾	RSIM7	RSIM6	RSIM5	RSIM4	RSIM3	RSIM2	RSIM1	RSIM0	0000 0000	-000 0000
718h	COG2RSIM1 ⁽²⁾	—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8	--00 0000	-000 0000
719h	COG2FIS0 ⁽²⁾	FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0	0000 0000	-000 0000
71Ah	COG2FIS1 ⁽²⁾	—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8	--00 0000	-000 0000
71Bh	COG2FSIM0 ⁽²⁾	FSIM7	FSIM6	FSIM5	FSIM4	FSIM3	FSIM2	FSIM1	FSIM0	0000 0000	-000 0000
71Ch	COG2FSIM1 ⁽²⁾	—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8	--00 0000	-000 0000
71Dh	COG2ASD0 ⁽²⁾	ASE	ARSEN	ASDBD<1:0>	ASDAC<1:0>	—	—	—	—	0001 01--	0001 01--
71Eh	COG2ASD1 ⁽²⁾	AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	0000 0000	0000 0000
71Fh	COG2STR ⁽²⁾	SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA	0000 0000	0000 0000
Bank 15											
78Ch — 793h	—	未实现								—	—
794h	PRG1RTSS	—	—	—	—	RTSS<3:0>			---- 0000	---- 0000	
795h	PRG1FTSS	—	—	—	—	FTSS<3:0>			---- 0000	---- 0000	
796h	PRG1INS	—	—	—	—	—	INS<2:0>		---- -000	---- -000	
797h	PRG1CON0	EN	—	FEDG	REDG	MODE<1:0>	OS	GO	0-000 0000	0-00 0000	
798h	PRG1CON1	—	—	—	—	—	RDY	FPOL	RPOL	---- -000	---- -000
799h	PRG1CON2	—	—	—	—	ISET<4:0>			--0 0000	--0 0000	
79Ah	PRG2RTSS ⁽²⁾	—	—	—	—	RTSS<3:0>			---- 0000	---- 0000	
79Bh	PRG2FTSS ⁽²⁾	—	—	—	—	FTSS<3:0>			---- 0000	---- 0000	
79Ch	PRG2INS ⁽²⁾	—	—	—	—	—	INS<2:0>		---- -000	---- -000	
79Dh	PRG2CON0 ⁽²⁾	EN	—	FEDG	REDG	MODE<1:0>	OS	GO	0-000 0000	0-00 0000	
79Eh	PRG2CON1 ⁽²⁾	—	—	—	—	—	RDY	FPOL	RPOL	---- -000	---- -000
79Fh	PRG2CON2 ⁽²⁾	—	—	—	—	ISET<4:0>			--0 0000	--0 0000	
Bank 16-26											
x0Ch/ x8Ch — x1Fh/ x9Fh	—	未实现								—	—

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
Bank 27											
D8Ch — D8Dh	—	未实现								—	—
D8Eh	PWMEN	—	—	MPWM6EN ⁽²⁾	MPWM5EN	—	—	—	—	--00 ----	--00 ----
D8Fh	PWMLD	—	—	MPWM6LD ⁽²⁾	MPWM5LD	—	—	—	—	--00 ----	--00 ----
D90h	PWMOUT	—	—	MPWM6OUT ⁽²⁾	MPWM5OUT	—	—	—	—	--00 ----	--00 ----
D91h	PWM5PHL				PH<7:0>					xxxx xxxx	uuuu uuuu
D92h	PWM5PHH				PH<15:8>					xxxx xxxx	uuuu uuuu
D93h	PWM5DCL				DC<7:0>					xxxx xxxx	uuuu uuuu
D94h	PWM5DCH				DC<15:8>					xxxx xxxx	uuuu uuuu
D95h	PWM5PRL				PR<7:0>					xxxx xxxx	uuuu uuuu
D96h	PWM5PRH				PR<15:8>					xxxx xxxx	uuuu uuuu
D97h	PWM5OFL				OF<7:0>					xxxx xxxx	uuuu uuuu
D98h	PWM5OFH				OF<15:8>					xxxx xxxx	uuuu uuuu
D99h	PWM5TMRL				TMR<7:0>					0000 0000	0000 0000
D9Ah	PWM5TMRH				TMR<15:8>					0000 0000	0000 0000
D9Bh	PWM5CON	EN	—	OUT	POL	MODE<1:0>	—	—	—	0-00 00--	0-00 00--
D9Ch	PWM5INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 0000	---- 0000
D9Dh	PWM5INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 0000	---- 0000
D9Eh	PWM5CLKCON	—		PS<2:0>		—	—		CS<1:0>	-000 --00	-000 --00
D9Fh	PWM5LDCON	LDA	LDT ⁽²⁾	—	—	—	—	—	LDS ⁽²⁾	00-- ---0	00-- ---0
DA0h	PWM5OFCON	—		OFM<1:0> ⁽²⁾	OFO	—	—	—	OFS ⁽²⁾	-000 ---0	-000 ---0
DA1h	PWM6PHL ⁽²⁾				PH<7:0>					xxxx xxxx	uuuu uuuu
DA2h	PWM6PHH ⁽²⁾				PH<15:8>					xxxx xxxx	uuuu uuuu
DA3h	PWM6DCL ⁽²⁾				DC<7:0>					xxxx xxxx	uuuu uuuu
DA4h	PWM6DCH ⁽²⁾				DC<15:8>					xxxx xxxx	uuuu uuuu
DA5h	PWM6PRL ⁽²⁾				PR<7:0>					xxxx xxxx	uuuu uuuu
DA6h	PWM6PRH ⁽²⁾				PR<15:8>					xxxx xxxx	uuuu uuuu
DA7h	PWM6OFL ⁽²⁾				OF<7:0>					xxxx xxxx	uuuu uuuu
DA8h	PWM6OFH ⁽²⁾				OF<15:8>					xxxx xxxx	uuuu uuuu
DA9h	PWM6TMRL ⁽²⁾				TMR<7:0>					0000 0000	0000 0000
DAAh	PWM6TMRH ⁽²⁾				TMR<15:8>					0000 0000	0000 0000
DABh	PWM6CON ⁽²⁾	EN	—	OUT	POL	MODE<1:0>	—	—	—	0-00 00--	0-00 00--
DACh	PWM6INTE ⁽²⁾	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 0000	---- 0000
DADh	PWM6INTF ⁽²⁾	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 0000	---- 0000
DAEh	PWM6CLKCON ⁽²⁾	—		PS<2:0>		—	—		CS<1:0>	-000 --00	-000 --00
DAFh	PWM6LDCON ⁽²⁾	LDA	LDT	—	—	—	—	—	LDS	00-- ---0	00-- ---0
DB0h	PWM6OFCON ⁽²⁾	—		OFM<1:0>	OFO	—	—	—	OFS	-000 ---0	-000 ---0
DB1h 至 DBFh	—	未实现								—	—

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

PIC16(L)F1764/5/8/9

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
Bank 28											
E0Ch — E0Eh	—	未实现								—	—
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 0101	---u uuuu
E15h	CCP2PPS ⁽²⁾	—	—	—		CCP2PPS<4:0>				---1 0011	---u uuuu
E16h	COG1INPPS	—	—	—		COG1INPPS<4:0>				---0 0010	---u uuuu
E17h	COG2INPPS ⁽²⁾	—	—	—		COG2INPPS<4:0>				---0 0010	---u uuuu
E18h	—	未实现								—	—
E19h	T2CKIPPS	—	—	—		T2CKIPPS<4:0>				---0 0101	---u uuuu
E1Ah	T3CKIPPS	—	—	—		T3CKIPPS<4:0>				---1 0101	---u uuuu
E1Bh	T3GPPS	—	—	—		T3GPPS<4:0>				---1 0100	---u uuuu
E1Ch	T4CKIPPS	—	—	—		T4CKIPPS<4:0>				---1 0001	---u uuuu
E1Dh	T5CKIPPS	—	—	—		T5CKIPPS<4:0>				---1 0000	---u uuuu
E1Eh	T5GPPS	—	—	—		T5GPPS<4:0>				---1 0011	---u uuuu
E1Fh	T6CKIPPS	—	—	—		T6CKIPPS<4:0>				---0 0011	---u uuuu
E20h	SSPCLKPPS	—	—	—		SSPCLKPPS<4:0>				---1 0000 ⁽³⁾	---u uuuu
		—	—	—		SSPCLKPPS<4:0>				---0 1110 ⁽²⁾	---u uuuu
E21h	SSPDATPPS	—	—	—		SSPDATPPS<4:0>				---1 0001 ⁽³⁾	---u uuuu
		—	—	—		SSPDATPPS<4:0>				---0 1100 ⁽²⁾	---u uuuu
E22h	SSPSSPPS	—	—	—		SSPSSPPS<4:0>				---1 0011 ⁽³⁾	---u uuuu
		—	—	—		SSPSSPPS<4:0>				---1 0110 ⁽²⁾	---u uuuu
E23h	—	未实现								—	—
E24h	RXPPS	—	—	—		RXPPS<4:0>				---1 0101 ⁽³⁾	---u uuuu
		—	—	—		RXPPS<4:0>				---0 1101 ⁽²⁾	---u uuuu
E25h	CKPPS	—	—	—		CKPPS<4:0>				---1 0100 ⁽³⁾	---u uuuu
		—	—	—		CKPPS<4:0>				---0 1111 ⁽²⁾	---u uuuu
E26h	—	未实现								—	—
E27h	—	未实现								—	—
E28h	CLCIN0PPS	—	—	—		CLCIN0PPS<4:0>				---1 0011	---u uuuu
E29h	CLCIN1PPS	—	—	—		CLCIN1PPS<4:0>				---1 0100	---u uuuu
E2Ah	CLCIN2PPS	—	—	—		CLCIN2PPS<4:0>				---1 0001	---u uuuu
E2Bh	CLCIN3PPS	—	—	—		CLCIN3PPS<4:0>				---0 0101	---u uuuu
E2Ch	PRG1RPPS	—	—	—		PRG1RPPS<4:0>				---1 0100	---u uuuu
E2Dh	PRG1FPPS	—	—	—		PRG1FPPS<4:0>				---1 0101	---u uuuu
E2Eh	PRG2RPPS ⁽²⁾	—	—	—		PRG2RPPS<4:0>				---1 0100	---u uuuu
E2Fh	PRG2FPPS ⁽²⁾	—	—	—		PRG2FPPS<4:0>				---1 0101	---u uuuu
E30h	MD1CHPPS	—	—	—		MD1CHPPS<4:0>				---0 0011	---u uuuu
E31h	MD1CLPPS	—	—	—		MD1CLPPS<4:0>				---0 0100	---u uuuu

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

注 1: 未实现位, 读为 1。

2: 仅限 PIC16(L)F1768/9。

3: 仅限 PIC16(L)F1764/5。

4: PIC16LF1764/5/8/9 上未实现。

表 3-16：特殊功能寄存器汇总（续）

地址	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
Bank 28 (续)											
E32h	MD1MODPPS	—	—	—	MD1MODPPS<4:0>				---0 0101	---u uuuu	
E33h	MD2CHPPS ⁽²⁾	—	—	—	MD2CHPPS<4:0>				---0 0011	---u uuuu	
E34h	MD2CLPPS ⁽²⁾	—	—	—	MD2CLPPS<4:0>				---0 0100	---u uuuu	
E35h	MD2MODPPS ⁽²⁾	—	—	—	MD2MODPPS<4:0>				---0 0101	---u uuuu	
E36h 至 E7Fh	—	未实现							—	—	

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

- 注 1: 未实现位，读为 1。
 2: 仅限 PIC16(L)F1768/9。
 3: 仅限 PIC16(L)F1764/5。
 4: PIC16LF1764/5/8/9 上未实现。

PIC16(L)F1764/5/8/9

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
Bank 29											
E8Ch — E8Fh —	未实现	—	—	—	RA0PPS<4:0>	—	—	—	—	—	—
E90h	RA0PPS	—	—	—	RA1PPS<4:0>	—	—	—	—	—	—
E91h	RA1PPS	—	—	—	RA2PPS<4:0>	—	—	—	—	—	—
E92h	RA2PPS	—	—	—	RA4PPS<4:0>	—	—	—	—	—	—
E93h —	未实现	—	—	—	RA5PPS<4:0>	—	—	—	—	—	—
E94h	RA4PPS	—	—	—	RA5PPS<4:0>	—	—	—	—	—	—
E95h	RA5PPS	—	—	—	RB4PPS<4:0>	—	—	—	—	—	—
E96h —	未实现	—	—	—	RB5PPS<4:0>	—	—	—	—	—	—
E97h —	未实现	—	—	—	RB6PPS<4:0>	—	—	—	—	—	—
E98h —	未实现	—	—	—	RB7PPS<4:0>	—	—	—	—	—	—
E99h —	未实现	—	—	—	RC0PPS<4:0>	—	—	—	—	—	—
E9Ah —	未实现	—	—	—	RC1PPS<4:0>	—	—	—	—	—	—
E9Bh —	未实现	—	—	—	RC2PPS<4:0>	—	—	—	—	—	—
E9Ch	RB4PPS ⁽²⁾	—	—	—	RC3PPS<4:0>	—	—	—	—	—	—
E9Dh	RB5PPS ⁽²⁾	—	—	—	RC4PPS<4:0>	—	—	—	—	—	—
E9Eh	RB6PPS ⁽²⁾	—	—	—	RC5PPS<4:0>	—	—	—	—	—	—
E9Fh	RB7PPS ⁽²⁾	—	—	—	RC6PPS<4:0>	—	—	—	—	—	—
EA0h	RC0PPS	—	—	—	RC7PPS<4:0>	—	—	—	—	—	—
EA1h	RC1PPS	—	—	—	—	—	—	—	—	—	—
EA2h	RC2PPS	—	—	—	—	—	—	—	—	—	—
EA3h	RC3PPS	—	—	—	—	—	—	—	—	—	—
EA4h	RC4PPS	—	—	—	—	—	—	—	—	—	—
EA5h	RC5PPS	—	—	—	—	—	—	—	—	—	—
EA6h	RC6PPS ⁽²⁾	—	—	—	—	—	—	—	—	—	—
EA7h	RC7PPS ⁽²⁾	—	—	—	—	—	—	—	—	—	—
EA8h — EEFh —	未实现	—	—	—	—	—	—	—	—	—	—

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
Bank 30											
F0Ch — F0Eh —	未实现	—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT	----000	----000
F0Fh	CLCDATA	—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT	----000	----000
F10h	CLC1CON	EN	—	OUT	INTP	INTN	MODE<2:0>			0-00 0000	0-00 0000
F11h	CLC1POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	0---xxxx	0---uuuu
F12h	CLC1SEL0	—	—	—	D1S<4:0>					---x xxxx	---u uuuu
F13h	CLC1SEL1	—	—	—	D2S<4:0>					---x xxxx	---u uuuu
F14h	CLC1SEL2	—	—	—	D3S<4:0>					---x xxxx	---u uuuu
F15h	CLC1SEL3	—	—	—	D4S<4:0>					---x xxxx	---u uuuu
F16h	CLC1GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	xxxx xxxx	uuuuu uuuuu
F17h	CLC1GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuuu uuuuu
F18h	CLC1GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuuu uuuuu
F19h	CLC1GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuuu uuuuu
F1Ah	CLC2CON	EN	—	OUT	INTP	INTN	MODE<2:0>			0-00 0000	0-00 0000
F1Bh	CLC2POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	0---xxxx	0---uuuu
F1Ch	CLC2SEL0	—	—	—	D1S<4:0>					---x xxxx	---u uuuu
F1Dh	CLC2SEL1	—	—	—	D2S<4:0>					---x xxxx	---u uuuu
F1Eh	CLC2SEL2	—	—	—	D3S<4:0>					---x xxxx	---u uuuu
F1Fh	CLC2SEL3	—	—	—	D4S<4:0>					---x xxxx	---u uuuu
F20h	CLC2GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	xxxx xxxx	uuuuu uuuuu
F21h	CLC2GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuuu uuuuu
F22h	CLC2GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuuu uuuuu
F23h	CLC2GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuuu uuuuu
F24h	CLC3CON	EN	—	OUT	INTP	INTN	MODE<2:0>			0-00 0000	0-00 0000
F25h	CLC3POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	0---xxxx	0---uuuu
F26h	CLC3SEL0	—	—	—	D1S<4:0>					---x xxxx	---u uuuu
F27h	CLC3SEL1	—	—	—	D2S<4:0>					---x xxxx	---u uuuu
F28h	CLC3SEL2	—	—	—	D3S<4:0>					---x xxxx	---u uuuu
F29h	CLC3SEL3	—	—	—	D4S<4:0>					---x xxxx	---u uuuu
F2Ah	CLC3GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	xxxx xxxx	uuuuu uuuuu
F2Bh	CLC3GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuuu uuuuu
F2Ch	CLC3GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuuu uuuuu
F2Dh	CLC3GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuuu uuuuu
F2Eh — F6Fh	—	未实现						—		—	—

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

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

PIC16(L)F1764/5/8/9

表 3-16: 特殊功能寄存器汇总 (续)

地址	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
Bank 31											
F8Ch 至 FE3h	—	未实现								—	—
FE4h	STATUS_SHAD	—	—	—	—	—	Z	DC	C	---- -xxx	---- -uuu
FE5h	WREG_SHAD	工作寄存器的影子寄存器								xxxx xxxx	uuuu uuuu
FE6h	BSR_SHAD	—	—	—	存储区选择寄存器的影子寄存器					---x xxxx	---u uuuu
FE7h	PCLATH_SHAD	—	程序计数器锁存器高字节寄存器的影子寄存器							-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	间接数据存储器地址 0 低字节指针的影子寄存器								xxxx xxxx	uuuu uuuu
FE9h	FSR0H_SHAD	间接数据存储器地址 0 高字节指针的影子寄存器								xxxx xxxx	uuuu uuuu
FEAh	FSR1L_SHAD	间接数据存储器地址 1 低字节指针的影子寄存器								xxxx xxxx	uuuu uuuu
FEBh	FSR1H_SHAD	间接数据存储器地址 1 高字节指针的影子寄存器								xxxx xxxx	uuuu uuuu
FECh	—	未实现								—	—
FEDh	STKPTR	—	—	—	当前堆栈指针					---1 1111	---1 1111
FEEh	TOSL	栈顶低字节								xxxx xxxx	uuuu uuuu
FEFh	TOSH	—	栈顶高字节							-xxx xxxx	-uuu uuuu

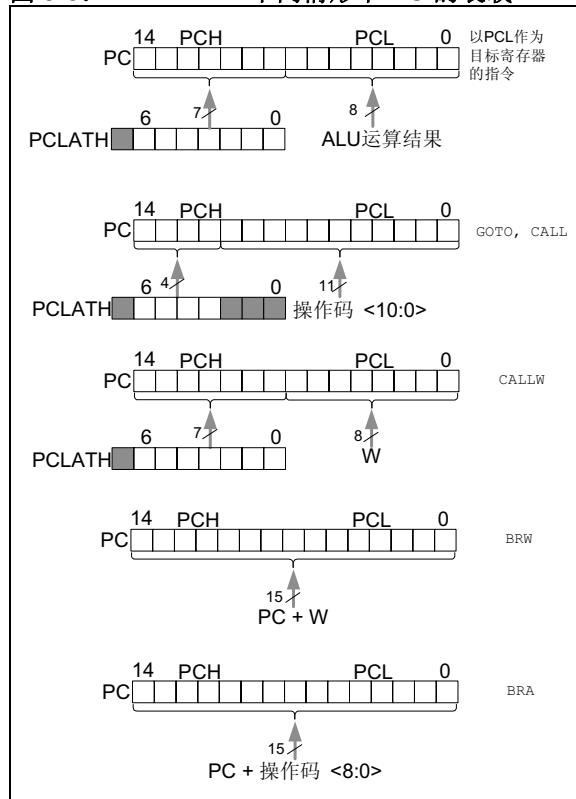
图注: x = 未知, u = 不变, q = 值取决于具体条件, — = 未实现, 读为 0, r = 保留。
阴影单元未实现, 读为 0。

- 注 1: 未实现位, 读为 1。
2: 仅限 PIC16(L)F1768/9。
3: 仅限 PIC16(L)F1764/5。
4: PIC16LF1764/5/8/9 上未实现。

3.5 PCL 和 PCLATH

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

图 3-3: 不同情形下 PC 的装载



3.5.1 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。

3.5.2 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)。

3.5.3 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。

3.5.4 跳转

跳转指令会将一个偏移量与 PC 相加。这使得可以实现可重定位代码和跨越页边界的代码。存在两种跳转形式: BRW 和 BRA。在两种形式中, PC 都会发生递增, 以便取下一条指令。使用任一跳转指令时, 都可以跨越 PCL 存储器边界。

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

如果使用 BRA, 整个 PC 中将装入 PC + 1 + BRA 指令操作数的有符号值。

3.6 堆栈

所有器件都有 16 级 x 15 位宽的硬件堆栈（见图 3-1）。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，或者中断导致程序跳转时，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 位被设定为 0（配置字），堆栈将作为循环缓冲区工作。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。

注：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。

3.6.1 访问堆栈

通过 TOSH、TOSL 和 STKPTR 寄存器可以使用堆栈。STKPTR 是堆栈指针的当前值。TOSH:TOSL 寄存器对指向栈顶。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。在任意时刻，都可以通过检查 STKPTR 来确定所剩余的堆栈空间。STKPTR 总是指向堆栈中当前使用的位置。因此，CALL 或 CALLW 会先递增 STKPTR，然后再写入 PC，而返回操作则会先取出 PC，然后再递减 STKPTR。

关于访问堆栈的示例，请参见图 3-4 至图 3-7。

图 3-4：访问堆栈示例 1

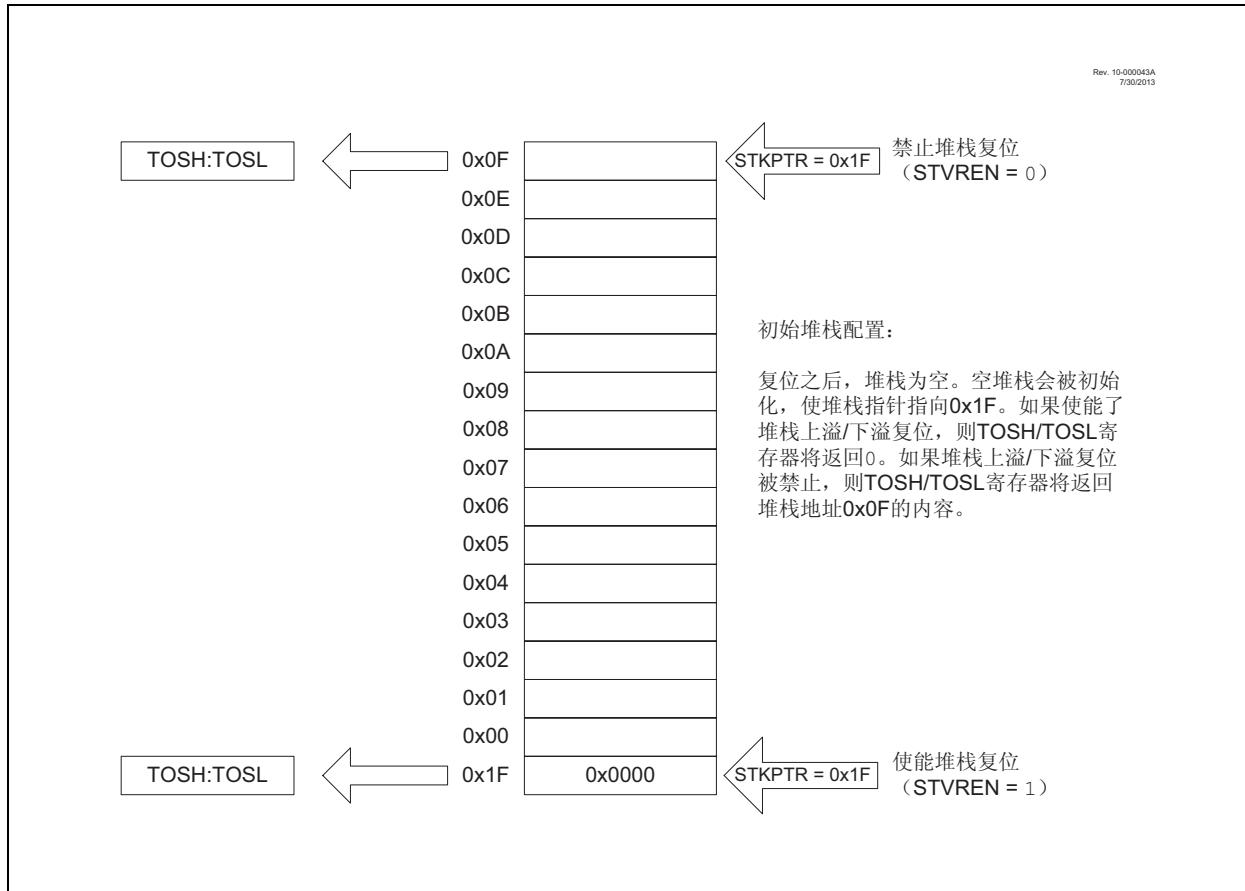


图 3-5: 访问堆栈示例 2

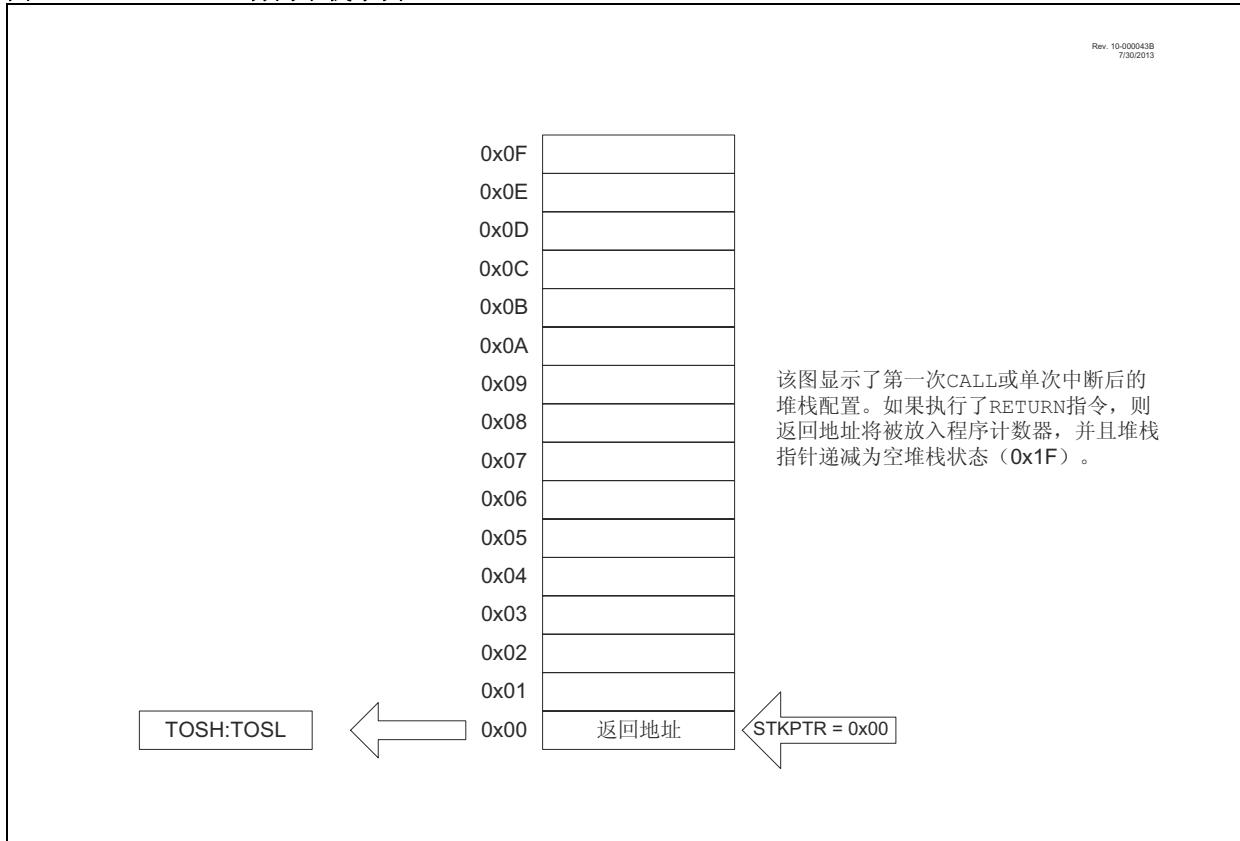


图 3-6: 访问堆栈示例 3

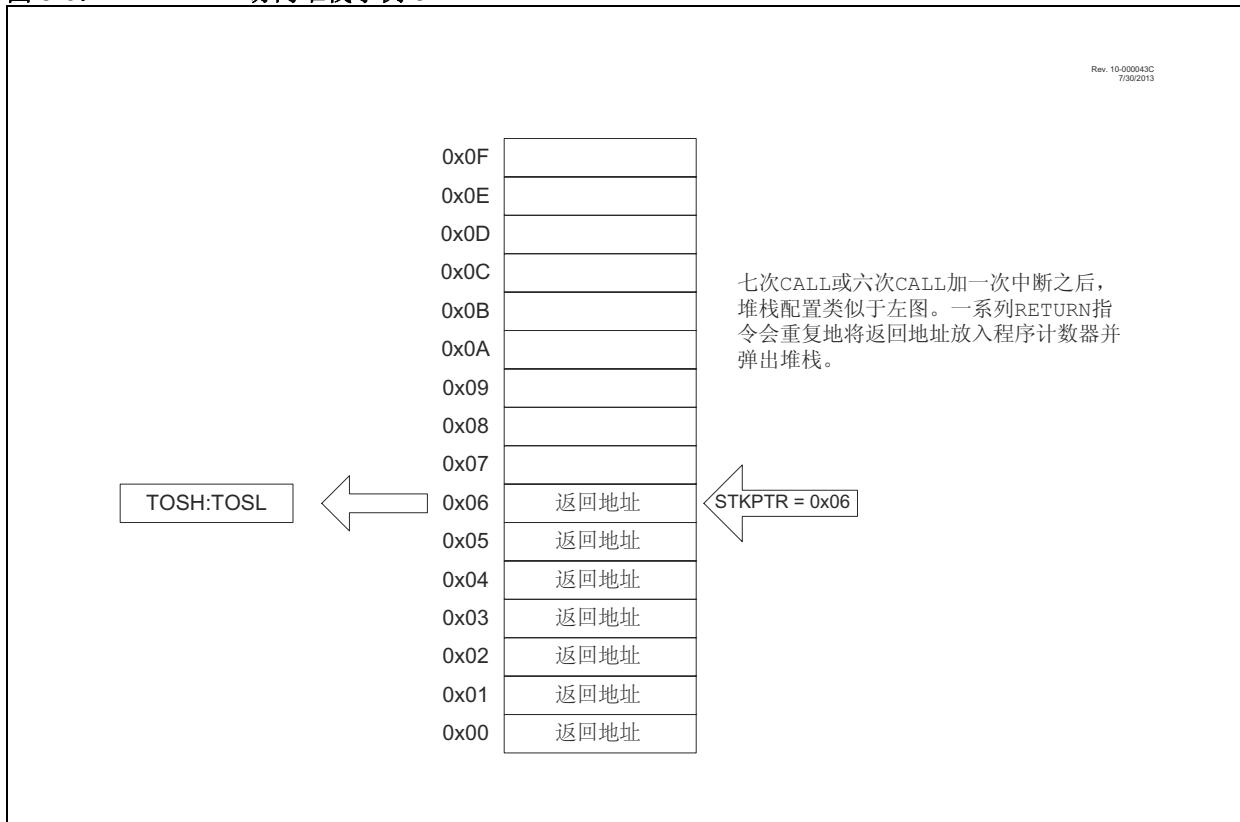
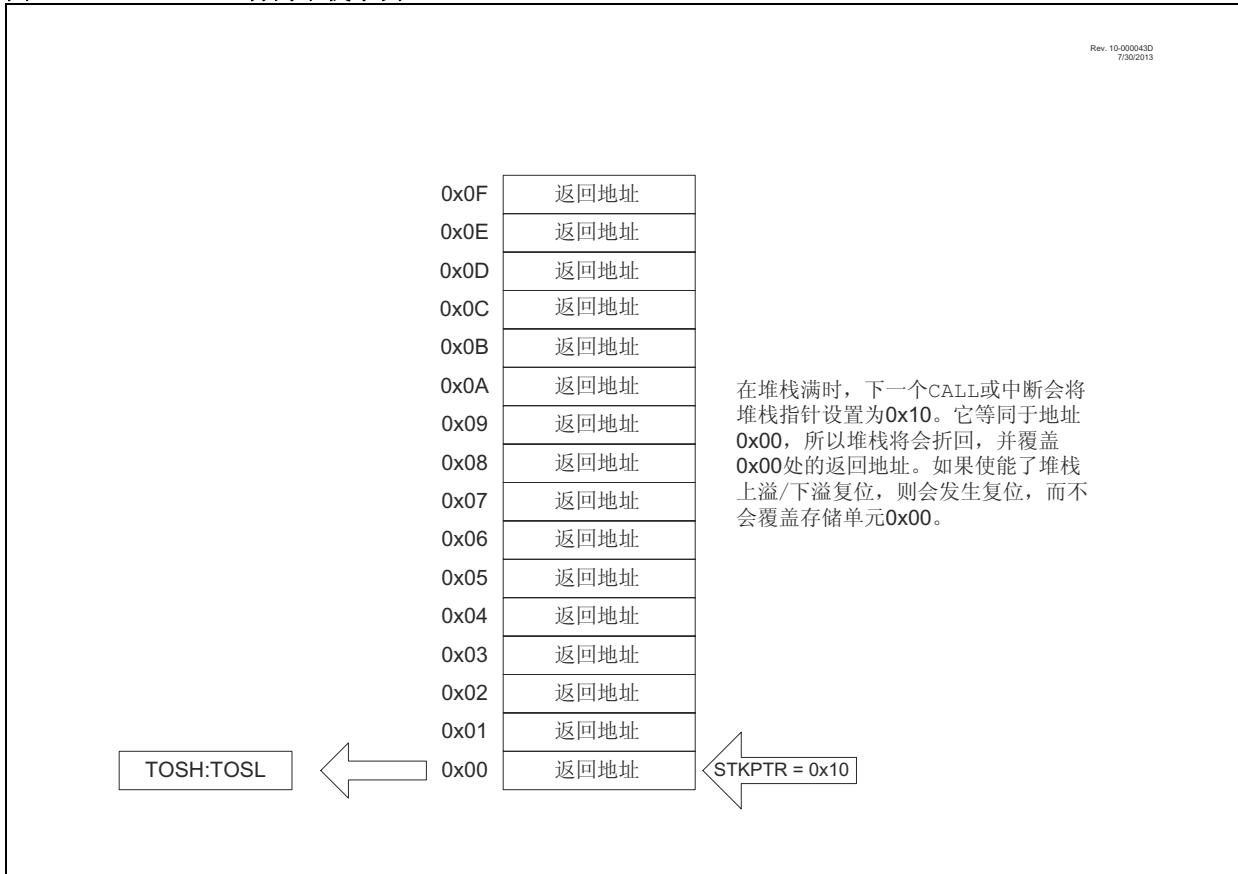


图 3-7：访问堆栈示例 4



3.6.2 上溢 / 下溢复位

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。

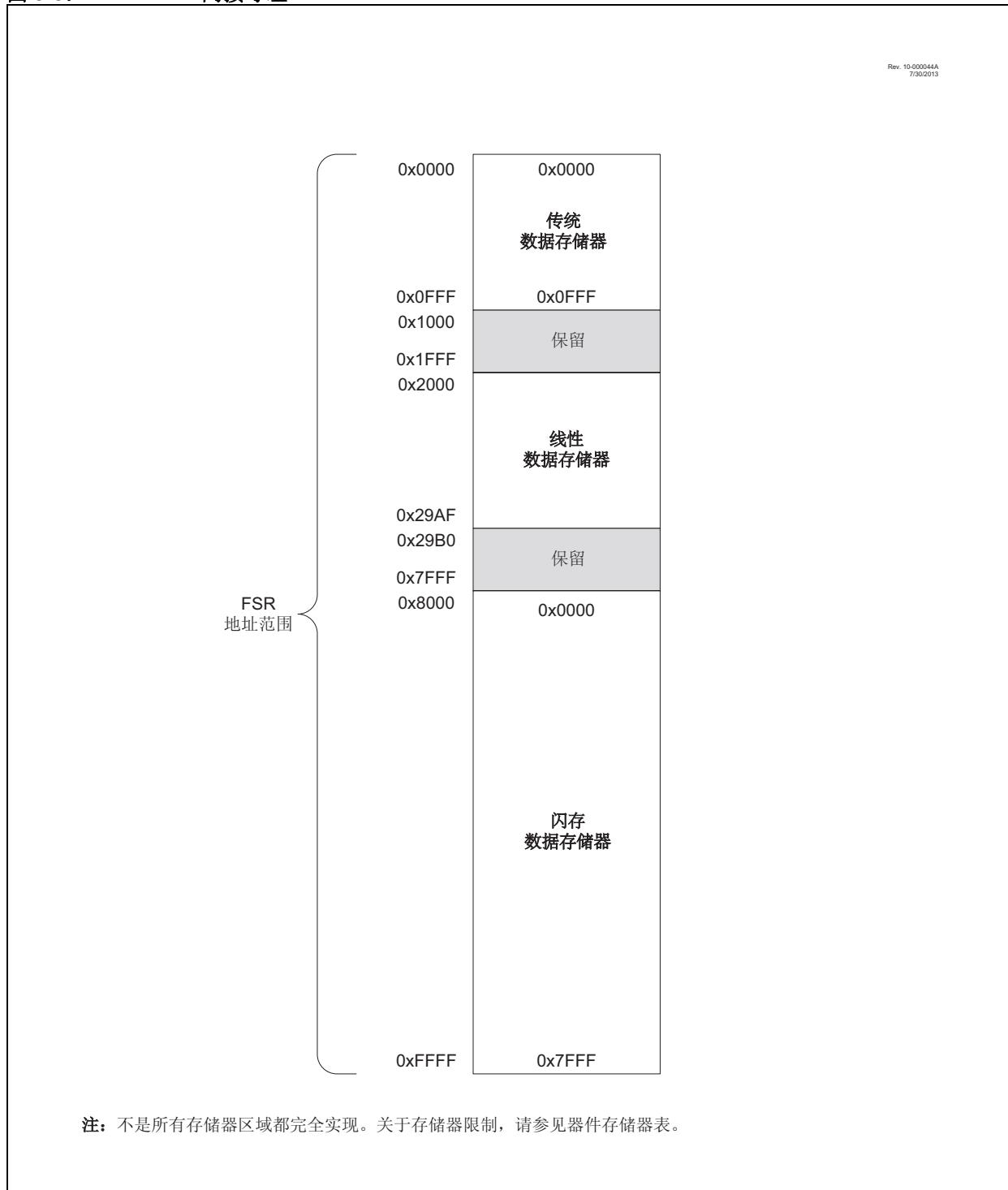
3.7 间接寻址

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 3 个存储器区域：

- 传统数据存储器
- 线性数据存储器
- 闪存程序存储器

图 3-8： 间接寻址

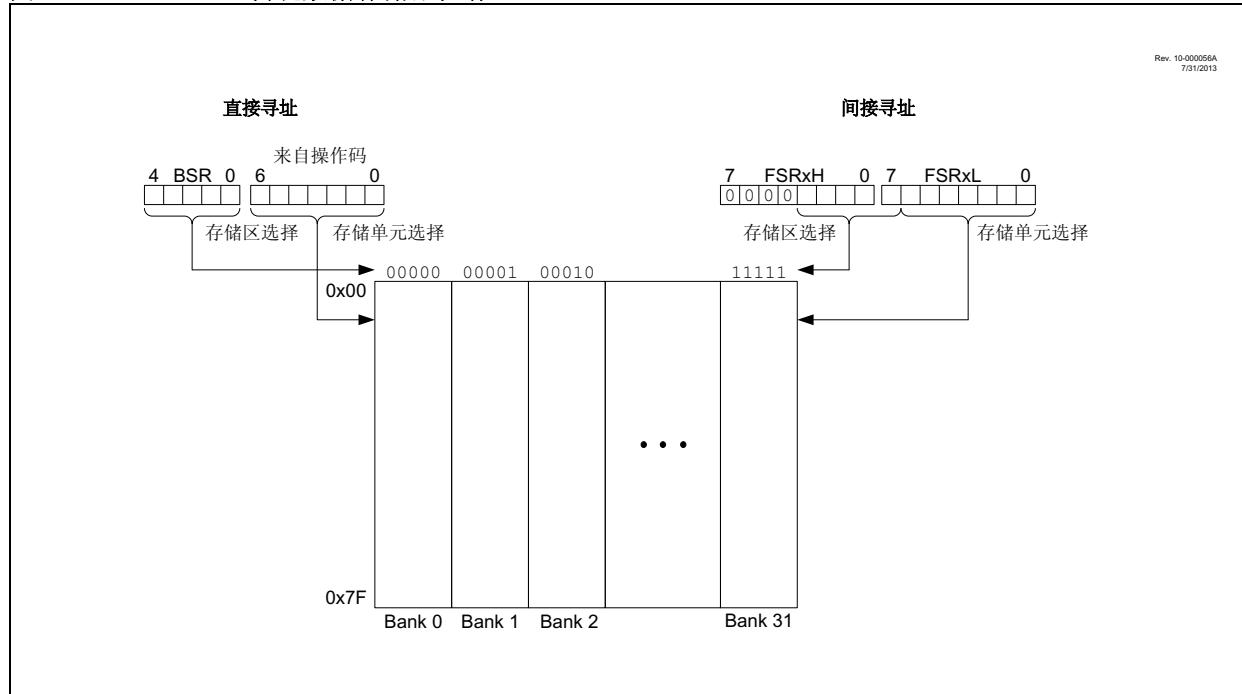


PIC16(L)F1764/5/8/9

3.7.1 传统数据存储器

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

图 3-9： 传统数据存储器映射



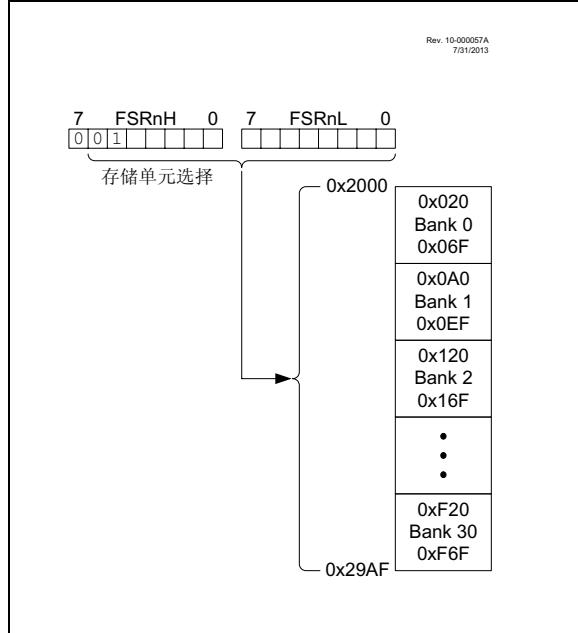
3.7.2 线性数据存储器

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。

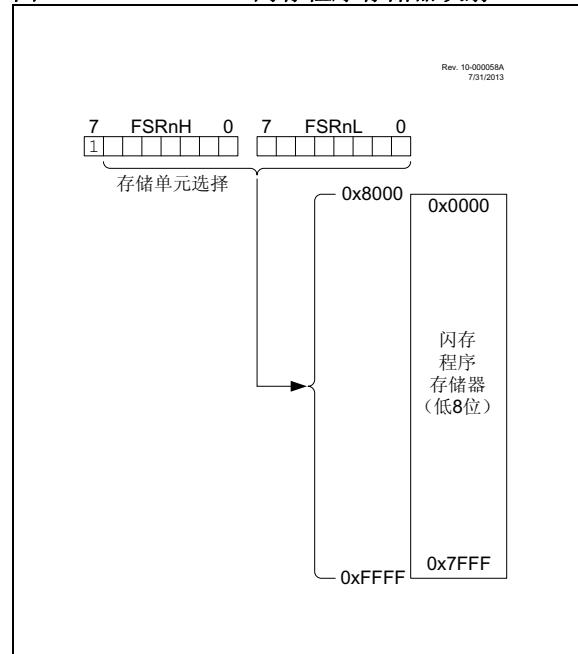
图 3-10: 线性数据存储器映射



3.7.3 闪存程序存储器

为了更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 3-11: 闪存程序存储器映射



4.0 器件配置

器件配置功能由配置字、代码保护和器件 ID 组成。

4.1 配置字

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

注: 配置字中的 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。

PIC16(L)F1764/5/8/9

寄存器 4-1: CONFIG1: 配置字 1 (续)

bit 2-0

FOSC<2:0>: 振荡器选择位

111 = ECH	外部时钟, 高功耗模式: CLKIN 送至 OSC1/CLKIN 引脚
110 = ECM	外部时钟, 中等功耗模式: CLKIN 送至 OSC1/CLKIN 引脚
101 = ECL	外部时钟, 低功耗模式: CLKIN 送至 OSC1/CLKIN 引脚
100 = INTOSC	内部 HFINTOSC。CLKIN 引脚为 I/O 功能
011 = EXTRC	外部 RC 电路连接到 CLKIN 引脚
010 = HS	高速晶振 / 谐振器连接在 OSC1 和 OSC2 引脚之间
001 = XT	晶振 / 谐振器连接在 OSC1 和 OSC2 引脚之间
000 = LP	低功耗晶振连接在 OSC1 和 OSC2 引脚之间

注 1: 在擦除期间关闭代码保护功能时, 将擦除整个闪存程序存储器的内容。执行批量擦除程序存储器命令时, 将擦除整个闪存程序存储器和配置存储器的内容。

寄存器 4-2:

CONFIG2: 配置字 2

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
LVP ⁽¹⁾	DEBUG ⁽²⁾	LPBOR	BORV ⁽³⁾	STVREN	PLLEN
bit 13					bit 8

R/P-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1	R/P-1
ZCD	—	—	—	—	PPS1WAY	WRT<1:0>	
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 0 = OFF 必须使用 MCLR 上的高电压进行编程
bit 12	DEBUG: 在线调试器模式位 ⁽²⁾ 1 = OFF 禁止在线调试器， ICSPCLK 和 ICSPDAT 是通用 I/O 引脚 0 = ON 使能在线调试器， ICSPCLK 和 ICSPDAT 专用于调试器
bit 11	LPBOR: 低功耗 BOR 使能位 1 = OFF 禁止低功耗欠压复位 0 = ON 使能低功耗欠压复位
bit 10	BORV: 欠压复位电压选择位 ⁽³⁾ 1 = LO 选择欠压复位电压 (VBOR) 低跳变点 0 = HI 选择欠压复位电压 (VBOR) 高跳变点
bit 9	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 8	PLLEN: PLL 使能位 1 = ON 使能 4xPLL 0 = OFF 禁止 4xPLL
bit 7	ZCD: ZCD 使能位 1 = OFF 禁止 ZCD。可通过将 ZCDCON 寄存器的 ZCDSEN 位置 1 使能 ZCD 0 = ON 总是使能 ZCD
bit 6-3	未实现：读为 1
bit 2	PPS1WAY: PPSLOCK 位一次置 1 使能位 1 = ON PPSLOCK 位只能在执行解锁序列之后置 1 一次； PPSLOCK 置 1 之后，将阻止对 PPS 寄存器的所有未来更改
	0 = OFF PPSLOCK 位可以根据需要置 1 和清零（前提是已执行解锁序列）
bit 1-0	WRT<1:0>: 闪存自写保护位 4 kW 闪存 (PIC16(L)F1764/8) 11 = OFF 写保护关闭 10 = BOOT 0000h 至 01FFh 被写保护， 0200h 至 0FFFh 可通过 PMCON 控制修改 01 = HALF 0000h 至 07FFh 被写保护， 0800h 至 0FFFh 可通过 PMCON 控制修改 00 = ALL 0000h 至 0FFFh 被写保护， 没有地址可通过 PMCON 控制修改 8 kW 闪存 (PIC16(L)F1765/9) 11 = OFF 写保护关闭 10 = BOOT 0000h 至 01FFh 被写保护， 0200h 至 1FFFh 可通过 PMCON 控制修改 01 = HALF 0000h 至 0FFFh 被写保护， 1000h 至 1FFFh 可通过 PMCON 控制修改 00 = ALL 0000h 至 1FFFh 被写保护， 没有地址可通过 PMCON 控制修改

注 1: 当通过 LVP 进入编程模式时，不能将 LVP 位编程为 0。

2: 配置字中的 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。

3: 关于具体跳变点电压，请参见 **VBOR** 参数。

4.3 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。

4.3.1 程序存储器保护

整个程序存储空间都通过配置字中的 \overline{CP} 位来防止外部读写操作。当 $CP = 0$ 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。对程序存储器的写操作则取决于写保护设置。更多信息，请参见第 4.4 节“写保护”。

4.4 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。

4.5 用户 ID

有 4 个存储单元（8000h-8003h）被指定为 ID 存储单元，供用户存储校验和或其他代码标识号。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 10.4 节“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)F170X Memory Programming Specification*”(DS40001683)。

4.6 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 10.4 节 “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。

4.7 寄存器定义：器件和版本

寄存器 4-3: DEVID: 器件 ID 寄存器

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	DEVID<13:0> 值
PIC16F1764	11 0000 1000 0000 (3080h)
PIC16F1765	11 0000 1000 0001 (3081h)
PIC16F1768	11 0000 1000 0100 (3084h)
PIC16F1769	11 0000 1000 0101 (3085h)
PIC16LF1764	11 0000 1000 0010 (3082h)
PIC16LF1765	11 0000 1000 0011 (3083h)
PIC16LF1768	11 0000 1000 0110 (3086h)
PIC16LF1769	11 0000 1000 0111 (3087h)

PIC16(L)F1764/5/8/9

寄存器 4-4:

REVID: 版本 ID 寄存器

R	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

5.0 振荡器模块（带故障保护时钟监视器）

5.1 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，同时最大限度地发挥性能并降低功耗。[图5-1](#)给出了振荡器模块的框图。

时钟源可由外部振荡器、石英晶体谐振器、陶瓷谐振器以及阻容（Resistor-Capacitor, RC）电路提供。此外，系统时钟源可由两个内部振荡器之一和 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。
- 双速启动模式，最大限度地缩短外部振荡器起振与代码执行之间的延时。
- 故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM），用来检测外部时钟源（LP、XT、HS、ECH、ECM、ECL 或 EXTRC 模式）故障并自动切换到内部振荡器。
- 振荡器起振定时器（OST），可确保晶振源的稳定性。

振荡器模块可配置为以下时钟模式之一。

1. ECL——外部时钟低功耗模式（0 MHz 至 0.5 MHz）
2. ECM——外部时钟中等功耗模式（0.5 MHz 至 4 MHz）
3. ECH——外部时钟高功耗模式（4 MHz 至 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模式（最高 4 MHz）
6. HS——高增益晶振或陶瓷谐振器模式（4 MHz 至 20 MHz）
7. EXTRC——外部阻容
8. INTOSC——内部振荡器（31 kHz 至 32 MHz）

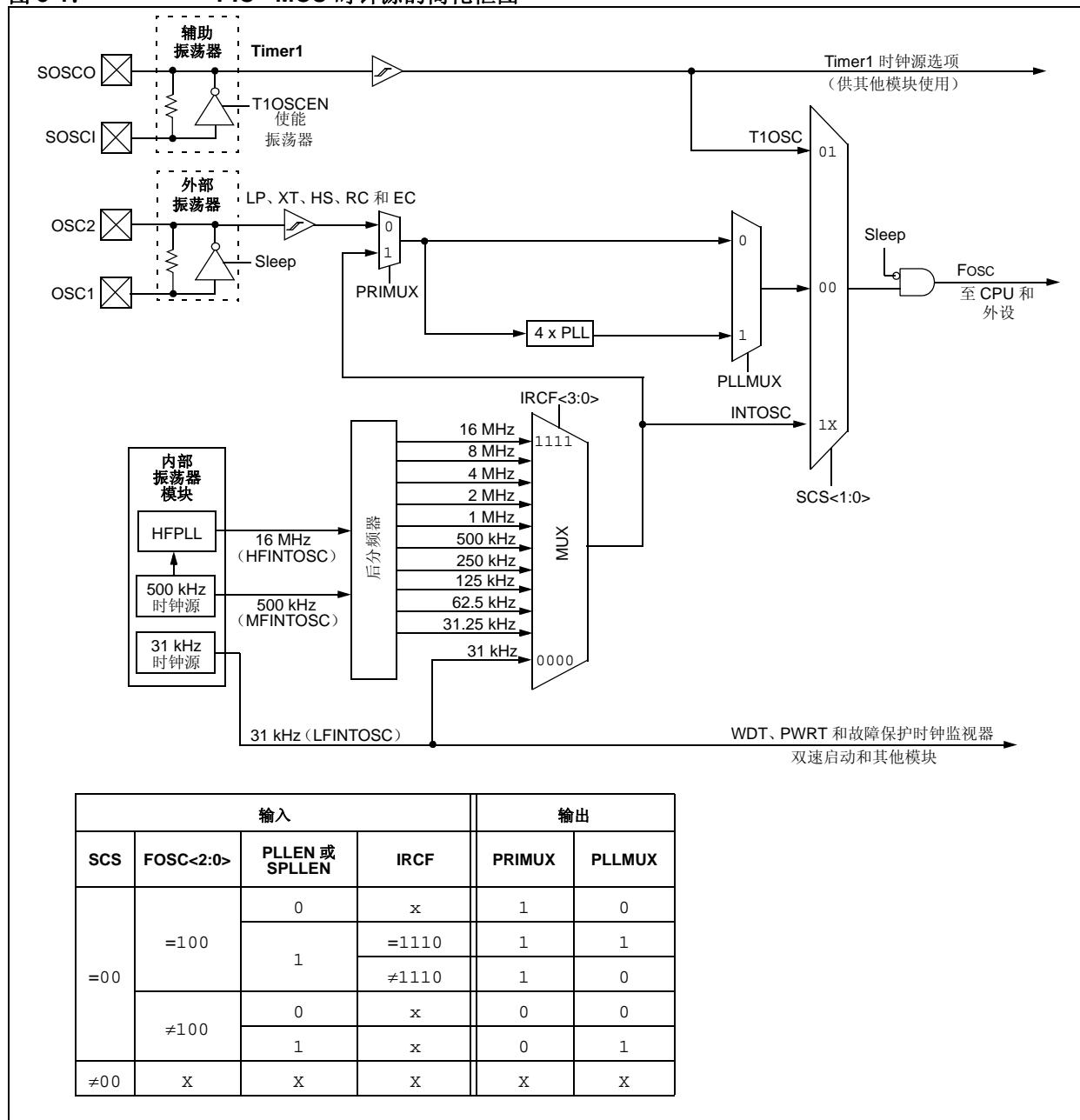
时钟源模式通过配置字中的 **FOSC<2:0>** 位进行选择。**FOSC** 位决定在器件刚上电时使用的振荡器类型。

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。EXTRC 时钟模式需要一个外部电阻和电容来设置振荡器频率。

INTOSC 内部振荡器模块可以产生低频、中频和高频时钟源，分别用 LFINTOSC、MFINTOSC 和 HFINTOSC 表示。（见内部振荡器模块，[图 5-1](#)）。基于这三个时钟源，可以产生多种器件时钟频率选择。

PIC16(L)F1764/5/8/9

图 5-1: PIC® MCU 时钟源的简化框图



5.2 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）、石英晶体谐振器或陶瓷谐振器（LP、XT 和 HS 模式）以及阻容（EXTRC）模式电路。

内部时钟源内置于振荡器模块中。内部振荡器模块具有两个内部振荡器和一个专用锁相环（HFPLL），用于产生 3 个内部系统时钟源：16 MHz 高频内部振荡器（HFINTOSC）、500 kHz（MFINTOSC）和 31 kHz 低频内部振荡器（LFINTOSC）。

通过 OSCCON 寄存器中的系统时钟选择（SCS）位在外部和内部时钟源之间选择系统时钟。更多信息，请参见第 5.3 节“时钟切换”。

5.2.1 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 FOSC<2:0> 位，选择在器件复位时用作默认系统时钟的外部时钟源。
- 写入 OSCCON 寄存器中的 SCS<1:0> 位，将系统时钟源切换为：
 - 辅助振荡器（可在运行时切换），或者
 - 由 FOSC 位的值决定的外部时钟源。

更多信息，请参见第 5.3 节“时钟切换”。

5.2.1.1 EC 模式

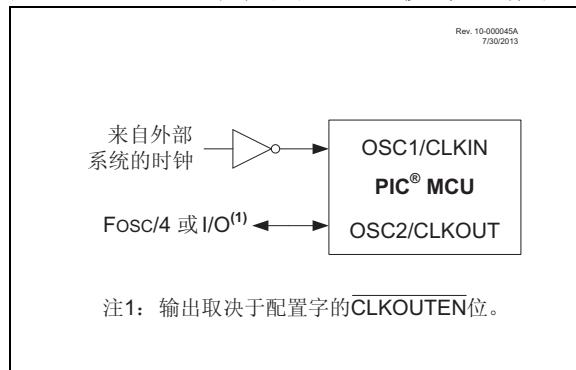
外部时钟（EC）模式允许外部产生的逻辑电平信号作为系统时钟源。工作在该模式下时，外部时钟源连接到 OSC1 输入。OSC2/CLKOUT 可用作通用 I/O 或 CLKOUT。图 5-2 给出了 EC 模式的引脚连接图。

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，4-32 MHz
- ECM——中等功耗，0.5-4 MHz
- ECL——低功耗，0-0.5 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 5-2：外部时钟（EC）模式的工作原理



5.2.1.2 LP、XT 和 HS 模式

LP、XT 和 HS 模式支持使用连接到 OSC1 和 OSC2 之间的石英晶体谐振器或陶瓷谐振器（图 5-3）。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仅工作在 32.768 kHz 的音叉（Tuning Fork）型晶振（时钟晶振）。

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。该模式最适合驱动具备中等驱动电流要求的谐振器。

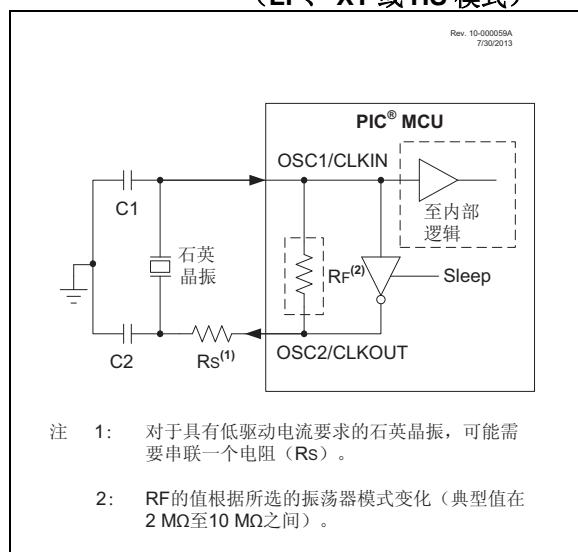
HS 振荡器模式选择内部反相放大器的最高增益设置。HS 模式的电流消耗在三种模式中最大。该模式最适合驱动需要高驱动电流的谐振器。

图 5-3 和图 5-4 分别给出了石英晶体谐振器和陶瓷谐振器的典型电路。

PIC16(L)F1764/5/8/9

图 5-3:

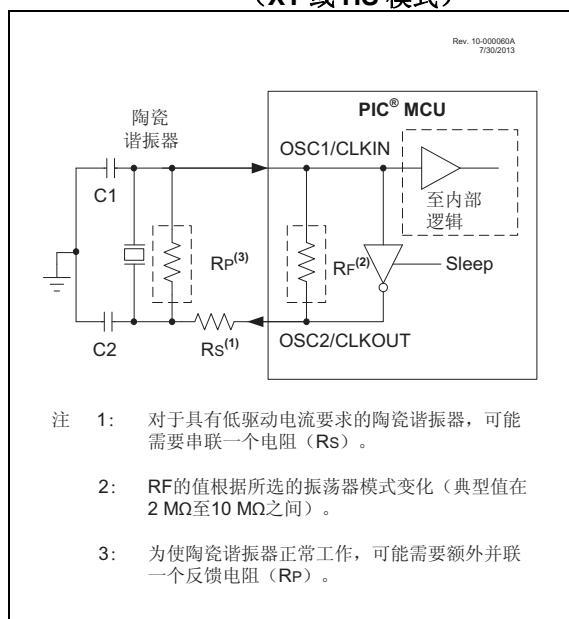
石英晶振的工作原理
(LP、XT 或 HS 模式)



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

图 5-4:

陶瓷谐振器的工作原理
(XT 或 HS 模式)



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

- 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 PIC® Oscillator Design” (DS00849)
 - AN943, “Practical PIC® Oscillator Analysis and Design” (DS00943)
 - AN949, “Making Your Oscillator Work” (DS00949)

5.2.1.3 振荡器起振定时器 (OST)

如果振荡器模块被配置为 LP、XT 或 HS 模式, 则振荡器起振定时器 (OST) 对来自 OSC1 的振荡计数 1024 次。这发生在上电复位 (POR) 和上电延时定时器 (PWRT) 延时结束后 (如果已配置), 或从休眠中唤醒后。在此期间, 程序计数器不递增, 程序执行暂停, 除非使能 FSCM 或双速启动。在这种情况下, 代码将继续以选定 INTOSC 频率执行, 同时 OST 进行计数。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经起振并为振荡器模块提供稳定的系统时钟。

为了使外部振荡器起振和代码执行之间的延时最小, 可选择双速时钟启动模式 (见第 5.4 节 “双速时钟启动模式”)。

5.2.1.4 4x PLL

振荡器模块包含了一个 4x PLL，它可以与外部和内部时钟源配合使用，用于提供系统时钟源。4x PLL 的输入频率必须处于规范值范围内。请参见表 36-9 中的 PLL 时钟时序规范。

4x PLL 可以通过两种方法之一使能：

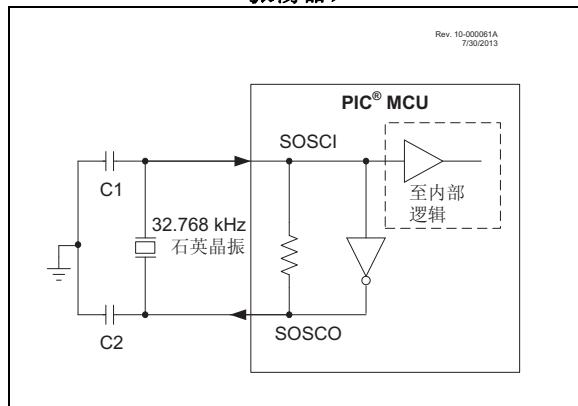
1. 将配置字中的 PLLEN 位编程为 1。
2. 将 OSCCON 寄存器中的 SPLLEN 位写为 1。如果配置字中的 PLLEN 位编程为 1，则 SPLLEN 的值会被忽略。

5.2.1.5 辅助振荡器

辅助振荡器是与 Timer1 外设关联的独立晶振。在 SOSCO 和 SOSCI 器件引脚之间连接一个 32.768 kHz 晶振，对计时操作进行优化。

辅助振荡器可以用作备用系统时钟源，并且可以在运行时通过时钟切换选择。更多信息，请参见第 5.3 节“时钟切换”。

图 5-5: 石英晶振的工作原理（辅助振荡器）



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 PIC® Oscillator Design*” (DS00849)
- AN943, “*Practical PIC® 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)

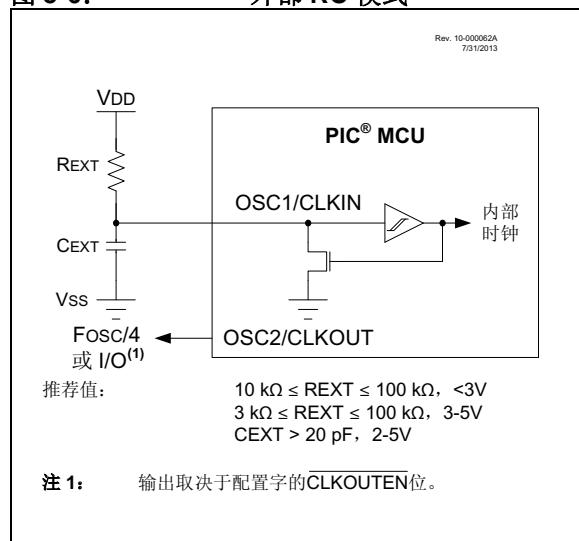
5.2.1.6 外部 RC 模式

外部阻容 (EXTRC) 模式支持使用外部 RC 电路。对时钟精度要求不高时，这使设计人员有了很大的频率选择空间，且保持成本最低。

RC 电路连接到 OSC1。OSC2/CLKOUT 可用作通用 I/O 或 CLKOUT。OSC2/CLKOUT 引脚的功能由配置字中的 CLKOUTEN 位决定。

图 5-6 给出了外部 RC 模式的连接电路。

图 5-6： 外部 RC 模式



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

- 阈值电压变化
- 元件容差
- 不同封装类型的电容差异

用户还应考虑因所使用的外部 RC 元件的容差而导致的差异。

5.2.2 内部时钟源

通过执行以下操作之一，可以将器件配置为使用内部振荡器模块作为系统时钟：

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

在 INTOSC 模式下，OSC1/CLKIN 可用作通用 I/O。OSC2/CLKOUT 可用作通用 I/O 或 CLKOUT。

OSC2/CLKOUT 引脚的功能由配置字中的 CLKOUTEN 位决定。

内部振荡器模块具有两个独立振荡器和一个专用锁相环 HFPLL，可以产生以下 3 个内部系统时钟源中的一个。

1. **HFINTOSC**（高频内部振荡器）出厂时已校准，工作频率为 16 MHz。HFINTOSC 源通过 500 kHz MFINTOSC 源和专用锁相环 HFPLL 产生。使用 OSCTUNE 寄存器（寄存器 5-3），用户可通过软件调整 HFINTOSC 的频率。
2. **MFINTOSC**（中频内部振荡器）出厂时已校准，工作频率为 500 kHz。使用 OSCTUNE 寄存器（寄存器 5-3），用户可通过软件调整 MFINTOSC 的频率。
3. **LFINTOSC**（低频内部振荡器）未经校准，工作频率为 31 kHz。

5.2.2.1 HFINTOSC

高频内部振荡器（HFINTOSC）在出厂时已校准，为16 MHz 内部时钟源。使用 OSCTUNE 寄存器（[寄存器 5-3](#)），可通过软件改变 HFINTOSC 的频率。

HFINTOSC 的输出连接到后分频器和多路开关（见图 5-1）。使用 OSCCON 寄存器的 IRCF<3:0> 位，可通过软件选择基于 HFINTOSC 产生的多个频率中的一个。更多信息，请参见[第 5.2.2.7 节 “内部振荡器时钟切换时序”](#)。

通过以下方式使能 HFINTOSC：

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

快速启动振荡器使内部电路可以在切换至 HFINTOSC 之前上电并稳定下来。

OSCSTAT 寄存器的高频内部振荡器就绪位（HFIOFR）指示 HFINTOSC 何时运行。

OSCSTAT 寄存器的高频内部振荡器状态锁定位（HFIOFL）指示 HFINTOSC 何时在距离其最终值的 2% 范围内运行。

OSCSTAT 寄存器的高频内部振荡器稳定位（HFIOFS）指示 HFINTOSC 何时在距离其最终值的 0.5% 范围内运行。

5.2.2.2 MFINTOSC

中频内部振荡器（MFINTOSC）在出厂时已校准，为500 kHz 内部时钟源。使用 OSCTUNE 寄存器（[寄存器 5-3](#)），可通过软件改变 MFINTOSC 的频率。

MFINTOSC 的输出连接到后分频器和多路开关（见图 5-1）。使用 OSCCON 寄存器的 IRCF<3:0> 位，可通过软件选择基于 MFINTOSC 产生的 9 个频率中的一个。更多信息，请参见[第 5.2.2.7 节 “内部振荡器时钟切换时序”](#)。

通过以下方式使能 MFINTOSC：

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

OSCSTAT 寄存器的中频内部振荡器就绪位（MFIOFR）指示 MFINTOSC 何时运行。

5.2.2.3 内部振荡器频率调整

500 kHz 内部振荡器是出厂时校准的。该内部振荡器可以通过用软件写入 OSCTUNE 寄存器（[寄存器 5-3](#)）进行调整。由于 HFINTOSC 和 MFINTOSC 时钟源是基于 500 kHz 内部振荡器产生的，所以 OSCTUNE 寄存器值的变化将同时应用于两者。

OSCTUNE 寄存器的默认值为 0。该值是一个 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，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。

OSCTUNE 不会影响 LFINTOSC 频率。依赖于 LFINTOSC 时钟源频率的功能，如上电延时定时器（PWRT）、看门狗定时器（WDT）、故障保护时钟监视器（FSCM）以及外设等，其工作不受频率改变的影响。

5.2.2.4 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准的 31 kHz 内部时钟源。

LFINTOSC 的输出连接到多路开关（见图 5-1）。使用 OSCCON 寄存器的 IRCF<3:0> 位，通过软件选择 31 kHz。更多信息，请参见[第 5.2.2.7 节 “内部振荡器时钟切换时序”](#)。LFINTOSC 还是上电延时定时器（PWRT）、看门狗定时器（WDT）以及故障保护时钟监视器（FSCM）的时钟源。

LFINTOSC 可以通过选择 31 kHz（OSCCON 寄存器的 IRCF<3:0> 位 = 000）作为系统时钟源（OSCCON 寄存器的 SCS 位 = 1x）进行使能，也可以通过以下方式使能：

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

使用 LFINTOSC 的外设有：

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

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

5.2.2.5 内部振荡器频率选择

使用 OSCCON 寄存器的内部振荡器频率选择位 $\text{IRCF}\langle 3:0 \rangle$ ，可通过软件选择系统时钟速度。

16 MHz HFINTOSC、500 kHz MFINTOSC 和 31 kHz LFINTOSC 的后分频输出连接到一个多路开关（见图 5-1）。OSCCON 寄存器的内部振荡器频率选择位 $\text{IRCF}\langle 3:0 \rangle$ 用于选择内部振荡器的频率输出。可通过软件选择以下频率中的一个：

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

注：任何复位后，OSCCON 寄存器的 $\text{IRCF}\langle 3:0 \rangle$ 位都被设置为 0111，频率选择被设置为 500 kHz。用户可以修改 IRCF 位来选择其他频率。

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

5.2.2.6 32 MHz 内部振荡器频率选择

内部振荡器模块可以与和外部振荡器模块关联的 4x PLL 配合使用，产生 32 MHz 内部系统时钟源。要使用 32 MHz 内部时钟源，需要以下设置：

- 配置字中的 FOSC 位必须设置为使用 INTOSC 源作为器件系统时钟 ($\text{FOSC}\langle 2:0 \rangle = 100$)。
- OSCCON 寄存器中的 SCS 位必须清零 ($\text{SCS}\langle 1:0 \rangle = 00$)，以使用由配置字中 $\text{FOSC}\langle 2:0 \rangle$ 决定的时钟。
- OSCCON 寄存器中的 IRCF 位必须设置为使用 8 MHz HFINTOSC ($\text{IRCF}\langle 3:0 \rangle = 1110$)。
- OSCCON 寄存器中的 SPLLEN 位必须置 1，以使能 4x PLL，或者配置字的 PLLLEN 位必须编程为 1。

注：使用配置字的 PLLLEN 位时，将无法用软件禁止 4x PLL，并且 SPLLEN 选项将不可用。

当 OSCCON 寄存器的 SCS 位设置为 1x 时，4x PLL 将不能用于内部振荡器。要将 4x PLL 用于内部振荡器，SCS 位必须设置为 00。

5.2.2.7 内部振荡器时钟切换时序

当在 HFINTOSC、MFINTOSC 和 LFINTOSC 之间切换时，新振荡器可能已经关闭以节省功耗（见图 5-7）。如果是这种情况，则在修改 OSCCON 寄存器的 IRCF<3:0> 位之后，会经过一定的延时，新选择的频率才会生效。OSCSTAT 寄存器将反映 HFINTOSC、MFINTOSC 和 LFINTOSC 振荡器的当前工作状态。频率选择序列如下：

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

更多详细信息，请参见图 5-7。

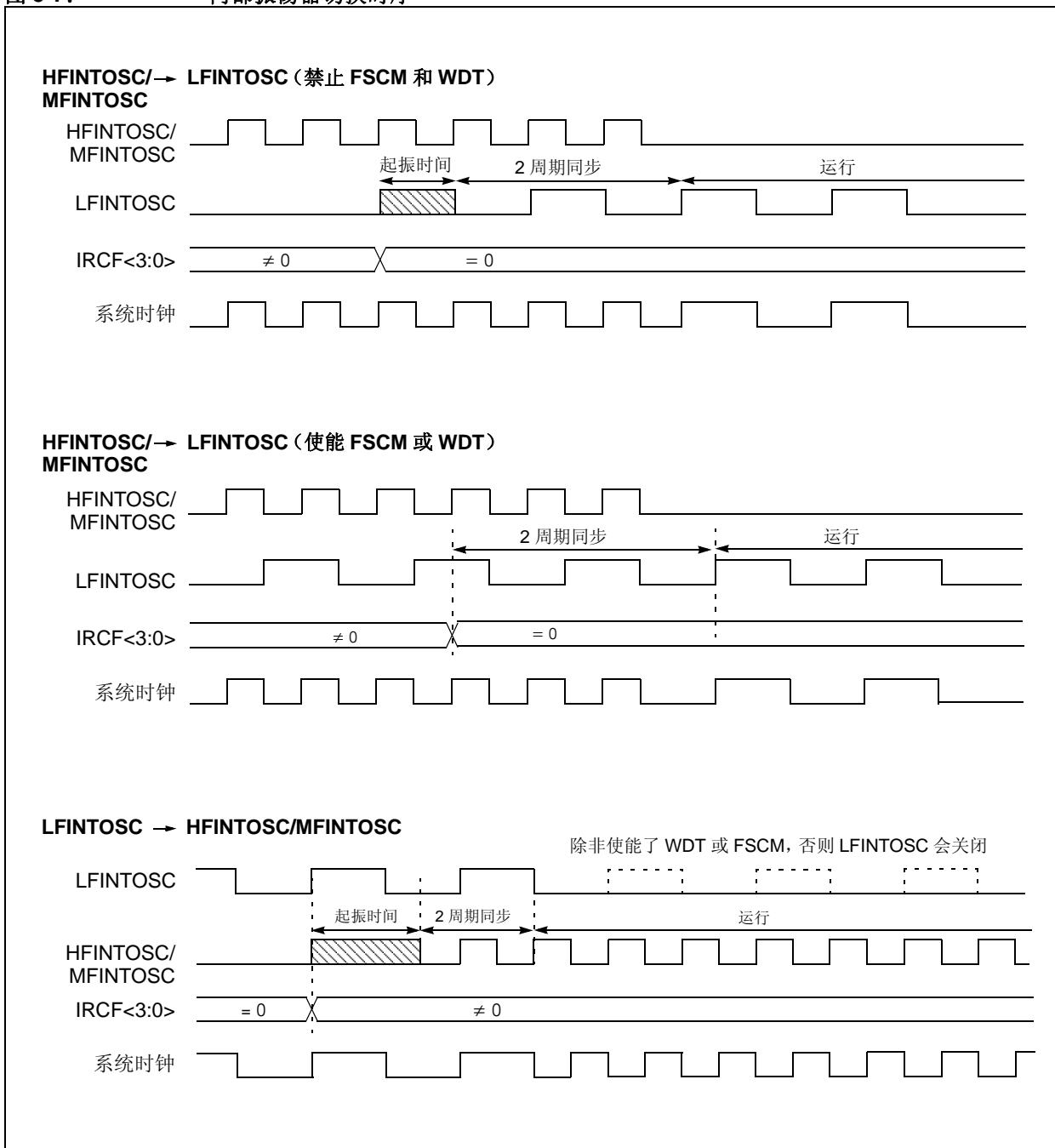
如果在同一个内部时钟源的两个时钟之间进行切换（改变内部振荡器速度），则选择新频率不存在起振延时。

表 5-1 中列出了时钟切换延时。

起振延时规范请参见第 36.0 节 “电气规范” 中的振荡器表。

PIC16(L)F1764/5/8/9

图 5-7： 内部振荡器切换时序



5.3 时钟切换

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

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

5.3.1 系统时钟选择（SCS）位

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

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

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

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

5.3.2 振荡器起振定时器状态（OSTS）位

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

5.3.3 辅助振荡器

辅助振荡器是与 Timer1 外设关联的独立晶振。在 SOSCO 和 SOSCI 器件引脚之间连接一个 32.768 kHz 晶振，对计时操作进行优化。

通过 T1CON 寄存器中的 OSCEN 控制位来使能辅助振荡器。关于 Timer1 外设的更多信息，请参见第 22.0 节“带门控的 Timer1/3/5 模块”。

5.3.4 辅助振荡器就绪（SOSCR）位

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

5.4 双速时钟启动模式

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

当振荡器模块被配置为 LP、XT 或 HS 模式时，双速启动可以带来一些益处。对于这些模式，振荡器起振定时器（OST）会被使能，并且它必须在计数 1024 次振荡之后，振荡器才能用作系统时钟源。

如果振荡器模块被配置为除 LP、XT 或 HS 模式以外的任何模式，则双速启动将被禁止。这是因为 POR 后或从休眠中退出时，外部时钟振荡器不需要花时间稳定。

如果在器件进入休眠模式之前 OST 计数到 1024，则 OSCSTAT 寄存器的 OSTS 位置 1，并且程序执行切换到外部振荡器。但是，如果唤醒所需的时间极短，则系统可能永远不会使用外部振荡器工作。

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

5.4.1 双速启动模式配置

通过以下设定来配置双速启动模式：

- IESO（在配置字中）= 1；内部 / 外部时钟源切换位（使能双速启动模式）。
- SCS（在 OSCCON 寄存器中）= 00。
- 配置字中的 FOSC<2:0> 位被配置为 LP、XT 或 HS 模式。
- 发生以下事件之后，进入双速启动模式：
 - 上电复位（POR）以及在上电延时定时器（PWRT）延时结束（如果使能）后，或者
 - 从休眠中唤醒。

表 5-1：振荡器切换延时

切换自	切换到	频率	振荡器延时
休眠 /POR	LFINTOSC ⁽¹⁾ MFINTOSC ⁽¹⁾ HFINTOSC ⁽¹⁾	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	振荡器预热延时（TWARM）
休眠 /POR	EC 或 RC ⁽¹⁾	DC – 32 MHz	2 个周期
LFINTOSC	EC 或 RC ⁽¹⁾	DC – 32 MHz	每次一周期
休眠 /POR	辅助振荡器 LP、XT 或 HS ⁽¹⁾	32 kHz-20 MHz	1024 个时钟周期（OST）
任何时钟源	MFINTOSC ⁽¹⁾ HFINTOSC ⁽¹⁾	31.25 kHz-500 kHz 31.25 kHz-16 MHz	2 μs（近似值）
任何时钟源	LFINTOSC ⁽¹⁾	31 kHz	每次一周期
任何时钟源	辅助振荡器	32 kHz	1024 个时钟周期（OST）
PLL 无效	PLL 有效	16-32 MHz	2 ms（近似值）

注 1： PLL 无效。

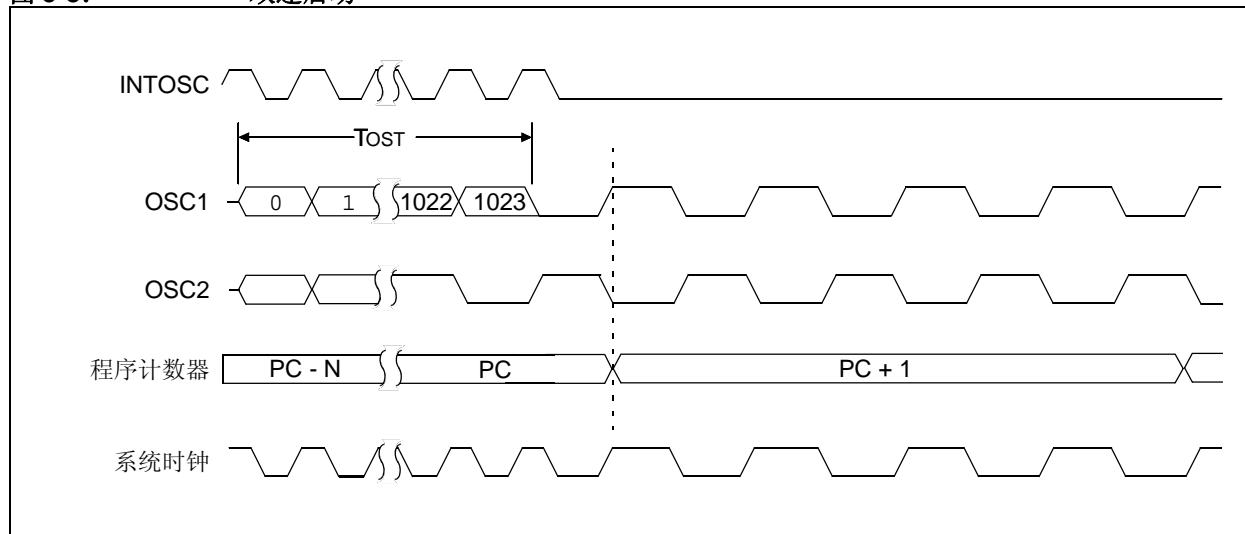
5.4.2 双速启动顺序

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

5.4.3 检查双速时钟状态

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

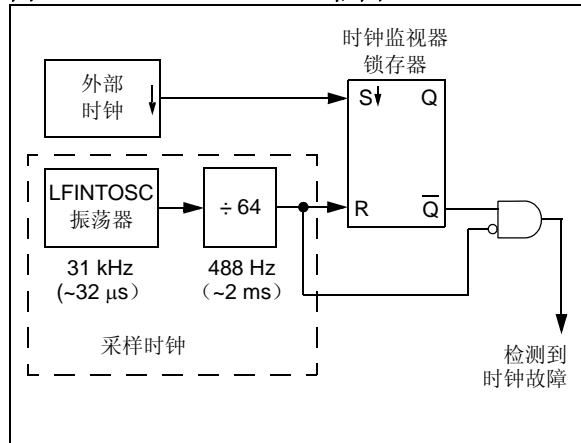
图 5-8： 双速启动



5.5 故障保护时钟监视器

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

图 5-9： FSCM 框图



5.5.1 故障保护检测

FSCM 模块通过将外部振荡器与 FSCM 采样时钟比较来检测振荡器故障。将 LFINTOSC 64 分频，就产生了采样时钟。请参见图 5-9。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。

5.5.2 故障保护操作

当外部时钟出现故障时，FSCM 将器件时钟切换到内部时钟源，并将 PIR2 寄存器的 OSFIF 标志位置 1。如果 PIE2 寄存器的 OSFIE 位也置 1，则 OSFIF 标志置 1 会产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续由内部时钟源提供，直到器件固件成功重启外部振荡器并切换回外部振荡器进行工作。

FSCM 选定的内部时钟源由 OSCCON 寄存器的 IRCF<3:0> 位决定。这使得可以在故障发生前配置内部振荡器。

5.5.3 故障保护条件清除

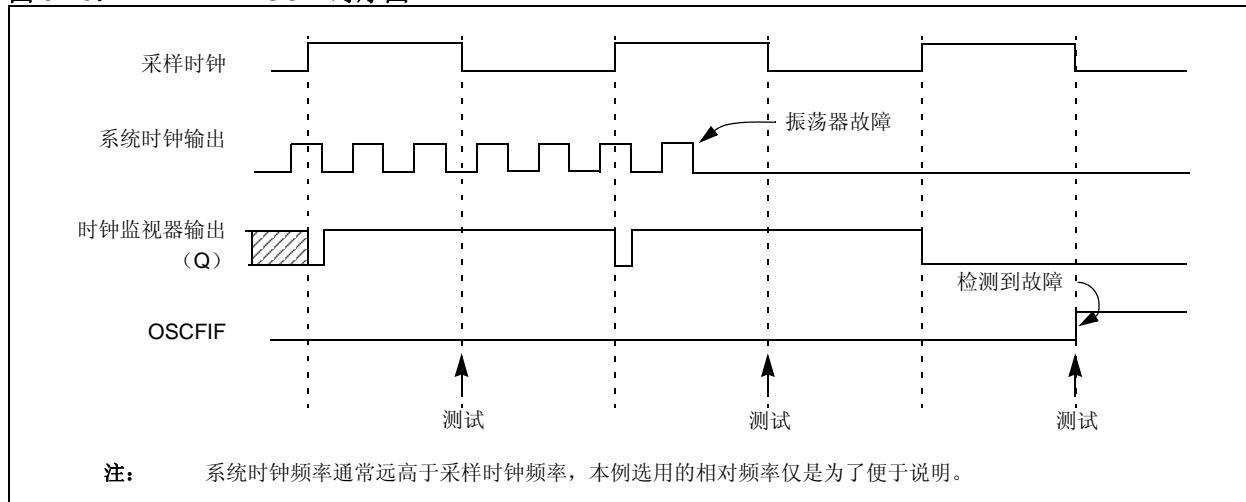
在复位、执行 SLEEP 指令或更改 OSCCON 寄存器的 SCS 位之后，故障保护条件被清除。SCS 位被更改后，OST 将重新启动。OST 运行时，器件将依靠 OSCCON 中选定的 INTOSC 工作。OST 超时后，故障保护条件会在成功切换到外部时钟源之后被清除。在切换到外部时钟源之前，应先清零 OSFIF 位。如果故障保护条件仍然存在，硬件会再次将 OSFIF 标志置 1。

5.5.4 复位或从休眠中唤醒

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

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

图 5-10: FSCM 时序图



PIC16(L)F1764/5/8/9

5.6 寄存器定义：振荡器控制

寄存器 5-1: OSCCON: 振荡器控制寄存器

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN		IRCF<3:0>		—		SCS<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

SPLLEN: 软件 PLL 使能位

如果配置字中的 PLLLEN = 1:

SPLLEN 位被忽略。总是使能 4x PLL (受振荡器要求制约)

如果配置字中的 PLLLEN = 0:

1 = 使能 4x PLL

0 = 禁止 4x PLL

bit 6-3

IRCF<3:0>: 内部振荡器频率选择位

1111 = 16 MHz HF

1110 = 8 MHz 或 32 MHz HF⁽²⁾

1101 = 4 MHz HF

1100 = 2 MHz HF

1011 = 1 MHz HF

1010 = 500 kHz HF⁽¹⁾

1001 = 250 kHz HF⁽¹⁾

1000 = 125 kHz HF⁽¹⁾

0111 = 500 kHz MF (复位时的默认值)

0110 = 250 kHz MF

0101 = 125 kHz MF

0100 = 62.5 kHz MF

0011 = 31.25 kHz HF⁽¹⁾

0010 = 31.25 kHz MF

000x = 31 kHz LF

bit 2

未实现: 读为 0

bit 1-0

SCS<1:0>: 系统时钟选择位

1x = 内部振荡器模块

01 = 辅助振荡器

00 = 由配置字中 FOSC<2:0> 决定的时钟

注 1: 基于 HFINTOSC 产生的重复频率。

2: 在 SPLLEN 位置 1 时为 32 MHz。请参见第 5.2.2.6 节 “32 MHz 内部振荡器频率选择”。

寄存器 5-2: OSCSTAT: 振荡器状态寄存器

R-1/q	R-0/q	R-q/q	R-0/q	R-0/q	R-q/q	R-0/0	R-0/q
SOSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
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 = 条件值

bit 7 **SOSCR:** 辅助振荡器就绪位

如果 T1OSCEN = 1:

1 = 辅助振荡器就绪
0 = 辅助振荡器未就绪

如果 T1OSCEN = 0:

1 = 辅助时钟源始终就绪

bit 6 **PLLR:** 4x PLL 就绪位

1 = 4x PLL 就绪
0 = 4x PLL 未就绪

bit 5 **OSTS:** 振荡器起振定时器状态位

1 = 依靠由配置字中 FOSC<2:0> 位定义的时钟运行
0 = 依靠内部振荡器 (FOSC<2:0> = 100) 运行

bit 4 **HFIOFR:** 高频内部振荡器就绪位

1 = HFINTOSC 就绪
0 = HFINTOSC 未就绪

bit 3 **HFIOFL:** 高频内部振荡器锁定位

1 = HFINTOSC 的精度小于等于 2%
0 = HFINTOSC 的精度大于 2%

bit 2 **MFIOFR:** 中频内部振荡器就绪位

1 = MFINTOSC 就绪
0 = MFINTOSC 未就绪

bit 1 **LFIOFR:** 低频内部振荡器就绪位

1 = LFINTOSC 就绪
0 = LFINTOSC 未就绪

bit 0 **HFIOFS:** 高频内部振荡器稳定位

1 = HFINTOSC 的精度小于等于 0.5%
0 = HFINTOSC 的精度大于 0.5%

PIC16(L)F1764/5/8/9

寄存器 5-3: OSCTUNE: 振荡器调节寄存器

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
—	—			TUN<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 TUN<5:0>: 频率调节位

100000 = 最低频率

•

•

•

111111 =

000000 = 振荡器模块以出厂时校准的频率运行

000001 =

•

•

•

011110 =

011111 = 最高频率

表 5-2: 与时钟源相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		86
OSCSTAT	SOSCR	PLLRL	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	87
OSCTUNE	—	—	TUN<5:0>						88
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	RPC4R ⁽¹⁾	C2IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
T1CON	CS<1:0>		CKPS<1:0>		OSCEN	SYNC	—	ON	228

图注: — = 未实现位, 读为 0。时钟源不使用阴影单元。

注 1: 仅限 PIC16(L)F1768/9。

表 5-3: 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	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>			FOSC<2:0>		

图注: — = 未实现位, 读为 0。时钟源不使用阴影单元。

6.0 复位

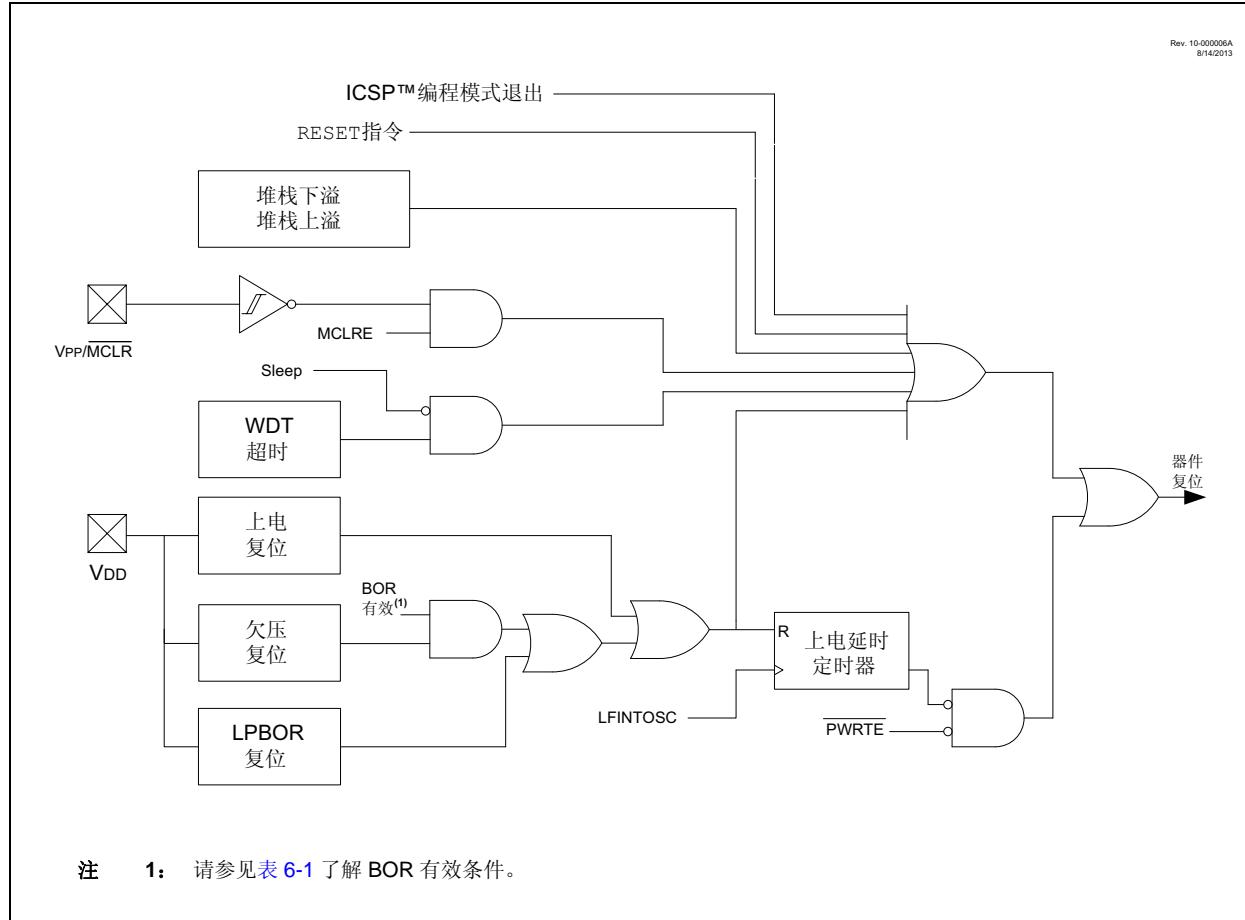
这些器件的复位有几种方式:

- 上电复位 (POR)
- 欠压复位 (BOR)
- 低功耗欠压复位 (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。

6.1.1 上电延时定时器 (PWRT)

上电延时定时器在 POR 或欠压复位时提供一段 64 ms 标称值的延时。

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

上电延时定时器会在 POR 和 BOR 释放之后启动。

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

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	等待 BOR 就绪 ⁽¹⁾ (BORRDY = 1)
10	X	唤醒	有效	等待 BOR 就绪 (BORRDY = 1)
		休眠	禁止	
01	1	X	有效	等待 BOR 就绪 ⁽¹⁾ (BORRDY = 1)
	0	X	禁止	
00	X	X	禁止	立即开始 (BORRDY = x)

注 1: 在“POR 释放”和“从休眠模式唤醒”的特殊情况下，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。

6.2.1 BOR 总是使能

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

BOR 保护在休眠期间有效。BOR 不会延迟从休眠中唤醒。

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 将开启。器件启动会被延迟，直到 BOR 就绪，且 VDD 高于 BOR 阈值为止。

BOR 保护在休眠期间无效。器件唤醒会被延迟，直到 BOR 就绪为止。

6.2 欠压复位 (BOR)

当 VDD 达到可选的最低电压时，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 降至低于 VBOR 的时间大于参数 TBORDC，器件将会发生复位。更多信息，请参见图 6-2。

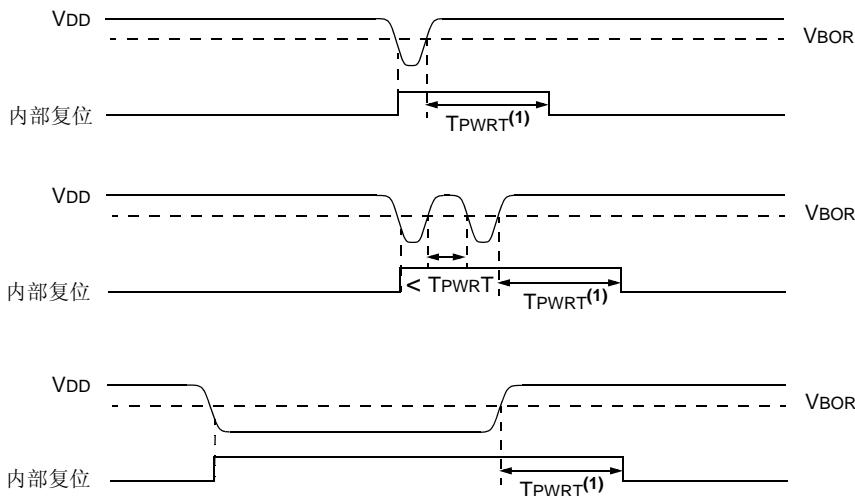
6.2.3 通过软件对 BOR 进行控制

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。器件启动不会受 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



注 1: 仅在 PWRTE 位被编程为 0 时, 才应用 TPWRT 延时。

6.3 寄存器定义: BOR 控制

寄存器 6-1: BORCON: 欠压复位控制寄存器

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS ⁽¹⁾	—	—	—	—	—	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

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

BORFS: 欠压复位快速启动位⁽¹⁾

如果 BOREN<1:0> = 11 (总是开启) 或 BOREN<1:0> = 00 (总是关闭):

BORFS 可读 / 写, 但不起任何作用。

如果 BOREN<1:0> = 10 (在休眠模式下禁止) 或 BOREN<1:0> = 01 (受软件控制):

1 = 总是强制开启带隙 (包括休眠 / 唤醒 / 工作等情形)

0 = 带隙正常工作, 并且可以关闭

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> 位位于配置字中。

6.4 低功耗欠压复位 (LPBOR)

低功耗欠压复位 (LPBOR) 是复位子系统的一个重要组成部分。关于 BOR 如何与其他模块进行交互的信息，请参见图 6-1。

LPBOR 用于监视外部 VDD 引脚。当检测到电压太低时，器件将保持在复位状态。发生这种情况时，有一个寄存器位 (BOR) 会发生改变，指示发生了 BOR 复位。BOR 和 LPBOR 事件都会将该位置 1。请参见寄存器 6-2。

6.4.1 使能 LPBOR

LPBOR 由配置字的 LPBOR 位控制。在器件被擦除后，LPBOR 模块默认设为禁止。

6.4.1.1 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运算，用以提供通用 BOR 信号，并送至 PCON 寄存器和电源控制模块。

6.5 MCLR

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

表 6-2: MCLR 配置

MCLRE	LVP	MCLR
0	0	禁止
1	0	使能
x	1	使能

6.5.1 MCLR 使能

当使能 MCLR 并且引脚保持低电平时，器件会保持在复位状态。MCLR 引脚通过内部弱上拉与 VDD 连接。

器件在 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 MCLR 引脚驱动为低电平。

6.5.2 MCLR 禁止

当 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 11.1 节 “PORTA 寄存器”。

6.6 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。STATUS 寄存器中的 TO 和 PD 位会改变，指示发生了 WDT 复位。更多信息，请参见第 9.0 节 “看门狗定时器 (WDT)”。

6.7 RESET 指令

RESET 指令会引起器件复位。PCON 寄存器中的 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。

6.8 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。这些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 3.6.2 节 “上溢 / 下溢复位”。

6.9 编程模式退出

在退出编程模式时，器件的行为与刚刚发生 POR 时的情况相同。

6.10 上电延时定时器

上电延时定时器可用于在 BOR 或 POR 事件之后延迟器件执行。该定时器通常用于使 VDD 在允许器件开始运行之前先稳定下来。

上电延时定时器由配置字的 PWRTE 位控制。

6.11 启动序列

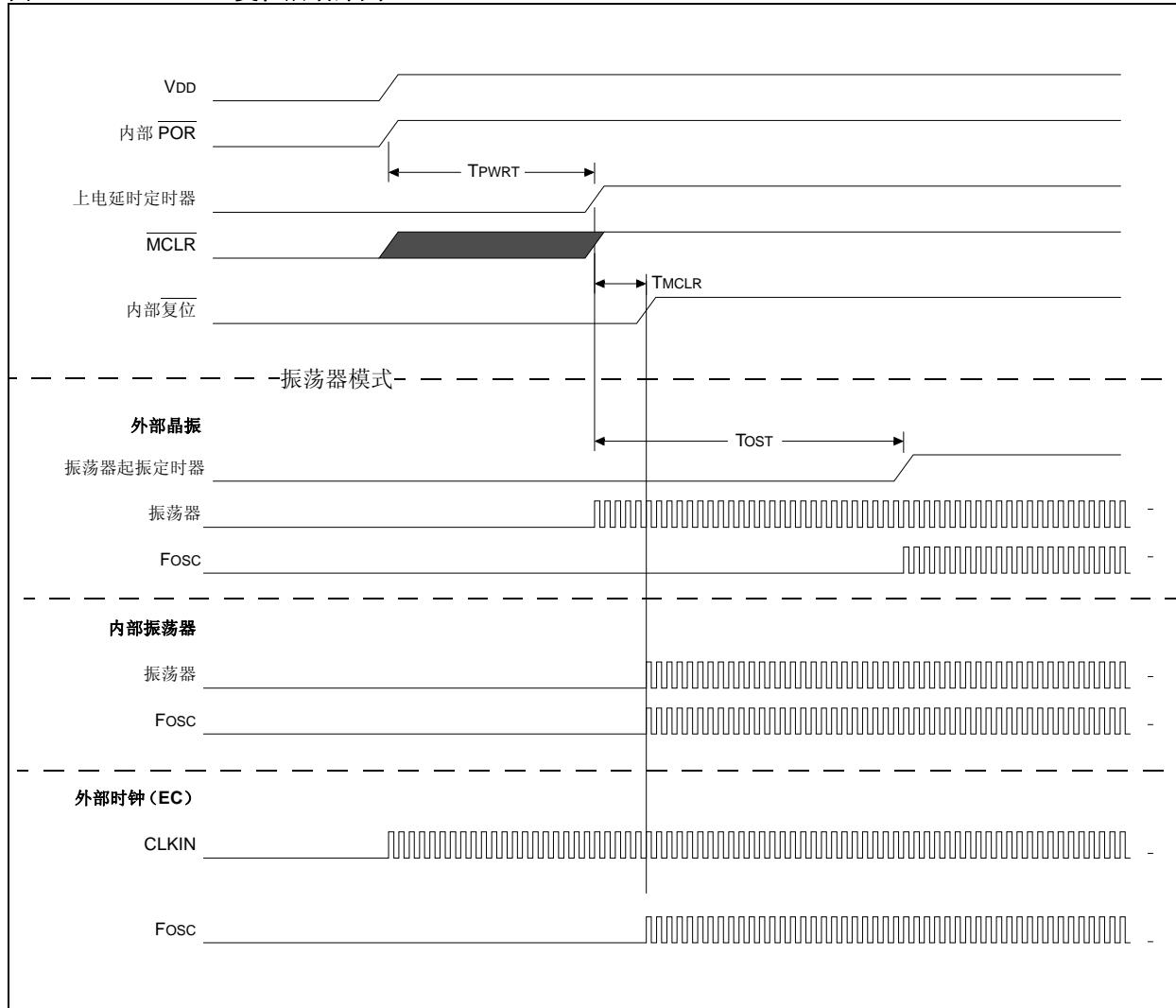
在 POR 或 BOR 释放时，只有先发生以下事件，器件才会开始执行：

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

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 5.0 节 “振荡器模块（带故障保护时钟监视器）”。

上电延时定时器和振荡器起振定时器的运行与 MCLR 复位无关。如果 MCLR 保持低电平的时间足够长，上电延时定时器和振荡器起振定时器将会延时结束。当 MCLR 变为高电平时，器件将在 10 个 Fosc 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



6.12 确定复位原因

在发生任何复位时，STATUS 和 PCON 寄存器中会有很多位发生更新，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，POR 时 TO 被置 1
0	0	1	1	1	0	x	x	0	非法的，POR 时 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 MCLR 复位
u	u	u	0	u	u	u	1	0	休眠期间的 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	PCON 寄存器
上电复位	0000h	---1 1000	00-- 110x
正常工作期间的 MCLR 复位	0000h	---u uuuu	uu-- 0uuu
休眠期间的 MCLR 复位	0000h	---1 0uuu	uu-- 0uuu
WDT 复位	0000h	---0 uuuu	uu-- uuuu
被 WDT 从休眠模式唤醒	PC + 1	---0 0uuu	uu-- uuuu
欠压复位	0000h	---1 1uuu	00-- 11u0
被中断从休眠模式唤醒	PC + 1 ⁽¹⁾	---1 0uuu	uu-- uuuu
执行了 RESET 指令	0000h	---u uuuu	uu-- u0uu
堆栈上溢复位 (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
堆栈下溢复位 (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

图注：u = 不变，x = 未知，- = 未实现，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。

6.13 电源控制（PCON）寄存器

PCON 寄存器位如[寄存器 6-2](#) 所示。

电源控制（PCON）寄存器包含区分以下各种复位的标志位：

- 上电复位（**POR**）
- 欠压复位（**BOR**）
- RESET 指令复位（**RI**）
- MCLR 复位（**RMCLR**）
- 看门狗定时器复位（**RWDT**）
- 堆栈下溢复位（**STKUNF**）
- 堆栈上溢复位（**STKOVF**）

6.14 寄存器定义：电源控制

寄存器 6-2: **PCON:** 电源控制寄存器

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	—	<u>RWDT</u>	<u>RMCLR</u>	<u>RI</u>	<u>POR</u>	<u>BOR</u>
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 = 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	STKOVF: 堆栈上溢标志位 1 = 发生了堆栈上溢 0 = 未发生堆栈上溢或由固件清零
bit 6	STKUNF: 堆栈下溢标志位 1 = 发生了堆栈下溢 0 = 未发生堆栈下溢或由固件清零
bit 5	<u>未实现:</u> 读为 0
bit 4	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设置为 1 0 = 发生了看门狗定时器复位（由硬件清零）
bit 3	RMCLR: <u>MCLR</u> 复位标志位 1 = 未发生 <u>MCLR</u> 复位或由固件设置为 1 0 = 发生了 <u>MCLR</u> 复位（由硬件清零）
bit 2	RI: <u>RESET</u> 指令标志位 1 = 未执行 <u>RESET</u> 指令或由固件设置为 1 0 = 执行了 <u>RESET</u> 指令（由硬件清零）
bit 1	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）

PIC16(L)F1764/5/8/9

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	BORFS	—	—	—	—	—	BORRDY	91
PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	95
STATUS	—	—	—	TO	PD	Z	DC	C	28
WDTCON	—	—		WDTPS<4:0>			SWDTEN		117

图注：— = 未实现，读为 0。复位不使用阴影单元。

7.0 中断

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

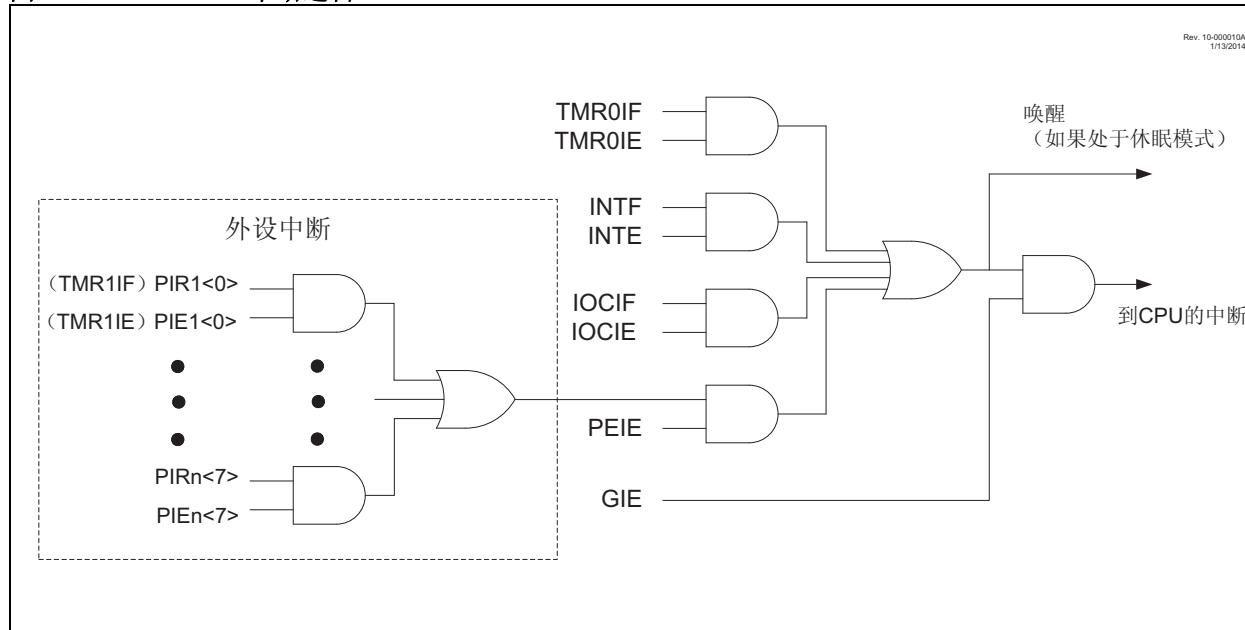
本章包含了关于中断的以下信息：

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

许多外设都会产生中断。详情请参见相应章节。

图 7-1 给出了中断逻辑的框图。

图 7-1： 中断逻辑



7.1 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
- INTCON 寄存器的 PEIE 位（如果中断事件的中断允许位包含在 PIE1 或 PIE2 寄存器中）

INTCON、PIR1 和 PIR2 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 7.5 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，RETFIE 指令退出 ISR。

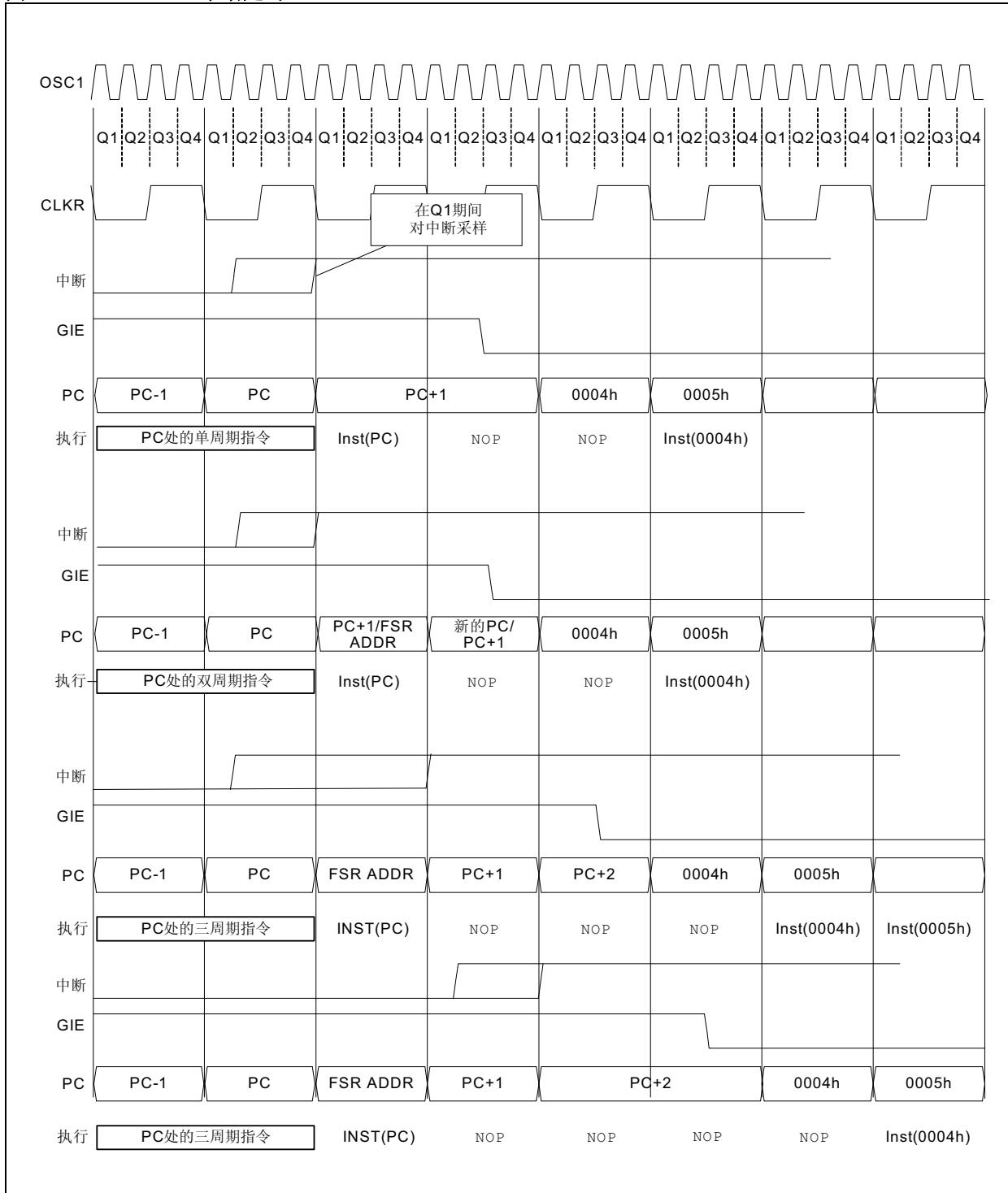
关于特定中断操作的更多信息，请参见其外设章节。

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

7.2 中断延时

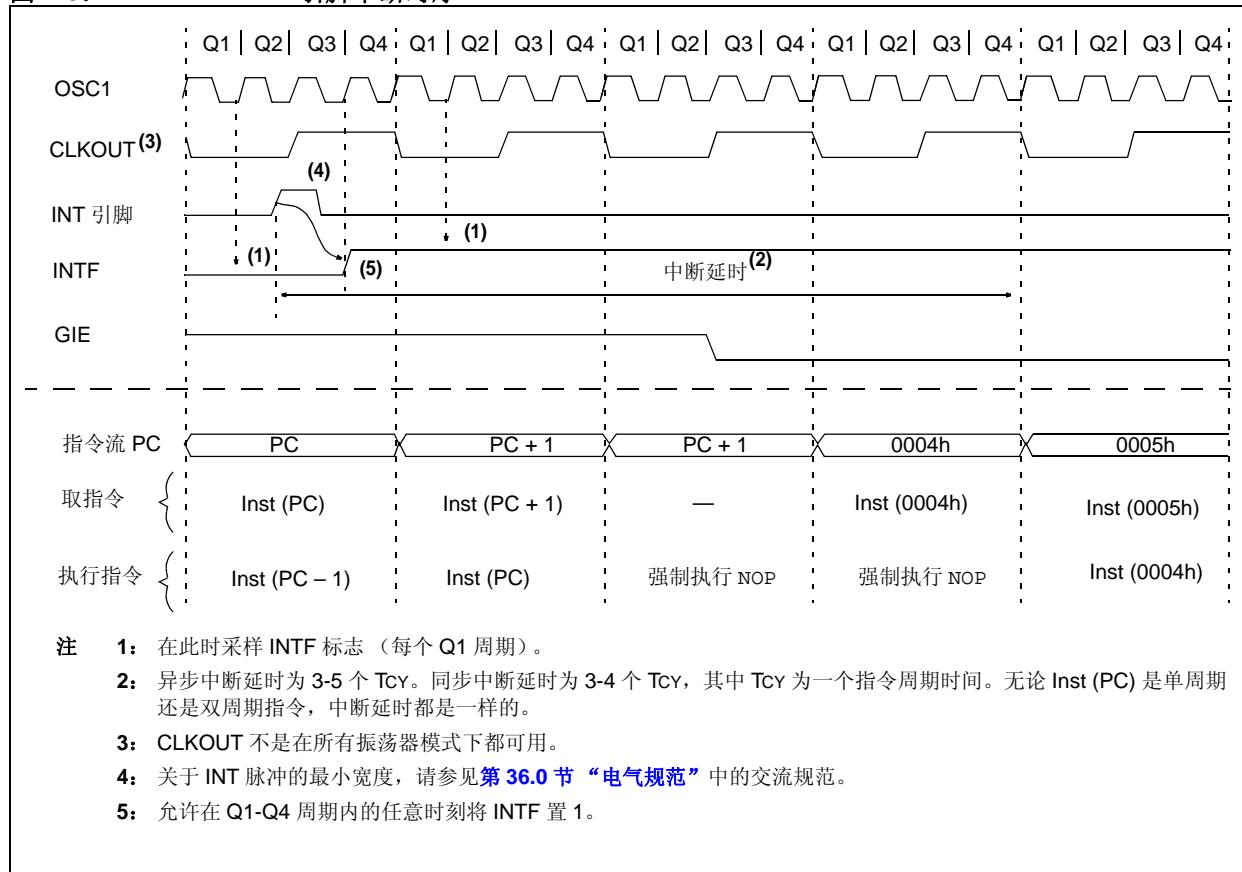
中断延时定义为从发生中断事件到开始执行中断向量处代码经过的时间。同步中断的延时为 3 或 4 个指令周期。对于异步中断，延时为 3 至 5 个指令周期，这取决于中断何时发生。更多详细信息，请参见[图 7-2](#) 和 [图 7-3](#)。

图 7-2: 中断延时



PIC16(L)F1764/5/8/9

图 7-3: INT 引脚中断时序



7.3 休眠期间的中断

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

从休眠模式唤醒时，如果 GIE 位也置 1，则处理器将跳转到中断向量。否则，处理器将继续执行 SLEEP 指令后的指令。紧接 SLEEP 指令后的指令总是会在跳转到 ISR 前执行。更多详细信息，请参见第 8.0 节“掉电模式（休眠）”。

7.4 INT 引脚

INT 引脚可用于产生异步边沿触发中断。可以通过将 INTCON 寄存器的 INTE 位置 1 来允许该中断。OPTION_REG 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。INTCON 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。

7.5 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。

PIC16(L)F1764/5/8/9

7.6 寄存器定义：中断控制

寄存器 7-1: INTCON: 中断控制寄存器

R/W-0/0	R-0/0						
GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF ⁽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 **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 **TMR0IE:** Timer0 溢出中断允许位

1 = 允许 Timer0 中断
0 = 禁止 Timer0 中断

bit 4 **INTE:** INT 外部中断允许位

1 = 允许 INT 外部中断
0 = 禁止 INT 外部中断

bit 3 **IOCIE:** 电平变化中断允许位

1 = 允许电平变化中断
0 = 禁止电平变化中断

bit 2 **TMR0IF:** Timer0 溢出中断标志位

1 = TMR0 寄存器已溢出
0 = TMR0 寄存器未溢出

bit 1 **INTF:** INT 外部中断标志位

1 = 发生了 INT 外部中断
0 = 未发生 INT 外部中断

bit 0 **IOCIF:** 电平变化中断标志位⁽¹⁾

1 = 至少有一个电平变化中断引脚改变了状态
0 = 没有任何电平变化中断引脚的状态发生改变

注 1: IOCIF 标志位是只读位，它在 IOCxF 寄存器中的所有电平变化中断标志都已由软件清零后清零。

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。用户软件应在允许一个中断前，先将相应的中断标志位清零。

寄存器 7-2: PIE1: 外设中断允许寄存器 1

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TMR1GIE | ADIE | RCIE | TXIE | SSP1IE | CCP1IE | 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 **CCP1IE:** CCP1 中断允许位

1 = 允许 CCP1 中断
0 = 禁止 CCP1 中断

bit 1 **TMR2IE:** TMR2 与 T2PR 匹配中断允许位

1 = 允许 Timer2 与 T2PR 匹配中断
0 = 禁止 Timer2 与 T2PR 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: 必须将 INTCON 寄存器的 PEIE 位置 1,
以允许任何外设中断。

PIC16(L)F1764/5/8/9

寄存器 7-3: PIE2: 外设中断允许寄存器 2

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽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 **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 **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 未实现: 读为 0

bit 3 **BCL1IE:** MSSP 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 2 **C4IE:** 比较器 C4 中断允许位⁽¹⁾

1 = 允许比较器 C4 中断

0 = 禁止比较器 C4 中断

bit 1 **C3IE:** 比较器 C3 中断允许位⁽¹⁾

1 = 允许比较器 C3 中断

0 = 禁止比较器 C3 中断

bit 0 **CCP2IE:** CCP2 中断允许位⁽¹⁾

1 = 允许 CCP2 中断

0 = 禁止 CCP2 中断

注 1: 仅限 PIC16(L)F1768/9。

注: 必须将 INTCON 寄存器的 PEIE 位置 1,
以允许任何外设中断。

寄存器 7-4: PIE3: 外设中断允许寄存器 3

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
PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE
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 **PWM6IE:** PWM6 中断允许位⁽¹⁾

1 = 允许 PWM6 中断

0 = 禁止 PWM6 中断

bit 6 **PWM5IE:** PWM5 中断允许位

1 = 允许 PWM5 中断

0 = 禁止 PWM5 中断

bit 5 **COG1IE:** COG1 自动关断中断允许位

1 = 允许 COG1 中断

0 = 禁止 COG1 中断

bit 4 **ZCDIE:** 过零检测中断允许位

1 = 允许 ZCD 中断

0 = 禁止 ZCD 中断

bit 3 **COG2IE:** COG2 自动关断中断允许位⁽¹⁾

1 = 允许 COG2 中断

0 = 禁止 COG2 中断

bit 2 **CLC3IE:** CLC3 中断允许位

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 1: 仅限 PIC16(L)F1768/9。

注: 必须将 INTCON 寄存器的 PEIE 位置 1,
以允许任何外设中断。

PIC16(L)F1764/5/8/9

寄存器 7-5: PIR1: 外设中断请求寄存器 1

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	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

bit 7 **TMR1GIF:** Timer1 门控中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 6 **ADIF:** 模数转换器 (ADC) 中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 5 **RCIF:** EUSART 接收中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 4 **TXIF:** EUSART 发送中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 3 **SSP1IF:** 同步串行端口 (MSSP) 中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 2 **CCP1IF:** CCP1 中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 1 **TMR2IF:** Timer2 与 T2PR 匹配中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 0 **TMR1IF:** Timer1 溢出中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

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。用户软件应在允许一个中断前, 先将相应的中断标志位清零。

寄存器 7-6: PIR2: 外设中断请求寄存器 2

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽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 **OSFIF:** 振荡器故障中断标志位

1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit 6 **C2IF:** 比较器 C2 中断标志位

1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit 5 **C1IF:** 比较器 C1 中断标志位

1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit 4 未实现: 读为 0

bit 3 **BCL1IF:** MSSP 总线冲突中断标志位

1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit 2 **C4IF:** 比较器 C4 中断标志位⁽¹⁾

1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit 1 **C3IF:** 比较器 C3 中断标志位⁽¹⁾

1 = 中断处于待处理状态
0 = 中断不处于待处理状态

bit 0 **CCP2IF:** CCP2 中断标志位⁽¹⁾

1 = 中断处于待处理状态
0 = 中断不处于待处理状态

注 1: 仅限 PIC16(L)F1768/9。

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。用户软件应在允许一个中断前, 先将相应的中断标志位清零。

PIC16(L)F1764/5/8/9

寄存器 7-7: PIR3: 外设中断请求寄存器 3

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
PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	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

bit 7 **PWM6IF:** PWM6 中断标志位⁽¹⁾

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 6 **PWM5IF:** PWM5 中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 5 **COG1IF:** COG1 自动关断中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 4 **ZCDIF:** 过零检测中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 3 **COG2IF:** COG2 自动关断中断标志位⁽¹⁾

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 2 **CLC3IF:** CLC3 中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 1 **CLC2IF:** CLC2 中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 0 **CLC1IF:** CLC1 中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

注 1: 仅限 PIC16(L)F1768/9。

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。用户软件应在允许一个中断前, 先将相应的中断标志位清零。

表 7-1：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	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			219
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108

图注：— = 未实现位，读为 0。中断不使用阴影单元。

注 1：仅限 PIC16(L)F1768/9。

8.0 掉电模式（休眠）

可通过执行 SLEEP 指令进入掉电模式。

在进入休眠模式时，会存在以下条件：

1. 如果在休眠期间使能 WDT，则 WDT 会清零，但保持运行。
2. STATUS 寄存器的 **PD** 位被清零。
3. STATUS 寄存器的 **TO** 位被置 1。
4. CPU 时钟被禁止。
5. 31 kHz LFINTOSC 不受影响，使用它工作的外设可以在休眠模式下继续工作。
6. 当所选的 Timer1 时钟源为以下时钟源时，Timer1 和使用 Timer1 工作的外设可以在休眠模式下继续工作：
 - LFINTOSC
 - T1CKI
 - 辅助振荡器
7. 如果选择了专用 FRC 振荡器，则 ADC 不受影响。
8. I/O 端口保持执行 SLEEP 指令之前的状态（驱动为高电平、低电平或高阻态）。
9. WDT 之外的其他复位都不会受休眠模式影响。

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 31 kHz LFINTOSC 的模块
- 使用辅助振荡器的模块

为了避免输入引脚悬空而引入开关电流，应在外部将高阻抗输入的 I/O 引脚拉到 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 17.0 节“5 位数模转换器（DAC）模块”](#)和[第 14.0 节“固定参考电压（FVR）”](#)。

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

前三个事件会导致器件复位。后三个事件被认为是程序执行的继续。要确定是发生了器件复位还是唤醒事件，请参见[第 6.12 节“确定复位原因”](#)。

当执行 SLEEP 指令时，下一条指令（PC + 1）被预取出。如果希望通过中断事件唤醒器件，则必须允许相应的中断允许位。唤醒与 GIE 位的状态无关。如果 GIE 位被禁止，器件将继续执行 SLEEP 指令之后的指令。如果 GIE 位被允许，器件将执行 SLEEP 指令之后的指令，然后器件将调用中断服务程序。如果不希望执行 SLEEP 指令之后的指令，用户应在 SLEEP 指令后面放置一条 NOP 指令。

器件从休眠模式唤醒时，WDT 都将被清零，而与唤醒原因无关。

8.1.1 使用中断唤醒

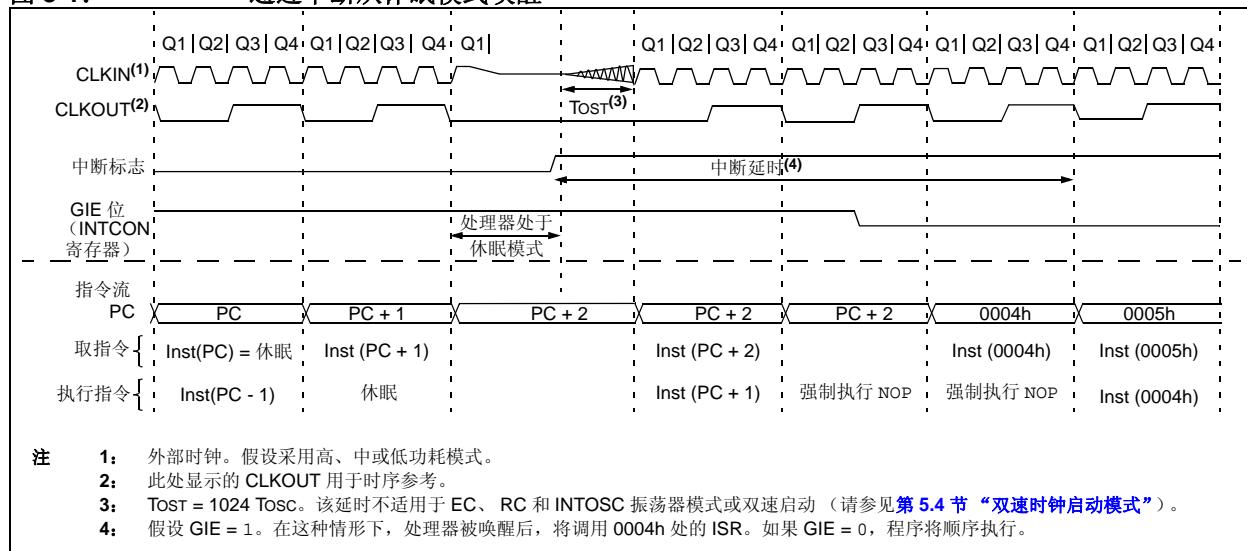
当禁止全局中断（GIE 被清零）时，并且任一中断源的中断允许位和中断标志位都置 1，将会发生以下事件之一：

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

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

即使在执行 SLEEP 指令之前检查了标志位，这些标志位也有可能在 SLEEP 指令执行完毕之前被置 1。要确定是否执行了 SLEEP 指令，可测试 PD 位。如果 PD 位置 1，则说明 SLEEP 指令作为 NOP 指令执行了。

图 8-1：通过中断从休眠模式唤醒



8.2 低功耗休眠模式

PIC16F1764/5/8/9 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 及其关联的参考电压电路必须保持活动状态。PIC16F1764/5/8/9 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。如果该位置 1，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。

8.2.1 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，所以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。

8.2.2 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仅与以下外设配合使用：

- 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 (外部时钟源 < 100 kHz)

注:

PIC16LF1764/5/8/9不具有可配置的低功耗休眠模式。PIC16LF1764/5/8/9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。这些器件的最大 V_{DD} 和 I/O 电压低于 PIC16F1764/5/8/9。更多信息，请参见[第 36.0 节 “电气规范”](#)。

8.3 寄存器定义：稳压器控制

寄存器 8-1: 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 PIC16F1764/5/8/9。

2: 请参见第 36.0 节 “电气规范”。

表 8-1: 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	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	166
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	166
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	167
IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	167
IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	168
IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	168
IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	169
IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	169
IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	169
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
PIR2	OSFIG	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C2IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IE ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108
STATUS	—	—	—	TO	PD	Z	DC	C	28
VREGCON ⁽²⁾	—	—	—	—	—	—	VREGPM	保留	113
WDTCON	—	—	—	—	WDTPS<4:0>	—	—	SWDTEN	117

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)F1768/9。

2: 仅限 PIC16F1764/5/8/9。

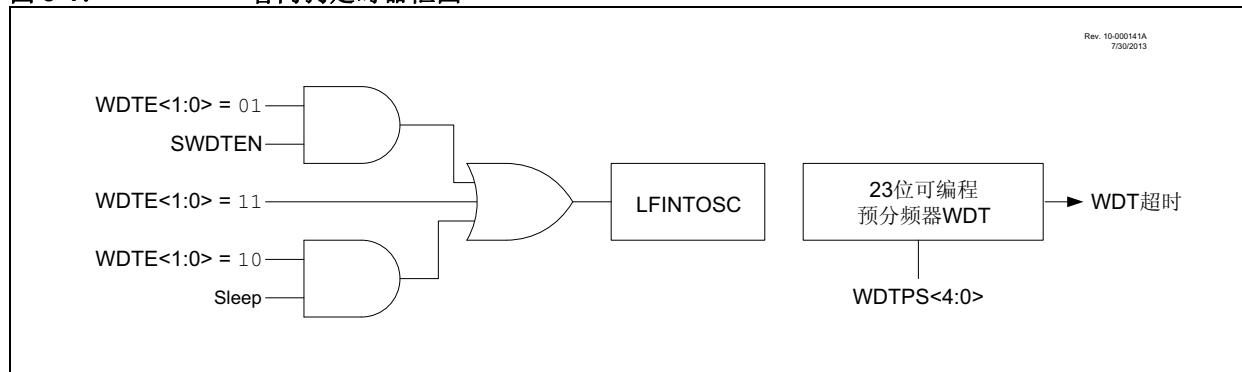
9.0 看门狗定时器 (WDT)

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）
- 多种复位条件
- 休眠期间的操作

图 9-1：看门狗定时器框图



9.1 独立时钟源

WDT 以 31 kHz LFINTOSC 内部振荡器作为其工作的时基。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 36-8。

9.2 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 9-1。

9.2.1 WDT 总是开启

当配置字的 WDTE 位设置为 11 时，WDT 将总是开启。

WDT 保护在休眠期间有效。

9.2.2 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 将开启。

WDT 保护在休眠期间无效。

9.2.3 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 9-1。

表 9-1：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
			禁止
00	X	X	禁止

9.3 超时周期

WDTCON 寄存器的 WDTPS 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。

9.4 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 9-2。

9.5 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 5.0 节“振荡器模块（带故障保护时钟监视器）”。

在器件处于休眠模式的情况下发生 WDT 超时时，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 位会发生改变，指示发生的事件。更多信息，请参见 STATUS 寄存器（寄存器 3-1）。

表 9-2: WDT 清零条件

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

9.6 寄存器定义：看门狗控制

寄存器 9-1： 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

-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 5-1 **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。

PIC16(L)F1764/5/8/9

表 9-3: 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
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		86
STATUS	—	—	—	TO	PD	Z	DC	C	28
WDTCON	—	—	WDTPS<4:0>				SWDTEN	—	117

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 9-4: 与看门狗定时器相关的配置字汇总

名称	位	Bit -7	Bit -6	Bit 13/5	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	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

图注: — = 未实现位, 读为 0。看门狗定时器不使用阴影单元。

10.0 闪存程序存储器控制

在整个 V_{DD} 范围内的正常工作期间，闪存程序存储器都是可读写的。程序存储器通过特殊功能寄存器（SFR）来间接寻址。用于访问程序存储器的 SFR 有：

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

当访问程序存储器时，PMDATH:PMDATL 寄存器对组成双字节字，保存 14 位读/写数据，而 PMADRH:PMADRL 寄存器对组成双字节字，保存 15 位被读取的程序存储单元的地址。

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。

闪存程序存储器可以通过两种方式进行保护：代码保护（配置字中的 \overline{CP} 位）和写保护（配置字中的 $WRT<1:0>$ 位）。

代码保护 ($\overline{CP} = 0$) (1) 会禁止通过外部器件编程器对闪存程序存储器进行访问（读写操作）。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，从而清除所有闪存程序存储器、配置位和用户 ID 而复位。

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

注 1: 整个闪存程序存储器阵列的代码保护通过清零配置字的 \overline{CP} 位来使能。

10.1 PMADRL 和 PMADRH 寄存器

PMADRH:PMADRL 寄存器对能寻址最大 32K 字的程序存储器。当选择程序地址值时，地址的 MSB 被写入 PMADRH 寄存器，而 LSB 被写入 PMADRL 寄存器。

10.1.1 PMCON1 和 PMCON2 寄存器

PMCON1 是访问闪存程序存储器的控制寄存器。

控制位 RD 和 WR 分别用于启动读和写操作。用软件只能将这两位置 1 而无法清零。在读或写操作完成后，它们由硬件清零。由于无法用软件将 WR 位清零，可避免意外地过早终止写操作。

当 WREN 位置 1 时，允许进行写操作。上电时，WREN 位被清零。在正常工作期间，如果写操作被复位中断，WRERR 位会置 1。在这些情况下，复位后用户可以检查 WRERR 位并执行相应的错误处理程序。

PMCON2 寄存器是只写寄存器。尝试读 PMCON2 寄存器将返回全 0。

要使能对程序存储器的写操作，必须向 PMCON2 寄存器中写入特定的模式（解锁序列）。必需的解锁序列可以防止对程序存储器写锁存器和闪存程序存储器的意外写操作。

10.2 闪存程序存储器概述

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。每一行都包含固定数量的 14 位程序存储字。行是可以通过用户软件擦除的最小大小。

在擦除某行之后，用户可以对该行的全部或部分内容进行再编程。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写锁存器，但可以通过连续写入 PMDATL:PMDATH 寄存器对来装入数据。

注: 如果用户只希望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 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。但对于任何未经过编程的单元，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单元。

关于闪存程序存储器的擦除行大小和写锁存器数量，请参见表 10-1。

PIC16(L)F1764/5/8/9

表 10-1: 闪存构成 (按器件)

器件	行擦除 (字)	写锁存器 (字)
PIC16(L)F1764		
PIC16(L)F1765		
PIC16(L)F1768		
PIC16(L)F1769	32	32

10.2.1 读取闪存程序存储器

要读取程序存储单元，用户必须：

1. 将所需地址写入 PMADRH:PMADRL 寄存器对。
2. 将 PMCON1 寄存器的 CFGS 位清零。
3. 然后，将 PMCON1 寄存器的控制位 RD 置 1。

一旦读控制位置 1，闪存程序存储器控制器将使用第二个指令周期来读取数据。这会导致紧随“BSF PMCON1, RD”指令的第二条指令被忽略。在紧接着的下一个周期，PMDATH:PMDATL 寄存器对中的数据即可使用；因此，可在随后的指令中读取为两个字节。

PMDATH:PMDATL 寄存器对将保存该值直到另一次读操作开始或用户写入新值为止。

注：程序存储器读操作后的两条指令必须为 NOP，从而阻止用户在 RD 位置 1 后的下一条指令执行双周期指令。

图 10-1: 闪存程序存储器读操作流程图

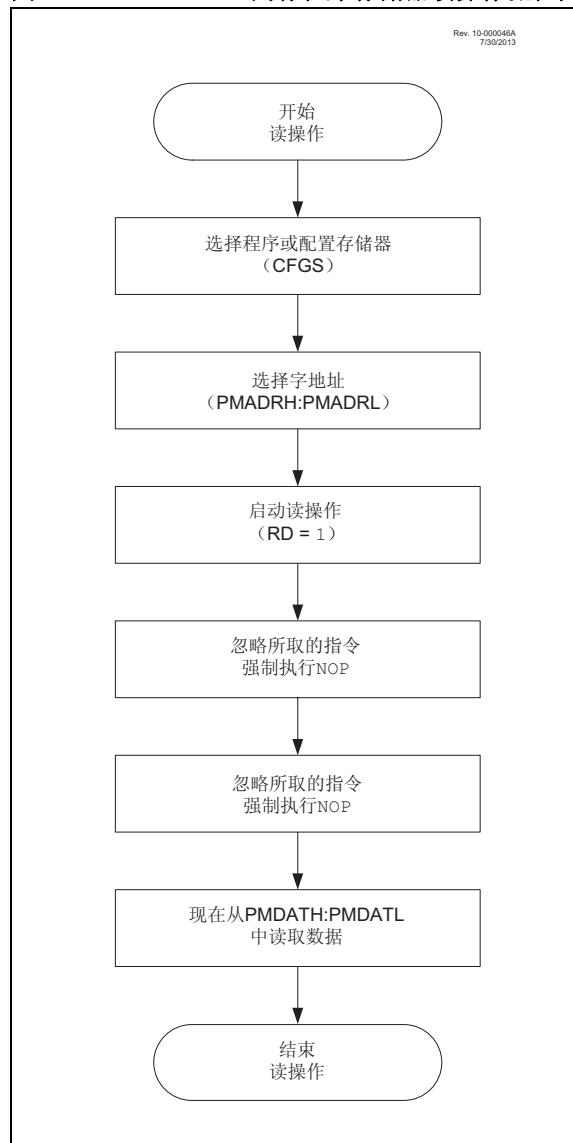
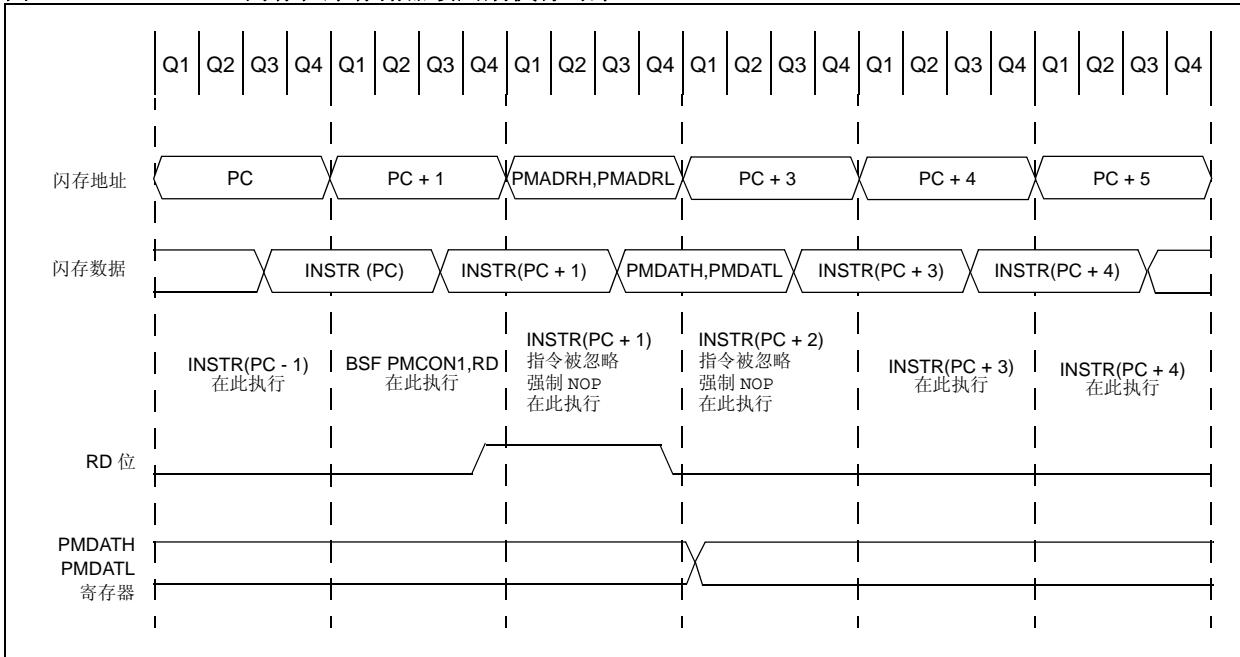


图 10-2: 闪存程序存储器读周期执行时序



例 10-1: 读取闪存程序存储器

```

* 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    PMADRL          ; Select Bank for PMCON registers
    MOVLW     PROG_ADDR_LO      ;
    MOVWF     PMADRL          ; Store LSB of address
    MOVLW     PROG_ADDR_HI      ;
    MOVWF     PMADRH          ; Store MSB of address

    BCF      PMCON1,CFG5        ; Do not select Configuration Space
    BSF      PMCON1,RD          ; Initiate read
    NOP          ; Ignored (Figure 10-1)
    NOP          ; Ignored (Figure 10-1)

    MOVF     PMDATL,W           ; Get LSB of word
    MOVWF    PROG_DATA_LO       ; Store in user location
    MOVF     PMDATH,W           ; Get MSB of word
    MOVWF    PROG_DATA_HI       ; Store in user location

```

10.2.2 闪存解锁序列

解锁序列是一种用于保护闪存程序存储器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：

- 行擦除
- 向程序存储器写锁存器装入数据
- 将程序存储器写锁存器内容写入程序存储器
- 将程序存储器写锁存器内容写入用户 ID

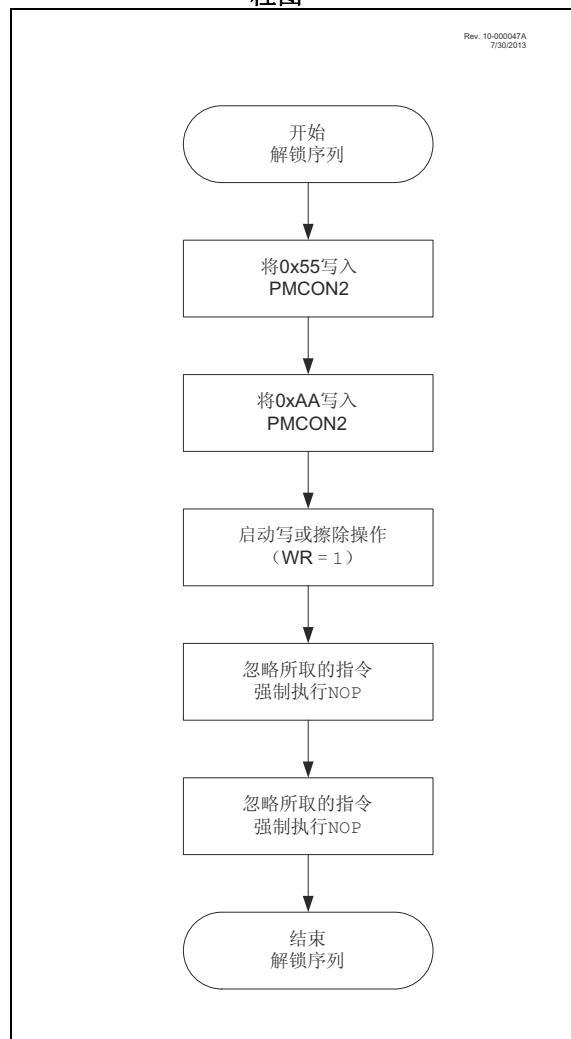
解锁序列包含以下步骤：

1. 将 55h 写入 PMCON2
2. 将 AAh 写入 PMCON2
3. 将 PMCON1 中的 WR 位置 1
4. NOP 指令
5. NOP 指令

在 WR 位置 1 之后，处理器总是会强制执行两条 NOP 指令。在执行擦除行或编程行操作时，处理器会暂停内部操作（通常为 2 ms），直到操作完成为止，然后再继续执行下一条指令。当操作向程序存储器写锁存器装入数据时，处理器总是会强制执行两条 NOP 指令，然后继续无中断地执行下一条指令。

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 10-3：闪存程序存储器解锁序列流程图



10.2.3 擦除闪存程序存储器

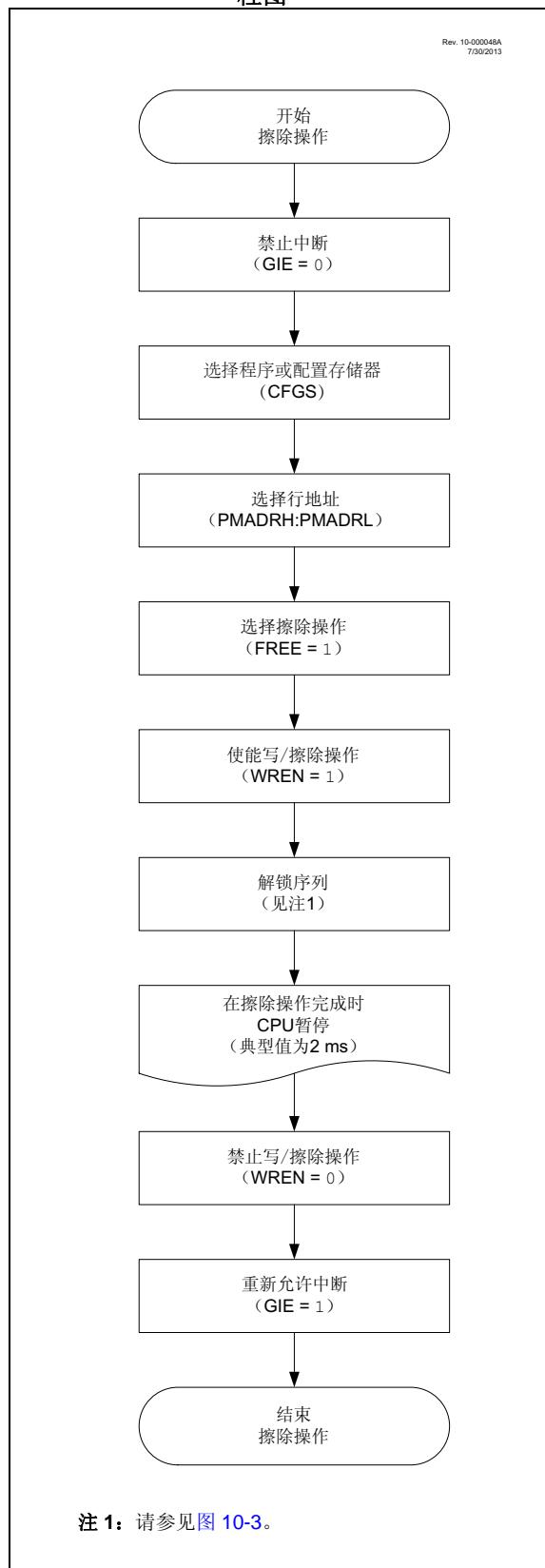
在执行代码时，程序存储器只能按行进行擦除。要擦除某行，请执行以下步骤：

1. 将要擦除的行内的任意地址装入 PMADRH:PMADRL 寄存器对。
2. 将 PMCON1 寄存器的 CFGS 位清零。
3. 将 PMCON1 寄存器的 FREE 和 WREN 位置 1。
4. 向 PMCON2 中先写入 55h，然后写入 AAh（闪存编程解锁序列）。
5. 将 PMCON1 寄存器的控制位 WR 置 1，以开始擦除操作。

请参见例 10-2。

在“BSF PMCON1,WR”指令之后，处理器需要两个周期来设置擦除操作。用户必须在紧跟 WR 位置 1 指令之后放置两条 NOP 指令。处理器将暂停内部操作，产生 2 ms（典型值）的擦除时间。这不是休眠模式，因为时钟和外设会继续运行。在擦除周期之后，处理器将继续处理 PMCON1 写指令之后的第三条指令。

图 10-4: 闪存程序存储器擦除操作流程图



PIC16(L)F1764/5/8/9

例 10-2: 擦除程序存储器的一行

```
; This row erase routine assumes the following:  
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL  
; 2. ADDRH and ADDRDL are located in shared data memory 0x70 - 0x7F (common RAM)  
  
BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly  
BANKSEL PMADRL  
MOVF    ADDR, W          ; Load lower 8 bits of erase address boundary  
MOVWF  PMADRL  
MOVF    ADDRH, W         ; Load upper 6 bits of erase address boundary  
MOVWF  PMADRH  
BCF      PMCON1, CFGS    ; Not configuration space  
BSF      PMCON1, FREE    ; Specify an erase operation  
BSF      PMCON1, WREN    ; Enable writes  
  
[必需的序列]  
MOVLW  55h              ; Start of required sequence to initiate erase  
MOVWF  PMCON2            ; Write 55h  
MOVLW  0AAh              ;  
MOVWF  PMCON2            ; Write AAh  
BSF    PMCON1, WR        ; Set WR bit to begin erase  
NOP                ; NOP instructions are forced as processor starts  
NOP                ; row erase of program memory.  
;  
; The processor stalls until the erase process is complete  
; after erase processor continues with 3rd instruction  
  
BCF      PMCON1, WREN    ; Disable writes  
BSF      INTCON, GIE     ; Enable interrupts
```

10.2.4 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. 将需要编程的行的地址装入 PMADRH:PMADRL。
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 10-5（使用 32 个写锁存器对程序存储器进行写操作）。

写锁存器将对齐到由 PMADRH:PMADRL 高 10 位 (PMADRH<6:0>:PMADRL<7:5>) 定义的闪存行地址边界处，PMADRL 的低 5 位 (PMADRL<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 PMDATH:PMDATL 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 PMCON1 寄存器的 WREN 位置 1。
2. 将 PMCON1 寄存器的 CFGS 位清零。
3. 将 PMCON1 寄存器的 LWLO 位置 1。当 PMCON1 寄存器的 LWLO 位为 1 时，写序列之后只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 PMADRH:PMADRL 寄存器对。
5. 将要写入的程序存储器数据装入 PMDATH:PMDATL 寄存器对。
6. 执行解锁序列（[第 10.2.2 节“闪存解锁序列”](#)）。此时，将数据装入写锁存器。
7. 递增 PMADRH:PMADRL 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 PMCON1 寄存器的 LWLO 位清零。当 PMCON1 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 PMDATH:PMDATL 寄存器对。
11. 执行解锁序列（[第 10.2.2 节“闪存解锁序列”](#)）。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 10-3 给出了一个完整写序列的示例。初始地址装入 PMADRH:PMADRL 寄存器对；数据使用间接寻址方式装入。

图 10-5: 使用 32 个写锁存器对闪存程序存储器进行块写操作

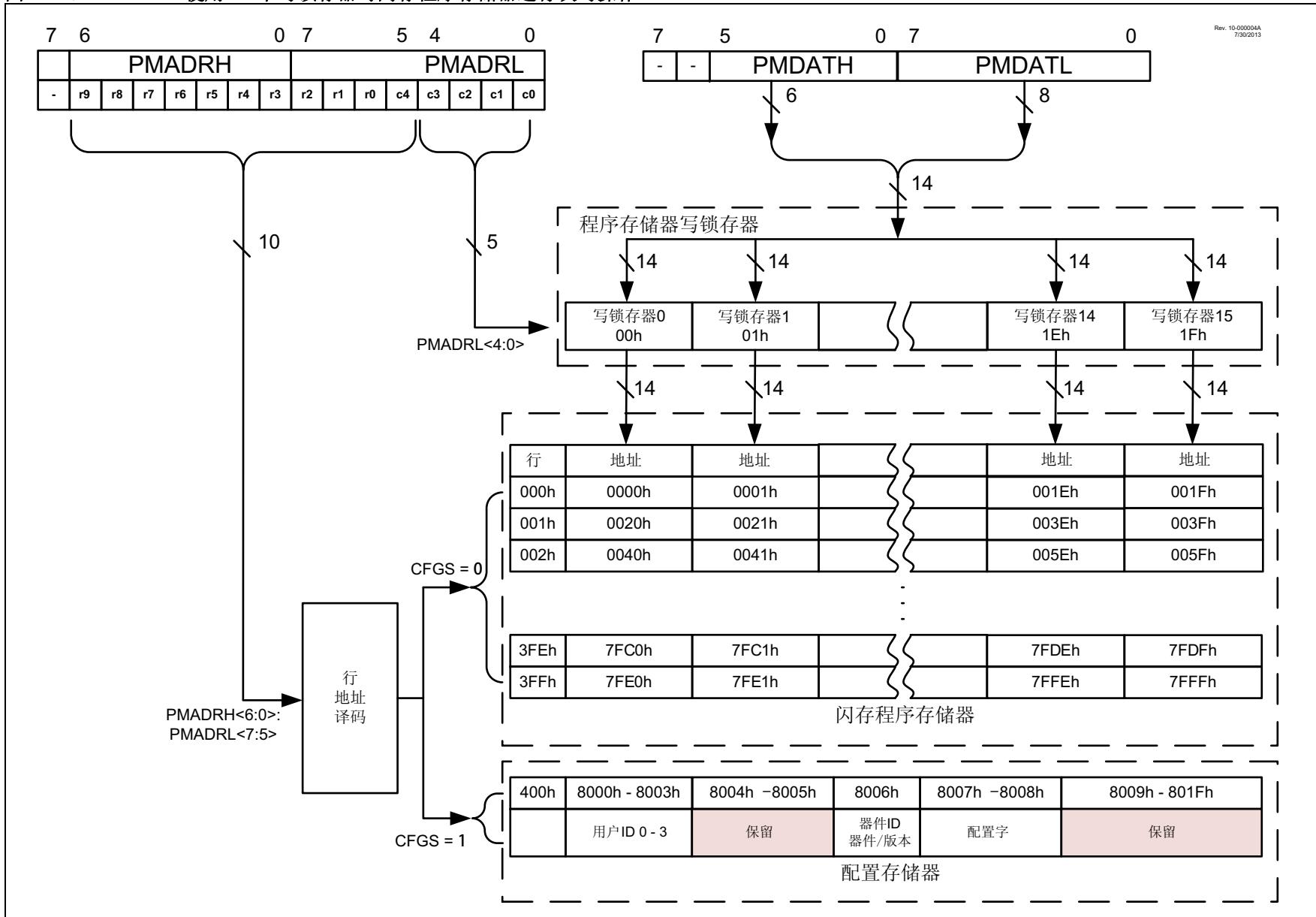
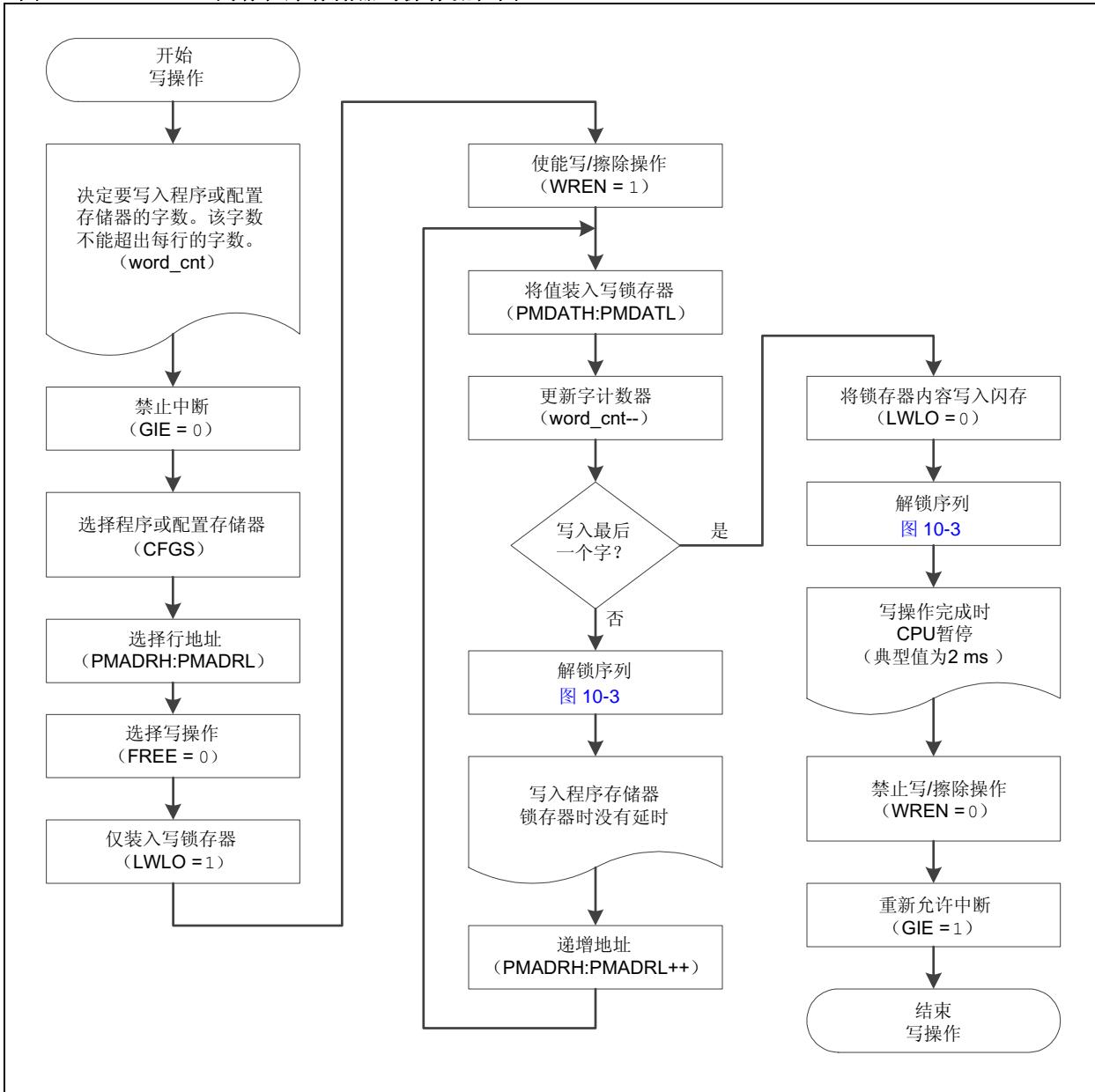


图 10-6: 闪存程序存储器写操作流程图



PIC16(L)F1764/5/8/9

例 10-3：写入闪存程序存储器

```
; This write routine assumes the following:  
; 1. 64 bytes 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 shared data memory 0x70 - 0x7F (common RAM)  
  
        BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly  
        BANKSEL  PMADRH        ; Bank 3  
        MOVF    ADDR,W         ; Load initial address  
        MOVWF   PMADRH        ;  
        MOVF    ADDR,W         ;  
        MOVWF   PMADRL        ;  
        MOVLW   LOW DATA_ADDR ; Load initial data address  
        MOVWF   FSROL         ;  
        MOVLW   HIGH DATA_ADDR; Load initial data address  
        MOVWF   FSROH         ;  
        BCF      PMCON1,CFG5  ; Not configuration space  
        BSF      PMCON1,WREN  ; Enable writes  
        BSF      PMCON1,LWLO  ; Only Load Write Latches  
  
LOOP  
        MOVIW   FSR0++        ; Load first data byte into lower  
        MOVWF   PMDATL        ;  
        MOVIW   FSR0++        ; Load second data byte into upper  
        MOVWF   PMDATH        ;  
  
        MOVF    PMADRL,W      ; Check if lower bits of address are '00000'  
        XORLW   0x1F          ; Check if we're on the last of 32 addresses  
        ANDLW   0x1F          ;  
        BTFSC  STATUS,Z       ; Exit if last of 32 words,  
        GOTO   START_WRITE    ;  
  
        MOVLW   55h           ; Start of required write sequence:  
        MOVWF   PMCON2        ; Write 55h  
        MOVLW   0AAh          ;  
        MOVWF   PMCON2        ; Write AAh  
        BSF    PMCON1,WR       ; Set WR bit to begin write  
        NOP                 ; NOP instructions are forced as processor  
                           ; loads program memory write latches  
        NOP                 ;  
  
        INCF    PMADRL,F      ; Still loading latches Increment address  
        GOTO   LOOP           ; Write next latches  
  
START_WRITE  
        BCF    PMCON1,LWLO    ; No more loading latches - Actually start Flash program  
                           ; memory write  
  
        MOVLW   55h           ; Start of required write sequence:  
        MOVWF   PMCON2        ; Write 55h  
        MOVLW   0AAh          ;  
        MOVWF   PMCON2        ; Write AAh  
        BSF    PMCON1,WR       ; Set WR bit to begin write  
        NOP                 ; NOP instructions are forced as processor writes  
                           ; all the program memory write latches simultaneously  
                           ; to program memory.  
        NOP                 ; After NOPs, the processor  
                           ; stalls until the self-write process is complete  
                           ; after write processor continues with 3rd instruction  
        BCF    PMCON1,WREN    ; Disable writes  
        BSF    INTCON,GIE     ; Enable interrupts
```

必需的序列

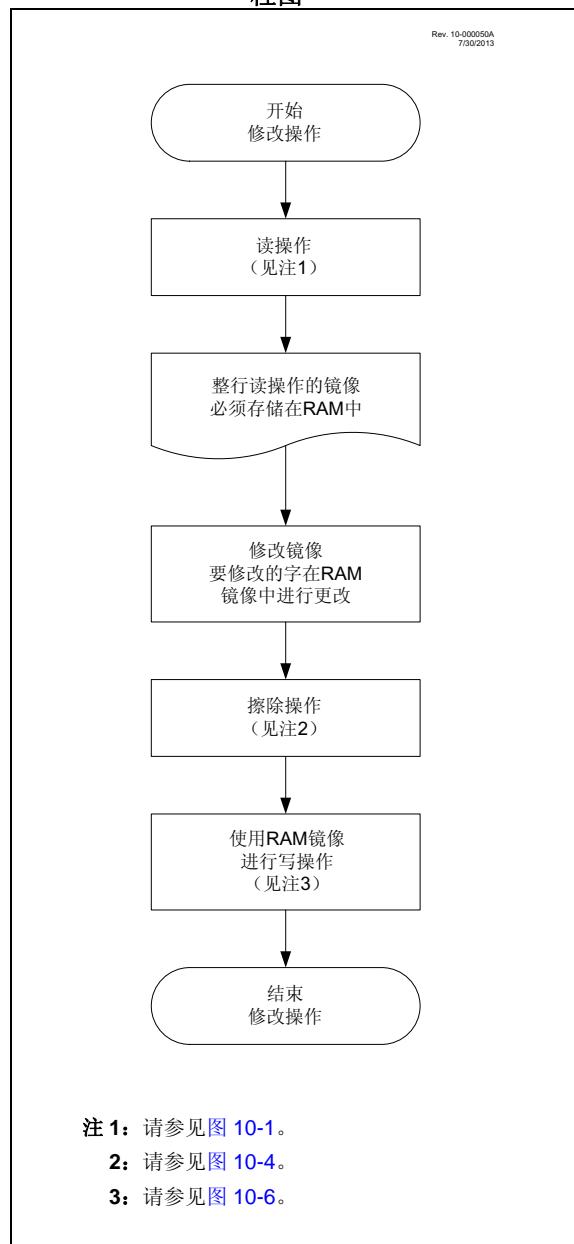
必需的序列

10.3 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 10-7：闪存程序存储器修改操作流程图



10.4 用户 ID、器件 ID 和配置字访问

当 PMCON1 寄存器中的 CFGS = 1 时，用户可以访问用户 ID、器件 ID/ 版本 ID 和配置字，而不是访问程序存储器。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 10-2。

对表 10-2 中所列参数之外的地址启动读访问时，PMDATH:PMDATL 寄存器对会被清零，读回 0。

表 10-2： 用户 ID、器件 ID 和配置字访问 (CFGs = 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-8008h	配置字 1 和 2	是	否

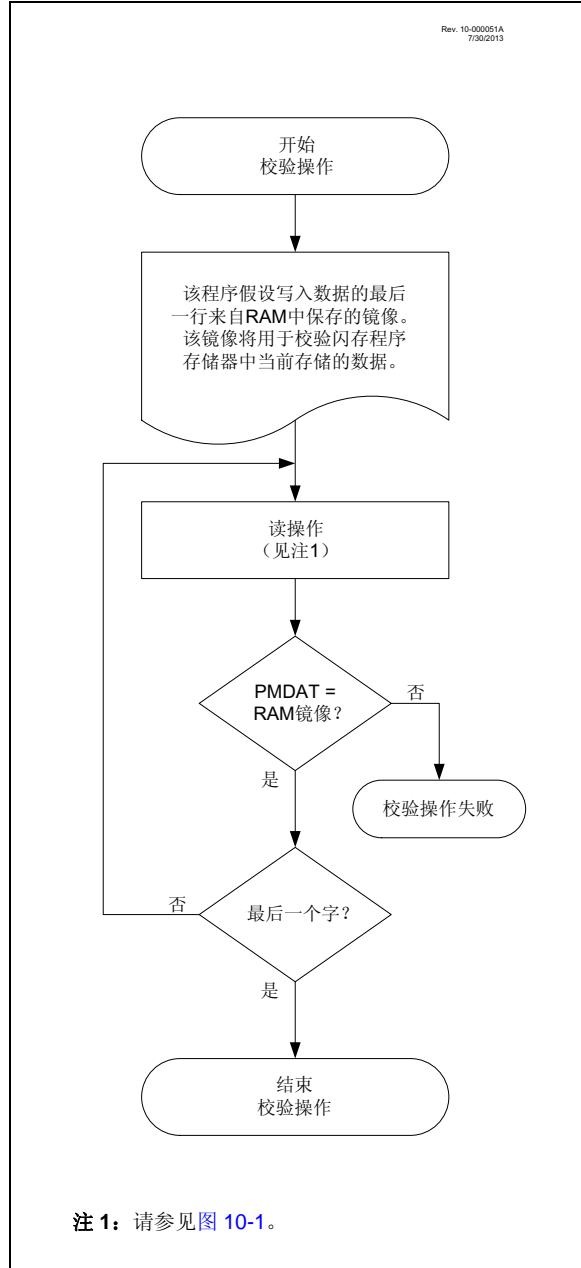
例 10-4： 配置字和器件 ID 访问

```
* This code block will read 1 word of program memory at the memory address:  
* PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;  
* PROG_DATA_HI, PROG_DATA_LO  
  
BANKSEL PMADRL ; Select correct Bank  
MOVLW PROG_ADDR_LO ;  
MOVWF PMADRL ; Store LSB of address  
CLRF PMADRH ; Clear MSB of address  
  
BSF PMCON1,CFGS ; Select Configuration Space  
BCF INTCON,GIE ; Disable interrupts  
BSF PMCON1,RD ; Initiate read  
NOP ; Executed (See Figure 10-2)  
NOP ; Ignored (See Figure 10-2)  
BSF INTCON,GIE ; Restore interrupts  
  
MOVF PMDATL,W ; Get LSB of word  
MOVWF PROG_DATA_LO ; Store in user location  
MOVF PMDATL,W ; Get MSB of word  
MOVWF PROG_DATA_HI ; Store in user location
```

10.5 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页形式存储，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 10-8：闪存程序存储器校验操作流程图



PIC16(L)F1764/5/8/9

10.6 寄存器定义：闪存程序存储器控制

寄存器 10-1: **PMDATL:** 程序存储器数据低字节寄存器

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
PMDAT<7:0>							
bit 7							

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 **PMDAT<7:0>:** 程序存储器最低有效位的读 / 写值

寄存器 10-2: **PMDATH:** 程序存储器数据高字节寄存器

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
—	—	PMDAT<13:8>					
bit 7							

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 **PMDAT<13:8>:** 程序存储器最高有效位的读 / 写值

寄存器 10-3: **PMADRL:** 程序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
PMADR<7:0>							
bit 7							

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 **PMADR<7:0>:** 指定程序存储器地址的最低有效位

寄存器 10-4: PMADRH: 程序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
—(1)				PMADR<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 PMADR<14:8>: 指定程序存储器地址的最高 7 位

注 1: 未实现位, 读为 1。

PIC16(L)F1764/5/8/9

寄存器 10-5: PMCON1: 程序存储器控制 1 寄存器

U-1	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q ⁽²⁾	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
— ⁽¹⁾	CFGs	LWLO ⁽³⁾	FREE	WRERR	WREN	写	RD
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

HC = 硬件清零位

- bit 7 **未实现:** 读为 1
- bit 6 **CFGs:** 配置选择位
1 = 访问配置、用户 ID 和器件 ID 寄存器
0 = 访问闪存程序存储器
- bit 5 **LWLO:** 仅装入写锁存器位⁽³⁾
1 = 在下一条 WR 命令时仅装入 / 更新所寻址的程序存储器写锁存器
0 = 在下一条 WR 命令时装入 / 更新所寻址的程序存储器写锁存器, 并启动对于所有程序存储器写锁存器的写操作
- bit 4 **FREE:** 程序闪存擦除使能位
1 = 在下一条 WR 命令时执行擦除操作 (完成后由硬件清零)
0 = 在下一条 WR 命令时执行写操作
- bit 3 **WRERR:** 编程 / 擦除错误标志位
1 = 该条件指示试图 / 终止执行不合法的编程或擦除序列 (试图将 WR 位置 1 (写入 1) 时自动将该位置 1)
0 = 编程或擦除操作正常完成
- 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
1 = 启动程序闪存的编程或擦除操作。
 操作是自定时的, 一旦该操作完成, 该位即由硬件清零。
 用软件只能将 WR 位置 1 (不能清零)。
0 = 对闪存的编程 / 擦除操作已完成并且变为无效
- bit 0 **RD:** 读控制位
1 = 启动程序闪存的读操作。读操作需要一个周期。 RD 由硬件清零。用软件只能将 RD 位置 1 (不能清零)。
0 = 不启动程序闪存的读操作
- 注**
- 1:** 未实现位, 读为 1。
 - 2:** 在程序存储器写操作或擦除操作启动 (WR = 1) 时, 硬件会自动将 WRERR 位置 1。
 - 3:** 在程序存储器擦除操作期间 (FREE = 1), LWLO 位会被忽略。

寄存器 10-6: PMCON2: 程序存储器控制 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
程序存储器控制寄存器 2							
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

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 PMCON1 寄存器的 WR 位置 1。写入该寄存器的值用于对写操作进行解锁。对于这些写操作, 存在一些特定的时序要求。

表 10-3: 与闪存程序存储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
INTCON	GIE	PEIE	TMROIE	INTE	IOCIE	TMROIF	INTF	IOCIF	102
PMCON1	— ⁽¹⁾	CFGs	LWLO	FREE	WRERR	WREN	写	RD	134
PMCON2	程序存储器控制寄存器 2								135
PMADRL	PMADRL<7:0>								132
PMADRH	— ⁽¹⁾	PMADRH<6:0>							
PMDATL	PMDATL<7:0>								132
PMDATH	—	—	PMDATH<5:0>						132

图注: — = 未实现位, 读为 0。闪存程序存储器不使用阴影单元。

注 1: 未实现位, 读为 1。

表 10-4: 与闪存程序存储器相关的配置字汇总

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	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	\overline{CP}	MCLRE	\overline{PWRT}	WDTE<1:0>		FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	\overline{LPBOR}	BORV	STVREN	PLLEN	67
	7:0	ZCD	—	—	—	—	PPS1WAY	WRT<1:0>		

图注: — = 未实现位, 读为 0。闪存程序存储器不使用阴影单元。

11.0 I/O 端口

每个端口都有用于控制其操作的 6 个标准寄存器。这些寄存器是：

- TRIS_x 寄存器（数据方向）
- PORT_x 寄存器（读取器件引脚的电平）
- LAT_x 寄存器（输出锁存器）
- INLVL_x（输入电平控制）
- ODCON_x 寄存器（漏极开路）
- SLRCON_x 寄存器（压摆率）

一些端口可能还具有以下一个或多个额外的寄存器。这些寄存器是：

- ANSEL_x（模拟选择）
- WPU_x（弱上拉）

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。

表 11-1：每款器件可用的端口

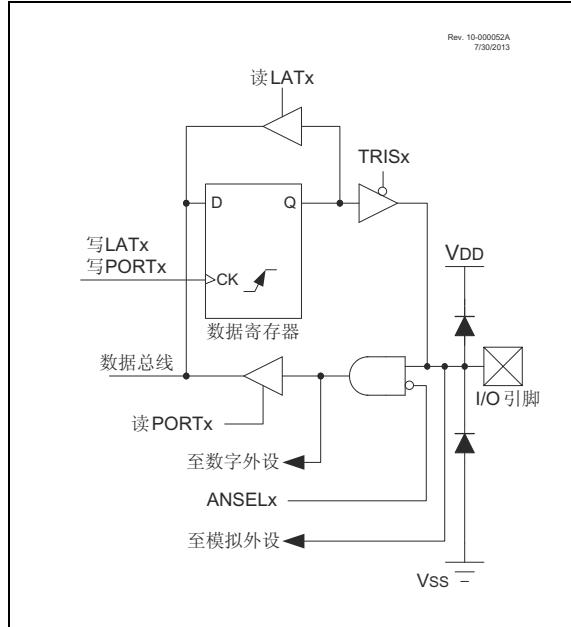
器件	PORTA	PORTB	PORTC
PIC16(L)F1764	•		•
PIC16(L)F1765	•		•
PIC16(L)F1768	•	•	•
PIC16(L)F1769	•	•	•

数据锁存器（LAT_x 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 LAT_x 寄存器的写操作与写入相应 PORT_x 寄存器的效果相同。读取 LAT_x 寄存器时，将会读取 I/O 端口锁存器中保存的值，而读取 PORT_x 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 ANSEL_x 寄存器。当某个 ANSEL 位置 1 时，与该位相关的数字输入缓冲器会被禁止。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平在逻辑输入电路中产生过大的电流。图 11-1 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 11-1：通用 I/O 端口的工作原理



11.1 PORTA 寄存器

11.1.1 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 11-8](#)）。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 11-1](#) 显示了如何初始化 PORTA。

读 PORTA 寄存器（[寄存器 11-1](#)）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总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器（LATA）。

11.1.2 方向控制

TRISA 寄存器（[寄存器 11-2](#)）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。

11.1.3 漏极开路控制

ODCONA 寄存器（[寄存器 11-6](#)）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。

11.1.4 压摆率控制

SLRCONA 寄存器（[寄存器 11-7](#)）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。

11.1.5 输入阈值控制

INVLVA 寄存器（[寄存器 11-8](#)）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）。关于电压阈值的更多信息，请参见[表 36-4](#)。

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。

11.1.6 模拟控制

ANSEL A 寄存器（[寄存器 11-4](#)）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确工作。

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。

例 11-1： 初始化 PORTA

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

BANKSEL PORTA           ;
CLRF    PORTA            ; Init PORTA
BANKSEL LATA             ; Data Latch
CLRF    LATA              ;
BANKSEL ANSEL A          ;
CLRF    ANSEL A           ; digital I/O
BANKSEL TRISA             ;
MOVlw   B'00111000'        ; Set RA<5:3> as inputs
MOVwf   TRISA             ; and set RA<2:0> as
                           ; outputs
```

11.1.7 PORTA 功能和输出优先级

每个 PORTA 引脚都与其他功能复用。

在发生复位之后，每个引脚均默认设为 PORT 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 12.0 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 ANSELA 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。

11.2 寄存器定义：PORTA

寄存器 11-1： PORTA： PORTA 寄存器

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

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 $\geq V_{IH}$

0 = 端口引脚电压 $\leq V_{IL}$

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。

寄存器 11-2： 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	— ⁽¹⁾	TRISA2	TRISA1	TRISA0
bit 7							

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

注 1： 未实现位，读为 1。

PIC16(L)F1764/5/8/9

寄存器 11-3: 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>: RA<5:4>** 输出锁存值位 ⁽¹⁾

bit 3 未实现: 读为 0

bit 2-0 **LATA<2:0>: RA<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 11-4: ANSELA: PORTA 模拟选择寄存器

U-0	U-0	U-0	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	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 7-5 未实现: 读为 0

bit 4 **ANSA<4>: 将 RA4 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 11-5: WPUA: 弱上拉 PORTA 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
—	—	WPUA5	WPUA4	WPUA3	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 (1,2)

1 = 使能上拉

0 = 禁止上拉

注 1: 必须清零 OPTION_REG 寄存器的全局 **WPUEN** 位, 从而使能各个上拉功能。

2: 如果引脚被配置为输出, 则自动禁止弱上拉器件。

寄存器 11-6: 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
—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0
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 **ODA<5:4>**: PORTA 漏极开路使能位

对于 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 **ODA<2:0>**: PORTA 漏极开路使能位

对于 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)F1764/5/8/9

寄存器 11-7: 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 RA<5:4> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率将为最大值

bit 3 未实现: 读为 0

bit 2-0 **SLRA<2:0>**: PORTA 压摆率使能位

对于 RA<2:0> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率将为最大值

寄存器 11-8: INLVLA: PORTA 输入电平控制寄存器

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
—	—	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 RA<5:0> 引脚

1 = 端口引脚数字输入使用 ST 阈值工作

0 = 端口引脚数字输入使用 TTL 阈值工作

表 11-2: 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
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	142
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	140
ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	141
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			219
POR TA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	139
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	142
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	141

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

注 1: 未实现位, 读为 1。

表 11-3: 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
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>			FOSC<2:0>		

图注: — = 未实现位, 读为 0。PORTA 不使用阴影单元。

11.3 PORTB 寄存器 (仅限 PIC16(L)F1768/9)

11.3.1 数据寄存器

PORTB 是一个 4 位宽的双向端口。对应的数据方向寄存器是 TRISB (寄存器 11-10)。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 11-1 显示了如何初始化 I/O 端口。

读 PORTB 寄存器 (寄存器 11-9) 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)。

11.3.2 方向控制

TRISB 寄存器 (寄存器 11-10) 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。

11.3.3 漏极开路控制

ODCONB 寄存器 (寄存器 11-14) 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。

11.3.4 压摆率控制

SLRCONB 寄存器 (寄存器 11-15) 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。

11.3.5 输入阈值控制

INLVLB 寄存器 (寄存器 11-16) 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)。关于电压阈值的更多信息，请参见表 36-4。

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。

11.3.6 模拟控制

ANSELB 寄存器 (寄存器 11-12) 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确工作。

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。

11.3.7 PORTB 功能和输出优先级

在发生复位之后，每个引脚均默认设为 PORT 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 12.0 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 和运放输入。这些输入在使用 ANSELB 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。

11.4 寄存器定义: PORTB

寄存器 11-9: PORTB: PORTB 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
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 通用 I/O 引脚位⁽¹⁾

1 = 端口引脚电平 $\geq V_{IH}$

0 = 端口引脚电平 $\leq V_{IL}$

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 11-10: 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 三态控制位

1 = PORTB 引脚被配置为输入 (三态)

0 = PORTB 引脚被配置为输出

bit 3-0 未实现: 读为 0

PIC16(L)F1764/5/8/9

寄存器 11-11: 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>**: PORTB 输出锁存值位 ⁽¹⁾

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 11-12: 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>**: 将 RB<7:4> 引脚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: 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 11-13: WPUB: 弱上拉 PORTB 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
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 = 使能上拉

0 = 禁止上拉

bit 3-0 未实现: 读为 0

注 1: 必须清零 OPTION_REG 寄存器的全局 WPUEN 位, 从而使能各个上拉功能。

2: 如果引脚被配置为输出, 则自动禁止弱上拉器件。

寄存器 11-14: 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
ODB7	ODB6	ODB5	ODB4	—	—	—	—
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 **ODB<7:4>**: PORTB 漏极开路使能位

对于 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)F1764/5/8/9

寄存器 11-15: 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>: PORTB 压摆率使能位**

对于 RB<7:4> 引脚

1 = 端口引脚的压摆率受到限制
0 = 端口引脚的压摆率将为最大值

bit 3-0 未实现: 读为 0

寄存器 11-16: INLVLB: PORTB 输入电平控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
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>: PORTB 输入电平选择位**

对于 RB<7:4> 引脚

1 = 端口引脚数字输入使用 ST 阈值工作
0 = 端口引脚数字输入使用 TTL 阈值工作

bit 3-0 未实现: 读为 0

表 11-4: 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
ANSELB	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	148
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	146
ODCONB	ODB7	ODB6	ODB5	ODB4	—	—	—	—	147
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	145
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	148
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	148
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	147

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

11.5 PORTC 寄存器

11.5.1 数据寄存器

在 PIC16(L)F1764/5 器件中，PORTC 是 6 位宽的双向端口；在 PIC16(L)F1768/9 器件中，它是 8 位宽的双向端口。对应的数据方向寄存器是 TRISC（[寄存器 11-18](#)）。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 11-1](#) 显示了如何初始化 I/O 端口。读 PORTC 寄存器（[寄存器 11-17](#)）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，最再写入该端口的数据锁存器（LATC）。

11.5.2 方向控制

TRISC 寄存器（[寄存器 11-18](#)）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。

11.5.3 输入阈值控制

INLVLC 寄存器（[寄存器 11-24](#)）用于控制每个可用 PORTC 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 PORTC 寄存器的读取值很重要，同时它也是发生电平变化中断（如果使能该功能）的电压。关于电压阈值的更多信息，请参见表 36-4。

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。

11.5.4 漏极开路控制

ODCONC 寄存器（[寄存器 11-22](#)）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。

11.5.5 压摆率控制

SLRCONC 寄存器（[寄存器 11-23](#)）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。

11.5.6 模拟控制

ANSELC 寄存器（[寄存器 11-20](#)）用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSELC 位设置为高电平将使引脚上的所有数字读操作都读为 0，并允许引脚上的模拟功能正确工作。

ANSELC 位的状态不会影响数字输出功能。TRIS 清零且 ANSELC 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，ANSELC 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。

11.5.7 PORTC 功能和输出优先级

在发生复位之后，每个引脚均默认设为 PORT 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 12.0 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 ANSELC 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。

11.5.8 高电流驱动控制

RC4 和 RC5 上的输出驱动器最高能够拉出和灌入 100 mA 的电流。这种额外的驱动能力可以通过 HIDRVC 寄存器（[寄存器 11-25](#)）中的控制位使能和禁止。

PIC16(L)F1764/5/8/9

11.6 寄存器定义: PORTC

寄存器 11-17: 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 ⁽²⁾	RC6 ⁽²⁾	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 7-0 **RC<7:0>**: PORTC 通用 I/O 引脚位 ^(1, 2)

1 = 端口引脚电平 $\geq V_{IH}$

0 = 端口引脚电平 $\leq V_{IL}$

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

2: RC<7:6> 仅在 PIC16(L)F1768/9 上可用。

寄存器 11-18: 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 7-0 **TRISC<7:0>**: PORTC 三态控制位 ⁽¹⁾

1 = PORTC 引脚被配置为输入 (三态)

0 = PORTC 引脚被配置为输出

注 1: TRISC<7:6> 仅在 PIC16(L)F1768/9 上可用。

寄存器 11-19: 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 7-0 **LATC<7:0>: PORTC 输出锁存值位⁽¹⁾**

注 1: LATC<7:6> 仅在 PIC16(L)F1768/9 上可用。

寄存器 11-20: ANSELC: PORTC 模拟选择寄存器

R/W-1/1	R/W-1/1	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	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> 引脚选择为模拟或数字功能^(1,2)**

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 5-4 未实现: 读为 0

bit 3-0 **ANSC<3:0>: 将 RC<3:0> 引脚选择为模拟或数字功能⁽¹⁾**

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。

2: ANSC<7:6> 仅在 PIC16(L)F1768/9 上可用。

PIC16(L)F1764/5/8/9

寄存器 11-21: WPUC: 弱上拉 PORTC 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
WPUC7 ⁽³⁾	WPUC6 ⁽³⁾	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 7-0 **WPUC<7:0>**: 弱上拉寄存器位^(1,2)

1 = 使能上拉

0 = 禁止上拉

注 1: 必须清零 OPTION_REG 寄存器的全局 WPUEN 位, 从而使能各个上拉功能。

2: 如果引脚被配置为输出, 则自动禁止弱上拉器件。

3: WPUC<7:6> 仅在 PIC16(L)F1768/9 上可用。

寄存器 11-22: ODCCONC: 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
ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0
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 **ODC<7:0>**: PORTC 漏极开路使能位⁽¹⁾

对于 RC<7:0> 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: ODC<7:6> 仅在 PIC16(L)F1768/9 上可用。

寄存器 11-23: 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 7-0 **SLRC<7:0>: PORTC 压摆率使能位⁽¹⁾**

对于 RC<7:0> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率将为最大值

注 1: SLRC<7:6> 仅在 PIC16(L)F1768/9 上可用。

寄存器 11-24: 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 7-0 **INLVLC<7:0>: PORTC 输入电平选择位⁽¹⁾**

对于 RC<7:0> 引脚

1 = 端口引脚数字输入使用 ST 阈值工作

0 = 端口引脚数字输入使用 TTL 阈值工作

注 1: INLVLC<7:6> 仅在 PIC16(L)F1768/9 上可用。

PIC16(L)F1764/5/8/9

寄存器 11-25: HIDRVC: PORTC 高驱动控制寄存器

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	
—	—	HIDC5	HIDC4	—	—	—	—	
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 **HIDC<5:4>**: PORTC 高驱动使能位

对于 RC<5:4> 引脚

1 = 使能高拉电流和灌电流

0 = 标准拉电流和灌电流

bit 3-0 未实现: 读为 0

表 11-5：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
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
HIDRVC	—	—	HIDC5	HIDC4	—	—	—	—	154
INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	153
LATC	LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	151
ODCONC	ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	152
PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	150
SLRCONC	SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	153
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	152

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

注 1：仅限 PIC16(L)F1768/9。

12.0 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 12-1 中的简化框图所示。

12.1 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。

多个外设可以同时使用同一输入源工作。端口读操作总是返回引脚电平，无论外设 PPS 选择如何。如果引脚还具备关联模拟功能，则必须清零该引脚的 ANSEL 位才会使能数字输入缓冲器。

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 12-1 所示。

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

12.2 PPS 输出

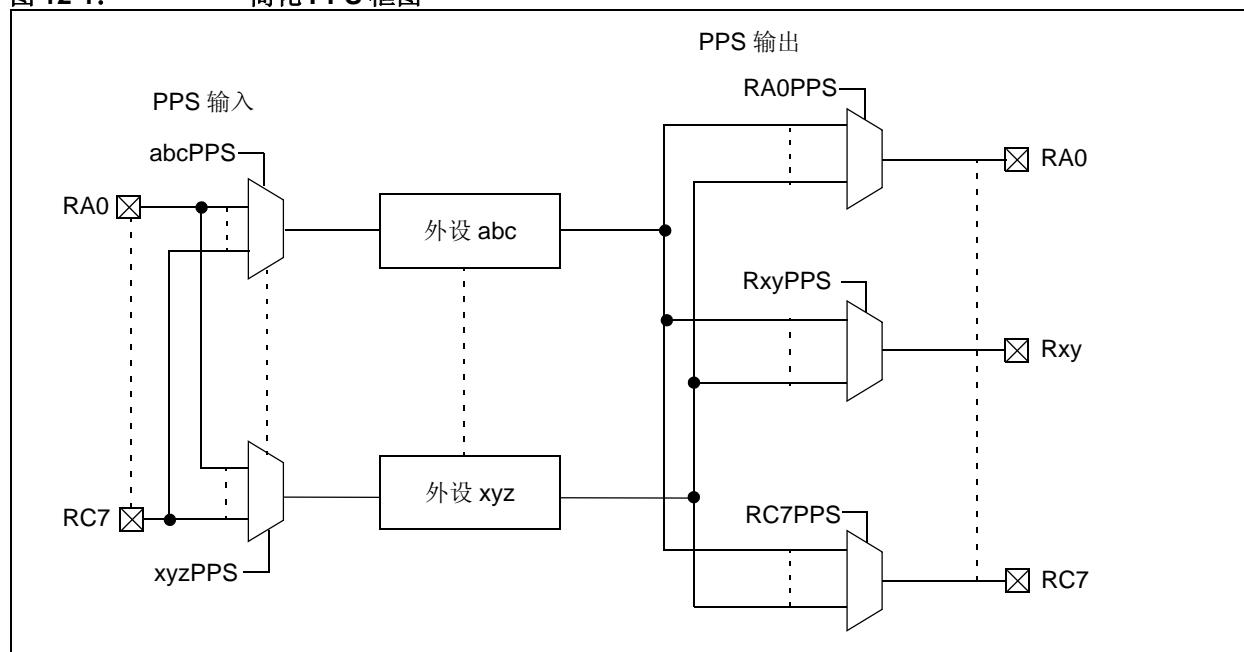
每个 I/O 引脚均具有一个用于选择引脚输出源的 PPS 寄存器。除了少数例外情况，与该引脚相关的端口 TRIS 控制将保持对引脚输出驱动器的控制权。作为外设操作的一部分，控制引脚输出驱动器的外设将根据需要改写 TRIS 控制。这些外设包括：

- EUSART (同步操作)
- MSSP (I^2C)
- COG (自动关断)

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 12-2 所示。

注： “**R_{xy}**”符号是引脚标识符的占位符。例如，RA0PPS。

图 12-1： 简化 PPS 框图



12.3 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：

- USART（同步操作）
- MSSP（I²C）

注： I²C 默认输入引脚与 I²C 和 SMBus 兼容，并且是器件上唯一具有这种兼容性的引脚。

12.4 PPS 锁定

PPS 包含了一种锁定模式，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 12-1 给出了置 1 和清零 PPSLOCKED 位的示例。

例 12-1: 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
```

12.5 PPS 永久锁定

通过将 PPS1WAY 配置位置 1，可以永久锁定 PPS。当该位置 1 时，PPSLOCKED 位只能在器件复位之后清零和置 1 一次。这使得可以清零 PPSLOCKED 位，从而可以在初始化过程中进行输入和输出选择。在进行所有选择之后将 PPSLOCKED 置位 1 时，它将一直保持置 1，直到下一个器件复位事件之后才能清零。

12.6 休眠期间的操作

PPS 输入和输出选择不会受休眠影响。

12.7 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。所有其他复位会将选择保留不变。[表 12-1](#) 列出了默认的输入选择。

PIC16(L)F1764/5/8/9

12.8 寄存器定义：PPS 输入选择

寄存器 12-1： 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 4-3 xxxPPS<4:3>：外设 xxx 输入端口选择位

11 = 保留。不要使用。

10 = 外设输入为 PORTC

01 = 外设输入为 PORTB⁽²⁾

00 = 外设输入为 PORTA

bit 2-0 xxxPPS<2:0>：外设 xxx 输入位选择位⁽¹⁾

111 = 外设输入来自 PORTx Bit 7 (Rx7)

110 = 外设输入来自 PORTx Bit 6 (Rx6)

101 = 外设输入来自 PORTx Bit 5 (Rx5)

100 = 外设输入来自 PORTx Bit 4 (Rx4)

011 = 外设输入来自 PORTx Bit 3 (Rx3)

010 = 外设输入来自 PORTx Bit 2 (Rx2)

001 = 外设输入来自 PORTx Bit 1 (Rx1)

000 = 外设输入来自 PORTx Bit 0 (Rx0)

注 1：关于 xxxPPS 寄存器列表和复位值，请参见表 12-1。

2：仅限 PIC16(L)F1768/9。

寄存器 12-2： 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

选择代码决定端口引脚上的输出信号。

关于选择代码，请参见表 12-2。

寄存器 12-3: PPSLOCK: PPS 锁定寄存器

U-0	R/W-0/0						
—	—	—	—	—	—	—	PPSLOCKED 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。

PIC16(L)F1764/5/8/9

表 12-1: PPS 输入寄存器复位值

外设	xxxPPS 寄存器	默认引脚选择		复位值 (xxxPPS<4:0>)	
		PIC16(L)F1768/9	PIC16(L)F1764/5	PIC16(L)F1768/9	PIC16(L)F1764/5
电平变化中断	INTPPS	RA2	RA2	00010	00010
Timer0 时钟	T0CKIPPS	RA2	RA2	00010	00010
Timer1 时钟	T1CKIPPS	RA5	RA5	00101	00101
Timer1 门控	T1GPPS	RA4	RA4	00100	00100
Timer2 时钟	T2CKIPPS	RA5	RA5	0101	0101
Timer3 时钟	T3CKIPPS	RC5	RC5	10101	10101
Timer3 门控	T3GPPS	RC4	RC4	10100	10100
Timer4 时钟	T4CKIPPS	RC1	RC1	10001	10001
Timer5 时钟	T5CKIPPS	RC0	RC0	10000	10000
Timer5 门控	T5GPPS	RC3	RC3	10011	10011
Timer6 时钟	T6CKIPPS	RA3	RA3	00011	00011
CCP1	CCP1PPS	RC5	RC5	10101	10101
CCP2	CCP2PPS ⁽¹⁾	RC3	—	10011	—
COG1	COG1INPPS	RA2	RA2	00010	00010
COG2	COG2INPPS ⁽¹⁾	RA2	—	00010	—
SPI 和 I ² C 时钟	SSPCLKPPS	RB6	RC0	01110	10000
SPI 和 I ² C 数据	SSPDATPPS	RB4	RC1	01100	10001
SPI 从选择	SSPSSPPS	RC6	RC3	10110	10011
EUSART RX	RXPPS	RB5	RC5	01101	10101
EUSART CK	CKPPS	RB7	RC4	01111	10100
所有 CLC	CLCIN0PPS	RC3	RC3	10011	10011
所有 CLC	CLCIN1PPS	RC4	RC4	10100	10100
所有 CLC	CLCIN2PPS	RC1	RC1	10001	10001
所有 CLC	CLCIN3PPS	RA5	RA5	00101	00101
PRG1 置为上升	PRG1RPPS	RC4	RC4	10100	10100
PRG1 置为下降	PRG1FPPS	RC5	RC5	10101	10101
PRG2 置为上升	PRG2RPPS ⁽¹⁾	RC4	—	10100	—
PRG2 置为下降	PRG2FPPS ⁽¹⁾	RC5	—	10101	—
DSM1 高载波信号	MD1CHPPS	RA3	RA3	00011	00011
DSM1 低载波信号	MD1CLPPS	RA4	RA4	00100	00100
DSM1 调制	MD1MODPPS	RA5	RA5	00101	00101
DSM2 高载波信号	MD2CHPPS ⁽¹⁾	RA3	—	00011	—
DSM2 低载波信号	MD2CLPPS ⁽¹⁾	RA4	—	00100	—
DSM2 调制	MD2MODPPS ⁽¹⁾	RA5	—	00101	—

例: CCP1PPS = 0x13 会选择 RC3 作为 CCP1 输入。

注 1: 仅限 PIC16(L)F1768/9。

表 12-2: 外设输出的可用端口 (2)

RxyPPS<4:0>	输出信号	PIC16(L)F1768/9			PIC16(L)F1764/5	
		PORTA	PORTB	PORTC	PORTA	PORTC
1111x	保留	—	—	—	—	—
11101	MD2_out	•	•	•	—	—
11100	MD1_out	•	•	•	•	•
11011	sync_C4OUT	•	•	•	—	—
11010	sync_C3OUT	•	•	•	—	—
11001	sync_C2OUT	•	•	•	•	•
11000	sync_C1OUT	•	•	•	•	•
10111	DT ⁽¹⁾	•	•	•	•	•
10110	TX/CK ⁽¹⁾	•	•	•	•	•
10101	保留	—	—	—	—	—
10100	SDO	•	•	•	•	•
10011	SDA	•	•	•	•	•
10010	SCK/SCL ⁽¹⁾	•	•	•	•	•
10001	PWM6_out	•	•	•	—	—
10000	PWM5_out	•	•	•	•	•
01111	PWM4_out	•	•	•	—	—
01110	PWM3_out	•	•	•	•	•
01101	CCP2_out	•	•	•	•	•
01100	CCP1_out	•	•	•	•	•
01011	COG2D ⁽¹⁾	•	•	•	—	—
01010	COG2C ⁽¹⁾	•	•	•	—	—
01001	COG2B ⁽¹⁾	•	•	•	—	—
01000	COG2A ⁽¹⁾	•	•	•	—	—
00111	COG1D ⁽¹⁾	•	•	•	•	•
00110	COG1C ⁽¹⁾	•	•	•	•	•
00101	COG1B ⁽¹⁾	•	•	•	•	•
00100	COG1A ⁽¹⁾	•	•	•	•	•
00011	LC3_out	•	•	•	•	•
00010	LC2_out	•	•	•	•	•
00001	LC1_out	•	•	•	•	•
00000	LATxy	•	•	•	•	•

注 1: TRIS 控制由外设根据需要改写。

2: 不受支持的外设将输出 0。

PIC16(L)F1764/5/8/9

表 12-3：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	159
INTPPS	—	—	—		INTPPS<4:0>				158
T0CKIPPS	—	—	—		T0CKIPPS<4:0>				158
T1CKIPPS	—	—	—		T1CKIPPS<4:0>				158
T1GPPS	—	—	—		T1GPPS<4:0>				158
T2CKIPPS	—	—	—		T2CKIPPS<4:0>				158
T3CKIPPS	—	—	—		T3CKIPPS<4:0>				158
T3GPPS	—	—	—		T3GPPS<4:0>				158
T4CKIPPS	—	—	—		T4CKIPPS<4:0>				158
T5CKIPPS	—	—	—		T5CKIPPS<4:0>				158
T5GPPS	—	—	—		T5GPPS<4:0>				158
T6CKIPPS	—	—	—		T6CKIPPS<4:0>				158
CCP1PPS	—	—	—		CCP1PPS<4:0>				158
CCP2PPS ⁽¹⁾	—	—	—		CCP2PPS<4:0>				158
COG1INPPS	—	—	—		COG1INPPS<4:0>				158
COG2INPPS ⁽¹⁾	—	—	—		COG2INPPS<4:0>				158
SSPCLKPPS	—	—	—		SSPCLKPPS<4:0>				158
SSPDATPPS	—	—	—		SSPDATPPS<4:0>				158
SSPSSPPS	—	—	—		SSPSSPPS<4:0>				158
RXPPS	—	—	—		RXPPS<4:0>				158
CKPPS	—	—	—		CKPPS<4:0>				158
CLCIN0PPS	—	—	—		CLCIN0PPS<4:0>				158
CLCIN1PPS	—	—	—		CLCIN1PPS<4:0>				158
CLCIN2PPS	—	—	—		CLCIN2PPS<4:0>				158
CLCIN3PPS	—	—	—		CLCIN3PPS<4:0>				158
PRG1RPPS	—	—	—		PRG1RPPS<4:0>				158
PRG1FPPS	—	—	—		PRG1FPPS<4:0>				158
PRG2RPPS ⁽¹⁾	—	—	—		PRG2RPPS<4:0>				158
PRG2FPPS ⁽¹⁾	—	—	—		PRG2FPPS<4:0>				158
MD1CHPPS	—	—	—		MD1CHPPS<4:0>				158
MD1CLPPS	—	—	—		MD1CLPPS<4:0>				158
MD1MODPPS	—	—	—		MD1MODPPS<4:0>				158
MD2CHPPS ⁽¹⁾	—	—	—		MD2CHPPS<4:0>				158
MD2CLPPS ⁽¹⁾	—	—	—		MD2CLPPS<4:0>				158
MD2MODPPS ⁽¹⁾	—	—	—		MD2MODPPS<4:0>				158
RA0PPS	—	—	—		RA0PPS<4:0>				158
RA1PPS	—	—	—		RA1PPS<4:0>				158
RA2PPS	—	—	—		RA2PPS<4:0>				158
RA4PPS	—	—	—		RA4PPS<4:0>				158
RA5PPS	—	—	—		RA5PPS<4:0>				158
RB4PPS ⁽¹⁾	—	—	—		RB4PPS<4:0>				158
RB5PPS ⁽¹⁾	—	—	—		RB5PPS<4:0>				158

表 12-3：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
RB6PPS ⁽¹⁾	—	—	—	RB6PPS<4:0>					158
RB7PPS ⁽¹⁾	—	—	—	RB7PPS<4:0>					158
RC0PPS	—	—	—	RC0PPS<4:0>					158
RC1PPS	—	—	—	RC1PPS<4:0>					158
RC2PPS	—	—	—	RC2PPS<4:0>					158
RC3PPS	—	—	—	RC3PPS<4:0>					158
RC4PPS	—	—	—	RC4PPS<4:0>					158
RC5PPS	—	—	—	RC5PPS<4:0>					158
RC6PPS ⁽¹⁾	—	—	—	RC6PPS<4:0>					158
RC7PPS ⁽¹⁾	—	—	—	RC7PPS<4:0>					158

图注：— = 未实现位，读为 0。PPS 模块不使用阴影单元。

注 1：仅限 PIC16(L)F1768/9。

13.0 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。中断可以通过检测具有上升沿或下降沿的信号而产生。任意一个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（主开关）
- 独立的引脚配置
- 上升沿和下降沿检测
- 独立的引脚中断标志

图 13-1 给出了 IOC 模块的框图。

13.1 使能模块

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

13.2 独立的引脚配置

对于每个引脚，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 IOCN 寄存器的相关位置 1。

通过同时将 IOCXP 和 IOCN 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。

13.3 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 IOCE 位也置 1，则还会产生中断。INTCON 寄存器的 IOCIF 位会反映所有 IOCxF 位的状态。

13.4 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，关联的状态标志都会在序列结束时置 1。

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

例 13-1：清零中断标志 (以 PORTA 为例)

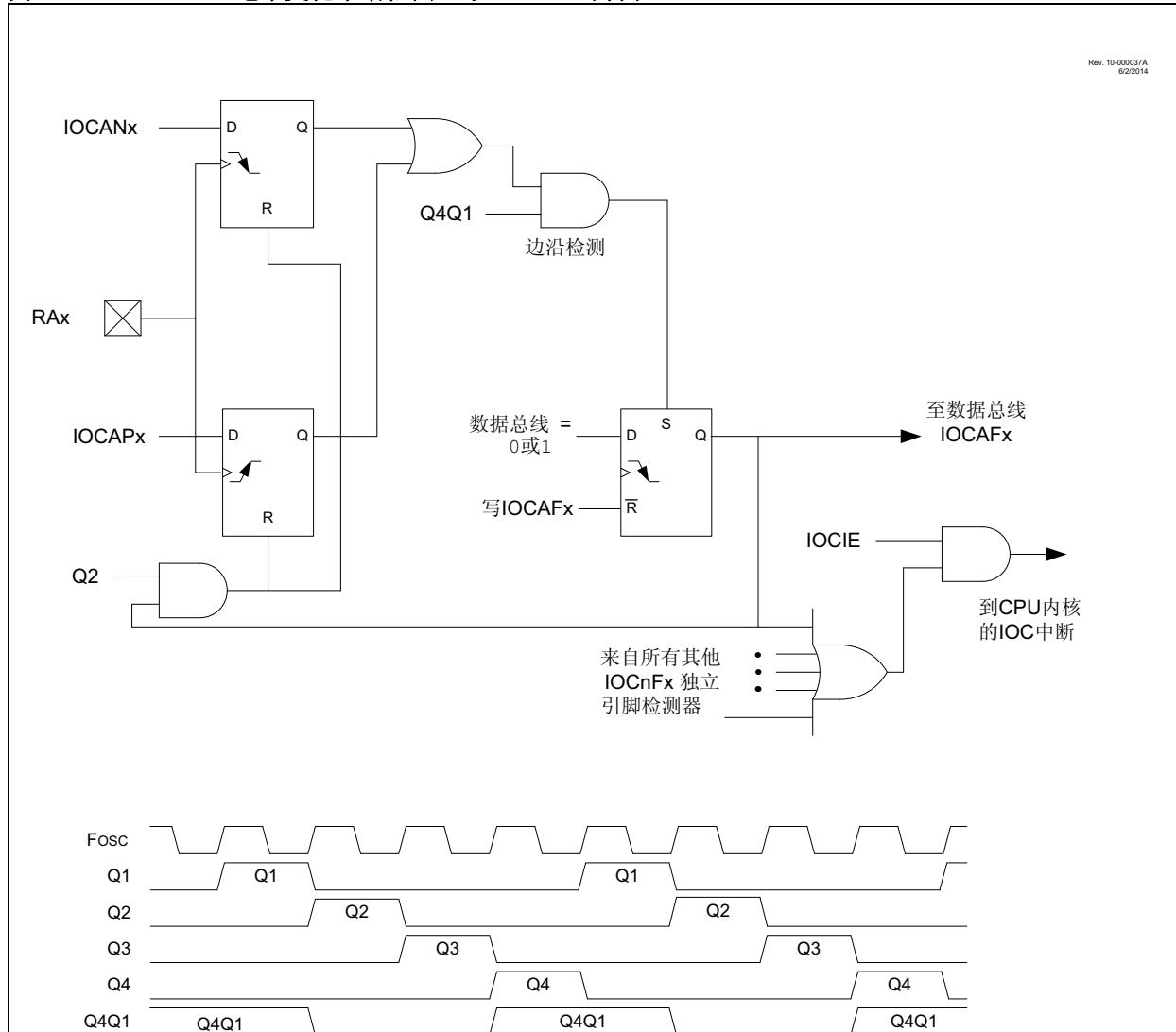
```
MOVLW 0xff
XORWF IOCAF, W
ANDWF IOCAF, F
```

13.5 休眠模式下的操作

如果 IOCE 位置 1，电平变化中断序列会将器件从休眠模式唤醒。

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

图 13-1：电平变化中断框图（以 PORTA 为例）



PIC16(L)F1764/5/8/9

13.6 寄存器定义：电平变化中断控制

寄存器 13-1: **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 正边沿使能位

1 = 允许由引脚电平的正边沿触发电平变化中断。 IOCAF_x 位和 IOCIF 标志将在检测到边沿时置 1。
0 = 禁止相应引脚的电平变化中断。

寄存器 13-2: **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 负边沿使能位

1 = 允许由引脚电平的负边沿触发电平变化中断。 IOCAF_x 位和 IOCIF 标志将在检测到边沿时置 1。
0 = 禁止相应引脚的电平变化中断。

寄存器 13-3: 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 = 在相应引脚上检测到有效的电平变化。

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 13-4: 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 正边沿使能位

1 = 允许由引脚电平的正边沿触发电平变化中断。 IOCBFx 位和 IOCIF 标志将在检测到边沿时置 1。

0 = 禁止相应引脚的电平变化中断。

bit 3-0 未实现: 读为 0

注 1: 仅限 PIC16(L)F1768/9。

PIC16(L)F1764/5/8/9

寄存器 13-5: **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 负边沿使能位

1 = 允许由引脚电平的负边沿触发电平变化中断。 IOCBFx 位和 IOCIF 标志将在检测到边沿时置 1。
0 = 禁止相应引脚的电平变化中断。

bit 3-0

未实现: 读为 0

注 1: 仅限 PIC16(L)F1768/9。

寄存器 13-6: **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 = 在相应引脚上检测到有效的电平变化。
在 IOCBPx = 1, 并在 RBx 上检测到上升沿时置 1, 或者在 IOCBNx = 1, 并在 RBx 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)F1768/9。

寄存器 13-7: 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 7-0

IOCCP<7:0>: 电平变化中断 PORTC 正边沿使能位

1 = 允许由引脚电平的正边沿触发电平变化中断。 IOCCFx 位和 IOCIF 标志将在检测到边沿时置 1。

0 = 禁止相应引脚的电平变化中断。

注 1: 仅限 PIC16(L)F1768/9。

寄存器 13-8: 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 7-0

IOCCN<7:0>: 电平变化中断 PORTC 负边沿使能位

1 = 允许由引脚电平的负边沿触发电平变化中断。 IOCCFx 位和 IOCIF 标志将在检测到边沿时置 1。

0 = 禁止相应引脚的电平变化中断。

注 1: 仅限 PIC16(L)F1768/9。

寄存器 13-9: 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 7-0

IOCCF<7:0>: 电平变化中断 PORTC 标志位

1 = 在相应引脚上检测到有效的电平变化。

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)F1768/9。

PIC16(L)F1764/5/8/9

表 13-1：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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	167
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	166
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	166
IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	168
IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	168
IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	167
IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	169
IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	169
IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	169
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

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)F1768/9。

2：未实现位，读为 1。

14.0 固定参考电压 (FVR)

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

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

FVR 可以通过将 FVRCON 寄存器的 FVREN 位置 1 来使能。

14.1 独立的增益放大器

送到 ADC、比较器和 DAC 的 FVR 输出会经过两个独立的可编程增益放大器。每个放大器都可以设定为增益为 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 16.0 节“模数转换器 \(ADC\) 模块”](#)。

FVRCON 寄存器的 CDAFVR<1:0> 位用于使能和配置送到 DAC 和比较器模块的参考电压的增益放大器设置。更多信息，请参见[第 17.0 节“5 位数模转换器 \(DAC\) 模块”](#)和[第 19.0 节“比较器模块”](#)。

14.2 FVR 稳定期

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。请参见[图 37-19](#)。

14.3 FVR 缓冲器稳定周期

在使能 FVR 缓冲器 1 或缓冲器 2 时，缓冲器放大电路需要经过 30 μ s 才会稳定。即使 FVR 已在工作且已稳定时，也需要该稳定时间。

PIC16(L)F1764/5/8/9

图 14-1: 参考电压框图

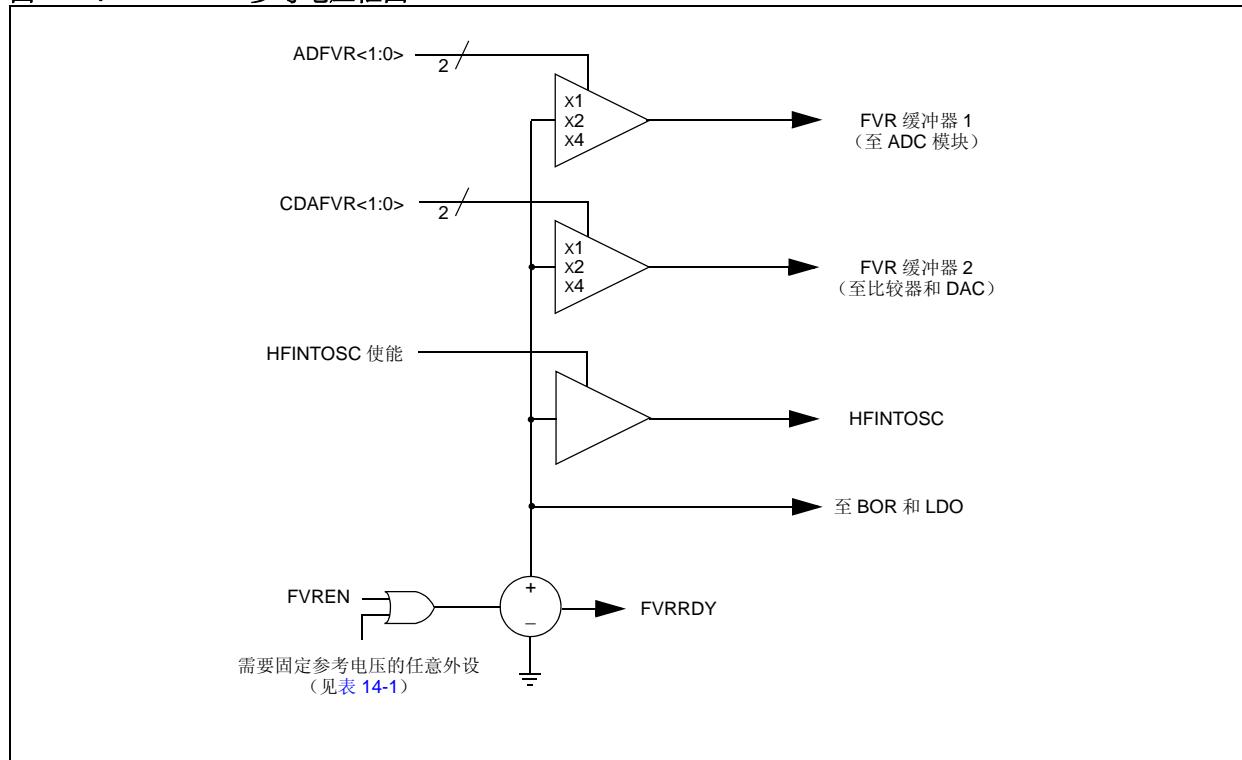


表 14-1: 需要固定参考电压 (FVR) 的外设

外设	条件	说明
HFINTOSC	$\text{FOSC}<2:0> = 100$ 且 $\text{IRCF}<3:0> \neq 000x$	INTOSC 有效且器件不处于休眠模式
BOR	$\text{BOREN}<1:0> = 11$	BOR 总是使能
	$\text{BOREN}<1:0> = 10$ 且 $\text{BORFS} = 1$	BOR 在休眠模式下被禁止, BOR 快速启动使能
	$\text{BOREN}<1:0> = 01$ 且 $\text{BORFS} = 1$	BOR 受软件控制, BOR 快速启动使能
LDO	当 $\text{VREGPM} = 1$ 且不处于休眠模式时, 所有 PIC16F1764/5/8/9 器件	处于休眠模式时, 器件依靠 ULP 稳压器工作

14.4 寄存器定义：FVR 控制

寄存器 14-1： 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
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, 输出 $V_{CDAFVR} = 4x V_{FVR}$ ⁽²⁾
10 = 比较器 FVR 缓冲器增益为 2x, 输出 $V_{CDAFVR} = 2x V_{FVR}$ ⁽²⁾
01 = 比较器 FVR 缓冲器增益为 1x, 输出 $V_{CDAFVR} = 1x V_{FVR}$
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, 输出 $V_{ADFVR} = 4x V_{FVR}$ ⁽²⁾
10 = ADC FVR 缓冲器增益为 2x, 输出 $V_{ADFVR} = 2x V_{FVR}$ ⁽²⁾
01 = ADC FVR 缓冲器增益为 1x, 输出 $V_{ADFVR} = 1x V_{FVR}$
00 = ADC FVR 缓冲器关闭

注 1: 仅对于 PIC16F1764/5/8/9, FVRRDY 总是为 1。

2: 固定参考电压输出不能超出 VDD。

3: 更多信息，请参见第 15.0 节 “温度指示器模块”。

表 14-2： 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>		173

图注： 固定参考电压不使用阴影单元。

15.0 温度指示器模块

本器件系列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 $+85^{\circ}\text{C}$ 之间。其输出是与器件温度成正比的电压。温度指示器的输出在内部与器件 ADC 连接。

电路可以用作温度阈值检测器，也可以用作更精确的温度指示器，这取决于所执行的校准级别。执行单点校准时，电路可以指示邻近该点的温度。执行双点校准时，电路可以更精确地检测整个温度范围。关于校准过程的更多详细信息，请参见应用笔记 AN1333，《内部温度指示器的使用和校准》(DS01333A_CN)。

15.1 电路工作原理

图 15-1 给出了温度电路的简化框图。与温度成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 15-1 描述了温度指示器的输出特性。

公式 15-1: V_{OUT} 范围

$$\text{高电压范围: } V_{\text{OUT}} = V_{\text{DD}} - 4V_t$$

$$\text{低电压范围: } V_{\text{OUT}} = V_{\text{DD}} - 2V_t$$

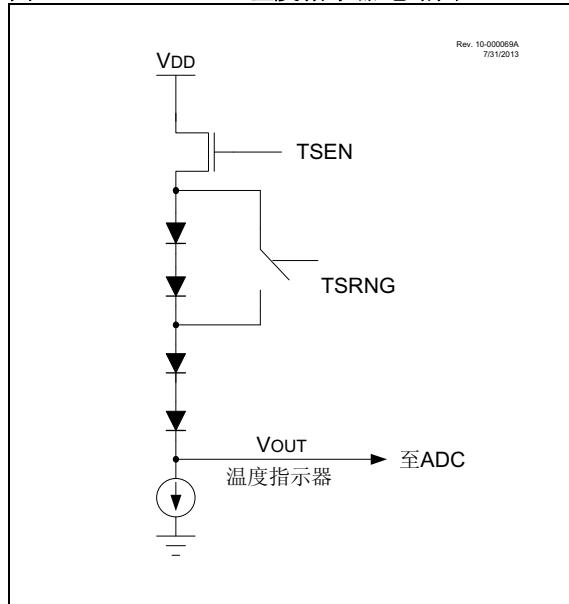
温度检测电路集成了固定参考电压 (FVR) 模块。更多信息，请参见第 14.0 节“固定参考电压 (FVR)”。

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，它可提供较宽的输出电压。这可以在整个温度量程中提供更高的分辨率，但各部件之间的一致性较低。该电压范围需要较高的偏置电压才能工作，所以需要较高的 V_{DD}。

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，所以只需较低的偏置电压就可以让电路工作。低电压范围旨在用于进行低电压操作。

图 15-1: 温度指示器电路图



15.2 最小工作电压 V_{DD}

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 V_{DD} 必须足够高，以确保正确地偏置温度电路。

表 15-1 给出了建议的最小 V_{DD} 与范围设置。

表 15-1: 建议的 V_{DD} 与范围

最小 V _{DD} , TSRNG = 1	最小 V _{DD} , TSRNG = 0
3.6V	1.8V

15.3 温度输出

电路的输出使用内部模数转换器测量。保留一路通道用于温度电路输出。详细信息，请参见第 16.0 节“模数转换器 (ADC) 模块”。

15.4 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 μs ，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 μs 。

表 15-2：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	CDFVR<1:0>	ADFVR<1:0>			173

图注： 温度指示器模块不使用阴影单元。

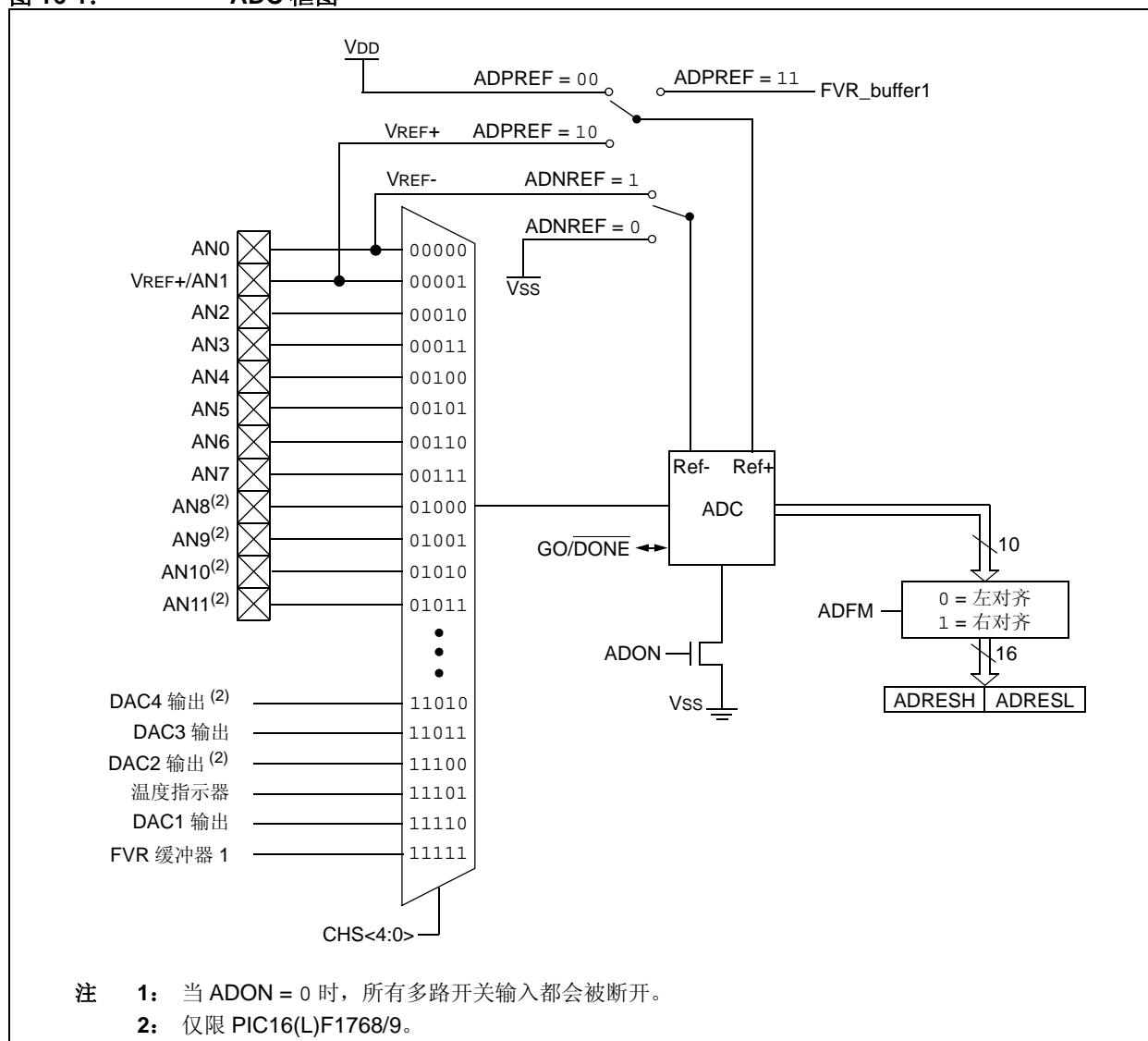
16.0 模数转换器 (ADC) 模块

模数转换器 (ADC) 可将模拟输入信号转换为信号的 10 位二进制表示。该模块使用模拟输入，这些输入通过多路开关连接到同一个采样和保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制结果，并将转换结果存储在 ADC 结果寄存器 (ADRESH:ADRESL 寄存器对) 中。图 16-1 给出了 ADC 的框图。

可通过软件方式选择内部产生的电压或外部提供的电压作为 ADC 参考电压。

ADC 可在转换完成时产生中断。该中断可用于将器件从休眠模式唤醒。

图 16-1: ADC 框图



注 1: 当 ADON = 0 时，所有多路开关输入都会被断开。

2: 仅限 PIC16(L)F1768/9。

16.1 ADC 配置

配置和使用 ADC 时必须考虑以下功能：

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

16.1.1 端口配置

ADC 可用于转换模拟信号和数字信号。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 11.0 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。

16.1.2 通道选择

有最多 18 个通道选择可供使用：

- AN<7:0> 引脚
- AN<11:8> 引脚（仅限 PIC16(L)F1768/9）
- 温度指示器
- DAC1 输出和 DAC3 输出
- DAC2 输出和 DAC4 输出（仅限 PIC16(L)F1768/9）
- FVR 缓冲器 1

ADCON0 寄存器（[寄存器 16-1](#)）的 CHS 位决定与采样保持电路相连接的通道。

当改变通道时，在开始下一次转换前需要一段延时。更多信息，请参见[第 16.2 节 “ADC 工作原理”](#)。

16.1.3 ADC 正参考电压

ADCON1 寄存器的 ADPREF 位用于控制正参考电压。

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）
- Vss

关于固定参考电压的更多详细信息，请参见[第 16.0 节 “模数转换器（ADC）模块”](#)。

16.1.4 ADC 负参考电压

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

16.1.5 转换时钟

可通过软件方式设置 ADCON1 寄存器的 ADCS 位来选择转换时钟源。有以下 7 种时钟频率可供选择：

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- FRC（内部 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 11.5 个 TAD 周期，如[图 16-2](#) 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 36-16](#)。[表 16-1](#) 给出了适当的 ADC 时钟选择的示例。

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

PIC16(L)F1764/5/8/9

表 16-1: 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 μ s
Fosc/4	100	125 ns ⁽²⁾	200 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	1.0 μ s	4.0 μ s
Fosc/8	001	0.5 μ s ⁽²⁾	400 ns ⁽²⁾	0.5 μ s ⁽²⁾	1.0 μ s	2.0 μ s	8.0 μ s ⁽³⁾
Fosc/16	101	800 ns	800 ns	1.0 μ s	2.0 μ s	4.0 μ s	16.0 μ s ⁽³⁾
Fosc/32	010	1.0 μ s	1.6 μ s	2.0 μ s	4.0 μ s	8.0 μ s ⁽³⁾	32.0 μ s ⁽²⁾
Fosc/64	110	2.0 μ s	3.2 μ s	4.0 μ s	8.0 μ s ⁽³⁾	16.0 μ s ⁽²⁾	64.0 μ s ⁽²⁾
FRC	x11	1.0-6.0 μ s ^(1,4)					

图注: 阴影单元表示超出了建议范围。

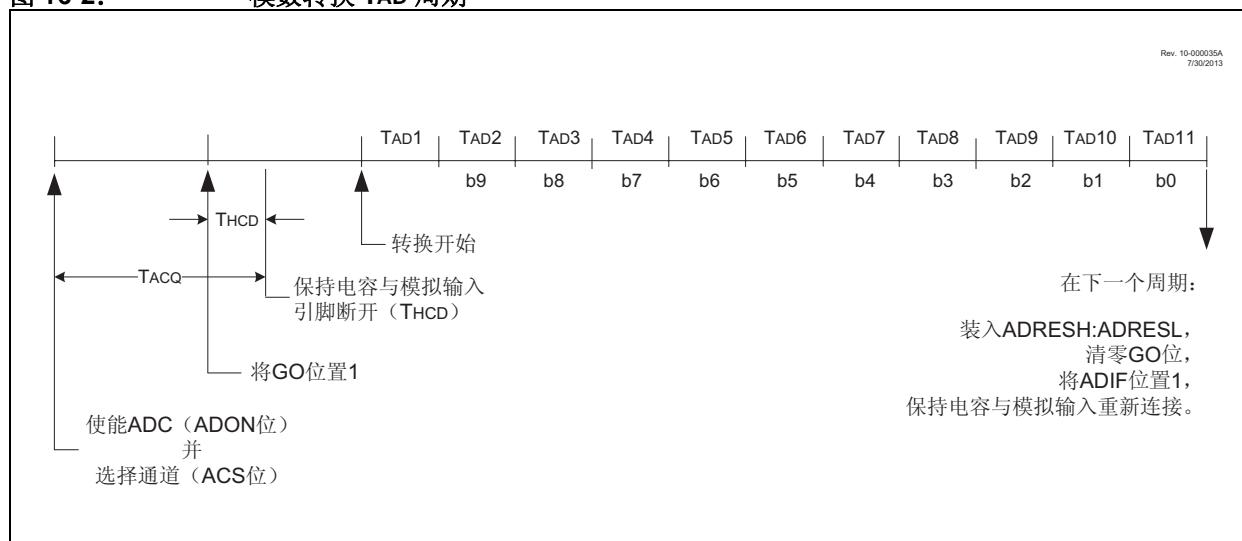
注 1: 关于 FRC 时钟源的典型 TAD 值, 请参见 TAD 参数。

2: 这些值均违反了所需的 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, 可以最大程度降低 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 FRC 振荡器源。

图 16-2: 模数转换 TAD 周期



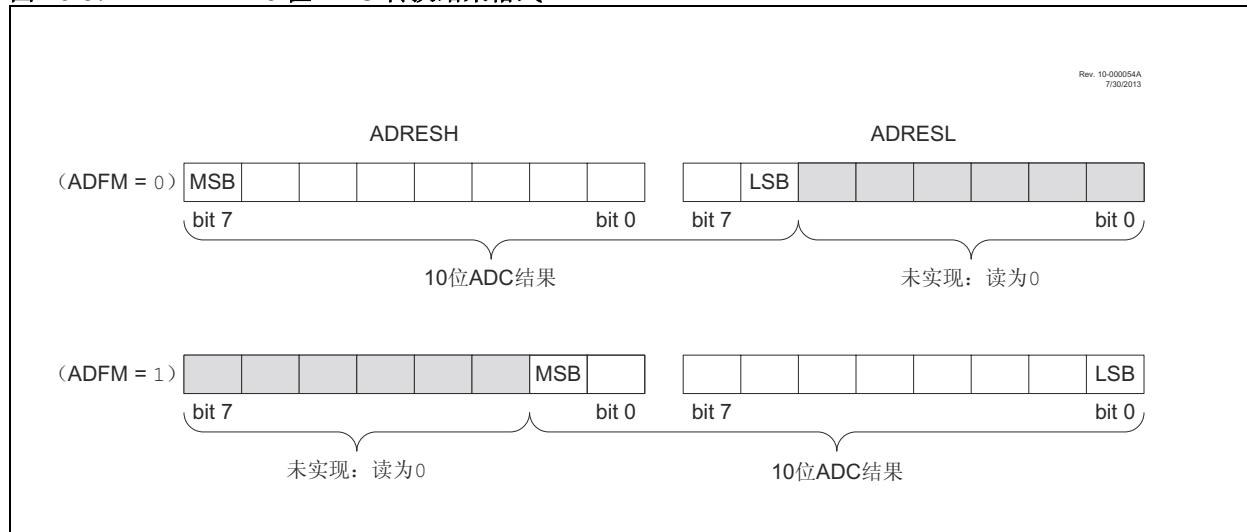
16.1.6 中断

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 FRC 振荡器时，ADC 才能在休眠模式下工作。

器件工作或休眠时都可产生该中断。如果器件处于休眠模式，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。

图 16-3： 10 位 ADC 转换结果格式



16.2 ADC 工作原理

16.2.1 启动转换

要使能 ADC 模块, ADCON0 寄存器的 ADON 位必须设置为 1。将 ADCON0 寄存器的 GO/DONE 位设置为 1 将启动模数转换。

注: 不应在启动 ADC 的同一条指令中将 GO/DONE 位置 1。请参见第 16.2.6 节“ADC 转换步骤”。

16.2.2 转换完成

转换完成时, ADC 模块将:

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

16.2.3 终止转换

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

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

16.2.4 休眠期间的 ADC 操作

ADC 模块可以在休眠模式下工作。这需要将 ADC 时钟源设置为 FRC 选项。当选择 FRC 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 FRC 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。

16.2.5 自动转换触发器

自动转换触发器允许定期进行 ADC 测量而无需软件干预。当出现选定源的上升沿时, GO/DONE 位由硬件置 1。

自动转换触发源使用 ADCON2 寄存器的 TRIGSEL<4:0> 位进行选择。

使用自动转换触发器不能确保正确的 ADC 时序。用户需负责确保满足 ADC 时序要求。

关于自动转换源, 请参见表 16-2。

表 16-2: 自动转换源

源外设	信号名称
CCP1	CCP1_trigger
CCP2 ⁽¹⁾	CCP2_trigger
Timer0	T0_overflow
Timer1	T1_overflow
Timer3	T3_overflow
Timer5	T5_overflow
Timer2	T2_postscaled
Timer4	T4_postscaled
Timer6	T6_postscaled
比较器 C1	sync_C1OUT
比较器 C2	sync_C2OUT
比较器 C3 ⁽¹⁾	sync_C3OUT
比较器 C4 ⁽¹⁾	sync_C4OUT
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
PWM3	PWM3OUT
PWM4 ⁽¹⁾	PWM4OUT
PWM5	PR/PH/OF/DC5_match
PWM6 ⁽¹⁾	PR/PH/OF/DC6_match

注 1: 仅限 PIC16(L)F1768/9。

16.2.6 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
 - 配置参考电压
 - 选择 ADC 输入通道
 - 开启 ADC 模块
3. 配置 ADC 中断（可选）：
 - 清零 ADC 中断标志
 - 允许 ADC 中断
 - 允许外设中断
 - 允许全局中断 **(1)**
4. 等待所需采集时间 **(2)**。
5. 通过将 GO/DONE 位置 1 启动转换。
6. 通过以下方式之一等待 ADC 转换完成：
 - 查询 GO/DONE 位
 - 等待 ADC 中断（已允许中断）
7. 读取 ADC 结果。
8. 清零 ADC 中断标志（如果已允许中断则需要此操作）。

例 16-1：ADC 转换

```
;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
;are included.
;
BANKSEL    ADCON1      ;
MOVLW     B' 11110000' ;Right justify, FRC
;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   ;Acquisition delay
BSF        ADCON0,ADGO ;Start conversion
BTFSR    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
```

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 16.4 节 “ADC 采集要求”**。

PIC16(L)F1764/5/8/9

16.3 寄存器定义：ADC 控制

寄存器 16-1： ADCON0: ADC 控制寄存器 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	R/W-0/0
—			CHS<4:0>		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

未实现：读为 0

bit 6-2

CHS<4:0>: 模拟通道选择位

11111 = FVR (固定参考电压) 缓冲器 1 输出 (2)

11110 = DAC1 输出 (1)

11101 = 温度指示器 (3)

11100 = DAC2 输出 (1,5)

11011 = DAC3 输出 (4)

11010 = DAC4 输出 (4,5)

11001 = 保留。不连接任何通道

•

•

•

01100 = 保留。不连接任何通道

01011 = AN11(5)

01010 = AN10(5)

01001 = AN9(5)

01000 = AN8(5)

00111 = AN7

00110 = AN6

00101 = AN5

00100 = AN4

00011 = AN3

00010 = AN2

00001 = AN1

00000 = AN0

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 17.0 节 “5 位数模转换器（DAC）模块”。

2: 更多信息，请参见第 14.0 节 “固定参考电压（FVR）”。

3: 更多信息，请参见第 15.0 节 “温度指示器模块”。

4: 更多信息，请参见第 18.0 节 “10 位数模转换器（DAC）模块”。

5: 仅限 PIC16(L)F1768/9。

寄存器 16-2: 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 = FRC (由内部 RC 振荡器提供的时钟)
110 = Fosc/64
101 = Fosc/16
100 = Fosc/4
011 = FRC (由内部 RC 振荡器提供的时钟)
010 = Fosc/32
001 = Fosc/8
000 = Fosc/2

bit 3

未实现: 读为 0

bit 2

ADNREF: ADC 负参考电压配置位

1 = VREF- 连接到外部 VREF- 引脚
0 = VREF- 连接到 Vss

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。详情请参见表 36-16。

PIC16(L)F1764/5/8/9

寄存器 16-3: ADCON2: ADC 控制寄存器 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0
		TRIGSEL<4:0> ⁽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 7-3 TRIGSEL<4:0>: 自动转换触发源选择位⁽¹⁾

11111 = 保留

•
•
•

11011 = 保留

11010 = PWM6——OF6_match⁽²⁾

11001 = PWM6——PH6_match⁽²⁾

11000 = PWM6——PR6_match⁽²⁾

10111 = PWM6——DC6_match⁽²⁾

10110 = PWM5——OF5_match

10101 = PWM5——PH5_match

10100 = PWM5——PR5_match

10011 = PWM5——DC5_match

10010 = PWM4——PWM4OUT⁽²⁾

10001 = PWM3——PWM3OUT

10000 = CCP2——CCP2_trigger⁽²⁾

01111 = CCP1——CCP1_trigger

01110 = CLC3——LC3_out

01101 = CLC2——LC2_out

01100 = CLC1——LC1_out

01011 = 比较器 C4——sync_C4OUT⁽²⁾

01010 = 比较器 C3——sync_C3OUT⁽²⁾

01001 = 比较器 C2——sync_C2OUT

01000 = 比较器 C1——sync_C1OUT

00111 = Timer6——T6_postscaled

00110 = Timer5——T5_overflow

00101 = Timer4——T4_postscaled

00100 = Timer3——T3_overflow

00011 = Timer2——T2_postscaled

00010 = Timer1——T1_overflow

00001 = Timer0——T0_overflow

00000 = 未选择任何自动转换触发源

bit 3-0

未实现: 读为 0

注 1: 这是所有触发源的上升沿敏感输入。

2: 仅限 PIC16(L)F1768/9。对于其他器件保留。

寄存器 16-4: 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 16-5: 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)F1764/5/8/9

寄存器 16-6: **ADRESH: ADC 结果寄存器高字节 (ADRESH)** ADFM = 1

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|------------|
| — | — | — | — | — | — | — | ADRES<9:8> |
| bit 7 | | | | | | | |

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 16-7: **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							

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

16.4 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 16-4 所示。模拟信号源阻抗（Rs）和内部采样开关阻抗（RSS）直接影响电容 CHOLD 的充电时间。采样开关阻抗（RSS）随器件电压（VDD）的变化而变化，参见图 16-4。模拟信号源的最大阻抗推荐值为 10 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 16-1 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 16-1：采集时间示例

假设： 温度 = 50°C，外部阻抗为 10 kΩ，VDD 为 5.0V

$$\begin{aligned} TACQ &= \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数} \\ &= TAMP + TC + TCOFF \\ &= 2 \mu s + T_C + [(\text{温度} - 25^\circ C) (0.05 \mu s / ^\circ C)] \end{aligned}$$

TC 值可以用以下公式近似计算：

$$V_{APPLIED} \left(1 - \frac{1}{(2^{n+1}) - 1} \right) = V_{CHOLD} \quad ;[1] \text{ 充电 } V_{CHOLD} \text{ (1/2 LSb 误差范围)}$$

$$V_{APPLIED} \left(1 - e^{-\frac{T_C}{RC}} \right) = V_{CHOLD} \quad ;[2] \text{ 响应 } V_{APPLIED} \text{ 充电 } V_{CHOLD}$$

$$V_{APPLIED} \left(1 - e^{-\frac{T_C}{RC}} \right) = V_{APPLIED} \left(1 - \frac{1}{(2^{n+1}) - 1} \right) \quad ; \text{ 合并 [1] 和 [2]}$$

注 其中 n = ADC 的位数。

求解 TC：

$$\begin{aligned} TC &= -CHOLD(RI C + RSS + RS) \ln(1/2047) \\ &= -10 pF(1 k\Omega + 7 k\Omega + 10 k\Omega) \ln(0.0004885) \\ &= 1.37 \mu s \end{aligned}$$

因此

$$\begin{aligned} TACQ &= 2 \mu s + 892 ns + [(50^\circ C - 25^\circ C)(0.05 \mu s / ^\circ C)] \\ &= 4.62 \mu s \end{aligned}$$

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。

PIC16(L)F1764/5/8/9

图 16-4: 模拟输入模型

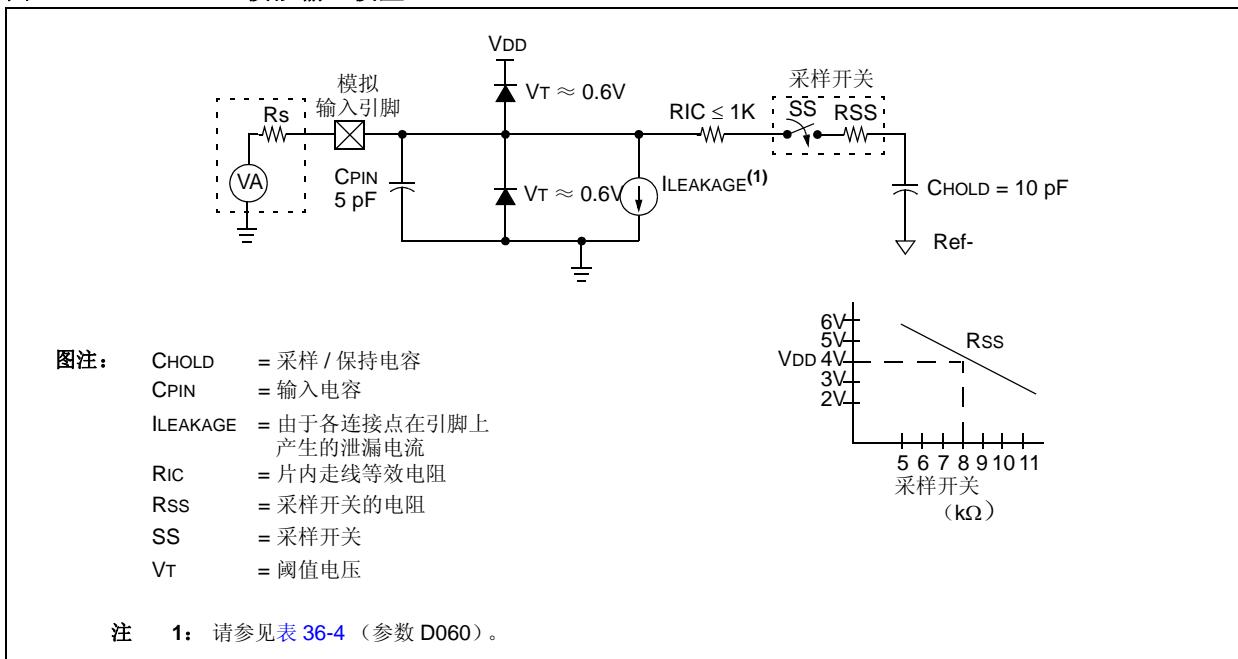


图 16-5: ADC 传递函数

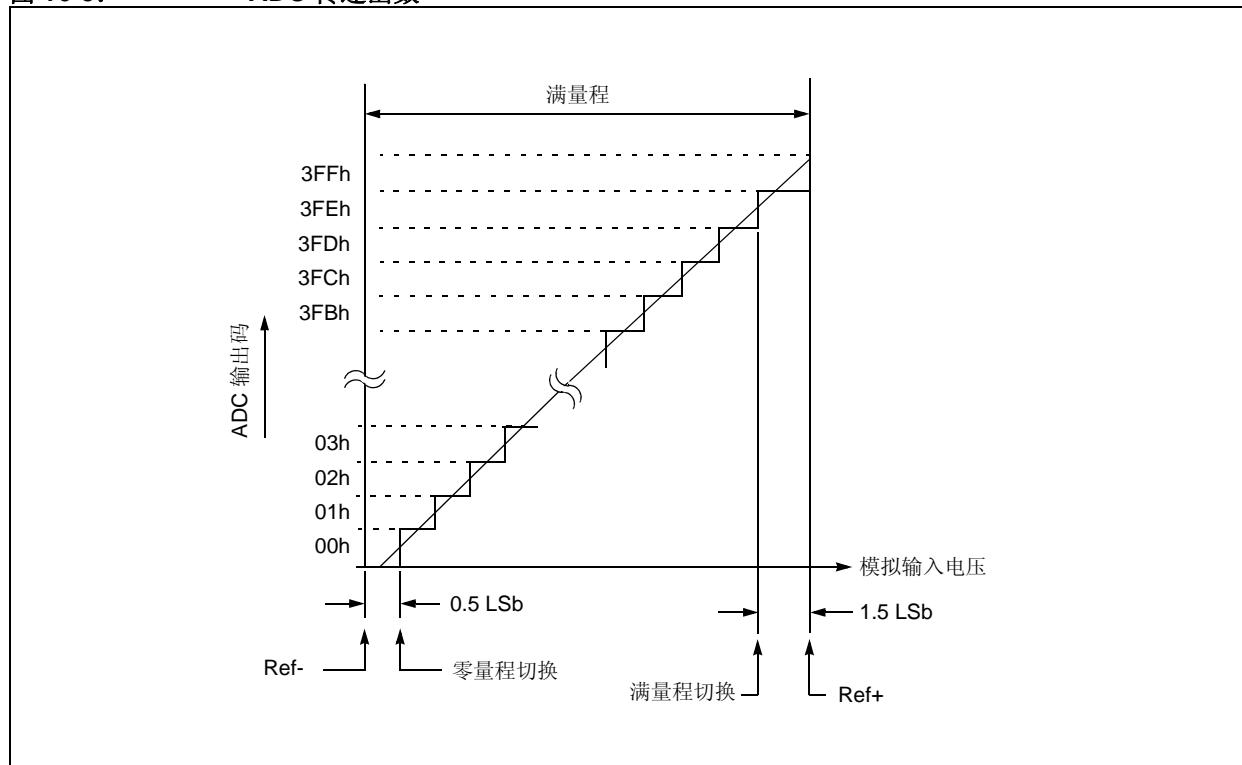


表 16-3：与 ADC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页			
ADCON0	—	CHS<4:0>				GO/DONE	ADON	182				
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		183			
ADCON2	TRIGSEL<4:0>				—	—	—	—	184			
ADRESH	ADC 结果寄存器的高字节								185, 186			
ADRESL	ADC 结果寄存器的低字节								185, 186			
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140			
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146			
ANSEL C	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151			
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102			
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103			
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106			
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139			
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145			
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150			
FVRCON	FVREN	FVR RDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		173			
DAC1CON0	EN	FM	OE1	—	PSS<1:0>		NSS<1:0>		193			

图注：x = 未知，u = 不变，— = 未实现，读为 0，q = 值取决于具体条件。ADC 模块不使用阴影单元。

注 1：仅限 PIC16(L)F1768/9。

2：未实现位，读为 1。

17.0 5 位数模转换器 (DAC) 模块

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

DAC 的输入可以连接到：

- 外部 VREF 引脚
- VDD 供电电压
- 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
- 运算放大器的反相和同相输入
- ADC 输入通道
- DACxOUT1 引脚

表 17-1：可用的 5 位 DAC

器件	D3	D4
PIC16(L)F1764	•	
PIC16(L)F1765	•	
PIC16(L)F1768	•	•
PIC16(L)F1769	•	•

数模转换器 (DAC) 可以通过将 DACxCON0 寄存器的 EN 位置 1 来使能。

17.1 输出电压选择

DAC 具有 32 个输出电压可供选择。32 个电压通过 DACxREF 寄存器的 REF<4:0> 位进行设置。

DAC 输出电压由公式 17-1 确定：

公式 17-1：DAC 输出电压

如果 $DACxEN = 1$

$$V_{OUT} = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACxR[4:0]}{2^5} \right) + V_{SOURCE-}$$

$V_{SOURCE+} = VDD, VREF$ 或 FVR 缓冲器 2

$V_{SOURCE-} = VSS$

17.2 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 36-20 中给出了梯形电阻网络中各个电阻的阻值。

17.3 DAC 参考电压输出

通过将 DACxCON0 寄存器的 OE1 位置 1，可以将 DAC 电压输出到 DACxOUT1 引脚上。选择将 DAC 电压输出到 DACxOUT1 引脚会自动改写数字输出缓冲器和该引脚的数字输入阈值检测器功能。当 DACxOUT1 引脚已被配置为 DAC 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，必须在 DAC 电压输出引脚 DACxOUT1 上外接缓冲器。图 17-2 举例说明了这一缓冲技术。

图 17-1: 数模转换器框图

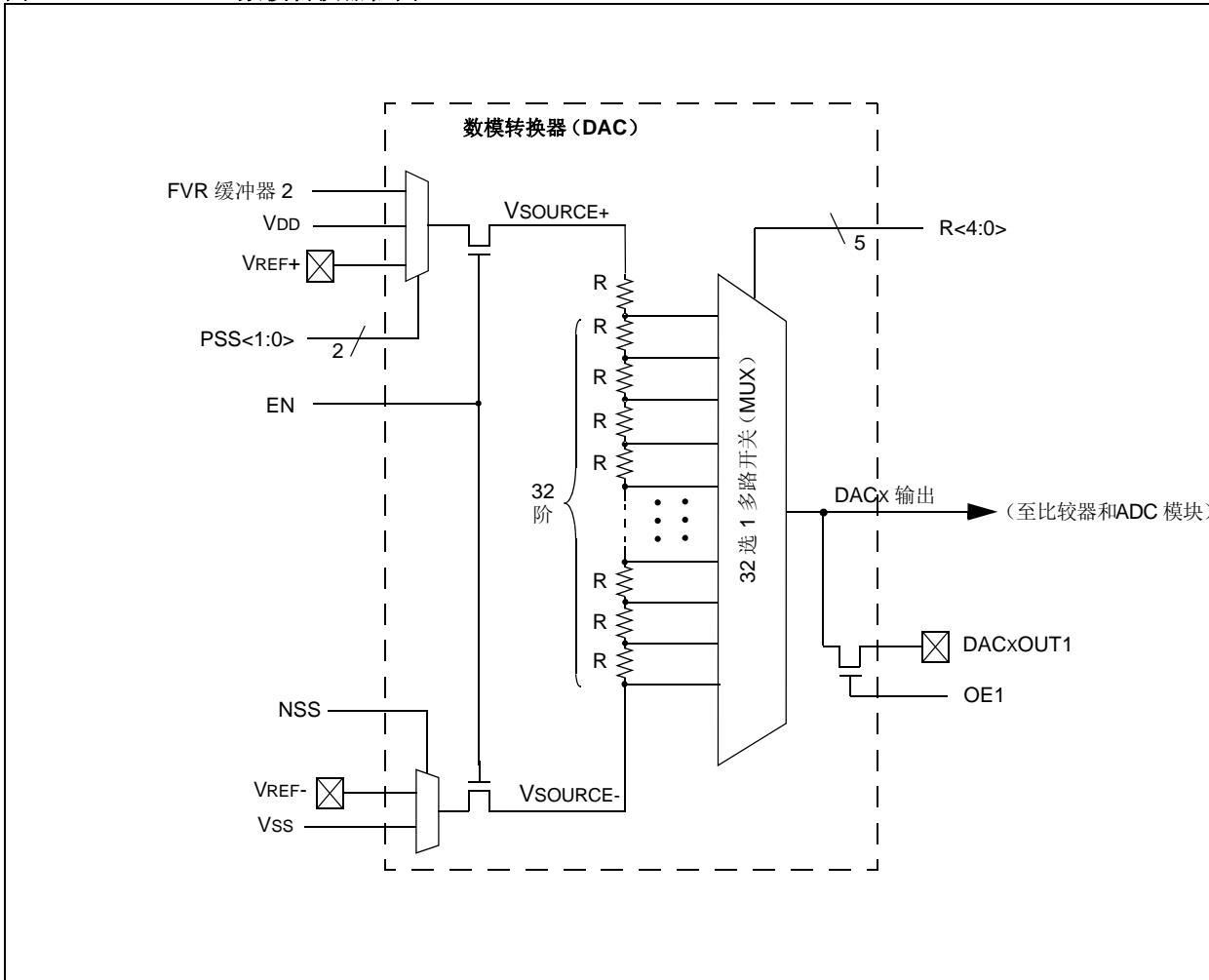
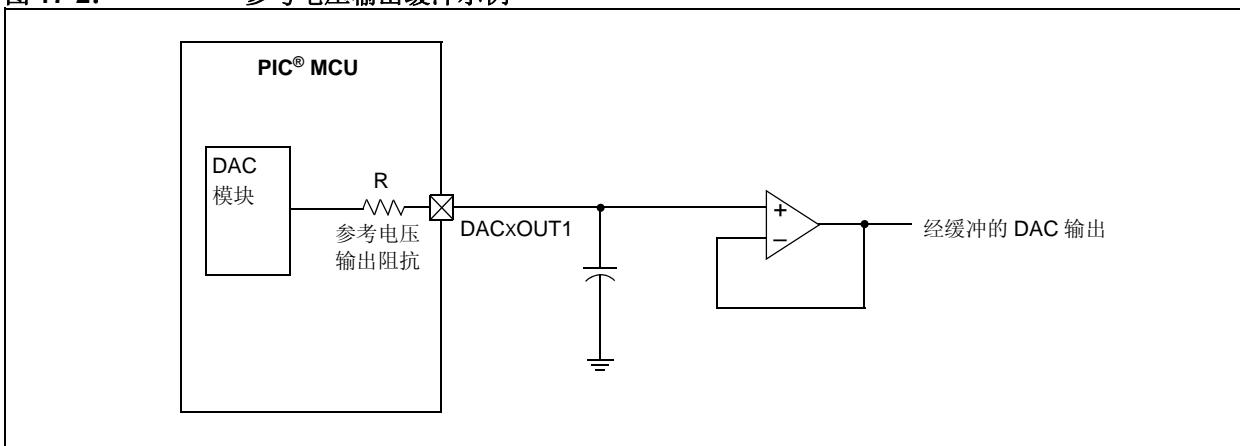


图 17-2: 参考电压输出缓冲示例



17.4 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，**DACxCON0** 寄存器的内容将不受影响。为了最大程度降低休眠模式下的电流消耗，应禁止参考电压模块。

17.5 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 **DACxOUT1** 引脚上被移除。
- **REF<4:0>** 参考电压控制位被清零。

17.6 寄存器定义：DAC 控制

表 17-2 列出了 5 位 DAC 外设的长位名称前缀。更多信息，请参见第 1.1 节“寄存器和位命名约定”。

表 17-2：

外设	位名称前缀
DAC3	DAC3
DAC4 ⁽¹⁾	DAC4

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 17-1： DACxCON0: DACx 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
EN	—	OE1	—	PSS<1:0>	—	—	NSS
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 **EN:** DAC 使能位

1 = 使能 DAC

0 = 禁止 DAC

bit 6 **未实现:** 读为 0

bit 5 **OE1:** DAC 电压输出使能位

1 = DAC 电压也从 DACxOUT1 引脚输出

0 = DAC 电压从 DACxOUT1 引脚断开

bit 4 **未实现:** 读为 0

bit 3-2 **PSS<1:0>:** DAC 正参考电压源选择位

11 = 保留；不要使用

10 = FVR 缓冲器 2 输出

01 = VREF+ 引脚

00 = VDD

bit 1 **未实现:** 读为 0

bit 0 **NSS:** DAC 负参考电压源选择位

1 = VREF- 引脚

0 = VSS

寄存器 17-2： DACxREF: DACx 参考电压输出选择寄存器

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
—	—	—			REF<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 **REF<4:0>:** DACx 参考电压输出选择位（见公式 17-1）

PIC16(L)F1764/5/8/9

表 17-3: 与 DACx 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
DAC3CON0	EN	—	OE1	—	PSS<1:0>	—	NSS	193	
DAC4CON0 ⁽¹⁾	EN	---	OE1	---	PSS<1:0>	---	NSS	193	
DAC3REF	---	---	---		REF<4:0>				193
DAC4REF ⁽¹⁾	---	---	---		REF<4:0>				193

图注: — = 未实现位, 读为 0。 DACx 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F1768/9。

18.0 10 位数模转换器 (DAC) 模块

10 位数模转换器 (DAC) 提供了一个可变参考电压，它与输入源成比例，具有 1024 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
- DACxOUT1 引脚
- 运放

数模转换器可以通过将 DACxCON0 寄存器的 EN 位置 1 来使能。

公式 18-1: DAC 输出电压

如果 EN = 1

$$DAC_x_output = \left((VSOURCE+ - VSOURCE-) \times \frac{DACxR[9:0]}{2^{10}} \right) + VSOURCE-$$

VSOURCE+ = VDD、VREF+ 或 FVR_buffer2

VSOURCE- = VSS 或 VREF-

表 18-1: 可用的 10 位 DAC

器件	D1	D2
PIC16(L)F1764	•	
PIC16(L)F1765	•	
PIC16(L)F1768	•	•
PIC16(L)F1769	•	•

18.1 输出电压选择

DAC 具有 1024 个电压，它们通过 DACxREFH 和 DACxREFL 寄存器中包含的 10 位参考电压选择字设置。这个 10 位的字可以进行左对齐或右对齐。更多详细信息，请参见第 18.4 节“DAC 参考电压选择对齐”。

可以使用公式 18-1 来确定 DAC 输出电压：

18.2 比例输出电压

DAC 输出电压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电压源和负电压源连接。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 36-20 中给出了梯形电阻网络中各个电阻的阻值。

18.3 DAC 输出

DAC 电压对于使用它的内部外设总是可用的。通过将 DACxCON0 寄存器的 OE1 位置 1，可以将 DAC 电压输出到 DACxOUT1 引脚上。选择将 DAC 电压输出到 DACxOUT1 引脚会自动改写数字输出缓冲器和该引脚的数字输入阈值检测器功能。当 DACxOUT1 引脚已被配置为 DAC 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，必须在 DAC 电压输出引脚 DACxOUT1 上外接缓冲器。图 18-3 举例说明了这一缓冲技术。

18.4 DAC 参考电压选择对齐

DAC 参考电压选择可以配置为进行左对齐或右对齐。当 DACxCON0 寄存器的 FM 位置 1 时，10 位字进行左对齐，以使高 8 位填充 DACxREFH 寄存器，低 2 位在 DACxREFL 寄存器中进行左对齐。当 FM 位清零时，10 位字进行右对齐，以使低 8 位填充 DACxREFL 寄存器，高 2 位在 DACxREFH 寄存器中进行右对齐。请参见图 18-1。

DACxREFL 和 DACxREFH 寄存器是双重缓冲的。对这两个寄存器的写操作不会立即生效。向 DACLD 寄存器的 DACxLD 位写入 1 时，会将 DACxREFH 和 DACxREFL 寄存器的内容传送到缓冲区，从而同时更改 DAC 参考电压选择的全部 10 位。

PIC16(L)F1764/5/8/9

图 18-1: DAC 对齐

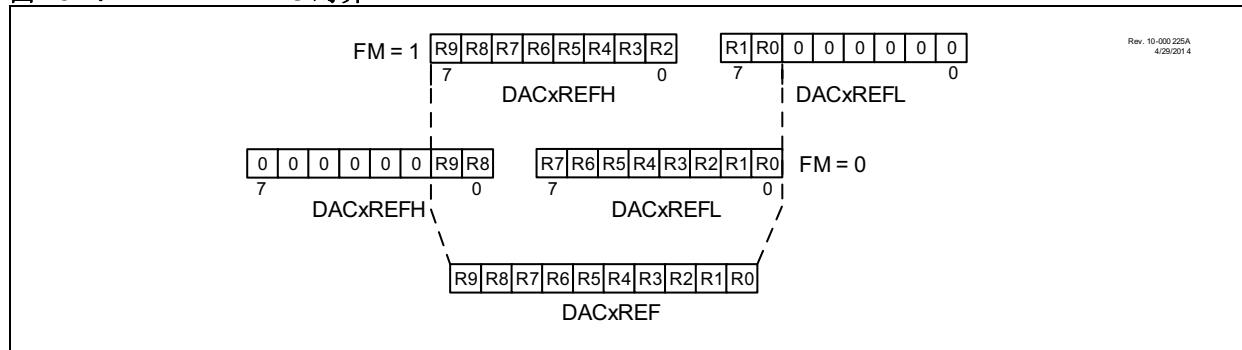


图 18-2: 数模转换器框图

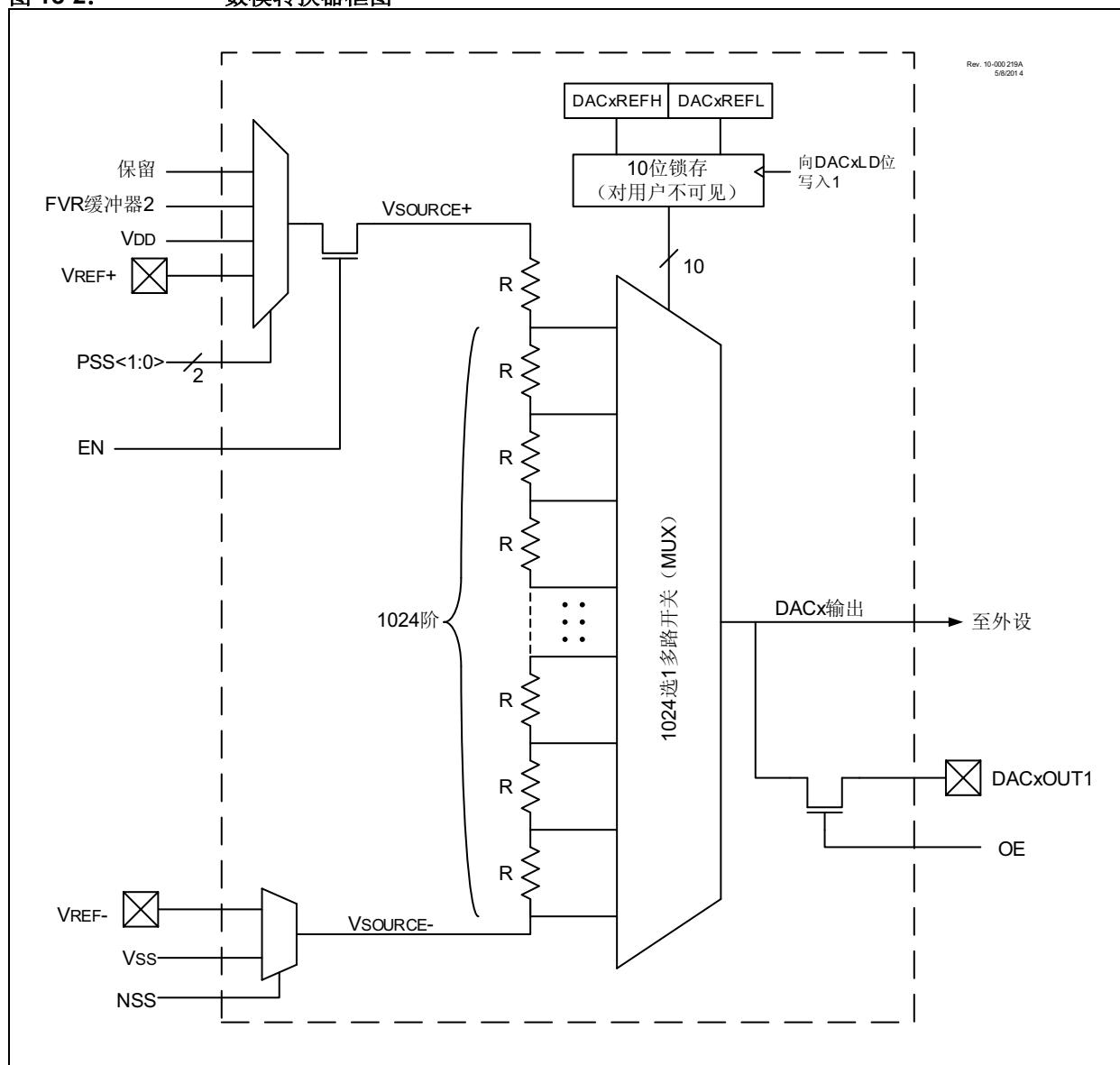
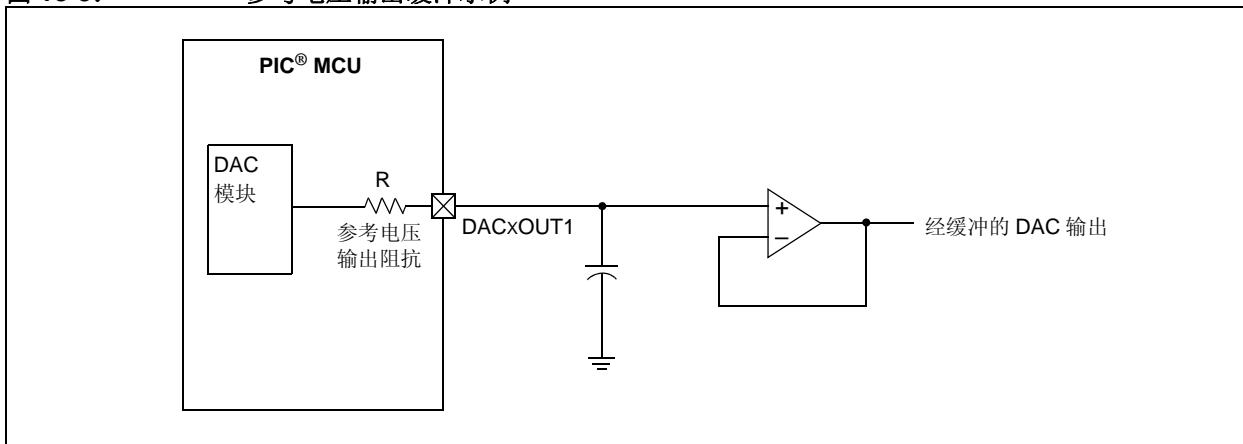


图 18-3: 参考电压输出缓冲示例



18.5 休眠期间的操作

如果因中断或看门狗定时器超时将器件从休眠模式唤醒，**DACxCON0** 寄存器的内容将不受影响。为了最大程度降低休眠模式下的电流消耗，应禁止参考电压模块。

18.6 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 **DACxOUT1** 引脚上被移除
- **REF<9:0>** 参考电压选择位被清零

18.7 寄存器定义：DAC 控制

表 18-2 列出了 10 位 DAC 外设的长位名称前缀。更多信息，请参见第 1.1 节 “寄存器和位命名约定”。

表 18-2：

外设	位名称前缀
DAC1	DAC1
DAC2 ⁽¹⁾	DAC2

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 18-1： DACxCON0：DAC 控制寄存器 0

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
EN	FM	OE1	—	PSS<1:0>	—	—	NSS
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 **EN:** DAC 使能位
1 = 使能 DACx
0 = 禁止 DACx
- bit 6 **FM:** DAC 参考电压格式位
1 = DACx 参考电压选择进行左对齐
0 = DACx 参考电压选择进行右对齐
- bit 5 **OE1:** DAC 电压输出使能位
1 = DACx 电压也从 DACxOUT1 引脚输出
0 = DACx 电压从 DACxOUT1 引脚断开
- bit 4 未实现：读为 0
- bit 3-2 **PSS<1:0>:** DAC 正参考电压源选择位
11 = 保留。不要使用。
10 = FVR 缓冲器 2
01 = VREF+ 引脚
00 = VDD
- bit 1 未实现：读为 0
- bit 0 **NSS:** DAC 负参考电压源选择位
1 = VREF- 引脚
0 = VSS

PIC16(L)F1764/5/8/9

寄存器 18-2: DACxREFH: DAC 参考电压选择高字节寄存器

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
REF<9:x> (x 取决于 FM 位)							
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

FM = 1 时 (左对齐)

bit 7-0 **REF<9:2>**: DAC 参考电压输出选择位
DACxOUT1 = f(REF<9:0>) (见公式 18-1)

FM = 0 时 (右对齐)

bit 7-2 未实现: 读为 0

bit 1-0 **REF<9:8>**: DAC 参考电压输出选择位
DACxOUT1 = f(REF<9:0>) (见公式 18-1)

寄存器 18-3: DACxREFL: DAC 参考电压选择低字节寄存器

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
REF<x-1:0> (x 取决于 FM 位)							
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

FM = 1 时 (左对齐)

bit 7-6 **REF<1:0>**: DAC 参考电压输出选择位
DACxOUT1 = f(REF<9:0>) (见公式 18-1)

bit 5-0 未实现: 读为 0

FM = 0 时 (右对齐)

bit 7-0 **REF<7:0>**: DAC 参考电压输出选择位
DACxOUT1 = f(REF<9:0>) (见公式 18-1)

寄存器 18-4: DACLD: DAC 缓冲区装入寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	DAC2LD ⁽¹⁾	DAC1LD
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 = 值取决于配置位

bit 7-2 未实现: 读为 0

bit 1 **DAC2LD: DAC2 双重缓冲区装入位⁽¹⁾**

1 = 将 DAC2REFH:DAC2REFL 值传送到双重缓冲区。该位由硬件自动清零。

0 = DAC2REFH:DAC2REFL 双重缓冲区保持不变。

bit 0 **DAC1LD: DAC1 双重缓冲区装入位**

1 = 将 DAC1REFH:DAC1REFL 值传送到双重缓冲区。该位由硬件自动清零。

0 = DAC1REFH:DAC1REFL 双重缓冲区保持不变。

注 1: 仅限 PIC16(L)F1768/9。

表 18-3: 与 DACx 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
DAC1CON0	EN	FM	OE1	—	PSS<1:0>	—	NSS	199	
DAC2CON0 ⁽¹⁾	EN	FM	OE1	—	PSS<1:0>	—	NSS	199	
DAC1REFH	REF<9:x> (x 取决于 FM 位)								200
DAC2REFH ⁽¹⁾	REF<9:x> (x 取决于 FM 位)								200
DAC1REFL	REF<x-1:0> (x 取决于 FM 位)								200
DAC2REFL ⁽¹⁾	REF<x-1:0> (x 取决于 FM 位)								200
DACLD	—	—	—	—	—	—	DAC2LD ⁽¹⁾	DAC1LD	201

图注: — = 未实现位, 读为 0。 DACx 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F1768/9。

19.0 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，建立模拟电路与数字电路的接口。比较器是非常有用的混合信号模块，因为它们提供了与程序执行相独立的模拟功能。模拟比较器模块具有以下特性

- 独立的比较器控制
- 可编程输入选择
- 可供内部 / 外部使用的比较器输出
- 可编程输出极性
- 电平变化中断
- 从休眠模式唤醒
- 可编程的速度 / 功耗优化
- PWM 关闭
- 可编程和固定参考电压

19.1 比较器概述

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

表 19-1 列出了该器件可用的比较器。

表 19-1：可用的比较器

器件	C1	C2	C3	C4
PIC16(L)F1764	•	•		
PIC16(L)F1765	•	•		
PIC16(L)F1768	•	•	•	•
PIC16(L)F1769	•	•	•	•

图 19-1：单个比较器

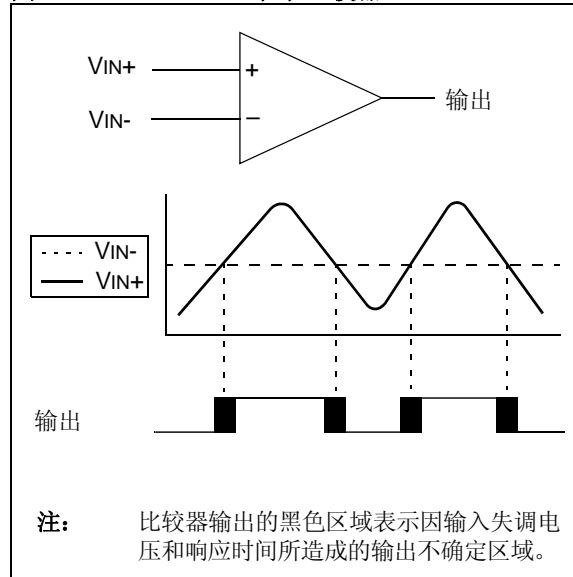
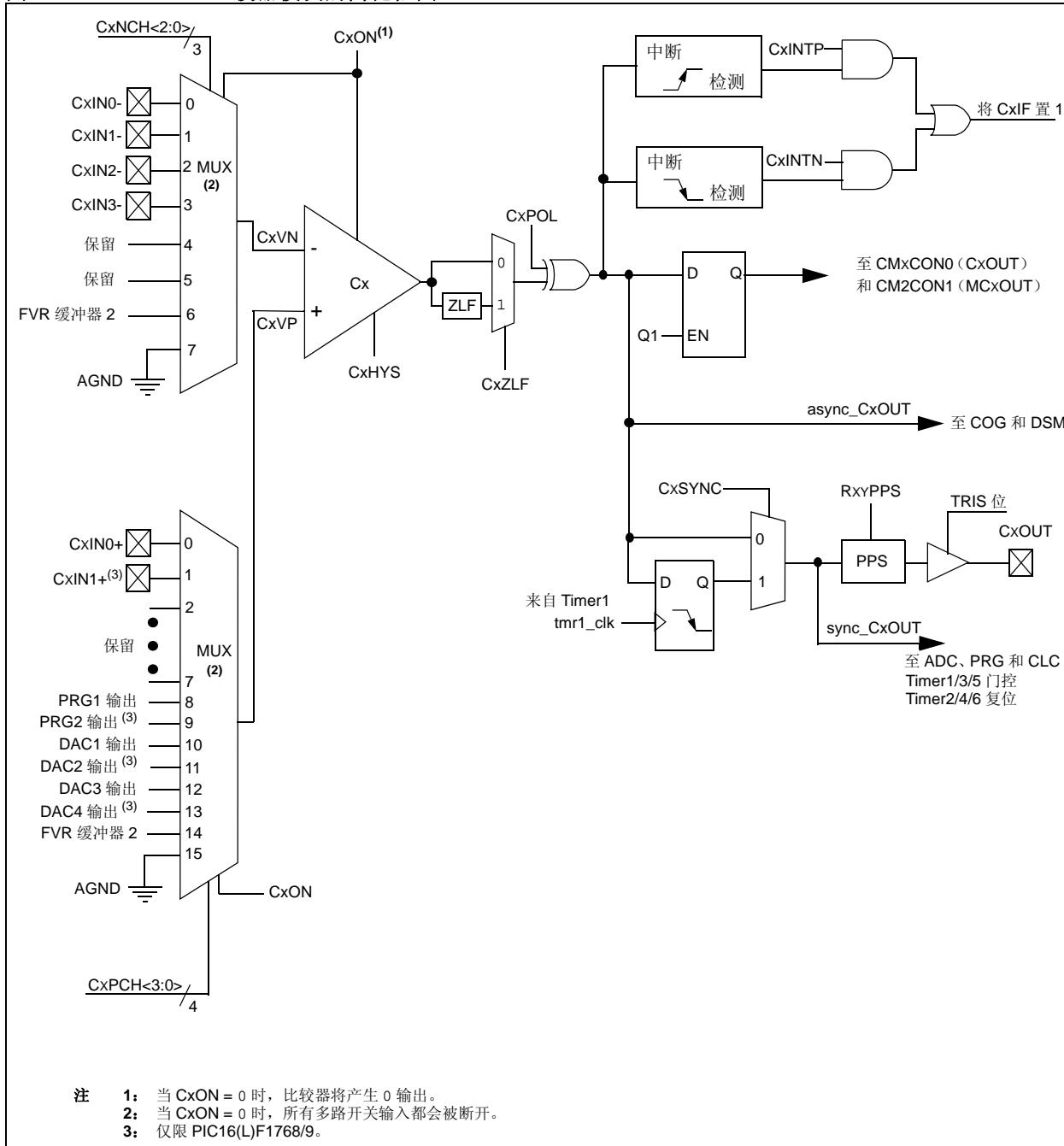


图 19-2: 比较器模块的简化框图



19.2 比较器控制

每个比较器都具有 2 个控制寄存器：CMxCON0 和 CMxCON1。

CMxCON0 寄存器（见[寄存器 19-1](#)）包含以下控制和状态位：

- 使能
- 输出
- 输出极性
- 零延时滤波器
- 速度 / 功耗选择
- 滞后使能
- 输出同步

CMxCON1 寄存器（见[寄存器 19-2](#)）包含以下控制位：

- 中断允许
- 中断边沿极性
- 同相输入通道选择
- 反相输入通道选择

19.2.1 比较器使能

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

19.2.2 比较器输出选择

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

- 所需的引脚 PPS 控制
- 必须清零相应的 TRIS 位
- 必须将 CMxCON0 寄存器的 ON 位置 1

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。

19.2.3 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 POL 位置 1 来使比较器输出的极性反相。清零 POL 位输出不反相。

表 19-2 给出了输出状态与输入条件的关系（包括极性控制）。

表 19-2： 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

19.3 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 HYS 位置 1 来使能。

更多信息，请参见表 36-19 中的比较器规范。

19.4 Timer1 门控操作

比较器操作产生的输出可以用作 Timer1 的门控源。更多信息，请参见第 22.6 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中发生变化时，Timer1 不会递增。

19.4.1 比较器输出同步

通过将 CMxCON0 寄存器的 SYNC 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 时钟源的下降沿被锁存。如果 Timer1 使用了预分频器，则比较器的输出在经过预分频后被锁存。为了防止发生竞争，比较器的输出在 Timer1 时钟源的下降沿被锁存，而 Timer1 在其时钟源的上升沿递增。更多信息，请参见比较器框图（图 19-2）和 Timer1 框图（图 22-1）。

19.5 比较器中断

比较器可以在输出值发生改变时产生中断；对于每个比较器，都提供了上升沿检测器和下降沿检测器。

当触发任一边沿检测器时，如果它关联的允许位已置 1（CMxCON1 寄存器的 INTP 和 / 或 INTN 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 ON 和 POL 位
- PIE2 寄存器的 CxIE 位
- CMxCON1 寄存器的 INTP 位（对于上升沿检测）
- CMxCON1 寄存器的 INTN 位（对于下降沿检测）
- INTCON 寄存器的 PEIE 和 GIE 位

关联的中断标志位（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 POL 位更改输出极性来产生中断，或者通过使用 CMxCON0 寄存器的 ON 位开启或关闭比较器来产生中断。

19.6 比较器同相输入选择

通过配置 CMxCON1 寄存器的 PCH<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。

- CxIN+ 模拟引脚
- 可编程斜坡发生器输出
- DAC 输出
- 固定参考电压（FVR）
- Vss（地）

关于固定参考电压模块的更多信息，请参见第 14.0 节“**固定参考电压（FVR）**”。

关于 DAC 输入信号的更多信息，请参见第 17.0 节“**5 位数模转换器（DAC）模块**”。

每当禁止比较器（CxON = 0）时，所有比较器输入都会被禁止。

19.7 比较器反相输入选择

CMxCON0 寄存器的 NCH<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
- 固定参考电压（FVR）
- 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。

19.8 比较器响应时间

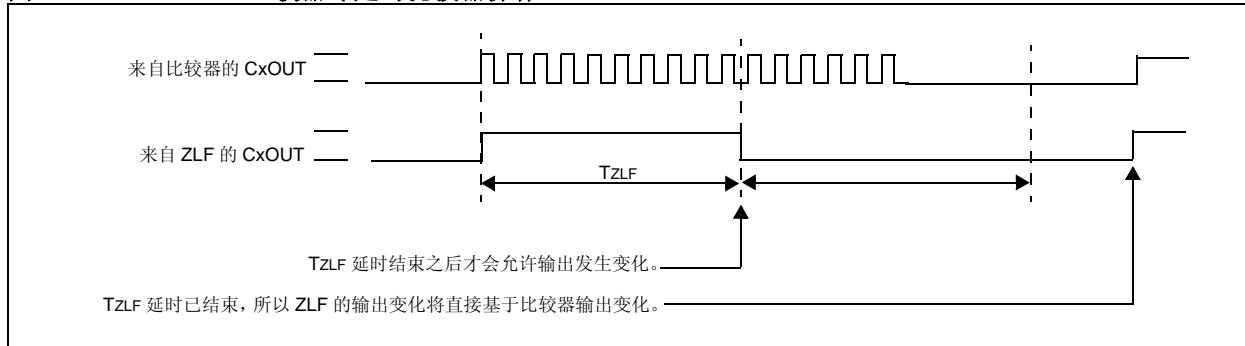
在改变输入源或选择新的参考电压后，一段时间内比较器的输出状态都是不确定的。这段时间被称为响应时间。比较器的响应时间不同于参考电压的稳定时间。因此，在确定比较器输入改变的总响应时间时，必须考虑这两个时间。更多详细信息，请参见表 36-19 中的比较器和参考电压规范。

19.9 零延时滤波器

在高速工作并处于适当电路条件下时，比较器输出可能会发生振荡。这种振荡会对依赖于该信号的硬件和软件产生不利影响。因此，在比较器输出上添加了一个数字

滤波器，用以抑制比较器输出振荡。当比较器输出发生变化时，将会在 20 ns 的标称时间内阻止输出发生反向变化。这使比较器输出可以稳定下来，而不会影响其他依赖器件。请参见图 19-3。

图 19-3: 比较器零延时滤波器操作



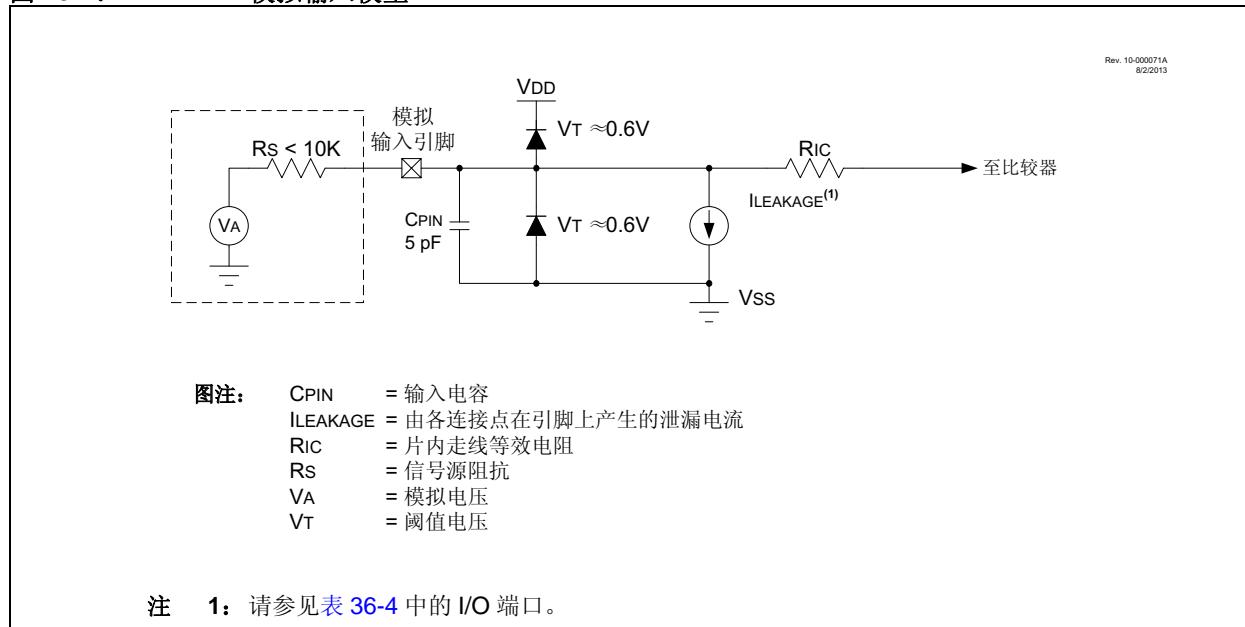
19.10 模拟输入连接注意事项

模拟输入的简化电路如图 19-4 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 $10\text{ k}\Omega$ 。任何连接到模拟输入引脚的外部元件（如电容或齐纳二极管），应保证其泄漏电流极小以使引入的误差降至最低。

- 注 1:** 读 PORT 寄存器时，所有配置为模拟输入的引脚均读为 0。配置为数字输入的引脚将根据输入规范转换为模拟输入。
- 2:** 定义为数字输入引脚上的模拟电平可能会使输入缓冲器的电流消耗超过规定值。

图 19-4: 模拟输入模型



PIC16(L)F1764/5/8/9

19.11 寄存器定义：比较器控制

表 19-3 列出了比较器外设的长位名称前缀。更多信息，请参见第 1.1.2.2 节“长位名称”。

表 19-3:

外设	位名称前缀
比较器 1	C1
比较器 2 ⁽¹⁾	C2

注 1: 仅限 PIC16(L)F1768/9 器件。

寄存器 19-1: CMxCON0: 比较器 Cx 控制寄存器 0

R/W-0/0	R-0/0	U-0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-0/0
ON	OUT	—	POL	ZLF	保留	HYS	SYNC
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 **ON:** 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 **OUT:** 比较器输出位
如果 POL = 1 (极性反相):
1 = CxVP < CxVN
0 = CxVP > CxVN
如果 POL = 0 (极性不反相):
1 = CxVP > CxVN
0 = CxVP < CxVN
- bit 5 未实现: 读为 0
- bit 4 **POL:** 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 **ZLF:** 比较器零延时滤波器使能位
1 = 对比较器输出进行滤波
0 = 不对比较器输出进行滤波
- bit 2 **保留:** 读为 1。保持该位置 1。
- bit 1 **HYS:** 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 **SYNC:** 比较器输出同步模式位
1 = 送到 Timer1 和 I/O 引脚的比较器输出与 Timer1 时钟源的变化进行同步。输出在 Timer1 时钟源的下降沿进行更新。
0 = 送到 Timer1 和 I/O 引脚的比较器输出是异步的

寄存器 19-2: CMxCON1: 比较器 Cx 控制寄存器 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	INTP	INTN
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 **INTP:** 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 0 **INTN:** 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

寄存器 19-3: CMxNSEL: 比较器 Cx 反相通道选择寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	NCH<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 7-3 未实现: 读为 0

bit 2-0 **NCH<2:0>:** 比较器反相输入通道选择位

111 = CxVN 连接到 AGND

110 = CxVN 连接到 FVR 缓冲器 2

101 = CxVN 连接到 PRG2 输出⁽¹⁾

100 = CxVN 连接到 PRG1 输出

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

注 1: 仅限 PIC16(L)F1768/9。

PIC16(L)F1764/5/8/9

寄存器 19-4: CMxPSEL: 比较器 Cx 同相通道选择寄存器 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
—	—	—	—	PCH<3:0>			
bit 7							

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 PCH<3:0>: 比较器同相输入通道选择位

1111 = CxVP 连接到 AGND

1110 = CxVP 连接到 FVR 缓冲器 2

1101 = CxVP 连接到 DAC4 输出

1100 = CxVP 连接到 DAC3 输出

1011 = CxVP 连接到 DAC2 输出

1010 = CxVP 连接到 DAC1 输出

1001 = CxVP 连接到 PRG2 输出

1000 = CxVP 连接到 PRG1 输出

0111 = CxVP 不连接, 输入悬空

0110 = CxVP 不连接, 输入悬空

0101 = CxVP 不连接, 输入悬空

0100 = CxVP 不连接, 输入悬空

0011 = CxVP 不连接, 输入悬空

0010 = CxVP 不连接, 输入悬空

0001 = CxVP 连接到 CxIN0+ 引脚

0000 = CxVP 连接到 CxIN0+ 引脚

注: 以下镜像寄存器没有长位名称和短位名称形式。

寄存器 19-5: CMOUT: 比较器输出寄存器

U-0	U-0	U-0	U-0	R-0/0	R-0/0	R-0/0	R-0/0
—	—	—	—	MC4OUT ⁽¹⁾	MC3OUT ⁽¹⁾	MC2OUT	MC1OUT
bit 7							

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

bit 7-2 未实现: 读为 0

bit 3 MC4OUT: C4OUT 的镜像副本位⁽¹⁾

bit 2 MC3OUT: C3OUT 的镜像副本位⁽¹⁾

bit 1 MC2OUT: C2OUT 的镜像副本位

bit 0 MC1OUT: C1OUT 的镜像副本位

注 1: 仅限 PIC16(L)F1768/9。

表 19-4：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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140							
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146							
ANSEL C	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151							
CM1CON0	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	208							
CM2CON0	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	208							
CM3CON0 ⁽¹⁾	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	208							
CM4CON0 ⁽¹⁾	ON	OUT	—	POL	ZLF	保留	HYS	SYNC	208							
CM1CON1	—	—	—	—	—	—	INTP	INTN	209							
CM2CON1	—	—	—	—	—	—	INTP	INTN	209							
CM3CON1 ⁽¹⁾	—	—	—	—	—	—	INTP	INTN	209							
CM4CON1 ⁽¹⁾	—	—	—	—	—	—	INTP	INTN	209							
CM1NSEL	—	—	—	—	—	NCH<2:0>										
CM2NSEL	—	—	—	—	—	NCH<2:0>										
CM3NSEL ⁽¹⁾	—	—	—	—	—	NCH<2:0>										
CM4NSEL ⁽¹⁾	—	—	—	—	—	NCH<2:0>										
CM1PSEL	—	—	—	—	PCH<3:0>											
CM2PSEL	—	—	—	—	PCH<3:0>											
CM3PSEL ⁽¹⁾	—	—	—	—	PCH<3:0>											
CM4PSEL ⁽¹⁾	—	—	—	—	PCH<3:0>											
CMOUT	—	—	—	—	MC4OUT ⁽¹⁾	MC3OUT ⁽¹⁾	MC2OUT	MC1OUT	210							
FVRCON	FVREN	FVR RDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		173							
DAC1CON0	EN	FM	OE1	—	PSS<1:0>											
DAC2CON0 ⁽¹⁾	EN	FM	OE1	—	PSS<1:0>											
DAC3CON0	EN	—	OE1	—	PSS<1:0>											
DAC4CON0 ⁽¹⁾	EN	—	OE1	—	PSS<1:0>											
DAC3REF	---	---	---	REF<4:0>												
DAC4REF ⁽¹⁾	---	---	---	REF<4:0>												
DAC1REFH	REF<9:x> (x 取决于 FM 位)								200							
DAC2REFH ⁽¹⁾	REF<9:x> (x 取决于 FM 位)								200							
DAC1REFL	REF<x-1:0> (x 取决于 FM 位)								200							
DAC2REFL ⁽¹⁾	REF<x-1:0> (x 取决于 FM 位)								200							
INTCON	GIE	PEIE	TMROIE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102							
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104							
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C3IF ⁽¹⁾	CCP2IF ⁽¹⁾	107							
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139							
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145							
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150							

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)F1768/9。

2：未实现位，读为 1。

20.0 过零检测（ZCD）模块

ZCD 模块用于检测交流信号越过地电位的时刻。实际的过零阈值为过零参考电压 $ZCPINV$ ，它通常比地电位高 $0.75V$ 。

要检测的信号通过一个串联限流电阻连接。该模块会在 ZCD 引脚上施加拉电流或灌电流，以便在引脚上维持恒定的电压，从而防止引脚电压使 ESD 保护二极管发生正向偏置。当施加的电压大于参考电压时，该模块会灌电流。当施加的电压小于参考电压时，该模块会拉电流。拉电流和灌电流操作会使引脚电压在所施加电压的完整范围内保持恒定。图 20-2 给出了 ZCD 模块的简化框图。

ZCD 模块在针对（但不限于）以下目的而监视交流波形时非常有用：

- 交流周期测量
- 精确的长时间测量
- 调光器相位延时驱动
- 低 EMI 周期切换

20.1 外部电阻选择

ZCD 模块要求将一个限流电阻与外部电压源串联。该电阻的阻抗和参数取决于外部源峰值电压。在通过电阻的电流标称值为 $300 \mu A$ 时，选择的阻值产生的压降需要等于峰值电压。请参见公式 20-1 和图 20-1。请确保禁止 ZCD I/O 引脚内部弱上拉，使它不会干扰拉电流和灌电流。

公式 20-1： 外部电阻

$$R_{series} = \frac{V_{peak}}{3 \times 10^{-4}}$$

图 20-1： 外部电压

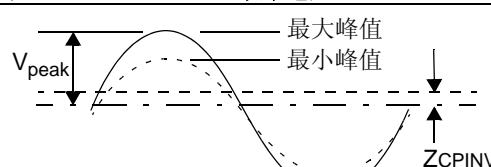
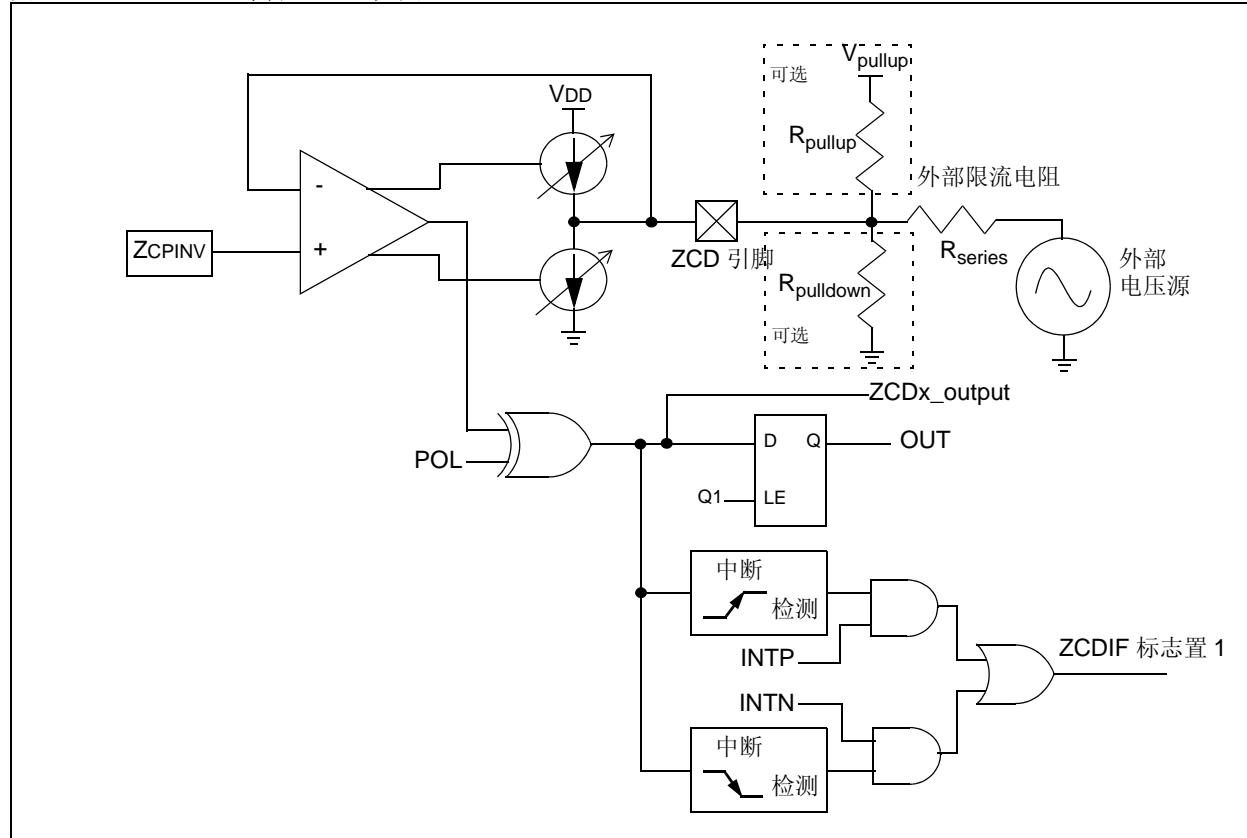


图 20-2： 简化 ZCD 框图



20.2 ZCD 逻辑输出

ZCD 模块包含了一个状态位，通过读取它可以确定处于工作状态的是拉电流还是灌电流。当处于工作状态的是灌电流时，ZCDCON 寄存器的 OUT 位会置 1，当处于工作状态的是拉电流时，该位会清零。OUT 位会受极性位影响。

20.3 ZCD 逻辑极性

ZCDxCON 寄存器的 POL 位用于将 OUT 位相对于拉电流和灌电流输出进行反相。当 POL 位置 1 时，OUT 为高电平表示处于工作状态的是拉电流，低电平输出表示处于工作状态的是灌电流。

POL 位会影响 ZCD 中断。请参见[第 20.4 节“ZCD 中断”](#)。

20.4 ZCD 中断

如果相应的中断允许位置 1，则在 ZCD 逻辑输出发生改变时，将会产生中断。因此，ZCD 中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，且其关联的中断允许位置 1 时，PIR3 寄存器的 ZCDIF 位会置 1。INTP 位用于允许上升沿中断，INTN 位用于允许下降沿中断。它们都位于 ZCDxCON 寄存器中。

要完全允许中断，必须将以下位置 1：

- PIE3 寄存器的 ZCDIE 位
- ZCDxCON 寄存器的 INTP 位
(用于上升沿检测)
- ZCDxCON 寄存器的 INTN 位
(用于下降沿检测)
- INTCON 寄存器的 PEIE 和 GIE 位

更改 POL 位将导致中断，无论 EN 位的电平如何。

作为中断服务的一部分，必须用软件将 PIR3 寄存器的 ZCDIF 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。

20.5 修正 ZCPINV 偏移

ZCD 发生开关的实际电压是 ZCD 运放同相输入上的参考电压。对于除方波之外的其他外部电压源波形，该电压相对于零电压发生偏移会导致过早或过迟地发生过零事件。当波形相对于 V_{SS} 发生变化时，在波形下降时会过早地检测到过零事件，在波形上升时会过迟地检测到过零事件。当波形相对于 V_{DD} 发生变化时，在波形上升时会过迟地检测到过零事件，在波形下降时会过早地检测到过零事件。使用[公式 20-2](#) 所示的相应公式，可以确定正弦波形的实际偏移时间。

公式 20-2: ZCD 事件偏移

当外部电压源相对于 V_{SS} 时：

$$T_{offset} = \frac{\arcsin\left(\frac{Z_{cpinv}}{V_{peak}}\right)}{2\pi \cdot \text{频率}}$$

当外部电压源相对于 V_{DD} 时：

$$T_{offset} = \frac{\arcsin\left(\frac{V_{DD}-Z_{cpinv}}{V_{peak}}\right)}{2\pi \cdot \text{频率}}$$

这种偏移时间可以通过在 ZCD 引脚添加上拉或下拉偏置电阻进行补偿。在外部电压源相对于 V_{SS} 变化时使用上拉电阻。在电压相对于 V_{DD} 变化时使用下拉电阻。该电阻会在 ZCD 引脚上添加偏置电压，使得只有目标外部电压源达到零电压时，才会将引脚电压拉至 ZCPINV 开关电压。上拉或下拉值可以使用[公式 20-3](#) 所示的公式确定。

公式 20-3: ZCD 上拉 / 下拉

当外部信号相对于 V_{SS} 时：

$$R_{pullup} = \frac{R_{series}(V_{pullup} - Z_{cpinv})}{Z_{cpinv}}$$

当外部信号相对于 V_{DD} 时：

$$R_{pulldown} = \frac{R_{series}(Z_{cpinv})}{(V_{DD} - Z_{cpinv})}$$

ZCPINV 的很小变化都会显著影响上拉和下拉电阻值。测量 ZCPINV 是很困难的，特别是在波形相对于 VDD 时。但是，通过结合 [公式 20-2](#) 和 [公式 20-3](#)，可以根据 ZCDOUT 高电平和低电平周期之间的时间差来确定电阻值。请注意，时间差 ΔT 等于 $4 \cdot T_{\text{offset}}$ 。[公式 20-4](#) 给出了根据高电平和低电平 ZCDOUT 周期确定上拉和下拉电阻值的公式。通过将 ZCDOUT 信号连接到某个 CLC，可以直接在引脚上观察 ZCDOUT 信号。

公式 20-4:

$$R = R_{\text{series}} \left(\frac{V_{\text{bias}}}{V_{\text{peak}} \left(\sin \left(\pi Freq \frac{(\Delta T)}{2} \right) \right)} - 1 \right)$$

R 为上拉或下拉电阻

V_{bias} 为 V_{pullup} (R 为上拉电阻时) 或 V_{DD} (R 为下拉电阻时)

ΔT 为 ZCDOUT 高电平和低电平周期的差值

20.7 休眠期间的操作

ZCD 拉电流和中断不会受休眠影响。

20.8 复位的影响

ZCD 电路可以配置为在发生上电复位 (POR) 时默认设为工作或非工作状态。当 \overline{ZCD} 配置位清零时，ZCD 电路将在 POR 时处于工作状态。当 \overline{ZCD} 配置位置 1 时，必须通过将 ZCDCON 寄存器的 ZCDEN 位置 1 来使能 ZCD 模块。

20.6 处理 V_{peak} 变化

如果预期外部电压的峰值大小会发生变化，则选择的串联电阻必须使 ZCD 拉电流和灌电流对于最大预期电压保持低于 $\pm 600 \mu\text{A}$ 的最大设计范围，并且也够高，足以在最小峰值电压下精确检测到。一般的经验是，最大峰值电压不能超出最小峰值电压的 6 倍。为了确保最大电流不超出 $\pm 600 \mu\text{A}$ ，最小值至少为 $\pm 100 \mu\text{A}$ ，需要按照 [公式 20-5](#) 所示的方式计算串联电阻。该串联电阻的补偿上拉电阻可以使用 [公式 20-3](#) 确定，因为上拉值与峰值电压无关。

公式 20-5: 对应于 V 范围的串联电阻 R

$$R_{\text{series}} = \frac{V_{\text{maxpeak}} + V_{\text{minpeak}}}{7 \times 10^{-4}}$$

20.9 寄存器定义：ZCD 控制

表 20-1 列出了过零检测外设的长位名称前缀。更多信息，请参见第 1.1.2.2 节“长位名称”。

表 20-1：

外设	位名称前缀
ZCD1	ZCD1

寄存器 20-1： ZCDxCON：过零检测控制寄存器

R/W-0/0	U-0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
EN	—	OUT	POL	—	—	INTP	INTN
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 = 值取决于配置位

- bit 7 **EN:** 过零检测使能位⁽¹⁾
1 = 使能过零检测。将 ZCD 引脚强制为输出拉电流和灌电流。
0 = 禁止过零检测。ZCD 引脚根据 PPS 和 TRIS 控制工作。
- bit 6 未实现：读为 0
- bit 5 **OUT:** 过零检测逻辑电平位
POL 位 = 0:
1 = ZCD 引脚灌电流
0 = ZCD 引脚拉电流
POL 位 = 1:
1 = ZCD 引脚拉电流
0 = ZCD 引脚灌电流
- bit 4 **POL:** 过零检测逻辑输出极性位
1 = 对 ZCD 逻辑输出进行反相
0 = 不对 ZCD 逻辑输出进行反相
- bit 3-2 未实现：读为 0
- bit 1 **INTP:** 过零正边沿中断允许位
1 = 在 OUT 从低电平变为高电平时，ZCDIF 位置 1
0 = 在 OUT 从低电平变为高电平时，ZCDIF 位不受影响
- bit 0 **INTN:** 过零负边沿中断允许位
1 = 在 OUT 从高电平变为低电平时，ZCDIF 位置 1
0 = 在 OUT 从高电平变为低电平时，ZCDIF 位不受影响

注 1：当 ZCD 配置位清零时，EN 位没有任何作用。

PIC16(L)F1764/5/8/9

表 20-2: 与 ZCD 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
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108
ZCD1CON	EN	—	OUT	POL	—	—	INTP	INTN	215

图注: — = 未实现位, 读为 0。ZCD 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F1768/9。

表 20-3: 与 ZCD 模块相关的配置字汇总

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	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	67
	7:0	ZCD	—	—	—	—	PPS1WAY	WRT<1:0>		

图注: — = 未实现位, 读为 0。ZCD 模块不使用阴影单元。

21.0 TIMER0 模块

Timer0 模块是 8 位定时器 / 计数器，具有以下特性：

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

图 21-1 给出了 Timer0 模块的框图。

21.1 Timer0 工作原理

Timer0 模块可被用作 8 位定时器或 8 位计数器。

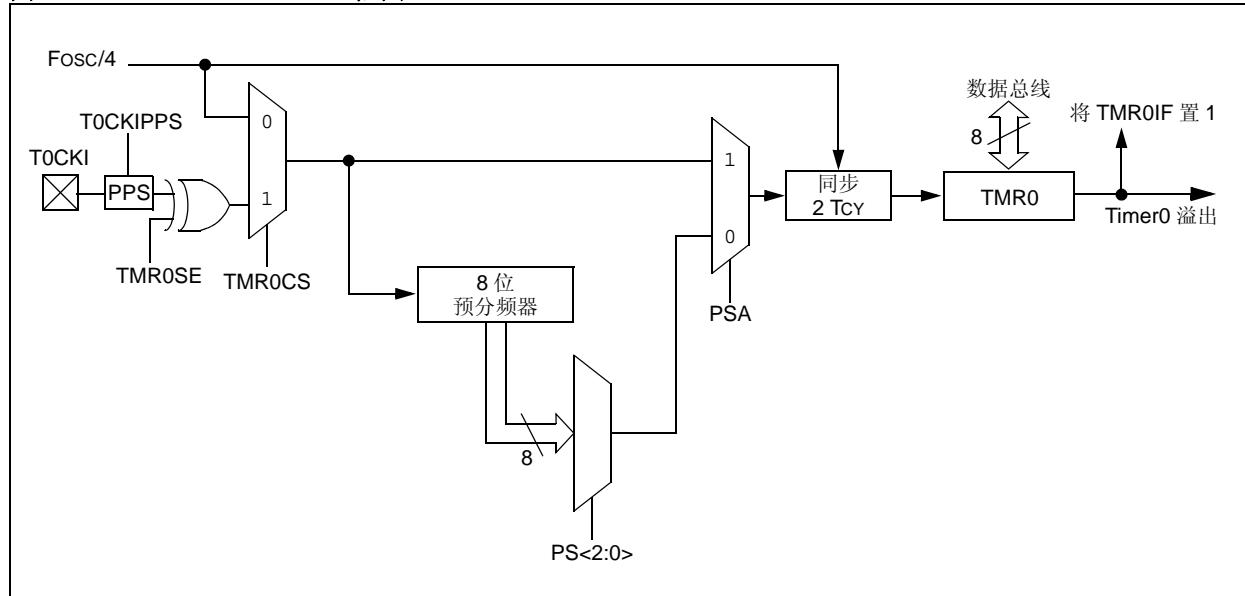
21.1.1 8 位定时器模式

如果 Timer0 模块不使用预分频器，则 Timer0 模块将在每个指令周期递增。8 位定时器模式可通过清零 OPTION_REG 寄存器的 TMR0CS 位选择。

当写 TMR0 时，紧跟写操作之后的两个指令周期内禁止 TMR0 递增。

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

图 21-1：TIMER0 框图



21.1.2 8 位计数器模式

在 8 位计数器模式下，Timer0 模块将在 T0CKI 引脚的每个上升沿或下降沿递增。

使用 T0CKI 引脚的 8 位计数器模式，可通过将 OPTION_REG 寄存器中的 TMR0CS 位设置为 1 来选择。

两个输入源递增边沿是上升沿还是下降沿由 OPTION_REG 寄存器中的 TMROSE 位决定。

21.1.3 软件可编程的预分频器

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

注: 看门狗定时器 (WDT) 使用它自己的独立预分频器。

Timer0 模块有 8 个预分频比选项，范围从 1:2 至 1:256。预分频值可通过 OPTION_REG 寄存器的 PS_{2:0} 位进行选择。为了让 Timer0 模块使用 1:1 预分频值，必须通过将 OPTION_REG 寄存器的 PSA 位置 1 来禁止预分频器。

预分频器是不可读写的。写 TMR0 寄存器的所有指令都会清零预分频器。

21.1.4 TIMER0 中断

TMR0 寄存器从 FFh 溢出到 00h 时，将产生 Timer0 中断。每次 TMR0 寄存器溢出时都会将 INTCON 寄存器的 TMROIF 中断标志位置 1，这与是否允许 Timer0 中断无关。TMROIF 位只能用软件清零。Timer0 中断允许位是 INTCON 寄存器的 TMROIE 位。

注: 由于定时器在休眠模式下是停止的，所以 Timer0 中断无法将处理器从休眠模式唤醒。

21.1.5 8 位计数器模式同步

在 8 位计数器模式下，T0CKI 引脚的递增边沿必须与指令时钟保持同步。同步可通过在指令时钟的 Q2 和 Q4 周期对预分频器的输出进行采样实现。外部时钟源的高低电平周期必须满足表 36-12 中所示的时序要求。

21.1.6 休眠期间的操作

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

21.2 寄存器定义：选项寄存器

寄存器 21-1: **OPTION_REG:** 选项寄存器

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| WPUEN | INTEDG | TMR0CS | TMR0SE | PSA | PS<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 **WPUEN:** 弱上拉使能位

1 = 禁止所有弱上拉 (MCLR 除外, 如果已使能)
0 = 通过各个 WPUx 锁存值使能弱上拉

bit 6 **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

bit 5 **TMR0CS:** Timer0 时钟源选择位

1 = T0CKI 引脚上的电平跳变
0 = 内部指令周期时钟 (Fosc/4)

bit 4 **TMR0SE:** Timer0 时钟源边沿选择位

1 = 在 T0CKI 引脚信号从高至低跳变时, 递增计数
0 = 在 T0CKI 引脚信号从低至高跳变时, 递增计数

bit 3 **PSA:** 预分频器分配位

1 = 预分频器未分配给 Timer0 模块
0 = 预分频器分配给 Timer0 模块

bit 2-0 **PS<2:0>:** 预分频比选择位

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

表 21-1: 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
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			219
TMR0	Timer0 模块寄存器								217*
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	139

图注：— = 未实现位，读为 0。Timer0 模块不使用阴影单元。

* 提供寄存器信息的页。

注 1：未实现位，读为 1。

22.0 带门控的 TIMER1/3/5 模块

Timer1 模块是 16 位定时器 / 计数器，具有以下特性：

- 16 位定时器 / 计数器寄存器对 (TMR1H:TMR1L)
- 可编程内部或外部时钟源
- 2 位预分频器
- 专用的 32 kHz 振荡器电路
- 可选的同步比较器输出
- 多个 Timer1 门控（计数使能）源
- 溢出时产生中断
- 溢出触发唤醒（仅限外部时钟，异步模式）
- 捕捉 / 比较功能的时基
- 自动转换触发（带 CCP）

- 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

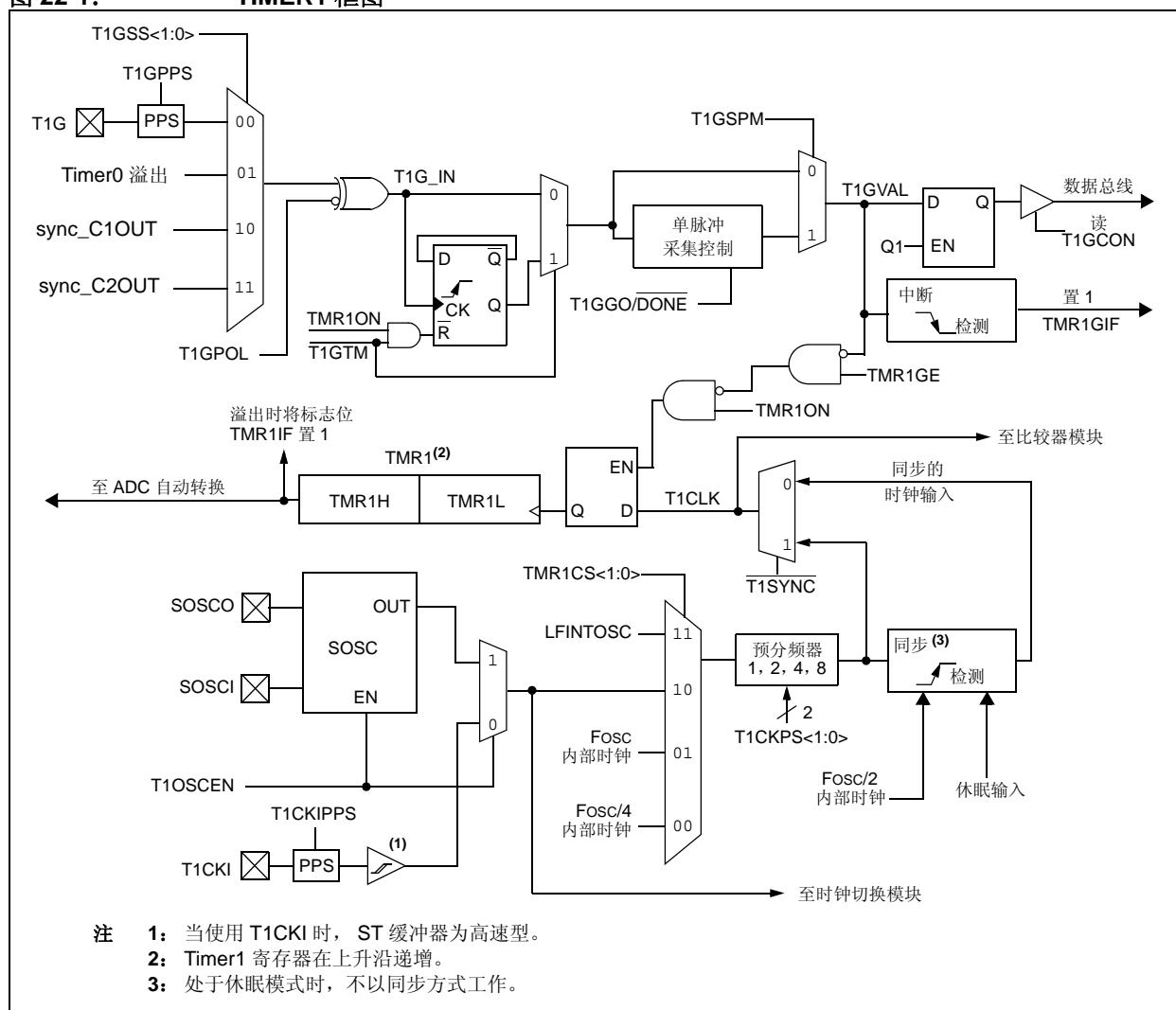
图 22-1 给出了 Timer1 模块的框图。

该器件具有 3 个 Timer1 类型的模块。它们包括：

- Timer1
- Timer3
- Timer5

对 Timer1 和 Timer1 门控的所有引用均适用于 Timer3 和 Timer5。

图 22-1: TIMER1 框图



22.1 Timer1 工作原理

Timer1模块是16位递增计数器，可通过TMR1H:TMR1L寄存器对访问。写TMR1H或TMR1L会直接更新计数器。

Timer1模块与内部时钟源一起使用时，模块用作定时器，并在每个指令周期递增。Timer1模块与外部时钟源一起使用时，模块可用作定时器或计数器，在外部时钟源的每个选定边沿递增。

Timer1分别通过配置T1CON和T1GCON寄存器中的ON和GE位使能。[表22-1](#)显示了Timer1使能选择。

表 22-1: 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

22.2 时钟源选择

T1CON寄存器的CS<1:0>和OSCEN位用于选择Timer1的时钟源。[表 22-2](#)显示了时钟源选择。

22.2.1 内部时钟源

当选择内部时钟源时，TMR1H:TMR1L寄存器对的递增频率将为Fosc的整数倍（取决于Timer1预分频器）。

选择Fosc内部时钟源时，Timer1寄存器的值将在每个指令时钟周期中递增4次。由于这个原因，在读取Timer1值时，分辨率将会出现2个LSB的误差。为了利用Timer1的全部分辨率，必须使用异步输入信号来对Timer1时钟输入进行门控。

可以使用以下异步源：

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

22.2.2 外部时钟源

当选择外部时钟源时，Timer1模块可以作为定时器或计数器工作。

当使能Timer1计数时，Timer1模块在外部时钟输入T1CKI的上升沿递增，该外部时钟源既可以与单片机系统时钟同步，也可以异步运行。

作为定时器采用时钟振荡器工作时，可以将外部32.768 kHz晶振与专用内部振荡器电路一起使用。

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

- POR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 22-2: 时钟源选择

TMR1CS<1:0>	T1OSCEN	时钟源
11	x	LFINTOSC
10	0	T1CKI引脚上的外部时钟源
01	x	系统时钟(Fosc)
00	x	指令时钟(Fosc/4)

22.3 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 CKPS 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。

22.4 Timer1（辅助）振荡器

在引脚 SOSCI（输入）和 SOSCO（放大器输出）之间接有一个内置专用低功耗 32.768 kHz 振荡器电路。该内部电路与一个外部 32.768 kHz 晶振配合使用。

通过将 T1CON 寄存器的 OSCEN 位置 1 可使能振荡器电路。在休眠期间，振荡器将继续运行。

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

22.5 异步计数器模式下的 Timer1 操作

如果 T1CON 寄存器的控制位 SYNC 置 1，外部时钟输入将不同步。定时器异步于内部相位时钟进行递增计数。如果选择了外部时钟源，在休眠期间定时器将继续运行，并在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 22.5.1 节“[在异步计数器模式下读写 Timer1](#)”）。

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

22.5.1 在异步计数器模式下读写 TIMER1

当定时器采用外部异步时钟运行时，对 TMR1H 或 TMR1L 的读操作将确保为有效读操作（由硬件实现）。但是，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。

22.6 Timer1 门控

Timer1 可配置为自由计数或用 Timer1 门控电路使能和禁止计数。这也称为 Timer1 门控使能。

Timer1 门控也可由多个可选择源驱动。

22.6.1 TIMER1 门控使能

通过将 T1GCON 寄存器的 GE 位置 1 使能 Timer1 门控使能模式。使用 T1GCON 寄存器的 GPOL 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 22-3。

表 22-3：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

22.6.2 TIMER1 门控源选择

表22-4列出了Timer1门控源选择。源的选择由T1GCON寄存器的T1GSS位控制。每个可用源的极性也是可选择的。极性的选择由T1GCON寄存器的T1GPOL位控制。

表 22-4: 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 sync_C1OUT (可选择 Timer1 同步输出)
11	比较器 2 的输出 sync_C2OUT (可选择 Timer1 同步输出)

22.6.2.1 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。

22.6.2.2 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。

22.6.2.3 比较器 C1 门控操作

比较器 1 操作产生的输出可以选择作为 Timer1 的门控源。比较器 1 输出 (sync_C1OUT) 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 19.4.1 节“比较器输出同步”。

22.6.2.4 比较器 C2 门控操作

比较器 2 操作产生的输出可以选择作为 Timer1 的门控源。比较器 2 输出 (sync_C2OUT) 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 19.4.1 节“比较器输出同步”。

22.6.3 TIMER1 门控翻转模式

使能 Timer1 门控翻转模式时，可测量 Timer1 门控信号整个周期的长度，而不是单电平脉冲的持续时间。

Timer1 门控源经由一个触发器输送到 Timer1，该单稳态触发器在信号的每个递增边沿改变状态。时序详细信息请参见图 22-4。

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。

22.6.4 TIMER1 门控单脉冲模式

使能 Timer1 门控单脉冲模式时，可能会捕捉到一个单脉冲门控事件。Timer1 门控单脉冲模式首先通过将 T1GCON 寄存器中的 T1GSPM 位置 1 来使能。接下来必须将 T1GCON 寄存器中的 T1GGO/DONE 位置 1。Timer1 将在下一个递增边沿完全使能。在脉冲的下一个边沿，将自动清零 T1GGO/DONE 位。不允许其他门控事件递增 Timer1，直到 T1GGO/DONE 位再次由软件置 1。时序详细信息请参见图 22-5。

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 门控源的周期时间。时序详细信息请参见图 22-6。

22.6.5 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。

22.6.6 TIMER1 门控事件中断

允许 Timer1 门控事件中断时，可在门控事件完成时产生一个中断。出现 T1GVAL 的下降沿时，PIR1 寄存器中的 TMR1GIF 标志位将置 1。如果 PIE1 寄存器中的 TMR1GIE 位置 1，则会识别出一个中断。

即使 Timer1 门控未使能 (TMR1GE 位清零)，TMR1GIF 标志位也能工作。

22.7 Timer1 中断

Timer1 寄存器对 (TMR1H:TMR1L) 递增到 FFFFh，然后计满返回到 0000h。当 Timer1 计满返回时，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 ON 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。

22.8 休眠期间的 Timer1 操作

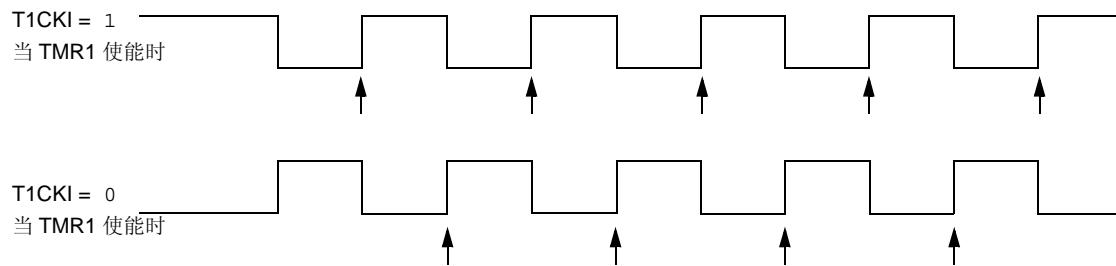
只有在设置为异步计数器模式时，Timer1 才能在休眠模式下工作。在该模式下，可使用外部晶振或时钟源使计数器递增计数。要设置定时器以唤醒器件：

- 必须将 T1CON 寄存器的 ON 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 SYNC 位置 1
- 必须配置 T1CON 寄存器的 CS 位
- 必须配置 T1CON 寄存器的 OSCEN 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 SYNC 位的设置如何，辅助振荡器都会在休眠模式下继续工作。

图 22-2: TIMER1 递增边沿



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

22.9 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 24.0 节“捕捉/比较/PWM 模块”。

22.10 CCP 自动转换触发信号

当将任一 CCP 配置为触发自动转换时，触发信号将清零 TMR1H:TMR1L 寄存器对。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 Fosc/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 的写操作和来自 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 24.2.1 节“自动转换触发器”。

图 22-3: TIMER1 门控使能模式

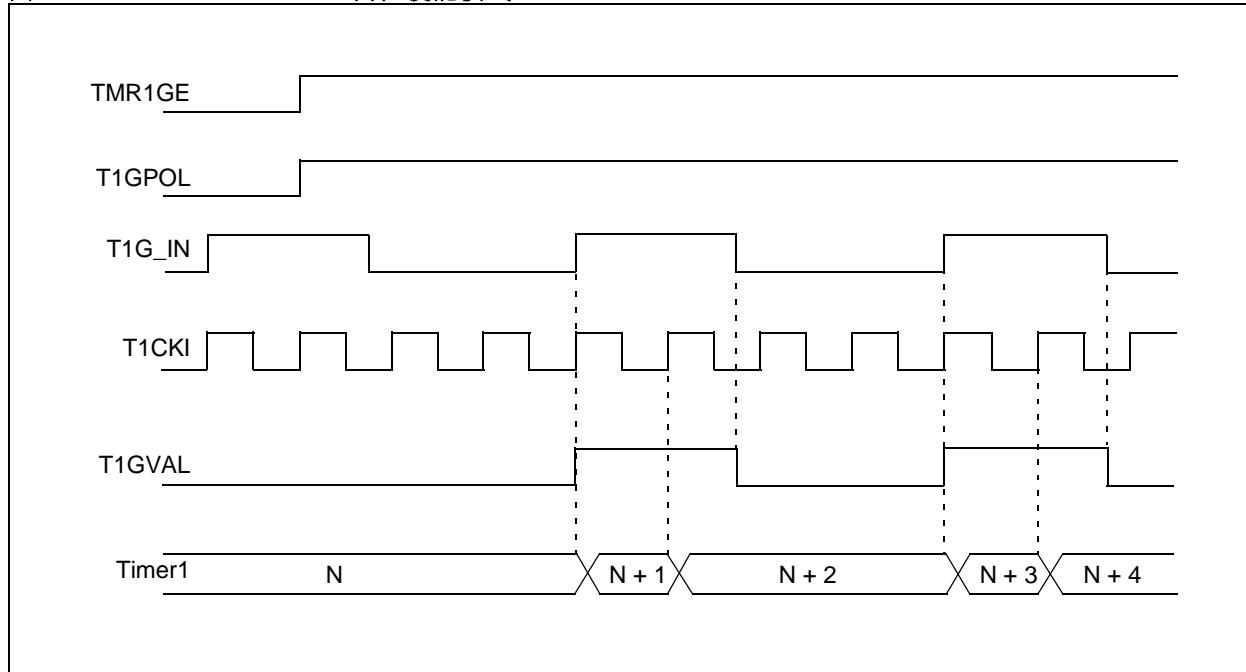
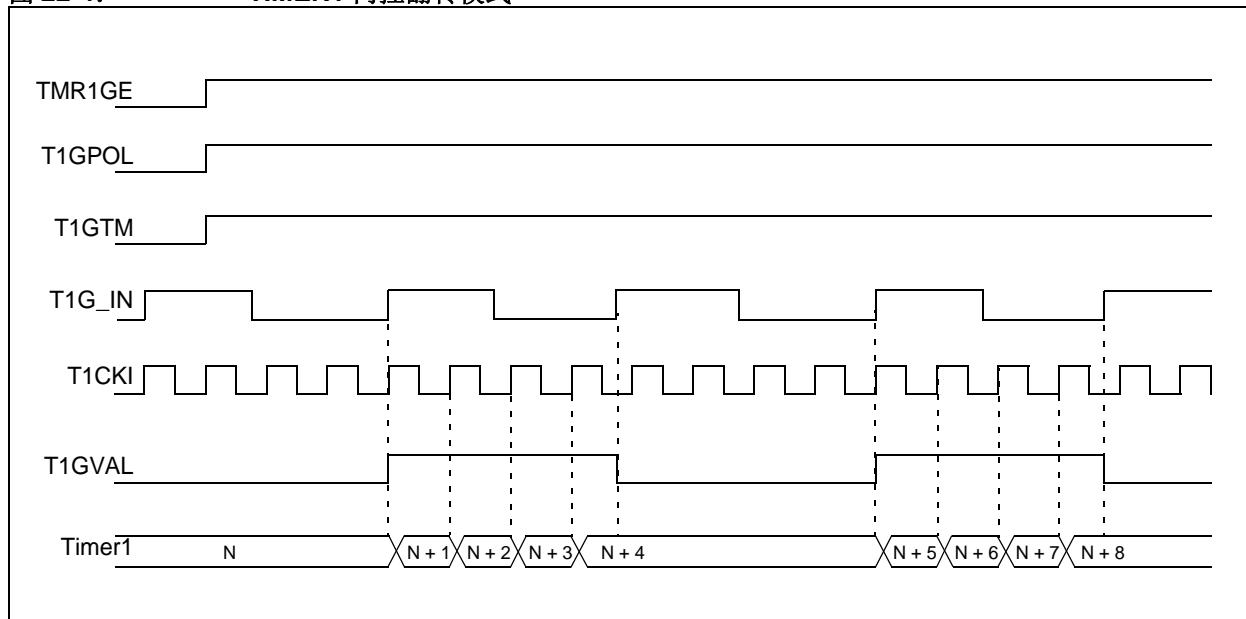


图 22-4: TIMER1 门控翻转模式



PIC16(L)F1764/5/8/9

图 22-5: TIMER1 门控单脉冲模式

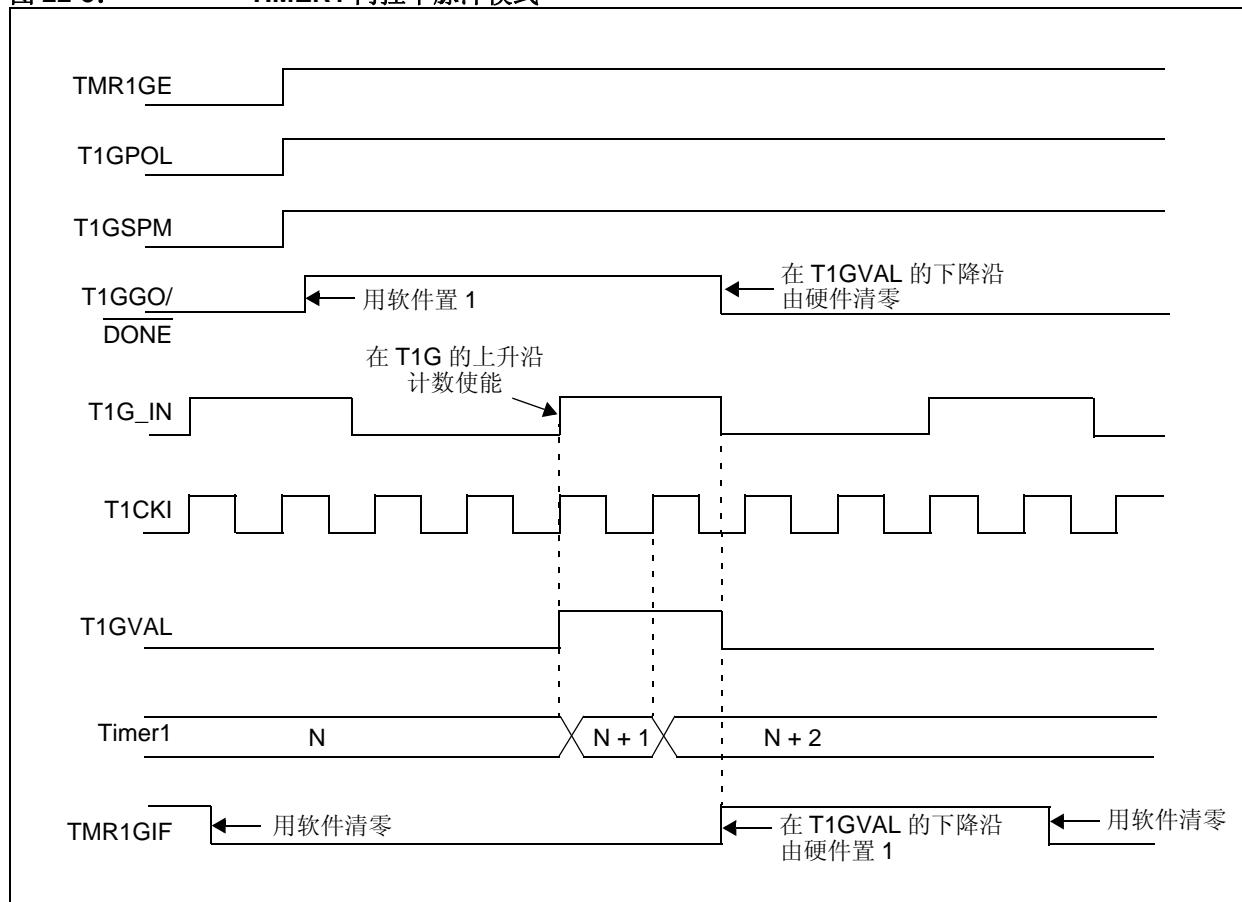
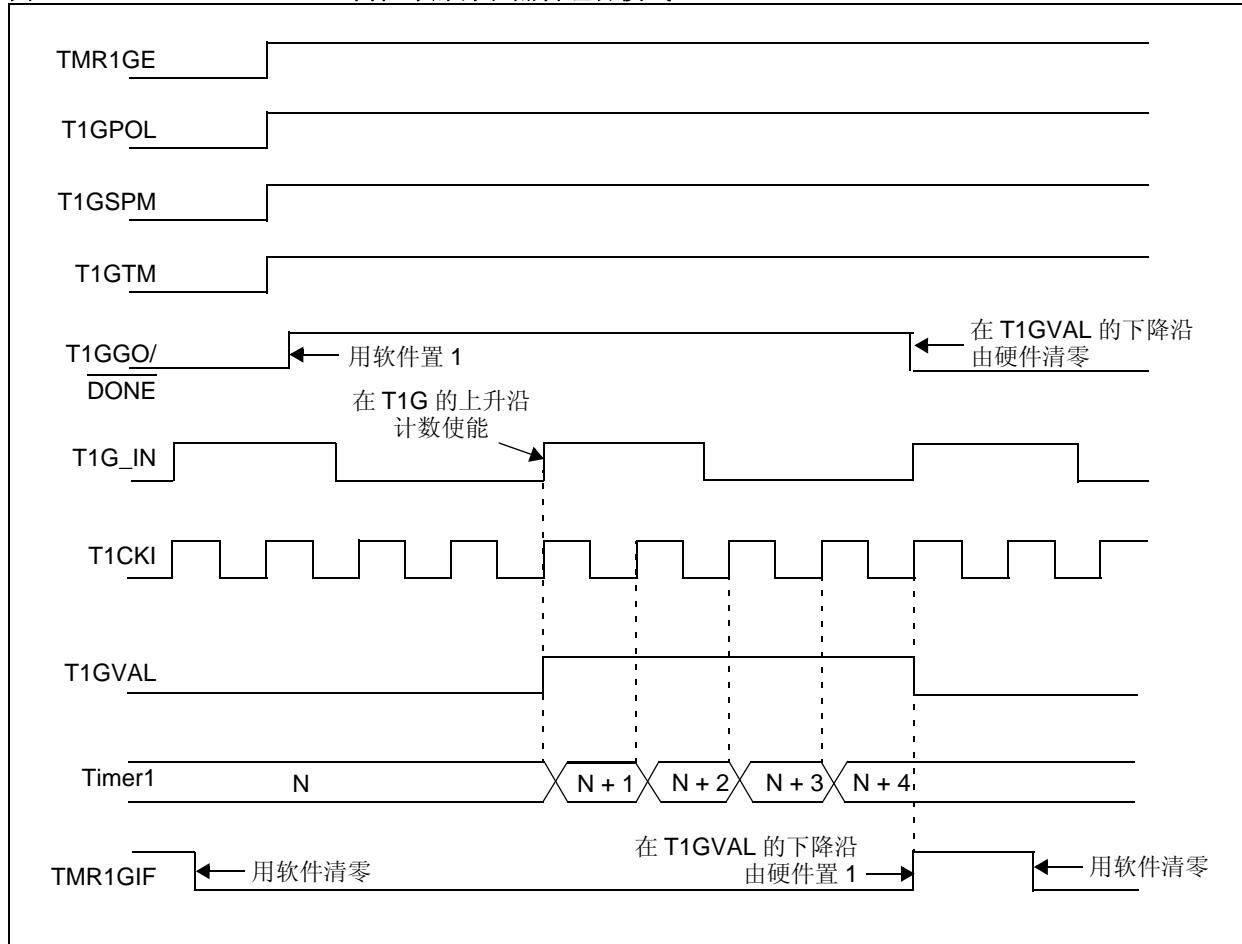


图 22-6: TIMER1 门控单脉冲和翻转组合模式



PIC16(L)F1764/5/8/9

22.11 寄存器定义: Timer1 控制

表 22-5 列出了 Timer1 外设的长位名称前缀。更多信息，请参见第 1.1.2.2 节“长位名称”。

表 22-5:

外设	位名称前缀
Timer1	T1
Timer3	T3
Timer5	T5

寄存器 22-1: T1CON: TIMER1 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
CS<1:0>		CKPS<1:0>		OSCEN	SYNC	—	ON
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 **CS<1:0>**: Timer1 时钟源选择位

11 = 保留, 不要使用。

10 = Timer1 时钟源为引脚或振荡器: (1)

如果 T1OSCEN = 0:

来自 T1CKI 引脚的外部时钟 (上升沿触发计数)

如果 T1OSCEN = 1:

SOSCI/SOSCO 引脚上的晶振

01 = Timer1 时钟源为系统时钟 (Fosc)

00 = Timer1 时钟源为指令时钟 (Fosc/4)

bit 5-4

CKPS<1:0>: Timer1 输入时钟预分频比选择位

11 = 1:8 预分频比

10 = 1:4 预分频比

01 = 1:2 预分频比

00 = 1:1 预分频比

bit 3

OSCEN: LP 振荡器使能控制位 (1)

1 = 使能专用的辅助振荡器电路

0 = 禁止专用的辅助振荡器电路

bit 2

SYNC: Timer1 同步控制位

1 = 不同步异步时钟输入

0 = 将异步时钟输入与系统时钟 (Fosc) 同步

bit 1

未实现: 读为 0

bit 0

ON: Timer1 使能位

1 = 使能 Timer1

0 = 停止 Timer1 并清零 Timer1 门控触发器

注 1: 仅用于 Timer1。保留, 不用于 Timer3 和 Timer5。

寄存器 22-2: T1GCON: TIMER1 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
GE	GPOL	GTM	GSPM	GGO/ DONE	GVAL	GSS<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 | GE: Timer1 门控使能位
<u>如果 TMR1ON = 0:</u>
该位被忽略
<u>如果 TMR1ON = 1:</u>
1 = Timer1 计数由 Timer1 门控功能控制
0 = Timer1 计数与 Timer1 门控功能无关 |
| bit 6 | GPOL: Timer1 门控极性位
1 = Timer1 门控为高电平有效 (当门控信号为高电平时 Timer1 计数)
0 = Timer1 门控为低电平有效 (当门控信号为低电平时 Timer1 计数) |
| bit 5 | GTM: Timer1 门控翻转模式位
1 = 使能 Timer1 门控翻转模式
0 = 禁止 Timer1 门控翻转模式并清零 T 触发器的输出
Timer1 门控触发器在每个上升沿翻转。 |
| bit 4 | GSPM: Timer1 门控单脉冲模式位
1 = 使能 Timer1 门控单脉冲模式, 控制 Timer1 门控
0 = 禁止 Timer1 门控单脉冲模式 |
| bit 3 | GGO/DONE: Timer1 门控单脉冲采集状态位
1 = Timer1 门控单脉冲采集就绪, 正在等待一个边沿
0 = Timer1 门控单脉冲采集已经结束或尚未开始 |
| bit 2 | GVAL: Timer1 门控值状态位
指示可提供给 TMR1H:TMR1L 的 Timer1 门控信号的当前状态。
不受 Timer1 门控使能 (TMR1GE) 的影响 |
| bit 1-0 | GSS<1:0>: Timer1 门控源选择位
11 = 比较器 2 的可选同步输出 (<code>sync_C2OUT</code>)
10 = 比较器 1 的可选同步输出 (<code>sync_C1OUT</code>)
01 = Timer0 溢出输出
00 = Timer1 门控引脚 |

PIC16(L)F1764/5/8/9

表 22-6：与 TIMER1/3/5 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
ANSEL _A	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
CCPxCON	EN	OE	OUT	FMT		MODE<3:0>			259
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
TMRxH	16 位 TMR1/3/5 寄存器高字节的保持寄存器								220*
TMRxL	16 位 TMR1/3/5 寄存器低字节的保持寄存器								220*
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
TxCON	CS<1:0>		CKPS<1:0>		OSCEN	SYNC	—	1ON	228
TxGCON	GE	GPOL	GTM	GSPM	GGO/ DONE	GVAL	GSS<1:0>		229

图注：— = 未实现位，读为 0。Timer1/3/5 模块不使用阴影单元。

* 提供寄存器信息的页。

注 1: 未实现位，读为 1。

2: 仅限 PIC16(L)F1768/9。

23.0 TIMER2/4/6 模块

Timer2/4/6 模块是 8 位定时器，可以作为自由运行周期计数器工作，也可以在单事件和单稳态工作模式下与控制启动、运行、冻结和复位操作的外部信号配合工作。通过将这些定时器与诸如比较器和 CCP 模块之类的其他内部外设配合工作，可以实现诸如脉冲密度调制之类的复杂波形控制。定时器的特性包括：

- 8 位定时器寄存器
- 8 位周期寄存器
- 可选的外部硬件定时器复位
- 可编程的预分频器（分频比为 1:1 至 1:128）
- 可编程的后分频器（分频比为 1:1 至 1:16）
- 可选的同步 / 异步操作
- 备用时钟源
- 周期中断

三种工作模式：

- 自由运行周期
- 单事件
- 单稳态

Timer2 框图请参见图 23-1。时钟源块框图请参见图 23-2。

注：该器件上实现了 3 个相同的 Timer2 模块。这些定时器命名为 Timer2、Timer4 和 Timer6。对 Timer2 的所有引用均适用于 Timer4 和 Timer6。对 T2PR 的所有引用也均适用于 T4PR 和 T6PR。

图 23-1：TIMER2 框图

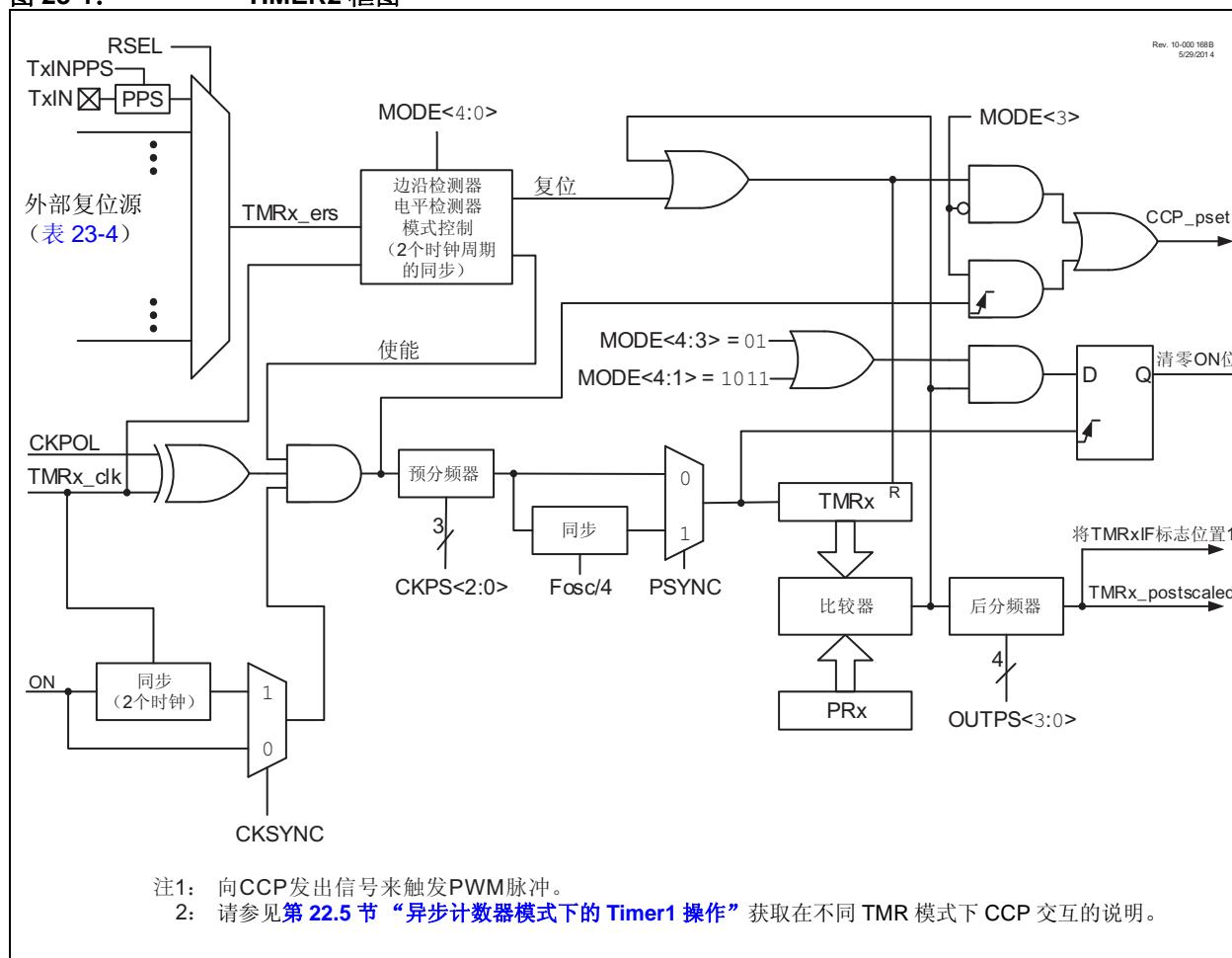
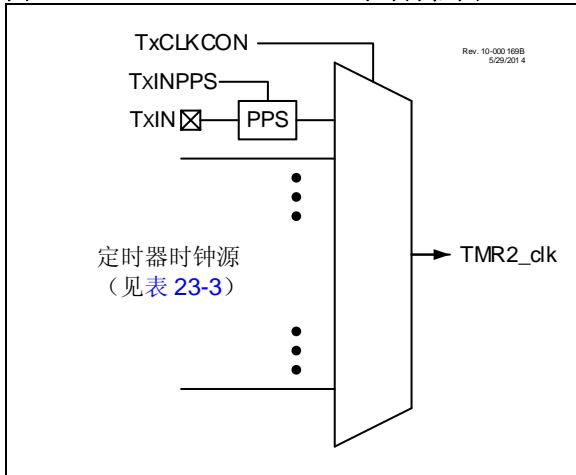


图 23-2: TIMER2 时钟源框图



23.1 Timer2 工作原理

Timer2 具有 3 种主要工作模式：

- 自由运行周期
- 单事件
- 单稳态

在每种模式内，都有一些用于启动、停止和复位的选项。表 23-1 列出了这些选项。

在所有模式下，TMR2 计数寄存器都是在来自可编程预分频器的时钟信号的上升沿发生递增。当 TMR2 等于 T2PR 时，会向后分频器计数器输出高电平。TMR2 在下一个时钟输入清零。

此外，还可以配置来自硬件的外部信号，对定时器操作进行门控或强制产生 TMR2 计数复位。在门控模式下，计数器在门控被禁止时停止，在门控被使能时恢复。在复位模式下，TMR2 计数在外部源出现任一电平或边沿时复位。

TMR2 和 T2PR 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 T2PR 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
- 任何器件复位
- 使定时器复位的外部复位源事件。

注： 写 T2CON 时 TMR2 不会清零。

23.1.1 自由运行周期模式

在每个时钟周期，TMR2 的值都会与周期寄存器 T2PR 中的值进行比较。当两个值匹配时，比较器会在下一个周期将 TMR2 的值复位为 00h，并递增输出后分频器计数器。当后分频器计数等于 TMRxCON1 寄存器的

OUTPS<4:0> 位的值时，会在 TMR2_postscaled 输出上产生一个时钟周期宽的脉冲，并清零后分频器计数。

23.1.2 单事件模式

单事件模式与自由运行周期模式基本相同，只是在 TMR2 与 T2PR 匹配时会清零 ON 位并停止定时器，并且只有在关闭并开启 T2ON 位之后才会重新启动定时器。在该模式下，后分频器的 OUTPS<4:0> 值不为 0 是没有意义的，因为定时器会在发生第一个周期事件时停止，而后分频器会在定时器重新启动时复位。

23.1.3 单稳态模式

单稳态模式类似于单事件模式，只是 ON 位不会被清零，定时器可以通过外部复位事件重新启动。

23.2 Timer2 输出

Timer2 模块的主输出为 TMR2_postscaled，它在后分频器计数器与 TMRxCON 寄存器的 OUTPS 位中的值匹配时，产生单个 TMR2_clk 周期的脉冲。T2PR 后分频器在每次 TMR2 值与 T2PR 值匹配时发生递增。可以选择该信号作为几个其他输入模块的输入：

- ADC 模块，作为自动转换触发信号
- COG，作为自动关断源

此外，CCP 模块在 PWM 模式下也会使用 Timer2 来进行脉冲生成。实际 TMR2 值及其他内部信号都会被发送到 CCP 模块，以正确地控制 PWM 信号的周期和脉宽。关于设置 Timer2 来用于 CCP 的更多详细信息，请参见第 24.6 节“CCP/PWM 时钟选择”；关于不同 Timer2 模式如何影响 CCP PWM 输出的示例，请参见第 23.5 节“操作示例”中的时序图。

23.3 外部复位源

除了时钟源之外，Timer2 还可接受外部复位源。Timer2、Timer4 和 Timer6 的这种外部复位源分别使用 T2RST、T4RST 和 T6RST 寄存器进行选择。该复位源可以控制定时器的启动和停止，以及定时器的复位，具体取决于定时器所处的模式。定时器的模式由 TMRxHLT 寄存器的 MODE<4:0> 位控制。边沿触发模式要求在外部触发信号之间有 6 个定时器时钟周期。电平触发模式要求触发电平至少为 3 个定时器时钟周期长。在调试冻结模式下，外部触发信号会被忽略。

表 23-1: TIMER2 工作模式

模式	MODE<4:0>		输出操作	操作	定时器控制		
	<4:3>	<2:0>			启动	复位	停止
自由运行周期	00	000	周期脉冲	软件门控 (图 23-4)	ON = 1	—	ON = 0
		001		硬件门控, 高电平有效 (图 23-5)	ON = 1, TMRx_ers = 1	—	ON = 0 或 TMRx_ers = 0
		010		硬件门控, 低电平有效	ON = 1, TMRx_ers = 0	—	ON = 0 或 TMRx_ers = 1
		011	周期脉冲与硬件复位	上升沿或下降沿复位	ON = 1	TMRx_ers ↓	ON = 0
		100		上升沿复位 (图 23-6)		TMRx_ers ↑	
		101		下降沿复位		TMRx_ers ↓	
		110		低电平复位		TMRx_ers = 0	ON = 0 或 TMRx_ers = 0
		111		高电平复位 (图 23-7)		TMRx_ers = 1	ON = 0 或 TMRx_ers = 1
单事件	01	000	单事件	软件启动 (图 23-8)	ON = 1	—	ON = 0 或 TMRx = PRx 之后的下一个时钟 (注 2)
		001	边沿触发启动 (注 1)	上升沿启动 (图 23-9)	ON = 1, TMRx_ers ↑	—	
		010		下降沿启动	ON = 1, TMRx_ers ↓	—	
		011		任意边沿启动	ON = 1, TMRx_ers ↓	—	
		100	边沿触发启动和硬件复位 (注 1)	上升沿启动和上升沿复位 (图 23-10)	ON = 1, TMRx_ers ↑	TMRx_ers ↑	
		101		下降沿启动和下降沿复位	ON = 1, TMRx_ers ↓	TMRx_ers ↓	
		110		上升沿启动和低电平复位 (图 23-11)	ON = 1, TMRx_ers ↑	TMRx_ers = 0	
		111		下降沿启动和高电平复位	ON = 1, TMRx_ers ↓	TMRx_ers = 1	
单稳态	10	000	保留				
		001	边沿触发启动 (注 1)	上升沿启动 (图 23-12)	ON = 1, TMRx_ers ↑	—	ON = 0 或 TMRx = PRx 之后的下一个时钟 (注 3)
		010		下降沿启动	ON = 1, TMRx_ers ↓	—	
		011		任意边沿启动	ON = 1, TMRx_ers ↓	—	
		100	保留				
保留		101	保留				
单事件	11	110	电平触发启动和硬件复位	高电平启动和低电平复位 (图 23-13)	ON = 1, TMRx_ers = 1	TMRx_ers = 0	ON = 0 或 保持复位状态 (注 2)
		111		低电平启动和高电平复位	ON = 1, TMRx_ers = 0	TMRx_ers = 1	
保留	11	xxx	保留				

注 1: 如果 ON = 0, 则在 ON = 1 后需要一个边沿来重新启动定时器。

2: 当 TMRx = PRx 时, 在下一个时钟清零 ON, 并将 TMRx 停在 00h。

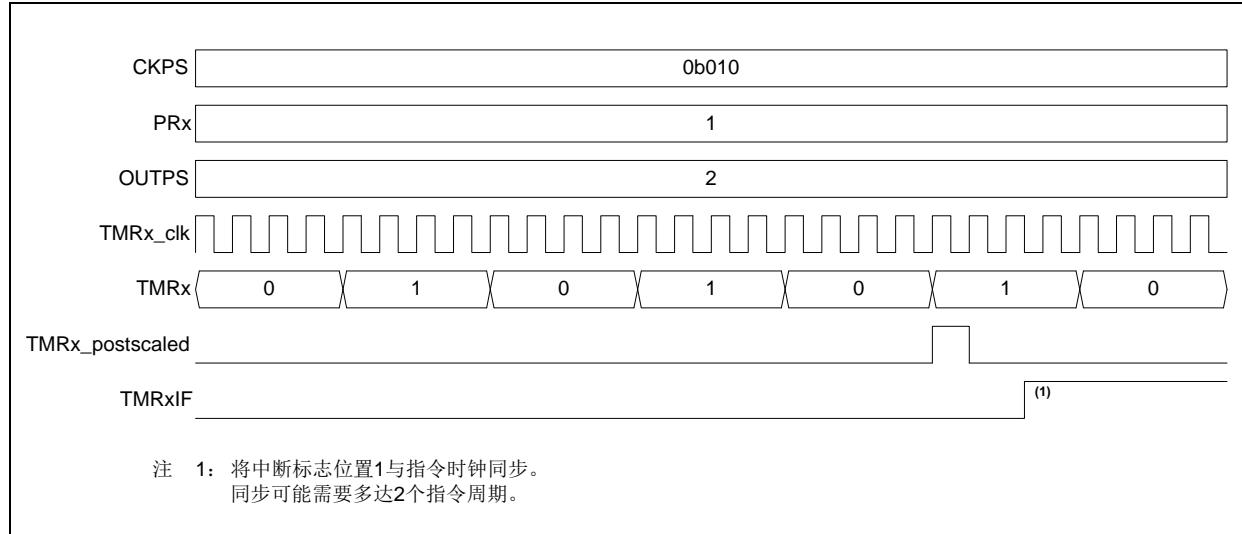
3: 当 TMRx = PRx 时, 在下一个时钟将 TMRx 停在 00h, 但不清零 ON。

PIC16F1764/5/8/9

23.4 Timer2 中断

Timer2 也可以产生器件中断。当后分频器计数器与 16 个后分频选项 (1:1 至 1:16) 之一匹配时，会产生中断；后分频选项使用 T2CON 寄存器的后分频控制位 OUTPS<3:0> 进行选择。可以通过将 PIE1 寄存器的 TMR2IE 中断允许位置 1 来允许该中断。中断时序如图 23-3 所示。

图 23-3: TIMER2 预分频器、后分频器和中断时序图



23.5 操作示例

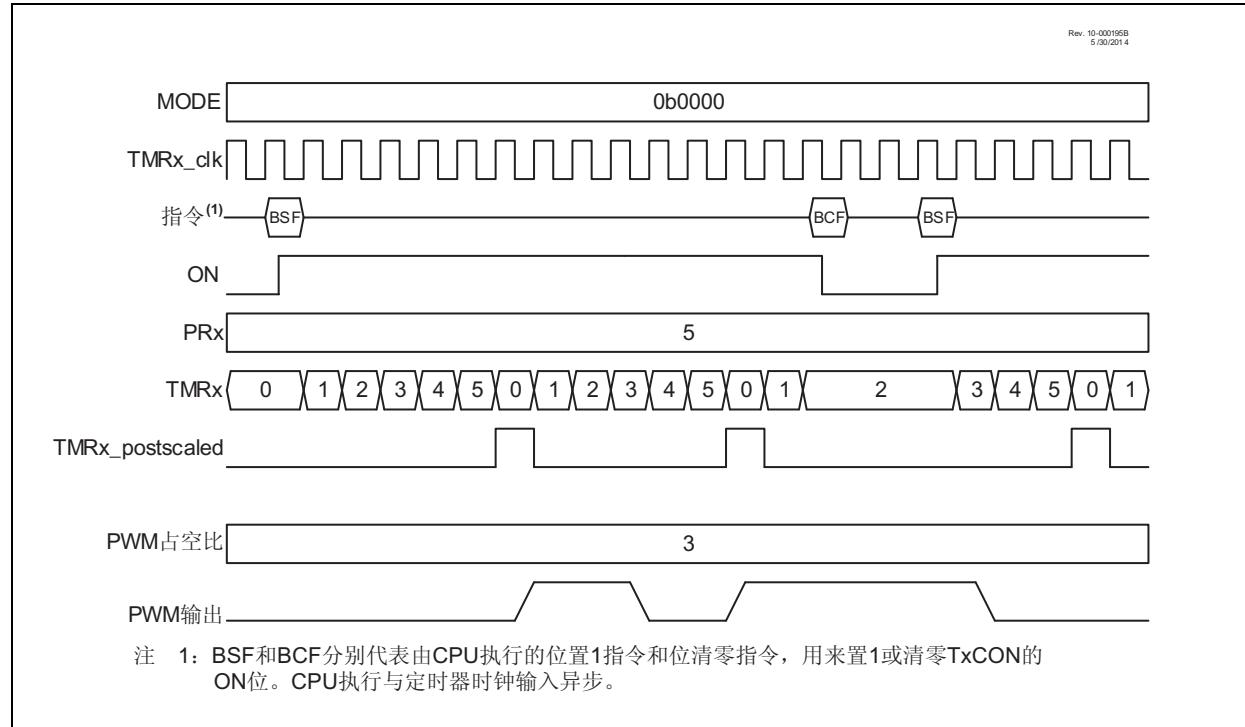
除非另有说明，否则以下说明适用于后面的时序图：

- 预分频比和后分频比均设置为 1:1（TxCON 寄存器中的 CKPS 和 OUTPS 位均清零）。
- 这些时序图说明的是使用除 Fosc/4 之外任何其他时钟时的操作，对于 ON 和 Timer2_ers 显示了至少两个完整周期的时钟同步延时。使用 Fosc/4 时，Timer2_ers 的时钟同步延时至少为 1 个指令周期；ON 在下一个指令周期应用。
- 图中显示了 PWM 占空比和 PWM 输出，即假定定时器用于 CCP 模块的 PWM 功能，如[第 24.6 节“CCP/PWM 时钟选择”](#)所述。这些信号不属于 Timer2 模块的一部分。

23.5.1 软件门控模式

该模式对应于传统 Timer2 操作。当 ON = 1 时，定时器会随每个时钟输入发生递增；当 ON = 0 时，它不会发生递增。当 TMRx 计数等于 PRx 周期计数时，定时器会在下一个时钟发生复位，并继续从 0 开始计数。[图 23-4](#) 显示了 ON 位由软件控制时的操作。PRx = 5 时，计数器会一直递增至 TMRx = 5，并在下一个时钟变为 0。

图 23-4： 软件门控模式时序图（MODE = 00000）



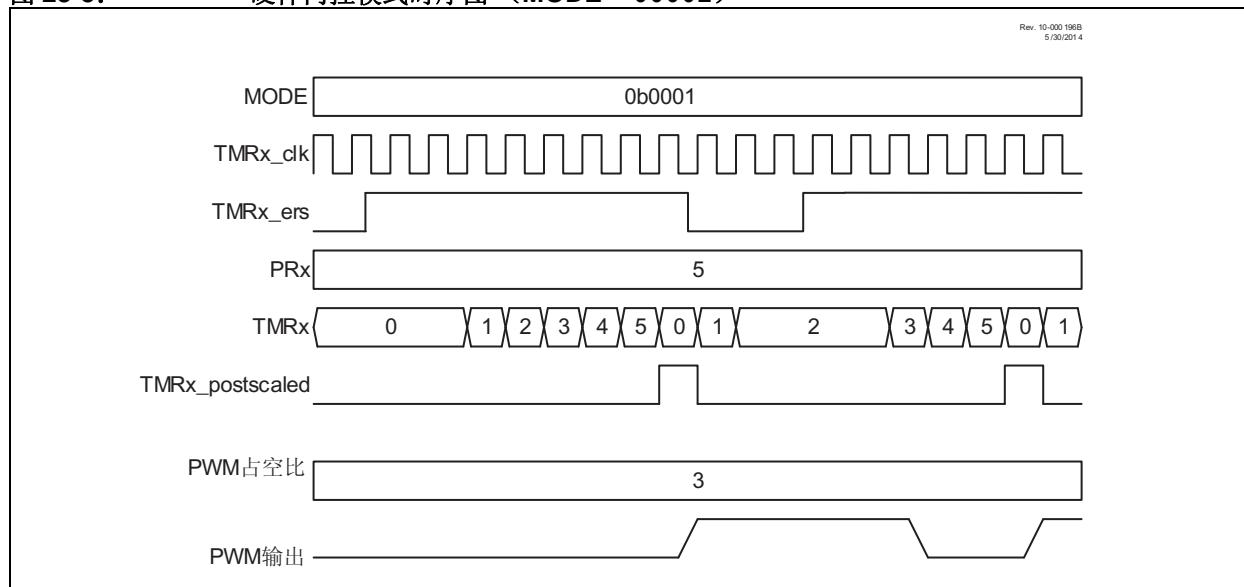
23.5.2 硬件门控模式

硬件门控模式的工作方式与软件门控模式基本相同，只是 `TMRx_ers` 外部信号也可以对定时器进行门控。与 CCP 配合使用时，门控会延长 PWM 周期。如果定时器在 PWM 输出为高电平时停止，则也会增大占空比。

当 `MODE<4:0> = 00001` 时，定时器在外部信号为高电平时停止。当 `MODE<4:0> = 00010` 时，定时器在外部信号为低电平时停止。

图 23-5 显示了 `MODE<4:0> = 00001` 时的硬件门控模式，在该模式下使用高输入电平来启动计数器。

图 23-5：硬件门控模式时序图（`MODE = 00001`）



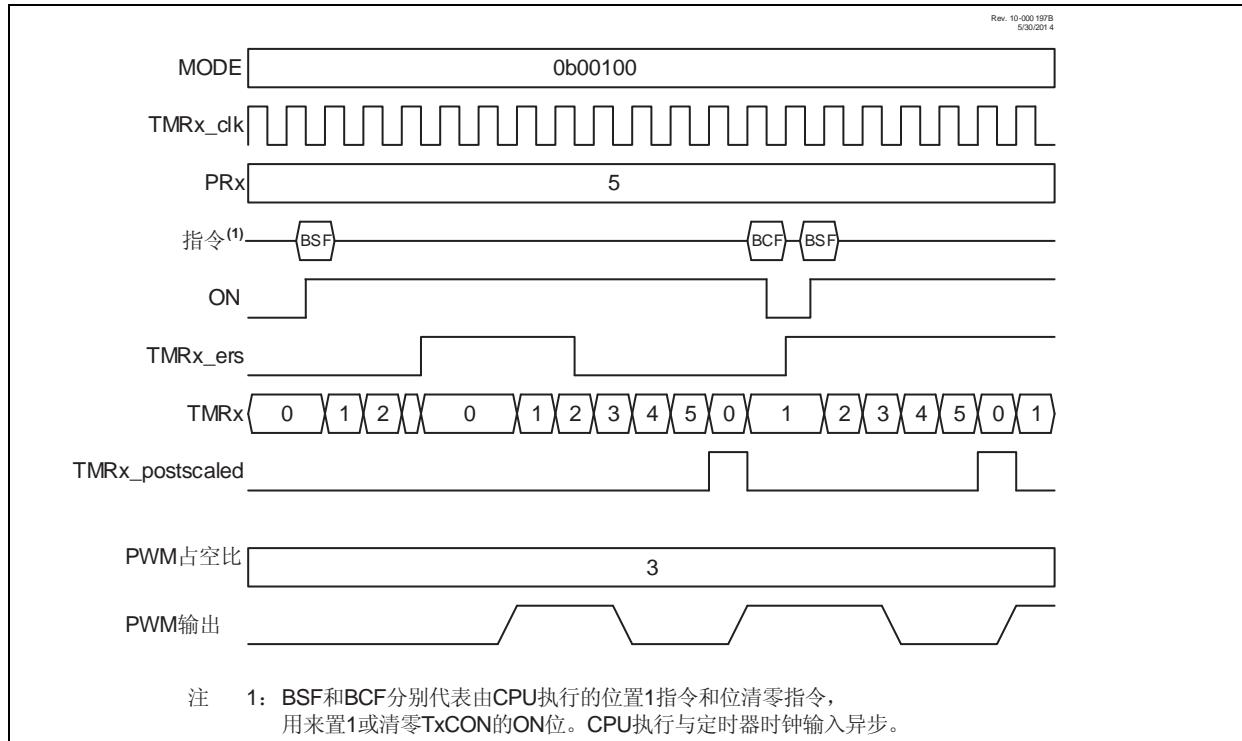
23.5.3 边沿触发硬件限制模式

在硬件限制模式下，可以在定时器达到周期计数之前通过 TMRx_ers 外部信号复位定时器。支持3种类型的复位：

- 在上升沿或下降沿复位（MODE<4:0> = 00011）
- 在上升沿复位（MODE<4:0> = 00100）
- 在下降沿复位（MODE<4:0> = 00101）

当定时器与 PWM 模式下的 CCP 配合使用时，提前复位会缩短周期，并在两个时钟的延时之后重新启动 PWM 脉冲。请参见图 23-6。

图 23-6：边沿触发硬件限制模式时序图（MODE = 00100）



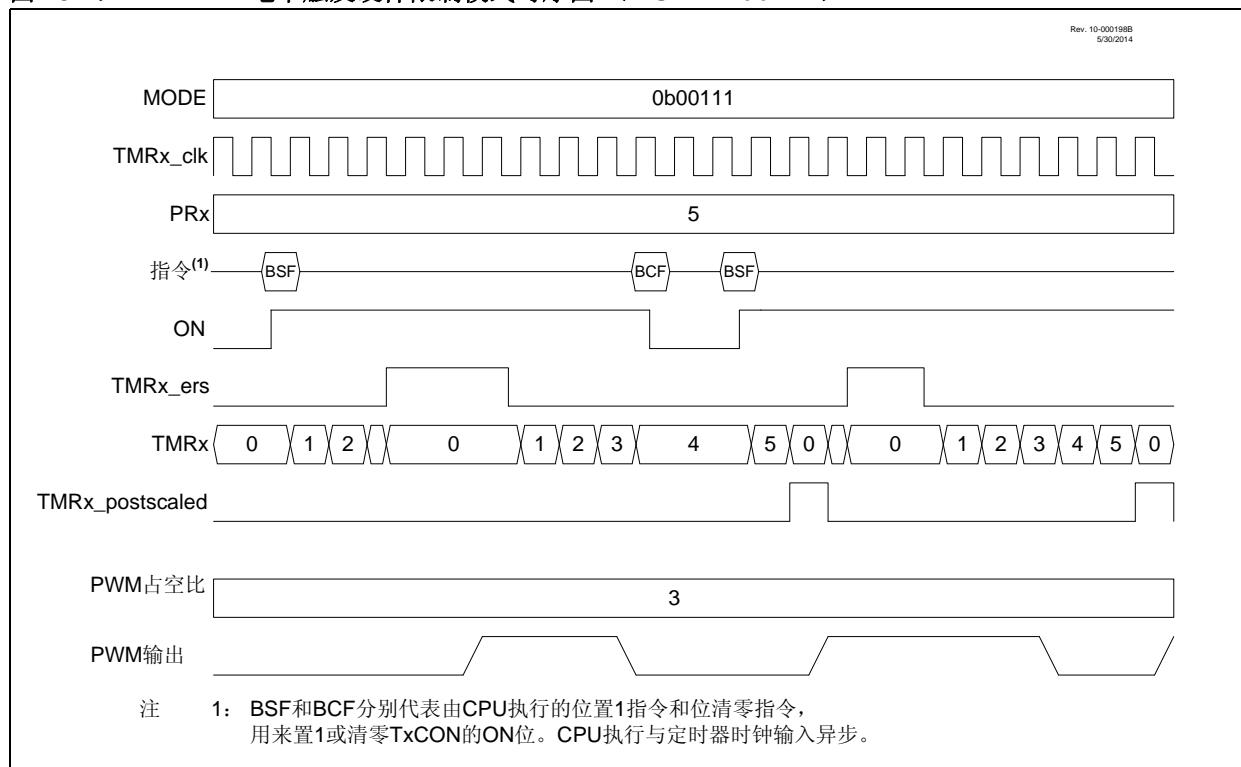
23.5.4 电平触发硬件限制模式

在电平触发硬件限制定时器模式下，计数器通过外部信号 TMRx_ers 的高电平或低电平进行复位，如图 23-7 所示。选择 MODE<4:0> = 00110 会导致定时器在出现低电平外部信号时复位。选择 MODE<4:0> = 00111 会导致定时器在出现高电平外部信号时复位。在示例中，计数器在 TMRx_ers = 1 时复位。ON 由 BSF 和 BCF 指令控制。当 ON = 0 时，外部信号会被忽略。

当 CCP 使用定时器作为 PWM 时基时，PWM 输出将在定时器开始计数时设置为高电平，之后仅在定时器计数与CCPRx值匹配时设置为低电平。在定时器计数与PRx值匹配时或在外部复位信号变为真并保持为真的两个时钟周期之后，定时器发生复位。

在发生PRx匹配的下一个时钟或在外部复位信号释放复位的两个时钟后，定时器开始计数，PWM输出设置为高电平。PWM输出将一直保持高电平，直到定时器递增计数至与CCPRx脉宽值匹配为止。如果外部复位信号在PWM输出为高电平时变为真，则PWM输出将一直保持高电平，直到复位信号被释放，使定时器可以递增计数至与CCPRx值匹配为止。

图 23-7：电平触发硬件限制模式时序图（MODE = 00111）

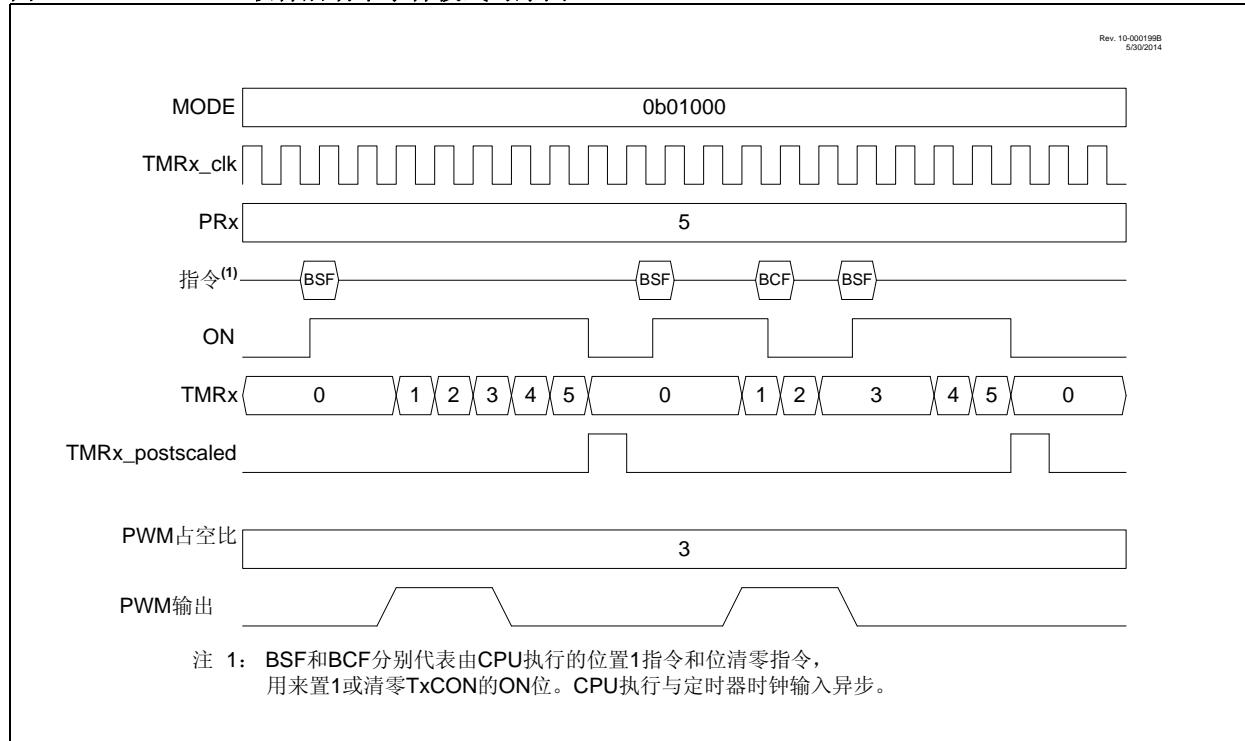


23.5.5 软件启动单事件模式

在单事件模式下，当定时器值与 PR_x 周期值匹配时，定时器发生复位，ON 位清零。要启动另一个定时器周期，必须用软件将 ON 位置 1。设置 MODE<4:0> = 01000 将选择单事件模式，如图 23-8 所示。在示例中，ON 由 BSF 和 BCF 指令控制。在第一种情形中，BSF 指令将 ON 置 1，计数器运行至完成并清零 ON。在第二种情形中，BSF 指令启动周期，BCF/BSF 指令在该周期期间关闭并开启计数器，然后它运行至完成。

当单事件模式与 CCP PWM 操作配合使用时，PWM 脉冲驱动会随 ON 位置 1 而同时启动。在 PWM 驱动处于有效状态时清零 ON 位会延长 PWM 驱动。PWM 驱动将在定时器值与 CCPRx 脉宽值匹配时终止。PWM 驱动将保持关闭，直到软件通过将 ON 位置 1 而启动另一个周期为止。如果软件在 CCPRx 匹配之后但在 PR_x 匹配之前清零 ON 位，则 PWM 驱动会被延长，延长量等于 ON 位保持清零状态的时间长度。只有在 ON 位由 PR_x 周期计数匹配事件清零之后，才能通过将 ON 位置 1 来启动另一个计时周期。

图 23-8：软件启动单事件模式时序图（MODE = 01000）



23.5.6 边沿触发单事件模式

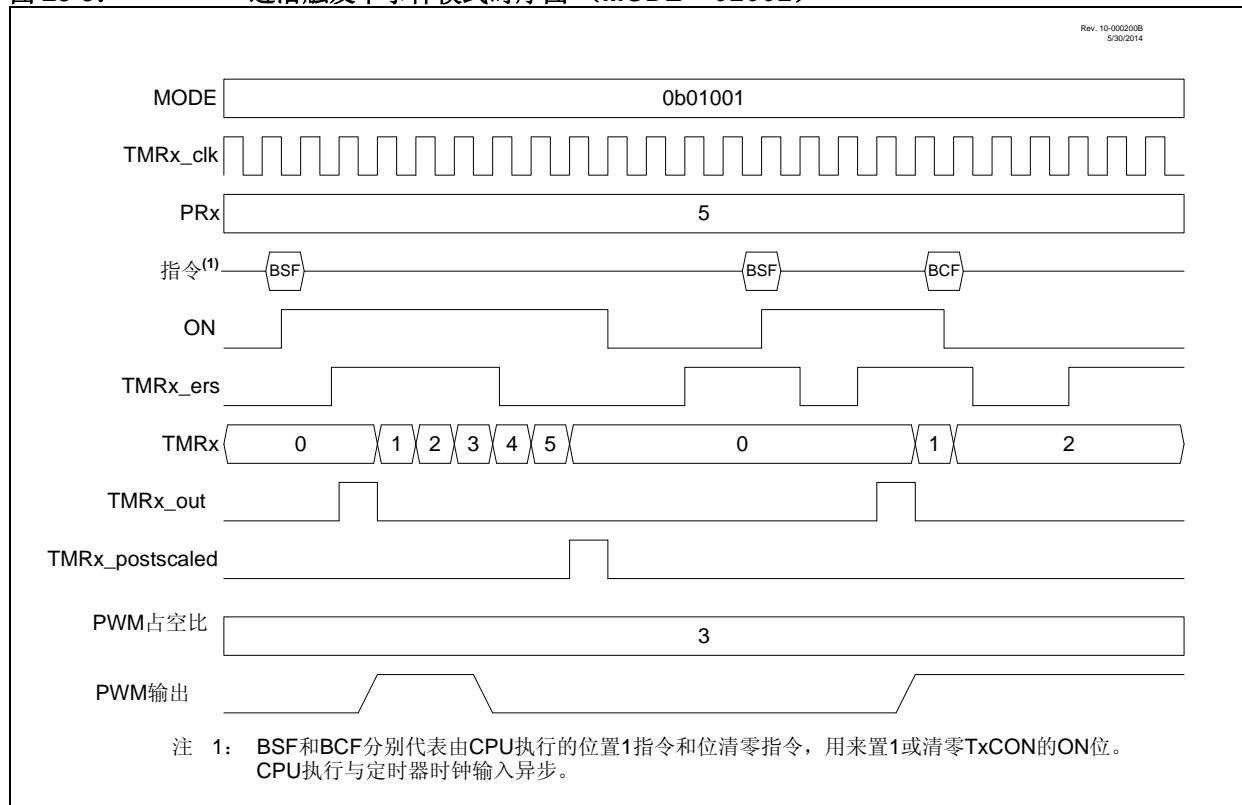
在 ON 位置 1 之后，边沿触发单事件模式会在外部信号输入出现边沿时启动定时器，在定时器与 PR_x 周期值匹配时清零 ON 位。以下边沿将启动定时器：

- 上升沿 (MODE<4:0> = 01001)
- 下降沿 (MODE<4:0> = 01010)
- 上升沿或下降沿 (MODE<4:0> = 01011)

如果定时器由于 ON 位清零而暂停，则在 ON 位置 1 后需要另一个 TMR_{x_ers} 边沿来恢复计数。图 23-9 显示了上升沿单事件模式下的操作。

当边沿触发单事件模式与 CCP 配合使用时，边沿触发信号会激活 PWM 驱动，PWM 驱动将在定时器与 CCPRx 脉宽值匹配时变为无效，并在定时器由于 PR_x 周期计数匹配而暂停时保持无效。

图 23-9： 边沿触发单事件模式时序图 (MODE = 01001)



23.5.7 边沿触发硬件限制单事件模式

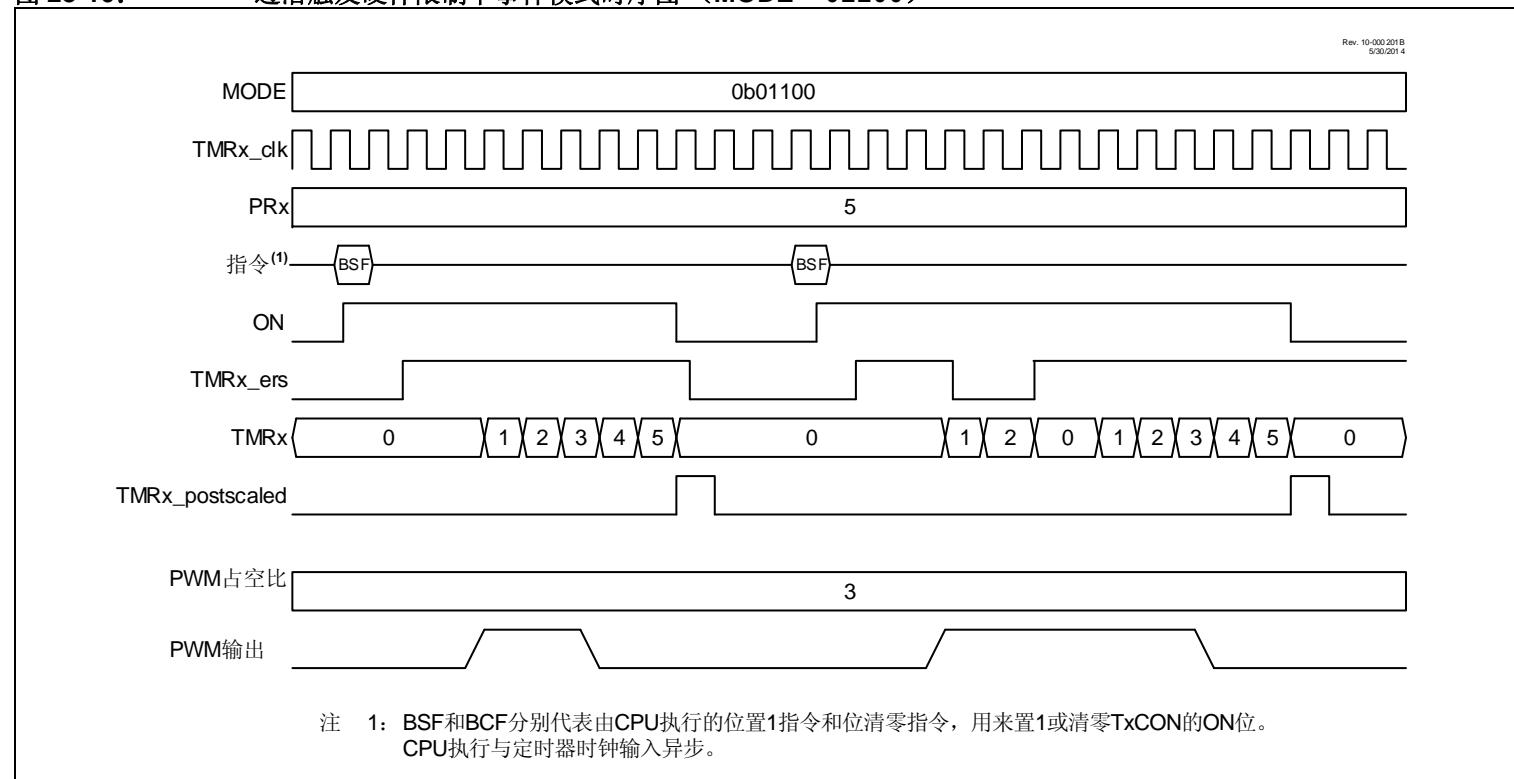
在边沿触发硬件限制单事件模式下，定时器在 ON 位置 1 后出现第一个外部信号边沿时启动，并在出现所有后续边沿时复位。在 ON 位置 1 后，只需要通过第一个边沿来启动定时器。计数器将在所有后续外部复位边沿的两个时钟后自动恢复计数。边沿触发信号如下：

- 上升沿启动和复位 (MODE<4:0> = 01100)
- 下降沿启动和复位 (MODE<4:0> = 01101)

当定时器值与 PR_x 周期值匹配时，定时器会发生复位，并清零 ON 位。只有在软件将 ON 位置 1 之后，外部信号边沿才会产生作用。图 23-10 显示了上升沿硬件限制单事件操作。

当该模式与 CCP 配合使用时，第一个启动边沿会触发 PWM 驱动，所有后续复位边沿会激活 PWM 驱动。除非外部信号边沿在发生匹配之前使定时器复位，否则在定时器与 CCPRx 脉宽值匹配时，PWM 驱动会变为无效，并保持无效直到定时器由于 PR_x 周期匹配而暂停。

图 23-10： 边沿触发硬件限制单事件模式时序图 (MODE = 01100)



23.5.8 电平复位边沿触发硬件限制单事件模式

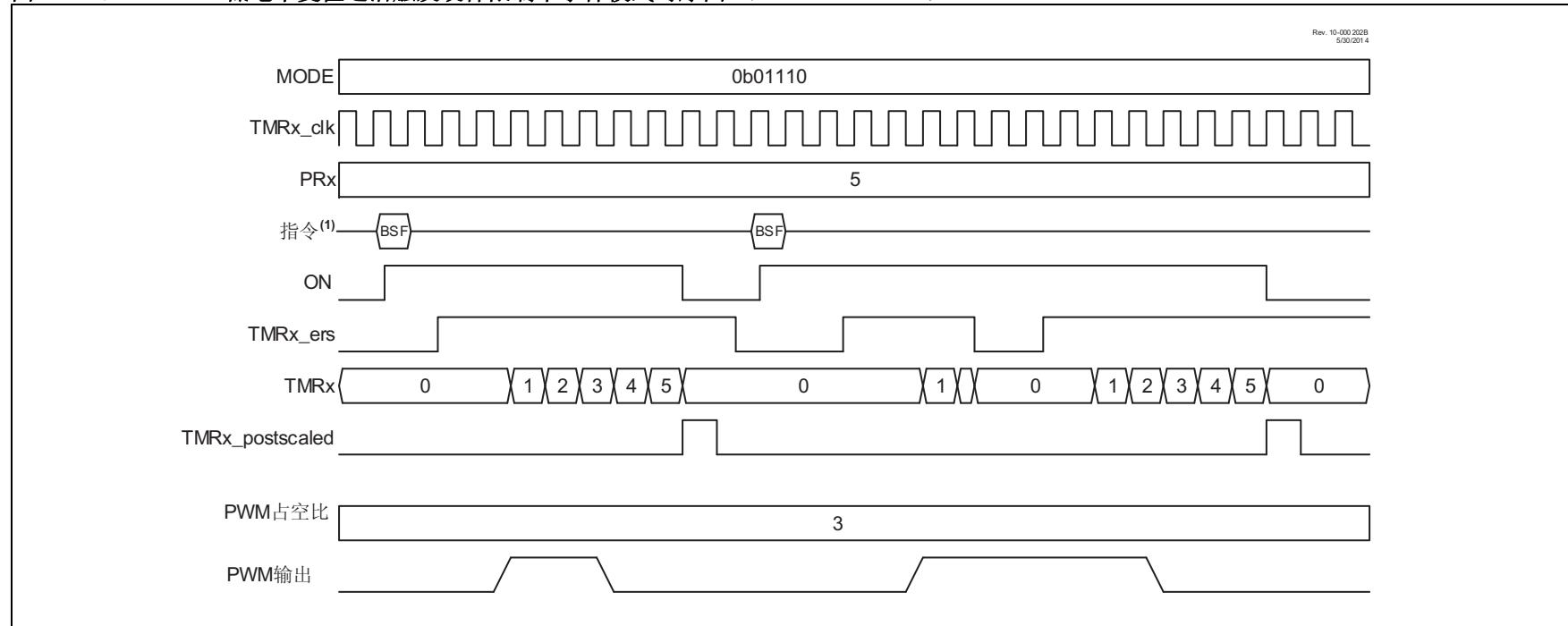
在电平复位边沿触发单事件模式下，当 ON 位置 1 时，定时器计数在出现外部信号电平时复位，在从复位电平变为有效电平的上升沿 / 下降沿开始计数。可选择的复位电平如下：

- 低复位电平 (MODE<4:0> = 01110)
- 高复位电平 (MODE<4:0> = 01111)

当定时器计数与 PRx 周期计数匹配时，定时器会发生复位，ON 位会被清零。当 ON 位由于发生 PRx 匹配或在软件控制下清零时，在 ON 位置 1 之后需要一个新的外部信号边沿来启动计数器。

当电平复位边沿触发单事件模式与 CCP PWM 操作配合使用时，PWM 驱动会随启动定时器的外部信号边沿变为有效。PWM 驱动会在定时器计数等于 CCPRx 脉宽计数时变为无效。当定时器计数由于 PRx 周期计数匹配而清零时，PWM 驱动不会变为有效。

图 23-11：低电平复位边沿触发硬件限制单事件模式时序图 (MODE = 01110)

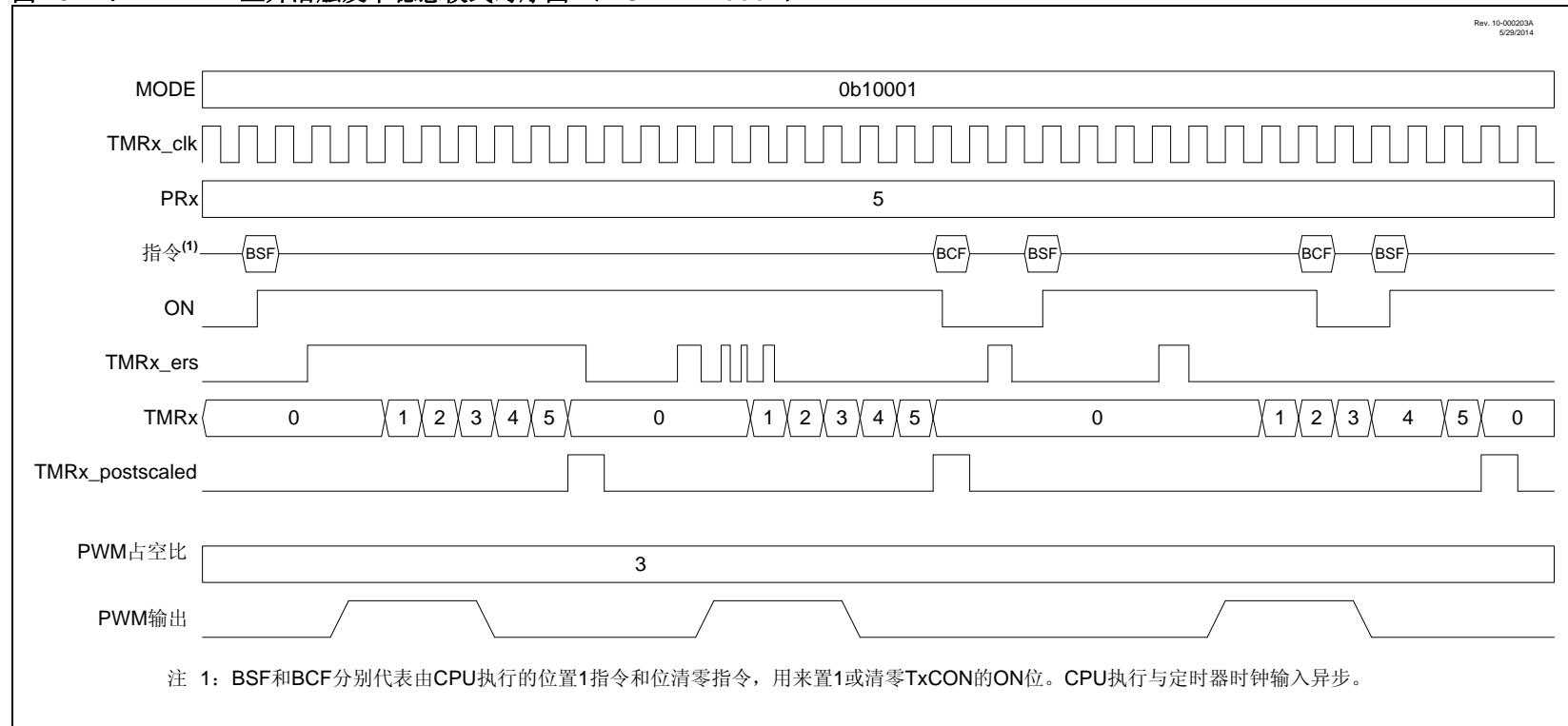


23.5.9 边沿触发单稳态模式

在 ON 位置 1 之后，边沿触发单稳态模式会在外部复位信号输入出现边沿时启动定时器，在定时器与 PR_x 周期值匹配时停止递增定时器。以下边沿将启动定时器：

- 上升沿 (MODE<4:0> = 10001)
- 下降沿 (MODE<4:0> = 10010)
- 上升沿或下降沿 (MODE<4:0> = 10011)

图 23-12：上升沿触发单稳态模式时序图 (MODE = 10001)



当边沿触发单稳态模式与 CCP PWM 操作配合使用时，PWM 驱动会随启动定时器的外部复位信号边沿而变为有效，但不会在定时器与 PR_x 值匹配时变为有效。当定时器在递增时，外部复位信号上的额外边沿不会影响 CCP PWM。

23.5.10 电平触发硬件限制单事件模式

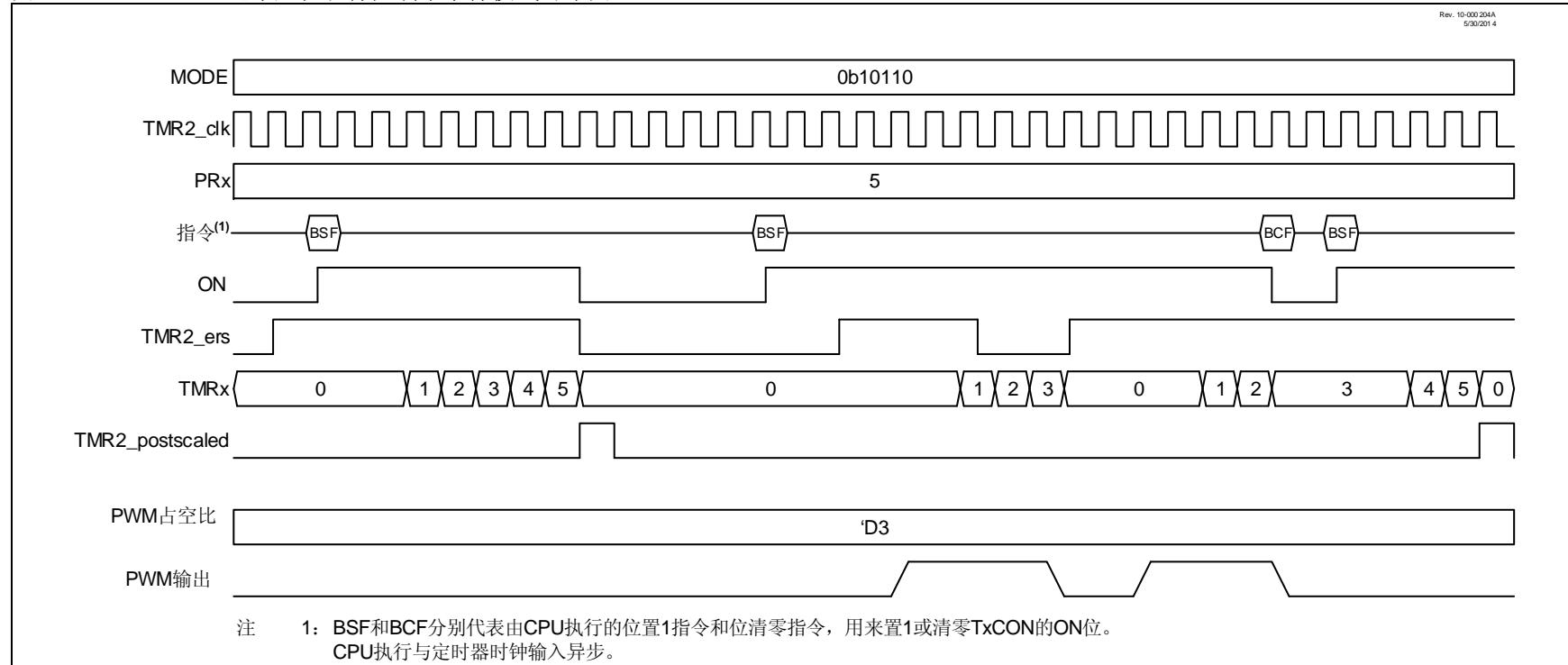
电平触发硬件限制单事件模式会在出现外部复位电平时将定时器保持在复位状态，并在 ON 位置 1 且外部信号不为复位电平时启动计数。如果存在外部信号不为复位电平或 ON 位置 1 这两者中的一种条件，则在另一信号置 1/ 变为有效时会启动定时器。可选择的复位电平如下：

- 低复位电平 (MODE<4:0> = 10110)
- 高复位电平 (MODE<4:0> = 10111)

当定时器计数与 PRx 周期计数匹配时，定时器会发生复位，ON 位会被清零。当 ON 位由于发生 PRx 匹配或在软件控制下清零时，定时器将保持复位状态，直到 ON 位置 1 且外部信号不为复位电平为止。

当电平触发硬件限制单事件模式与 CCP PWM 操作配合使用时，PWM 驱动会在出现外部信号边沿或 ON 位置 1 (启动定时器的两个条件之一) 时变为有效。

图 23-13：电平触发硬件限制单事件模式时序图 (MODE = 10110)



23.6 休眠期间的 Timer2 操作

当 PSYNC = 1 时，在处理器处于休眠模式时，Timer2 无法工作。在处理器处于休眠模式时，TMR2 和 T2PR 寄存器的内容将保持不变。

当 PSYNC = 0 时，只要所选择的时钟源仍在运行，Timer2 就会在休眠模式下工作。选择 LFINTOSC、MFINTOSC 或 HFINTOSC 振荡器作为定时器时钟源会使选定的振荡器在休眠期间保持运行。

PIC16F1764/5/8/9

23.7 寄存器定义: Timer2/4/6 控制

表 23-2 列出了 Timer2/4/6 外设的长位名称前缀。更多信息, 请参见第 1.1.2.2 节 “长位名称”。

表 23-2:

外设	位名称前缀
Timer2	T2
Timer4	T4
Timer6	T6

寄存器 23-1: TxCLKCON: TIMERx 时钟选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CS<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 CS<3:0>: Timerx 时钟选择位

请参见表 23-3。

表 23-3: TIMERx 时钟源

CS<3:0>	Timer2	Timer4	Timer6
1011-1111	保留	保留	保留
1010	LC3_out	LC3_out	LC3_out
1001	LC2_out	LC2_out	LC2_out
1000	LC1_out	LC1_out	LC1_out
0111	ZCD_out	ZCD_out	ZCD_out
0110	SOSC	SOSC	SOSC
0101	MFINTOSC	MFINTOSC	MFINTOSC
0100	LFINTOSC	LFINTOSC	LFINTOSC
0011	HFINTOSC	HFINTOSC	HFINTOSC
0010	Fosc	Fosc	Fosc
0001	Fosc/4	Fosc/4	Fosc/4
0000	通过 T2INPPS 选择的引脚	通过 T4INPPS 选择的引脚	通过 T6INPPS 选择的引脚

寄存器 23-2: TxCON: TIMERx 控制寄存器

R/W/HC-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
ON ⁽¹⁾		CKPS<2:0>			OUTPS<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

HC = 硬件清零位

bit 7 **ON:** Timerx 使能位

1 = 开启 Timerx

0 = 关闭 Timerx: 复位所有计数器和状态机

bit 6-4 **CKPS<2:0>:** Timer2 型时钟预分频比选择位

111 = 1:128 预分频比

110 = 1:64 预分频比

101 = 1:32 预分频比

100 = 1:16 预分频比

011 = 1:8 预分频比

010 = 1:4 预分频比

001 = 1:2 预分频比

000 = 1:1 预分频比

bit 3-0 **OUTPS<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

注 1: 在某些模式下, ON 位将由硬件自动清零。请参见[第 23.5 节 “操作示例”](#)。

寄存器 23-3: TxHLT: TIMERx 硬件限制控制寄存器

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
PSYNC ^(1, 2)	CKPOL ⁽³⁾	CKSYNC ^(4, 5)			MODE<4:0> ^(6, 7)		
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 **PSYNC:** Timerx 预分频器同步使能位 ^(1, 2)

1 = TMRx 预分频器输出与 Fosc/4 进行同步
0 = TMRx 预分频器输出不与 Fosc/4 进行同步

bit 6 **CKPOL:** Timerx 时钟极性选择位 ⁽³⁾

1 = 定时器 / 预分频器基于输入时钟的下降沿进行计数
0 = 定时器 / 预分频器基于输入时钟的上升沿进行计数

bit 5 **CKSYNC:** Timerx 时钟同步使能位 ^(4, 5)

1 = ON 寄存器位与 TMR2_clk 输入进行同步
0 = ON 寄存器位不与 TMR2_clk 输入进行同步

bit 4-0 **MODE<4:0>:** Timerx 控制模式选择位 ^(6, 7)

请参见表 23-1。

注 1: 置 1 该位可确保读取 TMRx 将返回一个有效的值。

2: 当该位为 1 时, Timer2 无法在休眠模式下工作。

3: 不应在 ON = 1 时更改 CKPOL。

4: 置 1 该位可确保在使能或禁止 ON 位时产生无毛刺的操作。

5: 当该位置 1 时, 定时器操作将在 ON 位置 1 后延迟两个 TMRx 输入时钟。

6: 除非另有说明, 否则所有模式均在 ON = 1 时启动, 在 ON = 0 时停止 (发生停止不会影响 TMRx 的值)。

7: 当 TMRx = PRx 时, 无论工作模式如何, 下一个时钟会清零 TMRx。

寄存器 23-4: TxRST: TIMERx 外部复位信号选择寄存器

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
—	—	—			RSEL<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 RSEL<4:0>: Timerx 外部复位信号源选择位

请参见表 23-4。

表 23-4: 外部复位源

RSEL<4:0>	Timer2	Timer4	Timer6
10010-11111	保留	保留	保留
10001	LC3_out	LC3_out	LC3_out
10000	LC2_out	LC2_out	LC2_out
01111	LC1_out	LC1_out	LC1_out
01110	ZCD_out	ZCD_out	ZCD_out
01101	sync_C4OUT ⁽¹⁾	sync_C4OUT ⁽¹⁾	sync_C4OUT ⁽¹⁾
01100	sync_C3OUT ⁽¹⁾	sync_C3OUT ⁽¹⁾	sync_C3OUT ⁽¹⁾
01011	sync_C2OUT	sync_C2OUT	sync_C2OUT
01010	sync_C1OUT	sync_C1OUT	sync_C1OUT
01001	PWM6_out ⁽¹⁾	PWM6_out ⁽¹⁾	PWM6_out ⁽¹⁾
01000	PWM5_out	PWM5_out	PWM5_out
00111	PWM4_out ⁽¹⁾	PWM4_out ⁽¹⁾	PWM4_out ⁽¹⁾
00110	PWM3_out	PWM3_out	PWM3_out
00101	CCP2_out ⁽¹⁾	CCP2_out ⁽¹⁾	CCP2_out ⁽¹⁾
00100	CCP1_out	CCP1_out	CCP1_out
00011	TMR6_postscaled	TMR6_postscaled	保留
00010	TMR4_postscaled	保留	TMR4_postscaled
00001	保留	TMR2_postscaled	TMR2_postscaled
00000	通过 T2INPPS 选择的引脚	通过 T4INPPS 选择的引脚	通过 T6INPPS 选择的引脚

注 1: 仅限 PIC16(L)F1768/9。

PIC16F1764/5/8/9

表 23-5：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									
CCP1CON	EN	OE	OUT	FMT	MODE<3:0>				259									
CCP2CON ⁽²⁾	EN	OE	OUT	FMT	MODE<3:0>				259									
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102									
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103									
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106									
T2PR	Timer2 模块周期寄存器								232*									
TMR2	8 位 TMR2 寄存器的保持寄存器								232*									
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				247									
T2CLKCON	—	—	—	—	CS<3:0>				246									
T2RST	—	—	—	—	RSEL<3:0>				249									
T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					248									
T4PR	Timer4 模块周期寄存器								232*									
TMR4	8 位 TMR4 寄存器的保持寄存器								232*									
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				247									
T4CLKCON	—	—	—	—	CS<3:0>				246									
T4RST	—	—	—	—	RSEL<3:0>				249									
T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					248									
T6PR	Timer6 模块周期寄存器								232*									
TMR6	8 位 TMR6 寄存器的保持寄存器								232*									
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				247									
T6CLKCON	—	—	—	—	CS<3:0>				246									
T6RST	—	—	—	—	RSEL<3:0>				249									
T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					248									

图注：— = 未实现位，读为 0。Timer2/4/6 模块不使用阴影单元。

* 提供寄存器信息的页。

注 1：仅限 PIC16(L)F1768/9。

24.0 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。

表 24-1： 可用的 CCP 模块

器件	CCP1	CCP2
PIC16(L)F1764	•	•
PIC16(L)F1765	•	•
PIC16(L)F1768	•	•
PIC16(L)F1769	•	•

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 模块相同的运行情况。

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 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。

24.1 捕捉模式

本节介绍的捕捉模式功能对于所有 CCP 模块均可用且相同。

捕捉模式使用 16 位 Timer1 资源。当 CCPx 输入上发生事件时，16 位 CCPRxH:CCPRxL 寄存器对会分别捕捉和存储 TMR1H:TMR1L 寄存器对的 16 位值。这些事件定义如下，可通过 CCPxCON 寄存器的 MODE<3:0> 位进行配置：

- 每个边沿（上升沿或下降沿）
- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

CCPx 捕捉输入信号通过 CCPxCAP 寄存器中的 CTS 位配置，具有以下选项：

- CCPx 引脚
- 比较器 1 输出（C1_OUT_sync）
- 比较器 2 输出（C2_OUT_sync）
- 比较器 3 输出（C3_OUT_sync）
- 比较器 4 输出（C4_OUT_sync）
- LC2_output
- LC3_output
- 电平变化中断触发（IOC_interrupt）

进行捕捉时，PIRx 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。

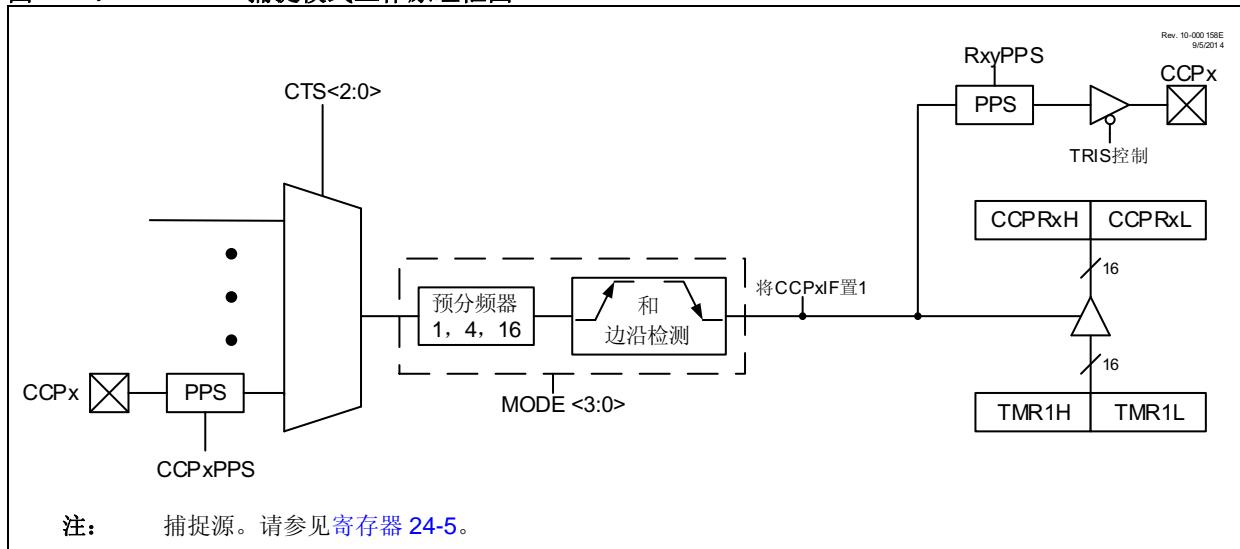
图 24-1 给出了捕捉操作的简化框图。

24.1.1 CCP 引脚配置

在捕捉模式下，使用 CCPxCAP 寄存器的 CTS 位来选择中断源。如果选择了 CCPx 引脚，则应通过将相关的 TRIS 控制位置 1，将该引脚配置为输入。

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。

图 24-1: 捕捉模式工作原理框图



24.1.2 TIMER1 模式资源

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

关于配置 Timer1 的更多信息, 请参见第 22.0 节“带门控的 Timer1/3/5 模块”。

24.1.3 软件中断模式

当捕捉模式改变时, 可能会产生错误的捕捉中断。用户应保持 PIE_x 寄存器的 CCP_xIE 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 PIR_x 寄存器的 CCP_xIF 中断标志位。

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

24.1.4 CCP 预分频器

通过 CCP_xCON 寄存器的 MODE<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 CCP_xCON 寄存器的 EN 位来关闭模块。

24.1.5 休眠期间的捕捉操作

捕捉模式的正常工作依赖于 Timer1 模块。有两个选项可用于在捕捉模式下驱动 Timer1 模块。它可由指令时钟 (Fosc/4) 驱动, 或由外部时钟源驱动。

当 Timer1 时钟源由 Fosc/4 提供时, Timer1 将不会在休眠期间递增。当器件被从休眠模式唤醒时, Timer1 将从先前状态继续工作。

当 Timer1 通过外部时钟源提供时钟时, 捕捉模式会在休眠模式期间继续工作。

24.1.6 备用引脚位置

该模块包含可通过使用 PPS 控制转移到其他位置的 I/O 引脚。更多详细信息, 请参见第 12.0 节“外设引脚选择 (PPS) 模块”。

24.1.7 捕捉输出

每当发生捕捉时, CCP 的输出会变为高电平, 并保持 1 个系统时钟周期 (1/Fosc)。该输出可用作以下外设的输入信号:

- ADC 触发
- COG
- PRG
- DSM
- CLC
- 运放改写
- Timer2/4/6 复位
- 任意器件引脚

此外, CCP 输出可以通过任意引脚的 PPS 控制输出到相应引脚。

24.2 比较模式

本节介绍的比较模式功能对于所有 CCP 模块均可用且相同。

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

- 翻转 CCPx 输出
- 将 CCPx 输出置 1
- 将 CCPx 输出清零
- 在 CCPx 上产生脉冲输出
- 产生软件中断
- 自动转换触发信号

引脚的动作由 CCPxCON 寄存器的 MODE<3:0> 控制位的值决定。同时，中断标志位 CCPxIF 置 1。

所有比较模式都能产生中断。

图 24-2 给出了比较操作的简化框图。

24.2.1 自动转换触发器

当选择自动转换触发模式 (CCPxM<3:0> = 1011) 时，CCPx 模块将进行以下操作：

- 复位 Timer1
- 如果 ADC 被使能，则启动 ADC 转换

在该模式下，CCPx 模块不会对 CCPx 引脚进行控制。

一旦 TMR1H 和 TMR1L 寄存器对与 CCPRxH 和 CCPRxL 寄存器对之间发生匹配，便会发生 CCP 的自动转换触发信号输出。TMR1H 和 TMR1L 寄存器对在 Timer1 时钟的下一个上升沿到来之前不会复位。自动转换触发信号输出也会启动一次 ADC 转换（如果 ADC 模块被使能）。这使 CCPRxH 和 CCPRxL 寄存器对实际上作为 Timer1 的 16 位可编程周期寄存器。

更多信息，请参见第 16.2.5 节“自动转换触发器”。

注 1: CCP 模块的自动转换触发信号不会将 PIR1 寄存器的中断标志位 TMR1IF 置 1。

2: 通过在产生自动转换触发信号的时钟边沿和使 Timer1 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。

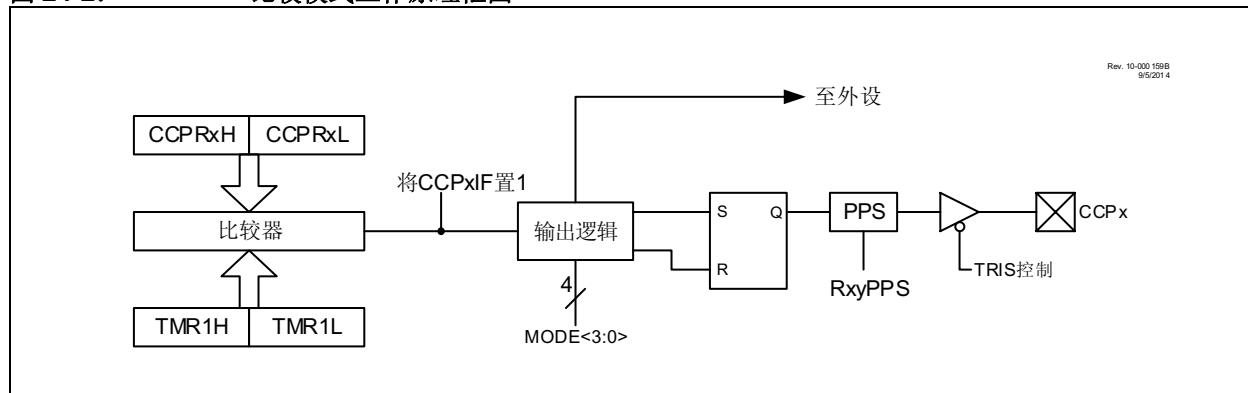
24.2.2 CCPx 引脚配置

用户必须通过将相关的 TRIS 位清零，将 CCPx 引脚配置为输出。

此外，还可以通过使用 PPS 控制将 CCPx 引脚功能转移到备用引脚上。更多详细信息，请参见第 12.0 节“外设引脚选择 (PPS) 模块”。

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。

图 24-2: 比较模式工作原理框图



24.2.3 TIMER1 模式资源

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

关于配置 Timer1 的更多信息，请参见[第 22.0 节 “带门控的 Timer1/3/5 模块”](#)。

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

24.2.4 软件中断模式

当选择产生软件中断模式（ $\text{MODE}\langle 3:0 \rangle = 1010$ ）时，CCPx 模块不会对 CCPx 引脚进行控制（见 CCPxCON 寄存器）。

24.2.5 休眠期间的比较操作

比较模式的正常工作依赖于系统时钟（ Fosc ）。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。

24.2.6 备用引脚位置

该模块包含可通过使用 PPS 控制转移到其他位置的 I/O 引脚。更多详细信息，请参见[第 12.0 节 “外设引脚选择（PPS）模块”](#)。

24.2.7 捕捉输出

在比较模式下，CCP 会在 CCPRxH:CCPRxL 寄存器对的 16 位值与 TMR1H:TMR1L 寄存器对匹配时提供输出。比较输出取决于为 CCP 配置的比较模式。如果 CCPxCON 寄存器的 MODE 位等于 1011 或 1010，CCP 模块将在 TMR1 等于 CCPRxH:CCPRxL 寄存器对时输出高电平。这意味着脉宽由 TMR1 预分频器决定。如果 CCPxCON 的 MODE 位等于 0001 或 0010，输出将在发生匹配时翻转，从 0 变为 1 或反之。如果 CCPxCON 的 MODE 位等于 1001，输出将在发生匹配时清零；如果 MODE 位等于 1000，输出将在发生匹配时置 1。该输出可供以下外设使用：

- ADC 触发器
- COG
- PRG
- DSM
- CLC
- 运放改写
- Timer2/4/6 复位
- 任意器件引脚

24.3 PWM 概述

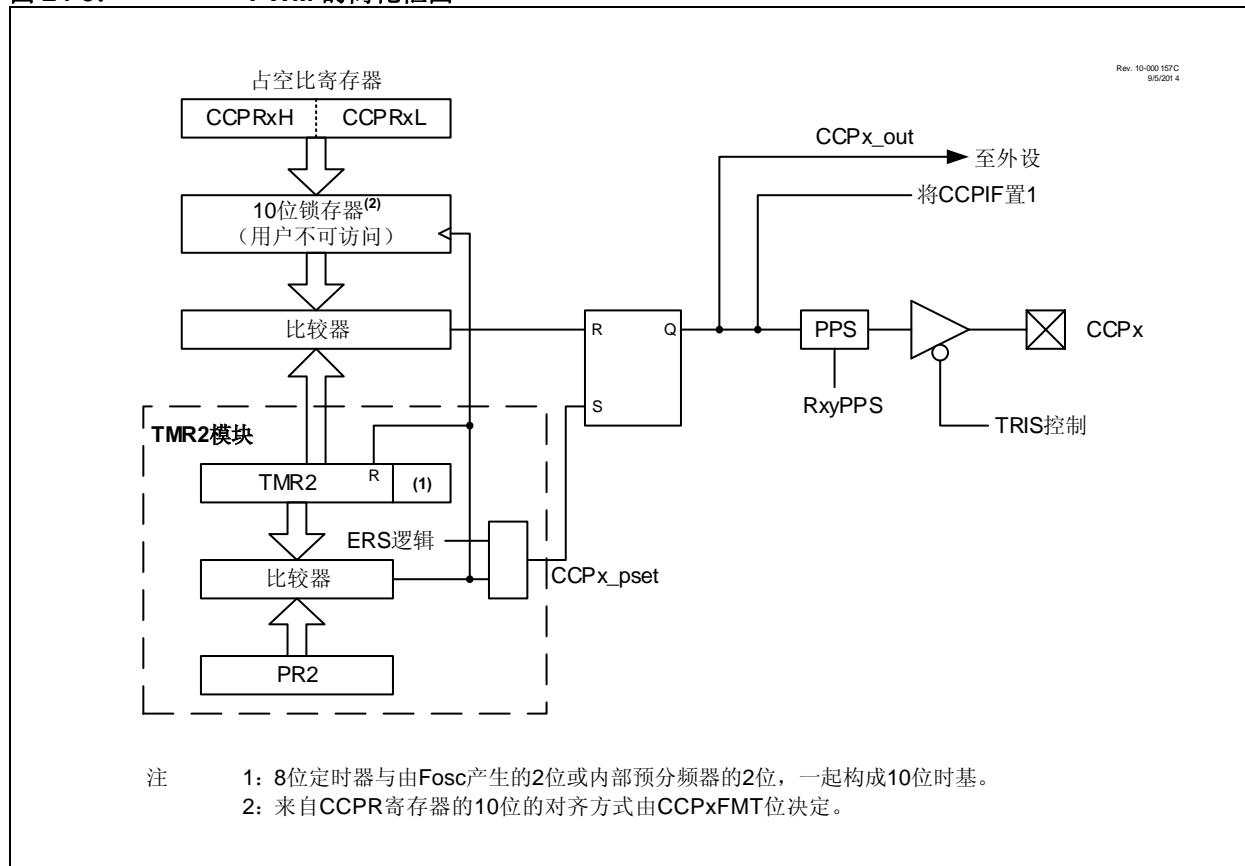
脉宽调制（PWM）是一种通过在完全开启和完全关闭状态之间进行快速切换而为负载供电的方案。PWM信号类似于方波，信号的高电平部分视为开启状态，信号的低电平部分视为关闭状态。高电平部分（也称为脉宽）可以随时间而变，并以步为单位进行定义。施加的步数越多（这会增大脉宽），为负载提供的功率就越大。施加的步数降低时（这会缩短脉宽），提供的功率就越小。PWM周期定义为一个完整周期的持续时间，或者开启和关闭时间相加的总时间。

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 24-3 给出了 PWM 的简化框图。

图 24-3: PWM 的简化框图



24.3.1 标准 PWM 操作

本节介绍的标准 PWM 功能对于所有 CCP 模块均可用且相同。

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

- T2PR/T4PR/T6PR 寄存器
- T2CON/T4CON/T6CON 寄存器
- CCPRxH:CCPRxL 寄存器对

图 24-3 给出了 PWM 操作的简化框图。

- 注 1:** 要使能 CCPx 引脚上的 PWM 输出，必须清零相应的 TRIS 位。
- 2:** 清零 CCPxCON 寄存器会放弃对 CCPx 引脚的控制。

24.3.2 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. 通过将相关的 TRIS 位置 1，禁止 CCPx 引脚输出驱动器。
2. 通过设置 CCPTMRS 寄存器选择与 PWM 相关的定时器。
3. 将 PWM 周期值装入相关的 T2PR/T4PR/T6PR 寄存器。
4. 通过将相应值装入 CCPxCON 寄存器，将 CCP 模块配置为 PWM 模式。
5. 将 PWM 占空比值装入 CCPRxH:CCPRxL 寄存器对。
6. 配置并启动在步骤 2 中选择的定时器：
 - 清零 PIRx 寄存器的定时器中断标志位。请参见下面的“注”。
 - 用定时器预分频值配置 TxCON 寄存器的 CKPS 位。
 - 通过将 TxCON 寄存器的 ON 位置 1 使能定时器。
7. 使能 PWM 输出引脚：
 - 等待直到定时器溢出，PIRx 寄存器的定时器中断标志位置 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。

24.4 CCP/PWM 时钟选择

PIC16(L)F1764/5/8/9 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。

最多有3个具有自动重载功能的8位定时器 (Timer2/4/6)。CCP 和 PWM 模块上的 PWM 模式可以使用这些定时器中的任意定时器。

CCPTMRS 寄存器用于选择要使用的定时器。

24.4.1 对 CCP 模块使用 TMR2/4/6

该器件具有一个新版本的 TMR2 模块，该模块具有许多新的模式，通过这些模式可以对 PWM 信号进行比先前器件更多的定制和控制。关于使用 Timer2 的不同模式生成 PWM 信号的示例，请参见[第 23.5 节“操作示例”](#)。CCP 操作要求用作 PWM 时基的定时器选择 Fosc/4 时钟源。

24.4.2 PWM 周期

PWM 周期可通过 Timer2/4/6 的 T2PR/T4PR/T6PR 寄存器来指定。PWM 周期可由[公式 24-1](#) 计算。

公式 24-1: PWM 周期

$$\text{PWM 周期} = [(PR2 + 1) \cdot 4 \cdot Tosc] \quad (TM2 \text{ 预分频值})$$

注 1: $Tosc = 1/Fosc$

当 TMR2/4/6 中的值与其相应的 T2PR/T4PR/T6PR 寄存器中的值相等时，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 CCPRxH:CCPRxL 寄存器对锁存到内部 10 位锁存器中。

注： 在确定 PWM 频率时不会用到定时器后分频比（见[图 24-1](#)）。

24.4.3 PWM 占空比

通过将 10 位值写入两个寄存器来指定 PWM 占空比：CCPRxH:CCPRxL 寄存器对。其中，各位的写入目标位置由 CCPxCON 寄存器的 FMT 位决定。如果 FMT = 0，则应将占空比值的高 2 位写入 CCPRxH 寄存器的 bit <1:0>，剩余 8 位写入 CCPRxL 寄存器。如果 FMT = 1，

则应将占空比的低 2 位写入 CCPRxL 寄存器的 bit <7:6>，高 8 位写入 CCPRxH 寄存器。如[图 24-4](#) 所示。这些位可以在任意时间写入。在周期结束（即，T2PR/T4PR/T6PR 和 TMR2/4/6 寄存器发生匹配）之后，占空比值才会被锁存到内部锁存器中。

[公式 24-2](#) 用于计算 PWM 脉宽。[公式 24-3](#) 用于计算 PWM 占空比。

公式 24-2: 脉冲宽度

$$\text{脉冲宽度} = CCPRxH:CCPRxL \bullet Tosc \\ \bullet (TM2 \text{ 预分频值})$$

公式 24-3: 占空比

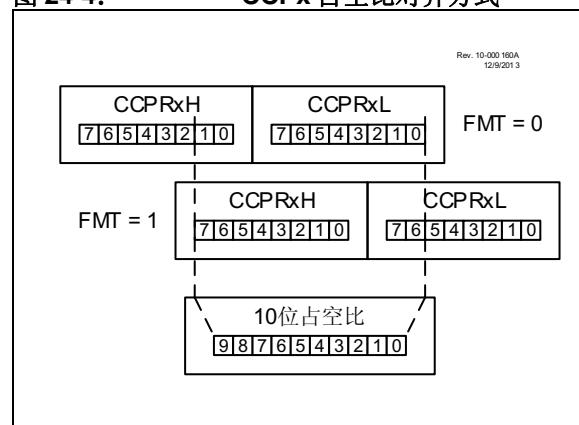
$$\text{占空比} = \frac{(CCPRxH:CCPRxL)}{4(PRx + 1)}$$

PWM 占空比寄存器是进行双重缓冲的，以确保无毛刺的 PWM 操作。

8 位定时器 TMR2/4/6 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 Timer2/4/6 预分频比设置为 1:1，则使用系统时钟。

当 10 位时基与内部缓冲寄存器匹配时，CCPx 引脚被清零（见[图 24-3](#)）。

图 24-4: CCPx 占空比对齐方式



24.4.4 PWM 分辨率

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个离散的占空比，而8位分辨率将可得到256个离散的占空比。

当T2PR/T4PR/T6PR为255时，PWM最大分辨率为10位。分辨率是T2PR/T4PR/T6PR寄存器值的函数，如公式24-4所示。

公式 24-4: PWM 分辨率

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

注：如果脉宽值比周期长，则指定的 PWM 引脚将保持不变。

表 24-2: 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
T2PR 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率 (位)	10	10	10	8	7	6

表 24-3: 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
T2PR 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率 (位)	8	8	8	6	5	5

24.4.5 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5.0节“振荡器模块（带故障保护时钟监视器）”。

24.4.6 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CCP寄存器为其复位状态。

24.4.7 PWM 输出

在 PWM 模式下，CCP 的输出是由模块生成的 PWM 信号，如上文所述。该输出可供以下外设使用：

- ADC 触发
- COG
- PRG
- DSM
- CLC
- 运放改写
- Timer2/4/6 复位
- 任意器件引脚

24.5 寄存器定义：CCPx 控制

寄存器 24-1： CCPxCON： CCPx 控制寄存器

R/W-0/0	U-0	R-x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	OUT	FMT		MODE<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 **EN:** CCPx 模块使能位

1 = 使能 CCPx

0 = 禁止 CCPx

bit 6 未实现：读为 0

bit 5 **OUT:** CCPx 输出数据位（只读）

bit 4 **FMT:** CCPW（脉宽）对齐方式位

如果 **MODE = PWM 模式**

1 = 左对齐格式，CCPRxH<7> 是 PWM 占空比的 MSB

0 = 右对齐格式，CCPRxL<0> 是 PWM 占空比的 LSB

bit 3-0 **MODE<3:0>:** CCPx 模式选择位

11xx = PWM 模式

1011 = 比较模式：脉冲输出，清零 TMR1

1010 = 比较模式：脉冲输出（0 - 1 - 0）

1001 = 比较模式：发生比较匹配时清零输出。输出在选择该模式时置 1。

1000 = 比较模式：发生比较匹配时置 1 输出。输出在选择该模式时置 1。

0111 = 捕捉模式：每 16 个上升沿

0110 = 捕捉模式：每 4 个上升沿

0101 = 捕捉模式：每个上升沿

0100 = 捕捉模式：每个下降沿

0011 = 捕捉模式：每个上升沿或下降沿

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并清零 TMR1

0000 = 捕捉 / 比较 /PWM 关闭（复位 CCPx 模块）（为保持向后兼容性而保留）

PIC16(L)F1764/5/8/9

寄存器 24-2: CCPTMRS: PWM 定时器选择控制寄存器 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
P4TSEL<1:0> ⁽¹⁾	P3TSEL<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 **P4TSEL<1:0>: PWM4 定时器选择位⁽¹⁾**

11 = 保留

10 = 在 PWM 模式下, PWM4 基于 Timer6 工作

01 = 在 PWM 模式下, PWM4 基于 Timer4 工作

00 = 在 PWM 模式下, PWM4 基于 Timer2 工作

bit 5-4 **P3TSEL<1:0>: PWM3 定时器选择位**

11 = 保留

10 = 在 PWM 模式下, PWM3 基于 Timer6 工作

01 = 在 PWM 模式下, PWM3 基于 Timer4 工作

00 = 在 PWM 模式下, PWM3 基于 Timer2 工作

bit 3-2 **C2TSEL<1:0>: CCP2 (PWM2) 定时器选择位**

11 = 保留

10 = 在 PWM 模式下, CCP2 基于 Timer6 工作

01 = 在 PWM 模式下, CCP2 基于 Timer4 工作

00 = 在 PWM 模式下, CCP2 基于 Timer2 工作

bit 1-0 **C1TSEL<1:0>: CCP1 (PWM1) 定时器选择位**

11 = 保留

10 = 在 PWM 模式下, CCP1 基于 Timer6 工作

01 = 在 PWM 模式下, CCP1 基于 Timer4 工作

00 = 在 PWM 模式下, CCP1 基于 Timer2 工作

寄存器 24-3: CCPRxL: CCPx 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
CCPR<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

MODE = 捕捉模式

CCPRxL<7:0>: 捕捉的 TMR1 值的 LSB

MODE = 比较模式

CCPRxL<7:0>: LSB 与 TMR1 值进行比较

MODE = PWM 模式且 FMT = 0

CCPRxL<7:0>: CCPW<7:0>——脉宽低 8 位

MODE = PWM 模式且 FMT = 1

CCPRxL<7:6>: CCPW<1:0>——脉宽低 2 位

CCPRxL<5:0>: 不使用

寄存器 24-4: CCPRxH: CCPx 高字节寄存器

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
CCPR<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

MODE = 捕捉模式

CCPRxH<7:0>: 捕捉的 TMR1 值的 MSB

MODE = 比较模式

CCPRxH<7:0>: MSB 与 TMR1 值进行比较

MODE = PWM 模式且 FMT = 0

CCPRxH<7:2>: 不使用

CCPRxH<1:0>: CCPW<9:8>——脉宽高 2 位

MODE = PWM 模式且 FMT = 1

CCPRxH<7:0>: CCPW<9:2>——脉宽高 8 位

PIC16(L)F1764/5/8/9

寄存器 24-5: CCPxCAP: CCPx 捕捉输入选择寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CTS<2:0>		
bit 7							

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 7-3 未实现: 读为 0

bit 2-0 CTS<2:0>: 捕捉触发输入选择位

111 = IOC_event

110 = LC3_output

101 = LC2_output

100 = C4_sync_out⁽¹⁾

011 = C3_sync_out⁽¹⁾

010 = C2_sync_out

001 = C1_sync_out

000 = 通过 CCPxPPS 寄存器选择的引脚

注 1: 仅限 PIC16(L)F1768/9。PIC16(L)F1764/5 上未实现。

表 24-4：与捕捉 / 比较 / PWM 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页							
CCPxCAP	—	—	—	—	—	CTS<2:0>			262							
CCPxCON	EN	OE	OUT	FMT	MODE<3:0>				259							
CCPRxL	捕捉 / 比较 / PWM 寄存器 x (LSB)								261							
CCPRxH	捕捉 / 比较 / PWM 寄存器 x (MSB)								261							
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		260							
INTCON	GIE	PEIE	TMROIE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102							
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106							
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	107							
T2PR	Timer2 周期寄存器								244*							
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				257							
TMR2	Timer2 模块寄存器								244							
T4PR	Timer4 周期寄存器								244*							
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				257							
TMR4	Timer4 模块寄存器								244							
T6PR	Timer6 周期寄存器								244*							
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				257							
TMR6	Timer6 模块寄存器								244							

图注：— = 未实现位，读为 0。捕捉 / 比较 / PWM 不使用阴影单元。

* 提供寄存器信息的页。

注 1：仅限 PIC16(L)F1768/9。

24.6 CCP/PWM 时钟选择

这些器件允许每个 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。

共有 3 个具有自动重载功能的 8 位定时器（Timer2、Timer4 和 Timer6）。CCP 和 10 位 PWM 模块上的 PWM 模式可以使用这些定时器中的任意定时器。

CCPTMRS 寄存器用于选择要使用的定时器。

24.7 寄存器定义：CCP/PWM 定时器控制

寄存器 24-6： CCPTMRS： PWM 定时器选择控制寄存器 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
P4TSEL<1:0>		P3TSEL<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 **P4TSEL<1:0>:** PWM4 定时器选择位

11 = 保留

10 = PWM4 基于 Timer6 工作

01 = PWM4 基于 Timer4 工作

00 = PWM4 基于 Timer2 工作

bit 5-4 **P3TSEL<1:0>:** PWM3 定时器选择位

11 = 保留

10 = PWM3 基于 Timer6 工作

01 = PWM3 基于 Timer4 工作

00 = PWM3 基于 Timer2 工作

bit 3-2 **C2TSEL<1:0>:** CCP2 (PWM2) 定时器选择位

11 = 保留

10 = 在 PWM 模式下，CCP2 基于 Timer6 工作

01 = 在 PWM 模式下，CCP2 基于 Timer4 工作

00 = 在 PWM 模式下，CCP2 基于 Timer2 工作

bit 1-0 **C1TSEL<1:0>:** CCP1 (PWM1) 定时器选择位

11 = 保留

10 = 在 PWM 模式下，CCP1 基于 Timer6 工作

01 = 在 PWM 模式下，CCP1 基于 Timer4 工作

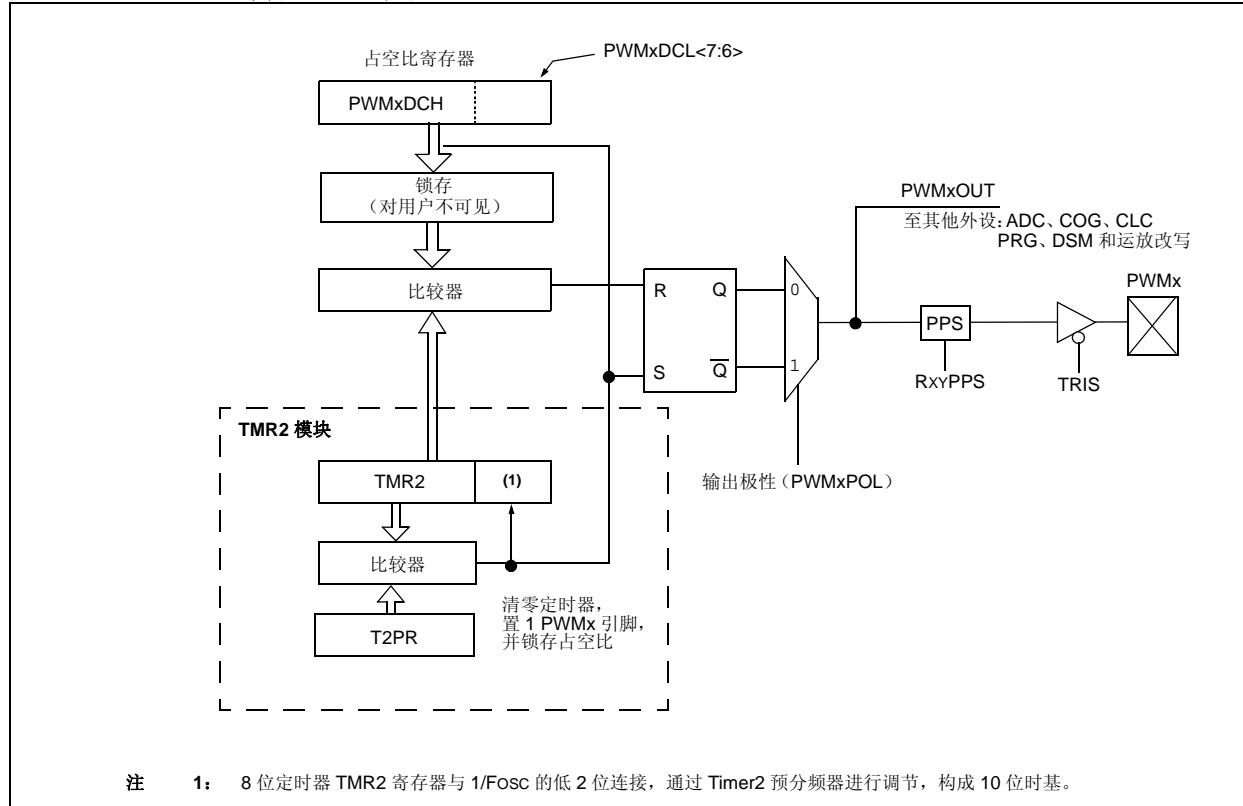
00 = 在 PWM 模式下，CCP1 基于 Timer2 工作

25.0 10 位脉宽调制 (PWM) 模块

10 位 PWM 模块可产生由占空比、周期和分辨率决定的脉宽调制信号，占空比、周期和分辨率则通过以下寄存器进行配置：

- T2PR
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

图 25-1：简化 PWM 框图



关于如何设置该模块使之进行 PWM 操作的详细步骤，请参见第 25.1.9 节“设置使用 PWMx 输出引脚的 PWM 操作”。

图 25-2：PWM 输出

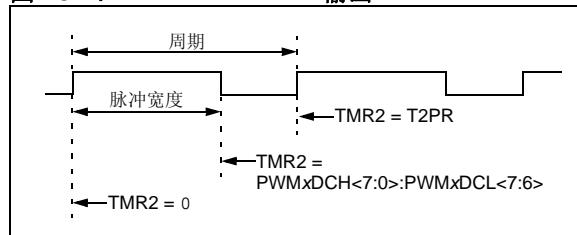


图 25-1 给出了 PWM 操作的简化框图。

图 25-2 给出了 PWM 信号的典型波形图。

25.1 PWM_x 引脚配置

所有 PWM 输出都与端口数据锁存器复用。用户必须通过清零相关的 TRIS 位将引脚配置为输出。

25.1.1 基本操作

PWM 模块可产生一个 10 位分辨率的输出。Timer2 和 T2PR 用于设置 PWM 的周期。PWMxDCL 和 PWMxDCH 寄存器用于配置占空比。周期由所有 PWM 模块共用，而占空比则独立进行控制。

注: 在确定 PWM 频率时不会用到 Timer2 后分频比。后分频器可用不同于 PWM 输出频率的频率进行伺服数据更新。

当 TMR2 清零时，与 Timer2 相关的所有 PWM 输出都会置 1。当 TMR2 等于相应 PWMxDCH（高 8 位）和 PWMxDCL<7:6>（低 2 位）寄存器指定的值时，每个 PWM_x 都会清零。当值大于等于 T2PR 时，PWM 输出永远不会清零（占空比为 100%）。

注: PWMxDCH 和 PWMxDCL 寄存器是进行双重缓冲的。当 Timer2 与 T2PR 匹配时，缓冲区会发生更新。在定时器匹配发生之前更新两个寄存器时需要非常小心。

25.1.2 PWM 输出极性

输出极性通过将 PWMxCON 寄存器的 PWMxPOL 位置 1 来进行反相。

25.1.3 PWM 周期

PWM 周期可通过 Timer2 的 T2PR 寄存器来指定。PWM 周期可由公式 25-1 计算。

公式 25-1: PWM 周期

$$\text{PWM 周期} = [T2PR + 1] \cdot 4 \cdot Tosc \cdot (TMR2 \text{ 预分频值})$$

注: Tosc = 1/Fosc

当 TMR2 中的值与 T2PR 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
- PWM 输出有效。（例外情况：当 PWM 占空比 = 0% 时，PWM 输出将保持无效。）
- PWMxDCH 和 PWMxDCL 寄存器的值被锁存到缓冲区中。

注: Timer2 后分频器对 PWM 操作没有任何作用。

25.1.4 PWM 占空比

PWM 占空比通过将一个 10 位值写入 PWMxDCH 和 PWMxDCL 寄存器来指定。PWMxDCH 寄存器包含高 8 位，PWMxDCL<7:6> 包含低 2 位。PWMxDCH 和 PWMxDCL 寄存器可以在任意时刻写入。

公式 25-2 用于计算 PWM 脉冲宽度。

公式 25-3 用于计算 PWM 占空比。

公式 25-2: 脉冲宽度

$$\text{脉冲宽度} = (PWMxDCH:PWMxDCL<7:6>) \cdot Tosc \cdot (TMR2 \text{ 预分频值})$$

注: Tosc = 1/Fosc

公式 25-3: 占空比

$$\text{占空比} = \frac{(PWMxDCH:PWMxDCL<7:6>)}{4(T2PR + 1)}$$

8 位定时器 TMR2 寄存器与 1/Fosc 的低 2 位连接，通过 Timer2 预分频器进行调节，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。

25.1.5 PWM 分辨率

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个离散的占空比，而8位分辨率将可得到256个离散的占空比。

当T2PR为255时，PWM最大分辨率为10位。分辨率是T2PR寄存器值的函数，如[公式 25-4](#)所示。

公式 25-4: **PWM 分辨率**

$$\text{分辨率} = \frac{\log[4(T2PR + 1)]}{\log(2)} \text{ 位}$$

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 25-1: PWM 频率和分辨率示例 (Fosc = 20 MHz)

PWM 频率	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频值	64	4	1	1	1	1
T2PR 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率 (位)	10	10	10	8	7	6.6

表 25-2: PWM 频率和分辨率示例 (Fosc = 8 MHz)

PWM 频率	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频值	64	4	1	1	1	1
T2PR 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率 (位)	8	8	8	6	5	5

25.1.6 休眠模式下的操作

在休眠模式下，TMR2寄存器将不会递增，模块状态也不会改变。如果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。

25.1.7 改变系统时钟频率

PWM频率是由系统时钟频率(FOSC)产生的。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 5.0 节“振荡器模块（带故障保护时钟监视器）”](#)。

25.1.8 复位的影响

任何复位都将强制所有端口为输入模式，并强制PWM寄存器为其复位状态。

25.1.9 设置使用 PWM_x 输出引脚的 PWM 操作

当将模块配置为使用 PWM_x 输出引脚的 PWM 操作时，可采用以下步骤：

1. 通过将相关的 TRIS 位置 1，禁止 PWM_x 引脚输出驱动器。
2. 清零 PWM_xCON 寄存器。
3. 将 PWM 周期值装入 T2PR 寄存器。
4. 将 PWM 占空比值装入 PWM_xDCH 寄存器和 PWM_xDCL 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。请参见下面的“注”。
 - 用 Timer2 预分频值配置 T2CON 寄存器的 CKPS 位。
 - 通过将 T2CON 寄存器的 ON 位置 1 来使能 Timer2。
6. 使能 PWM 输出引脚并等待直到 Timer2 溢出，PIR1 寄存器的 TMR2IF 位置 1。请参见下面的“注”。
7. 通过将相关的 TRIS 位清零并将所需的引脚 PPS 控制位置 1，使能 PWM_x 引脚输出驱动器。
8. 通过将相应值装入 PWM_xCON 寄存器来配置 PWM 模块。

注 1：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。如果以一个完整 PWM 信号开始并非至关重要，则用步骤 8 来代替步骤 4。
2：对于仅针对其他外设的操作，请禁止 PWM_x 引脚输出。

25.1.10 设置其他外设的 PWM 操作

将模块配置为由其他器件外设使用的 PWM 操作时，可采用以下步骤：

1. 通过将相关的 TRIS 位置 1，禁止 PWM_x 引脚输出驱动器。
2. 清零 PWM_xCON 寄存器。
3. 将 PWM 周期值装入 T2PR 寄存器。
4. 将 PWM 占空比值装入 PWM_xDCH 寄存器和 PWM_xDCL 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。请参见下面的“注”。
 - 用 Timer2 预分频值配置 T2CON 寄存器的 CKPS 位。
 - 通过将 T2CON 寄存器的 ON 位置 1 来使能 Timer2。
6. 使能 PWM 输出引脚：
 - 等待直到 Timer2 溢出，PIR1 寄存器的 TMR2IF 位置 1。请参见下面的“注”。
7. 通过将相应值装入 PWM_xCON 寄存器来配置 PWM 模块。

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。

25.2 寄存器定义：10 位 PWM 控制

表 25-3 列出了 10 位 PWM 外设的长位名称前缀。更多信息，请参见第 1.1.2.2 节“长位名称”。

表 25-3：

外设	位名称前缀
PWM3	PWM3
PWM4 ⁽¹⁾	PWM4

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 25-1：PWMxCON：PWM 控制寄存器

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	U-0
EN	—	OUT	POL	—	—	—	—
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 **EN:** 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 **OUT:** 读取该位时的 PWM 模块输出电平。

bit 4 **POL:** 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

PIC16(L)F1764/5/8/9

寄存器 25-2: PWMxDCH: PWM 占空比高位

R/W-x/u							
DC<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

DC<9:2>: PWM 占空比高 8 位

这些位是 PWM 占空比的高位。低 2 位位于 PWMxDCL 寄存器中。

寄存器 25-3: 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
DC<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

DC<1:0>: PWM 占空比低 2 位

这些位是 PWM 占空比的低位。高位位于 PWMxDCH 寄存器中。

bit 5-0

未实现: 读为 0

表 25-4: 与 10 位 PWM 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0> ⁽¹⁾		C1TSEL<1:0>		264
PWMxCON	EN	—	OUT	POL	MODE<1:0>		—	—	269
PWMxDCH	DC<9:2>								270
PWMxDCL	DC<1:0>		—	—	—	—	—	—	270
RxyPPS	—	—	—	RxyPPS<4:0>					158
TxCON	ON	CKPS<2:0>			OUTPS<3:0>				247
TxCLKCON	—	—	—	—	CS<3:0>				246
TxPR	TMRx 周期寄存器								232
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISO	150

图注: — = 未实现位, 读为 0, u = 不变, x = 未知。10 位 PWM 不使用阴影单元。

注 1: 未实现位, 读为 1。

2: 仅限 PIC16(L)F1768/9。

26.0 16 位脉宽调制 (PWM) 模块

脉宽调制 (Pulse Width Modulation, PWM) 模块可产生由相位、占空比、周期和偏移事件计数决定的脉宽调制信号，相位、占空比、周期和偏移事件计数则包含在以下寄存器中：

- PWMxPH 寄存器
- PWMxDC 寄存器
- PWMxPR 寄存器
- PWMxOF 寄存器

图 26-1 给出了 PWM 操作的简化框图。

每个 PWM 模块具有 4 种工作模式：

- 标准
- 匹配时置 1
- 匹配时翻转
- 中心对齐

关于每种 PWM 模式的更详细说明，请参见第 26.2 节“[“PWM 模式”](#)”。

每个 PWM 模块具有 4 种偏移模式：

- 独立运行模式
- 带同步启动的从运行模式
- 带同步启动的单事件从模式
- 带同步启动和定时器复位的连续运行从模式

通过使用偏移模式，每个 PWM 模块可以使其波形相对于同一器件中的任何其他 PWM 模块发生偏移。关于偏移模式的更详细说明，请参见[第 26.3 节“偏移模式”](#)。

每个 PWM 模块都具有可配置的重载操作，用于确保所有事件计数缓冲区在周期结束时发生更改，从而避免信号毛刺。图 26-2 给出了重载操作的简化框图。关于重载操作的更详细说明，请参见[第 26.4 节“重载操作”](#)。

图 26-1： 16 位 PWM 框图

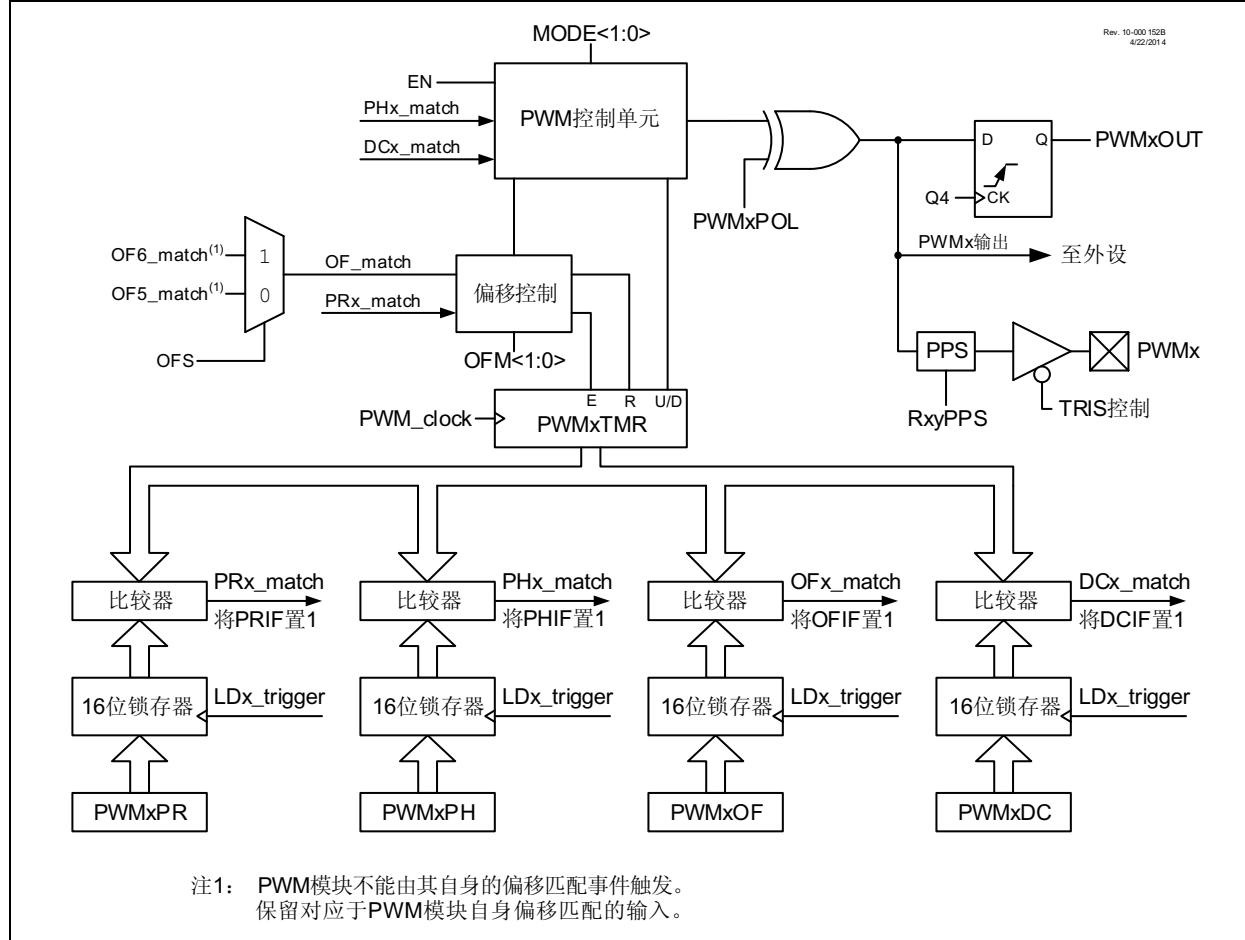
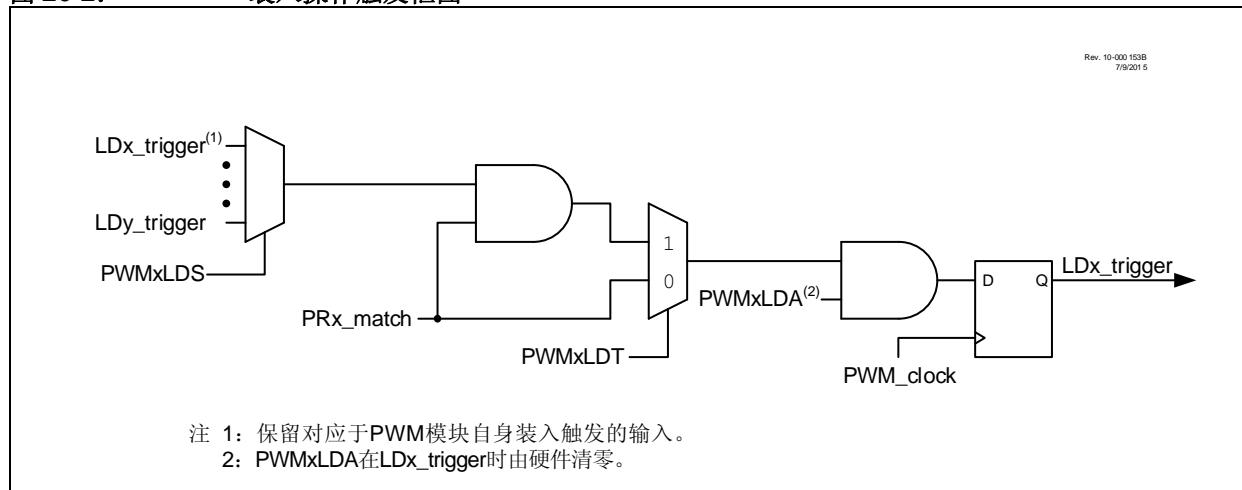


图 26-2: 装入操作触发框图



26.1 基本操作

PWM 模块可产生一个 16 位分辨率的脉宽调制输出。

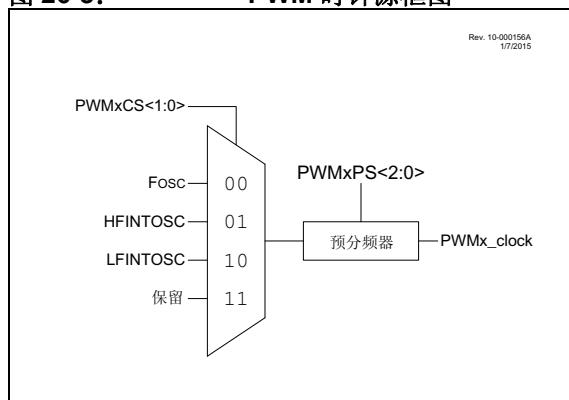
每个 PWM 模块均具有一个独立定时器，它通过由 PWMxCLKCON 寄存器（[寄存器 26-4](#)）决定的时钟源驱动。定时器值与事件计数寄存器进行比较，以生成 PWM 波形的各种事件，例如周期和占空比。关于描述时钟源的框图，请参见图 26-3。

可以使用 PWMxCON 寄存器的 EN 位单独使能每个 PWM 模块，也可以使用 PWMEN 寄存器的 MPWMxEN 位同时使能几个 PWM 模块。

可以使用 PWMxCON 寄存器的 OUT 位读取 PWM 输出的当前状态。在一些模式下，该位可以用软件清零和置 1，从而对 PWM 波形提供额外的软件控制。该位与 Fosc/4 同步，因此不会相对于 PWM_clock 实时变化。

注：如果 PWM_clock > Fosc/4，OUT 位可能不会精确表示 PWM 的输出状态。

图 26-3: PWM 时钟源框图



26.1.1 PWMx 引脚配置

这些器件使用 PPS 控制电路来将外设与任意器件 I/O 引脚连接。使用器件引脚 RxyPPS 控制寄存器（[寄存器 12-2](#)）选择所需的一个或多个引脚来用于 PWM 输出。

所有 PWM 输出都与端口数据锁存器复用，因此还必须通过清零相关的端口 TRIS 位将这些引脚配置为输出。

可以通过配置压摆率功能来优化要与 PWM 输出配合使用的速率。通过清零相关的端口 SLRCON 位可以实现高速输出开关。

通过将相关的端口 ODCON 位置 1，可以将 PWM 输出配置为漏极开路输出。

26.1.2 PWMx 输出极性

输出极性通过将 PWMxCON 寄存器的 POL 位置 1 来进行反相。即使未使能模块，极性控制也会影响 PWM 输出。

26.2 PWM 模式

PWM 模式使用 PWMxCON 寄存器（[寄存器 26-1](#)）的 MODE<1:0> 位进行选择。

在所有 PWM 模式下，还可以使用偏移匹配事件以 3 种偏移模式来同步 PWMxTMR。更多信息，请参见[第 26.3 节“偏移模式”](#)。

26.2.1 标准模式

标准模式（MODE = 00）将选择单相 PWM 输出。在该模式下，PWM 输出由周期、占空比和相位计数与 PWMxTMR 值匹配的时间决定。占空比在发生相位匹配时开始，在发生占空比匹配时结束。周期匹配会使定时器复位。此外，还可以使用偏移匹配以偏移模式来同步 PWMxTMR。更多信息，请参见[第 26.3 节“偏移模式”](#)。

[公式 26-1](#) 用于计算标准模式下的 PWM 周期。

[公式 26-2](#) 用于计算标准模式下的 PWM 占空比。

公式 26-1: 标准模式下的 PWM 周期

$$\text{周期} = \frac{(P\text{WM}x\text{PR} + 1) \cdot \text{预分频比}}{\text{PWM_clock}}$$

公式 26-2: 标准模式下的 PWM 占空比

$$\text{占空比} = \frac{(\text{P}\text{WM}x\text{DC} - \text{P}\text{WM}x\text{PH})}{\text{P}\text{WM}x\text{PR} + 1}$$

[图 26-4](#) 给出了标准模式的详细时序图。

26.2.2 匹配时置 1 模式

匹配时置 1 模式（MODE = 01）会在相位计数与 PWMxTMR 值匹配时产生有效输出。输出会一直保持有效，直到 PWMxCON 寄存器的 OUT 位清零或 PWM 模块被禁止为止。在该模式下，占空比计数没有任何作用。周期计数仅决定最大 PWMxTMR 值，高于该值时不会发生相位匹配。

在该模式下，可以使用 PWMxOUT 位来置 1 或清零 PWM 输出。写入该位之后，对该位的写操作将在 PWM_clock 的下一个上升沿发生。

[图 26-5](#) 给出了匹配时置 1 模式的详细时序图。

26.2.3 匹配时翻转模式

匹配时翻转模式（MODE = 10）会生成 50% 占空比的 PWM，周期为对于标准 PWM 模式计算的两倍长。在该模式下，占空比计数没有任何作用。相位计数决定在周期事件的多少个 PWMxTMR 周期后输出发生翻转。

在该模式下，对 PWMxCON 寄存器的 OUT 位的写操作没有任何作用。

[图 26-6](#) 给出了匹配时翻转模式的详细时序图。

26.2.4 中心对齐模式

中心对齐模式（MODE = 11）会生成按周期中心对齐的 PWM 波形。在该模式下，周期是 PWMxPR 计数的两倍。PWMxTMR 会递增至周期值，然后递减回到 0。占空比计数同时决定有效 PWM 输出的开始和结束。占空比在 PWMxTMR 进行递增情况下发生匹配事件时开始，在 PWMxTMR 进行递减情况下发生匹配事件时结束。递增匹配值等于周期计数减去占空比计数。递减匹配值等于递增匹配值加 1。

[公式 26-3](#) 用于计算中心对齐模式下的 PWM 周期。

公式 26-3: 中心对齐模式下的 PWM 周期

$$\text{周期} = \frac{(P\text{WM}x\text{PR} + 1) \cdot 2 \cdot \text{预分频比}}{\text{PWM_clock}}$$

[公式 26-4](#) 用于计算中心对齐模式下的 PWM 占空比。

公式 26-4: 中心对齐模式下的 PWM 占空比

$$\text{占空比} = \frac{\text{P}\text{WM}x\text{DC} \cdot 2}{(\text{P}\text{WM}x\text{PR} + 1) \cdot 2}$$

在该模式下，对 PWMxOUT 的写操作没有任何作用。

[图 26-7](#) 给出了中心对齐模式的详细时序图。

图 26-4: 标准 PWM 模式时序图

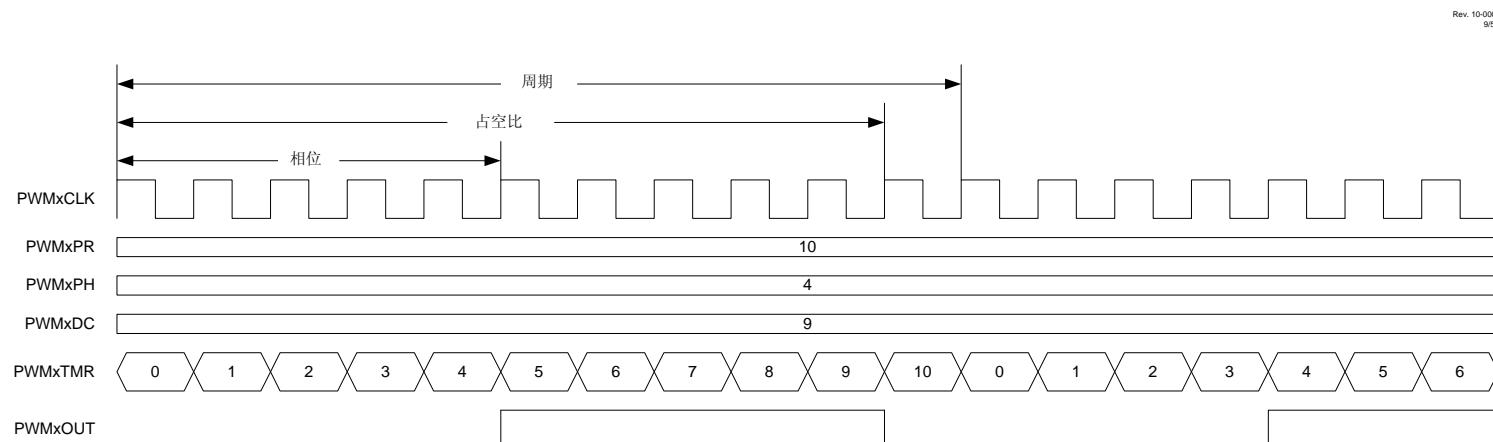


图 26-5: 匹配时置 1 PWM 模式时序图

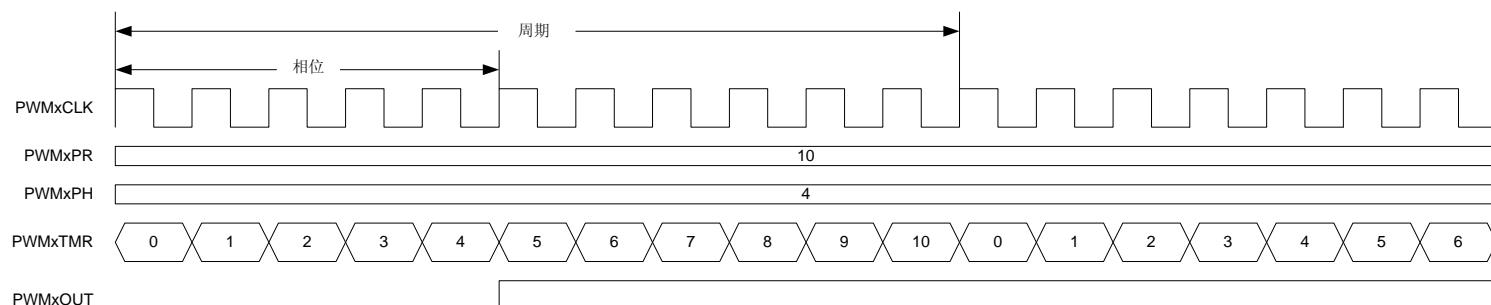


图 26-6：匹配时翻转 PWM 模式时序图

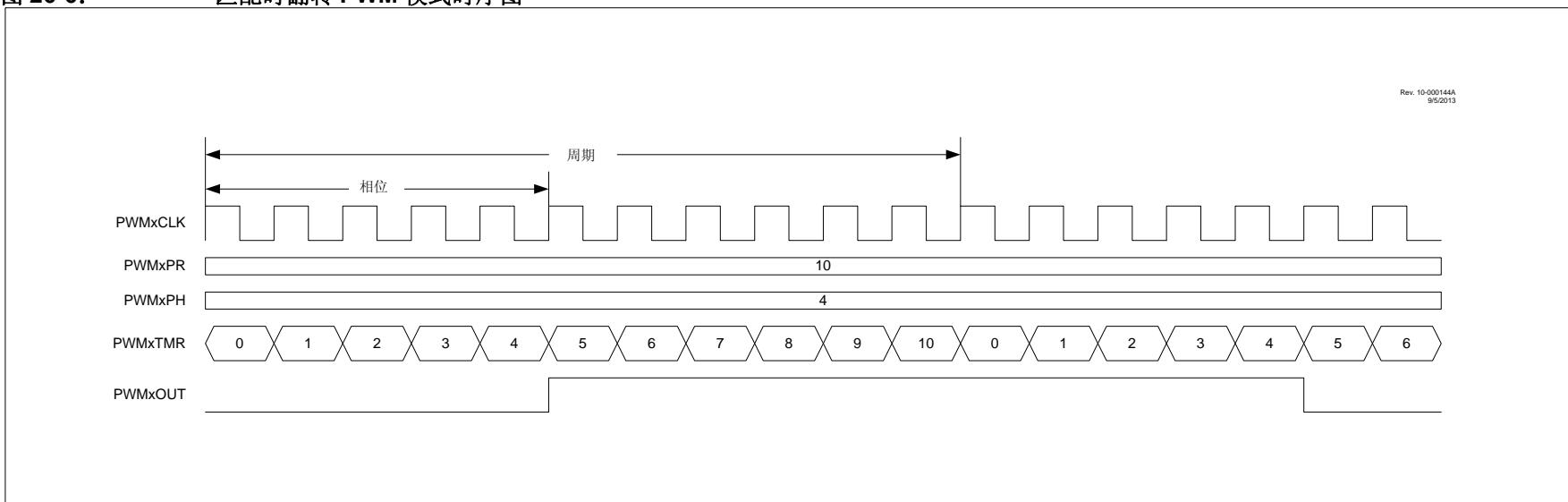
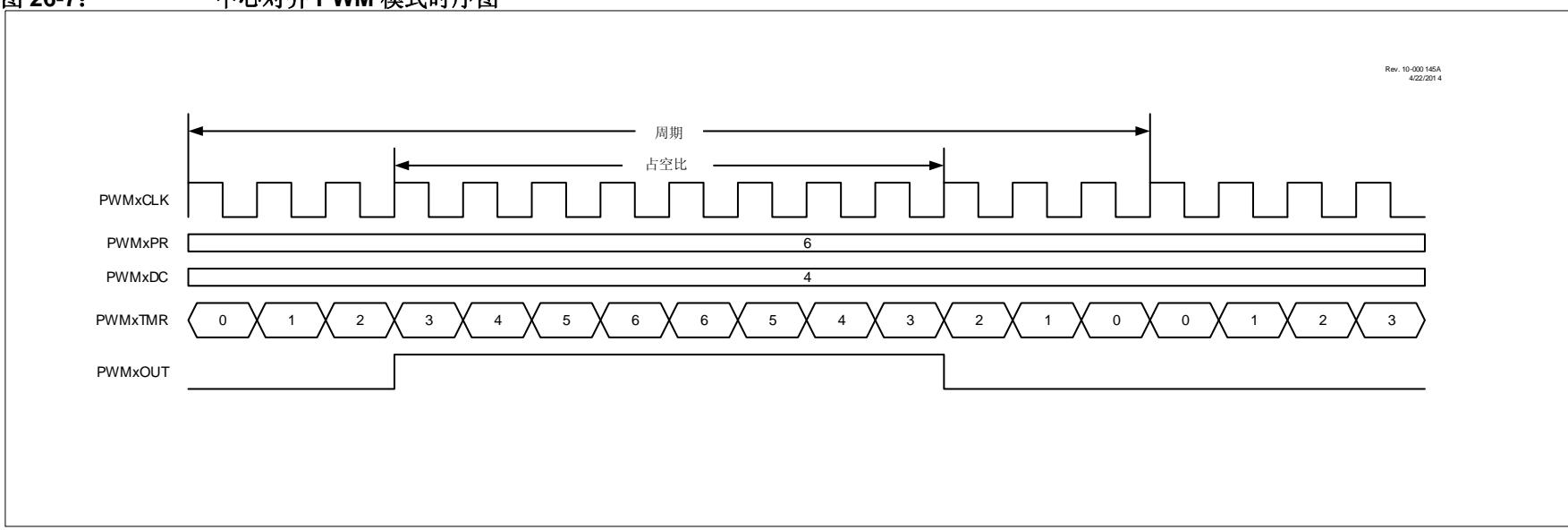


图 26-7：中心对齐 PWM 模式时序图



26.3 偏移模式

偏移模式用于将从 PWM 模块的波形相对于同一器件中的主 PWM 模块的波形进行调节。

26.3.1 独立运行模式

在独立运行模式 ($OFM = 00$) 下, PWM 模块不受器件中的其他 PWM 模块影响。与该模式下的 PWM 模块相关的 **PWMxTMR** 会在与该 PWM 模块相关的 EN 位置 1 时立即开始计数, 并一直保持计数, 直到 EN 位清零为止。周期事件会使 **PWMxTMR** 复位为零, 之后定时器会继续计数。

图 26-8 给出了在标准 PWM 模式下使用该模式的详细时序图。

26.3.2 带同步启动的从运行模式

在带同步启动的从运行模式 ($OFM = 01$) 下, 从模块 **PWMxTMR** 一直等待主模块的 **OF_match** 事件。当发生该事件时, 如果 EN 位置 1, **PWMxTMR** 会开始计数并保持计数, 直到软件清零 EN 位为止。从模块周期事件会使 **PWMxTMR** 复位为零, 之后定时器会继续计数。

图 26-9 给出了在标准 PWM 模式下使用该模式的详细时序图。

26.3.3 带同步启动的单事件从模式

在带同步启动的单事件从模式 ($OFM = 10$) 下, 从模块 **PWMxTMR** 一直等待主模块的 **OF_match** 事件。然后, 定时器从定时器中的已有值开始计数并保持计数, 直到发生周期匹配事件为止。发生周期事件时, 定时器会复位为零并停止计数。然后, 定时器会一直等到发生下一个主模块 **OF_match** 事件, 之后它会再次开始计数, 重复这一循环。

图 26-10 给出了在标准 PWM 模式下使用该模式的详细时序图。

26.3.4 带同步启动和定时器复位的连续运行从模式

在带同步启动和定时器复位的连续运行从模式 ($OFM = 11$) 下, 会禁止从模块 **PWMxTMR** 在从 PWM 模块使能位置 1 之后开始计数。第一个主模块 **OF_match** 事件会启动从模块 **PWMxTMR**。随后的主模块 **OF_match** 事件会使从模块 **PWMxTMR** 定时器值复位为 1, 之后从模块 **PWMxTMR** 会继续计数。下一个主模块 **OF_match** 事件会使从模块 **PWMxTMR** 复位为 1, 重复这一循环。在主模块的 **OF_match** 事件之前发生的从模块周期事件会使从模块 **PWMxTMR** 复位为零, 之后定时器会继续

计数。对于在该模式下工作的从模块, **PWMxPH** 寄存器对的值必须等于或大于 1, 否则不会发生相位匹配事件, 从而无法启动 PWM 输出占空比。

如果主模块和从模块的 **PWMxPR** 值相等, 并且在 PWM 模块工作时从模块偏移模式变为独立运行模式, 偏移时序将保持下去。

图 26-11 给出了在标准 PWM 模式下使用该模式的详细时序图。

注: 如果从模块 **PWM_clock** 的频率高于主模块 **PWM_clock** 的频率, 则会发生意外的结果。

26.3.5 中心对齐模式下的偏移匹配

当主模块在中心对齐模式下工作时, 偏移匹配事件将取决于 **PWMxTMR** 的计数方向。将 **PWMxOFCON** 寄存器的 **OFO** 位清零会导致在定时器递增时发生 **OF_match** 事件。将 **PWMxOFCON** 寄存器的 **OFO** 位置 1 会导致在定时器递减时发生 **OF_match** 事件。在非中心对齐模式下, **OFO** 位会被忽略。

OFO 位是进行双重缓冲的, 它在 PWM 模块工作时, 需要将 **LDA** 位置 1 才会生效。

图 26-12 和图 26-13 给出了在带同步启动的独立从模式下使用偏移匹配控制的中心对齐模式的详细时序图。

图 26-8：独立运行模式的时序图

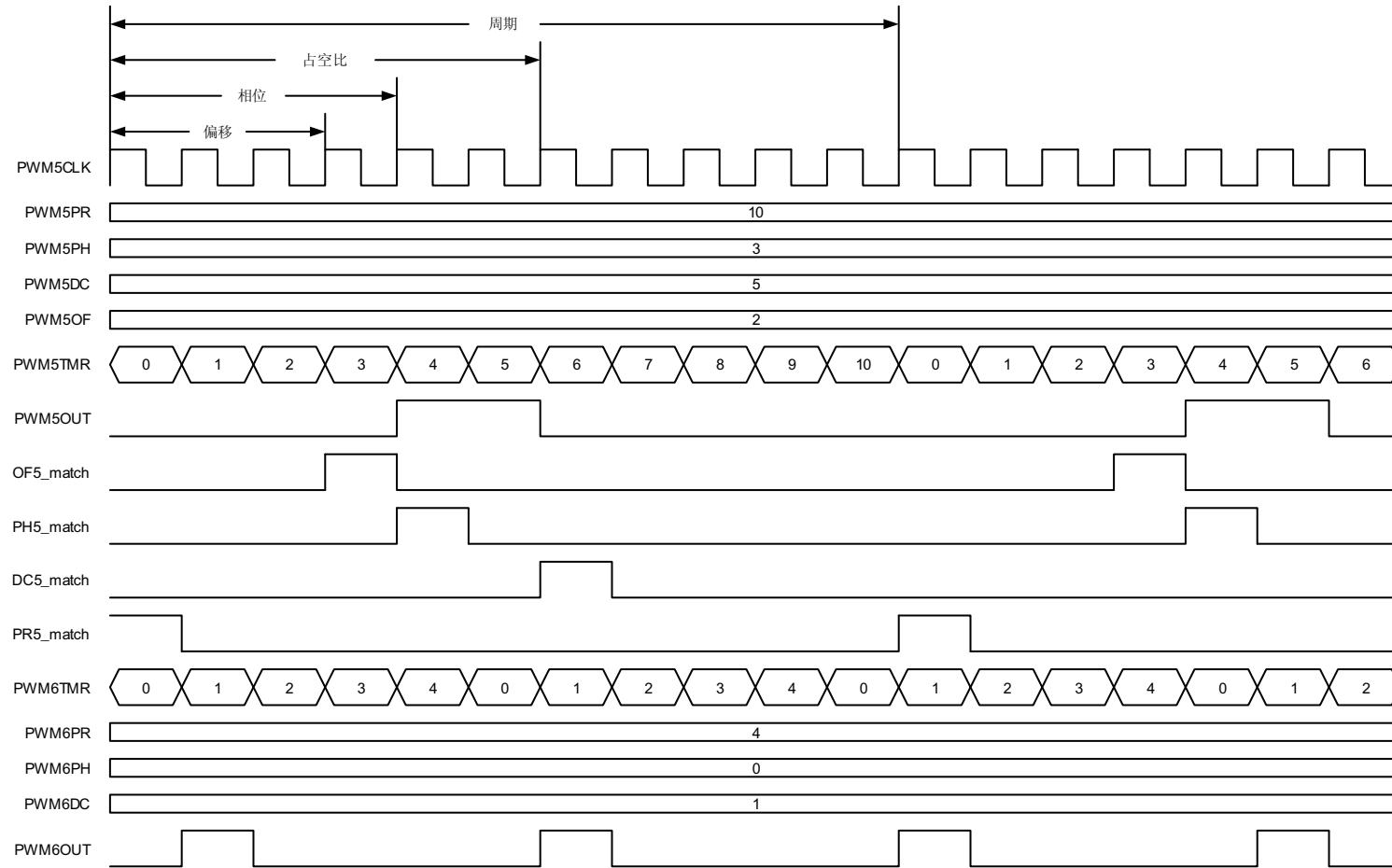


图 26-9: 带同步启动的从运行模式的时序图

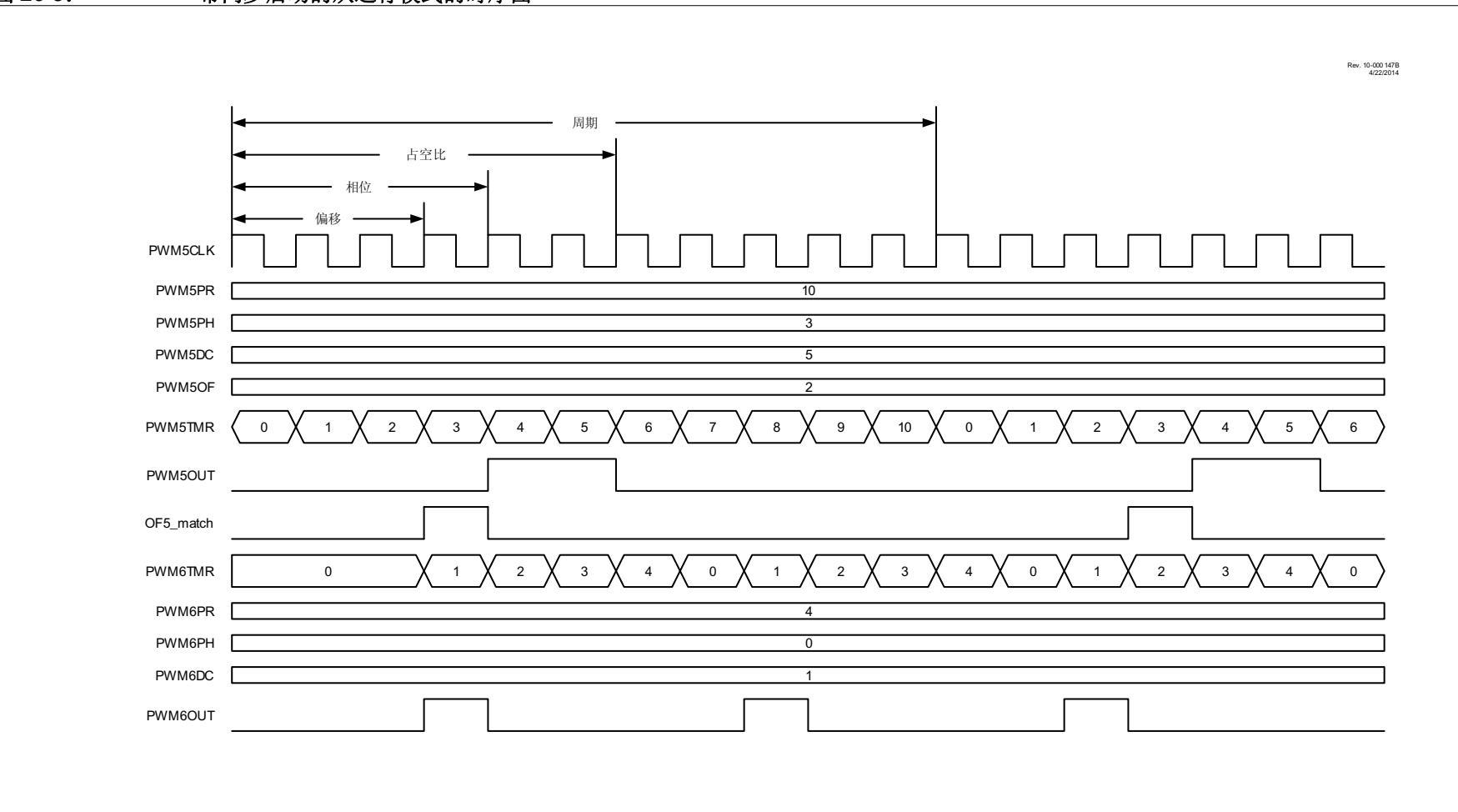


图 26-10：带同步启动的单事件从运行模式的时序图

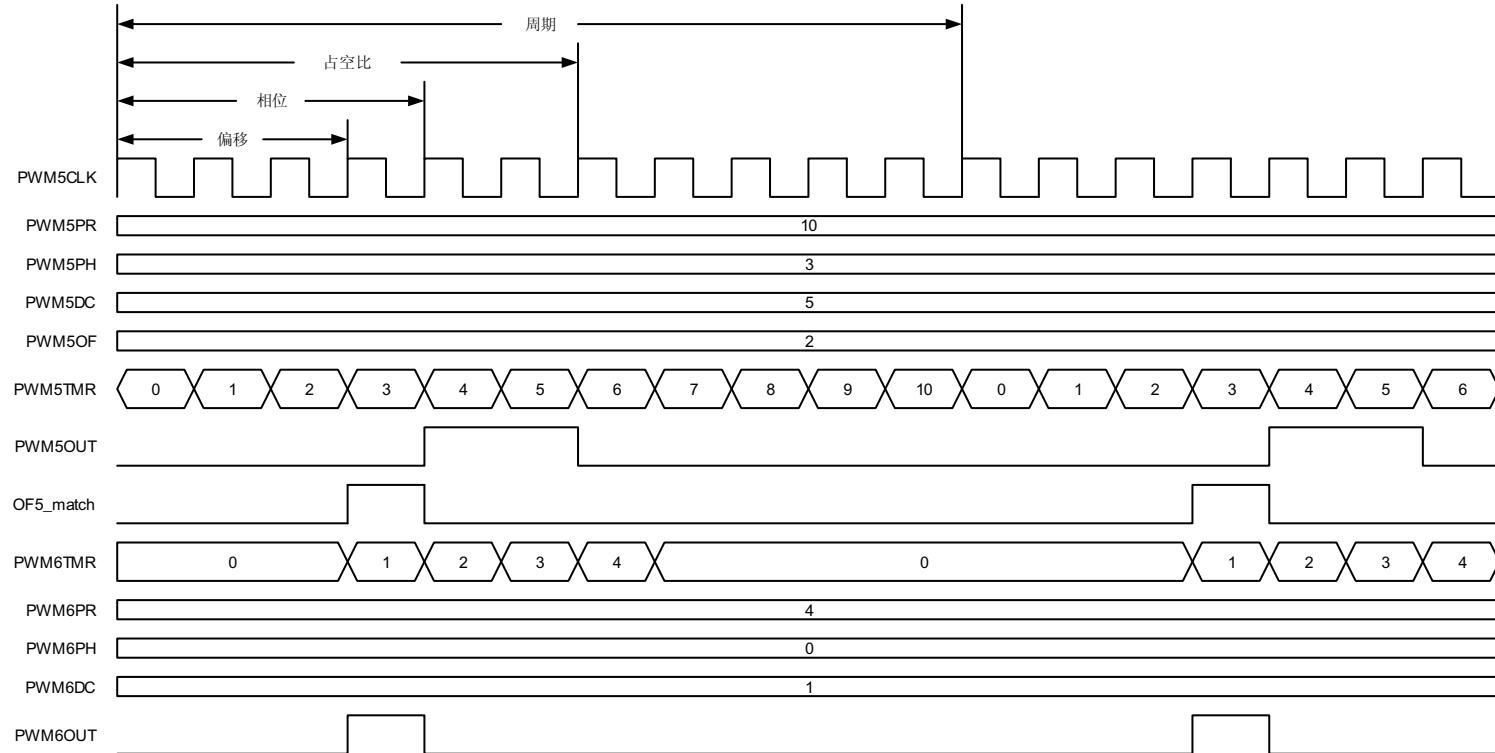


图 26-11：带立即复位和同步启动的连续从运行模式的时序图

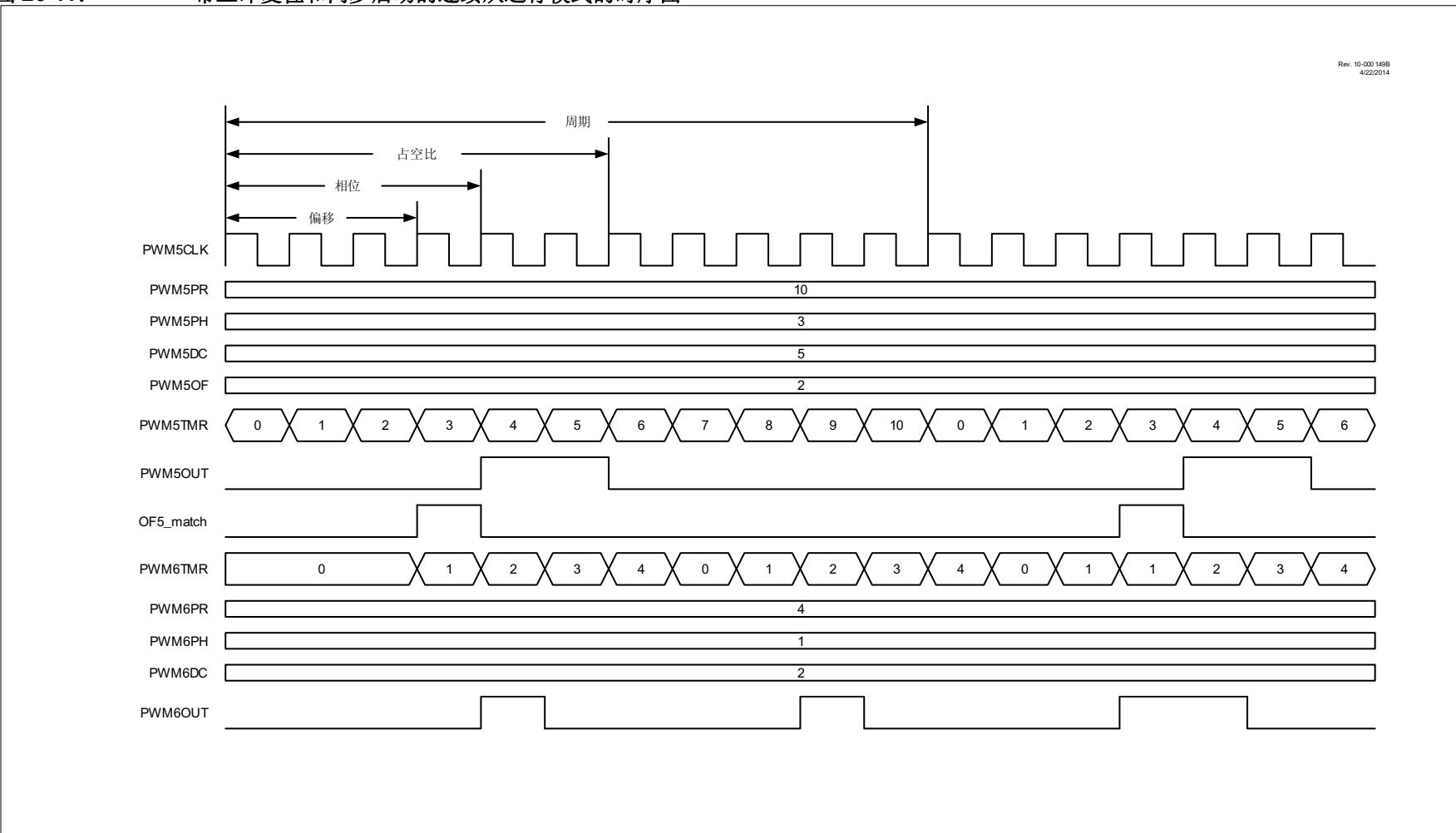


图 26-12: 在递增定时器时发生偏移匹配的时序图

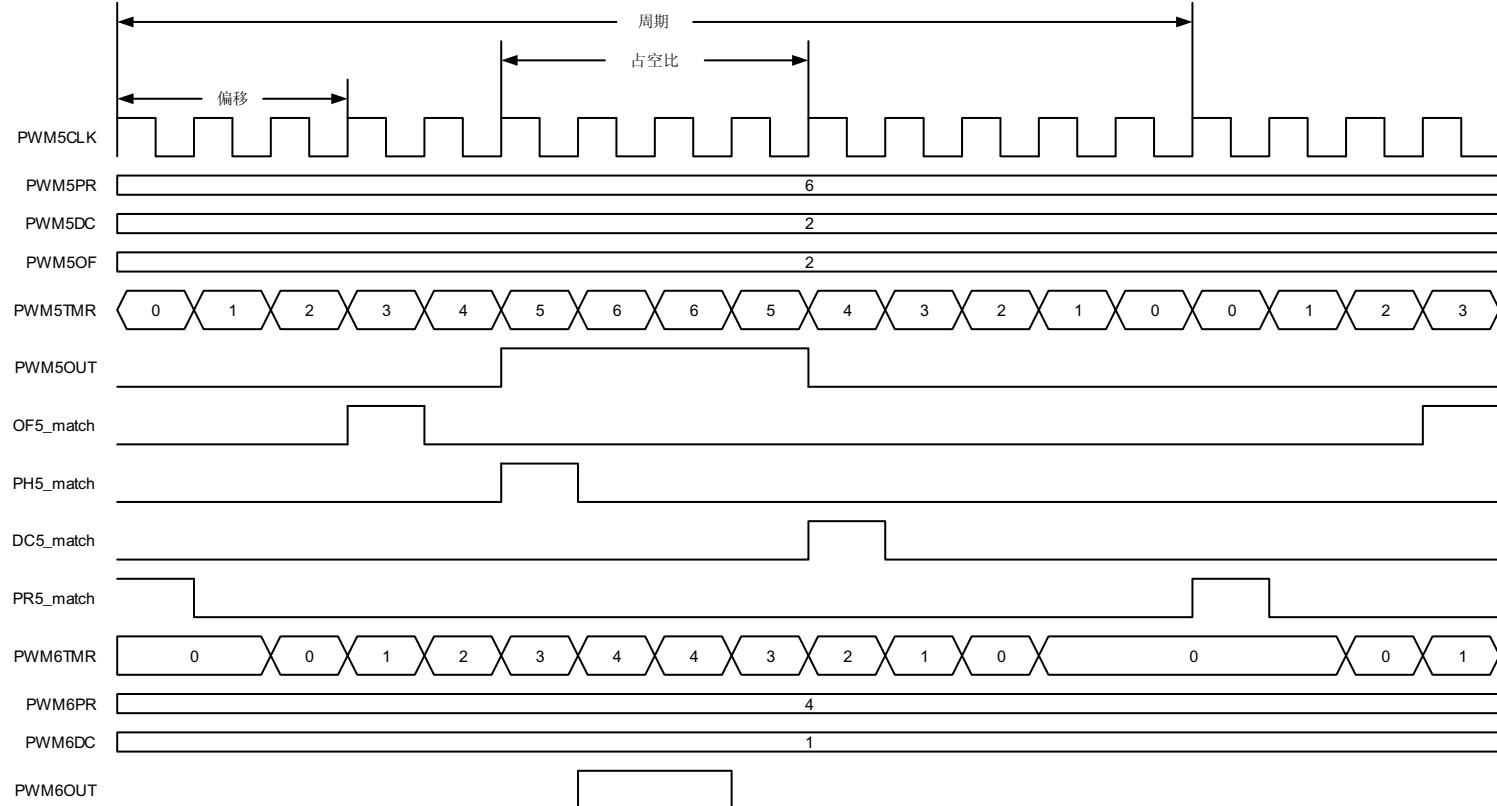
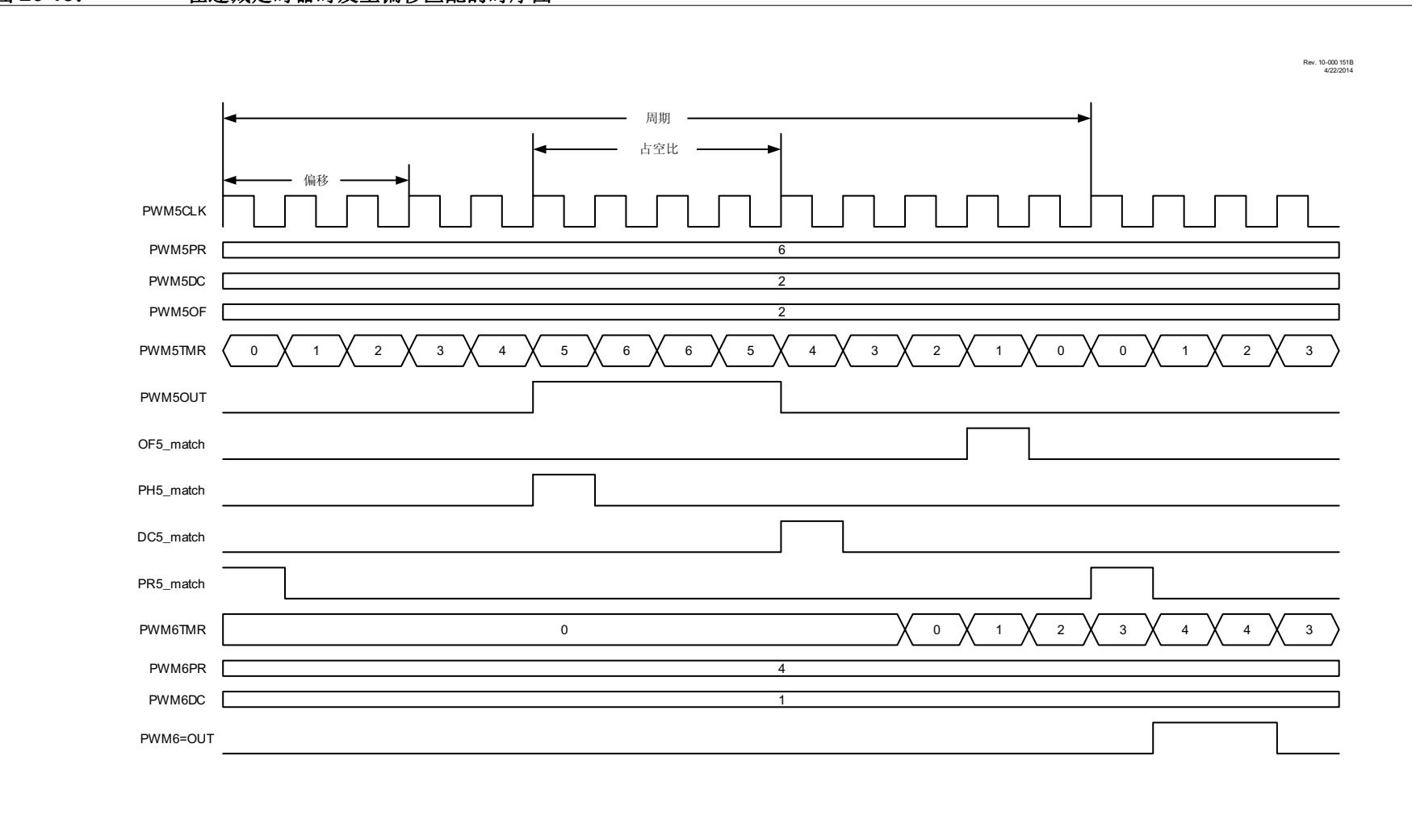


图 26-13: 在递减定时器时发生偏移匹配的时序图



26.4 重载操作

PWM 模块的 4 个控制寄存器对和 1 个控制位都是双重缓冲的，因此可以同时更新它们。它们包括：

- PWMxPHH:PWMxPHL 寄存器对
- PWMxDCH:PWMxDCL 寄存器对
- PWMxPRH:PWMxPRL 寄存器对
- PWMxOFL:PWMxOFL 寄存器对
- ODO 控制位

在被写入时，这些寄存器不会立即影响 PWM 的操作。默认情况下，只有满足激活条件之后，写入这些寄存器的内容才会被装入 PWM 操作缓冲寄存器。激活控制具有两种操作方法：

- 立即
- 触发

PWMxLDCON 寄存器的 LDT 位控制激活方法。这两种方法都要求 LDA 位置 1。全部 4 个缓冲寄存器对将在发生装入事件时同时装入。

26.4.1 立即重载

当 LDT 位清零时，即表示选择立即模式，缓冲区将在 LDA 位置 1 后的第一个周期事件时装入。当 PWM 模块独立工作时或当 PWM 模块作为其他从 PWM 模块的主要模块工作时，需要采用立即重载。

26.4.2 触发重载

当 LDT 位置 1 时，即选择触发模式，需要发生触发事件 LDA 位才会生效。触发源是器件中其他某个 PWM 模块的缓冲区装入事件。触发源通过 PWMxLDCON 寄存器的 LDS_{1:0} 位进行选择。缓冲区将在触发事件后的第一个周期事件时装入。当 PWM 模块作为另一个 PWM 的从模块工作，并且需要对两个模块中的缓冲区重载进行同步时，需要采用触发重载。

注 1：缓冲区装入操作会清零 LDA 位。

2：如果 LDA 位在 PWMxTMR = PWMxPR 的同时置 1，则 LDA 位会被忽略，直到发生下一个周期事件为止。当选择触发重载，并且触发事件与目标的周期事件同时发生时，就会发生这种情况。

26.5 休眠模式下的操作

如果通过 PWMxCLKCON<1:0> 选择 HFINTOSC 或 LFINTOSC 作为时钟源，则每个 PWM 模块将在休眠模式下继续工作。

26.6 中断

每个 PWM 模块都具有基于相位、占空比、周期和偏移匹配事件的 4 个独立中断。每个中断标志在相应信号的上升沿置 1。关于匹配信号的详细时序图，请参见 [图 26-8](#) 和 [图 26-12](#)。

26.7 寄存器定义：16 位 PWM 控制

表 26-1 列出了 16 位 PWM 外设的长位名称前缀。更多信息，请参见第 1.1 节“寄存器和位命名约定”。

表 26-1：

外设	位名称前缀
PWM5	PWM5
PWM6 ⁽¹⁾	PWM6

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 26-1： PWMxCON： PWM 控制寄存器

R/W-0/0	U-0	R/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EN	—	OUT	POL	MODE<1:0>	—	—	bit 0
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 = 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 **EN:** PWM 模块使能位

1 = 使能模块

0 = 禁止模块

bit 6 未实现：读为 0

bit 5 **OUT:** PWM 模块的输出状态

bit 4 **POL:** PWM 输出极性控制位

1 = PWM 输出有效状态为低电平

0 = PWM 输出有效状态为高电平

bit 3-2 **MODE<1:0>:** PWM 模式控制位

11 = 中心对齐模式

10 = 匹配时翻转模式

01 = 匹配时置 1 模式

00 = 标准 PWM 模式

bit 1-0 未实现：读为 0

寄存器 26-2: PWMxINTE: PWM 中断允许寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	OFIE	PHIE	DCIE	PRIE
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 **OFIE:** 偏移中断允许位

1 = 发生偏移匹配时中断 CPU

0 = 发生偏移匹配时不中断 CPU

bit 2 **PHIE:** 相位中断允许位

1 = 发生相位匹配时中断 CPU

0 = 发生相位匹配时不中断 CPU

bit 1 **DCIE:** 占空比中断允许位

1 = 发生占空比匹配时中断 CPU

0 = 发生占空比匹配时不中断 CPU

bit 0 **PRIE:** 周期中断允许位

1 = 发生周期匹配时中断 CPU

0 = 发生周期匹配时不中断 CPU

寄存器 26-3: PWMxINTF: PWM 中断请求寄存器

U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	OFIF	PHIF	DCIF	PRIF
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 = 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 **OFIF:** 偏移中断标志位⁽¹⁾

1 = 发生了偏移匹配事件

0 = 未发生偏移匹配事件

bit 2 **PHIF:** 相位中断标志位⁽¹⁾

1 = 发生了相位匹配事件

0 = 未发生相位匹配事件

bit 1 **DCIF:** 占空比中断标志位⁽¹⁾

1 = 发生了占空比匹配事件

0 = 未发生占空比匹配事件

bit 0 **PRIF:** 周期中断标志位⁽¹⁾

1 = 发生了周期匹配事件

0 = 未发生周期匹配事件

注 1: 当模块被禁止 (EN = 0) 时, 该位会被硬件强制清零。

PIC16F1764/5/8/9

寄存器 26-4: PWMxCLKCON: PWM 时钟控制寄存器

U-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
—		PS<2:0>		—	—	CS<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 6-4 **PS<2:0>**: 时钟源预分频比选择位

- 111 = 对时钟源进行 128 分频
- 110 = 对时钟源进行 64 分频
- 101 = 对时钟源进行 32 分频
- 100 = 对时钟源进行 16 分频
- 011 = 对时钟源进行 8 分频
- 010 = 对时钟源进行 4 分频
- 001 = 对时钟源进行 2 分频
- 000 = 无预分频器

bit 3-2 未实现: 读为 0

bit 1-0 **CS<1:0>**: 时钟源选择位

- 11 = 保留
- 10 = LFINTOSC (在休眠期间继续工作)
- 01 = HFINTOSC (在休眠期间继续工作)
- 00 = FOSC

寄存器 26-5: PWMxLDCON: PWM 重载触发源选择寄存器

R/W/HC-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
LDA ⁽¹⁾	LDT ⁽³⁾	—	—	—	—	—	LDS ^(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

HC = 硬件清零位

bit 7

LDA: 装入缓冲区激活位⁽¹⁾

如果 LDT = 1:

1 = 在发生选定触发事件的周期结束时装入 ODO 位以及 OFx、PHx、DCx 和 PRx 缓冲区。

0 = 不装入缓冲区 / 装入已完成

如果 LDT = 0:

1 = 在当前周期结束时装入 ODO 位以及 OFx、PHx、DCx 和 PRx 缓冲区

0 = 不装入缓冲区 / 装入已完成

bit 6

LDT: 触发时装入缓冲区位⁽³⁾

1 = 等待出现通过 LDS<1:0> 位选择的触发信号, 之后再使能 LDA 位

0 = 禁止装入触发。缓冲区装入由 LDA 位单独控制。

bit 5-1

未实现: 读为 0

bit 0

LDS: 装入触发源选择位^(2, 3)

1 = LD6_trigger

0 = LD5_trigger

注

1: 该位在重载操作之后由模块清零。它可以用软件清零, 以清除现有的激活事件。

2: 对应于 PWM 模块自己的 LDx_trigger 的触发源是保留的。

3: 仅限 PIC16(L)F1768/9。

PIC16F1764/5/8/9

寄存器 26-6: PWMxOFCON: PWM 偏移触发源选择寄存器

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
—	OFM<1:0> ⁽²⁾	OFO ⁽¹⁾	—	—	—	—	OFS ⁽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 未实现: 读为 0

bit 6-5 OFM<1:0>: 偏移模式选择位 ⁽²⁾

11 = 带偏移触发定时器复位和同步启动的连续运行从模式

10 = 带偏移触发同步启动的单事件从模式

01 = 带偏移触发同步启动的从运行模式

00 = 独立运行模式

bit 4 OFO: 偏移匹配输出控制位 ⁽²⁾

如果 MODE<1:0> = 11 (PWM 中心对齐模式):

1 = 在 PWMxTMR 递增时发生 OFx_match

0 = 在 PWMxTMR 递减时发生 OFx_match

如果 MODE<1:0> = 00、01 或 10 (所有其他模式):

该位被忽略

bit 3-1 未实现: 读为 0

bit 0 OFS: 偏移触发源选择位 ⁽²⁾

1 = OF6_match

0 = OF5_match

注 1: 对应于 PWM 模块自己的 OFx_match 的触发源是保留的。

2: 仅限 PIC16(L)F1768/9。

寄存器 26-7: PWMxPHH: PWMx 相位计数高字节寄存器

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
PH<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

PH<15:8>: PWM 相位高字节位
PWM 相位计数的高 8 位

寄存器 26-8: PWMxPHL: PWMx 相位计数低字节寄存器

R/W-x/u							
PH<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

PH<7:0>: PWM 相位低字节位
PWM 相位计数的低 8 位

PIC16F1764/5/8/9

寄存器 26-9: PWMxDCH: PWMx 占空比计数高字节寄存器

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
DC<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 **DC<15:8>**: PWM 占空比高字节位
PWM 占空比计数的高 8 位

寄存器 26-10: PWMxDCL: PWMx 占空比计数低字节寄存器

R/W-x/u							
DC<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 **DC<7:0>**: PWM 占空比低字节位
PWM 占空比计数的低 8 位

寄存器 26-11: PWMxPRH: PWM_x 周期计数高字节寄存器

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
PR<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

PR<15:8>: PWM 周期高字节位

PWM 周期计数的高 8 位

寄存器 26-12: PWMxPRL: PWM_x 周期计数低字节寄存器

R/W-x/u							
PR<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

PR<7:0>: PWM 周期低字节位

PWM 周期计数的低 8 位

PIC16F1764/5/8/9

寄存器 26-13: PWMxOFH: PWMx 偏移计数高字节寄存器

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
OF<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 **OF<15:8>**: PWM 偏移高字节位
PWM 偏移计数的高 8 位

寄存器 26-14: PWMxOFL: PWMx 偏移计数低字节寄存器

R/W-x/u							
OF<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 **OF<7:0>**: PWM 偏移低字节位
PWM 偏移计数的低 8 位

寄存器 26-15: PWMxTMRH: PWMx 定时器高字节寄存器

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
TMR<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

TMR<15:8>: PWM 定时器高字节位
PWM 定时器计数器的高 8 位

寄存器 26-16: PWMxTMRL: PWMx 定时器低字节寄存器

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
TMR<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

TMR<7:0>: PWM 定时器低字节位
PWM 定时器计数器的低 8 位

PIC16F1764/5/8/9

注：以下 3 个镜像寄存器没有长位名称和短位名称形式。

寄存器 26-17： PWMEN: PWMEN 位镜像寄存器

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	MPWM6EN ⁽¹⁾	MPWM5EN	—	—	—	—
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

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

bit 7-6 未实现：读为 0

bit 5-4 **PWMMxEN:** PWM6/PWM5 使能位

每个 PWM 模块的 PWMMCON<7> 位的镜像副本

bit 3-0 未实现：读为 0

注 1：仅限 PIC16(L)F1768/9。（也适用于 [寄存器 26-18](#) 和 [寄存器 26-19](#)）

寄存器 26-18： PWMLD: LDA 位镜像寄存器

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	MPWM6LD ⁽¹⁾	MPWM5LD	—	—	—	—
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

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

bit 7-6 未实现：读为 0

bit 5-4 **MPWMxLDA:** PWM6/PWM5 LDA 位

每个 PWM 模块的 PWMMLDCON<7> 位的镜像副本

bit 3-0 未实现：读为 0

寄存器 26-19： PWMOUT: PWMOUT 位镜像寄存器

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	MPWM6OUT ⁽¹⁾	MPWM5OUT	—	—	—	—
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

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

bit 7-3 未实现：读为 0

bit 5-4 **MPWMxOUT:** PWM6/PWM5 OUT 位

每个 PWM 模块的 PWMMOUTCON<5> 位的镜像副本

bit 3-0 未实现：读为 0

表 26-2: 与 16 位 PWM 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页		
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		86		
PWMEN	—	—	MPWM6EN ⁽¹⁾	MPWM5EN	—	—	—	—	294		
PWMLD	—	—	MPWM6LD ⁽¹⁾	MPWM5LD	—	—	—	—	294		
PWMOUT	—	—	MPWM6OUT ⁽¹⁾	MPWM5OUT	—	—	—	—	294		
PWM5PHL	PH<7:0>								289		
PWM5PHH	PH<15:8>								289		
PWM5DCL	DC<7:0>								290		
PWM5DCH	DC<15:8>								290		
PWM5PRH	PR<15:8>								291		
PWM5PRL	PR<7:0>								291		
PWM5OFH	OF<15:8>								292		
PWM5OFL	OF<7:0>								292		
PWM5TMRH	TMR<15:8>								293		
PWM5TMRL	TMR<7:0>								293		
PWM5CON	EN	—	OUT	POL	MODE<1:0>		—	—	284		
PWM5INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	285		
PWM5INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	285		
PWM5CLKCON	—	PS<2:0>			—	—	CS<1:0>		286		
PWM5LDCON	LDA	LDT ⁽¹⁾	—	—	—	—	—	LDS ⁽¹⁾	287		
PWM5OFCON	—	OFM<1:0> ⁽¹⁾		OFO	—	—	—	OFS ⁽¹⁾	288		
PWM6PHL ⁽¹⁾	PH<7:0>								289		
PWM6PHH ⁽¹⁾	PH<15:8>								289		
PWM6DCL ⁽¹⁾	DC<7:0>								290		
PWM6DCH ⁽¹⁾	DC<15:8>								290		
PWM6PRL ⁽¹⁾	PR<7:0>								291		
PWM6PRH ⁽¹⁾	PR<15:8>								291		
PWM6OFL ⁽¹⁾	OF<7:0>								292		
PWM6OFH ⁽¹⁾	OF<15:8>								292		
PWM6TMRL ⁽¹⁾	TMR<7:0>								293		
PWM6TMRH ⁽¹⁾	TMR<15:8>								293		
PWM6CON ⁽¹⁾	EN	—	OUT	POL	MODE<1:0>		—	—	284		
PWM6INTE ⁽¹⁾	—	—	—	—	OFIE	PHIE	DCIE	PRIE	285		
PWM6INTF ⁽¹⁾	—	—	—	—	OFIF	PHIF	DCIF	PRIF	285		
PWM6CLKCON ⁽¹⁾	—	PS<2:0>			—	—	CS<1:0>		286		
PWM6LDCON ⁽¹⁾	LDA	LDT ⁽¹⁾	—	—	—	—	—	LDS ⁽¹⁾	287		
PWM6OFCON ⁽¹⁾	—	OFM<1:0> ⁽¹⁾		OFO	—	—	—	OFS ⁽¹⁾	288		

图注: — = 未实现位, 读为 0。16 位 PWM 不使用阴影单元。

注 1: 仅限 PIC16(L)F1768/9。

表 26-3: 与 16 位 PWM 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	IESO	CLKOUTEN	BOREN<1:0>		—	65
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

图注: — = 未实现位, 读为 0。16 位 PWM 不使用阴影单元。

27.0 互补输出发生器（COG）模块

互补输出发生器（COG）的主要用途是将单个输出 PWM 信号转换为两个互补的输出 PWM 信号。此外，COG 还可以将两个独立的输入事件转换为单个或互补的 PWM 输出。

COG PWM 频率和占空比由上升沿事件输入和下降沿事件输入决定。上升沿事件和下降沿事件可以使用相同的输入源。输入源可以与 COG_clock 同步或异步。

上升沿事件的发生速率决定 PWM 频率。从上升沿事件到下降沿事件的时间决定占空比。

一个可选的时钟输入用于产生相位延时时间、消隐时间和死区时间。此外，还可以生成具有可编程延时链的死区时间，该延时链是独立于所有时钟源的。

图 27-2 至图 27-6 给出了各种 COG 模式的简化框图。

COG 模块具有以下特性：

- 6 种工作模式：
 - 转向 PWM 模式
 - 同步转向 PWM 模式
 - 正向全桥模式
 - 反向全桥模式
 - 半桥模式
 - 推挽模式
- 可选择的 COG_clock 时钟源
- 可独立选择的上升沿事件源
- 可独立选择的下降沿事件源
- 可独立选择的边沿或电平事件敏感性
- 独立的输出极性选择
- 具有独立的上升沿和下降沿延时的相位延时
- 死区控制：
 - 独立的上升沿和下降沿事件死区时间
 - 同步和异步时序
- 具有独立的上升沿和下降沿事件消隐时间的消隐控制
- 自动关断控制：
 - 可独立选择的关断源
 - 自动重启使能
 - 自动关断引脚改写控制
(高电平、低电平、关闭和高阻态)

27.1 输出到引脚（所有模式）

COG 外设具有 4 个输出：COGA、COGB、COGC 和 COGD。

使用 COGxCON0 寄存器的 MD<2:0> 位选择的工作模式决定在每个输出上提供的波形。每个器件引脚的独立外设源控制用于选择输出将出现在哪个（些）引脚上。更多信息，请参见 RxyPPS 寄存器（[寄存器 12-2](#)）。

27.2 事件驱动的 PWM（所有模式）

除了基于单个 PWM 输入生成 PWM 和互补输出之外，COG 还可以基于周期性的上升沿事件和独立的下降沿事件生成 PWM 波形。在这种情况下，下降沿事件通常基于外部 PWM 驱动电路内的模拟反馈产生。在这种配置中，高功率开关瞬变可能会触发需要进行消隐的虚假下降沿事件。COG 可以配置为在紧接上升沿（下降沿）事件驱动输出之后的一段时间内对下降沿（上升沿）事件输入进行消隐。这被称为输入消隐，[第 27.8 节“消隐控制”](#) 对它进行了介绍。

有可能需要防范发生外部电路故障的可能性。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。这被称为自动关断，[第 27.10 节“自动关断控制”](#) 对它进行了介绍。

COG 可以配置为以相位延时的方式与另一个 PWM 配合工作。有效驱动周期通过一个相位延时定时器相对于上升沿事件进行延时。[第 27.9 节“相位延时”](#) 更详细地介绍了相位延时。

图 27-10 给出了基于单个 CCP1 输入产生的带相位延时和死区的典型工作波形。

27.3 工作模式

27.3.1 转向 PWM 模式

在转向 PWM 模式下，基于输入事件源产生的 PWM 信号以单相 PWM 的形式输出，它可以转向到 4 个 COG 输出的任意组合。输出转向在写入 COGxSTR 寄存器之后的指令周期生效。

同步转向 PWM 模式与转向 PWM 模式基本相同，只是对输出转向的更改在 COGxSTR 寄存器写操作之后的第一个上升沿事件时生效。静态输出数据不进行同步。

图 27-2 和图 27-3 给出了转向模式的配置。

转向 PWM 和同步转向 PWM 模式分别通过将 COGxCON0 寄存器（寄存器 27-1）的 MD_{2:0} 位设置为 000 和 001 进行选择。

27.3.2 全桥模式

在正向和反向全桥模式下，4 个 COG 输出中的两个输出有效，其他两个无效。在两个有效输出中，一个通过 PWM 输入信号进行调制，另一个则以 100% 占空比开启。当方向发生改变时，将会插入死区时间，对调制输出进行延时。这使不调制的驱动器有时间可以关断，从而防止串联的功率器件中产生直通电流。

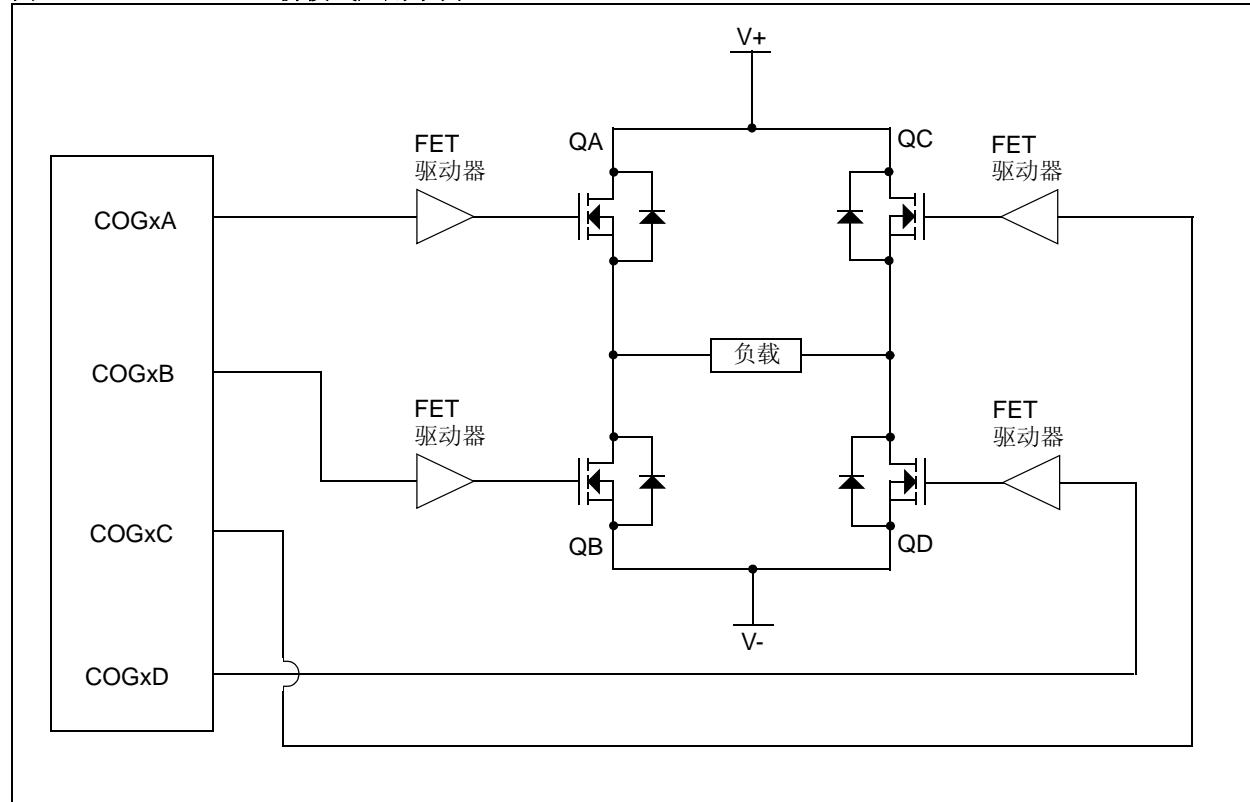
在正向全桥模式下，PWM 输入用于调制 COGxD 输出，并以 100% 占空比驱动 COGxA 输出。

在反向全桥模式下，PWM 输入用于调制 COGxB 输出，并以 100% 占空比驱动 COGxC 输出。

全桥配置如图 27-4 所示。典型的全桥波形如图 27-12 和图 27-13 所示。

全桥正向和全桥反向模式分别通过将 COGxCON0 寄存器的 MD_{2:0} 位设置为 010 和 011 进行选择。

图 27-1：全桥模式应用示例



27.3.3 半桥模式

在半桥模式下，COG 基于上升沿和下降沿事件源生成两个互补的 PWM 输出波形。在最简单的配置下，上升沿和下降沿事件源使用相同的信号，即具有所需周期和占空比的 PWM 信号。COG 将此单个 PWM 输入转换为两个互补的 PWM 输出。两个 PWM 输出的频率和占空比与单输入 PWM 信号的频率和占空比相匹配。可以对每个输出由关到开的跳变进行延时，使之避开另一个输出由开到关的跳变，从而在 PWM 跳变后立即产生一段不驱动任一输出的时间。这被称为死区时间，[第 27.7 节“死区控制”](#) 对它进行了介绍。

半桥配置如图 27-5 所示。图 27-9 给出了基于单个 CCP1 输入产生的带死区的典型工作波形。

主输出可送到 COGxA 和 / 或 COGxC 上。互补输出可送到 COGxB 和 / 或 COGxD 上。

半桥模式通过将 COGxCON0 寄存器的 MD<2:0> 位设置为 100 进行选择。

27.3.4 推挽模式

在推挽模式下，COG 生成在每个 PWM 周期在两对 COG 输出之间交替的单个 PWM 输出。COGxA 的信号与 COGxC 相同。COGxB 的信号与 COGxD 相同。输出驱动器通过上升沿输入事件激活，通过下降沿事件输入终止。每个上升沿事件会启动一个新的周期，导致输出切换到前一周期中未使用的 COG 对。

推挽配置如图 27-6 所示。图 27-11 给出了基于单个 CCP1 输入产生的典型推挽波形。

推挽模式通过将 COGxCON0 寄存器的 MD<2:0> 位设置为 101 进行选择。

图 27-2: 简化 COG 框图 (转向 PWM 模式, MD = 0)

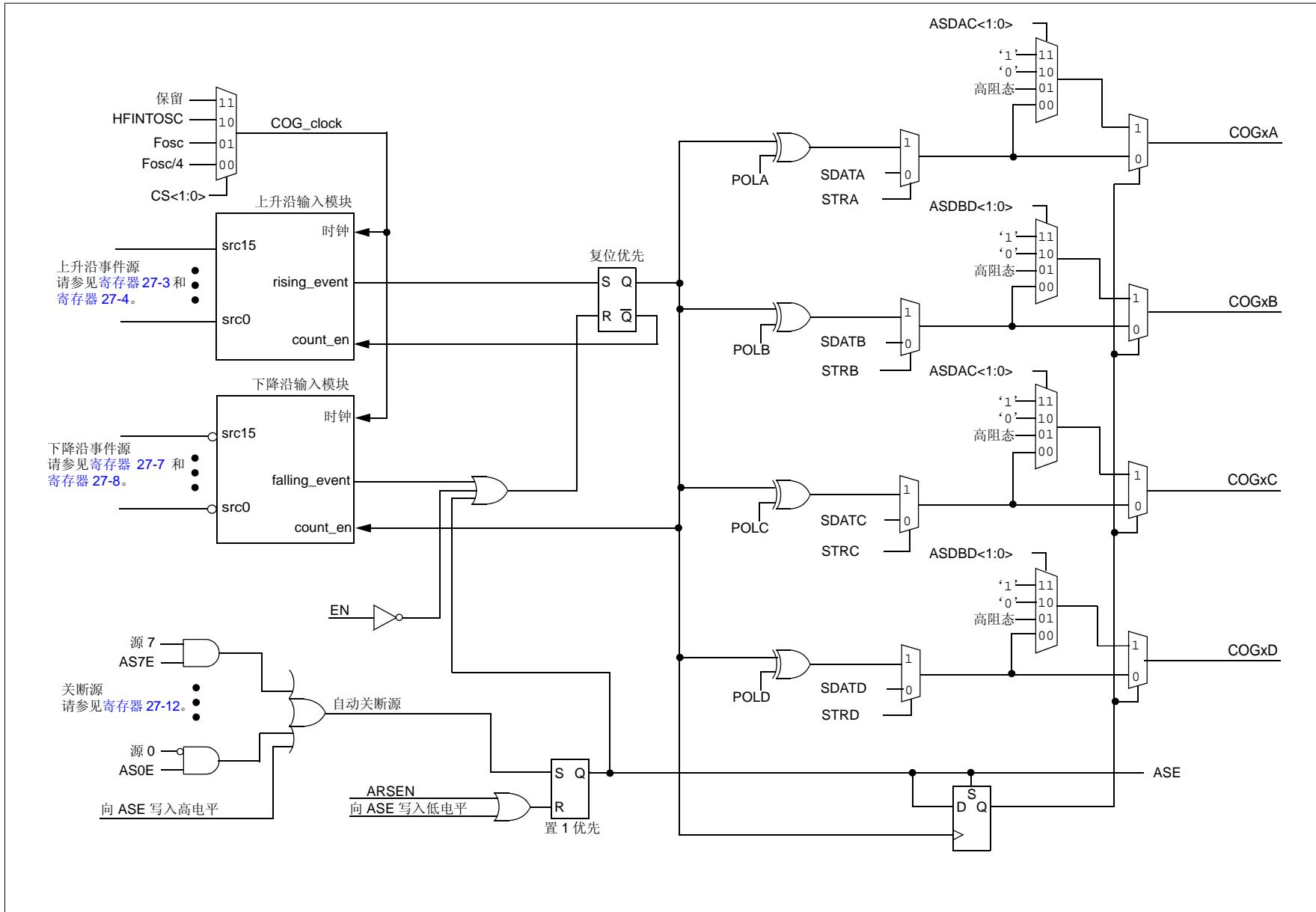


图 27-3: 简化 COG 框图 (同步转向 PWM 模式, MD = 1)

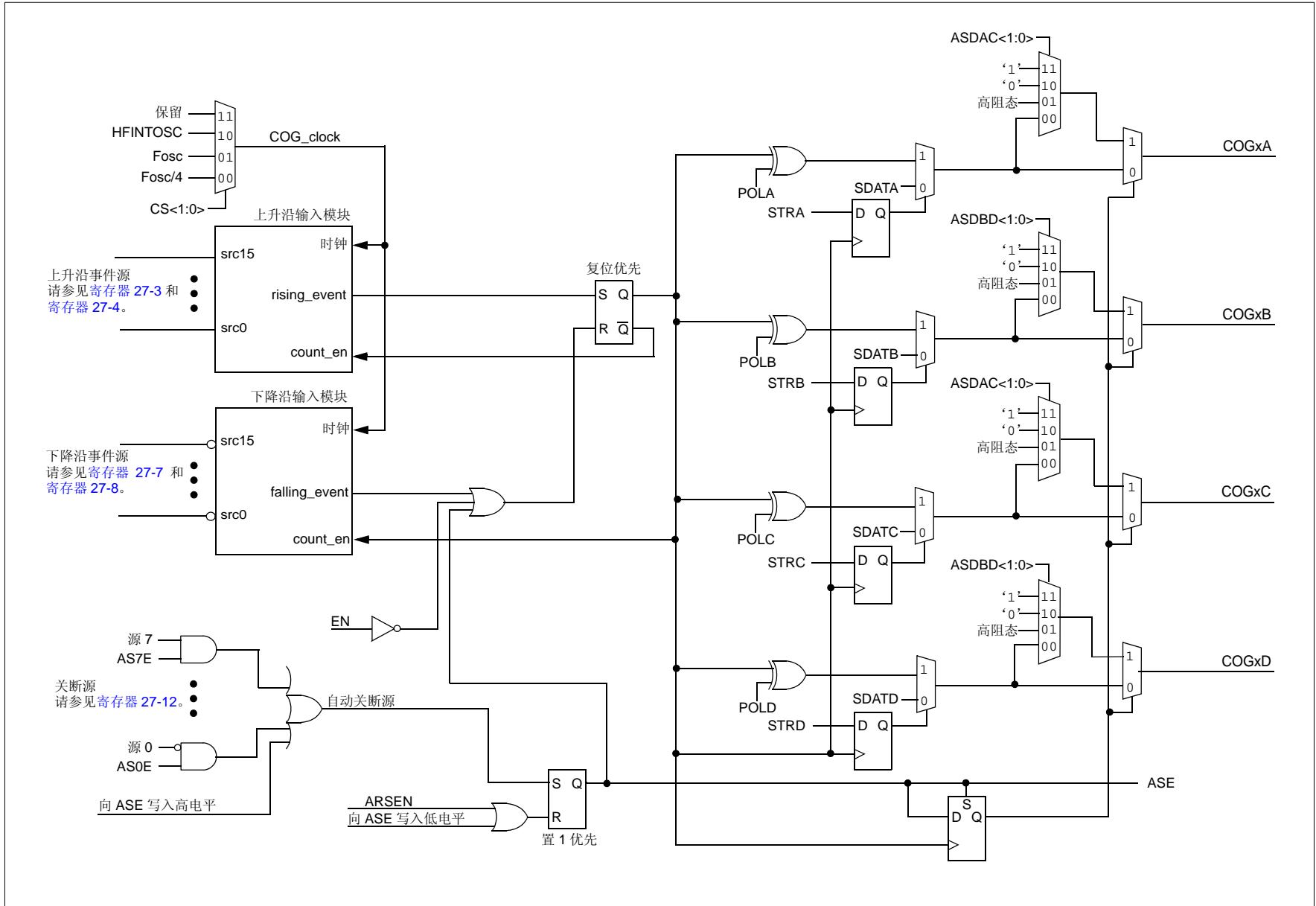


图 27-4: 简化 COG 框图 (全桥模式, 正向: MD = 2, 反向: MD = 3)

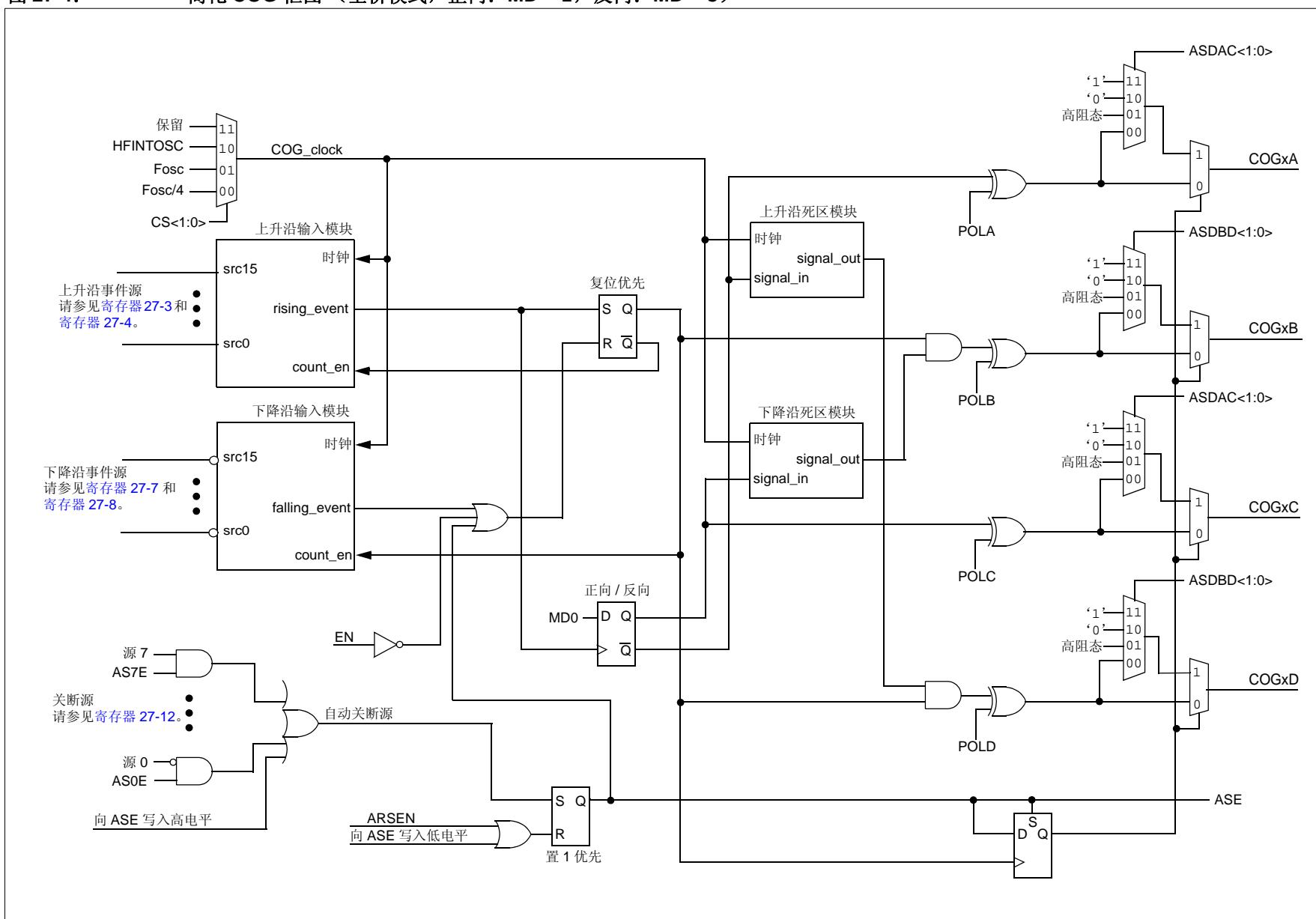


图 27-5: 简化 COG 框图 (半桥模式, MD = 4)

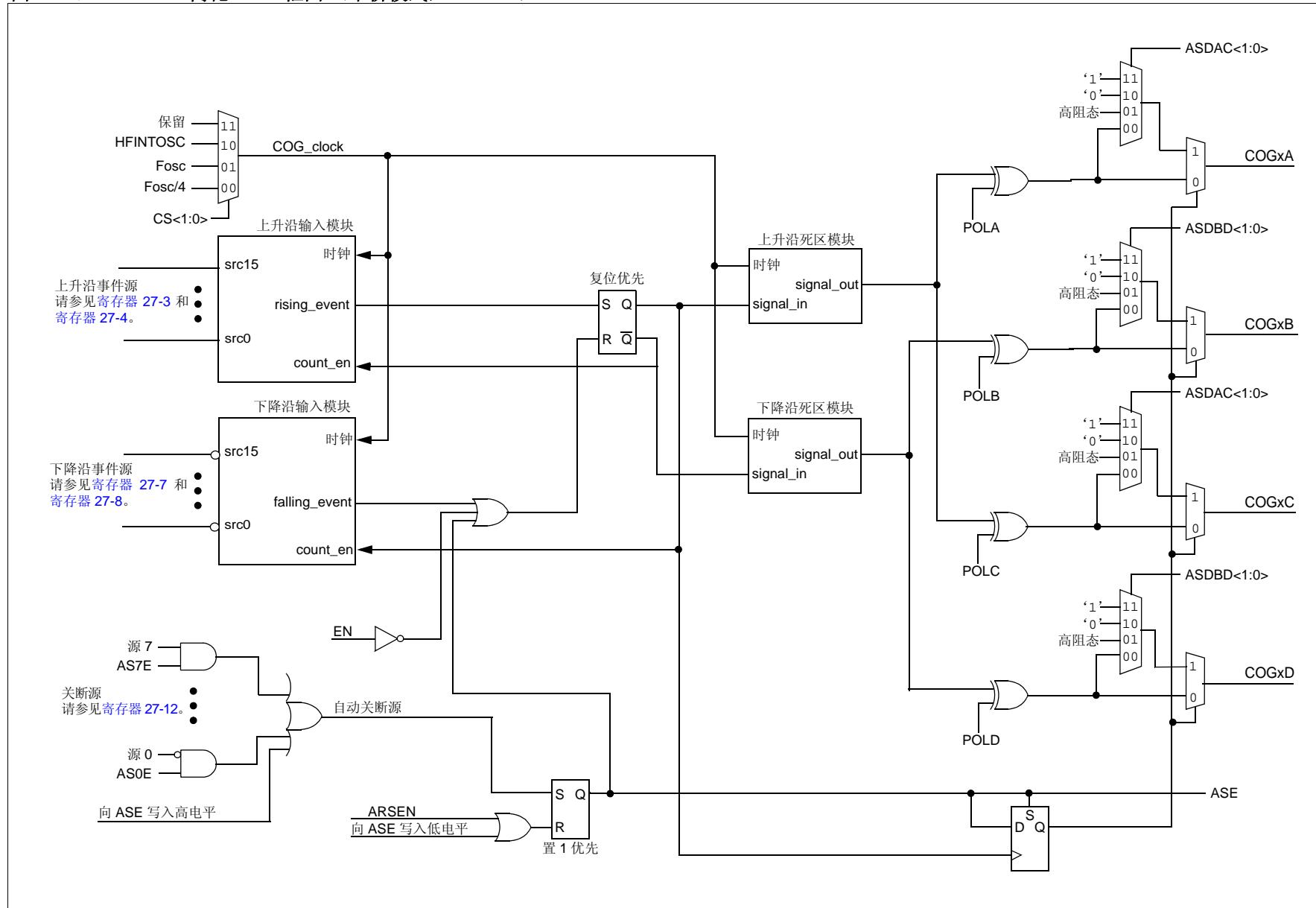
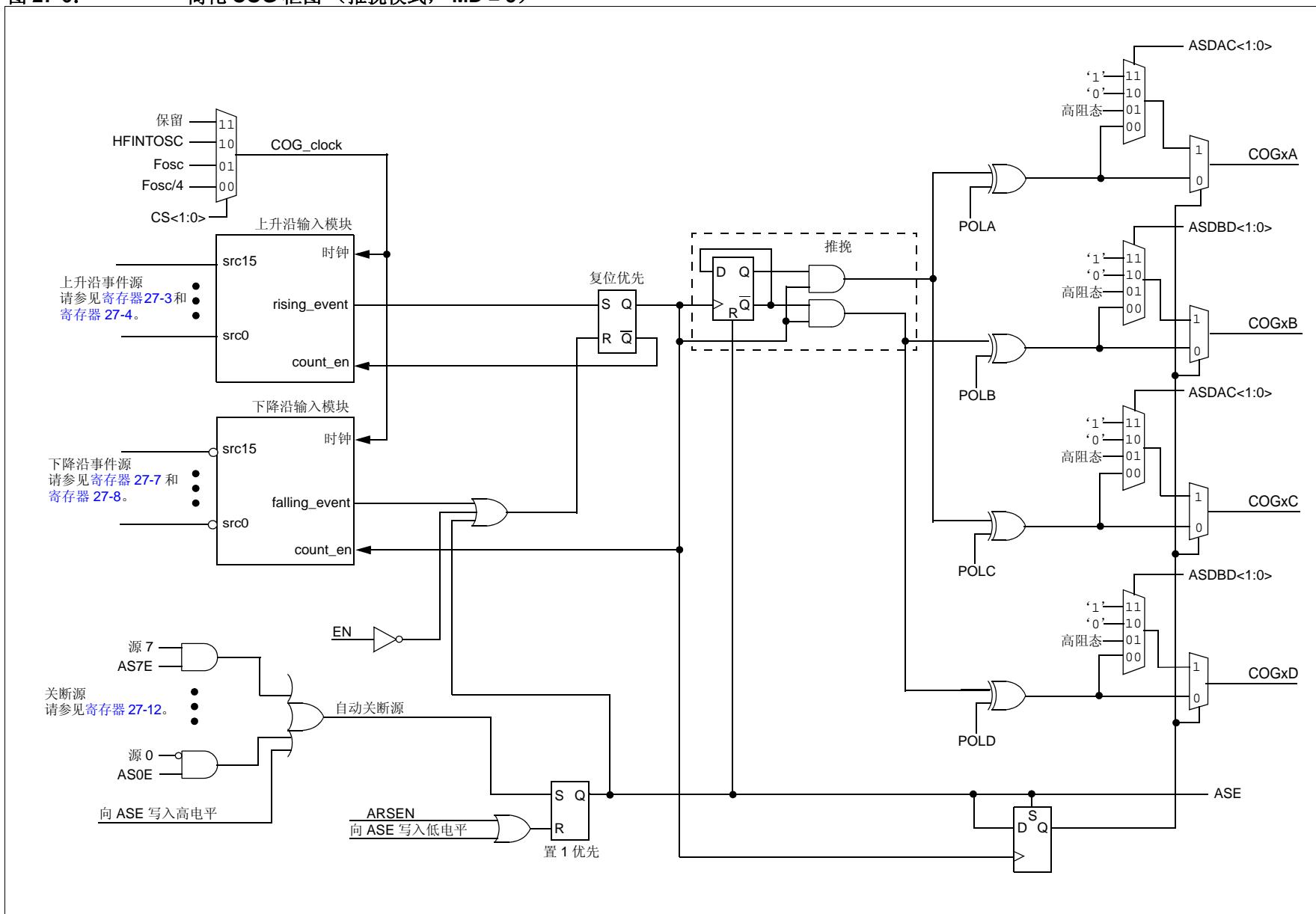


图 27-6: 简化 COG 框图 (推挽模式, MD = 5)



PIC16(L)F1764/5/8/9

图 27-7: COG (上升沿 / 下降沿) 输入模块

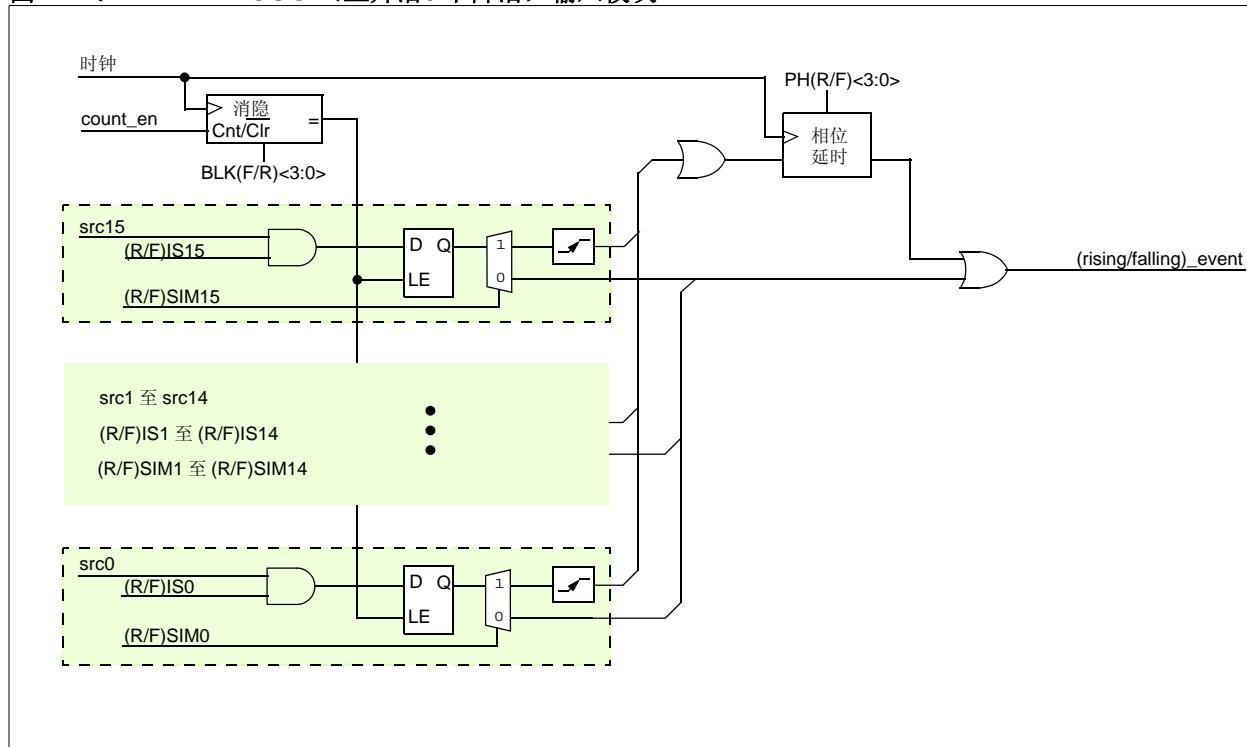


图 27-8: COG (上升沿 / 下降沿) 死区模块

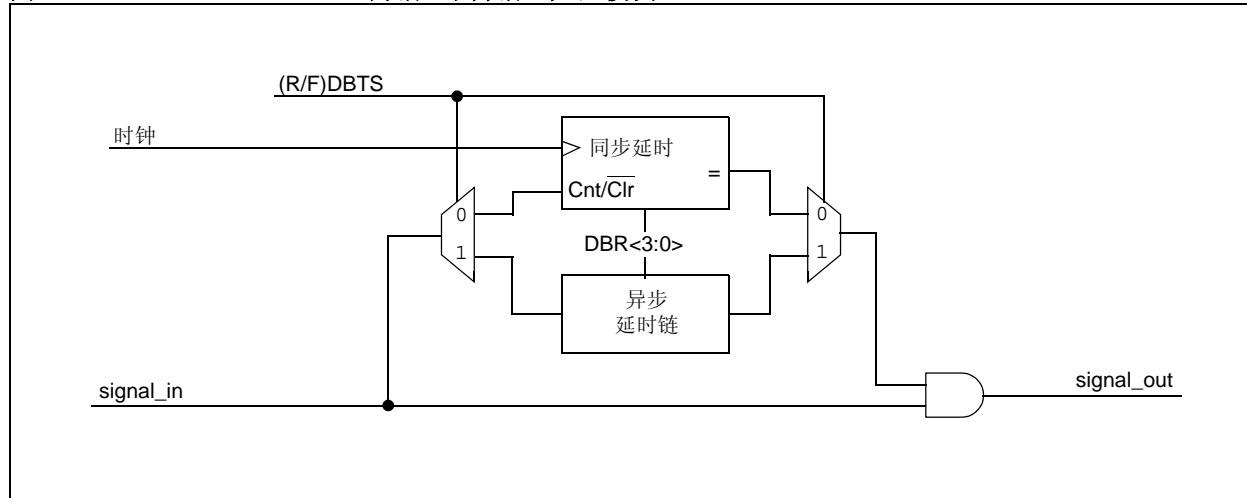


图 27-9: 使用 CCP1 时的典型半桥模式 COG 操作

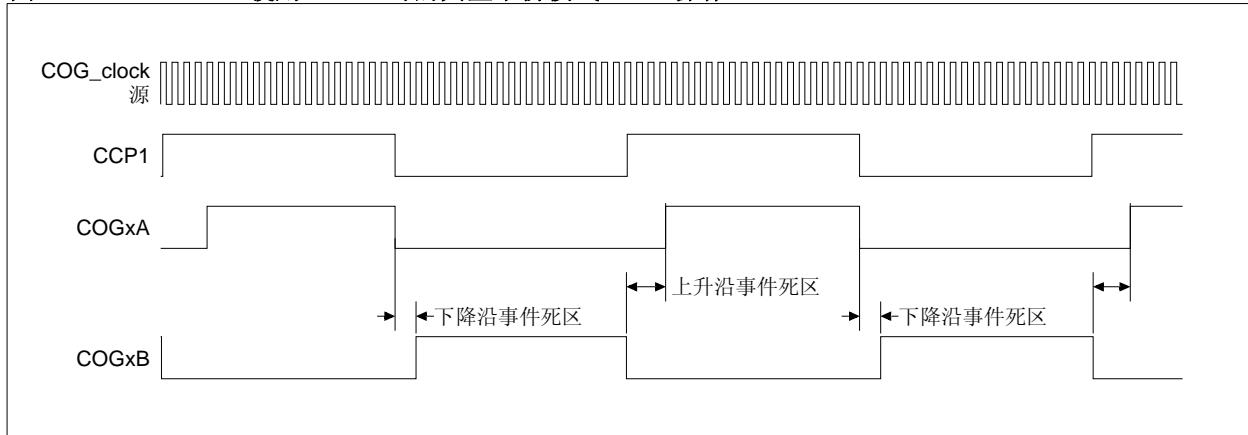


图 27-10: 使用 CCP1 和相位延时时的半桥模式 COG 操作

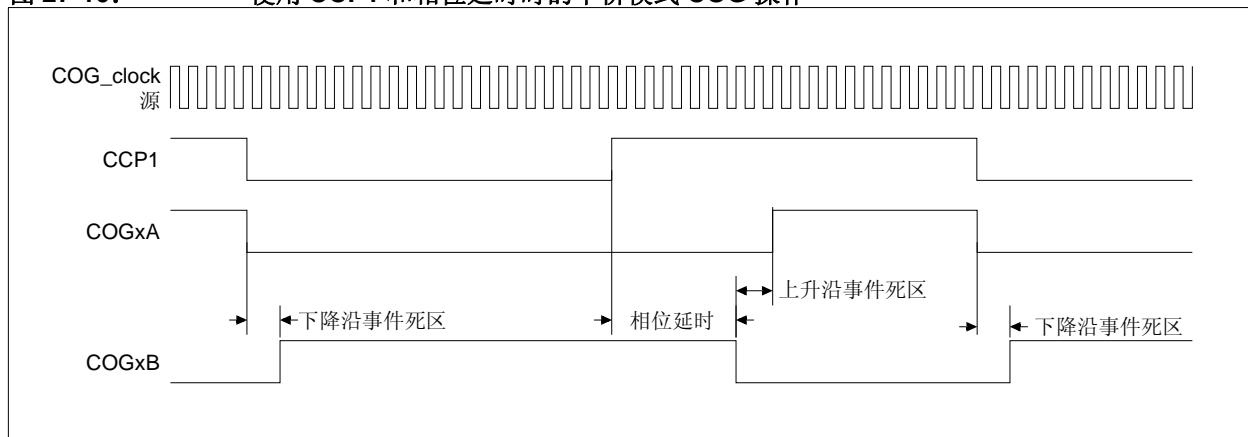
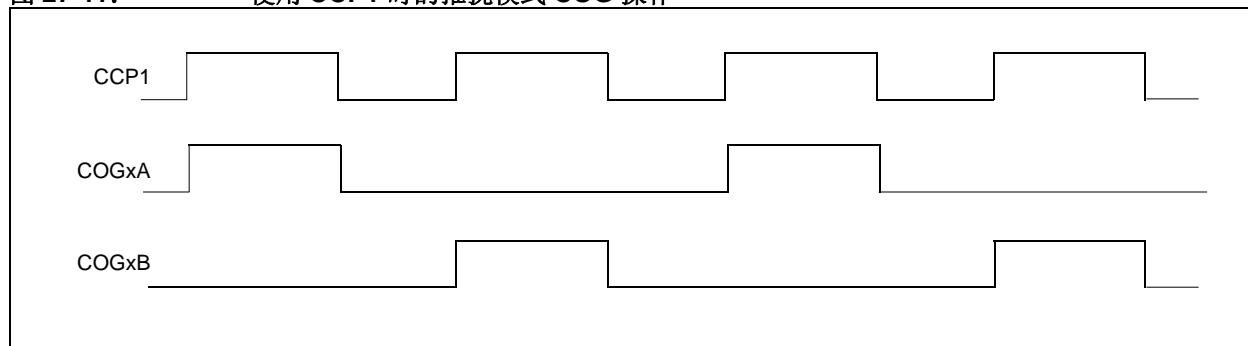


图 27-11: 使用 CCP1 时的推挽模式 COG 操作



PIC16(L)F1764/5/8/9

图 27-12: 使用 CCP1 时的全桥正向模式 COG 操作

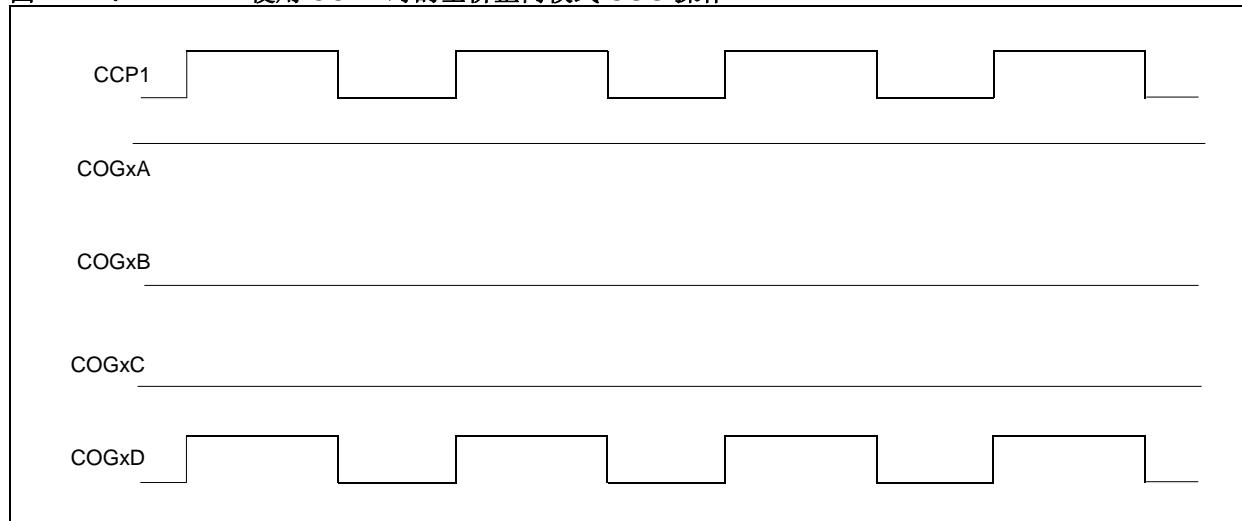
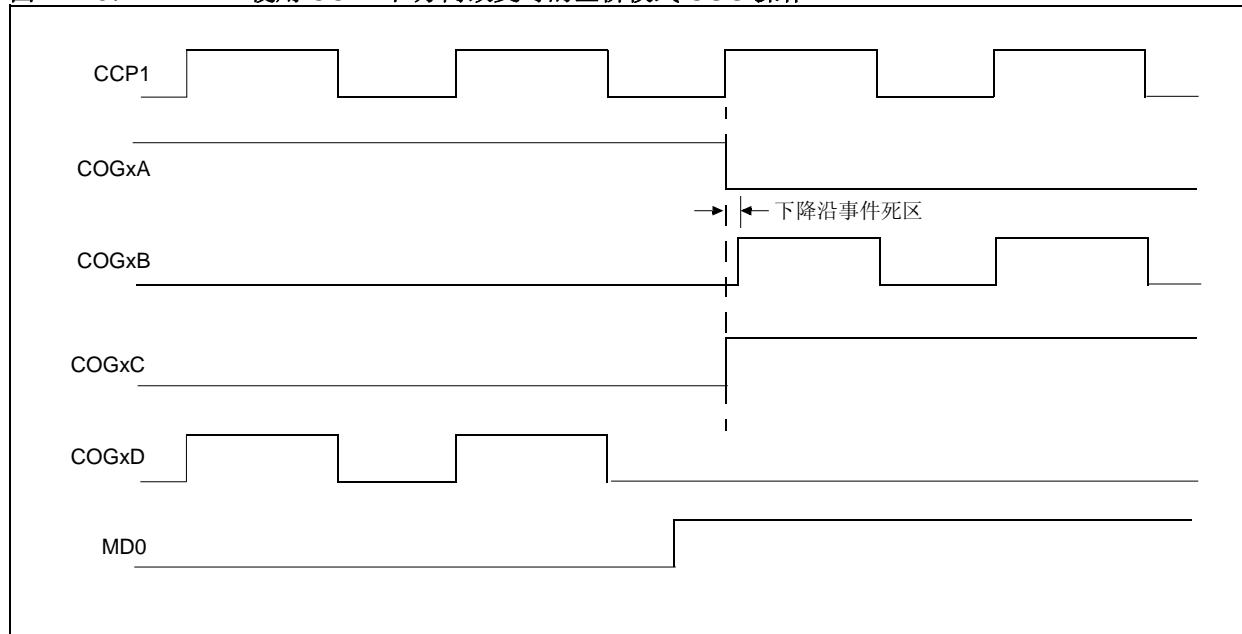


图 27-13: 使用 CCP1 和方向改变时的全桥模式 COG 操作



27.4 时钟源

COG_clock 用作外设中各种定时器的参考时钟。使用 COG_clock 的定时器包括：

- 上升沿和下降沿死区时间
- 上升沿和下降沿消隐时间
- 上升沿和下降沿事件相位延时

可供选择的时钟源包括：

- 16 MHz HFINTOSC (休眠期间工作)
- 指令时钟 (Fosc/4)
- 系统时钟 (Fosc)

时钟源使用 COGxCONO 寄存器（[寄存器 27-1](#)）的 CS<1:0> 位进行选择。

27.5 可选事件源

COG 使用可独立选择的事件源的任意组合来产生互补波形。事件源可分为两类：

- 上升沿事件源
- 下降沿事件源

上升沿事件源通过设置 COGxRIS0 和 COGxRIS1 寄存器（[寄存器 27-3](#) 和 [寄存器 27-4](#)）中的位来进行选择。下降沿事件源通过设置 COGxFIS0 和 COGxFIS1 寄存器（[寄存器 27-7](#) 和 [寄存器 27-8](#)）中的位来进行选择。所有选定的事件源进行逻辑或运算，以生成相应的事件信号。请参见图 27-7。

27.5.1 边沿与电平检测

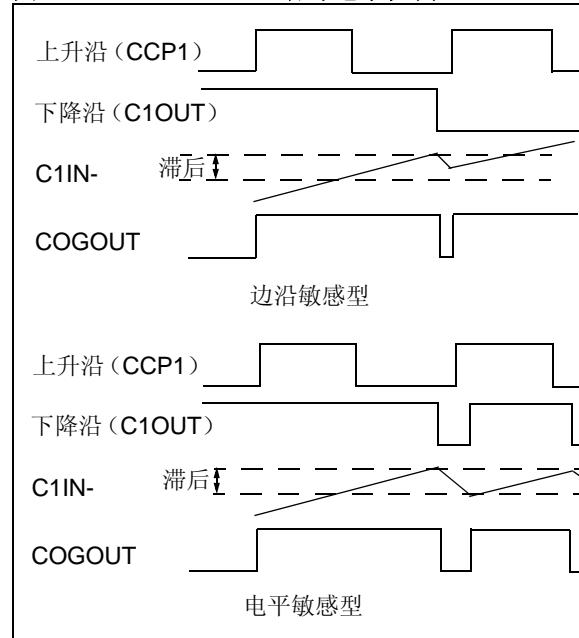
事件输入检测可以选择为电平或边沿敏感型。可单独选择每个事件源的检测模式。上升沿事件源检测模式使用 COGxRSIM0 和 COGxRSIM1 寄存器（[寄存器 27-6](#) 和 [寄存器 27-6](#)）进行选择。下降沿事件源检测模式使用 COGxFSIM0 和 COGxFSIM1 寄存器（[寄存器 27-9](#) 和 [寄存器 27-10](#)）进行选择。某个位置 1 会对相应事件源选择边沿检测。某个位清零会选择电平检测。

一般情况下，基于周期事件源驱动的事件应采用边沿检测，基于目标电路电压阈值产生的事件应为电平敏感型。假设存在以下两个示例：

1. 第一个示例是这样一个应用：周期由上升沿事件输入上的 50% 占空比时钟决定，COG 输出占空比由下降沿事件输入上通过比较器反馈的电压决定。如果时钟输入是电平敏感的，由于电平敏感时钟会抑制比较器反馈，占空比小于 50% 会表现出不稳定的操作。

2. 第二个示例类似于第一个示例，只是占空比接近于 100%。反馈比较器从高电平到低电平的跳变会使 COG 驱动器关闭，但周期事件源会几乎立刻重新开启该驱动器。如果关闭周期足够短，比较器输入可能无法达到滞后区间的下限，阻止输出发生变化。比较器输出会保持低电平，不会发生从高电平到低电平的跳变来触发边沿检测，COG 输出的驱动器会停滞在恒定的驱动器开启条件。请参见图 27-14。

图 27-14： 边沿与电平检测



27.5.2 上升沿事件

上升沿事件会启动 PWM 输出有效占空比周期。`rising_event`输出从低电平变为高电平即为上升沿事件。当上升沿事件相位延时和死区时间值均为零时，主输出会立即启动。否则，主输出会被延迟。上升沿事件源会导致以下所有操作：

- 启动上升沿事件相位延时计数器（如使能）
- 在相位延时后清零互补输出
- 启动下降沿事件输入消隐（如使能）
- 启动死区延时（如使能）
- 在死区延时到期之后置 1 主输出

27.5.3 下降沿事件

下降沿事件会终止 PWM 输出有效占空比周期。`falling_event`输出从高电平变为低电平即为下降沿事件。当下降沿事件相位延时和死区时间值均为零时，互补输出会立即启动。否则，互补输出会被延迟。下降沿事件源会导致以下所有操作：

- 启动下降沿事件相位延时计数器（如使能）
- 清零主输出
- 启动上升沿事件输入消隐（如使能）
- 启动下降沿事件死区延时（如使能）
- 在死区延时到期之后置 1 互补输出

27.6 输出控制

在禁止时，或紧接着在使能 COG 模块之后，主 COG 输出处于无效状态，互补 COG 输出处于有效状态。

27.6.1 输出使能

COG 模块中不存在输出使能控制。每个器件引脚具有独立的输出选择控制，称为 PPS 寄存器。全部 4 个 COG 输出均可在每个引脚的 PPS 寄存器中进行选择。

当通过 PPS 选择使能某个 COG 输出时，引脚上的输出具有几种可能性，这取决于模式、转向控制、EN 位和关断状态，如表 27-1 和表 27-2 所示。

表 27-1：引脚输出状态，MD_{2:0} = 00X

EN	STR 位	关断	输出
x	0	无效	静态转向数据
x	1	有效	关断改写
0	1	无效	无效状态
1	1	无效	有效 PWM 信号

表 27-2：引脚输出状态，MD_{2:0} > 001

EN	STR 位	关断	输出
x	x	无效	无效状态
x	x	有效	关断改写
1	x	无效	有效 PWM 信号

27.6.2 极性控制

每个 COG 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，极性只会在 4 种关断改写模式之下影响输出。更多详细信息，请参见第 27.10 节“自动关断控制”。

输出极性通过 COGxCON1 寄存器（寄存器 27-2）的 POLA 至 POLD 位进行选择。

27.7 死区控制

死区控制用于提供不重叠的 PWM 输出信号，以防止外部功率开关中产生直通电流。只有在半桥模式下，以及在全桥模式下改变方向时，死区时间才会影响输出。

COG 包含两个死区定时器。一个死区定时器用于上升沿事件死区控制。另一个用于下降沿事件死区控制。定时器模式可选择为：

- 异步延时链
- 同步计数器

上升沿事件和下降沿事件死区时间的死区定时器模式通过 COGxCON1 寄存器（寄存器 27-2）中相应的 RDBS 和 FDBS 位进行选择。

在半桥模式下，上升沿事件死区时间会在上升沿事件之后延迟所有选定的主输出，阻止它们在选定的死区时间内变为有效。COGxA 和 COGxC 是半桥模式下的主输出。

在半桥模式下，下降沿事件死区时间会在下降沿事件之后延迟所有选定的互补输出，阻止它们在选定的死区时间内变为有效。COGxB 和 COGxD 是半桥模式下的互补输出。

在全桥模式下，只有在方向改变过程中才会出现死区延时。在方向从正向变为反向之后，调制输出会被延迟，延时等于下降沿事件死区时间。在方向从反向变为正向之后，调制输出会被延迟，延时等于上升沿事件死区时间。

27.7.1 异步延时链死区延时

异步死区延时由输入传播通过一系列延时元件所需的时间决定。每个延时元件的标称延时为 5 纳秒。

对于上升沿事件异步死区延时，将 COGxCON0 寄存器的 RDBS 位置 1，并将 COGxDBR 寄存器（[寄存器 27-14](#)）值设置为上升沿事件死区时间中所需的延时元件数。

对于下降沿事件异步死区延时，将 COGxCON0 寄存器的 FDBS 位置 1，并将 COGxDBF 寄存器（[寄存器 27-15](#)）值设置为下降沿事件死区时间中所需的延时元件数。

将该值设置为零会禁止死区延时。

27.7.2 同步计数器死区延时

同步计数器死区的计时方式是对 COG_clock 周期进行计数，计数范围为零至死区计数寄存器中的值。使用[公式 27-1](#) 可计算死区时间。

对于上升沿事件同步死区延时，将 COGxCON0 寄存器的 RDBS 位清零，并将 COGxDBR 计数寄存器值设置为上升沿事件死区时间中的 COG_clock 周期数。

对于下降沿事件同步死区延时，将 COGxCON0 寄存器的 FDBS 位清零，并将 COGxDBF 计数寄存器值设置为下降沿事件死区时间中的 COG_clock 周期数。

值为零时，死区延时会被禁止。

27.7.3 同步计数器死区时间的偏差

当触发死区计数器的上升沿和下降沿事件来自异步输入时，它会在同步计数器死区时间中产生偏差。最大偏差等于 1 个 COG_clock 周期。更多详细信息，请参见[例 27-1](#)。

当事件输入源是异步的，且没有相位延时时，可以使用异步延时链死区模式来避免死区时间的偏差。

27.7.4 上升沿事件死区

上升沿事件死区会延迟主输出的开启，使之与互补输出的关闭时间点相偏离。上升沿事件死区时间在 rising_event 输出变为真时开始。

关于设置上升沿死区时间的更多信息，请参见[第 27.7.1 节 “异步延时链死区延时”](#) 和 [第 27.7.2 节 “同步计数器死区延时”](#)。

27.7.5 下降沿事件死区

下降沿事件死区会延迟互补输出的开启，使之与主输出的关闭时间点相偏离。下降沿事件死区时间在 falling_event 输出变为真时开始。

关于设置上升沿死区时间的更多信息，请参见[第 27.7.1 节 “异步延时链死区延时”](#) 和 [第 27.7.2 节 “同步计数器死区延时”](#)。

27.7.6 死区重叠

可能出现的死区重叠有两种情况：

- 上升沿到下降沿
- 下降沿到上升沿

27.7.6.1 上升沿到下降沿的重叠

这种情况下，在上升沿事件死区计数器仍在计数时发生下降沿事件。发生这种情况时，主驱动器会被抑制，死区会发生延长，延长量等于下降沿事件死区时间。在延长的死区时间终止时，互补驱动器会变为真。

27.7.6.2 下降沿到上升沿的重叠

这种情况下，在下降沿事件死区计数器仍在计数时发生上升沿事件。发生这种情况时，互补驱动器会被抑制，死区会发生延长，延长量等于上升沿事件死区时间。在延长的死区时间终止时，主驱动器会变为真。

27.8 消隐控制

输入消隐是一种可将事件输入屏蔽或消隐一小段时间的功能。它用于防止由于开启 / 关闭功率元件时的电瞬变所导致的错误输入事件。

COG 包含两个消隐计数器，一个由上升沿事件触发，另一个由下降沿事件触发。这两个计数器与它们消隐的事件交叉配合使用。下降沿事件消隐计数器用于消隐上升沿输入事件，上升沿事件消隐计数器用于消隐下降沿输入事件。在启动之后，消隐将持续相应的消隐计数器所指定的时间。

消隐的计时方式是对 COG_clock 周期计数，计数范围为零至消隐计数寄存器中的值。使用[公式 27-1](#) 可计算消隐时间。

27.8.1 上升沿事件输入的下降沿事件消隐

下降沿事件消隐计数器会禁止上升沿事件输入触发上升沿事件。下降沿事件消隐时间在 **rising_event** 输出驱动器变为假时开始。

下降沿事件消隐时间通过 COGxBLKF 寄存器（[寄存器 27-17](#)）中包含的值设置。消隐时间使用 [公式 27-1](#) 计算。

当 COGxBLKF 值为零时，下降沿事件消隐会被禁止，消隐计数器输出为真，从而使事件信号可以直接传递到事件触发电路。

27.8.2 下降沿事件输入的上升沿事件消隐

上升沿事件消隐计数器会禁止下降沿事件输入触发下降沿事件。上升沿事件消隐时间在 **falling_event** 输出驱动器变为假时开始。

上升沿事件消隐时间通过 COGxBLKR 寄存器（[寄存器 27-16](#)）中包含的值设置。

当 COGxBLKR 值为零时，上升沿事件消隐会被禁止，消隐计数器输出为真，从而使事件信号可以直接传递到事件触发电路。

27.8.3 消隐时间的偏差

当触发消隐计数器的上升沿和下降沿事件源与 COG_clock 异步时，它会在消隐时间中产生偏差。最大偏差等于 1 个 COG_clock 周期。更多详细信息，请参见 [公式 27-1](#) 和 [例 27-1](#)。

27.9 相位延时

可以对上升沿事件和 / 或下降沿事件置为有效的时间进行延迟。这通过将一个非零值分别放入 COGxPHR 或 COGxPHF 相位延时计数寄存器（[寄存器 27-18](#) 和 [寄存器 27-19](#)）来实现。关于使用 CCP1 和相位延时的 COG 操作，请参见 [图 27-10](#)。从输入上升沿事件信号切换到事件实际置为有效的延时的计算方式与死区和消隐延时相同。请参见 [公式 27-1](#)。

当相位延时计数值为零时，相位延时会被禁止，相位延时计数器输出为真，从而使事件信号可以直接传递到互补输出驱动器触发器。

27.9.1 累积的偏差

连续几个阶段产生的偏差不会大于 1 个 COG_clock。假设相位延时阶段位于消隐阶段之后，死区阶段位于消隐或相位延时阶段之后，消隐阶段位于死区阶段之后。当使能前一个阶段时，该阶段的输出一定与 COG_clock 同步，这可以消除在后继阶段中产生偏差的任何可能性。

公式 27-1：相位、死区和消隐时间计算

$$T_{\min} = \frac{\text{Count}}{F_{COG_clock}}$$

$$T_{\max} = \frac{\text{Count} + 1}{F_{COG_clock}}$$

$$T_{\text{uncertainty}} = T_{\max} - T_{\min}$$

并且：

$$T_{\text{uncertainty}} = \frac{1}{F_{COG_clock}}$$

其中：

T	计数
上升沿相位延时	COGxPHR
下降沿相位延时	COGxPHF
上升沿死区	COGxDBR
下降沿死区	COGxDBF
上升沿事件消隐	COGxBLKR
下降沿事件消隐	COGxBLKF

例 27-1：定时器偏差

给定：

$$Count = Ah = 10d$$

$$F_{COG_Clock} = 8 \text{ MHz}$$

因此：

$$\begin{aligned} T_{\text{uncertainty}} &= \frac{1}{F_{COG_clock}} \\ &= \frac{1}{8 \text{ MHz}} = 125 \text{ ns} \end{aligned}$$

证明：

$$\begin{aligned} T_{\min} &= \frac{Count}{F_{COG_clock}} \\ &= 125 \text{ ns} \cdot 10d = 1.25 \mu\text{s} \\ T_{\max} &= \frac{Count + 1}{F_{COG_clock}} \\ &= 125 \text{ ns}^2 (10d + 1) \\ &= 1.375 \mu\text{s} \end{aligned}$$

因此：

$$\begin{aligned} T_{\text{uncertainty}} &= T_{\max} - T_{\min} \\ &= 1.375 \mu\text{s} - 1.25 \mu\text{s} \\ &= 125 \text{ ns} \end{aligned}$$

27.10 自动关断控制

自动关断是一种使用特定改写立即改写 COG 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。在两种情况下，关断改写都会一直保持有效，直到清除关断之后发生第一个上升沿事件为止。

27.10.1 关断

关断状态可以通过以下两种机制之一进入：

- 软件生成
- 外部输入

27.10.1.1 由软件产生的关断

将 COGxASD0 寄存器（寄存器 27-11）的 ASE 位置 1 可以强制 COG 进入关断状态。

在禁止自动重启时，关断状态会一直保持，直到用软件清零 ASE 位之后发生第一个上升沿事件为止。

在使能自动重启时，ASE 位会自动清零，并在关断输入清除之后发生第一个上升沿事件时继续工作。请参见图 27-15 和第 27.10.3.2 节“自动重启”。

27.10.1.2 外部关断源

外部关断输入提供了在出现故障条件时安全地暂停 COG 工作的最快办法。当选定的任意关断输入变为真时，输出驱动器锁存器会发生复位，COG 输出会立即变为选定的改写电平，无任何软件延时。

可以选择输入源的任意组合来产生关断条件。当选定的输入源为低电平时会发生关断。关断输入源包括：

- 通过 COGxINPPS 控制选择的任意输入引脚
- 比较器 1
- 比较器 2
- 比较器 3
- 比较器 4
- CLC2 输出
- Timer2 输出
- Timer4 输出

关断输入通过 COGxASD1 寄存器（寄存器 27-12）的位独立进行选择。

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。

27.10.2 引脚改写电平

在关断有效时驱动到输出引脚的电平由 COGxASD0 寄存器（寄存器 27-11）的 ASDAC<1:0> 和 ASDBC<1:0> 位控制。ASDAC<1:0> 控制 COGxA 和 COGxC 的改写电平，ASDBC<1:0> 控制 COGxB 和 COGxD 的改写电平。每个输出对有 4 个改写选项：

- 强制低电平
- 强制高电平
- 三态
- PWM 无效状态（与下降沿事件引起的状态相同）

注： 极性控制不适用于强制低电平和强制高电平改写电平，但适用于 PWM 无效状态。

27.10.3 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

重启方法使用 COGxASD0 寄存器的 ARSEN 位进行选择。[图 27-15](#) 给出了软件控制的自动重启的波形。

27.10.3.1 软件控制重启

当 COGxASD0 寄存器的 ARSEN 位清零时，在发生自动关断事件之后，软件必须通过清零 ASE 位来重启 COG 操作。

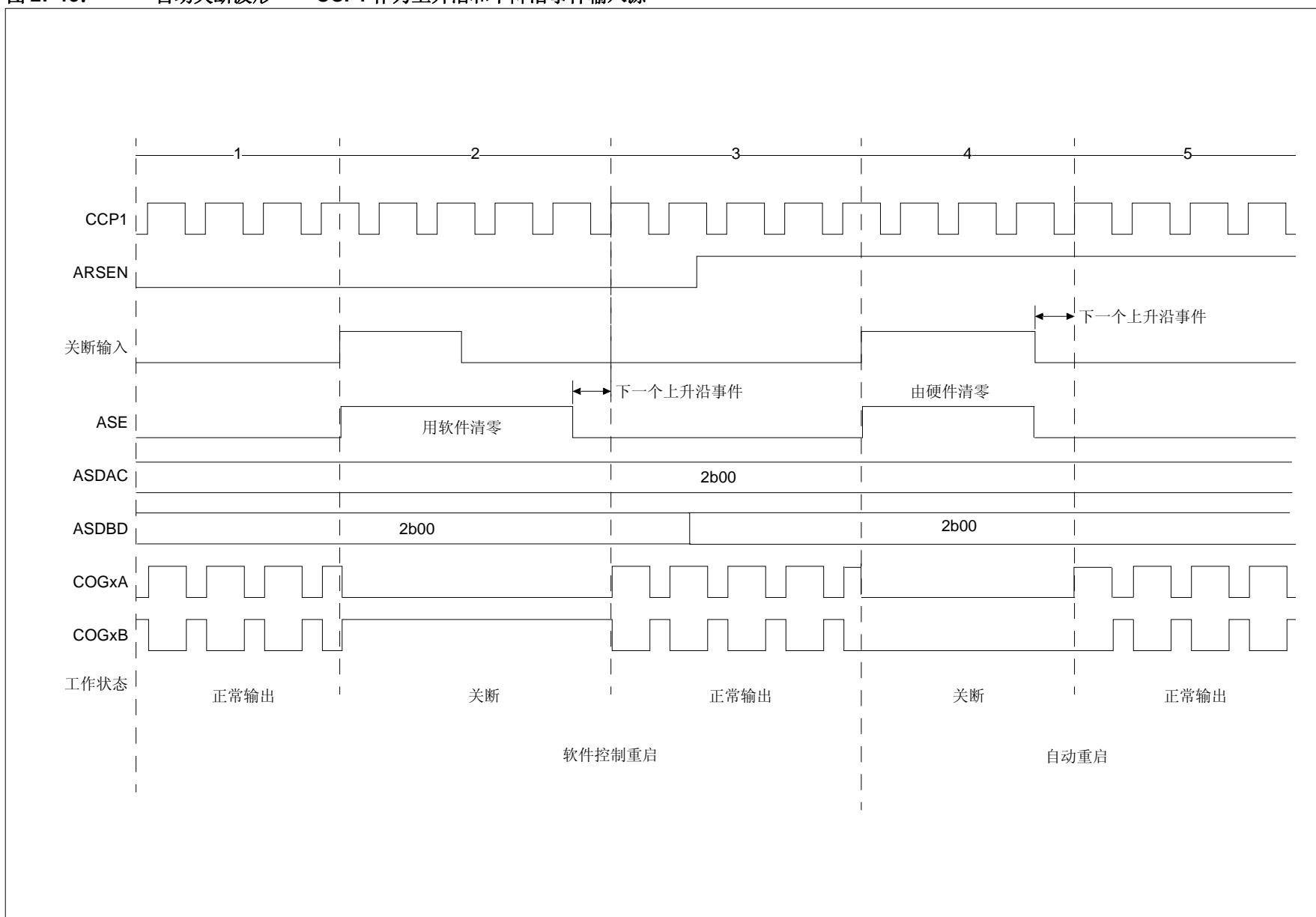
COG 将在 ASE 位清零之后发生第一个上升沿事件时恢复工作。清除关断状态要求所有选定的关断输入为假，否则 ASE 位将保持置 1。

27.10.3.2 自动重启

当 COGxASD0 寄存器的 ARSEN 位置 1 时，COG 将从自动关断状态中自动重启。

ASE 位将自动清零，COG 将在所有选定关断输入变为假之后发生第一个上升沿事件时恢复工作。

图 27-15: 自动关断波形 —— CCP1 作为上升沿和下降沿事件输入源



27.11 缓冲区更新

在 COG 操作期间，对相位、死区和消隐计数寄存器的更改需要同时发生，以避免由于每个寄存器写操作之间的延时而可能出现的意外操作。这通过 COGxCON0 寄存器的 LD 位以及相位、消隐和死区计数寄存器的双重缓冲功能实现。

在使能 COG 模块之前，写计数寄存器会向计数缓冲区中装入数据，不需要 LD 位。但是，当使能 COG 时，在写计数寄存器之后，计数缓冲区更新会被暂停，直到 LD 位置 1 为止。当 LD 位置 1 时，相位、死区和消隐寄存器值会被传送到相应的缓冲区，与 COG 操作同步完成。LD 位在传送操作完成时由硬件清零。

27.12 输入和输出引脚选择

COG 可以选择一个器件引脚作为输入。这个输入可以用作上升沿和下降沿事件源或故障源。COGxINPPS 寄存器用于选择该引脚。请参见寄存器 [xxxPPS](#)（[寄存器 12-1](#)）和 [RxyPPS](#)（[寄存器 12-2](#)）。

引脚 PPS 控制寄存器用于使能 COG 输出。引脚可以使用任意输出组合，包括多个引脚使用相同的输出。更多详细信息，请参见 [RxyPPS](#) 控制寄存器和[第 12.2 节“PPS 输出”](#)。

27.13 休眠期间的操作

在休眠模式下，只要 COG_clock、上升沿事件和下降沿事件源仍保持工作，COG 就会继续工作。

如果使能了 COG，并选择 HFINTOSC 作为 COG_clock 源，则 HFINTSOC 会在休眠期间保持工作。

27.14 配置 COG

以下步骤说明了如何正确配置 COG 来确保使用上升沿事件输入同步启动：

1. 如果要将某个引脚用作 COG 故障或事件输入，则使用 COGxINPPS 寄存器来配置所需的引脚。
2. 清零与用于 COG 功能的引脚相关联的所有 ANSEL 寄存器位。
3. 确保对应于要使用的 COG 输出的 TRIS 控制位置 1，从而将它们全部配置为输入。COG 模块稍后将根据需要使能输出驱动器。
4. 清零 EN 位（如果尚未清零）。

5. 使用 COGxDBR 和 COGxDBF 寄存器设置所需的死区时间，并使用 COGxCON1 寄存器的 RDBS 和 FDBS 位选择时序源。
6. 使用 COGxBLKR 和 COGxBLKF 寄存器设置所需的消隐时间。
7. 使用 COGxPHR 和 COGxPHF 寄存器设置所需的相位延时。
8. 使用 COGxASD1 寄存器选择所需的关断源。
9. 在 COGxASD0 自动关断寄存器中设置以下控制：
 - 将两个输出改写控制选择为所需电平（这是必需的，即使不使用自动关断，因为启动将从关断状态开始）。
 - 将 ASE 位置 1，将 ARSEN 位清零。
10. 使用 COGxRIS0、COGxRIS1、COGxFIS0 和 COGxFIS1 寄存器选择所需的上升沿和下降沿事件源。
11. 使用 COGxRSIM0、COGxRSIM1、COGxFSIM0 和 COGxFSIM1 寄存器选择所需的上升沿和下降沿事件模式。
12. 在 COGxCON1 寄存器中配置以下控制：
 - 设置每个输出的极性
 - 选择所需的死区时序源
13. 在 COGxCON0 寄存器中配置以下控制：
 - 设置所需的工作模式
 - 选择所需的时钟源
14. 如果选择了某个转向模式，则在 COGxSTR 寄存器中配置以下控制：
 - 将要使用的输出的转向位置 1。
 - 设置所需的静态电平。
15. 将 EN 位置 1。
16. 设置引脚 PPS 控制，将 COG 输出定向到所需的引脚。
17. 如果要使用自动重启，则将 ARSEN 位置 1，ASE 将会自动清零。否则，通过清零 ASE 位来启动 COG。

27.15 寄存器定义：COG 控制

表 27-3 列出了 COG 外设的长位名称前缀。更多信息，请参见第 1.1 节 “寄存器和位命名约定”。

表 27-3：

外设	位名称前缀
COG1	G1
COG2 ⁽¹⁾	G2

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 27-1：COGxCON0：COG 控制寄存器 0

R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD	—	CS<1:0>		MD<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

q = 值取决于具体条件

- bit 7 **EN:** COGx 使能位
1 = 使能模块
0 = 禁止模块
- bit 6 **LD:** COGx 装入缓冲区位
1 = 相位、消隐和死区缓冲区将在发生下一个输入事件时装入寄存器值
0 = 寄存器到缓冲区的传送操作已完成
- bit 5 未实现：读为 0
- bit 4-3 **CS<1:0>:** COGx 时钟选择位
11 = 保留。不要使用。
10 = COG_clock 为 HFINTOSC（在休眠期间保持工作状态）
01 = COG_clock 为 Fosc
00 = COG_clock 为 Fosc/4
- bit 2-0 **MD<2:0>:** COGx 模式选择位
11x = 保留。不要使用。
101 = COG 输出在推挽模式下工作
100 = COG 输出在半桥模式下工作
011 = COG 输出在反向全桥模式下工作
010 = COG 输出在正向全桥模式下工作
001 = COG 输出在同步转向 PWM 模式下工作
000 = COG 输出在转向 PWM 模式下工作

寄存器 27-2: COGxCON1: COG 控制寄存器 1

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
RDBS	FDBS	—	—	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 **RDBS:** COGx 上升沿事件死区时序源选择位
1 = 延时链和 COGxDBR 用于死区时序生成
0 = COGx_clock 和 COGxDBR 用于死区时序生成
- bit 6 **FDBS:** COGx 下降沿事件死区时序源选择位
1 = 延时链和 COGxDBF 用于死区时序生成
0 = COGx_clock 和 COGxDBF 用于死区时序生成
- bit 5-4 未实现: 读为 0
- bit 3 **POLD:** COGxD 输出极性控制位
1 = COGxD 输出的有效电平为低电平
0 = COGxD 输出的有效电平为高电平
- bit 2 **POLC:** COGxC 输出极性控制位
1 = COGxC 输出的有效电平为低电平
0 = COGxC 输出的有效电平为高电平
- bit 1 **POLB:** COGxB 输出极性控制位
1 = COGxB 输出的有效电平为低电平
0 = COGxB 输出的有效电平为高电平
- bit 0 **POLA:** COGxA 输出极性控制位
1 = COGxA 输出的有效电平为低电平
0 = COGxA 输出的有效电平为高电平

寄存器 27-3: COGxRIS0: COG 上升沿事件输入选择寄存器

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| RIS7 | RIS6 | RIS5 | RIS4 | RIS3 | RIS2 | RIS1 | RIS0 |
| 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 | RIS7: COGx 上升沿事件输入源 7 使能位
1 = 使能 PWM3 输出作为上升沿事件输入
0 = PWM3 输出对上升沿事件没有任何作用 |
| bit 6 | RIS6: COGx 上升沿事件输入源 6 使能位
1 = 使能 CCP2 输出作为上升沿事件输入
0 = CCP2 输出对上升沿事件没有任何作用 |
| bit 5 | RIS5: COGx 上升沿事件输入源 5 使能位
1 = 使能 CCP1 输出作为上升沿事件输入
0 = CCP1 输出对上升沿事件没有任何作用 |
| bit 4 | RIS4: COGx 上升沿事件输入源 4 使能位
1 = 使能比较器 4 输出作为上升沿事件输入
0 = 比较器 4 输出对上升沿事件没有任何作用 |
| bit 3 | RIS3: COGx 上升沿事件输入源 3 使能位
1 = 使能比较器 3 输出作为上升沿事件输入
0 = 比较器 3 输出对上升沿事件没有任何作用 |
| bit 2 | RIS2: COGx 上升沿事件输入源 2 使能位
1 = 使能比较器 2 输出作为上升沿事件输入
0 = 比较器 2 输出对上升沿事件没有任何作用 |
| bit 1 | RIS1: COGx 上升沿事件输入源 1 使能位
1 = 使能比较器 1 输出作为上升沿事件输入
0 = 比较器 1 输出对上升沿事件没有任何作用 |
| bit 0 | RIS0: COGx 上升沿事件输入源 0 使能位
1 = 使能通过 COGxINPPS 控制寄存器选择的引脚作为上升沿事件输入
0 = 通过 COGxINPPS 控制寄存器选择的引脚对上升沿事件没有任何作用 |

PIC16(L)F1764/5/8/9

寄存器 27-4: COGxRIS1: COG 上升沿事件输入选择寄存器

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
—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8
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 **RIS13:** COGx 上升沿事件输入源 13 使能位
1 = 使能 CLC3 输出作为上升沿事件输入
0 = CLC3 输出对上升沿事件没有任何作用
- bit 4 **RIS12:** COGx 上升沿事件输入源 12 使能位
1 = 使能 CLC2 输出作为上升沿事件输入
0 = CLC2 输出对上升沿事件没有任何作用
- bit 3 **RIS11:** COGx 上升沿事件输入源 11 使能位
1 = 使能 CLC1 输出作为上升沿事件输入
0 = CLC1 输出对上升沿事件没有任何作用
- bit 2 **RIS10:** COGx 上升沿事件输入源 10 使能位
1 = 使能 PWM6 输出作为上升沿事件输入
0 = PWM6 输出对上升沿事件没有任何作用
- bit 1 **RIS9:** COGx 上升沿事件输入源 9 使能位
1 = 使能 PWM5 输出作为上升沿事件输入
0 = PWM5 输出对上升沿事件没有任何作用
- bit 0 **RIS8:** COGx 上升沿事件输入源 8 使能位
1 = 使能 PWM4 输出作为上升沿事件输入
0 = PWM4 输出对上升沿事件没有任何作用

寄存器 27-5: COGxRSIM0: COG 上升沿事件源输入模式寄存器

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| RSIM7 | RSIM6 | RSIM5 | RSIM4 | RSIM3 | RSIM2 | RSIM1 | RSIM0 |
| 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	RSIM7: COGx 上升沿事件输入源 7 模式位 <u>RIS7 = 1:</u> 1 = PWM3 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = PWM3 输出高电平时, 将立即产生上升沿事件 <u>RIS7 = 0:</u> PWM3 输出对上升沿事件没有任何作用
bit 6	RSIM6: COGx 上升沿事件输入源 6 模式位 <u>RIS6 = 1:</u> 1 = CCP2 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = CCP2 输出高电平时, 将立即产生上升沿事件 <u>RIS6 = 0:</u> CCP2 输出对上升沿事件没有任何作用
bit 5	RSIM5: COGx 上升沿事件输入源 5 模式位 <u>RIS5 = 1:</u> 1 = CCP1 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = CCP1 输出高电平时, 将立即产生上升沿事件 <u>RIS5 = 0:</u> CCP1 输出对上升沿事件没有任何作用
bit 4	RSIM4: COGx 上升沿事件输入源 4 模式位 <u>RIS4 = 1:</u> 1 = 比较器 4 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = 比较器 4 输出高电平时, 将立即产生上升沿事件 <u>RIS4 = 0:</u> 比较器 4 对上升沿事件没有任何作用
bit 3	RSIM3: COGx 上升沿事件输入源 3 模式位 <u>RIS3 = 1:</u> 1 = 比较器 3 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = 比较器 3 输出高电平时, 将立即产生上升沿事件 <u>RIS3 = 0:</u> 比较器 3 输出对上升沿事件没有任何作用
bit 2	RSIM2: COGx 上升沿事件输入源 2 模式位 <u>RIS2 = 1:</u> 1 = 比较器 2 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = 比较器 2 输出高电平时, 将立即产生上升沿事件 <u>RIS2 = 0:</u> 比较器 2 对上升沿事件没有任何作用
bit 1	RSIM1: COGx 上升沿事件输入源 1 模式位 <u>RIS1 = 1:</u> 1 = 比较器 1 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = 比较器 1 输出高电平时, 将立即产生上升沿事件 <u>RIS1 = 0:</u> 比较器 1 对上升沿事件没有任何作用
bit 0	RSIM0: COGx 上升沿事件输入源 0 模式位 <u>RIS0 = 1:</u> 1 = 通过 COGxINS 寄存器选择的引脚从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件 0 = 通过 COGxINPPS 寄存器选择的引脚为高电平时, 将立即产生上升沿事件 <u>RIS0 = 0:</u> 通过 COGxINPPS 寄存器选择的引脚对上升沿事件没有任何作用

PIC16(L)F1764/5/8/9

寄存器 27-6: COGxRSIM1: COG 上升沿事件源输入模式寄存器

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
—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8
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 RSIM13: COGx 上升沿事件输入源 13 模式位

RIS13 = 1:

1 = CLC3 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件

0 = CLC3 输出高电平时, 将立即产生上升沿事件

RIS13 = 0:

CLC3 输出对上升沿事件没有任何作用

bit 4 RSIM12: COGx 上升沿事件输入源 12 模式位

RIS12 = 1:

1 = CLC2 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件

0 = CLC2 输出高电平时, 将立即产生上升沿事件

RIS12 = 0:

CLC2 输出对上升沿事件没有任何作用

bit 3 RSIM11: COGx 上升沿事件输入源 11 模式位

RIS11 = 1:

1 = CLC1 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件

0 = CLC1 输出高电平时, 将立即产生上升沿事件

RIS11 = 0:

CLC1 输出对上升沿事件没有任何作用

bit 2 RSIM10: COGx 上升沿事件输入源 10 模式位

RIS10 = 1:

1 = PWM6 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件

0 = PWM6 输出高电平时, 将立即产生上升沿事件

RIS10 = 0:

PWM6 输出对上升沿事件没有任何作用

bit 1 RSIM9: COGx 上升沿事件输入源 9 模式位

RIS9 = 1:

1 = PWM5 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件

0 = PWM5 输出高电平时, 将立即产生上升沿事件

RIS9 = 0:

PWM5 输出对上升沿事件没有任何作用

bit 0 RSIM8: COGx 上升沿事件输入源 8 模式位

RIS8 = 1:

1 = PWM4 输出从低电平变为高电平时, 将在上升沿事件相位延时之后产生上升沿事件

0 = PWM4 输出高电平时, 将立即产生上升沿事件

RIS8 = 0:

PWM4 输出对上升沿事件没有任何作用

寄存器 27-7: COGxFIS0: COG 下降沿事件输入选择寄存器

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| FIS7 | FIS6 | FIS5 | FIS4 | FIS3 | FIS2 | FIS1 | FIS0 |
| 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 | FIS7: COGx 下降沿事件输入源 7 使能位
1 = 使能 PWM3 输出作为下降沿事件输入
0 = PWM3 输出对下降沿事件没有任何作用 |
| bit 6 | FIS6: COGx 下降沿事件输入源 6 使能位
1 = 使能 CCP2 输出作为下降沿事件输入
0 = CCP2 输出对下降沿事件没有任何作用 |
| bit 5 | FIS5: COGx 下降沿事件输入源 5 使能位
1 = 使能 CCP1 输出作为下降沿事件输入
0 = CCP1 输出对下降沿事件没有任何作用 |
| bit 4 | FIS4: COGx 下降沿事件输入源 4 使能位
1 = 使能比较器 4 输出作为下降沿事件输入
0 = 比较器 4 输出对下降沿事件没有任何作用 |
| bit 3 | FIS3: COGx 下降沿事件输入源 3 使能位
1 = 使能比较器 3 输出作为下降沿事件输入
0 = 比较器 3 输出对下降沿事件没有任何作用 |
| bit 2 | FIS2: COGx 下降沿事件输入源 2 使能位
1 = 使能比较器 2 输出作为下降沿事件输入
0 = 比较器 2 输出对下降沿事件没有任何作用 |
| bit 1 | FIS1: COGx 下降沿事件输入源 1 使能位
1 = 使能比较器 1 输出作为下降沿事件输入
0 = 比较器 1 输出对下降沿事件没有任何作用 |
| bit 0 | FIS0: COGx 下降沿事件输入源 0 使能位
1 = 使能通过 COGxINPPS 控制寄存器选择的引脚作为下降沿事件输入
0 = 通过 COGxINPPS 控制寄存器选择的引脚对下降沿事件没有任何作用 |

PIC16(L)F1764/5/8/9

寄存器 27-8: COGxFIS1: COG 下降沿事件输入选择寄存器

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
—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8
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 **FIS13:** COGx 下降沿事件输入源 13 使能位
1 = 使能 CLC3 输出作为下降沿事件输入
0 = CLC3 输出对下降沿事件没有任何作用
- bit 4 **FIS12:** COGx 下降沿事件输入源 12 使能位
1 = 使能 CLC2 输出作为下降沿事件输入
0 = CLC2 输出对下降沿事件没有任何作用
- bit 3 **FIS11:** COGx 下降沿事件输入源 11 使能位
1 = 使能 CLC1 输出作为下降沿事件输入
0 = CLC1 输出对下降沿事件没有任何作用
- bit 2 **FIS10:** COGx 下降沿事件输入源 10 使能位
1 = 使能 PWM6 输出作为下降沿事件输入
0 = PWM6 输出对下降沿事件没有任何作用
- bit 1 **FIS9:** COGx 下降沿事件输入源 9 使能位
1 = 使能 PWM5 输出作为下降沿事件输入
0 = PWM5 输出对下降沿事件没有任何作用
- bit 0 **FIS8:** COGx 下降沿事件输入源 8 使能位
1 = 使能 PWM4 输出作为下降沿事件输入
0 = PWM4 输出对下降沿事件没有任何作用

寄存器 27-9: COGxFSIM0: COG 下降沿事件源输入模式寄存器

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| FSIM7 | FSIM6 | FSIM5 | FSIM4 | FSIM3 | FSIM2 | FSIM1 | FSIM0 |
| 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	FSIM7: COGx 下降沿事件输入源 7 模式位 <u>FIS7 = 1:</u> 1 = PWM3 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = PWM3 输出低电平时, 将立即产生下降沿事件 <u>FIS7 = 0:</u> PWM3 输出对下降沿事件没有任何作用
bit 6	FSIM6: COGx 下降沿事件输入源 6 模式位 <u>FIS6 = 1:</u> 1 = CCP2 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = CCP2 输出低电平时, 将立即产生下降沿事件 <u>FIS6 = 0:</u> CCP2 输出对下降沿事件没有任何作用
bit 5	FSIM5: COGx 下降沿事件输入源 5 模式位 <u>FIS5 = 1:</u> 1 = CCP1 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = CCP1 输出低电平时, 将立即产生下降沿事件 <u>FIS5 = 0:</u> CCP1 输出对下降沿事件没有任何作用
bit 4	FSIM4: COGx 下降沿事件输入源 4 模式位 <u>FIS4 = 1:</u> 1 = 比较器 4 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = 比较器 4 输出低电平时, 将立即产生下降沿事件 <u>FIS4 = 0:</u> 比较器 4 对下降沿事件没有任何作用
bit 3	FSIM3: COGx 下降沿事件输入源 3 模式位 <u>FIS3 = 1:</u> 1 = 比较器 3 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = 比较器 3 输出低电平时, 将立即产生下降沿事件 <u>FIS3 = 0:</u> 比较器 3 对下降沿事件没有任何作用
bit 2	FSIM2: COGx 下降沿事件输入源 2 模式位 <u>FIS2 = 1:</u> 1 = 比较器 2 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = 比较器 2 输出低电平时, 将立即产生下降沿事件 <u>FIS2 = 0:</u> 比较器 2 对下降沿事件没有任何作用
bit 1	FSIM1: COGx 下降沿事件输入源 1 模式位 <u>FIS1 = 1:</u> 1 = 比较器 1 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = 比较器 1 输出低电平时, 将立即产生下降沿事件 <u>FIS1 = 0:</u> 比较器 1 对下降沿事件没有任何作用
bit 0	FSIM0: COGx 下降沿事件输入源 0 模式位 <u>FIS0 = 1:</u> 1 = 通过 COGxINPPS 控制选择的引脚从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件 0 = 通过 COGxINPPS 控制选择的引脚低电平时, 将立即产生下降沿事件 <u>FIS0 = 0:</u> 通过 COGxINPPS 控制选择的引脚对下降沿事件没有任何作用

PIC16(L)F1764/5/8/9

寄存器 27-10: COGxFSIM1: COG 下降沿事件源输入模式寄存器

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
—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8
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

FSIM13: COGx 下降沿事件输入源 13 模式位

FIS13 = 1:

1 = CLC3 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件

0 = CLC3 输出低电平时, 将立即产生下降沿事件

FIS13 = 0:

CLC3 输出对下降沿事件没有任何作用

bit 4

FSIM12: COGx 下降沿事件输入源 12 模式位

FIS12 = 1:

1 = CLC2 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件

0 = CLC2 输出低电平时, 将立即产生下降沿事件

FIS12 = 0:

CLC2 输出对下降沿事件没有任何作用

bit 3

FSIM11: COGx 下降沿事件输入源 11 模式位

FIS11 = 1:

1 = CLC1 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件

0 = CLC1 输出低电平时, 将立即产生下降沿事件

FIS11 = 0:

CLC1 输出对下降沿事件没有任何作用

bit 2

FSIM10: COGx 下降沿事件输入源 10 模式位

FIS10 = 1:

1 = PWM6 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件

0 = PWM6 输出低电平时, 将立即产生下降沿事件

FIS10 = 0:

PWM6 输出对下降沿事件没有任何作用

bit 1

FSIM9: COGx 下降沿事件输入源 9 模式位

FIS9 = 1:

1 = PWM5 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件

0 = PWM5 输出低电平时, 将立即产生下降沿事件

FIS9 = 0:

PWM5 输出对下降沿事件没有任何作用

bit 0

FSIM8: COGx 下降沿事件输入源 8 模式位

FIS8 = 1:

1 = PWM4 输出从高电平变为低电平时, 将在下降沿事件相位延时之后产生下降沿事件

0 = PWM4 输出低电平时, 将立即产生下降沿事件

FIS8 = 0:

PWM4 输出对下降沿事件没有任何作用

寄存器 27-11: COGxASD0: COG 自动关断控制寄存器 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	U-0	U-0
ASE	ARSEN	ASDBD<1:0>		ASDAC<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 **ASE:** 自动关断事件状态位
1 = COG 处于关断状态
0 = COG 不处于关断状态或将在发生下一个上升沿事件时退出关断状态
- bit 6 **ARSEN:** 自动重启使能位
1 = 使能自动重启
0 = 禁止自动重启
- bit 5-4 **ASDBD<1:0>:** COGxB 和 COGxD 自动关断改写电平选择位
11 = 在关断有效时, 将逻辑 1 放置在 COGxB 和 COGxD 上
10 = 在关断有效时, 将逻辑 0 放置在 COGxB 和 COGxD 上
01 = 在关断有效时, COGxB 和 COGxD 处于三态
00 = 在关断有效时, 将引脚的无效状态 (包括极性) 放置在 COGxB 和 COGxD 上
- bit 3-2 **ASDAC<1:0>:** COGxA 和 COGxC 自动关断改写电平选择位
11 = 在关断有效时, 将逻辑 1 放置在 COGxA 和 COGxC 上
10 = 在关断有效时, 将逻辑 0 放置在 COGxA 和 COGxC 上
01 = 在关断有效时, COGxA 和 COGxC 处于三态
00 = 在关断有效时, 将引脚的无效状态 (包括极性) 放置在 COGxA 和 COGxC 上
- bit 1-0 未实现: 读为 0

PIC16(L)F1764/5/8/9

寄存器 27-12: COGxASD1: COG 自动关断控制寄存器 1

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| AS7E | AS6E | AS5E | 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 **AS7E:** COGx 自动关断源使能位 7

1 = 在 Timer4_output 为高电平时关断 COGx

0 = Timer4_output 对关断没有任何作用

bit 6 **AS6E:** COGx 自动关断源使能位 6

1 = 在 Timer2_output 为高电平时关断 COGx

0 = Timer2_output 对关断没有任何作用

bit 5 **AS5E:** COGx 自动关断源使能位 5

1 = 在 CLC LC2_out 为低电平时关断 COGx

0 = CLC2 输出对关断没有任何作用

bit 4 **AS4E:** COGx 自动关断源使能位 4

1 = 在比较器 sync_C4OUT 为低电平时关断 COGx

0 = 比较器 4 的输出对关断没有任何作用

bit 3 **AS3E:** COGx 自动关断源使能位 3

1 = 在比较器 sync_C3OUT 为低电平时关断 COGx

0 = 比较器 3 的输出对关断没有任何作用

bit 2 **AS2E:** COGx 自动关断源使能位 2

1 = 在比较器 sync_C2OUT 为低电平时关断 COGx

0 = 比较器 2 的输出对关断没有任何作用

bit 1 **AS1E:** COGx 自动关断源使能位 1

1 = 在比较器 sync_C1OUT 为低电平时关断 COGx

0 = 比较器 1 的输出对关断没有任何作用

bit 0 **AS0E:** COGx 自动关断源使能位 0

1 = 在通过 COGxINPPS 寄存器选择的引脚为低电平时关断 COGx

0 = 通过 COGxINPPS 寄存器选择的引脚对关断没有任何作用

寄存器 27-13: COGxSTR: COG 转向控制寄存器 1⁽¹⁾

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| SDATD | SDATC | SDATB | SDATA | STRD | STRC | STRB | STRA |
| 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 | SDATD: COGxD 静态输出数据位
1 = COGxD 静态数据为高电平
0 = COGxD 静态数据为低电平 |
| bit 6 | SDATC: COGxC 静态输出数据位
1 = COGxC 静态数据为高电平
0 = COGxC 静态数据为低电平 |
| bit 5 | SDATB: COGxB 静态输出数据位
1 = COGxB 静态数据为高电平
0 = COGxB 静态数据为低电平 |
| bit 4 | SDATA: COGxA 静态输出数据位
1 = COGxA 静态数据为高电平
0 = COGxA 静态数据为低电平 |
| bit 3 | STRD: COGxD 转向控制位
1 = COGxD 输出具有通过 POLD 位进行极性控制的 COGxD 波形
0 = COGxD 输出为由 SDATD 位决定的静态数据电平 |
| bit 2 | STRC: COGxC 转向控制位
1 = COGxC 输出具有通过 POLC 位进行极性控制的 COGxC 波形
0 = COGxC 输出为由 SDATC 位决定的静态数据电平 |
| bit 1 | STRB: COGxB 转向控制位
1 = COGxB 输出具有通过 POLB 位进行极性控制的 COGxB 波形
0 = COGxB 输出为由 SDATB 位决定的静态数据电平 |
| bit 0 | STRA: COGxA 转向控制位
1 = COGxA 输出具有通过 POLA 位进行极性控制的 COGxA 波形
0 = COGxA 输出为由 SDATA 位决定的静态数据电平 |

注 1: 转向仅在 COGxCON0 寄存器的 MD<1:0> 位等于 00x 时有效。(见 [寄存器 27-1](#))

PIC16(L)F1764/5/8/9

寄存器 27-14: COGxDBR: COG 上升沿事件死区计数寄存器

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>: 上升沿事件死区计数值位

RDBS = 0:

= 在上升沿事件之后将主输出延迟的 COGx 时钟周期数

RDBS = 1:

= 在上升沿事件之后将主输出延迟的延时链元件周期数

寄存器 27-15: COGxDBF: COG 下降沿事件死区计数寄存器

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>: 下降沿事件死区计数值位

FDBS = 0:

= 在下降沿事件输入之后将互补输出延迟的 COGx 时钟周期数

FDBS = 1:

= 在下降沿事件输入之后将互补输出延迟的延时链元件周期数

寄存器 27-16: COGxBLKR: COG 上升沿事件消隐计数寄存器

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
BLKR<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 BLKR<5:0>: 上升沿事件消隐计数值位

= 禁止下降沿事件输入的 COGx 时钟周期数

寄存器 27-17: COGxBLKF: COG 下降沿事件消隐计数寄存器

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
BLKF<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

BLKF<5:0>: 下降沿事件消隐计数值位

= 禁止上升沿事件输入的 COGx 时钟周期数

寄存器 27-18: COGxPHR: COG 上升沿事件相位延时计数寄存器

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
—	—	PHR<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

PHR<5:0>: 上升沿事件相位延时计数值位

= 将上升沿事件延迟的 COGx 时钟周期数

寄存器 27-19: COGxPHF: COG 下降沿事件相位延时计数寄存器

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
—	—	PHF<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

PHF<5:0>: 下降沿事件相位延时计数值位

= 将下降沿事件延迟的 COGx 时钟周期数

PIC16(L)F1764/5/8/9

表 27-4: 与 COGx 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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB6	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSEL C	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC0	ANSC0	151
COGxASD0	ASE	ARSEN	ASDBD<1:0>		ASDAC<1:0>		—	—	325
COGxASD1	AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	326
COGxBLKR	—	—	BLKR<5:0>						328
COGxBLKF	—	—	BLKF<5:0>						329
COGxCON0	EN	LD	—	CS<1:0>		MD<2:0>			315
COGxCON1	RDBS	FDBS	—	—	POLD	POLC	POLB	POLA	316
COGxDBR	—	—	DBR<5:0>						328
COGxDBF	—	—	DBF<5:0>						328
COGxFIS0	FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0	321
COGxFIS1	—	—	FIS13	FIS12	FIS11	FIS10	FIS9	FIS8	322
COGxFSIM0	FSIM7	FSIM6	FSIM5	FSIM4	FSIM3	FSIM2	FSIM1	FSIM0	323
COGxFSIM1	—	—	FSIM13	FSIM12	FSIM11	FSIM10	FSIM9	FSIM8	324
COGxPHR	—	—	PHR<5:0>						329
COGxPHF	—	—	PHF<5:0>						329
COGxPPS	—	—	—	COG1PPS<4:0>					158, 160
COGxRIS0	RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0	317
COGxRIS1	—	—	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8	318
COGxRSIM0	RSIM7	RSIM6	RSIM5	RSIM4	RSIM3	RSIM2	RSIM1	RSIM0	319
COGxRSIM1	—	—	RSIM13	RSIM12	RSIM11	RSIM10	RSIM9	RSIM8	320
COGxSTR	SDATD	SDATC	SDATB	SDATA	STRD	STRC	STRB	STRA	327
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
RxyPPS	—	—	—	RxyPPS<4:0>					158

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

注 1: 仅限 PIC16(L)F1768/9。

28.0 可配置逻辑单元 (CLCx)

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 32 个输入信号，并通过使用可配置门将 32 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

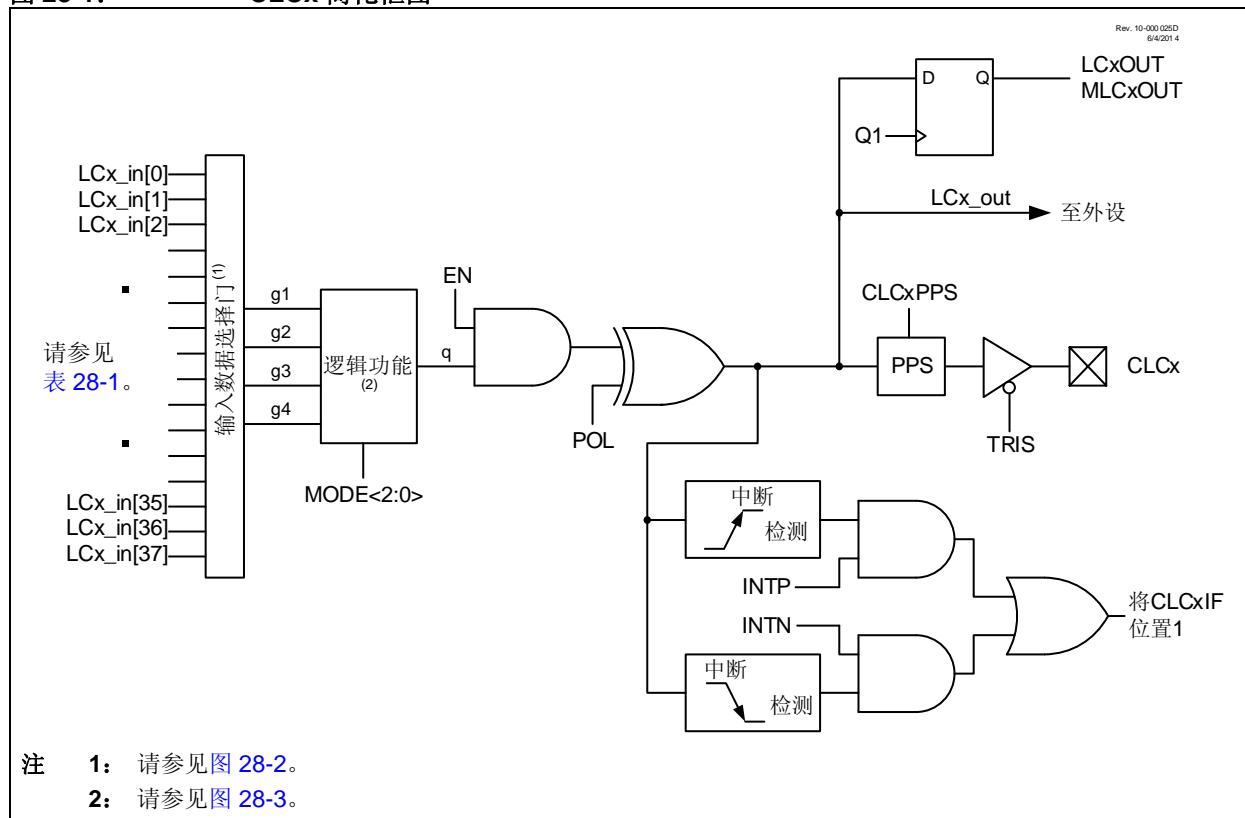
可将输出内部连接到外设和输出引脚。

关于说明通过CLCx的信号流的简化图，请参见[图28-1](#)。

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 28-1: CLCx 简化框图



28.1 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。

28.1.1 数据选择

有 32 个信号可用作可配置逻辑的输入。使用 4 个 32 输入多路开关来选择要传递到下一级的输入。

数据选择通过图 28-2 左侧所示的 4 个多路开关来进行。图中的数据输入使用通用编号的输入名称来表示。

表 28-1 列出了每个 CLC 模块中的通用编号的输入名称和实际信号的关联。标有 dy 的列给出了选定数据输入的多路开关选择代码。DxS 是多路开关选择输入代码 D1S<4:0> 至 D4S<4:0> 的缩写。.

数据输入使用 CLCxSEL0 至 CLCxSEL3 寄存器（[寄存器 28-3 至寄存器 28-6](#)）进行选择。

注： 数据选择在上电时是未定义的。

表 28-1： CLCx 数据输入选择

数据输入	dy DxS	CLCx
LCx_in[37]	100101	FOSC
LCx_in[36]	100100	HFINTOSC
LCx_in[35]	100011	LFINTOSC
LCx_in[34]	100010	FRC (ADC RC 时钟)
LCx_in[33]	100001	IOCIF 置 1
LCx_in[32]	100000	Timer6_postscaled
LCx_in[31]	011111	Timer4_postscaled
LCx_in[30]	011110	Timer2_postscaled
LCx_in[29]	011101	Timer5 溢出
LCx_in[28]	011100	Timer3 溢出
LCx_in[27]	011011	Timer1 溢出
LCx_in[26]	011010	Timer0 溢出
LCx_in[25]	011001	EUSART RX
LCx_in[24]	011000	EUSART TX
LCx_in[23]	010111	ZCD1_output
LCx_in[22]	010110	MSSP1 SDO/SDA
LCx_in[21]	010101	MSSP1 SCL/SCK
LCx_in[20]	010100	PWM6_out
LCx_in[19]	010011	PWM5_out
LCx_in[18]	010010	PWM4_out
LCx_in[17]	010001	PWM3_out
LCx_in[16]	010000	CCP2_out
LCx_in[15]	001111	CCP1_out
LCx_in[14]	001110	COG2B
LCx_in[13]	001101	COG2A
LCx_in[12]	001100	COG1B
LCx_in[11]	001011	COG1A
LCx_in[10]	001010	sync_C4OUT
LCx_in[9]	001001	sync_C3OUT
LCx_in[8]	001000	sync_C2OUT
LCx_in[7]	000111	sync_C1OUT
LCx_in[6]	000110	来自 CLC3 的 LC3_out
LCx_in[5]	000101	来自 CLC2 的 LC2_out
LCx_in[4]	000100	来自 CLC1 的 LC1_out
LCx_in[3]	000011	在 CLCIN3PPS 寄存器中选择的 CLCIN3 引脚输入
LCx_in[2]	000010	在 CLCIN2PPS 寄存器中选择的 CLCIN2 引脚输入
LCx_in[1]	000001	在 CLCIN1PPS 寄存器中选择的 CLCIN1 引脚输入
LCx_in[0]	000000	在 CLCINOOPPS 寄存器中选择的 CLCINO 引脚输入

28.1.2 数据门

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。

注： 在上电时，数据门是未定义的。

门级不仅仅是信号方向。可将门配置为将每个输入信号指定为反相或同相数据。在每个门中，将定向后的信号进行与运算。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 个输入的 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运算。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运算。

表 28-2 总结了可以通过使用门逻辑选择位在门 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 28-2： 数据门逻辑

CLCxGLS0	G1POL	门逻辑
0x55	1	AND
0x55	0	NAND
0xAA	1	NOR
0xAA	0	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。

数据门使用如下逻辑门选择寄存器进行配置：

- 门 1: CLCxGLS0 (寄存器 28-7)
- 门 2: CLCxGLS1 (寄存器 28-8)
- 门 3: CLCxGLS2 (寄存器 28-9)
- 门 4: CLCxGLS3 (寄存器 28-10)

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 28-2 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。

28.1.3 逻辑功能

有 8 种可用的逻辑功能，包括：

- AND-OR
- OR-XOR
- AND
- S-R 锁存器
- 带置 1 和复位功能的 D 触发器
- 带复位功能的 D 触发器
- 带复位功能的 J-K 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 28-3 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、输出引脚，然后回到 CLCx。

28.1.4 输出极性

可配置逻辑单元中的最后一个级是输出极性。将 CLCxCON 寄存器的 POL 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发生中断。

28.1.5 CLCx 设置步骤

在设置 CLCx 时，应遵循以下步骤：

- 通过清零 EN 位来禁止 CLCx。
- 使用 CLCxSEL0 至 CLCxSEL3 寄存器选择所需的输入（见表 28-1）。
- 清零所有相关的 ANSEL 位。
- 将与输入相关的所有 TRIS 位置 1。
- 将与输出相关的所有 TRIS 位清零。
- 使用 CLCxGLS0、CLCxGLS1、CLCxGLS2 和 CLCxGLS3 寄存器通过 4 个门来使能所选输入。
- 使用 CLCxPOL 寄存器的 POL 位选择门输出极性。
- 使用 CLCxCON 寄存器的 MODE<2:0> 位选择所需的逻辑功能。
- 使用 CLCxPOL 寄存器的 POL 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）。
- 如果要驱动某个器件引脚，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 INTP 位置 1。
 - 下降沿事件时，将 CLCxCON 寄存器中的 INTN 位置 1。
 - 将相关的 PIE 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 EN 位置 1 来使能 CLCx。

28.2 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，且其相关的中断位置 1 时，相关 PIR 寄存器的 CLCxIF 位会置 1。INTP 位用于允许上升沿中断，INTN 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：

- CLCxCON 寄存器的 ON 位
- 关联 PIE 寄存器的 CLCxIE 位
- CLCxCON 寄存器的 INTP 位（对于上升沿检测）
- CLCxCON 寄存器的 INTN 位（对于下降沿检测）
- INTCON 寄存器的 PEIE 和 GIE 位

作为中断服务的一部分，必须用软件将相关 PIR 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。

28.3 输出镜像副本

所有 LCxCON 输出位的镜像副本包含在 CLCxDATA 寄存器中。读取该寄存器将同时读取所有 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 位而导致读取差错。

28.4 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。

28.5 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。

图 28-2: 输入数据选择和门

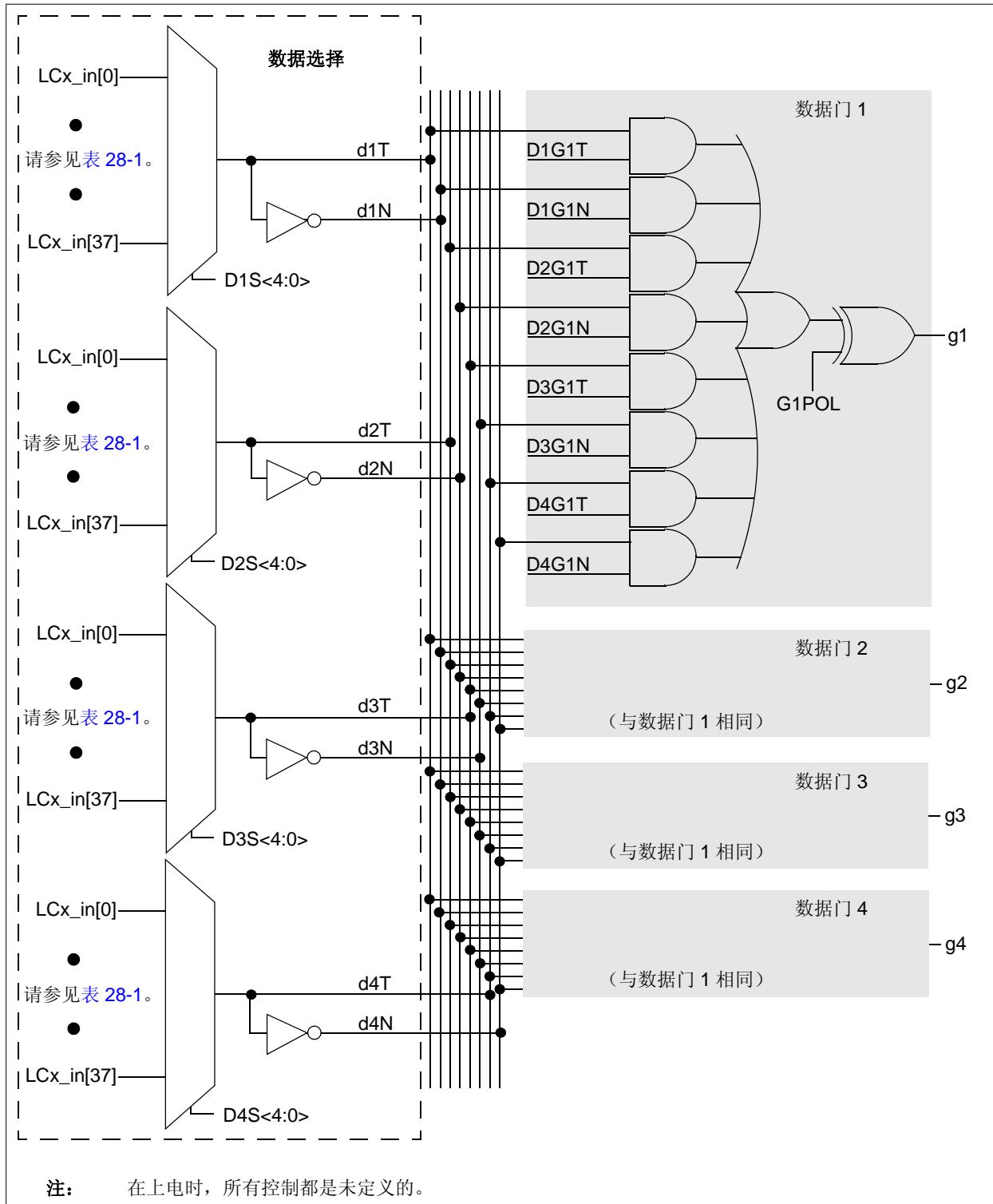
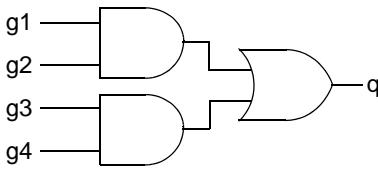
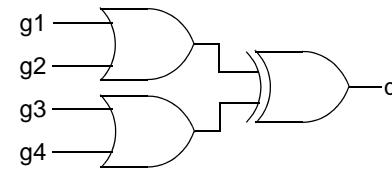
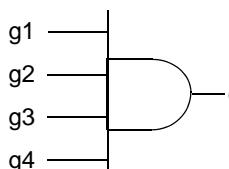
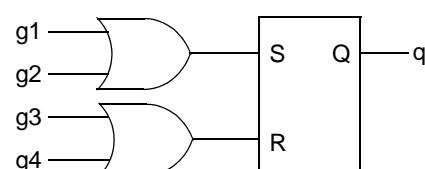
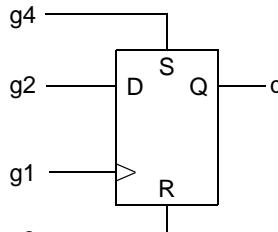
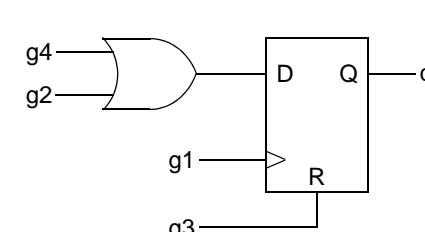
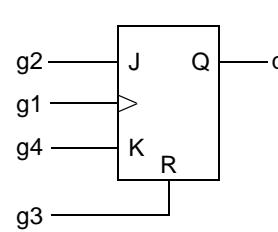
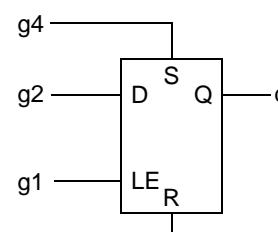


图 28-3: 可编程逻辑功能

AND – OR  MODE<2:0> = 000	OR – XOR  MODE<2:0> = 001
4 输入 AND  MODE<2:0> = 010	S-R 锁存器  MODE<2:0> = 011
带置 1 和复位功能的 1 输入 D 触发器  MODE<2:0> = 100	带复位功能的 2 输入 D 触发器  MODE<2:0> = 101
带复位功能的 J-K 触发器  MODE<2:0> = 110	带置 1 和复位功能的 1 输入透明锁存器  MODE<2:0> = 111

28.6 寄存器定义：CLC 控制

表 28-3 列出了 CLC 外设的长位名称前缀。更多信息，请参见第 1.1 节“寄存器和位命名约定”。

表 28-3：

外设	位名称前缀
CLC1	LC1
CLC2	LC2

寄存器 28-1： 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
EN	—	OUT	INTP	INTN		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

bit 7

EN: 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

OUT: 可配置逻辑单元数据输出位

只读：经过 POL 之后的逻辑单元输出数据；从 lcx_out 线采样。

bit 4

INTP: 可配置逻辑单元上升边沿中断允许位

1 = CLCxIF 将在 lcx_out 上出现上升沿时置 1
0 = CLCxIF 不会置 1

bit 3

INTN: 可配置逻辑单元下降边沿中断允许位

1 = CLCxIF 将在 lcx_out 上出现下降沿时置 1
0 = CLCxIF 不会置 1

bit 2-0

MODE<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 触发器
101 = 单元是带复位功能的 2 输入 D 触发器
100 = 单元是带置 1 和复位功能的 1 输入 D 触发器
011 = 单元是 S-R 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)F1764/5/8/9

寄存器 28-2: 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
POL	—	—	—	G4POL	G3POL	G2POL	G1POL
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 **POL:** LCOUT 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 **G4POL:** 门 4 输出极性控制位

1 = 门 4 的输出在施加到逻辑单元时反相

0 = 门 4 的输出不反相

bit 2 **G3POL:** 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 1 **G2POL:** 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 0 **G1POL:** 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

寄存器 28-3: CLCxSEL0: 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
—	—			D1S<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 D1S<5:0>: CLCx 数据 1 输入选择位
请参见表 28-1。

寄存器 28-4: CLCxSEL1: 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
—	—			D2S<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 D2S<5:0>: CLCx 数据 2 输入选择位
请参见表 28-1。

寄存器 28-5: CLCxSEL2: 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
—	—			D3S<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 D3S<5:0>: CLCx 数据 3 输入选择位
请参见表 28-1。

PIC16(L)F1764/5/8/9

寄存器 28-6: CLCxSEL3: 通用 CLCx 数据 4 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
—	—			D4S<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 D4S<5:0>: CLCx 数据 4 输入选择位
请参见表 28-1。

寄存器 28-7: CLCxGLS0: 门 1 逻辑选择寄存器

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| G1D4T | G1D4N | G1D3T | G1D3N | G1D2T | G1D2N | G1D1T | G1D1N |
| 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 G1D4T: 门 1 数据 4 (同相) 位

1 = d4T 通过门输入到 g1

0 = d4T 不通过门输入到 g1

bit 6 G1D4N: 门 1 数据 4 取反 (反相) 位

1 = d4N 通过门输入到 g1

0 = d4N 不通过门输入到 g1

bit 5 G1D3T: 门 1 数据 3 (同相) 位

1 = d3T 通过门输入到 g1

0 = d3T 不通过门输入到 g1

bit 4 G1D3N: 门 1 数据 3 取反 (反相) 位

1 = d3N 通过门输入到 g1

0 = d3N 不通过门输入到 g1

bit 3 G1D2T: 门 1 数据 2 (同相) 位

1 = d2T 通过门输入到 g1

0 = d2T 不通过门输入到 g1

bit 2 G1D2N: 门 1 数据 2 取反 (反相) 位

1 = d2N 通过门输入到 g1

0 = d2N 不通过门输入到 g1

bit 1 G1D1T: 门 1 数据 1 (同相) 位

1 = d1T 通过门输入到 g1

0 = d1T 不通过门输入到 g1

bit 0 G1D1N: 门 1 数据 1 取反 (反相) 位

1 = d1N 通过门输入到 g1

0 = d1N 不通过门输入到 g1

寄存器 28-8: CLCxGLS1: 门 2 逻辑选择寄存器

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| G2D4T | G2D4N | G2D3T | G2D3N | G2D2T | G2D2N | G2D1T | G2D1N |
| 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 **G2D4T:** 门 2 数据 4 (同相) 位
1 = d4T 通过门输入到 g2
0 = d4T 不通过门输入到 g2
- bit 6 **G2D4N:** 门 2 数据 4 取反 (反相) 位
1 = d4N 通过门输入到 g2
0 = d4N 不通过门输入到 g2
- bit 5 **G2D3T:** 门 2 数据 3 (同相) 位
1 = d3T 通过门输入到 g2
0 = d3T 不通过门输入到 g2
- bit 4 **G2D3N:** 门 2 数据 3 取反 (反相) 位
1 = d3N 通过门输入到 g2
0 = d3N 不通过门输入到 g2
- bit 3 **G2D2T:** 门 2 数据 2 (同相) 位
1 = d2T 通过门输入到 g2
0 = d2T 不通过门输入到 g2
- bit 2 **G2D2N:** 门 2 数据 2 取反 (反相) 位
1 = d2N 通过门输入到 g2
0 = d2N 不通过门输入到 g2
- bit 1 **G2D1T:** 门 2 数据 1 (同相) 位
1 = d1T 通过门输入到 g2
0 = d1T 不通过门输入到 g2
- bit 0 **G2D1N:** 门 2 数据 1 取反 (反相) 位
1 = d1N 通过门输入到 g2
0 = d1N 不通过门输入到 g2

PIC16(L)F1764/5/8/9

寄存器 28-9: CLCxGLS2: 门 3 逻辑选择寄存器

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| G3D4T | G3D4N | G3D3T | G3D3N | G3D2T | G3D2N | G3D1T | G3D1N |
| 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 **G3D4T:** 门 3 数据 4 (同相) 位

1 = d4T 通过门输入到 g3

0 = d4T 不通过门输入到 g3

bit 6 **G3D4N:** 门 3 数据 4 取反 (反相) 位

1 = d4N 通过门输入到 g3

0 = d4N 不通过门输入到 g3

bit 5 **G3D3T:** 门 3 数据 3 (同相) 位

1 = d3T 通过门输入到 g3

0 = d3T 不通过门输入到 g3

bit 4 **G3D3N:** 门 3 数据 3 取反 (反相) 位

1 = d3N 通过门输入到 g3

0 = d3N 不通过门输入到 g3

bit 3 **G3D2T:** 门 3 数据 2 (同相) 位

1 = d2T 通过门输入到 g3

0 = d2T 不通过门输入到 g3

bit 2 **G3D2N:** 门 3 数据 2 取反 (反相) 位

1 = d2N 通过门输入到 g3

0 = d2N 不通过门输入到 g3

bit 1 **G3D1T:** 门 3 数据 1 (同相) 位

1 = d1T 通过门输入到 g3

0 = d1T 不通过门输入到 g3

bit 0 **G3D1N:** 门 3 数据 1 取反 (反相) 位

1 = d1N 通过门输入到 g3

0 = d1N 不通过门输入到 g3

寄存器 28-10: CLCxGLS3: 门 4 逻辑选择寄存器

| R/W-x/u |
|---------|---------|---------|---------|---------|---------|---------|---------|
| G4D4T | G4D4N | G4D3T | G4D3N | G4D2T | G4D2N | G4D1T | G4D1N |
| 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 **G4D4T:** 门 4 数据 4 (同相) 位
1 = d4T 通过门输入到 g4
0 = d4T 不通过门输入到 g4
- bit 6 **G4D4N:** 门 4 数据 4 取反 (反相) 位
1 = d4N 通过门输入到 g4
0 = d4N 不通过门输入到 g4
- bit 5 **G4D3T:** 门 4 数据 3 (同相) 位
1 = d3T 通过门输入到 g4
0 = d3T 不通过门输入到 g4
- bit 4 **G4D3N:** 门 4 数据 3 取反 (反相) 位
1 = d3N 通过门输入到 g4
0 = d3N 不通过门输入到 g4
- bit 3 **G4D2T:** 门 4 数据 2 (同相) 位
1 = d2T 通过门输入到 g4
0 = d2T 不通过门输入到 g4
- bit 2 **G4D2N:** 门 4 数据 2 取反 (反相) 位
1 = d2N 通过门输入到 g4
0 = d2N 不通过门输入到 g4
- bit 1 **G4D1T:** 门 4 数据 1 (同相) 位
1 = d1T 通过门输入到 g4
0 = d1T 不通过门输入到 g4
- bit 0 **G4D1N:** 门 4 数据 1 取反 (反相) 位
1 = d1N 通过门输入到 g4
0 = d1N 不通过门输入到 g4

PIC16(L)F1764/5/8/9

寄存器 28-11: CLCDATA: CLC 数据输出

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	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 7-3 未实现: 读为 0

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 28-4: 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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
CLCxCON	EN	—	OUT	INTP	INTN	MODE<2:0>			337
CLCxDATA	—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT	344
CLCxGLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	340
CLCxGLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	341
CLCxGLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	342
CLCxGLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	343
CLCxPOL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	338
CLCxSEL0	—	—			D1S<5:0>				339
CLCxSEL1	—	—			D2S<5:0>				339
CLCxSEL2	—	—			D3S<5:0>				339
CLCxSEL3	—	—			D4S<5:0>				340
CLCxPPS	—	—	—		CLCxPPS<4:0>				158, 160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE3	PWM6IE ⁽¹⁾	PWM5IE	COG1IE	ZCDIE	COG2IE ⁽¹⁾	CLC3IE	CLC2IE	CLC1IE	105
PIR3	PWM6IF ⁽¹⁾	PWM5IF	COG1IF	ZCDIF	COG2IF ⁽¹⁾	CLC3IF	CLC2IF	CLC1IF	108
RxyPPS	—	—	—		RxyPPS<4:0>				158
TRISA	—	—	TRISA5	TRISA4	— ⁽³⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150

图注: — = 未实现位, 读为 0。CLC 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F1768/9。

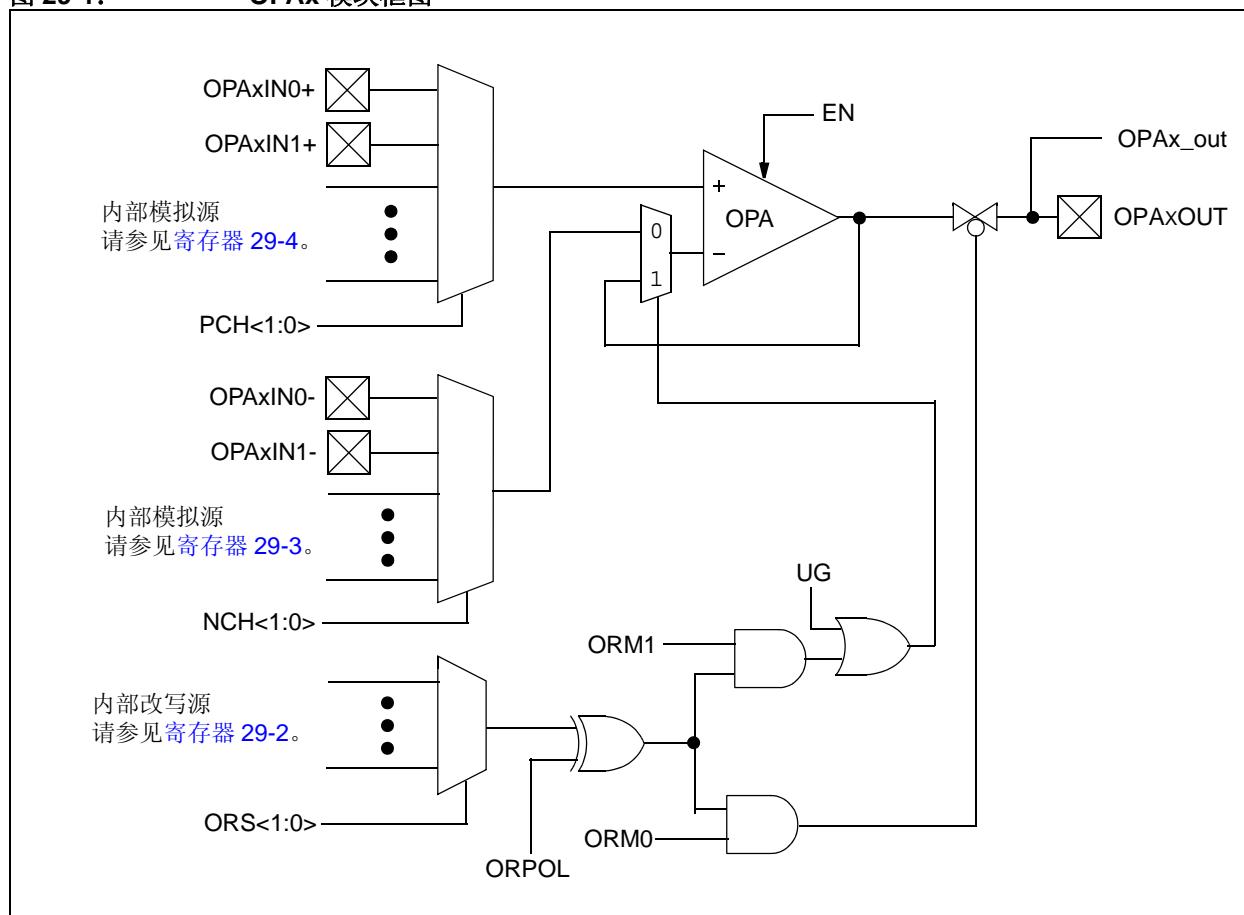
2: 未实现位, 读为 1。

29.0 运算放大器 (OPA) 模块

运算放大器 (Operational Amplifier, OPA) 是一种需要使用外部反馈进行工作的标准三端器件。OPA 模块具有以下特性：

- 外部连接到 I/O 端口
- 低泄漏电流输入
- 出厂时已校准输入失调电压
- 单位增益控制
- 可编程同相和反相源选择
- 改写控制
 - 强制三态输出
 - 强制单位增益

图 29-1: OPAx 模块框图



29.1 OPA 模块性能

OPA 模块的常见交流和直流性能规范:

- 共模电压范围
- 泄漏电流
- 输入失调电压
- 开环增益
- 增益带宽积

共模电压范围 是 OPA+ 和 OPA- 输入的规定电压范围，在此电压范围内 OPA 模块将在其规范内工作。OPA 模块设计为在 VSS 和 VDD 之间的输入电压下工作。共模电压大于 VDD 或小于 VSS 时，将无法保证模块的性能。

泄漏电流 用来度量 OPA+ 和 OPA- 输入上小的拉电流或灌电流。要最大程度降低泄漏电流的影响，连接到 OPA+ 和 OPA- 输入的有效电阻应尽可能小且相等。

输入失调电压 用来度量闭环电路中 OPA 在其线性区内时 OPA+ 和 OPA- 输入之间的电压差。在输出中将显示为直流失调的失调电压等于输入失调电压乘以电路增益。输入失调电压也受共模电压的影响。OPA 出厂时经过校准，以最大程度降低模块的输入失调电压。

开环增益 是输出电压与差分输入电压的比率，差分输入电压为 (OPA+) - (OPA-)。增益在直流时最大，并且随频率降低而下降。

增益带宽积 或 GBWP 是开环增益下降至 0 dB 时的频率。

29.2 OPA 模块控制

OPA 模块通过将 OPAxCON 寄存器（[寄存器 29-1](#)）的 OPAxEN 位置 1 来使能。当使能时，OPA 会强制 OPAxOUT 引脚的输出驱动器为三态，以防止驱动器和 OPA 输出之间发生争用。

注: 使能 OPA 模块时，OPAxOUT 引脚将由运放输出驱动，而不是由端口数字驱动器驱动。关于运放输出驱动能力的信息，请参见 [表 36-17：运算放大器（OPA）](#)。

29.2.1 单位增益模式

OPAxCON 寄存器（[寄存器 29-1](#)）的 OPAxUG 位用于选择单位增益模式。选择单位增益时，OPA 输出连接到反相输入，OPAxIN 引脚会被释放，使该引脚可用作通用输入和输出。

29.2.2 可编程的输入源选择

反相和同相源分别使用 OPAxNCHS（[寄存器 29-3](#)）和 OPAxPCHS（[寄存器 29-4](#)）寄存器进行选择。输入源包括：

- 内部 DAC
- 器件引脚
- 内部斜率补偿斜坡发生器
- 器件中的其他运放

29.3 改写控制

29.3.1 改写模式

运放操作可以通过两种方式改写：

- 强制三态输出
- 强制单位增益

改写模式使用 OPxCON 寄存器（[寄存器 29-1](#)）的 ORM<1:0> 位进行选择。改写在选择了模式且改写源为真时生效。

29.3.2 改写源

改写源使用 OPAxORS 寄存器（[寄存器 29-2](#)）进行选择。改写源来自内部外设，包括：

- CCP 输出
- PWM 输出
- 比较器输出
- 过零检测输出
- 可配置逻辑单元输出
- COG 输出

29.3.3 改写源极性

可以对改写源极性进行反相，从而在选定改写源的高电平或低电平发生改写。改写极性由 OPAxCON 寄存器（[寄存器 29-1](#)）的 ORPOL 位控制。

29.4 复位的影响

器件复位将强制所有寄存器为复位状态。这会禁止 OPA 模块。

29.5 休眠的影响

当器件被置于休眠模式时，运算放大器会继续工作。

29.6 寄存器定义：运放控制

表 29-1 列出了运放外设的长位名称前缀。更多信息，请参见第 1.1 节“[寄存器和位命名约定](#)”。

表 29-1：

外设	位名称前缀
OPA1	OPA1
OPA2 ⁽¹⁾	OPA2

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 29-1： OPAXCON：运算放大器（OPAx）控制寄存器

R/W-0/0	U-0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	—	UG	—	ORPOL	ORM<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

EN：运放使能位

1 = 使能运放

0 = 禁止运放，不消耗有功功率

bit 6-5

未实现：读为 0

bit 4

UG：运放单位增益选择位

1 = OPA 输出连接到反相输入。OPAxIN- 引脚可用作通用 I/O。

0 = 反相输入连接到 OPAxIN- 引脚

bit 3

未实现：读为 0

bit 2

ORPOL：运放改写源极性位

1 = 对改写源极性进行反相。在改写源为高电平时发生改写。

0 = 不对改写源极性进行反相。在改写源为低电平时发生改写。

bit 1-0

ORM<1:0>：运放改写模式选择位

11 = 保留。不要使用。

10 = 当改写源为真时，将运放强制为单位增益。

01 = 当改写源为真时，运放输出为三态。

00 = 禁止输出改写功能。

寄存器 29-2: OPAxORS: 运放改写源选择寄存器

U-0	U-0	U-0	R/W-0/0	R/W-0/x	R/W-0/x	R/W-0/0	R/W-0/x
—	—	—		ORS<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 = 值取决于具体条件

bit 7-5 未实现: 读为 0

bit 4-0 **ORS<4:0>: 运放输出改写源选择位**

11111 = 保留。不要使用。

•
•
•

10110 = 保留。不要使用。

10101 = 改写源为 COG2D⁽¹⁾

10100 = 改写源为 COG2C⁽¹⁾

10011 = 改写源为 COG2B⁽¹⁾

10010 = 改写源为 COG2A⁽¹⁾

10001 = 改写源为 COG1C

10000 = 改写源为 COG1C

01111 = 改写源为 COG1B

01110 = 改写源为 COG1A

01101 = 改写源为 LC3_out

01100 = 改写源为 LC2_out

01011 = 改写源为 LC1_out

01010 = 改写源为 ZCD1_output

01001 = 改写源为 sync_C4OUT⁽¹⁾

01000 = 改写源为 sync_C3OUT⁽¹⁾

00111 = 改写源为 sync_C2OUT

00110 = 改写源为 sync_C1OUT

00101 = 改写源为 PWM6_out⁽¹⁾

00100 = 改写源为 PWM5_out

00011 = 改写源为 PWM4_out⁽¹⁾

00010 = 改写源为 PWM3_out

00001 = 改写源为 CCP2_out⁽¹⁾

00000 = 改写源为 CCP1_out

注 1: 仅限 PIC16(L)F1768/9。

PIC16(L)F1764/5/8/9

寄存器 29-3: OPAxNCHS: 运放反相通道源选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCH<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 NCH<3:0>: 运放反相输入通道选择位

1111 = 保留。不要使用。

•
•
•

1010 = 保留。不要使用。

1001 = 可编程斜坡发生器 PRG2_out⁽¹⁾

1000 = 可编程斜坡发生器 PRG1_out

0111 = 保留。不要使用。

0110 = FVR_Buffer2

0101 = DAC4_out⁽¹⁾

0100 = DAC3_out

0011 = DAC2_out⁽¹⁾

0010 = DAC1_out

0001 = OPAxIN1- 引脚⁽¹⁾

0000 = OPAxIN0- 引脚

注 1: 仅限 PIC16(L)F1768/9。

寄存器 29-4: OPAxPCHS: 运放同相通道源选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—		PCH<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 **PCH<3:0>: 运放同相输入通道选择位**
1111 = 保留。不要使用。

•

•

•

1010 = 保留。不要使用。

1001 = 可编程斜坡发生器 PRG2_out⁽¹⁾

1000 = 可编程斜坡发生器 PRG1_out

0111 = 保留。不要使用。

0110 = FVR_Buffer2

0101 = DAC4_out⁽¹⁾

0100 = DAC3_out

0011 = DAC2_out⁽¹⁾

0010 = DAC1_out

0001 = OPAxIN1+ 引脚⁽¹⁾

0000 = OPAxIN0+ 引脚

注 1: 仅限 PIC16(L)F1768/9。

表 29-2: 与运放相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELB ⁽²⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC ⁽²⁾	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
ANSELC ⁽³⁾	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
DACxCON0	EN	FM	OE1	—	PSS<1:0>		NSS<1:0>		193
DACxREF	---	---	---		REF<4:0>				193
DACxREFL ⁽²⁾					REF<7:0>				200
DACxREFH ⁽²⁾					REF<15:8>				200
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		173
OPAxCON	EN	—	—	UG	—	ORPOL	ORM<1:0>		348
OPAxNCHS	—	—	—	—			NCH<3:0>		350
OPAxPCHS	—	—	—	—			PCH<3:0>		351
OPAxORS	—	—	—		ORS<4:0>				349
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC ⁽²⁾	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TRISC ⁽³⁾	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	1	0	150

图注: — = 未实现位, 读为 0。运放不使用阴影单元。

注 1: 未实现位, 读为 1。

2: 仅限 PIC16(L)F1768/9。

3: 仅限 PIC16(L)F1764/5。

30.0 可编程斜坡发生器（PRG）模块

可编程斜坡发生器（PRG）模块用于提供上升和下降线性斜坡。典型应用包括固定频率、连续电流和电流模式开关电源的斜率补偿。斜率补偿是这些电源必需的特性，因为它可以防止占空比大于 50% 时的频率不稳定。

PRG 具有以下特性：

- 线性正电压斜坡和负电压斜坡输出
- 可编程的拉 / 灌电流源
- 内部和外部参考电压选择
- 内部和外部时序源选择

图 30-1 给出了 PRG 的简化框图。

30.1 基本操作

PRG 可以在 3 种电压斜坡发生器模式下工作：

- 下降电压（斜率补偿）
- 上升电压
- 交替上升和下降电压

在上升或下降模式下，内部电容在 `set_falling` 时序输入为真时放电，在 `set_rising` 时序输入为真时通过内部产生的恒定电流充电。所产生的线性斜坡从选定的电压输入值开始，并在 `set_falling` 时序输入终止斜坡时恢复为该电压值。当两个时序输入均为真时，`set_falling` 输入优先。

要使用单端源来控制操作，可以为 `set_rising` 和 `set_falling` 输入选择同一输入源，并使用相应的极性控制位对其中某个输入的极性进行反相。

在交替模式下，不对电容进行放电，而是先在一个方向充电，然后在另一个方向充电，在这两者之间交替。

所有模式的输入选择是相同的。输入电压由以下任意输入源提供：

- PRGxIN0 或 PRGxIN1 引脚
- 内部固定电压参考（FVR）的缓冲输出
- 任一内部 DAC

时序源从以下来源选择：

- 任意比较器的同步输出
- 任意 PWM 输出
- 任意 I/O 引脚

斜坡输出可用作任意比较器或运放的输入。

30.1.1 斜率补偿

斜率补偿的工作方式是在每个 PWM 周期开始时对内部电容进行快速放电。内部电容的一端与电压输入源连接，另一端与内部灌电流源连接。内部灌电流源以可编程速率对该电容进行充电。在电容充电时，将从电压源电压中减去电容电压，从而以所需的速率产生线性电压衰减（见图 30-2）。当 `set_falling` 时序输入变为真时，斜坡终止，电容进行放电。下一个斜坡在 `set_rising` 时序输入变为真时开始。

通过将 PRGxCON0 寄存器的 OS 位置 1，可以使能可选的单次定时，这种情况下会改写 `set_rising` 时序输入并使短路开关保持闭合至少单次定时周期（通常为 50 ns），确保电容完全放电。在单次定时周期期间出现的边沿敏感时序输入会被忽略。在单次定时周期期间出现并延续到该周期之后的电平敏感时序输入会被暂停，直到单次定时时间结束为止。

30.1.2 斜坡发生

斜坡发生类似于斜率补偿，只是其斜率可以为既上升也下降，也可以为仅上升。

30.1.2.1 交替上升 / 下降斜坡

交替上升 / 下降斜坡发生功能的工作方式是，利用内置拉电流源和灌电流源，依靠内部模拟开关和时序源的同步控制，使模块的输出电压先发生斜升然后发生斜降。

在初始化之后，输出电压通过拉电流源以可编程速率线性斜升，直到 `set_falling` 时序源变为真，此时会断开拉电流源。同时会接通灌电流源，同样也是以可编程速率使输出电压线性斜降，直到 `set_rising` 时序输入变为真，从而使斜坡斜率反转方向。然后重复这一过程，产生类似于锯齿的波形，如图 30-3 和图 30-4 所示。

`set_rising` 和 `set_falling` 时序输入可以是边沿或电平敏感的，这使用 PRGxCON0 寄存器中相应的 REDG 和 FEDG 位进行选择。对于周期信号（如时钟），建议采用边沿敏感操作，而对于模拟电压限制触发信号（如比较器输出），建议采用电平敏感操作。

当使能单次定时（OS 位置 1）时，下降和上升斜坡将至少持续单次定时周期的时间。在单次定时周期期间出现的边沿敏感时序输入会被忽略。在单次定时周期期间出现并延续到该周期之后的电平敏感时序输入会被暂停，直到单次定时时间结束为止。

30.1.2.2 上升斜坡

上升斜坡模式与斜率补偿模式基本相同，只是斜坡具有上升斜率而不是下降斜率。内部电容的一端与电压输入源连接，另一端与内部拉电流源连接。内部拉电流源以可编程速率对该电容进行充电。在电容充电时，电容电压将与电压源电压相加，从而以所需的速率产生线性电压上升（见图 30-5）。当 **set_falling** 时序输入变为真时，斜坡终止，电容进行放电。下一个斜坡在 **set_rising** 时序输入变为真时开始。

通过将 PRGxCON0 寄存器的 OS 位置 1，可以使能可选的单次定时，这种情况下会改写 **set_rising** 时序输入并使短路开关保持闭合至少单次定时周期（通常为 50 ns），确保电容完全放电。在单次定时周期期间出现的边沿敏感时序输入会被忽略。在单次定时周期期间出现并延续到该周期之后的电平敏感时序输入会被暂停，直到单次定时时间结束为止。

30.2 使能、就绪和运行

PRGxCON0 寄存器的 EN 位用于使能模拟电路，包括电流源。这使 PRG 模块可以做好准备，可以在开始工作之前达到稳定状态。当 EN 位置 1 时，时序输入会被使能，从而可以在 GO 位置 1 之前确定初始斜坡操作。电容短路开关在 EN 位置 1 时闭合，并在 GO 位为零时保持闭合状态。

PRGxCON1 寄存器的 RDY 位用于指示模拟电路和电流源是否已稳定。

PRGxCON0 寄存器的 GO 位用于使能开关控制电路，从而使 PRG 开始工作。GO 从清零变为置 1 会触发单次定时，从而使电容短路开关闭合延长单次定时周期的时间。

为了确保产生可预测的操作，请先将 EN 位置 1，然后等待 RDY 位变为高电平，之后再将 GO 位置 1。

30.3 独立的 **set_rising** 和 **set_falling** 时序输入

时序输入决定斜坡何时开始和停止。在交替上升 / 下降模式下，斜坡在 **set_rising** 输入变为真时上升，在 **set_falling** 输入变为真时下降。在斜率补偿和上升斜坡模式下，电容在 **set_falling** 时序输入变为真时放电，斜坡在 **set_rising** 时序输入变为真时开始。**set_falling** 输入优先于 **set_rising** 输入。

30.4 电平与边沿时序敏感性

set_rising 和 **set_falling** 时序输入可以单独配置为电平或边沿敏感。

如果需要检测改写状态结束之后的时序输入真状态，则电平敏感操作会很有用。例如，对于可能会被单次定时的改写操作抑制的电容生成时序输入，电平敏感会很有用。配置为电平敏感时，如果电容输出在单次定时周期期间发生改变，则在单次定时周期结束时会检测到该变化。配置为边沿敏感时，该变化会被忽略。

对于周期性时序输入（如 PWM 产生的输入以及时钟），边沿敏感操作很有用。电平敏感周期信号的占空比可能会干扰其他时序输入。可以考虑对于交替斜坡模式使用电平敏感的 50% PWM 作为 **set_rising** 时序源，使用电平敏感的比较器作为 **set_falling** 时序源。如果比较器输出在 PWM 信号仍为高电平时反转斜坡方向，则在比较器信号变为低电平时，斜坡会错误地再次反转方向。对于设置为边沿敏感的 **set_rising** 时序输入，在相同情况下，只会在 PWM 信号的上升沿正确地使斜坡输出变为上升。

set_rising 和 **set_falling** 时序输入边沿敏感性使用 PRGxCON1 寄存器中相应的 REDG 和 FEDG 位进行选择。

30.5 单次定时最长时间

在斜率补偿和上升斜坡模式下，单次定时定时器可以确保一个最小电容放电时间；在交替斜坡模式下，可以确保一个最小上升或下降斜坡持续时间。将 PRGxCON0 寄存器的 OS 位置 1 会使能单次定时定时器。

30.6 DAC 电压源

当使用任一 DAC 作为电压源时，预期的电压偏移量等于电流设置乘以 DAC 等效电阻。在斜率补偿和斜坡模式下，它是一个恒定的偏移量，在交替模式下，它是一个正 / 负阶跃偏移量。为了避免这种限制，可以通过一个设置为单位增益的运放将 DAC 输出传送至 PRG 输入。

30.7 休眠期间的操作

RG 模块不会受休眠影响。

30.8 复位的影响

RG 模块会复位为禁止状态。

图 30-1: PRG 模块简化框图

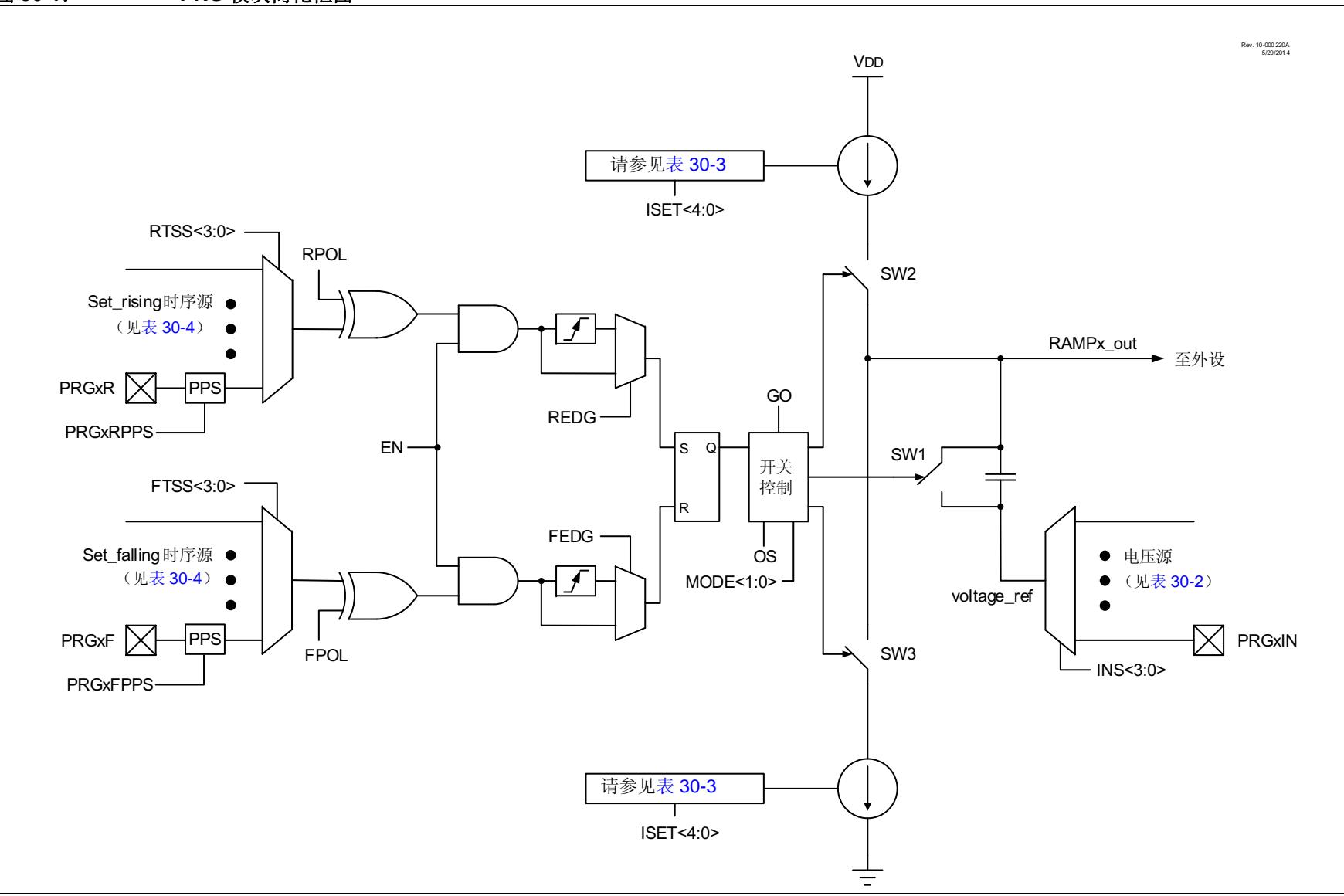


图 30-2: 斜率补偿 (下降斜坡) 时序图 (MODE = 00)

Rev. 10-000223A
5/2/2014

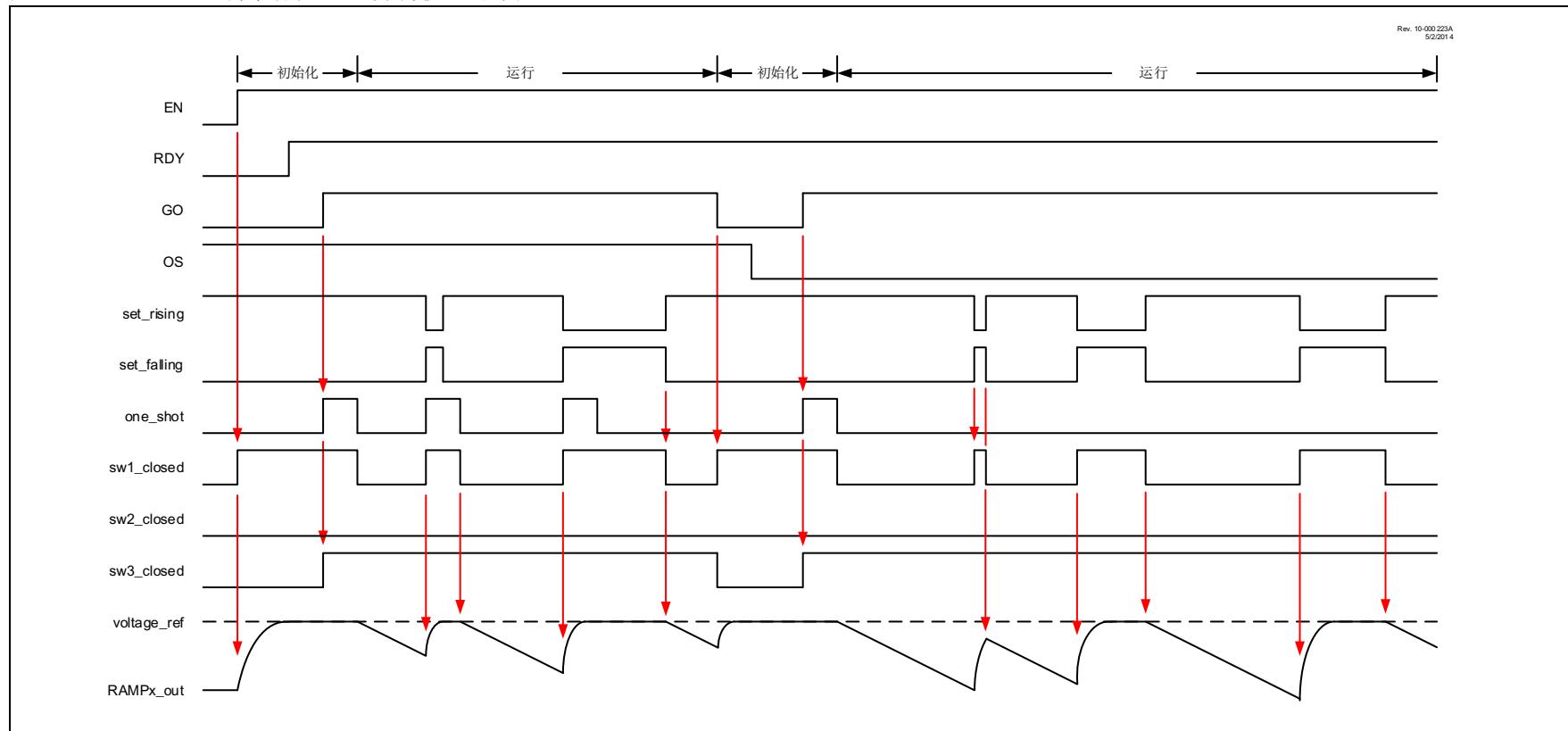


图 30-3: 交替上升 / 下降斜坡发生时序图 (OS = 0, MODE = 01)

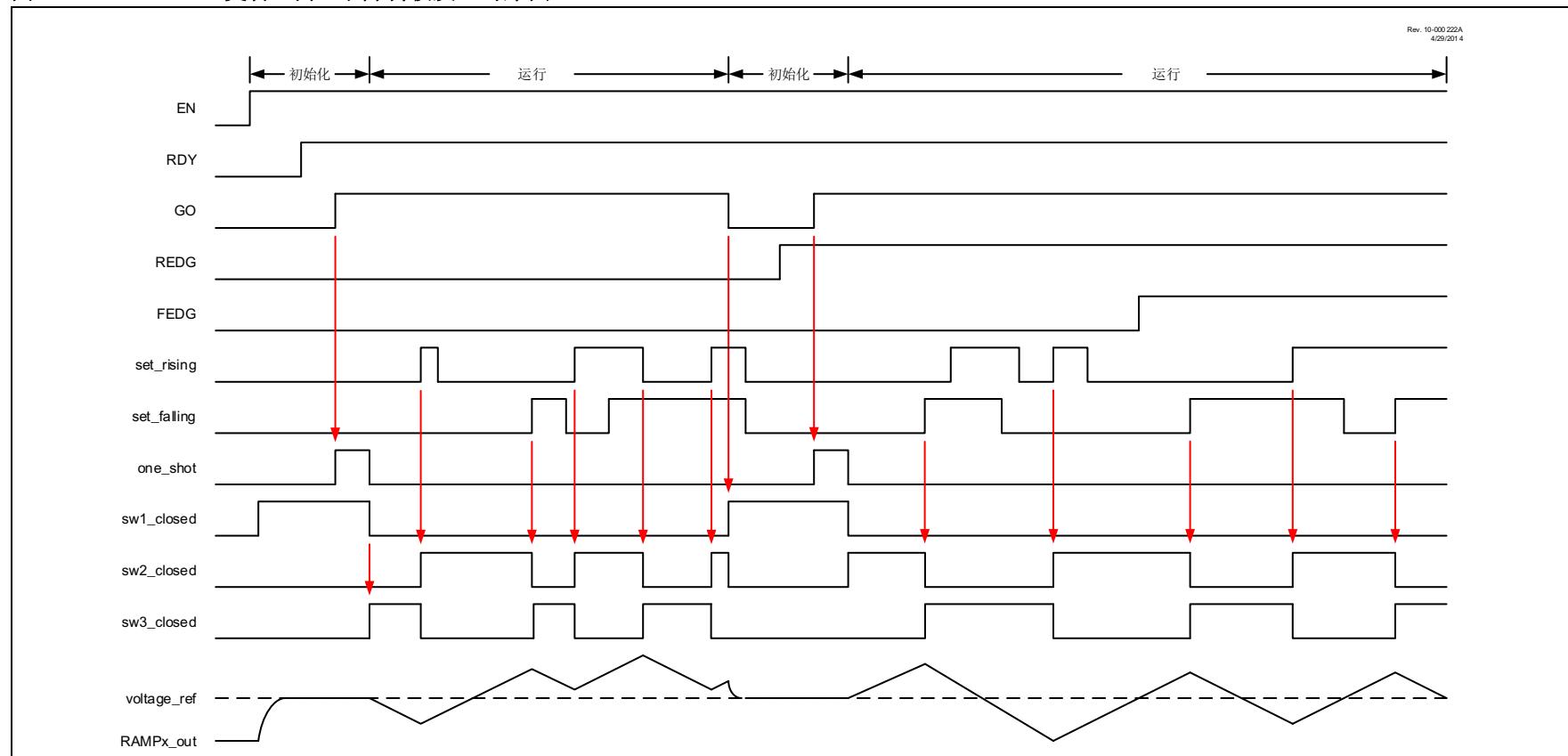


图 30-4: 交替上升 / 下降斜坡发生时序图 (OS = 1, MODE = 01)

Rev. 10.00/23A
5/2/2014

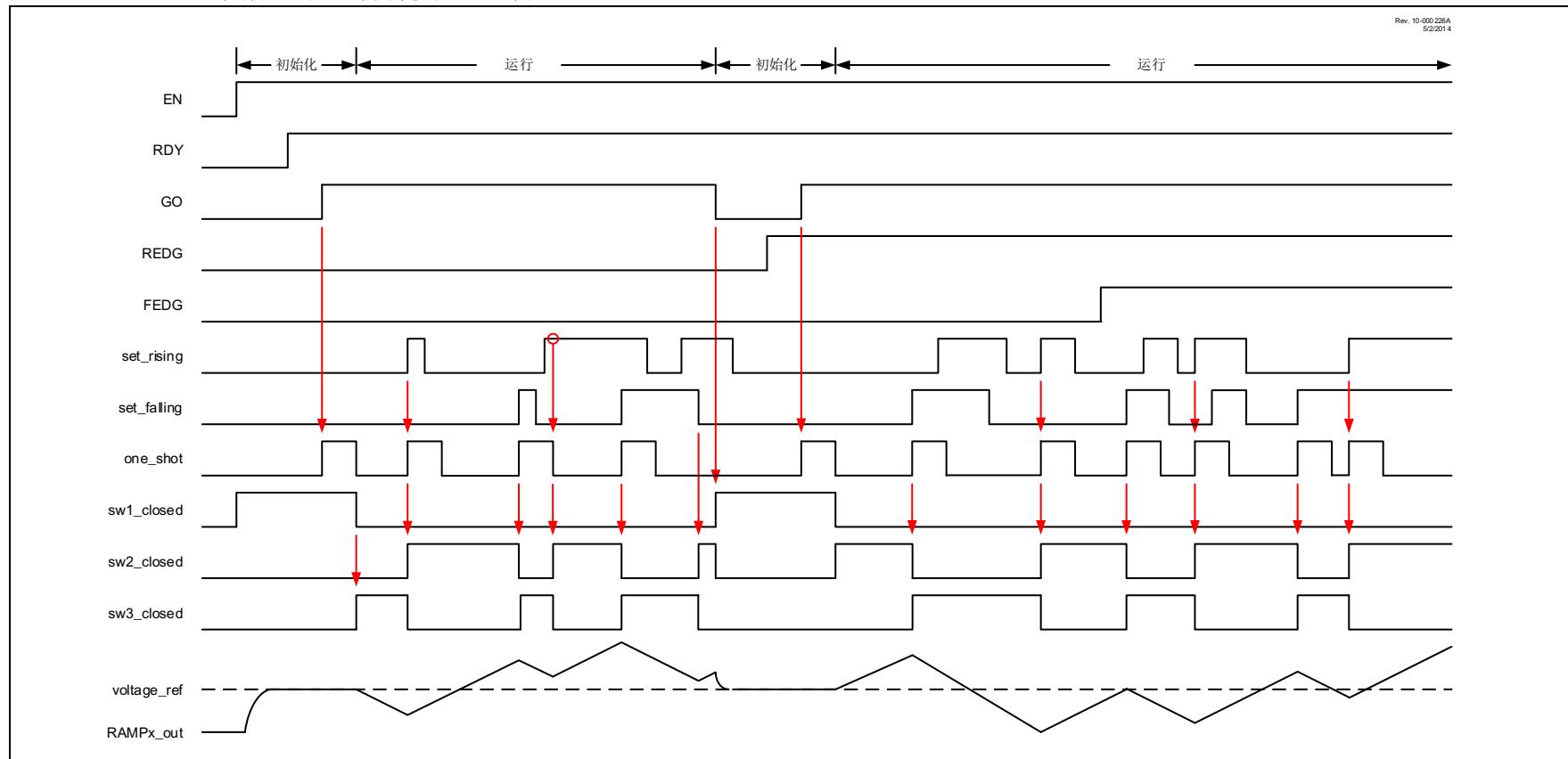
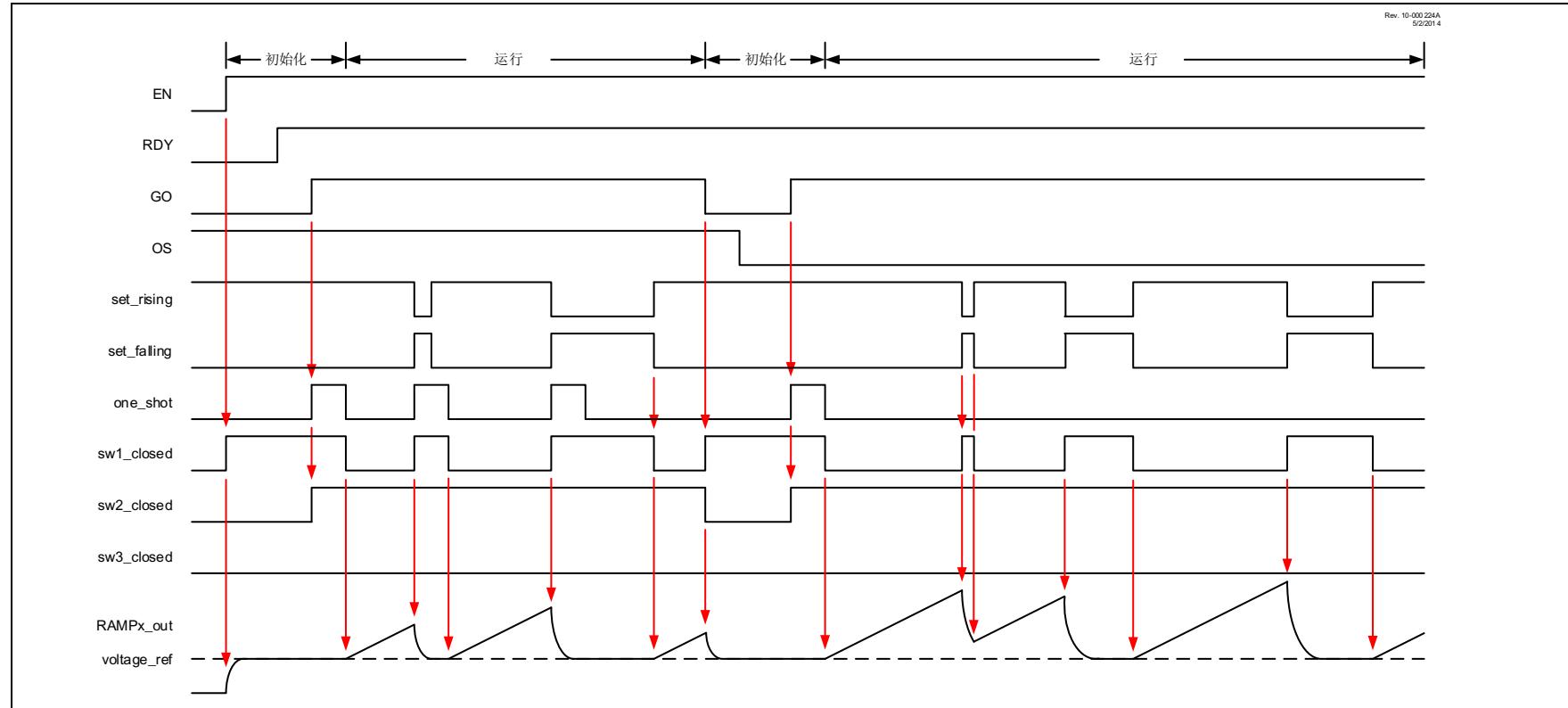


图 30-5: 上升斜坡发生时序图 (MODE = 10)



30.9 斜率补偿应用

图 30-6 给出了一个斜率补偿电路示例。PRG 输入电压为 PRGxIN，它与运放输出共用一个 I/O 引脚。运放输出设计为在预期的峰值电流检测电压（称为 VREF）下工作。PRG 输出电压的起始值为 VREF，并以小于目标电路电流检测电压上升速率一半的速率下降。因此，可以使用公式 30-1 计算以 $V/\mu s$ 表示的补偿器斜率。

公式 30-1：

$$\frac{V}{\mu s} \geq \frac{V_{REF}}{2 \text{ PWM 周期 } (\mu s)}$$

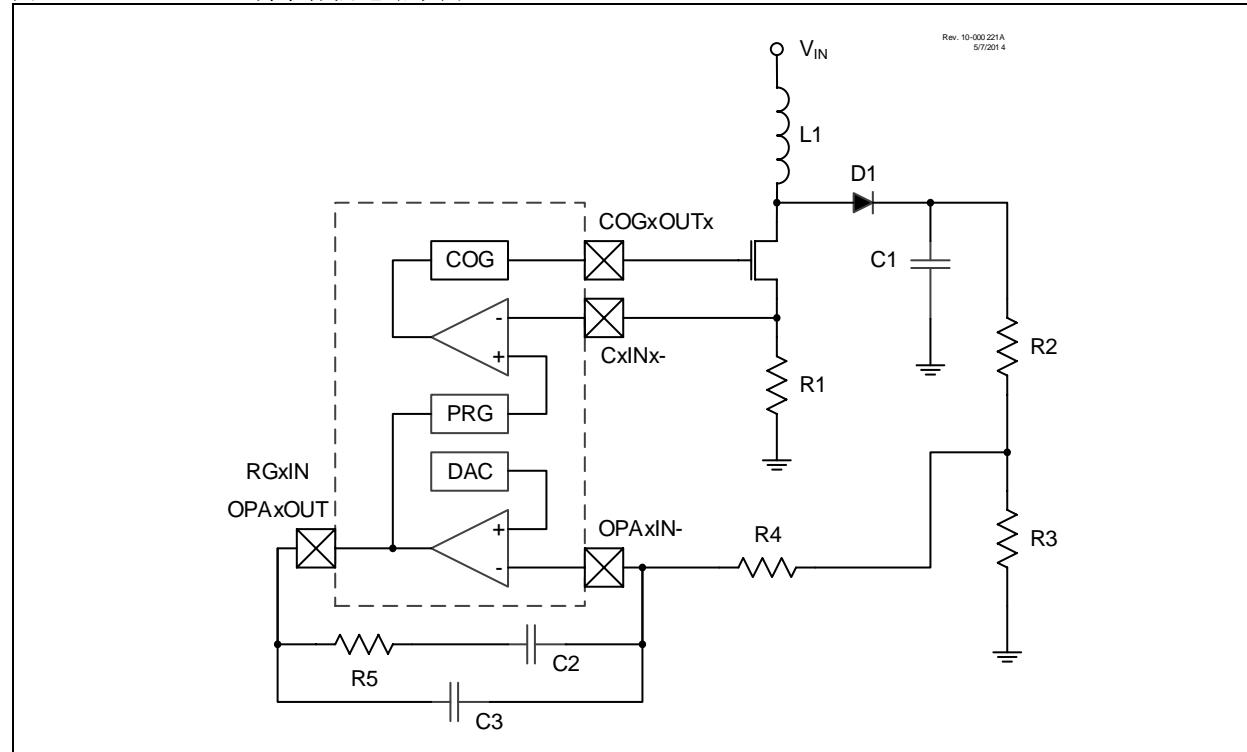
例如，当电路使用 1Ω 电流检测电阻，峰值电流为 $1A$ 时，以电压形式表示的峰值电流为 $1V$ 。因此，对于该示例，运放输出应设计为在 $1V$ 下工作。如果电源 PWM 频率为 1MHz ，则周期为 $1\mu s$ 。因此，所需的斜率为 $0.5V/\mu s$ ，其计算方式如公式 30-2 所示。

公式 30-2：

$$\frac{V_{REF}}{2 \text{ PWM 周期 } (\mu s)} = \frac{I}{2 \mu s} = 0.5V/\mu s$$

注：对应于 $0.5V/\mu s$ 的设置为 $ISET<4:0> = 6$

图 30-6：斜率补偿电路示例



PIC16(L)F1764/5/8/9

30.10 寄存器定义：斜率补偿控制

表 30-1 列出了 PRG 外设的长位名称前缀。更多信息，请参见第 1.1 节“寄存器和位命名约定”。

表 30-1：

外设	位名称前缀
PRG1	RG1
PRG2 ⁽¹⁾	RG2

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 30-1： PRGxCON0：可编程斜坡发生器控制 0 寄存器

R/W-0/0	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
EN	—	FEDG	REDG	MODE<1:0>		OS	GO
bit 7							

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 = 值取决于配置位

bit 7 **EN:** 可编程斜坡发生器使能位

1 = 使能 PRG 模块

0 = 禁止 PRG 模块

bit 6 未实现：读为 0

bit 5 **FEDG:** set_falling 输入模式选择位

1 = set_falling 时序输入为电平敏感型

0 = set_falling 时序输入为边沿敏感型

bit 4 **REDG:** set_rising 输入模式选择位

1 = set_rising 时序输入为电平敏感型

0 = set_rising 时序输入为边沿敏感型

bit 3-2 **MODE<1:0>:** 可编程斜坡发生器模式选择位

11 = 保留

10 = 上升斜坡发生器

01 = 交替上升 / 下降斜坡发生器

00 = 斜率补偿

bit 1 **OS:** 单次定时使能位

1 = 使能单次定时。通过单次定时在内部控制最小电容放电时间。

0 = 禁止单次定时。电容在时序输入为真时放电。

bit 0 **GO:** 斜坡发生控制启动位

如果 EN = 0

该位强制设为 0

如果 EN = 1

1 = 斜率或斜坡功能正在工作

0 = 斜率或斜坡功能未在工作。所有电流源开关均开路，电容放电开关闭合。

寄存器 30-2: PRGxCON1: 可编程斜坡发生器控制 1 寄存器

U-0	U-0	U-0	U-0	U-0	R-0	R/W-0/0	R/W-0/0
—	—	—	—	—	RDY	FPOL	RPOL
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 = 值取决于配置位

bit 7-3 未实现: 读为 0

bit 2 **RDY:** 斜率发生器就绪状态位

1 = PRG 就绪

0 = PRG 未就绪

bit 1 **FPOL:** 下降事件极性选择位

1 = set_falling 时序输入为低电平有效

0 = set_falling 时序输入为高电平有效

bit 0 **RPOL:** 上升事件极性选择位

1 = set_rising 时序输入为低电平有效

0 = set_rising 时序输入为高电平有效

寄存器 30-3: PRGxINS: 电压输入选择寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	INS<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

q = 值取决于配置位

bit 7-3 未实现: 读为 0

bit 2-0 **INS<2:0>:** 电压输入选择位

选择斜坡起始电压的电压源。请参见表 30-2。

表 30-2: 电压输入源

INS<2:0>	PIC16(L)F1764/5 电压源	PIC16(L)F1768/9 电压源	
111	保留	保留	保留
110	保留	DAC4_output	DAC4_output
101	DAC3_output	DAC3_output	DAC3_output
100	保留	DAC2_output	DAC2_output
011	DAC1_output	DAC1_output	DAC1_output
010	FVR_buffer1	FVR_buffer1	FVR_buffer1
001	保留	PRG1IN1/OPA2OUT	PRG2IN1/OPA1OUT
000	PRG1IN0/OPA1OUT	PRG1IN0/OPA1OUT	PRG2IN0/OPA2OUT

PIC16(L)F1764/5/8/9

寄存器 30-4: PRGxCON2: 可编程斜坡发生器控制 2 寄存器

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
—	—	—		ISET<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 = 值取决于配置位

bit 7-5 未实现: 读为 0

bit 4-0 ISET<4:0>: PRG 拉电流 / 灌电流源设置位

拉电流 / 灌电流源设置和斜率速率。请参见表 30-3。

表 30-3: 可编程斜坡发生器电流设置

ISET<4:0>	电流设置 (μ A)	斜率速率 (V/ μ s)	ISET<4:0>	电流设置 (μ A)	斜率速率 (V/ μ s)
0h	2	0.2	10h	10	1.0
1h	2.5	0.25	11h	11	1.1
2h	3	0.3	12h	12	1.2
3h	3.5	0.35	13h	13	1.3
4h	4	0.4	14h	14	1.4
5h	4.5	0.45	15h	15	1.5
6h	5	0.5	16h	16	1.6
7h	5.5	0.55	17h	17	1.7
8h	6	0.6	18h	18	1.8
9h	6.5	0.65	19h	19	1.9
Ah	7	0.7	1Ah	20	2.0
Bh	7.5	0.75	1Bh	21	2.1
Ch	8	0.8	1Ch	22	2.2
Dh	8.5	0.85	1Dh	23	2.3
Eh	9	0.9	1Eh	24	2.4
Fh	9.5	0.95	1Fh	25	2.5

寄存器 30-5: PRGxRTSS: SET_RISING 时序源选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	RTSS<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 = 值取决于配置位

bit 7-4 未实现: 读为 0

bit 3-0 RTSS<3:0>: set_rising 时序源选择位

请参见表 30-4。

寄存器 30-6: PRGxFTSS: SET_FALLING 时序源选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	FTSS<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 = 值取决于配置位

bit 7-4 未实现: 读为 0

bit 3-0 FTSS<3:0>: set_falling 时序源选择位

请参见表 30-4。

表 30-4: 可编程斜坡发生器时序源

RTSS<3:0>/FTSS<3:0>	时序源	RTSS<3:0>/FTSS<3:0>	时序源
0000	sync_C1OUT	1000	PWM6_output ⁽²⁾
0001	sync_C2OUT	1001	CCP1_out
0010	sync_C3OUT ⁽²⁾	1010	CCP2_out ⁽²⁾
0011	sync_C4OUT ⁽²⁾	1011	保留
0100	PRGxR/PRGxF 引脚 ⁽¹⁾	1100	保留
0101	PWM3_output	1101	保留
0110	PWM4_output ⁽²⁾	1110	保留
0111	PWM5_output	1111	保留

注 1: 输入引脚使用 PRGxRPPS 或 PRGxFPPS 寄存器进行选择。

2: 仅限 PIC16(L)F1768/9。

PIC16(L)F1764/5/8/9

表 30-5: 与 PRG 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
PRG1CON0	EN	—	FEDG	REDG	MODE<1:0>	OS	GO	360	
PRG1CON1	—	—	—	—	—	RDY	FPOL	RPOL	361
PRG1CON2	—	—	—	—	ISET<4:0>	—	—	—	362
PRG1INS	—	—	—	—	—	—	INS<2:0>	—	361
PRG1RPPS	—	—	—	—	PRG1RPPS<4:0>	—	—	—	363
PRG1FPPS	—	—	—	—	PRG1FPPS<4:0>	—	—	—	363
PRG1RTSS	—	—	—	—	—	RTSS<3:0>	—	—	158, 160
PRG1FTSS	—	—	—	—	—	FTSS<3:0>	—	—	158, 160
PRG2CON0 ⁽¹⁾	EN	—	FEDG	REDG	MODE<1:0>	OS	GO	360	
PRG2CON1 ⁽¹⁾	—	—	—	—	—	RDY	FPOL	RPOL	361
PRG2CON2 ⁽¹⁾	—	—	—	—	ISET<4:0>	—	—	—	362
PRG2INS ⁽¹⁾	—	—	—	—	—	—	INS<2:0>	—	361
PRG2RPPS ⁽¹⁾	—	—	—	—	PRG2RPPS<4:0>	—	—	—	363
PRG2FPPS ⁽¹⁾	—	—	—	—	PRG2FPPS<4:0>	—	—	—	363
PRG2RTSS ⁽¹⁾	—	—	—	—	—	RTSS<3:0>	—	—	158, 160
PRG2FTSS ⁽¹⁾	—	—	—	—	—	FTSS<3:0>	—	—	158, 160
PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	150
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	152

图注: — = 未实现位, 读为 0。PRG 模块不使用阴影单元。

注 1: 仅限 PIC16(L)F1768/9。

31.0 数据信号调制器 (DSM)

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 MDxOUT 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：

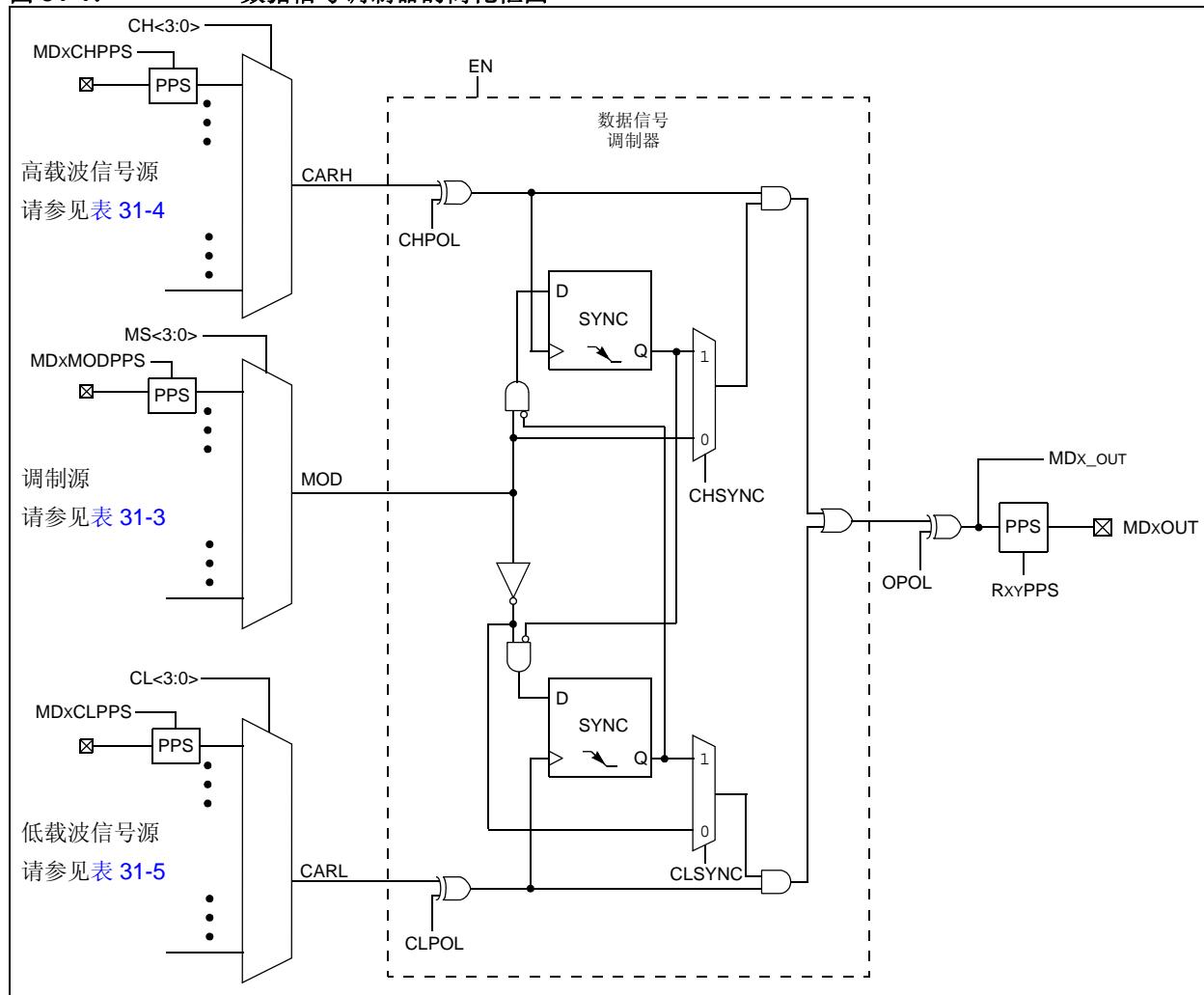
- 频移键控 (Frequency-Shift Keying, FSK)
- 相移键控 (Phase-Shift Keying, PSK)
- 开关键控 (On-Off Keying, OOK)

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 31-1 给出了数据信号调制器外设的简化框图。

图 31-1: 数据信号调制器的简化框图



31.1 DSM 工作原理

DSM 模块通过将 MDxCON 寄存器的 EN 位置 1 来使能。清零 MDxCON 寄存器中的 EN 位时，将会通过自动将高载波信号和低载波信号切换至 V_{SS} 信号源来禁止 DSM 模块。调制器信号源也会被切换至 MDxCON0 寄存器中的 BIT 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 EN 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 EN 位置 1 且 DSM 模块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。

通过在引脚的 PPS 控制寄存器（见 [寄存器 12-2](#)）中选择所需的 DSM 模块，可以将调制输出信号输出到任意器件 I/O 引脚上。如果输出不送到任何 I/O 引脚，则 DSM 模块将保持工作状态并继续混合信号，但输出值不会送到任何引脚。

31.2 调制器信号源

调制器信号通过配置 MDxSRC 寄存器的 MS<4:0> 位来进行选择。[表 31-3](#) 列出了选择范围。

31.3 载波信号源

高载波信号通过配置 MDxCARH 寄存器的 CH<3:0> 位来进行选择。[表 31-4](#) 列出了选择范围。

低载波信号通过配置 MDxCARL 寄存器的 CL<3:0> 位来进行选择。[表 31-5](#) 列出了选择范围。

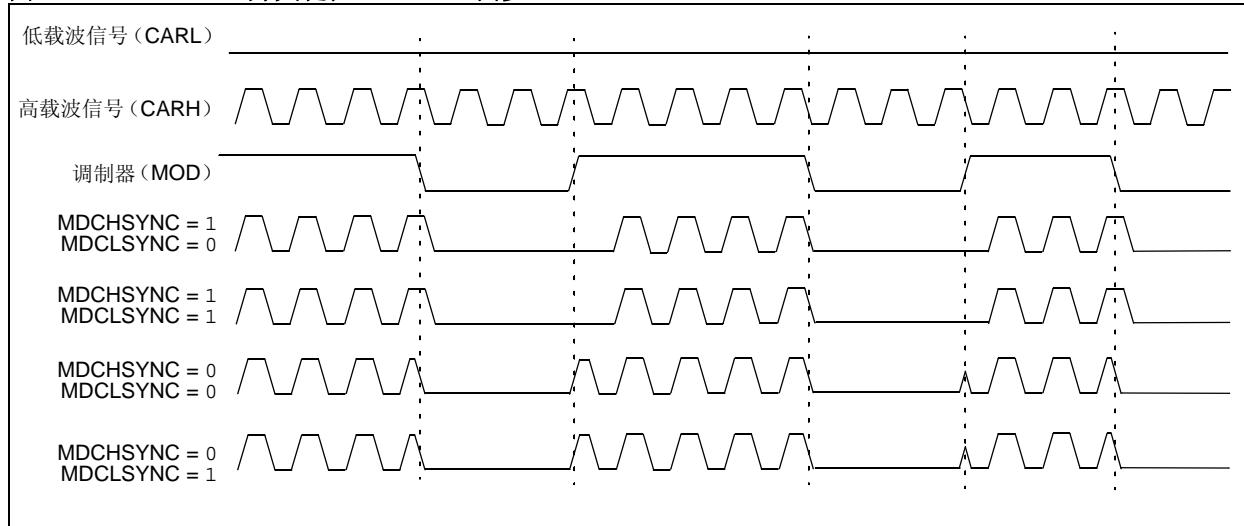
31.4 载波同步

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。当使能同步时，DSM 将允许在切换时进行混合的载波脉冲先变为低电平，然后再切换为另一个载波源。

对于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 MDxCON1 寄存器的 CHSYNC 位置 1 来使能。低载波信号的同步通过将 MDxCON1 寄存器的 CLSYNC 位置 1 来使能。

[图 31-1](#) 至 [图 31-5](#) 给出了使用各种同步方法的时序图。

图 31-2: 开关键控 (OOK) 同步



例 31-1: 无同步 (MDSHSYNC = 0, MDCLSYNC = 0)

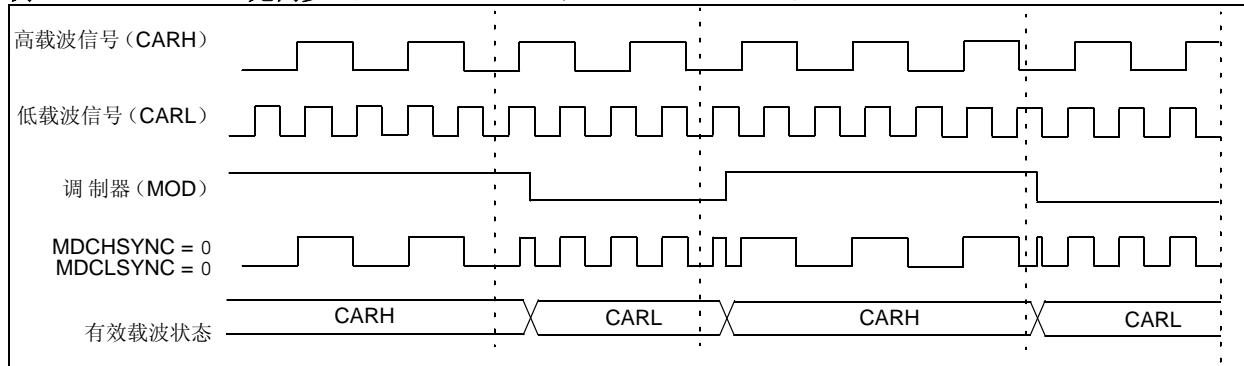
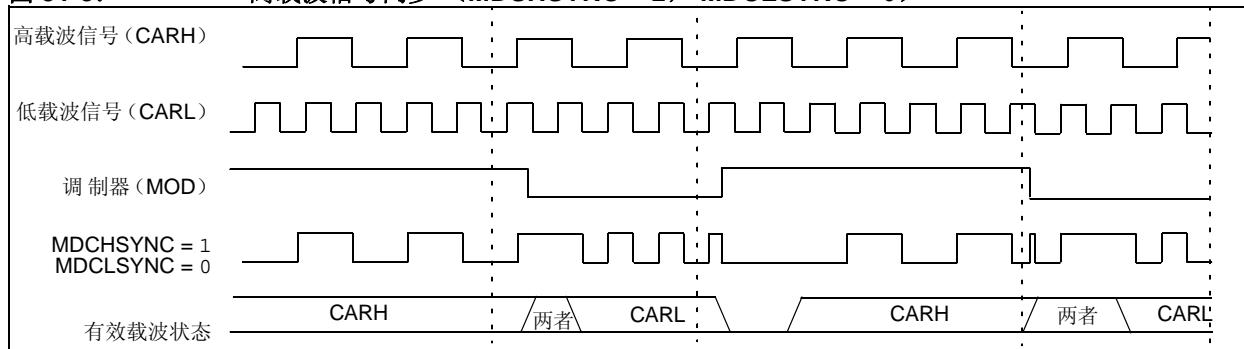


图 31-3: 高载波信号同步 (MDSHSYNC = 1, MDCLSYNC = 0)



PIC16(L)F1764/5/8/9

图 31-4: 低载波信号同步 ($\text{MDSHSYNC} = 0$, $\text{MDCLSYNC} = 1$)

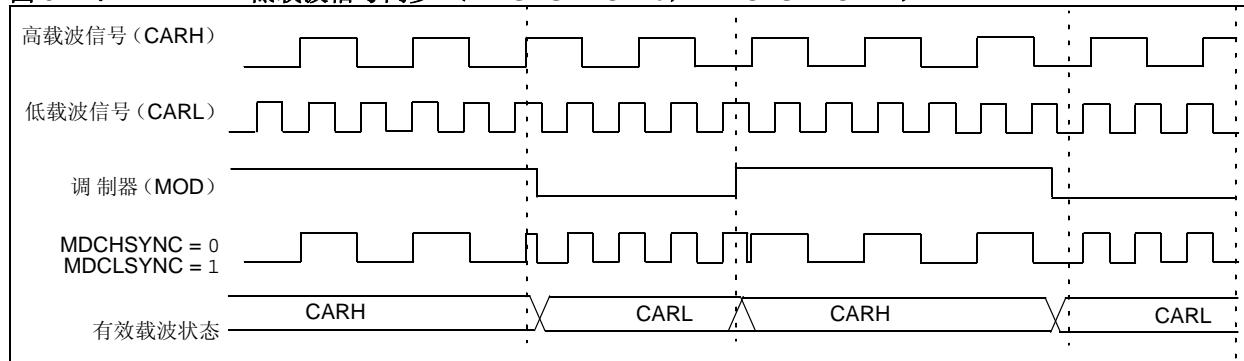
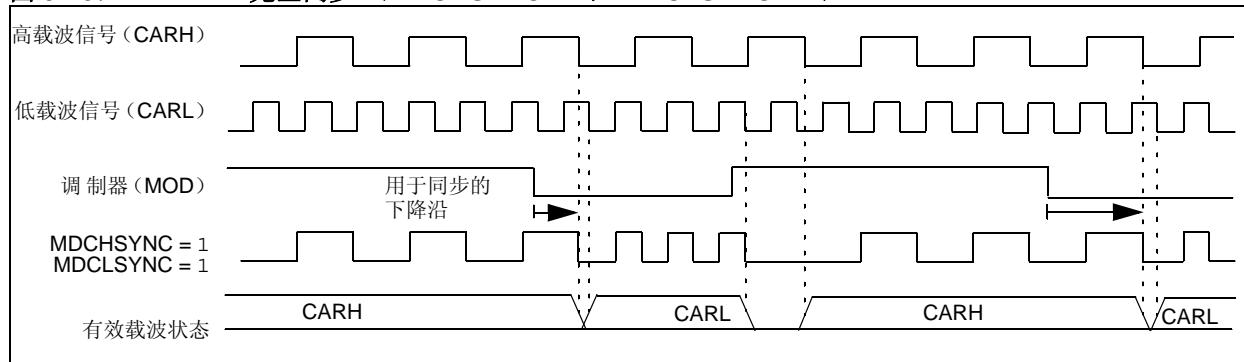


图 31-5: 完全同步 ($\text{MDCHSYNC} = 1$, $\text{MDCLSYNC} = 1$)



31.5 输入和输出引脚

通过 PPS 控制逻辑，可以选择调制源和载波源来自任意器件引脚。选择引脚需要两项设置：源选择决定将使用的 PPS，PPS 控制选择所需的引脚。[表 31-1](#) 列出了源寄存器和 PPS 寄存器。[寄存器 12-1](#) 和 [寄存器 12-2](#) 列出了 PPS 寄存器引脚选择范围。

表 31-1：

源	源寄存器	PPS 寄存器
调制	MDxSRC	MDxMODPPS
高载波信号	MDxCARH	MDxCHPPS
低载波信号	MDxCARL	MDxCLPPS

通过各个引脚 PPS 控制，可以选择任意器件引脚作为调制输出。关于引脚输出选择，请参见 [寄存器 12-2](#)。

31.6 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 MDxCON1 寄存器的 CHPOL 位置 1 来使能。对低载波信号源信号反相通过将 MDxCON1 寄存器的 CLPOL 位置 1 来使能。

31.7 可编程调制器数据

用户可以选择 MDxCON0 寄存器的 BIT 位作为调制器信号的信号源。当选择 BIT 源时，软件会通过在所需的调制高电平和低电平时间相应地置 1 和清零 BIT 位来生成调制信号。

31.8 调制输出极性

送到 MDxOUT 引脚上的调制输出信号也可以进行反相。调制输出信号反相通过将 MDxCON0 寄存器的 OPOL 位置 1 来使能。

31.9 休眠模式下的操作

DSM 模块不受休眠模式的影响。如果载波源和调制器输入源在休眠期间工作，则 DSM 也会在休眠期间工作。

31.10 复位的影响

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

31.11 寄存器定义：数据信号调制器

表 31-2 列出了 DSM 外设的长位名称前缀。更多信息，请参见第 1.1.2.2 节“长位名称”。

表 31-2：

外设	位名称前缀
DSM1	MD1
DSM2 ⁽¹⁾	MD2

注 1：仅限 PIC16(L)F1768/9 器件。

寄存器 31-1： MDxCON0：调制控制寄存器 0

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
EN	—	OUT	OPOL	—	—	—	BIT
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

EN： 调制器模块使能位

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

bit 6

未实现： 读为 0

bit 5

OUT： 调制器输出位

指示调制器模块的当前输出值。⁽¹⁾

bit 4

OPOL： 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 3-1

未实现： 读为 0

bit 0

BIT： 允许软件直接控制模块的调制源输入 ⁽²⁾

1 = 调制器使用高载波信号源
0 = 调制器使用低载波信号源

注 1：调制输出频率可能会高于更新该寄存器位的时钟，与该时钟异步；位值对于速度较高的调制器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 BIT 作为调制源。

寄存器 31-2: MDxCON1: 调制控制寄存器 1

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC
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 **CHPOL:** 调制高载波信号极性选择位

1 = 选定的高载波信号源反相

0 = 选定的高载波信号源不反相

bit 4 **CHSYNC:** 调制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 3-2 未实现: 读为 0

bit 1 **CLPOL:** 调制低载波信号极性选择位

1 = 选定的低载波信号源反相

0 = 选定的低载波信号源不反相

bit 0 **CLSYNC:** 调制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⁾

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

PIC16(L)F1764/5/8/9

寄存器 31-3: MDxSRC: 调制源控制寄存器

U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—		MS<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 MS<3:0> 调制源选择位

请参见表 31-3。

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

表 31-3: 调制源

MS<4:0>	调制源 PIC16(L)F1764/5	调制源 PIC16(L)F1768/9
11111-10100	固定低电平	固定低电平
10011	固定低电平	sync_C4OUT
10010	固定低电平	sync_C3OUT
10001	sync_C2OUT	sync_C2OUT
10000	sync_C1OUT	sync_C1OUT
01111	LC3_out	LC3_out
01110	LC2_out	LC2_out
01101	LC1_out	LC1_out
01100	固定低电平	PWM6_out
01011	PWM5_out	PWM5_out
01010	固定低电平	PWM4_out
01001	PWM3_out	PWM3_out
01000	固定低电平	CCP2_out
00111	CCP1_out	CCP1_out
00110	SDO_out	SDO_out
00101	固定低电平	COG2A
00100	DT	DT
00011	TX_out	TX_out
00010	COG1A	COG1A
00001	MDxBIT	MDxBIT
00000	MDxMODPPS 引脚选择	MDxMODPPS 引脚选择

寄存器 31-4: MDxCARH: 调制高载波信号控制寄存器

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u			
—	—	—	—	CH<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 CH<3:0> 调制器数据高载波信号选择位⁽¹⁾

请参见表 31-4。

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

表 31-4: 高载波信号源

CH<3:0>	高载波信号源 PIC16(L)F1764/5	高载波信号源 PIC16(L)F1768/9
1111	LC3_out	LC3_out
1110	LC2_out	LC2_out
1101	LC1_out	LC1_out
1100	固定低电平	PWM6_out
1011	PWM5_out	PWM5_out
1010	固定低电平	PWM4_out
1001	PWM3_out	PWM3_out
1000	固定低电平	CCP2_out
0111	CCP1_out	CCP1_out
0110	固定低电平	固定低电平
0101	固定低电平	固定低电平
0100	固定低电平	固定低电平
0011	固定低电平	固定低电平
0010	HFINTOSC	HFINTOSC
0001	Fosc	Fosc
0000	MDxCHPPS 引脚选择	MDxCHPPS 引脚选择

PIC16(L)F1764/5/8/9

寄存器 31-5: MDxCARL: 调制低载波信号控制寄存器

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—		CL<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 CL<3:0> 调制器数据低载波信号选择位⁽¹⁾

请参见表 31-5。

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

表 31-5: 低载波信号源

CL<3:0>	低载波信号源 PIC16(L)F1764/5	低载波信号源 PIC16(L)F1768/9
1111	LC3_out	LC3_out
1110	LC2_out	LC2_out
1101	LC1_out	LC1_out
1100	固定低电平	PWM6_out
1011	PWM5_out	PWM5_out
1010	固定低电平	PWM4_out
1001	PWM3_out	PWM3_out
1000	固定低电平	CCP2_out
0111	CCP1_out	CCP1_out
0110	固定低电平	固定低电平
0101	固定低电平	固定低电平
0100	固定低电平	固定低电平
0011	固定低电平	固定低电平
0010	HFINTOSC	HFINTOSC
0001	Fosc	Fosc
0000	MDxCLPPS 引脚选择	MDxCLPPS 引脚选择

表 31-6: 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
MDxCARH	—	—	—	—	CH<3:0>				373
MDxCARL	—	—	—	—	CL<3:0>				374
MDxSRC	—	—	—	MS<4:0>					371
MDxCON0	EN	—	OUT	OPOL	—	—	—	BIT	370
MDxCON1	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	370

图注: — = 未实现, 读为 0。数据信号调制器模块不使用阴影单元。

32.0 主同步串行端口（MSSP）模块

32.1 MSSP 模块概述

主同步串行端口（MSSP）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 ADC 转换器等。MSSP 模块有以下两种工作模式：

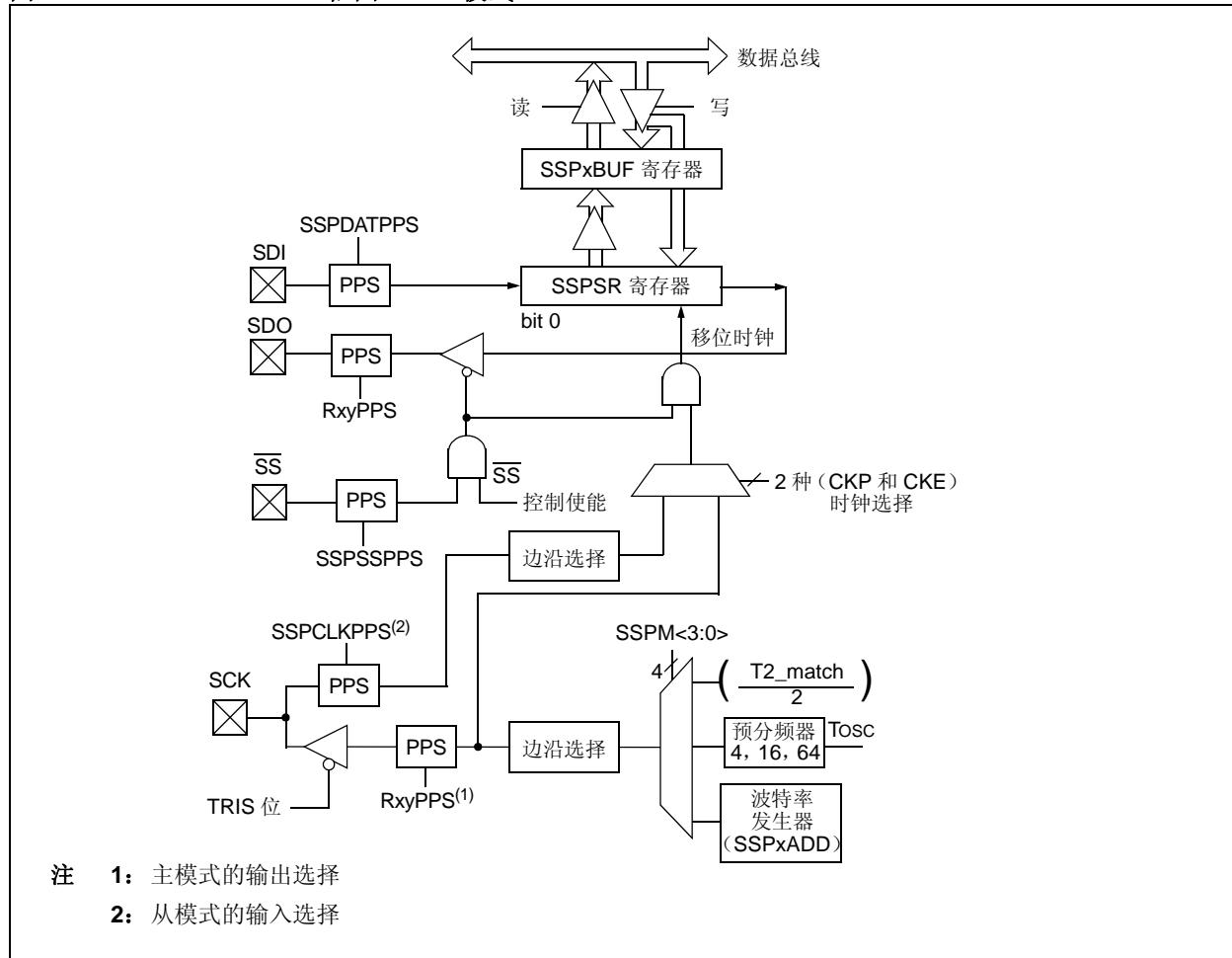
- 串行外设接口（SPI）
- I²C

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 32-1 给出了 SPI 接口模块的框图。

图 32-1: MSSP 框图（SPI 模式）



I²C 接口支持以下模式和特性：

- 主模式
- 从模式
- 字节无应答（从模式）
- 有限多主器件支持
- 7 位和 10 位寻址
- 启动和停止中断
- 中断屏蔽
- 时钟延长
- 总线冲突检测
- 广播呼叫地址匹配
- 地址掩码
- 地址保持模式和数据保持模式
- 可选的 SDA 保持时间

图 32-2 给出了主模式下 I²C 接口模块的框图。图 32-3

给出了从模式下 I²C 接口模块的框图。

图 32-2: MSSP 框图 (I²C 主模式)

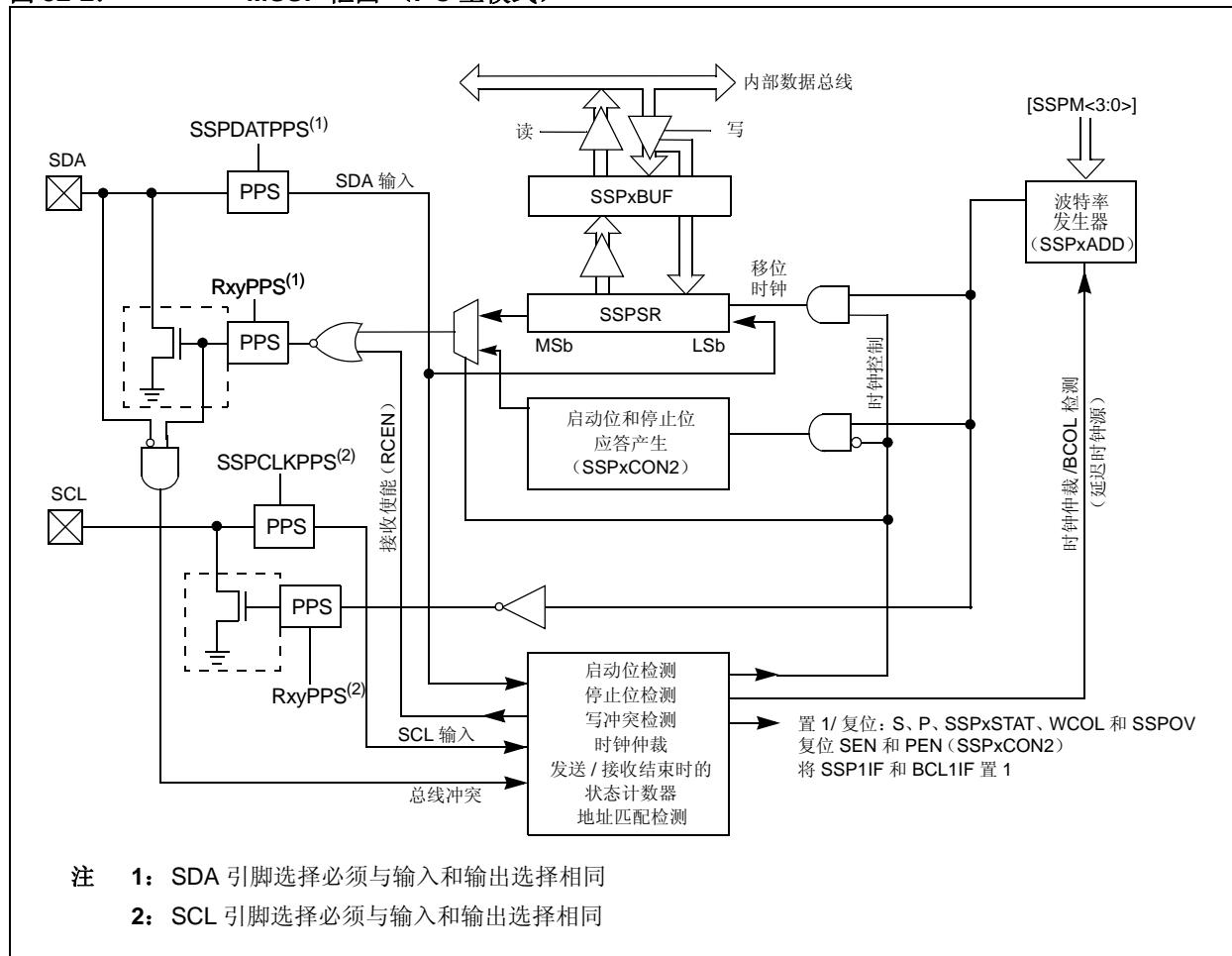
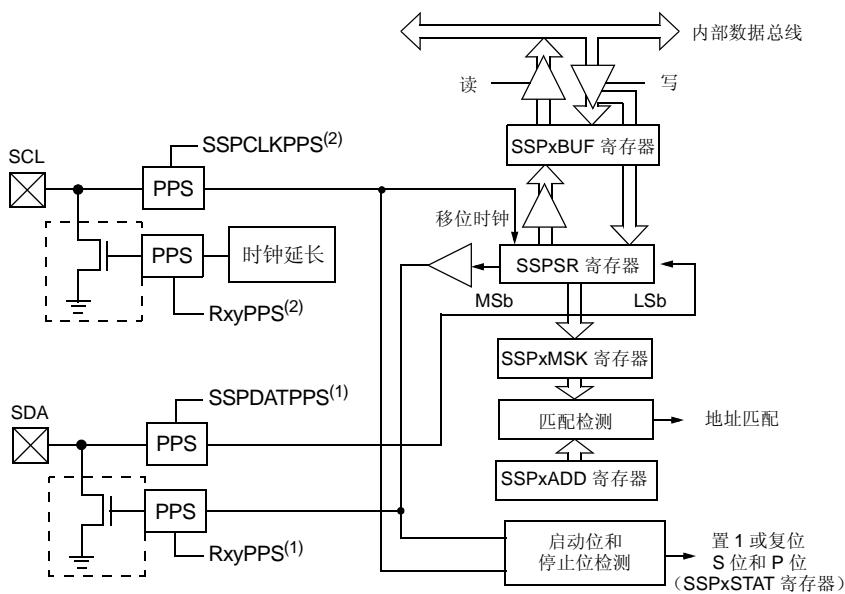


图 32-3: MSSP 框图 (I²C 从模式)



注 1: SDA 引脚选择必须与输入和输出选择相同

2: SCL 引脚选择必须与输入和输出选择相同

32.2 SPI 模式概述

串行外设接口（SPI）总线是以全双工模式工作的同步串行数据通信总线。器件在由主器件启动通信的主 / 从器件环境中进行通信。从器件通过称为从选择的片选进行控制。

SPI 总线规定了 4 种信号连接：

- 串行时钟（SCK）
- 串行数据输出（SDO）
- 串行数据输入（SDI）
- 从选择（SS）

图 32-1 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，从主器件到每个从器件都需要独立的从选择连接。

图 32-4 给出了主器件和多个从器件之间的典型连接。

主器件每次仅选择一个从器件。大多数从器件都具有三态输出，所以在未选择它们时，它们的输出信号会看起来好像与总线断开。

数据发送涉及到两个移位寄存器，它们都为 8 位，一个在主器件中，一个在从器件中。不论是对于主器件还是从器件，数据总是每次移出一位，最高有效位（MSb）先移出。与此同时，新的最低有效位（LSb）会被移入同一寄存器。

图 32-5 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。

在两个处理器的移位寄存器之间，数据在编程设定的时钟边沿被传送，并在相反的时钟边沿被锁存。

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。

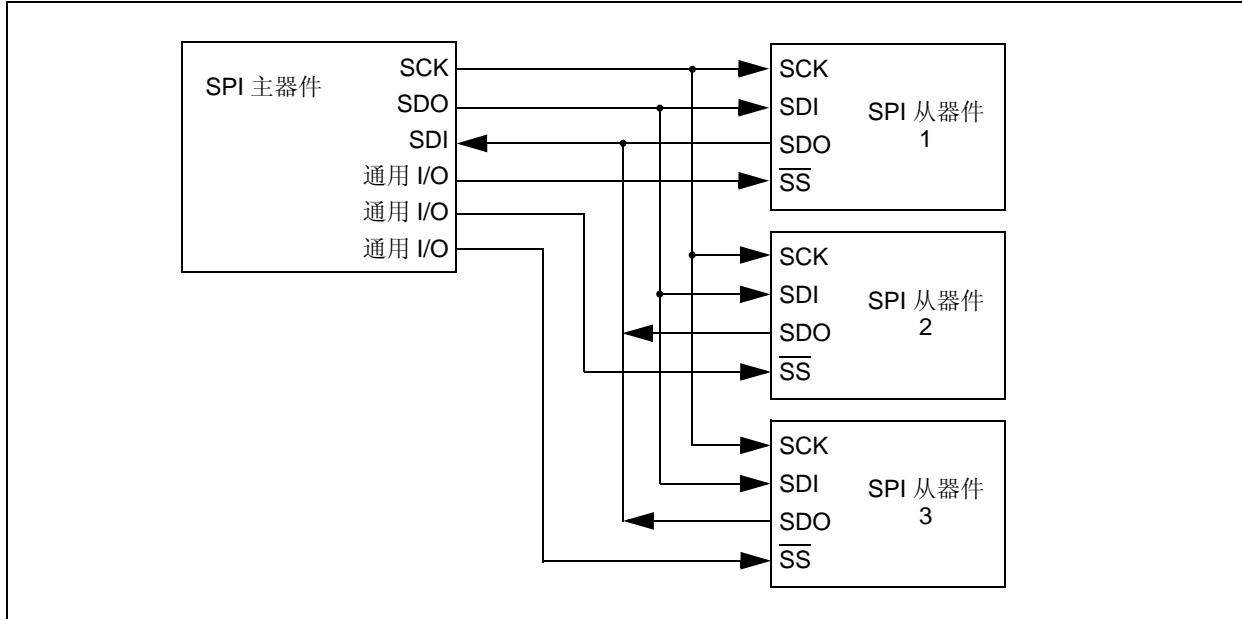
数据是否有意义（无效数据），取决于应用软件。这就导致以下三种数据传输情形：

- 主器件发送有用数据，从器件发送无效数据。
- 主器件发送有用数据，从器件发送有用数据。
- 主器件发送无效数据，从器件发送有用数据。

数据发送可能会需要不定数量的时钟周期。在没有更多数据需要发送时，主器件会停止发送时钟信号，并取消选择从器件。

每个与总线连接、但未被通过其从选择线选择的从器件，都必须忽略时钟和数据发送信号，并且不能发送自己的任何数据。

图 32-4: SPI 主器件和多个从器件连接



32.2.1 SPI 模式寄存器

MSSP 模块有 6 个寄存器用于 SPI 工作模式。它们包括

- MSSP 状态寄存器 (SSPxSTAT)
- MSSP 控制寄存器 1 (SSPxCON1)
- MSSP 控制寄存器 3 (SSPxCON3)
- MSSP 数据缓冲寄存器 (SSPxBUF)
- MSSP 地址寄存器 (SSPxADD)
- MSSP 移位寄存器 (SSPSR) (不可直接访问)

SSPxCON1 和 SSPxSTAT 是 SPI 工作模式下的控制寄存器和状态寄存器。SSPxCON1 寄存器是可读写的。SSPxSTAT 的低 6 位是只读的，SSPxSTAT 的高 2 位是可读写的。

在一种 SPI 主模式下，SSPxADD 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 32.7 节“波特率发生器”。

SSPSR 是用来将数据移入和移出的移位寄存器。

SSPxBUF 用于间接访问 SSPSR 寄存器。SSPxBUF 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，SSPSR 和 SSPxBUF 共同构成一个缓冲接收器。当 SSPSR 接收到一个完整字节时，字节会被传输到 SSPxBUF 中，并且 SSPxIF 中断标志会置 1。

在发送期间，SSPxBUF 不是可缓冲的。对 SSPxBUF 的写操作将同时写入 SSPxBUF 和 SSPSR。

32.2.2 SPI 模式操作

初始化 SPI 时需要指定几个选项。可以通过编程相应的控制位 (SSPxCON1<5:0> 和 SSPxSTAT<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)

要使能串口，SSPxCON1 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 SSPxCONx 寄存器，然后再将 SSPEN 置 1。这会将 SDI、SDO、SCK 和 SS 引脚配置为串口引脚。要让上述引脚充当串口，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

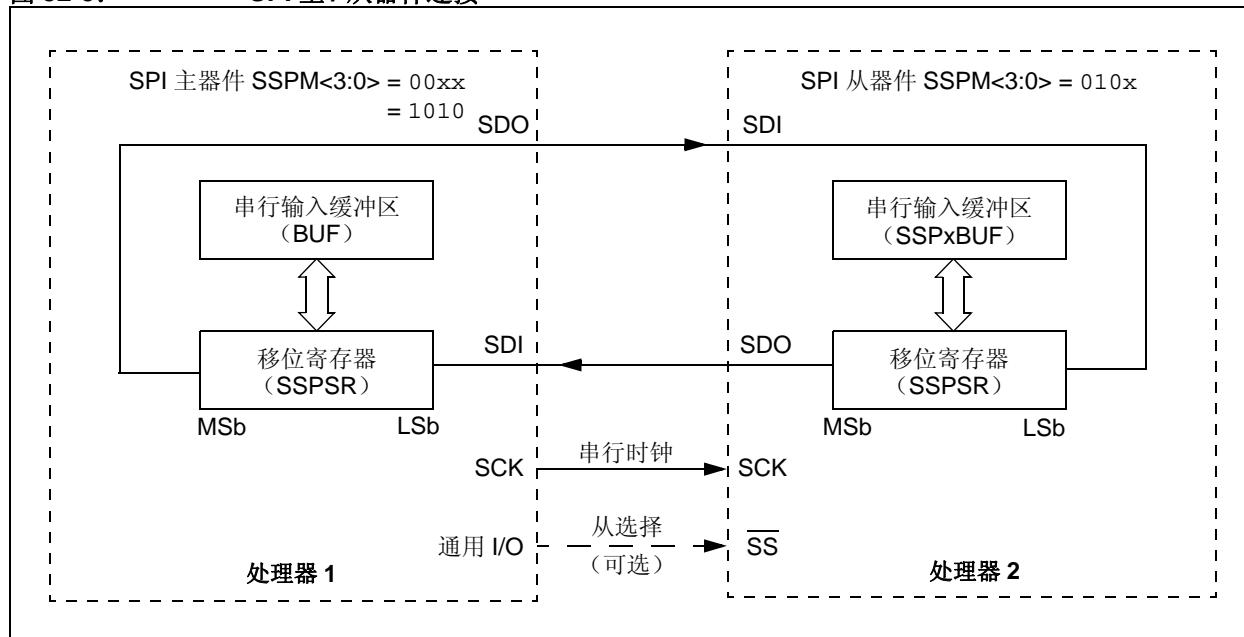
对于不需要的串口功能，可通过将相应的数据方向 (TRIS) 寄存器编程为相反值来改写。

MSSP 模块由一个发送 / 接收移位寄存器 (SSPSR) 和一个缓冲寄存器 (SSPxBUF) 组成。SSPSR 将数据移入 / 移出器件，先移位 MSb。SSPxBUF 会保存先前写入 SSPSR 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 SSPx BUF 寄存器。然后，SSPxSTAT 寄存器的缓冲区满检测位 BF 和中断标志位 SSPxIF 被置 1。这种双重缓冲数据接收方式 (SSPxBUF) 允许在读取刚接收的数据之前就开始接收下一个字节。当 SSPxBUF 寄存器正在发送 / 接收数据时，对它写入的任何数据都将被忽略，同时 SSPxCON1 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 SSPxBUF 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 SSPxBUF 之前，读取 SSPxBUF 中现有的数据。SSPxSTAT 寄存器的缓冲区满位 BF 用于指示何时 SSPxBUF 装入了接收到的数据（发送完成）。SSPxBUF 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 中断用于检测发送 / 接收何时结束。如果不打算使用中断方法，用软件查询的方法同样可确保不会发生写冲突。

SSPSR 不能直接读写，只能通过寻址 SSPxBUF 寄存器来进行访问。此外，SSPxSTAT 寄存器用于指示各种状态条件。

图 32-5：SPI 主 / 从器件连接



32.2.3 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 32-5 中的处理器 2）在何时广播数据。

在主模式下，数据一写入 SSPxBUF 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）。SSPSR 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 SSPxBUF 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 SSPxCON1 寄存器的 CKP 位和 SSPxSTAT 寄存器的 CKE 位，可以选择时钟极性。图 32-6、图 32-8、图 32-9 和图 32-10 给出了 SPI 通信的波形图，其中 MSb 先发送。在主模式下，SPI 时钟速率（比特率）可由用户编程为以下几种之一：

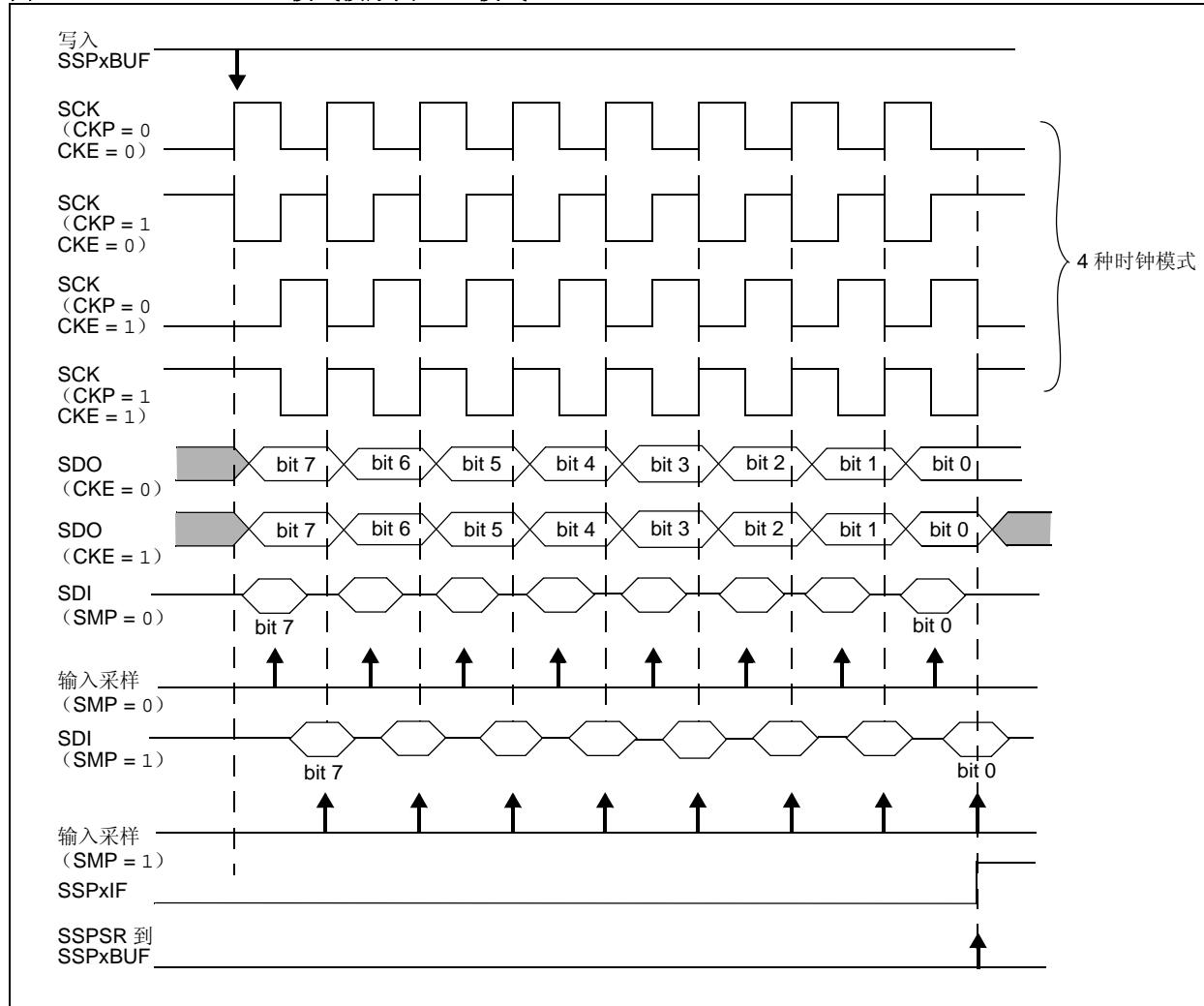
- Fosc/4（或 T_{CY}）
- Fosc/16（或 4 * T_{CY}）
- Fosc/64（或 16 * T_{CY}）
- Timer2 输出 /2
- Fosc/(4 * (SSPxADD + 1))

图 32-6 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 SSPxBUF 的时刻。

注： 在主模式下，送至 SCK 引脚的时钟信号输出也是送至外设的时钟信号输入。通过 RxyPPS 寄存器选择作为输出的引脚也必须通过 SSPCLKPPS 寄存器选择作为外设输入。

图 32-6: SPI 模式波形图 (主模式)



32.2.4 SPI 从模式

在从模式下，当 SCK 上出现外部时钟脉冲时发送和接收数据。锁存最后一位数据之后，SSPxIF 中断标志位会置 1。

在 SPI 从模式下使能该模块前，时钟线必须处于相应的空闲状态。时钟线可通过读 SCK 引脚来查看。空闲状态由 SSPxCON1 寄存器的 CKP 位决定。

在从模式下，外部时钟由 SCK 引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。

在休眠模式下，从器件仍可发送 / 接收数据。移位寄存器通过 SCK 引脚输入提供时钟，当接收到一个字节时，器件会产生中断。如果允许发生中断，器件会从休眠模式唤醒。

32.2.4.1 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。整个链充当一个很大的通信移位寄存器。菊花链功能只需要从主器件引出一条从选择线。

图 32-7 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 SSPxCON3 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 SSPxBUF 寄存器。这使软件可以忽略不适用于它的数据。

32.2.5 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。

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($SSPxCON1<3:0> = 0100$)。

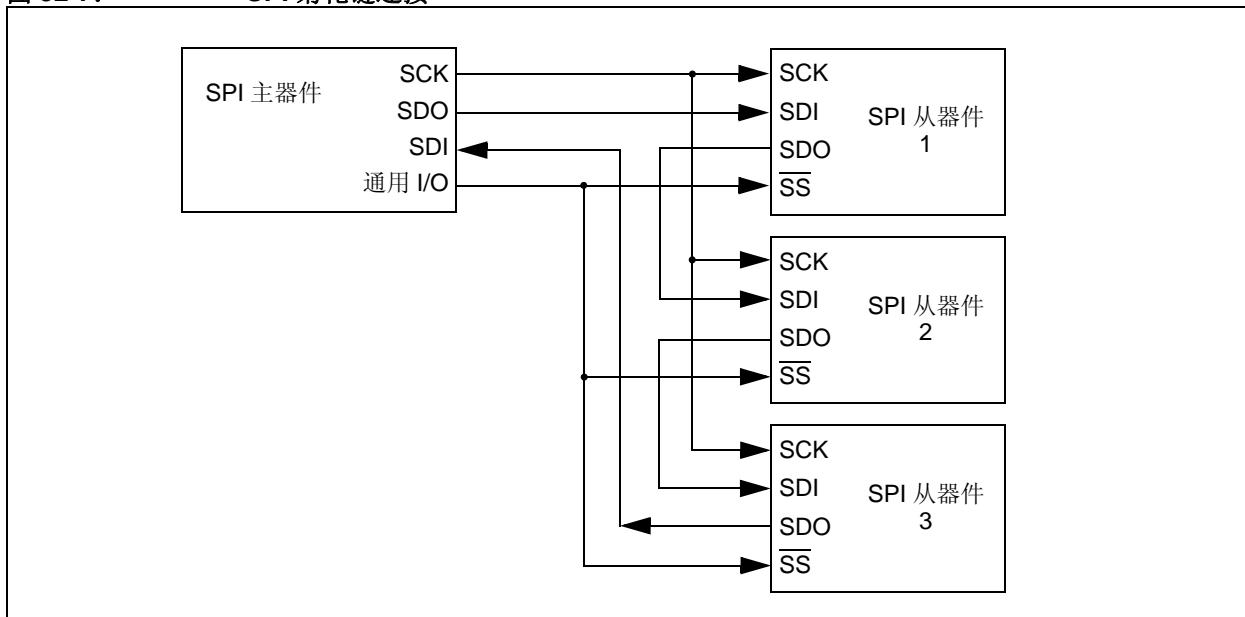
当SS引脚为低电平时，使能数据的发送和接收，同时驱动SDO引脚。

当SS引脚变为高电平时，即使是在字节的发送过程中，也不再驱动SDO引脚，而是将其变成悬空输出状态。根据具体应用，可能需要使用外部上拉/下拉电阻。

- 注 1:** 当 SPI 处于从模式且使能 SS 引脚控制 ($SSPxCON1<3:0> = 0100$) 时，如果 SS 引脚设置为 VDD，SPI 模块将会复位。
- 2:** 当 SPI 用于从模式且 CKE 置 1 时，用户必须使能 SS 引脚控制。
- 3:** 工作于 SPI 从模式时，SSPxSTAT 寄存器的 SMP 位必须保持清零。

当 SPI 模块复位时，位计数器被强制为 0。这通过强制将 SS 引脚拉为高电平或将 SSPEN 位清零来实现。

图 32-7: SPI 菊花链连接



PIC16(L)F1764/5/8/9

图 32-8：从选择同步波形图

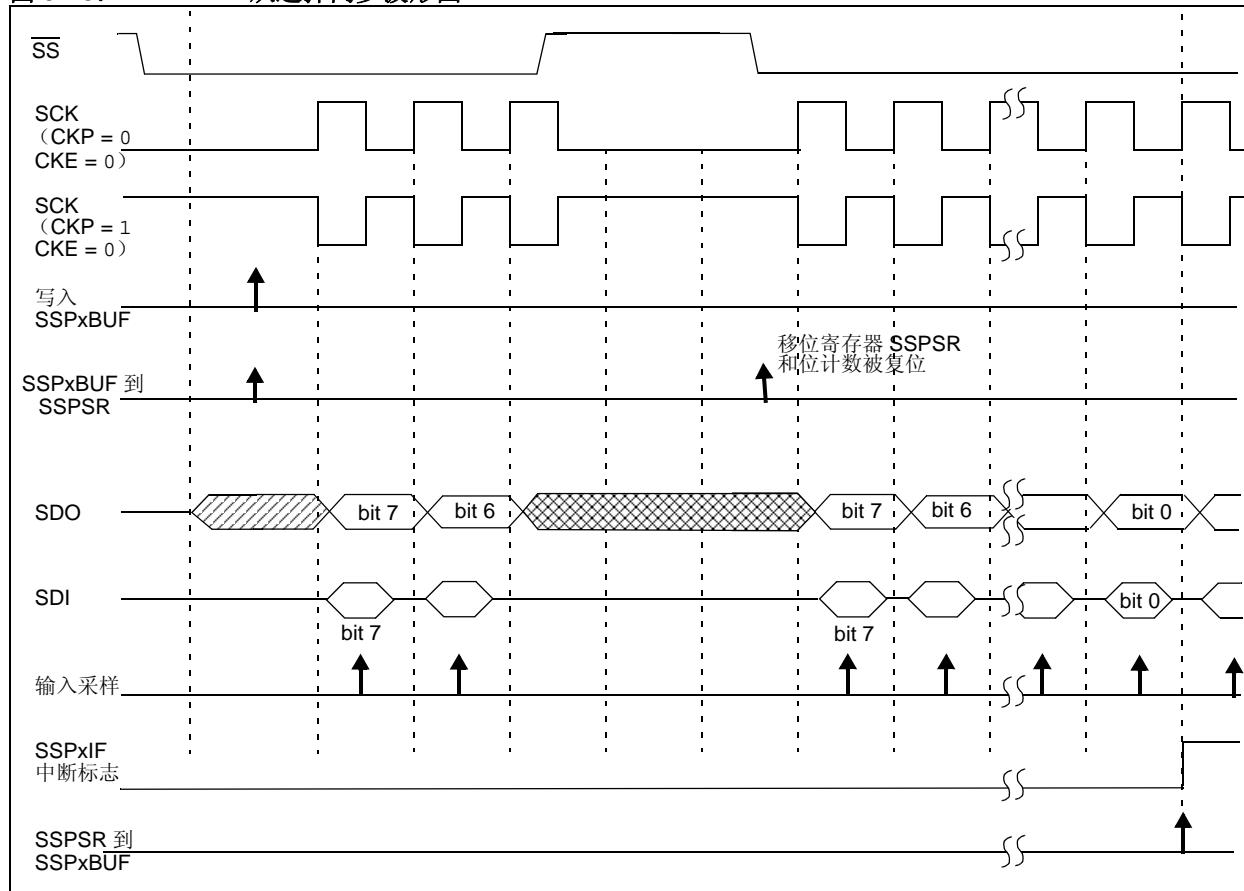


图 32-9: SPI 模式波形图 (从模式, CKE = 0)

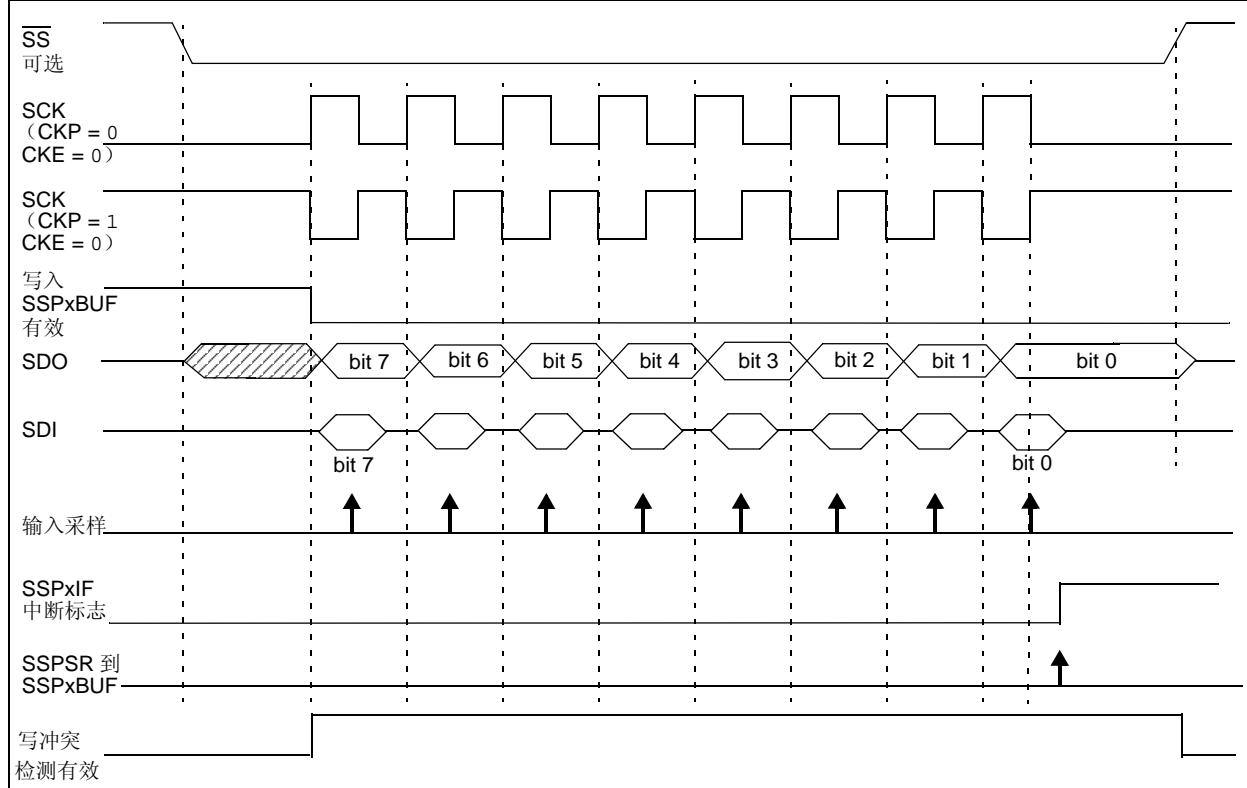
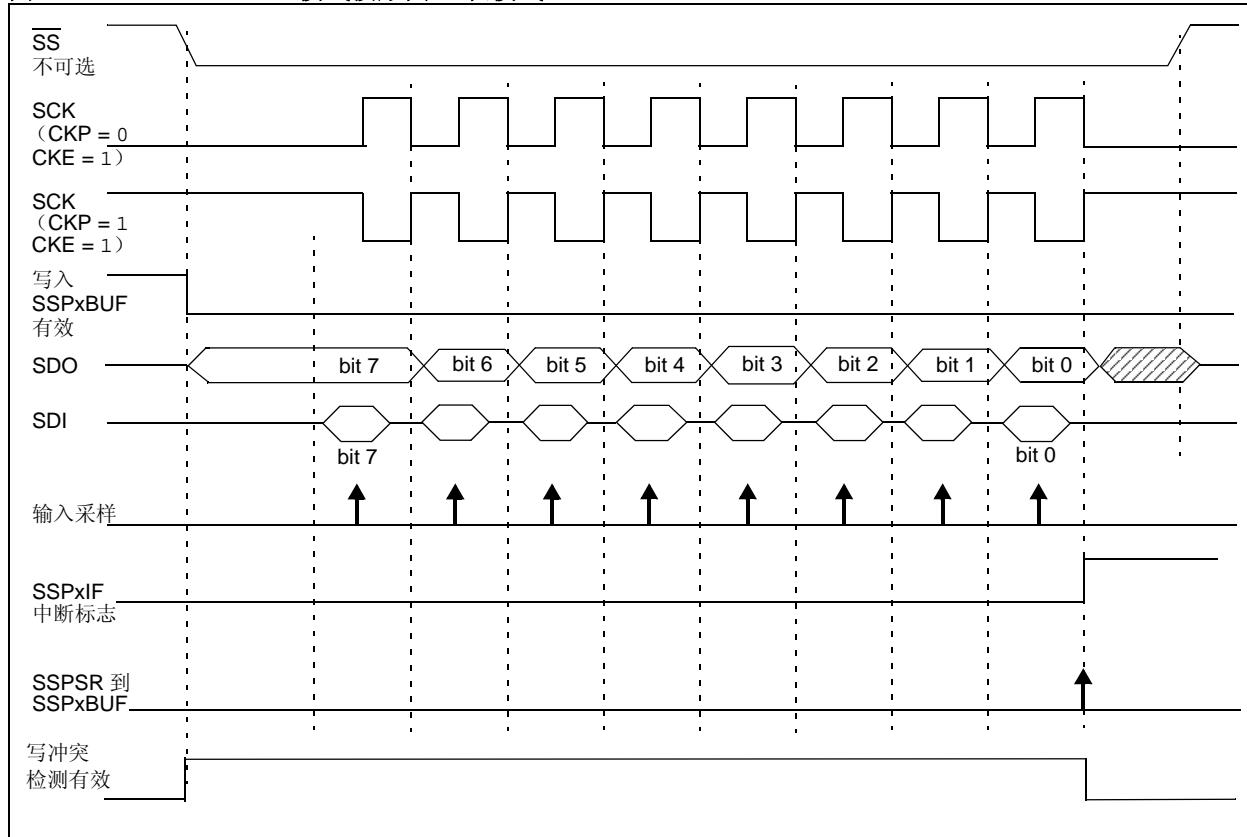


图 32-10: SPI 模式波形图 (从模式, CKE = 1)



32.2.6 休眠模式下的 SPI 操作

在 SPI 主模式下，模块时钟速度可能与全功耗模式下的不同；处于休眠模式时，所有时钟都暂停。

在 MSSP 时钟速度远高于系统时钟时，用户需要特别小心。

在从模式下，当允许 MSSP 中断时，在主器件发送完数据时，MSSP 中断会将控制器从休眠模式唤醒。

如果不从休眠模式退出，应该禁止 MSSP 中断。

在 SPI 主模式下，当选择休眠模式时，所有模块的时钟都将暂停，并且在器件被唤醒前，发送 / 接收将保持此暂停状态。器件返回到运行模式之后，模块将恢复发送和接收数据。

在 SPI 从模式下，SPI 发送 / 接收移位寄存器与器件异步工作。这可使器件置于休眠模式，仍能将数据移入 SPI 发送 / 接收移位寄存器。当接收到全部 8 位数据时，MSSP 中断标志位将置 1，并且如果允许中断的话，将唤醒器件。

表 32-1：与 SPI 操作相关的寄存器汇总

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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSEL0	ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RxyPPS	—	—	—	RxyPPS<4:0>					158
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					158, 160
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					158, 160
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					158, 160
SSP1BUF	同步串行端口接收缓冲区 / 发送寄存器							379*	
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				424
SSP1CON3	ACKTMR	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	423
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	423
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽²⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽²⁾	TRISC6 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	150

图注：— = 未实现位，读为 0。SPI 模式下的 MSSP 不使用阴影单元。

* 提供寄存器信息的页。

注 1：未实现位，读为 1。

2：仅限 PIC16(L)F1768/9。

32.3 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 32-11 给出了 MSSP 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，信号视为逻辑 0；线路保持悬空时，信号视为逻辑 1。

图 32-11 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：

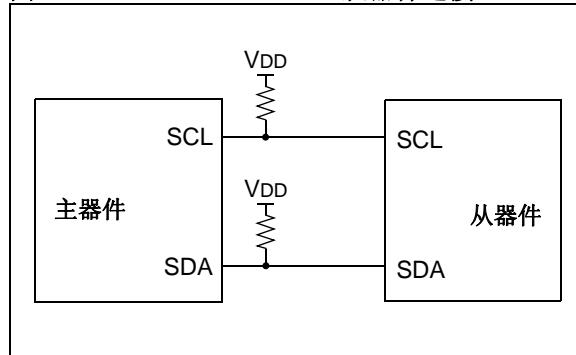
- 主发送模式
(主器件向从器件发送数据)
- 主接收模式
(主器件从从器件接收数据)
- 从发送模式
(从器件向主器件发送数据)
- 从接收模式
(从器件从主器件接收数据)

要开始进行通信，主器件需要以主发送模式启动。主器件发送出启动位，后面跟随它希望进行通信的从器件的地址字节。后面再跟随单个读 / 写位，该位决定主器件是向从器件发送数据还是从从器件接收数据。

如果总线上存在所请求的从器件，从器件会使用应答位（也称为 ACK）进行响应。然后，主器件会以发送模式或接收模式继续通信，从器件则以互补模式（分别为接收模式或发送模式）继续通信。

启动位由 SCL 线保持为高电平时 SDA 线的由高至低跳变来指示。地址和数据字节随后送出，先发送最高有效位 (MSb)。在主器件希望从从器件读取数据时，发送出的读 / 写位为逻辑 1，在主器件希望向从器件写入数据时，该位为逻辑 0。

图 32-11: I²C 主 / 从器件连接



应答位 (ACK) 是低电平有效信号，它会将 SDA 线保持为低电平，用于指示发送器，从器件已接收到发送数据，并已准备好接收更多数据。

数据位的跳变总是在 SCL 线保持低电平时执行。在 SCL 线保持高电平时发生的跳变用于指示启动位和停止位。

如果主器件希望向从器件写入数据，则它会重复发送一个字节的数据，而从器件则在接收每个字节之后使用 ACK 位进行响应。在该示例中，主器件处于主发送模式，从器件处于从接收模式。

如果主器件希望从从器件读取数据，则它会从从器件重复接收一个字节的数据，并在接收每个字节之后使用 ACK 位进行响应。在该示例中，主器件处于主接收模式，从器件处于从发送模式。

在传输最后一个数据字节之后，主器件可以通过发送停止位来结束数据发送。如果主器件处于接收模式，它会发送停止位来代替最后一个 ACK 位。停止位由 SCL 线保持为高电平时 SDA 线的由低至高跳变来指示。

在某些情况下，主器件可能希望维持对总线的控制，并重新启动另一次数据发送。如果是这样，主器件可以在它处于接收模式时，发送另一个启动位来代替停止位或最后一个 ACK 位。

I²C 总线规定了三种报文协议：

- 主器件向从器件写数据的单一报文。
- 主器件从从器件读数据的单一报文。
- 主器件对一个或多个从器件启动至少两次写操作或读操作，或者读写操作组合的组合报文。

在一个器件发送逻辑 1（或将线路保留悬空），第二个器件发送逻辑 0（或将线路保持为低电平）时，第一个器件会检测到线路不为逻辑 1。这种检测在用于 **SCL** 线时，称为时钟延长。时钟延长为从器件提供了一种控制数据流的机制。这种检测在用于 **SDA** 线时，称为仲裁。仲裁可以确保任意时刻只有一个主器件在进行通信。

32.3.1 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上拉 **SCL** 线，以传输下一位数据，但它会检测到时钟线尚未被释放。由于 **SCL** 连接是漏极开路，所以从器件可以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。

32.3.2 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第一个发送器会在仲裁中失败，必须停止在 **SDA** 线上发送数据。

例如，如果一个发送器将 **SDA** 线保持为逻辑 1（保留悬空），而第二个发送器将它保持为逻辑 0（下拉为低电平），则结果是 **SDA** 线将为低电平。那么，第一个发送器会发现线路电平与期望电平不同，并断定有另一个发送器正在进行通信。

发现电平不同的第一个发送器将是仲裁失败的发送器，必须停止驱动 **SDA** 线。如果该发送器同时也是主器件，则它还必须停止驱动 **SCL** 线。然后，它可以在尝试重新启动数据发送之前监视线路上是否出现停止条件。与此同时，另一个未发现期望电平与 **SDA** 线实际电平不同的器件将继续原来的数据发送。它可以无需进行任何复杂处理，因为到目前为止，发送条件与所期望的完全相同，没有其他发送器对报文产生干扰。

当主器件对多个从器件进行寻址时，也会对从发送模式进行仲裁，但这种情况较少见。

如果有两个主器件在地址阶段向两个不同的从器件发送报文，则发送较小从器件地址的主器件总是会在仲裁中获胜。当两个主器件向同一从器件地址发送报文时，地址有时会指向多个从器件，仲裁过程必须继续进入到数据阶段。

仲裁通常极少发生，但它是正确支持多主器件所必需的过程。

32.4 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。有 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。

32.4.1 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。

32.4.2 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。

32.4.3 SDA 和 SCL 引脚

在 SSPEN 位置 1 的情况下选择任意 I²C 模式时，SCL 和 SDA 引脚将会强制设为漏极开路。用户应通过将相应的 TRIS 位置 1 来将这些引脚设置为输入。

- 注 1：**在使能 I²C 模式时，数据将设为输出 0。
- 2：**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。

32.4.4 SDA 保持时间

SDA 引脚的保持时间通过 SSPxCON3 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 32-2: 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	送入从器件中、并与 SSPxADD 中的存储值匹配的地址字节。
写请求	从器件接收到 R/W 位清零的匹配地址，并已准备好移入数据。
读请求	主器件发送 R/W 位置 1 的地址字节，表示要求从器件在时钟控制下将数据移出。从器件在接收到该地址字节后会立即移出所有数据字节，直到发生重复启动或停止条件。
时钟延长	总线上的器件通过将 SCL 保持为低电平来暂停通信的时间。
总线冲突	每当模块进行输出并期望 SDA 线为高电平，却采样到 SDA 线为低电平时。

32.4.5 启动条件

I^2C 规范将启动条件定义为在 SCL 线为高电平时, SDA 从高电平变为低电平状态。启动条件总是由主器件产生, 指示总线从空闲状态变为有效状态。图 32-12 给出了启动和停止条件的波形图。

如果模块在将 SDA 线置为低电平之前采样到 SDA 线为低电平, 则会在产生启动条件时发生总线冲突。这一点不符合 I^2C 规范, 该规范规定启动时不能发生总线冲突。

32.4.6 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。

图 32-12: I^2C 启动和停止条件

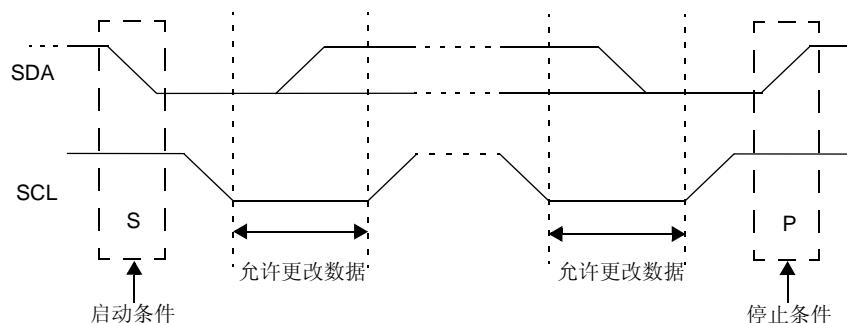
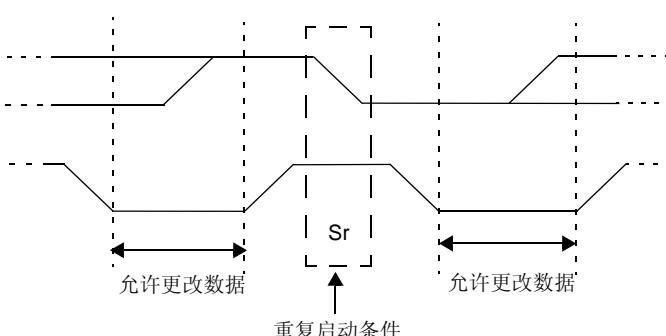


图 32-13: I^2C 重复启动条件



32.4.7 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 32-13 给出了重复启动条件的波形图。

在 10 位寻址从模式下, 要从寻址到的从器件中移出数据, 主器件需要产生重复启动条件。从器件完全寻址(高地址字节和低地址字节均匹配)之后, 主器件可以发出重复启动条件和 R/W 位置 1 的高地址字节。然后, 从器件逻辑会保持时钟, 并准备送出数据。

在 10 位模式下, 与 R/W 清零的地址字节完全匹配之后, 前一次匹配标志会置 1 并保持置 1, 直到出现停止条件、出现 R/W 清零的高地址字节或发生高地址字节匹配失败为止。

32.4.8 启动 / 停止条件中断屏蔽

SSPxCON3 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。

32.4.9 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。

ACK 的结果会被放入 SSPxCON2 寄存器的 ACKSTAT 位中。

当 AHEN 和 DHEN 位置 1 时，从器件软件允许用户设置要回送到发送器的 ACK 值。用户可以通过置 1/ 清零 SSPxCON2 寄存器的 ACKDT 位来决定响应。

如果 SSPxCON3 寄存器的 AHEN 和 DHEN 位清零，从器件硬件会产生 ACK 响应。

有一些条件下，从器件不会发送 ACK。如果在接收到数据字节时，SSPxSTAT 寄存器的 BF 位或 SSPxCON1 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，SSPxCON3 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。

32.5 I²C 从模式操作

MSSP 从模式可以在 4 种模式下工作，这些模式通过 SSPxCON1 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 SSPxIF 置 1。

32.5.1 从模式地址

SSPxADD 寄存器（[寄存器 32-6](#)）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 SSPxBUF 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 32-5](#)）会影响地址匹配过程。更多信息，请参见 [第 32.5.8 节 “SSP 掩码寄存器”](#)。

32.5.1.1 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。

32.5.1.2 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 SSPxADD 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 SSPxADD 为止。在低地址字节送入之后，全部 8 位将与 SSPxADD 中的低地址值进行比较。即使地址不匹配，SSPxIF 和 UA 也会置 1，SCL 会保持低电平，直到 SSPxADD 发生更新可再次接收高字节为止。当 SSPxADD 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。

32.5.2 从接收

当接收到的匹配地址字节的 $\overline{R/W}$ 位清零时，SSPxSTAT 寄存器的 $\overline{R/W}$ 位会清零。接收到的地址被装入 SSPxBUF 寄存器并产生应答。

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 SSPxSTAT 寄存器的 BF 位被置 1，或 SSPxCON1 寄存器的 SSPOV 位被置 1。SSPxCON3 寄存器的 BOEN 位会修改该操作。更多信息，请参见[寄存器 32-4](#)。

每个传输的数据字节都会产生 MSSP 中断。标志位 SSPxIF 必须用软件清零。

当 SSPxCON2 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 SSPxCON1 寄存器的 CKP 位置 1 来释放时钟，10 位模式下的特殊情况除外。更多详细信息，请参见[第 32.5.6.2 节“10 位寻址模式”](#)。

32.5.2.1 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 MSSP 模块的标准事件序列。[图 32-14](#) 和 [图 32-15](#) 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. SSPxSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPxIF 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 SSPxIF 置 1。
5. 用软件清零 SSPxIF 位。
6. 软件从 SSPxBUF 中读取接收的地址，使 BF 标志清零。
7. 如果 SEN = 1，从器件软件会通过将 CKP 位置 1 来释放 SCL 线。
8. 主器件送出数据字节。
9. 从器件通过将 SDA 驱动为低电平而向主器件发送 ACK，并将 SSPxIF 置 1。
10. 用软件清零 SSPxIF。
11. 软件从 SSPxBUF 中读取接收的字节，使 BF 清零。
12. 对于从主器件接收到的所有字节重复步骤 8-12。
13. 主器件发送停止条件，将 SSPxSTAT 的 P 位置 1，总线变为空闲状态。

32.5.2.2 使用 AHEN 和 DHEN 时的 7 位接收

在 AHEN 和 DHEN 置 1 时，从器件接收的工作方式与不使用这些选项时的工作方式相同，只是在 SCL 的第 8 个下降沿之后添加了额外的中断和时钟延长。这些额外中断允许从器件软件决定是否应答（ACK）接收的地址或数据字节，而不是由硬件决定。该功能增加了对 PMBus™ 的支持，先前版本的该模块不支持这一功能。

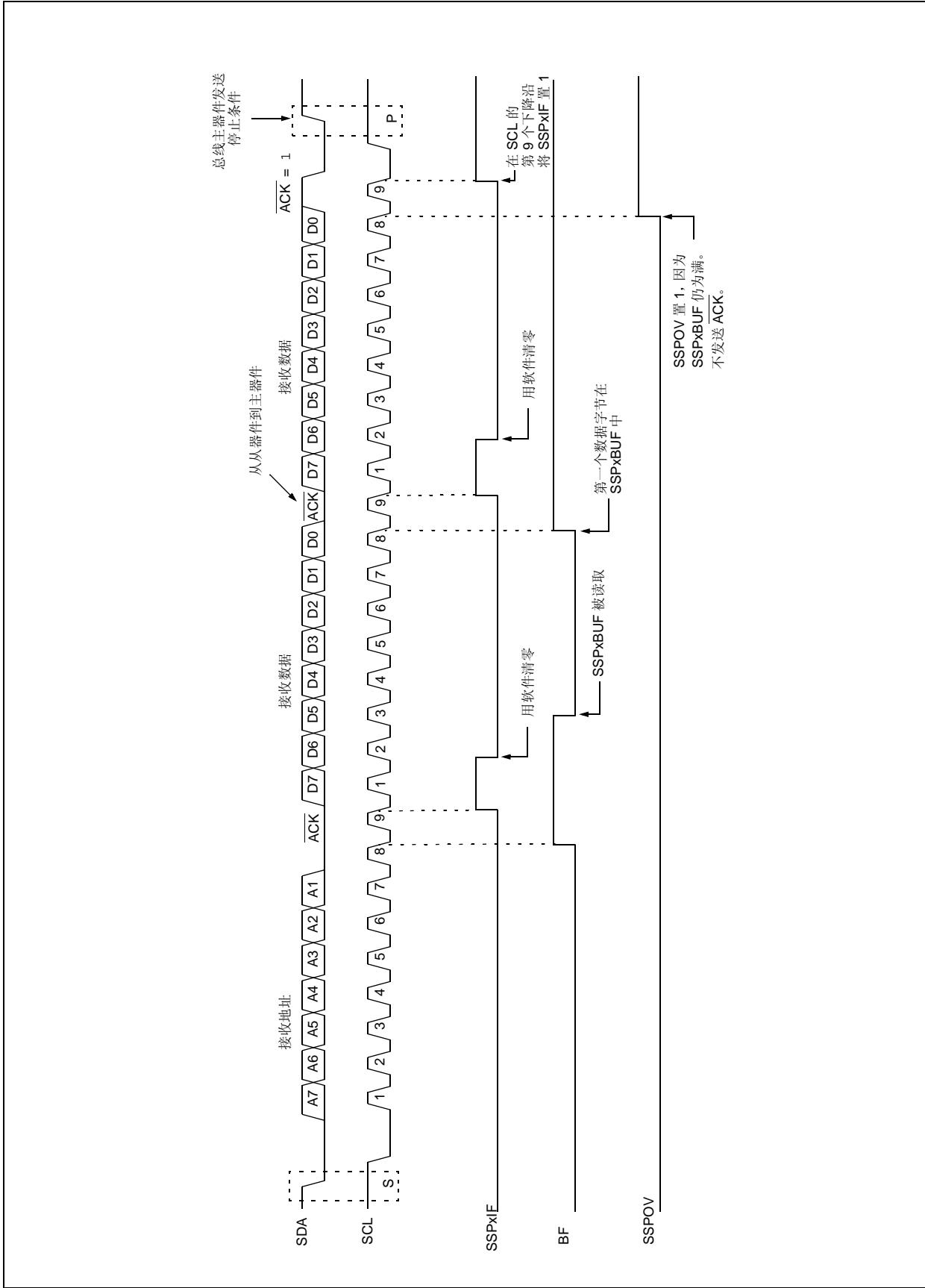
以下列表介绍了要对 I²C 通信使用这些选项时，从器件软件需要执行的步骤。[图 32-16](#) 显示了同时使用地址和数据保持功能的模块。[图 32-17](#) 包含了 SSPxCON2 寄存器的 SEN 位置 1 时的操作。

1. SSPxSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPxIF 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，SSPxIF 置 1，CKP 清零。
3. 从器件清零 SSPxIF。
4. 从器件可以查看 SSPxCON3 寄存器的 ACKTIM 位，以确定 SSPxIF 是在 ACK 之前还是之后置 1。
5. 从器件从 SSPxBUF 中读取地址值，使 BF 标志清零。
6. 从器件通过设置 ACKDT 来设置要送到主器件的 ACK 值。
7. 从器件通过将 CKP 置 1 来释放时钟。
8. SSPxIF 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 SSPxIF。

注： 即使不进行时钟延长，且 BF 已清零，SSPxIF 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，SSPxIF 才不会置 1。

11. 在所接收数据字节的第 8 个 SCL 下降沿之后，SSPxIF 置 1，CKP 清零。
12. 从器件通过查看 SSPxCON3 的 ACKTIM 位来确定中断源。
13. 从器件从 SSPxBUF 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 SSPSTAT 寄存器的 P 位才能检测停止条件。

图 32-14: I²C 从模式, 7 位地址, 接收 (SEN = 0, AHEN = 0, DHEN = 0)



PIC16(L)F1764/5/8/9

图 32-15: I²C 从模式, 7 位地址, 接收 (SEN = 1, AHEN = 0, DHEN = 0)

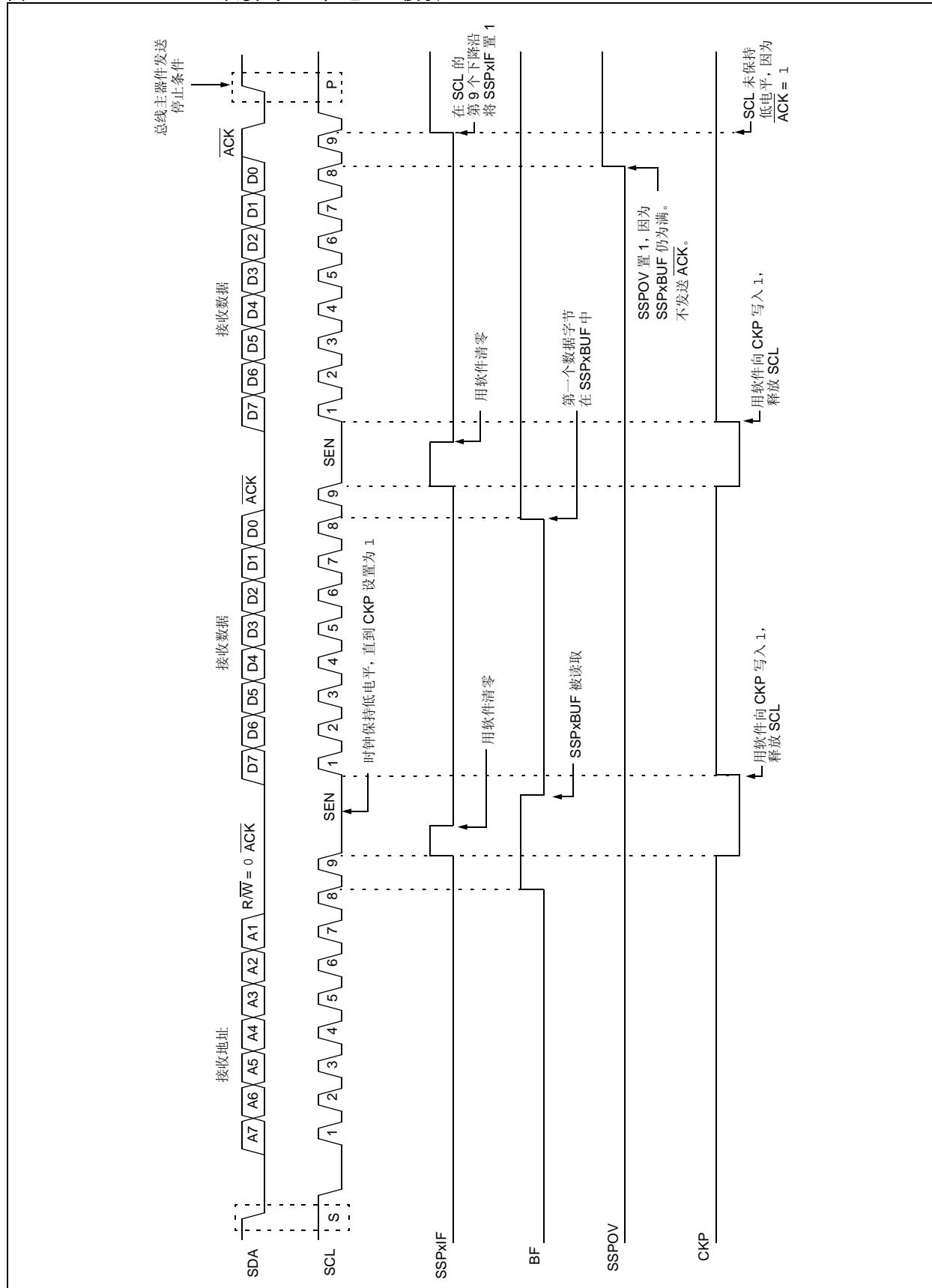


图 32-16: I²C 从模式, 7 位地址, 接收 (SEN = 0, AHEN = 1, DHEN = 1)

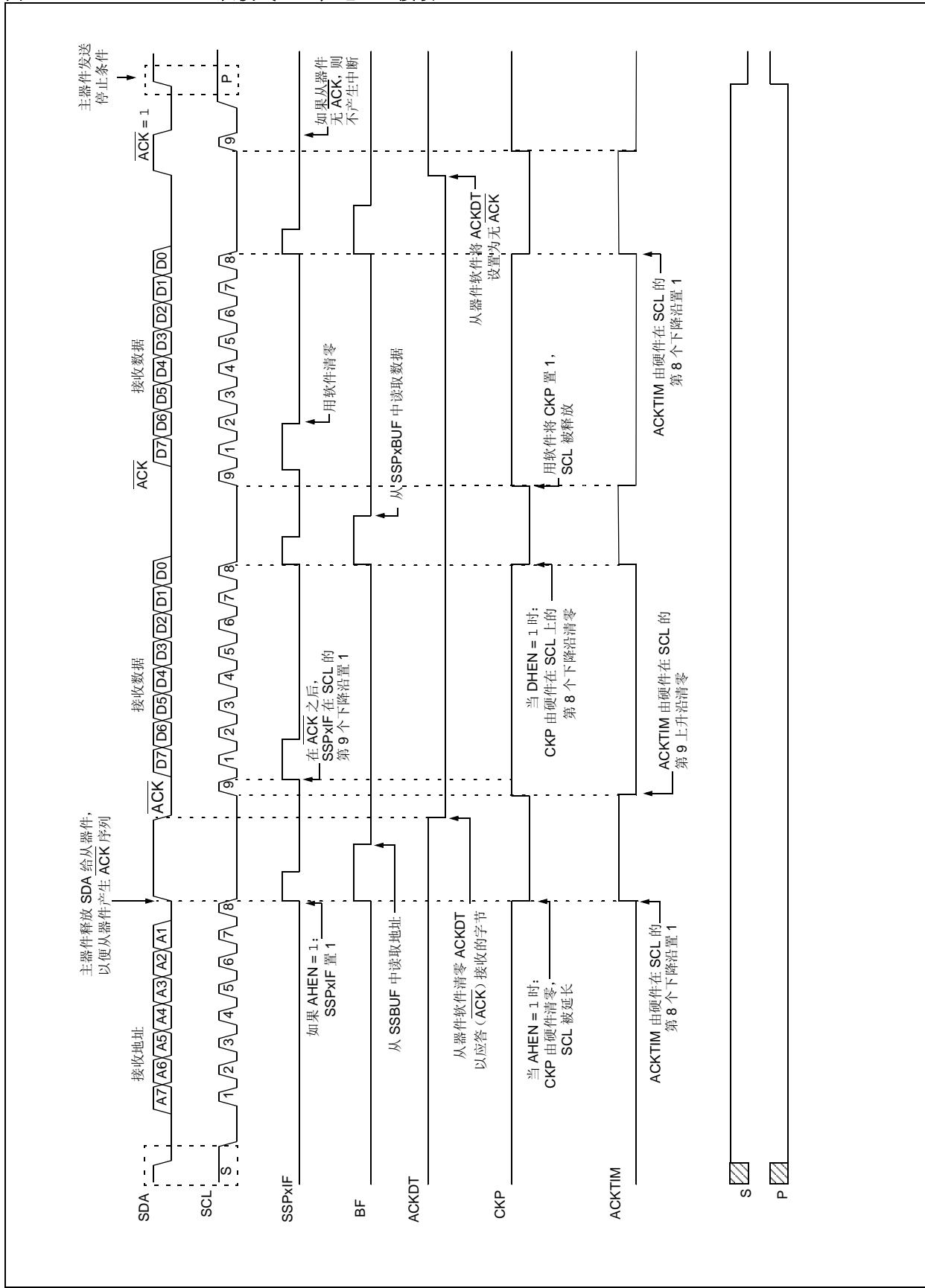
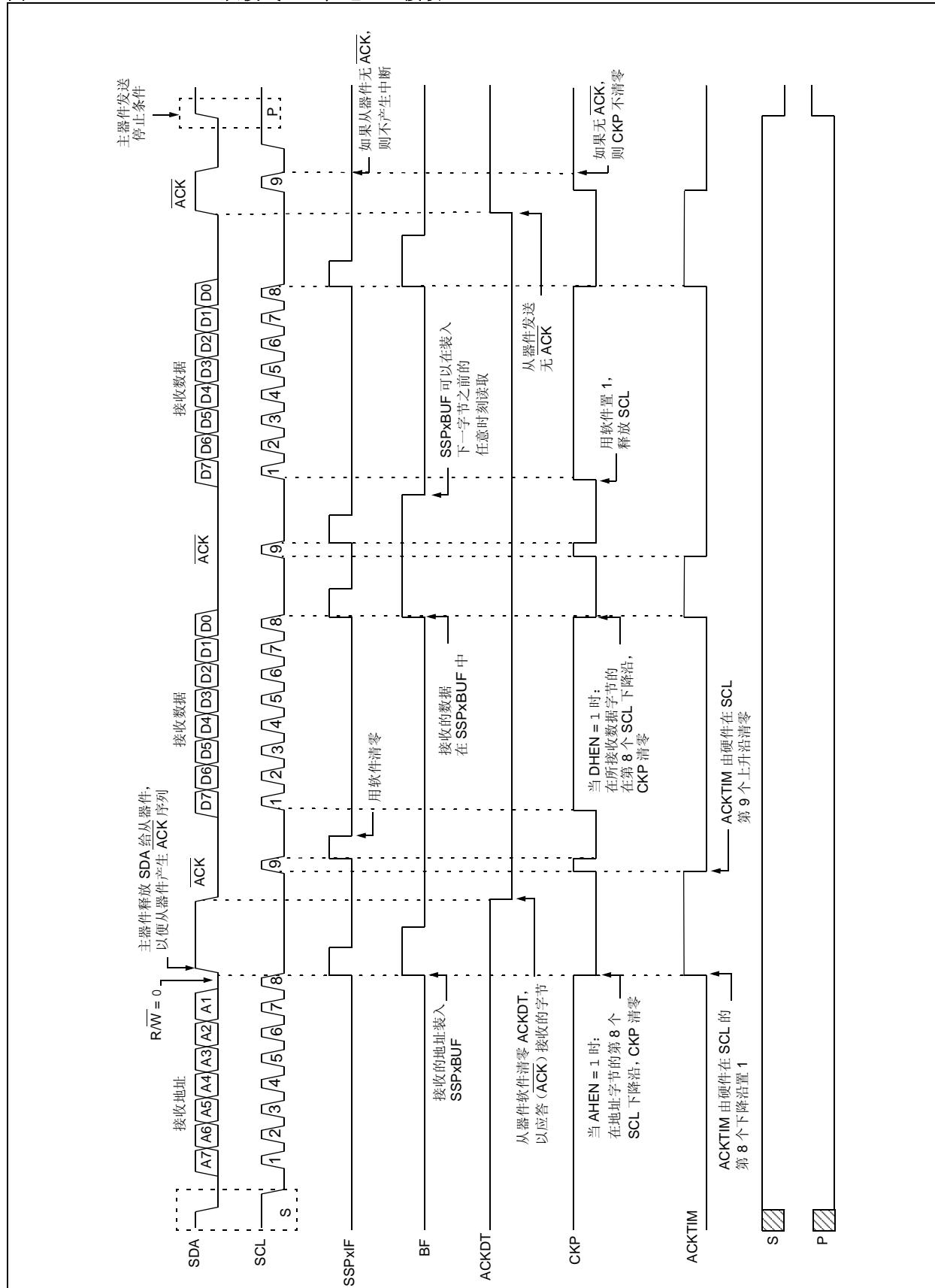


图 32-17: I²C 从模式, 7 位地址, 接收 (SEN = 1, AHEN = 1, DHEN = 1)



32.5.3 从发送

当输入地址字节的 R/W 位置 1 并发生地址匹配时，SSPxSTAT 寄存器的 R/W 位被置 1。接收到的地址会被装入 SSPxBUF 寄存器，并且从器件会在第 9 个位发送 ACK 脉冲。

在 ACK 之后，从器件硬件会清零 CKP 位，并且 SCL 引脚保持低电平（更多详细信息，请参见第 32.5.6 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 SSPxBUF 寄存器，此时该数据也会被装入 SSPSR 寄存器。然后，应通过将 SSPxCON1 寄存器的 CKP 位置 1 来释放 SCL 引脚。8 个数据位在 SCL 输入的下降沿被移出。这可确保在 SCL 为高电期间 SDA 信号是有效的。

来自主接收器的 ACK 脉冲将在第 9 个 SCL 输入脉冲的上升沿锁存。该 ACK 值会被复制到 SSPxCON2 寄存器的 ACKSTAT 位中。如果 ACKSTAT 置 1（无 ACK 应答信号），那么表示数据传输已完成。这种情况下，在从器件锁存无 ACK 时，从器件会进入空闲状态，并等待出现另一个启动位。如果 SDA 线为低电平（ACK），则必须将下一个要发送的数据装入 SSPxBUF 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。SSPxIF 位必须用软件清零，SSPxSTAT 寄存器用于确定字节的状态。SSPxIF 位在第 9 个时钟脉冲的下降沿被置 1。

32.5.3.1 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，SSPxCON3 寄存器的 SBCDE 位会置 1，PIR 寄存器的 BCLIF 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 BCLIF 位来处理从器件总线冲突。

32.5.3.2 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 32-18 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. SSPxSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPxIF 会置 1。
3. 从器件接收到 R/W 位置 1 的匹配地址，将 SSPxIF 位置 1。
4. 从器件硬件产生 ACK，并将 SSPxIF 置 1。
5. 用户将 SSPxIF 位清零。
6. 软件从 SSPxBUF 中读取接收的地址，使 BF 清零。
7. R/W 置 1，所以 CKP 会在 ACK 之后自动清零。
8. 从器件软件将发送数据装入 SSPxBUF。
9. CKP 位置 1，释放 SCL，使主器件可以从从器件中移出数据。
10. 来自主器件的 ACK 响应装入 ACKSTAT 寄存器之后，SSPxIF 置 1。
11. SSPxIF 位清零。
12. 从器件软件通过检查 ACKSTAT 位来确定主器件是否要移出更多数据。

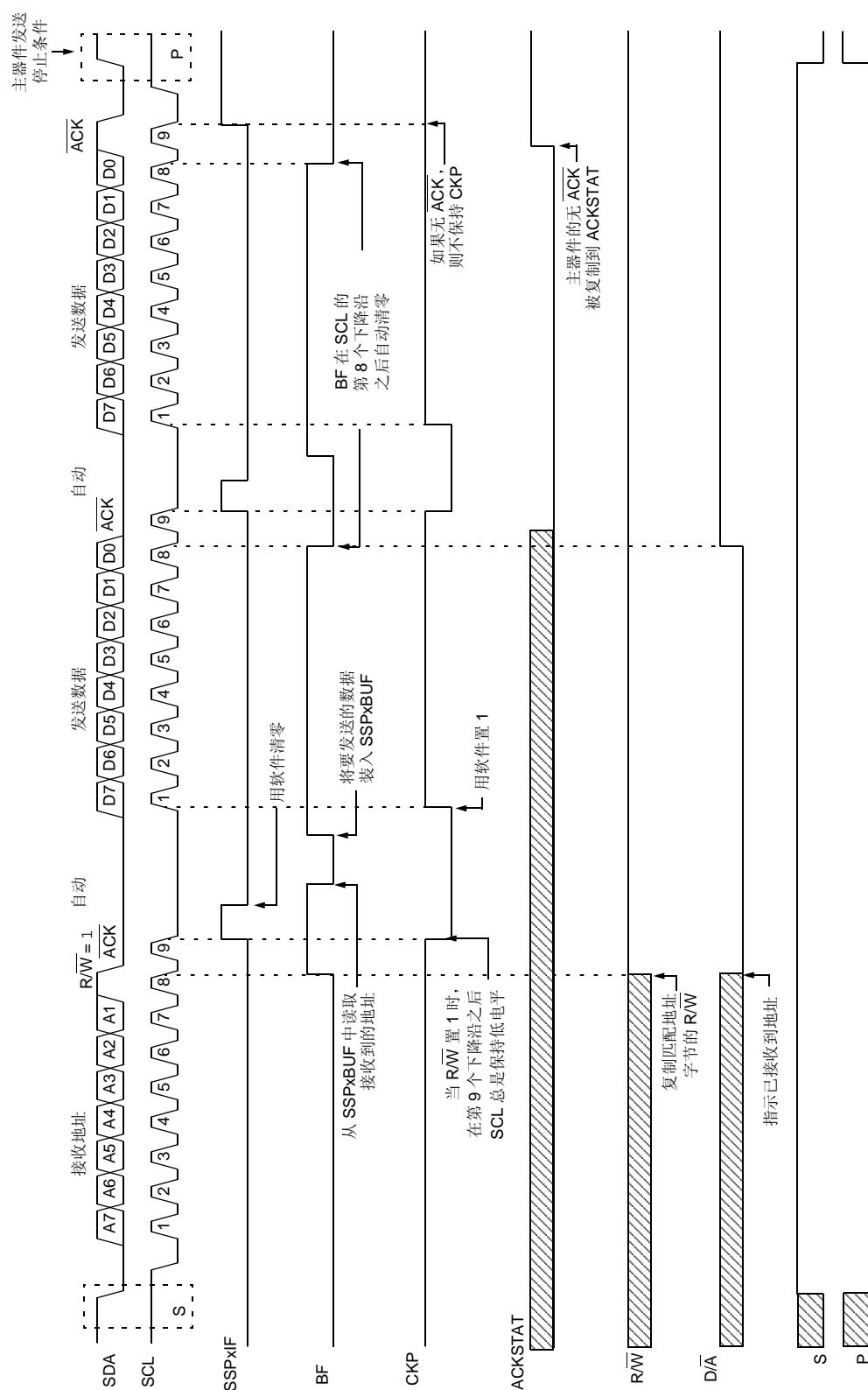
注 1： 如果主器件应答（ACK），时钟将被延长。

2： ACKSTAT 是唯一一个在 SCL 上升沿（第 9 个）而不是下降沿发生更新的位。

13. 对于每个发送字节重复步骤 9-13。
14. 如果主器件发送无 ACK，则不会保持时钟，但 SSPxIF 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。

PIC16(L)F1764/5/8/9

图 32-18: I²C 从模式, 7 位地址, 发送 (AHEN = 0)



32.5.3.3 使能地址保持时的 7 位发送

将 SSPxCON3 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，SSPxIF 中断标志会置 1。

图 32-19 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；SSPxSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPxIF 会置 1。
3. 主器件发送 R/W 位置 1 的匹配地址。在 SCL 线的第 8 个下降沿之后，CKP 位清零，并产生 SSPxIF 中断。
4. 从器件软件清零 SSPxIF。
5. 从器件软件读取 SSPxCON3 寄存器的 ACKTIM 位，以及 SSPxSTAT 寄存器的 R/W 和 D/A 位，以确定中断源。
6. 从器件从 SSPxBUF 寄存器中读取地址值，使 BF 位清零。
7. 从器件软件根据该信息确定它是产生 ACK 还是产生无 ACK，并相应地设置 SSPxCON2 寄存器的 ACKDT 位。
8. 从器件将 CKP 位置 1，释放 SCL。
9. 主器件移入来自从器件的 ACK 值。
10. 如果 R/W 位置 1，则在 ACK 之后，从器件硬件会自动将 CKP 位清零，将 SSPxIF 置 1。
11. 从器件软件清零 SSPxIF。
12. 从器件将要发送给主器件的值装入 SSPxBUF，使 BF 位置 1。

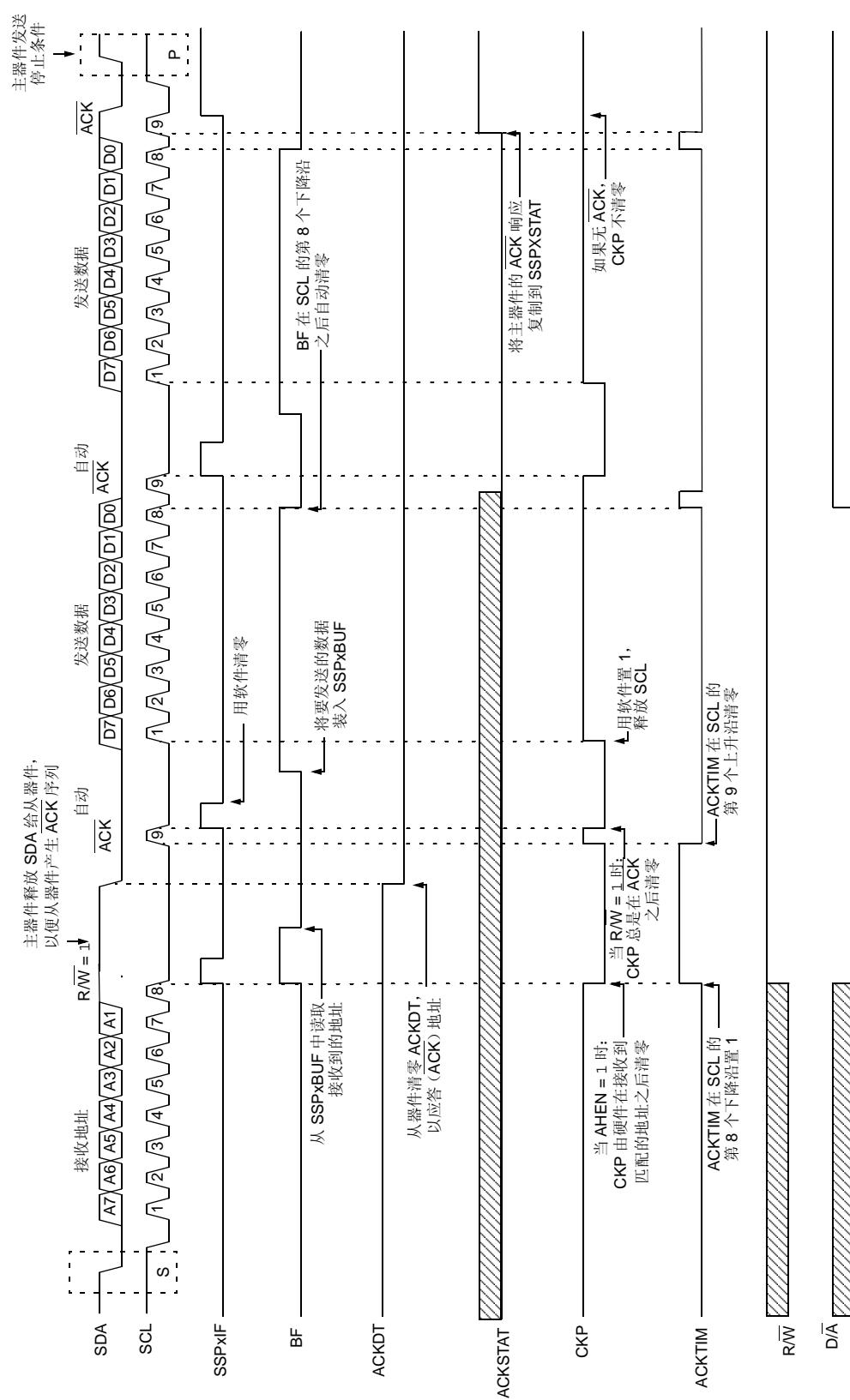
注： 只有在 ACK 之后，才能装入 SSPxBUF。

13. 从器件将 CKP 位置 1，释放时钟。
14. 主器件从从器件中移出数据，并在第 9 个 SCL 脉冲发送 ACK 值。
15. 从器件硬件将 ACK 值复制到 SSPxCON2 寄存器的 ACKSTAT 位中。
16. 对于从从器件发送到主器件的每个字节重复步骤 10-15。
17. 如果主器件发送无 ACK，从器件会释放总线，让主器件可以发送停止条件和结束通信。

注： 主器件必须对于最后一个字节发送无 ACK，以确保从器件释放 SCL 线来接收停止条件。

PIC16(L)F1764/5/8/9

图 32-19: I²C 从模式, 7 位地址, 发送 (AHEN = 1)



32.5.4 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 MSSP 模块的标准事件序列。

图 32-20 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； SSPxSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPxIF 会置 1。
3. 主器件发送 R/W 位清零的匹配高地址； SSPxSTAT 寄存器的 UA 位置 1。
4. 从器件发送 ACK， SSPxIF 置 1。
5. 用软件清零 SSPxIF 位。
6. 软件从 SSPxBUF 中读取接收的地址，使 BF 标志清零。
7. 从器件将低地址装入 SSPxADD，释放 SCL。
8. 主器件向从器件发送匹配的低地址字节； UA 位置 1。

注： 只有在 ACK 序列之后，才允许更新 SSPxADD 寄存器。

9. 从器件发送 ACK， SSPxIF 置 1。

注： 如果低地址不匹配，SSPxIF 和 UA 仍然会置 1，从而让从器件软件可以将 SSPxADD 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 SSPxIF。
11. 从器件从 SSPxBUF 中读取接收的匹配地址，使 BF 清零。
12. 从器件将高地址装入 SSPxADD。
13. 主器件随着时钟将数据字节移入从器件，并在第 9 个 SCL 脉冲随着时钟将 ACK 移出从器件； SSPxIF 置 1。
14. 如果 SSPxCON2 的 SEN 位置 1，CKP 会被硬件清零，时钟会被延长。
15. 从器件清零 SSPxIF。
16. 从器件从 SSPxBUF 中读取接收的字节，使 BF 清零。
17. 如果 SEN 置 1，从器件会将 CKP 置 1，以释放 SCL。
18. 对于接收的每个字节重复步骤 13-17。
19. 主器件发送停止条件来结束数据发送。

32.5.5 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 SSPxADD 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 32-21 可以用作 AHEN 置 1 时 10 位寻址模式下从器件的参考图示。

图 32-22 给出了 10 位寻址模式下从发送器的标准波形图。

PIC16(L)F1764/5/8/9

图 32-20: I²C 从模式, 10 位地址, 接收 (SEN = 1, AHEN = 0, DHEN = 0)

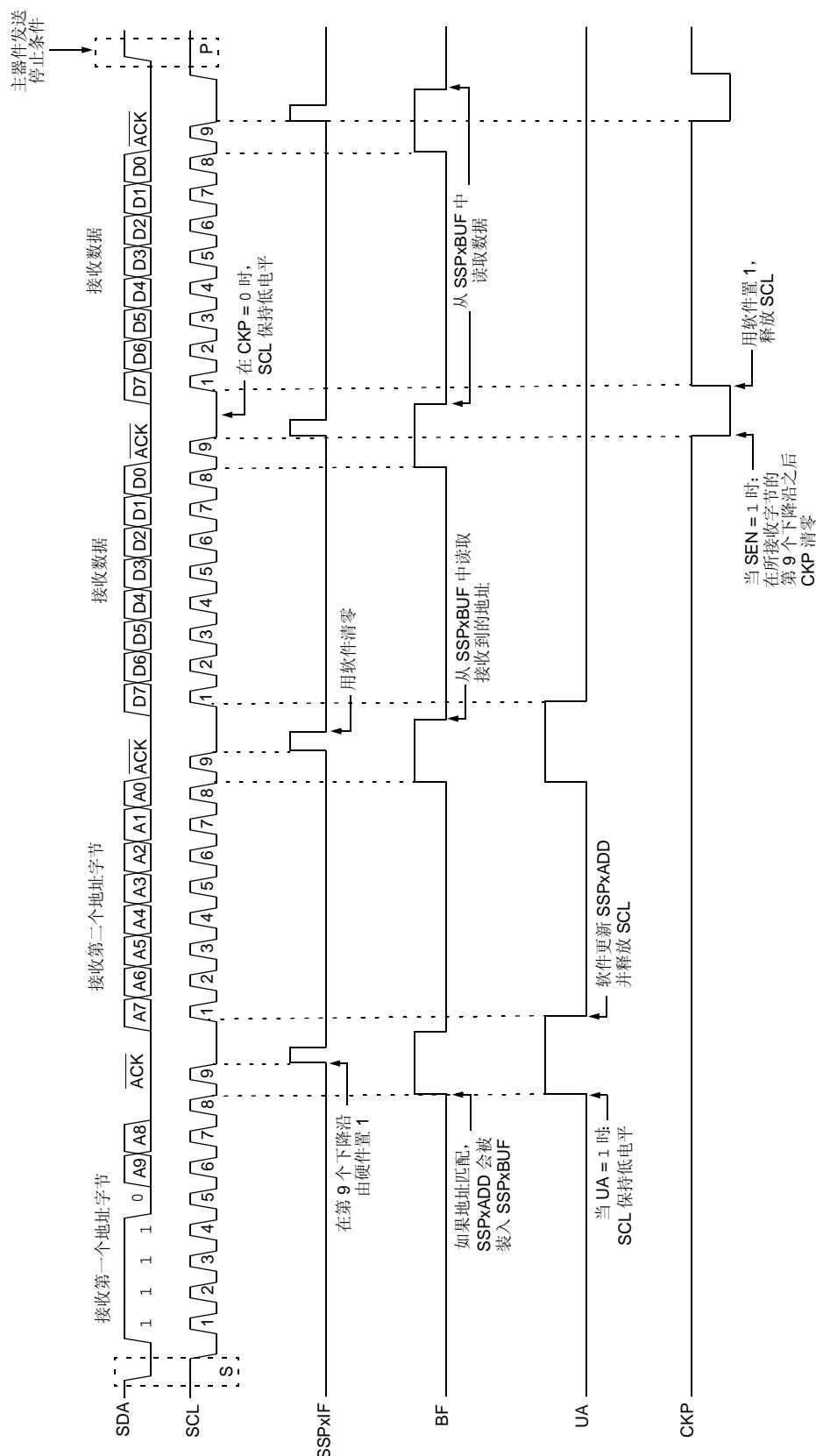
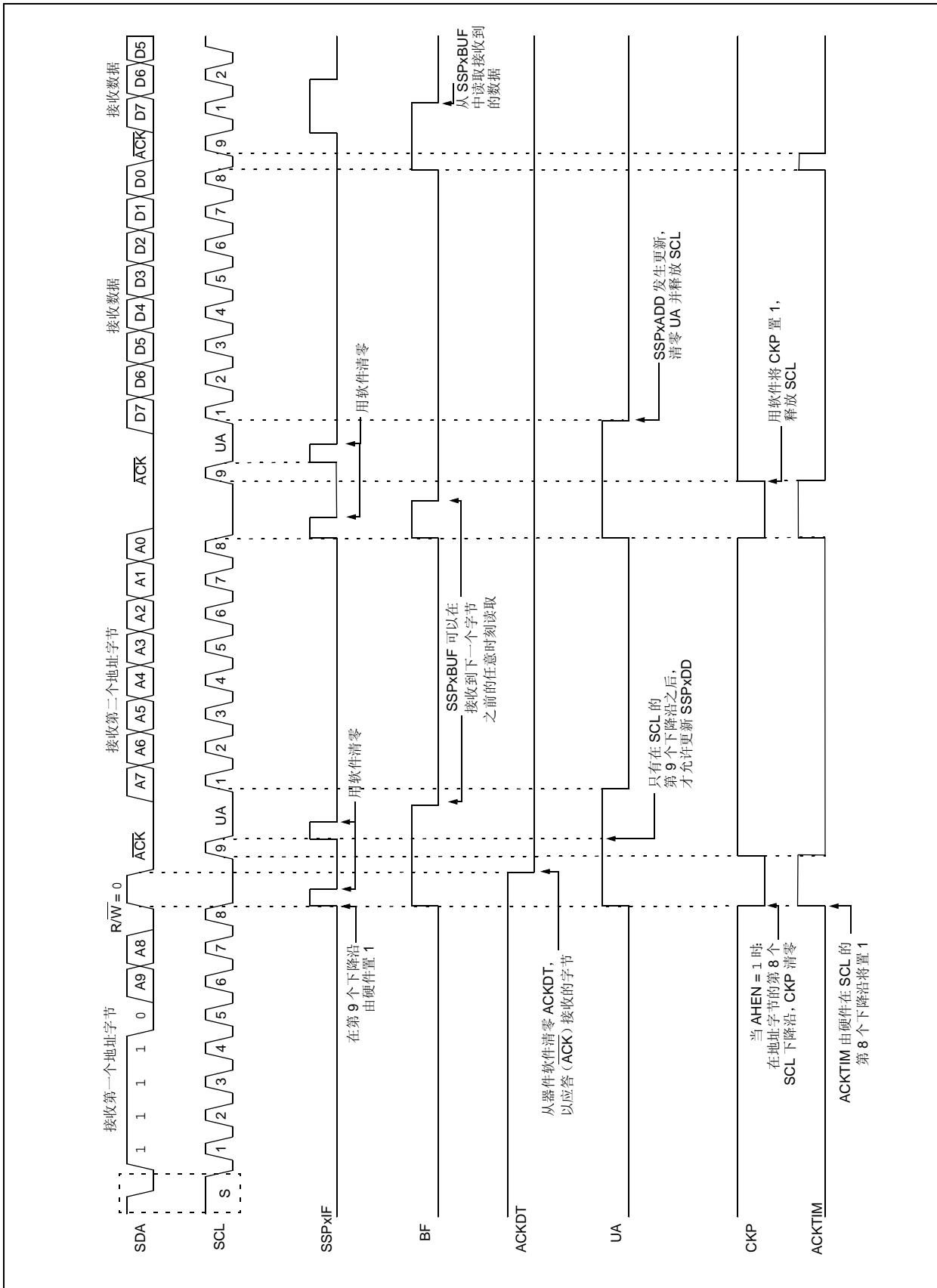
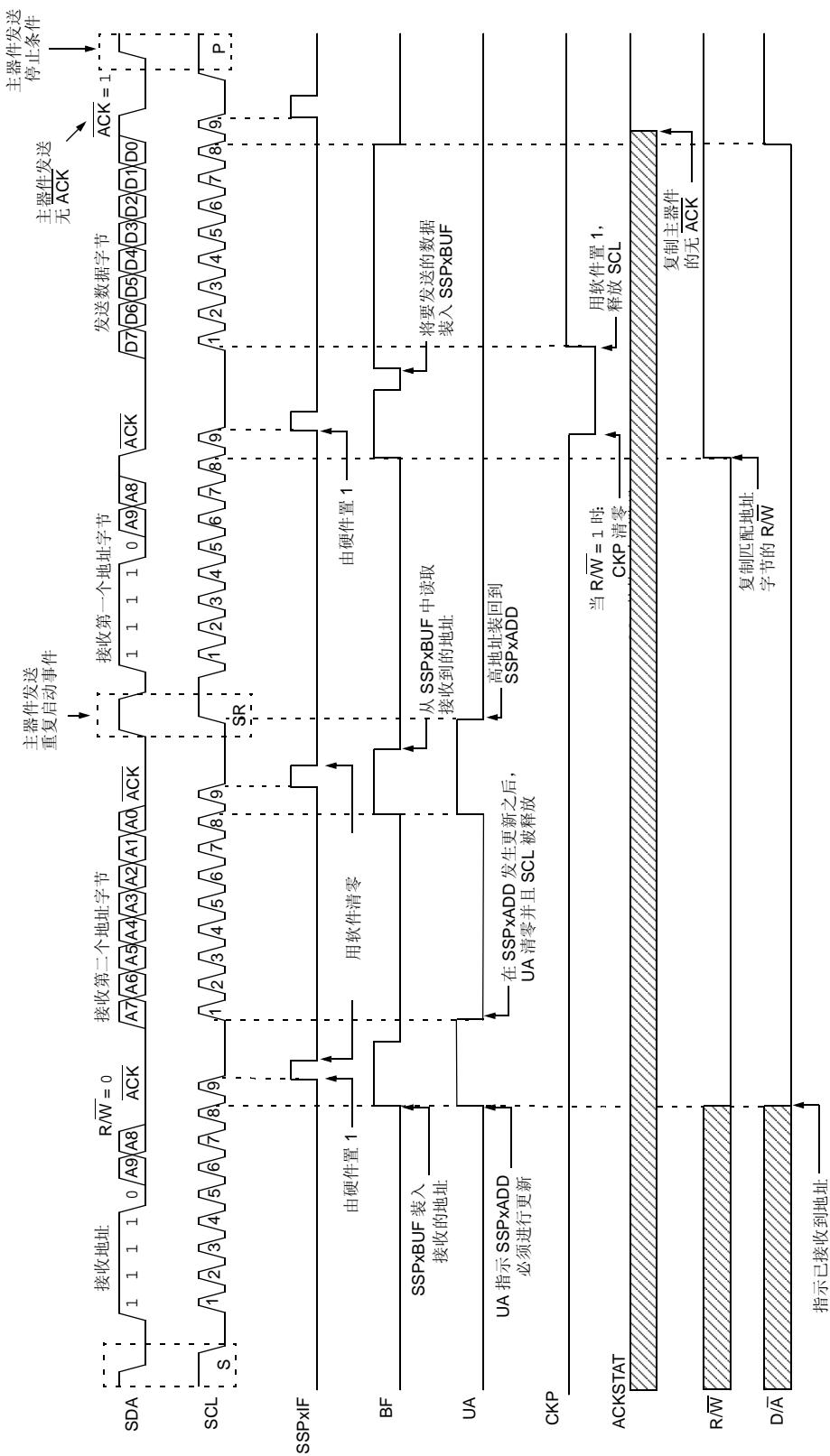


图 32-21: I²C 从模式, 10 位地址, 接收 (SEN = 0, AHEN = 1, DHEN = 0)



PIC16(L)F1764/5/8/9

图 32-22: I²C 从模式, 10 位地址, 发送 (**SEN** = 0, **AHEN** = 0, **DHEN** = 0)



32.5.6 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。

SSPxCON1 寄存器的 CKP 位用于在软件中控制时钟延长。每当 CKP 位清零时，模块就会等待 SCL 线变为低电平，然后保持低电平状态不变。将 CKP 置 1 将会释放 SCL，允许继续进行通信。

32.5.6.1 正常时钟延长

如果 SSPxSTAT 的 RW 位置 1（读请求），则在 ACK 之后，从器件硬件会清零 CKP。这让从器件可以有时间使用要传输给主器件的数据更新 SSPxBUF。如果 SSPxCON2 的 SEN 位置 1，则在 ACK 序列之后，从器件将总是延长时钟。在从器件就绪之后，软件会将 CKP 置 1，并继续进行通信。

- 注 1：** BF 位对于是否延长时钟没有任何影响。这一点与模块的先前版本不同：如果在 SCL 的第 9 个下降沿之前读取了 SSPxBUF，先前版本将不延长时钟，清零 CKP。
- 2：** 如果在 SCL 的第 9 个下降沿之前装入 SSPxBUF，则模块的先前版本不会为数据发送延长时钟。因此，对于读请求，总是会将该位清零。

32.5.6.2 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 的唯一情形。在写入 SSPxADD 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。

32.5.6.3 字节无应答

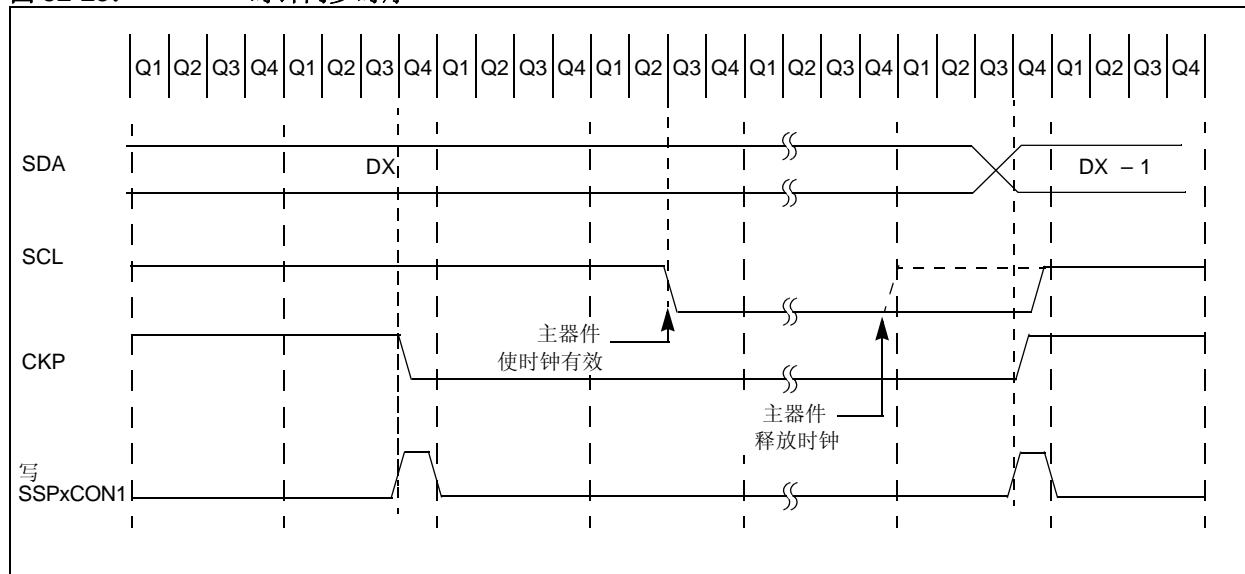
当 SSPxCON3 的 AHEN 位置 1 时，在所接收匹配地址字节的第 8 个 SCL 下降沿之后，硬件会将 CKP 清零。当 SSPxCON3 的 DHEN 位置 1 时，在所接收数据的第 8 个 SCL 下降沿之后，CKP 会被清零。

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。

32.5.6.4 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 32-23）。

图 32-23：时钟同步时序



32.5.7 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。

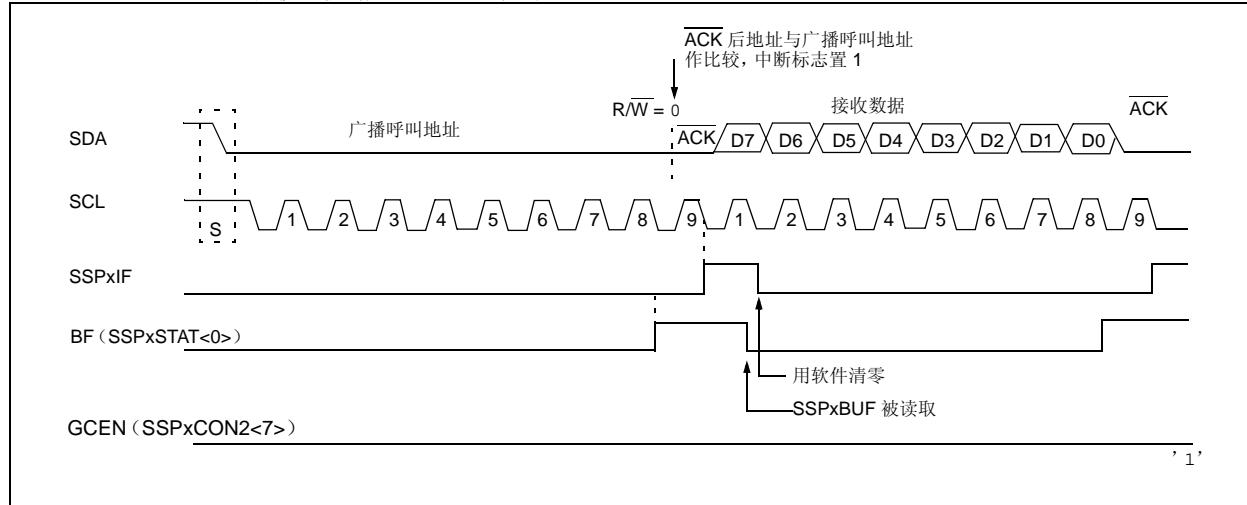
广播呼叫地址是 I²C 协议中的保留地址，定义为地址 0x00。如果 SSPxCON2 寄存器的 GCEN 位置 1，则无论 SSPxADD 中存储的值如何，在接收到该地址时，从模块都会自动发送 ACK。在从器件移入 R/W 位清零的

全零地址之后，将会产生中断，从器件软件可以读取 SSPxBUF 并进行响应。图 32-24 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 SSPxCON3 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 32-24：从模式广播呼叫地址序列



32.5.8 SSP 掩码寄存器

SSP 掩码 (SSPxMSK) 寄存器（[寄存器 32-5](#)）在 I²C 从模式下可用，用作地址比较操作期间 SSPSR 寄存器中保存的值的掩码。SSPxMSK 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。

32.6 I²C 主模式

通过将 SSPxCON1 中的相应 SSPM 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。

主模式通过在检测到启动条件和停止条件时产生中断来工作。停止 (P) 位和启动 (S) 位在复位或禁止 MSSP 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位条件检测执行所有的 I²C 总线操作。在该模式下，启动和停止条件检测是惟一有效的电路。所有其他通信都通过用户软件直接操作 SDA 和 SCL 线来完成。

以下事件会使 SSP 中断标志位 SSPxIF 置 1 (如果允许 SSP 中断，则产生中断)：

- 检测到启动条件
- 检测到停止条件
- 数据传输字节发送 / 接收
- 应答发送 / 接收
- 产生重复启动条件

注 1: 当配置为 I²C 主模式时，MSSP 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 SSPxBUF 寄存器以启动传输。在这种情况下，将不会写 SSPxBUF，WCOL 位将被置 1，指示没有发生对 SSPxBUF 的写操作。

2: 当通过 SEN/PEN 控制位发送启动 / 停止条件时，主模式暂停启动 / 停止检测。当硬件清零 SEN/PEN 控制位时，SSPxIF 位在启动 / 停止条件产生结束时置 1。

32.6.1 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。

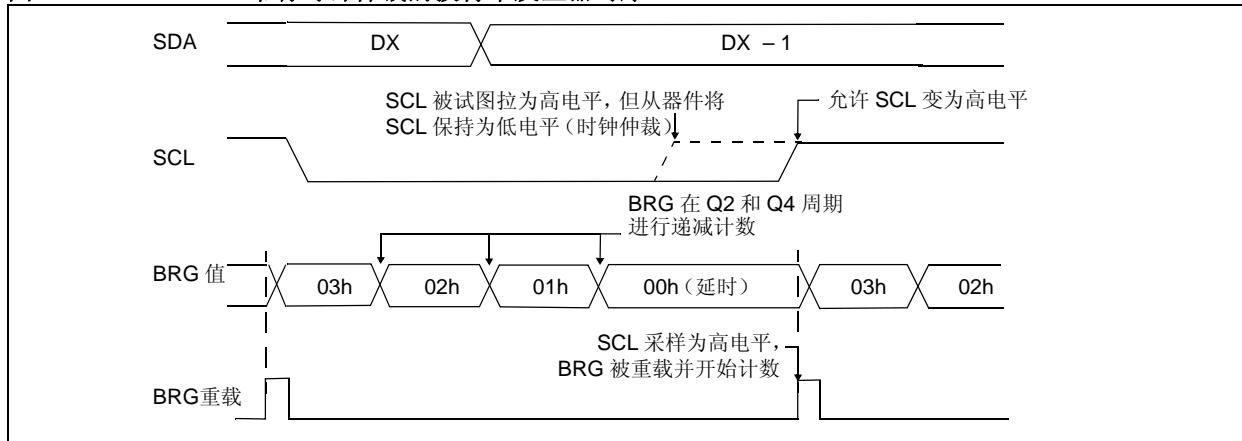
在主接收模式下，发送的第一个字节包括作为发送方的从器件地址 (7 位) 和 R/W 位。在这种情况下，R/W 将为逻辑 1。因此，发送的第一个字节是一个 7 位从器件地址，后跟 1 指示接收位。串行数据通过 SDA 接收，而串行时钟由 SCL 输出。一次接收 8 位串行数据。每接收到一个字节，都会发送一个应答位。启动条件和停止条件指示发送的开始和结束。

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 32.7 节 “波特率发生器”。

32.6.2 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 SSPxADD<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 32-25）。

图 32-25：带有时钟仲裁的波特率发生器时序



32.6.3 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 SSPxBUF，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 SSPxBUF 尝试了某个操作。

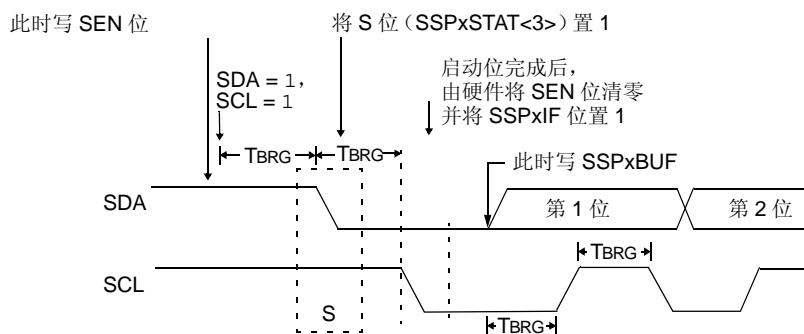
注：由于不允许事件排队，在启动条件结束之前，不能写 SSPxCON2 的低 5 位。

32.6.4 I²C 主模式启动条件时序

要发出启动条件（图 32-26），用户应将 SSPxCON2 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 SSPxADD<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 SSPxSTAT1 寄存器的 S 位置 1。随后波特率发生器重新装入 SSPxADD<7:0> 的内容并恢复计数。当波特率发生器再次超时（TBRG）时，SSPxCON2 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 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 BCLIF 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 32-26：第一个启动位时序



32.6.5 I²C 主模式重复启动条件时序

当 SSPxCON2 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 32-27）。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

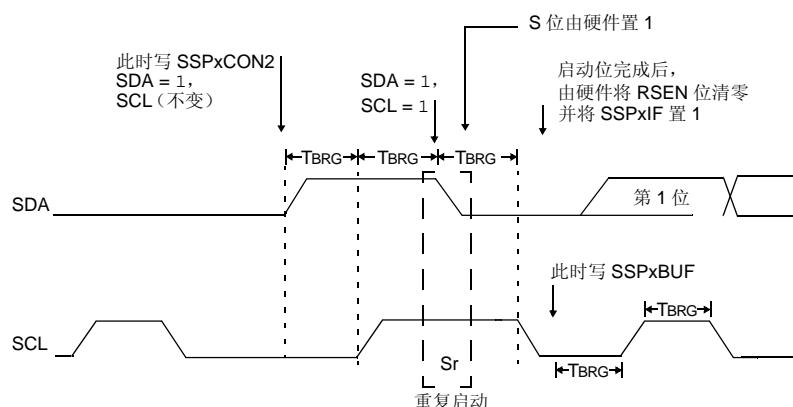
平。SCL 被置为低电平。随后 SSPxCON2 寄存器的 RSEN 位将自动清零，这次波特率发生器不会重载，SDA 引脚保持低电平。一旦在 SDA 和 SCL 引脚上检测到启动条件，SSPxSTAT 寄存器的 S 位将被置 1。SSPxIF 位在波特率发生器超时之前不会被置 1。

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尝试发送一个数据 1。

图 32-27: 重复启动条件波形图



32.6.6 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 SSPxBUF 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，SSPxIF 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 SSPxBUF，SCL 保持低电平，SDA 保持不变（图 32-28）。

在写 SSPxBUF 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 SSPxCON2 寄存器的 ACKSTAT 状态位。在发送地址的第 9 个时钟下降沿之后，SSPxIF 置 1，BF 标志清零，波特率发生器关闭直到发生下一次写 SSPxBUF，且 SCL 保持低电平，允许 SDA 悬空。

32.6.6.1 BF 状态标志

在发送模式下，SSPxSTAT 寄存器的 BF 位在 CPU 写 SSPxBUF 时置 1，在所有 8 位数据移出后清零。

32.6.6.2 WCOL 状态标志

如果在发送过程中（即，SSPSR 仍在移出数据字节时）用户写 SSPxBUF，则 WCOL 位被置 1，同时缓冲区内容不变（未发生写操作）。

在下一次发送前 WCOL 必须用软件清零。

32.6.6.3 ACKSTAT 状态标志

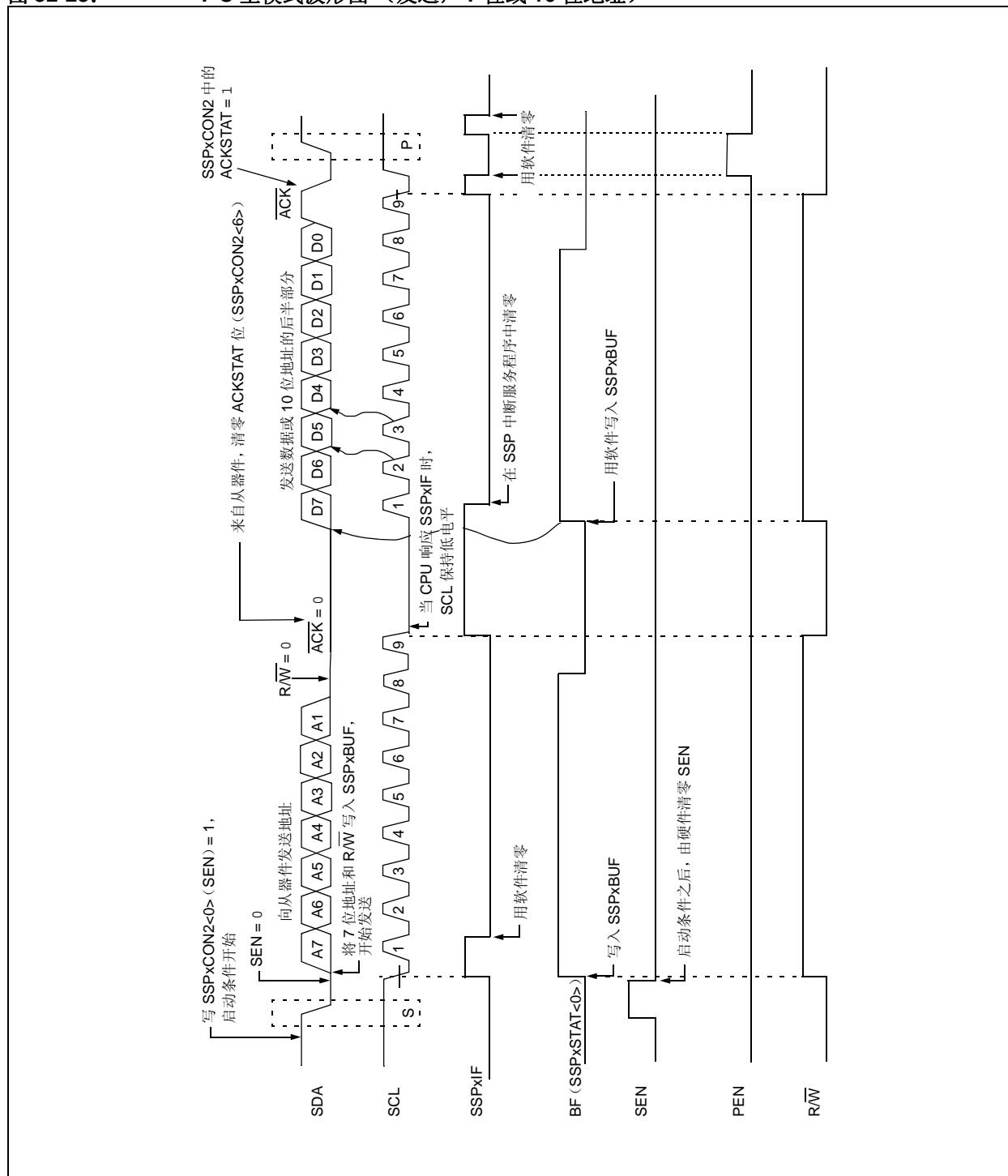
在发送模式下，当从器件发送应答 ($\overline{\text{ACK}} = 0$) 时，SSPxCON2 寄存器的 ACKSTAT 位被清零；当从器件没有应答 ($\overline{\text{ACK}} = 1$) 时，该位被置 1。从器件在识别出其地址（包括广播呼叫地址）或正确接收数据后，会发送一个应答。

32.6.6.4 典型的发送序列：

1. 用户通过将 SSPxCON2 寄存器的 SEN 位置 1，产生启动条件。
2. 在启动条件结束时，硬件将 SSPxIF 置 1。
3. SSPxIF 用软件清零。
4. 在进行任何其他操作前，MSSP 模块将等待所需的启动时间。
5. 用户将从器件地址装入 SSPxBUF 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 SSPxBUF 后立刻开始。
7. MSSP 模块移入来自从器件的 $\overline{\text{ACK}}$ 位，并将它的值写入 SSPxCON2 寄存器的 ACKSTAT 位。
8. MSSP 模块在第 9 个时钟周期结束时将 SSPxIF 置 1 产生中断。
9. 用户将 8 位数据装入 SSPxBUF。
10. 数据从 SDA 引脚移出，直到发送完所有 8 位数据。
11. MSSP 模块移入来自从器件的 $\overline{\text{ACK}}$ 位，并将它的值写入 SSPxCON2 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 SSPxCON2 寄存器的 PEN 或 RSEN 位置 1，产生停止或重复启动条件。停止 / 重复启动条件完成时产生中断。

PIC16(L)F1764/5/8/9

图 32-28: I²C 主模式波形图 (发送, 7 位或 10 位地址)



32.6.7 I²C 主模式接收

通过编程 SSPxCON2 寄存器的接收使能位 RCEN，可以使能主模式接收（图 32-29）。

注： 将 RCEN 位置 1 前，MSSP 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 SSPSR。在第 8 个时钟的下降沿之后，接收使能标志自动清零，SSPSR 的内容装入 SSPxBUF，BF 标志位置 1，SSPxIF 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 MSSP 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 SSPxCON2 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。

32.6.7.1 BF 状态标志

接收过程中，把地址或数据字节从 SSPSR 装入 SSPxBUF 时，BF 位置 1。在读 SSPxBUF 寄存器时将其清零。

32.6.7.2 SSPOV 状态标志

在接收操作中，当 SSPSR 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。

32.6.7.3 WCOL 状态标志

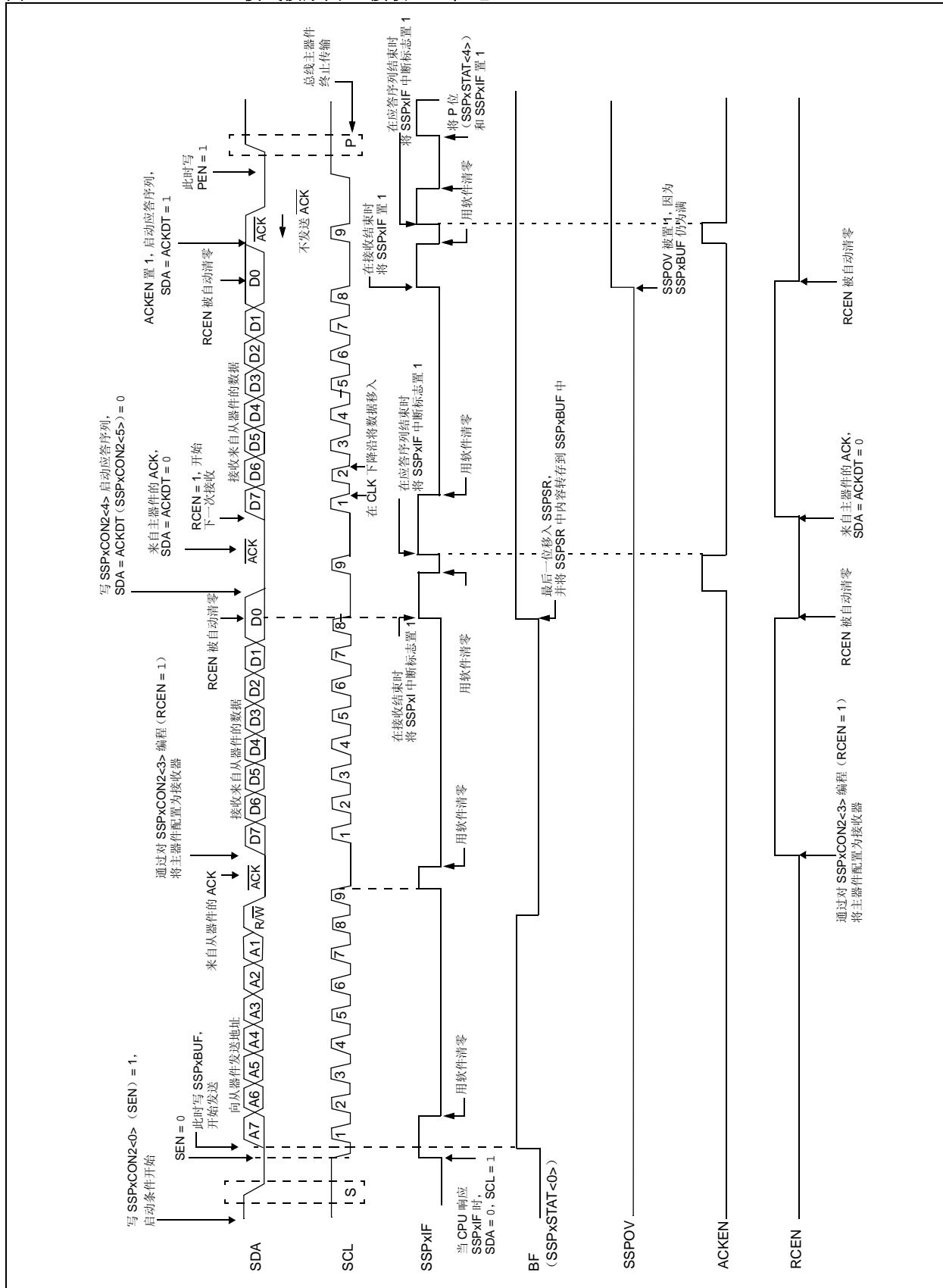
如果在接收过程中（即，SSPSR 仍在移入数据字节时）用户写 SSPxBUF，则 WCOL 位被置 1，同时缓冲区内容不变（未发生写操作）。

32.6.7.4 典型的接收序列：

1. 用户通过将 SSPxCON2 寄存器的 SEN 位置 1，产生启动条件。
2. 在启动条件结束时，硬件将 SSPxIF 置 1。
3. SSPxIF 用软件清零。
4. 用户将要发送的从器件地址写入 SSPxBUF 且 R/W 位置 1。
5. 器件地址从 SDA 引脚移出，直到发送完所有 8 位地址。发送会在写入 SSPxBUF 后立刻开始。
6. MSSP 模块移入来自从器件的 ACK 位，并将它的值写入 SSPxCON2 寄存器的 ACKSTAT 位。
7. MSSP 模块在第 9 个时钟周期结束时将 SSPxIF 置 1 产生中断。
8. 用户将 SSPxCON2 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，SSPxIF 和 BF 置 1。
10. 主器件清零 SSPxIF，并从 SSPxBUF 中读取接收到的字节，使 BF 清零。
11. 主器件在 SSPxCON2 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 SSPxIF 置 1。
13. 用户清零 SSPxIF。
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。

PIC16(L)F1764/5/8/9

图 32-29: I²C 主模式波形图 (接收, 7 位地址)



32.6.8 应答序列时序

将 SSPxCON2 寄存器的应答序列使能位 ACKEN 置 1 即可使能应答序列。当该位被置 1 时，SCL 引脚被拉为低电平，应答数据位的内容输出到 SDA 引脚上。如果用户希望产生一个应答，则应将 ACKDT 位清零。否则，用户应在应答序列开始前将 ACKDT 位置 1。然后波特率发生器进行一个计满返回周期 (TBRG) 的计数，随后 SCL 引脚电平被置为无效（拉为高电平）。当 SCL 引脚被采样为高电平（时钟仲裁）时，波特率发生器再进行一个 TBRG 周期的计数。然后 SCL 引脚被拉为低电平。在这之后，ACKEN 位自动清零，波特率发生器关闭，MSSP 模块进入空闲模式（图 32-30）。

32.6.8.1 WCOL 状态标志

如果在应答序列进行过程中用户写 SSPxBUF，则 WCOL 位被置 1，同时缓冲区内容不变（未发生写操作）。

图 32-30：应答序列波形图

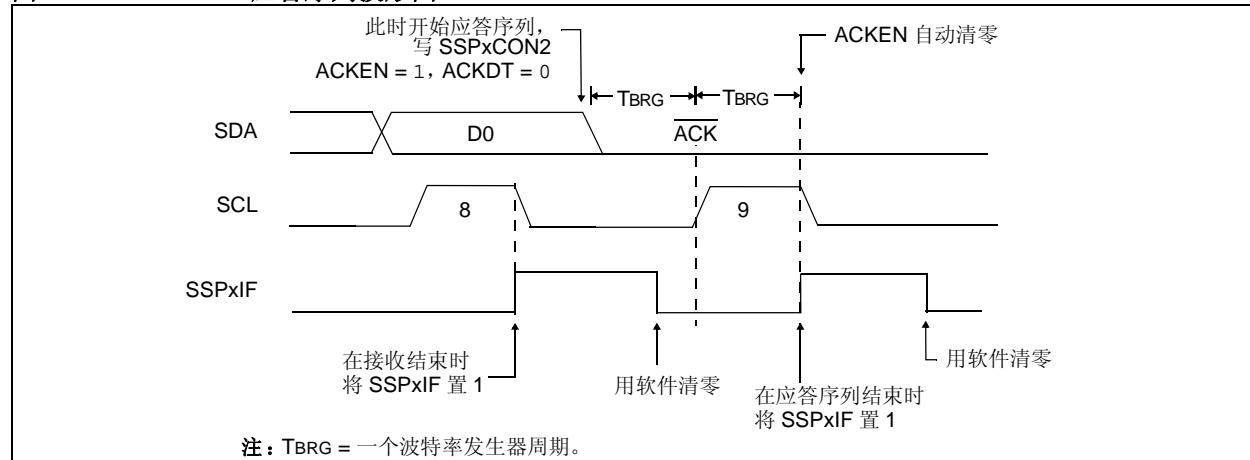
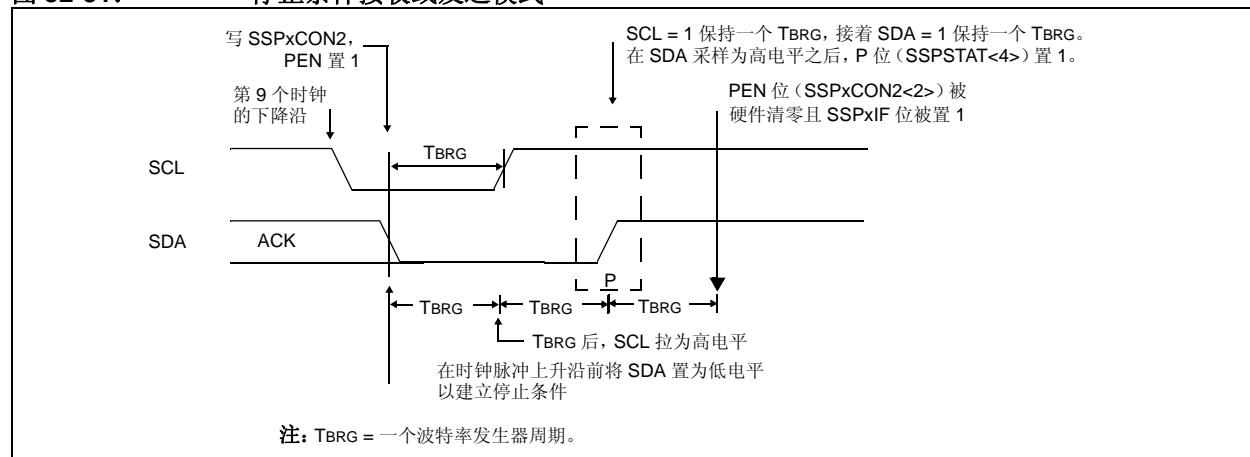


图 32-31：停止条件接收或发送模式



32.6.9 停止条件时序

如果将 SSPxCON2 寄存器的停止序列使能位 PEN 置 1，则在接收 / 发送结束后，SDA 引脚上将产生停止位。在接收 / 发送结束时，SCL 线在第 9 个时钟的下降沿后保持低电平。当 PEN 位置 1 时，主器件将 SDA 线置为低电平。当 SDA 线被采样为低电平时，波特率发生器被重载并递减计数至 0。当波特率发生器超时时，SCL 引脚被拉为高电平，在一个 TBRG (波特率发生器计满返回周期) 之后，SDA 引脚将被置为无效。当 SDA 引脚采样为高电平且 SCL 也是高电平时，SSPxSTAT 寄存器的 P 位置 1。另一个 TBRG 之后，PEN 位被清零，同时 SSPxIF 位被置 1 (图 32-31)。

32.6.9.1 WCOL 状态标志

如果在停止序列进行过程中用户写 SSPxBUF，则 WCOL 位被置 1，同时缓冲区内容不变（未发生写操作）。

32.6.10 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。

32.6.11 复位的影响

复位会禁止 MSSP 模块并终止当前的数据传输。

32.6.12 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 MSSP 模块时清零。当 SSPxSTAT 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 BCLIF 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

32.6.13 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 上输出 1 (将 SDA 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 BCLIF 置 1, 并将 I²C 端口复位到空闲状态 (图 32-32)。

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 SSPxBUF。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。

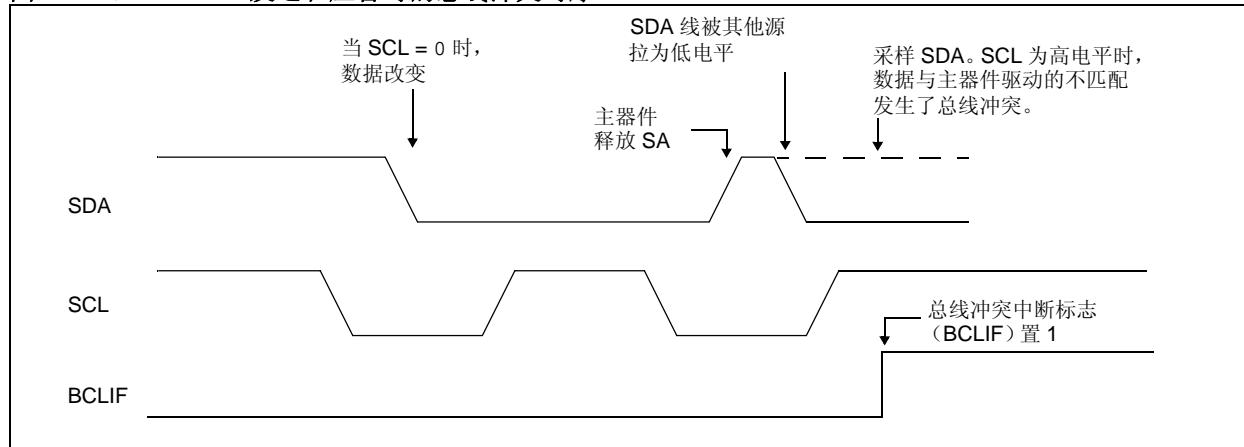
如果在启动、重复启动、停止或应答信号的进行过程中发生总线冲突, 则这种状态被中止, SDA 和 SCL 线被置为无效, SSPxCON2 寄存器中的对应控制位清零。当执行总线冲突中断服务程序时, 如果 I²C 总线空闲, 用户可通过发出启动条件恢复通信。

主器件将继续监视 SDA 和 SCL 引脚。一旦出现停止条件, SSPxIF 位将被置 1。

发生总线冲突时无论发送的进度如何, 写入 SSPxBUF 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 SSPxSTAT 寄存器中的 P 位置 1 时, 可以获取 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 32-32: 发送和应答时的总线冲突时序



32.6.13.1 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 32-33）。
- SDA 置为低电平之前，SCL 采样为低电平（图 32-34）。

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，
- BCLIF 标志置 1，并且
- MSSP 模块复位为空闲状态（图 32-33）。

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，则发生了总线冲突，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 32-35）。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 32-33：启动条件期间的总线冲突（仅用于 SDA）

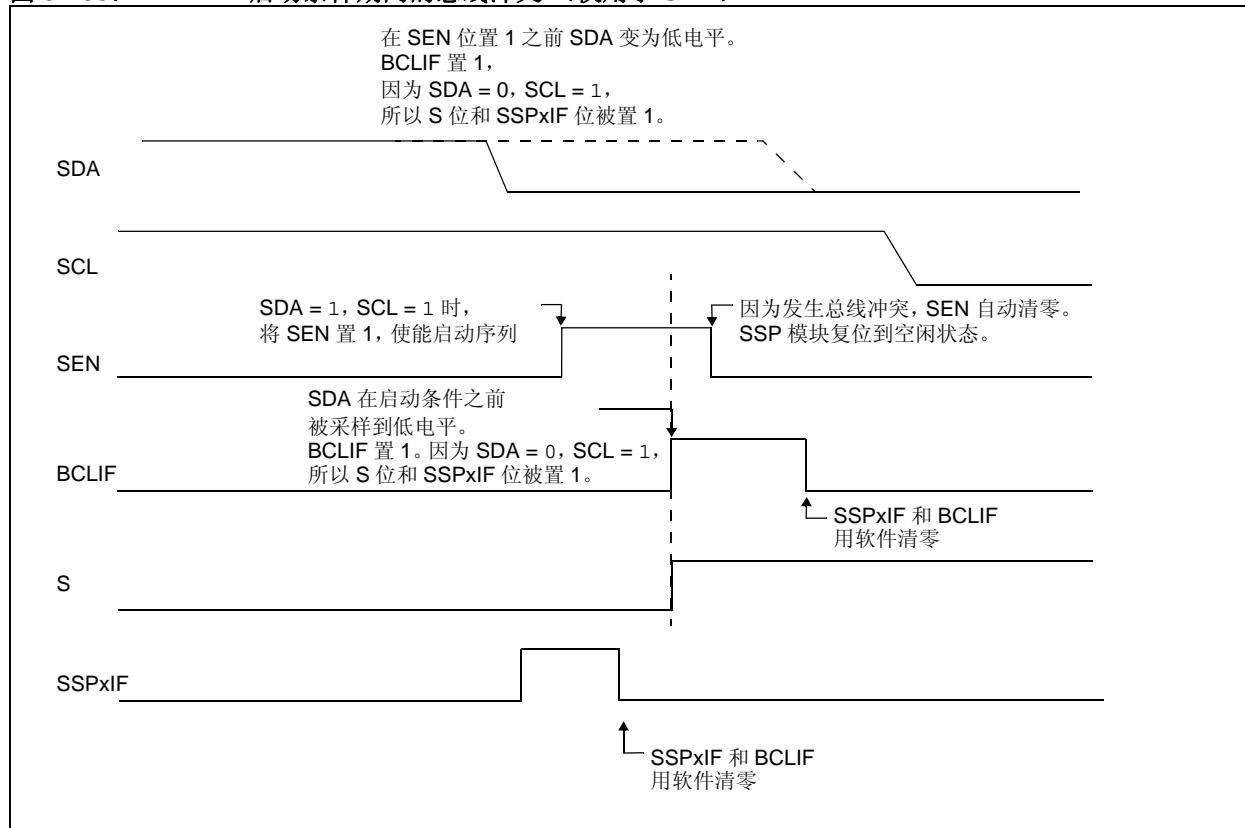


图 32-34: 启动条件期间的总线冲突 ($SCL = 0$)

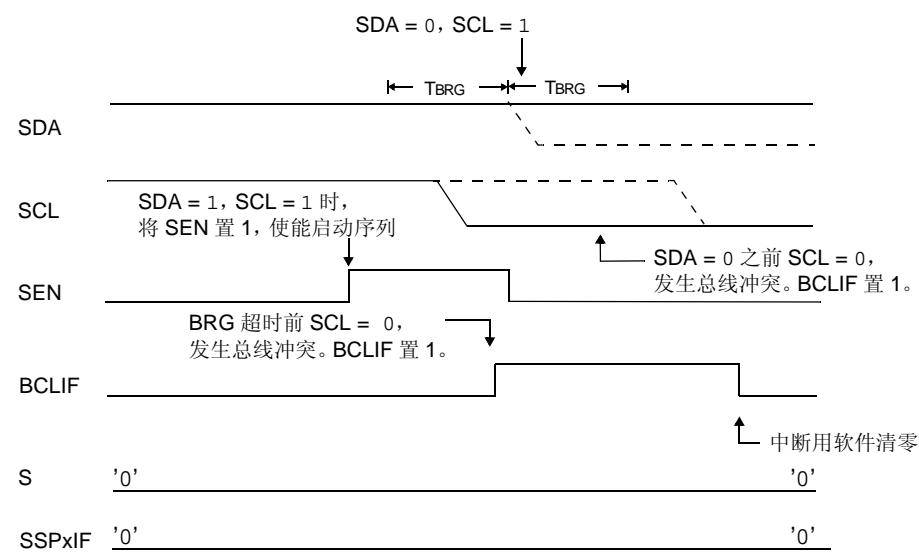
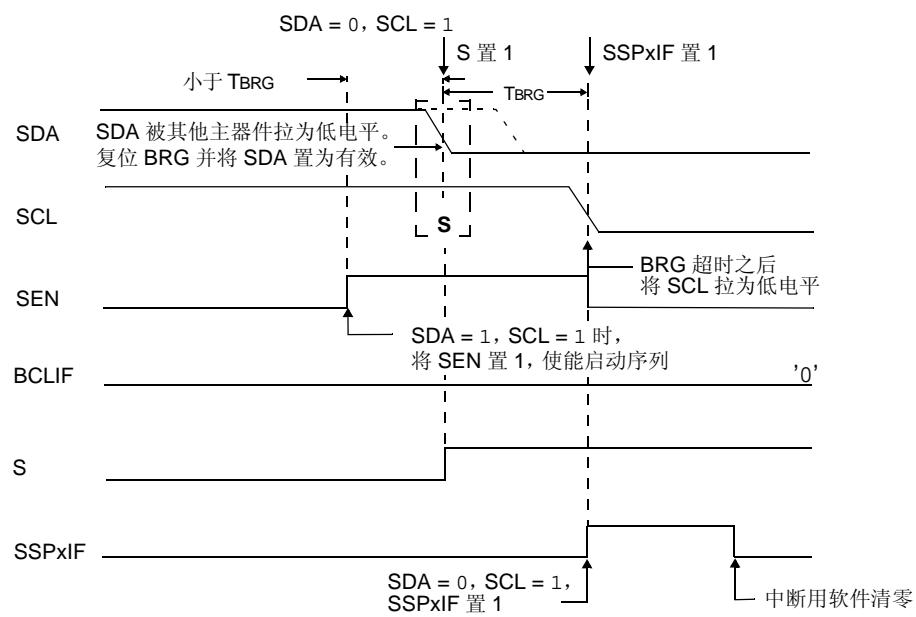


图 32-35: 启动条件期间由 SDA 仲裁引起的 BRG 复位



32.6.13.2 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 SSPxADD 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 32-36）。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尝试发送一个数据 1（见图 32-27）。

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 32-36：重复启动条件期间的总线冲突（情形 1）

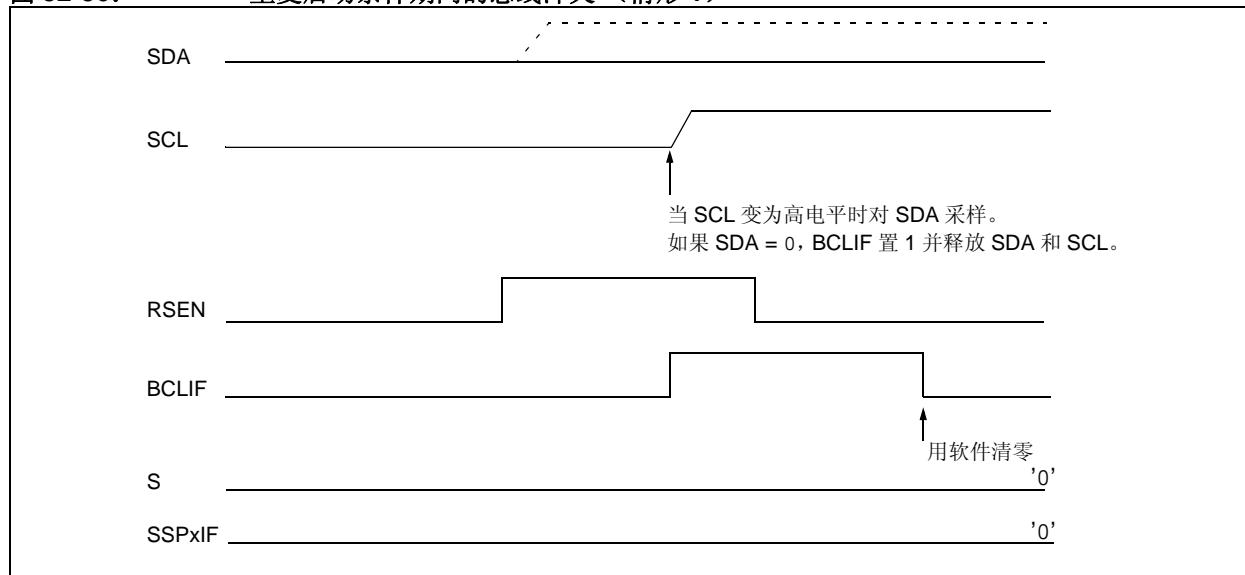
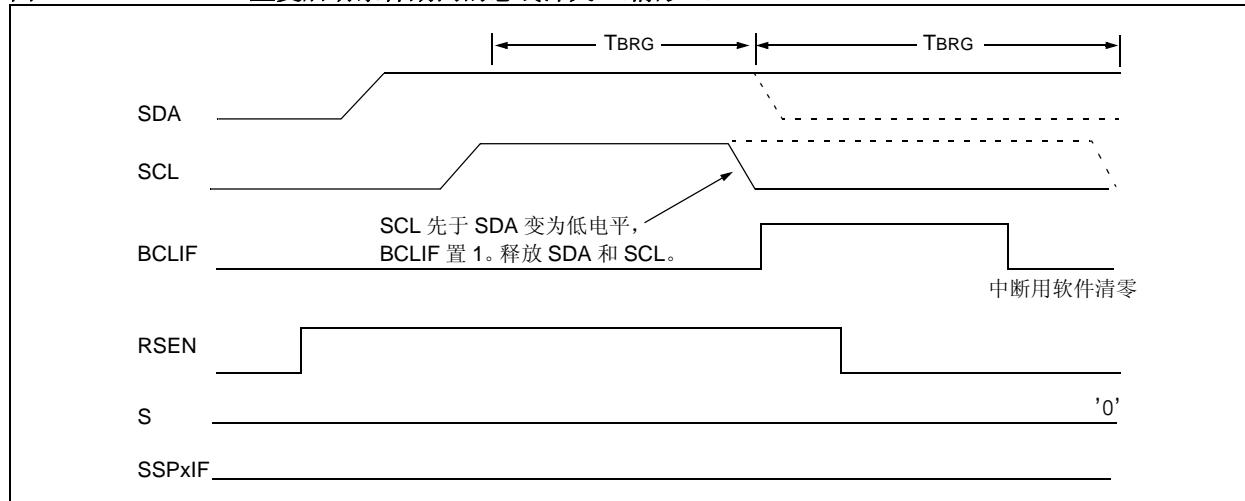


图 32-37：重复启动条件期间的总线冲突（情形 2）



32.6.13.3 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：

- 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）。
- 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 SSPxADD 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尝试发送一个数据 0（图 32-38）。如果在允许 SDA 悬空为高电平前 SCL 引脚被采样到低电平，也会发生总线冲突。这是另一个主器件正尝试发送一个数据 0 的另外一种情况（图 32-39）。

图 32-38：停止条件期间的总线冲突（情形 1）

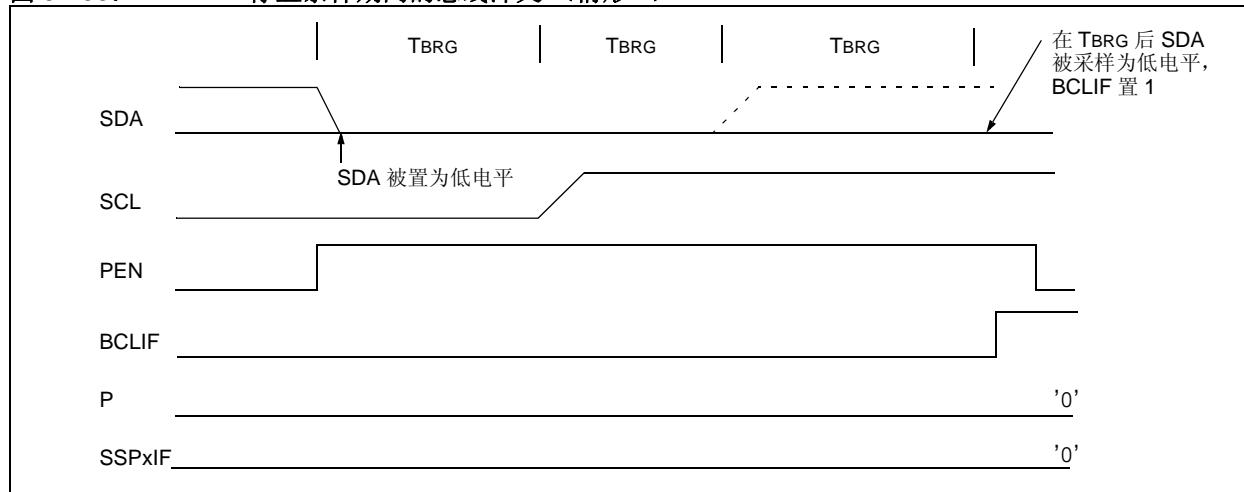


图 32-39：停止条件期间的总线冲突（情形 2）

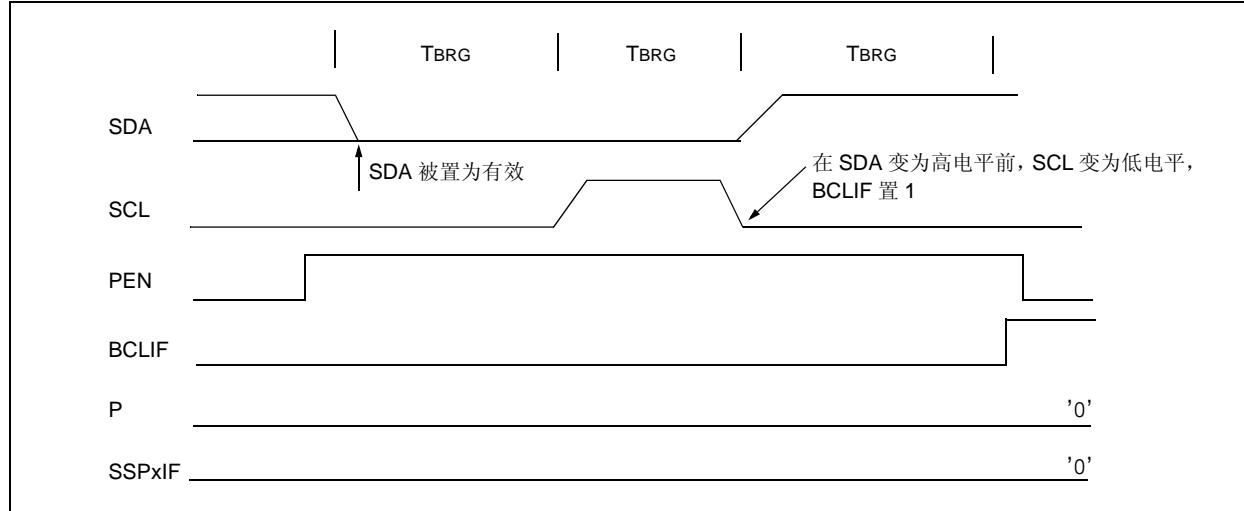


表 32-3：与 I²C 操作相关的寄存器总汇

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页:
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE ⁽¹⁾	C3IE ⁽¹⁾	CCP2IE ⁽¹⁾	104
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF ⁽¹⁾	C2IF ⁽¹⁾	CCP2IF ⁽¹⁾	107
RxyPPS	—	—	—	RxyPPS<4:0>					158
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					158, 160
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					158, 160
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					158, 160
SSP1ADD	ADD<7:0>							427	
SSP1BUF	同步串行端口接收缓冲区 / 发送寄存器								379*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				424
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	425
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	426
SSP1MSK	MSK<7:0>							427	
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	423
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	150

图注: — = 未实现, 读为 0。I²C 模式下的 MSSP 模块不使用阴影单元。

* 提供寄存器信息的页。

注 1: 仅限 PIC16(L)F1768/9。

2: 未实现位, 读为 1。

32.7 波特率发生器

MSSP 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 SSPxADD 寄存器（[寄存器 32-6](#)）中。当发生对 SSPxBUF 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 32-40](#) 中的内部信号“重载”会触发将 SSPxADD 值装入 BRG 计数器。对于模块时钟线的每次振荡，这会

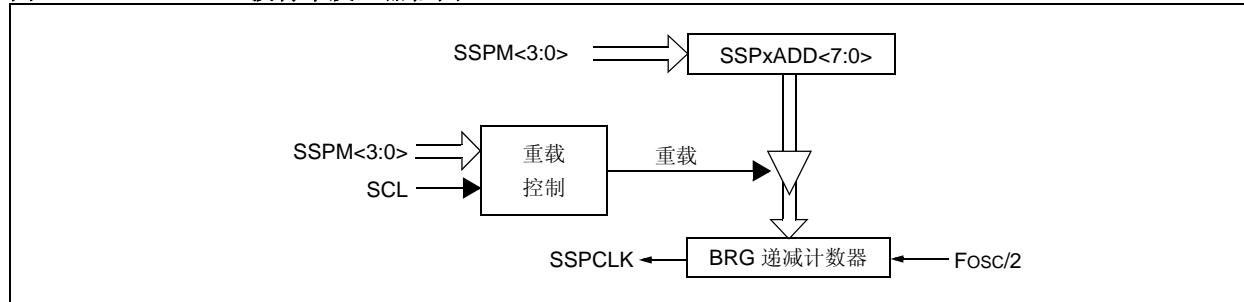
发生两次。指定重载信号何时置为有效的逻辑依赖于 MSSP 当前的工作模式。

[表 32-4](#) 列出了不同的指令周期下的时钟速率以及装入 SSPxADD 的 BRG 值。

公式 32-1：

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

[图 32-40：](#) 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 SSPxADD 是无效的。这是实现限制。

[表 32-4：](#) 使用 BRG 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 36-4](#) 中的 I/O 端口电气规范。

32.8 寄存器定义：MSSP 控制

寄存器 32-1： SSP1STAT： SSP 状态寄存器

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
SMP	CKE	D/A	P	S	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

bit 7	SMP: SPI 数据输入采样位 <u>SPI 主模式:</u> 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 <u>SPI 从模式:</u> 当 SPI 工作在从模式时，必须将 SMP 清零 在 I ² C 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 和 1 MHz） 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） 在 SPI 主模式或从模式下： 1 = 时钟状态从有效转换到空闲时发送 0 = 时钟状态从空闲转换到有效时发送 仅在 I ² C 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/A: 数据 / 地址位（仅限 I ² C 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 (仅限 I ² C 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 (仅限 I ² C 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 I ² C 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 I ² C 从模式下： 1 = 读 0 = 写 在 I ² C 主模式下： 1 = 正在进行发送 0 = 未进行发送 将该位与 SEN、RSEN、PEN、RCEN 或 ACKEN 进行逻辑或运算将指示 MSSP 是否处于空闲模式。
bit 1	UA: 更新地址位（仅限 10 位 I ² C 模式） 1 = 指示用户需要更新 SSP1ADD 寄存器中的地址 0 = 不需要更新地址
bit 0	BF: 缓冲区满状态位 接收（SPI 和 I ² C 模式）： 1 = 接收完成，SSP1BUF 已满 0 = 接收未完成，SSP1BUF 为空 发送（仅限 I ² C 模式）： 1 = 数据发送正在进行（不包括 ACK 位和停止位），SSP1BUF 为满 0 = 数据发送完成（不包括 ACK 位和停止位），SSP1BUF 为空

PIC16(L)F1764/5/8/9

寄存器 32-2: SSP1CON1: SSP 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

主模式:

1 = 当 I²C 不满足启动发送条件时, 试图写 SSP1BUF 寄存器

0 = 未发生冲突

从模式:

1 = 正在发送前一个字时写 SSP1BUF 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 = SSP1BUF 寄存器中仍保存前一数据时, 又接收到一个新的字节。如果发生溢出, SSPSR 中的数据会丢失。溢出只会在从模式下发生。在从模式下, 即使只是发送数据, 用户也必须读 SSP1BUF, 以避免将溢出位置 1。在主模式下, 溢出位不会被置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSP1BUF 寄存器启动的 (必须用软件清零)。

0 = 无溢出

在 I²C 模式下:

1 = SSP1BUF 寄存器中仍保存前一字节时, 又接收到一个新的字节。在发送模式下, 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, 必须将这些引脚正确地配置为输入或输出

在 SPI 模式下:

1 = 使能串口并将 SCK、SDO、SDI 和 SS 配置为串口引脚源⁽²⁾

0 = 禁止串口并将上述引脚配置为 I/O 端口引脚

在 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

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 = Fosc/(4 * (SSP1ADD+1))⁽⁵⁾

1001 = 保留

1000 = I²C 主模式, 时钟 = Fosc / (4 * (SSP1ADD+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 SS 引脚控制, SS 可用作 I/O 引脚

0100 = SPI 从模式, 时钟 = SCK 引脚, 使能 SS 引脚控制

0011 = SPI 主模式, 时钟 = T2_match/2

0010 = SPI 主模式, 时钟 = Fosc/64

0001 = SPI 主模式, 时钟 = Fosc/16

0000 = SPI 主模式, 时钟 = Fosc/4

注

1: 在主模式下, 溢出位不会被置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSP1BUF 寄存器启动的。

2: 当使能时, 必须将这些引脚正确地配置为输入或输出。使用 SSPSSPPS、SSPCLKPPS、SSPDATPPS 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 SSPCLKPPS、SSPDATPPS 和 RxyPPS 来选择引脚。

4: 对于 I²C 模式, 不支持 SSP1ADD 值 0、1 或 2。

5: 不支持 SSP1ADD 值 0。而使用 SSPM = 0000。

寄存器 32-3: SSP1CON2: SSP 控制寄存器 2⁽¹⁾

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-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 = 当 SSPSR 接收到广播呼叫地址 (0x00) 时允许中断
 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)
SCKMSSP 释放控制:
 1 = 在 SDA 和 SCL 引脚上发出停止条件。由硬件自动清零。
 0 = 停止条件空闲
- 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 模块不处于空闲模式, 该位可能不会被置 1 (不支持并行操作), 并且可能不能写入 SSP1BUF (或禁止写入 SSP1BUF)。

PIC16(L)F1764/5/8/9

寄存器 32-4: SSP1CON3: SSP 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 = 允许检测到停止条件时产生中断

0 = 禁止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 = 允许检测到启动或重复启动条件时产生中断

0 = 禁止检测到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 = SSP1BUF 在每次新的数据字节移入时更新, 并忽略 BF 位

0 = 如果在接收到新字节时 SSP1STAT 寄存器的 BF 位已置 1, 则 SSP1CON1 寄存器的 SSPOV 位会置 1, 并且不会更新缓冲区

在 I²C 主模式和 SPI 主模式下:

该位被忽略。

在 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 SSP1BUF 并产生 ACK 信号, 忽略 SSPOV 位的状态。

0 = 只有在 SSPOV 清零时才更新 SSP1BUF

bit 3

SDAHT: SDA 保持时间选择位 (仅限 I²C 模式)

1 = SCL 下降沿之后, 在 SDA 上最少有 300 ns 的保持时间

0 = SCL 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 PIR2 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, SSP1CON1 寄存器的 CKP 位将清零, 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 = 在所接收数据字节的第 8 个 SCL 下降沿之后, 从器件硬件清零 SSP1CON1 寄存器的 CKP 位, 而 SCL 则保持低电平。

0 = 禁止数据保持

注 1: 用于菊花链 SPI 操作; 使用户可以忽略除最后一个接收到的字节之外的所有字节。在接收到新字节且 BF = 1 时, SSPOV 仍然会置 1, 但硬件会继续将最新字节写入 SSP1BUF。

2: 在启动和停止条件检测明确列为使能的从模式下, 该位没有任何作用。

3: ACKTIM 状态位仅在 AHEN 位或 DHEN 位置 1 时有效。

寄存器 32-5: SSP1MSK: SSP 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
MSK<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 **MSK<7:1>:** 掩码位

1 = 接收到的地址 bit n 与 SSP1ADD<n> 相比较来检测 I²C 模式下地址是否匹配

0 = 接收到的地址 bit n 不用于检测 I²C 模式下地址是否匹配

bit 0 **MSK<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 SSP1ADD<0> 相比较来检测 I²C 模式下地址是否匹配

0 = 接收到的地址 bit 0 不用于检测 I²C 模式下地址是否匹配

I²C 从模式、7 位地址时, 该位被忽略

寄存器 32-6: SSP1ADD: MSSP 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
ADD<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 **ADD<7:0>:** 波特率时钟分频比位

SCL 引脚时钟周期 = ((ADD<7:0> + 1) * 4)/Fosc

10 位从模式 —— 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”。主器件发送的位格式由 I²C 规范确定, 必须等于 11110。但是, 那些位通过硬件进行比较, 并且不受该寄存器中的值影响。

bit 2-1 **ADD<2:1>:** 10 位地址的高 2 位

bit 0 **未使用:** 在此模式下未使用。位状态为“无关”。

10 位从模式 —— 地址低字节:

bit 7-0 **ADD<7:0>:** 10 位地址的低 8 位

7 位从模式:

bit 7-1 **ADD<7:1>:** 7 位地址

bit 0 **未使用:** 在此模式下未使用。位状态为“无关”。

33.0 增强型通用同步 / 异步收发器 (EUSART)

增强型通用同步 / 异步收发器 (EUSART) 模块是一种串行 I/O 通信外设。它包含用来完成与器件程序执行独立的输入或输出串行数据传输所需的所有时钟发生器、移位寄存器和数据缓冲区等。EUSART 也可称为串行通信接口 (Serial Communication Interface, SCI)，可配置为全双工异步系统或半双工同步系统。全双工模式可用来与外设系统通信，如 CRT 终端和个人计算机。半双工同步模式用于与外设通信，如 A/D 或 D/A 集成电路、串行 EEPROM 或其他单片机。这些器件通常不具备用以产生波特率的内部时钟，并需要由主同步器件提供外部时钟信号。

EUSART 模块具备以下功能：

- 全双工异步收发
- 双字符输入缓冲区
- 单字符输出缓冲区
- 可编程 8 位或 9 位字符长度
- 9 位模式下的地址检测
- 输入缓冲区溢出错误检测
- 接收字符帧错误检测
- 半双工同步主模式
- 半双工同步从模式
- 同步模式下的可编程时钟极性
- 休眠模式下的操作

EUSART 模块还具备以下特性，使其成为局域互联网 (Local Interconnect Network, LIN) 总线系统的理想选择：

- 波特率的自动检测和校准
- 接收到间隔字符时唤醒
- 13 位间隔字符发送

EUSART 发送器和接收器的框图如图 33-1 和图 33-2 所示。

EUSART 发送输出 (TX_out) 可送至 TX/CK 引脚和在内部送至以下外设：

- 可配置逻辑单元 (CLC)
- 数据信号调制器 (DSM)

图 33-1：EUSART 发送框图

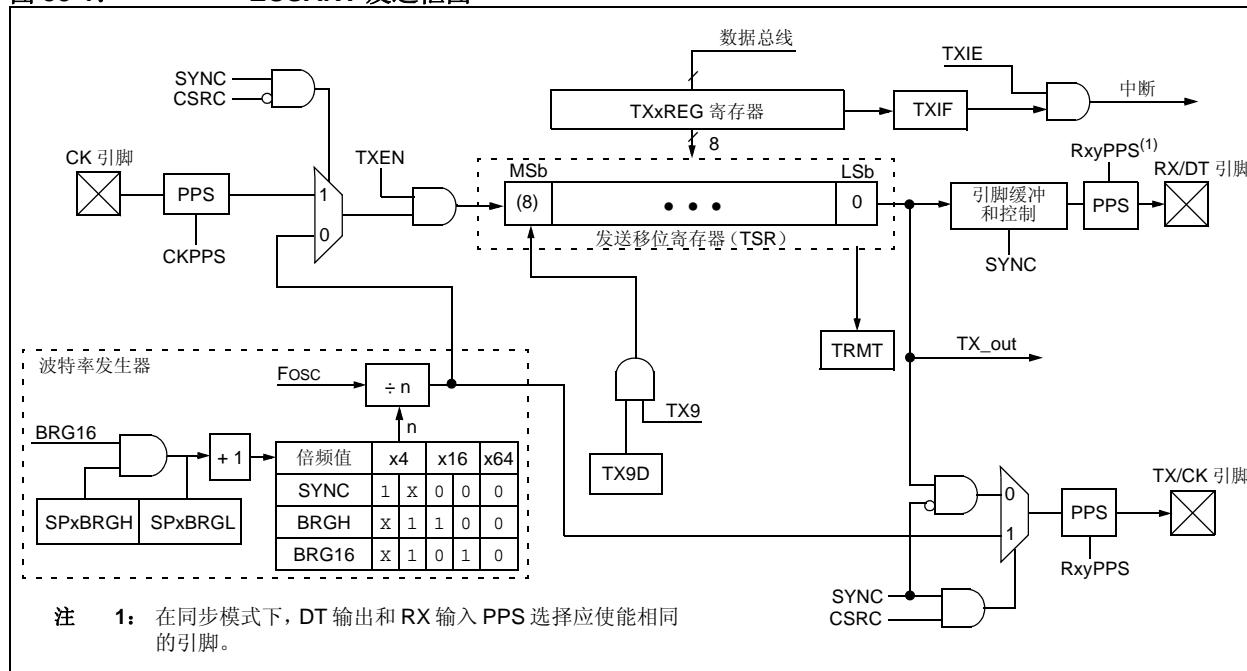
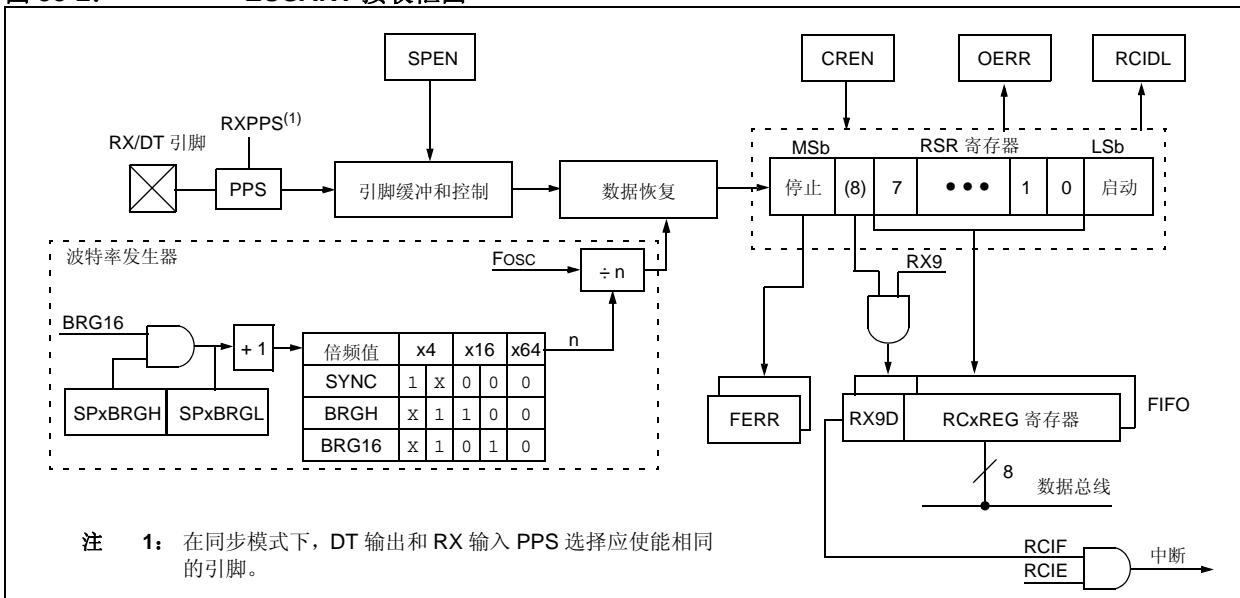


图 33-2: EUSART 接收框图



EUSART 模块的操作由以下 3 个寄存器控制:

- 发送状态和控制寄存器 (TXxSTA)
- 接收状态和控制寄存器 (RCxSTA)
- 波特率控制寄存器 (BAUDxCON)

这些寄存器的详细信息请分别参见[寄存器 33-1](#)、[寄存器 33-2](#) 和[寄存器 33-3](#)。

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 RxPPS 寄存器进行选择。由于在同步模式下 RX 输入与 DT 输出耦合, 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。EUSART 控制逻辑将自动控制数据方向驱动器。

33.1 EUSART 异步模式

EUSART 采用标准不归零 (non-return-to-zero, NRZ) 格式发送和接收数据。NRZ 实现为两种电平: VOH 标记状态 (mark state) 代表 “1” 数据位, 而 VOL 空格状态 (space state) 代表 “0” 数据位。” NRZ 指的是连续发送的具有相同值的数据位, 保持在位的输出电平, 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。每个发送位保持时间为 $1/(\text{波特率})$ 。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 33-5。

EUSART 先发送和接收 Lsb。EUSART 的发送器和接收器在功能上是相互独立的, 但它们的数据格式和波特率相同。硬件不支持奇偶校验, 但可通过软件实现奇偶校验, 并将奇偶校验位作为第 9 个数据位存储。

33.1.1 EUSART 异步发送器

图 33-1 给出了 EUSART 发送器框图。发送器的核心是串行发送移位寄存器 (Transmit Shift Register, TSR), 该寄存器不可用软件直接访问。TSR 从发送缓冲区 (即 TXxREG 寄存器) 取得数据。

33.1.1.1 使能发送器

EUSART 发送器可通过配置以下 3 个控制位使能为异步操作:

- TXEN = 1
- SYNC = 0
- SPEN = 1

假定所有其他 EUSART 控制位均处于其默认状态。

将 TXxSTA 寄存器的 TXEN 位置 1, 可以使能 EUSART 的发送器电路。清零 TXxSTA 寄存器的 SYNC 位将 EUSART 配置为异步操作。将 RCxSTA 寄存器的 SPEN 位置 1, 可以使能 EUSART 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。

33.1.1.2 发送数据

向 TXxREG 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, TXxREG 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 TXxREG 中, 直到前一个字符的停止位被发送。之后, 在 TXxREG 中等待的字符在停止位发送后 1 个 TCY 内被传送到 TSR 中。TXxREG 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。

33.1.1.3 发送数据极性

可通过 BAUDxCON 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 33.5.1.2 节 “时钟极性”。

33.1.1.4 发送中断标志

只要 EUSART 发送器被使能且 TXxREG 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 TXxREG 中还有一个排队等待发送的新字符时, TXIF 位才被清零。写入 TXxREG 后并不立即清零 TXIF 标志位, 而是在之后的第二个指令周期将其清零。写入 TXxREG 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 TXxREG 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 TXxREG 时应清零 TXIE 中断允许位。

33.1.1.5 TSR 状态

TXxSTA 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 TXxREG 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器中，因此用户无法使用。

33.1.1.6 发送 9 位字符

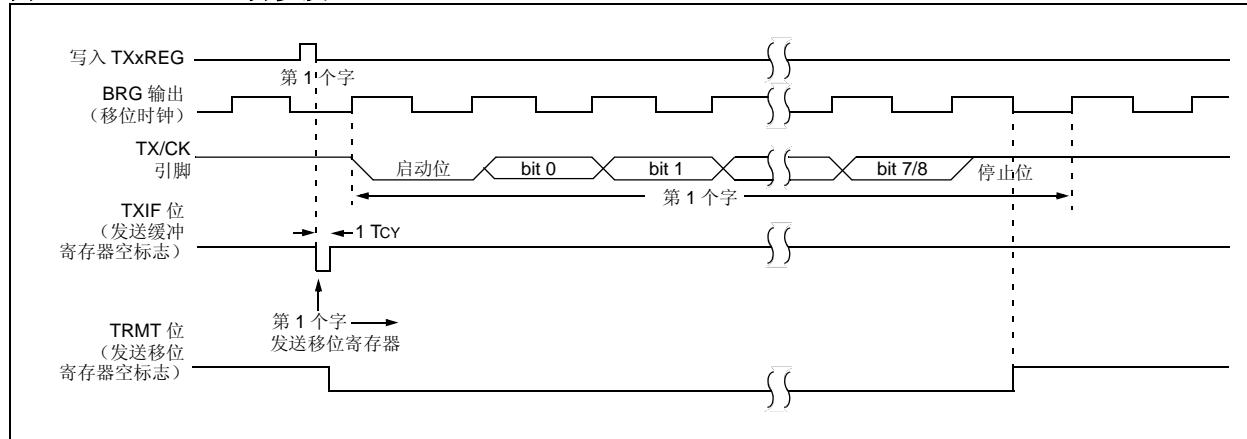
EUSART 支持 9 位字符发送。当 TXxSTA 寄存器的 TX9 位置 1 时，EUSART 将在发送每个字符时移出 9 位。TXxSTA 寄存器的 TX9D 位是第 9 个数据位，也是最高有效位。发送 9 位数据时，TX9D 数据位必须先于低 8 位写入 TXxREG。写入 TXxREG 后，所有 9 个位将被立即传递到 TSR 移位寄存器中。

有多个接收器时，可使用一种特殊的 9 位地址模式。关于地址模式的更多信息，请参见 [第 33.1.2.7 节“地址检测”](#)。

33.1.1.7 异步发送设置

1. 初始化 SPxBRGH 和 SPxBRGL 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率（见 [第 33.4 节“EUSART 波特率发生器（BRG）”](#)）。
2. 通过清零 SYNC 位并将 SPEN 位置 1，使能异步串口。
3. 如果需要 9 位发送，将 TX9 控制位置 1。接收器置于地址检测模式时，第 9 个数据位置 1 表示 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 TXxREG 寄存器。这将启动发送。

图 33-3： 异步发送



PIC16(L)F1764/5/8/9

图 33-4: 异步发送 (连续)

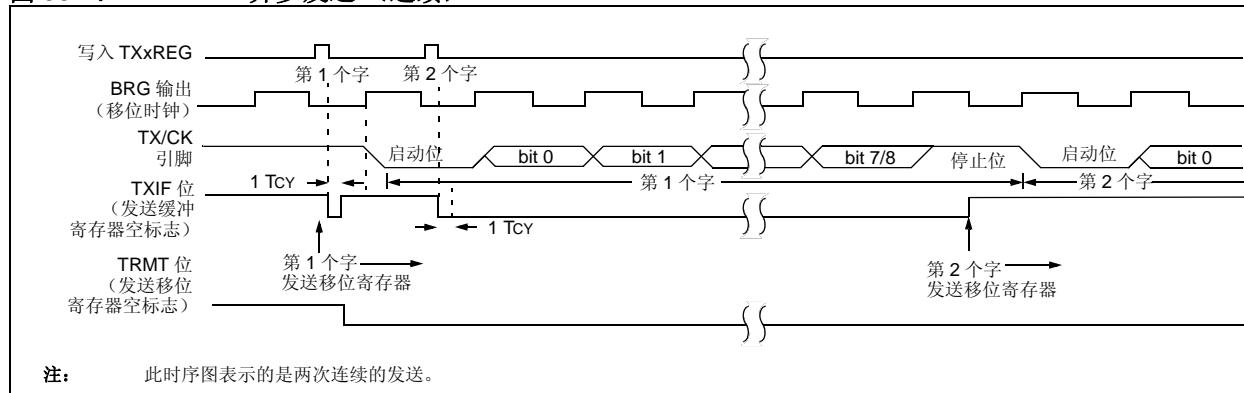


表 33-1: 与异步发送相关的寄存器汇总

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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140					
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146					
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151					
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440					
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102					
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103					
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106					
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439					
RxyPPS	—	—	—	RxyPPS<4:0>					158, 160					
SP1BRGL	BRG<7:0>								441*					
SP1BRGH	BRG<15:8>								441*					
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139					
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145					
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISCO	150					
TX1REG	EUSART 发送数据寄存器								430*					
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438					

图注: — = 未实现位, 读为 0。异步发送不使用阴影单元。

* 提供寄存器信息的页。

注 1: 仅限 PIC16(L)F1768/9。

2: 未实现位, 读为 1。

33.1.2 EUSART 异步接收器

异步模式通常用于 RS-232 系统中。图 33-2 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递到双字符的先进先出（First-In-First-Out, FIFO）存储区中。软件开始处理 EUSART 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 RCxREG 寄存器访问接收数据。

33.1.2.1 使能接收器

EUSART 接收器可通过配置以下 3 个控制位使能为异步操作：

- CREN = 1
- SYNC = 0
- SPEN = 1

假定所有其他 EUSART 控制位均处于其默认状态。

将 RCxSTA 寄存器的 CREN 位置 1 使能 EUSART 的接收器电路。清零 TXxSTA 寄存器的 SYNC 位将 EUSART 配置为异步操作。通过将 RCxSTA 寄存器的 SPEN 位置 1，可使能 EUSART。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。

33.1.2.2 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。数据恢复电路计数传输半个位的时间至启动位的中点并验证该位是否仍为零。如果该位非零，则数据恢复电路中止字符接收，不产生错误，并恢复寻找启动位的下降沿。如果启动位被验证为零，则数据恢复电路计数一个位时间至下个位的中点。该位被一个择多检测电路采样，其结果（0 或 1）被移入 RSR。重复此过程直到所有数据位均被采样并移入 RSR。最后一个位时间被测量且其电平被采样。此位为停止（Stop）位，始终为 1。如果数据恢复电路在停止位处采样到 0，则置 1 此字符的帧错误标志位，否则清零此字符的帧错误标志位。关于帧错误的更多信息，请参见第 33.1.2.4 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递到 EUSART 接收 FIFO，且 PIR1 寄存器的 RCIF 中断标志位被置 1。读取 RCxREG 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 33.1.2.5 节“接收溢出错误”。

33.1.2.3 接收中断

只要 EUSART 接收器被使能且接收 FIFO 中存在未被读取的字符，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。

33.1.2.4 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 RCxSTA 寄存器的 FERR 位可访问帧错误状态。FERR 位表示接收 FIFO 中顶部的未读字符的状态。因此，在读取 RCxREG 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 RCxSTA 寄存器的 SPEN 位清零可复位 EUSART，这样就可将 FERR 位强制清零。将 RCxSTA 寄存器的 CREN 位清零不影响 FERR 位。自身产生的帧错误不会产生中断。

注： 如果接收 FIFO 中的所有接收字符均有帧错误，反复读取 RCxREG 不会将 FERR 位清零。

33.1.2.5 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，RCxSTA 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 RCxSTA 寄存器的 CREN 位清零或通过将 RCxSTA 寄存器的 SPEN 位清零复位 EUSART，可清除该错误。

33.1.2.6 接收 9 位字符

EUSART 支持 9 位字符接收。当 RCxSTA 寄存器的 RX9 位置 1 时，EUSART 将在接收每个字符时将 9 个位移入 RSR。RCxSTA 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 RCxREG 的低 8 位前必须先读取 RX9D 数据位。

33.1.2.7 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 RCxSTA 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。

33.1.2.8 异步接收设置

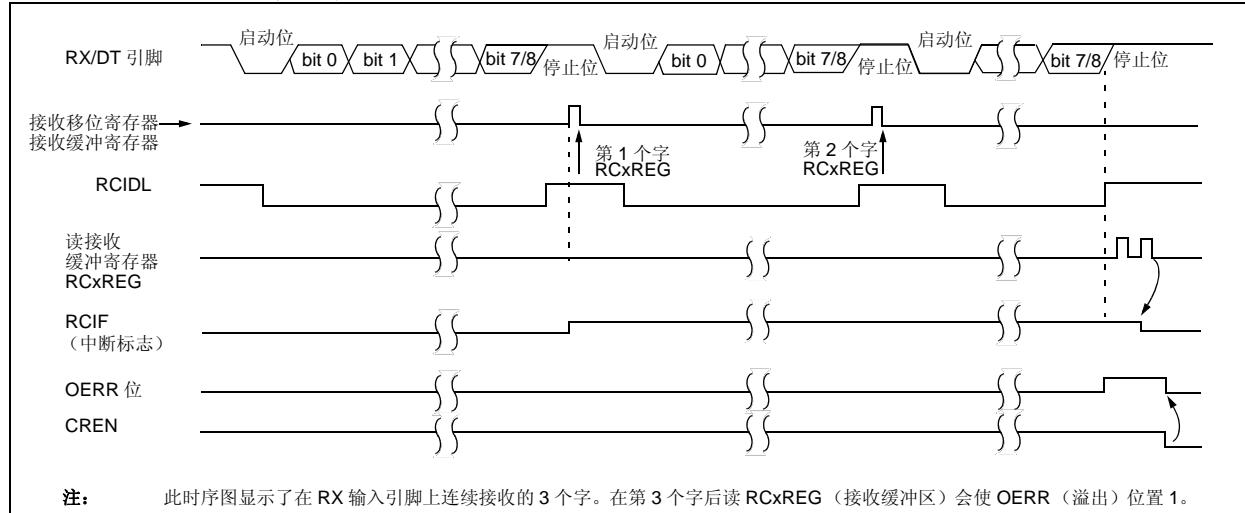
1. 初始化 SPxBRGH 和 SPxBRGL 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率（见第 33.4 节“EUSART 波特率发生器（BRG）”）。
2. 清零 RX 引脚的 ANSEL 位（如适用）。
3. 将 SPEN 位置 1 使能串口。SYNC 位必须清零才能进行异步操作。
4. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 如果需要接收 9 位数据，将 RX9 位置 1。
6. 将 CREN 位置 1 使能接收。
7. 当字符从 RSR 被移入接收缓冲区时，RCIF 中断标志位将被置 1。如果 RCIE 中断允许位也置 1，则产生中断。
8. 读取 RCxSTA 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 RCxREG 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。

33.1.2.9 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 SPxBRGH 和 SPxBRGL 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率（见第 33.4 节“EUSART 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 RCxSTA 寄存器取得错误标志。第 9 个数据位将始终置 1。
10. 读取 RCxREG 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 33-5： 异步接收



PIC16(L)F1764/5/8/9

表 33-2: 与异步接收相关的寄存器汇总

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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1REG	EUSART 接收数据寄存器								433*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RxyPPS	—	—	—	RxyPPS<4:0>					
SP1BRGL	BRG<7:0>								441
SP1BRGH	BRG<15:8>								441
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

图注: — = 未实现位, 读为 0。异步接收不使用阴影单元。

* 提供寄存器信息的页。

注 1: 仅限 PIC16(L)F1768/9。

2: 未实现位, 读为 1。

33.2 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。请参见 更多信息，请参见[第 5.2.2.3 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。自动波特率检测功能可自动完成这种调整（见[第 33.4.1 节“自动波特率检测”](#)）。通过调整波特率发生器来补偿外设时钟频率的逐渐变化时，可能无法足够细微地调节分辨率。

PIC16(L)F1764/5/8/9

33.3 寄存器定义: EUSART 控制

寄存器 33-1: TX1STA: 发送状态和控制寄存器

R/W-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	SEND B	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:

无关位

同步模式:

1 = 主模式 (时钟由内部 BRG 产生)

0 = 从模式 (时钟来自外部时钟源)

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:** EUSART 模式选择位

1 = 同步模式

0 = 异步模式

bit 3 **SEND B:** 发送间隔字符位

异步模式:

1 = 在下一次发送时发送同步间隔字符 (完成后由硬件清零)

0 = 同步间隔字符发送完成

同步模式:

无关位

bit 2 **BRGH:** 高波特率选择位

异步模式:

1 = 高速

0 = 低速

同步模式:

在此模式下未使用

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。

寄存器 33-2: 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-0/0
SPEN	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 = 使能串行口
 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:
 无关位
同步主模式:
 1 = 使能单字节接收
 0 = 禁止单字节接收
 此位在接收完成后清零。
同步从模式
 无关位
- bit 4 **CREN:** 连续接收使能位
异步模式:
 1 = 使能接收器
 0 = 禁止接收器
同步模式:
 1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 的优先级高于 SREN)
 0 = 禁止连续接收
- bit 3 **ADDEN:** 地址检测使能位
9 位异步模式 (RX9 = 1):
 1 = 当 RSR<8> 置 1 时, 使能地址检测, 允许中断并装入接收缓冲区
 0 = 禁止地址检测, 接收所有字节并且第 9 位可作为奇偶校验位
8 位异步模式 (RX9 = 0):
 无关位
- bit 2 **FERR:** 帧错误位
 1 = 帧错误 (可以通过读 RCxREG 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。

PIC16(L)F1764/5/8/9

寄存器 33-3: 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:** 同步时钟极性选择位

异步模式:

1 = 将反相数据发送到 TX/CK 引脚

0 = 将未反相数据发送到 TX/CK 引脚

同步模式:

1 = 数据在时钟上升沿改变

0 = 数据在时钟下降沿改变

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 = 接收器正在等待下降沿。不会接收到任何字符, 但 RCIF 位将被置 1。RCIF 置 1 后 WUE 将被自动清零。

0 = 接收器正常工作

同步模式:

无关位

bit 0 **ABDEN:** 自动波特率检测使能位

异步模式:

1 = 使能自动波特率模式 (完成自动波特率检测后清零)

0 = 禁止自动波特率模式

同步模式:

无关位

33.4 EUSART 波特率发生器 (BRG)

波特率发生器 (BRG) 是 8 位或 16 位定时器，专用于支持异步和同步 EUSART 操作。默认情况下，BRG 工作在 8 位模式下。将 BAUDxCON 寄存器的 BRG16 位置 1 可选择 16 位模式。

SPxBRGH 和 SPxBRGL 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 TXxSTA 寄存器的 BRGH 位和 BAUDxCON 寄存器的 BRG16 位决定。在同步模式下，BRGH 位被忽略。

表 33-3 提供了确定波特率的公式。例 33-1 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 33-5 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 SPxBRGH 和 SPxBRGL 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 33-1：计算波特率误差

针对工作在异步模式下， $F_{OSC} = 16\text{ MHz}$ ，目标波特率 = 9600 且采用 8 位 BRG 的器件：

$$\text{目标波特率} = \frac{F_{OSC}}{64([SPBRGH:SPBRGL] + 1)}$$

求解 SPxBRGH:SPxBRGL：

$$\begin{aligned} X &= \frac{\frac{F_{OSC}}{\text{目标波特率}} - 1}{64} \\ &= \frac{\frac{16000000}{9600} - 1}{64} \\ &= [25.042] = 25 \end{aligned}$$

$$\text{计算波特率} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\begin{aligned} \text{误差} &= \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}} \\ &= \frac{(9615 - 9600)}{9600} = 0.16\% \end{aligned}$$

PIC16(L)F1764/5/8/9

表 33-3: 波特率公式

配置位			BRG/EUSART 模式	波特率公式
SYNC	BRG16	BRGH		
0	0	0	8 位 / 异步	Fosc/[64 (n+1)]
0	0	1	8 位 / 异步	Fosc/[16 (n+1)]
0	1	0	16 位 / 异步	
0	1	1	16 位 / 异步	Fosc/[4 (n+1)]
1	0	x	8 位 / 同步	
1	1	x	16 位 / 同步	

图注: x = 无关位, n = SPxBRGH 和 SPxBRGL 寄存器对的值。

表 33-4: 与波特率发生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
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
SP1BRGL	BRG<7:0>								441
SP1BRGH	BRG<15:8>								441
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

图注: — = 未实现位, 读为 0。波特率发生器不使用阴影单元。

* 提供寄存器信息的页。

表 33-5: 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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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	—	—	—	—	—	—	—	—	—	—	—	—

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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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	117k	2.12	16	113.64	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

PIC16(L)F1764/5/8/9

表 33-5： 异步模式下的波特率（续）

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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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

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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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	—	—	—

表 33-5：异步模式下的波特率（续）

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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 值 (十进制)	实际 波特率	% 误差	SPxBRG 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	—	—	—

33.4.1 自动波特率检测

EUSART 模块支持波特率自动检测和校准。

在自动波特率检测（Auto-Baud Rate Detect, ABD）模式下，提供给 BRG 的时钟信号是反向的。BRG 并不为传入的 RX 信号提供时钟信号，而是由 RX 信号为 BRG 定时。波特率发生器用于为接收的 55h (ASCII “U”) 的周期定时，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 BAUDxCON 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，EUSART 状态机保持在空闲状态。在接收线的第一个上升沿（启动位之后），SPxBRG 使用 BRG 计数器时钟递增计数，如图 33-6 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，累计值即正确的 BRG 周期总数被保存在 SPxBRGH 和 SPxBRGL 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 RCxREG 中的值。RCxREG 的内容应该被丢弃。在不使用 SPxBRGH 寄存器的模式下进行校准时，用户可通过检查 SPxBRGH 寄存器的值是否为 00h 来验证 SPxBRGL 寄存器是否溢出。

BRG 自动波特率时钟由 BRG16 和 BRGH 位决定，如表 33-6 所示。在 ABD 期间，SPxBRGH 和 SPxBRGL 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，SPxBRGH 和 SPxBRGL 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。

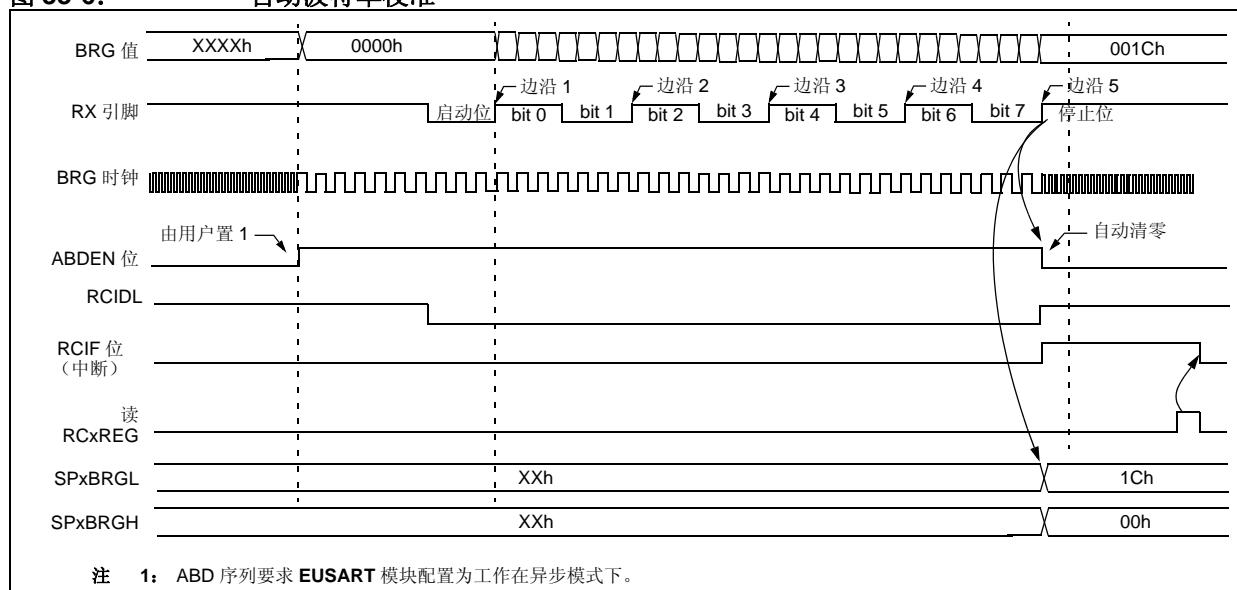
- 注 1:** 如果 WUE 位和 ABDEN 位都置 1，自动波特率检测将从间隔字符之后的字节开始（见第 33.4.3 节“接收到间隔字符时自动唤醒”）。
- 2:** 需要由用户来判断输入字符的波特率是否处于所选 BRG 时钟源范围内。可能无法实现某些振荡器频率和 EUSART 波特率组合。
- 3:** 在自动波特率过程中，自动波特率计数器从 1 开始计数。自动波特率序列完成后，为了得到最准确的结果，应从 SPxBRGH:SPxBRGL 寄存器对的值中减去 1。

表 33-6: 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，SPxBRGL 和 SPxBRGH 寄存器都被用作 16 位计数器，与 BRG16 的设置无关。

图 33-6: 自动波特率校准



33.4.2 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 BAUDCON 寄存器的 ABDOVF 位将被置 1。ABDOVF 位指示计数器已超出 SPxBRGH:SPxBRGL 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 位将保持为假 (0)，直到出现第 5 个上升沿，此时 RCIDL 位会置 1。如果在发生溢出之后但在第 5 个上升沿之前读 RCREG，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 RCREG 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。

33.4.3 接收到间隔字符时自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于不工作状态，不能正常进行字符接收。自动唤醒功能使控制器可被 RX/DT 线上的活动唤醒。该功能只在异步模式下可用。

自动唤醒功能可通过将 BAUDCON 寄存器的 WUE 位置 1 来使能。一旦置 1，RX/DT 上的正常接收序列就被禁止，EUSART 保持在空闲状态，监视与 CPU 模式无关的唤醒事件。唤醒事件包含 RX/DT 线上电平由高至低的跳变。(这与同步间隔字符或 LIN 协议的唤醒信号字符的启动条件一致。)

EUSART 模块产生的 RCIF 中断与唤醒事件同时发生。在正常 CPU 工作模式下，与 Q 时钟同步产生中断（图 33-7）；在休眠模式下，与 Q 时钟异步产生中断（图 33-8）。通过读 RCRxEG 寄存器可清除中断条件。

RX 线在间隔字符末尾由低至高的跳变将自动清零 WUE 位。这向用户表明间隔事件结束。此时，EUSART 模块处于空闲模式，等待接收下一个字符。

33.4.3.1 特殊注意事项

间隔字符

在发生唤醒事件期间为了避免字符错误或字符碎片，唤醒字符必须为全零。

唤醒被使能时，其工作状况与数据流的低电平时间无关。如果 WUE 位置 1 并接收到有效的非零字符，则从启动位至第一个上升沿的低电平时间将被解读为唤醒事件。字符的其余位将作为碎片字符接收，后续字符有可能产生帧错误或溢出错误。

因此，发送的首字符必须为全 0。这必须持续 10 个或更长的位时间，对于 LIN 总线，建议持续 13 个位时间，而对于标准 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 EUSART 进行正确初始化。

WUE 位

唤醒事件会通过将 RCIF 位置 1 产生一个接收中断。WUE 位在 RX/DT 的上升沿由硬件清零。之后软件通过读取 RCxREG 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)F1764/5/8/9

图 33-7：正常工作时的自动唤醒位（WUE）时序

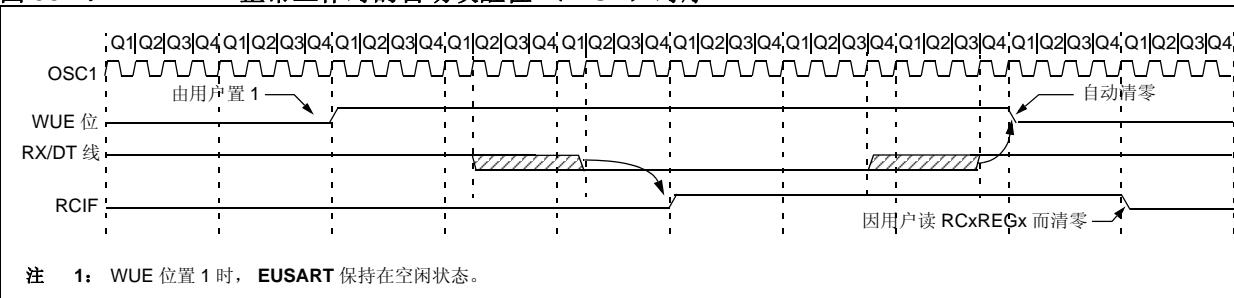
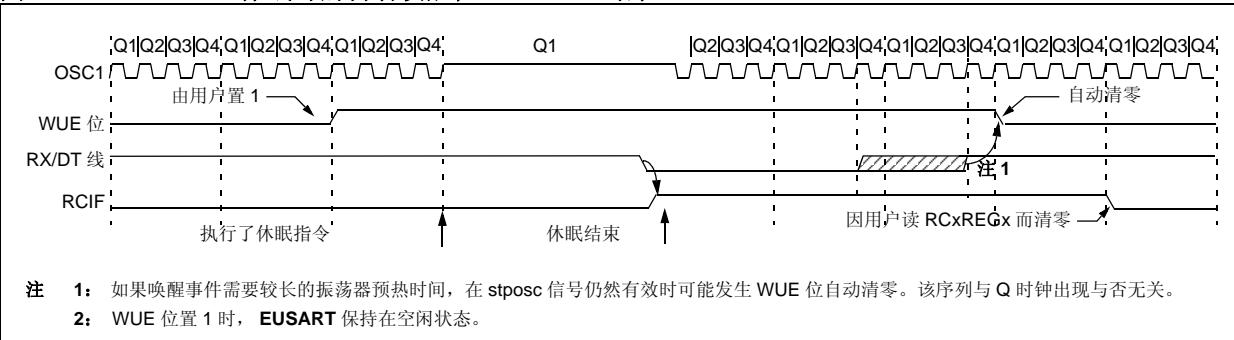


图 33-8：休眠时的自动唤醒位（WUE）时序



33.4.4 间隔字符序列

EUSART 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。间隔字符包含 1 个启动位以及随后的 12 个 0 位和 1 个停止位。

要发送间隔字符，应将 TXxSTA 寄存器的 SENDB 和 TXEN 位置 1。随后对 TXxREG 执行写操作可启动间隔字符发送。写入 TXxREG 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。

TXxSTA 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 33-9 给出了发送间隔字符的时序。

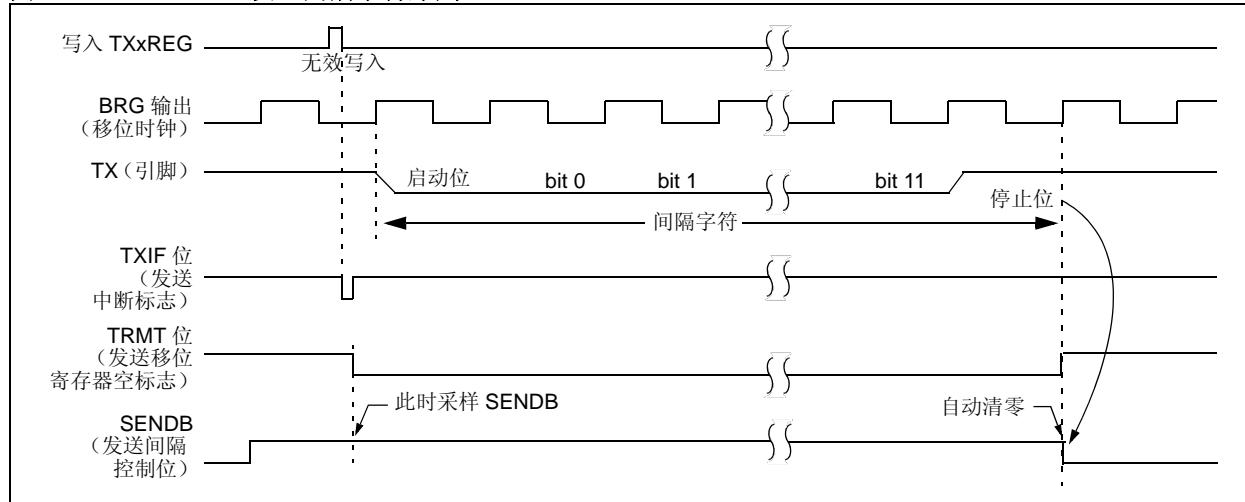
33.4.4.1 间隔和同步发送序列

以下序列将启动报文帧头，它由间隔字符和其后的自动波特率同步字节组成。这是 LIN 总线主器件的典型序列。

1. 将 EUSART 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置 1 使能间隔序列。
3. 将无效字符装入 TXxREG，启动发送（该值会被忽略）。
4. 将 “55h” 写入 TXxREG，以便将同步字符装入发送 FIFO 缓冲区。
5. 发送间隔字符后，SENDB 位被硬件复位，同步字符随后被发送。

当 TXxREG 为空时（由 TXIF 指示），下一个数据字节会写入 TXxREG。

图 33-9：发送间隔字符序列



33.4.5 接收间隔字符

增强型 EUSART 模块接收间隔字符有两种方法。

第一种检测间隔字符的方法采用 RCxSTA 寄存器的 FERR 位和如 RCxREG 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
- RCxREG = 00h

第二种方法采用第 33.4.3 节“接收到间隔字符时自动唤醒”中所述的自动唤醒功能。通过使能此功能，EUSART 将采样 RX/DT 上的下两次跳变，产生 RCIF 中断，并接收下一个数据字节并再产生一次中断。

请注意，在间隔字符后，用户通常希望使能自动波特率检测功能。采用这两种方法时，用户均可在 EUSART 进入休眠模式前将 BAUDxCON 寄存器的 ABDEN 位置 1。

33.5 EUSART 同步模式

同步串行通信通常用于具有一个主器件和一个或多个从器件的系统中。主器件包含生成波特率所需的电路，可将时钟提供给系统中的所有器件。从器件使用主时钟，可不再需要内部时钟生成电路。

同步模式下有两条信号线：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。EUSART 可作为主器件，也可作为从器件。

同步发送时不使用启动位和停止位。

33.5.1 同步主模式

使用以下位将 EUSART 配置为同步主操作：

- SYNC = 1
- CSRC = 1
- SREN = 0 (用于发送) ; SREN = 1 (用于接收)
- CREN = 0 (用于发送) ; CREN = 1 (用于接收)
- SPEN = 1

通过将 TXxSTA 寄存器的 SYNC 位置 1，可将器件配置为同步操作。通过将 TXxSTA 寄存器的 CSRC 位置 1，可将器件配置为主器件。通过将 RCxSTA 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 RCxSTA 寄存器的 SPEN 位置 1，可使能 EUSART。

33.5.1.1 主时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为主器件的器件将时钟信号发送到 TX/CK 线上。EUSART 配置为同步发送或接收操作时，自动使能 TX/CK 引脚输出驱动器。串行数据位在时钟前沿改变，以确保其在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。

33.5.1.2 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 BAUDxCON 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。

33.5.1.3 同步主发送

从器件的 RX/DI 引脚发送出数据。EUSART 配置为同步主发送操作时，RX/DI 和 TX/CK 引脚的输出驱动器被自动使能。

向 TXxREG 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 TXxREG 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，TXxREG 中的数据就立即被传送到 TSR。字符发送在数据从 TXxREG 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器中，因此用户无法使用。

33.5.1.4 同步主发送设置

1. 初始化 SPxBRGH 和 SPxBRGL 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率（见第 33.4 节“EUSART 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 TXxREG 寄存器，启动发送。

图 33-10: 同步发送

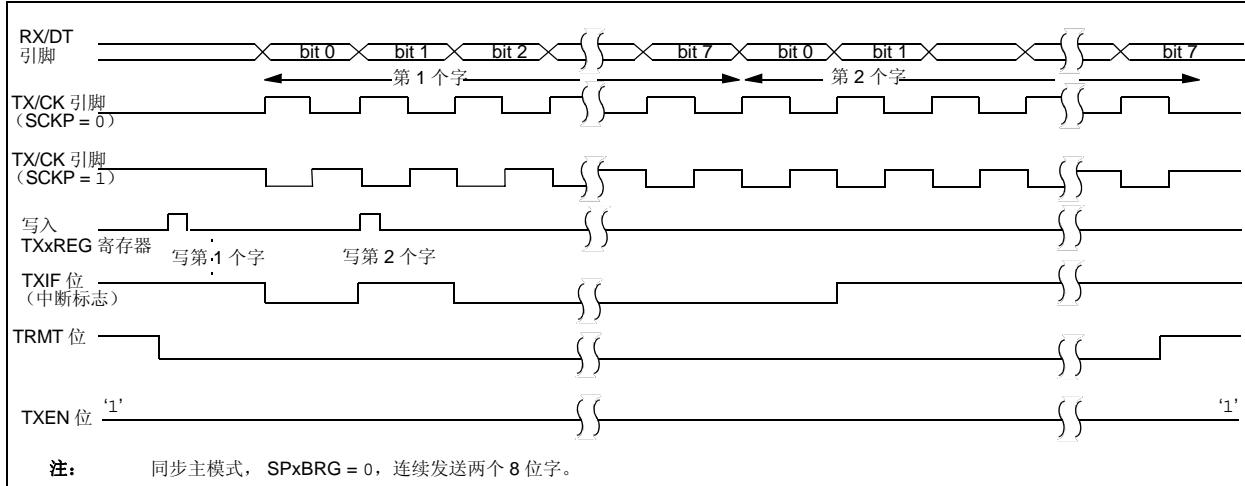
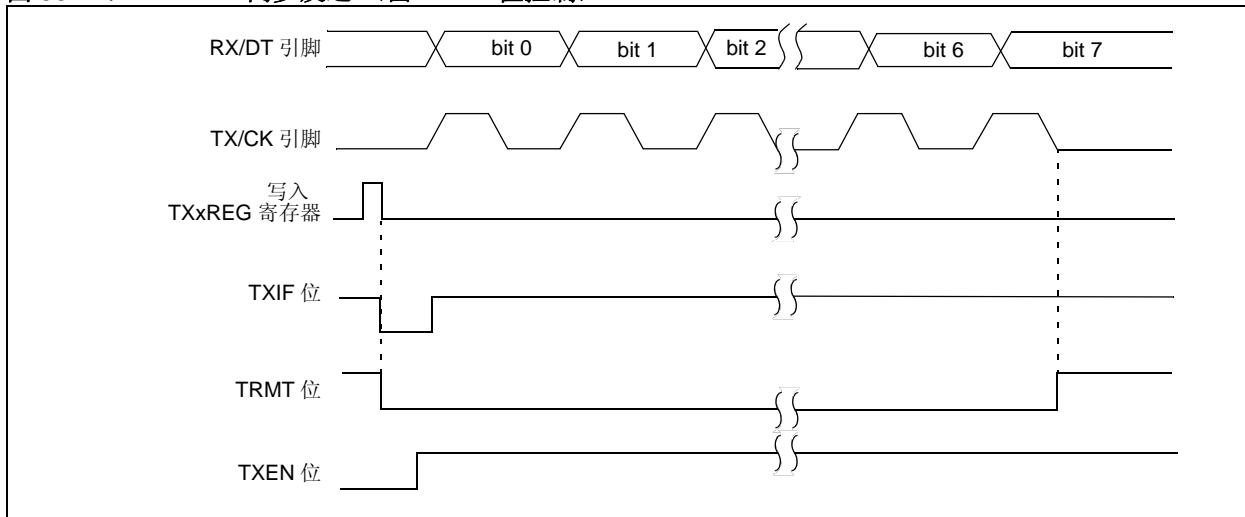


图 33-11: 同步发送 (由 TXEN 位控制)



PIC16(L)F1764/5/8/9

表 33-7：与同步主发送相关的寄存器汇总

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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140					
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146					
ANSEL C	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151					
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440					
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102					
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103					
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106					
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439					
RxyPPS	—	—	—	RxyPPS<4:0>					158					
SP1BRGL	BRG<7:0>								441					
SP1BRGH	BRG<15:8>								441					
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139					
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145					
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150					
TX1REG	EUSART 发送数据寄存器								430*					
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438					

图注：— = 未实现位，读为 0。同步主发送不使用阴影单元。

* 提供寄存器信息的页。

注 1：仅限 PIC16(L)F1768/9。

2：未实现位，读为 1。

33.5.1.5 同步主接收

数据在 RX/DT 引脚上接收。将 EUSART 配置为同步主接收操作时，自动禁止 RX/DT 引脚输出驱动器。

在同步模式下，可通过将单字符接收使能位（RCxSTA 寄存器的 SREN）或连续接收使能位（RCxSTA 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 RCxREG 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。

33.5.1.6 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，就接收多少个时钟周期。

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。

33.5.1.7 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。在 RCxREG 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，RCxSTA 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 RCxREG 可清除错误。如果 CREN 位置 1 时发生溢出错误，则通过清零 RCxSTA 寄存器的 CREN 位或清零 SPEN 位（该位清零将 EUSART 复位），可以清除错误条件。

33.5.1.8 接收 9 位字符

EUSART 支持 9 位字符接收。当 RCxSTA 寄存器的 RX9 位置 1 时，EUSART 将在接收每个字符时将 9 个位移入 RSR。RCxSTA 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 RCxREG 的低 8 位前必须先读取 RX9D 数据位。

33.5.1.9 同步主接收设置

1. 根据相应的波特率初始化 SPxBRGH 和 SPxBRGL 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，以便能同步主串口。
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 RCxSTA 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 RCxREG 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 RCxSTA 寄存器的 CREN 位或清零 SPEN 位（该位清零将 EUSART 复位），可以清除错误。

PIC16(L)F1764/5/8/9

图 33-12: 同步接收 (主模式, SREN)

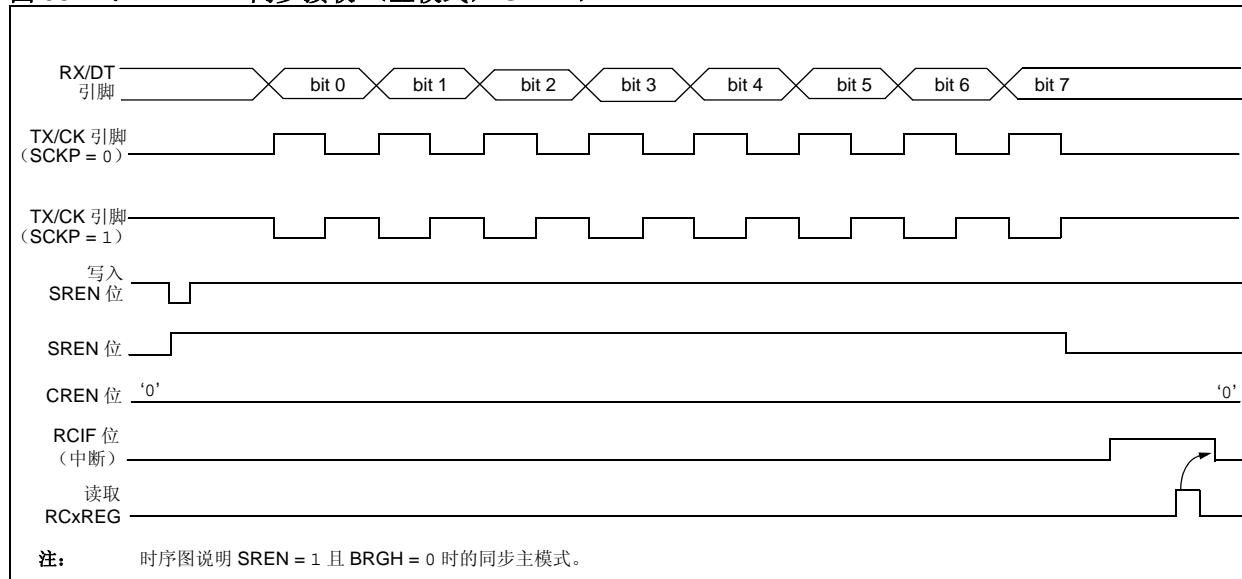


表 33-8: 与同步主接收相关的寄存器汇总

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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140					
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146					
ANSELc	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151					
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440					
CKPPS	—	—	—	CKPPS<4:0>					158, 160					
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102					
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103					
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106					
RC1REG	EUSART 接收数据寄存器								433*					
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439					
RXPPS	—	—	—	RXPPS<4:0>					158, 160					
RxyPPS	—	—	—	RxyPPS<4:0>					158					
SP1BRGL	BRG<7:0>								441*					
SP1BRGH	BRG<15:8>								441*					
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139					
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145					
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150					
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438					

图注: — = 未实现位, 读为 0。同步主接收不使用阴影单元。

* 提供寄存器信息的页。

注 1: 仅限 PIC16(L)F1768/9。

2: 未实现位, 读为 1。

33.5.2 同步从模式

使用以下位将 EUSART 配置为同步从操作：

- SYNC = 1
- CSRC = 0
- SREN = 0 (用于发送); SREN = 1 (用于接收)
- CREN = 0 (用于发送); CREN = 1 (用于接收)
- SPEN = 1

通过将 TXxSTA 寄存器的 SYNC 位置 1，可将器件配置为同步操作。通过将 TXxSTA 寄存器的 CSRC 位清零，可将器件配置为从器件。通过将 RCxSTA 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 RCxSTA 寄存器的 SPEN 位置 1，可使能 EUSART。

33.5.2.1 EUSART 同步从发送

除了休眠模式以外，同步主模式和从模式的工作是相同的（见 [第 33.5.1.2 节 “时钟极性”](#)）。

如果向 TXxREG 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. 第二个字符将保留在 TXxREG 寄存器中。
3. TXIF 位不会被置 1。
4. 第一个字符移出 TSR 后，TXxREG 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。

33.5.2.2 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 TXxREG 寄存器，启动发送。

PIC16(L)F1764/5/8/9

表 33-9：与同步从发送相关的寄存器汇总

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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
CKPPS	—	—	—	CKPPS<4:0>					158, 160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RXPPS	—	—	—	RXPPS<4:0>					158, 160
RxyPPS	—	—	—	RxyPPS<4:0>					158
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1REG	EUSART 发送数据寄存器							430*	
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

图注：— = 未实现位，读为 0。同步从发送不使用阴影单元。

* 提供寄存器信息的页。

注 1：仅限 PIC16(L)F1768/9。

2：未实现位，读为 1。

33.5.2.3 EUSART 同步从接收

除下列各项外，同步主模式和从模式的工作是相同的（[第 33.5.1.5 节 “同步主接收”](#)）：

- 休眠
- CREN 位始终置 1，因此接收器从不空闲
- SREN 位在从模式下为“无关位”

进入休眠前将 CREN 位置 1，可在休眠模式下接收一个字符。接收到数据字后，RSR 寄存器会将数据传送到 RCxREG 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。

33.5.2.4 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，以便能接收。
6. 接收完成时 RCIF 位将被置 1。如果 RCIE 位已置 1，则产生中断。
7. 如果使能了 9 位模式，从 RCSTA 寄存器的 RX9D 位取出最高有效位。
8. 读取 RCxREG 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 RCxSTA 寄存器的 CREN 位或清零 SPEN 位（清零该位将 EUSART 复位），可以清除错误。

表 33-10：与同步从接收相关的寄存器汇总

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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	140
ANSELB ⁽¹⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	146
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	151
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	440
CKPPS	—	—	—	CKPPS<4:0>					158, 160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	102
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	106
RC1REG	EUSART 接收数据寄存器								433*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	439
RXPPS	—	—	—	RXPPS<4:0>					158, 160
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	139
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	145
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	150
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	438

图注：— = 未实现位，读为 0。同步从接收不使用阴影单元。

* 提供寄存器信息的页。

注 1：仅限 PIC16(L)F1768/9。

2：未实现位，读为 1。

33.6 休眠期间的 EUSART 操作

EUSART 只有在同步从模式下，才会在休眠模式下保持工作状态。所有其他模式都需要系统时钟，因此在休眠模式下无法产生运行发送或接收移位寄存器必需的信号。

同步从模式使用外部产生的时钟运行发送和接收移位寄存器。

33.6.1 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：

- RCxSTA 和 TXxSTA 控制寄存器必须配置为同步从接收（见第 33.5.2.4 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 RCxREG 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。

33.6.2 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：

- RCxSTA 和 TXxSTA 控制寄存器必须配置为同步从发送（请参见第 33.5.2.2 节“同步从发送设置”）。
- 必须通过将输出数据写入 TXxREG 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，TXxREG 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，TXxREG 可接收其他字符进行发送，此操作将清零 TXIF 标志位。

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。

34.0 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
- MCLR/VPP
- VDD
- VSS

在编程 / 校验模式下，通过串行通信对程序存储器、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)F170X Memory Programming Specification” (DS40001683)。

34.1 高电压编程模式

通过将 ICSPCLK 和 ICSPDAT 引脚保持为低电平，然后将 MCLR/VPP 上的电压升至 V_{IH} ，将器件置于高电压编程模式。

34.2 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。

进入低电压编程模式需要执行以下步骤：

1. 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。

完成密钥序列后，在需要维持编程/校验模式的时间内，必须将 MCLR 保持为 V_{IL} 。

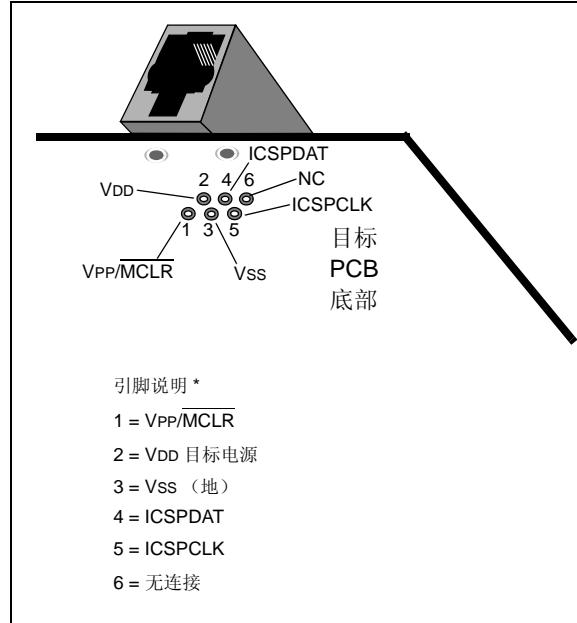
如果使能了低电压编程 ($LVP = 1$)，则 MCLR 复位功能会被自动使能，无法禁止。更多信息，请参见第 6.5 节 “MCLR”。

LVP 位只能通过使用高压编程模式重新设定为 0。

34.3 常用编程接口

与目标器件的连接通常通过 ICSP™ 插头来实现。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 34-1。

图 34-1： ICD RJ-11 型连接器接口



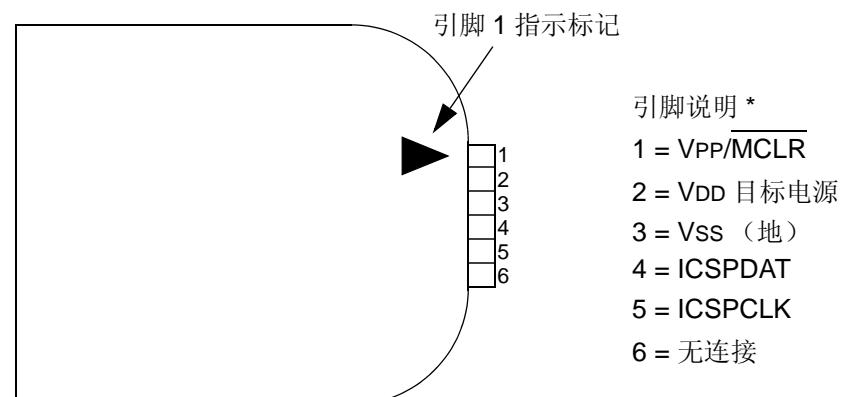
另一种常用于 PICkit™ 编程器的连接器是间距为 0.1 英寸的标准 6 引脚插头。请参见图 34-2。

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元件。更多信息，请参见图 34-3。

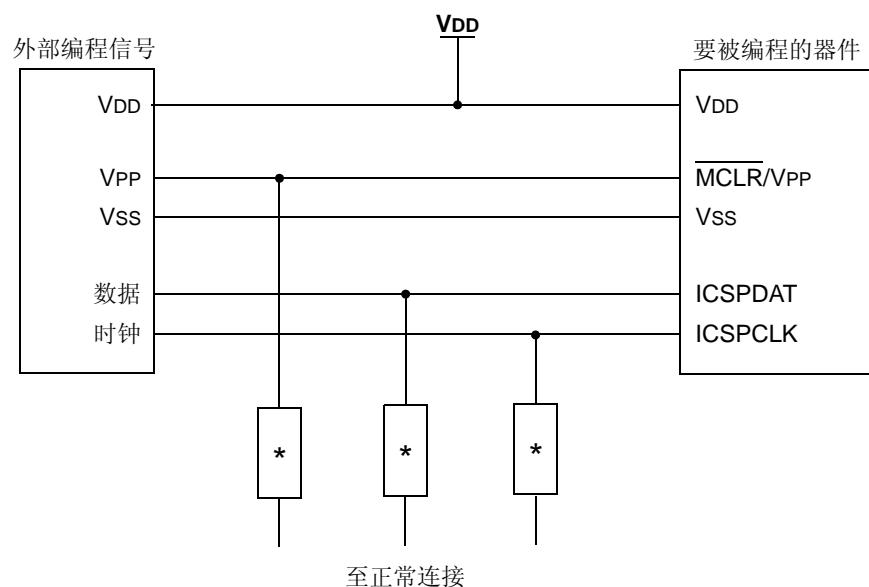
PIC16(L)F1764/5/8/9

图 34-2: PICkit™ 编程器型连接器接口



* 此 6 引脚插头 (0.100" 间距) 可连接 0.025" 的方形引脚。

图 34-3: ICSP™ 编程的典型连接



* 隔离器件 (根据需要而定)。

35.0 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 35-3 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：

- 子程序指令需要两个周期（CALL 和 CALLW）
- 中断或子程序返回指令需要两个周期（RETURN、RETLW 和 RETFIE）
- 程序跳转指令需要两个周期（GOTO、BRA、BRW、BTFSZ、BTFSZ、DECFSZ 和 INCFSZ）
- 当任意指令引用某个间接文件寄存器，并且文件选择寄存器指向程序存储器时，将需要使用一个额外的指令周期。

一个指令周期包含 4 个振荡器周期；振荡器频率为 4 MHz 时，得到的标称指令执行速率为 1 MHz。

所有指令示例均使用格式 “0xhh” 来表示一个十六进制数，其中 “h” 表示一个十六进制数字。

35.1 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指令都进行读 - 修改 - 写（Read-Modify-Write, R-M-W）操作。根据指令或目标标识符 “d” 读寄存器、修改数据和存储结果。即使指令写入寄存器，还是会执行对该寄存器的读操作。

表 35-1：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。默认值 d = 1。
n	FSR 或 INDF 编号。（0-1）
mm	预 / 后递增 / 递减模式选择

表 35-2：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

PIC16(L)F1764/5/8/9

图 35-1： 指令的通用格式

针对字节的文件寄存器操作类指令

13	8	7	6	0
操作码	d	f (寄存器地址)		

d = 0, 结果存入 W

d = 1, 结果存入 f

f = 7 位文件寄存器地址

针对位的文件寄存器操作类指令

13	10	9	7	6	0
操作码	b (位地址)	f (寄存器地址)			

b = 3 位位地址

f = 7 位文件寄存器地址

立即数和控制操作类指令

一般格式

13	8	7	0
操作码		k (立即数)	

k = 8 位立即数值

仅限 CALL 和 GOTO 指令

13	11	10	0
操作码		k (立即数)	

k = 11 位立即数值

仅限 MOVLW 指令

13	7	6	0
操作码		k (立即数)	

k = 7 位立即数值

仅限 MOVLB 指令

13	5	4	0
操作码		k (立即数)	

k = 5 位立即数值

仅限 BRA 指令

13	9	8	0
操作码		k (立即数)	

k = 9 位立即数值

FSR 偏移指令

13	7	6	5	0
操作码	n		k (立即数)	

n = 相应的 FSR

k = 6 位立即数值

FSR 递增指令

13	3	2	1	0
操作码	n	m(模式)		

n = 相应的 FSR

m = 2 位模式值

仅限操作码

13	0
操作码	

表 35-3: PIC16(L)F1764/5/8/9 指令集

助记符, 操作数	说明	周期数	14 位操作码				受影响的 状态位	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	ffff	ffff	C、DC 和 Z	2
ADDWFC f, d	W 与 f 相加 (带进位)	1	11	1101	ffff	ffff	C、DC 和 Z	2
ANDWF f, d	W 和 f 作逻辑与运算	1	00	0101	ffff	ffff	Z	2
ASRF f, d	算术右移	1	11	0111	ffff	ffff	C 和 Z	2
LSLF f, d	逻辑左移	1	11	0101	ffff	ffff	C 和 Z	2
LSRF f, d	逻辑右移	1	11	0110	ffff	ffff	C 和 Z	2
CLRF f	将 f 清零	1	00	0001	ffff	ffff	Z	2
CLRW -	将 W 清零	1	00	0001	0000	00xx	Z	
COMF f, d	对 f 取反	1	00	1001	ffff	ffff	Z	2
DECf f, d	f 递减 1	1	00	0011	ffff	ffff	Z	2
INCf f, d	f 递增 1	1	00	1010	ffff	ffff	Z	2
IORWF f, d	W 和 f 作逻辑或运算	1	00	0100	ffff	ffff	Z	2
MOVF f, d	传送 f	1	00	1000	ffff	ffff	Z	2
MOVWF f	将 W 的内容传送到 f	1	00	0000	ffff	ffff		2
RLF f, d	f 带进位循环左移	1	00	1101	ffff	ffff	C	2
RRF f, d	f 带进位循环右移	1	00	1100	ffff	ffff	C	2
SUBWF f, d	f 减去 W	1	00	0010	ffff	ffff	C、DC 和 Z	2
SUBWFB f, d	f 减去 W (带借位)	1	11	1011	ffff	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	ffff	ffff		2
XORWF f, d	W 和 f 作逻辑异或运算	1	00	0110	ffff	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	ffff	ffff		1, 2
INCFSZ f, d	f 递增 1, 为 0 则跳过	1 (2)	00	1111	ffff	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								
BTFS f, b	测试 f 中的某位, 为 0 则跳过	1 (2)	01	10bb	bfff	ffff		1, 2
BTFF 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	立即数和 W 作逻辑与运算	1	11	1001	kkkk	kkkk	Z	
IORLW k	立即数和 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	立即数和 W 作逻辑异或运算	1	11	1010	kkkk	kkkk	Z	

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。

PIC16(L)F1764/5/8/9

表 35-3: PIC16(L)F1764/5/8/9 指令集 (续)

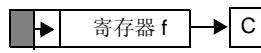
助记符, 操作数	说明	周期数	14 位操作码				受影响的 状态	注	
			MSb		Lsb				
控制操作类指令									
BRA k	相对跳转	2	11	001k	kkkk	kkkk			
BRW -	使用 W 进行相对跳转	2	00	0000	0000	1011			
CALL k	调用子程序	2	10	0kkk	kkkk	kkkk			
CALLW -	使用 W 调用子程序	2	00	0000	0000	1010			
GOTO k	跳转到地址	2	10	1kkk	kkkk	kkkk			
RETFIE k	从中断返回	2	00	0000	0000	1001			
RETLW 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			
固有操作类指令									
CLRWDT -	将看门狗定时器清零	1	00	0000	0110	0100	TO 和 PD		
NOP -	空操作	1	00	0000	0000	0000			
OPTION -	将 W 的内容装入 OPTION_REG 寄存器	1	00	0000	0110	0010			
RESET -	软件器件复位	1	00	0000	0000	0001			
SLEEP -	进入待机模式	1	00	0000	0110	0011			
TRIS f	将 W 的内容装入 TRIS 寄存器	1	00	0000	0110	0fff	TO 和 PD		
针对 C 编译器优化的指令									
ADDFSR n, k	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	
	将 INDFn 送入 W, 间接变址寻址	1	11	1111	0nkk	kkkk		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	1nmm		2, 3	
	将 W 送入 INDFn, 间接变址寻址	1	11	1111	1nkk	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。

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: 请参见 MOVIW 和 MOVWI 指令说明表。

35.2 指令说明

ADDSR	将立即数加到 FSRn	ANDLW	立即数和 W 作逻辑与运算
语法:	[标号] ADDSR FSRn, k	语法:	[标号] ANDLW k
操作数:	-32 ≤ k ≤ 31	操作数:	0 ≤ k ≤ 255
	n ∈ [0, 1]		
操作:	FSR(n) + k → FSR(n)	操作:	(W) .AND.(k) → (W)
受影响的状态位:	无	受影响的状态位:	Z
说明:	将有符号 6 位立即数 k 加到 FSRnH:FSRnL 寄存器对的内容。	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。
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	ANDWF	W 和 f 作逻辑与运算
语法:	[标号] ADDLW k	语法:	[标号] ANDWF f,d
操作数:	0 ≤ k ≤ 255	操作数:	0 ≤ f ≤ 127
			d ∈ [0,1]
操作:	(W) + k → (W)	操作:	(W) .AND.(f) → (目标寄存器)
受影响的状态位:	C、DC 和 Z	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 相加, 结果存入 W 寄存器。	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。
ADDWF	W 与 f 相加	ASRF	算术右移
语法:	[标号] ADDWF f,d	语法:	[标号] ASRF f {,d}
操作数:	0 ≤ f ≤ 127	操作数:	0 ≤ f ≤ 127
	d ∈ [0,1]		d ∈ [0,1]
操作:	(W) + (f) → (目标寄存器)	操作:	(f<7>) → dest<7> (f<7:1>) → dest<6:0> (f<0>) → C
受影响的状态位:	C、DC 和 Z	受影响的状态位:	C 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。	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, 结果存入 W。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 ≤ f ≤ 127		
	d ∈ [0,1]		
操作:	(W) + (f) + (C) → 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, 结果存入 W。如果 d 为 1, 结果存入数据存储单元 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。

BTFS 测试 f 中的某位, 为 0 则跳过

语法:	[标号] BTFS 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 相对跳转

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 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 $\text{PC} + 1 + k$ 。该指令为一条双周期指令。该跳转的地址范围存在限制。

BTFS 测试 f 中的某位, 为 1 则跳过

语法:	[标号] BTFS 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 进行相对跳转

语法:	[标号] BRW
操作数:	无
操作:	$(\text{PC}) + (W) \rightarrow \text{PC}$
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 $\text{PC} + 1 + (W)$ 。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 的 bit b 置 1。

CALL	调用子程序
语法:	[标号] CALL k
操作数:	$0 \leq k \leq 2047$
操作:	$(PC) + 1 \rightarrow TOS$, $k \rightarrow PC<10:0>$, $(PCLATH<6:3>) \rightarrow PC<14:11>$
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 的高 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:	$(PC) + 1 \rightarrow TOS$, $(W) \rightarrow PC<7:0>$, $(PCLATH<6:0>) \rightarrow PC<14:8>$
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:	$00h \rightarrow (f)$ $1 \rightarrow Z$
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:	$00h \rightarrow (W)$ $1 \rightarrow Z$
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:	$00h \rightarrow WDT$ $0 \rightarrow WDT$ 预分频器 $1 \rightarrow \overline{TO}$ $1 \rightarrow PD$
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:	$(f) \rightarrow (\text{目标寄存器})$
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:	$(f) - 1 \rightarrow (\text{目标寄存器})$
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:	$(f) - 1 \rightarrow (\text{目标寄存器})$; 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, 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 无条件跳转

语法:	[标号] GOTO k
操作数:	$0 \leq k \leq 2047$
操作:	$k \rightarrow \text{PC}_{<10:0>}$ $\text{PCLATH}_{<6:3>} \rightarrow \text{PC}_{<14:11>}$
受影响的状态位:	无
说明:	GOTO 是一条无条件跳转指令。11 位立即数值被装入 PC 的 bit <10:0>。PC 的高位从 PCLATH<4:3> 装入。GOTO 是一条双周期指令。

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:	$(f) + 1 \rightarrow (\text{目标寄存器})$
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。

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:	$(f) + 1 \rightarrow (\text{目标寄存器})$, 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, 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。

IORLW 立即数和 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。

IORWF W 和 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:	$(W) .OR.(f) \rightarrow (\text{目标寄存器})$
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。

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:	$(f<7>) \rightarrow C$ $(f<6:0>) \rightarrow dest<7:1>$ $0 \rightarrow dest<0>$
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，结果存入 W。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:	$0 \rightarrow dest<7>$ $(f<7:1>) \rightarrow dest<6:0>$ $(f<0>) \rightarrow C$
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，结果存入 W。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 要受影响，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:	<code>MOVF FSR, 0</code>

执行指令后
 $W = FSR$ 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 <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:	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 / 递减指针来更新指针 (FSRn)。

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 的内容送入 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

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:	$W \rightarrow INDFn$ 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															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>模式</th> <th>语法</th> <th>mm</th> </tr> </thead> <tbody> <tr> <td>预递增</td> <td>++FSRn</td> <td>00</td> </tr> <tr> <td>预递减</td> <td>--FSRn</td> <td>01</td> </tr> <tr> <td>后递增</td> <td>FSRn++</td> <td>10</td> </tr> <tr> <td>后递减</td> <td>FSRn--</td> <td>11</td> </tr> </tbody> </table>		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
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 / 递减指针来更新指针 (FSRn)。															
<p>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。</p> <p>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。</p> <p>对于 FSRn 的递增 / 递减操作不会影响任何状态位。</p>																

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
OPTION	将 W 的内容装入 OPTION_REG 寄存器
语法:	[标号] OPTION
操作数:	无
操作:	(W) \rightarrow OPTION_REG
受影响的状态位:	无
说明:	将 W 寄存器的数据传送到 OPTION_REG 寄存器。
指令字数:	1
指令周期数:	1
示例:	OPTION 执行指令前 OPTION_REG = 0xFF W = 0x4F 执行指令后 OPTION_REG = 0x4F W = 0x4F

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 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 → PC, 1 → 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 (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:	<pre>RETFIE ; 中断后 ; PC = TOS ; GIE = 1</pre>

RETLW	返回并将立即数送入 W
语法:	[标号] RETLW k
操作数:	0 ≤ k ≤ 255
操作:	k → (W) ; TOS → 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:	<pre>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</pre>
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 → 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 ≤ f ≤ 127 d ∈ [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:	<pre>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</pre>

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。



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	$W<3:0> > k<3:0>$
DC = 1	$W<3:0> \leq k<3:0>$

SLEEP 进入休眠模式

语法: [标号] SLEEP
 操作数: 无
 操作: $00h \rightarrow WDT$,
 $0 \rightarrow WDT$ 预分频器,
 $1 \rightarrow \overline{TO}$,
 $0 \rightarrow PD$
 受影响的状态位: \overline{TO} 和 \overline{PD}
 说明: 掉电状态位 \overline{PD} 被清零。超时状态位 TO 被置 1。看门狗定时器及其预分频器被清零。
 振荡器停振, 处理器进入休眠模式。

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: $(f) - (W) \rightarrow (目标寄存器)$
 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	$W<3:0> > f<3:0>$
DC = 1	$W<3:0> \leq f<3:0>$

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: $(f) - (W) - (\bar{B}) \rightarrow dest$
 受影响的状态位: C、DC 和 Z
 说明: 用 f 寄存器的内容减去 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。

PIC16(L)F1764/5/8/9

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:	$(f<3:0>) \rightarrow (\text{目标寄存器 } <7:4>)$, $(f<7:4>) \rightarrow (\text{目标寄存器 } <3:0>)$
受影响的状态位:	无
说明:	f 寄存器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, 结果存回寄存器 f。

XORLW 立即数和 W 作逻辑异或运算

语法:	[标号] XORLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) .XOR. k \rightarrow (W)$
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 进行逻辑异或运算。结果存入 W 寄存器。

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:	$(W) \rightarrow \text{TRIS} \text{ 寄存器 } f$
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。

XORWF W 和 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:	$(W) .XOR.(f) \rightarrow (\text{目标寄存器})$
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。

36.0 电气规范

36.1 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 Vss 的电压		
VDD 引脚		
PIC16F1764/5/8/9	-0.3V 至 +6.5V
PIC16LF1764/5/8/9	-0.3V 至 +4.0V
MCLR 引脚	-0.3V 至 +9.0V
所有其他引脚	-0.3V 至 (VDD + 0.3V)
最大电流		
Vss 引脚 ⁽¹⁾		
-40°C ≤ TA ≤ +85°C	170 mA
-40°C ≤ TA ≤ +125°C	70 mA
VDD 引脚 ⁽¹⁾		
-40°C ≤ TA ≤ +85°C	170 mA
-40°C ≤ TA ≤ +125°C	70 mA
标准 I/O 引脚	±25 mA
高电流 I/O 引脚	±100 mA
钳位电流, Ik (Vpin < 0 或 Vpin > Vdd)	±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。最大电流值可以通过器件封装功率耗散特性进行限制，请参见表 36-6 来计算器件规范值。

2: 功耗按如下公式计算:

$$P_{dis} = V_{DD} * \{I_{dd} - \sum I_{oh}\} + \sum \{(V_{DD} - V_{oh}) * I_{oh}\} + \sum (V_{ol} * I_{ol})$$

† 注: 如果器件运行条件超过上述各项“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。

PIC16(L)F1764/5/8/9

36.2 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⁾

PIC16LF1764/5/9

V_{DDMIN} ($F_{osc} \leq 16$ MHz)	+1.8V
V_{DDMIN} ($F_{osc} > 16$ MHz)	+2.5V
V_{DDMAX}	+3.6V

PIC16F1764/5/9

V_{DDMIN} ($F_{osc} \leq 16$ MHz)	+2.3V
V_{DDMIN} ($F_{osc} > 16$ MHz)	+2.5V
V_{DDMAX}	+5.5V

T_A —— 工作环境温度范围

工业级温度

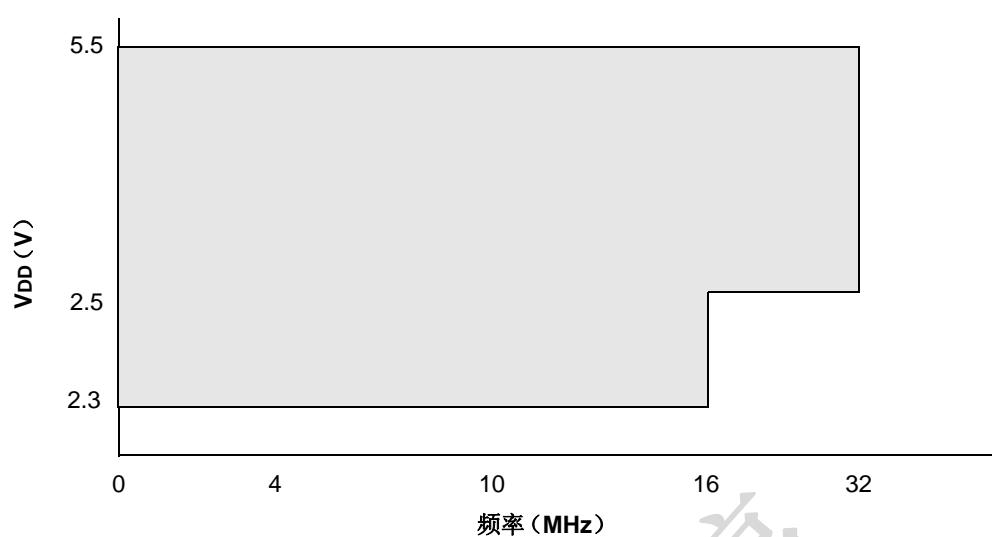
T_{A_MIN}	-40°C
T_{A_MAX}	+85°C

扩展级温度

T_{A_MIN}	-40°C
T_{A_MAX}	+125°C

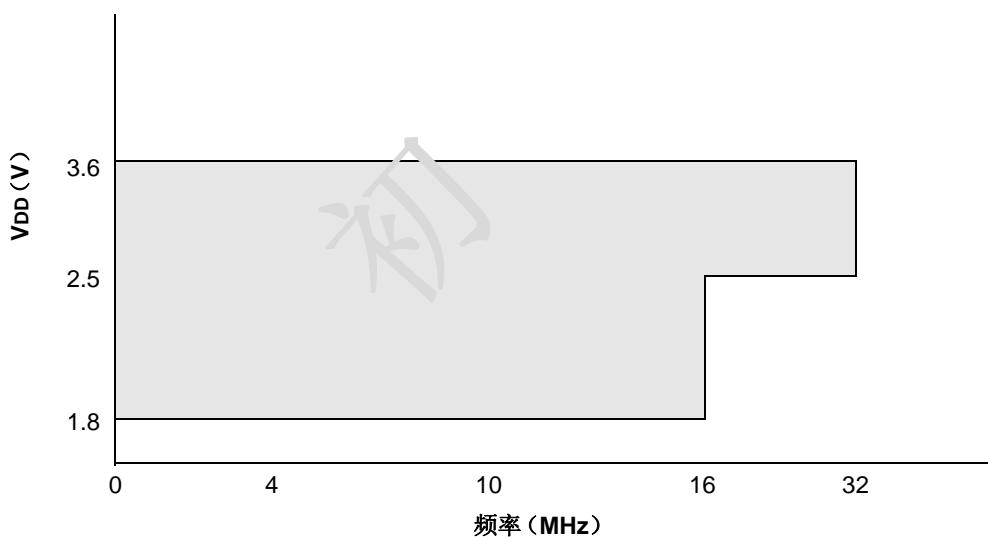
注 1：请参见直流特性：电源电压中的参数 D001。

图 36-1: 电压—频率关系图, $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$, 仅限 PIC16F1764/5/8/9



注 1: 阴影区域表示允许的电压频率组合。
2: 请参见表 36-7 了解每种振荡器模式所支持的频率。

图 36-2: 电压—频率关系图, $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$, 仅限 PIC16LF1764/5/8/9



注 1: 阴影区域表示允许的电压频率组合。
2: 请参见表 36-7 了解每种振荡器模式所支持的频率。

PIC16(L)F1764/5/8/9

36.3 直流特性

表 36-1：电源电压

PIC16LF1764/5/8/9			标准工作条件（除非另外声明）				
PIC16F1764/5/8/9							
参数 编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D001	VDD	电源电压					
			VDDMIN 1.8 2.5	— —	VDDMAX 3.6 3.6	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz (注 2)
D001		PIC16F1764/5/8/9	2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz (注 2)
D002*	VDR	RAM 数据保持电压 ⁽¹⁾					
			1.5	—	—	V	器件处于休眠模式
D002*			1.7	—	—	V	器件处于休眠模式
D002A*	VPOR	上电复位释放电压 ⁽³⁾					
			—	1.6	—	V	
D002A*			—	1.6	—	V	
D002B*	VPORR*	上电复位重新激活电压 ⁽³⁾					
			—	0.8	—	V	
D002B*			—	1.5	—	V	
D003	VFVR	固定参考电压 ⁽⁴⁾	-4	—	+4	%	1 倍增益, 1.024, VDD ≥ 2.5V, -40°C 至 +85°C
			-4	—	+4	%	2 倍增益, 2.048, VDD ≥ 2.5V, -40°C 至 +85°C
			-5	—	+5	%	4 倍增益, 4.096, VDD ≥ 4.5V, -40°C 至 +85°C
D004*	SVDD	Vdd 上升速率 ⁽²⁾	0.05	—	—	V/ms	请确保正确释放上电复位信号。

* 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。

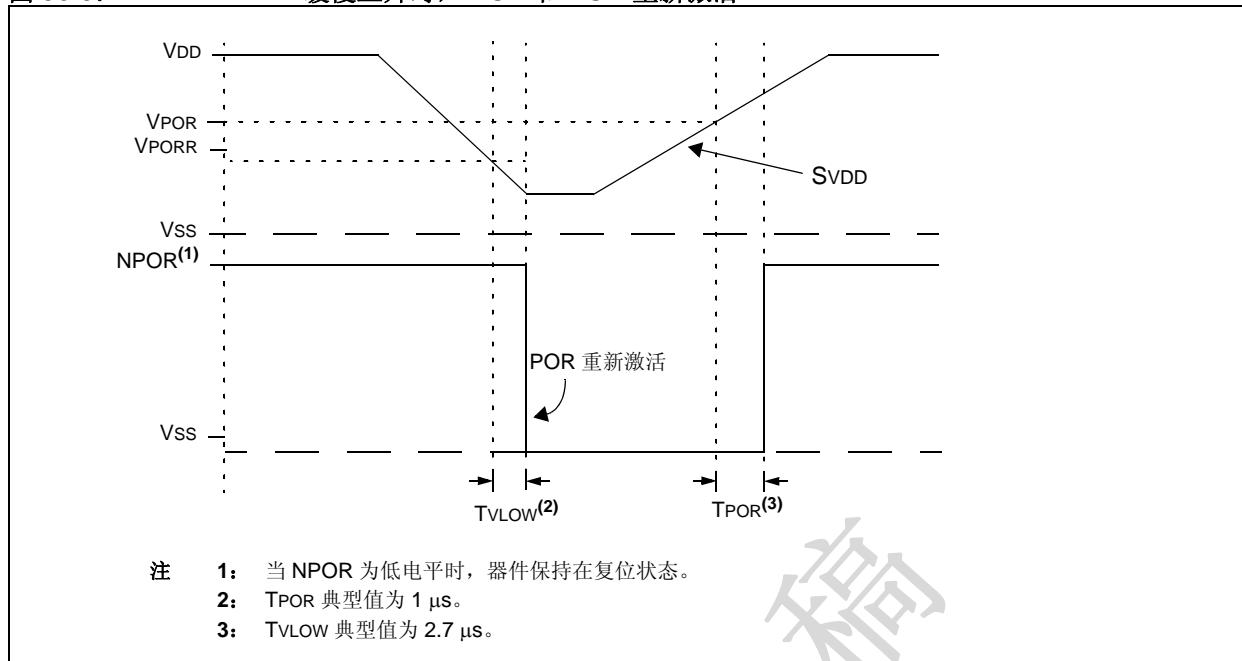
注 1: 这是在不丢失 RAM 数据的前提下，休眠模式下 VDD 的下限值。

2: 32 MHz 工作需要使用 PLL。

3: 请参见图 36-3。

4: 仅限工业级温度范围。

图 36-3: V_{DD} 缓慢上升时, POR 和 POR 重新激活



PIC16(L)F1764/5/8/9

表 36-2： 供电电流 (IDD) (1,2)

PIC16LF1764/5/8/9		标准工作条件 (除非另外声明)					
参数 编号	器件特性	最小值	典型值†	最大值	单位	条件	
						VDD	注
D009	LDO 稳压器	—	75	—	μA	—	高功耗模式, 正常工作
		—	15	—	μA	—	休眠, VREGCON<1> = 0
		—	0.3	—	μA	—	休眠, VREGCON<1> = 1
D010		—	8	—	μA	1.8	Fosc = 32 kHz, LP 振荡器模式, -40°C ≤ TA ≤ +85°C
		—	12	—	μA	3.0	
D010		—	15	—	μA	2.3	Fosc = 32 kHz, LP 振荡器模式 (注 4) -40°C ≤ TA ≤ +85°C
		—	17	—	μA	3.0	
		—	21	—	μA	5.0	
D012		—	140	—	μA	1.8	Fosc = 4 MHz, XT 振荡器模式
		—	250	—	μA	3.0	
D012		—	210	—	μA	2.3	Fosc = 4 MHz, XT 振荡器模式
		—	280	—	μA	3.0	
		—	340	—	μA	5.0	
D014		—	115	—	μA	1.8	Fosc = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	210	—	μA	3.0	
D014		—	180	—	μA	2.3	Fosc = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	240	—	μA	3.0	
		—	300	—	μA	5.0	
D015		—	2.1	—	mA	3.0	Fosc = 32 MHz, 外部时钟 (ECH), 高功耗模式
		—	2.5	—	mA	3.6	
D015		—	2.1	—	mA	3.0	Fosc = 32 MHz, 外部时钟 (ECH), 高功耗模式
		—	2.2	—	mA	5.0	

† 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: 对于 EXTRC 振荡器配置, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $IR = VDD/2REXT$ (mA) 来估算, 其中 REXT 的单位是 kΩ。
- 4: 禁止 FVR 和 BOR。
- 5: 8 MHz 晶振, 使能 4x PLL。

表 36-2: 供电电流 (IDD) (1,2) (续)

PIC16LF1764/5/8/9		标准工作条件 (除非另外声明)					
参数 编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						VDD	注
D017		—	130	—	μA	1.8	Fosc = 500 kHz, MFINTOSC 模式
		—	150	—	μA	3.0	
D017		—	150	—	μA	2.3	Fosc = 500 kHz, MFINTOSC 模式
		—	170	—	μA	3.0	
		—	220	—	μA	5.0	
D019		—	0.8	—	mA	1.8	Fosc = 16 MHz, HFINTOSC 模式
		—	1.2	—	mA	3.0	
D019		—	1.0	—	mA	2.3	Fosc = 16 MHz, HFINTOSC 模式
		—	1.3	—	mA	3.0	
		—	1.4	—	mA	5.0	
D020		—	2.1	—	mA	3.0	Fosc = 32 MHz, HFINTOSC 模式
		—	2.5	—	mA	3.6	
D020		—	2.1	—	mA	3.0	Fosc = 32 MHz, HFINTOSC 模式
		—	2.2	—	mA	5.0	
D022		—	2.1	—	mA	3.0	Fosc = 32 MHz, HS 振荡器模式 (注 5)
		—	2.5	—	mA	3.6	
D022		—	2.1	—	mA	3.0	Fosc = 32 MHz HS 振荡器模式 (注 5)
		—	2.2	—	mA	5.0	

† 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: 对于 EXTRC 振荡器配置，该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 $I_R = VDD/2REXT$ (mA) 来估算，其中 REXT 的单位是 kΩ。
- 4: 禁止 FVR 和 BOR。
- 5: 8 MHz 晶振，使能 4x PLL。

PIC16(L)F1764/5/8/9

表 36-3: 掉电电流 (IPD) (1,2)

PIC16LF1764/5/9		工作条件: (除非另外声明) 低功耗休眠模式						
PIC16F1764/5/9		低功耗休眠模式, VREGPM = 1						
参数 编号	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
D023	基本 IPD 电流	—	0.05	1.0	8.0	μA	1.8	禁止 WDT、BOR、FVR 和 SOSC, 所有外设不工作
		—	0.08	2.0	9.0	μA	3.0	
D023	基本 IPD 电流	—	0.3	3	11	μA	2.3	禁止 WDT、BOR、FVR 和 SOSC, 所有外设不工作, 低功耗休眠模式
		—	0.4	4	12	μA	3.0	
		—	0.5	6	15	μA	5.0	
D023A	基本 IPD 电流	—	9.8	16	18	μA	2.3	禁止 WDT、BOR、FVR 和 SOSC, 所有外设不工作, 正常功耗休眠模式 VREGPM = 0
		—	10.3	18	20	μA	3.0	
		—	11.5	21	26	μA	5.0	
D024		—	0.5	6	14	μA	1.8	WDT 电流
		—	0.8	7	17	μA	3.0	
D024		—	0.8	6	15	μA	2.3	WDT 电流
		—	0.9	7	20	μA	3.0	
		—	1.0	8	22	μA	5.0	
D025		—	15	28	30	μA	1.8	FVR 电流
		—	18	30	33	μA	3.0	
D025		—	18	33	35	μA	2.3	FVR 电流
		—	19	35	37	μA	3.0	
		—	20	37	39	μA	5.0	
D026		—	7.5	25	28	μA	3.0	BOR 电流
D026		—	10	25	28	μA	3.0	BOR 电流
		—	12	28	31	μA	5.0	
D027		—	0.5	4	10	μA	3.0	LPBOR 电流
D027		—	0.8	6	14	μA	3.0	LPBOR 电流
		—	1	8	17	μA	5.0	
		—	0.5	5	9	μA	1.8	
D028		—	0.8	8.5	12	μA	3.0	SOSC 电流
		—	1.1	6	10	μA	2.3	
D028		—	1.3	8.5	20	μA	3.0	SOSC 电流
		—	1.4	10	25	μA	5.0	
		—	0.05	2	9	μA	1.8	
D029		—	0.08	3	10	μA	3.0	ADC 电流 (注 3), 不在进行转换
		—	0.3	4	12	μA	2.3	
D029		—	0.4	5	13	μA	3.0	ADC 电流 (注 3), 不在进行转换
		—	0.5	7	16	μA	5.0	

* 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: ADC 时钟源是 FRC。

表 36-3: 掉电电流 (IPD) (1,2) (续)

PIC16LF1764/5/8/9		工作条件: (除非另外声明) 低功耗休眠模式						
PIC16F1764/5/8/9		低功耗休眠模式, VREGPM = 1						
参数 编号	器件特性	最小值	典型值 †	最大值 +85°C	最大值 +125°C	单位	条件	
		—	250	—	—		VDD	注
D030		—	280	—	—	μA	3.0	ADC 电流 (注 3), 转换正在进行
		—	230	—	—	μA	2.3	
D030		—	250	—	—	μA	3.0	ADC 电流 (注 3), 转换正在进行
		—	350	—	—	μA	5.0	
		—	250	650	—	μA	3.0	
D031		—	250	650	—	μA	3.0	运放 (高功耗)
		—	350	650	—	μA	5.0	
D032		—	250	600	—	μA	1.8	比较器
		—	300	650	—	μA	3.0	
D032		—	280	600	—	μA	2.3	比较器, VREGPM = 0。
		—	300	650	—	μA	3.0	
		—	310	650	—	μA	5.0	

* 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: ADC 时钟源是 FRC。

PIC16(L)F1764/5/8/9

表 36-4: I/O 端口

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		
D034 D034A D035 D036 D036A	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		
			—	—	0.15 V _{DD}	V	1.8V ≤ V _{DD} ≤ 4.5V		
		带施密特触发器缓冲器	—	—	0.2 V _{DD}	V	2.0V ≤ V _{DD} ≤ 5.5V		
		I ² C 电平	—	—	0.3 V _{DD}	V			
		SMBus 电平	—	—	0.8	V	2.7V ≤ V _{DD} ≤ 5.5V		
		MCLR 和 OSC1 (EXTRC 模式)	—	—	0.2 V _{DD}	V	(注 1)		
		OSC1 (HS 模式)	—	—	0.3 V _{DD}	V			
D040 D040A D041 D042 D043A D043B	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		
			0.25 V _{DD} + 0.8	—	—	V	1.8V ≤ V _{DD} ≤ 4.5V		
		带施密特触发器缓冲器	0.8 V _{DD}	—	—	V	2.0V ≤ V _{DD} ≤ 5.5V		
		I ² C 电平	0.7 V _{DD}	—	—	V			
		SMBus 电平	2.1	—	—	V	2.7V ≤ V _{DD} ≤ 5.5V		
		MCLR	0.8 V _{DD}	—	—	V			
		OSC1 (HS 模式)	0.7 V _{DD}	—	—	V			
		OSC1 (EXTRC 振荡器)	0.9 V _{DD}	—	—	V	V _{DD} > 2.0V (注 1)		
D060 D061	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		
			—	± 5	± 1000	nA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, 125°C		
		MCLR ⁽³⁾	—	± 50	± 200	nA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, 85°C		
D070*	I _{PUR}	弱上拉电流							
			25	100	200	μA	V _{DD} = 3.3V, V _{PIN} = V _{SS}		
			25	140	300	μA	V _{DD} = 5.0V, V _{PIN} = V _{SS}		
D080 D080A	V _{OL}	输出低电压 ⁽⁴⁾							
		标准 I/O 端口	—	—	0.6	V	I _{OL} = 8 mA, V _{DD} = 5V I _{OL} = 6 mA, V _{DD} = 3.3V I _{OL} = 1.8 mA, V _{DD} = 1.8V		
		高驱动 I/O 端口	—	—	0.6	V	I _{OH} = 10 mA, V _{DD} = 2.3V, HIDCX = 1 I _{OH} = 32 mA, V _{DD} = 3.0V, HIDCX = 1 I _{OH} = 51 mA, V _{DD} = 5.0V, HIDCX = 1		
			—	0.6	—	V	I _{OH} = 10 mA, V _{DD} = 2.3V, HIDCX = 1 I _{OH} = 32 mA, V _{DD} = 3.0V, HIDCX = 1 I _{OH} = 51 mA, V _{DD} = 5.0V, HIDCX = 1		
D090	V _{OH}	输出高电压 ⁽⁴⁾							
		标准 I/O 端口	V _{DD} - 0.7	—	—	V	I _{OH} = 3.5 mA, V _{DD} = 5V I _{OH} = 3 mA, V _{DD} = 3.3V I _{OH} = 1 mA, V _{DD} = 1.8V		
D090A		高驱动 I/O 端口	V _{DD} - 0.7	—	—	V	I _{OH} = 10 mA, V _{DD} = 2.3V, HIDCX = 1 I _{OH} = 37 mA, V _{DD} = 3.0V, HIDCX = 1 I _{OH} = 54 mA, V _{DD} = 5.0V, HIDCX = 1		

* 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: 在 EXTRC 振荡器配置中, OSC1/CLKIN 引脚被配置为施密特触发器输入。在 EXTRC 模式下, 建议不要使用外部时钟。

2: 负电流定义为引脚的拉电流。

3: 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。

4: 在 CLKOUT 模式下包括 OSC2。

表 36-4: I/O 端口 (续)

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
D101*	输出引脚上的容性负载规范						
D101*	COSC2	OSC2 引脚	—	—	15	pF	当外部时钟用于驱动 OSC1 时, 处于 XT、HS 和 LP 模式下
D101A*	CIO	所有 I/O 引脚	—	—	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: 在 EXTRC 振荡器配置中, OSC1/CLKIN 引脚被配置为施密特触发器输入。在 EXTRC 模式下, 建议不要使用外部时钟。

2: 负电流定义为引脚的拉电流。

3: 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。

4: 在 CLKOUT 模式下包括 OSC2。

表 36-5: 存储器编程规范

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	
D110	VIHH	程序存储器编程规范	8.0	—	9.0	V	(注 2)	
		MCLR/VPP 引脚上的电压			10	mA		
D111	IDDP	编程时的供电电流	—	—	—	—	—	
D112	VBE	批量擦除时的 VDD	2.7	—	VDDMAX	V	—	
D113	VPEW	写或行擦除时的 VDD	VDDMIN	—	VDDMAX	V	—	
D114	I _{PPPGM}	擦除 / 写操作时 MCLR/VPP 上的电流	—	1.0	—	mA	—	
D115	I _{DDPGM}	擦除 / 写操作时 VDD 上的电流	—	5.0	—	mA	—	
D121-D125	EP	闪存程序存储器	10K	—	—	E/W	-40°C ≤ TA ≤ +85°C (注 1)	
		单元耐擦写能力						
		读 / 写操作时的 VDD		VDDMIN	2	V		
		自定时写周期时间			2.5	ms		
		特性保持时间			40	年	假设没有违反其他规范	
D125	EHEFC	高耐用性闪存单元	100K	—	—	E/W	-0°C ≤ TA ≤ +60°C, 最后 128 个地址的低字节	

† 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: 自写和块擦除。

2: 仅当禁止单电源编程时才需要。

表 36-6: 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	θJA	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 QFN 4x4 mm 封装
			62.2	°C/W	20 引脚 PDIP 封装
			87.3	°C/W	20 引脚 SSOP
			77.7	°C/W	20 引脚 SOIC 封装
			43.0	°C/W	20 引脚 QFN 4x4 mm 封装
TH02	θJC	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 QFN 4x4 mm 封装
			27.5	°C/W	20 引脚 PDIP 封装
			31.1	°C/W	20 引脚 SSOP
			23.1	°C/W	20 引脚 SOIC 封装
			5.3	°C/W	20 引脚 QFN 4x4 mm 封装
TH03	TJMAX	最高结温	150	°C	
TH04	PD	功耗	—	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	内部功耗	—	W	PINTERNAL = IDD × VDD ⁽¹⁾
TH06	PI/O	I/O 功耗	—	W	PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))
TH07	PDER	降额功耗	—	W	PDER = PDMAX (TJ - TA)/θJA ⁽²⁾

注 1: IDD 为不驱动输出引脚上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)F1764/5/8/9

36.4 交流特性

根据以下格式之一创建了时序参数符号：

1. T_{ppS2ppS}

2. T_{ppS}

T	
F	频率

小写字母 (pp) 及其含义：

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	\overline{RD}
cs	\overline{CS}	rw	\overline{RD} 或 \overline{WR}
di	SDI	sc	SCK
do	SDO	ss	\overline{SS}
dt	数据输入	t0	T0CKI
io	I/O 端口	t1	T1CKI
mc	MCLR	wr	\overline{WR}

大写字母及其含义：

S		P	
F	下降	P	周期
H	高	R	上升
I	无效 (高阻)	V	有效
L	低	Z	高阻

图 36-4：负载条件

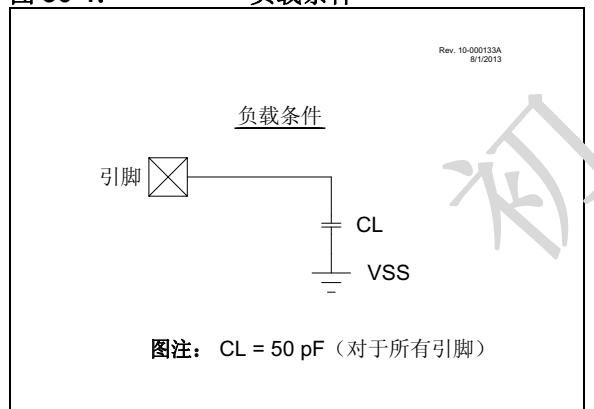


图 36-5: 时钟时序

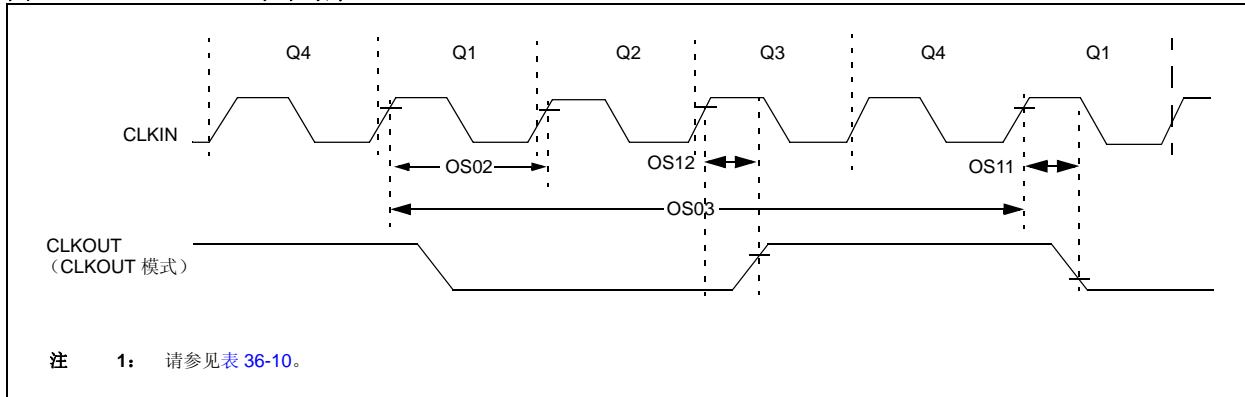


表 36-7: 时钟振荡器时序要求

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
OS01	FOSC	外部 CLKIN 频率 (1)	DC	—	0.5	MHz	外部时钟 (ECL)
			DC	—	4	MHz	外部时钟 (ECM)
			DC	—	20	MHz	外部时钟 (ECH)
		振荡器频率 (1)	—	32.768	—	kHz	LP 振荡器
			0.1	—	4	MHz	XT 振荡器
			1	—	4	MHz	HS 振荡器
			1	—	20	MHz	HS 振荡器, VDD > 2.7V
			DC	—	4	MHz	EXTRC, VDD > 2.0V
		外部 CLKIN 周期 (1)	27	—	∞	μs	LP 振荡器
			250	—	∞	ns	XT 振荡器
			50	—	∞	ns	HS 振荡器
			50	—	∞	ns	外部时钟 (EC)
			—	30.5	—	μs	LP 振荡器
		振荡器周期 (1)	—	—	10,000	ns	XT 振荡器
			250	—	1,000	ns	HS 振荡器
			50	—	—	ns	EXTRC
			250	—	—	ns	—
OS03	T _{CY}	指令周期 (1)	125	T _{CY}	DC	ns	T _{CY} = 4/FOSC
OS04*	TosH, TosL	外部 CLKIN 高电平时间, 外部 CLKIN 低电平时间	2	—	—	μs	LP 振荡器
			100	—	—	ns	XT 振荡器
			20	—	—	ns	HS 振荡器
OS05*	TosR, TosF	外部 CLKIN 上升时间, 外部 CLKIN 下降时间	0	—	∞	ns	LP 振荡器
			0	—	∞	ns	XT 振荡器
			0	—	∞	ns	HS 振荡器

* 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。

注 1: 指令周期 (T_{CY}) 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)。

PIC16(L)F1764/5/8/9

表 36-8: 振荡器参数

标准工作条件 (除非另外声明)								
参数 编号	符号	特性	频率 容差	最小值	典型值 †	最大值	单位	条件
OS08	HFOSC	内部已校准的 HFINTOSC 频率 (1)	±2%	—	16.0	—	MHz	VDD = 3.0V, TA = 25°C (注 2)
OS08A	MFOSC	内部已校准的 MFINTOSC 频率 (1)	±2%	—	500	—	kHz	VDD = 3.0V, TA = 25°C (注 2)
OS09	LFOSC	内部 LFINTOSC 频率	—	—	31	—	kHz	-40°C ≤ TA ≤ +125°C
OS10*	Tiosc ST	HFINTOSC 从休眠模式唤醒的启动时间	—	—	3.2	8	μs	
		MFINTOSC 从休眠模式唤醒的启动时间	—	—	24	35	μs	
OS10A*	TLFOSC ST	LFINTOSC 从休眠模式唤醒的启动时间	—	—	0.5	—	ms	-40°C ≤ TA ≤ +125°C

* 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 36-6、图 37-22 和图 36-6。

3: 请参见图 37-20 和图 37-21。

图 36-6: 器件 VDD 和温度范围内的 HFINTOSC 频率精度

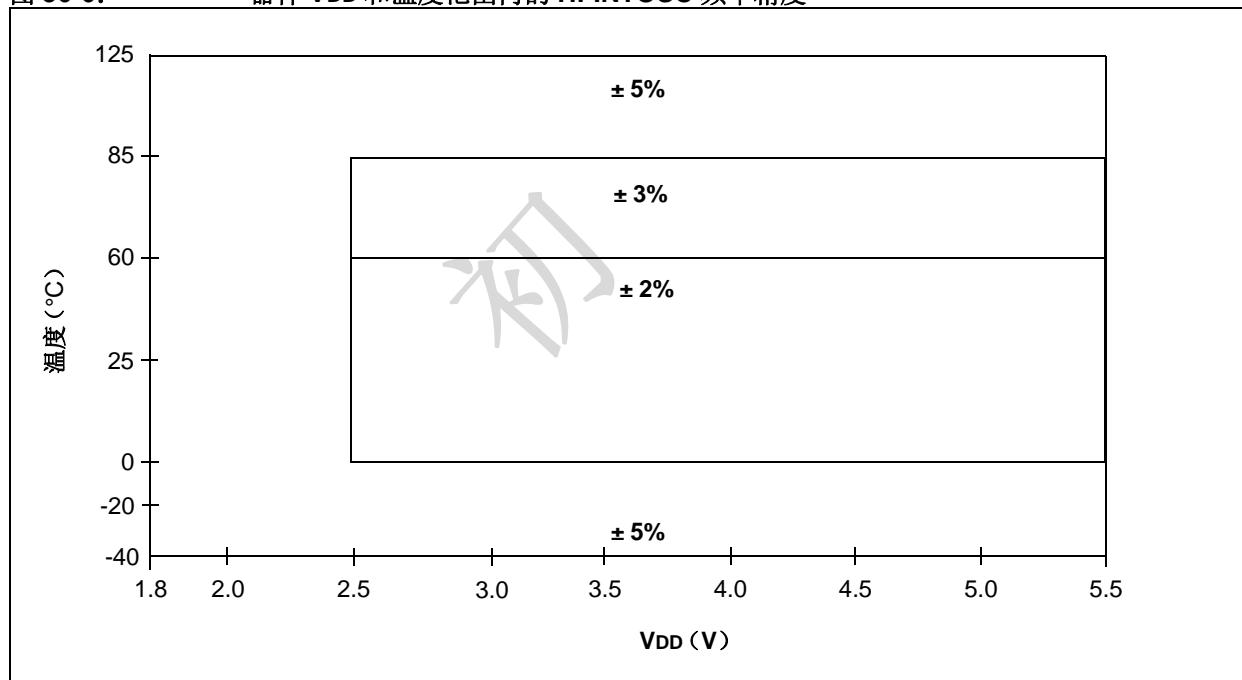


表 36-9: PLL 时钟时序规范

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
F10	FOSC	振荡器频率范围	4	—	8	MHz	
F11	FSYS	片上 VCO 系统频率	16	—	32	MHz	
F12	TRC	PLL 起振时间 (锁定时间)	—	—	2	ms	
F13*	ΔCLK	CLKOUT 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。

PIC16(L)F1764/5/8/9

图 36-7: CLKOUT 和 I/O 时序

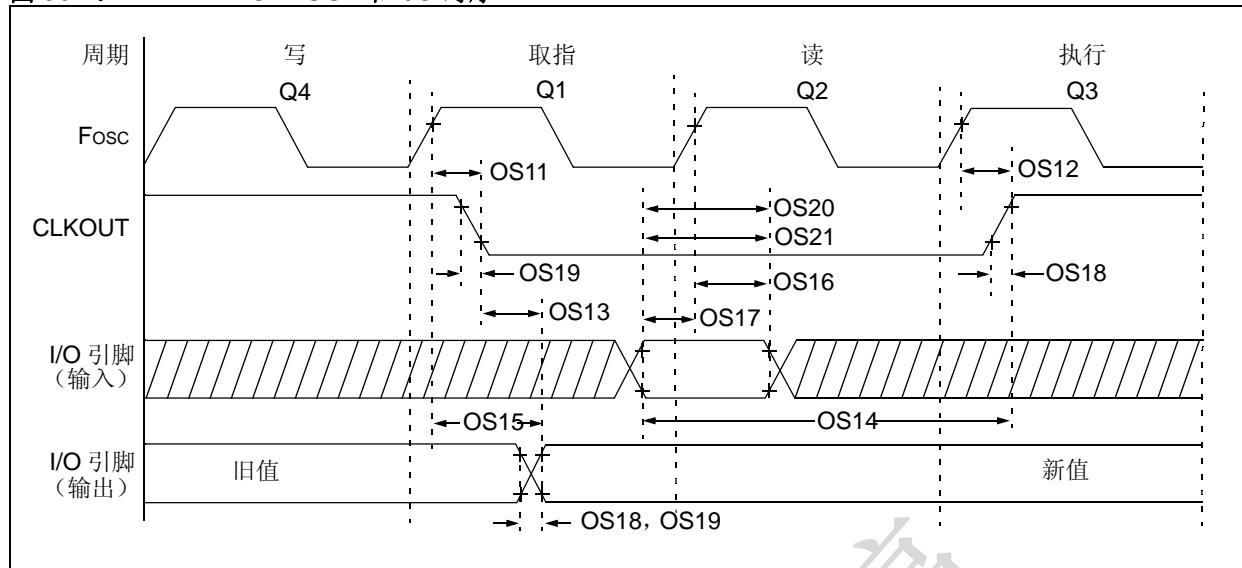


表 36-10: CLKOUT 和 I/O 时序参数

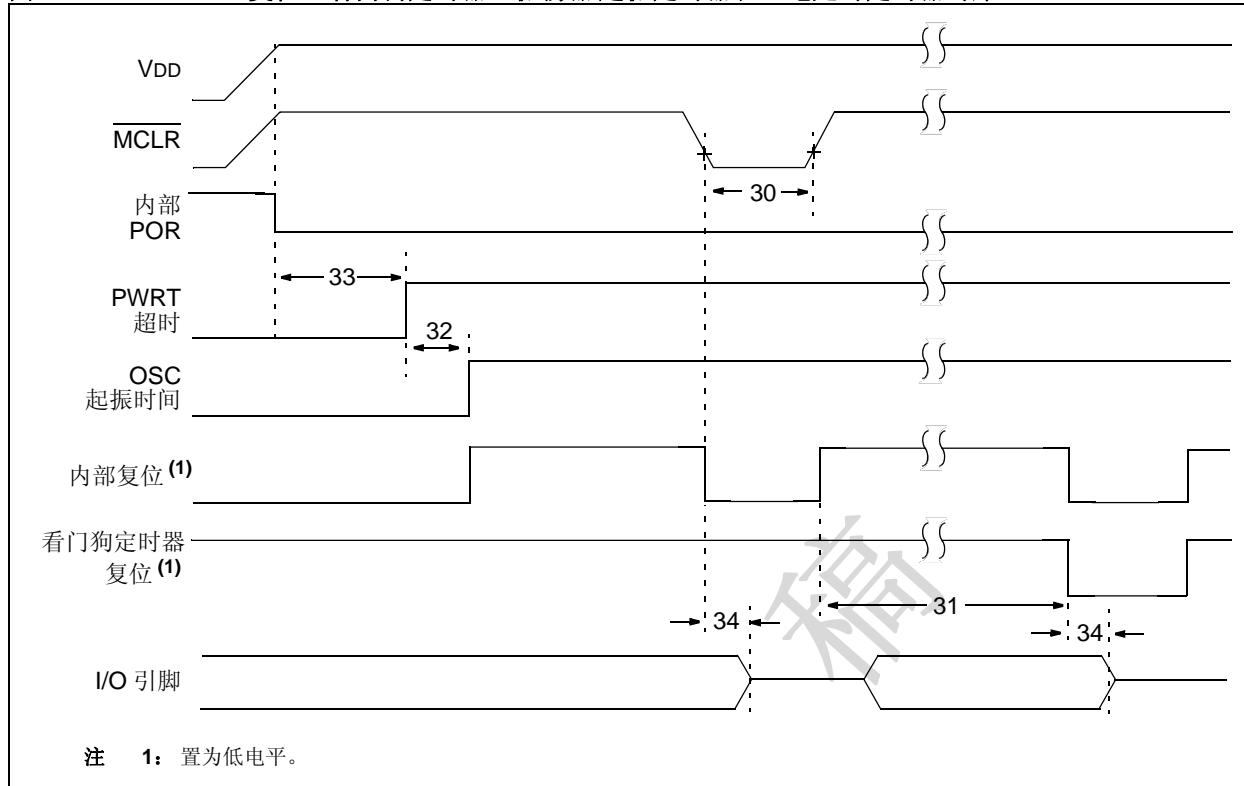
标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS11	TosH2ckL	Fosc↑ 到 CLKOUT↓ 的时间 (1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	Fosc↑ 到 CLKOUT↑ 的时间 (1)	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓ 到端口输出有效的时间 (1)	—	—	20	ns	
OS14	TioV2ckH	CLKOUT↑ 之前端口输入有效的时间 (1)	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 周期) 到端口输出有效的时间	—	50	70*	ns	3.3V ≤ VDD ≤ 5.0V
OS16	TosH2iol	Fosc↑ (Q2 周期) 到端口输入无效的时间 (I/O 输入保持时间)	50	—	—	ns	3.3V ≤ VDD ≤ 5.0V
OS17	TioV2osH	端口输入有效到 Fosc↑ (Q2 周期) 的时间 (I/O 输入建立时间)	20	—	—	ns	
OS18*	TioR	端口输出上升时间	—	40 15	72 32	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS19*	TioF	端口输出下降时间	—	28 15	55 30	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS20*	Tinp	INT 引脚输入高电平或低电平时间	25	—	—	ns	
OS21*	Tioc	电平变化中断新输入电平时间	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。

注 1: 测量是在 EXTRC 模式下进行的，其中 CLKOUT 输出为 4 × Tosc。

图 36-8: 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



PIC16(L)F1764/5/8/9

表 36-11：复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位参数

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
30	TMCL	MCLR 脉冲宽度（低电平）	2	—	—	μs	
31	TWDTLP	低功耗看门狗定时器超时周期	10	16	27	ms	VDD = 3.3V-5V 使用 1:16 预分频比
32	TOST	振荡器起振定时器周期 (1)	—	1024	—	Tosc	
33*	TPWRT	上电延时定时器周期, $\overline{\text{PWRTE}} = 0$	40	65	140	ms	
34*	TIOZ	自 $\overline{\text{MCLR}}$ 低电平或看门狗定时器复位起到 I/O 处于高阻态的时间	—	—	2.0	μs	
35	VBOR	欠压复位 (2)	2.55 2.30 1.80	2.70 2.45 1.90	2.85 2.60 2.10	V V V	BORV = 0 BORV = 1 (PIC16F1764/5/8/9) BORV = 1 (PIC16LF1764/5/8/9)
35A	VLPBOR	低功耗欠压	1.8	2.1	2.5	V	LPBOR = 1
36*	VHYST	欠压复位滞后电压	0	25	75	mV	-40°C ≤ TA ≤ +85°C
37*	TBORDC	欠压复位直流响应时间	1	3	35	μs	VDD ≤ VBOR

* 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：根据设计，振荡器起振定时器 (OST) 计数前 1024 个周期，与频率无关。

2：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。

图 36-9: TIMER0 和 TIMER1 外部时钟时序

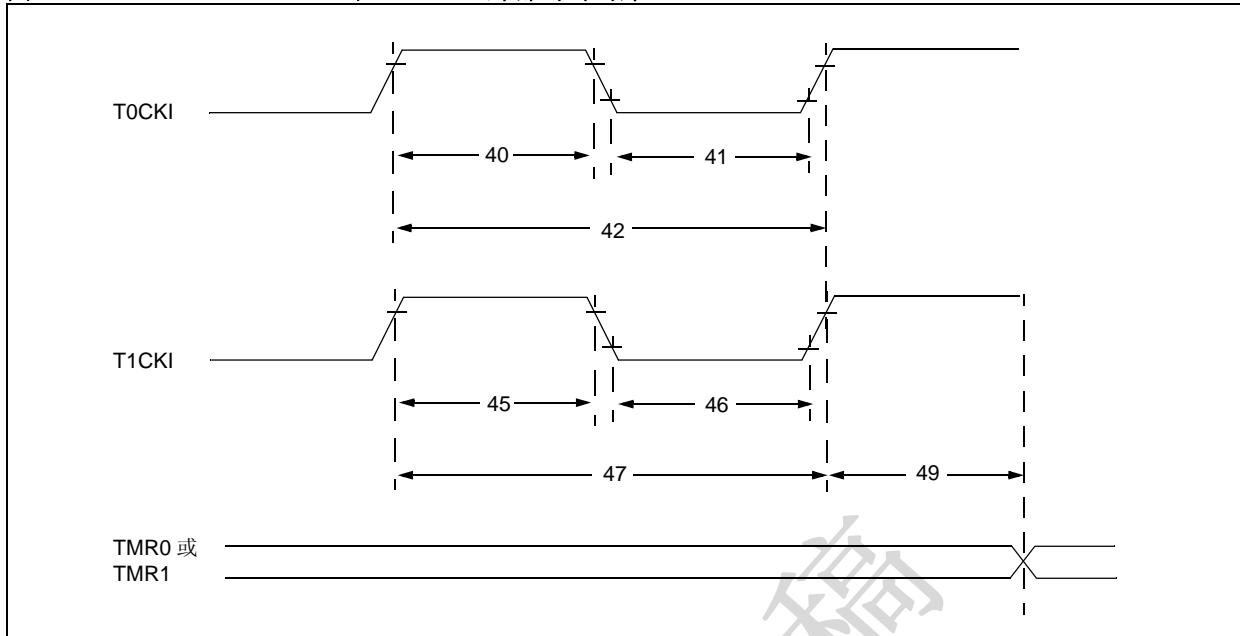
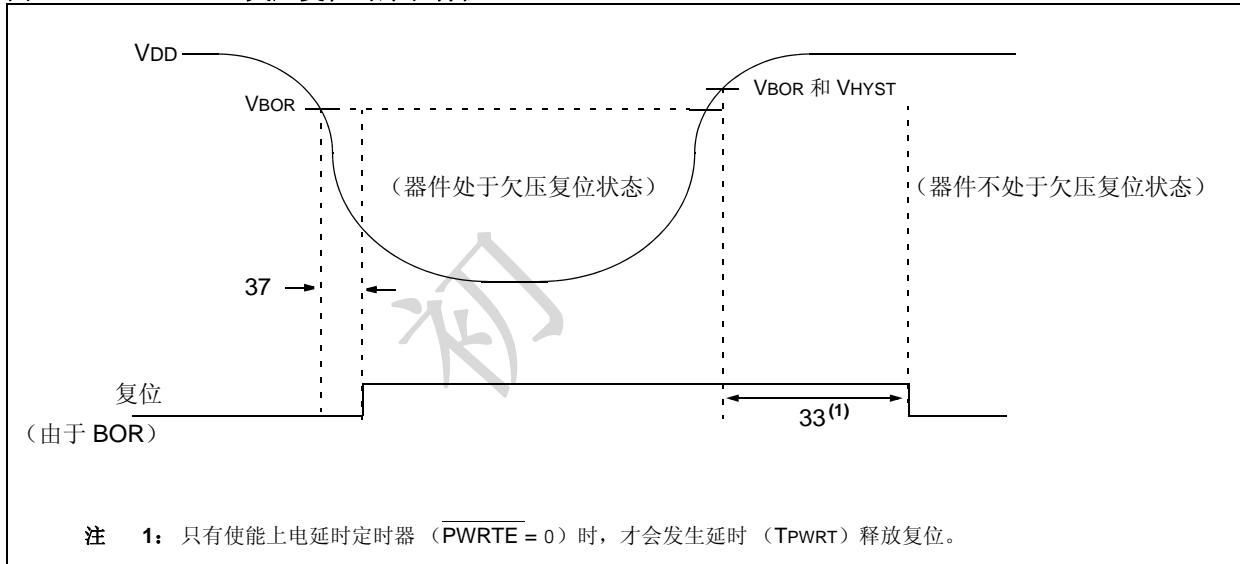


图 36-10: 欠压复位时序和特性



PIC16(L)F1764/5/8/9

表 36-12: TIMER0 和 TIMER1 外部时钟要求

标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C									
参数 编号	符号	特性		最小值	典型值 †	最大值	单位	条件	
40*	TT0H	T0CKI 高电平脉冲宽度	无预分频器	0.5 TCY + 20	—	—	ns		
			带预分频器	10	—	—	ns		
41*	TT0L	T0CKI 低电平脉冲宽度	无预分频器	0.5 TCY + 20	—	—	ns		
			带预分频器	10	—	—	ns		
42*	TT0P	T0CKI 周期		取如下二者中 较大值: 20 或 $\frac{TCY + 40}{N}$	—	—	ns	N = 预分频值	
45*	TT1H	T1CKI 高电 平时间	同步, 无预分频器		—	—	ns		
46*	TT1L		同步, 带预分频器	15	—	—	ns		
			异步	30	—	—	ns		
			同步, 无预分频器	0.5 TCY + 20	—	—	ns		
47*	TT1P	T1CKI 输入 周期	同步, 带预分频器	15	—	—	ns		
			异步	30	—	—	ns		
			同步	取如下二者中 较大值: 30 或 $\frac{TCY + 40}{N}$	—	—	ns	N = 预分频值	
48	FT1	辅助振荡器输入频率范围 (通过将 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 Tosc	—	7 Tosc	—	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。

图 36-11: 捕捉 / 比较 /PWM 时序 (CCP)

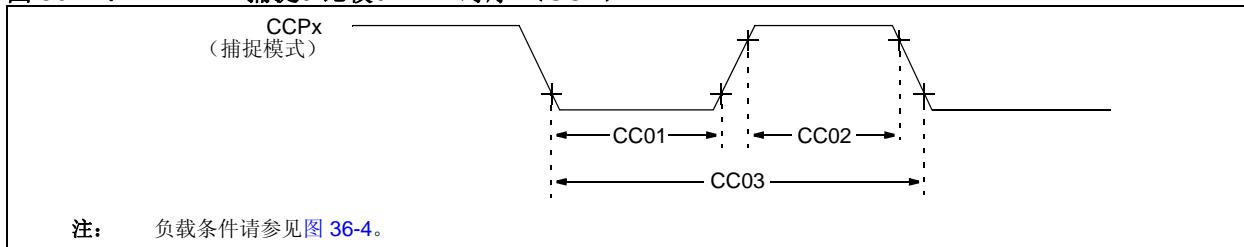


表 36-13: 捕捉 / 比较 /PWM 要求 (CCP)

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	0.5TCY + 20	—	—	ns	
			带预分频器	20	—	—	ns	
CC02*	TccH	CCPx 输入高电平时间	无预分频器	0.5TCY + 20	—	—	ns	
			带预分频器	20	—	—	ns	
CC03*	TccP	CCPx 输入周期		$\frac{3TCY + 40}{N}$	—	—	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 36-12: CLC 传播时序

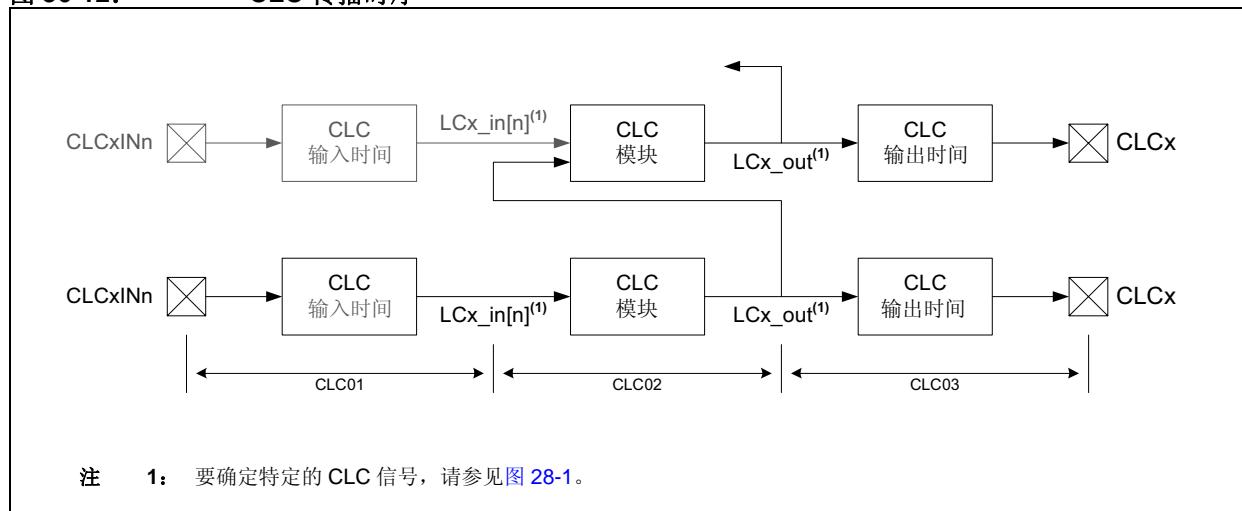


表 36-14: 可配置逻辑单元 (CLC) 特性

标准工作条件 (除非另外声明)							
工作温度 -40°C ≤ TA ≤ +125°C							
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	VDD = 1.8V VDD > 3.6V
CLC03*	TCLCOUT	CLC 输出时间	—	OS18	—	—	(注 1)
			—	OS19	—	—	(注 1)
CLC04*	FCLCMAX	CLC 最大开关频率	—	45	—	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 上升时间和下降时间, 请参见表 36-10。

表 36-15: 模数转换器 (ADC) 特性^(1,2,3,4):

工作条件 (除非另外声明) $VDD = 3V$, $T_A = 25^\circ C$, 单端, $2 \mu s TAD$, $VREF+ = 3V$, $VREF- = VSS$							
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	—	—	± 1.7	LSb	$VREF = 3.0V$
AD03	EDL	微分误差	—	—	± 1	LSb	无丢失编码, $VREF = 3.0V$
AD04	EOFF	失调误差	—	—	± 2.5	LSb	$VREF = 3.0V$
AD05	EGL	增益误差	—	—	± 2.0	LSb	$VREF = 3.0V$
AD06	VREF	参考电压	1.8	—	VDD	V	$VREF = (VREF+ - VREF-)$
AD07	VAIN	满量程	VSS	—	$VREF$	V	
AD08	ZAIN	模拟信号源的推荐阻抗	—	—	10	kΩ	如果输入引脚上接有 $0.01 \mu F$ 的外部电容，则该值可以更高。

* 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 $25^\circ C$ 条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 总的绝对误差包括积分误差、微分误差、失调误差和增益误差。

2: ADC 转换结果不会因输入电压的增加而减小, 并且不会丢失编码。

3: ADC VREF 来自选择作为参考输入的外部 VREF+ 引脚、VDD 引脚或 FVR。

4: 关于工作特性, 请参见第 37.0 节 “直流和交流特性图表”。

表 36-16: ADC 转换要求

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
AD130*	TAD	ADC 时钟周期 (TADC) ADC 内部 FRC 振荡器周期 (TFRC)	1.0 1.0	— 2.5	9.0 6.0	μs μs	基于 FOSC ADCS<1:0> = 11 (ADC FRC 模式)
AD131	TCNV	转换时间 (不包括采集时间) (1)	—	13	—	TAD	将 GO/DONE 位置 1 以完成转换
AD132*	TACQ	采集时间	—	5.0	—	μs	
AD133*	THCD	保持电容断开时间	—	1/2 TAD	—		ADCS<2:0> ≠ x11 (基于 FOSC)
			—	1/2 TAD + 1TCY	—		ADCS<2:0> = x11 (基于 FRC)

* 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 $25^\circ C$ 条件下的值。这些参数仅供设计参考, 未经测试。

注 1: ADRES 寄存器可在下一个 TCY 周期被读取。

PIC16(L)F1764/5/8/9

图 36-13: ADC 转换时序 (ADC 时钟基于 Fosc)

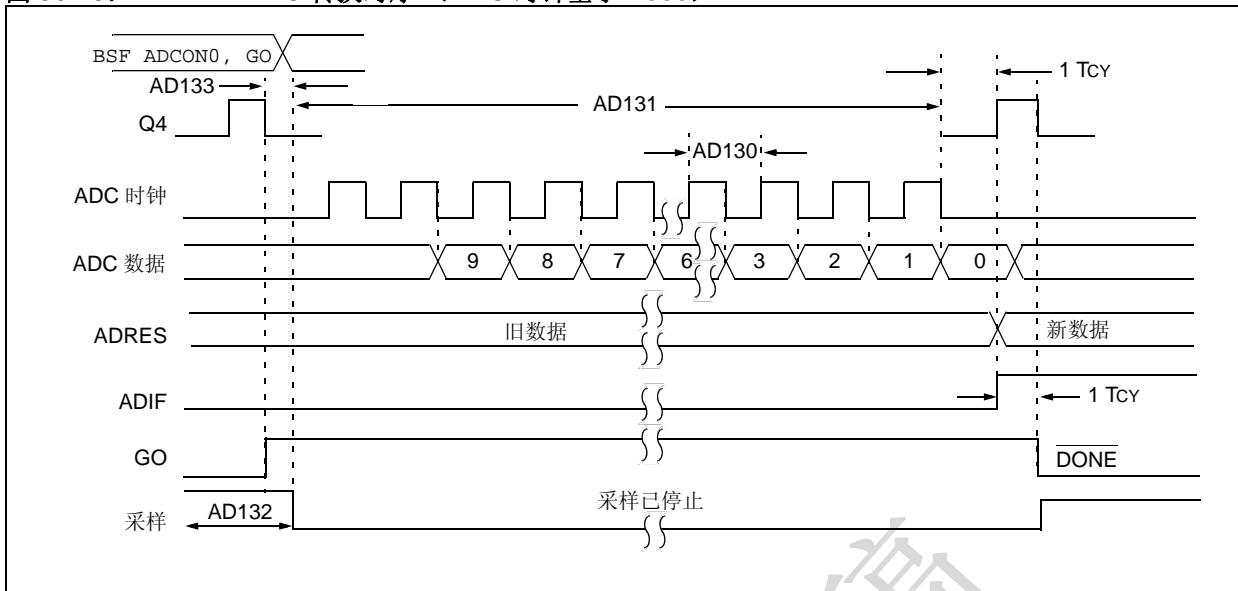
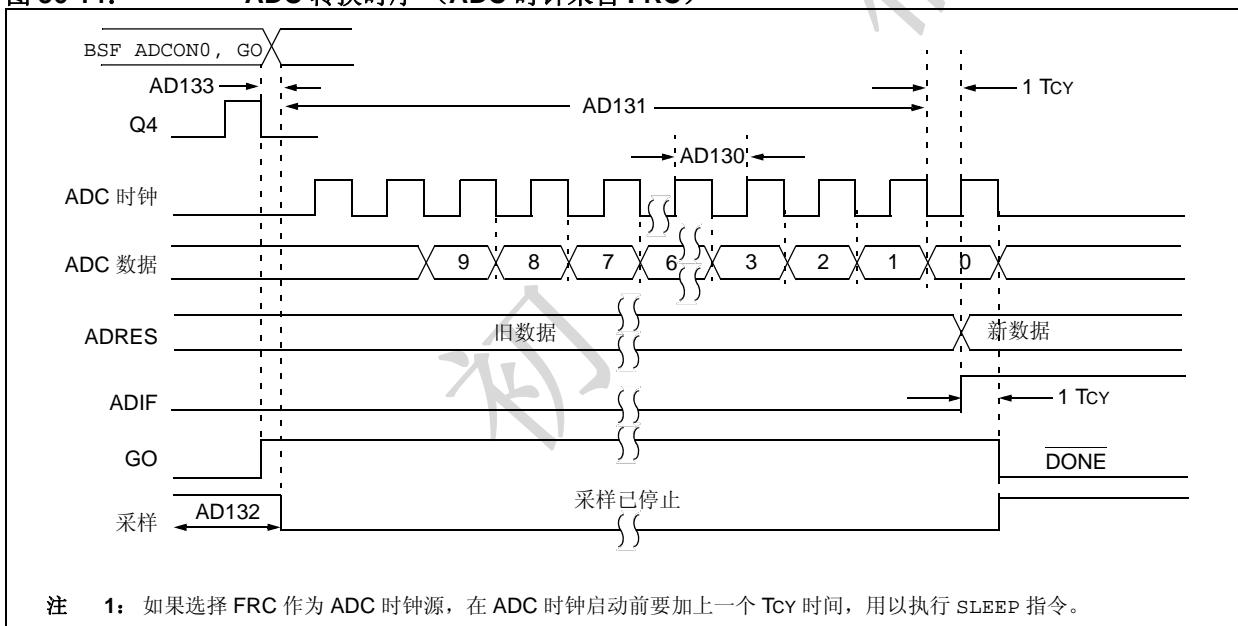


图 36-14: ADC 转换时序 (ADC 时钟来自 FRC)



注 1: 如果选择 FRC 作为 ADC 时钟源, 在 ADC 时钟启动前要加上一个 Tcy 时间, 用以执行 SLEEP 指令。

表 36-17: 运算放大器 (OPA)

参数 编号	符号	参数	最小值	典型值	最大值	单位	条件
OPA01*	GBWP	增益带宽积	—	2	—	MHz	
OPA02*	TON	开启时间	—	10	—	μs	
OPA03*	PM	相位裕量	—	40	—	度	
OPA04*	SR	压摆率	—	3	—	V/μs	
OPA05	OFF	失调	—	±3	±9	mV	
OPA06	CMRR	共模抑制比	52	70	—	dB	
OPA07*	AOL	开环增益	—	90	—	dB	
OPA08	VICM	输入共模电压	0	—	VDD	V	VDD > 2.5V
OPA09*	PSRR	电源抑制比	—	80	—	dB	
OPA10*	HZ	高阻抗开/关时间	—	50	—	ns	

* 这些参数为特性值，但未经测试。

表 36-18: 可编程斜坡发生器 (PRG) 规范

工作条件 (除非另外声明) VDD = 3.0V, TA = 25°C (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	备注
PRG01	RRR	上升斜坡速率	—	1	—	V/μs	PRGxCON2 = 10h
PRG02	FRR	下降斜坡速率	—	1	—	V/μs	PRGxCON2 = 10h

* 这些参数为特性值，但未经测试。

表 36-19: 比较器规范

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 37.0 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	VIOFF	输入失调电压	—	±2.5	±5	mV	VICM = VDD/2
CM02	VICM	输入共模电压	0	—	VDD	V	
CM03	CMRR	共模抑制比	40	50	—	dB	
CM04A	TRESP(1)	响应时间上升沿	—	60	85	ns	
CM04B		响应时间下降沿	—	60	90	ns	
CM05*	TMC2OV	比较器模式改变到输出有效的时间 *	—	—	10	μs	
CM06	CHYSTER	比较器滞后	20	45	75	mV	CxHYS = 1

* 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 VDD/2，而另一个输入端从 Vss 跳变到 VDD 时测得的。

PIC16(L)F1764/5/8/9

表 36-20: 10 位数模转换器 (DAC) 规范

参数 编号	符号	特性	最小值	典型值	最大值	单位	备注
DAC01*	CLSB	步长	—	VDD/1024	—	V	
DAC02	CINL	积分误差 (2)	—	—	± 1.5	LSb	对于编码 0x004 至 0x3FB
DAC03	CDNL	差分误差 (2)	—	—	± 1	LSb	
DAC04	COFF	失调误差 (2)	—	—	± 3	LSb	
DAC05	CGN	增益误差 (2)	—	—	± 3	LSb	
DAC06*	CR	单位电阻值 (R)	—	300	—	Ω	
DAC07*	CST	稳定时间 (1)	—	—	10	μs	

* 这些参数为特性值，但未经测试。

注 1: 稳定时间是在 DACR<9:0> 从 0x000 跳变到 0x3FF 时测得的。

2: 通过单位增益的运放进行缓冲。

表 36-21: 5 位数模转换器 (DAC) 规范

参数 编号	符号	特性	最小值	典型值	最大值	单位	备注
DAC10*	CLSB	步长	—	VDD/32	—	V	
DAC11	CACC	绝对精度 (2)	—	—	± 0.5	LSb	
DAC12*	CR	单位电阻值 (R)	—	6000	—	Ω	
DAC13*	CST	稳定时间 (1)	—	—	10	μs	

* 这些参数为特性值，但未经测试。

注 1: 稳定时间是在 DACR<4:0> 从 0x00 跳变到 0x1F 时测得的。

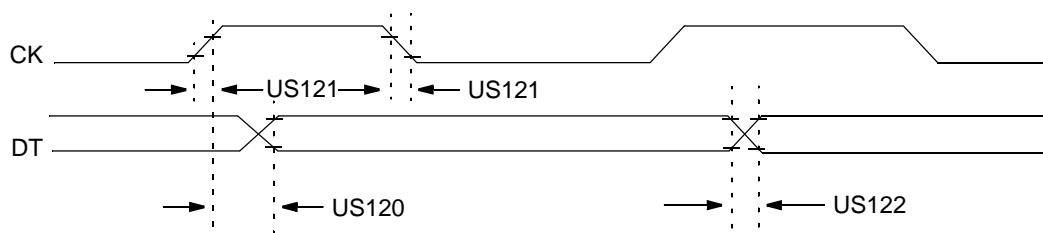
2: 通过单位增益的运放进行缓冲。

表 36-22: 过零引脚规范

参数 编号	符号	特性	最小值	典型值	最大值	单位	备注
ZC01	ZCPINV	过零引脚上的电压	—	0.75	—	V	
ZC02	ZCSRC	拉电流	—	-300	-600	μA	
ZC03	ZCSNK	灌电流	—	300	600	μA	
ZC04	ZCISW	响应时间上升沿	—	1	—	μs	
		响应时间下降沿	—	1	—	μs	
ZC05	ZCOUT	响应时间上升沿	—	1	—	μs	
		响应时间下降沿	—	1	—	μs	

* 这些参数为特性值，但未经测试。

图 36-15: EUSART 同步发送（主 / 从）时序



注：负载条件请参见图 36-4。

表 36-23: EUSART 同步发送要求

标准工作条件（除非另外声明）						
参数编号	符号	特性	最小值	最大值	单位	条件
US120	TCKH2DTV	同步发送（主 / 从） 时钟高电平到数据输出有效的时间	—	80	ns	3.0V ≤ VDD ≤ 5.5V
			—	100	ns	1.8V ≤ VDD ≤ 5.5V
US121	TCKRF	时钟输出上升时间和下降时间 (主模式)	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V
US122	TDTRF	数据输出上升时间和下降时间	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V

图 36-16: EUSART 同步接收（主 / 从）时序



注：负载条件请参见图 36-4。

表 36-24: EUSART 同步接收要求

标准工作条件（除非另外声明）						
参数编号	符号	特性	最小值	最大值	单位	条件
US125	TDTV2CKL	同步接收（主 / 从） CK ↓ 之前的数据建立时间（DT 保持时间）	10	—	ns	
			—	—	ns	
US126	TckL2DTL	CK ↓ 之后的数据保持时间（DT 保持时间）	15	—	ns	

PIC16(L)F1764/5/8/9

图 36-17: SPI 主模式时序 ($\text{CKE} = 0, \text{SMP} = 0$)

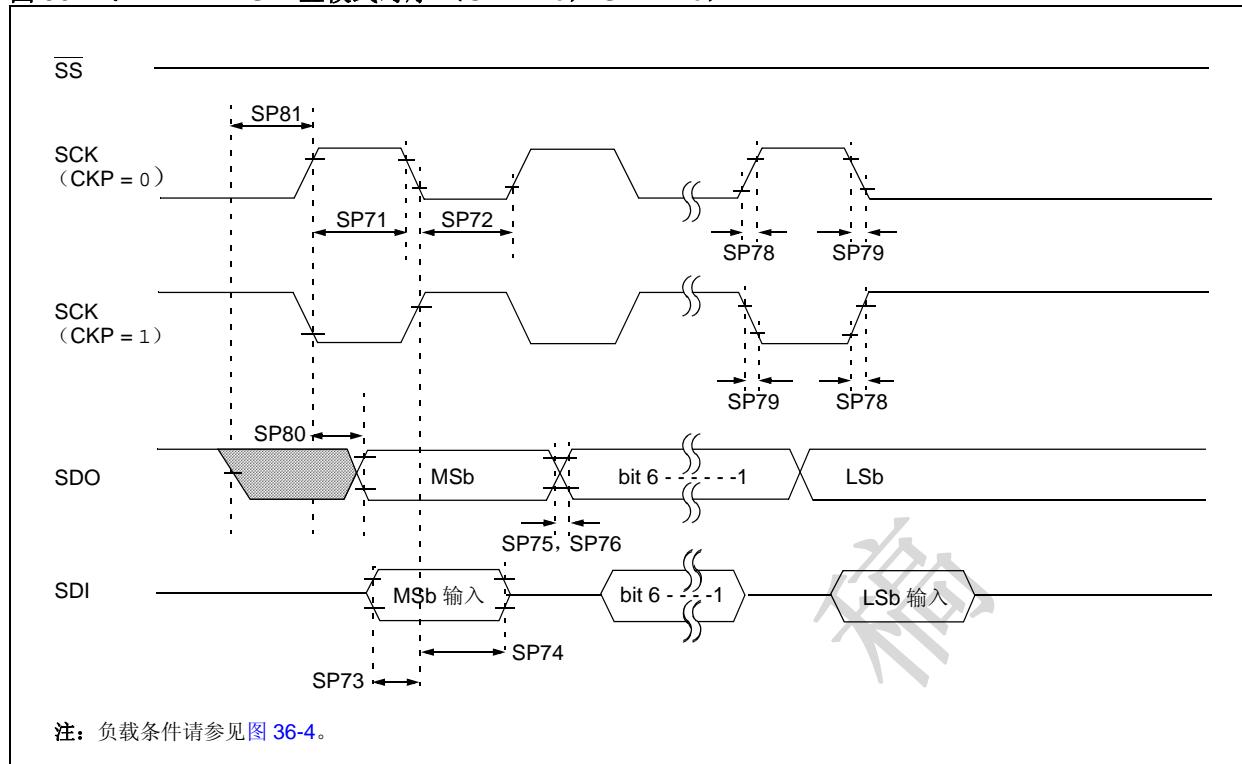


图 36-18: SPI 主模式时序 ($\text{CKE} = 1, \text{SMP} = 1$)

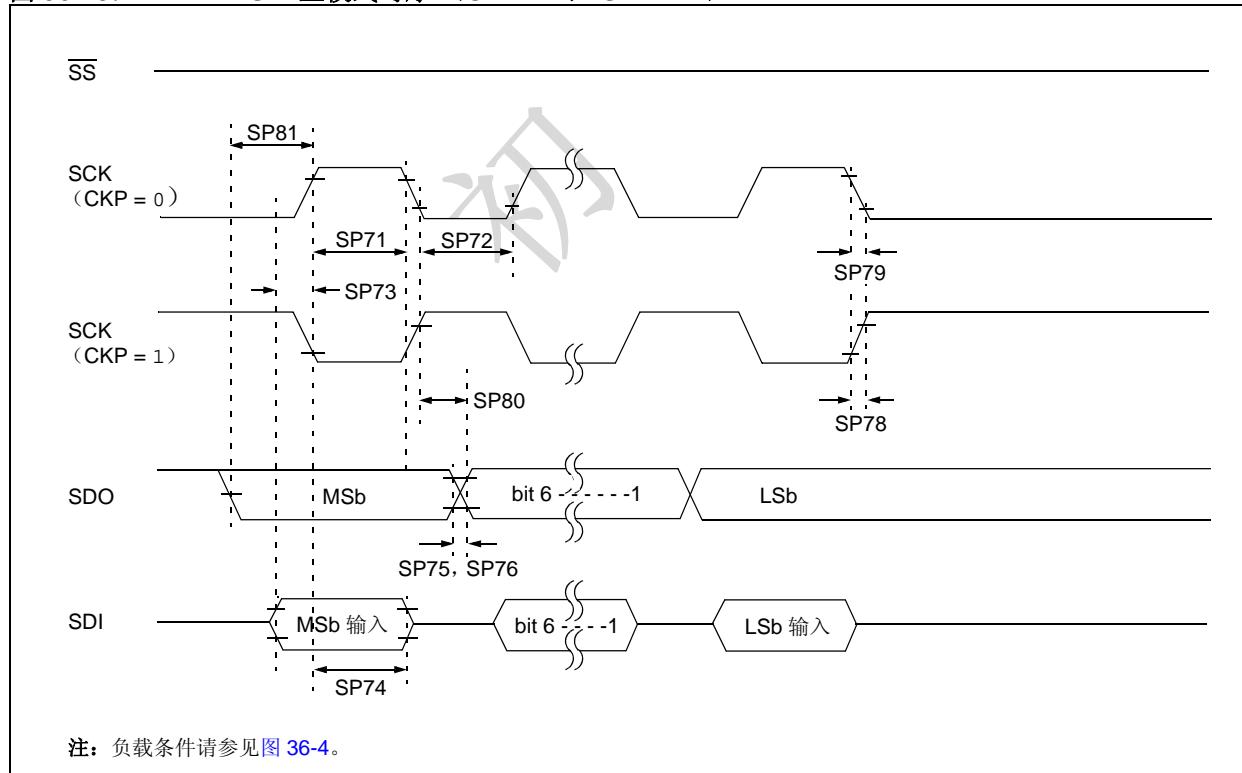


图 36-19: SPI 从模式时序 ($CKE = 0$)

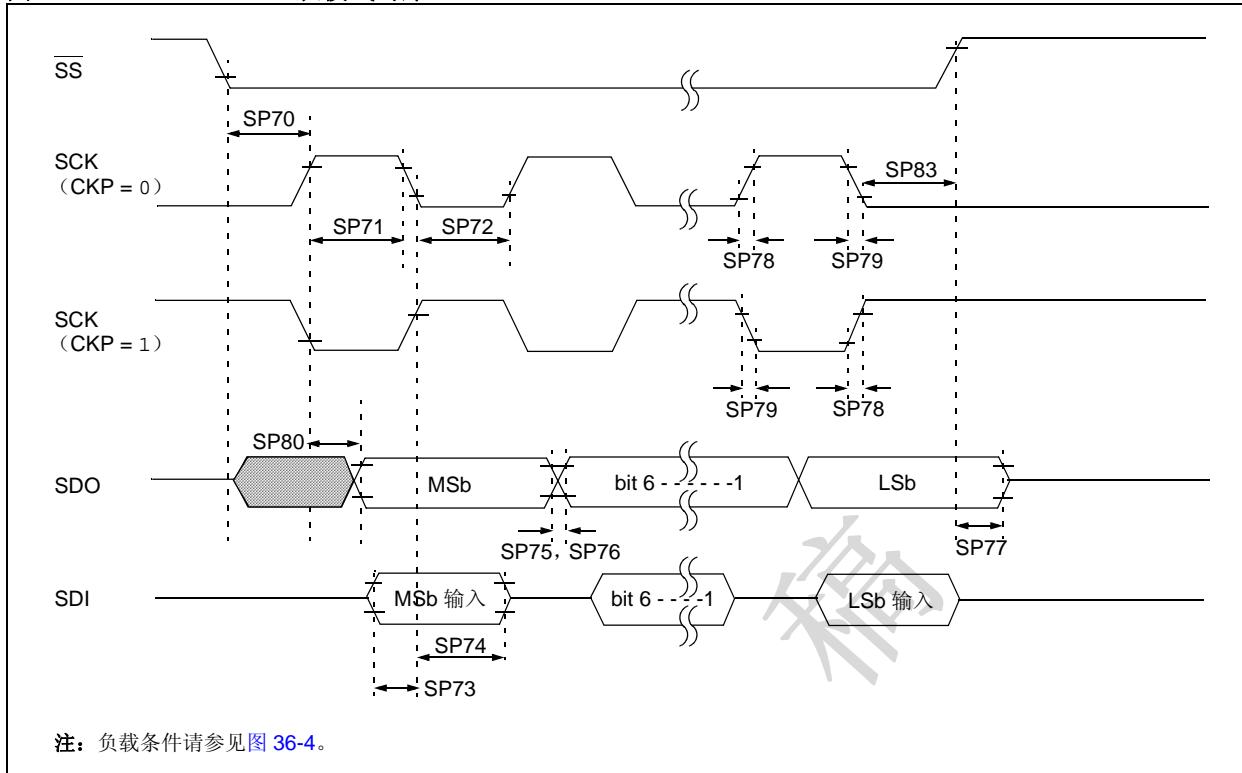
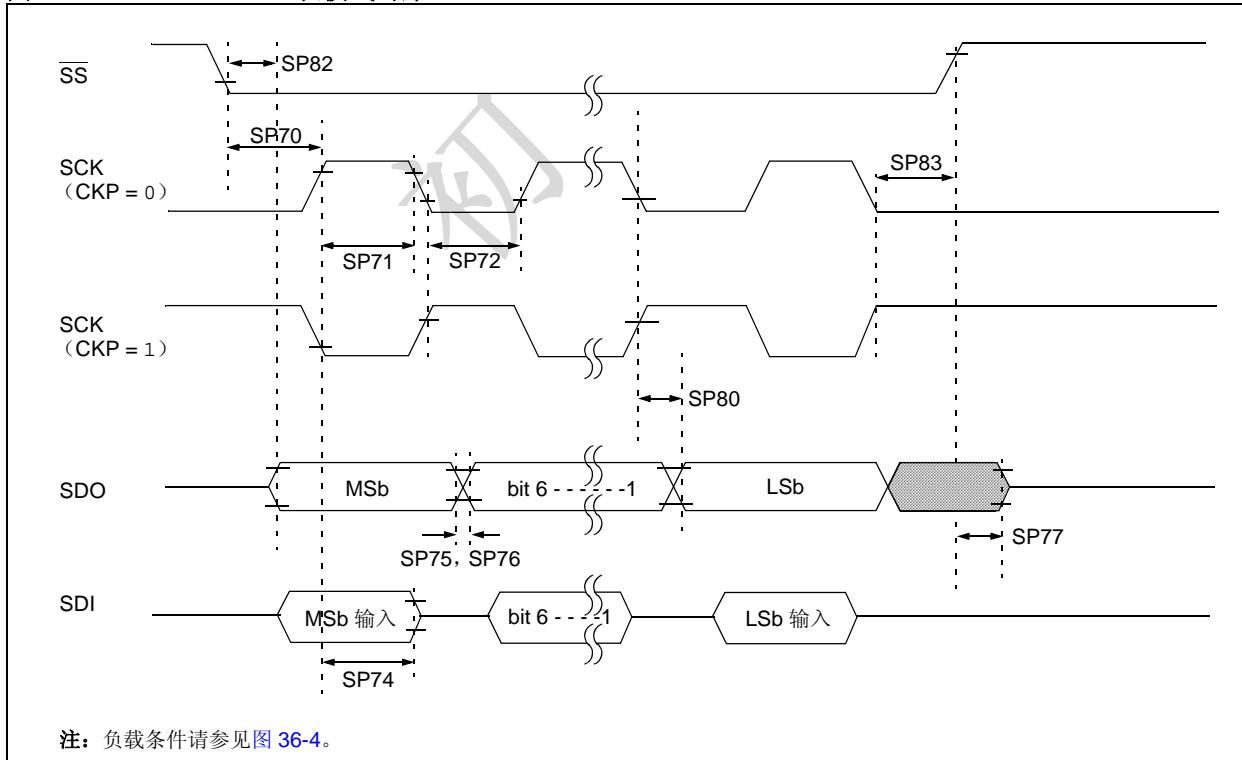


图 36-20: SPI 从模式时序 ($CKE = 1$)



PIC16(L)F1764/5/8/9

表 36-25: SPI 模式要求

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
SP70*	TssL2sCH, TssL2sCL	SS↓ 到 SCK↓ 或 SCK↑ 输入的时间	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*	TssH2D0Z	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*	TscH2D0V, Tscl2D0V	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*	TssL2D0V	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 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。这些参数仅供设计参考，未经测试。

图 36-21: I²C 总线启动位 / 停止位时序

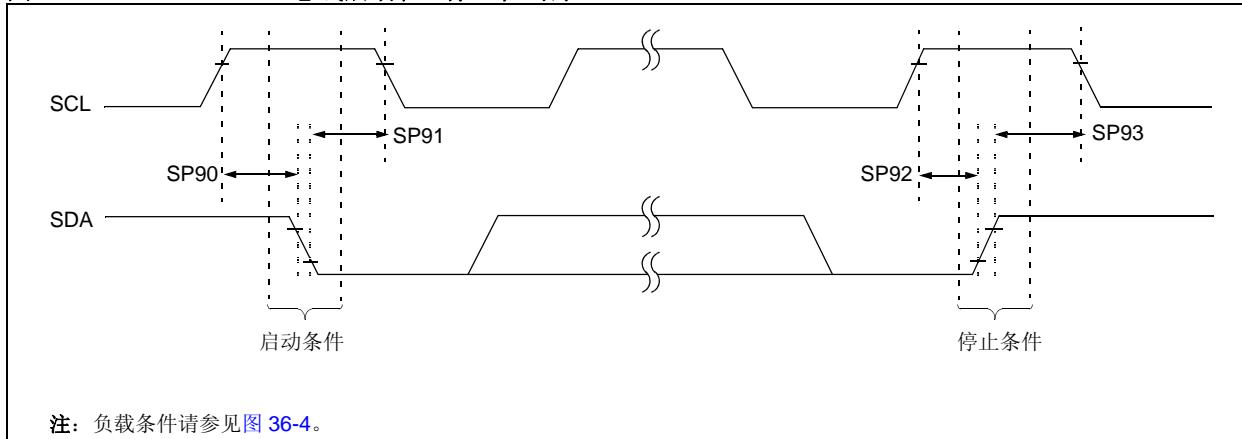
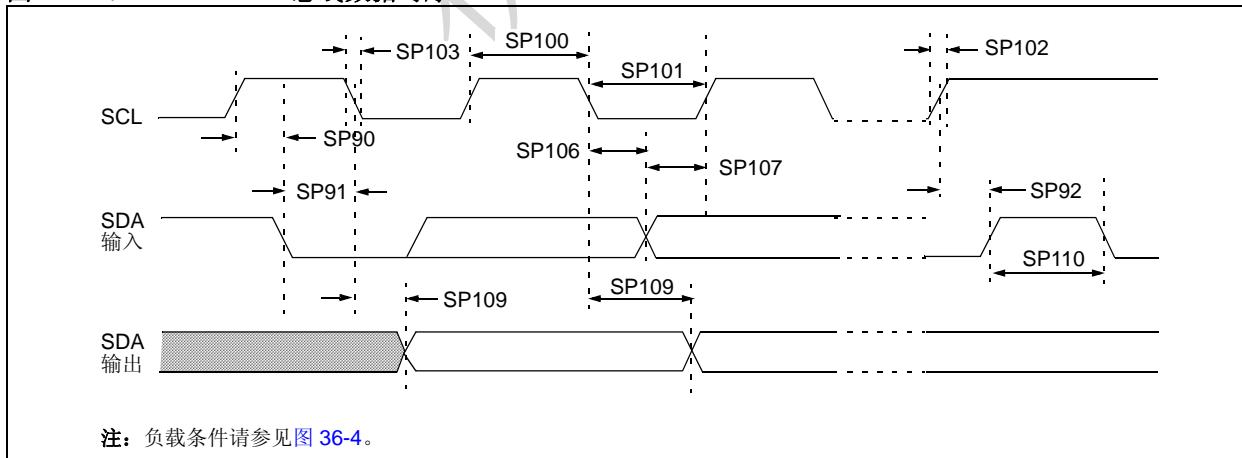


表 36-26: I²C 总线启动位 / 停止位要求

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
SP90*	TSU:STA	启动条件 建立时间	100 kHz 模式 400 kHz 模式	4700 600	—	—	ns 仅与重复启动条件相关
SP91*	THD:STA	启动条件 保持时间	100 kHz 模式 400 kHz 模式	4000 600	—	—	ns 这个周期后产生第一个时钟脉冲
SP92*	TSU:STO	停止条件 建立时间	100 kHz 模式 400 kHz 模式	4700 600	—	—	ns
SP93	THD:STO	停止条件 保持时间	100 kHz 模式 400 kHz 模式	4000 600	—	—	ns

* 这些参数为特性值, 但未经测试。

图 36-22: I²C 总线数据时序



PIC16(L)F1764/5/8/9

表 36-27: I²C 总线数据要求

标准工作条件（除非另外声明）							
参数编号	符号	特性		最小值	最大值	单位	条件
SP100*	THIGH	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	1.5TCY	—		
SP101*	TLLOW	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	1.5TCY	—		
SP102*	TR	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	20 + 0.1CB	300	ns	CB 值规定在 10-400 pF 之间
SP103*	TF	SDA 和 SCL 下降时间	100 kHz 模式	—	250	ns	
			400 kHz 模式	20 + 0.1CB	250	ns	CB 值规定在 10-400 pF 之间
SP106*	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
SP107*	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	(注 2)
			400 kHz 模式	100	—	ns	
SP109*	TAA	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*	TBUF	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	CB	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 $TSU:DAT \geq 250$ 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 信号的低电平周期，它必须将下一个数据位输出到 SDA 线。SCL 线被释放前，根据标准模式 I²C 总线规范， $TR_{max.} + TSU:DAT = 1000 + 250 = 1250$ ns。

37.0 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。所列特性未经测试，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（Typical）”代表 25°C 时的分布的平均值。“最大值（Max.）”和“最小值（Min.）”分别代表（平均值 + 3σ）或（平均值 - 3σ）；其中，σ 是每个温度范围内的标准差。

PIC16(L)F1764/5/8/9

图 37-1: 整个温度范围内的 V_{OH} — I_{OH} 曲线, $VDD = 5.5V$, 仅限 PIC16F1764/5/8/9

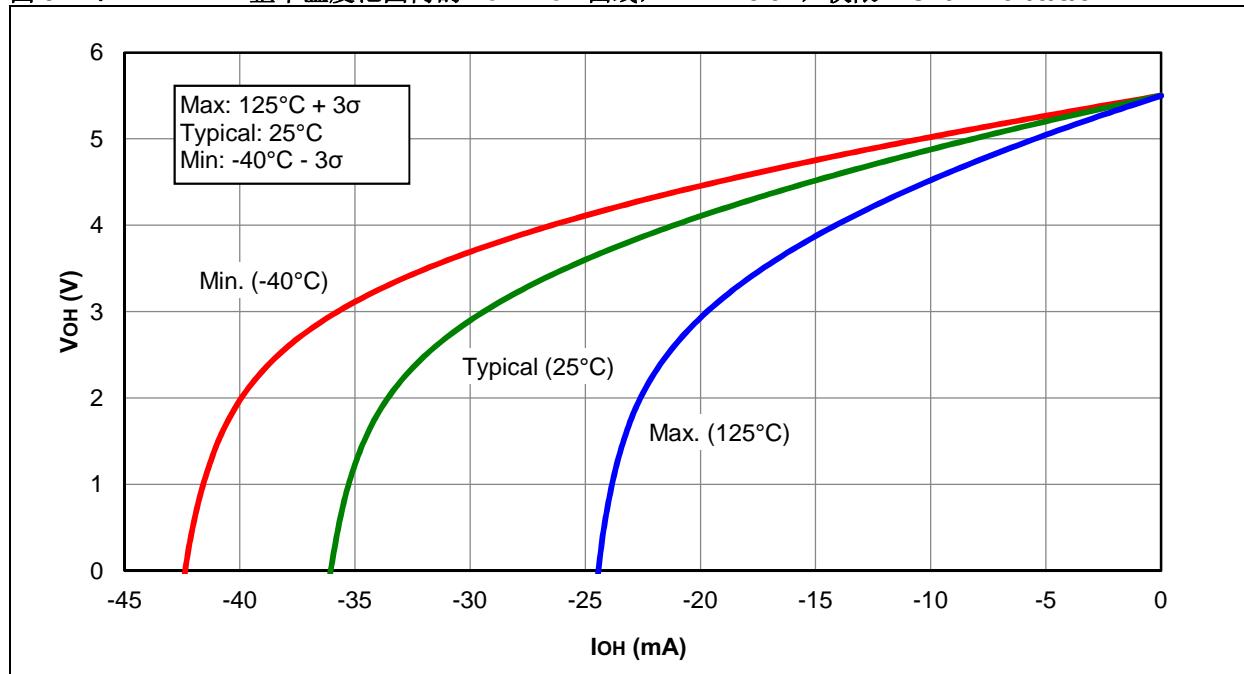


图 37-2: 整个温度范围内的 V_{OL} — I_{OL} 曲线, $VDD = 5.5V$, 仅限 PIC16F1764/5/8/9

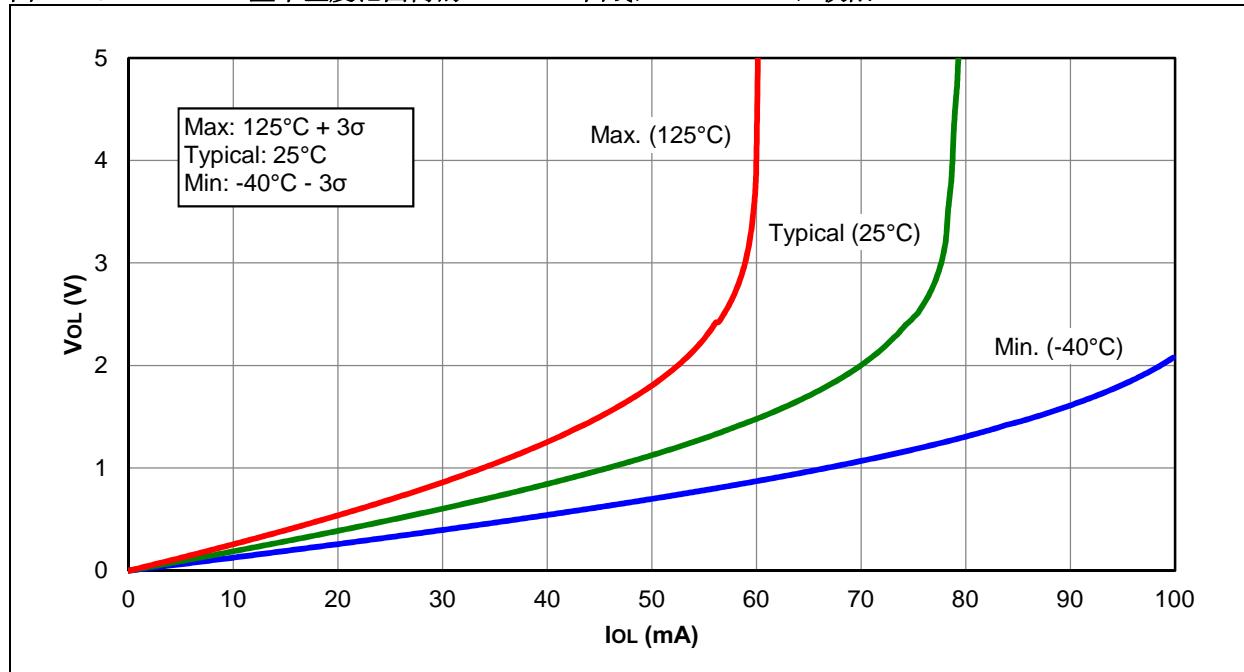


图 37-3: 整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 3.0V$

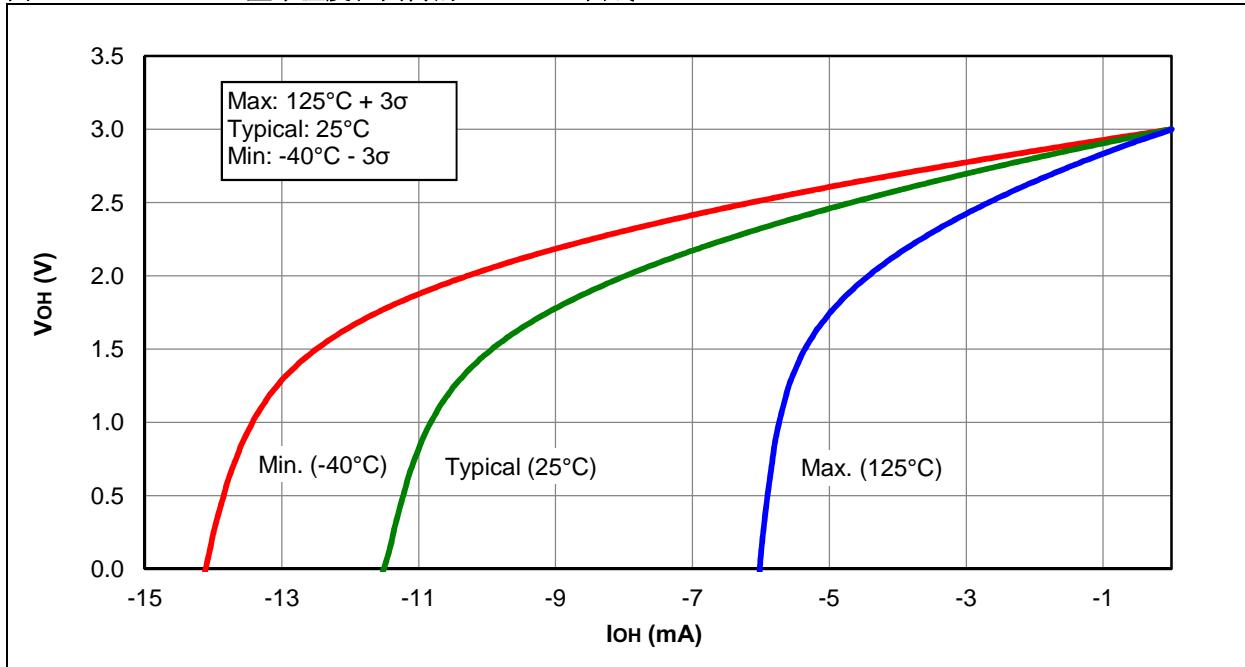
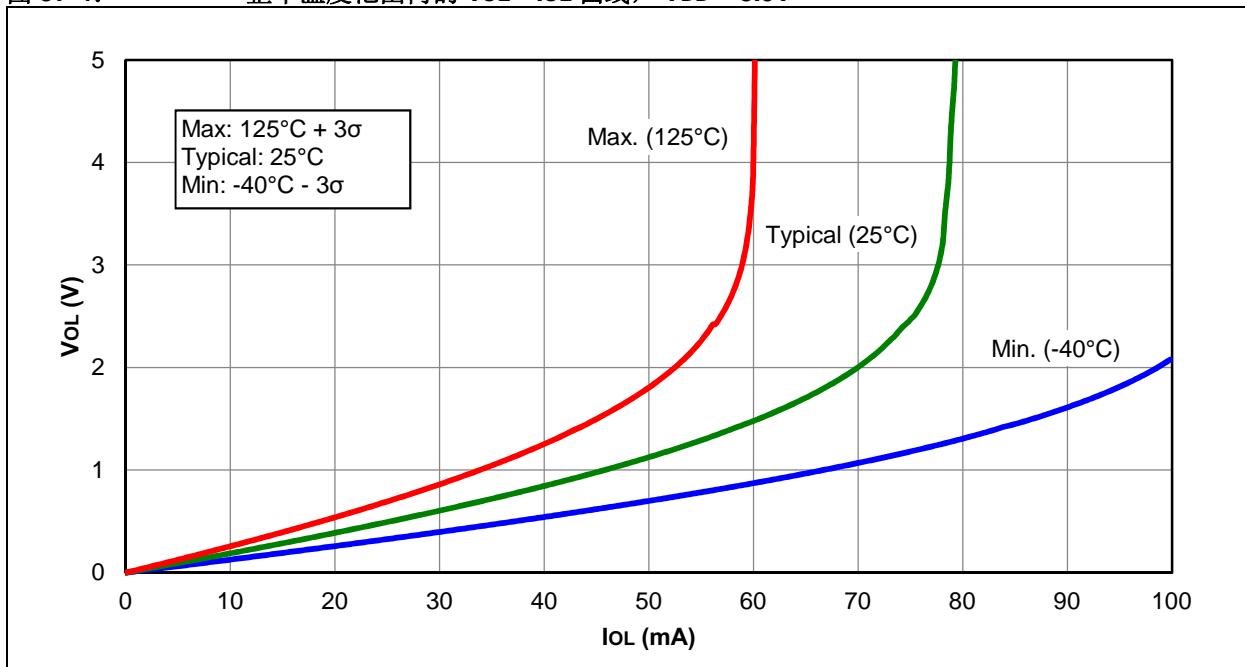


图 37-4: 整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 3.0V$



PIC16(L)F1764/5/8/9

图 37-5: 整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 1.8V$, 仅限 PIC16LF1764/5/8/9

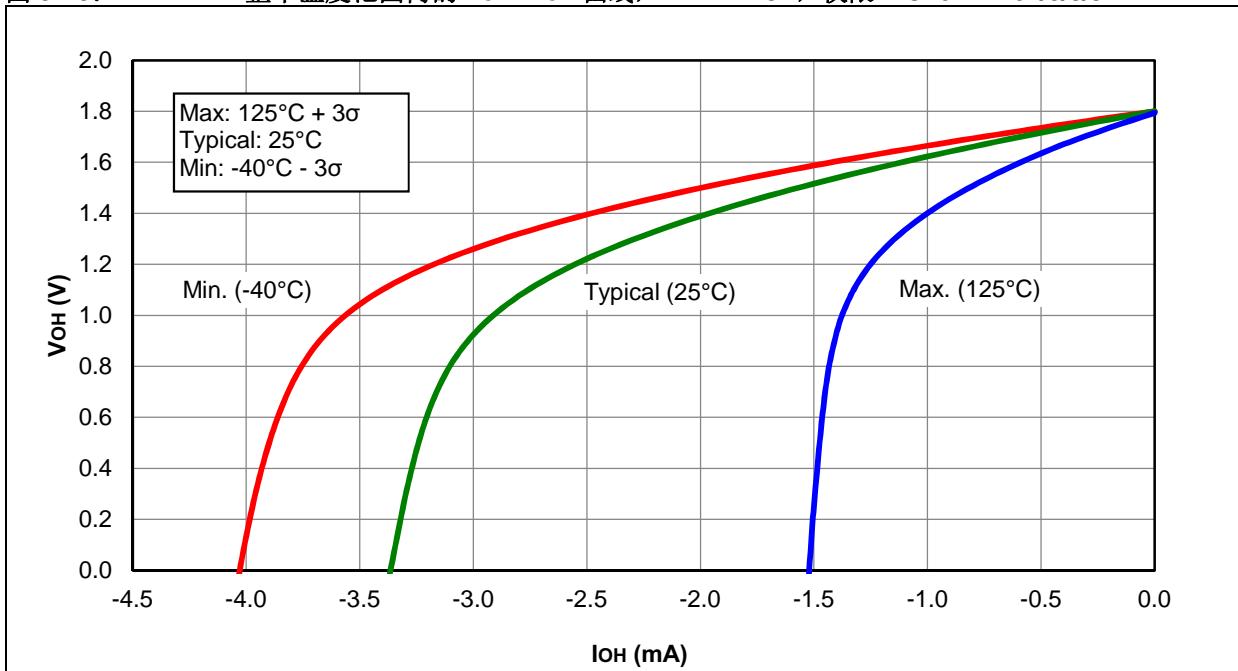


图 37-6: 整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 1.8V$, 仅限 PIC16LF1764/5/8/9

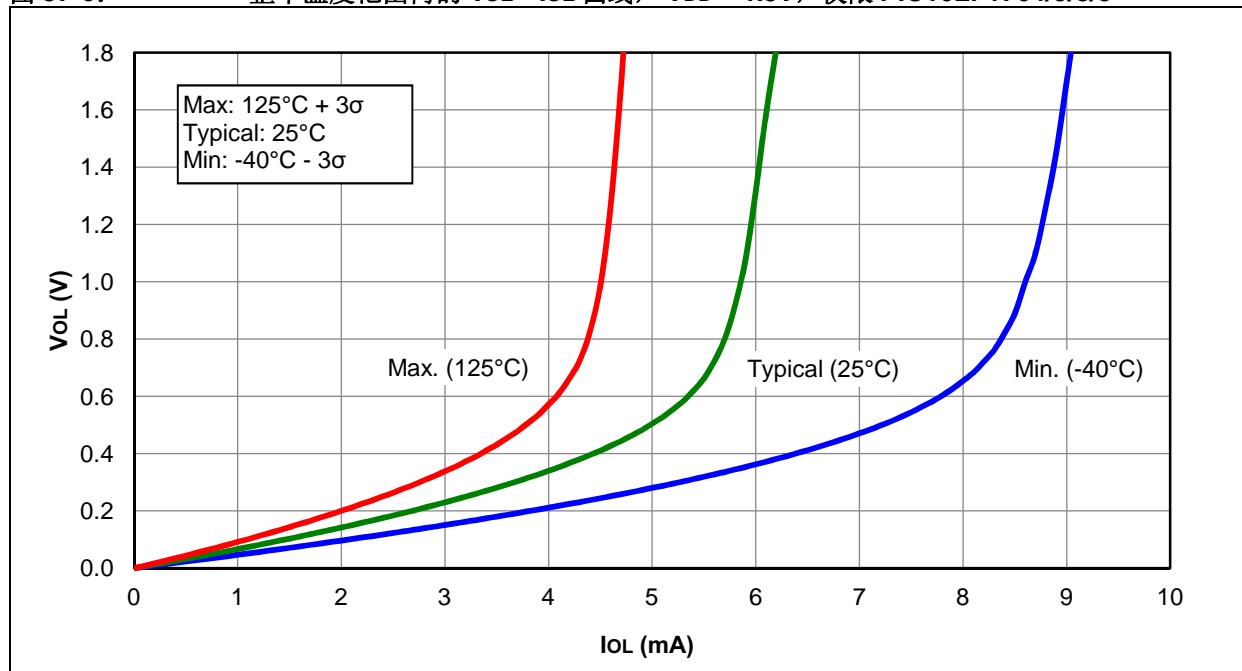


图 37-7: POR 释放电压

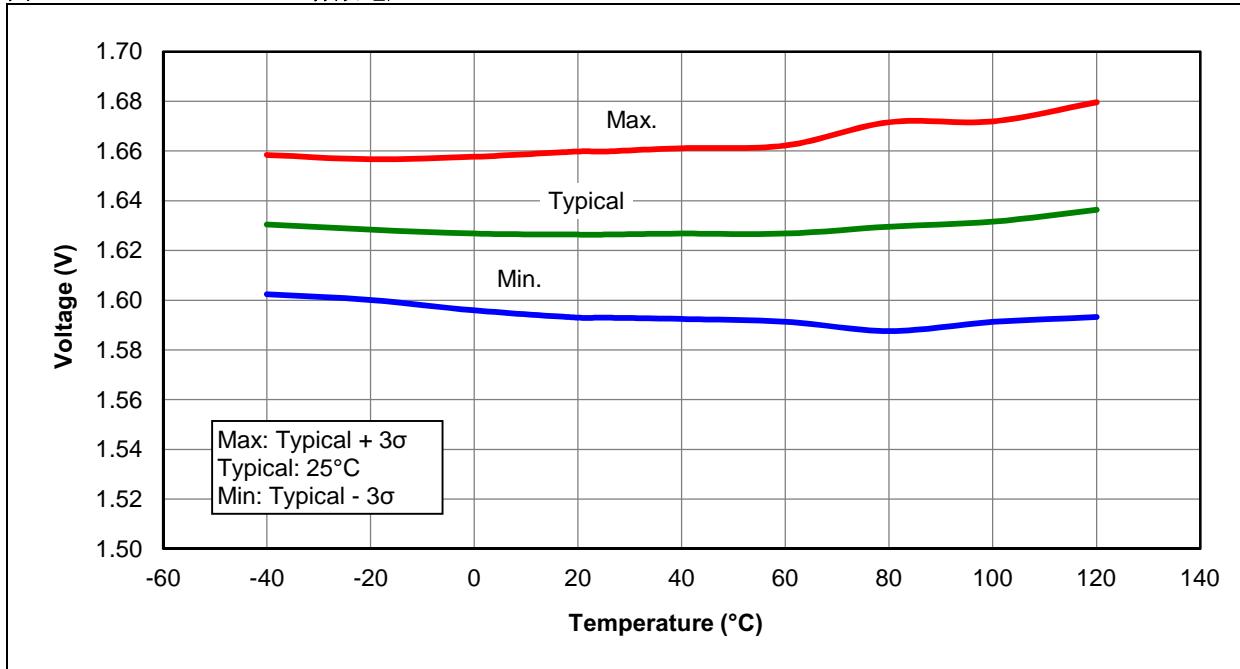
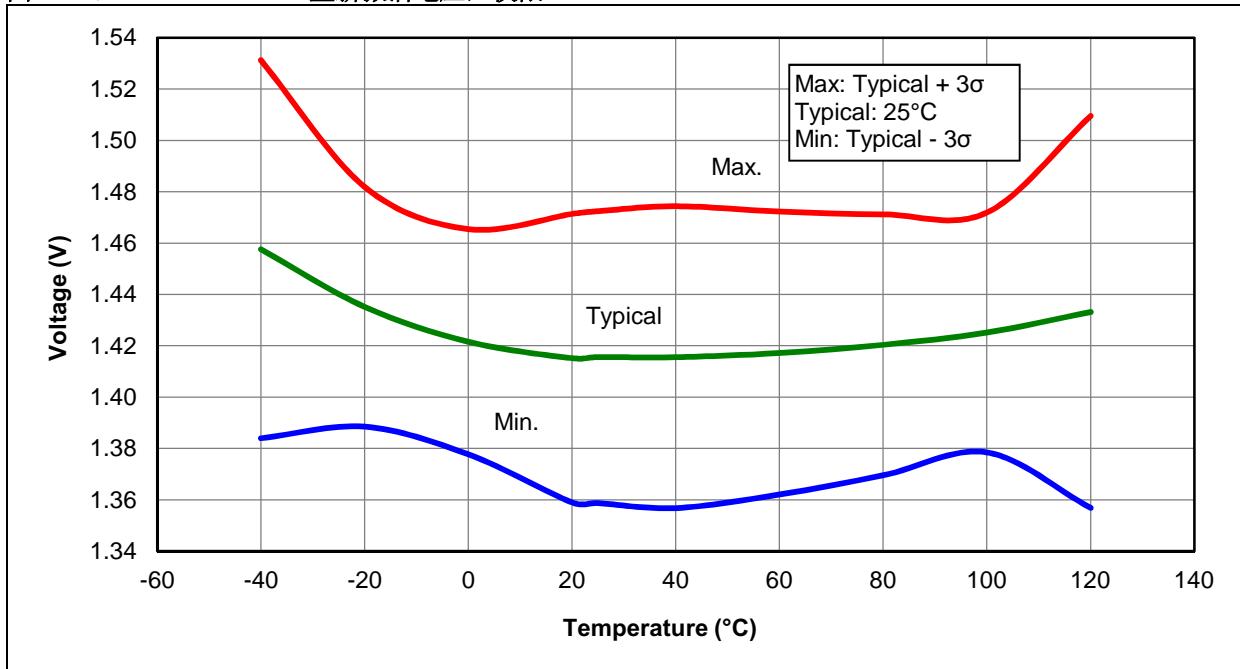


图 37-8: POR 重新激活电压, 仅限 PIC16F1764/5/8/9



PIC16(L)F1764/5/8/9

图 37-9: 欠压复位电压, $BORV = 1$, 仅限 PIC16LF1764/5/8/9

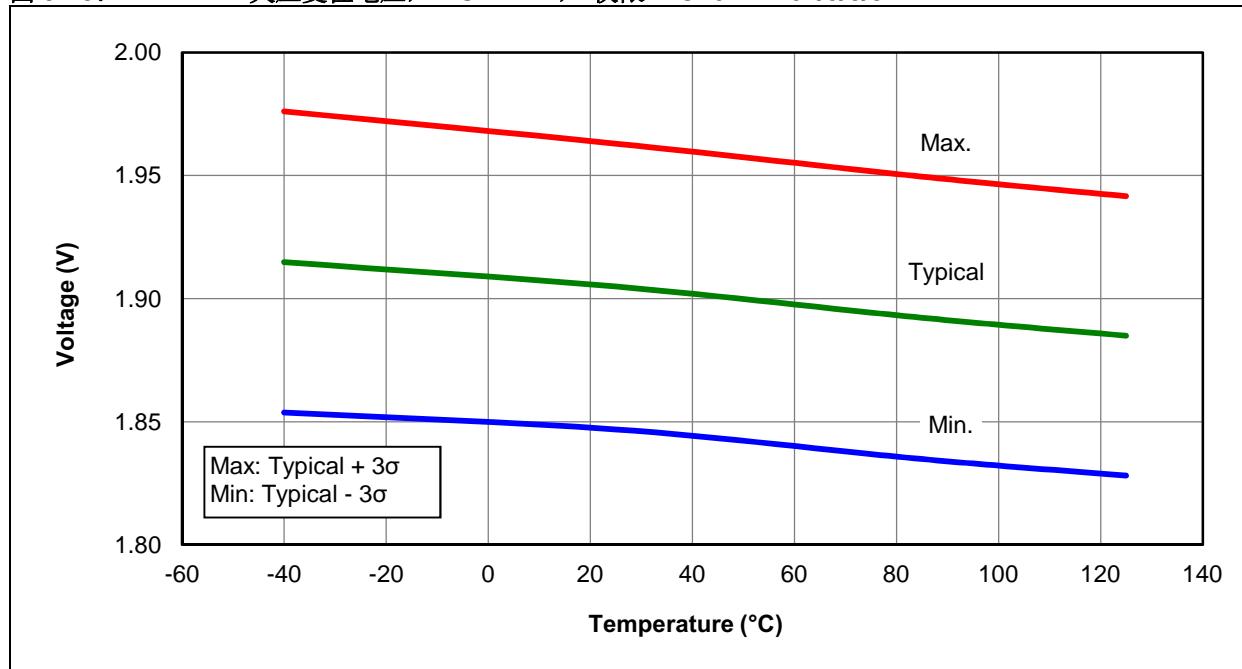


图 37-10: 欠压复位滞后电压, $BORV = 1$, 仅限 PIC16LF1764/5/8/9

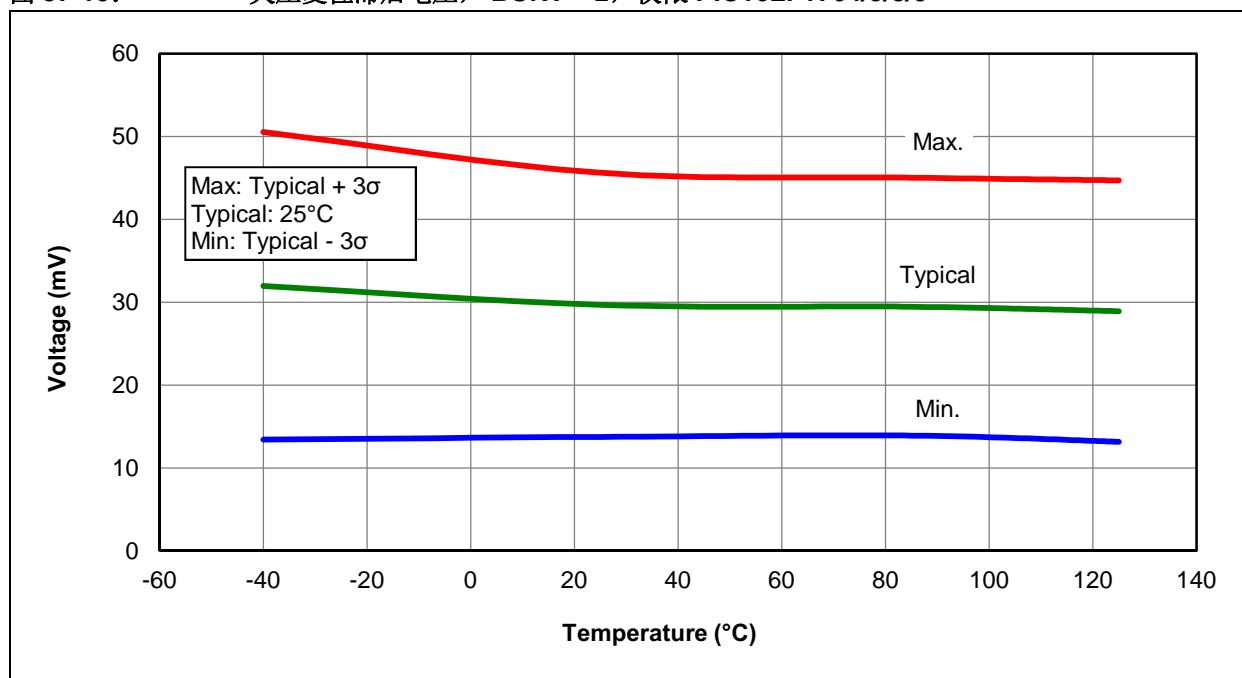


图 37-11: 欠压复位电压, $BORV = 1$, 仅限 PIC16F1764/5/8/9

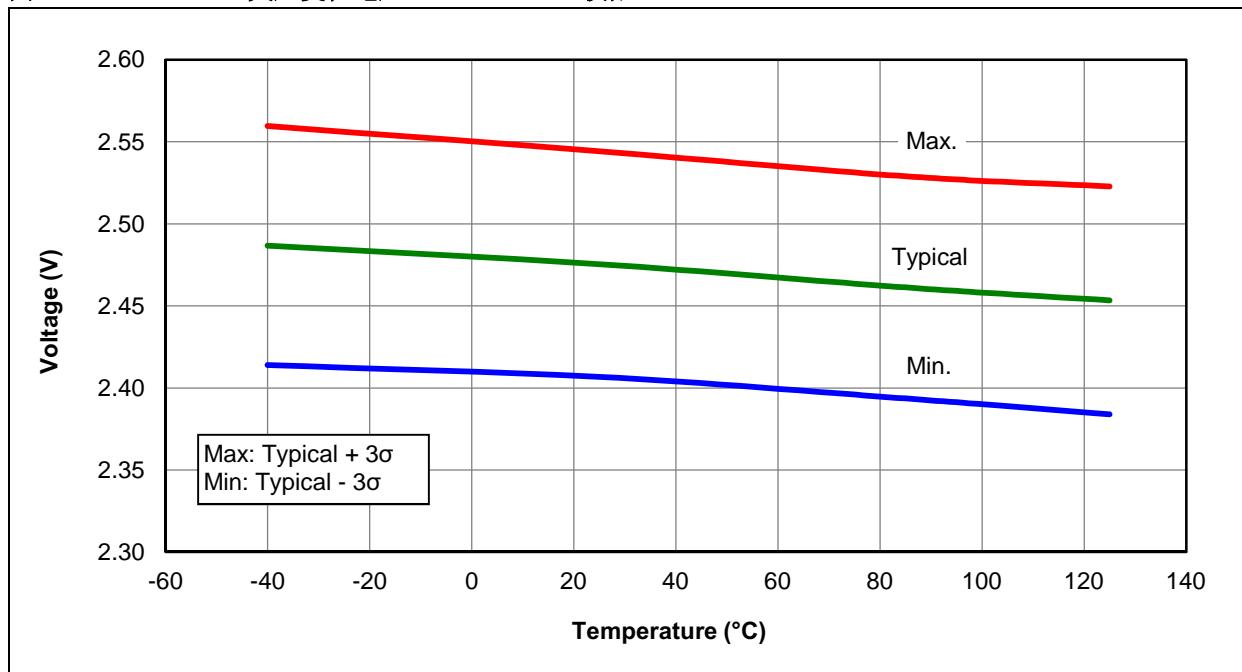
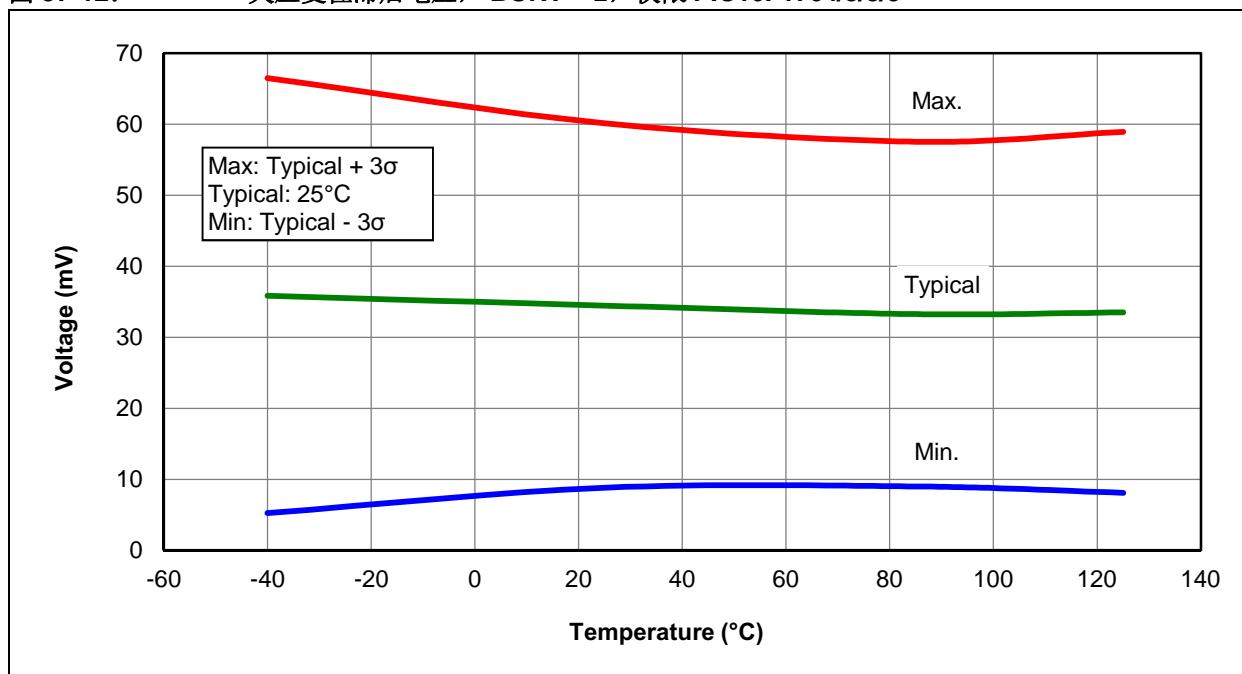


图 37-12: 欠压复位滞后电压, $BORV = 1$, 仅限 PIC16F1764/5/8/9



PIC16(L)F1764/5/8/9

图 37-13: 欠压复位电压, $BORV = 0$

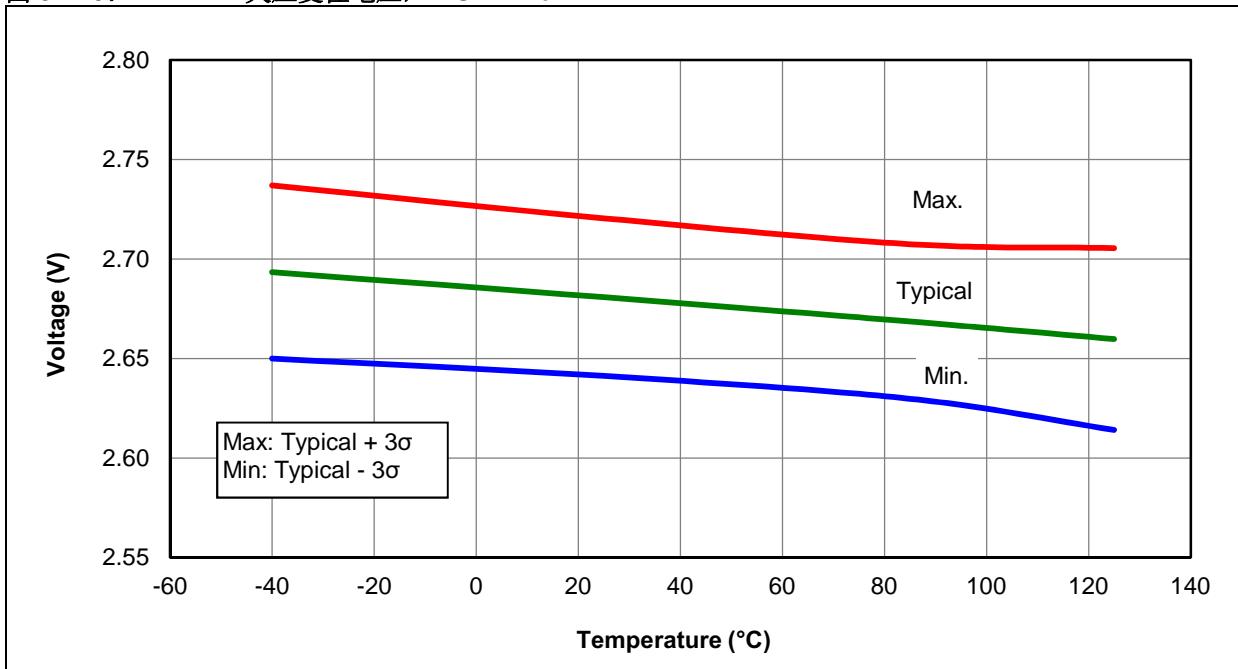


图 37-14: 欠压复位滞后电压, $BORV = 0$

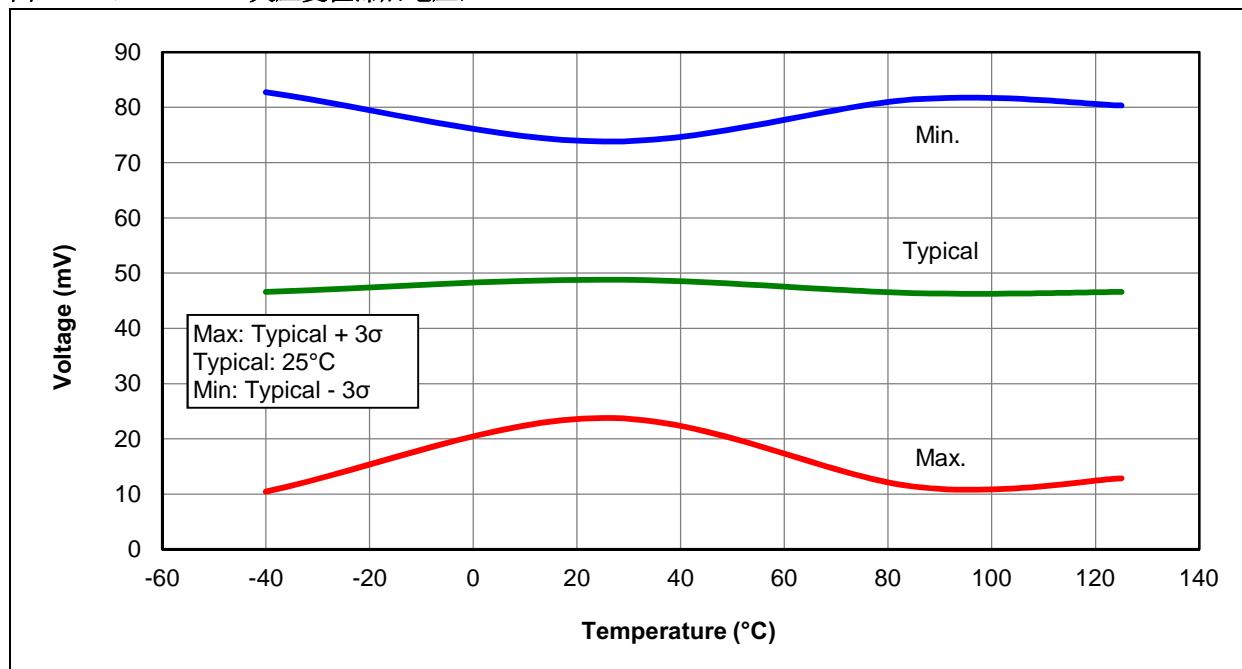


图 37-15: 低功耗欠压复位电压, LPBOR = 0

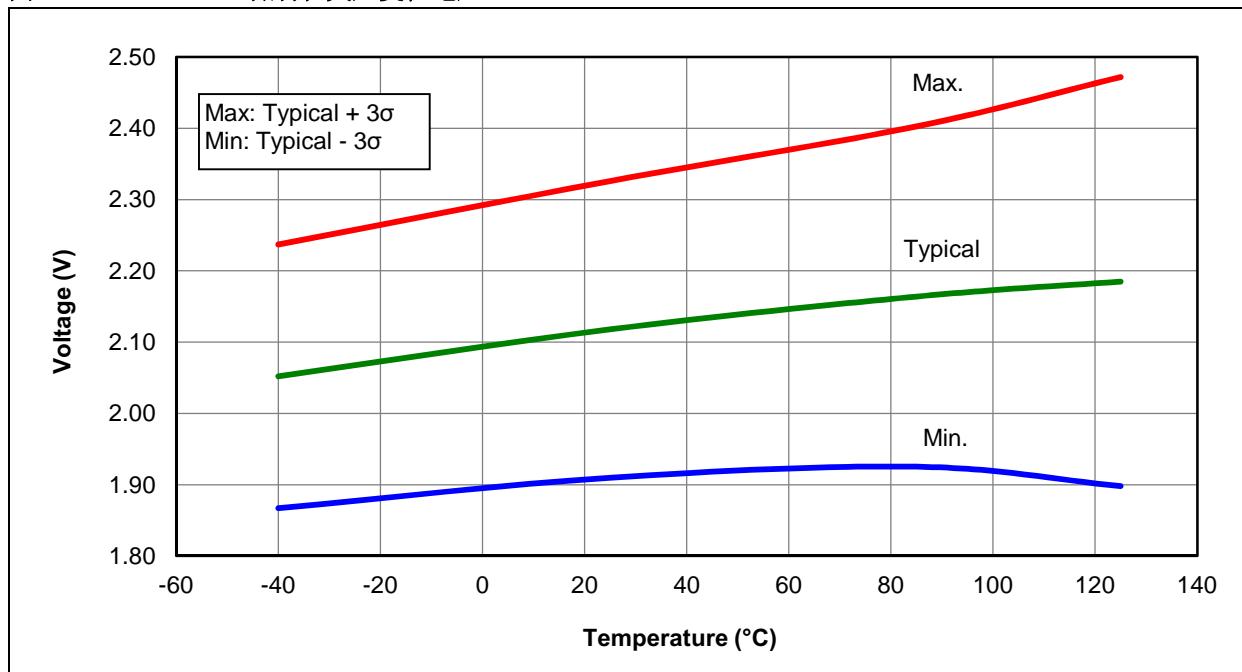
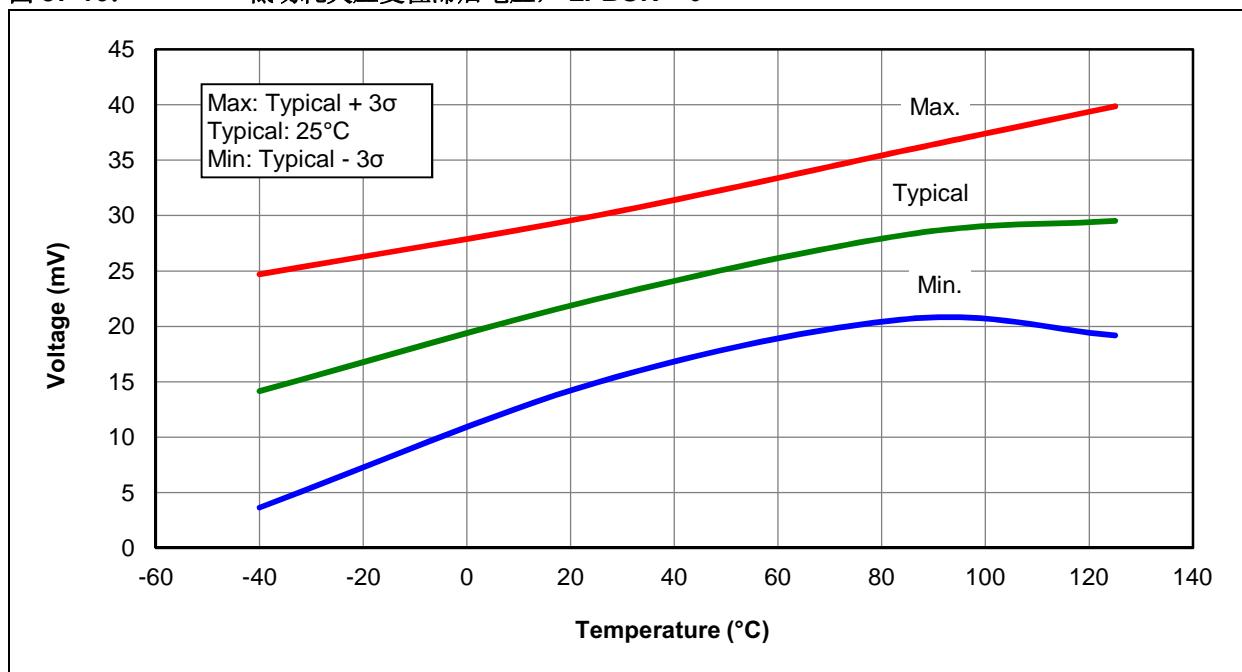


图 37-16: 低功耗欠压复位滞后电压, LPBOR = 0



PIC16(L)F1764/5/8/9

图 37-17: WDT 超时周期

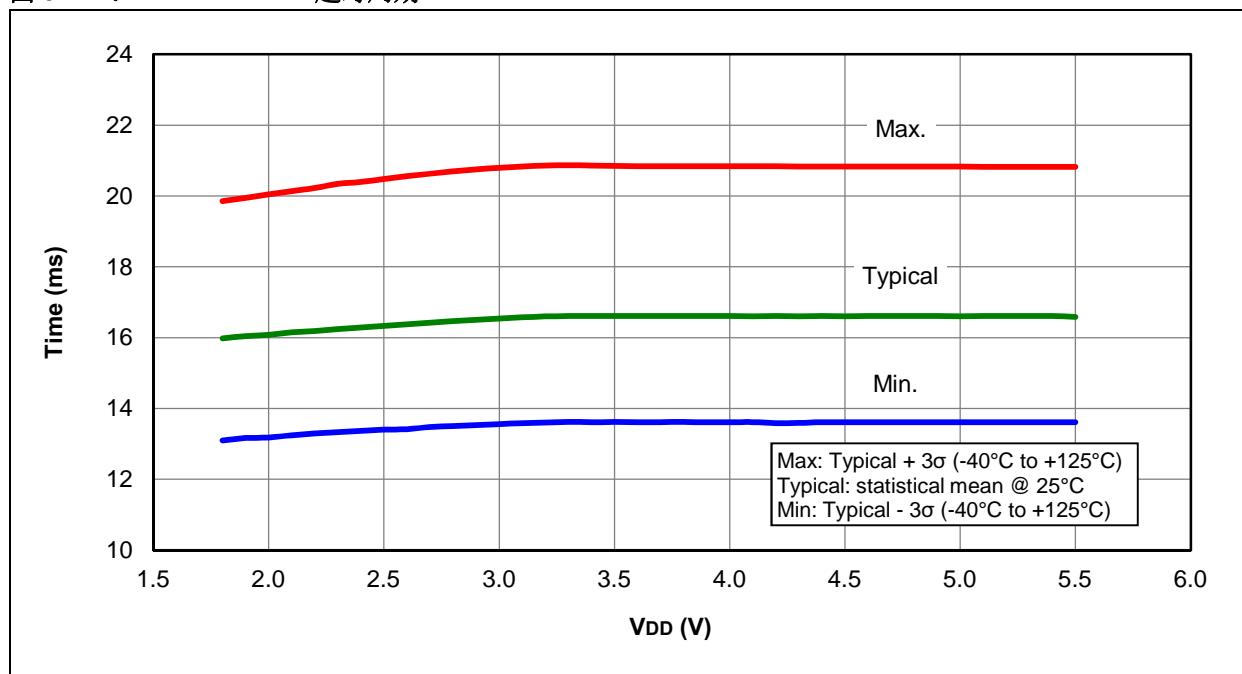


图 37-18: PWRT 周期

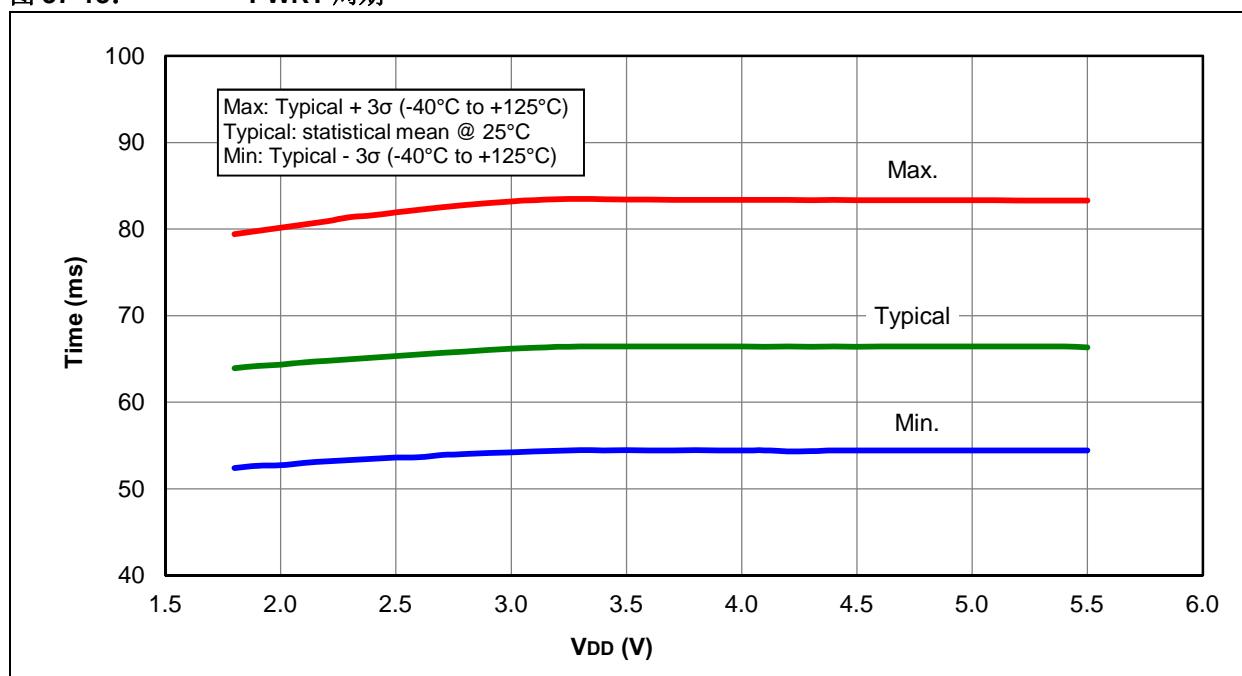


图 37-19: FVR 稳定周期

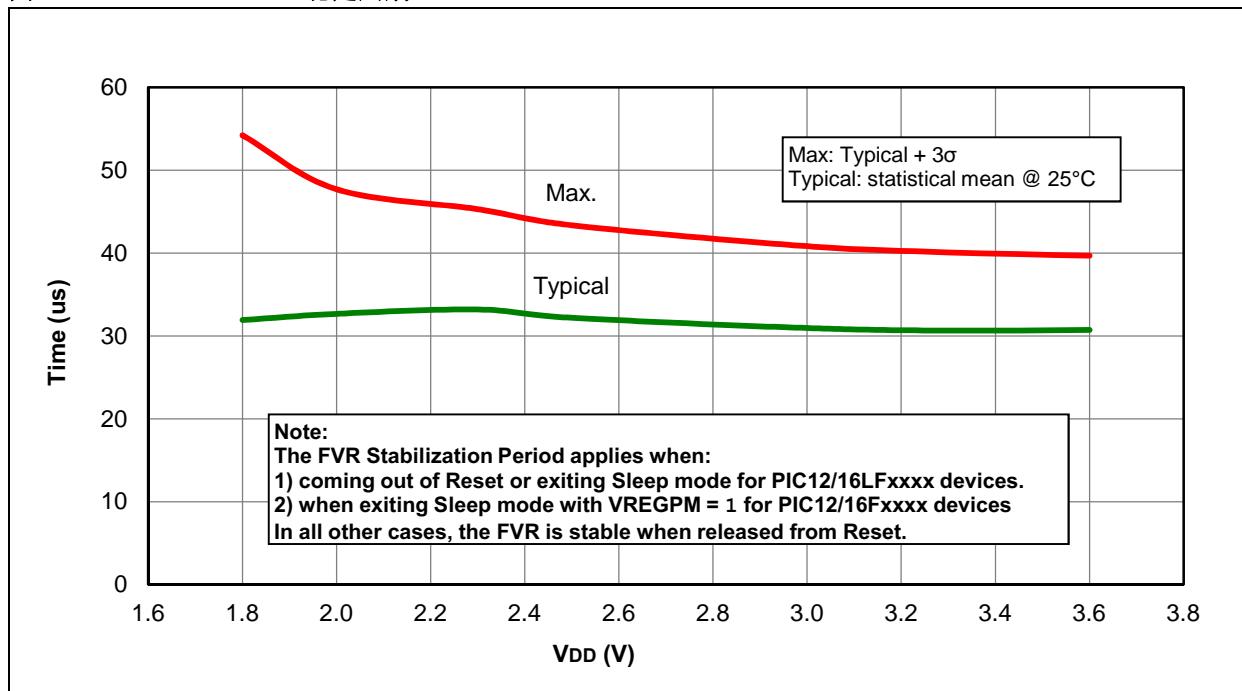
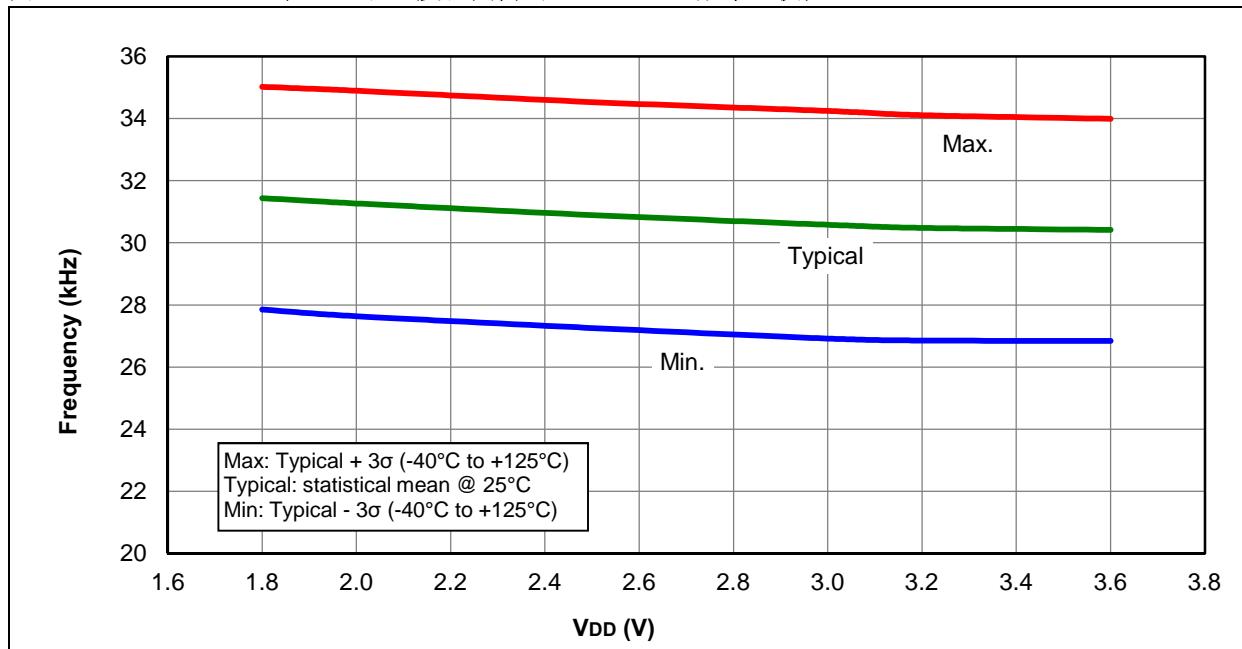


图 37-20: 整个 VDD 和温度范围内的 LFINTOSC 频率，仅限 PIC16LF1764/5/8/9



PIC16(L)F1764/5/8/9

图 37-21：整个 V_{DD} 和温度范围内的 LFINTOSC 频率，仅限 PIC16F1764/5/8/9

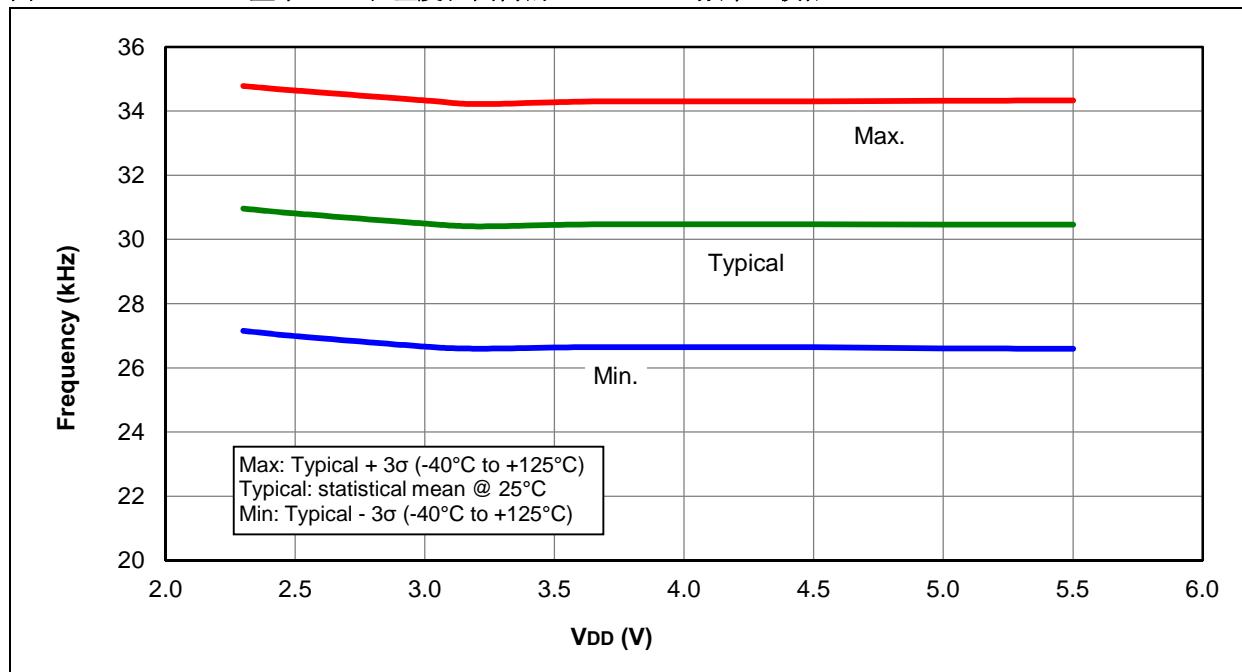


图 37-22：休眠模式，唤醒周期与 HFINTOSC 源，仅限 PIC16LF1764/5/8/9

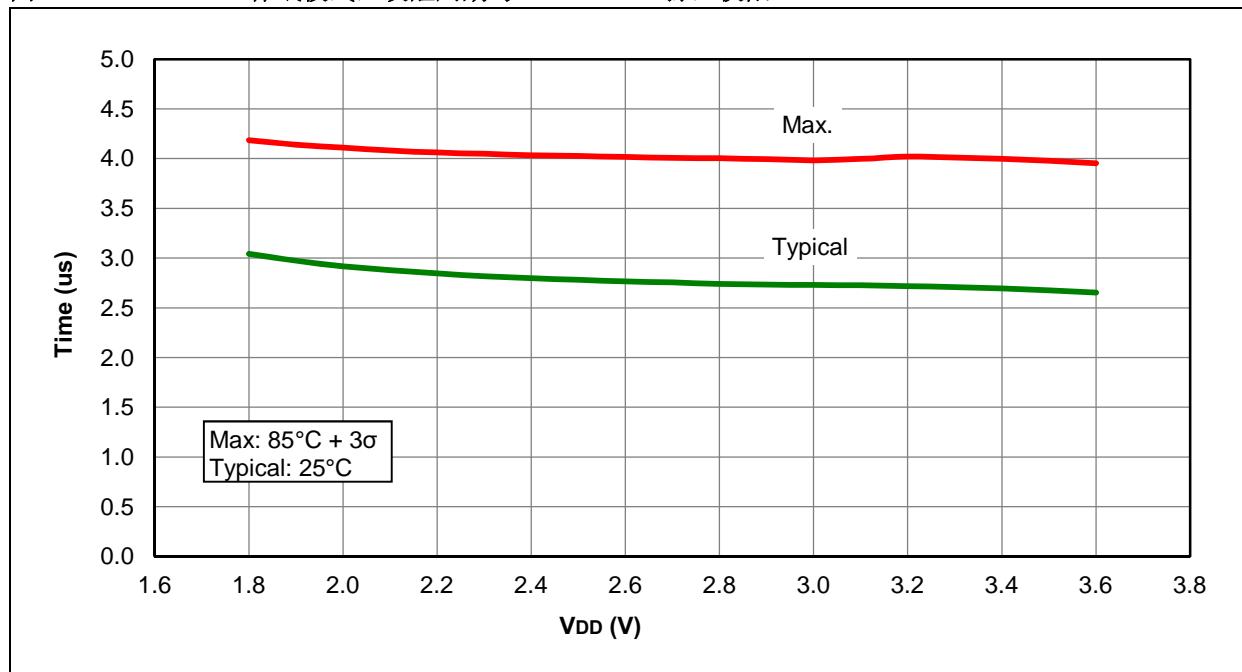


图 37-23: 低功耗休眠模式, 唤醒周期与 HFINTOSC 源, VREGPM = 1, 仅限 PIC16F1764/5/8/9

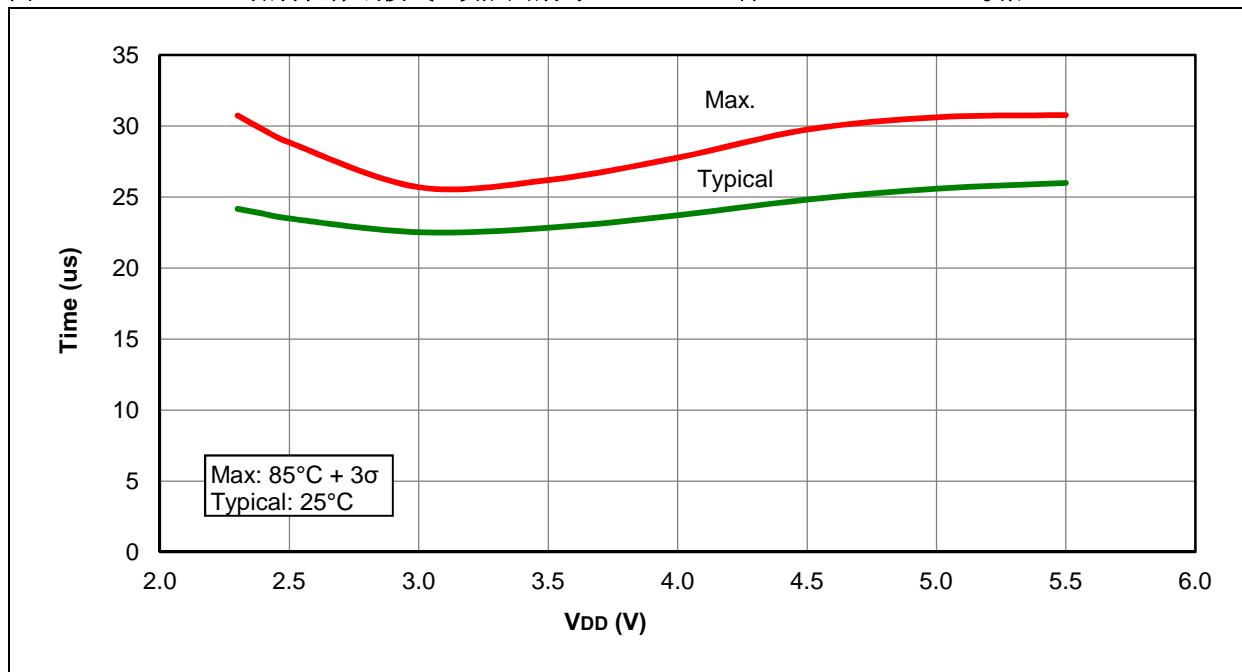
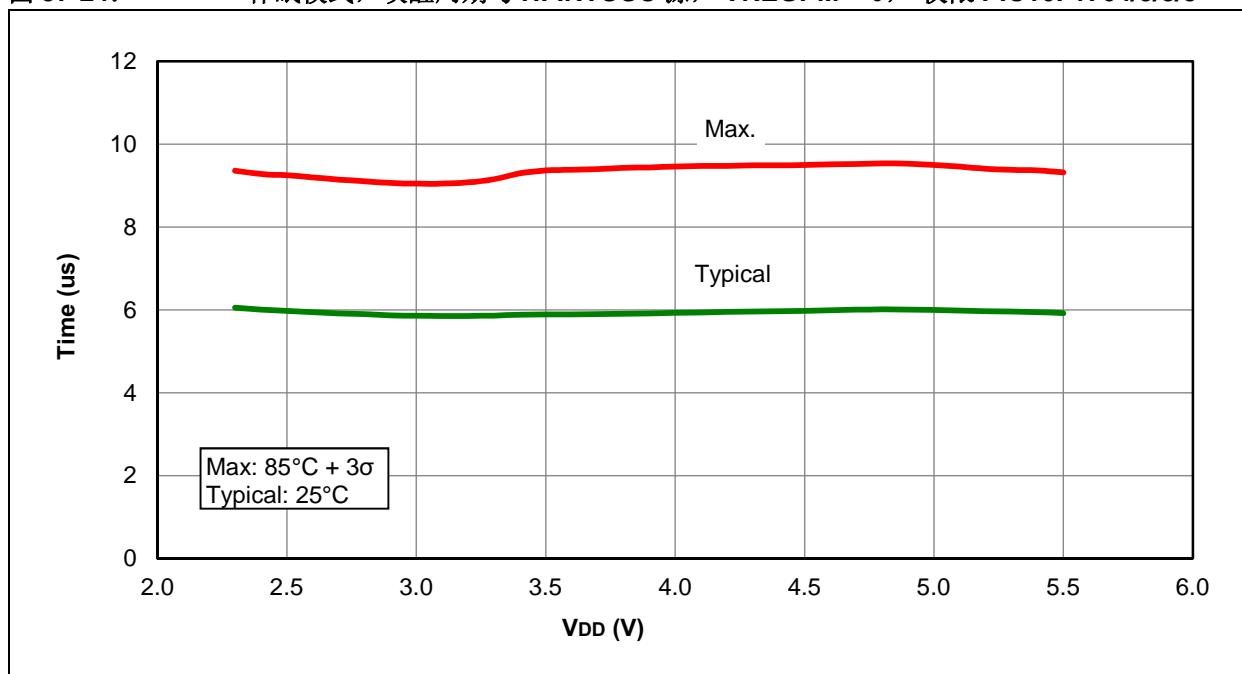


图 37-24: 休眠模式, 唤醒周期与 HFINTOSC 源, VREGPM = 0, 仅限 PIC16F1764/5/8/9



38.0 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
- 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

38.1 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

38.2 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

38.3 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

38.4 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，只有包含该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

38.5 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

38.6 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。

38.7 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（RJ-11）或新型抗噪声、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。

38.8 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（RJ-11）与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 连接器。

38.9 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）。

38.10 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。

38.11 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)。

38.12 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。

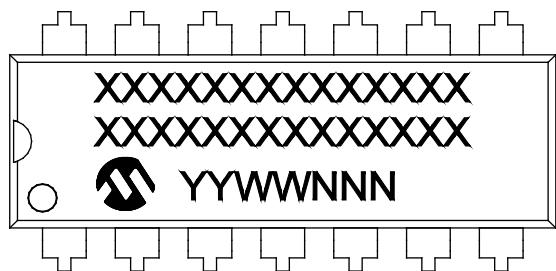
- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 IPLogika® 等公司提供的嵌入式以太网解决方案

PIC16(L)F1764/5/8/9

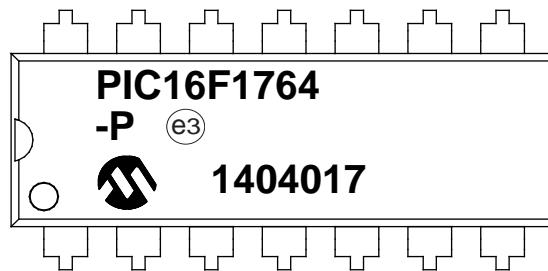
39.0 封装信息

39.1 封装标识信息

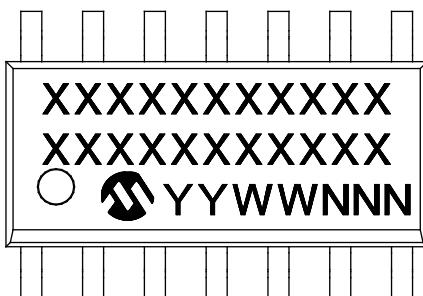
14 引脚 PDIP (300 mil)



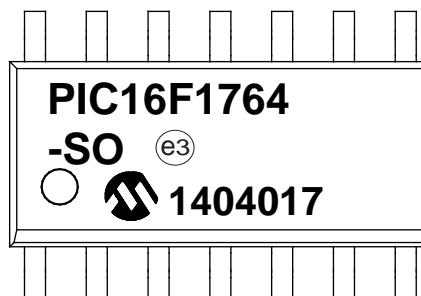
示例



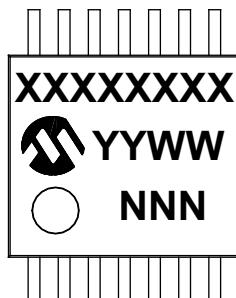
14 引脚 SOIC (3.90 mm)



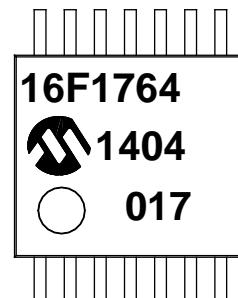
示例



14 引脚 TSSOP (4.4 mm)



示例

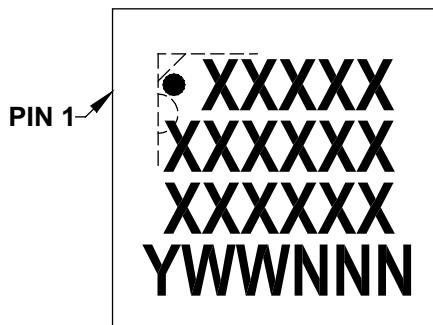


图注:	XX...X	客户指定信息
	Y	年份代码（日历年的最后一位数字）
	YY	年份代码（日历年的最后两位数字）
	WW	星期代码（一月一日的星期代码为“01”）
	NNN	以字母数字排序的追踪代码
	(e3)	雾锡（Matte Tin, Sn）的 JEDEC 无铅标志
*		本封装为无铅封装。JEDEC 无铅标志（e3）标示于此种封装的外包装上。

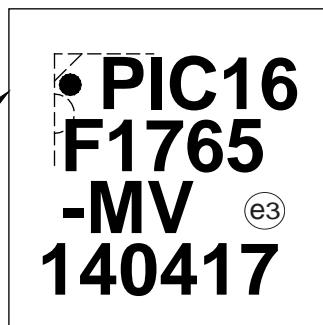
注: Microchip 部件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

封装标识信息（续）

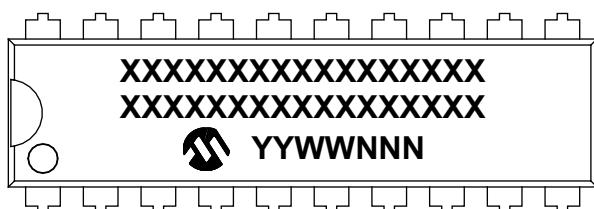
16 引脚 QFN (4x4x0.9 mm)



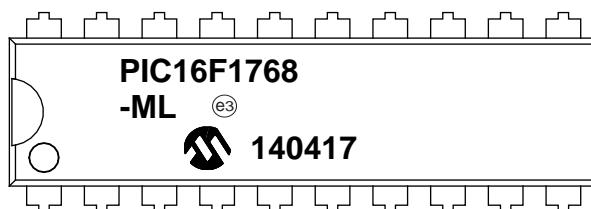
示例



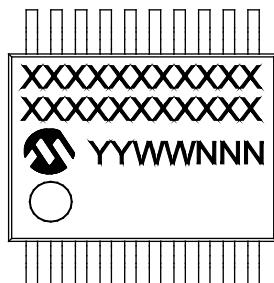
20 引脚 PDIP (300 mil)



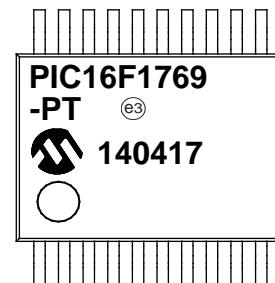
示例



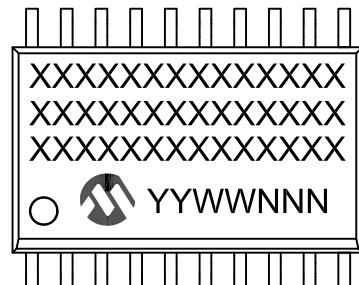
20 引脚 SSOP (5.30 mm)



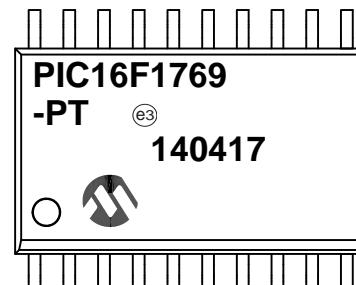
示例



20 引脚 SOIC (7.50 mm)



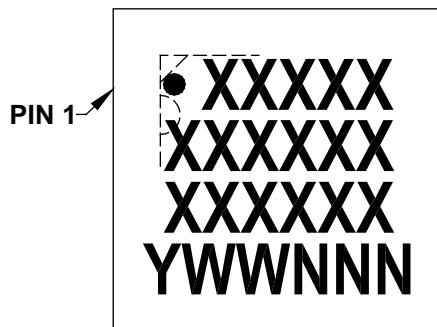
示例



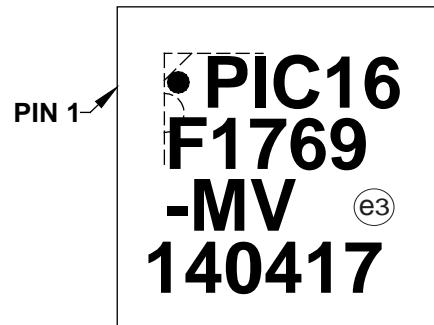
PIC16(L)F1764/5/8/9

封装标识信息（续）

20 引脚 QFN (4x4x0.9 mm)



示例

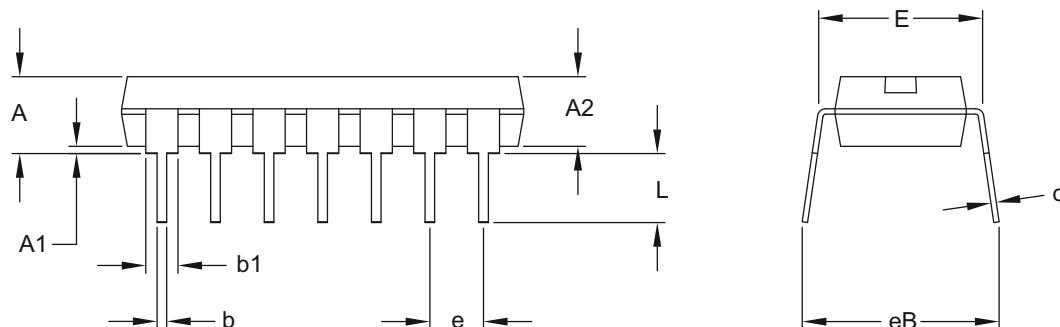
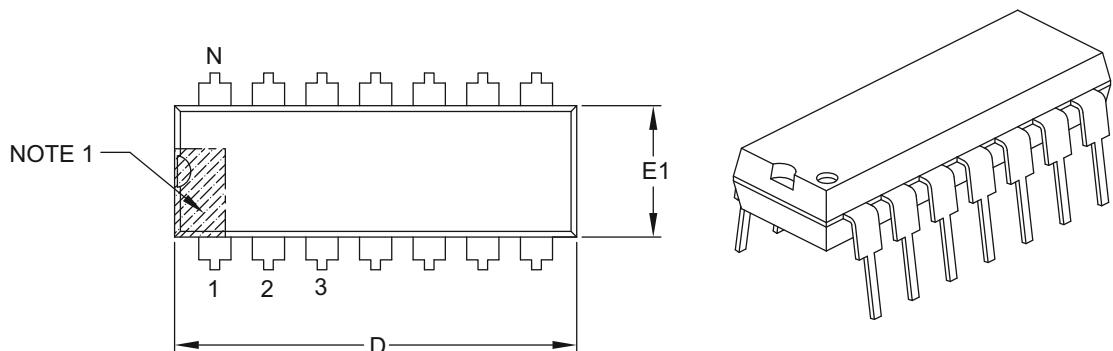


39.2 封装详细信息

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。



	Units	INCHES		
	Dimension Limits	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:

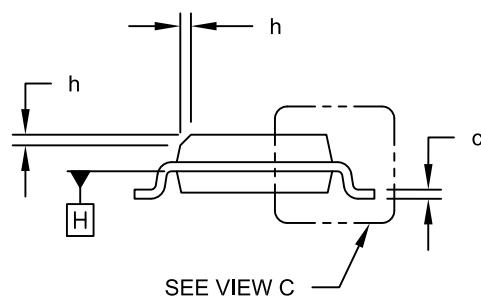
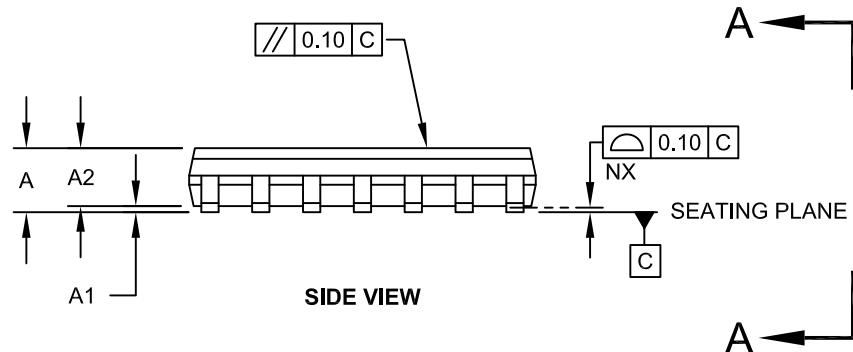
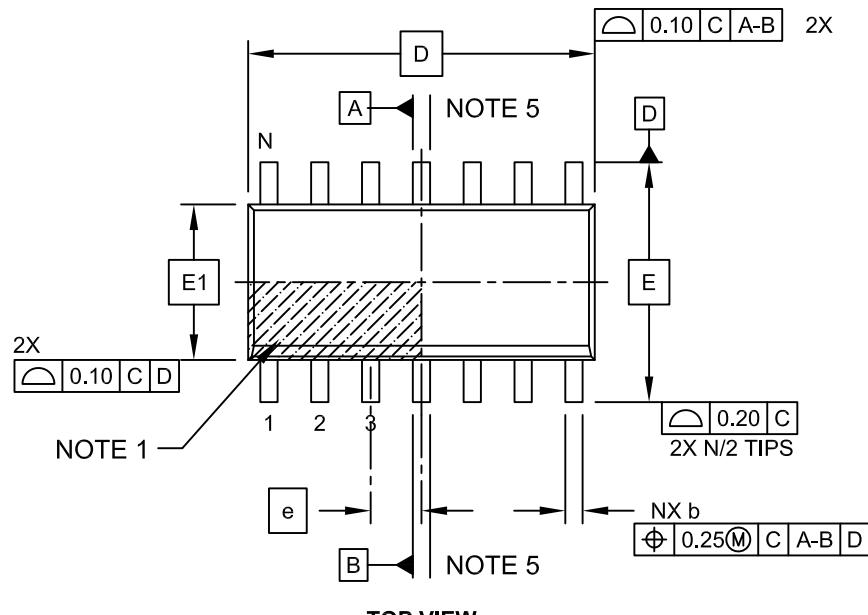
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

PIC16(L)F1764/5/8/9

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。

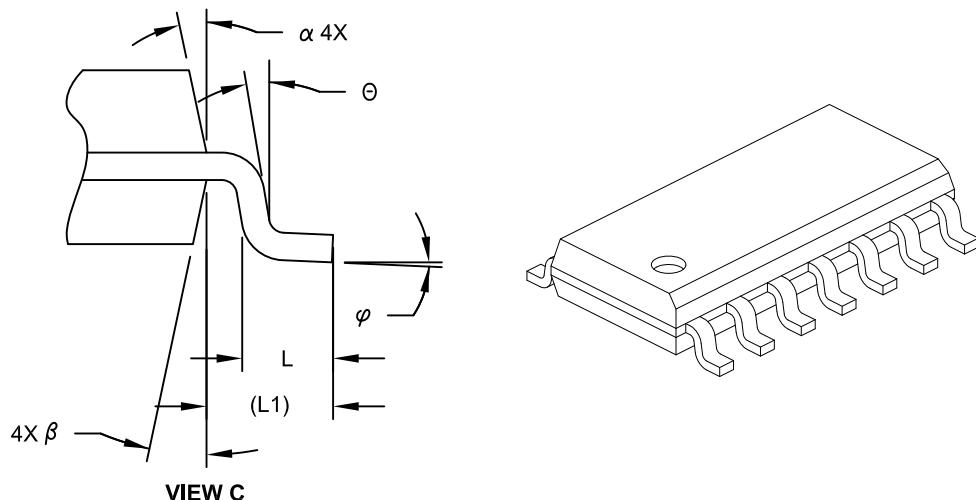


VIEW A-A

Microchip Technology Drawing No. C04-065C Sheet 1 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。



VIEW C

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins		N		
Pitch		e		
Overall Height		A		
Molded Package Thickness		A2		
Standoff §		A1		
Overall Width		E		
Molded Package Width		E1		
Overall Length		D		
Chamfer (Optional)		h		
Foot Length		L		
Footprint		L1		
Lead Angle		Θ		
Foot Angle		φ		
Lead Thickness		c		
Lead Width		b		
Mold Draft Angle Top		α		
Mold Draft Angle Bottom		β		

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. § Significant Characteristic

3. 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.

4. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

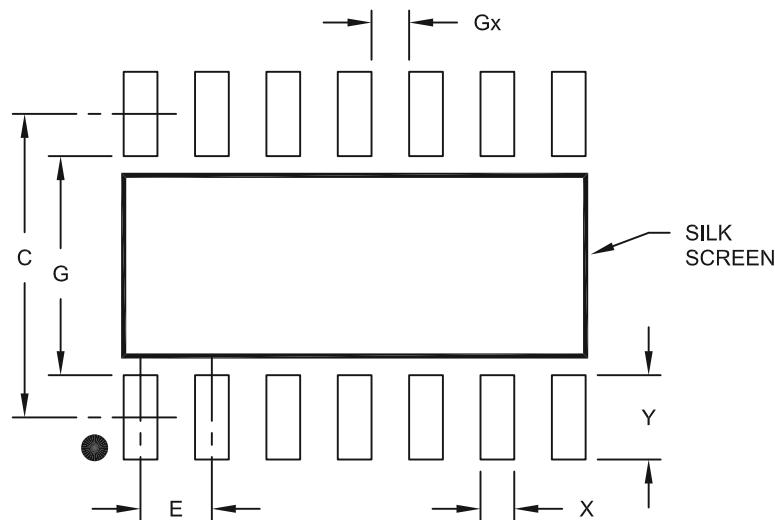
REF: Reference Dimension, usually without tolerance, for information purposes only.

5. Datums A & B to be determined at Datum H.

PIC16(L)F1764/5/8/9

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

Units		MILLIMETERS		
Dimension	Limits	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:

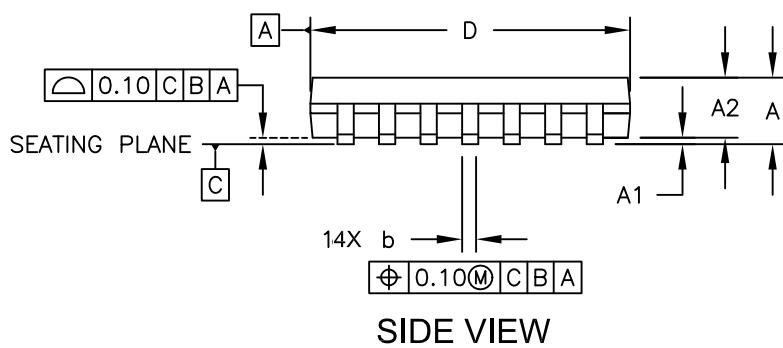
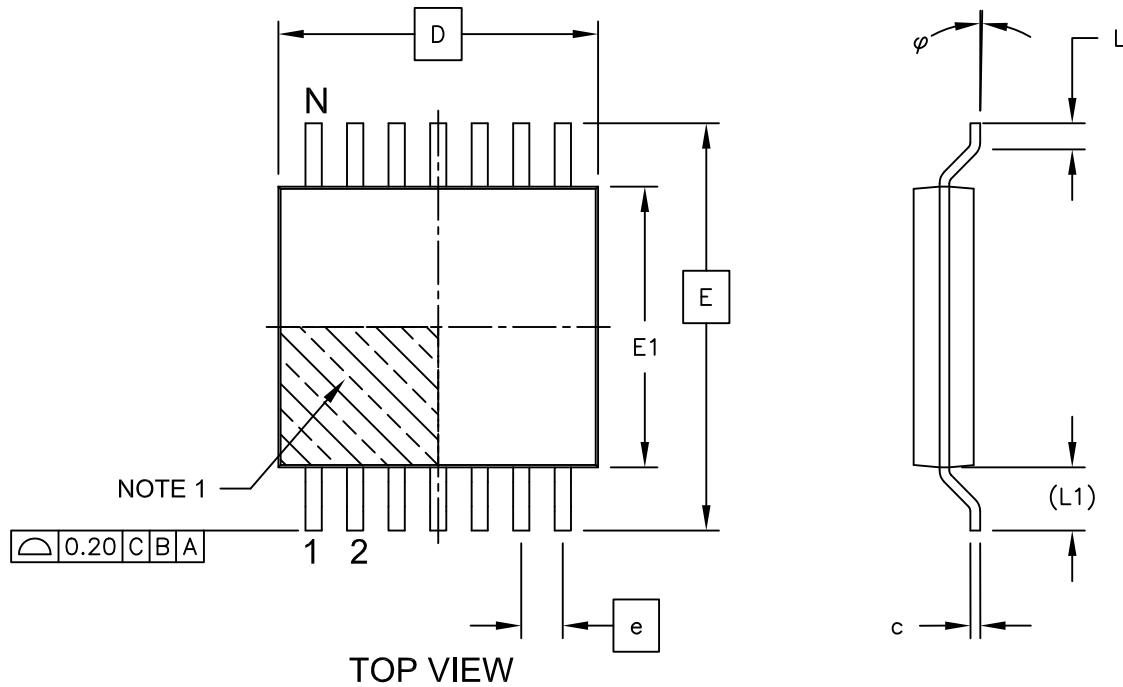
- Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

14 引脚塑封薄型缩小外形封装 (ST) —— 主体 4.4 mm [TSSOP]

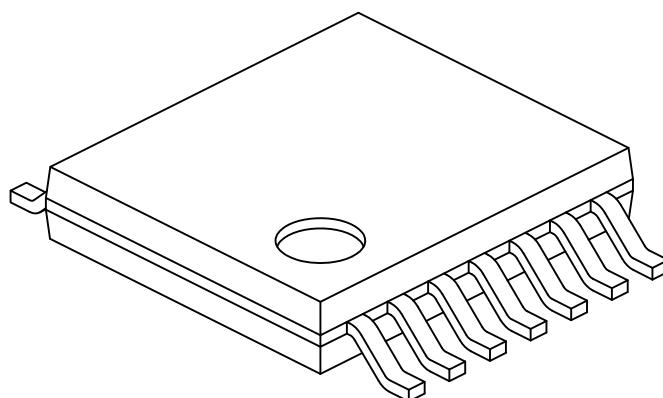
注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



PIC16(L)F1764/5/8/9

14 引脚塑封薄型缩小外形封装 (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

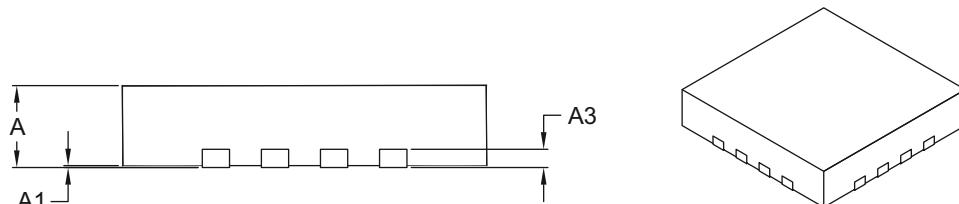
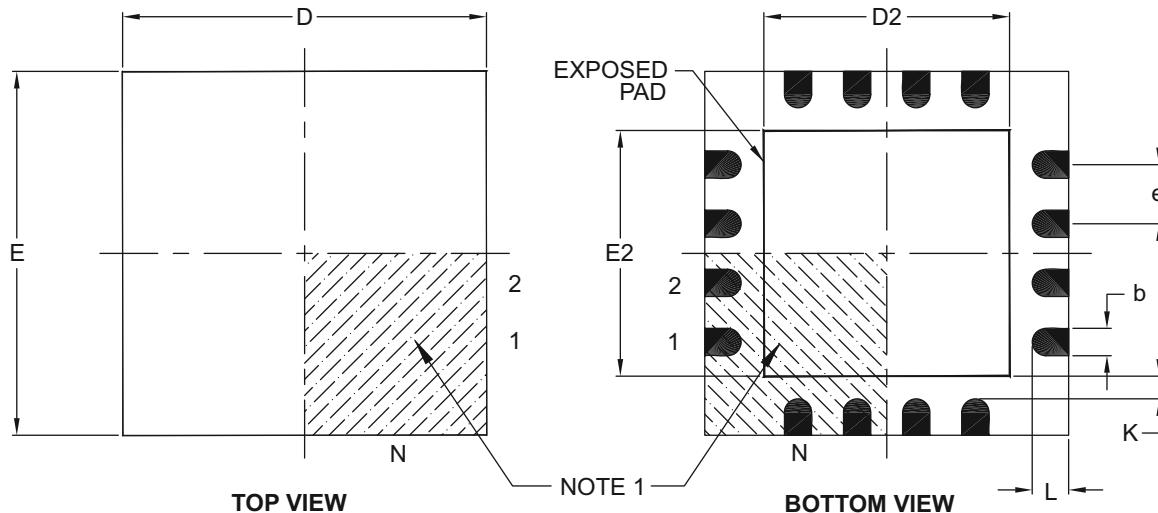
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

16 引脚塑封正方扁平无脚封装 (ML) —— 主体 4x4x0.9 mm [QFN]

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		16		
Pitch	e		0.65	BSC	
Overall Height	A	0.80	0.90	1.00	
Standoff	A1	0.00	0.02	0.05	
Contact Thickness	A3	0.20 REF			
Overall Width	E	4.00 BSC			
Exposed Pad Width	E2	2.50	2.65	2.80	
Overall Length	D	4.00 BSC			
Exposed Pad Length	D2	2.50	2.65	2.80	
Contact Width	b	0.25	0.30	0.35	
Contact Length	L	0.30	0.40	0.50	
Contact-to-Exposed Pad	K	0.20	-	-	

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

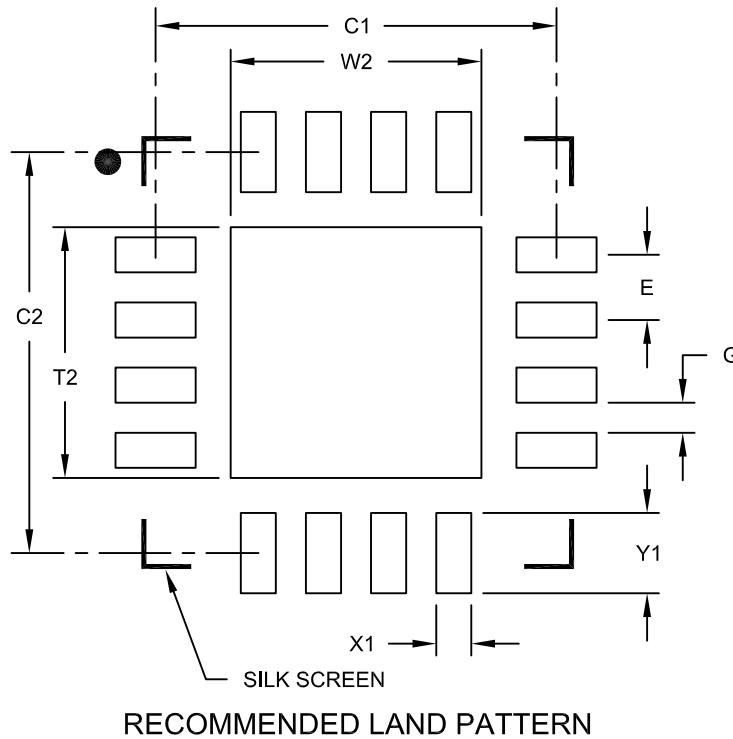
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127B

PIC16(L)F1764/5/8/9

16 引脚塑封正方扁平无脚封装 (ML) —— 主体 4x4x0.9 mm [QFN]

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
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			2.50
Optional Center Pad Length	T2			2.50
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
Distance Between Pads	G	0.30		

Notes:

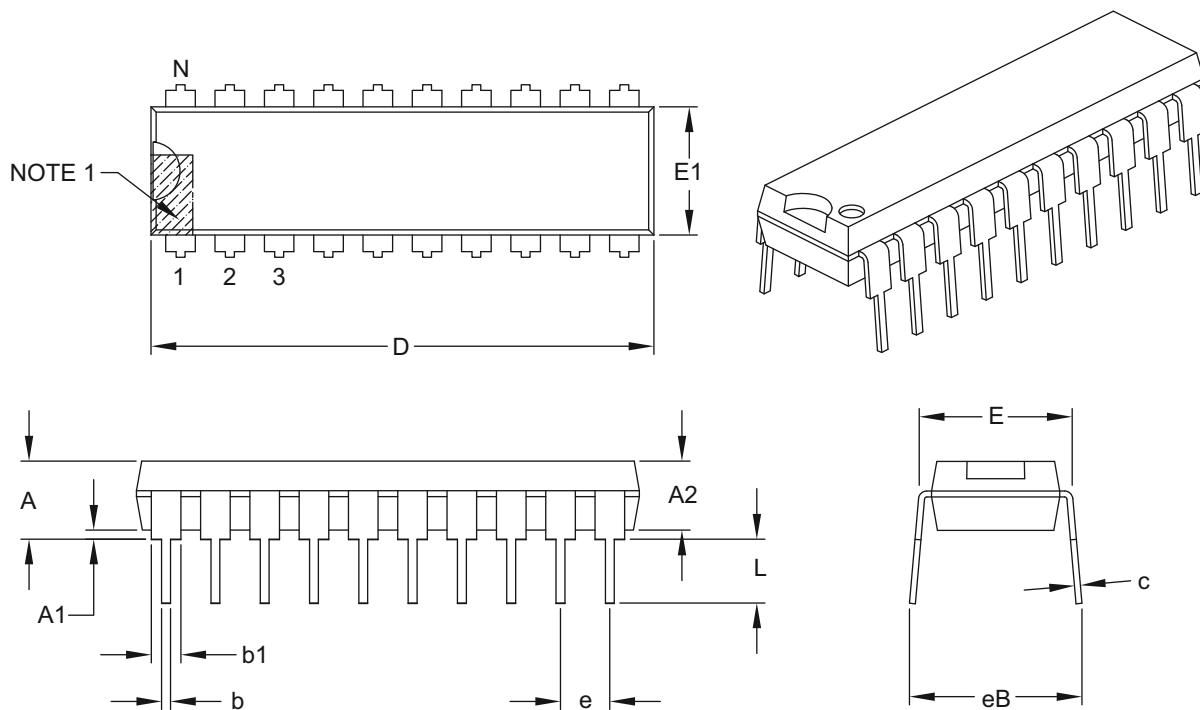
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2127A

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:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

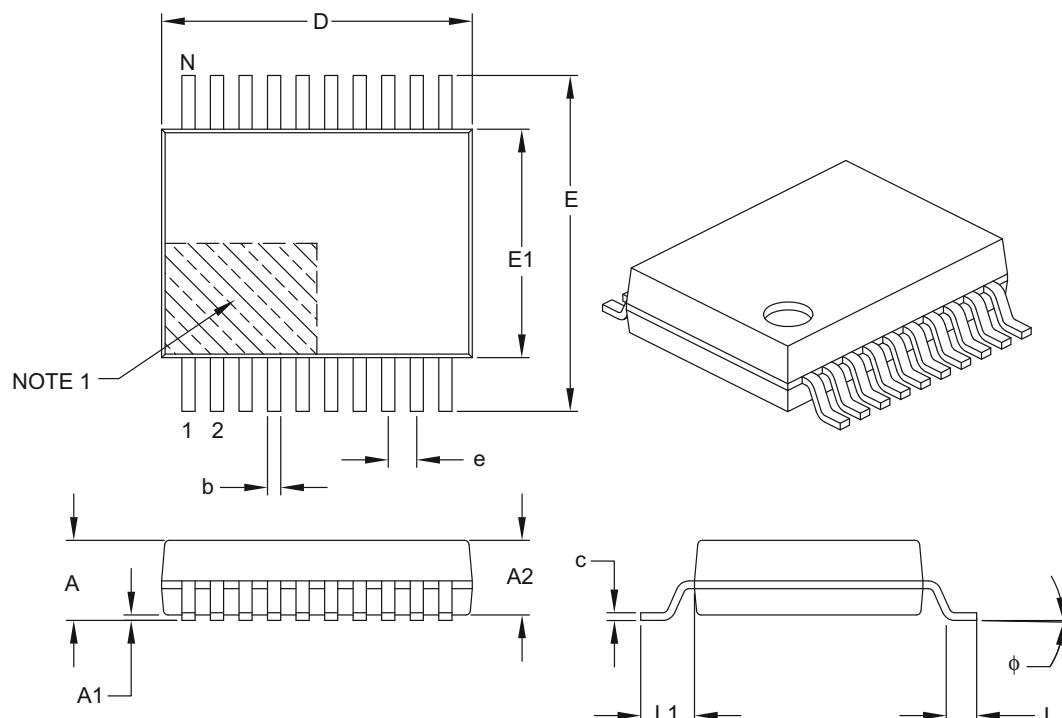
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B

PIC16(L)F1764/5/8/9

20 引脚塑封缩小外形封装 (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。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins		N		20
Pitch		e		0.65 BSC
Overall Height		A		—
Molded Package Thickness		A2		1.65
Standoff		A1		0.05
Overall Width		E		7.40
Molded Package Width		E1		5.00
Overall Length		D		6.90
Foot Length		L		0.55
Footprint		L1		1.25 REF
Lead Thickness		c		0.09
Foot Angle		φ		0°
Lead Width		b		0.22
				0.38

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 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

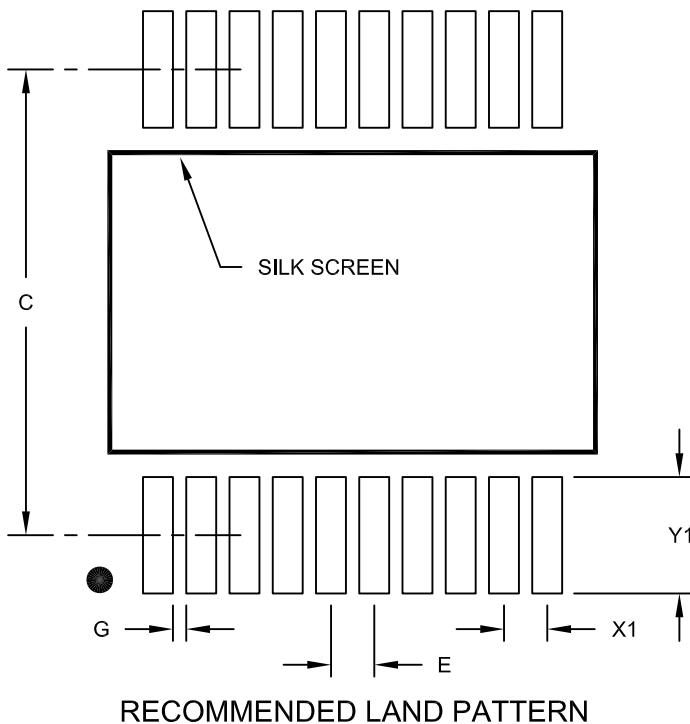
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

20 引脚塑封缩小外形封装 (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。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch		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:

- Dimensioning and tolerancing per ASME Y14.5M

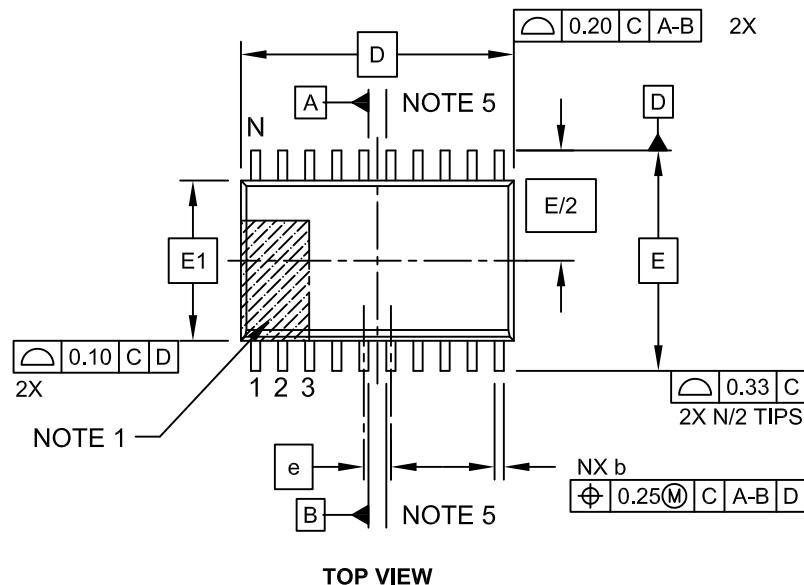
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

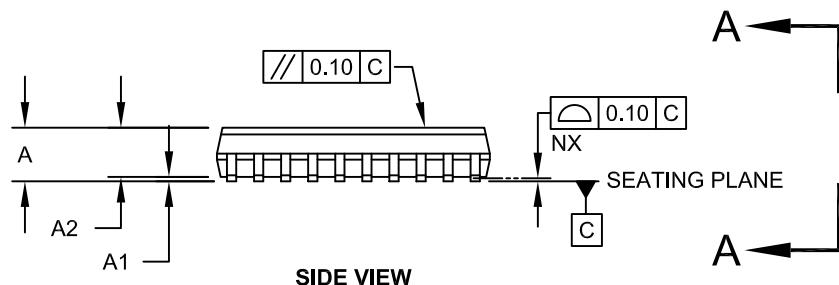
PIC16(L)F1764/5/8/9

20 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

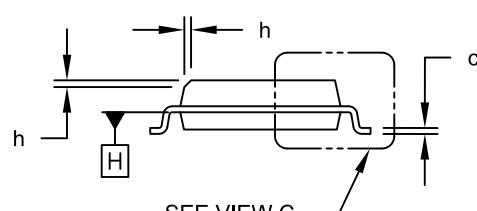
注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



TOP VIEW



SIDE VIEW

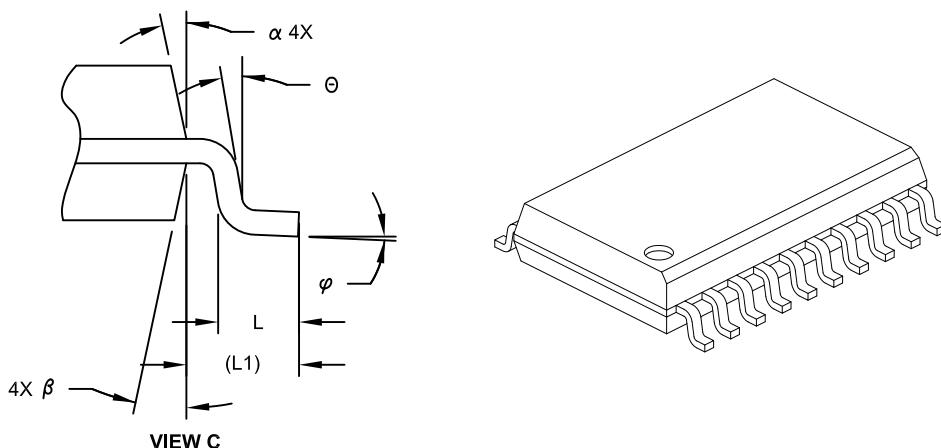


VIEW A-A

Microchip Technology Drawing C04-094C Sheet 1 of 2

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:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. § Significant Characteristic

3. 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.

4. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

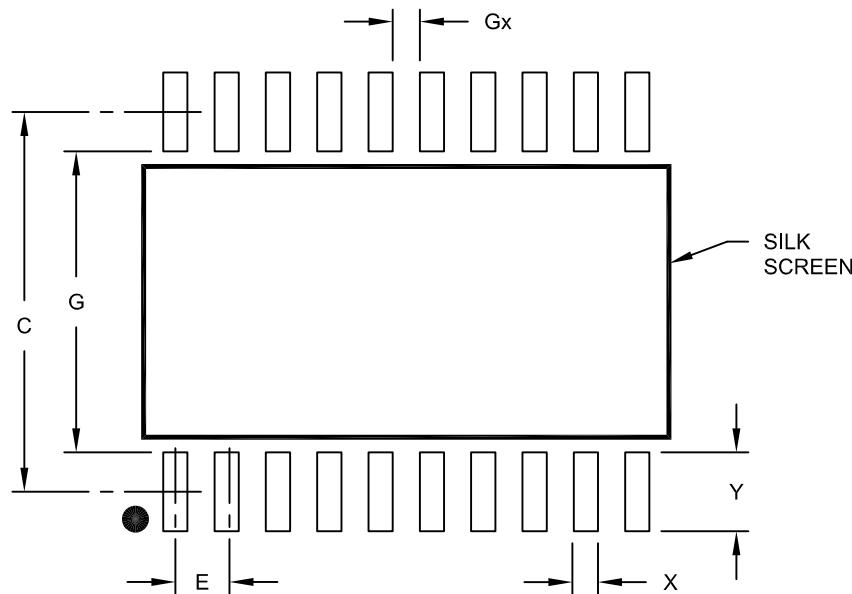
REF: Reference Dimension, usually without tolerance, for information purposes only.

5. Datums A & B to be determined at Datum H.

PIC16(L)F1764/5/8/9

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		
Dimension Limits		MIN	NOM	MAX
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:

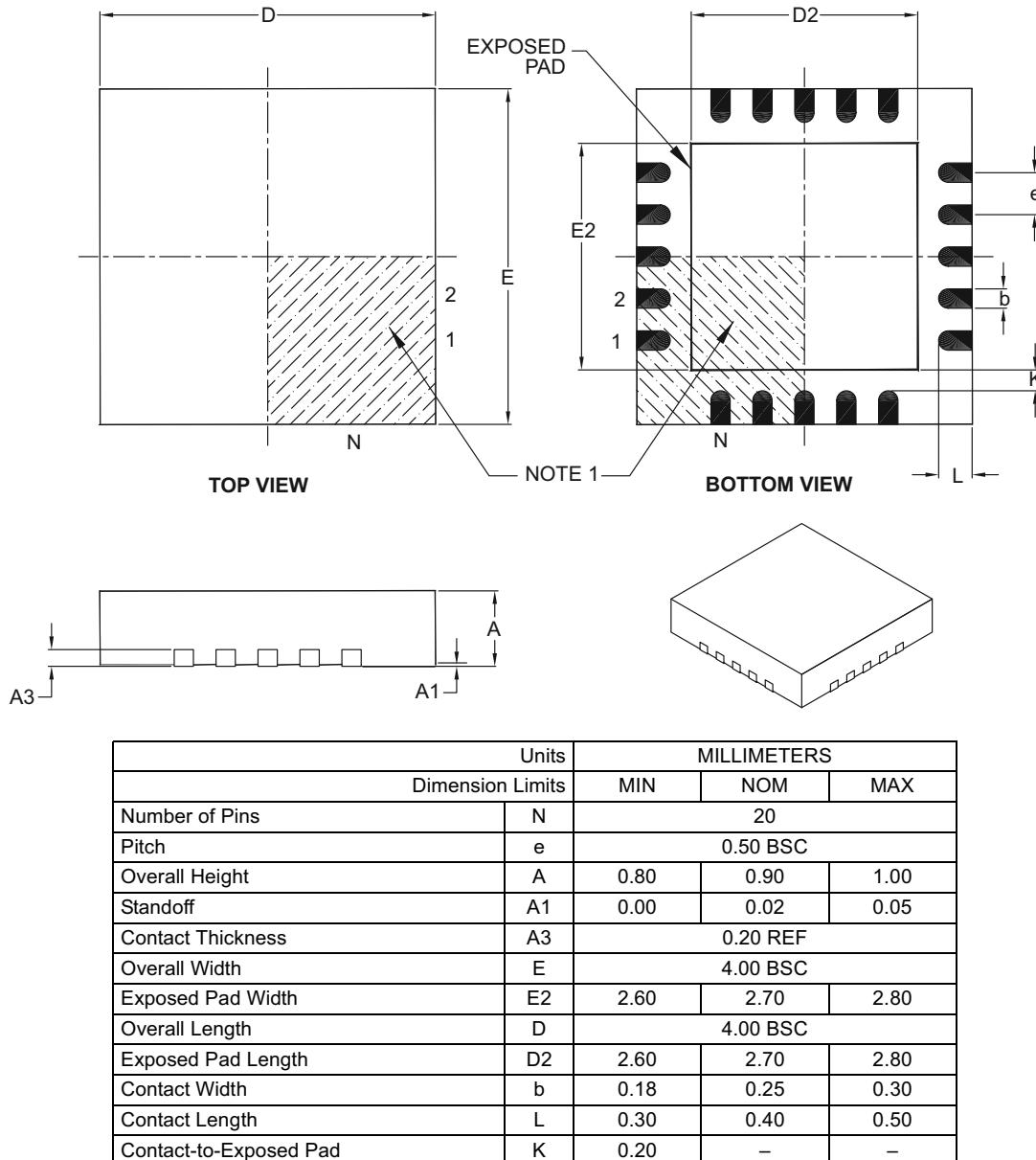
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

20 引脚塑封正方扁平无脚封装 (ML) —— 主体 4x4x0.9 mm [QFN]

注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



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.

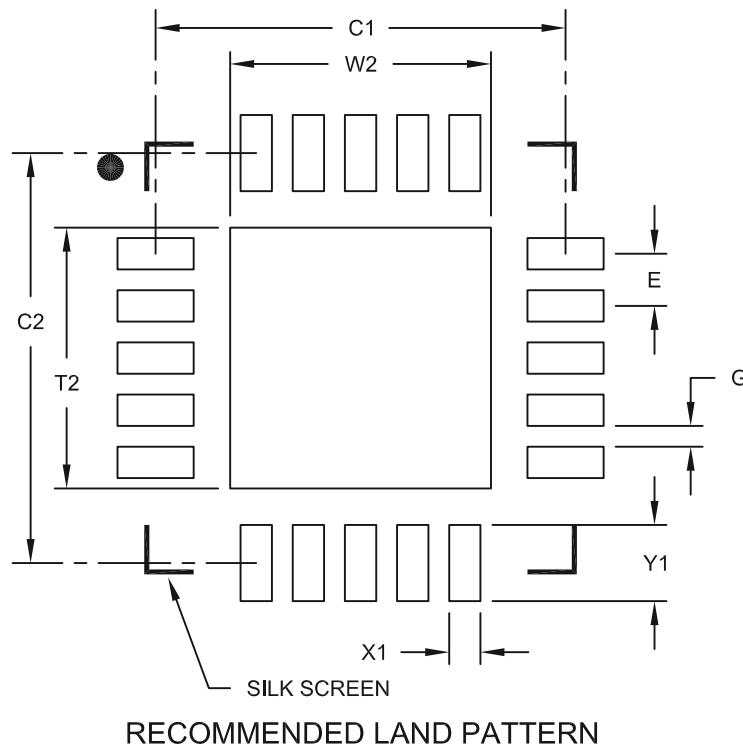
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

PIC16(L)F1764/5/8/9

20 引脚塑封正方扁平无脚封装 (ML) —— 主体 4x4 mm [QFN], 触点长度为 0.40 mm

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
Contact Pitch		E		0.50 BSC
Optional Center Pad Width	W2			2.50
Optional Center Pad Length	T2			2.50
Contact Pad Spacing	C1		3.93	
Contact Pad Spacing	C2		3.93	
Contact Pad Width	X1			0.30
Contact Pad Length	Y1			0.73
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. C04-2126A

附录 A: 版本历史

版本 A (2014 年 12 月)

本文档的初始版本。

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

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。

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。

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。

部件编号	[X] ⁽¹⁾	-	X	XX	XXX	示例:
器件	卷带式选项	温度范围	封装	定制信息		a) PIC16LF1764- I/P 工业级温度, PDIP 封装
器件:	PIC16F1764, PIC16LF1764, PIC16F1765, PIC16LF1765, PIC16F1768, PIC16LF1768, PIC16F1769, PIC16LF1769					b) PIC16F1769 - E/SS 扩展级温度, SSOP 封装
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⁾	ML = QFN P = PDIP SL = SOIC SO = SOIC SS = SSOP ST = TSSOP					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: 可能提供小型封装选项。关于小型封装的供货信息，请访问 www.microchip.com/packaging , 或联系您当地的销售办事处。
定制编号:	QTP、SQTP、代码或特殊要求（其他情况空白）					

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最安全的产品之一。
- 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、适销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，不得暗中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、设计和晶圆生产厂以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、非易失性存储器和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 —



MICROCHIP

全球销售及服务网点

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](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

克里夫兰 Cleveland

Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

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

洛杉矶 Los Angeles

Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

纽约 New York, NY

Tel: 1-631-435-6000

圣何塞 San Jose, CA

Tel: 1-408-735-9110

加拿大多伦多 Toronto

Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong

Tel: 852-2943-5100

Fax: 852-2401-3431

中国 - 北京

Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆

Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 东莞

Tel: 86-769-8702-9880

中国 - 杭州

Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 香港特别行政区

Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳

Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门

Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海

Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄

Tel: 886-7-213-7828

台湾地区 - 台北

Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹

Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore

Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune

Tel: 91-20-3019-1500

日本 Japan - Osaka

Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 Japan - Tokyo

Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韩国 Korea - Daegu

Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul

Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang

Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila

Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore

Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok

Tel: 66-2-694-1351
Fax: 66-2-694-1350

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

法国 France - Paris

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Dusseldorf

Tel: 49-2129-3766400
Tel: 49-721-625370

德国 Germany - Karlsruhe

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan

Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Venice

Tel: 39-049-7625286
Tel: 31-416-690399
Fax: 31-416-690340

波兰 Poland - Warsaw

Tel: 48-22-3325737

西班牙 Spain - Madrid

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Stockholm

Tel: 46-8-5090-4654

英国 UK - Wokingham

Tel: 44-118-921-5800
Fax: 44-118-921-5820