

---

## 14/20 引脚8位闪存单片机

---

### 说明

PIC16(L)F1615/9 单片机以 14/20 引脚封装提供了用于设计小型电机嵌入式控制和通用应用的独特片上功能。10 位 ADC、CCP、24 位 SMT 和过零检测等功能为各种应用提供了出色的解决方案。该产品系列还具有 CRC+ 存储器扫描功能和窗口化 WDT，支持家用电器、白色家电和其他终端设备中安全要求极高的系统。

### 内核特性

- 优化的 C 编译器 RISC 架构
- 仅需学习 49 条指令
- 工作速度：
  - DC – 32 MHz 时钟输入
  - 最小指令周期为 125 ns
- 中断功能
- 16 级深硬件堆栈
- 1 个 8 位定时器
- 4 个 16 位定时器
- 低电流上电复位 (Power-on Reset, POR)
- 可配置的上电延时定时器 (Power-up Timer, PWRT)
- 跳变点可选的欠压复位 (Brown-out Reset, BOR)
- 窗口看门狗定时器 (Windowed Watchdog Timer, WWDT)：
  - 可变预分频比选择
  - 可变窗口大小选择
  - 可用硬件或软件配置所有时钟源

### 存储器

- 8K 字的闪存程序存储器
- 1024 字节的数据 SRAM 存储器
- 直接、间接和相对寻址模式
- 高耐用性闪存 (High-Endurance Flash, HEF) 数据存储器：
  - 128B (非易失性)，100,000 次擦/写

### 工作特性

- 工作电压范围：
  - 1.8V 至 3.6V (PIC16LF1615/9)
  - 2.3V 至 5.5V (PIC16F1615/9)
- 温度范围：
  - 工业级: -40°C 至 85°C
  - 扩展级: -40°C 至 125°C

### 超低功耗 (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  $\mu$ A (32 kHz、1.8V 时, 典型值)
  - 32  $\mu$ A/MHz (1.8V 时, 典型值)

### 数字外设

- 可配置逻辑单元 (Configurable Logic Cell, CLC)：
  - 4 个 CLC
  - 集成组合和顺序逻辑
- 互补波形发生器 (Complementary Waveform Generator, CWG)：
  - 上升沿和下降沿死区控制
  - 全桥、半桥和单通道驱动
  - 多个信号源
- 两个捕捉/比较/PWM (Capture/Compare/PWM, CCP) 模块
- PWM: 两个 10 位脉宽调制器
- 两个信号测量定时器 (Signal Measurement Timer, SMT)：
  - 带有预分频器的 24 位定时器/计数器
  - 多个门控输入和时钟输入
- 角度定时器：
  - 单脉冲
  - 多个具有缺失脉冲恢复功能的脉冲
- 8 位定时器 (TMR2+HLT/4/6)：
  - 最多 3 个具有硬件限制定时器 (Hardware Limit Timer, HLT) 的 Timer2/4/6
  - 监视故障条件: 暂停和停止等
  - 多种模式
  - 带有预分频器的 8 位定时器/计数器
  - 8 位周期寄存器和后分频器
  - 异步硬件复位源
- 带比例-积分-微分 (Proportional-Integral-Derivative, PID) 功能的数学加速器：
  - 四种工作模式
  - 相加并相乘
  - 简单的乘法器
  - 相乘并累加
  - 可编程 PID 控制器
- 带有存储器扫描功能的循环冗余校验 (CRC/SCAN)：
  - 可通过软件配置
- 串行通信：
  - 增强型 USART (EUSART)
  - 兼容 SPI、I<sup>2</sup>C、RS-232、RS-485 和 LIN
  - 自动波特率检测, 检测到启动位时自动唤醒

- 最多 17 个 I/O 引脚和 1 个仅用作输入的引脚：
  - 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
  - 2 个高电流驱动引脚
- 外设引脚选择 (Peripheral Pin Select, PPS) :
  - 允许对数字 I/O 进行引脚映射

## 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
- 2 个比较器 (Comparator, COMP) :
  - 低功耗/高速模式
  - 最多 3 个外部反相输入
  - 同相输入端上的固定参考电压
  - 比较器输出可从外部访问
- 8 位数模转换器 (Digital-to-Analog Converter, DAC) :
  - 8 位分辨率, 轨到轨
  - 正参考电压选择
- 参考电压:
  - 固定参考电压 (Fixed Voltage Reference, FVR) : 1.024V、2.048V 和 4.096V 输出级别
- 过零检测 (Zero-Cross Detect, ZCD) :
  - 检测引脚上的交流信号何时越过地电位
- 2 个高电流驱动引脚:
  - 100 mA (5V 时)

## 时钟结构

- 16 MHz 内部振荡器:
  - 校准时精度为  $\pm 1\%$
  - 可选频率范围: 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
  - 1 种晶振模式, 频率最高为 32 MHz
- 故障保护时钟监视器:
  - 允许在外设时钟停止时安全关断
- 双速振荡器启动
- 振荡器起振定时器 (Oscillator Start-up Timer, OST)

表1: PIC12/16(L)F161X系列类型

器件	数据手册索引	闪存程序存储器 (字)	闪存程序存储器 (kB)	数据SRAM (字节)	I/O引脚	带 HLT 的 8 位定时器	16 位定时器	角度定时器	窗口看门狗定时器	24 位 SMT	比较器	10 位 ADC (通道)	过零检测	CCP/10 位 PWM	CWG	CLC	带存储器扫描的 CRC	带 PID 的数学加速器	100 mA 高电流 I/O	PPS	EUSART	I <sup>2</sup> C/SPI
PIC12(L)F1612	(A)	2048	3.5	256	6	4	1	0	有	1	1	4	1	2/0	1	0	有	0	0	无	0	0
PIC16(L)F1613	(A)	2048	3.5	256	12	4	1	0	有	2	2	8	1	2/0	1	0	有	0	0	无	0	0
PIC16(L)F1614	(B)	4096	7	512	12	4	3	1	有	2	2	8	1	2/2	1	2	有	1	2	有	1	1
PIC16(L)F1615	(C)	8192	14	1024	12	4	3	1	有	2	2	8	1	2/2	1	4	有	1	2	有	1	1
PIC16(L)F1618	(B)	4096	7	512	18	4	3	1	有	2	2	12	1	2/2	1	2	有	1	2	有	1	1
PIC16(L)F1619	(C)	8192	14	1024	18	4	3	1	有	2	2	12	1	2/2	1	4	有	1	2	有	1	1

注 1: 调试方法: (I) —— 集成在芯片上; (H) —— 通过 ICD 转接头; E —— 使用仿真产品

数据手册索引:

- A. DS40001737A\_CN PIC12(L)F1612/16(L)F1613 数据手册, 8/14 引脚 8 位闪存单片机
- B. DS40001769 PIC16(L)F1614/8 数据手册, 14/20 引脚 8 位闪存单片机
- C. DS40001770B\_CN PIC16(L)F1615/9 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> 或联系当地的销售办事处。

# PIC16(L)F1615/9

表2: 封装

封装	PDIP	SOIC	DFN	UDFN	TSSOP	QFN	UQFN	SSOP
PIC16(L)F1615	●	●			●	●		
PIC16(L)F1619	●	●				●	●	●

注： 引脚详细信息可能有所变动。

## 引脚图

14 引脚PDIP、SOIC和TSSOP

VDD 1 14 VSS  
RA5 2 13 RA0/ICSPDAT  
RA4 3 12 RA1/ICSPCLK  
MCLR/VPP/RA3 4 11 RA2  
RC5 5 10 RC0  
RC4 6 9 RC1  
RC3 7 8 RC2

16 引脚UQFN

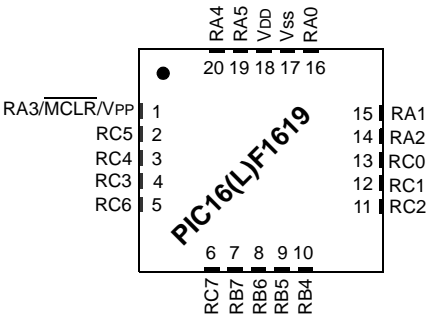
VDD 16 15 14 13 VSS  
RA5 1 12 RA0  
RA4 2 11 RA1  
RA3/MCLR/VPP 3 10 RA2  
RC5 4 9 RC0  
RC4 5 8 RC1  
RC3 6  
RC2 7  
RC1 8

20 引脚PDIP、SOIC和SSOP

VDD 1 20 VSS  
RA5 2 19 RA0  
RA4 3 18 RA1  
MCLR/VPP/RA3 4 17 RA2  
RC5 5 16 RC0  
RC4 6 15 RC1  
RC3 7 14 RC2  
RC6 8 13 RB4  
RC7 9 12 RB5  
RB7 10 11 RB6



20 引脚 QFN 和 UQFN



## 引脚分配表

表3: 14/16 引脚分配表 (PIC16(L)F1615)

I/O	14 引脚 PDIP/SOIC/TSSOP	16 引脚 UQFN	A/D	参考电压	比较器	定时器	CCP	CWG	ZCD	CLC	EUSART	SMT	角度定时器	MSSP	PWM	高电流 I/O	中断	上拉	基本功能
RA0	13	12	AN0	DAC1OUT1	C1IN+	—	—	—	—	—	—	—	—	—	—	—	IOC	有	ICSPDAT
RA1	12	11	AN1	VREF+	C1IN0- C2IN0-	—	—	—	—	—	—	—	—	—	—	—	IOC	有	ICSPCLK
RA2	11	10	AN2	—	—	T0CKI <sup>(1)</sup>	—	CWG1IN <sup>(1)</sup>	ZCD1IN	—	—	—	—	—	—	—	INT IOC	有	—
RA3	4	3	—	—	—	T6IN <sup>(1)</sup>	—	—	—	—	—	SMTWIN2 <sup>(1)</sup>	—	—	—	—	IOC	有	MCLR/VPP
RA4	3	2	AN3	—	—	T1G <sup>(1)</sup>	—	—	—	—	—	SMTSIG1 <sup>(1)</sup>	—	—	—	—	IOC	有	CLKOUT
RA5	2	1	—	—	—	T1CKI <sup>(1)</sup> T2IN <sup>(1)</sup>	—	—	—	—	—	SMTWIN1 <sup>(1)</sup>	—	—	—	—	IOC	有	CLKIN
RC0	10	9	AN4	—	C2IN+	T5CKI <sup>(1)</sup>	—	—	—	—	—	—	—	SCK <sup>(1,3)</sup>	—	—	IOC	有	—
RC1	9	8	AN5	—	C1IN1- C2IN1-	T4IN <sup>(1)</sup>	—	—	—	—	—	SMTSIG2 <sup>(1)</sup>	—	SDI <sup>(1)</sup>	—	—	IOC	有	—
RC2	8	7	AN6	—	C1IN2- C2IN2-	—	—	—	—	—	—	—	—	—	—	—	IOC	有	—
RC3	7	6	AN7	—	C1IN3- C2IN3-	T5G <sup>(1)</sup>	CCP2 <sup>(1)</sup>	—	—	CLCIN0 <sup>(1)</sup>	—	—	ATCC <sup>(1)</sup>	SS <sup>(1)</sup>	—	—	IOC	有	—
RC4	6	5	—	—	—	T3G <sup>(1)</sup>	—	—	—	CLCIN1 <sup>(1)</sup>	CK <sup>(1)</sup>	—	—	—	—	HIC4	IOC	有	—
RC5	5	4	—	—	—	T3CKI <sup>(1)</sup>	CCP1 <sup>(1)</sup>	—	—	—	RX <sup>(1,3)</sup>	—	ATIN <sup>(1)</sup>	—	—	HIC5	IOC	有	—
VDD	1	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	14	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VSS
OUT <sup>(2)</sup>	—	—	—	—	C1OUT	—	CCP1	CWG1A	ZCD1OUT	CLC1OUT	DT <sup>(3)</sup>	—	—	SDO	PWM3OUT	—	—	—	—
	—	—	—	—	C2OUT	—	CCP2	CWG1B	—	CLC2OUT	CK	—	—	SCK <sup>(3)</sup>	PWM4OUT	—	—	—	—
	—	—	—	—	—	—	—	CWG1C	—	—	TX	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	CWG1D	—	—	—	—	—	—	—	—	—	—	—

- 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)F1619)

I/O	20引脚PDIP、SOIC和SSOP	20引脚UQFN	A/D	参考电压	比较器	定时器	CCP	CWG	ZCD	CLC	EUSART	SMT	角度定时器	MSSP	PWM	高电流I/O	中断	上拉	基本功能
RA0	19	16	AN0	DAC1OUT	C1IN+	—	—	—	—	—	—	—	—	—	—	—	IOC	有	ICSPDAT
RA1	18	15	AN1	VREF+	C1IN0-C2IN0-	—	—	—	—	—	—	—	—	—	—	—	IOC	有	ICSPCLK
RA2	17	14	AN2	—	—	T0CKI <sup>(1)</sup>	—	CWG1IN <sup>(1)</sup>	ZCD1IN	—	—	—	—	—	—	—	INT IOC	有	—
RA3	4	1	—	—	—	T6IN <sup>(1)</sup>	—	—	—	—	—	SMTWIN2 <sup>(1)</sup>	—	—	—	—	IOC	有	MCLR V <sub>PP</sub>
RA4	3	20	AN3	—	—	T1G <sup>(1)</sup>	—	—	—	—	—	SMTSIG1 <sup>(1)</sup>	—	—	—	—	IOC	有	CLKOUT
RA5	2	19	—	—	—	T1CKI <sup>(1)</sup> T2IN <sup>(1)</sup>	—	—	—	CLCIN3 <sup>(1)</sup>	—	SMTWIN1 <sup>(1)</sup>	—	—	—	—	IOC	有	CLKIN
RB4	13	10	AN10	—	—	—	—	—	—	—	—	—	—	SDI <sup>(1)</sup>	—	—	IOC	有	—
RB5	12	9	AN11	—	—	—	—	—	—	—	RX <sup>(1,3)</sup>	—	—	—	—	—	IOC	有	—
RB6	11	8	—	—	—	—	—	—	—	—	—	—	—	SCK <sup>(1,3)</sup>	—	—	IOC	有	—
RB7	10	7	—	—	—	—	—	—	—	—	CK <sup>(1)</sup>	—	—	—	—	—	IOC	有	—
RC0	16	13	AN4	—	C2IN+	T5CKI <sup>(1)</sup>	—	—	—	—	—	—	—	—	—	—	IOC	有	—
RC1	15	12	AN5	—	C1IN1-C2IN1-	T4IN <sup>(1)</sup>	—	—	—	CLCIN2 <sup>(2)</sup>	—	SMTSIG2 <sup>(1)</sup>	—	—	—	—	IOC	有	—
RC2	14	11	AN6	—	C1IN2-C2IN2-	—	—	—	—	—	—	—	—	—	—	—	IOC	有	—
RC3	7	4	AN7	—	C1IN3-C2IN3-	T5G <sup>(1)</sup>	CCP2 <sup>(1)</sup>	—	—	CLCIN0 <sup>(1)</sup>	—	—	ATCC <sup>(1)</sup>	—	—	—	IOC	有	—
RC4	6	3	—	—	—	T3G <sup>(1)</sup>	—	—	—	CLCIN1 <sup>(1)</sup>	—	—	—	—	—	HIC4	IOC	有	—
RC5	5	2	—	—	—	T3CKI <sup>(1)</sup>	CCP1 <sup>(1)</sup>	—	—	—	—	—	ATIN <sup>(1)</sup>	—	—	HIC5	IOC	有	—
RC6	8	5	AN8	—	—	—	—	—	—	—	—	—	—	SS <sup>(1)</sup>	—	—	IOC	有	—
RC7	9	6	AN9	—	—	—	—	—	—	—	—	—	—	—	—	—	IOC	有	—
VDD	1	18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
VSS	20	17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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)F1619）（续）

I/O	20 引脚 PDIP、SOIC 和 SSOP	20 引脚 UQFN	A/D	参考电压	比较器	定时器	CCP	CWG	ZCD	CLC	EUSART	SMT	角度定时器	MSSP	PWM	高电流 I/O	中断	上拉	基本功能
OUT <sup>(2)</sup>	—	—	—	—	C1OUT	—	CCP1	CWG1A	ZCD1OUT	CLC1OUT	DT <sup>(3)</sup>	—	—	SDO	PWM3OUT	—	—	—	—
	—	—	—	—	C2OUT	—	CCP2	CWG1B	—	CLC2OUT	CK	—	—	SCK <sup>(3)</sup>	PWM4OUT	—	—	—	—
	—	—	—	—	—	—	—	CWG1C	—	CLC3OUT	TX	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	CWG1D	—	CLC4OUT	—	—	—	—	—	—	—	—	—

- 注 1：默认外设输入。通过 PPS 输入选择寄存器，可以将输入移至任何其他引脚。
- 2：所有引脚输出均默认设为 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。

## 目录

1.0	器件概述 .....	11
2.0	增强型中档CPU .....	20
3.0	存储器构成 .....	22
4.0	器件配置 .....	68
5.0	振荡器模块 .....	76
6.0	复位 .....	92
7.0	中断 .....	100
8.0	掉电模式（休眠） .....	117
9.0	窗口看门狗定时器（WDT） .....	120
10.0	闪存程序存储器控制 .....	128
11.0	循环冗余校验（CRC）模块 .....	144
12.0	I/O端口 .....	156
13.0	外设引脚选择（PPS）模块 .....	178
14.0	电平变化中断 .....	186
15.0	固定参考电压（FVR） .....	192
16.0	温度指示器模块 .....	195
17.0	模数转换器（ADC）模块 .....	197
18.0	8位数模转换器（DAC1）模块 .....	211
19.0	比较器模块 .....	215
20.0	过零检测（ZCD）模块 .....	223
21.0	Timer0 模块 .....	228
22.0	带门控控制的Timer1/3/5 模块 .....	231
23.0	Timer2/4/6 模块 .....	243
24.0	主同步串行端口（MSSP）模块 .....	267
25.0	增强型通用同步/异步收发器（EUSART） .....	320
26.0	捕捉/比较/PWM 模块 .....	352
27.0	脉宽调制（PWM）模块 .....	366
28.0	互补波形发生器（CWG）模块 .....	372
29.0	可配置逻辑单元（CLC） .....	398
30.0	信号测量定时器（SMT） .....	414
31.0	角度定时器（AT）模块 .....	460
32.0	带比例-积分-微分（PID）模块的数学加速器 .....	489
33.0	在线串行编程（ICSPTM） .....	505
34.0	指令集汇总 .....	507
35.0	电气规范 .....	521
36.0	直流和交流特性图表 .....	545
37.0	开发支持 .....	563
38.0	封装信息 .....	567
附录A:	版本历史 .....	590

## 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](mailto:CTRC@microchip.com)。我们期待您的反馈。

### 最新数据手册

欲获得本数据手册的最新版本，请访问我公司网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中紧跟数字串后的字母是版本号，例如：DS30000000A\_CN 是文档的 A 版本。

### 勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

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

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

### 客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 [www.microchip.com](http://www.microchip.com) 上注册。

## 1.0 器件概述

本数据手册介绍了PIC16(L)F1615/9。这些器件的框图如图1-1所示，可用的外设如表1-1所示，引脚说明如表1-2和表1-3所示。

表1-1： 器件外设汇总

外设		PIC16(L)F1615	PIC16(L)F1619
模数转换器（ADC）		•	•
互补波形发生器（CWG）		•	•
循环冗余校验（Cyclic Redundancy Check, CRC）		•	•
数模转换器（DAC）		•	•
增强型通用同步/异步收发器（Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter, EUSART）		•	•
固定参考电压（FVR）		•	•
温度指示器		•	•
窗口看门狗定时器（Watchdog Timer, WDT）		•	•
过零检测（Zero Cross Detection, ZCD）		•	•
捕捉/比较/PWM（CCP）模块			
	CCP1	•	•
	CCP2	•	•
比较器			
	C1	•	•
	C2	•	•
可配置逻辑单元（CLC）			
	CLC1	•	•
	CLC2	•	•
	CLC3	•	•
	CLC4	•	•
主同步串行端口			
	MSSP1	•	•
脉宽调制器（Pulse-Width Modulator, PWM）			
	PWM3	•	•
	PWM4	•	•
信号测量定时器（SMT）			
	SMT1	•	•
	SMT2	•	•
定时器			
	Timer0	•	•
	Timer1	•	•
	Timer2	•	•
	Timer3	•	•
	Timer4	•	•
	Timer5	•	•
	Timer6	•	•

## 1.1 寄存器和位的命名约定

### 1.1.1 寄存器名称

当器件中存在同一外设的多个实例时，外设控制寄存器将描述为外设标识符、外设实例和控制标识符的组合。控制寄存器部分将只显示所有寄存器名称的一个实例，用“x”代替外设实例号。当器件中只有该外设的一个实例时，该命名约定也适用于该外设，以保持与系列中包含多个实例的其他器件的兼容性。

### 1.1.2 位名称

有两种形式的位名称：

- 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

短位名称是位功能的缩写。例如，一些外设通过EN位来使能。寄存器中的位名称是短名称。

在C程序中访问位时，短位名称十分有用。用短名称访问位时的通用格式为 *RegisterNamebits.ShortName*。例如，COG1CON0寄存器中的使能位EN可以在C程序中使用指令 `COG1CON0bits.EN = 1` 来置1。

短名称在汇编程序中通常无用，因为不同外设可能会在不同位位置使用同一名称。出现这种情况时，在包含文件生成期间，相应短位名称的所有实例均附加下划线和寄存器名称，寄存器中的位保留以避免命名冲突。

#### 1.1.2.2 长位名称

长位名称的构建方式是向短名称添加外设缩写前缀。该前缀对外设而言是惟一的，因此可使每个长位名称也惟一。COG1使能位的长位名称为COG1前缀G1附加使能位短名称EN，最终得到惟一的位名称G1EN。

长位名称在C程序和汇编程序中均有用。例如，在C程序中，COG1CON0使能位可以用 `G1EN = 1` 指令置1。在汇编程序中，该位可以用 `BSF COG1CON0,G1EN` 指令置1。

### 1.1.2.3 位域

位域是同一寄存器中的两个或多个相邻位。位域仅遵循短位命名约定。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，没有对应的长位名称。位域访问只能在C程序中进行。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;
```

位域中的各个位也可用长位名称和短位名称访问。每个位均为位域名称附加位域内相应位位置的编号。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：

示例1：

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

示例2：

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

## 1.1.3 寄存器和位命名约定的例外情况

### 1.1.3.1 状态位、中断位和镜像位

状态位、中断使能位、中断标志位和镜像位位于多个外设的寄存器中。在这些情况下，位名称是惟一的，因此没有前缀或短名称。

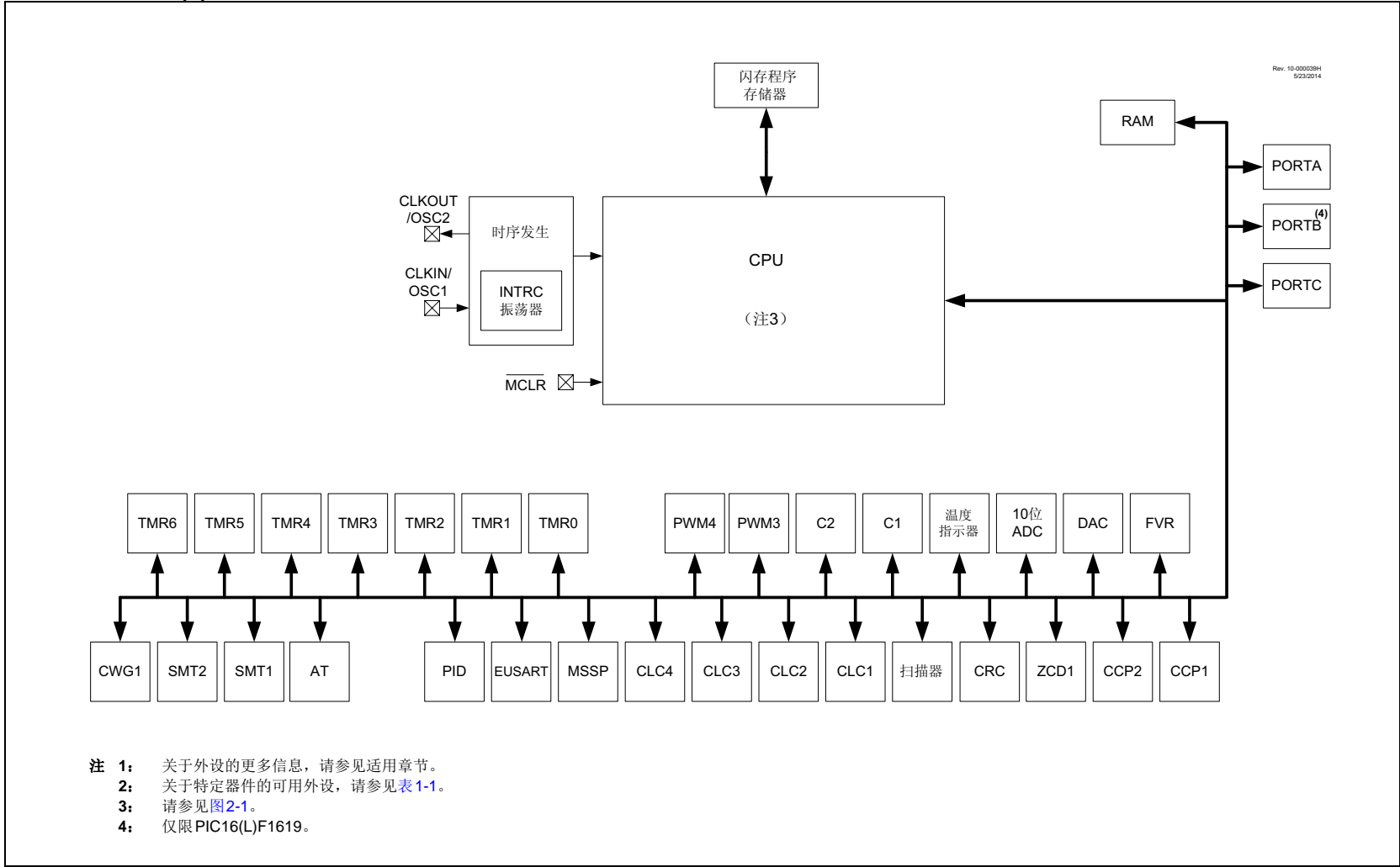
### 1.1.3.2 传统外设

有一些外设并未严格遵循这些命名约定。例外情况是存在多年且出现在几乎每个器件中的外设。这些例外有助于限制新约定对传统代码的不利影响。在寄存器部分会列出一个遵循新约定的外设的表，包含每个外设实例的长名称前缀。归入例外情况的外设没有此表。这些外设包括但不限于：

- EUSART
- MSSP



图 1-1: PIC16(L)F1615/9框图



# PIC16(L)F1615/9

表1-2: PIC16(L)F1615 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/DAC1OUT1/ ICSPDAT	RA0	TTL/ST	CMOS/OD	通用 I/O。
	AN0	AN	—	ADC 通道输入。
	C1IN+	AN	—	比较器的同相输入。
	DAC1OUT1	—	AN	数模转换器输出。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RA1/AN1/VREF+/C1IN0-/C2IN0-/ ICSPCLK	RA1	TTL/ST	CMOS/OD	通用 I/O。
	AN1	AN	—	ADC 通道输入。
	VREF+	AN	—	参考电压输入。
	C1IN0-	AN	—	比较器的反相输入。
	C2IN0-	AN	CMOS/OD	比较器的反相输入。
	ICSPCLK	ST	—	ICSP 编程时钟。
RA2/AN2/T0CKI <sup>(1)</sup> /CWG1IN <sup>(1)</sup> / ZCD1IN/INT	RA2	TTL/ST	CMOS/OD	通用 I/O。
	AN2	AN	—	ADC 通道输入。
	T0CKI	TTL/ST	—	Timer0 时钟输入。
	CWG1IN	TTL/ST	—	CWG 互补输入。
	ZCD1IN	AN	—	过零检测输入。
	INT	TTL/ST	—	外部中断。
RA3/VPP/T6IN <sup>(1)</sup> /SMTWIN2 <sup>(1)</sup> / MCLR	RA3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	VPP	HV	—	编程电压。
	T6IN	TTL/ST	—	Timer6 输入。
	SMTWIN2	TTL/ST	—	SMT2 窗口输入。
	MCLR	TTL/ST	—	带内部上拉的主复位。
RA4/AN3/T1G <sup>(1)</sup> /SMTSIG1 <sup>(1)</sup> / CLKOUT	RA4	TTL/ST	CMOS/OD	通用 I/O。
	AN3	AN	—	ADC 通道输入。
	T1G	TTL/ST	—	Timer1 门控输入。
	SMTSIG1	TTL/ST	—	SMT1 信号输入。
	CLKOUT	—	CMOS	Fosc/4 输出。
RA5/CLKIN/T1CKI <sup>(1)</sup> /T2IN <sup>(1)</sup> / SMTWIN1 <sup>(1)</sup>	RA5	TTL/ST	CMOS/OD	通用 I/O。
	CLKIN	CMOS	—	外部时钟输入（EC 模式）。
	T1CKI	TTL/ST	—	Timer1 时钟输入。
	T2IN	TTL/ST	—	Timer2 输入。
	SMTWIN1	TTL/ST	—	SMT1 窗口输入。
RC0/AN4/C2IN+/T5CKI <sup>(1)</sup> / SCK <sup>(1)</sup>	RC0	TTL/ST	CMOS/OD	通用 I/O。
	AN4	AN	—	ADC 通道输入。
	C2IN+	AN	—	比较器的同相输入。
	T5CKI	TTL/ST	—	Timer5 时钟输入。
	SCK	ST	CMOS	SPI 时钟。

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 = 带 CMOS 电平的施密特触发器输入      I<sup>2</sup>C = 带 I<sup>2</sup>C 电平的施密特触发器输入  
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，可以将输入移至任何其他引脚。

2: 所有引脚输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见寄存器 13-1。

表1-2: PIC16(L)F1615引脚说明（续）

名称	功能	输入类型	输出类型	说明
RC1/AN5/C1IN1-/C2IN1-/T4IN <sup>(1)</sup> /SMTSIG2 <sup>(1)</sup> /SDI <sup>(1)</sup>	RC1	TTL/ST	CMOS/OD	通用 I/O。
	AN5	AN	—	ADC 通道输入。
	C1IN1-	AN	—	比较器的反相输入。
	C2IN1-	AN	—	比较器的反相输入。
	T4IN	TTL/ST	—	Timer4 输入。
	SMTSIG2	TTL/ST	—	SMT2 信号输入。
	SDI	CMOS	—	SPI 数据输入。
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	通用 I/O。
	AN6	AN	—	ADC 通道输入。
	C1IN2-	AN	—	比较器的反相输入。
	C2IN2-	AN	—	比较器的反相输入。
RC3/AN7/C1IN3-/C2IN3-/T5G <sup>(1)</sup> /CCP2 <sup>(1)</sup> /CLCIN0 <sup>(1)</sup> /ATCC <sup>(1)</sup> /SS	RC3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	AN7	AN	—	ADC 通道输入。
	C1IN3-	AN	—	比较器的反相输入。
	C2IN3-	AN	—	比较器的反相输入。
	T5G	ST	—	Timer5 门控输入。
	CCP2	TTL/ST	CMOS/OD	捕捉/比较/PWM2。
	CLCIN0	ST	—	可配置逻辑单元源输入。
	ATCC	ST	—	角度定时器捕捉/比较输入。
	SS	ST	—	从选择输入。
RC4/T3G <sup>(1)</sup> /CLCIN1 <sup>(1)</sup> /CK <sup>(1)</sup> /HIC4	RC4	TTL/ST	CMOS/OD	通用 I/O。
	T3G	TTL/ST	—	Timer3 门控输入。
	CLCIN1	ST	—	可配置逻辑单元源输入。
	CK	ST	CMOS	EUSART 同步时钟。
	HIC4	TTL	CMOS	高电流 I/O。
RC5/T3CKI <sup>(1)</sup> /CCP1 <sup>(1)</sup> /RX <sup>(1)</sup> /ATIN <sup>(1)</sup> /HIC5	RC5	TTL/ST	CMOS/OD	通用 I/O。
	T3CKI	TTL/ST	—	Timer3 时钟输入。
	CCP1	TTL/ST	CMOS/OD	捕捉/比较/PWM1。
	RX	ST	—	EUSART 异步输入。
	ATIN	TTL/ST	—	角度定时器时钟输入。
	HIC5	TTL	CMOS	高电流 I/O。

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 = 带 CMOS 电平的施密特触发器输入      I<sup>2</sup>C = 带 I<sup>2</sup>C 电平的施密特触发器输入  
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，可以将输入移至任何其他引脚。

2: 所有引脚输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见寄存器 13-1。

# PIC16(L)F1615/9

表1-2: PIC16(L)F1615引脚说明（续）

名称	功能	输入类型	输出类型	说明
OUT <sup>(2)</sup>	C1OUT	—	CMOS	比较器输出。
	C2OUT	—	CMOS	比较器输出。
	CCP1	—	CMOS	捕捉/比较/PWM1输出。
	CCP2	—	CMOS	捕捉/比较/PWM2输出。
	PWM3OUT	—	CMOS	PWM3输出。
	PWM4OUT	—	CMOS	PWM4输出。
	CWG1A	—	CMOS	互补输出发生器输出A。
	CWG1B	—	CMOS	互补输出发生器输出B。
	CWG1C	—	CMOS	互补输出发生器输出C。
	CWG1D	—	CMOS	互补输出发生器输出D。
	SDO	—	CMOS	SPI数据输出。
	SCK	—	CMOS	SPI时钟输出。
	TX/CK	—	CMOS	EUSART异步发送数据/同步时钟输出。
	DT	—	CMOS	EUSART同步数据输出E。
	CLC1OUT	—	CMOS	可配置逻辑单元1源输出。
	CLC2OUT	—	CMOS	可配置逻辑单元2源输出。
	ZCD1OUT	—	CMOS	过零检测输出。
VDD	VDD	电源	—	正电源。
VSS	VSS	电源	—	参考地。

**图注:** AN = 模拟输入或输出      CMOS = CMOS兼容输入或输出      OD = 漏极开路  
TTL = TTL兼容输入      ST = 带CMOS电平的施密特触发器输入      I<sup>2</sup>C = 带I<sup>2</sup>C电平的施密特触发器输入  
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，可以将输入移至任何其他引脚。

**2:** 所有引脚输出均默认设为PORT锁存器数据。通过PPS输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见[寄存器13-1](#)。

**表 1-3: PIC16(L)F1619 引脚说明**

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/DAC1OUT/ICSPDAT	RA0	TTL/ST	CMOS/OD	通用 I/O。
	AN0	AN	—	ADC 通道输入。
	C1IN+	AN	—	比较器的同相输入。
	DAC1OUT	—	AN	数模转换器输出。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RA1/AN1/VREF+/C1IN0-/C2IN0-/ICSPCLK	RA1	TTL/ST	CMOS/OD	通用 I/O。
	AN1	AN	—	ADC 通道输入。
	VREF+	AN	—	参考电压输入。
	C1IN0-	AN	—	比较器的反相输入。
	C2IN0-	AN	CMOS/OD	比较器的反相输入。
	ICSPCLK	ST	—	ICSP 编程时钟。
RA2/AN2/T0CKI <sup>(1)</sup> /CWG1IN <sup>(1)</sup> /ZCD1IN/INT	RA2	TTL/ST	CMOS/OD	通用 I/O。
	AN2	AN	—	ADC 通道输入。
	T0CKI	TTL/ST	—	Timer0 时钟输入。
	CWG1IN	TTL/ST	—	CWG 互补输入。
	ZCD1IN	AN	—	过零检测输入。
	INT	TTL/ST	—	外部中断。
RA3/VPP/T6IN <sup>(1)</sup> /SMTWIN2 <sup>(1)</sup> /MCLR	RA3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	VPP	HV	—	编程电压。
	T6IN	TTL/ST	—	Timer6 输入。
	SMTWIN2	TTL/ST	—	SMT2 窗口输入。
	MCLR	TTL/ST	—	带内部上拉的主复位。
RA4/AN3/T1G <sup>(1)</sup> /SMTSIG1 <sup>(1)</sup> /CLKOUT	RA4	TTL/ST	CMOS/OD	通用 I/O。
	AN3	AN	—	ADC 通道输入。
	T1G	TTL/ST	—	Timer1 门控输入。
	SMTSIG1	TTL/ST	—	SMT1 信号输入。
	CLKOUT	—	CMOS	Fosc/4 输出。
RA5/CLKIN/T1CKI <sup>(1)</sup> /T2IN <sup>(1)</sup> /CLCIN3 <sup>(1)</sup> /SMTWIN1	RA5	TTL/ST	CMOS/OD	通用 I/O。
	CLKIN	CMOS	—	外部时钟输入（EC 模式）。
	T1CKI	TTL/ST	—	Timer1 时钟输入。
	T2IN	TTL/ST	—	Timer2 输入。
	CLCIN3	ST	—	可配置逻辑单元源输入。
	SMTWIN1	TTL/ST	—	SMT1 窗口输入。
RB4/AN10/SDI <sup>(1)</sup>	RB4	TTL/ST	CMOS/OD	通用 I/O。
	AN10	AN	—	ADC 通道输入。
	SDI	CMOS	—	SPI 数据输入。
RB5/AN11/RX <sup>(1, 3)</sup>	RB5	TTL/ST	CMOS/OD	通用 I/O。
	AN11	AN	—	ADC 通道输入。
	RX	ST	—	EUSART 异步输入。

**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 = 带 CMOS 电平的施密特触发器输入      I<sup>2</sup>C = 带 I<sup>2</sup>C 电平的施密特触发器输入  
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，可以将输入移至任何其他引脚。

**2:** 所有引脚输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见 [寄存器 13-1](#)。

**3:** 这些 I<sup>2</sup>C 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)F1615/9

表1-3: PIC16(L)F1619引脚说明（续）

名称	功能	输入类型	输出类型	说明
RB6/SCK <sup>(1, 3)</sup>	RB6	TTL/ST	CMOS/OD	通用 I/O。
	SCK	ST	CMOS	SPI 时钟。
RB7/CK <sup>(1)</sup>	RB7	TTL/ST	CMOS/OD	通用 I/O。
	CK	ST	CMOS	EUSART 同步时钟。
RC0/AN4/C2IN+/T5CKI <sup>(1)</sup>	RC0	TTL/ST	CMOS/OD	通用 I/O。
	AN4	AN	—	ADC 通道输入。
	C2IN+	AN	—	比较器的同相输入。
	T5CKI	TTL/ST	—	Timer5 时钟输入。
RC1/AN5/C1IN1-/C2IN1-/T4IN <sup>(1)</sup> /CLCIN2 <sup>(2)</sup> /SMTSIG2 <sup>(1)</sup>	RC1	TTL/ST	CMOS/OD	通用 I/O。
	AN5	AN	—	ADC 通道输入。
	C1IN1-	AN	—	比较器的反相输入。
	C2IN1-	AN	—	比较器的反相输入。
	T4IN	TTL/ST	—	Timer4 输入。
	CLCIN2	ST	—	可配置逻辑单元源输入。
	SMTSIG2	TTL/ST	—	SMT2 信号输入。
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	通用 I/O。
	AN6	AN	—	ADC 通道输入。
	C1IN2-	AN	—	比较器的反相输入。
	C2IN2-	AN	—	比较器的反相输入。
RC3/AN7/C1IN3-/C2IN3-/T5G <sup>(1)</sup> /CCP2 <sup>(1)</sup> /CLCIN0 <sup>(1)</sup> /ATCC <sup>(1)</sup>	RC3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	AN7	AN	—	ADC 通道输入。
	C1IN3-	AN	—	比较器的反相输入。
	C2IN3-	AN	—	比较器的反相输入。
	T5G	ST	—	Timer5 门控输入。
	CCP2	ST	CMOS	捕捉/比较/PWM2。
	CLCIN0	ST	—	可配置逻辑单元源输入。
	ATCC	ST	—	角度定时器捕捉/比较输入。
RC4/T3G <sup>(1)</sup> /CLCIN1 <sup>(1)</sup> /HIC4	RC4	TTL/ST	CMOS/OD	通用 I/O。
	T3G	ST	—	Timer3 门控输入。
	CLCIN1	ST	—	可配置逻辑单元源输入。
	HIC4	TTL	CMOS	高电流 I/O。
RC5/T3CKI <sup>(1)</sup> /CCP1 <sup>(1)</sup> /ATIN <sup>(1)</sup> /HIC5	RC5	TTL/ST	CMOS/OD	通用 I/O。
	T3CKI	TTL/ST	—	Timer3 时钟输入。
	CCP1	TTL/ST	CMOS/OD	捕捉/比较/PWM1。
	ATIN	TTL/ST	—	角度定时器时钟输入。
	HIC5	TTL	CMOS	高电流 I/O。

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 = 带 CMOS 电平的施密特触发器输入      I<sup>2</sup>C = 带 I<sup>2</sup>C 电平的施密特触发器输入  
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，可以将输入移至任何其他引脚。  
2: 所有引脚输出均默认设为 PORT 锁存器数据。通过 PPS 输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见寄存器 13-1。  
3: 这些 I<sup>2</sup>C 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)F1619引脚说明（续）

名称	功能	输入类型	输出类型	说明
OUT <sup>(2)</sup>	C1OUT	—	CMOS	比较器输出。
	C2OUT	—	CMOS	比较器输出。
	CCP1	—	CMOS	捕捉/比较/PWM1输出。
	CCP2	—	CMOS	捕捉/比较/PWM2输出。
	PWM3OUT	—	CMOS	PWM3输出。
	PWM4OUT	—	CMOS	PWM4输出。
	CWG1A	—	CMOS	互补输出发生器输出A。
	CWG1B	—	CMOS	互补输出发生器输出B。
	CWG1C	—	CMOS	互补输出发生器输出C。
	CWG1D	—	CMOS	互补输出发生器输出D。
	SDO	—	CMOS	SPI数据输出。
	SCK	—	CMOS	SPI时钟输出。
	TX/CK	—	CMOS	EUSART异步发送数据/同步时钟输出。
	DT	—	CMOS	EUSART同步数据输出。
	CLC1OUT	—	CMOS	可配置逻辑单元1源输出。
	CLC2OUT	—	CMOS	可配置逻辑单元2源输出。
	CLC3OUT	—	CMOS	可配置逻辑单元3源输出。
	CLC4OUT	—	CMOS	可配置逻辑单元4源输出。
	ZCD1OUT	—	CMOS	过零检测输出。
VDD	VDD	电源	—	正电源。
VSS	VSS	电源	—	参考地。

图注: AN = 模拟输入或输出      CMOS = CMOS兼容输入或输出      OD = 漏极开路  
TTL = TTL兼容输入      ST = 带CMOS电平的施密特触发器输入      I<sup>2</sup>C = 带I<sup>2</sup>C电平的施密特触发器输入  
HV = 高电压      XTAL = 晶振

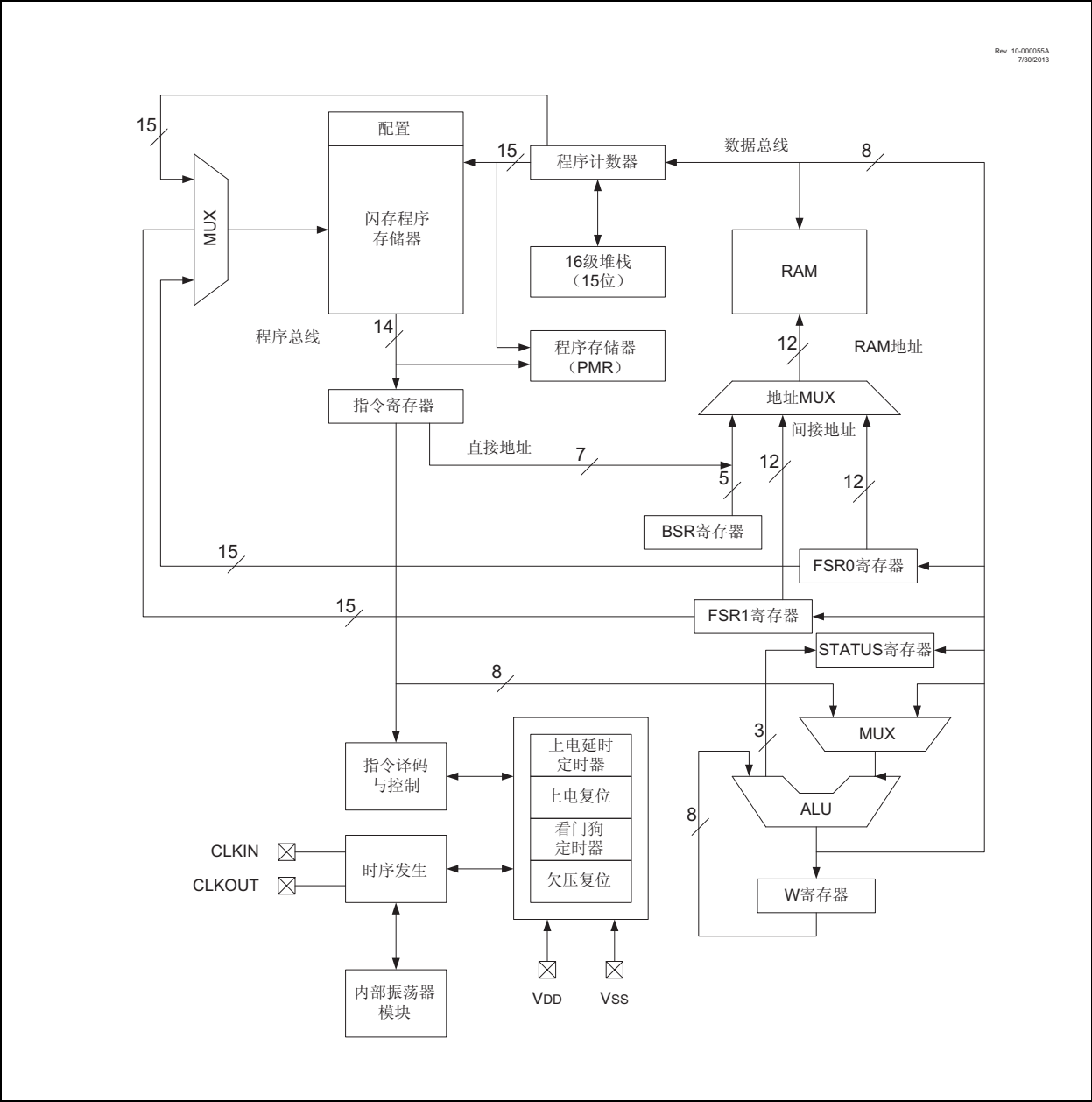
- 注 1: 默认外设输入。通过PPS输入选择寄存器, 可以将输入移至任何其他引脚。  
2: 所有引脚输出均默认设为PORT锁存器数据。通过PPS输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见[寄存器13-1](#)。  
3: 这些I<sup>2</sup>C功能是双向的。输出引脚选择必须与输入引脚选择相同。

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。器件提供了直接寻址、间接寻址和相对寻址模式。用户可以通过两个文件选择寄存器（File Select Register，FSR）来读取程序和数据存储器。

- 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至影子寄存器中，从中断返回时则会恢复这些寄存器。这可以节省堆栈空间 and 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，如果使能复位，则会导致软件复位。更多详细信息，请参见[第3.5节“堆栈”](#)。

## 2.3 文件选择寄存器

有两个16位文件选择寄存器（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3.6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34.0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

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

3.1 程序存储器构成

增强型中档内核具有一个15位程序计数器，能够寻址32K x 14的程序存储空间。表3-1列出了已实现的存储器大小。访问超出上述边界的存储单元，将返回到已实现的存储空间内。复位向量位于0000h，而中断向量位于0004h（见图3-1）。

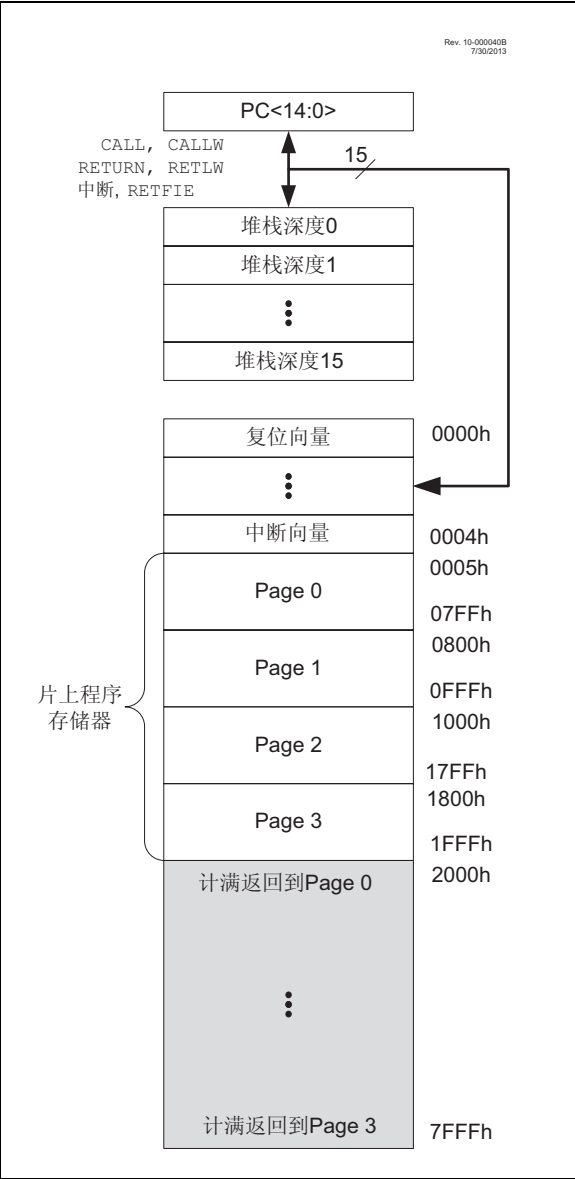
器件	程序存储空间（字）	最后的程序存储器地址	高耐用性闪存地址范围 <sup>(1)</sup>
PIC16(L)F1615/9	4,096	0FFFh	0F80h-0FFFh

注 1： 高耐用性闪存适用于该范围内每个地址的低字节。

3.2 高耐用性闪存

该器件有一个128字节的高耐用性闪存程序存储器（Program Flash Memory, PFM）来代替数据EEPROM。此区域尤其适合预期将在最终产品的整个生命周期内不断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PFM中存储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)F1615/9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。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，所以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FSR<sub>x</sub>H寄存器的bit 7置1，并读取匹配的INDF<sub>x</sub>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操作。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操作符会将bit<7>置1。

### 例3-2: 通过FSR访问程序存储器

```
constants
    DW DATA0           ;First constant
    DW DATA1           ;Second constant
    DW DATA2
    DW DATA3
my_function
    ;... LOTS OF CODE...
    MOVLW DATA_INDEX
    ADDLW LOW constants
    MOVWF FSR1L
    MOVLW HIGH constants;MSb sets
                                automatically
    MOVWF FSR1H
    BTFSC STATUS, C           ;carry from ADDLW?
    INCF FSR1h, f             ;yes
    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，每个存储区有128字节。每个存储区都包含（图3-2）：

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

通过向存储区选择寄存器（Bank Select Register, BSR）写入存储区编号，可以选择工作存储区。未实现的存储器将读为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3.6节“间接寻址”。

数据存储器使用一个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器区的前12个地址（地址x00h/x08h至x0Bh/x8Bh）。表3-1列出了这些寄存器。详细信息，请参见表3-13。

表3-1： 内核寄存器

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

3.3.1.1 状态寄存器

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 位均为不可写位。因此，当执行一条将 STATUS 寄存器作为目标寄存器的指令时，运行结果可能会与预想的不同。

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
—	—	—	TO	PD	Z	DC <sup>(1)</sup>	C <sup>(1)</sup>
bit 7							
							bit 0

图注:		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-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	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
bit 3	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） <sup>(1)</sup> 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 <sup>(1)</sup> （ADDWF、ADDLW、SUBLW 和 SUBWF 指令） <sup>(1)</sup> 1 = 结果的最高有效位发生了进位 0 = 结果的最高有效位未发生进位

注 1：对于借位，极性是相反的。减法是通过加上第二个操作数的二进制补码来执行的。对于移位指令（RRF 和 RLF），此位中将装入源寄存器的最高位或最低位。

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

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

注 1：在减法运算中，C 和 DC 位分别作为借位位和半借位位。

3.3.2 特殊功能寄存器

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

3.3.3 通用RAM

每个数据存储区中有最大80字节的GPR。通用RAM占用每个数据存储区中特殊功能寄存器之后的80字节（地址0x20h/0xA0h至0x6Fh/0xEFh）。

3.3.3.1 线性访问GPR

通用RAM可以通过FSR以非存储区方式访问。这可以简化对大存储器结构的访问。更多信息，请参见第3.6.2节“线性数据存储器”。

3.3.4 公共RAM

有16字节的公共RAM可以从所有存储区中进行访问。

3.3.5 器件存储器映射

表3-2至表3-12给出了存储器映射。

图3-2: 存储区分区

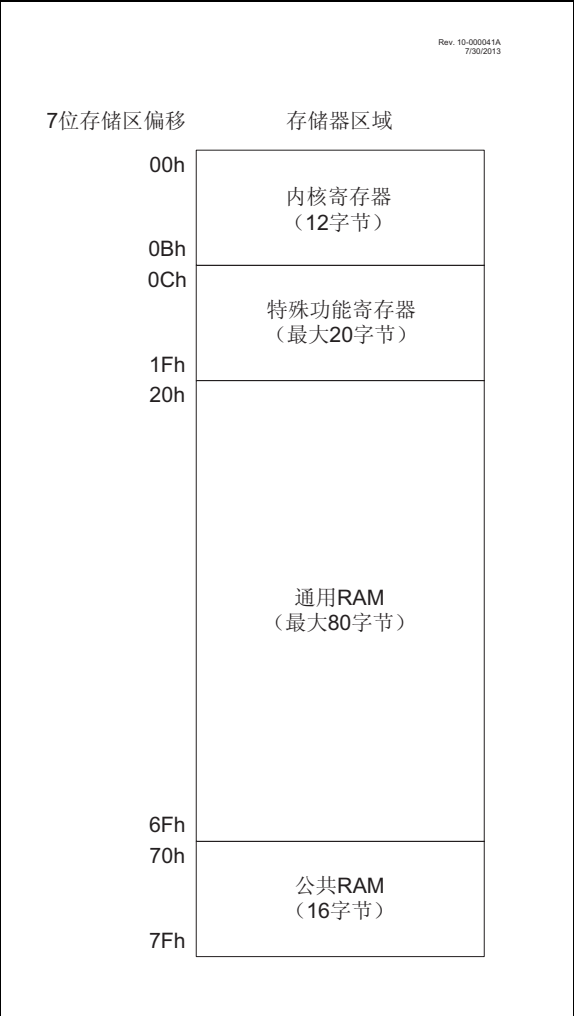


表3-2: PIC16(L)F1615 存储器映射, BANK 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	内核寄存器 (表3-1)	080h	内核寄存器 (表3-1)	100h	内核寄存器 (表3-1)	180h	内核寄存器 (表3-1)	200h	内核寄存器 (表3-1)	280h	内核寄存器 (表3-1)	300h	内核寄存器 (表3-1)	380h	内核寄存器 (表3-1)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	—	08Dh	—	10Dh	—	18Dh	—	20Dh	—	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PIR1	090h	PIE1	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR2	091h	PIE2	111h	CM1CON0	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	391h	IOCAP
012h	PIR3	092h	PIE3	112h	CM1CON1	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	392h	IOCAN
013h	PIR4	093h	PIE4	113h	CM2CON0	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR5	094h	PIE5	114h	CM2CON1	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	—
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	SSP1CON	295h	—	315h	—	395h	—
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	—
017h	TMR1H	097h	—	117h	FVRCON	197h	VREGCON	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	—	218h	—	298h	CCPR2L	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	RC1REG	219h	—	299h	CCPR2H	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh	—	19Dh	RC1STA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	TX1STA	21Eh	—	29Eh	CCPTMRS	31Eh	—	39Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	BAUD1CON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h	通用寄存器 96 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 80 字节	320h	通用寄存器 80 字节	3A0h	通用寄存器 80 字节
		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
		0F0h	公共RAM (快速操作 存储区 70h – 7Fh)	170h	公共RAM (快速操作 存储区 70h – 7Fh)	1F0h	公共RAM (快速操作 存储区 70h – 7Fh)	270h	公共RAM (快速操作 存储区 70h – 7Fh)	2F0h	公共RAM (快速操作 存储区 70h – 7Fh)	370h	公共RAM (快速操作 存储区 70h – 7Fh)	3F0h	公共RAM (快速操作 存储区 70h – 7Fh)
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

图注:  = 未实现的数据存储单元, 读为0。



表3-3: PIC16(L)F1619 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3-1)	080h	内核寄存器 (表3-1)	100h	内核寄存器 (表3-1)	180h	内核寄存器 (表3-1)	200h	内核寄存器 (表3-1)	280h	内核寄存器 (表3-1)	300h	内核寄存器 (表3-1)	380h	内核寄存器 (表3-1)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Eh	WPUB	28Eh	ODCONB	30Eh	SLRCONB	38Eh	INLVLB
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PIR1	090h	PIE1	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR2	091h	PIE2	111h	CM1CON0	191h	PMADRL	211h	SSP1BUF	291h	CCP1RL	311h	—	391h	IOCAP
012h	PIR3	092h	PIE3	112h	CM1CON1	192h	PMADRH	212h	SSP1ADD	292h	CCP1RH	312h	—	392h	IOCAN
013h	PIR4	093h	PIE4	113h	CM2CON0	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR5	094h	PIE5	114h	CM2CON1	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	SSP1CON	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	—	117h	FVRCON	197h	VREGCON	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	—	218h	—	298h	CCP2RL	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	RC1REG	219h	—	299h	CCP2RH	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh	—	19Dh	RC1STA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	TX1STA	21Eh	—	29Eh	CCPTMRS	31Eh	—	39Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	BAUD1CON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h	通用寄存器 96 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 80 字节	320h	通用寄存器 80 字节	3A0h	通用寄存器 80 字节
		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
		0F0h		170h		1F0h		270h		2F0h		370h		3F0h	
07Fh		0FFh	公共RAM (快速操作 存储区 70h – 7Fh)	17Fh	公共RAM (快速操作 存储区 70h – 7Fh)	1FFh	公共RAM (快速操作 存储区 70h – 7Fh)	27Fh	公共RAM (快速操作 存储区 70h – 7Fh)	2FFh	公共RAM (快速操作 存储区 70h – 7Fh)	37Fh	公共RAM (快速操作 存储区 70h – 7Fh)	3FFh	公共RAM (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3-4: PIC16(L)F1615/9 存储器映射, BANK 8-15

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	内核寄存器 (表3-1)	480h	内核寄存器 (表3-1)	500h	内核寄存器 (表3-1)	580h	内核寄存器 (表3-1)	600h	内核寄存器 (表3-1)	680h	内核寄存器 (表3-1)	700h	内核寄存器 (表3-1)	780h	内核寄存器 (表3-1)
40Bh	—	48Bh	—	50Bh	—	58Bh	PID1SETL	60Bh	PID1Z2L	68Bh	—	70Bh	—	78Bh	—
40Ch	—	48Ch	—	50Ch	—	58Ch	PID1SETH	60Ch	PID1Z2H	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	PID1INL	60Dh	PID1Z2U	68Dh	—	70Dh	—	78Dh	—
40Eh	HDRVENC	48Eh	—	50Eh	—	58Eh	PID1INH	60Eh	PID1ACCLL	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	PID1K1L	60Fh	PID1ACCLH	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	PID1K1H	610h	PID1ACCHL	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	PID1K2L	611h	PID1ACCHH	691h	CWG1DBR	711h	WDTC0N0	791h	CRCDATL
412h	—	492h	—	512h	—	592h	PID1K2H	612h	PID1ACCLL	692h	CWG1DBF	712h	WDTC0N1	792h	CRCDATL
413h	TMR4	493h	TMR3L	513h	—	593h	PID1K3L	613h	PID1ACCLL	693h	CWG1AS0	713h	WDTPSL	793h	CRCACCL
414h	PR4	494h	TMR3H	514h	—	594h	PID1K3H	614h	PID1CON	694h	CWG1AS1	714h	WDTPSH	794h	CRCACCH
415h	T4CON	495h	T3CON	515h	—	595h	PID1OUTLL	615h	—	695h	CWG1OCON0	715h	WDTTMR	795h	CRCCHIFTL
416h	T4HLT	496h	T3GCON	516h	—	596h	PID1OUTLH	616h	—	696h	CWG1CON1	716h	—	796h	CRCCHIFTH
417h	T4CLKCON	497h	—	517h	—	597h	PID1OUTH	617h	PWM3DCL	697h	—	717h	—	797h	CRCXORL
418h	T4RST	498h	—	518h	—	598h	PID1OUTH	618h	PWM3DCH	698h	—	718h	SCANLADRL	798h	CRCXORH
419h	—	499h	—	519h	—	599h	PID1OUTUL	619h	PWM3CON	699h	CWG1CLKCON	719h	SCANLADRH	799h	CRCCON0
41Ah	TMR6	49Ah	TMR5L	51Ah	—	59Ah	PID1Z1L	61Ah	PWM4DCL	69Ah	CWG1ISM	71Ah	SCANHADRL	79Ah	CRCCON1
41Bh	PR6	49Bh	TMR5H	51Bh	—	59Bh	PID1Z1H	61Bh	PWM4DCH	69Bh	—	71Bh	SCANHADRH	79Bh	—
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	PID1Z1U	61Ch	PWM4CON	69Ch	—	71Ch	SCANCON0	79Ch	—
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	SCANTRIG	79Dh	—
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	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	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	5EFh	快速操作存储区 70h – 7Fh	66Fh	快速操作存储区 70h – 7Fh	6EFh	快速操作存储区 70h – 7Fh	76Fh	快速操作存储区 70h – 7Fh	7EFh	快速操作存储区 70h – 7Fh
470h		4F0h		570h		5F0h		670h		6F0h		770h		7F0h	
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—

图注: ■ = 未实现的数据存储单元, 读为 0。

表3-5: PIC16(L)F1615/9 存储器映射, BANK 16-23

BANK 16				BANK 17				BANK 18				BANK 19				BANK 20				BANK 21				BANK 22				BANK 23			
800h	内核寄存器 (表3-1)			880h	内核寄存器 (表3-1)			900h	内核寄存器 (表3-1)			980h	内核寄存器 (表3-1)			A00h	内核寄存器 (表3-1)			A80h	内核寄存器 (表3-1)			B00h	内核寄存器 (表3-1)			B80h	内核寄存器 (表3-1)		
80Bh				88Bh				90Bh				98Bh				A0Bh				A8Bh				B0Bh				B8Bh			
80Ch	AT1RESL			88Ch	AT1CLK			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		
80Dh	AT1RESH			88Dh	AT1SIG																										
80Eh	AT1MISSL			88Eh	AT1CSEL1																										
80Fh	AT1MISSH			88Fh	AT1CC1L																										
810h	AT1PERL			890h	AT1CC1H																										
811h	AT1PERH			891h	AT1CCON1																										
812h	AT1PHSL			892h	AT1CSEL2																										
813h	AT1PHSH			893h	AT1CC2L																										
814h	AT1CON0			894h	AT1CC2H																										
815h	AT1CON1			895h	AT1CCON2																										
816h	AT1IR0			896h	AT1CSEL2																										
817h	AT1IE0			897h	AT1CC3L																										
818h	AT1IR1			898h	AT1CC3H																										
819h	AT1IE1			899h	AT1CCON3																										
81Ah	AT1STPTL			89Ah																											
81Bh	AT1STPTH																														
81Ch	AT1ERRL																														
81Dh	AT1ERRH																														
86Fh				8EFh				96Fh				9EFh				A6Fh				AEFh				B6Fh				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			9F0h	快速操作存储区 70h – 7Fh			A70h	快速操作存储区 70h – 7Fh			AF0h	快速操作存储区 70h – 7Fh			B70h	快速操作存储区 70h – 7Fh			BF0h	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3-6: PIC16(L)F1615/9 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	内核寄存器 (表3-1)	C80h	内核寄存器 (表3-1)	D00h	内核寄存器 (表3-1)	D80h	内核寄存器 (表3-1)	E00h	内核寄存器 (表3-1)	E80h	内核寄存器 (表3-1)	F00h	内核寄存器 (表3-1)	F80h	内核寄存器 (表3-1)
C0Bh		C8Bh		D0Bh		D8Bh		E0Bh		E8Bh		F0Bh		F8Bh	
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch		E0Ch		E8Ch		F0Ch		F8Ch	
C0Dh	—	C8Dh	—	D0Dh	—	D8Dh		E0Dh		E8Dh		F0Dh		F8Dh	
C0Eh	—	C8Eh	—	D0Eh	—	D8Eh		E0Eh		E8Eh		F0Eh		F8Eh	
C0Fh	—	C8Fh	—	D0Fh	—	D8Fh		E0Fh		E8Fh		F0Fh		F8Fh	
C10h	—	C90h	—	D10h	—	D90h		E10h		E90h		F10h		F90h	
C11h	—	C91h	—	D11h	—	D91h		E11h		E91h		F11h		F91h	
C12h	—	C92h	—	D12h	—	D92h		E12h		E92h		F12h		F92h	
C13h	—	C93h	—	D13h	—	D93h		E13h		E93h		F13h		F93h	
C14h	—	C94h	—	D14h	—	D94h		E14h		E94h		F14h		F94h	
C15h	—	C95h	—	D15h	—	D95h		E15h		E95h		F15h		F95h	
C16h	—	C96h	—	D16h	—	D96h		E16h		E96h		F16h		F96h	
C17h	—	C97h	—	D17h	—	D97h		E17h		E97h		F17h		F97h	
C18h	—	C98h	—	D18h	—	D98h	请参见表3-7 了解寄存器映射 详细信息	E18h	请参见表3-8 了解寄存器映射 详细信息	E98h	请参见表3-9和 表3-10了解寄存 器映射详细信息	F18h	请参见表3-12 了解寄存器映射 详细信息	F98h	请参见表3-12 了解寄存器映射 详细信息
C19h	—	C99h	—	D19h	—	D99h		E19h		E99h		F19h		F99h	
C1Ah	—	C9Ah	—	D1Ah	—	D9Ah		E1Ah		E9Ah		F1Ah		F9Ah	
C1Bh	—	C9Bh	—	D1Bh	—	D9Bh		E1Bh		E9Bh		F1Bh		F9Bh	
C1Ch	—	C9Ch	—	D1Ch	—	D9Ch		E1Ch		E9Ch		F1Ch		F9Ch	
C1Dh	—	C9Dh	—	D1Dh	—	D9Dh		E1Dh		E9Dh		F1Dh		F9Dh	
C1Eh	—	C9Eh	—	D1Eh	—	D9Eh		E1Eh		E9Eh		F1Eh		F9Eh	
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh		E1Fh		E9Fh		F1Fh		F9Fh	
C20h	未实现 读为0	CA0h	未实现 读为0	D20h	未实现 读为0	DA0h		E20h		EA0h		F20h		FA0h	
C6Fh		CEFh		D6Fh		DEFh		E6Fh		EEFh		F6Fh		FEFh	
C70h	快速操作存储区 70h – 7Fh	CF0h	快速操作存储区 70h – 7Fh	D70h	快速操作存储区 70h – 7Fh	DF0h	快速操作存储区 70h – 7Fh	E70h	快速操作存储区 70h – 7Fh	EF0h	快速操作存储区 70h – 7Fh	F70h	快速操作存储区 70h – 7Fh	FF0h	快速操作存储区 70h – 7Fh
CFFh		CFFh		D7Fh		DFFh		E7Fh		EFFh		F7Fh		FFFh	

图注: ■ = 未实现的数据存储单元, 读为0。

表 3-7: PIC16(L)F1615/9 存储器映射, BANK 27

Bank 27	
D8Ch	SMT1TMRL
D8Dh	SMT1TMRH
D8Eh	SMT1TMRU
D8Fh	SMT1CPRL
D90h	SMT1CPRH
D91h	SMT1CPRU
D92h	SMT1CPWL
D93h	SMT1CPWH
D94h	SMT1CPWU
D95h	SMT1PRL
D96h	SMT1PRH
D97h	SMT1PRU
D98h	SMT1CON0
D99h	SMT1CON1
D9Ah	SMT1STAT
D9Bh	SMT1CLK
D9Ch	SMT1SIG
D9Dh	SMT1WIN
D9Eh	SMT2TMRL
D9Fh	SMT2TMRH
DA0h	SMT2TMRU
DA1h	SMT2CPRL
DA2h	SMT2CPRH
DA3h	SMT2CPRU
DA4h	SMT2CPWL
DA5h	SMT2CPWH
DA6h	SMT2CPWU
DA7h	SMT2PRL
DA8h	SMT2PRH
DA9h	SMT2PRU
DAAh	SMT2CON0
DABh	SMT2CON1
DACH	SMT2STAT
DADh	SMT2CLK
DAEh	SMT2SIG
DAFh	SMT2WIN
DB0h	—
DEFh	—



图注:  = 未实现的数据存储单元, 读为 0。

表 3-8: PIC16(L)F1615/9 存储器映射, BANK 28

Bank 28	
E0Ch	---
E0Dh	---
E0Eh	---
E0Fh	PPSLOCK
E10h	INTPPS
E11h	T0CKIPPS
E12h	T1CKIPPS
E13h	T1GPPS
E14h	CCP1PPS
E15h	CCP2PPS
E16h	ATINPPS
E17h	CWGINPPS
E18h	T2PPS
E19h	T3CKIPPS
E1Ah	T3GPPS
E1Bh	T4PPS
E1Ch	T5CKIPPS
E1Dh	T5GPPS
E1Eh	T6PPS
E1Fh	ATCC1PPS
E20h	SSPCLKPPS
E21h	SSPDATPPS
E22h	SSPSSPPS
E23h	ATCC2PPS
E24h	RXPPS
E25h	CKPPS
E26h	SMT1SIGPPS
E27h	SMT1WINPPS
E28h	CLCIN0PPS
E29h	CLCIN1PPS
E2Ah	CLCIN2PPS
E2Bh	CLCIN3PPS
E2Ch	SMT2SIGPPS
E2Dh	SMT2WINPPS
E2Eh	ATCC3PPS
E2Fh	---
E6Fh	---

图注:  = 未实现的数据存储单元, 读为 0。

# PIC16(L)F1615/9

表 3-9: PIC16(L)F1615 存储器映射, BANK 29

Bank 29	
E8Ch	---
E8Dh	---
E8Eh	---
E8Fh	---
E90h	RA0PPS
E91h	RA1PPS
E92h	RA2PPS
E93h	---
E94h	RA4PPS
E95h	RA5PPS
E96h	---
E97h	---
E98h	---
E99h	---
E9Ah	---
E9Bh	---
E9Ch	---
E9Dh	---
E9Eh	---
E9Fh	---
EA0h	RC0PPS
EA1h	RC1PPS
EA2h	RC2PPS
EA3h	RC3PPS
EA4h	RC4PPS
EA5h	RC5PPS
EA6h	---
EEFh	---


图注:  = 未实现的数据存储单元, 读为 0。

表 3-10: PIC16(L)F1619 存储器映射, BANK 29

Bank 29	
E8Ch	---
E8Dh	---
E8Eh	---
E8Fh	---
E90h	RA0PPS
E91h	RA1PPS
E92h	RA2PPS
E93h	---
E94h	RA4PPS
E95h	RA5PPS
E96h	---
E97h	---
E98h	---
E99h	---
E9Ah	---
E9Bh	---
E9Ch	RB4PPS
E9Dh	RB5PPS
E9Eh	RB6PPS
E9Fh	RB7PPS
EA0h	RC0PPS
EA1h	RC1PPS
EA2h	RC2PPS
EA3h	RC3PPS
EA4h	RC4PPS
EA5h	RC5PPS
EA6h	RC6PPS
EA7h	RC7PPS
EA8h	---
EEFh	---


图注:  = 未实现的数据存储单元, 读为 0。

表3-11: PIC16(L)F1615/9 存储器映射, BANK 30

Bank 30	
F0Ch	---
F0Dh	---
F0Eh	---
F0Fh	CLCDATA
F10h	CLC1CON
F11h	CLC1POL
F12h	CLC1SEL0
F13h	CLC1SEL1
F14h	CLC1SEL2
F15h	CLC1SEL3
F16h	CLC1GLS0
F17h	CLC1GLS1
F18h	CLC1GLS2
F19h	CLC1GLS3
F1Ah	CLC2CON
F1Bh	CLC2POL
F1Ch	CLC2SEL0
F1Dh	CLC2SEL1
F1Eh	CLC2SEL2
F1Fh	CLC2SEL3
F20h	CLC2GLS0
F21h	CLC2GLS1
F22h	CLC2GLS2
F23h	CLC2GLS3
F24h	CLC3CON
F25h	CLC3POL
F26h	CLC3SEL0
F27h	CLC3SEL1
F28h	CLC3SEL2
F29h	CLC3SEL3
F2Ah	CLC3GLS0
F2Bh	CLC3GLS1
F2Ch	CLC3GLS2
F2Dh	CLC3GLS3
F2Eh	CLC4CON
F2Fh	CLC4POL
F30h	CLC4SEL0
F31h	CLC4SEL1
F32h	CLC4SEL2
F33h	CLC4SEL3
F34h	CLC4GLS0
F35h	CLC4GLS1
F36h	CLC4GLS2
F37h	CLC4GLS3
F38h	---
F6Fh	---



图注:  = 未实现的数据存储单元, 读为0。

表3-12: PIC16(L)F1615/9 存储器映射, BANK 31

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

图注:  = 未实现的数据存储单元, 读为0。

# PIC16(L)F1615/9

## 3.3.6 内核功能寄存器汇总

表3-13列出了内核功能寄存器，可从任何存储区访问这些寄存器。

表3-13: 内核功能寄存器汇总

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 0-31												
x00h或x80h	INDF0	通过用FSR0H/FSR0L的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx	uuuu uuuu	
x01h或x81h	INDF1	通过用FSR1H/FSR1L的内容寻址这个存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx	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	—	—	—	$\overline{TO}$	$\overline{PD}$	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	—	—	—	BSR<4:0>					---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	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	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。



表3-14: 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	--xx xxxx	
00Dh	PORTB <sup>(4)</sup>	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	xxxx ----	
00Eh	PORTC	RC7 <sup>(4)</sup>	RC6 <sup>(4)</sup>	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	xxxx xxxx	
00Fh	—	未实现								—	—	
010h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
011h	PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF	-00- 0000	-00- 0000	
012h	PIR3	—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	--00 0000	--00 0000	
013h	PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	0000 0000	0000 0000	
014h	PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF	0000 -000	0000 -000	
015h	TMR0	8位Timer0计数的保持寄存器								xxxx xxxx	uuuu uuuu	
016h	TMR1L	16位TMR1计数最低有效字节的保持寄存器								xxxx xxxx	uuuu uuuu	
017h	TMR1H	16位TMR1计数最高有效字节的保持寄存器								xxxx xxxx	uuuu uuuu	
018h	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	T1SYNC	—	TMR1ON	0000 -0-0	uuuu -u-u	
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		0000 0x00	uuuu uxuu	
01Ah	TMR2	Timer2模块寄存器								0000 0000	0000 0000	
01Bh	PR2	Timer2周期寄存器								1111 1111	1111 1111	
01Ch	T2CON	ON	CKPS<2:0>			OUTPS<3:0>					0000 0000	0000 0000
01Dh	T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					0000 0000	0000 0000	
01Eh	T2CLKCON	—	—	—	—	CS<3:0>					---- 0000	---- 0000
01Fh	T2RST	—	—	—	—	RSEL<3:0>					---- 0000	---- 0000

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

注 1: 仅限PIC16F1615/9。

2: 未实现, 读为1。

3: 仅限PIC16(L)F1615。

4: 仅限PIC16(L)F1619。

表3-14: 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 1											
08Ch	TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
08Dh	TRISB <sup>(4)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
08Eh	TRISC	TRISC7 <sup>(4)</sup>	TRISC6 <sup>(4)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
08Fh	—	未实现								—	—
090h	—	未实现								—	—
090h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
091h	PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	-00- 0000	-00- 0000
092h	PIE3	—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	--00 0000	--00 0000
093h	PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	0000 0000	0000 0000
094h	PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE	0000 -000	0000 -000
095h	OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111	1111 1111
096h	PCON	STKOVF	STKUNF	WDTWV	RWD $\overline{T}$	RMCLR	R $\overline{I}$	POR	BOR	00-1 11qq	qq-q qquu
097h	—	未实现								—	—
098h	OSCTUNE	—	—	TUN<5:0>						--00 0000	--00 0000
099h	OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		0011 1-00	0011 1-00
09Ah	OSCSTAT	—	PLL $\overline{R}$	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	-000 0000	-qqq qqqq
09Bh	ADRESL	ADC结果寄存器的低字节								xxxx xxxx	uuuu uuuu
09Ch	ADRESH	ADC结果寄存器的高字节								xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>					GO/ $\overline{DONE}$	ADON	-000 0000	-000 0000
09Eh	ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		0000 --00	0000 --00
09Fh	ADCON2	TRIGSEL<4:0>					—	—	—	0000 0---	0000 0---

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

注 1: 仅限 PIC16F1615/9。

2: 未实现, 读为 1。

3: 仅限 PIC16(L)F1615。

4: 仅限 PIC16(L)F1619。

表 3-14: 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	LATA3	LATA2	LATA1	LATA0	--xx xxxx	--uu uuuu
10Dh	LATB <sup>(4)</sup>	LATB7	LATB6	LATB5	LATB4	—	—	—	—	xxxx ----	uuuu ----
10Eh	LATC	LATC7 <sup>(4)</sup>	LATC6 <sup>(4)</sup>	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	uuuu uuuu
10Fh	—	未实现								—	—
110h	—	未实现								—	—
111h	CM1CON0	C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC	00-0 -100	00-0 -100
112h	CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			0000 -000	0000 -000
113h	CM2CON0 <sup>(4)</sup>	C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC	00-0 -100	00-0 -100
114h	CM2CON1 <sup>(4)</sup>	C2INTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			0000 -000	0000 -000
115h	CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	---- --00	---- --00
116h	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	10-- ---q	uu-- ---u
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		0q00 0000	0q00 0000
118h	DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	0-0- 00--	0-0- 00--
119h	DAC1CON1	DAC1R<7:0>								0000 0000	0000 0000
11Ah	—	未实现								—	—
11Bh	—	未实现								—	—
11Ch	ZCD1CON	ZCD1EN	—	ZCD1OUT	ZCD1POL	—	—	ZCD1INTP	ZCD1INTN	0-00 --00	0-00 --00
11Dh	—	未实现								—	—
11Eh	—	未实现								—	—
11Fh	—	未实现								—	—

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

注 1: 仅限 PIC16F1615/9。

2: 未实现, 读为 1。

3: 仅限 PIC16(L)F1615。

4: 仅限 PIC16(L)F1619。

表3-14: 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 3											
18Ch	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	---1 -111	---1 -111
18Dh	ANSELB <sup>(4)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	--11 ----	--11 ----
18Eh	ANSELC	ANSC7 <sup>(4)</sup>	ANSC6 <sup>(4)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	11-- 1111	11-- 1111
18Fh	—	未实现								—	—
190h	—	未实现								—	—
191h	PMADRL	闪存程序存储器地址寄存器的低字节								0000 0000	0000 0000
192h	PMADRH	— <sup>(2)</sup>	闪存程序存储器地址寄存器的高字节							1000 0000	1000 0000
193h	PMDATL	闪存程序存储器读数据寄存器的低字节								xxxx xxxx	uuuu uuuu
194h	PMDATH	—	—	闪存程序存储器读数据寄存器的高字节						--xx xxxx	--uu uuuu
195h	PMCON1	— <sup>(2)</sup>	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	1000 x000	1000 q000
196h	PMCON2	闪存程序存储器控制寄存器2								0000 0000	0000 0000
197h	VREGCON <sup>(1)</sup>	—	—	—	—	—	—	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	波特率通用寄存器的低字节								0000 0000	0000 0000
19Ch	SP1BRGH	波特率通用寄存器的高字节								0000 0000	0000 0000
19Dh	RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19Eh	TX1STA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	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, r = 保留。阴影单元未实现, 读为0。

注 1: 仅限 PIC16F1615/9。

2: 未实现, 读为1。

3: 仅限 PIC16(L)F1615。

4: 仅限 PIC16(L)F1619。

表 3-14: 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 <sup>(4)</sup>	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----
20Eh	WPUC	WPUC7 <sup>(4)</sup>	WPUC6 <sup>(4)</sup>	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	1111 1111	111 1111
20Fh	—	未实现								—	—
210h	—	未实现								—	—
211h	SSP1BUF	同步串行端口接收缓冲/发送寄存器								xxxx xxxx	xxxx xxxx
212h	SSP1ADD	ADD<7:0>								0000 0000	0000 0000
213h	SSP1MSK	MSK<7:0>								1111 1111	1111 1111
214h	SSP1STAT	SMP	CKE	D/ $\overline{A}$	P	S	R/ $\overline{W}$	UA	BF	0000 0000	0000 0000
215h	SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000	0000 0000
216h	SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000
217h	SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000
218h 至 21Fh	—	未实现								—	—

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

- 注 1: 仅限 PIC16F1615/9。  
2: 未实现, 读为 1。  
3: 仅限 PIC16(L)F1615。  
4: 仅限 PIC16(L)F1619。

表3-14: 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 5											
28Ch	ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	--00 -000	--00 -000
28Dh	ODCONB <sup>(4)</sup>	ODB7	ODB6	ODB5	ODB4	—	—	—	—	0000 ----	0000 ----
28Eh	ODCONC	ODC7 <sup>(4)</sup>	ODC6 <sup>(4)</sup>	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000 0000	0000 0000
28Fh	—	未实现								—	—
290h	—	未实现								—	—
291h	CCP1RL	捕捉/比较/PWM1 寄存器 (LSB)								xxxx xxxx	uuuu uuuu
292h	CCP1RH	捕捉/比较/PWM1 寄存器 (MSB)								xxxx xxxx	uuuu uuuu
293h	CCP1CON	EN	—	OUT	FMT	MODE<3:0>				0000 0000	0000 0000
294h	CCP1CAP	—	—	—	—	—	CTS<2:0>			---- -000	---- -000
295h — 297h	—	未实现								—	—
298h	CCP2RL	捕捉/比较/PWM2 寄存器 (LSB)								xxxx xxxx	uuuu uuuu
299h	CCP2RH	捕捉/比较/PWM2 寄存器 (MSB)								xxxx xxxx	uuuu uuuu
29Ah	CCP2CON	EN	—	OUT	FMT	MODE<3:0>				0000 0000	0000 0000
29Bh	CCP2CAP	—	—	—	—	—	CTS<2:0>			---- -000	---- -000
29Ch	—	未实现								—	—
29Dh	—	未实现								—	—
29Eh	CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		0000 0000	0000 0000
29Fh	—	未实现								—	—
Bank 6											
30Ch	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	--11 -111	--11 -111
30Dh	SLRCONB <sup>(4)</sup>	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	1111 ----	1111 ----
30Eh	SLRCONC	SLRC7 <sup>(4)</sup>	SLRC6 <sup>(4)</sup>	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	1111 1111
30Fh — 31Fh	—	未实现								—	—

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

注 1: 仅限 PIC16F1615/9。

2: 未实现, 读为 1。

3: 仅限 PIC16(L)F1615。

4: 仅限 PIC16(L)F1619。

表 3-14: 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 7											
38Ch	INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	--11 1111	--11 1111
38Dh	INVLVB <sup>(4)</sup>	INVLVB7	INVLVB6	INVLVB5	INVLVB4	—	—	—	—	1111 ----	1111 ----
38Eh	INLVLC	INLVLC7 <sup>(4)</sup>	INLVLC6 <sup>(4)</sup>	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	1111 1111	1111 1111
30Fh	—	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 <sup>(4)</sup>	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	0000 ----
395h	IOCBN <sup>(4)</sup>	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	0000 ----
396h	IOCBF <sup>(4)</sup>	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	0000 ----
397h	IOCCP	IOCCP7 <sup>(4)</sup>	IOCCP6 <sup>(4)</sup>	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
398h	IOCCN	IOCCN7 <sup>(4)</sup>	IOCCN6 <sup>(4)</sup>	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
399h	IOCCF	IOCCF7 <sup>(4)</sup>	IOCCF6 <sup>(4)</sup>	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000
39Ah 至 39Fh	—	未实现								—	—

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

- 注 1: 仅限 PIC16F1615/9。  
2: 未实现, 读为 1。  
3: 仅限 PIC16(L)F1615。  
4: 仅限 PIC16(L)F1619。

表3-14： 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	—	未实现								—	—
40Dh	—	未实现								—	—
40Eh	HIDRVC	—	—	HIDC5	HIDC4	—	—	—	—	--00 ----	--00 ----
40Fh 至 412h	—	未实现								—	—
413h	TMR4	Timer4 模块寄存器								0000 0000	0000 0000
414h	PR4	Timer4 周期寄存器								1111 1111	1111 1111
415h	T4CON	ON	CKPS<2:0>			OUTPS<3:0>				0000 0000	0000 0000
416h	T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>				0000 0000	0000 0000	
417h	T4CLKCON	—	—	—	—	CS<3:0>				---- 0000	---- 0000
418h	T4RST	—	—	—	—	RSEL<3:0>				---- 0000	---- 0000
419h	—	未实现								—	—
41Ah	TMR6	Timer6 模块寄存器								0000 0000	0000 0000
41Bh	PR6	Timer6 周期寄存器								1111 1111	1111 1111
41Ch	T6CON	ON	CKPS<2:0>			OUTPS<3:0>				0000 0000	0000 0000
41Dh	T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>				0000 0000	0000 0000	
41Eh	T6CLKCON	—	—	—	—	CS<3:0>				---- 0000	---- 0000
41Fh	T6RST	—	—	—	—	RSEL<3:0>				---- 0000	---- 0000

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

注 1： 仅限PIC16F1615/9。

2： 未实现，读为1。

3： 仅限PIC16(L)F1615。

4： 仅限PIC16(L)F1619。



表 3-14: 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 9											
48Ch 至 492h	—	未实现								—	—
493h	TMR3L	Timer3 模块寄存器								xxxx xxxx	xxxx xxxx
494h	TMR3H	Timer3 模块寄存器								xxxx xxxx	xxxx xxxx
495h	T3CON	TMR3CS<1:0>		T3CKPS<1:0>		—	T3SYNC	—	TMR3CON	xxxx -x-x	xxxx -x-x
496h	T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ DONE	T3GVAL	T3GSS<1:0>		xxxx xxxx	xxxx xxxx
497h 至 499h	—	未实现								—	—
49Ah	TMR5L	Timer5 模块寄存器								xxxx xxxx	xxxx xxxx
49Bh	TMR5H	Timer5 模块寄存器								xxxx xxxx	xxxx xxxx
49Ch	T5CON	TMR5CS<1:0>		T5CKPS<1:0>		—	T5SYNC	—	TMR5CON	xxxx -x-x	xxxx -x-x
49Dh	T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/ DONE	T5GVAL	T5GSS<1:0>		xxxx xxxx	xxxx xxxx
49Eh	—	未实现								—	—
49Fh	—	未实现								—	—
Bank 10											
50Ch 至 51Fh	—	未实现								—	—

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

- 注 1: 仅限 PIC16F1615/9。  
2: 未实现, 读为 1。  
3: 仅限 PIC16(L)F1615。  
4: 仅限 PIC16(L)F1619。

表3-14： 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 11											
58Ch	PID1SELT	SET<7:0>								xxxx xxxx	xxxx xxxx
58Dh	PID1SETH	SET<15:8>								xxxx xxxx	xxxx xxxx
58Eh	PID1INL	IN<7:0>								0000 0000	0000 0000
58Fh	PID1INH	IN<15:8>								0000 0000	0000 0000
590h	PID1K1L	K1<7:0>								xxxx xxxx	xxxx xxxx
591h	PID1K1H	K1<15:8>								xxxx xxxx	xxxx xxxx
592h	PID1K2L	K2<7:0>								xxxx xxxx	xxxx xxxx
593h	PID1K2H	K2<15:8>								xxxx xxxx	xxxx xxxx
594h	PID1K3L	K3<7:0>								xxxx xxxx	xxxx xxxx
595h	PID1K3H	K3<15:8>								xxxx xxxx	xxxx xxxx
596h	PID1OUTLL	OUT<7:0>								0000 0000	0000 0000
597h	PID1OUTLH	OUT<15:8>								0000 0000	0000 0000
598h	PID1OUTHL	OUT<23:16>								0000 0000	0000 0000
599h	PID1OUTHH	OUT<31:24>								0000 0000	0000 0000
59Ah	PID1OUTU	—	—	—	—	OUT<35:32>				---- 0000	---- 0000
59Bh	PID1Z1L	Z1<7:0>								0000 0000	0000 0000
59Ch	PID1Z1H	Z1<15:8>								0000 0000	0000 0000
59Dh	PID1Z1U	—	—	—	—	—	—	—	Z116	---- ---0	---- ---0
59Eh	—	未实现								—	—
59Fh	—	未实现								—	—

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

注 1： 仅限PIC16F1615/9。

2： 未实现，读为1。

3： 仅限PIC16(L)F1615。

4： 仅限PIC16(L)F1619。

表 3-14: 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	PID1Z2L	Z2<7:0>								0000 0000	0000 0000
60Dh	PID1Z2H	Z2<15:8>								0000 0000	0000 0000
60Eh	PID1Z2U	—	—	—	—	—	—	—	Z216	---- ---0	---- ---0
60Fh	PID1ACCLL	ACC<7:0>								0000 0000	0000 0000
610h	PID1ACCLH	ACC<15:8>								0000 0000	0000 0000
611h	PID1ACCHL	ACC<23:16>								0000 0000	0000 0000
612h	PID1ACCHH	ACC<31:24>								0000 0000	0000 0000
613h	PID1ACCU	—	—	—	—	—	ACC<34:32>			---- -000	---- -000
614h	PID1CON	EN	BUSY	—	—	—	MODE<2:0>			00-- 0000	00-- 0000
615h	—	未实现								—	—
616h	—	未实现								—	—
617h	PWM3DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	xx-- ----
618h	PWM3DCH	DC<9:2>								xxxx xxxx	xxxx xxxx
619h	PWM3CON	EN	—	OUT	POL	—	—	—	—	0-x0 ----	0-x0 ----
61Ah	PWM4DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	xx-- ----
61Bh	PWM4DCH	DC<9:2>								xxxx xxxx	xxxx xxxx
61Ch	PWM4CON	EN	—	OUT	POL	—	—	—	—	0-x0 ----	0-x0 ----
61Dh 至 61Fh	—	未实现								—	—

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

- 注 1: 仅限 PIC16F1615/9。  
2: 未实现, 读为 1。  
3: 仅限 PIC16(L)F1615。  
4: 仅限 PIC16(L)F1619。

表3-14： 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 13											
68Ch 至 690h	—	未实现								—	—
691h	CWG1DBR	—	—	DBR<5:0>						--00 0000	--00 0000
692h	CWG1DBF	—	—	DBF<5:0>						--xx xxxx	--xx xxxx
693h	CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	0001 01--	00001 01--
694h	CWG1AS1	—	TMR6AS	TMR4AS	TMR2AS	—	C2AS <sup>(4)</sup>	C1AS	INAS	-000 -000	-000 -000
695h	CWG1OCON0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	0000 0000	0000 0000
696h	CWG1CON0	EN	LD	—	—	—	MODE<2:0>			00-- -000	00-- -000
697h	CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	--x- 0000	--x- 0000
698h	—	未实现								—	—
699h	CWG1CLKCON	—	—	—	—	—	—	—	CS	---- ---0	---- --0
69Ah	CWG1ISM	—	—	—	—	IS<3:0>				---- 0000	---- 0000
69Bh 至 6EFh	—	未实现								—	—

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

注 1： 仅限PIC16F1615/9。

2： 未实现，读为1。

3： 仅限PIC16(L)F1615。

4： 仅限PIC16(L)F1619。

表 3-14: 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 至 710h	—	未实现								—	—
711h	WDTCN0	—	—	WDTPS<4:0>					SEN	--qq qqqq	--qq qqqq
712h	WDTCN1	—	WDTCS<2:0>			—	WINDOW<2:0>			-qqq -qqq	-qqq -qqq
713h	WDTPSL	PSCNT<7:0>								0000 0000	0000 0000
714h	WDTPSH	PSCNT<15:8>								0000 0000	0000 0000
715h	WDTTMR	WDTTMR<4:0>					STATE	PSCNT<17:16>		0000 0000	0000 0000
716h	—	未实现								—	—
717h	—	未实现								—	—
718h	SCANLADRL	LADR<7:0>								0000 0000	0000 0000
719h	SCANLADRH	LADR<15:8>								0000 0000	0000 0000
71Ah	SCANHADRL	HADR<7:0>								1111 1111	1111 1111
71Bh	SCANHADRH	HADR<15:8>								1111 1111	1111 1111
71Ch	SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		0000 0-00	0000 0-00
71Dh	SCANTRIG					TSEL<3:0>			---- 0000	---- 0000	
71Eh	—	未实现								—	—
71Fh	—	未实现								—	—

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

注 1: 仅限 PIC16F1615/9。

2: 未实现, 读为 1。

3: 仅限 PIC16(L)F1615。

4: 仅限 PIC16(L)F1619。

表3-14: 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 15											
78Ch 至 790h	—	未实现								—	—
791h	CRCDATL	DAT<7:0>								xxxx xxxx	xxxx xxxx
792h	CRCDATH	DAT<15:8>								xxxx xxxx	xxxx xxxx
793h	CRCACCL	ACC<7:0>								0000 0000	0000 0000
794h	CRCACCH	ACC<15:8>								0000 0000	0000 0000
795h	CRCSHIFTL	SHIFT<7:0>								0000 0000	0000 0000
796h	CRCSHIFTH	SHIFT<15:8>								0000 0000	0000 0000
797h	CRCXORL	XOR<7:1>							—	xxxx xxx-	xxxx xxx-
798h	CRCXORH	XOR<15:8>								xxxx xxxX	xxxx xxxX
799h	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	0000 --00	0000 -00
79Ah	CRCCON1	DLEN<3:0>				PLEN<3:0>				0000 0000	0000 0000
79Bh 至 79Fh	—	未实现								—	—

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

注 1: 仅限PIC16F1615/9。

2: 未实现, 读为1。

3: 仅限PIC16(L)F1615。

4: 仅限PIC16(L)F1619。

表 3-14: 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 16											
80Ch	AT1RESL	RES<7:0>								xxxx xxxx	xxxx xxxx
80Dh	AT1RESH	—	—	—	—	—	—	RES<9:8>		---- --xx	---- --xx
80Eh	AT1MISSL	MISS<7:0>								xxxx xxxx	xxxx xxxx
80Fh	AT1MISSH	MISS<15:8>								xxxx xxxx	xxxx xxxx
810h	AT1PERL	PER<7:0>								xxxx xxxx	xxxx xxxx
811h	AT1PERH	POV	PER<14:8>							xxxx xxxx	xxxx xxxx
812h	AT1PHSL	PHS<7:0>								xxxx xxxx	xxxx xxxx
813h	AT1PHSH	—	—	—	—	—	—	PHS<9:8>		---- --xx	---- --xx
814h	AT1CON0	EN	PREC	PS<1:0>		POL	—	APMOD	MODE	0x00 --00	0x00 -00
815h	AT1CON1	—	PHP	—	PRP	—	MPP	ACCS	VALID	0000 0000	0000 0000
816h	AT1IR0	—	—	—	—	—	PHSIF	MISSIF	PERIF	----000	----000
817h	AT1IE0	—	—	—	—	—	PHSIE	MISSIE	PERIE	----000	----000
818h	AT1IR1	—	—	—	—	—	CC3IF	CC2IF	CC1IF	----000	----000
819h	AT1IE1	—	—	—	—	—	CC3IE	CC2IE	CC1IE	----000	----000
81Ah	AT1STPTL	STPT<7:0>								xxxx xxxx	xxxx xxxx
81Bh	AT1STPTH	—	STPT<14:8>							-xxx xxxx	-xxx xxxx
81Ch	AT1ERRL	ERR<7:0>								xxxx xxxx	xxxx xxxx
81Dh	AT1ERRH	ERR<15:8>								xxxx xxxx	xxxx xxxx
81Eh	—	未实现								—	—
81Fh	—	未实现								—	—

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

- 注 1: 仅限 PIC16F1615/9。  
2: 未实现, 读为 1。  
3: 仅限 PIC16(L)F1615。  
4: 仅限 PIC16(L)F1619。

表3-14： 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 17											
88Ch	AT1CLK	—	—	—	—	—	—	—	CS0	---- --0	---- --0
88Dh	AT1SIG	—	—	—	—	—	SSEL<2:0>			---- -000	---- -000
88Eh	AT1CSEL1	—	—	—	—	—	CP1S<2:0>			---- -000	---- -000
88Fh	AT1CC1L	CC1<7:0>								0000 0000	0000 0000
890h	AT1CC1H	—	—	—	—	—	—	CC1<9:8>		---- -000	---- -000
891h	AT1CCON1	CC1EN	—	—	CC1POL	CAP1P	—	—	CC1MODE	0--0 0--0	0--0 0--0
892h	AT1CSEL2	—	—	—	—	—	CP2S<2:0>			---- -000	---- -000
893h	AT1CC2L	CC2<7:0>								0000 0000	0000 0000
894h	AT1CC2H	—	—	—	—	—	—	CC2<9:8>		---- -000	---- -000
895h	AT1CCON2	CC2EN	—	—	CC2POL	CAP2P	—	—	CC2MODE	0--0 0--0	0--0 0--0
896h	AT1CSEL3	—	—	—	—	—	CP3S<2:0>			---- -000	---- -000
897h	AT1CC1L	CC3<7:0>								0000 0000	0000 0000
898h	AT1CC1H	—	—	—	—	—	—	CC3<9:8>		---- -000	---- -000
899h	AT1CCON1	CC3EN	—	—	CC3POL	CAP3P	—	—	CC3MODE	0--0 0--0	0--0 0--0
89Ah 至 89Fh	—	未实现								—	—
Bank 18-26											
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，r = 保留。阴影单元未实现，读为0。

注 1： 仅限PIC16F1615/9。

2： 未实现，读为1。

3： 仅限PIC16(L)F1615。

4： 仅限PIC16(L)F1619。



表3-14: 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											
D80h 至 D8Bh	—	未实现								—	—
D8Ch	SMT1TMRL	SMT1TMR<7:0>								0000 0000	0000 0000
D8Dh	SMT1TMRH	SMT1TMR<15:8>								0000 0000	0000 0000
D8Eh	SMT1TMRU	SMT1TMR<23:16>								0000 0000	0000 0000
D8Fh	SMT1CPRL	SMT1CPR<7:0>								xxxx xxxx	xxxx xxxx
D90h	SMT1CPRH	SMT1CPR<15:8>								xxxx xxxx	xxxx xxxx
D91h	SMT1CPRU	SMT1CPR<23:16>								xxxx xxxx	xxxx xxxx
D92h	SMT1CPWL	SMT1CPW<7:0>								xxxx xxxx	xxxx xxxx
D93h	SMT1CPWH	SMT1CPW<15:8>								xxxx xxxx	xxxx xxxx
D94h	SMT1CPWU	SMT1CPW<23:16>								xxxx xxxx	xxxx xxxx
D95h	SMT1PRL	SMT1PR<7:0>								xxxx xxxx	xxxx xxxx
D96h	SMT1PRH	SMT1PR<15:8>								xxxx xxxx	xxxx xxxx
D97h	SMT1PRU	SMT1PR<23:16>								xxxx xxxx	xxxx xxxx
D98h	SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1PS<1:0>		0-00 0000	0-00 0000
D99h	SMT1CON1	SMT1GO	REPEAT	—	—	MODE<3:0>				00-- 0000	00-- 0000
D9Ah	SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	000- -000	000- -000
D9Bh	SMT1CLK	—	—	—	—	—	CSEL<2:0>			---- -000	---- -000
D9Ch	SMT1SIG	—	—	—	SSEL<4:0>					---0 0000	---0 0000
D9Dh	SMT1WIN	—	—	—	WSEL<4:0>					---0 0000	---0 0000
D9Eh	SMT2TMRL	SMT2TMR<7:0>								0000 0000	0000 0000
D9Fh	SMT2TMRH	SMT2TMR<15:8>								0000 0000	0000 0000
DA0h	SMT2TMRU	SMT2TMR<23:16>								0000 0000	0000 0000
DA1h	SMT2CPRL	SMT2CPR<7:0>								xxxx xxxx	xxxx xxxx
DA2h	SMT2CPRH	SMT2CPR<15:8>								xxxx xxxx	xxxx xxxx
DA3h	SMT2CPRU	SMT2CPR<23:16>								xxxx xxxx	xxxx xxxx
DA4h	SMT2CPWL	SMT2CPW<7:0>								xxxx xxxx	xxxx xxxx

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

- 注 1: 仅限PIC16F1615/9。  
 2: 未实现, 读为1。  
 3: 仅限PIC16(L)F1615。  
 4: 仅限PIC16(L)F1619。

表 3-14: 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（续）											
DA5h	SMT2CPWH	SMT2CPW<15:8>								xxxx xxxx	xxxx xxxx
DA6h	SMT2CPWU	SMT2CPW<23:16>								xxxx xxxx	xxxx xxxx
DA7h	SMT2PRL	SMT2PR<7:0>								xxxx xxxx	xxxx xxxx
DA8h	SMT2PRH	SMT2PR<15:8>								xxxx xxxx	xxxx xxxx
DA9h	SMT2PRU	SMT2PR<23:16>								xxxx xxxx	xxxx xxxx
DAAh	SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT2PS<1:0>		0-00 0000	0-00 0000
DABh	SMT2CON1	SMT2GO	REPEAT	—	—	MODE<3:0>				00-- 0000	00-- 0000
DACH	SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	000- -000	000- -000
DADh	SMT2CLK	—	—	—	—	—	CSEL<2:0>			---- -000	---- -000
DAEh	SMT2SIG	—	—	—	SSEL<4:0>					---0 0000	---0 0000
DAFh	SMT2WIN	—	—	—	WSEL<4:0>					---0 0000	---0 0000

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

注 1: 仅限 PIC16F1615/9。

2: 未实现, 读为 1。

3: 仅限 PIC16(L)F1615。

4: 仅限 PIC16(L)F1619。

表3-14: 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
<b>Bank 28</b>											
E0Ch 至 E0Eh	—	未实现								—	—
E0Fh	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	---- -00	---- -00
E10h	INTPPS	—	—	—	INTPPS<4:0>					---0 0010	---0 0010
E11h	T0CKIPPS	—	—	—	T0CKIPPS<4:0>					---0 0010	---0 0010
E12h	T1CKIPPS	—	—	—	T1CKIPPS<4:0>					---0 0101	---0 0101
E13h	T1GPPS	—	—	—	T1GPPS<4:0>					---0 0100	---0 0100
E14h	CCP1PPS	—	—	—	CCP1PPS<4:0>					---1 0101	---1 0101
E15h	CCP2PPS	—	—	—	CCP2PPS<4:0>					---1 0011	---1 0011
E16h	ATINPPS	—	—	—	ATINPPS<4:0>					---1 0101	---1 0101
E17h	CWGINPPS	—	—	—	CWGINPPS<4:0>					---0 0010	---0 0010
E18h	T2PPS	—	—	—	T2PPS<4:0>					---0 0101	---0 0101
E19h	T3CKIPPS	—	—	—	T3CKIPPS<4:0>					---1 0101	---1 0101
E1Ah	T3GPPS	—	—	—	T3GPPS<4:0>					---1 0100	---1 0100
E1Bh	T4PPS	—	—	—	T4PPS<4:0>					---1 0001	---1 0001
E1Ch	T5CKIPPS	—	—	—	T5CKIPPS<4:0>					---1 0000	---1 0000
E1Dh	T5GPPS	—	—	—	T5GPPS<4:0>					---1 0011	---1 0011
E1Eh	T6PPS	—	—	—	T6PPS<4:0>					---0 0011	---0 0011
E1Fh	ATCC1PPS	—	—	—	ATCC1PPS<4:0>					---1 0011	---1 0011
E20h	SSPCLKPPS <sup>(3)</sup>	—	—	—	SSPCLKPPS<4:0>					---1 0000	---1 0000
E20h	SSPCLKPPS <sup>(4)</sup>	—	—	—	SSPCLKPPS<4:0>					---1 0000	---0 1110
E21h	SSPDATPPS <sup>(3)</sup>	—	—	—	SSPDATPPS<4:0>					---1 0001	---1 0001
E21h	SSPDATPPS <sup>(4)</sup>	—	—	—	SSPDATPPS<4:0>					---1 0001	---0 1100
E22h	SSPSSPPS <sup>(3)</sup>	—	—	—	SSPSSPPS<4:0>					---1 0011	---1 0011
E22h	SSPSSPPS <sup>(4)</sup>	—	—	—	SSPSSPPS<4:0>					---1 0110	---1 0110
E23h	ATCC2PPS	—	—	—	ATCC2PPS<4:0>					---1 0100	---1 0100

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

- 注 1: 仅限 PIC16F1615/9。  
 2: 未实现, 读为1。  
 3: 仅限 PIC16(L)F1615。  
 4: 仅限 PIC16(L)F1619。

表3-14： 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）											
E24h	RXPPS <sup>(3)</sup>	—	—	—			RXPPS<4:0>			---1 0101	---1 0101
E24h	RXPPS <sup>(4)</sup>	—	—	—			RXPPS<4:0>			---0 1101	---0 1101
E25h	CKPPS <sup>(3)</sup>	—	—	—			CKPPS<4:0>			---1 0100	---1 0100
E25h	CKPPS <sup>(4)</sup>	—	—	—			CKPPS<4:0>			---0 1111	---0 1111
E26h	SMT1SIGPPS	—	—	—			SMT1SIGPPS<4:0>			---0 0100	---0 0100
E27h	SMT1WINPPS	—	—	—			SMT1WINPPS<4:0>			---0 0101	---0 0101
E28h	CLCIN0PPS	—	—	—			CLCIN0PPS<4:0>			---1 0011	---1 0011
E29h	CLCIN1PPS	—	—	—			CLCIN1PPS<4:0>			---1 0100	---1 0100
E2Ah	CLCIN2PPS	—	—	—			CLCIN2PPS<4:0>			---1 0001	---1 0001
E2Bh	CLCIN3PPS	—	—	—			CLCIN3PPS<4:0>			---0 0101	---0 0101
E2Ch	SMT2SIGPPS	—	—	—			SMT2SIGPPS<4:0>			---1 0001	---1 0001
E2Dh	SMT2WINPPS	—	—	—			SMT2WINPPS<4:0>			---0 0011	---0 0011
E2Eh	ATCC3PPS	—	—	—			ATCC3PPS<4:0>			---1 0101	---1 0101
E2Fh 至 E6Fh	—	未实现								—	—

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

注 1： 仅限PIC16F1615/9。

2： 未实现，读为1。

3： 仅限PIC16(L)F1615。

4： 仅限PIC16(L)F1619。

表 3-14: 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 至 E8Fh	—	未实现								—	—	
E90h	RA0PPS	—	—	—	RA0PPS<4:0>				---	0 0000	---	0 0000
E91h	RA1PPS	—	—	—	RA1PPS<4:0>				---	0 0000	---	0 0000
E92h	RA2PPS	—	—	—	RA2PPS<4:0>				---	0 0000	---	0 0000
E93h	—	未实现								—	—	
E94h	RA4PPS	—	—	—	RA4PPS<4:0>				---	0 0000	---	0 0000
E95h	RA5PPS	—	—	—	RA5PPS<4:0>				---	0 0000	---	0 0000
E96h 至 E9Bh	—	未实现								—	—	
E9Ch	RB4PPS <sup>(4)</sup>	—	—	—	RB4PPS<4:0>				---	0 0000	---	0 0000
E9Dh	RB5PPS <sup>(4)</sup>	—	—	—	RB5PPS<4:0>				---	0 0000	---	0 0000
E9Eh	RB6PPS <sup>(4)</sup>	—	—	—	RB6PPS<4:0>				---	0 0000	---	0 0000
E9Fh	RB7PPS <sup>(4)</sup>	—	—	—	RB7PPS<4:0>				---	0 0000	---	0 0000
EA0h	RC0PPS	—	—	—	RC0PPS<4:0>				---	0 0000	---	0 0000
EA1h	RC1PPS	—	—	—	RC1PPS<4:0>				---	0 0000	---	0 0000
EA2h	RC2PPS	—	—	—	RC2PPS<4:0>				---	0 0000	---	0 0000
EA3h	RC3PPS	—	—	—	RC3PPS<4:0>				---	0 0000	---	0 0000
EA4h	RC4PPS	—	—	—	RC4PPS<4:0>				---	0 0000	---	0 0000
EA5h	RC5PPS	—	—	—	RC5PPS<4:0>				---	0 0000	---	0 0000
EA6h	RC6PPS <sup>(4)</sup>	—	—	—	RC6PPS<4:0>				---	0 0000	---	0 0000
EA7h	RC7PPS <sup>(4)</sup>	—	—	—	RC7PPS<4:0>				---	0 0000	---	0 0000
EA8h 至 EEfh	—	未实现								—	—	

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

- 注 1: 仅限 PIC16F1615/9。  
2: 未实现, 读为 1。  
3: 仅限 PIC16(L)F1615。  
4: 仅限 PIC16(L)F1619。

表3-14: 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
<b>Bank 30</b>											
F0Ch 至 F0Eh	—	未实现								—	—
F0Fh	CLCDATA	—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	---- 0000	---- 0000
F10h	CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			0-x0 0000	0-x0 0000
F11h	CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	x--- xxxx	x--- xxxx
F12h	CLC1SEL0	—	—	LC1D1S<5:0>						--xx xxxx	--xx xxxx
F13h	CLC1SEL1	—	—	LC1D2S<5:0>						--xx xxxx	--xx xxxx
F14h	CLC1SEL2	—	—	LC1D3S<5:0>						--xx xxxx	--xx xxxx
F15h	CLC1SEL3	—	—	LC1D4S<5:0>						--xx xxxx	--xx xxxx
F16h	CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	xxxx xxxx	xxxx xxxx
F17h	CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	xxxx xxxx	xxxx xxxx
F18h	CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	xxxx xxxx	xxxx xxxx
F19h	CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	xxxx xxxx	xxxx xxxx
F1Ah	CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			0-x0 0000	0-x0 0000
F1Bh	CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	x--- xxxx	x--- xxxx
F1Ch	CLC2SEL0	—	—	LC2D1S<5:0>						--xx xxxx	--xx xxxx
F1Dh	CLC2SEL1	—	—	LC2D2S<5:0>						--xx xxxx	--xx xxxx
F1Eh	CLC2SEL2	—	—	LC2D3S<5:0>						--xx xxxx	--xx xxxx
F1Fh	CLC2SEL3	—	—	LC2D4S<5:0>						--xx xxxx	--xx xxxx
F20h	CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	xxxx xxxx	xxxx xxxx
F21h	CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	xxxx xxxx	xxxx xxxx
F22h	CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	xxxx xxxx	xxxx xxxx
F23h	CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	xxxx xxxx	xxxx xxxx

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

- 注 1: 仅限PIC16F1615/9。  
 2: 未实现, 读为1。  
 3: 仅限PIC16(L)F1615。  
 4: 仅限PIC16(L)F1619。

表3-14: 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
<b>Bank 30 (续)</b>											
F24h	CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			0-x0 0000	0-x0 0000
F25h	CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	x--- xxxx	x--- xxxx
F26h	CLC3SEL0	—	—	LC3D1S<5:0>						--xx xxxx	--xx xxxx
F27h	CLC3SEL1	—	—	LC3D2S<5:0>						--xx xxxx	--xx xxxx
F28h	CLC3SEL2	—	—	LC3D3S<5:0>						--xx xxxx	--xx xxxx
F29h	CLC3SEL3	—	—	LC3D4S<5:0>						--xx xxxx	--xx xxxx
F2Ah	CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	xxxx xxxx	xxxx xxxx
F2Bh	CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	xxxx xxxx	xxxx xxxx
F2Ch	CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	xxxx xxxx	xxxx xxxx
F2Dh	CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	xxxx xxxx	xxxx xxxx
F2Eh	CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			0-x0 0000	0-x0 0000
F2Fh	CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	x--- xxxx	x--- xxxx
F30h	CLC4SEL0	—	—	LC4D1S<5:0>						--xx xxxx	--xx xxxx
F31h	CLC4SEL1	—	—	LC4D2S<5:0>						--xx xxxx	--xx xxxx
F32h	CLC4SEL2	—	—	LC4D3S<5:0>						--xx xxxx	--xx xxxx
F33h	CLC4SEL3	—	—	LC4D4S<5:0>						--xx xxxx	--xx xxxx
F34h	CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	xxxx xxxx	xxxx xxxx
F35h	CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	xxxx xxxx	xxxx xxxx
F36h	CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	xxxx xxxx	xxxx xxxx
F37h	CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	xxxx xxxx	xxxx xxxx
F38h 至 F6Fh	—	未实现								—	—

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

- 注 1: 仅限PIC16F1615/9。  
 2: 未实现, 读为1。  
 3: 仅限PIC16(L)F1615。  
 4: 仅限PIC16(L)F1619。

表3-14： 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 — FE3h	—	未实现								—	—
FE4h	STATUS_ SHAD	—	—	—	—	—	Z_SHAD	DC_SHAD	C_SHAD	---- -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
FEEd	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，r = 保留。阴影单元未实现，读为0。

注 1： 仅限PIC16F1615/9。

2： 未实现，读为1。

3： 仅限PIC16(L)F1615。

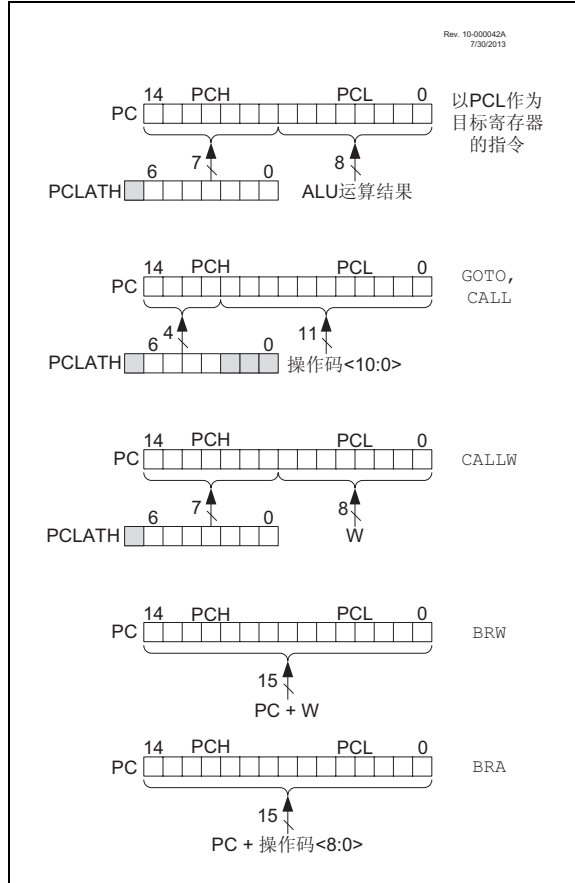
4： 仅限PIC16(L)F1619。



## 3.4 PCL和PCLATH

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

图3-3： 不同情形下PC的装载



### 3.4.1 修改PCL

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

### 3.4.2 计算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）。

### 3.4.3 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。

### 3.4.4 跳转

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

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

如果使用BRA，整个PC中将装入PC+1+ BRA指令操作数的有符号值。

3.5 堆栈

所有器件都具有 16 级 x15 位宽的硬件堆栈（见图 3-4 至图 3-7）。堆栈既不占用程序存储空间，也不占用数据存储空间。当执行 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 1：不存在被称为 PUSH 或 POP 的指令/助记符。堆栈的压入或弹出是源于执行了 CALL、CALLW、RETURN、RETLW 和 RETFIE 指令，或源于跳转到中断向量地址。

3.5.1 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

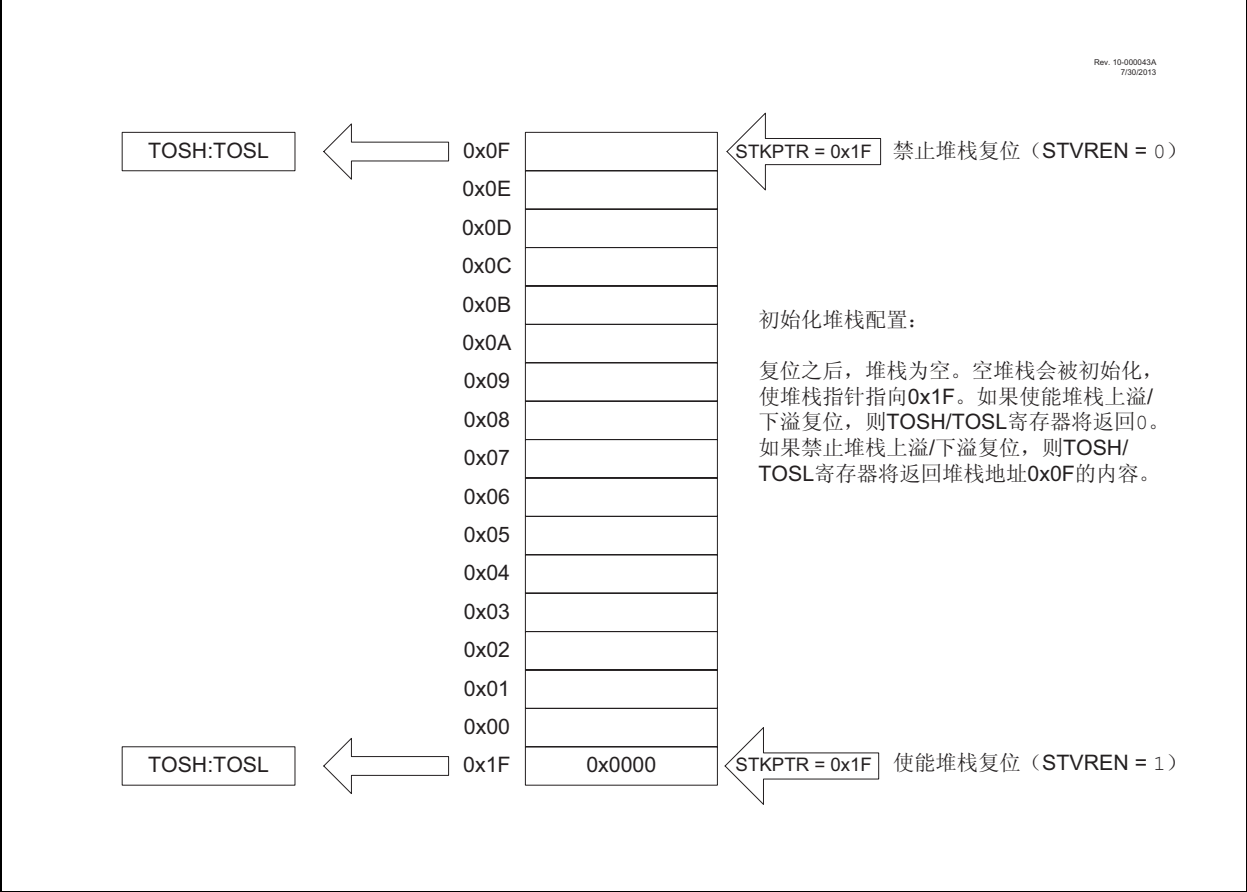


图3-5： 访问堆栈示例2

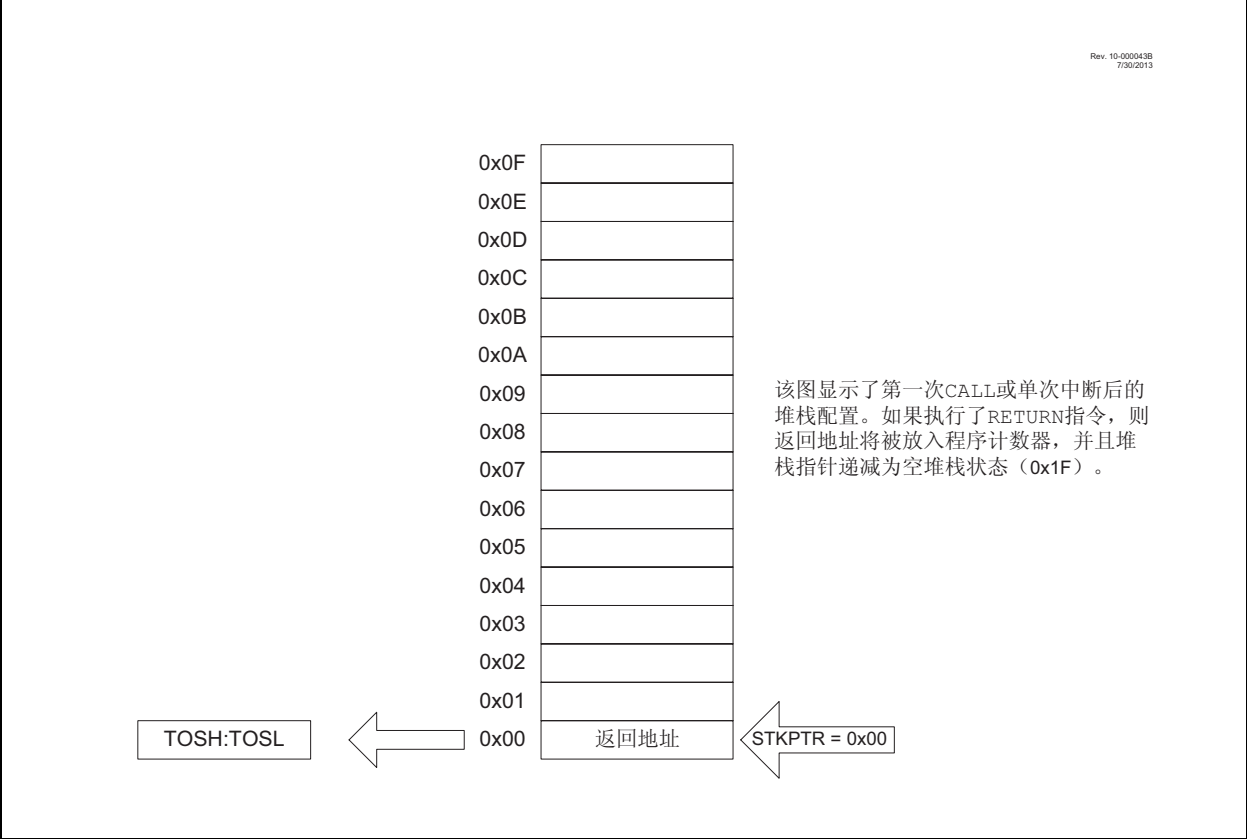


图3-6： 访问堆栈示例3

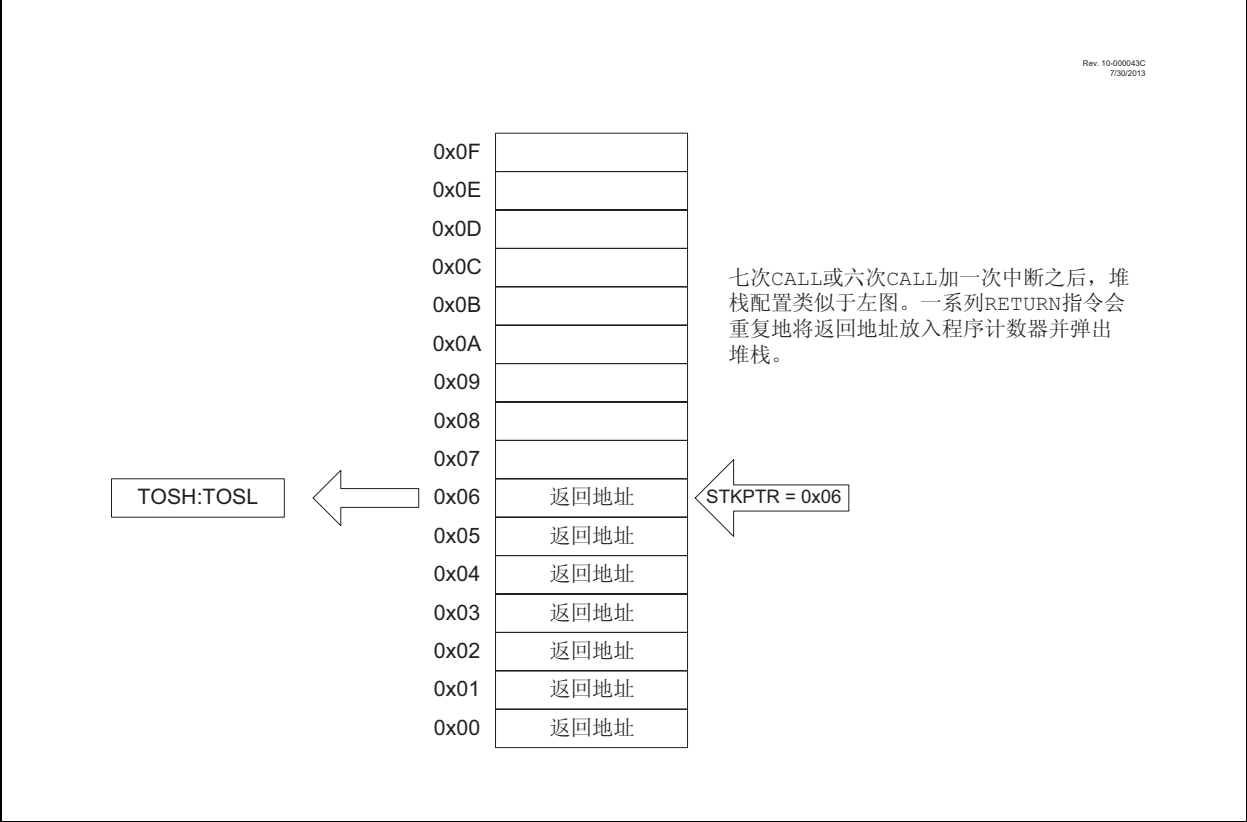
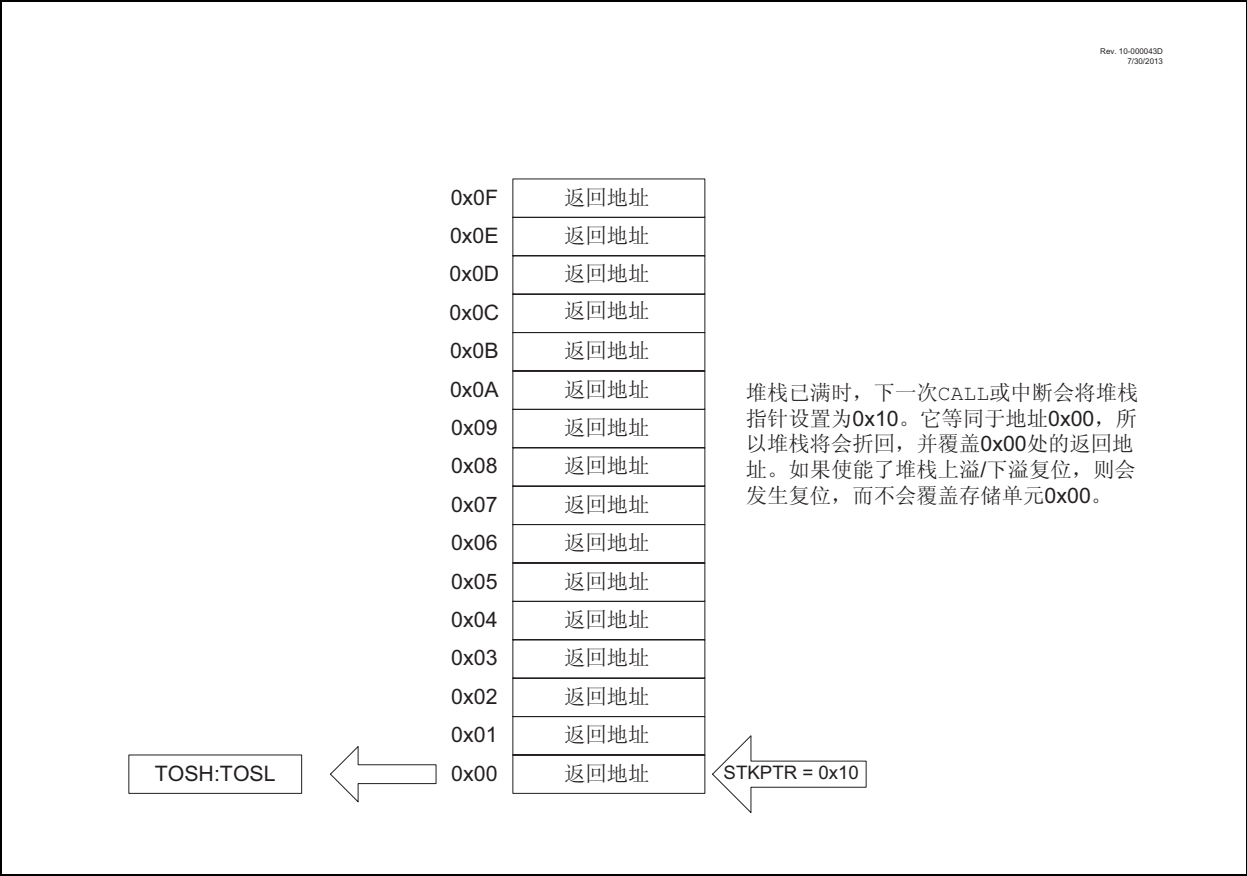


图3-7： 访问堆栈示例4



3.5.2 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。

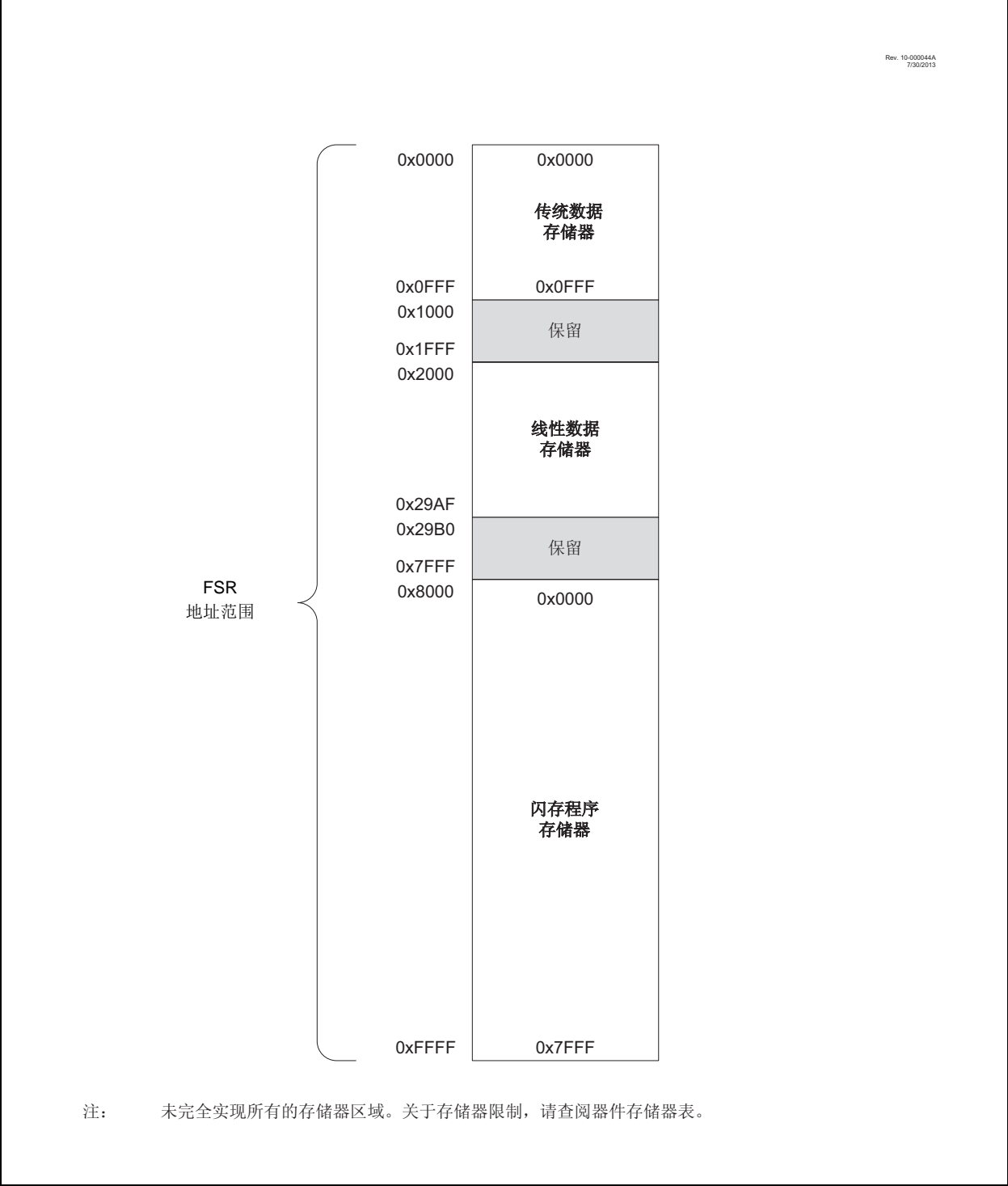
3.6 间接寻址

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

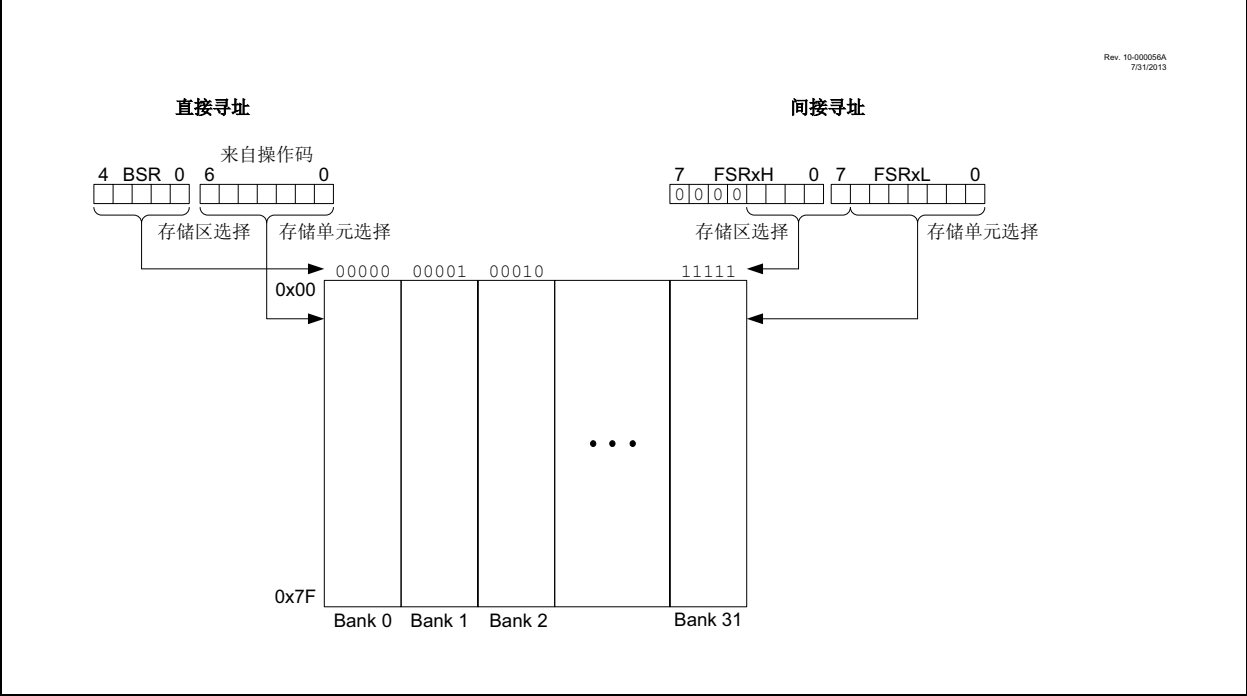


# PIC16(L)F1615/9

## 3.6.1 传统数据存储器

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

图 3-9: 传统数据存储器映射



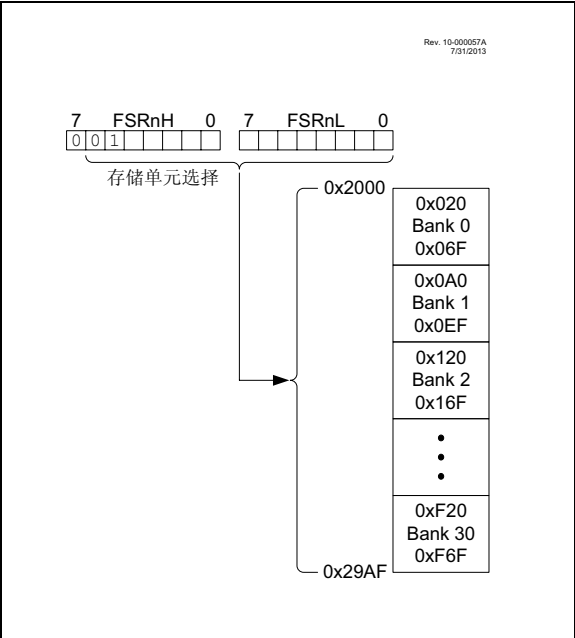
3.6.2 线性数据存储器

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。

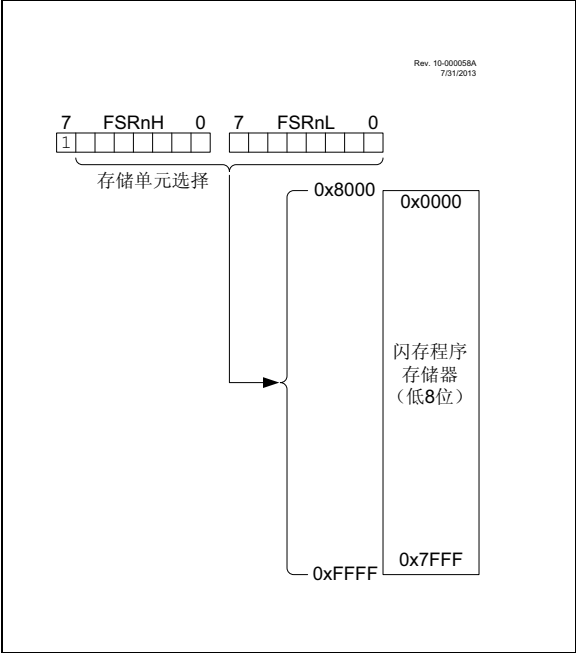
图3-10: 线性数据存储器映射



3.6.3 闪存程序存储器

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

图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，位于8008h的配置字2和位于8009h的配置字3。

<b>注：</b> 配置字中的 <b>DEBUG</b> 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。
--



## 4.2 寄存器定义：配置字

寄存器 4-1: CONFIG1: 配置字 1

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1
FCMEN	IESO	CLKOUTEN	BOREN<1:0> <sup>(1)</sup>	—	
bit 13					bit 8

R/P-1	R/P-1	R/P-1	U-1	U-1	R/P-1	R/P-1	R/P-1
CP <sup>(2)</sup>	MCLR	PWRT	—	—	FOSC<2:0>		
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 **FCMEN:** 故障时钟监视器使能位

1 = 使能故障保护时钟监视器

0 = 禁止故障保护时钟监视器

bit 12 **IESO:** 内部/外部时钟切换位

1 = 使能内/外部时钟切换模式

0 = 禁止内/外部时钟切换模式

bit 11 **CLKOUTEN:** 时钟输出使能位

1 = 禁止 CLKOUT 功能。CLKOUT 引脚为 I/O 功能

0 = 使能 CLKOUT 引脚上的 CLKOUT 功能

bit 10-9 **BOREN<1:0>:** 欠压复位使能位<sup>(1)</sup>

11 = 使能 BOR

10 = 在工作期间使能 BOR，在休眠期间禁止 BOR

01 = BOR 由 BORCON 寄存器的 SBOREN 位控制

00 = 禁止 BOR

bit 8 **未实现:** 读为 1

bit 7 **CP:** 代码保护位<sup>(2)</sup>

1 = 禁止程序存储器代码保护

0 = 使能程序存储器代码保护

bit 6 **MCLR:** MCLR/VPP 引脚功能选择位

如果 LVP 位 = 1:

该位被忽略。

如果 LVP 位 = 0:

1 = MCLR/VPP 引脚功能为 MCLR；使能弱上拉。

0 = MCLR/VPP 引脚功能为数字输入；MCLR 在内部被禁止；弱上拉由 WPUA3 位控制。

bit 5 **PWRT:** 上电延时定时器使能位

1 = 禁止 PWRT

0 = 使能 PWRT

bit 4-3 **未实现:** 读为 1

# PIC16(L)F1615/9

---

## 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: 在 CLKIN 引脚上

110 = ECM: 外部时钟, 中等功耗模式: 在 CLKIN 引脚上

101 = ECL: 外部时钟, 低功耗模式: 在 CLKIN 引脚上

100 = INTOSC 振荡器: CLKIN 引脚为 I/O 功能

011 = 保留

010 = HS: HS 振荡器, 高速晶振/谐振器连接在 OSC1 和 OSC2 引脚之间

001 = 保留

000 = 保留

**注 1:** 使能欠压复位时并不会自动使能上电延时定时器。

**2:** 使能代码保护之后, 只能通过批量擦除器件来禁止它。

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 <sup>(1)</sup>	DEBUG <sup>(3)</sup>	LPBOR	BORV <sup>(2)</sup>	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<sup>(1)</sup>

1 = 使能低电压编程

0 = 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<sup>(3)</sup>

1 = 禁止在线调试器, ICSPCLK 和 ICSPDAT 是通用 I/O 引脚

0 = 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 = 禁止低功耗欠压复位

0 = 使能低功耗欠压复位

bit 10 **BORV:** 欠压复位电压选择位<sup>(2)</sup>

1 = 欠压复位电压 (VBOR), 选择低跳变点

0 = 欠压复位电压 (VBOR), 选择高跳变点

bit 9 **STVREN:** 堆栈上溢/下溢复位使能位

1 = 堆栈上溢或下溢将导致复位

0 = 堆栈上溢或下溢将不会导致复位

bit 8 **PLLEN:** PLL 使能位

1 = 使能 4xPLL

0 = 禁止 4xPLL

bit 7 **ZCD:** ZCD 禁止位

1 = 禁止 ZCD。通过将 ZCD1CON 寄存器的 ZCD1EN 位置 1 可以使能 ZCD

0 = 总是使能 ZCD

bit 6-3 **未实现:** 读为 1

bit 2 **PPS1WAY:** PPSLOCK 一次置 1 使能位

1 = PPSLOCK 位只能在执行解锁序列后置 1 一次; 当 PPSLOCK 置 1 之后, 将阻止对 PPS 寄存器的所有未来更改

0 = PPSLOCK 位可根据需要置 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

8 KW 闪存 (PIC16(L)F1615/9):

11 = OFF——关闭写保护

10 = BOOT——000h 至 01FFh 被写保护, 200h 至 1FFFh 可由 PMCON 控制修改

01 = HALF——000h 至 0FFFh 被写保护, 1000h 至 1FFFh 可由 PMCON 控制修改

00 = ALL——000h 至 1FFFh 被写保护, 没有地址可由 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:** 关于具体跳变点电压, 请参见 VBOR 参数。

**3:** 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)F1615/9

寄存器 4-3: CONFIG3: 配置字 3

	R/P-0	R/P-0	R/P-1	R/P-1	R/P-1	R/P-1
	WDTCCS<2:0>			WDTCWS<2:0>		
bit 13	bit 8					

U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	WDTE<1:0>		WDTCPSS<4:0>				
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 13-11      **WDTCCS<2:0>**: WDT 配置时钟选择位

111 = 软件控制; WDT 时钟通过 CS<2:0> 选择

110 = 保留

•

•

•

010 = 保留

001 = WDT 参考时钟为 MFINTOSC, 31.25 kHz (默认值)

000 = WDT 参考时钟为 LFINTOSC, 31.00 kHz 输出

bit 10-8      **WDTCWS<2:0>**: WDT 配置窗口选择位

WDTCWS<2:0>	POR 时的窗口			软件控制窗口?	需要密钥访问?
	值	窗口延时的时间百分比	窗口打开的时间百分比		
111	111	n/a	100	是	否
110	111	n/a	100	否	是
101	101	25	75		
100	100	37.5	62.5		
011	011	50	50		
010	010	62.5	37.5		
001	001	75	25		
000	000	87.5	12.5 <sup>(1)</sup>		

默认熔丝值 = 111

bit 7      **未实现**: 读为 1

bit 6-5      **WDTE<1:0>**: 看门狗定时器使能位

11 = 在所有模式下使能 WDT, 忽略 WDTCON0 寄存器中的 SEN 位

10 = 在运行时使能 WDT, 在休眠时禁止 WDT

01 = WDT 由 WDTCON0 寄存器的 SEN 位控制

00 = 禁止 WDT

寄存器 4-3:           **CONFIG3: 配置字 3 (续)**  
bit 4-0           **WDTCP5<4:0>**: WDT 配置周期选择位

WDTCP5 <4:0>	POR 时的 WDTPS				软件控制 WDTPS
	值	分频比		典型超时 (F <sub>IN</sub> = 31 kHz)	
11111	01011	1:65536	2 <sup>16</sup>	2s	是
10011 ... 11110	10011 ... 11110	1:32	2 <sup>5</sup>	1 ms	否
10010	10010	1:8388608	2 <sup>23</sup>	256s	否
10001	10001	1:4194304	2 <sup>22</sup>	128s	
10000	10000	1:2097152	2 <sup>21</sup>	64s	
01111	01111	1:1048576	2 <sup>20</sup>	32s	
01110	01110	1:524299	2 <sup>19</sup>	16s	
01101	01101	1:262144	2 <sup>18</sup>	8s	
01100	01100	1:131072	2 <sup>17</sup>	4s	
01011	01011	1:65536	2 <sup>16</sup>	2s	
01010	01010	1:32768	2 <sup>15</sup>	1s	
01001	01001	1:16384	2 <sup>14</sup>	512 ms	
01000	01000	1:8192	2 <sup>13</sup>	256 ms	
00111	00111	1:4096	2 <sup>12</sup>	128 ms	
00110	00110	1:2048	2 <sup>11</sup>	64 ms	
00101	00101	1:1024	2 <sup>10</sup>	32 ms	
00100	00100	1:512	2 <sup>9</sup>	16 ms	
00011	00011	1:256	2 <sup>8</sup>	8 ms	
00010	00010	1:128	2 <sup>7</sup>	4 ms	
00001	00001	1:64	2 <sup>6</sup>	2 ms	
00000	00000	1:32	2 <sup>5</sup>	1 ms	

默认熔丝值 = 11111

注 1: 12.5% 的窗口延时仅在软件控制模式下通过 WDTCON1 寄存器使用。

## 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。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 $\overline{\text{CP}}$ 位来防止外部读写操作。当 $\overline{\text{CP}} = 0$ 时，将禁止对程序存储器的外部读写操作，读取时将返回全0。无论保护位的设置如何，CPU都可以继续读取程序存储器。对程序存储器的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“*PIC12(L)F1612/16(L)F161X Memory Programming Specification*”（DS40001720）。



## 5.0 振荡器模块

### 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，同时最大程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系统时钟源。
- 双速启动模式，最大程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（HS、ECH、ECM或ECL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. INTOSC——内部振荡器（31 kHz至32 MHz）
5. HS——高增益晶振或陶瓷谐振器模式（4 MHz至10 MHz）

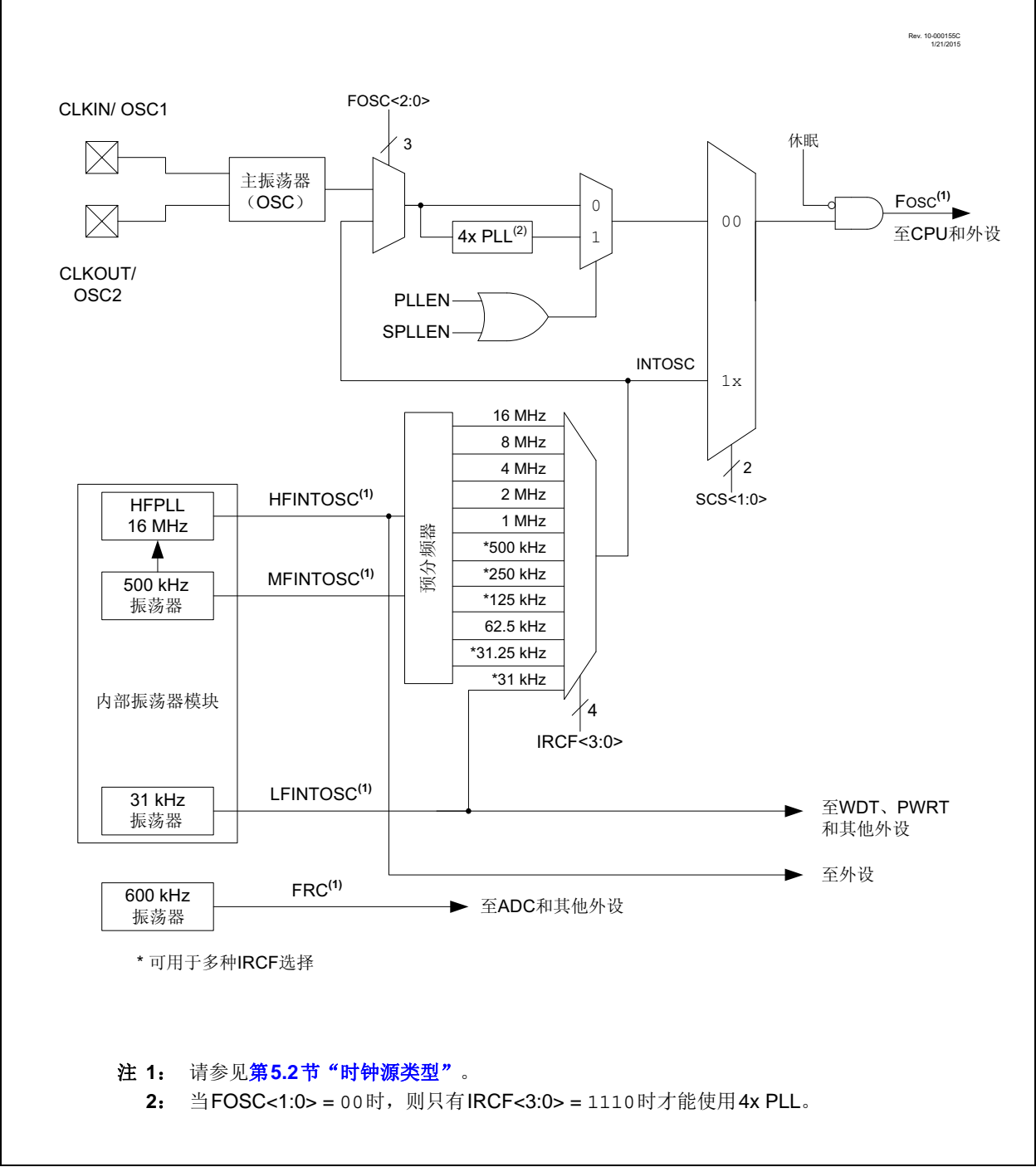
时钟源模式通过配置字中的FOSC<1:0>位进行选择。FOSC位决定在器件初次上电时使用的振荡器类型。

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。HS时钟模式要求器件在外部连接一个晶振或谐振器。

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。



图 5-1: PIC® MCU时钟源的简化框图



## 5.2 时钟源类型

时钟源可分为外部和内部模式。

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

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<1:0>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：
  - 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

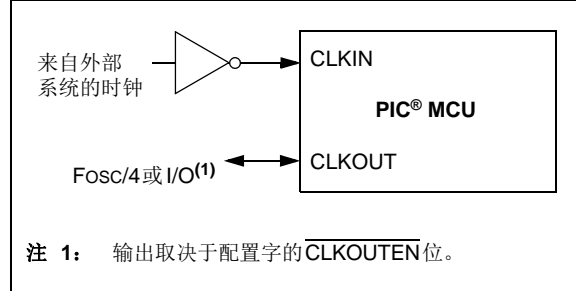
外部时钟（External Clock，EC）模式允许外部产生的逻辑电平信号作为系统时钟源。工作在该模式下时，外部时钟源连接到CLKIN输入。CLKOUT可用作通用I/O或CLKOUT。图5-2给出了EC模式的引脚连接图。

EC模式具有三种功耗模式，可通过配置字中的Fosc位进行选择：

- ECH——高功耗，4-20 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限制器件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 HS模式

HS模式支持使用连接到OSC1和OSC2的石英晶振或陶瓷谐振器。图5-3和图5-4分别给出了石英晶振和陶瓷谐振器的典型电路。

图5-3： 石英晶振的工作原理（HS模式）

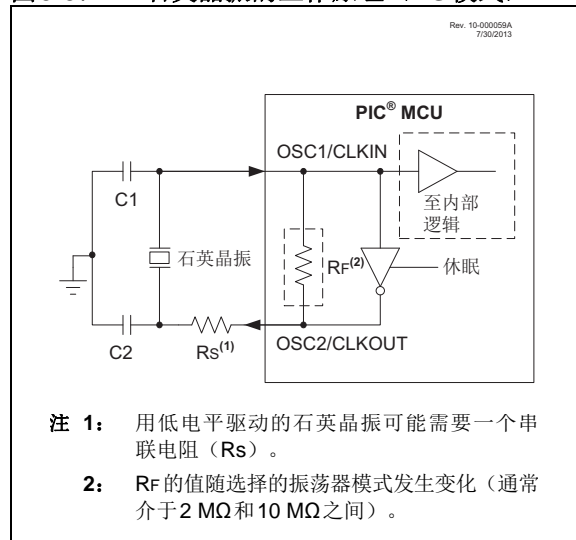
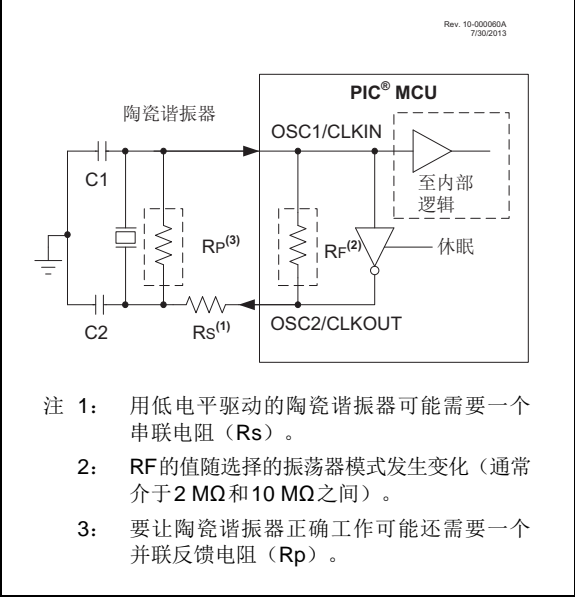


图 5-4： 陶瓷谐振器的工作原理（HS 模式）



5.2.1.3 振荡器起振定时器（OST）

如果振荡器模块被配置为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。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”）。

## 5.2.2 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<1:0> 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，CLKIN 可用作通用 I/O。CLKOUT 可用作通用 I/O 或 CLKOUT。

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

内部振荡器模块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 5.2.2.8 节“内部振荡器时钟切换时序”。

发生以下情况时，HFINTOSC 被使能：

- 根据所需的 HF 频率配置 OSCCON 寄存器的 IRCF<3:0> 位，并且
- FOSC<1:0> = 00，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 5.2.2.8 节“内部振荡器时钟切换时序”。

发生以下情况时，MFINTOSC 被使能：

- 根据所需的 MF 频率配置 OSCCON 寄存器的 IRCF<3:0> 位，并且
- FOSC<1:0> = 00，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。该值为 6 位的二进制补码。值为 1Fh 时，将调整为最高频率。值为 20h 时，将调整为最低频率。

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

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）以及外设等，其工作不受频率变化的影响。

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 5.2.2.8 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<1:0> = 00，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 FRC

FRC 时钟是未经校准的标称频率为 600 kHz 的外设时钟源。

FRC 由请求 FRC 时钟的外设自动开启。

在休眠期间，FRC 时钟将继续运行。

5.2.2.6 内部振荡器频率选择

使用 OSCCON 寄存器的内部振荡器频率选择位 IRCF<3:0>，可通过软件选择系统时钟速度。

16 MHz HFINTOSC、500 kHz MFINTOSC 和 31 kHz LFINTOSC 输出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 IRCF<3:0> 用于选择内部振荡器的频率输出。可通过软件选择以下频率中的一个：

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

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

## 5.2.2.7 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。可以使用8 MHz或16 MHz内部振荡器设置，16 MHz在输入PLL之前进行2分频。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（FOSC<1:0>=00），以使用INTOSC源作为器件系统时钟。
- OSCCON寄存器中的SCS位必须清零（SCS<1:0>=00），以使用由配置字中FOSC<1:0>决定的时钟。
- OSCCON寄存器中的IRCF位必须设置为16 MHz（IRCF<3:0>=1111）或8 MHz HFINTOSC（IRCF<3:0>=1110）。
- OSCCON寄存器中的SPLLEN位必须置1，以使能4x PLL，或者配置字的PLLEN位必须编程为1。

**注：** 使用配置字的PLLEN位时，将无法用软件禁止4x PLL，并且8/16 MHz HFINTOSC选项将不再可用。

当OSCCON寄存器的SCS位设置为1x时，4x PLL不可用于内部振荡器。要将4x PLL用于内部振荡器，必须将SCS位设置为00。

## 5.2.2.8 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5-5）。如果是这种情况，则在修改OSCCON寄存器的IRCF<3:0>位之后，进行频率选择之前，存在一定的延时。OSCSTAT寄存器将反映HFINTOSC、MFINTOSC和LFINTOSC振荡器的当前工作状态。频率选择序列如下：

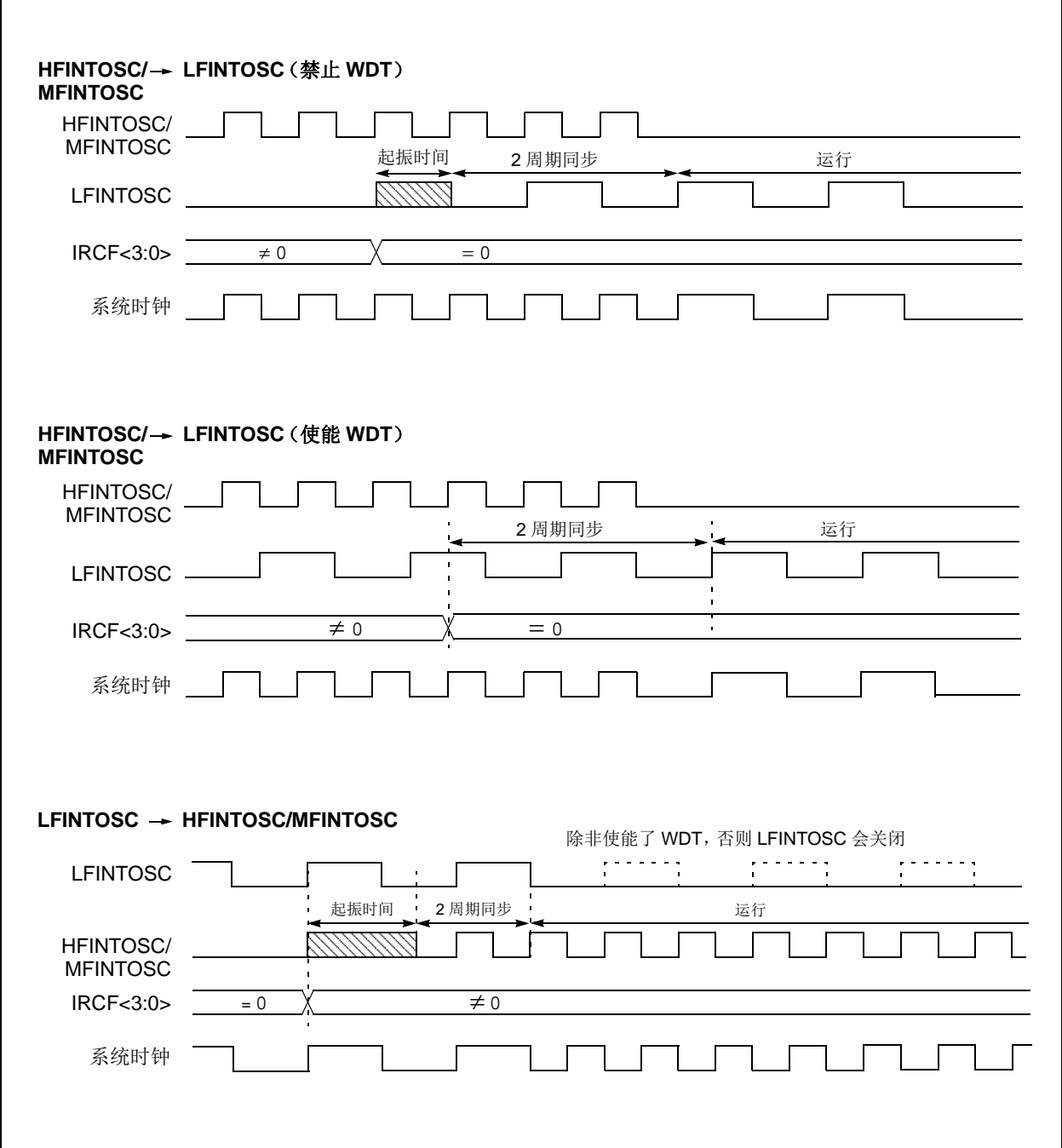
1. 修改OSCCON寄存器的IRCF<3:0>位。
2. 如果新时钟是关闭的，开始时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿出现。
4. 当前时钟保持为低电平，时钟切换电路等待新时钟上升沿出现。
5. 新时钟现在开始工作。
6. OSCSTAT寄存器按需要进行更新。
7. 时钟切换完成。

更多详细信息，请参见图5-5。

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

起振延时规范请参见第35.0节“电气规范”中的振荡器表。

图5-5： 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
- 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的FOSC<1:0>位的值决定。
- 当OSCCON寄存器的SCS位 = 1x时，系统时钟源由通过OSCCON寄存器的IRCF<3:0>位选择的内部振荡器频率选择。复位之后，OSCCON寄存器的SCS位总是被清零。

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。

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 <sup>(1)</sup> MFINTOSC <sup>(1)</sup> HFINTOSC <sup>(1)</sup>	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	振荡器预热延时（TWARM）
休眠/POR	EC <sup>(1)</sup>	DC – 32 MHz	2个周期
LFINTOSC	EC <sup>(1)</sup>	DC – 32 MHz	每次一周期
任何时钟源	MFINTOSC <sup>(1)</sup> HFINTOSC <sup>(1)</sup>	31.25 kHz-500 kHz 31.25 kHz-16 MHz	2 μs（近似值）
任何时钟源	LFINTOSC <sup>(1)</sup>	31 kHz	每次一周期
PLL无效	PLL有效	16-32 MHz	2 ms（近似值）

注 1： PLL无效。



5.4 双速时钟启动模式

双速启动模式通过最大程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，然后再返回休眠状态而无需等待外部振荡器稳定。

当振荡器模块被配置为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

通过以下设置来配置双速启动模式：

- IESO（在配置字中）= 1；内部/外部时钟切换位（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。

**注：** 使能FSCM时，将自动使能双速启动。

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（HS模式）。
7. 系统时钟切换到外部时钟源。

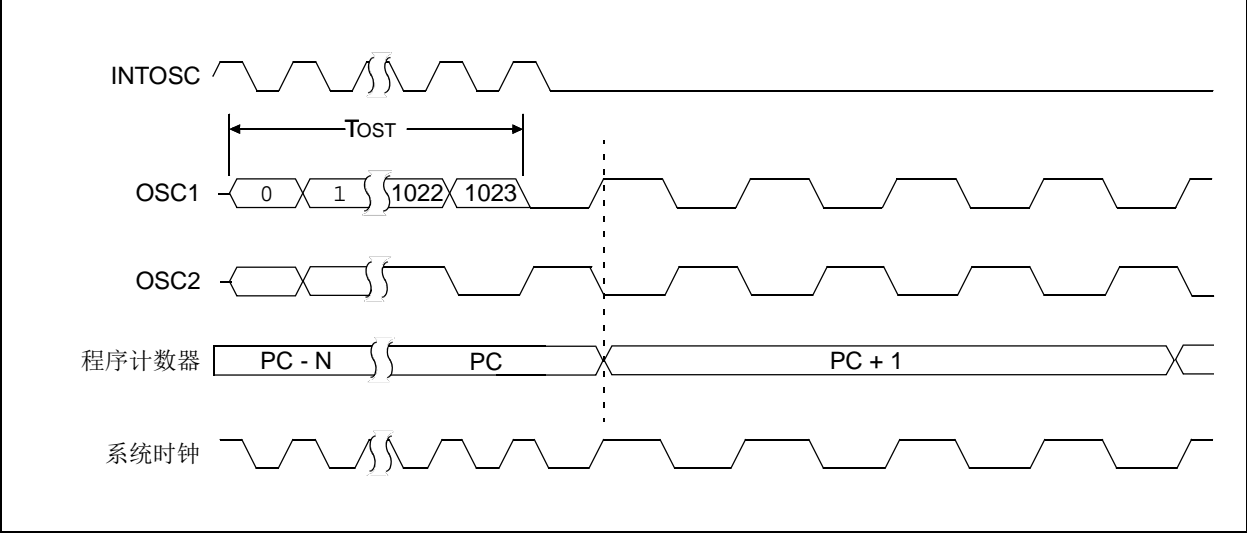
5.4.3 检查双速时钟状态

通过检查OSCSTAT寄存器的OSTS位的状态，可以确认CPU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。请参见表5-2。

表5-2: 振荡器切换延时

切换自	切换到	振荡器延时
任何时钟源	LFINTOSC	每个时钟源1个周期
	HFINTOSC	2 μs（近似值）
	ECH、ECM、ECL或EXTRC	2个周期
	HS	1024个时钟周期（OST）
	辅助振荡器	1024个辅助振荡器周期

图 5-6: 双速启动

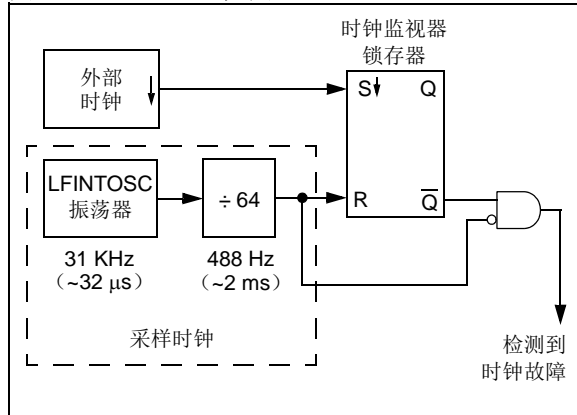


## 5.5 故障保护时钟监视器

故障保护时钟监视器（FSCM）使得器件在出现外部振荡器或外部时钟故障时仍能继续工作。如果选择了振荡器模式，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。当选择了外部时钟模式时，FSCM能在器件从复位释放后立即检测振荡器故障。

FSCM通过将配置字中的FCMEN位置1来使能。FSCM可用于外部振荡器模式（HS）、外部时钟模式（ECH、ECM、ECL和EXTRC）和辅助振荡器（SOSC）。

图5-7: FSCM框图



### 5.5.1 故障保护检测

FSCM模块通过监视时钟下降沿并将LFINTOSC用作时基来检测振荡器故障。请参见图5-7。检测振荡器故障将需要32至96个LFINTOSC周期。图5-8显示了FSCM模块的时序图。

### 5.5.2 故障保护操作

当外部时钟出现故障时，FSCM将CPU时钟切换到内部时钟源，并将PIR2寄存器的OSFIF位置1。内部时钟源由OSCCON寄存器的IRCF<3:0>位决定。

当OSFIF位置1时，如果PIE2寄存器的OSFIE位也使能，将产生中断。中断服务程序（Interrupt Service Routine, ISR）中的用户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，直到故障保护条件被清除，请参见第5.5.3节“故障保护条件清除”。

### 5.5.3 故障保护条件清除

当存在故障保护条件时，用户必须在使用外部时钟源返回正常工作前执行以下操作清除条件。

接下来的章节将介绍在指定的时钟选择（FOSC位）和时钟切换模式（SCS位设置）下，如何清除故障保护条件。

#### 5.5.3.1 外部振荡器（SCS<1:0> = 00）

当使用FOSC位选择外部振荡器（FOSC<2:0> = HS）且时钟切换选择为依靠FOSC选择（SCS<1:0> = 00）运行时发生了故障保护条件，则通过执行下列步骤清除该条件。

当SCS<1:0> = 00时（根据FOSC选择运行）

SCS<1:0> = 1x:

修改OSCCON寄存器中SCS位，以选择内部振荡器模块。这会复位OST定时器并允许其重新工作。

OSFIF = 0:

将PIR2寄存器的OSFIF位清零。

SCS<1:0> = 00:

修改OSCCON寄存器的SCS位，以选择FOSC配置字时钟选择。这会启动OST。CPU将继续依靠内部振荡器工作，直到达到OST计数。当OST超时后，时钟模块将切换到外部振荡器且故障保护条件将被清除。

如果故障保护条件仍然存在，硬件会再次将OSFIF位置1。

#### 5.5.3.2 外部时钟（SCS<1:0> = 00）

当使用FOSC位选择外部时钟（FOSC<2:0> = ECH、ECM、ECL或EXTRC）且时钟切换选择为依靠FOSC选择（SCS<1:0> = 00）运行时发生了故障保护条件，则通过执行下列步骤清除该条件。

当SCS<1:0> = 00时（根据FOSC选择运行）

SCS<1:0> = 1x:

修改OSCCON寄存器中SCS位，以选择内部振荡器模块。这会复位OST定时器并允许其重新工作。

OSFIF = 0:

将PIR2寄存器的OSFIF位清零。

SCS<1:0> = 00:

更改OSCCON寄存器的SCS位，以选择FOSC配置字时钟选择。由于OST不适用于外部时钟，时钟模块将立即切换到外部时钟且故障保护条件将被清除。

如果故障保护条件仍然存在，硬件会再次将OSFIF位置1。

5.5.4 复位或从休眠中唤醒

FSCM设计为用于检测外部振荡器或外部时钟故障。

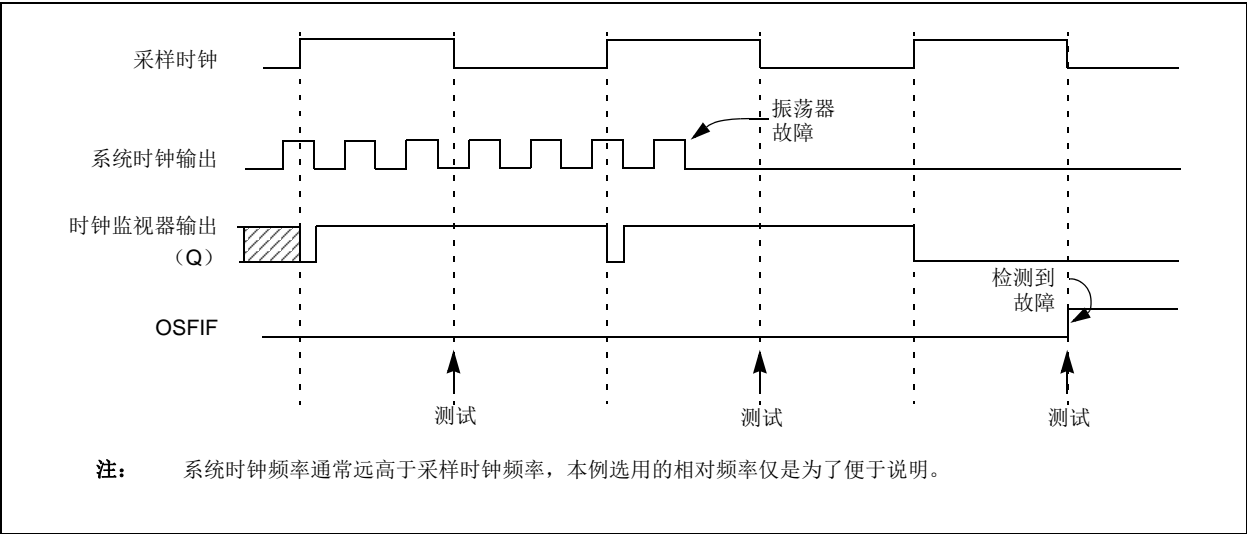
当FSCM使用外部振荡器时，振荡器起振定时器（OST）计数必须在FSCM处于活动状态前超时。从休眠状态唤醒后以及任何类型的复位后使用OST。

当FSCM使用外部时钟时，OST不使用，所以一旦复位或唤醒完成，FSCM就处于活动状态。

当FSCM被使能时，双速启动也被使能。因此，当OST运行时，器件总是处于代码执行阶段。

**注：** 由于振荡器起振时间范围较大，在振荡器起振期间（即，从复位或休眠中退出后），故障保护电路不处于活动状态。

图5-8: FSCM时序图



## 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 **PLLEN** = 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 **PLLEN** = 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 HF

1101 = 4 MHz HF

1100 = 2 MHz HF

1011 = 1 MHz HF

1010 = 500 kHz HF<sup>(1)</sup>

1001 = 250 kHz HF<sup>(1)</sup>

1000 = 125 kHz HF<sup>(1)</sup>

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<sup>(1)</sup>

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 = 保留 (默认设为内部振荡器模块)

00 = 由配置字中 **FOSC<1:0>** 决定的时钟。

**注 1:** 基于 HFINTOSC 产生的重复频率。

# PIC16(L)F1615/9

## 寄存器 5-2: OSCSTAT: 振荡器状态寄存器

U-0	R-0/q	U-0	R-0/q	R-0/q	R-q/q	R-0/q	R-0/q
—	PLL R	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

### 图注:

R = 可读位

u = 不变

1 = 置 1

W = 可写位

x = 未知

0 = 清零

U = 未实现位, 读为 0

-n/n = POR 和 BOR 时的值/所有其他复位时的值

q = 条件值

bit 7	<b>未实现:</b> 读为 0
bit 6	<b>PLL R:</b> 4x PLL 就绪位 1 = 4x PLL 就绪 0 = 4x PLL 未就绪
bit 5	<b>OSTS:</b> 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	<b>HFIOFR:</b> 高频内部振荡器就绪位 1 = HFINTOSC 就绪 0 = HFINTOSC 未就绪
bit 3	<b>HFIOFL:</b> 高频内部振荡器锁定位 1 = HFINTOSC 的精度至少在 2% 以内 0 = HFINTOSC 的精度在 2% 以外
bit 2	<b>MFIOFR:</b> 中频内部振荡器就绪位 1 = MFINTOSC 就绪 0 = MFINTOSC 未就绪
bit 1	<b>LFIOFR:</b> 低频内部振荡器就绪位 1 = LFINTOSC 就绪 0 = LFINTOSC 未就绪
bit 0	<b>HFIOFS:</b> 高频内部振荡器稳定位 1 = HFINTOSC 稳定 0 = HFINTOSC 不稳定

**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 5-3: 与时钟源相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		89
OSCSTAT	—	PLL R	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	90
OSCTUNE	—	—	TUN<5:0>						91

图注: — = 未实现位, 读为0。时钟源不使用阴影单元。

**表 5-4: 与时钟源相关的配置字汇总**

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	69
	7:0	CP	MCLRE	PWRTÉ	—	—	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

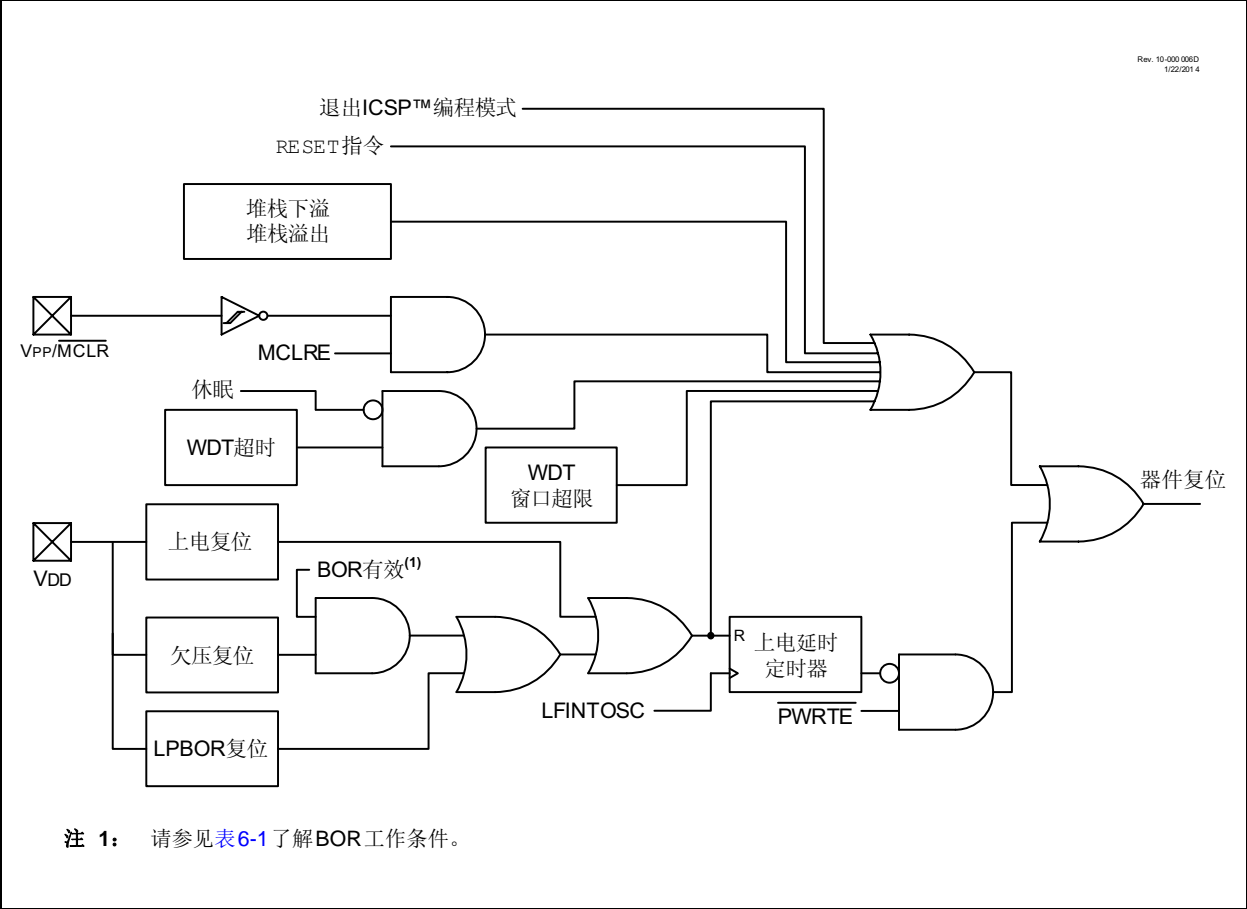
该器件的复位有几种方式：

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

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

图6-1 给出了片上复位电路的简化框图。

图6-1： 片上复位电路的简化框图





6.1 上电复位（POR）

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

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个64 ms标称值的延时。

只要PWRT处于活动状态，器件就保持在复位状态。PWRT延时使VDD有额外的时间上升到所需的电平。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）。

6.2 欠压复位（BOR）

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

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

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

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平。

VDD噪声抑制滤波器可以防止BOR在发生轻微事件时产生触发。如果VDD降至低于VBOR的时间大于参数TBORDC，器件将会发生复位。更多信息，请参见图6-2。

表6-1: BOR工作模式

BOREN<1:0>	SBOREN	器件模式	BOR模式	在以下情况下执行的指令： POR释放或从休眠模式唤醒
11	X	X	有效	等待BOR就绪 <sup>(1)</sup> (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 <sup>(1)</sup> (BORRDY = 1)
	0	X	禁止	立即开始 (BORRDY = x)
00	X	X	禁止	

注 1： 在“POR释放”和“从休眠模式唤醒”的特定情况下，启动时没有任何延时。在CPU准备好执行指令之前，BOR就绪标志会置1（BORRDY = 1），这是因为BOR电路通过BOREN<1:0>位被强制开启。

6.2.1 BOR总是开启

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

BOR保护在休眠期间有效。BOR不会延迟从休眠中唤醒。

6.2.2 BOR在休眠模式下关闭

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

BOR保护在休眠期间无效。器件唤醒会被延迟，直到BOR就绪为止。

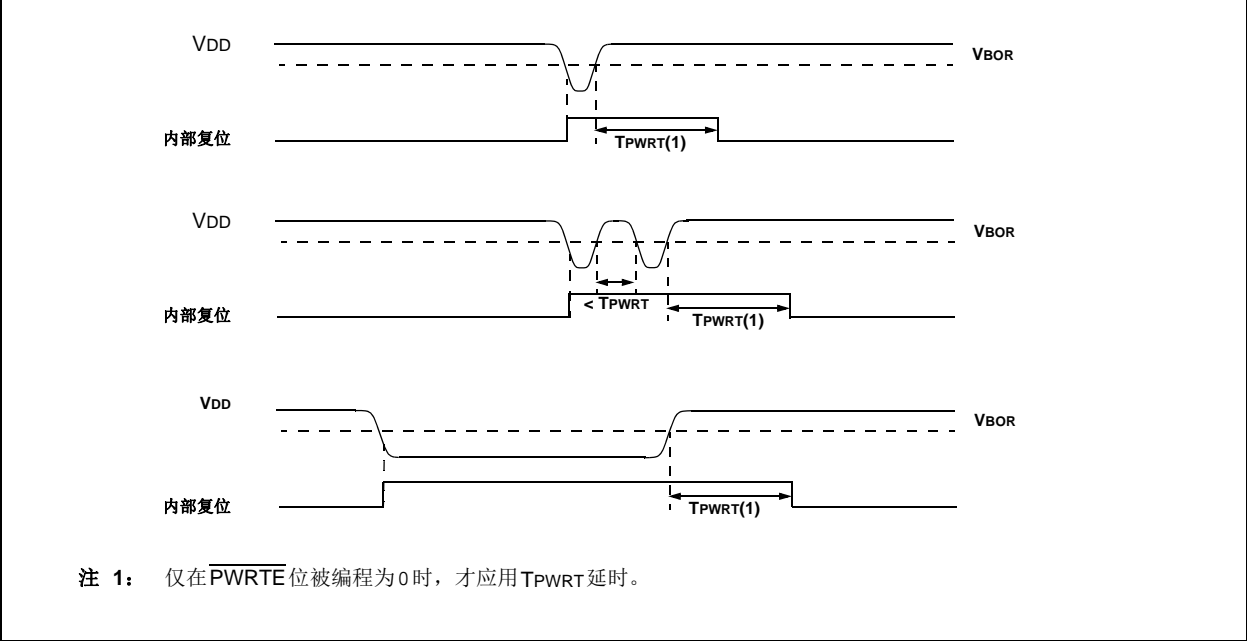
6.2.3 通过软件对BOR进行控制

当配置字的BOREN位编程为01时，BOR将通过BORCON寄存器的SBOREN位进行控制。器件启动不会受BOR就绪条件或VDD电平条件影响而延迟。

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

BOR保护在休眠期间不变。

图6-2: 欠压情形



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	—	—	—	—	—	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:  
1 = 使能BOR  
0 = 禁止BOR  
如果配置字中的BOREN<1:0> ≠ 01:  
SBOREN可读/写, 但对BOR没有任何作用
- bit 6

**BORFS:** 欠压复位快速启动位<sup>(1)</sup>  
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  
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。
- bit 5-1

未实现: 读为0
- bit 0

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

注 1: BOREN<1:0> 位于配置字中。

6.4 低功耗欠压复位 (LPBOR)

低功耗欠压复位 (LPBOR) 功能的工作方式类似于 BOR，即检测 VDD 引脚上的低电压条件。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。PCON 中的 BOR 位同时用于 BOR 和 LPBOR。请参见寄存器 6-2。

LPBOR 电压阈值 (VLPBOR) 的容限宽于 BOR (VBOR)，但工作时需要的电流却低得多 (LPBOR 电流)。LPBOR 旨在在 BOR 配置为禁止 (BOREN = 00) 或在休眠模式下禁止 (BOREN = 10) 时使用。

关于 LPBOR 如何与其他模块进行交互的信息，请参见图 6-1。

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。

6.5 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 12.1 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，且窗口处于打开状态，看门狗定时器会产生复位。STATUS 寄存器中的 TO 和 PD 位会发生改变，指示定时器溢出导致了 WDT 复位；PCON 寄存器中的 WDTWV 位会发生改变，指示窗口违例导致了 WDT 复位。更多信息，请参见第 9.0 节“窗口看门狗定时器 (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 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 3.5.2 节“上溢/下溢复位”。

6.9 编程模式退出

在退出编程模式时，器件的反应与刚刚发生 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 PWRT 位控制。

6.11 启动序列

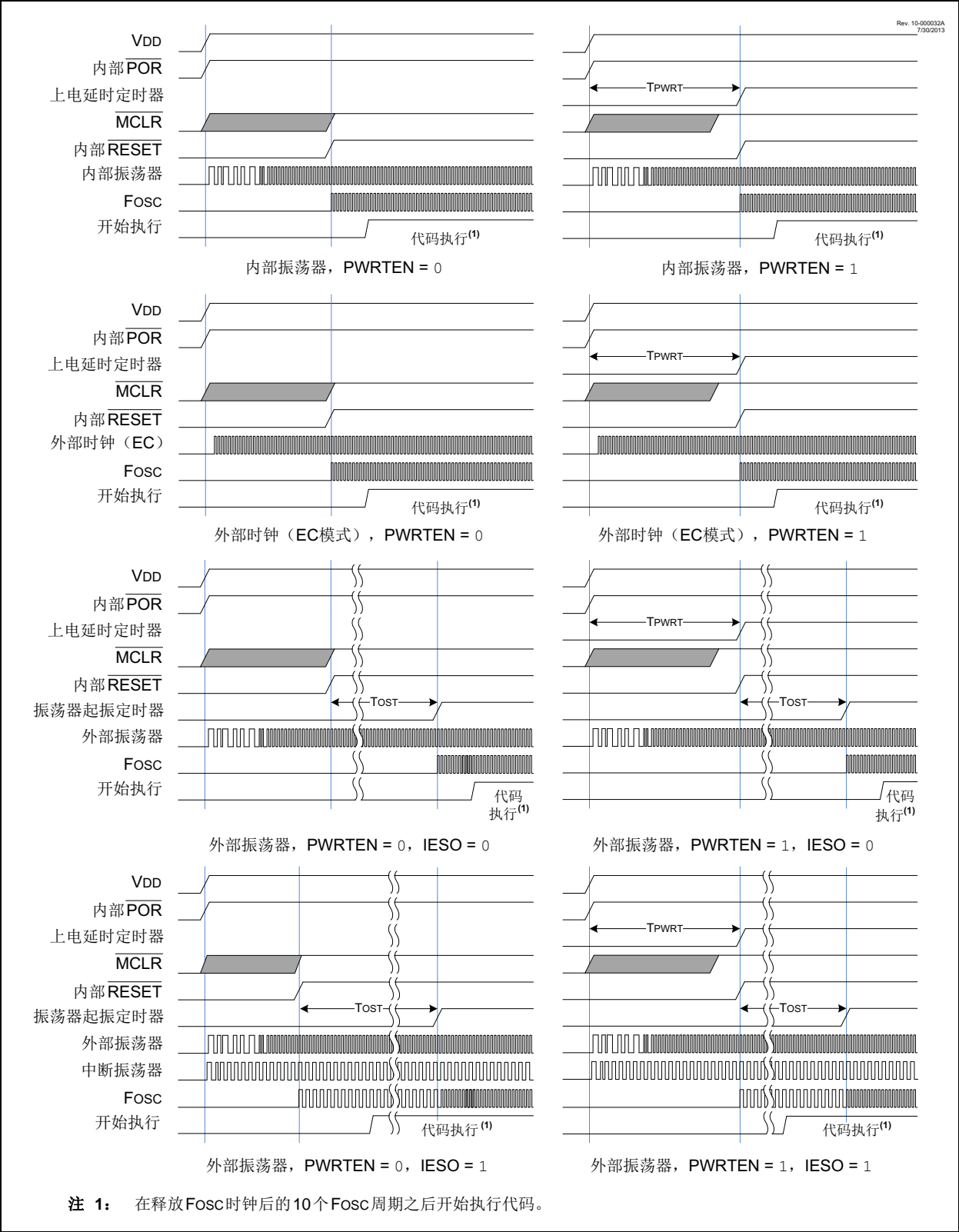
在 POR 或 BOR 释放时，只有先发生以下事件，器件才会开始执行：

- 1. 上电延时定时器运行完毕（如果使能）。
- 2. MCLR 必须被释放（如果使能）。

总延时时间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”。

上电延时定时器的运行与 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寄存器中会有多个位发生更新，以指示复位的原因。表6-3和表6-4列出了这些寄存器的复位条件。

表6-3: 复位状态位及其含义

STKOVF	STKUNF	RWD $\overline{T}$	RMCLR	$\overline{RI}$	$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	条件
0	0	1	1	1	0	x	1	1	上电复位
0	0	1	1	1	0	x	0	x	非法的， $\overline{POR}$ 时 $\overline{TO}$ 被置1
0	0	1	1	1	0	x	x	0	非法的， $\overline{POR}$ 时 $\overline{PD}$ 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从休眠状态唤醒
u	u	u	u	u	u	u	1	0	被中断从休眠状态唤醒
u	u	u	0	u	u	u	u	u	正常工作期间的 $\overline{MCLR}$ 复位
u	u	u	0	u	u	u	1	0	休眠期间的 $\overline{MCLR}$ 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	0011 110x
正常工作期间的 $\overline{MCLR}$ 复位	0000h	---u uuuu	uuuu 0uuu
休眠期间的 $\overline{MCLR}$ 复位	0000h	---1 0uuu	uuuu 0uuu
WDT复位	0000h	---0 uuuu	uuu0 uuuu
被WDT从休眠状态唤醒	PC + 1	---0 0uuu	uuuu uuuu
欠压复位	0000h	---1 1uuu	00uu 11u0
被中断从休眠状态唤醒	PC + 1 <sup>(1)</sup>	---1 0uuu	uuuu uuuu
执行了RESET指令	0000h	---u uuuu	uuuu u0uu
堆栈上溢复位（STVREN = 1）	0000h	---u uuuu	1uuu uuuu
堆栈下溢复位（STVREN = 1）	0000h	---u uuuu	u1uu uuuu
WDT窗口违例	0000h	---1 uuuu	uu0u 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，PC中装入中断向量（0004h）。

6.13 电源控制（PCON）寄存器

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

PCON寄存器位如寄存器6-2所示。

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	R/W/HC-1/q	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	WDTWV	RWD $\overline{\text{T}}$	RMCLR	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

图注：			
HC = 硬件清零位		HS = 硬件置1位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
u = 不变	x = 未知	-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	<b>STKOVF：</b> 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	<b>STKUNF：</b> 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	<b>WDTWV：</b> WDT窗口违例标志位 1 = 未发生WDT窗口违例复位或由固件置1 0 = 发生了WDT窗口违例复位（在未激活窗口的情况下或在窗口范围之外执行了CLRWD $\overline{\text{T}}$ 指令（由硬件清零））
bit 4	<b>RWD<math>\overline{\text{T}}</math>：</b> 看门狗定时器复位标志位 1 = 未发生看门狗定时器复位或由固件置1 0 = 发生了看门狗定时器复位（由硬件清零）
bit 3	<b>RMCLR：</b> MCLR复位标志位 1 = 未发生MCLR复位或由固件置1 0 = 发生了MCLR复位（由硬件清零）
bit 2	<b><math>\overline{\text{RI}}</math>：</b> RESET指令标志位 1 = 未执行RESET指令或由固件置1 0 = 执行了RESET指令（由硬件清零）
bit 1	<b><math>\overline{\text{POR}}</math>：</b> 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	<b><math>\overline{\text{BOR}}</math>：</b> 欠压复位状态位 1 = 未发生欠压复位 0 = 发生了欠压复位（发生上电复位或欠压复位后必须用软件置1）

**表 6-5: 与复位相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	94
PCON	STKOVF	STKUNF	WDTWV	RWD $\overline{T}$	RMCLR	R $\overline{I}$	POR	BOR	98
STATUS	—	—	—	T $\overline{O}$	P $\overline{D}$	Z	DC	C	26
WDTCON0	—	—	WDTPS<4:0>					SEN	124

图注: — = 未实现位, 读为0。复位不使用阴影单元。

注 1: 其他 (非上电) 复位包括在正常工作期间的MCLR复位和看门狗定时器复位。

**表 6-6: 与复位相关的配置字汇总**

名称	Bit	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>		—	69
	7:0	CP	MCLRE	PWRT $\overline{\text{E}}$	—	—	FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	71
	7:0	ZCD	—	—	—	—	PPS1WAY	WRT<1:0>		
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			72
	7:0	—	WDTE<1:0>		WDTGPS<4:0>					

图注: — = 未实现位, 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。

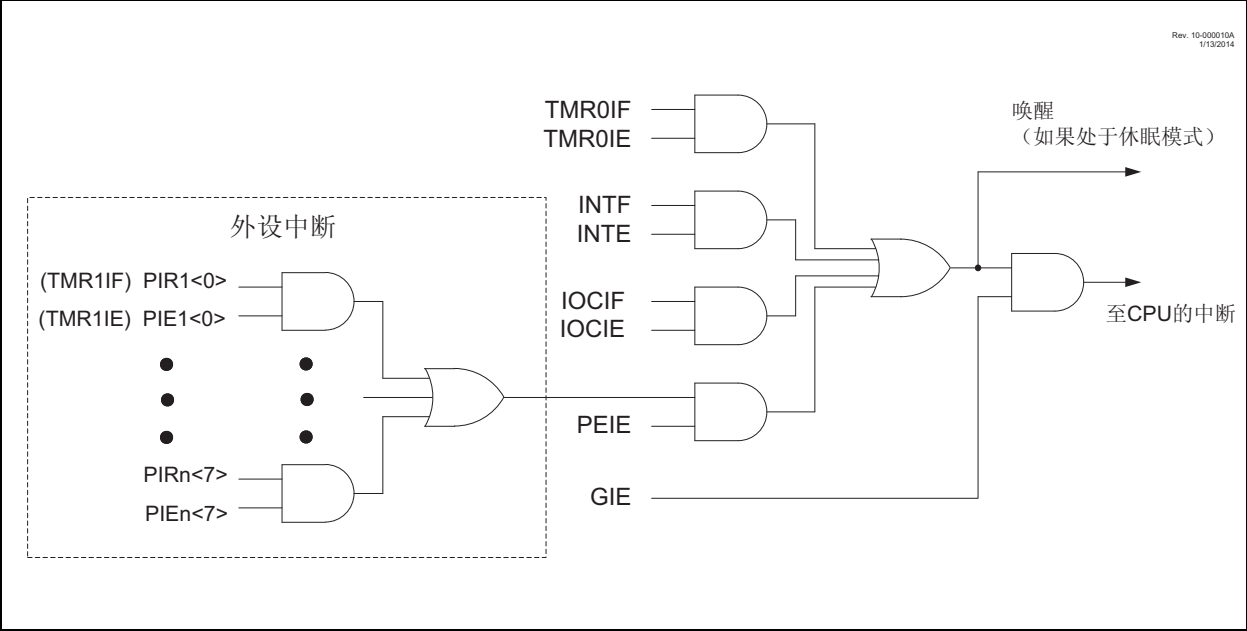
本章包含了关于中断的以下信息：

- 工作原理
- 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 工作原理

任何器件复位时都会禁止中断。通过将以下位置1允许相应中断：

- 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、PIE2和PIE3寄存器中）

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

当中断事件发生时，若GIE位置1，将发生以下事件：

- 清除当前的预取指令
- GIE位清零
- 程序计数器（PC）的当前值压入堆栈
- 自动将关键寄存器保存到影子寄存器中（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（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下来，但不会使处理器重定位到中断向量。

通过从堆栈弹出先前保存的地址、从影子寄存器恢复保存的现场数据并将GIE位置1，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

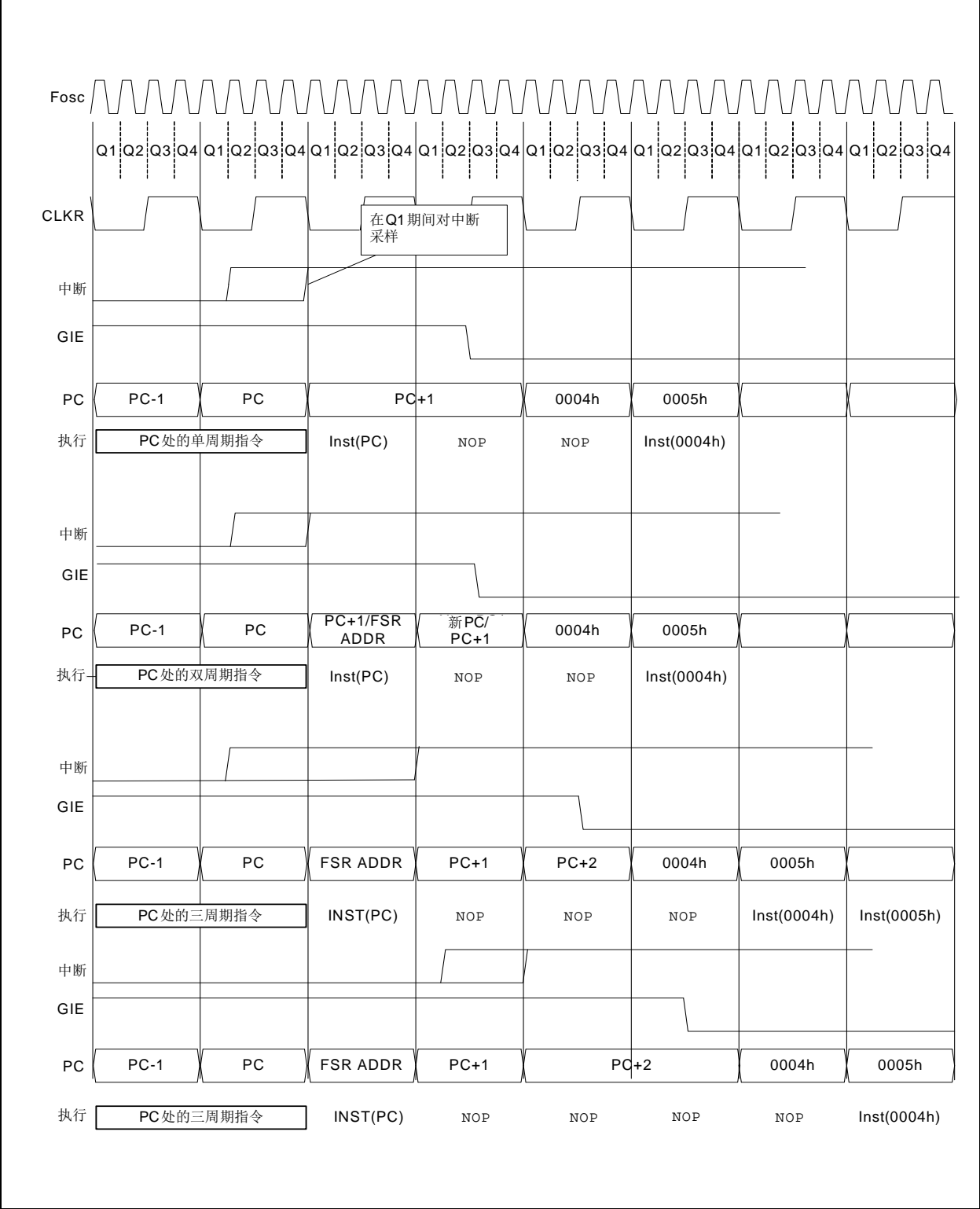
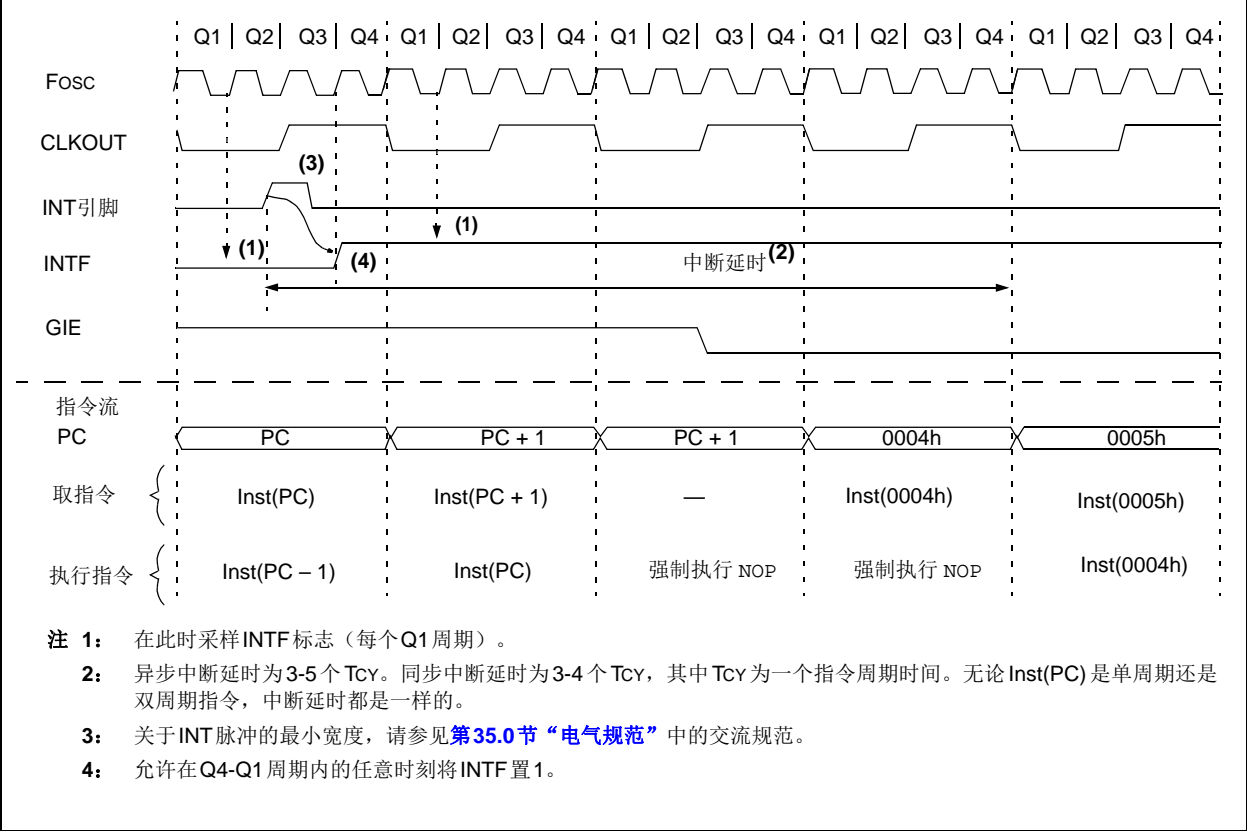


图7-3: INT引脚中断时序



## 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位到中断向量。

## 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（ $\overline{TO}$ 和 $\overline{PD}$ 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。

## 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/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0
GIE <sup>(1)</sup>	PEIE <sup>(2)</sup>	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF <sup>(3)</sup>
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位，读为0

u = 不变

x = 未知

-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<sup>(1)</sup>  
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<sup>(2)</sup>  
1 = 允许所有有效外设中断  
0 = 禁止所有外设中断
- bit 5      **TMR0IE**: Timer0 上溢中断允许位  
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 上溢中断标志位  
1 = TMR0 寄存器已上溢  
0 = TMR0 寄存器未上溢
- 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<sup>(3)</sup>  
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:** 当中断条件产生时，不管相应的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。

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

**3:** 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。

# PIC16(L)F1615/9

## 寄存器 7-2:      **PIE1: 外设中断允许寄存器 1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
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 PR2 匹配中断允许位  
1 = 允许 Timer2 与 PR2 匹配中断  
0 = 禁止 Timer2 与 PR2 匹配中断
- bit 0      **TMR1IE:** Timer1 上溢中断允许位  
1 = 允许 Timer1 上溢中断  
0 = 禁止 Timer1 上溢中断

**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 7-3:      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	TMR6IE	TMR4IE	CCP2IE
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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      **TMR6IE:** TMR6 与 PR6 匹配中断允许位  
1 = 允许 Timer6 与 PR6 匹配中断  
0 = 禁止 Timer6 与 PR6 匹配中断
- bit 1      **TMR4IE:** TMR4 与 PR4 匹配中断允许位  
1 = 允许 Timer4 与 PR4 匹配中断  
0 = 禁止 Timer4 与 PR4 匹配中断
- bit 0      **CCP2IE:** CCP2 中断允许位  
1 = 允许 CCP2 中断  
0 = 禁止 CCP2 中断

**注 1:** 要允许任何一个外设中断, 必须将 INTCON 寄存器的 PEIE 位置1。

# PIC16(L)F1615/9

寄存器 7-4:        **PIE3: 外设中断允许寄存器3**

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
—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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        **CWGIE:** 互补波形发生器 (CWG) 中断允许位  
              1 = 允许 CWG 中断  
              0 = 禁止 CWG 中断
- bit 4        **ZCDIE:** 过零检测 (ZCD) 中断允许位  
              1 = 允许 ZCD 中断  
              0 = 禁止 ZCD 中断
- bit 3        **CLC4IE:** 可配置逻辑模块4中断允许位  
              1 = 允许 CLC 4 中断  
              0 = 禁止 CLC 4 中断
- bit 2        **CLC3IE:** 可配置逻辑模块3中断允许位  
              1 = 允许 CLC 3 中断  
              0 = 禁止 CLC 3 中断
- bit 1        **CLC2IE:** 可配置逻辑模块2中断允许位  
              1 = 允许 CLC 2 中断  
              0 = 禁止 CLC 2 中断
- bit 0        **CLC1IE:** 可配置逻辑模块1中断允许位  
              1 = 允许 CLC 1 中断  
              0 = 禁止 CLC 1 中断

**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 7-5: PIE4: 外设中断允许寄存器 4

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
SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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	<b>SCANIE:</b> 扫描器中断允许位 1 = 允许扫描器中断 0 = 禁止扫描器中断
bit 6	<b>CRCIE:</b> CRC 中断允许位 1 = 允许 CRC 中断 0 = 禁止 CRC 中断
bit 5	<b>SMT2PWAIE:</b> SMT2 脉宽采集中断允许位 1 = 允许 SMT 采集中断 0 = 禁止 SMT 采集中断
bit 4	<b>SMT2PRAIE:</b> SMT2 周期采集中断允许位 1 = 允许 SMT 采集中断 0 = 禁止 SMT 采集中断
bit 3	<b>SMT2IE:</b> SMT2 匹配中断允许位 1 = 允许 SMT 周期匹配中断 0 = 禁止 SMT 周期匹配中断
bit 2	<b>SMT1PWAIE:</b> SMT1 脉宽采集中断允许位 1 = 允许 SMT 采集中断 0 = 禁止 SMT 采集中断
bit 1	<b>SMT1PRAIE:</b> SMT1 周期采集中断允许位 1 = 允许 SMT 采集中断 0 = 禁止 SMT 采集中断
bit 0	<b>SMT1IE:</b> SMT1 匹配中断允许位 1 = 允许 SMT 周期匹配中断 0 = 禁止 SMT 周期匹配中断

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

# PIC16(L)F1615/9

## 寄存器 7-6: PIE5: 外设中断允许寄存器5

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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      **TMR3GIE:** Timer3 门控中断允许位  
1 = 允许 Timer3 门控中断  
0 = 禁止 Timer3 门控中断
- bit 6      **TMR3IE:** Timer3 上溢中断允许位  
1 = 允许 Timer3 上溢中断  
0 = 禁止 Timer3 上溢中断
- bit 5      **TMR5GIE:** Timer5 门控中断允许位  
1 = 允许 Timer5 门控中断  
0 = 禁止 Timer5 门控中断
- bit 4      **TMR5IE:** Timer5 上溢中断允许位  
1 = 允许 Timer5 上溢中断  
0 = 禁止 Timer5 上溢中断
- bit 3      **未实现:** 读为0
- bit 2      **AT1IE:** 角度定时器1 中断允许位  
1 = 允许角度定时器1 中断  
0 = 禁止角度定时器1 中断
- bit 1      **PID1EIE:** PID 错误中断允许位  
1 = 允许 PID 错误中断  
0 = 禁止 PID 错误中断
- bit 0      **PID1DIE:** PID 中断允许位  
1 = 允许 PID 中断  
0 = 禁止 PID 中断

**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 7-7: PIR1: 外设中断请求寄存器 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	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	<b>TMR1GIF:</b> 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	<b>ADIF:</b> 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	<b>RCIF:</b> 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	<b>TXIF:</b> 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	<b>SSP1IF:</b> 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	<b>CCP1IF:</b> 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	<b>TMR2IF:</b> Timer2 与 PR2 匹配中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 0	<b>TMR1IF:</b> Timer1 上溢中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态

注:	当中断条件产生时, 不管相应的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)F1615/9

寄存器 7-8:        **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	TMR6IF	TMR4IF	CCP2IF
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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	<b>OSFIF:</b> 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	<b>C2IF:</b> 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	<b>C1IF:</b> 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	<b>未实现:</b> 读为 0
bit 3	<b>BCL1IF:</b> 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	<b>TMR6IF:</b> Timer6 与 PR6 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 1	<b>TMR4IF:</b> Timer4 与 PR4 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	<b>CCP2IF:</b> CCP2 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:** 当中断条件产生时，不管相应的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 7-9: PIR3: 外设中断请求寄存器3

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
—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF
bit 7							bit 0

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

bit 7-6	未实现: 读为0
bit 5	<b>CWGIF:</b> CWG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	<b>ZCDIF:</b> ZCD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	<b>CLC4IF:</b> 可配置逻辑模块4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	<b>CLC3IF:</b> 可配置逻辑模块3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	<b>CLC2IF:</b> 可配置逻辑模块2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	<b>CLC1IF:</b> 可配置逻辑模块1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断

注:	当中断条件产生时, 不管相应的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)F1615/9

## 寄存器 7-10: PIR4: 外设中断请求寄存器 4

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
SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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	<b>SCANIF:</b> 扫描器中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 6	<b>CRCIF:</b> CRC 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	<b>SMT2PWAIF:</b> SMT2 脉宽采集中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 4	<b>SMT2PRAIF:</b> SMT2 周期采集中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 3	<b>SMT2IF:</b> SMT2 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 2	<b>SMT1PWAIF:</b> SMT1 脉宽采集中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 1	<b>SMT1PRAIF:</b> SMT1 周期采集中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 0	<b>SMT1IF:</b> SMT1 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

**注:** 当中断条件产生时, 不管相应的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 7-11: PIR5: 外设中断请求寄存器 5

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF
bit 7				bit 0			

图注:		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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	<b>TMR3GIF:</b> Timer3 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	<b>TMR3IF:</b> Timer3 上溢中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 5	<b>TMR5GIF:</b> Timer5 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 4	<b>TMR5IF:</b> Timer5 上溢中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 3	<b>未实现:</b> 读为0
bit 2	<b>AT1IF:</b> 角度定时器 1 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	<b>PID1EIF:</b> PID 错误中断标志位 1 = 中断处于待处理状态 0 = 中断不处于待处理状态
bit 0	<b>PID1DIF:</b> PID 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:	当中断条件产生时, 不管相应的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)F1615/9

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	105
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			230
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	107
PIE3	—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	108
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IF	109
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE	110
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF	112
PIR3	—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	113
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	114
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF	115

图注： — = 未实现位，读为0。中断不使用阴影单元。



## 8.0 掉电模式（休眠）

通过执行 SLEEP 指令可进入掉电模式。

在进入休眠模式时，会存在以下条件：

1. 如果在休眠期间使能 WDT，则 WDT 会清零，但保持运行。
2. STATUS 寄存器的  $\overline{\text{PD}}$  位被清零。
3. STATUS 寄存器的  $\overline{\text{TO}}$  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
  - Timer1 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
- 来自 I/O 引脚的外部电路灌电流
- 来自 I/O 引脚的内部电路拉电流
- 从带弱上拉的引脚汲取的电流
- 使用 31 kHz LFINTOSC 的模块
- 使用 HFINTOSC 的 CWG 模块

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

可能产生拉电流的内部电路示例包括 FVR 模块。关于该模块的更多信息，请参见第 15.0 节“固定参考电压（FVR）”。

### 8.1 从休眠状态唤醒

发生以下任一事件时，器件将从休眠状态唤醒：

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

前三个事件会导致器件复位。后三个事件视为继续执行程序。要确定是发生了器件复位还是唤醒事件，请参见第 6.12 节“确定复位原因”。

当执行 SLEEP 指令时，下一条指令（PC + 1）被预先取出。如果希望通过中断事件唤醒器件，则必须允许相应的中断允许位。唤醒与 GIE 位的状态无关。如果 GIE 位被禁止，器件将继续执行 SLEEP 指令之后的指令。如果 GIE 位被允许，器件将执行 SLEEP 指令之后的指令，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。

器件从休眠状态唤醒时，WDT 都将被清零，而与唤醒原因无关。

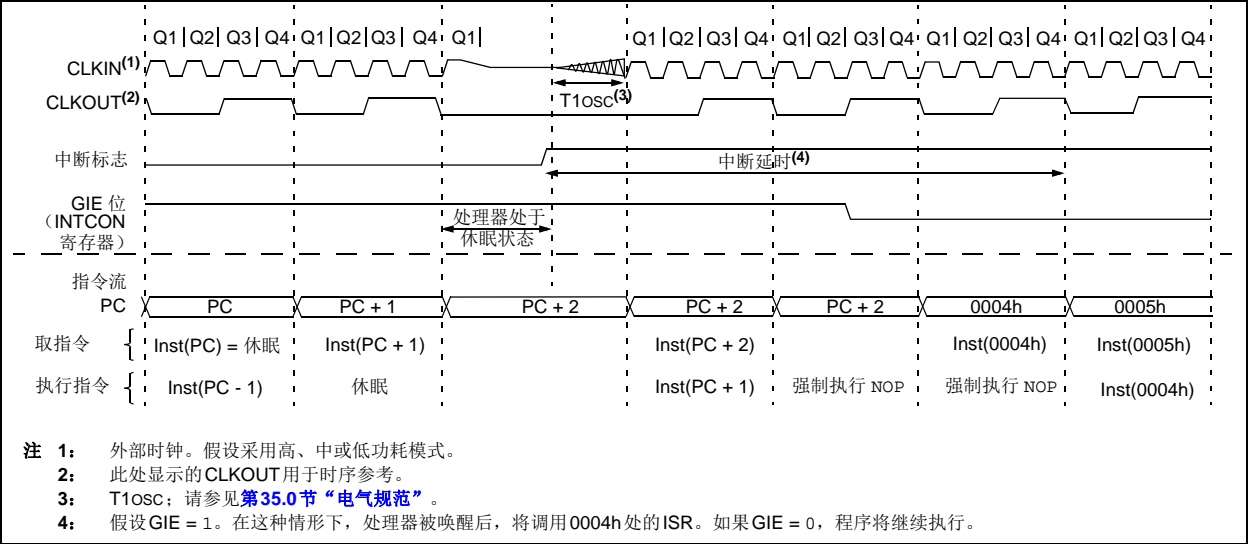
#### 8.1.1 使用中断唤醒

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

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

图8-1： 通过中断从休眠状态唤醒



8.2 低功耗休眠模式

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 及其关联的参考电路必须保持活动状态。

通过低功耗休眠模式, 用户可以优化休眠模式下的工作电流。选择低功耗休眠模式的方法是将 VREGCON 寄存器的 VREGPM 位置 1, 从而在每次器件处于休眠模式时, 将 LDO 和参考电压电路置于低功耗状态。

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。使能这些外设时, LDO 将保持在正常功耗模式。低功耗休眠模式旨在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 (带外部时钟源)

互补波形发生器 (CWG) 可以利用 HFINTOSC 振荡器作为时钟源或作为输入源。在某些条件下, 当选择 HFINTOSC 与 CWG 模块配合使用时, HFINTOSC 将在休眠期间保持活动状态。这会直接影响休眠模式的电流。

更多信息, 请参见第 28.11 节“休眠期间的操作”。

注: PIC16LF1615/9 不具有可配置的低功耗休眠模式。PIC16LF1615/9 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。该器件的最大 VDD 和 I/O 电压低于 PIC16F1615/9。更多信息, 请参见第 35.0 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<sup>(1)</sup>

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<sup>(2)</sup>

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<sup>(2)</sup>

休眠时消耗的电流较高, 唤醒速度较快

bit 0 保留: 读为1。该位保持置1。

注 1: 仅限 PIC16F1615/9。

2: 请参见第35.0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	IOCFIE	TMR0IF	INTF	IOCFIF	105
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	188
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	188
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	188
IOCCP	IOCCP7 <sup>(1)</sup>	IOCCP6 <sup>(1)</sup>	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	188
IOCCN	IOCCN7 <sup>(1)</sup>	IOCCN6 <sup>(1)</sup>	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	188
IOCCF	IOCCF7 <sup>(1)</sup>	IOCCF6 <sup>(1)</sup>	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	188
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	107
PIE3	—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	108
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IF	109
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE	110
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF	112
PIR3	—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	113
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	114
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF	115
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	26
WDTCON0	—	—	WDTPS<4:0>					SEN	124

图注: — = 未实现, 读为0。掉电模式下不使用阴影单元。

注 1: 仅限 PIC16(L)F1619。

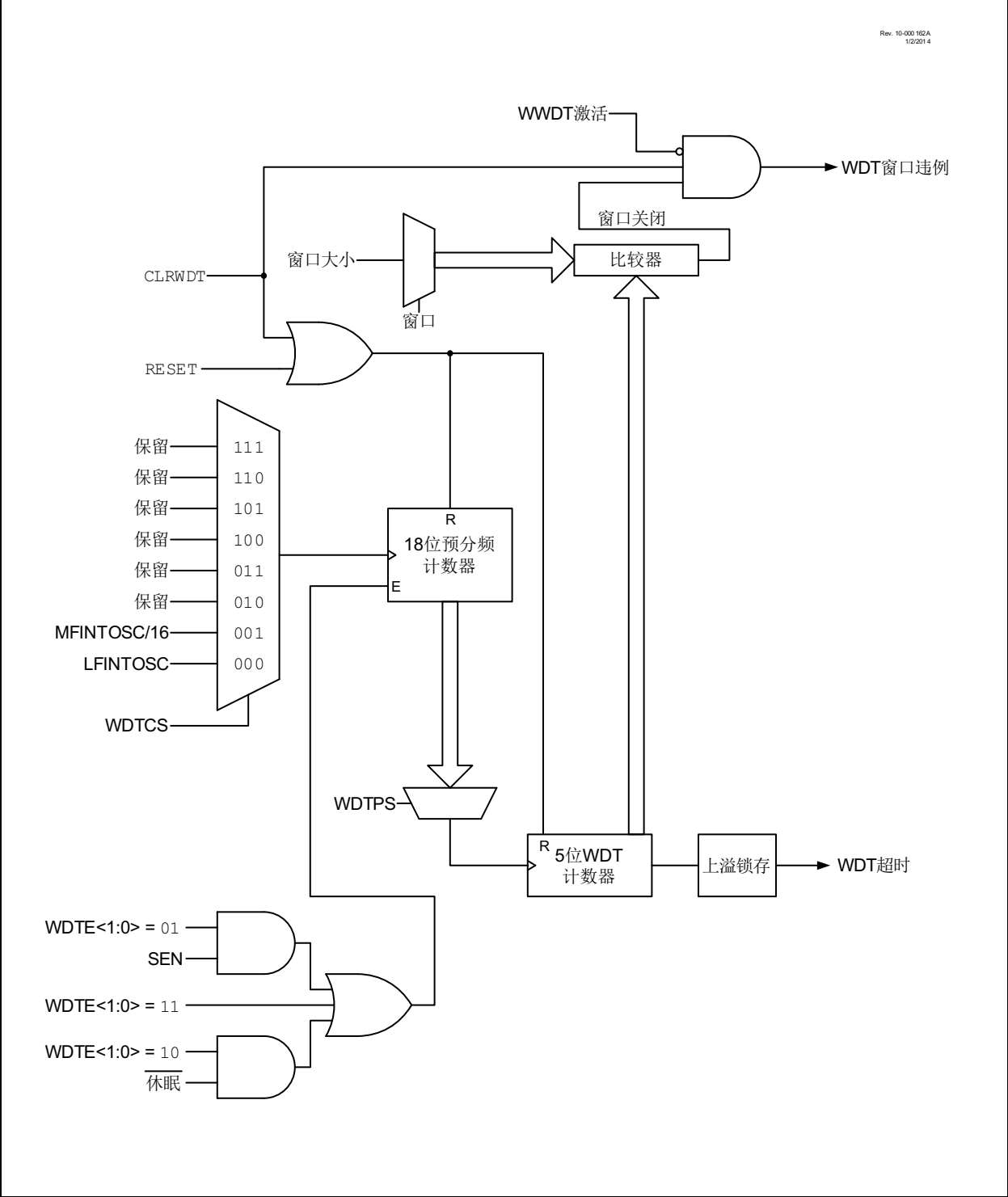
## 9.0 窗口看门狗定时器（WDT）

看门狗定时器（WDT）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。窗口看门狗定时器（WDT）的不同之处在于，只有在超时周期期间的特定窗口内执行 CLRWDT 指令时才会接受该指令。

WDT 具有以下特性：

- 可选择的时钟源
- 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）
- 可配置窗口大小范围为超时周期的 12.5% 至 100%
- 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可以根据WDTCCS<2:0>配置位或WDTCON1的WDTCS<2:0>位的值，基于31 kHz LFINTOSC或31.25 kHz MFINTOSC内部振荡器产生其时基。本章中的时间间隔均基于1 ms的最小标称时间间隔。关于LFINTOSC和MFINTOSC容差，请参见第35.0节“电气规范”。

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将总是使能。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将使能。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WDTCON0寄存器的SEN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>	SEN	器件模式	WDT模式
11	X	X	有效
10	X	唤醒	有效
		休眠	禁止
01	1	X	有效
	0	X	禁止
00	X	X	禁止

9.3 超时周期

WDTCON0寄存器的WDTPS位用于设置从1 ms至256秒（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 看门狗窗口

看门狗定时器具有一种可选的窗口模式，由WDTCWS<2:0>配置位和WDTCON1寄存器的WINDOW<2:0>位控制。在窗口模式下，CLRWDT指令必须在WDT周期的容许窗口内发生。在该窗口范围外发生的任何CLRWDT指令都会触发窗口违例，使WDT复位，这与WDT超时类似。图9-2给出了一个示例。

窗口大小由WDTCWS<2:0>配置位或WDTCON1的WINDOW<2:0>位控制（如果WDTCWS<2:0>=111）。

发生窗口违例时，将产生复位，并且PCON寄存器的WDTWV位将被清零。该位在POR时置1，也可以由固件置1。

9.5 清零WDT

当发生以下任何条件时，WDT被清零：

- 任何复位
- 执行了有效的CLRWDT指令
- 器件进入休眠模式
- 器件从休眠状态唤醒
- WDT被禁止
- 振荡器起振定时器（OST）正在运行
- 对WDTCON0或WDTCON1寄存器执行任意写操作时

9.5.1 CLRWDT注意事项（窗口模式）

在窗口模式下，必须先激活WDT，之后的CLRWDT指令才会清零定时器。这通过读取WDTCON0寄存器来执行。未执行这种激活操作的情况下执行CLRWDT指令会触发窗口违例。

更多信息，请参见表9-2。

9.6 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。

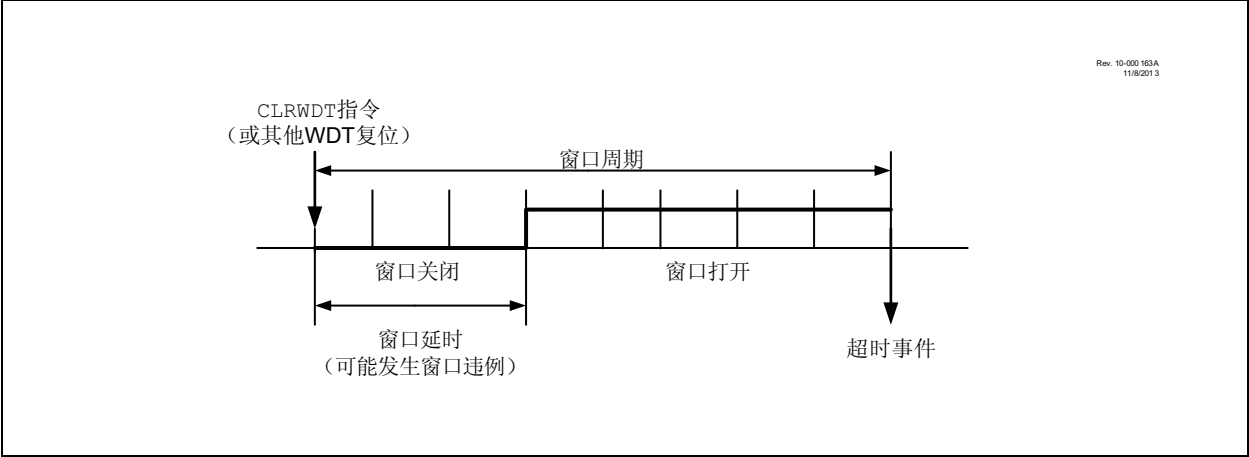
WDT一直保持清零，直到OST（如果使能）完成为止。关于OST的更多信息，请参见第5.0节“振荡器模块”。

在器件处于休眠模式的情况下发生WDT超时，不会产生复位。器件将会唤醒并继续工作。STATUS寄存器中的TO和PD位会发生改变，指示发生的事件。也可以使用PCON寄存器中的RWDT位。更多信息，请参见第3.0节“存储器构成”。

表9-2: WDT清零条件

条件	WDT
WDTE<1:0> = 00	清零
WDTE<1:0> = 01 且 SEN = 0	
WDTE<1:0> = 10 并进入休眠状态	
CLRWDT 命令	
检测到振荡器故障	
退出休眠 + 系统时钟 = T1OSC、EXTRC、INTOSC或EXTCLK	
更改INTOSC分频比（IRCF位）	不受影响

图9-2: 窗口周期和延时



# PIC16(L)F1615/9

## 9.7 寄存器定义：窗口看门狗定时器控制

寄存器9-1: **WDTCON0: 看门狗定时器控制寄存器0**

U-0	U-0	R/W <sup>(3)</sup> -q/q <sup>(2)</sup>	R/W <sup>(3)</sup> -q/q <sup>(2)</sup>	R/W <sup>(3)</sup> -q/q <sup>(2)</sup>	R/W <sup>(3)</sup> -q/q <sup>(2)</sup>	R/W <sup>(3)</sup> -q/q <sup>(2)</sup>	R/W-0/0
—	—	WDTPS<4:0> <sup>(1)</sup>					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

q = 值取决于具体条件

bit 7-6 未实现: 读为0

bit 5-1 **WDTPS<4:0>:** 看门狗定时器预分频比选择位<sup>(1)</sup>

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 = 1:262144 ( $2^{17}$ ) (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 **SEN:** 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。

**2:** 当CONFIG3中的WDTCP3<4:0> = 11111时, WDTPS<4:0>的复位值为01011。否则, WDTPS<4:0>的复位值等于CONFIG3中的WDTCP3<4:0>。

**3:** 当CONFIG3中的WDTCP3<4:0> ≠ 11111时, 这些位是只读的。



寄存器 9-2: WDTCON1: 看门狗定时器控制寄存器 1

U-0	R/W <sup>(3)</sup> -q/q <sup>(1)</sup>	R/W <sup>(3)</sup> -q/q <sup>(1)</sup>	R/W <sup>(3)</sup> -q/q <sup>(1)</sup>	U-0	R/W <sup>(4)</sup> -q/q <sup>(2)</sup>	R/W <sup>(4)</sup> -q/q <sup>(2)</sup>	R/W <sup>(4)</sup> -q/q <sup>(2)</sup>
—	WDTCS<2:0>			—	WINDOW<2:0>		
bit 7				bit 0			

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

- bit 7       **未实现:** 读为0
- bit 6-4     **WDTCS<2:0>:** 看门狗定时器时钟选择位
- 111 = 保留
- 
- 
- 
- 010 = 保留
- 001 = MFINTOSC 31.25 kHz
- 000 = LFINTOSC 31 kHz
- bit 3       **未实现:** 读为0
- bit 2-0     **WINDOW<2:0>:** 看门狗定时器窗口选择位

WINDOW<2:0>	窗口延时的 时间百分比	窗口打开的 时间百分比
111	N/A	100
110	12.5	87.5
101	25	75
100	37.5	62.5
011	50	50
010	62.5	37.5
001	75	25
000	87.5	12.5

- 注 1:** 如果CONFIG3中的WDTCCS<2:0> = 111, 则WDTCS<2:0>的复位值为000。
- 2:** WINDOW<2:0>的复位值由CONFIG3寄存器中WDTCWS<2:0>的值决定。
- 3:** 如果CONFIG3中的WDTCCS<2:0> ≠ 111, 则这些位是只读的。
- 4:** 如果CONFIG3中的WDTCWS<2:0> ≠ 111, 则这些位是只读的。

# PIC16(L)F1615/9

## 寄存器 9-3: WDTPSL: WDT 预分频比选择低字节寄存器 (只读)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<7:0> <sup>(1)</sup>							
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 **PSCNT<7:0>**: 预分频比选择低字节位<sup>(1)</sup>

**注 1:** 18 位 WDT 预分频值 PSCNT<17:0> 包括 WDTPSL、WDTPSH 和 WDTTMR 寄存器的低位。PSCNT<17:0> 用于调试操作, 应在正常操作过程中读取。

## 寄存器 9-4: WDTPSH: WDT 预分频比选择高字节寄存器 (只读)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
PSCNT<15:8> <sup>(1)</sup>							
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 **PSCNT<15:8>**: 预分频比选择高字节位<sup>(1)</sup>

**注 1:** 18 位 WDT 预分频值 PSCNT<17:0> 包括 WDTPSL、WDTPSH 和 WDTTMR 寄存器的低位。PSCNT<17:0> 用于调试操作, 应在正常操作过程中读取。

## 寄存器 9-5: WDTTMR: WDT 定时器寄存器 (只读)

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
WDTTMR<3:0>					STATE	PSCNT<17:16> <sup>(1)</sup>	
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 **WDTTMR<4:0>**: 看门狗定时器值

bit 2 **STATE**: WDT 激活状态位  
1 = WDT 已激活  
0 = WDT 未激活

bit 1-0 **PSCNT<17:16>**: 预分频比选择最高字节位<sup>(1)</sup>

**注 1:** 18 位 WDT 预分频值 PSCNT<17:0> 包括 WDTPSL、WDTPSH 和 WDTTMR 寄存器的低位。PSCNT<17:0> 用于调试操作, 应在正常操作过程中读取。

**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>		89
PCON	STKOVF	STKUNF	WDTWV	RWD $\overline{T}$	RMCLR	R $\overline{I}$	POR	BOR	98
STATUS	—	—	—	T $\overline{O}$	P $\overline{D}$	Z	DC	C	26
WDTCON0	—	—	WDTPS<4:0>					SEN	124
WDTCON1	—	WDTCS<2:0>			—	WINDOW<2:0>			124
WDTPSL	PSCNT<7:0>								124
WDTPSH	PSCNT<15:8>								124
WDTTMR	—	WDTTMR<4:0>				STATE	PSCNT<17:16>		124

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	bit	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>		—	69
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	—	—	FOSC<2:0>			
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			72
	7:0	—	WDTE<1:0>		WDTCPSC<4:0>					

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VDD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CP位）和写保护（配置字中的WRT<1:0>位）。

代码保护（CP=0）<sup>(1)</sup>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CP位来使能。

10.1 PMADRL和PMADRH寄存器

PMADRH:PMADRL寄存器对能寻址最大16K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。

要能使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PMDATH:PMDATL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。

表10-1： 闪存构成（按器件）

器件	行擦除（字）	写锁存器（字）
PIC16(L)F1615	32	32
PIC16(L)F1619		

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CFG5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寄存器对将保留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

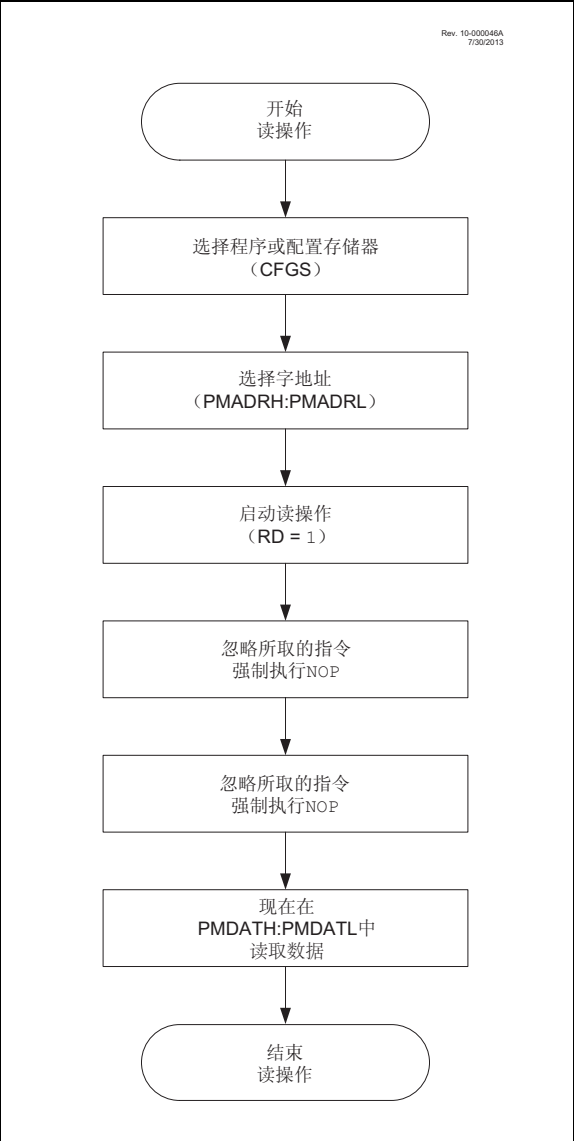
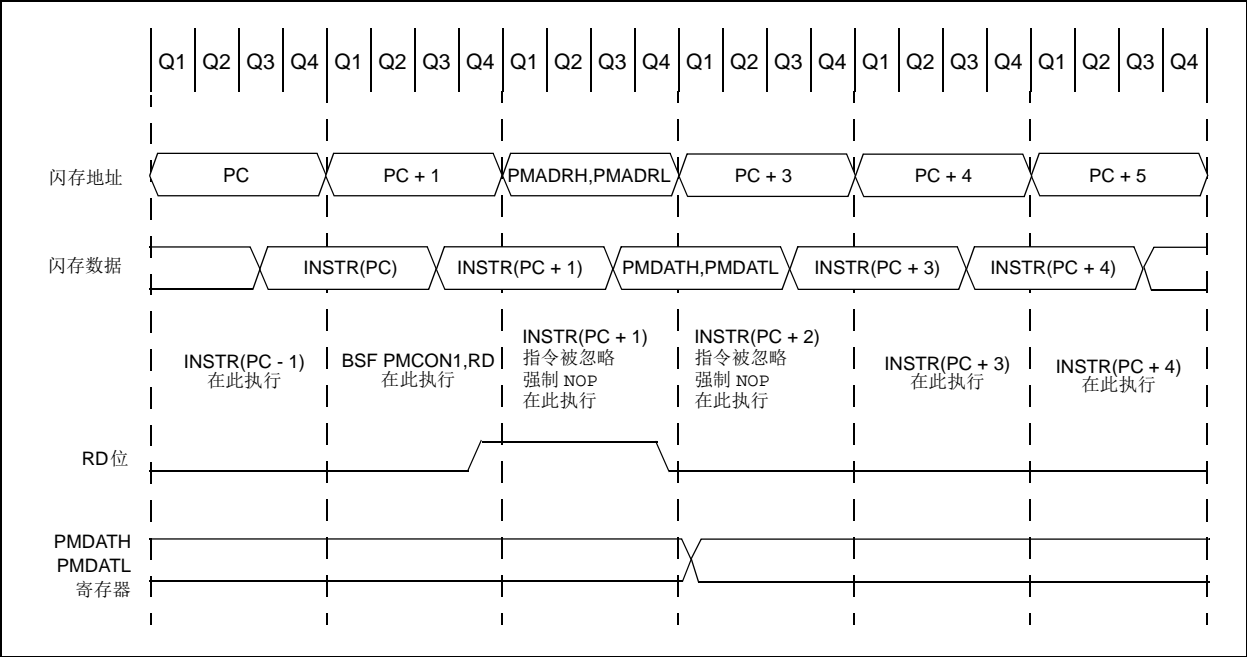


图10-2： 闪存程序存储器读周期执行时序



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, CFGS    ; Do not select Configuration Space
BSF       PMCON1, RD      ; Initiate read
NOP       ; Ignored (Figure 10-2)
NOP       ; Ignored (Figure 10-2)

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

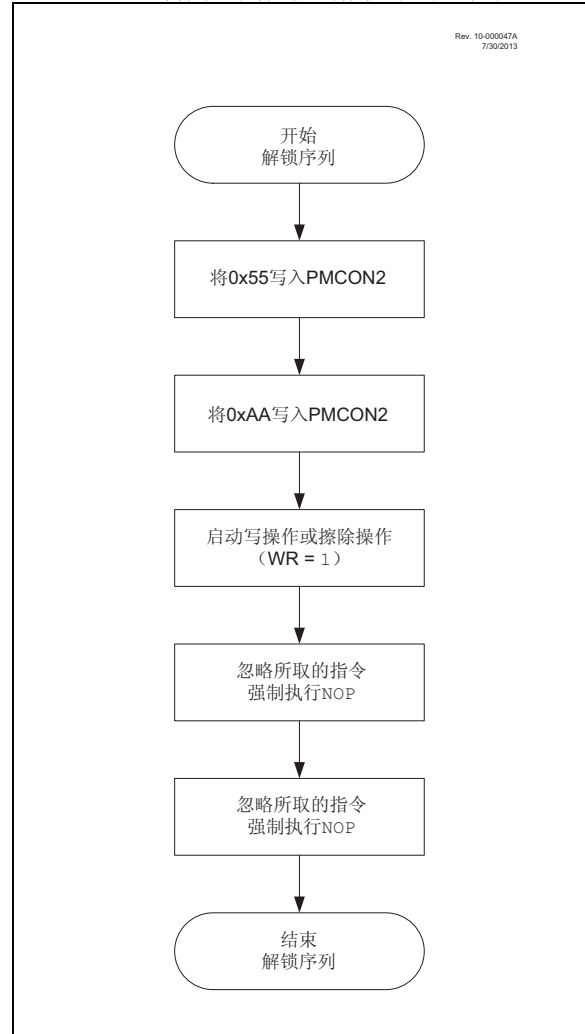
解锁序列包含以下步骤：

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。

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

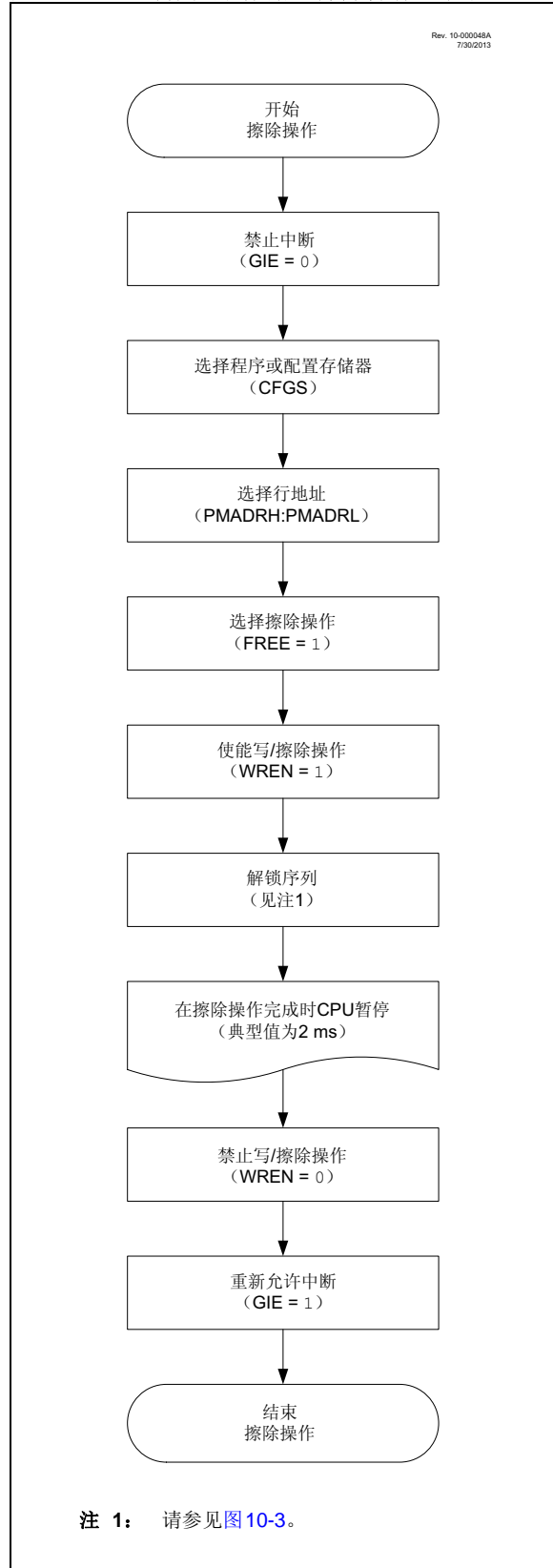
在执行代码时，程序存储器只能按行进行擦除。要擦除某行，请执行以下步骤：

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CFG5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完成擦除操作。用户必须在紧随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2ms（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





例10-2: 擦除程序存储器的一行

必需的序列

```
; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. 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  PMADRL
        MOVF     ADDRL,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,CFGSR     ; Not configuration space
        BSF      PMCON1,FREER     ; Specify an erase operation
        BSF      PMCON1,WREN      ; Enable writes

        MOVLW    55h             ; Start of required sequence to initiate erase
        MOVWF    PMCON2          ; Write 55h
        MOVLW    AAh             ;
        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至步骤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行写操作）。

写锁存器将对齐到由PMADRH:PMADRL高11位（PMADRH<6:0>:PMADRL<7:4>）定义的闪存行地址边界处，PMADRL的低4位（PMADRL<3:0>）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CFG5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，写操作之后只会向写锁存器装入数据，而不会启动对闪存程序存储器的写操作。
4. 将要写入的存储单元的地址装入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

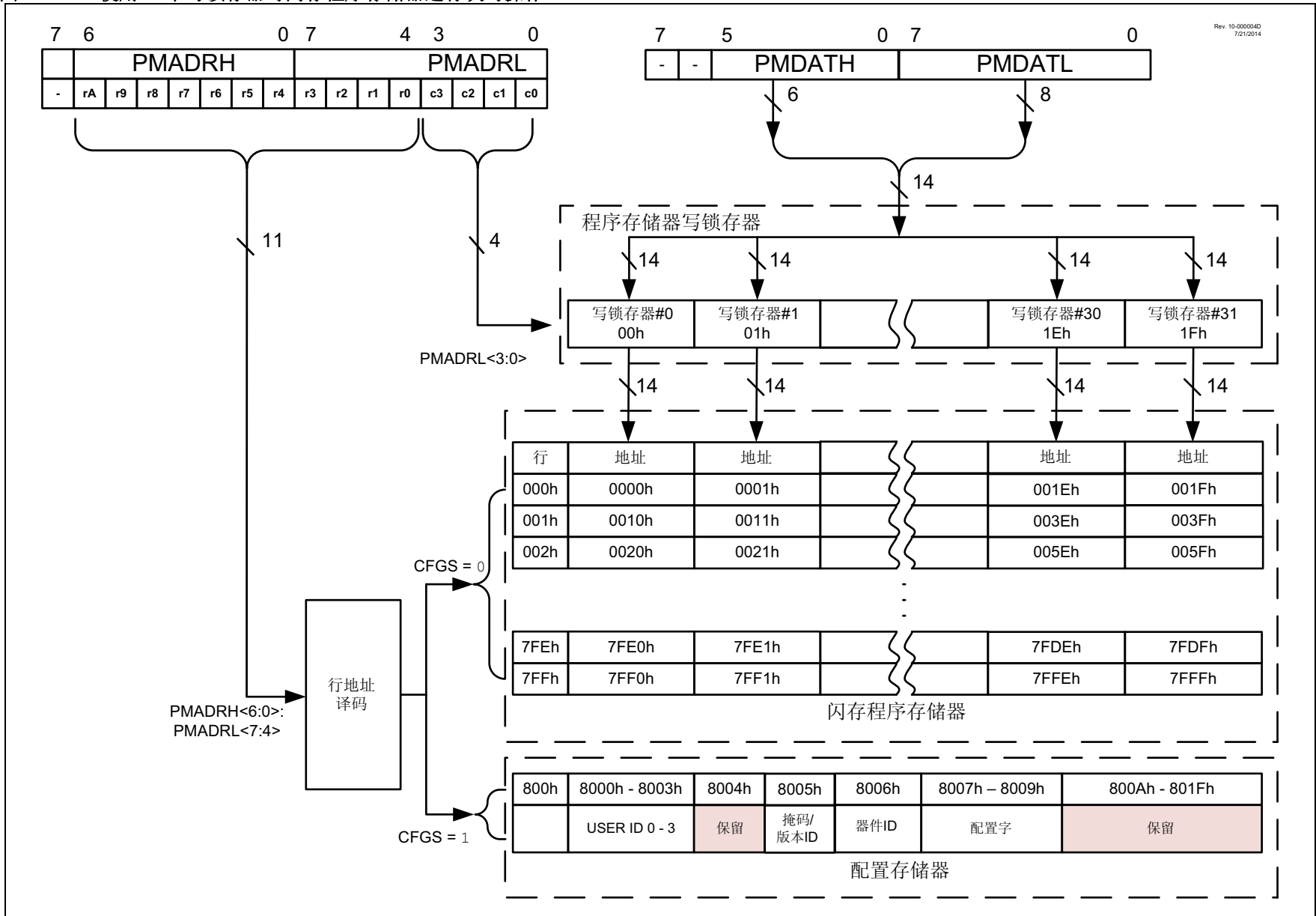
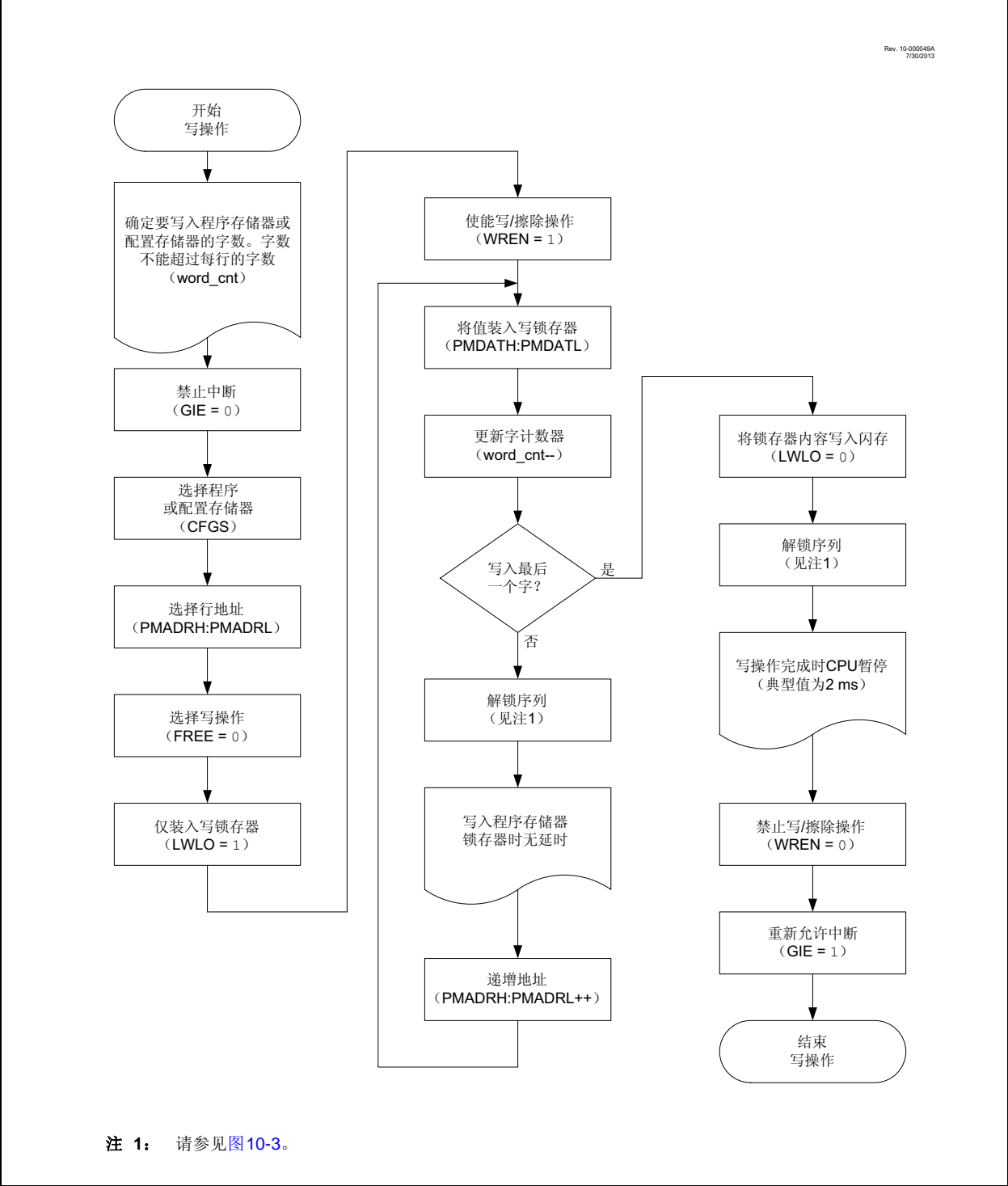


图10-6： 闪存程序存储器写操作流程



## 例10-3: 写入闪存程序存储器 (32个写锁存器)

```

; 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     ADDRH,W        ; Load initial address
    MOVWF    PMADRH         ;
    MOVF     ADDRL,W        ;
    MOVWF    PMADRL         ;
    MOVLW    LOW DATA_ADDR ; Load initial data address
    MOVWF    FSR0L          ;
    MOVLW    HIGH DATA_ADDR ; Load initial data address
    MOVWF    FSR0H          ;
    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    ;

    ; 必需的序列
    MOVLW    55h             ; Start of required write sequence:
    MOVWF    PMCON2          ; Write 55h
    MOVLW    AAh             ;
    MOVWF    PMCON2          ; Write AAh
    BSF      PMCON1,WR        ; Set WR bit to begin write
    NOP      ; NOP instructions are forced as processor
    NOP      ; 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

    ; 必需的序列
    MOVLW    55h             ; Start of required write sequence:
    MOVWF    PMCON2          ; Write 55h
    MOVLW    AAh             ;
    MOVWF    PMCON2          ; Write AAh
    BSF      PMCON1,WR        ; Set WR bit to begin write
    NOP      ; NOP instructions are forced as processor writes
    NOP      ; all the program memory write latches simultaneously
    NOP      ; to program memory.
    NOP      ; After NOPs, the processor
    NOP      ; stalls until the self-write process is complete
    NOP      ; after write processor continues with 3rd instruction

    BCF      PMCON1,WREN     ; Disable writes
    BSF      INTCON,GIE      ; Enable interrupts

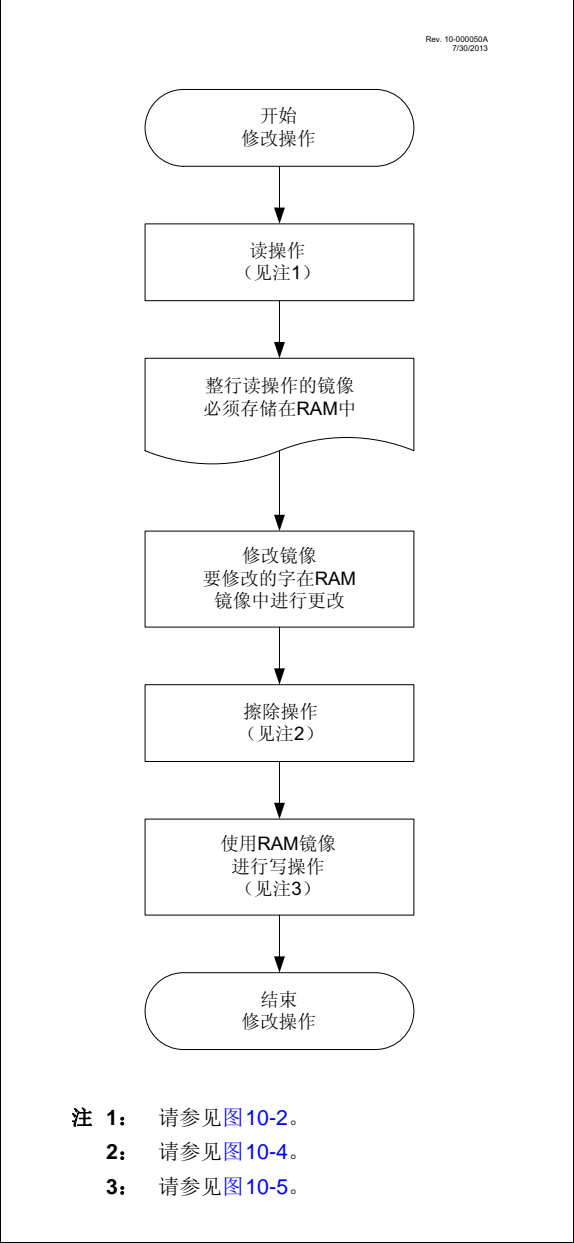
```

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



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。对于读操作和写操作，可能存在不同的访问权限。请参见表10-2。

对表10-2中所列地址外的其他地址启动读访问时，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
8006h/8005h	器件ID/版本ID	是	否
8007h-8009h	配置字1、2和3	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

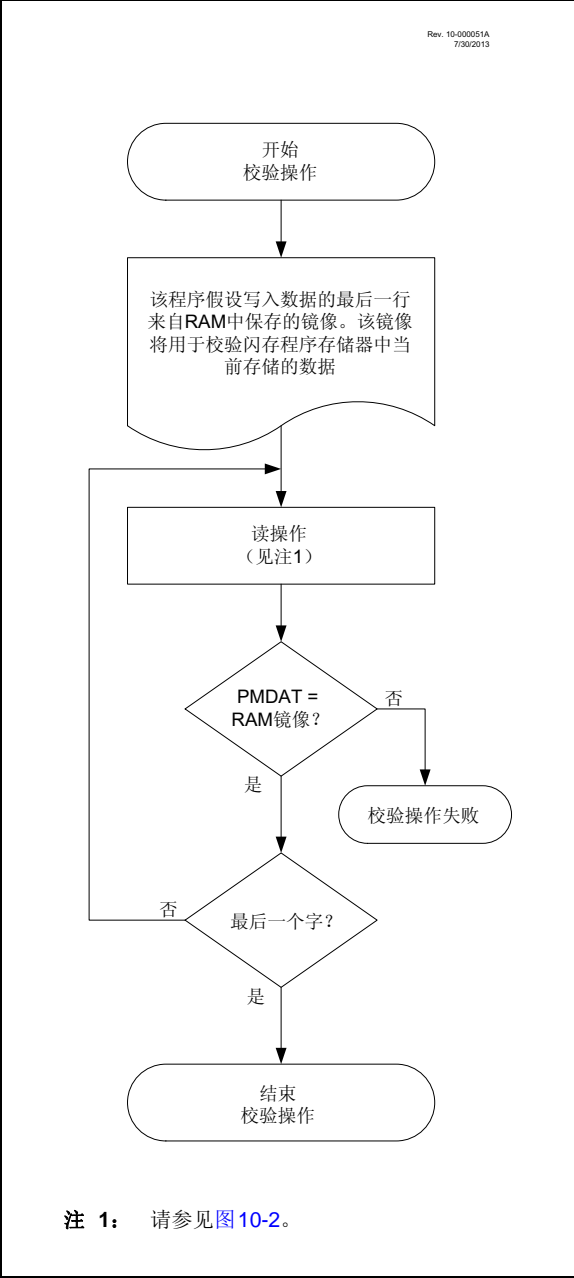
BSF      PMCON1,CFG8     ; Select Configuration Space
BCF      INTCON,GIE      ; Disable interrupts
BSF      PMCON1,RD       ; Initiate read
NOP      ; Executed (See Figure 10-2)
NOP      ; Ignored (See Figure 10-2)
BSF      INTCON,GIE      ; Restore interrupts

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





## 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							bit 0

#### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
u = 不变                      x = 未知                      -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							bit 0

#### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
u = 不变                      x = 未知                      -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							bit 0

#### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
u = 不变                      x = 未知                      -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
— <sup>(1)</sup>	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有效位

**注 1:** 未实现, 读为1。

# PIC16(L)F1615/9

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 <sup>(2)</sup>	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
— <sup>(1)</sup>	CFGS	LWLO	FREE	WRERR	WREN	WR	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	<b>未实现:</b> 读为1
bit 6	<b>CFGS:</b> 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	<b>LWLO:</b> 仅装入写锁存器位 <sup>(3)</sup> 1 = 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 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	<b>FREE:</b> 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	<b>WRERR:</b> 编程/擦除错误标志位 1 = 指示试图执行不合法的编程或擦除序列，或表示序列意外终止（试图将WR位置1（写入1）时自动将该位置1） 0 = 编程或擦除操作正常完成
bit 2	<b>WREN:</b> 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	<b>WR:</b> 写控制位 1 = 启动程序闪存的编程/擦除操作。 操作是自定时的，一旦该操作完成，该位即由硬件清零。 用软件只能将WR位置1（不能清零）。 0 = 对闪存的编程/擦除操作已完成并且变为无效
bit 0	<b>RD:</b> 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	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PMCON1	— <sup>(1)</sup>	CFG5	LWLO	FREE	WRERR	WREN	WR	RD	142
PMCON2	程序存储器控制寄存器2								143
PMADRL	PMADRL<7:0>								141
PMADRH	— <sup>(1)</sup>	PMADRH<6:0>							141
PMDATL	PMDATL<7:0>								141
PMDATH	—	—	PMDATH<5:0>						141

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, 读为1。

**表 10-4: 与闪存程序存储器相关的配置字汇总**

名称	Bit	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>		—	69
	7:0	CP	MCLRE	PWRTE	—	—	FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	71
	7:0	ZCD	—	—	—	—	PPS1WAY	WRT<1:0>		
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			72
	7:0	—	WDTE<1:0>		WDTCPs<4:0>					

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 循环冗余校验（CRC）模块

循环冗余校验（CRC）模块提供了一个软件可配置的硬件实现的CRC校验和发生器。该模块具有以下特性：

- 可使用最高16位的任意标准CRC
- 可配置多项式
- 可使用最高16位的任意种子值
- 标准和反向位序可用
- 可以自动或由用户添加扩充零
- 用于对程序存储器用户数据快速计算CRC的存储器扫描器
- 用于计算非存储器扫描器提供的CRC值的软件可装入数据寄存器

### 11.1 CRC模块概述

CRC模块提供了一种计算程序存储器校验值的方式。CRC模块与存储器扫描器配合使用，可实现更快的CRC计算。存储器扫描器可以自动向CRC模块提供数据。CRC模块也可以在直接向SFR写入数据、无需使用扫描器的情况下工作。

### 11.2 CRC功能概述

CRC模块可用于使用内置存储器扫描器或通过用户输入RAM存储器来检测闪存中的位错误。CRC模块可接受最高16位的多项式与最高16位的种子值。然后，CRC计算的校验值（或校验和）会被生成到CRCACC<15:0>寄存器中，供用户存储。CRC模块使用异或（XOR）移位寄存器实现来执行CRC计算所需的多项式除法。

例11-1:

Rev. 10-000206A  
1/8/2014

**CRC-16-ANSI**

$$x^{16} + x^{15} + x^2 + 1 \text{ (17位)}$$

标准16位表示 = 0x8005

CRCXORH = 0b10000000  
CRCXORL = 0b0000010- <sup>(1)</sup>

数据序列:

0x55, 0x66, 0x77, 0x88

DLEN = 0b0111  
PLEN = 0b1111

输入CRC的数据:

SHIFTM = 0:

01010101 01100110 01110111 10001000

SHIFTM = 1:

10101010 01100110 11101110 00010001

校验值 (ACCM = 1) :

SHIFTM = 0: 0x32D6  
CRCACCH = 0b00110010  
CRCACCL = 0b11010110

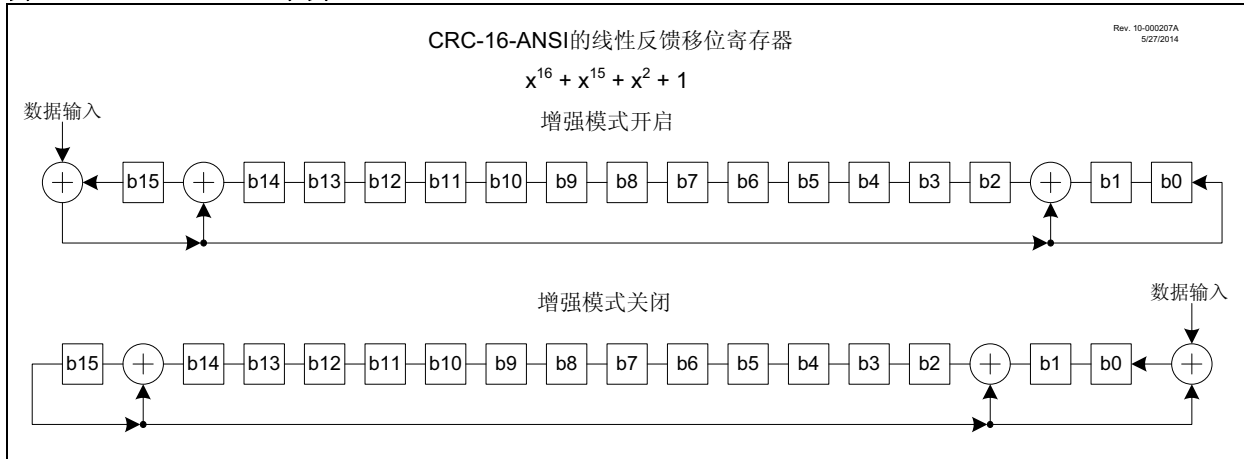
SHIFTM = 1: 0x6BA2  
CRCACCH = 0b01101011  
CRCACCL = 0b10100010

**注 1:** bit 0未实现。所有CRC多项式的LSb总是为1，CRC在计算CRC校验值时总是将它视为1。该位将在软件中读为0。

### 11.3 CRC多项式实现

可使用最高17位的任意标准多项式。PLEN<3:0>位用于指定所使用的多项式的长度。对于 $x^n$ 多项式，PLEN = n-2。在n位多项式中， $x^n$ 位和LSb将在CRC计算中用作1，因为CRC多项式的MSb和LSb必须总是为1。例如，如果使用CRC-16-ANSI，多项式的形式将类似于0x8005。它会被实现到CRCXOR<15:1>寄存器中，如例11-1所示。

例11-2: CRC LFSR 示例



## 11.4 CRC数据源

数据可以通过两种方式输入CRC模块：

- 用户数据，使用CRCDAT寄存器
- 闪存，使用程序存储器扫描器

要设置数据的位数（最高16位），必须相应地设置CRCCON1的DLEN位。该模块将仅使用CRCDATA寄存器中最高到DLEN的数据位，并忽略CRCDATA寄存器中的其他数据位。

数据会被移入CRCSHIFT，作为一种中间结果来计算位于CRCACC寄存器中的校验值。

SHIFTM位用于确定移入累加器的数据的位序。如果SHIFTM未置1，则先移入数据的MSb。DLEN的值将决定MSb。如果SHIFTM位置1，则将按反序（LSb最先）将数据移入累加器。

通过在开始CRC之前将CRCACC<15:0>寄存器设置为适当的值，可以设置CRC模块的初始种子值。

### 11.4.1 用户数据的CRC

要对用户的数据输入使用CRC模块，用户必须将数据写入CRCDAT寄存器。CRCDAT寄存器中的数据将会在每次写入CRCDATL寄存器时被锁存到移位寄存器中。

### 11.4.2 闪存的CRC

要对位于闪存中的数据使用CRC模块，用户可以初始化程序存储器扫描器，如第11.8节“程序存储器扫描配置”所述。

## 11.5 CRC校验值

在CRC计算完毕后，CRC校验值将位于CRCACC寄存器。校验值将取决于CRCCON的两种模式设置：ACCM和SHIFTM。

如果ACCM位置1，CRC模块会使用一定数量（等于多项式长度）的零来扩充数据，以确定最终的校验值。如果ACCM位未置1，CRC将在数据结束处停止。然后，可以输入一定数量（等于多项式长度）的零，以确定与扩充模式相同的校验值；或者也可以在此时输入预期的校验值，使最终结果等于0。

可能会需要最终的异或值，与校验值一起用于确定所需的CRC结果。

## 11.6 CRC中断

BUSY位从1变为0时，CRC将会产生中断。每次BUSY位发生变化时，不论是否允许CRC中断，PIR4寄存器的CRCIF中断标志位都会置1。CRCIF位只能用软件清零。CRC中断允许位是PIE4寄存器的CRCIE位。

## 11.7 配置CRC

以下步骤说明了如何正确配置CRC。

1. 确定是使用扫描器进行自动程序存储器扫描还是通过SFR接口进行手动计算，并根据所作的决定执行第11.4节“CRC数据源”中指定的操作。
2. 如果需要，在CRCACCH/L寄存器中设置起始CRC值作为种子。
3. 使用所需的发生器多项式设定CRCXORH/L寄存器。
4. 使用数据字长度-1设定CRCCON1寄存器的DLEN<3:0>位（请参见例11-1）。这将决定对于每个数据字，移位器将移入累加器多少次。
5. 使用多项式长度-2设定CRCCON1寄存器的PLEN<3:0>位（请参见例11-1）。
6. 确定是否需要移入尾随零，并相应地设置CRCCON0寄存器的ACCM位。
7. 类似地，确定是先移入MSb还是LSb，并相应地写入CRCCON0寄存器的SHIFTM位。
8. 写入CRCCON0寄存器的CRCGO位，开始移位过程。
- 9a. 如果使用手动SFR输入，则监视CRCCON0寄存器的FULL位。当FULL=0时，可以向CRCDATA/L寄存器写入另一个数据字；请记住，如果数据>8位，则应先写入CRCDATA，因为移位器会在写入CRCDATA/L寄存器时开始。
- 9b. 如果使用扫描器，则只要SCANGO位置1，扫描器就会根据需要自动在CRCDATA/L寄存器中填入数据字。
- 10a. 如果使用闪存扫描器，则通过监视SCANIF（或SCANGO位）来确定扫描器是否完成将信息压入CRCDATA寄存器。在扫描器完成之后，通过监视CRCIF（或BUSY位）来确定CRC是否已完成，之后可以从CRCACC寄存器中读取校验值。如果两个中断标志均置1（或BUSY和SCANGO位均清零），则可以从CRCACCH/L寄存器中读取完成的CRC计算结果。
- 10b. 如果使用手动输入，则通过监视CRCIF（或BUSY位）来确定CRCACC寄存器是否存放了校验值。

## 11.8 程序存储器扫描配置

如果需要，可以将程序存储器扫描模块与CRC模块配合使用，对一个程序存储器地址范围执行CRC计算。为了设置扫描器，使之与CRC配合工作，需要执行以下步骤：

1. 将EN位置1以使能模块。这可以在将SCANGO位置1之前的任意时刻进行，但如果模块被禁止，扫描器的所有内部状态都会发生复位（寄存器不受影响）。
2. 选择要使用哪种存储器访问模式（见第11.10节“扫描模式”），并相应地设置SCANCON0寄存器的MODE位。
3. 基于存储器访问模式，将SCANCON0寄存器的INTM位设置为相应的中断模式（见第11.10.5节“中断交互”）。
4. 使用要扫描的存储器的起始和结束位置设置SCANLADRL/H和SCANHADRL/H寄存器。
5. 通过将SCANCON0寄存器中的SCANGO位置1来开始扫描。扫描器将等待（CRCGO必须置1）CRC指示它已准备好处理第一个闪存单元的信号，然后开始将数据装入CRC。它将不断执行该操作，直到达到所配置的结束地址或器件上未实现的地址，此时SCANGO位会清零，扫描器功能将会停止，并触发SCANIF中断。或者，如果需要，可以用软件清零SCANGO位。

## 11.9 扫描器中断

SCANGO位从1变为0时，扫描器将会触发中断。当达到最后一个存储单元，并且数据已输入CRCDATA寄存器时，PIR4的SCANIF中断标志会置1。SCANIF位只能用软件清零。SCAN中断允许位是PIE4寄存器的SCANIE位。

## 11.10 扫描模式

存储器扫描器可以使用4种模式进行扫描：突发、窥探、并发和触发。这些模式由SCANCON0寄存器中的MODE位控制。表11-1中总结了这4种模式。

### 11.10.1 突发模式

当MODE = 01时，扫描器处于突发模式。在突发模式下，CPU操作会在将SCANGO位置1的操作之后停顿，扫描会使用指令时钟开始执行。CPU会被暂停，直到扫描停止为止。请注意，由于CPU不执行指令，无法用软

件清零 SCANGO 位，所以 CPU 将一直保持停顿，直到发生硬件结束条件之一为止。扫描器的突发模式具有最高的吞吐率，但代价是发生该模式时需要停顿其他代码执行。

11.10.2 并发模式

当 MODE = 00 时，扫描器处于并发模式。并发模式类似于突发模式，在执行存储器访问时会停顿 CPU。但是，突发模式会停顿到所有访问完成为止，而并发模式允许 CPU 在访问周期之间执行。

11.10.3 触发模式

当 MODE = 11 时，扫描器处于触发模式。触发模式的行为与并发模式相同，只是它不是在 SCANGO 位置 1 时立

即开始扫描，而是会等待来自一个独立触发时钟的上升沿，其时钟源由 SCANTRIG 寄存器决定。

11.10.4 窥探模式

当 MODE = 10 时，扫描器处于窥探模式。窥探模式会等待 CPU 不需要访问 NVM 的指令周期（如转移指令），并使用该周期执行它自己的 NVM 访问。这会导致 NVM 访问的吞吐率最低（并且需要比其他模式长得多的时间来完成扫描），但不同于其他模式，该模式不会对执行时间产生任何影响。

表 11-1: 扫描器模式汇总

MODE<1:0>		说明		
		第一次扫描访问	CPU 操作	
11	触发	在触发后尽快执行	NVM 访问期间停顿	每次访问后 CPU 恢复执行
10	窥探	在第一个死周期时	时序不受影响	每次访问后 CPU 继续执行
01	突发	尽快	NVM 访问期间停顿	CPU 被暂停，直到扫描完成为止
00	并发			每次访问后 CPU 恢复执行

11.10.5 中断交互

SCANCON0 寄存器的 INTM 位控制扫描器根据 NVM 扫描器所处的模式而对中断作出的响应，如表 11-2 所述。

表 11-2: 扫描中断模式

INTM	MODE<1:0>	
	MODE == 突发	MODE != 突发
1	中断会通过改写 SCANGO 来暂停突发操作，中断处理程序将以全速执行；扫描器突发操作在中断完成时恢复。	扫描器在中断响应期间暂停；中断将以全速执行，扫描在中断完成时恢复。
0	中断不会改写 SCANGO，扫描（突发）操作将继续进行；中断响应会被延迟，直到扫描完成为止（中断延时会增大）。	扫描器会在中断响应期间访问 NVM。如果 MODE != 窥探，中断处理程序执行速度会受影响。

一般来说，如果 INTM = 0，则扫描器优先于中断，导致中断处理速度下降和/或中断响应延时增大。如果

INTM = 1，则中断优先并具有更高的速度，使存储器扫描发生延迟。

11.10.6 WDT交互

WDT 的操作不会受扫描器活动影响。因此，长扫描（特别是在突发模式下）可能会超出WDT超时周期，导致非期望的器件复位。对于还采用了WDT的应用程序，执行存储器扫描时应考虑这一点。

11.10.7 在线调试（ICD）交互

在发生ICD暂停时，扫描器会停止并一直保持停止状态，直到恢复用户模式操作为止。调试器可以通过检查SCANCON0和SCANLADR寄存器来确定扫描的状态。

表 11-3 总结了ICD 与每种工作模式的交互。

表11-3: ICD和扫描器的交互

ICD 暂停	扫描器工作模式		
	窥探	并发触发	突发
外部暂停	如果扫描器要窥探未执行的指令（由于进入ICD），窥探操作会在ICD退出后在指令执行时发生。	如果外部暂停在扫描周期期间生效，则是否在进入ICD之前执行（被扫描延迟的）指令，这取决于外部暂停时序。	如果外部暂停在BSF(SCANCON.GO) 期间生效，则会进入ICD，而突发操作会被延迟到退出ICD为止。  否则，将会完成当前的NVM访问周期，然后中断扫描器来进入ICD。
PC断点		如果外部暂停在紧接扫描周期之前的周期期间生效，则扫描和指令执行均在退出ICD后发生。	如果外部暂停在突发操作期间生效，则突发操作会被暂停，并随着退出ICD而恢复。
数据断点		扫描周期在进入ICD前发生，指令执行在退出ICD后发生。	如果PC断点（或单步执行）处于BSF(SCANCON.GO) 上，则会在执行前进入ICD；突发操作将在退出ICD后执行，并且突发操作将运行至完成为止。  请注意，突发操作会被外部暂停中断。
单步执行		执行具有数据断点的指令，之后立即进入ICD。如果在该周期期间请求扫描，则扫描周期会被推迟到退出ICD为止。	
SWBP和ICDINST		如果扫描周期在执行调试指令后就绪，则扫描会读取PFM，然后重新进入ICD。	
		如果扫描会使SWBP停顿，则会发生扫描周期并进入ICD。	如果SWBP替换了BSF(SCANCON.GO)，则会进入ICD；指令将在退出ICD时（通过ICDINSTR寄存器）执行，突发操作将运行至完成为止。



## 11.11 寄存器定义：CRC和扫描器控制

**寄存器 11-1: CRCCON0: CRC控制寄存器0**

R/W-0/0	R/W-0/0	R-0	R/W-0/0	U-0	U-0	R/W-0/0	R-0
EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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:** CRC使能位  
1 = 从复位状态释放CRC模块  
0 = 禁止CRC, 不消耗工作电流
- bit 6 **CRCGO:** CRC启动位  
1 = 启动CRC串行移位器  
0 = 关闭CRC串行移位器
- bit 5 **BUSY:** CRC忙位  
1 = 正在进行或正在等待进行移位  
0 = 移位器中的所有有效位都已被移入到累加器, EMPTY = 1
- bit 4 **ACCM:** 累加器模式位  
1 = 使用零扩充数据  
0 = 不使用零扩充数据
- bit 3-2 **未实现:** 读为0
- bit 1 **SHIFTM:** 移位模式位  
1 = 右移 (LSb)  
0 = 左移 (MSb)
- bit 0 **FULL:** 数据路径满指示位  
1 = CRCDATH/L 寄存器已满  
0 = CRCDATH/L 寄存器已将其数据移入移位器

**寄存器 11-2: CRCCON1: CRC控制寄存器1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
DLEN<3:0>				PLEN<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 **DLEN<3:0>:** 数据长度位  
表示数据字长度 - 1 (见例 11-1)
- bit 3-0 **PLEN<3:0>:** 多项式长度位  
表示多项式长度 - 1 (见例 11-1)

# PIC16(L)F1615/9

寄存器 11-3: CRCDATAH: CRC数据高字节寄存器

R/W-xx	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
DAT<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 DAT<15:8>: CRC输入/输出数据位

寄存器 11-4: CRCDATL: CRC数据低字节寄存器

R/W-xx	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
DAT<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 DAT<7:0>: CRC输入/输出数据位  
写入该寄存器将会填充移位器。

寄存器 11-5: CRCACCH: CRC累加器高字节寄存器

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
ACC<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 ACC<15:8>: CRC累加器寄存器位  
写入该寄存器会写入CRC累加器寄存器。读取该寄存器会读取CRC累加器。

寄存器 11-6: CRCACCL: CRC累加器低字节寄存器

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
ACC<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 ACC<7:0>: CRC累加器寄存器位  
写入该寄存器会通过CRC写总线写入CRC累加器寄存器。读取该寄存器会读取CRC累加器。

寄存器 11-7:      CRCSHIFTH: CRC 移位高字节寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SHIFT<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      **SHIFT<15:8>**: CRC 移位器寄存器位  
读取该寄存器会读取CRC移位器。

寄存器 11-8:      CRCSHIFTL: CRC 移位低字节寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SHIFT<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      **SHIFT<7:0>**: CRC 移位器寄存器位  
读取该寄存器会读取CRC移位器。

寄存器 11-9:      CRCXORH: CRC 异或高字节寄存器

读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
XOR<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      **XOR<15:8>**: 多项式的项X<sub>N</sub>的异或使能位

寄存器 11-10:      CRCXORL: CRC 异或低字节寄存器

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	U-0
XOR<7:1>							—
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 7-1      **XOR<7:1>**: 多项式的项X<sub>N</sub>的异或使能位  
bit 0      未实现: 读为0

# PIC16(L)F1615/9

## 寄存器 11-11: SCANCON0: 扫描器访问控制寄存器0

R/W-0/0	R/W/HC-0/0	R-0	R-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
EN <sup>(1)</sup>	SCANGO <sup>(2, 3)</sup>	BUSY <sup>(4)</sup>	INVALID	INTM	—	MODE<1:0> <sup>(5)</sup>	
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 **EN:** 扫描器使能位<sup>(1)</sup>

1 = 使能扫描器

0 = 禁止扫描器, 内部状态复位

bit 6 **SCANGO:** 扫描器GO位<sup>(2, 3)</sup>

1 = 当CRC发送就绪信号时, 将根据MDx访问NVM, 并将数据传递到客户端外设。

0 = 不发生扫描器操作

bit 5 **BUSY:** 扫描器忙指示位<sup>(4)</sup>

1 = 正在执行扫描器周期

0 = 扫描器周期已完成 (或从未启动)

bit 4 **INVALID:** 扫描器中止信号位

1 = SCANLADRL/H已递增或包含无效的地址<sup>(6)</sup>

0 = SCANLADRL/H指向一个有效地址

bit 3 **INTM:** NVM扫描器中断管理模式选择位

如果MODE = 10:

该位被忽略

如果MODE = 01 (CPU停顿, 直至传输完所有数据为止):

1 = 在中断操作期间改写SCANGO (为零); 从中断返回后恢复扫描器

0 = SCANGO不受中断影响, 中断响应将受影响

如果MODE = 00或11:

1 = 在中断操作期间改写SCANGO (为零); 从中断返回后恢复扫描操作

0 = 中断不会阻止NVM访问

bit 2 **未实现:** 读为0

bit 1-0 **MODE<1:0>:** 存储器访问模式位<sup>(5)</sup>

11 = 触发模式

10 = 窥探模式

01 = 突发模式

00 = 并发模式

注 1: 设置EN = 0 (SCANCON0寄存器) 不会影响任何其他寄存器内容。

2: 该位在LADR > HADR时清零 (并且不会发生数据周期)。

3: 如果INTM = 1, 则在中断响应期间会改写该位 (为零, 但不是清零)。

4: 当对NVM进行访问时, 或当CRC发送就绪信号时, BUSY = 1。

5: 更多详细信息, 请参见表11-1。

6: 当扫描并完成PFM的整个范围时会出现无效的地址; 即, 器件存储器大小为0x4000, SCANHADR = 0x3FFF时, 在最后一次扫描之后, SCANLADR递增至0x4000, 该地址是无效的。

寄存器 11-12:     **SCANLADRH: 扫描低地址高字节寄存器**

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
LADR<15:8> <sup>(1, 2)</sup>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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           **LADR<15:8>:** 扫描起始/当前地址位<sup>(1, 2)</sup>  
                  要取数据的当前地址的高位, 其值在每次存储器取操作时递增。
- 注 1:** 寄存器 SCANLADRH/L 构成一个16位值, 但并不保证原子或异步访问; 应仅在 SCANGO = 0 (SCANCON0 寄存器) 时读取或写入寄存器。
- 2:** 当 SCANGO = 1 (SCANCON0 寄存器) 时, 对该寄存器的写操作会被忽略。

寄存器 11-13:     **SCANLADRL: 扫描低地址低字节寄存器**

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
LADR<7:0> <sup>(1, 2)</sup>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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           **LADR<7:0>:** 扫描起始/当前地址位<sup>(1, 2)</sup>  
                  要取数据的当前地址的低位, 其值在每次存储器取操作时递增。
- 注 1:** 寄存器 SCANLADRH/L 构成一个16位值, 但并不保证原子或异步访问; 应仅在 SCANGO = 0 (SCANCON0 寄存器) 时读取或写入寄存器。
- 2:** 当 SCANGO = 1 (SCANCON0 寄存器) 时, 对该寄存器的写操作会被忽略。

# PIC16(L)F1615/9

寄存器 11-14:     **SCANHADRH: 扫描高地址高字节寄存器**

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
HADR<15:8> <sup>(1, 2)</sup>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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           **HADR<15:8>: 扫描结束地址位<sup>(1, 2)</sup>**  
                  所指定扫描的结束地址的高位

- 注 1:** 寄存器 SCANHADRH/L 构成一个16位值, 但并不保证原子或异步访问; 应仅在 SCANGO = 0 (SCANCON0 寄存器) 时读取或写入寄存器。
- 2:** 当 SCANGO = 1 (SCANCON0 寄存器) 时, 对该寄存器的写操作会被忽略。

寄存器 11-15:     **SCANHADRL: 扫描高地址低字节寄存器**

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
HADR<7:0> <sup>(1, 2)</sup>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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           **HADR<7:0>: 扫描结束地址位<sup>(1, 2)</sup>**  
                  所指定扫描的结束地址的低位

- 注 1:** 寄存器 SCANHADRH/L 构成一个16位值, 但并不保证原子或异步访问; 应仅在 SCANGO = 0 (SCANCON0 寄存器) 时读取或写入寄存器。
- 2:** 当 SCANGO = 1 (SCANCON0 寄存器) 时, 对该寄存器的写操作会被忽略。

**寄存器 11-16: SCANTRIG: 扫描触发选择寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	TSEL<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

**TSEL<3:0>:** 扫描器数据触发输入选择位

1111-1010 = 保留

1001 = SMT2\_Match

1000 = SMT1\_Match

0111 = TMR0\_Overflow

0110 = TMR5\_Overflow

0101 = TMR3\_Overflow

0100 = TMR1\_Overflow

0011 = TMR6\_postscaled

0010 = TMR4\_postscaled

0001 = TMR2\_postscaled

0000 = LFINTOSC

**表11-4: 与CRC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
CRCACCH	ACC<15:8>								150
CRCACCL	ACC<7:0>								150
CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	149
CRCCON1	DLEN<3:0>				PLEN<3:0>				149
CRCDATH	DAT<15:8>								150
CRCDATL	DAT<7:0>								150
CRCSHIFTH	SHIFT<15:8>								151
CRCSHIFTL	SHIFT<7:0>								151
CRCXORH	XOR<15:8>								151
CRCXORL	XOR<7:1>							—	151
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	114
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	109
SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		152
SCANHADRH	HADR<15:8>								154
SCANHADRL	HADR<7:0>								154
SCANLADRH	LADR<15:8>								153
SCANLADRL	LADR<7:0>								153
SCANTRIG					TSEL<3:0>				155

图注: — = 未实现位, 读为0。CRC 模块不使用阴影单元。

\* 提供寄存器信息的页。

12.0 I/O 端口

每个端口都有三个标准工作寄存器。这些寄存器是：

- TRISx寄存器（数据方向）
- PORTx寄存器（读取器件引脚的电平）
- LATx寄存器（输出锁存器）
- INLVLx（输入电平控制）
- ODCONx寄存器（漏极开路）
- SLRCONx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：

- ANSELx（模拟选择）
- WPUx（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 12-1： 每款器件可用的端口

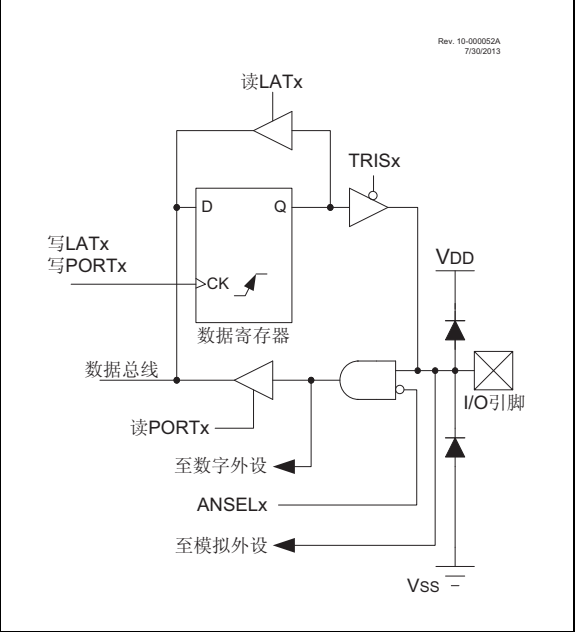
器件	PORTA	PORTB	PORTC
PIC16(L)F1619	•	•	•
PIC16(L)F1615	•		•

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

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

支持模拟输入的端口具有相关的ANSELx寄存器。当某个ANSEL位置1时，与该位相关的数字输入缓冲器会被禁止。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上产生过大的电流。图 12-1 给出了通用 I/O 端口的简化模型，没有给出与其他外设的接口。

图 12-1： 通用 I/O 端口的工作原理





## 12.1 PORTA 寄存器

### 12.1.1 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 12-2）。将 TRISA 某位置 1（= 1）时，会将 PORTA 的相应引脚设为输入（即，禁止输出驱动器）。将 TRISA 某位清零（= 0）时，会将 PORTA 的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选定的引脚）。RA3 是个例外，仅可作为输入引脚，其 TRIS 位总是读为 1。例 12-1 显示了如何初始化 I/O 端口。

读 PORTA 寄存器（寄存器 12-1）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读-修改-写操作。因此，对端口的写操作意味着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）。

### 12.1.2 方向控制

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

### 12.1.3 漏极开路控制

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

### 12.1.4 压摆率控制

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

### 12.1.5 输入阈值控制

INLVLA 寄存器（寄存器 12-8）用于控制每个可用 PORTA 输入引脚的输入阈值电压。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。如果使能该功能，则输入阈值对于确定 PORTA 寄存器的读取值很重要，同时对于确定发生电平变化中断的临界电压也很重要。关于阈值电压的更多信息，请参见第 35.3 节“直流特性”。

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

### 12.1.6 模拟控制

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

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

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

#### 例 12-1: 初始化 PORTA

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

## 12.1.7 PORTA 功能和输出优先级

在复位之后，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进行选择。更多信息，请参见第13.0节“[外设引脚选择（PPS）模块](#)”。外设引脚选择列表中未列出模拟输入功能，例如ADC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。

## 12.2 寄存器定义：PORTA

**寄存器 12-1: 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							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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<sup>(1)</sup>  
1 = 端口引脚电平  $\geq V_{IH}$   
0 = 端口引脚电平  $\leq V_{IL}$

**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 12-2: TRISA: PORTA 三态寄存器**

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0
bit 7							bit 0

**图注:**

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

bit 7-6      **未实现:** 读为0  
bit 5-4      **TRISA<5:4>:** PORTA 三态控制位  
1 = PORTA 引脚被配置为输入 (三态)  
0 = PORTA 引脚被配置为输出  
bit 3      **未实现:** 读为1  
bit 2-0      **TRISA<2:0>:** PORTA 三态控制位  
1 = PORTA 引脚被配置为输入 (三态)  
0 = PORTA 引脚被配置为输出

**注 1:** 未实现, 读为1。

# PIC16(L)F1615/9

寄存器 12-3: LATA: PORTA 数据锁存器寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LATA5	LATA4	LATA3	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 5-0 LATA<5:0>: RA<5:0> 输出锁存值位<sup>(1)</sup>

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

寄存器 12-4: ANSELA: PORTA 模拟选择寄存器

U-0	U-0	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 ANSA4: 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<sup>(1)</sup>。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<sup>(1)</sup>。数字输入缓冲器被禁止。  
0 = 数字 I/O。引脚被配置为端口或数字特殊功能。

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

**寄存器 12-5: WPUA: 弱上拉PORTA 寄存器**

U-0	U-0	R/W-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<sup>(3)</sup>

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:** 对于WPUA3位, 当MCLRE = 1时, 会在内部使能弱上拉, 但未在此处指出。

**寄存器 12-6: ODCONA: PORTA漏极开路控制寄存器**

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	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

对于各个RA<5:4> 引脚

1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)

0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

bit 3 **未实现:** 读为0

bit 2-0 **ODA<2:0>:** PORTA漏极开路使能位

对于各个RA<2:0> 引脚

1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)

0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

# PIC16(L)F1615/9

寄存器 12-7: SLRCONA: PORTA 压摆率控制寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-4 **SLRA<5:4>**: PORTA 压摆率使能位

对于各个 RA<5:4> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率将为最大值

bit 3 未实现: 读为0

bit 2-0 **SLRA<2:0>**: PORTA 压摆率使能位

对于各个 RA<2:0> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率将为最大值

寄存器 12-8: INLVLA: PORTA 输入电平控制寄存器

U-0	U-0	R/W-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

对于各个 RA<5:0> 引脚

1 = 对于端口读操作和电平变化中断, 使用 ST 输入

0 = 对于端口读操作和电平变化中断, 使用 TTL 输入

表12-2: 与PORTA相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	160
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	162
LATA	—	—	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	160
ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	161
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			230
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	159
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	162
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	161

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, 读为1。

表12-3: 与PORTA相关的配置字汇总

名称	Bit	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>		—	69
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	—	—	FOSC<2:0>			

图注: — = 未实现位, 读为0。PORTA不使用阴影单元。

## 12.3 PORTB 寄存器 (仅限 PIC16(L)F1619)

### 12.3.1 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 12-10)。将 TRISB 某位置 1 (= 1) 时, 会将 PORTB 的相应引脚设为输入 (即, 禁止输出驱动器)。将 TRISB 某位清零 (= 0) 时, 会将 PORTB 的相应引脚设为输出 (即, 使能输出驱动器并将输出锁存器中的内容输出到选定的引脚)。例 12-1 显示了如何初始化 I/O 端口。

读 PORTB 寄存器 (寄存器 12-9) 将读出相应引脚的状态, 而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读-修改-写操作。因此, 对端口的写操作意味着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 (LATB)。

### 12.3.2 方向控制

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

### 12.3.3 漏极开路控制

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

### 12.3.4 压摆率控制

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

### 12.3.5 输入阈值控制

INLVLB 寄存器 (寄存器 12-16) 用于控制每个可用 PORTB 输入引脚的输入阈值电压。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。如果使能该功能, 则输入阈值对于确定 PORTB 寄存器的读取值很重要, 同时对于确定发生电平变化中断的临界电压也很重要。关于阈值电压的更多信息, 请参见第 35.3 节“直流特性”。

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

### 12.3.6 模拟控制

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

ANSELB 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, 得到的结果可能与预期不符。

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



## 12.3.7 PORTB 功能和输出优先级

在复位之后，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进行选择。更多信息，请参见第13.0节“[外设引脚选择（PPS）模块](#)”。外设引脚选择列表中未列出模拟输入功能，例如ADC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。

# PIC16(L)F1615/9

## 12.4 寄存器定义：PORTB

寄存器 12-9: PORTB: PORTB 寄存器

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

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **RB<7:4>: PORTB I/O 值位<sup>(1)</sup>**

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 12-10: TRISB: PORTB 三态寄存器

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

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **TRISB<7:4>: PORTB 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**

## 寄存器 12-11: LATB: PORTB 数据锁存器寄存器

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

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

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

bit 3-0 **未实现**: 读为0

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

## 寄存器 12-12: ANSELB: PORTB 模拟选择寄存器

U-0	U-0	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
—	—	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 7-6 **未实现**: 读为0

bit 5-4 **ANSB<5:4>**: 将 RB<5:4> 引脚选择为模拟或数字功能

1 = 模拟输入。引脚被配置为模拟输入<sup>(1)</sup>。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 3-0 **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。

# PIC16(L)F1615/9

## 寄存器 12-13: WPUB: 弱上拉PORTB 寄存器<sup>(1),(2)</sup>

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 12-14: ODCONB: PORTB漏极开路控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
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

对于各个RB<7:4> 引脚

1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)

0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

bit 3-0 **未实现**: 读为0

**寄存器 12-15: SLRCONB: PORTB 压摆率控制寄存器**

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

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 **SLRB<7:4>: PORTB 压摆率使能位**

对于各个 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 12-16: 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**

对于各个 RB<7:4> 引脚

1 = 对于端口读操作和电平变化中断, 使用 ST 输入

0 = 对于端口读操作和电平变化中断, 使用 TTL 输入

bit 3-0 **未实现: 读为0**

**表 12-4: 与 PORTB 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	167
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	169
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	167
ODCONB	ODB7	ODB6	ODB5	ODB4	—	—	—	—	168
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			230
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	166
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	169
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	168

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

**注 1:** 未实现, 读为1。

# PIC16(L)F1615/9

表 12-5: 与 PORTB 相关的配置字汇总

名称	Bit	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>		—	69
	7:0	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE<1:0>		FOSC<2:0>			

图注： — = 未实现位，读为 0。PORTB 不使用阴影单元。

## 12.5 PORTC 寄存器

### 12.5.1 数据寄存器

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

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

### 12.5.2 方向控制

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

### 12.5.3 漏极开路控制

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

### 12.5.4 压摆率控制

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

### 12.5.5 输入阈值控制

INLVLC 寄存器（寄存器 12-24）用于控制每个可用 PORTC 输入引脚的输入阈值电压。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。如果使能该功能，则输入阈值对于确定 PORTC 寄存器的读取值很重要，同时对于确定发生电平变化中断的临界电压也很重要。关于阈值电压的更多信息，请参见第 35.3 节“直流特性”。

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

### 12.5.6 模拟控制

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

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，得到的结果可能与预期不符。

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

### 12.5.7 高驱动能力引脚

HIDRVC 寄存器（寄存器 12-25）控制 RC4 和 RC5 引脚上的高驱动能力选项。当 HIDRVC 位清零时，引脚具有正常驱动能力。当 HIDRVC 位置 1 时，相应引脚的灌电流或拉电流最高可达 100 mA。

## 12.5.8 PORTC 功能和输出优先级

在复位之后，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进行选择。更多信息，请参见第13.0节“[外设引脚选择（PPS）模块](#)”。外设引脚选择列表中未列出模拟输入功能，例如ADC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。



12.6 寄存器定义：PORTC

寄存器 12-17:     **PORTC: PORTC 寄存器**

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

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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 I/O 值位<sup>(1,2)</sup>**  
                  1 = 端口引脚电平 ≥ V<sub>IH</sub>  
                  0 = 端口引脚电平 ≤ V<sub>IL</sub>

- 注 1:  仅限PIC16(L)F1619上的RC<7:6>。  
2:  写入PORTC时，实际上会写入相应的LATC寄存器。读取PORTC寄存器时，将返回实际的I/O引脚值。

寄存器 12-18:     **TRISC: PORTC 三态寄存器**

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

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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<sup>(1)</sup>**  
                  1 = PORTC 引脚配置为输入（三态）  
                  0 = PORTC 引脚配置为输出

- 注 1:  仅限PIC16(L)F1619上的TRISC<7:6>。

# PIC16(L)F1615/9

## 寄存器 12-19: LATC: PORTC 数据锁存器寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATC7 <sup>(1)</sup>	LATC6 <sup>(1)</sup>	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>**: RC<7:0> 输出锁存器位<sup>(1)</sup>

1 = PORTC 引脚配置为输入 (三态)

0 = PORTC 引脚配置为输出

注 1: 仅限 PIC16(L)F1619 上的 LATC<7:6>。

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

## 寄存器 12-20: 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 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	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<sup>(1)</sup>

1 = 模拟输入。引脚被分配为模拟输入引脚<sup>(2)</sup>。禁止数字输入缓冲区。

0 = 数字 I/O。引脚被分配为端口或是特殊数字功能。

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 = 模拟输入。引脚被分配为模拟输入引脚<sup>(2)</sup>。禁止数字输入缓冲区。

0 = 数字 I/O。引脚被分配为端口或是特殊数字功能。

注 1: 仅限 PIC16(L)F1619 上的 ANSC<7:6>。

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

## 寄存器 12-21: WPUC: 弱上拉PORTC寄存器<sup>(2),(3)</sup>

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 <sup>(1)</sup>	WPUC6 <sup>(1)</sup>	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<sup>(1)</sup>

1 = 使能上拉

0 = 禁止上拉

注 1: 仅限PIC16(L)F1619上的WPUC<7:6>。

2: 必须清零OPTION\_REG寄存器的全局WPUEN位, 从而使能各个上拉功能。

3: 如果引脚配置为输出, 则弱上拉器件将自动禁止。

## 寄存器 12-22: ODCONC: PORTC漏极开路控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ODC7 <sup>(1)</sup>	ODC6 <sup>(1)</sup>	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<sup>(1)</sup>

对于各个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: 仅限PIC16(L)F1619上的ODC<7:6>。

# PIC16(L)F1615/9

寄存器 12-23: **SLRCONC: PORTC 压摆率控制寄存器**

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRC7 <sup>(1)</sup>	SLRC6 <sup>(1)</sup>	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<sup>(1)</sup>**

对于各个 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: 仅限 PIC16(L)F1619 上的 SLRC<7:6>。

寄存器 12-24: **INLVLC: PORTC 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	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
INLVLC7 <sup>(1)</sup>	INLVLC6 <sup>(1)</sup>	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<sup>(1)</sup>**

对于各个 RC<7:0> 引脚

1 = 对于端口读操作和电平变化中断使用 ST 输入

0 = 对于端口读操作和电平变化中断使用 TTL 输入

注 1: 仅限 PIC16(L)F1619 上的 INLVLC<7:6>。

**寄存器 12-25: HIDRVC: PORTC 高驱动能力寄存器**

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      **HIDC5:** 端口C5的高电流驱动使能  
1 = 使能高电流驱动  
0 = 禁止高电流驱动
- bit 4      **HIDC4:** 端口C4的高电流驱动使能  
1 = 使能高电流驱动  
0 = 禁止高电流驱动
- bit 3-0      **未实现:** 读为0

**表 12-6: 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
ANSEL	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
HIDRVC	—	—	HIDC5	HIDC4	—	—	—	—	177
INLVLC	INLVLC7 <sup>(1)</sup>	INLVLC6 <sup>(1)</sup>	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	176
LATC	LATC7 <sup>(1)</sup>	LATC6 <sup>(1)</sup>	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	174
ODCONC	ODC7 <sup>(1)</sup>	ODC6 <sup>(1)</sup>	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	175
PORTC	RC7 <sup>(1)</sup>	RC6 <sup>(1)</sup>	RC5	RC4	RC3	RC2	RC1	RC0	173
SLRCONC	SLRC7 <sup>(1)</sup>	SLRC6 <sup>(1)</sup>	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	176
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
WPUC	WPUC7 <sup>(1)</sup>	WPUC6 <sup>(1)</sup>	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	175

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

**注 1:** 仅限PIC16(L)F1619

13.0 外设引脚选择（PPS）模块

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

13.1 PPS输入

每个外设均具有一个用于选择外设输入的PPS寄存器。输入包括器件引脚。

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。如果某个引脚还具有关联的模拟功能，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13-1所示。

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

13.2 PPS输出

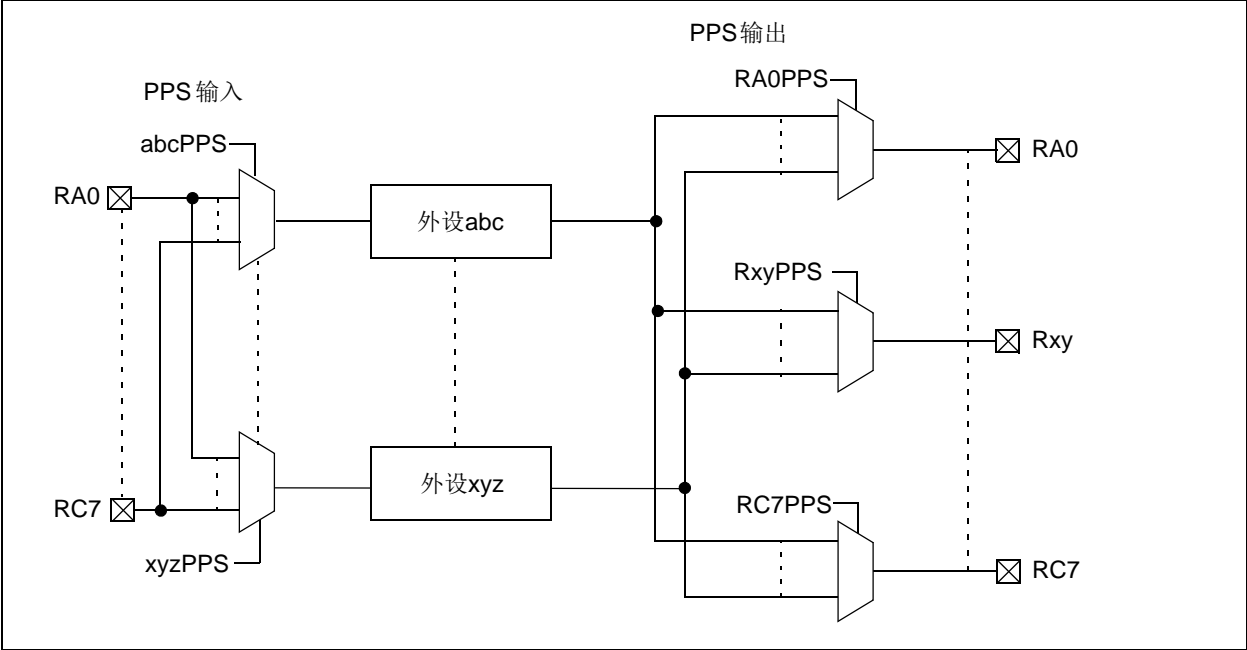
每个I/O引脚均具有一个用于选择引脚输出源的PPS寄存器。除了少数例外情况，与该引脚关联的端口TRIS控制将保持对引脚输出驱动器的控制权。作为外设操作一部分，控制引脚输出驱动器的外设将根据需要改写TRIS控制。这些外设包括：

- EUSART（同步操作）
- MSSP（I<sup>2</sup>C）
- CWG（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13-2所示。

注： 符号“Rxy”是引脚标识符的占位符。例如，RA0PPS。

图13-1： 简化PPS框图



13.3 双向引脚

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

- EUSART（同步操作）
- MSSP（I<sup>2</sup>C）

注： I<sup>2</sup>C 默认输入引脚与 I<sup>2</sup>C 和 SMBus 兼容，并且是器件上惟一具有这种兼容性的引脚。

13.4 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 13-1 给出了置 1 和清零 PPSLOCKED 位的示例。

例 13-1: PPS 锁定/解锁序列

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

13.5 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。在完成所有选择之后将 PPSLOCKED 位置 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。

13.6 休眠期间的操作

PPS 输入和输出选择不会受休眠影响。

13.7 复位的影响

器件上电复位（POR）会将所有 PPS 输入和输出选择清除为其默认值。所有其他复位会将选择保留不变。表 13-1 列出了默认的输入选择。

# PIC16(L)F1615/9

## 13.8 寄存器定义：PPS输入选择

寄存器 13-1: **xxxPPS**: 外设 xxx 输入选择

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于外设

bit 7-5 未实现: 读为0

bit 4-3 **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 (仅限 PIC16(L)F1619)  
00 = 外设输入为 PORTA

bit 2-0 **xxxPPS<2:0>**: 外设 xxx 输入位选择位<sup>(1)</sup>  
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: 关于复位值, 请参见表 13-1。

寄存器 13-2: **RxyPPS**: 引脚 Rxy 输出源选择寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-5 未实现: 读为0

bit 4-0 **RxyPPS<4:0>**: 引脚 Rxy 输出源选择位  
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 13-2。



寄存器 13-3:      PPSLOCK: PPS 锁定寄存器

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

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

bit 7-1      未实现: 读为0

bit 0      **PPSLOCKED:** PPS 锁定位

            1 = PPS 已锁定。不能更改 PPS 选择。

            0 = PPS 未锁定。可以更改 PPS 选择。

# PIC16(L)F1615/9

**表13-1: 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)F1619	PIC16(L)F1615	PIC16(L)F1619	PIC16(L)F1615
电平变化中断	INTPPS	RA2	RA2	00010	00010
定时器0时钟	T0CKIPPS	RA2	RA2	00010	00010
定时器1时钟	T1CKIPPS	RA5	RA5	00101	00101
定时器1门控	T1GPPS	RA4	RA4	00100	00100
定时器2时钟	T2CKIPPS	RA5	RA5	0101	0101
定时器3时钟	T3CKIPPS	RC5	RC5	10101	10101
定时器3门控	T3GPPS	RC4	RC4	10100	10100
定时器4时钟	T4CKIPPS	RC1	RC1	10001	10001
定时器5时钟	T5CKIPPS	RC0	RC0	10000	10000
定时器5门控	T5GPPS	RC3	RC3	10011	10011
定时器6时钟	T6CKIPPS	RA3	RA3	00011	00011
CCP1	CCP1PPS	RC5	RC5	10101	10101
CCP2	CCP2PPS	RC3	RC3	10011	10011
CWG1	CWG1INPPS	RA2	RA2	00010	00010
SPI和I <sup>2</sup> C时钟	SSPCLKPPS	RB6	RC0	01110	10000
SPI和I <sup>2</sup> 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
SMT1窗口输入	SMTWIN1PPS	RA5	RA5	00101	00101
SMT1信号输入	SMTSIG1PPS	RA4	RA4	00100	00100
SMT2窗口输入	SMTWIN2PPS	RA3	RA3	00101	00101
SMT2信号输入	SMTSIG2PPS	RC1	RC1	10001	10001
角度定时器1时钟输入	AT1INPPS	RC5	RC5	10101	10101
角度定时器1 CC1输入	AT1CC1PPS	RC3	RC3	10011	10011
角度定时器1 CC2输入	AT1CC2PPS	RC4	RC4	10100	10100
角度定时器1 CC3输入	AT1CC3PPS	RC5	RC5	10101	10101

示例: CCP1PPS = 0x13表示选择RC3作为CCP1输入。

表 13-2: 外设可用输出端口<sup>(2)</sup>

RxyPPS<4:0>	输出信号	PIC16(L)F1619			PIC16(L)F1615	
		PORTA	PORTB	PORTC	PORTA	PORTC
11xxx	保留	•	•	•	•	•
10111	保留	•	•	•	•	•
10110	保留	•	•	•	•	•
10101	保留	•	•	•	•	•
10100	保留	•	•	•	•	•
10011	DT	•	•	•	•	•
10010	TX/CK	•	•	•	•	•
10001	SDO/SDA <sup>(1)</sup>	•	•	•	•	•
10000	SCK/SCL <sup>(1)</sup>	•	•	•	•	•
01111	PWM4_out	•	•	•	•	•
01110	PWM3_out	•	•	•	•	•
01101	CCP2_out	•	•	•	•	•
01100	CCP1_out	•	•	•	•	•
01011	CWG1OUTD <sup>(1)</sup>	•	•	•	•	•
01010	CWG1OUTC <sup>(1)</sup>	•	•	•	•	•
01001	CWG1OUTB <sup>(1)</sup>	•	•	•	•	•
01000	CWG1OUTA <sup>(1)</sup>	•	•	•	•	•
00111	LC4_out	•	•	•	•	•
00110	LC3_out	•	•	•	•	•
00101	LC2_out	•	•	•	•	•
00100	LC1_out	•	•	•	•	•
00011	ZCD1_out	•	•	•	•	•
00010	sync_C2OUT	•	•	•	•	•
00001	sync_C1OUT	•	•	•	•	•
00000	LATxy	•	•	•	•	•

注 1: TRIS 控制由外设根据需要改写。

2: 不支持的外设将输出 0。

# PIC16(L)F1615/9

表13-3: 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	181
INTPPS	—	—	—	INTPPS<4:0>					180
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					180
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					180
T1GPPS	—	—	—	T1GPPS<4:0>					180
T2CKIPPS	—	—	—	T2CKIPPS<4:0>					180
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					180
T3GPPS	—	—	—	T3GPPS<4:0>					180
T4CKIPPS	—	—	—	T4CKIPPS<4:0>					180
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					180
T5GPPS	—	—	—	T5GPPS<4:0>					180
T6CKIPPS	—	—	—	T6CKIPPS<4:0>					180
CCP1PPS	—	—	—	CCP1PPS<4:0>					180
CCP2PPS	—	—	—	CCP2PPS<4:0>					180
CWG1INPPS	—	—	—	CWG1INPPS<4:0>					180
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					180
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					180
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					180
RXPPS	—	—	—	RXPPS<4:0>					180
CKPPS	—	—	—	CKPPS<4:0>					180
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					180
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					180
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					180
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					180
AT1INPPS	—	—	—	AT1INPPS<4:0>					180
ATCC1PPS	—	—	—	ATCC1PPS<4:0>					180
ATCC2PPS	—	—	—	ATCC2PPS<4:0>					180
ATCC3PPS	—	—	—	ATCC3PPS<4:0>					180
SMT1SIGPPS	—	—	—	SMT1SIGPPS<4:0>					180
SMT1WINPPS	—	—	—	SMT1WINPPS<4:0>					180
SMT2SIGPPS	—	—	—	SMT2SIGPPS<4:0>					180
SMT2WINPPS	—	—	—	SMT2WINPPS<4:0>					180
RA0PPS	—	—	—	RA0PPS<4:0>					180
RA1PPS	—	—	—	RA1PPS<4:0>					180
RA2PPS	—	—	—	RA2PPS<4:0>					180
RA4PPS	—	—	—	RA4PPS<4:0>					180
RA5PPS	—	—	—	RA5PPS<4:0>					180
RB4PPS <sup>(1)</sup>	—	—	—	RB4PPS<4:0>					180
RB5PPS <sup>(1)</sup>	—	—	—	RB5PPS<4:0>					180
RB6PPS <sup>(1)</sup>	—	—	—	RB6PPS<4:0>					180
RB7PPS <sup>(1)</sup>	—	—	—	RB7PPS<4:0>					180
RC0PPS	—	—	—	RC0PPS<4:0>					180

注 1: 仅限PIC16(L)F1619。

表13-3: 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
RC1PPS	—	—	—	RC1PPS<4:0>					<a href="#">180</a>
RC2PPS	—	—	—	RC2PPS<4:0>					<a href="#">180</a>
RC3PPS	—	—	—	RC3PPS<4:0>					<a href="#">180</a>
RC4PPS	—	—	—	RC4PPS<4:0>					<a href="#">180</a>
RC5PPS	—	—	—	RC5PPS<4:0>					<a href="#">180</a>
RC6PPS <sup>(1)</sup>	—	—	—	RC6PPS<4:0>					<a href="#">180</a>
RC7PPS <sup>(1)</sup>	—	—	—	RC7PPS<4:0>					<a href="#">180</a>

注 1: 仅限PIC16(L)F1619。

14.0 电平变化中断

PORTA、PORTB<sup>(1)</sup>和PORTC引脚可以配置为作为电平变化中断（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。任意一个端口引脚或端口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 14-1 给出了 IOC 模块的框图。

**注 1：** PORTB 引脚仅在 PIC16(L)F1619 上可用。

14.1 使能模块

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

14.2 独立的引脚配置

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

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

14.3 中断标志

分别位于 IOCAF、IOCBF 和 IOCCF 寄存器中的 IOCAF<sub>x</sub>、IOCBF<sub>x</sub> 和 IOCCF<sub>x</sub> 位是对应于关联端口的电平变化中断引脚的状态标志。如果在正确使能的引脚上检测到期望的边沿，则对应于该引脚的状态标志会置 1，并且如果 IOCIE 位也置 1，则还会产生中断。INTCON 寄存器的 IOCIF 位会反映所有 IOCAF<sub>x</sub>、IOCBF<sub>x</sub> 和 IOCCF<sub>x</sub> 位的状态。

14.4 清零中断标志

各个状态标志（IOCAF<sub>x</sub>、IOCBF<sub>x</sub> 和 IOCCF<sub>x</sub> 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 14-1： 清零中断标志（以 PORTA 为例）

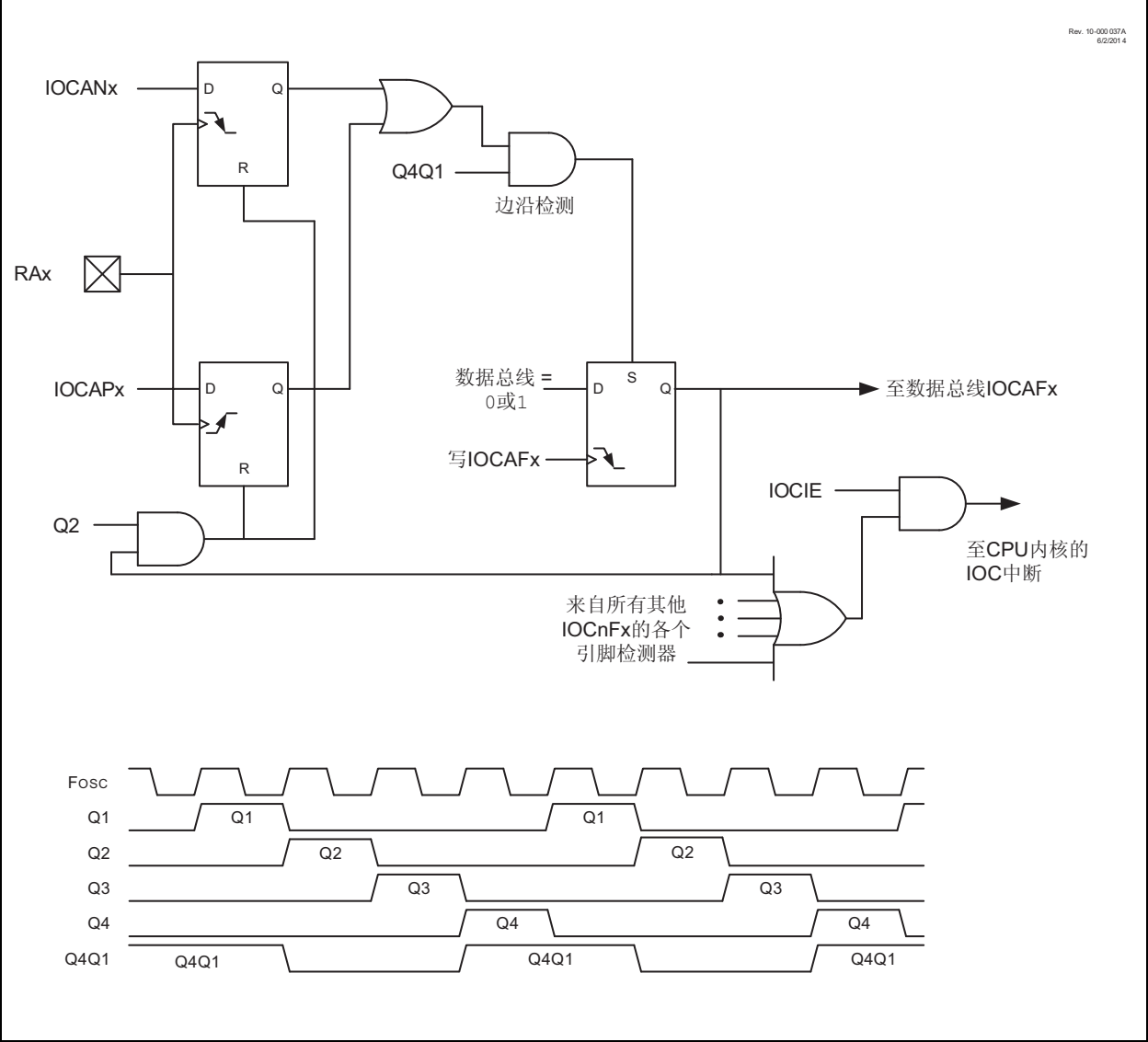
```
MOVLW    0xff
XORWF    IOCAF, W
ANDWF    IOCAF, F
```

14.5 休眠模式下的操作

如果 IOCIE 位置 1，电平变化中断的中断序列会将器件从休眠模式唤醒。

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

图14-1： 电平变化中断框图（以PORTA为例）



# PIC16(L)F1615/9

## 14.6 寄存器定义：电平变化中断控制

寄存器 14-1: IOCAP: 电平变化中断PORTA 正边沿寄存器

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

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

bit 7-6 未实现：读为0  
bit 5-0 IOCAP<5:0>: 电平变化中断PORTA 正边沿允许位  
1 = 在引脚上对于正向边沿允许电平变化中断。IOCAF<sub>x</sub> 位和 IOCIF 标志将在检测到边沿时置1。  
0 = 禁止关联引脚的电平变化中断。

寄存器 14-2: IOCAN: 电平变化中断PORTA 负边沿寄存器

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

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

bit 7-6 未实现：读为0  
bit 5-0 IOCAN<5:0>: 电平变化中断PORTA 负边沿允许位  
1 = 在引脚上对于负向边沿允许电平变化中断。IOCAF<sub>x</sub> 位和 IOCIF 标志将在检测到边沿时置1。  
0 = 禁止关联引脚的电平变化中断。

寄存器 14-3: IOCAF: 电平变化中断PORTA 标志寄存器

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

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

bit 7-6 未实现：读为0  
bit 5-0 IOCAF<5:0>: 电平变化中断PORTA 标志位  
1 = 在关联引脚上检测到使能的电平变化。  
在 IOCAP<sub>x</sub> = 1，并在 RAX 上检测到上升沿时置1，或者在 IOCAN<sub>x</sub> = 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 14-4: IOCBP: 电平变化中断PORTB 正边沿寄存器<sup>(1)</sup>

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

### 图注:

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

bit 7-4 **IOCBP<7:4>:** 电平变化中断PORTB 正边沿允许位  
 1 = 在引脚上对于正向边沿允许电平变化中断。IOCBFx 位和 IOCIF 标志将在检测到边沿时置1。  
 0 = 禁止关联引脚的电平变化中断。

bit 3-0 **未实现:** 读为0

注 1: 仅限PIC16(L)F1619。

## 寄存器 14-5: IOCBN: 电平变化中断PORTB 负边沿寄存器<sup>(1)</sup>

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

### 图注:

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

bit 7-4 **IOCBN<7:4>:** 电平变化中断PORTB 负边沿允许位  
 1 = 在引脚上对于负向边沿允许电平变化中断。IOCBFx 位和 IOCIF 标志将在检测到边沿时置1。  
 0 = 禁止关联引脚的电平变化中断。

bit 3-0 **未实现:** 读为0

注 1: 仅限PIC16(L)F1619。

## 寄存器 14-6: IOCBF: 电平变化中断PORTB 标志寄存器<sup>(1)</sup>

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

### 图注:

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

bit 7-4 **IOCBF<7:4>:** 电平变化中断PORTB 标志位  
 1 = 在关联引脚上检测到使能的电平变化。  
 在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)F1619。

# PIC16(L)F1615/9

## 寄存器 14-7: 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 <sup>(1)</sup>	IOCCP6 <sup>(1)</sup>	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正边沿允许位<sup>(1)</sup>

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

0 = 禁止关联引脚的电平变化中断。

注 1: 仅PIC16(L)F1619上提供IOCCP<7:6>。

## 寄存器 14-8: IOCCN: 电平变化中断PORTC负边沿寄存器<sup>(1)</sup>

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 <sup>(1)</sup>	IOCCN6 <sup>(1)</sup>	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负边沿允许位<sup>(1)</sup>

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

0 = 禁止关联引脚的电平变化中断。

注 1: 仅PIC16(L)F1619上提供IOCCN<7:6>。

## 寄存器 14-9: IOCCF: 电平变化中断PORTC标志寄存器<sup>(1)</sup>

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
IOCCF <sup>(1)</sup>	IOCCF6 <sup>(1)</sup>	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<sup>(1)</sup>

1 = 在关联引脚上检测到使能的电平变化。

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

0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

注 1: 仅PIC16(L)F1619上提供IOCCF<7:6>。

**表14-1: 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	160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	188
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	188
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	188
IOCBF <sup>(2)</sup>	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	189
IOCBN <sup>(2)</sup>	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	189
IOCBP <sup>(2)</sup>	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	189
IOCCF	IOCCF7 <sup>(2)</sup>	IOCCF6 <sup>(2)</sup>	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	190
IOCCN	IOCCN7 <sup>(2)</sup>	IOCCN6 <sup>(2)</sup>	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	190
IOCCP	IOCCP7 <sup>(2)</sup>	IOCCP6 <sup>(2)</sup>	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	190
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159
TRISC	TRISC7 <sup>(2)</sup>	TRISC7 <sup>(2)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173

**图注:** — = 未实现位，读为0。电平变化中断不使用阴影单元。

**注 1:** 未实现，读为1。

**2:** 仅限PIC16(L)F1619。

## 15.0 固定参考电压（FVR）

固定参考电压（FVR）是稳定的固定参考电压，独立于VDD，输出电压标称值（ $V_{FVR}$ ）为1.024V。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
- 比较器的同相输入
- 比较器的反相输入

FVR可以通过将FVRCON寄存器的FVREN位置1来使能。

### 15.1 独立的增益放大器

送到外设（如上所列）的FVR输出会经过一个可编程增益放大器。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17.0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19.0节“比较器模块”。

要最大程度降低禁止FVR时的电流消耗，应通过清零缓冲区增益选择位来关闭FVR缓冲区。

### 15.2 FVR稳定周期

当固定参考电压模块使能时，参考电压和放大电路需要一段时间才能达到稳定。在电路稳定下来、可供使用时，FVRCON寄存器的FVRRDY位将会置1。请参见图36-21。

图15-1： 参考电压框图

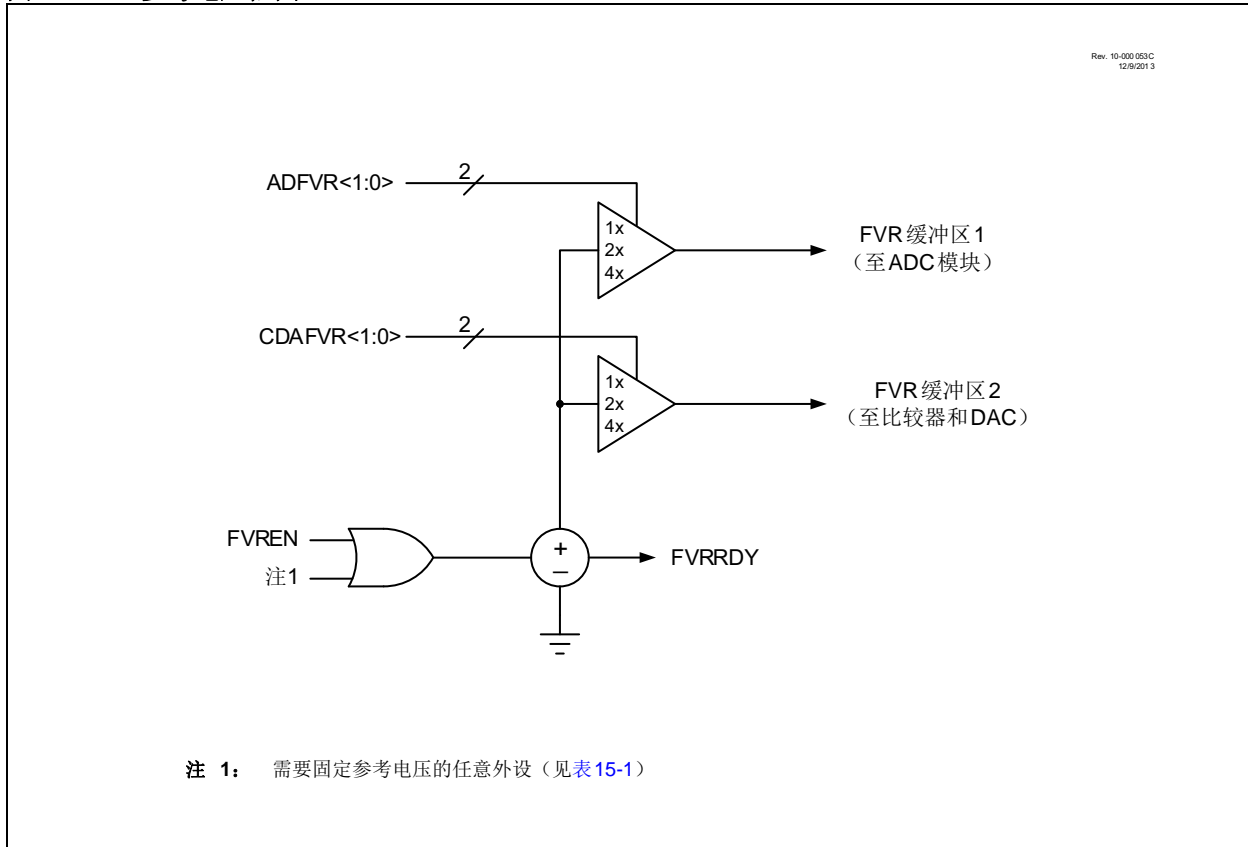


表15-1： 需要固定参考电压（FVR）的外设

外设	条件	说明
HFINTOSC	FOSC<2:0> = 010 且 IRCF<3:0> = 000x	INTOSC 有效且器件不处于休眠状态。
BOR	BOREN<1:0> = 11	BOR 总是使能。
	BOREN<1:0> = 10 且 BORFS = 1	BOR 在休眠模式下被禁止，BOR 快速启动使能。
	BOREN<1:0> = 01 且 BORFS = 1	BOR 受软件控制，BOR 快速启动使能。
LDO	当 VREGPM = 1 且不处于休眠模式时， 所有 PIC16F1615/9 器件	处于休眠模式时，器件会运行低功耗稳压器。

# PIC16(L)F1615/9

## 15.3 寄存器定义：FVR 控制

寄存器 15-1: 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 <sup>(1)</sup>	FVRRDY <sup>(2)</sup>	TSEN <sup>(3)</sup>	TSRNG <sup>(3)</sup>	CDAFVR<1:0> <sup>(1)</sup>		ADFVR<1:0> <sup>(1)</sup>	
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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<sup>(1)</sup>  
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<sup>(2)</sup>  
1 = 固定参考电压输出就绪备用  
0 = 固定参考电压输出未就绪或未使能
- bit 5 **TSEN**: 温度指示器使能位<sup>(3)</sup>  
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<sup>(3)</sup>  
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 缓冲区增益选择位<sup>(1)</sup>  
11 = 比较器 FVR 缓冲区增益为 4x, 输出  $V_{CDAFVR} = 4x V_{FVR}$ <sup>(4)</sup>  
10 = 比较器 FVR 缓冲区增益为 2x, 输出  $V_{CDAFVR} = 2x V_{FVR}$ <sup>(4)</sup>  
01 = 比较器 FVR 缓冲区增益为 1x, 输出  $V_{CDAFVR} = 1x V_{FVR}$   
00 = 比较器 FVR 缓冲区关闭
- bit 1-0 **ADFVR<1:0>**: ADC FVR 缓冲区增益选择位<sup>(1)</sup>  
11 = ADC FVR 缓冲区增益为 4x, 输出  $V_{ADFVR} = 4x V_{FVR}$ <sup>(4)</sup>  
10 = ADC FVR 缓冲区增益为 2x, 输出  $V_{ADFVR} = 2x V_{FVR}$ <sup>(4)</sup>  
01 = ADC FVR 缓冲区增益为 1x, 输出  $V_{ADFVR} = 1x V_{FVR}$   
00 = ADC FVR 缓冲区关闭

注 1: 要最大程度降低禁止 FVR 时的电流消耗, 应通过清零缓冲区增益选择位来关闭 FVR 缓冲区。

2: 对于 PIC16LF1615/9 器件, FVRRDY 总是为 1。

3: 更多信息, 请参见第 16.0 节“温度指示器模块”。

4: 固定参考电压输出不能超出  $V_{DD}$ 。

表 15-2: 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>		194

图注: 固定参考电压模块不使用阴影单元。

16.0 温度指示器模块

本器件系列配备了用于测量硅裸片工作温度的温度电路。电路的工作温度范围介于-40°C和+85°C之间。其输出是与器件温度成比例的电压。温度指示器的输出在内部与器件ADC连接。

电路可以用作温度阈值检测器，也可以用作更精确的温度指示器，这取决于所执行的校准级别。执行单点校准时，电路可以指示邻近该点的温度。执行双点校准时，电路可以更精确地检测整个温度量程。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与校准》（DS01333A\_CN）。

16.1 电路工作原理

图16-1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16-1描述了温度指示器的输出特性。

公式16-1: VOUT 范围

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

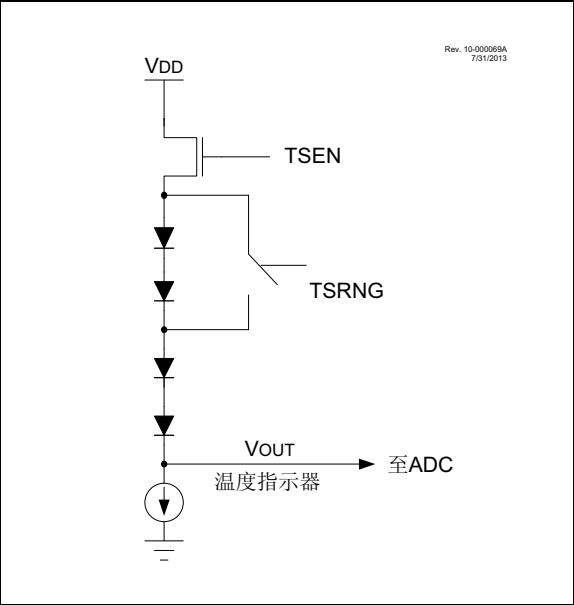
温度检测电路集成了固定参考电压（FVR）模块。更多信息，请参见第15.0节“固定参考电压（FVR）”。

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，但各器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VDD。

低电压范围的选择方式是将FVRCON寄存器的TSRNG位清零。低电压范围产生的压降较小，所以只需较低的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16-1: 温度指示器电路图



16.2 最小工作电压VDD

当温度电路工作于低电压范围时，器件可以在规范范围内的任意工作电压下工作。

当温度电路工作于高电压范围时，器件工作电压VDD必须足够高，以确保正确地偏置温度电路。

表16-1给出了建议的最小VDD与范围设置。

表16-1: 建议的VDD与范围

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

16.3 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17.0节“模数转换器（ADC）模块”。

16.4 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 16-2: 与温度指示器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器 所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		118

图注: 温度指示器模块不使用阴影单元。



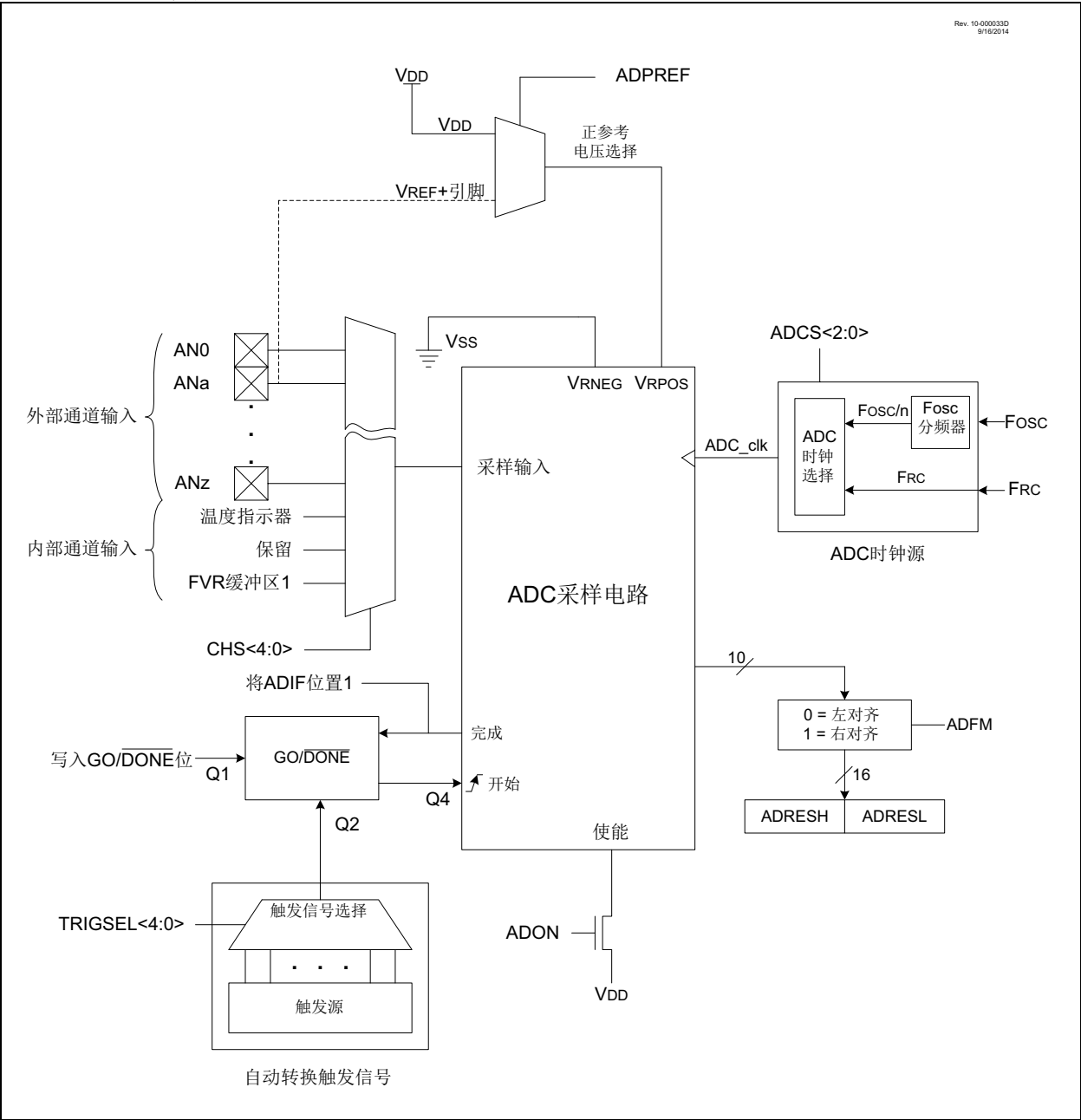
17.0 模数转换器（ADC）模块

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17-1给出了ADC的框图。

可通过软件方式选择内部产生的电压或外部提供的电压作为ADC参考电压。

ADC可在转换完成时产生中断。该中断可用于将器件从休眠状态唤醒。

图17-1： ADC框图



## 17.1 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

### 17.1.1 端口配置

ADC可用于将模拟信号转换为数字信号。转换模拟信号时，应通过设置相关的TRIS和ANSEL位将I/O引脚配置为模拟。更多信息，请参见第12.0节“[I/O端口](#)”。

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

### 17.1.2 通道选择

有最多15个通道选择可供使用：

- AN<11:0>引脚（仅限PIC16(L)F1619）
- AN<7:0>引脚（仅限PIC16(L)F1615）
- 温度指示器
- DAC1输出
- FVR缓冲区1

ADCON0寄存器的CHS位决定与采样保持电路相连接的通道。

当改变通道时，在开始下一次转换前需要一段延时（TACQ）。更多信息，请参见第17.2.6节“[ADC转换步骤](#)”。

### 17.1.3 ADC参考电压

ADC模块使用正参考电压和负参考电压。正参考电压标记为ref+，负参考电压标记为ref-。

正参考电压（ref+）通过ADCON1寄存器中的ADPREF位进行选择。正参考电压源可以是：

- VREF+引脚
- VDD
- FVR缓冲区1

负参考电压（ref-）源是：

- Vss

### 17.1.4 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17-2所示。

为正确转换，必须满足合适的TAD规范。更多信息，请参见第35.0节“[电气规范](#)”中的ADC转换要求。表17-1给出了适当的ADC时钟选择的示例。

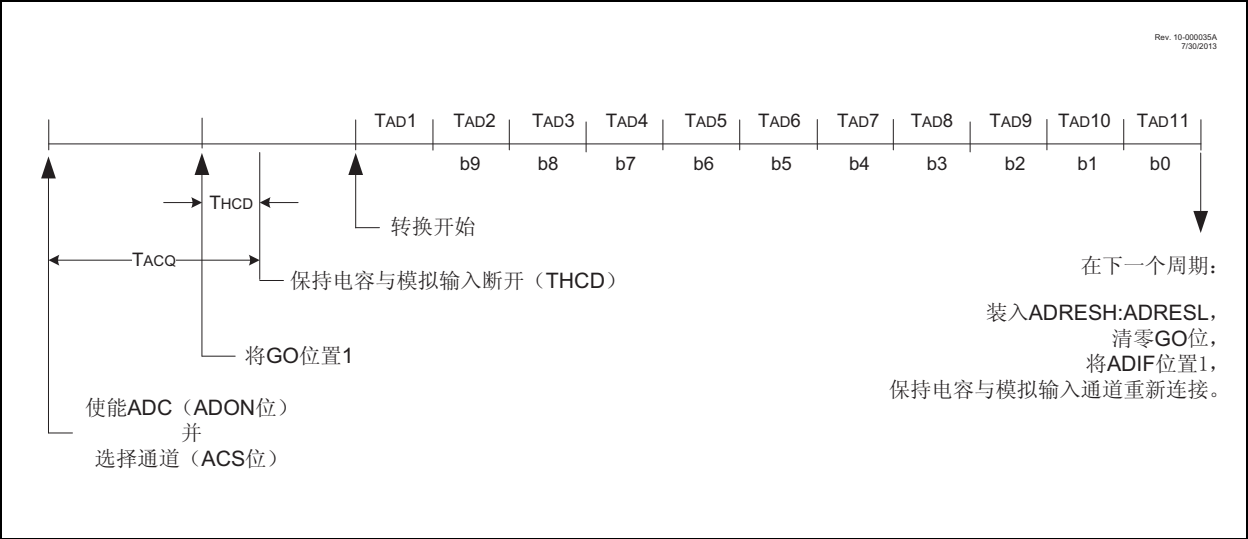
**注：** 除非使用FRC，否则系统时钟频率的任何改变都会改变ADC时钟频率，这会影响ADC结果。

表17-1： 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>	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	100 ns	125 ns	250 ns	500 ns	2.0 μs
Fosc/4	100	200 ns	250 ns	500 ns	1.0 μs	4.0 μs
Fosc/8	001	400 ns	500 ns	1.0 μs	2.0 μs	8.0 μs
Fosc/16	101	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs
Fosc/32	010	1.6 μs	2.0 μs	4.0 μs	8.0 μs	32.0 μs
Fosc/64	110	3.2 μs	4.0 μs	8.0 μs	16.0 μs	64.0 μs
FRC	x11	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs

- 图注： 阴影单元表示超出了建议范围。
- 注 1： FRC时钟源的典型TAD时间为1.7 ms。
- 2： 当器件频率高于1 MHz时，仅当在休眠状态下进行转换时才推荐使用FRC时钟源。
- 3： 当使用FRC时钟源时，TAD周期可以处于规定范围内（见TAD参数）。当使用基于Fosc的时钟源时，TAD周期可以配置为更精确的TAD周期。但是，如果要在器件处于休眠模式时执行转换，则必须使用FRC时钟源。

图17-2： 模数转换TAD周期



17.1.5 中断

ADC 模块可在模数转换完成时产生中断。ADC 中断标志位是 PIR1 寄存器中的 ADIF 位。ADC 中断允许位是 PIE1 寄存器中的 ADIE 位。ADIF 位必须用软件清零。

- 注 1:** ADIF 位在每次转换完成时置 1，与是否允许 ADC 中断无关。

**2:** 仅当选择了 FRC 振荡器时，ADC 才能在休眠模式下工作。

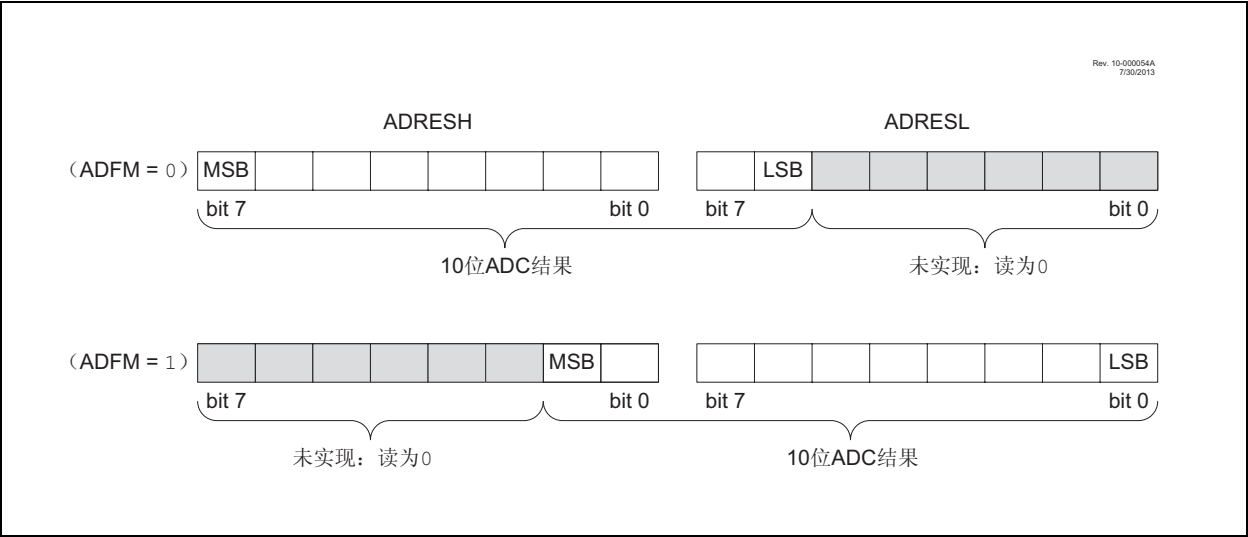
器件工作或休眠时都可产生该中断。如果器件处于休眠状态，该中断会唤醒器件。从休眠状态唤醒时，总是执行紧跟 SLEEP 指令后的下一条指令。如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止 INTCON 寄存器的 GIE 和 PEIE 位。如果使能了 INTCON 寄存器的 GIE 和 PEIE 位，执行将切换到中断服务程序。

17.1.6 结果格式

10 位 ADC 转换结果可以两种格式提供：左对齐或右对齐。ADCON1 寄存器的 ADFM 位控制输出格式。

图 17-3 给出了两种输出格式。

图 17-3: 10 位 ADC 转换结果格式



17.2 ADC工作原理

17.2.1 启动转换

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

**注：** 不应在启动ADC的同一条指令中将GO/DONE位置1。请参见第17.2.6节“ADC转换步骤”。

17.2.2 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

17.2.3 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。

17.2.4 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。在休眠期间执行ADC转换可以降低系统噪声。如果允许了ADC中断，转换完成时器件将从休眠状态唤醒。如果禁止了ADC中断，尽管ADON位仍保持置1，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。

17.2.5 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17-2。

表17-2: 自动转换源

转换源外设	信号名称
Timer0	T0_overflow
Timer1	T1_overflow
Timer2	TMR2_postscaled
Timer4	TMR4_postscaled
Timer6	TMR6_postscaled
比较器C1	C1_OUT_sync
比较器C2	C2_OUT_sync
SMT1	SMT1_CPW
SMT1	SMT1_CPR
SMT1	SMT1_PR
SMT2	SMT2_CPW
SMT2	SMT2_CPR
SMT2	SMT2_PR
CCP1	CCP1_out
CCP2	CCP2_out

17.2.6 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<sup>(1)</sup>
- 4. 等待所需采集时间<sup>(2)</sup>。
- 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）。

注 1： 如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止全局中断。

2： 请参见第17.4节“ADC采集要求”。

例17-1： 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     ;Acquisiton delay
BSF        ADCON0,ADGO    ;Start conversion
BTFSC     ADCON0,ADGO     ;Is conversion done?
GOTO      $-1             ;No, test again
BANKSEL    ADRESH         ;
MOVF      ADRESH,W        ;Read upper 2 bits
MOVWF     RESULTHI        ;store in GPR space
BANKSEL    ADRESL         ;
MOVF      ADRESL,W        ;Read lower 8 bits
MOVWF     RESULTLO        ;Store in GPR space
```

17.3 寄存器定义：ADC控制

寄存器 17-1:      **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

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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	<b>未实现:</b> 读为0
bit 6-2	<b>CHS&lt;4:0&gt;:</b> 模拟通道选择位 11111 = FVR (固定参考电压) 缓冲区1输出 <sup>(3)</sup> 11110 = DAC (数模转换器) <sup>(2)</sup> 11101 = 温度指示器 <sup>(1)</sup> 11100 = 保留。不连接任何通道 • • • 01100 = 保留。不连接任何通道 01011 = AN11 <sup>(4)</sup> 01010 = AN10 <sup>(4)</sup> 01001 = AN9 <sup>(4)</sup> 01000 = AN8 <sup>(4)</sup> 00111 = 保留。不连接任何通道 00110 = 保留。不连接任何通道 00101 = 保留。不连接任何通道 00100 = 保留。不连接任何通道 01000 = 保留。不连接任何通道 00111 = AN7 <sup>(4)</sup> 00110 = AN6 <sup>(4)</sup> 00101 = AN5 <sup>(4)</sup> 00100 = AN4 <sup>(4)</sup> 00011 = AN3 00010 = AN2 00001 = AN1 00000 = AN0
bit 1	<b>GO/DONE:</b> 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转换完成后, 该位由硬件自动清零。 0 = ADC转换已完成/未进行
bit 0	<b>ADON:</b> ADC使能位 1 = 使能ADC 0 = 禁止ADC, 不消耗工作电流

注 1: 请参见第16.0节“温度指示器模块”。

2: 更多信息, 请参见第18.0节“8位数模转换器 (DAC1) 模块”。

3: 更多信息, 请参见第15.0节“固定参考电压 (FVR)”。

4: AN<11:4> 仅在PIC16(L)F1619上可用。

# PIC16(L)F1615/9

寄存器 17-2:      **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	U-0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	—	ADPREF<1:0>	
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-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 3-2      **未实现: 读为 0**

bit 1-0      **ADPREF<1:0>: ADC 正参考电压配置位**  
11 = VRPOS 连接到内部固定参考电压 (FVR)  
10 = VRPOS 连接到外部 VREF+ 引脚<sup>(1)</sup>  
01 = 保留  
00 = VRPOS 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 35-13。



寄存器 17-3:      **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> <sup>(1)</sup>					—	—	—
bit 7					bit 0		

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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>: 自动转换触发器选择位<sup>(1)</sup>

11111 = 保留

•

•

•

10101 = 保留

10100 = AT1\_cmp3

10011 = AT1\_cmp2

10010 = AT1\_cmp1

10001 = CLC4OUT

10000 = CLC3OUT

01111 = CLC2OUT

01110 = CLC1OUT

01101 = TMR5\_overflow

01100 = TMR3\_overflow

01011 = SMT2\_match

01010 = SMT1\_match

01001 = TMR6\_postscaled

01000 = TMR4\_postscaled

00111 = C2\_OUT\_sync

00110 = C1\_OUT\_sync

00101 = TMR2\_postscaled

00100 = T1\_overflow<sup>(2)</sup>

00011 = T0\_overflow<sup>(2)</sup>

00010 = CCP2\_out

00001 = CCP1\_out

00000 = 未选择任何自动转换触发器

bit 2-0

未实现: 读为0

注 1: 这是所有触发源的上升沿敏感输入。

2: 信号还会将其相应的中断标志置1。

# PIC16(L)F1615/9

寄存器 17-4:      **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

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 17-5:      **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

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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。**

寄存器 17-6:      **ADRESH: ADC结果寄存器高字节 (ADRESH) 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<9:8>	
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 17-7:      **ADRESL: ADC结果寄存器低字节 (ADRESL) ADFM = 1**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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

## 17.4 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平。模拟输入模型如图17-4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17-4。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17-1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17-1： 采集时间示例

假设： 温度 = 50°C， 外部阻抗为(10 kΩ)， VDD为5.0V

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

Tc 值可以用以下公式近似计算：

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

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

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

注： 其中n = ADC 的位数。

求解TC：

$$\begin{aligned} T_C &= -CHOLD(RIC + RSS + RS) \ln(1/2047) \\ &= -12.5 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0004885) \\ &= 1.12 \mu s \end{aligned}$$

因此：

$$\begin{aligned} T_{ACQ} &= 2 \mu s + 1.12 \mu s + [(50^\circ\text{C} - 25^\circ\text{C}) (0.05 \mu s/^\circ\text{C})] \\ &= 4.37 \mu s \end{aligned}$$

**注 1：** 因为参考电压（VRPOS）自行抵消，因此它对该公式没有影响。

**2：** 充电保持电容（CHOLD）在每次转换后不会放电。

**3：** 模拟信号源的最大阻抗推荐值为10 kΩ。此要求是为了符合引脚泄漏电流规范。

图17-4： 模拟输入模型

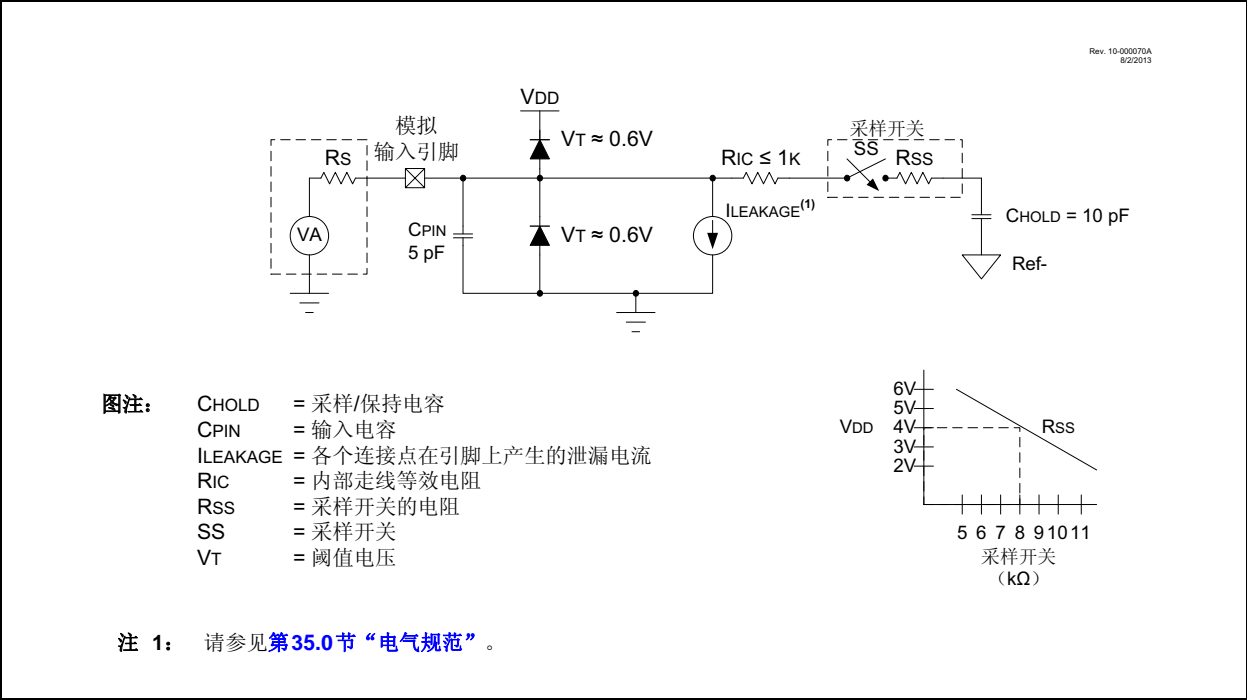
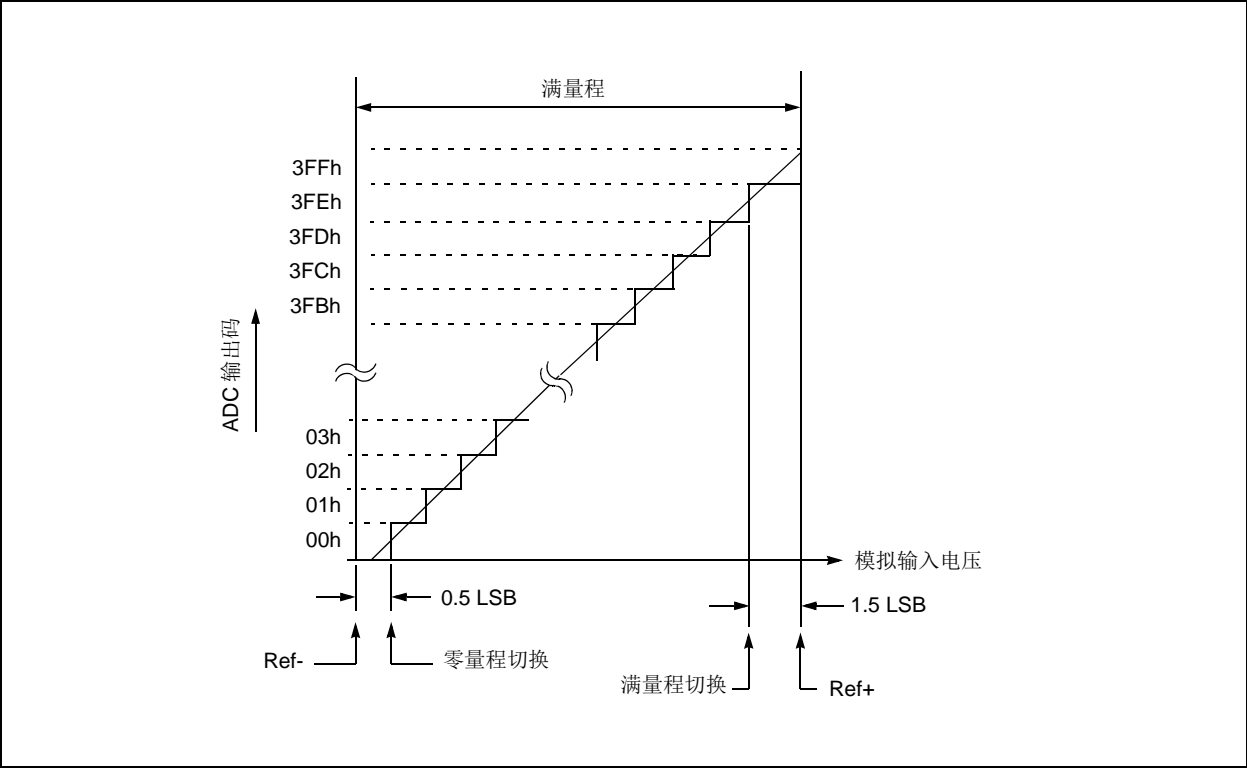


图17-5： ADC传递函数



# PIC16(L)F1615/9

表17-3: 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	203
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		204
ADCON2	TRIGSEL<4:0>					—	—	—	205
ADRESH	ADC结果寄存器的高字节								206和207
ADRESL	ADC结果寄存器的低字节								206和207
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	160
ANSELC	ANSC7 <sup>(2)</sup>	ANSC6 <sup>(2)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159
TRISC	TRISC7 <sup>(2)</sup>	TRISC6 <sup>(2)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		194

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: 未实现, 读为1。

2: 仅限PIC16(L)F1619。

18.0 8位数模转换器（DAC1）模块

数模转换器提供了一个可变参考电压，它与输入源成比例，具有256个可选输出电压。

DAC的输入可以连接到：

- 外部VREF引脚
- VDD供电电压
- FVR（固定参考电压）

DAC的输出可以配置为向以下对象提供参考电压：

- 比较器的同相输入
- ADC输入通道
- DACxOUT1引脚

数模转换器（DAC）可以通过将DAC1CON0寄存器的DAC1EN位置1来使能。

18.1 输出电压选择

DAC具有256个电平范围。256个电平通过DAC1CON1寄存器的DAC1R<7:0>位进行设置。

DAC输出电压由公式18-1确定：

公式18-1: DAC输出电压

如果  $DAC1EN = 1$

$$V_{OUT} = \left( (V_{SOURCE+} - V_{SOURCE-}) \times \frac{DAC1R[7:0]}{2^8} \right) + V_{SOURCE-}$$

$V_{SOURCE+} = V_{DD}, V_{REF} \text{ 或 } FVR \text{ 缓冲区 } 2$

$V_{SOURCE-} = V_{SS}$

18.2 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。

第35.0节“电气规范”中给出了梯形电阻网络中各个电阻的阻值。

18.3 DAC参考电压输出

通过将DAC1CON0寄存器的DAC1OE1引脚置1，可以将DAC电压输出到DACxOUT1引脚。选择将DAC参考电压输出到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参考电压输出时，读取该引脚将总是返回0。

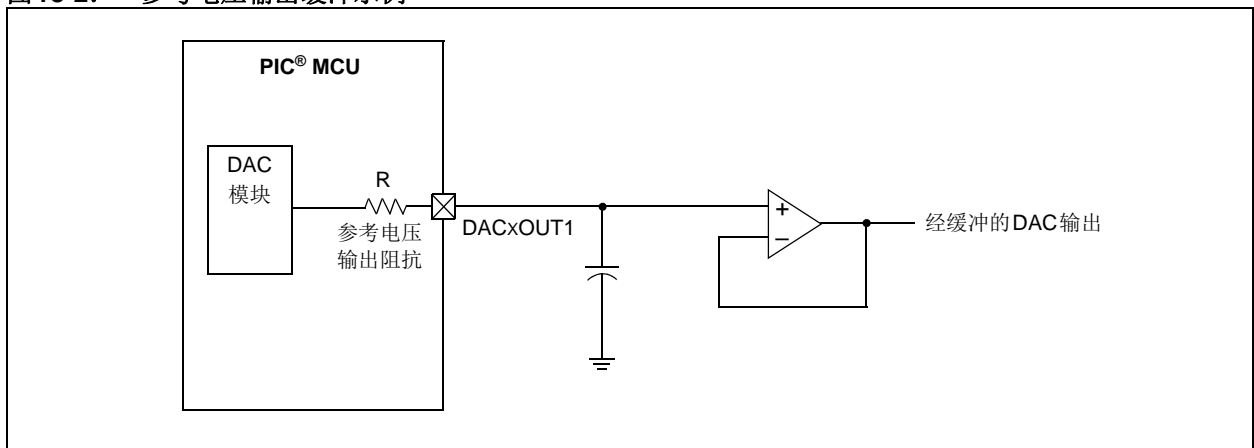
受电流驱动能力的限制，DAC参考电压输出到DACxOUT1引脚必须外接缓冲器。图18-2举例说明了这一缓冲技术。

---

图18-1: 数模转换器框图



图18-2: 参考电压输出缓冲示例





## 18.4 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，DAC1CON0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。

## 18.5 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。
- DAC1R<7:0>范围选择位被清零。

# PIC16(L)F1615/9

## 18.6 寄存器定义：DAC控制

寄存器 18-1: DAC1CON0: 参考电压控制寄存器0

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0
DAC1EN	—	DAC1OE1	—	DAC1PSS<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      **DAC1EN:** DAC1使能位  
1 = 使能DAC  
0 = 禁止DAC
- bit 6      **未实现:** 读为0
- bit 5      **DAC1OE1:** DAC1电压输出1使能位  
1 = DAC电平也从DACxOUT1引脚输出  
0 = DAC电平从DACxOUT1引脚断开
- bit 4      **未实现:** 读为0
- bit 3-2    **DAC1PSS<1:0>:** DAC1正参考电压源选择位  
11 = 保留, 不要使用  
10 = FVR缓冲区2输出  
01 = VREF+引脚  
00 = VDD
- bit 1-0    **未实现:** 读为0

寄存器 18-2: DAC1CON1: 参考电压控制寄存器1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
DAC1R<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    **DAC1R<7:0>:** DAC1电压输出选择位

表18-1: 与DAC1模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		194
DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	214
DAC1CON1	DAC1R<7:0>								214

图注: — = 未实现位, 读为0。DAC模块不使用阴影单元。

19.0 比较器模块

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

- 独立的比较器控制
- 可编程输入选择
- 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
- 从休眠状态唤醒
- 可编程的速度/功耗优化
- PWM关闭
- 可编程和固定参考电压

19.1 比较器概述

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

表 19-1 列出了该器件可用的比较器。

表 19-1： 每款器件可用的比较器

器件	C1	C2
PIC16(L)F1619	•	•
PIC16(L)F1615	•	•

图19-1： 单比较器

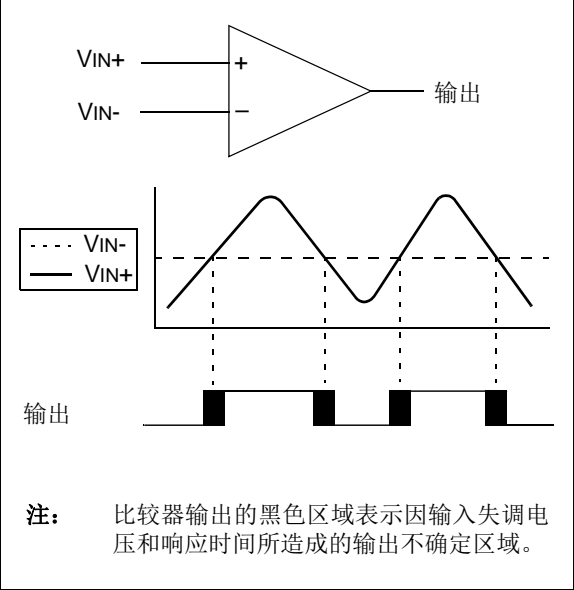
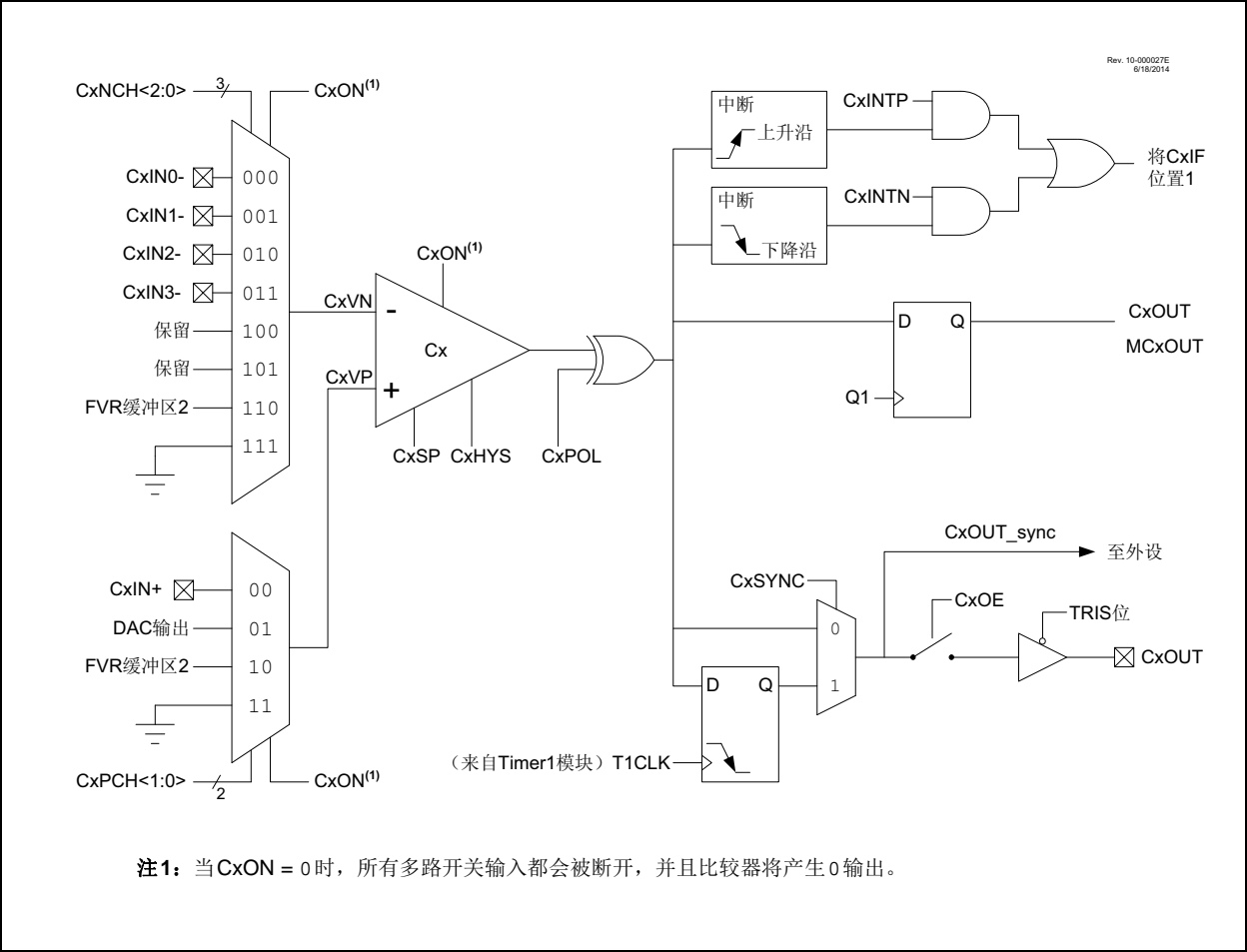


图 19-2: 比较器模块的简化框图



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
- 输出选择
- 输出极性
- 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 CxON 位置 1 可以使能比较器操作。清零 CxON 位可以禁止比较器，以使电流消耗降至最低。

19.2.2 比较器输出选择

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

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

**注 1：** CMxCON0 寄存器的 CxOE 位会改写端口数据锁存器。将 CMxCON0 寄存器的 CxON 位置 1 对端口改写没有影响。

**2：** 比较器的内部输出在每个指令周期被锁存。除非另外指定，否则不锁存外部输出。

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 CxPOL 位置 1 来使比较器输出的极性反相。清零 CxPOL 位得到的是同相的输出信号。

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

19.2.4 比较器速度/功耗选择

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

## 19.3 比较器滞后

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

更多信息，请参见第35.0节“电气规范”。

## 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22.5节“Timer1 门控”。该功能可用于对模拟事件的持续时间或间隔时间进行计时。

建议将比较器输出与Timer1进行同步。这可以确保在比较器中发生变化时，Timer1不会递增。

### 19.4.1 比较器输出同步

通过将CMxCON0寄存器的CxSYNC位置1，可以使比较器的输出与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CxINTP和/或CxINTN位），则相应的中断标志位（PIR2寄存器的CxIF位）会置1。

要允许中断，必须将以下位置1：

- CMxCON0寄存器的CxON、CxPOL和CxSP位
- PIE2寄存器的CxIE位
- CMxCON1寄存器的CxINTP位（对于上升沿检测）
- CMxCON1寄存器的CxINTN位（对于下降沿检测）
- INTCON寄存器的PEIE和GIE位

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CxPOL位更改输出极性来产生中断，或者通过使用CMxCON0寄存器的CxON位开启或关闭比较器来产生中断。

## 19.6 比较器同相输入选择

通过配置CMxCON1寄存器的CxPCH<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
- DAC输出
- FVR（固定参考电压）
- Vss（地）

关于固定参考电压模块的更多信息，请参见第15.0节“固定参考电压（FVR）”。

关于DAC输入信号的更多信息，请参见第18.0节“8位数模转换器（DAC1）模块”。

每当禁止比较器（CxON = 0）时，所有比较器输入都会被禁止。

## 19.7 比较器反相输入选择

CMxCON1寄存器的CxNCH<2:0>位指示模拟输入引脚或模拟地连接到比较器的反相输入：

- CxIN- 引脚
- 模拟地
- FVR缓冲区2

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，请参见第35.0节“电气规范”中的比较器和参考电压规范。

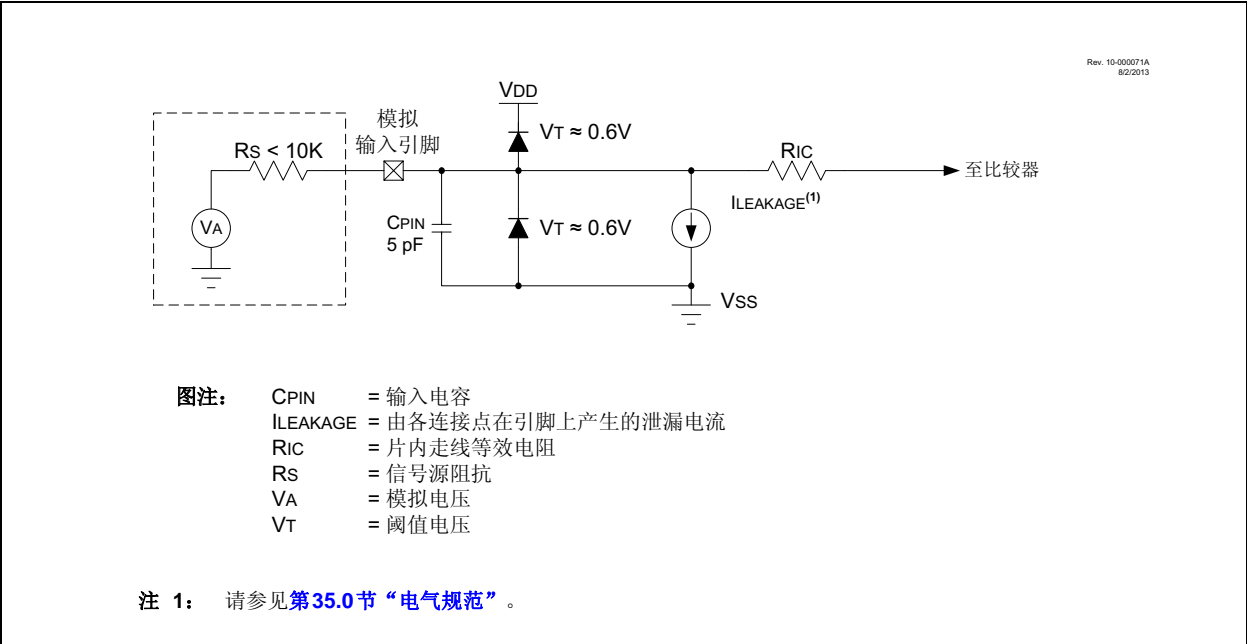
19.9 模拟输入连接注意事项

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

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

- 注 1： 当读端口寄存器时，所有配置为模拟输入的引脚都读为0。配置为数字输入的引脚将按照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19-3： 模拟输入模型



# PIC16(L)F1615/9

## 19.10 寄存器定义：比较器控制

寄存器 19-1: CMxCON0: 比较器 Cx 控制寄存器 0

R/W-0/0	R-0/0	U/U-0/0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7	<b>CxON:</b> 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	<b>CxOUT:</b> 比较器输出位 如果 $CxPOL = 1$ (极性反相): 1 = $CxVP < CxVN$ 0 = $CxVP > CxVN$ 如果 $CxPOL = 0$ (极性不反相): 1 = $CxVP > CxVN$ 0 = $CxVP < CxVN$
bit 5	未实现: 读为0
bit 4	<b>CxPOL:</b> 比较器输出极性选择位 1 = 比较器输出反相 0 = 比较器输出不反相
bit 3	未实现: 读为0
bit 2	<b>CxSP:</b> 比较器速度/功耗选择位 1 = 比较器工作在正常功耗、高速模式下 0 = 比较器工作在低功耗、低速模式下
bit 1	<b>CxHYS:</b> 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	<b>CxSYNC:</b> 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**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>	—		CxNCH<2:0>		
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7 **CxINTP:** 比较器正向边沿中断允许位  
1 = 在 CxOUT 位的正向边沿, CxIF 中断标志将置1  
0 = 在 CxOUT 位的正向边沿, CxIF 中断标志不会置1
- bit 6 **CxINTN:** 比较器负向边沿中断允许位  
1 = 在 CxOUT 位的负向边沿, CxIF 中断标志将置1  
0 = 在 CxOUT 位的负向边沿, CxIF 中断标志不会置1
- bit 5-4 **CxPCH<1:0>:** 比较器同相输入通道选择位  
11 = CxVP 连接到 AGND  
10 = CxVP 连接到 FVR 缓冲区 2  
01 = CxVP 连接到 VDAC  
00 = CxVP 连接到 CxIN+ 引脚
- bit 3 **未实现:** 读为0
- bit 2-0 **CxNCH<2:0>:** 比较器反相输入通道选择位  
111 = CxVN 连接到 AGND  
110 = CxVN 不连接, 输入悬空  
101 = CxVN 不连接, 输入悬空  
100 = 保留  
011 = CxVN 连接到 CxIN3- 引脚<sup>(1)</sup>  
010 = CxVN 连接到 CxIN2- 引脚<sup>(1)</sup>  
001 = CxVN 连接到 CxIN1- 引脚  
000 = CxVN 连接到 CxIN0- 引脚

**寄存器 19-3: CMOUT: 比较器输出寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7-2 **未实现:** 读为0
- bit 1 **MC2OUT:** C2OUT 的镜像副本位
- bit 0 **MC1OUT:** C1OUT 的镜像副本位

# PIC16(L)F1615/9

表 19-3: 与比较器模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	160
CM1CON0	C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC	220
CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			221
CM2CON0	C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC	220
CM2CON1	C2INTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			221
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	221
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		194
DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	214
DAC1CON1	DAC1R<7:0>								214
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	107
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF	112
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159
TRISC <sup>(2)</sup>	TRISC7 <sup>(2)</sup>	TRISC6 <sup>(2)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173

图注: — = 未实现位, 读为0。比较器模块不使用阴影单元。

注 1: 未实现, 读为1。

2: 仅限 PIC16F1619。

20.0 过零检测（ZCD）模块

ZCD 模块用于检测交流信号何时越过地电位。实际的过零阈值为过零参考电压  $V_{CPINV}$ ，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，该模块会产生灌电流。当施加的电压小于参考电压时，该模块会产生拉电流。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进行串联。该电阻的阻抗和额定值取决于外部源峰值电压。在通过电阻的电流标称值为 300  $\mu A$ ，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

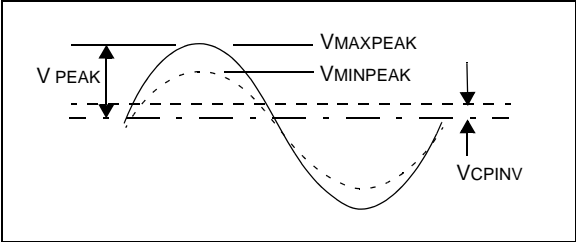
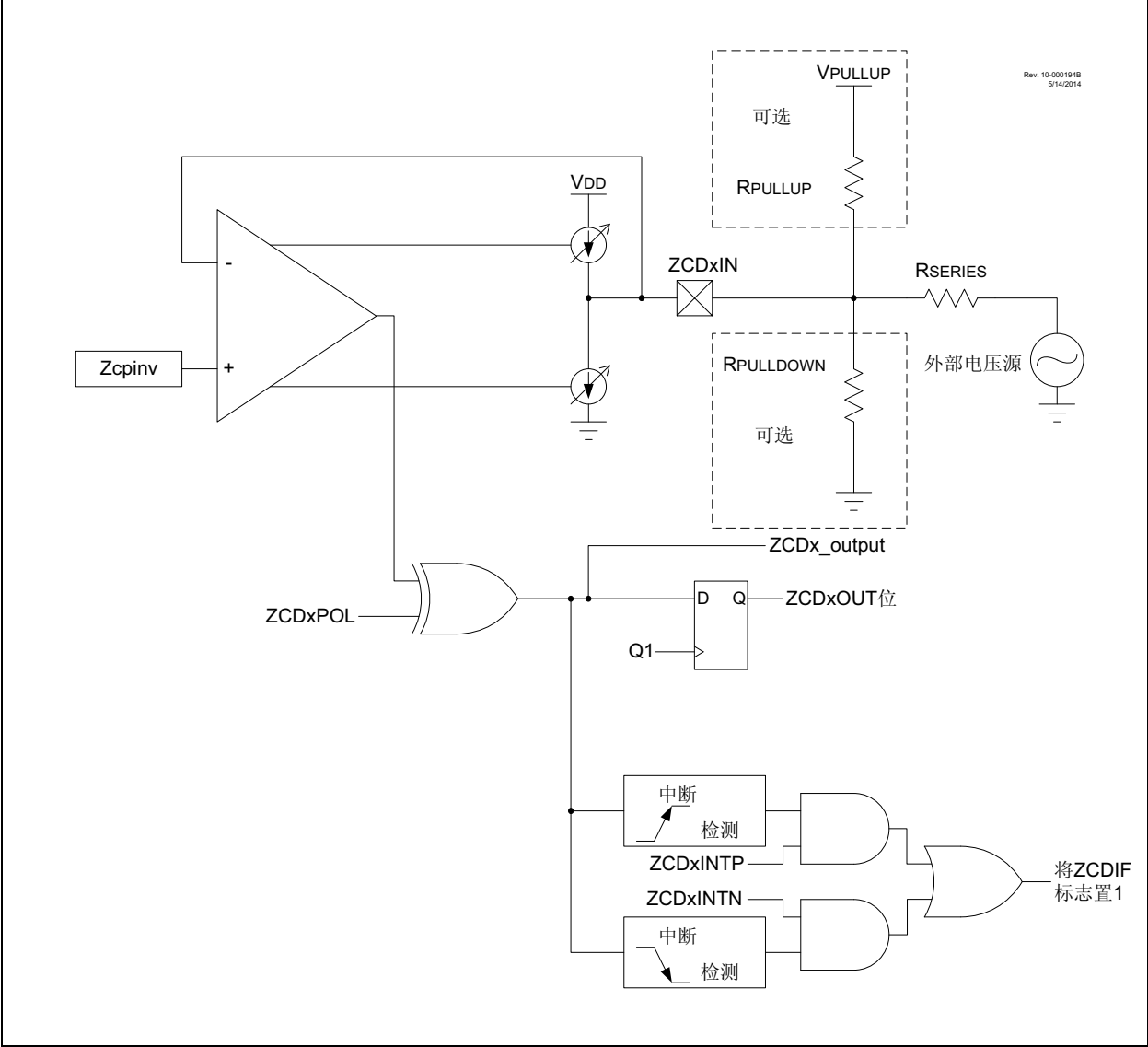


图 20-2: 简化ZCD框图



## 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，ZCDxCON寄存器的ZCDxOUT位会置1；当处于工作状态的是拉电流时，该位会清零。ZCDxOUT位会受极性位影响。

## 20.3 ZCD逻辑极性

ZCDxCON寄存器的ZCDxPOL位用于将ZCDxOUT位相对于拉电流和灌电流输出进行反相。当ZCDxPOL位置1时，ZCDxOUT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。

ZCDxPOL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，且其关联的使能位置1时，PIR3寄存器的ZCDIF位会置1。ZCDxINTP位用于允许上升沿中断，ZCDxINTN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ZCDxINTP位（用于上升沿检测）
- ZCDxCON寄存器的ZCDxINTN位（用于下降沿检测）
- INTCON寄存器的PEIE和GIE位

更改ZCDxPOL位将导致中断，无论ZCDxEN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VCPINV偏移

ZCD发生切换的实际电压是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Vss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VDD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Vss时：

$$T_{OFFSET} = \frac{\arcsin\left(\frac{V_{cpinv}}{V_{PEAK}}\right)}{2\pi \cdot \text{频率}}$$

当外部电压源相对于VDD时：

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

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Vss变化时使用上拉电阻。在电压相对于VDD变化时使用下拉电阻。该电阻会增大ZCD引脚的偏置电压，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VCPINV切换电压。上拉或下拉值可以使用公式20-3或公式20-4所示的公式确定。

### 公式20-3： ZCD上拉/下拉

当外部信号相对于Vss时：

$$R_{PULLUP} = \frac{R_{SERIES}(V_{PULLUP} - V_{cpinv})}{V_{cpinv}}$$

当外部信号相对于VDD时：

$$R_{PULLDOWN} = \frac{R_{SERIES}(V_{cpinv})}{(V_{DD} - V_{cpinv})}$$

VCPINV 的很小变化都会显著影响上拉和下拉电阻值。测量 VCPINV 是很困难的，特别是在波形是相对于 VDD 时。但是，通过结合 [公式 20-2](#) 和 [公式 20-3](#)，可以根据 ZCDx\_output 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  $\Delta T$  等于  $4 \cdot T_{OFFSET}$ 。[公式 20-4](#) 给出了根据高电平和低电平 ZCDx\_output 周期确定上拉和下拉电阻值的公式。通过将 ZCDxOE 位置 1，可以直接在 ZCDxOUT 引脚上观察 ZCDx\_output 信号。

**公式 20-4:**

$$R = R_{SERIES} \left( \frac{V_{BIAS}}{V_{PEAK} \left( \sin \left( \pi Freq \frac{(\Delta T)}{2} \right) \right)} - 1 \right)$$

R 为上拉或下拉电阻。

VBIAS 为 VPULLUP（R 为上拉电阻时）或 VDD（R 为下拉电阻时）。

$\Delta T$  为 ZCDxOUT 高电平和低电平周期的差值。

## 20.6 处理 VPEAK 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 拉电流和灌电流保持低于  $\pm 600 \mu A$  的最大设计范围，并高于合理的最小范围。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  $\pm 600 \mu A$ ，最小值至少为  $\pm 100 \mu A$ ，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**

$$R_{SERIES} = \frac{V_{MAXPEAK} + V_{MINPEAK}}{7 \times 10^{-4}}$$

## 20.7 休眠期间的操作

ZCD 电流源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 ZCD 配置位清零时，ZCD 电路将在 POR 时处于工作状态。当 ZCD 配置位置 1 时，必须通过将 ZCDxCON 寄存器的 ZCDxEN 位置 1 来使能 ZCD 模块。

## 20.9 寄存器定义：ZCD控制

**寄存器 20-1: ZCDxCON: 过零检测控制寄存器**

R/W-q/q	U-0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ZCDxEN	—	ZCDxOUT	ZCDxPOL	—	—	ZCDxINTP	ZCDxINTN
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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	<b>ZCDxEN:</b> 过零检测使能位 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	<b>未实现:</b> 读为0
bit 5	<b>ZCDxOUT:</b> 过零检测逻辑电平位 <u>ZCDxPOL位 = 0:</u> 1 = ZCD引脚产生灌电流 0 = ZCD引脚产生拉电流 <u>ZCDxPOL位 = 1:</u> 1 = ZCD引脚产生拉电流 0 = ZCD引脚产生灌电流
bit 4	<b>ZCDxPOL:</b> 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	<b>未实现:</b> 读为0
bit 1	<b>ZCDxINTP:</b> 过零正边沿中断允许位 1 = 在ZCDx_output从低电平变为高电平时, ZCDIF位置1 0 = 在ZCDx_output从低电平变为高电平时, ZCDIF位不受影响
bit 0	<b>ZCDxINTN:</b> 过零负边沿中断允许位 1 = 在ZCDx_output从高电平变为低电平时, ZCDIF位置1 0 = 在ZCDx_output从高电平变为低电平时, ZCDIF位不受影响

**表 20-1: 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	—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	<a href="#">108</a>
PIR3	—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	<a href="#">113</a>
ZCD1CON	ZCD1EN	—	ZCD1OUT	ZCD1POL	—	—	ZCD1INTP	ZCD1INTN	<a href="#">227</a>

图注: — = 未实现, 读为0。ZCD模块不使用阴影单元。

**表 20-2: 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	71
	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）
- 3位预分频器（独立于看门狗定时器）
- 可编程内部或外部时钟源
- 可编程外部时钟边沿选择
- 上溢时产生中断
- TMR0可用于门控Timer1

图21-1给出了Timer0模块的框图。

### 21.1 Timer0工作原理

Timer0模块可用作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时，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。

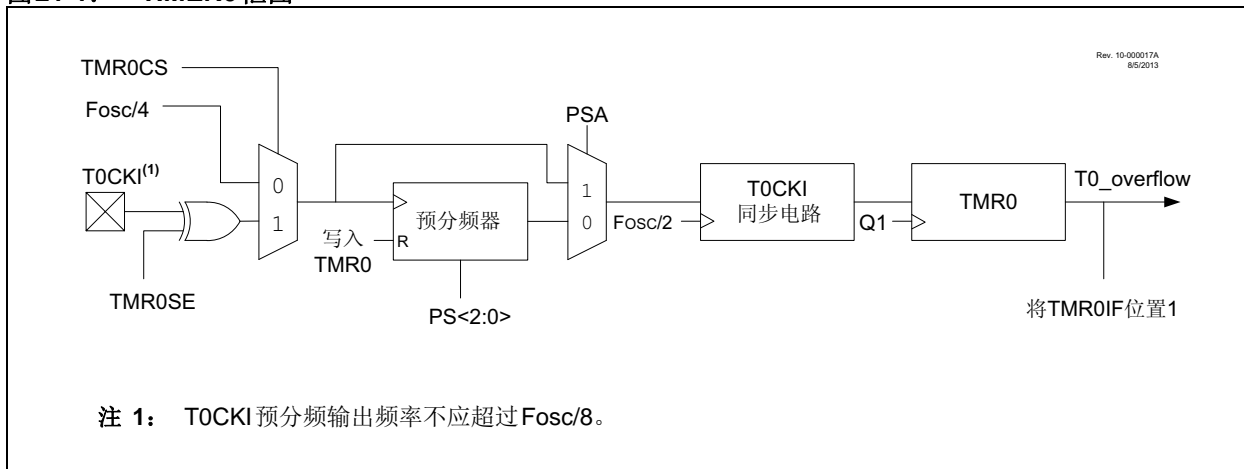
#### 21.1.2 8位计数器模式

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TMR0SE位决定。

图21-1: TIMER0框图





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

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

## 21.1.5 8位同步计数器模式

在8位计数器模式下，T0CKI引脚的递增边沿必须与指令时钟保持同步。同步可通过在指令时钟的Q2和Q4周期对预分频器的输出进行采样实现。外部时钟源的高低电平周期必须满足第35.0节“电气规范”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。

# PIC16(L)F1615/9

## 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	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
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 ( $\overline{\text{MCLR}}$  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 ( $F_{\text{osc}}/4$ )
- 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
ADCON2	TRIGSEL<4:0>					—	—	—	205
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			230
TMR0	8 位Timer0计数的保持寄存器								228*
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159

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, 读为 1。

## 22.0 带门控控制的TIMER1/3/5模块

Timer1/3/5模块是16位定时器/计数器，具有以下特性：

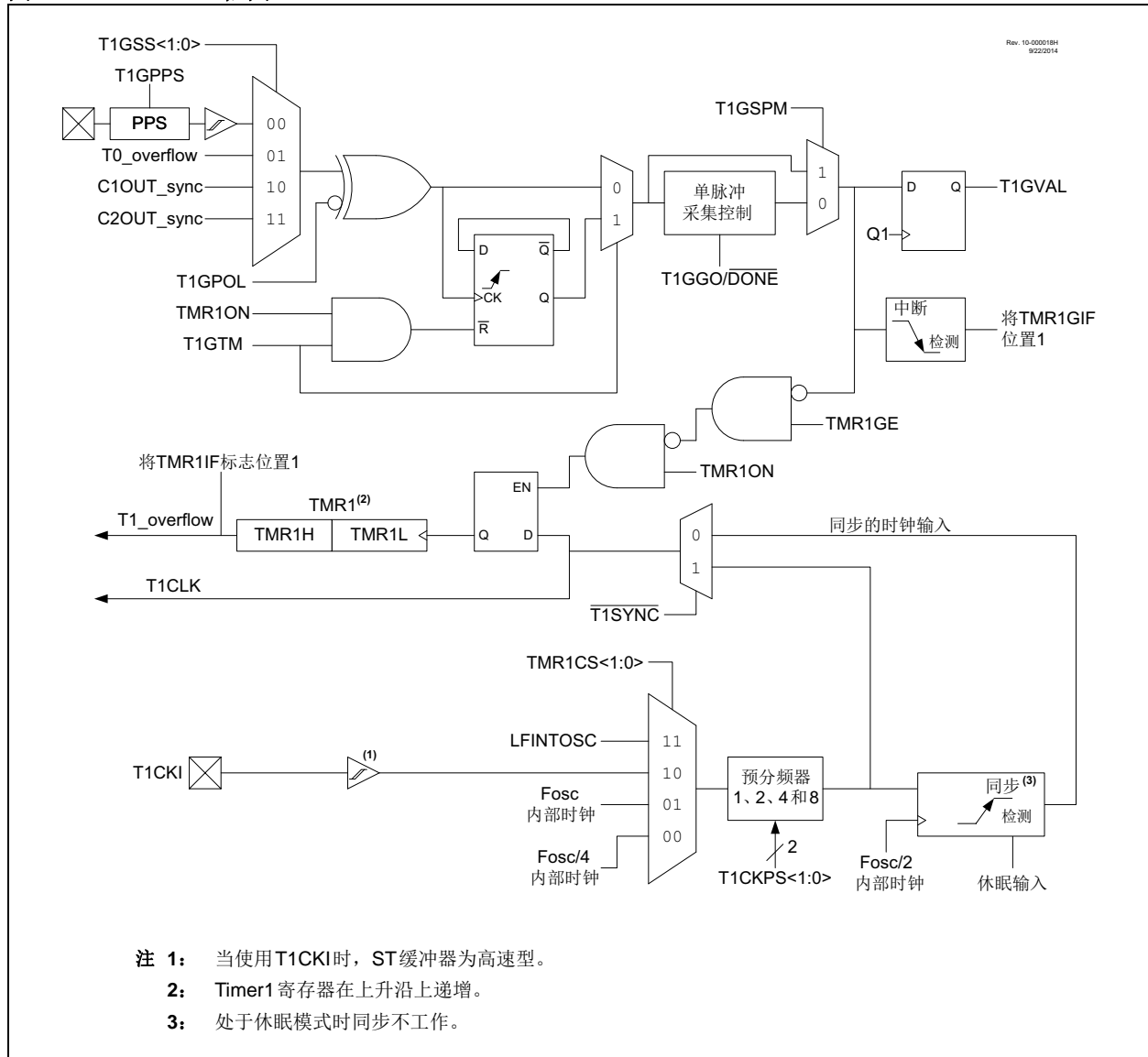
- 16位定时器/计数器寄存器对（TMR1H:TMR1L）
- 可编程内部或外部时钟源
- 2位预分频器
- 可选的同步比较器输出
- 多个Timer1门控（计数使能）源
- 上溢时产生中断
- 上溢触发唤醒（仅限外部时钟，异步模式）
- ADC自动转换触发器

- 可选的门控源极性
- 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。

**注：** 该器件上实现了3个相同的Timer1模块。这些定时器命名为Timer1、Timer3和Timer5。对Timer1及其相关寄存器的所有引用，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 TMR1ON 和 TMR1GE 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 TMR1CS<1:0> 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。

使能 Timer1 计数时，在外部时钟输入 T1CKI 的上升沿递增。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。

**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>	时钟源
11	LFINTOSC
10	T1CKI 引脚上的外部时钟源
01	系统时钟（Fosc）
00	指令时钟（Fosc/4）

## 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 T1CKPS 位控制预分频器计数器。对预分频器计数器不能直接进行读写操作；但是，通过写入 TMR1H 或 TMR1L 可将预分频器计数器清零。

## 22.4 异步计数器模式下的 Timer1 操作

如果 T1CON 寄存器的控制位  $\overline{\text{T1SYNC}}$  置 1，外部时钟输入将不同步。定时器异步于内部相位时钟进行递增计数。如果选择了外部时钟源，在休眠期间定时器将继续运行，并在上溢时产生中断以唤醒处理器。但是，用软件对定时器进行读/写操作时，要特别当心（见第 22.4.1 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。

### 22.4.1 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上溢。

对于写操作，建议用户直接停止定时器，然后写入所需的值。如果定时器寄存器正进行递增计数，对定时器寄存器进行写操作可能会导致写争用，这可能在 TMR1H:TMR1L 寄存器对中产生不可预测的值。

## 22.5 Timer1 门控

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

Timer1 门控也可由多个可选择源驱动。

### 22.5.1 TIMER1 门控使能

通过将 T1GCON 寄存器的 TMR1GE 位置 1 使能 Timer1 门控使能模式。使用 T1GCON 寄存器的 T1GPOL 位来配置 Timer1 门控使能模式的极性。

使能 Timer1 门控使能模式时，Timer1 将在 Timer1 时钟源的上升沿递增。禁止 Timer1 门控使能模式时，不会发生递增，Timer1 将保持当前计数。时序详细信息请参见图 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

### 22.5.2 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<1:0> 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 (T1G)
01	Timer0 上溢 (T0_overflow) (TMR0 从 FFh 递增到 00h)
10	比较器 1 输出 (C1_OUT_sync) <sup>(1)</sup>
11	比较器 2 输出 (C2_OUT_sync) <sup>(1)</sup>

注 1: 可选的同步比较器输出。

## 22.5.2.1 T1G 引脚门控操作

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

## 22.5.2.2 Timer0 上溢门控操作

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

## 22.5.3 TIMER1 门控翻转模式

使能 Timer1 门控翻转模式时，可测量 Timer1 门控信号整个周期的长度，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，将清除触发器并保持清零。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。

## 22.5.4 TIMER1 门控单脉冲模式

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

## 22.5.5 TIMER1 门控值状态

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

## 22.5.6 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。

22.6 Timer1 中断

Timer1 寄存器对 (TMR1H:TMR1L) 递增到 FFFFh, 然后返回到 0000h。当 Timer1 计满返回时, PIR1 寄存器的 Timer1 中断标志位将置 1。为允许计满返回时的中断, 必须将以下位置 1:

- T1CON 寄存器的 TMR1ON 位
- PIE1 寄存器的 TMR1IE 位
- INTCON 寄存器的 PEIE 位
- INTCON 寄存器的 GIE 位

在中断服务程序中将 TMR1IF 位清零将清除中断。

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

无论 T1SYNC 位的设置如何, Timer1 振荡器都会在休眠模式下继续工作。

22.7 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 TMR1ON 位置 1
- 必须将 PIE1 寄存器的 TMR1IE 位置 1
- 必须将 INTCON 寄存器的 PEIE 位置 1
- 必须将 T1CON 寄存器的 T1SYNC 位置 1
- 必须配置 T1CON 寄存器的 TMR1CS 位

器件将在上溢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。

图 22-2: TIMER1 递增边沿

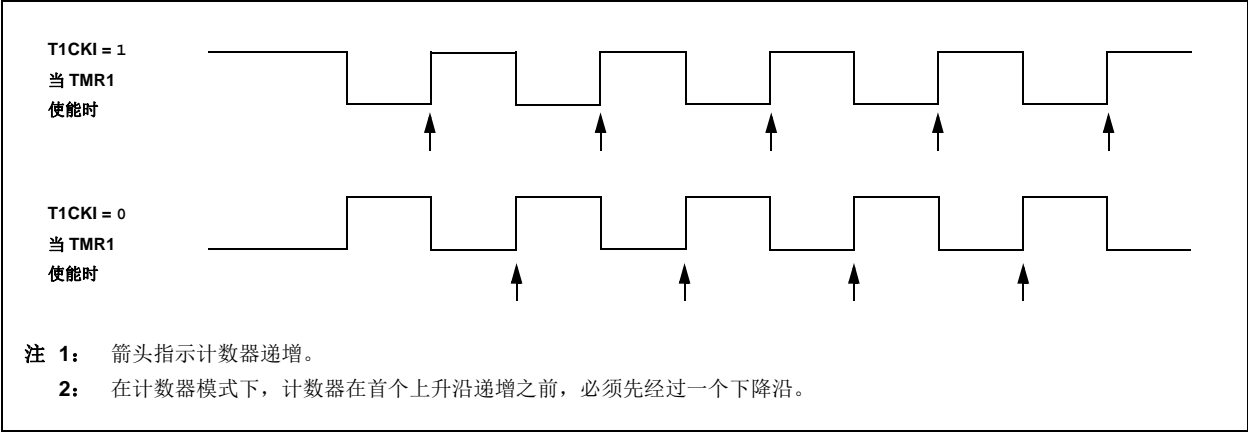


图22-3: TIMER1门控使能模式

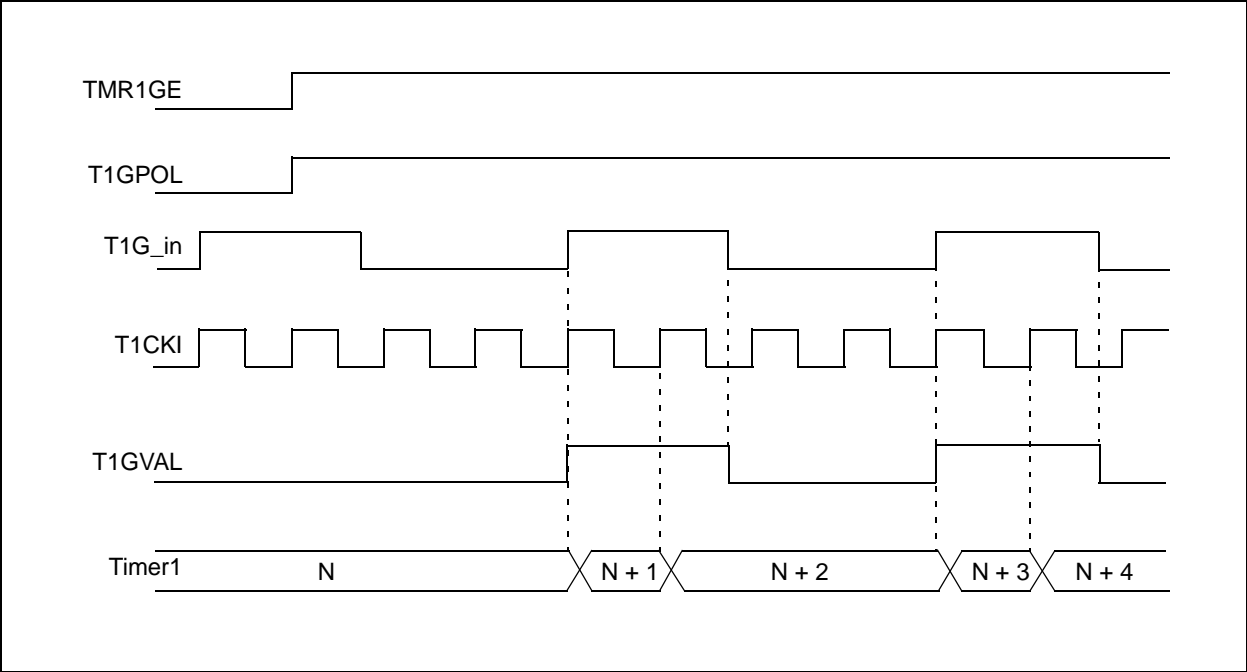


图22-4: TIMER1门控翻转模式

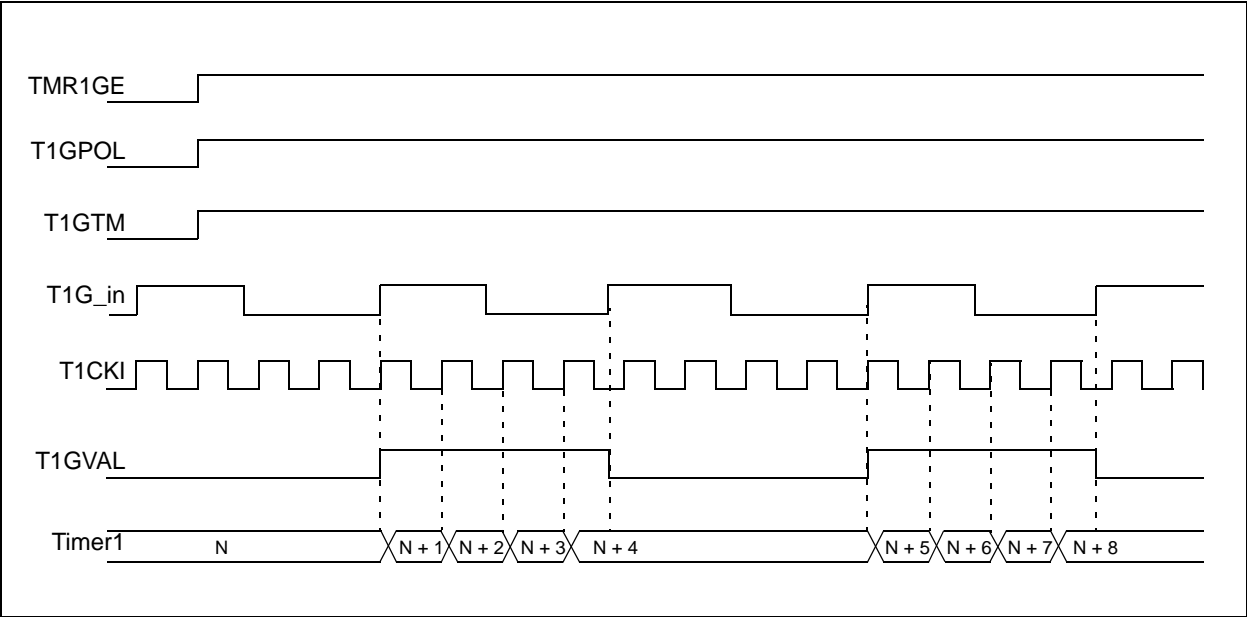




图22-5: TIMER1门控单脉冲模式

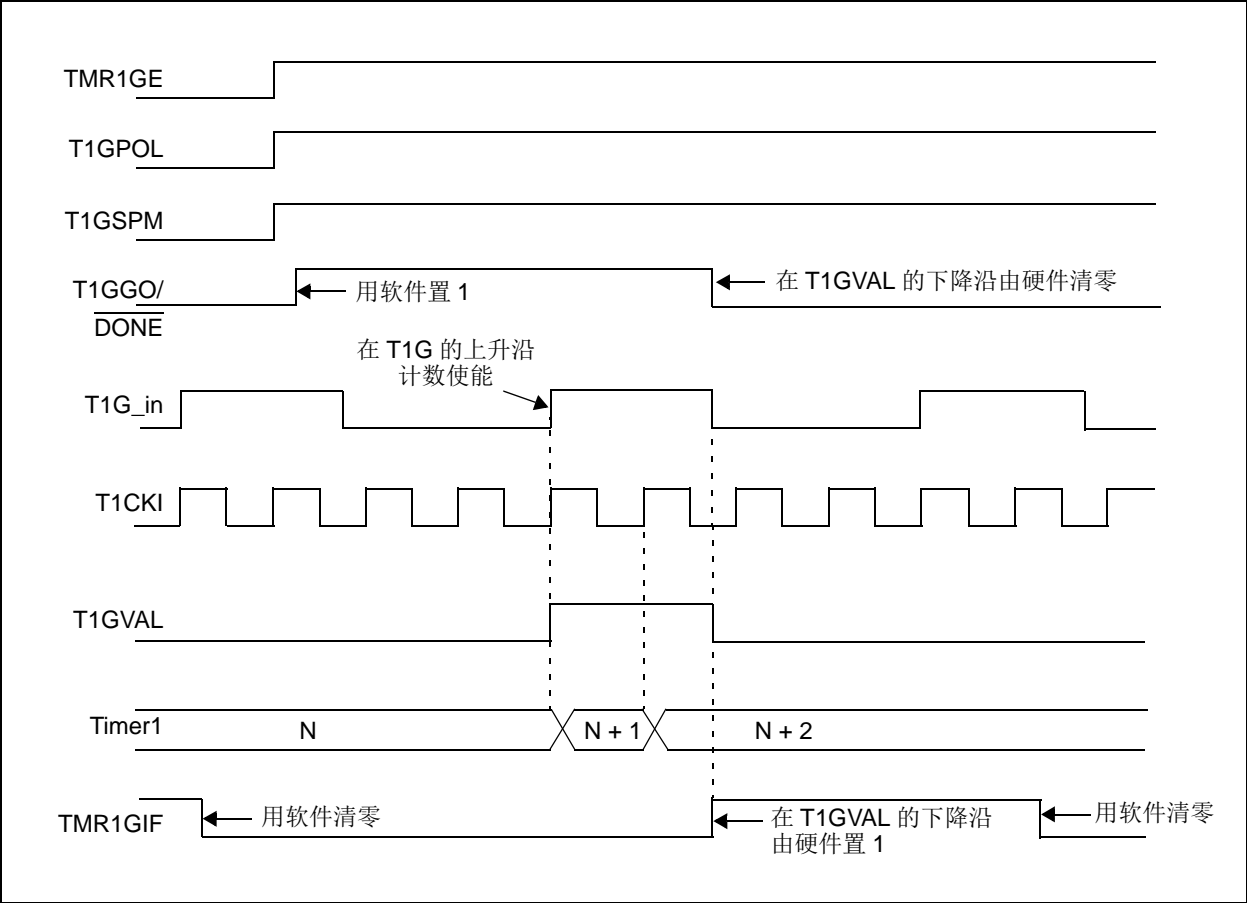
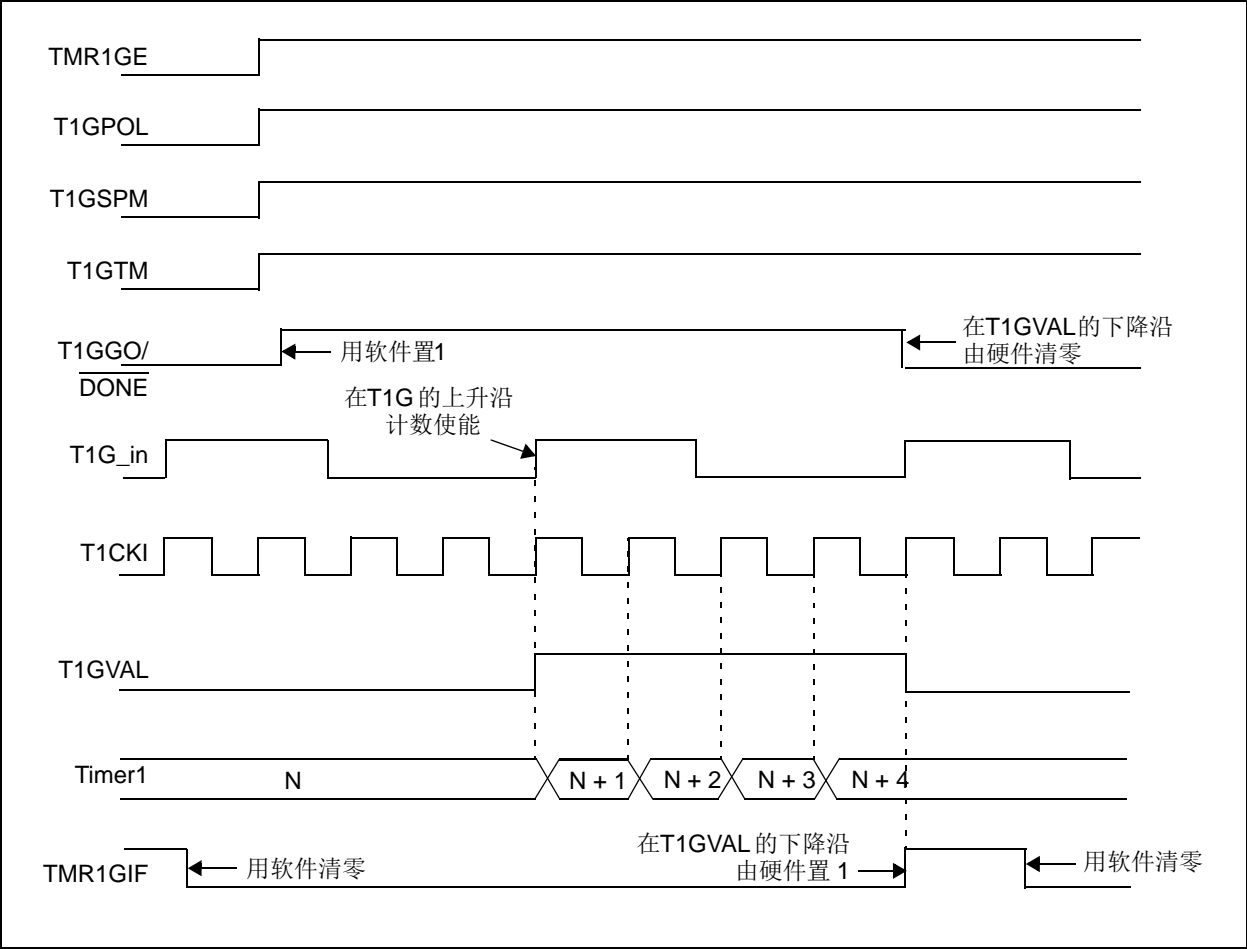


图22-6: TIMER1 门控单脉冲和翻转组合模式



22.8 寄存器定义：Timer1 控制

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	U-0	R/W-0/u	U-0	R/W-0/u
TMR1CS<1:0>		T1CKPS<1:0>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON
bit 7							bit 0

图注:		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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

**TMR1CS<1:0>**: Timer1 时钟源选择位  
11 = LFINTOSC  
10 = Timer1 时钟源为 T1CKI  
01 = Fosc  
00 = Fosc/4
- bit 5-4

**T1CKPS<1:0>**: Timer1 输入时钟预分频比选择位  
11 = 1:8 预分频比  
10 = 1:4 预分频比  
01 = 1:2 预分频比  
00 = 1:1 预分频比
- bit 3

**未实现**: 读为 0
- bit 2

**T1SYNC**: 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

**TMR1ON**: Timer1 使能位  
1 = 使能 Timer1  
0 = 停止 Timer1 并清零 Timer1 门控触发器

# PIC16(L)F1615/9

寄存器 22-2: T1GCON: TIMER1 门控控制寄存器

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
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<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 **TMR1GE:** Timer1 门控使能位  
如果 TMR1ON = 0:  
该位被忽略  
如果 TMR1ON = 1:  
1 = Timer1 计数由 Timer1 门控功能控制  
0 = Timer1 计数与 Timer1 门控功能无关
- bit 6 **T1GPOL:** Timer1 门控极性位  
1 = Timer1 门控为高电平有效 (当门控信号为高电平时 Timer1 计数)  
0 = Timer1 门控为低电平有效 (当门控信号为低电平时 Timer1 计数)
- bit 5 **T1GTM:** Timer1 门控翻转模式位  
1 = 使能 Timer1 门控翻转模式  
0 = 禁止 Timer1 门控翻转模式并清除触发器的输出  
Timer1 门控触发器在每个上升沿翻转。
- bit 4 **T1GSPM:** Timer1 门控单脉冲模式位  
1 = 使能 Timer1 门控单脉冲模式, 控制 Timer1 门控  
0 = 禁止 Timer1 门控单脉冲模式
- bit 3 **T1GGO/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 **T1GVAL:** Timer1 门控值状态  
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 0 **T1GSS<1:0>:** Timer1 门控源选择位  
11 = 比较器 2 的可选同步输出 (C2\_OUT\_sync)  
10 = 比较器 1 的可选同步输出 (C1\_OUT\_sync)  
01 = Timer0 上溢输出 (T0\_overflow)  
00 = Timer1 门控引脚 (T1G)

**表22-5: 与TIMER1相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器 所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
TMR1H	16位TMR1计数最高有效字节的保持寄存器								235*
TMR1L	16位TMR1计数最低有效字节的保持寄存器								235*
TMR3H	16位TMR3计数最高有效字节的保持寄存器								235*
TMR3L	16位TMR3计数最低有效字节的保持寄存器								235*
TMR5H	16位TMR5计数最高有效字节的保持寄存器								235*
TMR5L	16位TMR5计数最低有效字节的保持寄存器								235*
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	T1SYNC	—	TMR1ON	239
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		240
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		—	T3SYNC	—	TMR3ON	239
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ DONE	T3GVAL	T3GSS<1:0>		240
T5CON	TMR5CS<1:0>		T5CKPS<1:0>		—	T5SYNC	—	TMR5ON	239
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/ DONE	T5GVAL	T5GSS<1:0>		240

**图注:** — = 未实现位，读为0。Timer1模块不使用阴影单元。

\* 该页提供寄存器信息。

**注 1:** 未实现，读为1。

注:

## 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，或者在单事件和单稳态工作模式下与控制启动、运行、暂停和复位操作的外部信号配合工作。通过将这些定时器和其他内部外设（例如比较器和CCP模块）配合工作，可以实现复杂的波形控制（例如脉冲密度调制）。定时器的特性包括：

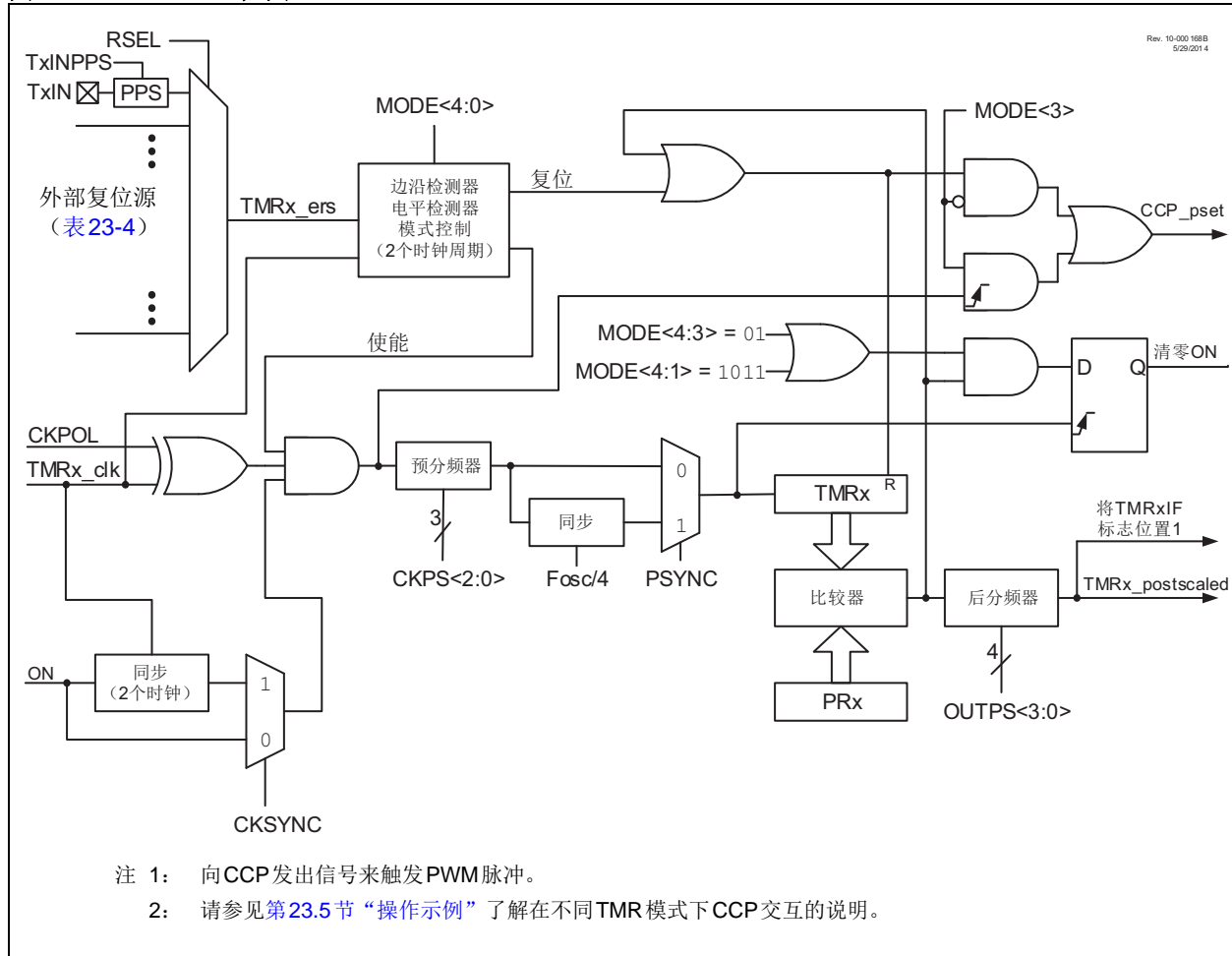
- 8位定时器寄存器
- 8位周期寄存器
- 可选的外部硬件定时器复位
- 可编程的预分频器（分频比为1:1至1:128）
- 可编程的后分频器（分频比为1:1至1:16）
- 可选的同步/异步操作
- 备用时钟源
- 周期中断

- 3种工作模式：
  - 自由运行周期
  - 单事件
  - 单稳态

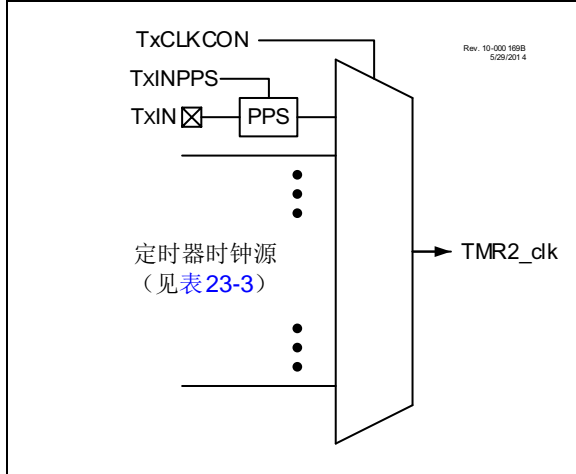
Timer2框图请参见图23-1。时钟源框图请参见图23-2。

**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的所有引用均同样适用于Timer4和Timer6。对T2PR的所有引用也同样适用于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具有以下3种工作模式：

- 自由运行周期
- 单事件
- 单稳态

每种模式下都有启动、停止和复位这几个选项。表23-1列出了这些选项。

在所有模式下，TMR2计数寄存器均在可编程的预分频器时钟信号的上升沿递增。当TMR2等于T2PR时，将输出高电平到后分频器计数器。TMR2在下一个时钟输入时清零。

硬件的外部信号也可以配置为门控定时器操作或强制TMR2计数复位。在门控模式下，计数器在门控禁用时停止，并在门控使能时继续工作。在复位模式下，TMR2计数由外部时钟源的电平或边沿复位。

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
- 外部复位源事件，它们会复位定时器。

**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，TMR2\_postscaled输出上会出现宽度为一个时钟周期的脉冲，同时后分频器计数清零。

### 23.1.2 单事件模式

单事件模式与自由运行周期模式相同，只是在TMR2与T2PR匹配时ON位会清零，同时定时器会停止并直到T2ON位循环清零并置1后重启。在该工作模式下，后分频器OUTPS<4:0>位的值仅在为0时有意义，因为定时器在发生第一个周期事件时停止，并且后分频器在定时器重启时复位。

### 23.1.3 单稳态模式

单稳态模式与单事件模式类似，只是ON位不清零，并且定时器可以通过外部复位事件重启。

## 23.2 Timer2输出

Timer2模式的主输出为TMR2\_postscaled，当后分频器计数器与TMR2xCON寄存器的OUTPS位的值匹配时，该输出将产生宽度为单个TMR2\_clk周期的脉冲。T2PR后分频器在每次TMR2值与T2PR值匹配时递增。可以将该信号选择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，以正确地按时钟控制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26.4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寄存器中的MODE<4:0>位控制。边沿触发模式在外部触发信号之间需要6个定时器时钟周期。电平触发模式需要至少3个定时器时钟周期长的触发电平。在调试暂停模式下，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 且 TMRx_ers = 1	—	ON = 0 或 TMRx_ers = 0	
		010		硬件门控，低电平有效	ON = 1 且 TMRx_ers = 0	—	ON = 0 或 TMRx_ers = 1	
		011	周期脉冲 和 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 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 且 TMRx_ers ↑	—		
		010		下降沿启动	ON = 1 且 TMRx_ers ↓	—		
		011		任意边沿启动	ON = 1 且 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 且 TMRx_ers ↑	TMRx_ers ↑		
		101		下降沿启动和 下降沿复位	ON = 1 且 TMRx_ers ↓	TMRx_ers ↓		
		110		上升沿启动和 低电平复位（图23-11）	ON = 1 且 TMRx_ers ↑	TMRx_ers = 0		
		111		下降沿启动和 高电平复位	ON = 1 且 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 且 TMRx_ers ↑	—	ON = 0 或 TMRx = PRx 后的下一个 时钟 （注3）	
		010		下降沿启动	ON = 1 且 TMRx_ers ↓	—		
		011		任意边沿启动	ON = 1 且 TMRx_ers ↓	—		
		保留	100	保留				
		保留	101	保留				
		单事件	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 且 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 且 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启定时器。

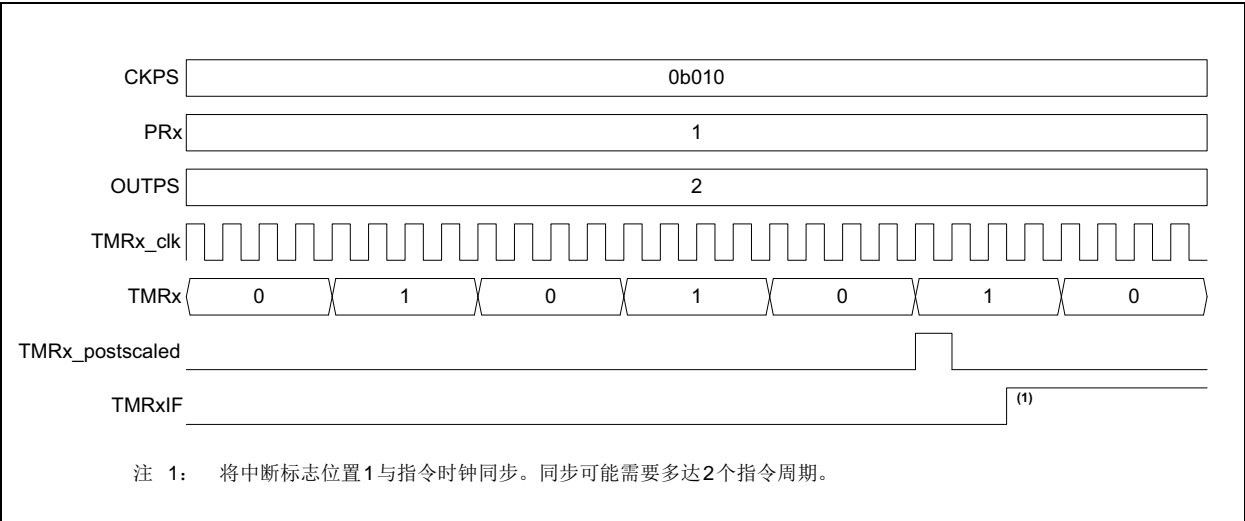
2: 当 TMRx = PRx 时, 下一个时钟将清零 ON 并在 00h 时停止 TMRx。

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

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

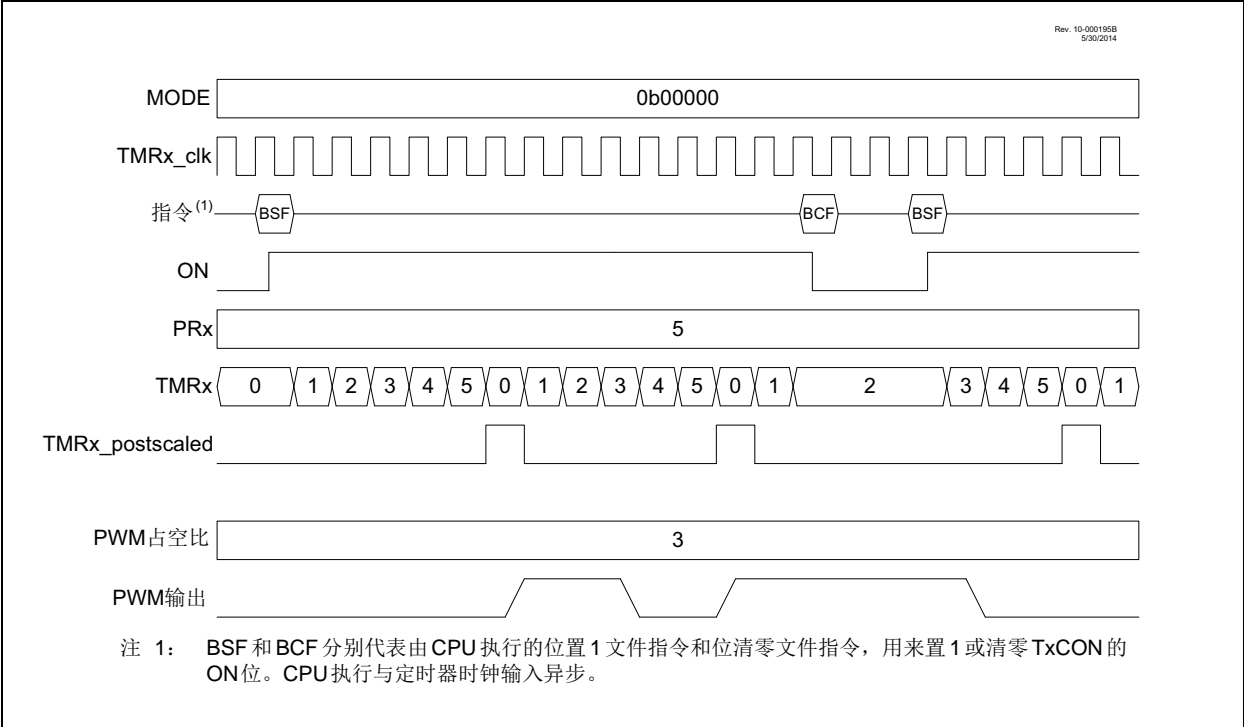
除非另有说明，否则以下说明适用于后面的时序图：

- 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26.4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 = 000000）



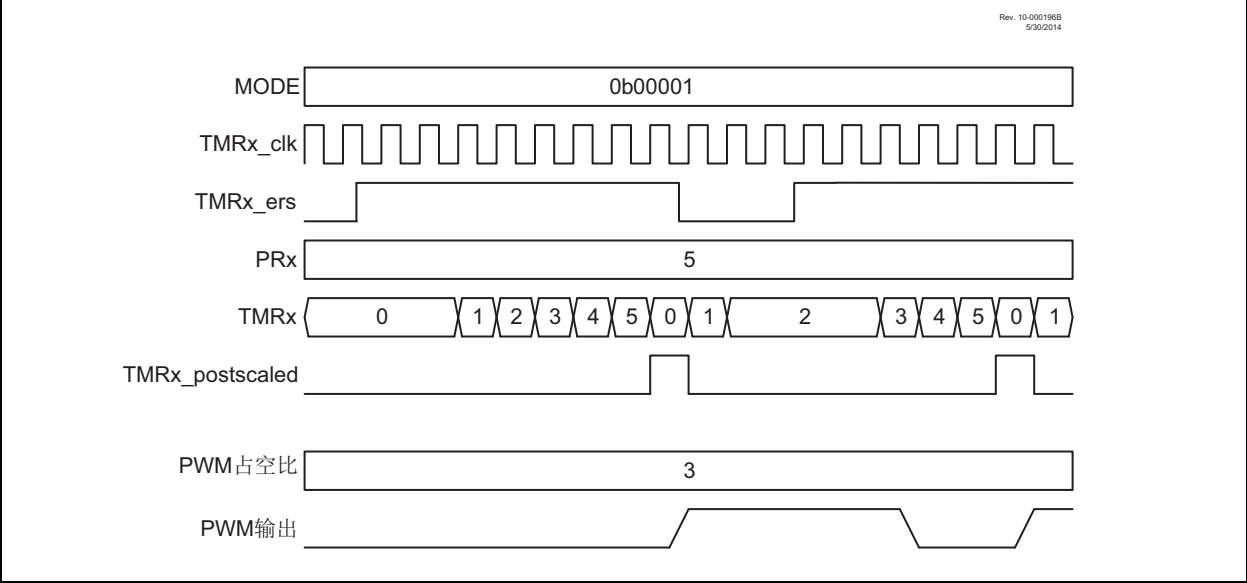
23.5.2 硬件门控模式

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，在该模式下输入高电平会启动计数器。

图23-5： 硬件门控模式时序图（MODE = 00001）



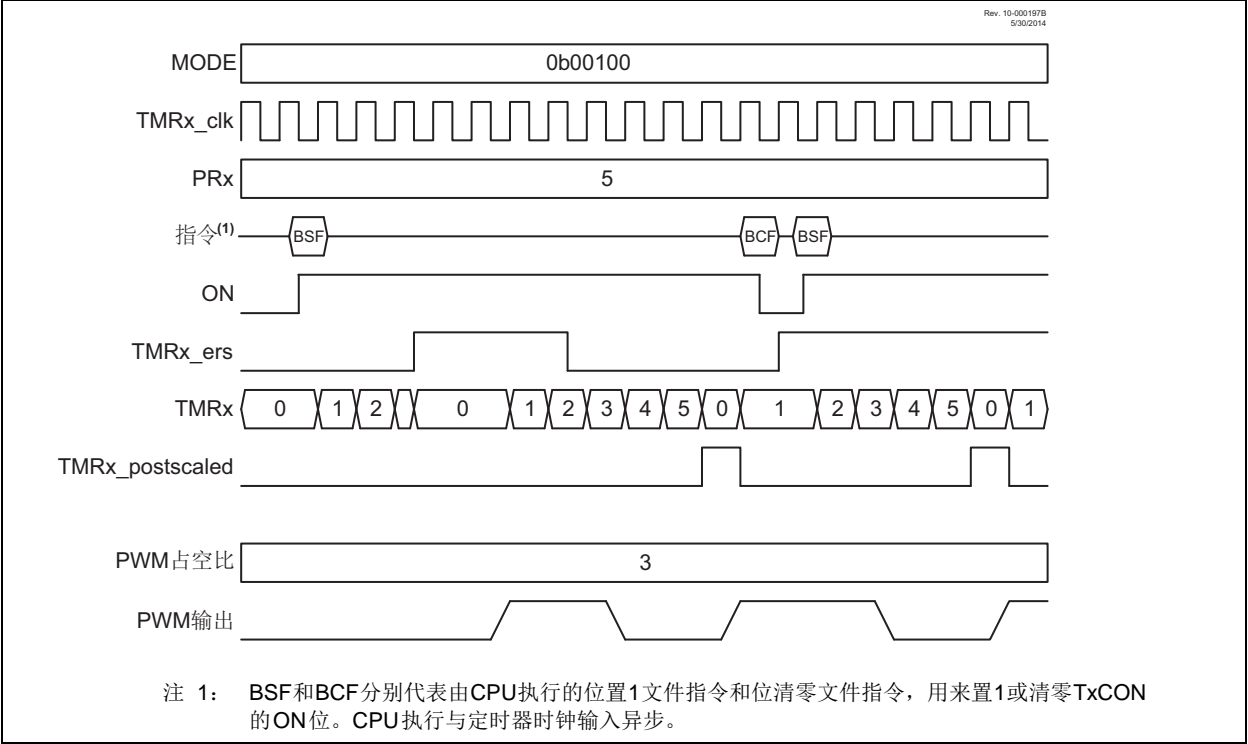
23.5.3 边沿触发硬件限制模式

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）



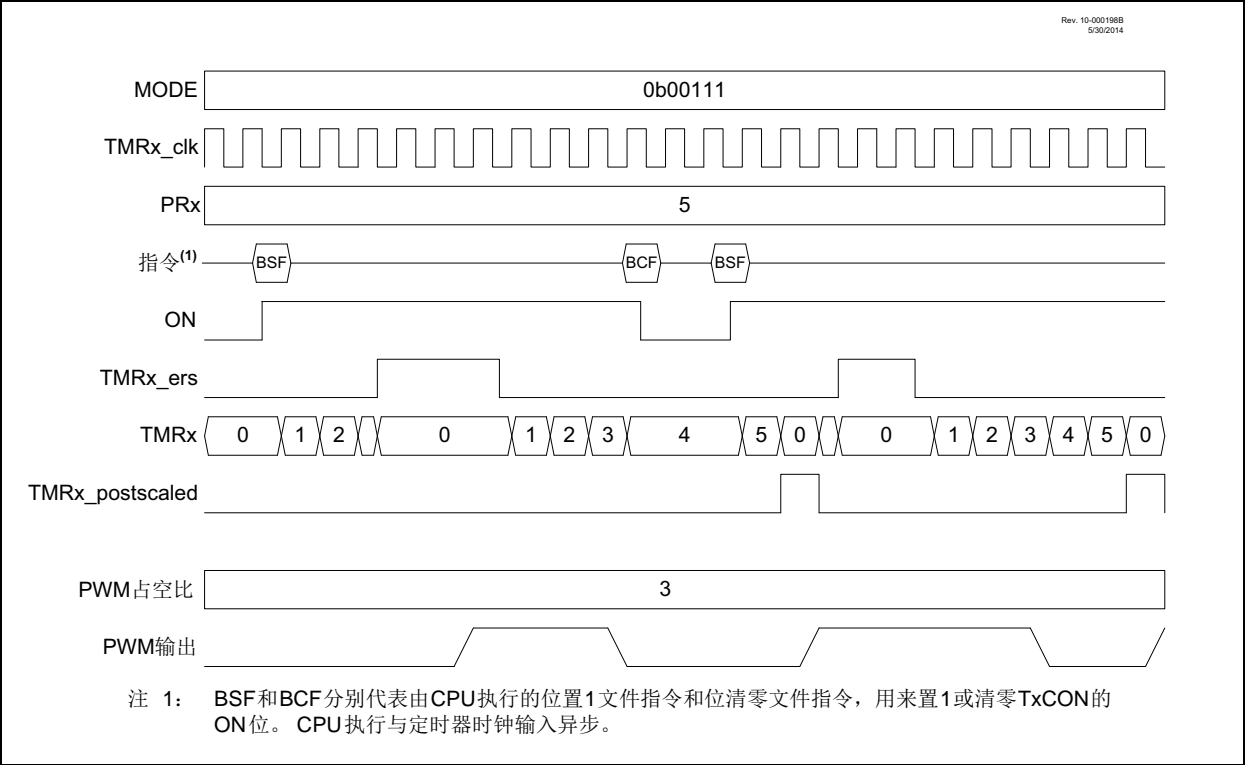
23.5.4 电平触发硬件限制模式

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置为高电平，并在定时器计数与CCPRx 值匹配时置为低电平。当定时器计数与PRx 值匹配时或者当外部复位信号变为有效并保持两个时钟周期后，定时器将复位。

在发生PRx匹配后的时钟周期内或者外部信号电平释放复位的两个时钟周期后，定时器开始计数，PWM输出置为高电平。PWM输出将一直保持高电平，直到定时器递增计数至与CCPRx 脉冲宽度值匹配为止。如果外部复位信号变为有效而PWM输出为高电平，则PWM输出将保持高电平直到复位信号被释放，在此期间定时器将递增计数至与CCPRx 值匹配。

图23-7： 电平触发硬件限制模式时序图（MODE = 00111）

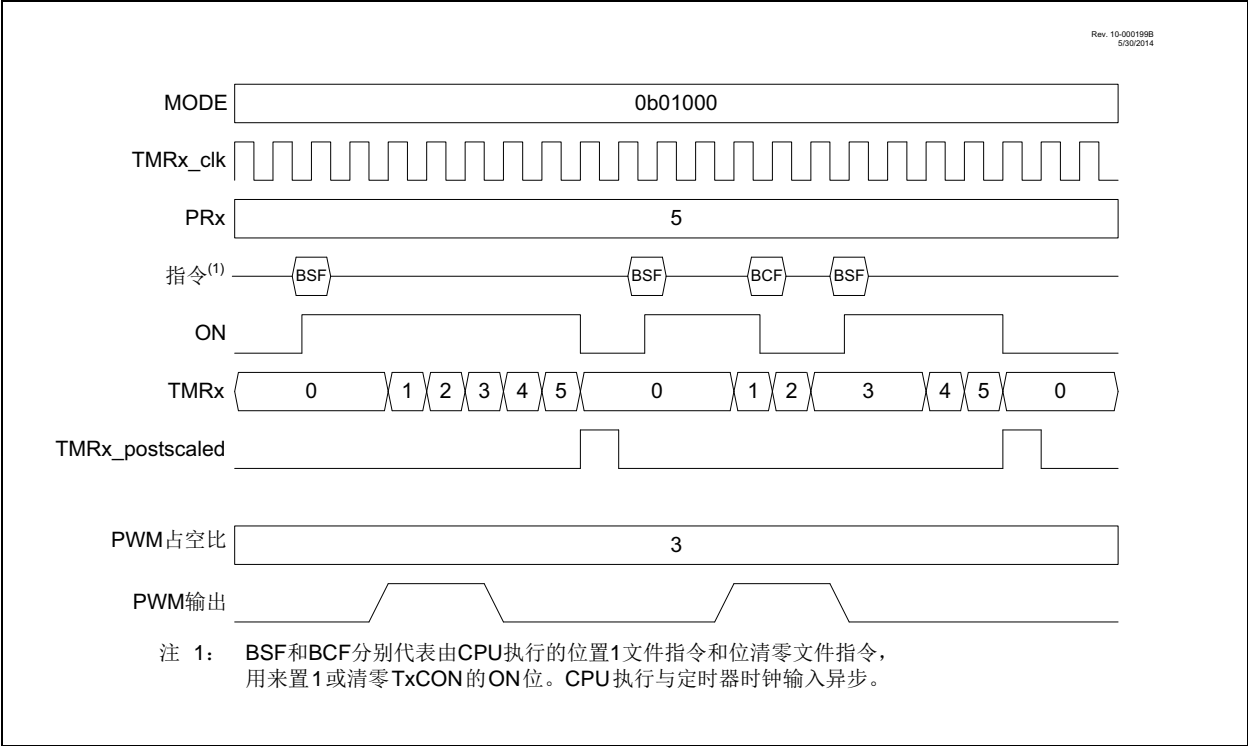


23.5.5 软件启动单事件模式

在单事件模式下，当定时器值与PRx周期值匹配时，定时器会发生复位，ON位会清零。ON位必须用软件置1，以启动另一个定时器周期。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。在第一种情况下，BSF指令将ON置1，计数器运行至完成，并清零ON。在第二种情况下，BSF指令启动计数周期，BCF/BSF指令在该周期期间关闭和开启计数器，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脉冲宽度值匹配时终止。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PRx匹配之前清零ON位，则PWM驱动将被延迟，延迟时间等于ON位保持清零状态的时间。在ON位由PRx周期计数匹配事件清零之后，只能通过将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

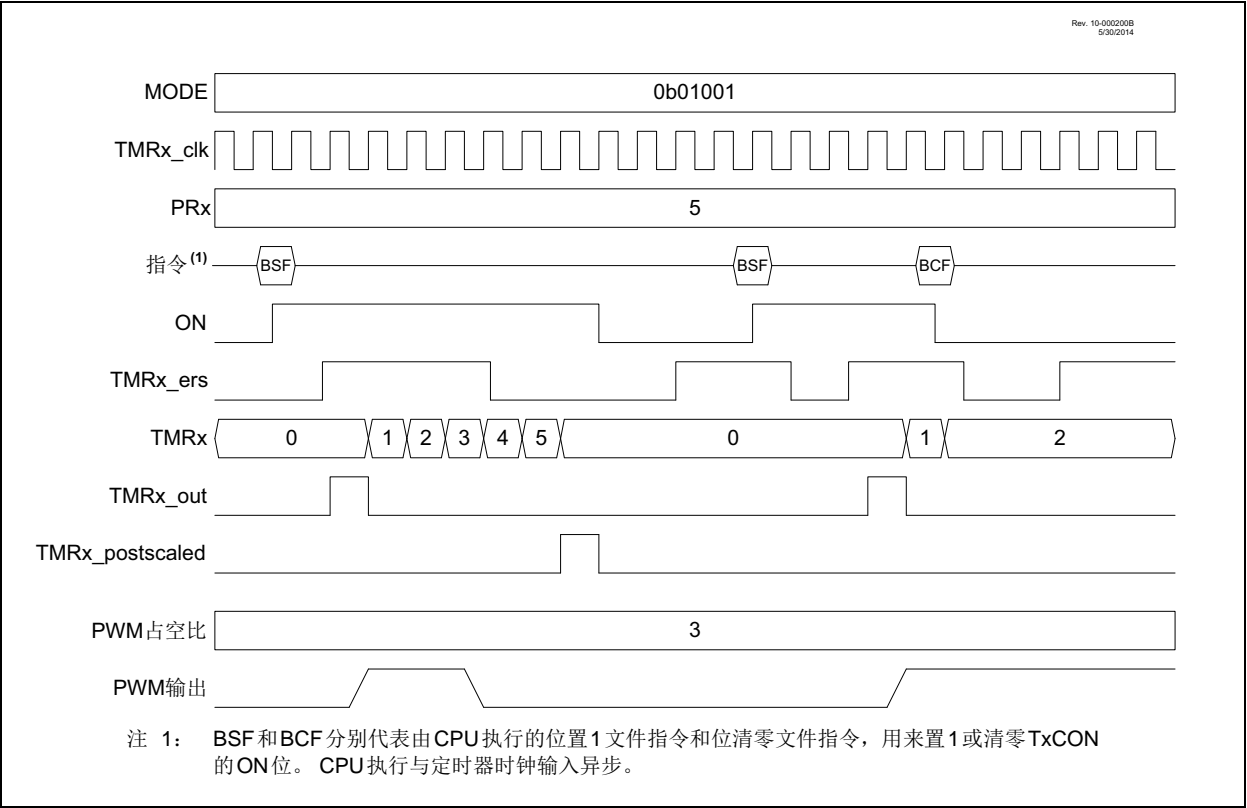
在ON位置1之后，边沿触发单事件模式会在外部信号输入出现边沿时启动定时器，在定时器与PRx周期值匹配时清零ON位。以下边沿可以启动定时器：

- 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TMRx\_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脉冲宽度值匹配时变为无效，并在定时器由于PRx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

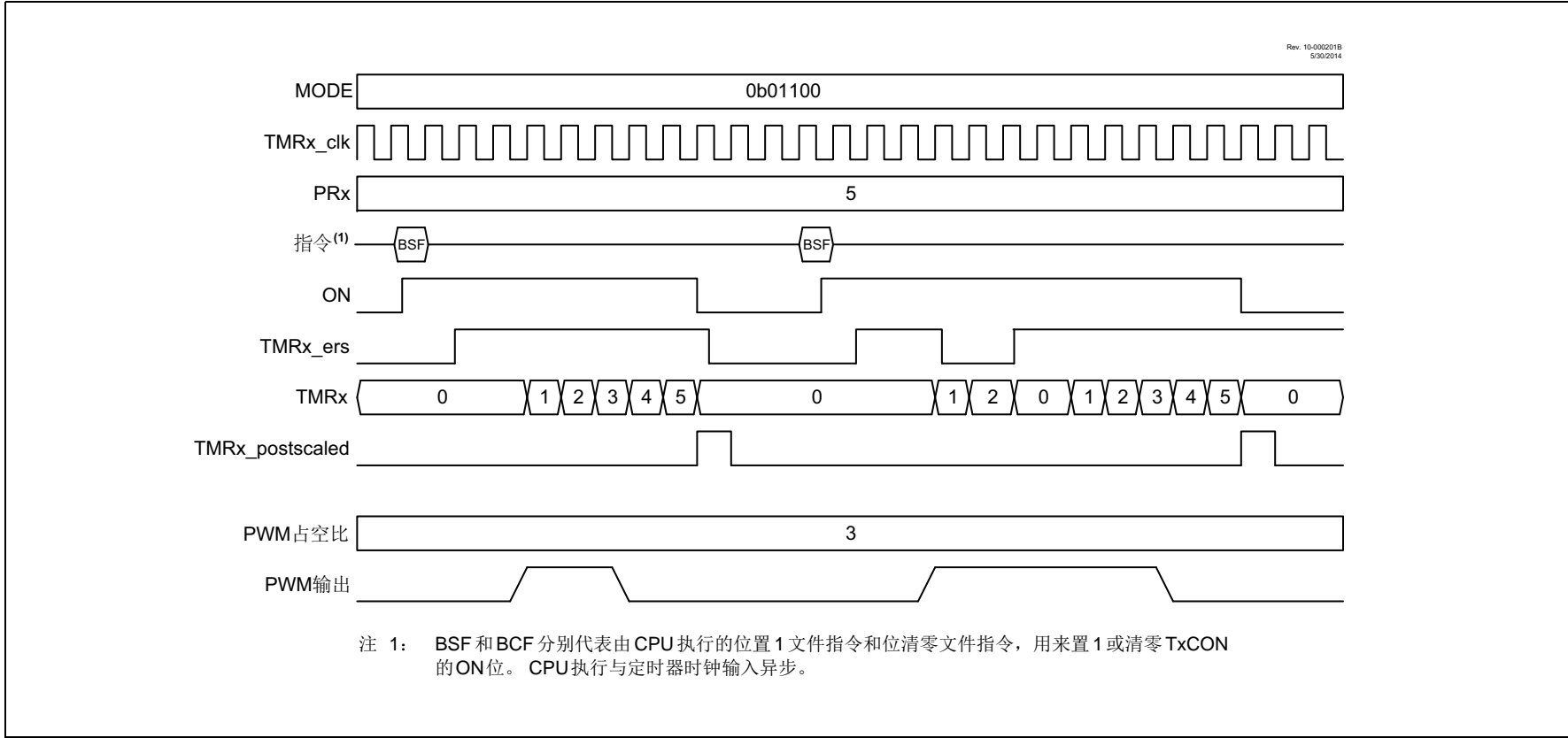
在边沿触发硬件限制单事件模式下，定时器在ON位置1之后出现第一个外部信号边沿时启动，并在出现所有后续边沿时复位。在ON位置1后，只需通过第一个边沿来启动定时器。计数器将在所有后续外部复位边沿的两个时钟后自动恢复计数。边沿触发信号如下：

- 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PRx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。当定时器与CCPRx脉冲宽度值匹配时，PWM驱动会变为无效并一直保持无效状态，直到定时器由于PRx周期匹配而暂停为止，除非外部信号边沿在发生匹配之前使定时器复位。

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

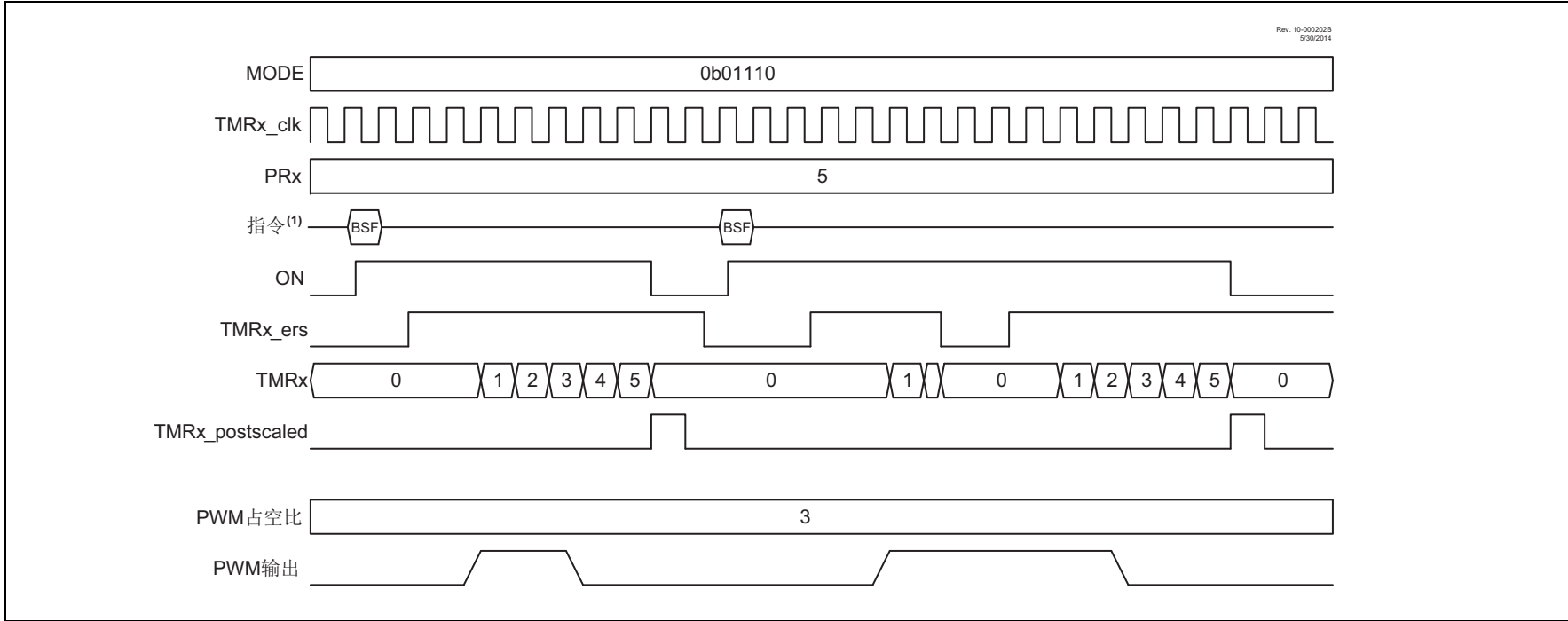
在电平触发单事件模式下，定时器计数在外部信号电平出现时复位，并在ON位置1时从复位电平变为有效电平的上升沿/下降沿开始计数。复位电平可选择如下：

- 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PRx匹配或由软件控制清零时，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。

将电平触发复位单事件模式与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脉冲宽度计数时变为无效。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）



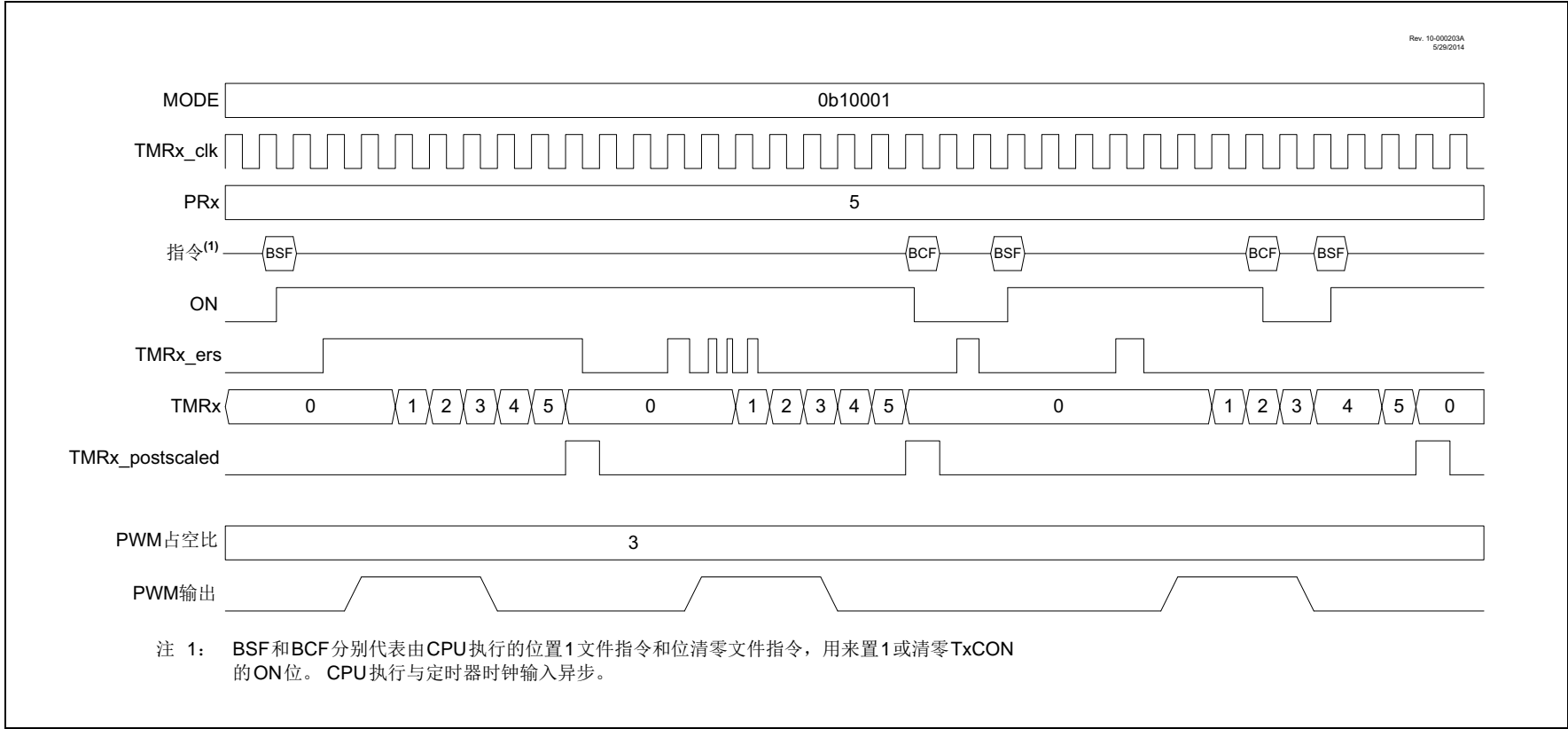
## 23.5.9 边沿触发单稳态模式

在ON位置1后，边沿触发单稳态模式在外部复位信号输入出现边沿时启动定时器，在定时器与PRx周期值匹配时停止递增定时器。以下边沿可以启动定时器：

- 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)

将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变为有效，但不会在定时器与PRx值匹配时变为有效。当定时器正在递增计数时，外部复位信号的其他边沿不会影响CCP PWM。

图23-12: 上升沿触发单稳态模式时序图 (MODE = 10001)



## 23.5.10 电平触发硬件限制单事件模式

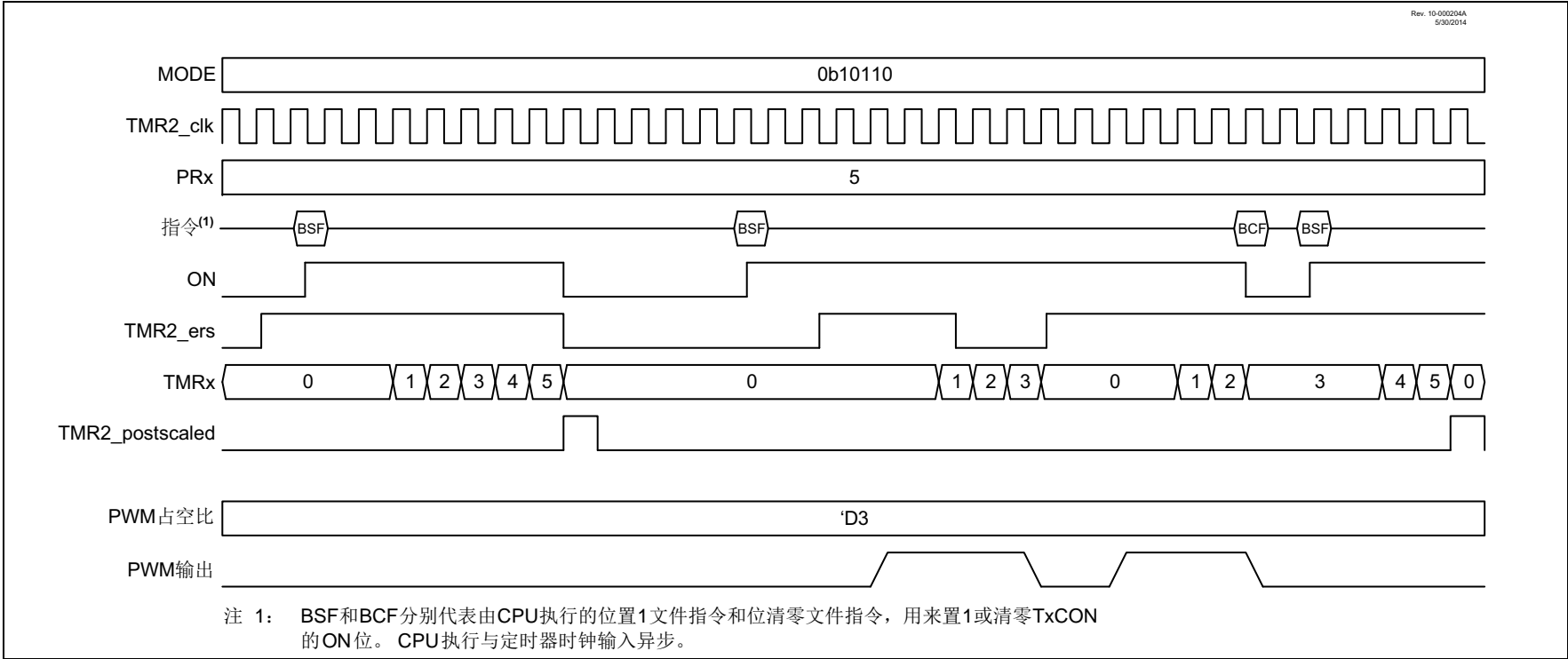
在电平触发硬件限制单事件模式下，计数器在外部复位电平出现时保持复位状态，并在 **ON** 位置 1 且外部信号不在复位电平时开始计数。如果外部信号不在复位电平或 **ON** 位置 1，则其他置为有效的信号将启动定时器。复位电平可选择如下：

- 低复位电平 ( $\text{MODE}<4:0> = 10110$ )
- 高复位电平 ( $\text{MODE}<4:0> = 10111$ )

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 **PRx** 匹配或软件控制清零时，定时器将保持复位状态，直到 **ON** 位置 1 且外部信号不在复位电平。

将电平触发硬件限制单事件模式与 **CCP PWM** 操作配合使用时，**PWM** 驱动会随外部信号边沿或 **ON** 位置 1（二者均会启动定时器）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，只要选择的时钟源也仍在运行，Timer2就会在休眠模式下工作。选择LFINTOSC、MFINTOSC或HFINTOSC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。

# PIC16(L)F1615/9

## 23.7 寄存器定义：Timer2/4/6控制

表23-2给出了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
1101-1111	保留	保留	保留
1011	AT1_perclk	AT1_perclk	AT1_perclk
1010	LC4_out	LC4_out	LC4_out
1001	LC3_out	LC3_out	LC3_out
1000	LC2_out	LC2_out	LC2_out
0111	LC1_out	LC1_out	LC1_out
0110	T2INPPS选择的引脚	T2INPPS选择的引脚	T2INPPS选择的引脚
0101	MFINTOSC 31.25 kHz	MFINTOSC 31.25 kHz	MFINTOSC 31.25 kHz
0100	ZCD1_output	ZCD1_output	ZCD1_output
0011	LFINTOSC	LFINTOSC	LFINTOSC
0010	HFINTOSC 16 MHz	HFINTOSC 16 MHz	HFINTOSC 16 MHz
0001	Fosc	Fosc	Fosc
0000	Fosc/4	Fosc/4	Fosc/4

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 <sup>(1)</sup>	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	<b>ON:</b> 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	<b>CKPS&lt;2:0&gt;:</b> 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	<b>OUTPS&lt;3:0&gt;:</b> Timerx 输出后分频比选择位 1111 = 1:16 预分频比 1110 = 1:15 预分频比 1101 = 1:14 预分频比 1100 = 1:13 预分频比 1011 = 1:12 预分频比 1010 = 1:11 预分频比 1001 = 1:10 预分频比 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”。

# PIC16(L)F1615/9

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 <sup>(1, 2)</sup>	CKPOL <sup>(3)</sup>	CKSYNC <sup>(4, 5)</sup>	MODE<4:0> <sup>(6, 7)</sup>				
bit 7							bit 0

图注:		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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<sup>(1, 2)</sup>  
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<sup>(3)</sup>  
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<sup>(4, 5)</sup>  
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<sup>(6, 7)</sup>  
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，下一个时钟会清零 TMRx，无论工作模式如何。

寄存器 23-4: TxRST: TIMERx外部复位信号选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	RSEL<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           **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
1111	保留	保留	保留
1110	PWM4_out	PWM4_out	PWM4_out
1101	PWM3_out	PWM3_out	PWM3_out
1100	LC4_out	LC4_out	LC4_out
1011	LC3_out	LC3_out	LC3_out
1010	LC2_out	LC2_out	LC2_out
1001	LC1_out	LC1_out	LC1_out
1000	ZCD1_out	ZCD1_out	ZCD1_out
0111	TMR6_postscaled	TMR6_postscaled	保留
0110	TMR4_postscaled	保留	TMR4_postscaled
0101	保留	TMR2_postscaled	TMR2_postscaled
0100	CCP2_out	CCP2_out	CCP2_out
0011	CCP1_out	CCP1_out	CCP1_out
0010	C2OUT_sync	C2OUT_sync	C2OUT_sync
0001	C1OUT_sync	C1OUT_sync	C1OUT_sync
0000	T2INPPS选择的引脚	T2INPPS选择的引脚	T2INPPS选择的引脚

# PIC16(L)F1615/9

表23-5: 与TIMER2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	—	OUT	FMT	MODE<3:0>				361
CCP2CON	EN	—	OUT	FMT	MODE<3:0>				361
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
PR2	Timer2 模块周期寄存器								244*
TMR2	8位TMR2寄存器的保持寄存器								244*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				263
T2CLKCON	—	—	—	—	CS<3:0>				262
T2RST	—	—	—	—	RSEL<3:0>				265
T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					264
PR4	Timer4 模块周期寄存器								244*
TMR4	8位TMR4寄存器的保持寄存器								244*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				263
T4CLKCON	—	—	—	—	CS<3:0>				262
T4RST	—	—	—	—	RSEL<3:0>				265
T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					264
PR6	Timer6 模块周期寄存器								244*
TMR6	8位TMR6寄存器的保持寄存器								244*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				263
T6CLKCON	—	—	—	—	—	T6CS<2:0>			262
T6RST	—	—	—	—	RSEL<3:0>				265
T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					264

图注: — = 未实现位, 读为0。Timer2 模块不使用阴影单元。

\* 提供寄存器信息的页。



# PIC16(L)F1615/9

I<sup>2</sup>C 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 24-2 给出了主模式下 I<sup>2</sup>C 接口模块的框图。图 24-3 给出了从模式下 I<sup>2</sup>C 接口模块的框图。

图 24-2: MSSP 框图 (I<sup>2</sup>C 主模式)

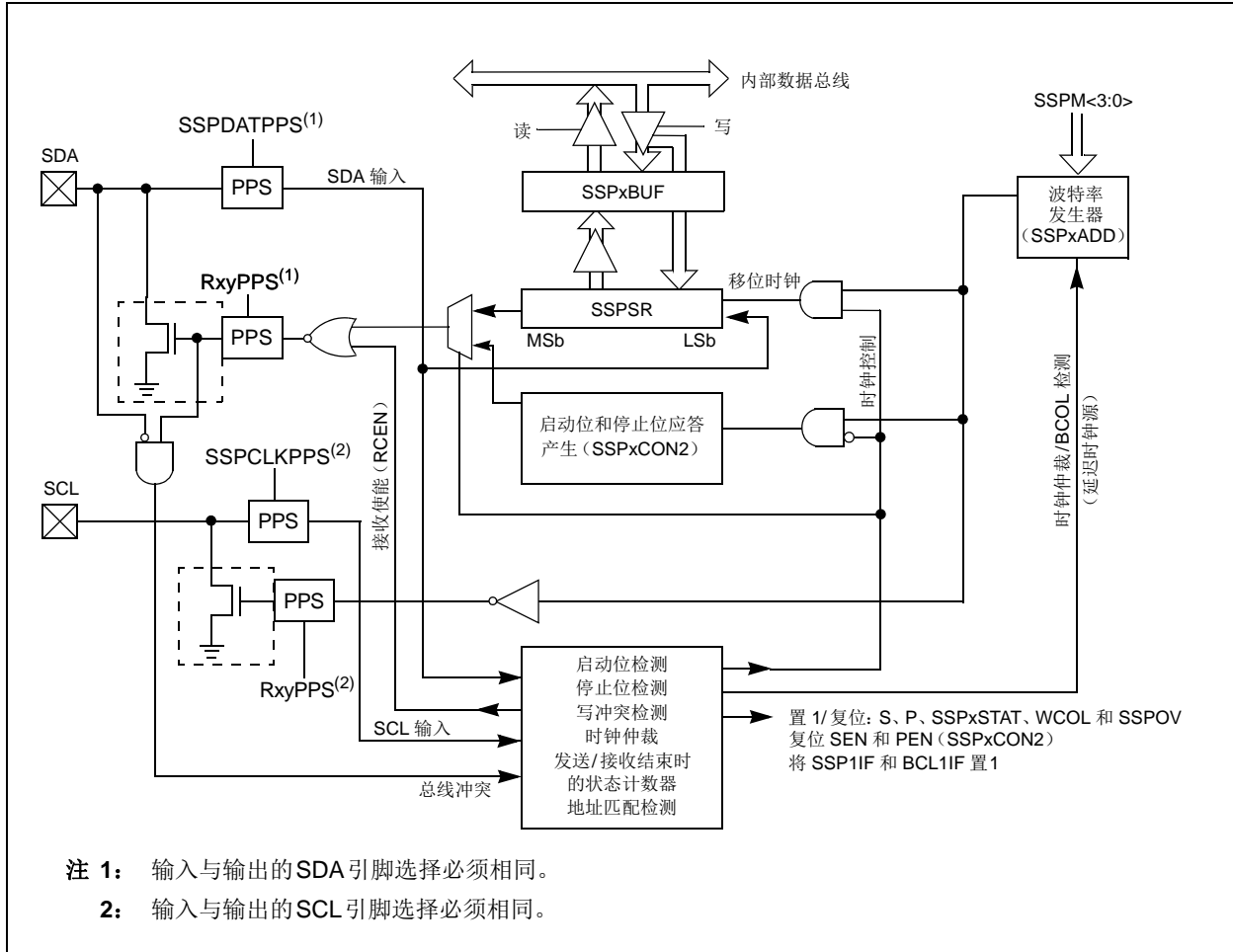
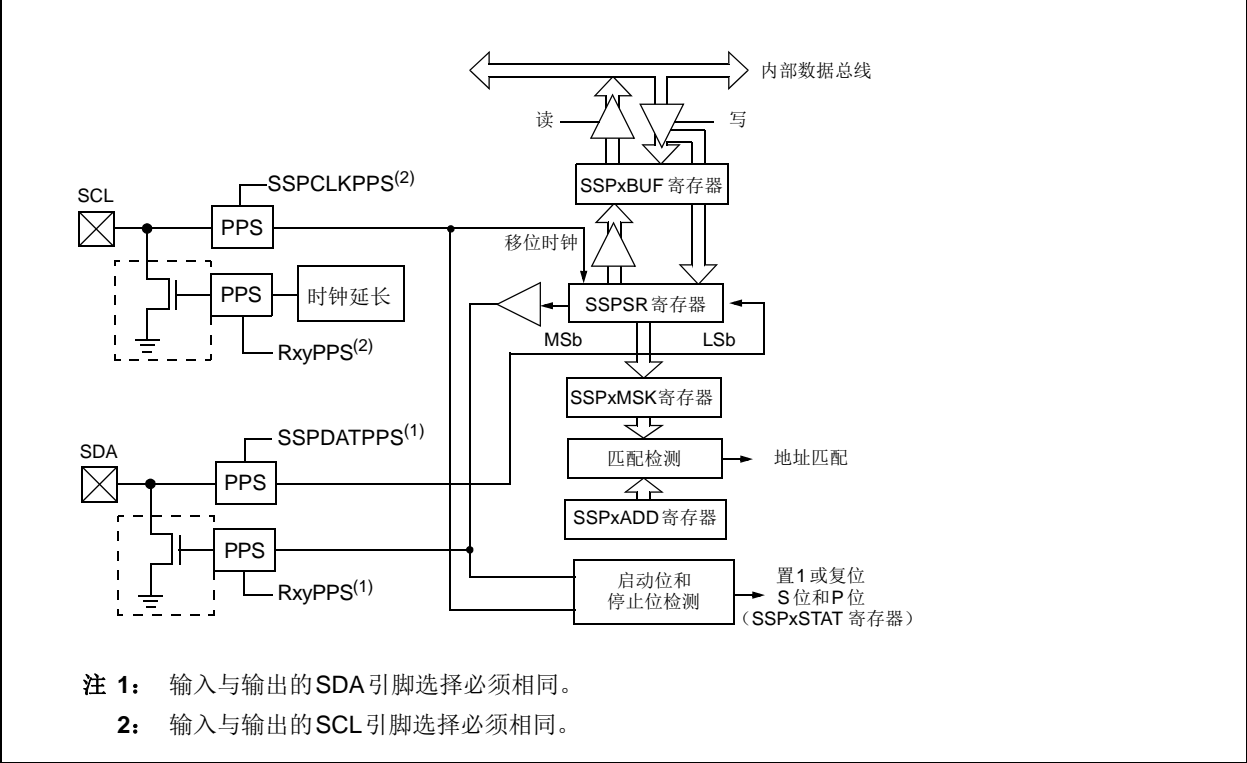




图24-3: MSSP框图 (I<sup>2</sup>C从模式)



## 24.2 SPI模式概述

串行外设接口（SPI）总线是以全双工模式工作的同步串行数据通信总线。器件在由主器件启动通信的主/从器件环境中进行通信。从器件通过称为从选择的片选进行控制。

SPI总线规定了4种信号连接：

- 串行时钟（SCK）
- 串行数据输出（SDO）
- 串行数据输入（SDI）
- 从选择（ $\overline{SS}$ ）

图24-1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24-4给出了主器件和多个从器件之间的典型连接。

主器件每次仅选择一个从器件。大多数从器件都具有三态输出，所以在未选择它们时，它们的输出信号会看起来好像与总线断开。

数据发送涉及到两个移位寄存器，它们大小都为8位，一个在主器件中，一个在从器件中。不论是对于主器件还是从器件，数据总是每次移出一位，最高有效位（Most Significant bit, MSb）先移出。与此同时，新的最低有效位（Least Significant bit, LSb）会被移入同一寄存器。

图24-5给出了分别配置为主器件和从器件的两个处理器之间的典型连接。

数据在所设定的时钟边沿从两个移位寄存器移出，并在相反的时钟边沿锁存。

主器件通过它的SDO输出引脚发送信息，并由该引脚所连接的从器件SDI输入引脚接收。从器件通过它的SDO输出引脚发送信息，并由该引脚所连接的主器件SDI输入引脚接收。

要开始进行通信，主器件需要先送出时钟信号。主器件和从器件应配置为相同的时钟极性。

主器件会通过从它的移位寄存器中发送MSb而启动数据发送。从器件会从同一条线上读取该位，并将它保存到其移位寄存器的LSb单元中。

在每个SPI时钟周期中，会发生全双工数据发送。这意味着，在主器件从其移位寄存器中送出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。

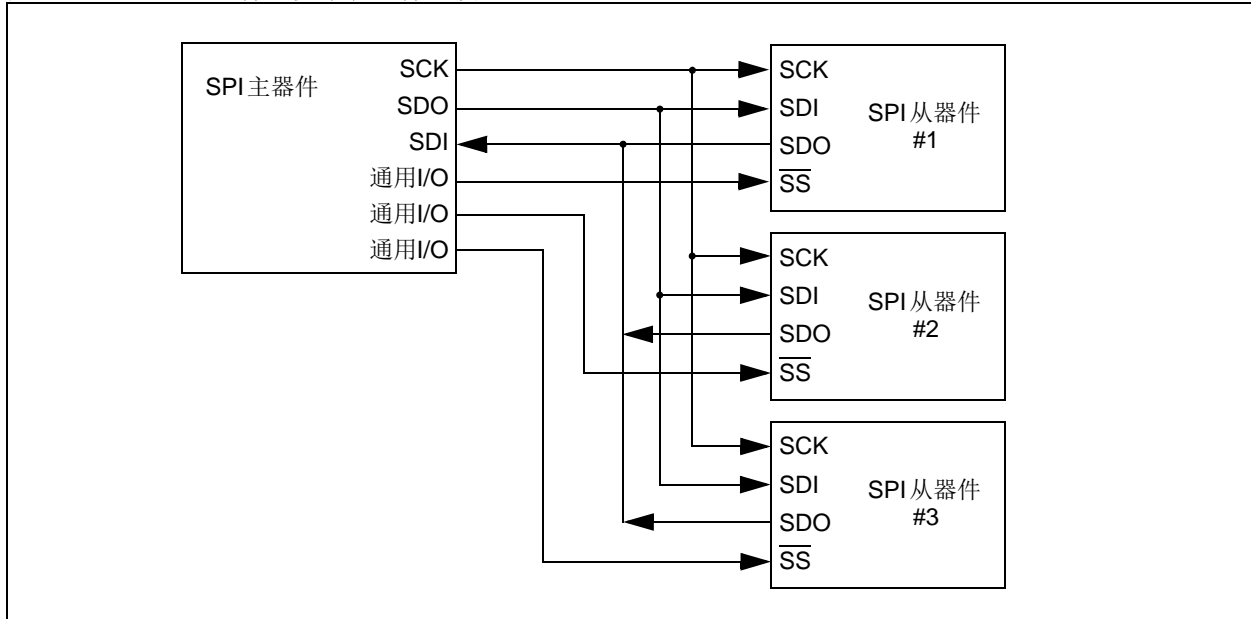
数据是否有意义（无效数据），取决于应用软件。这就导致以下三种数据传输情形：

- 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24-4: SPI主器件和多个从器件连接



#### 24.2.1 SPI模式寄存器

MSSP模块有5个寄存器用于SPI工作模式。这些寄存器分别为:

- 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模式操作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24.7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。

#### 24.2.2 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位置1。这会将SDI、SDO、SCK和SS引脚配置为串口引脚。要将引脚用作串口功能, 必须正确设置其中一些引脚的数据方向位 (在TRIS寄存器中):

- SDI必须将相应的TRIS位置1
- SDO必须将相应的TRIS位清零
- SCK (主模式) 必须将相应的TRIS位清零
- SCK (从模式) 必须将相应的TRIS位置1
- SS必须将相应的TRIS位置1

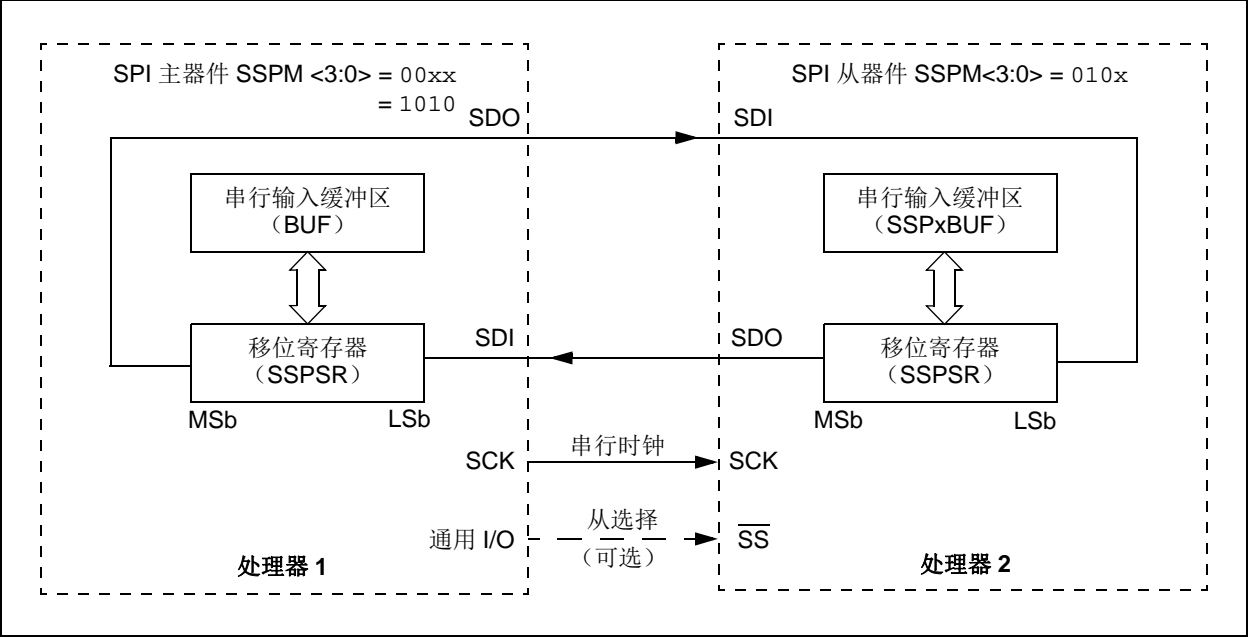
对于不需要的串口功能, 可通过将相应的数据方向 (TRIS) 寄存器编程为相反值来改写。

MSSP 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 会一直保存先前写入 SSPSR 的数据, 直到被接收数据就绪为止。一旦 8 位数据接收完毕, 该字节就被移入 SSPxBUF 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 位清零才能使以后对 SSPxBUF 寄存器的写入成功完成。

为确保应用软件能接收有效数据, 在下一个要发送的数据字节写入 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24-5: SPI主/从器件连接



24.2.3 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24-5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24-6、图24-8、图24-9和图24-10给出了SPI通信的波形图，其中MSB先发送。在主模式下，SPI时钟速率（比特率）可由用户编程为以下几种之一：

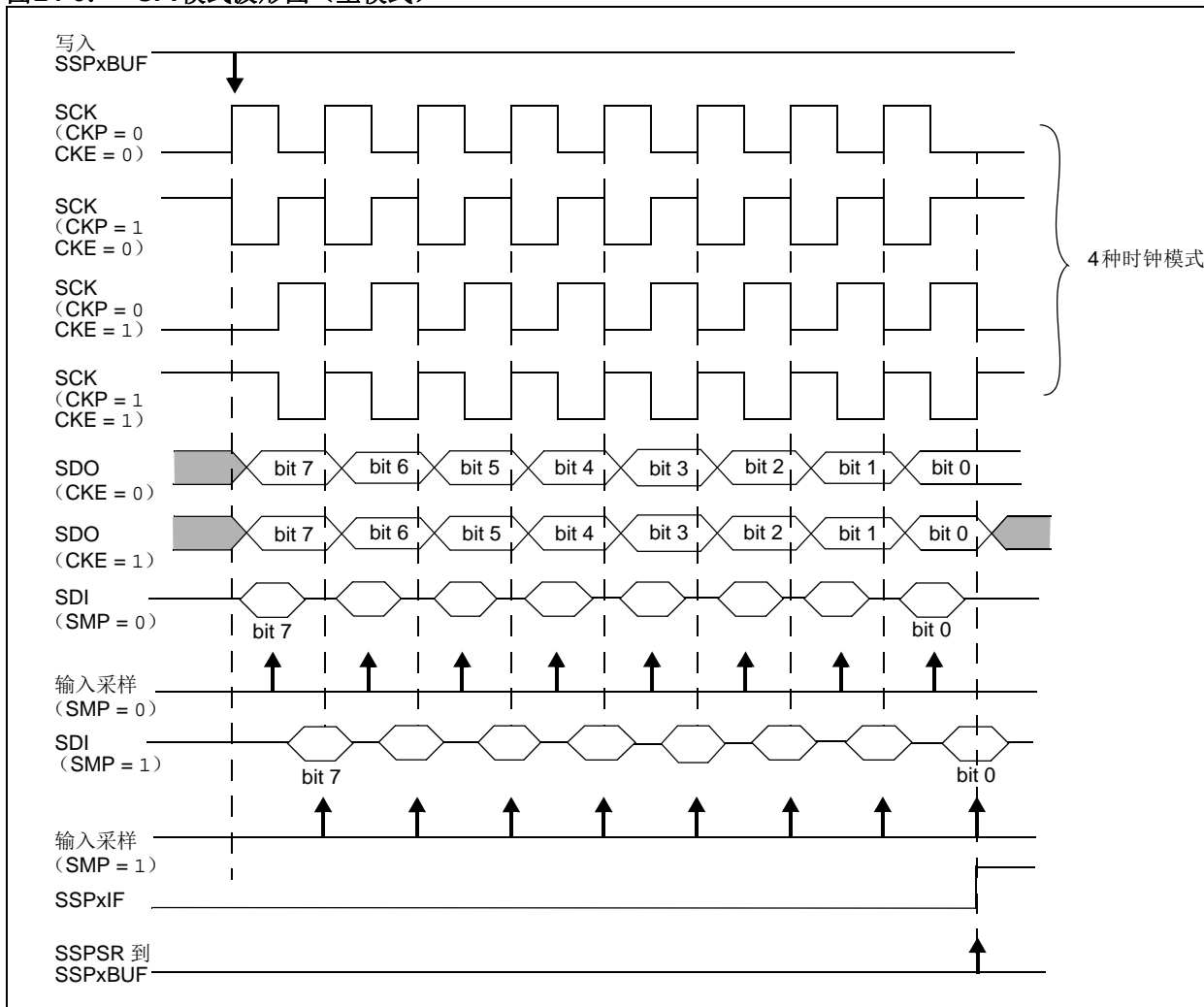
- Fosc/4（或Tcy）
- Fosc/16（或4 \* Tcy）
- Fosc/64（或16 \* Tcy）
- Timer2输出/2
- Fosc/(4 \* (SSPxADD + 1))

图24-6给出了主模式的波形图。

当CKE位置1时，SDO数据在SCK上出现时钟边沿前一直有效。图中所示的输入采样的变化由SMP位的状态反映。图中给出了将接收到的数据装入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24-6: SPI模式波形图 (主模式)



## 24.2.4 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。

### 24.2.4.1 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24-7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。

24.2.5 从选择同步

从选择也可以用于对通信进行同步。从选择线会一直保持高电平，直到主器件准备好进行通信。当从选择线下拉为低电平时，从器件就知道新的数据发送正在启动。

如果从器件未能正确地接收到通信，它会在从选择线恢复为高电平状态、数据发送结束时发生复位。然后，从器件会在从选择线再次下拉为低电平时准备好接收新的发送数据。如果不使用从选择线，则会存在从器件最终与主器件脱离同步的风险。如果从器件丢失了某个位，则在之后的数据发送中，它将总是偏离一位。使用从选择线可以让从器件和主器件在每次发送开始时保持同步。

$\overline{SS}$  引脚允许器件工作于同步从模式。SPI 必须处于从模式，并使能  $\overline{SS}$  引脚控制（ $SSPxCON1<3:0> = 0100$ ）。

当  $\overline{SS}$  引脚为低电平时，使能数据的发送和接收，同时驱动 SDO 引脚。

当  $\overline{SS}$  引脚变为高电平时，即使是在字节的发送过程中，也不再驱动 SDO 引脚，而是将其变成悬空输出状态。根据具体应用，可能需要使用外部上拉/下拉电阻。

- 注 1:** 当 SPI 处于从模式且使能  $\overline{SS}$  引脚控制（ $SSPxCON1<3:0> = 0100$ ）时，如果  $\overline{SS}$  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  $\overline{SS}$  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，位计数器会被强制设为 0。这可以通过将  $\overline{SS}$  引脚强制设为高电平或清零 SSPEN 位实现。

图24-7: SPI菊花链连接

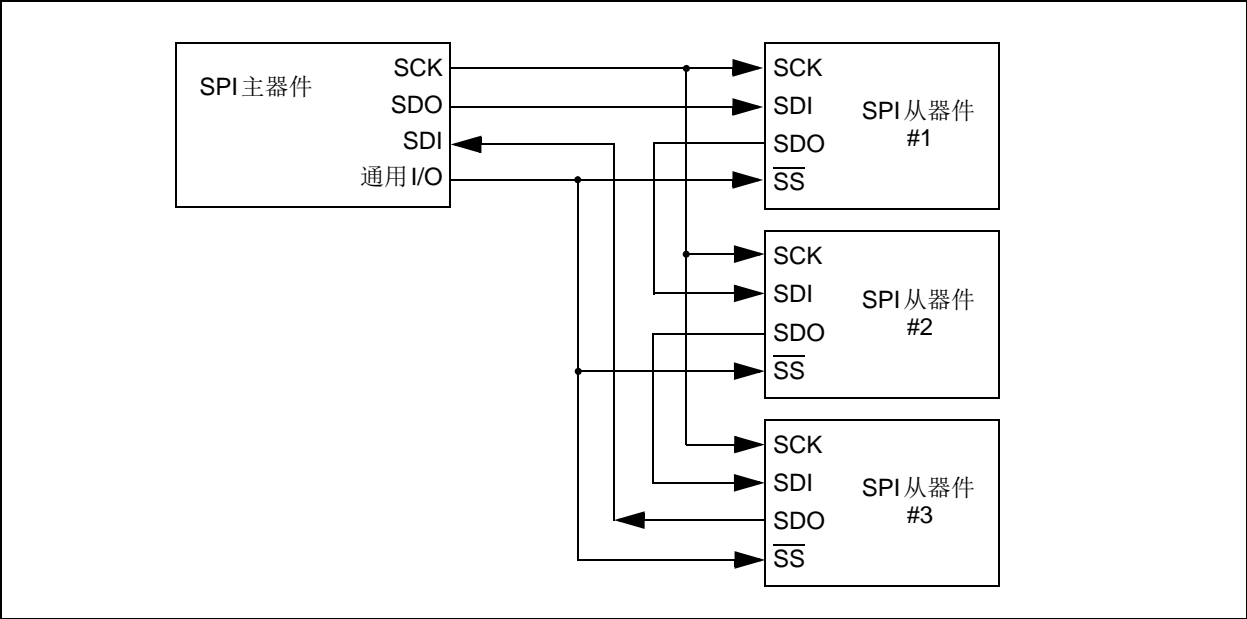


图24-8： 从选择同步波形图

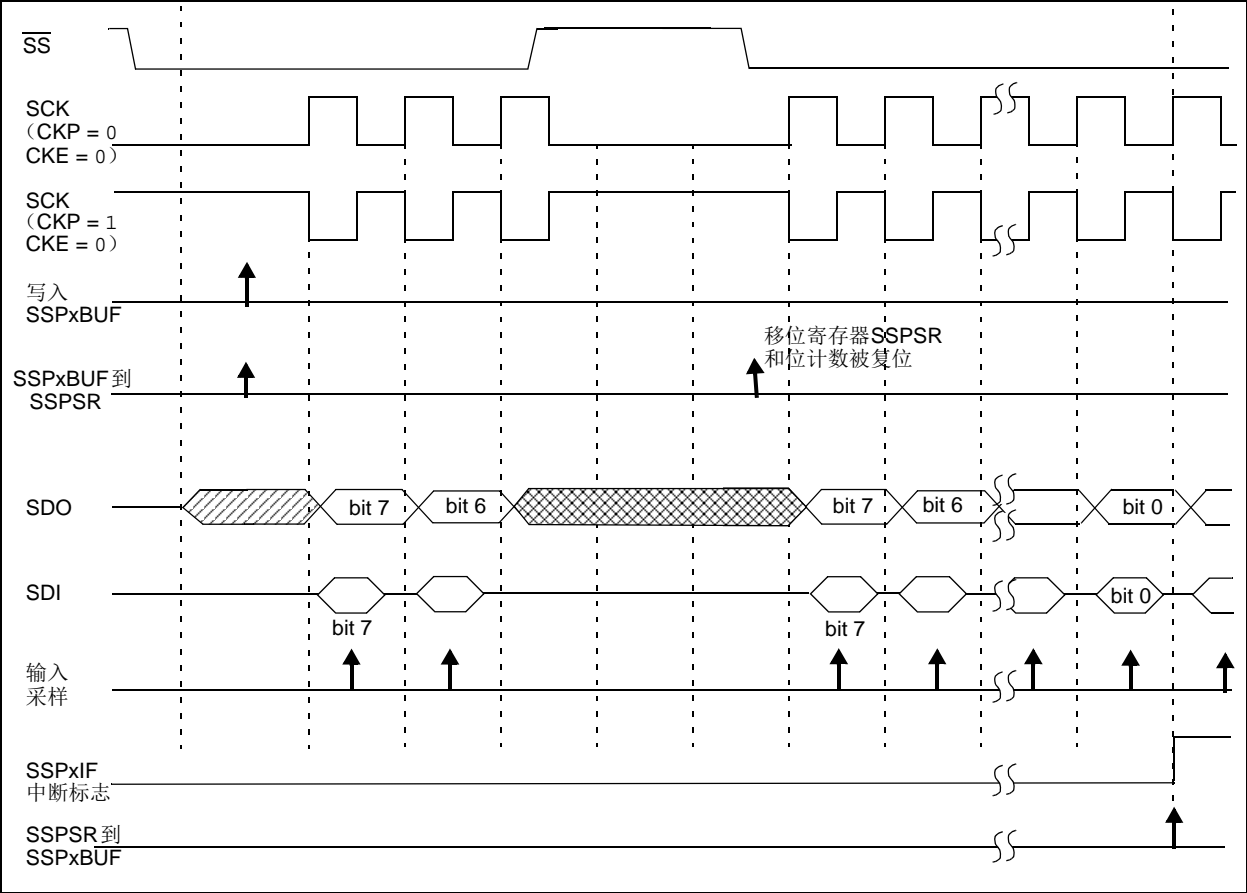




图24-9: SPI模式波形图（从模式，CKE = 0）

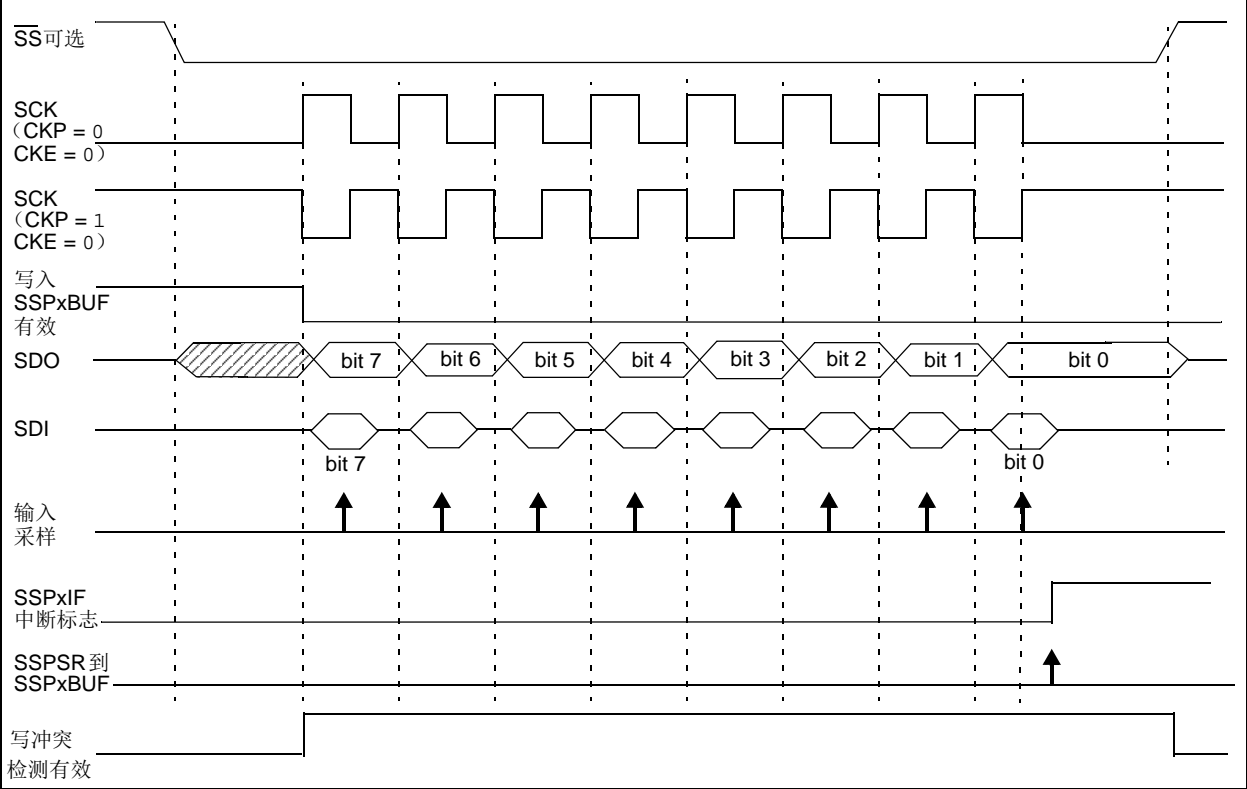
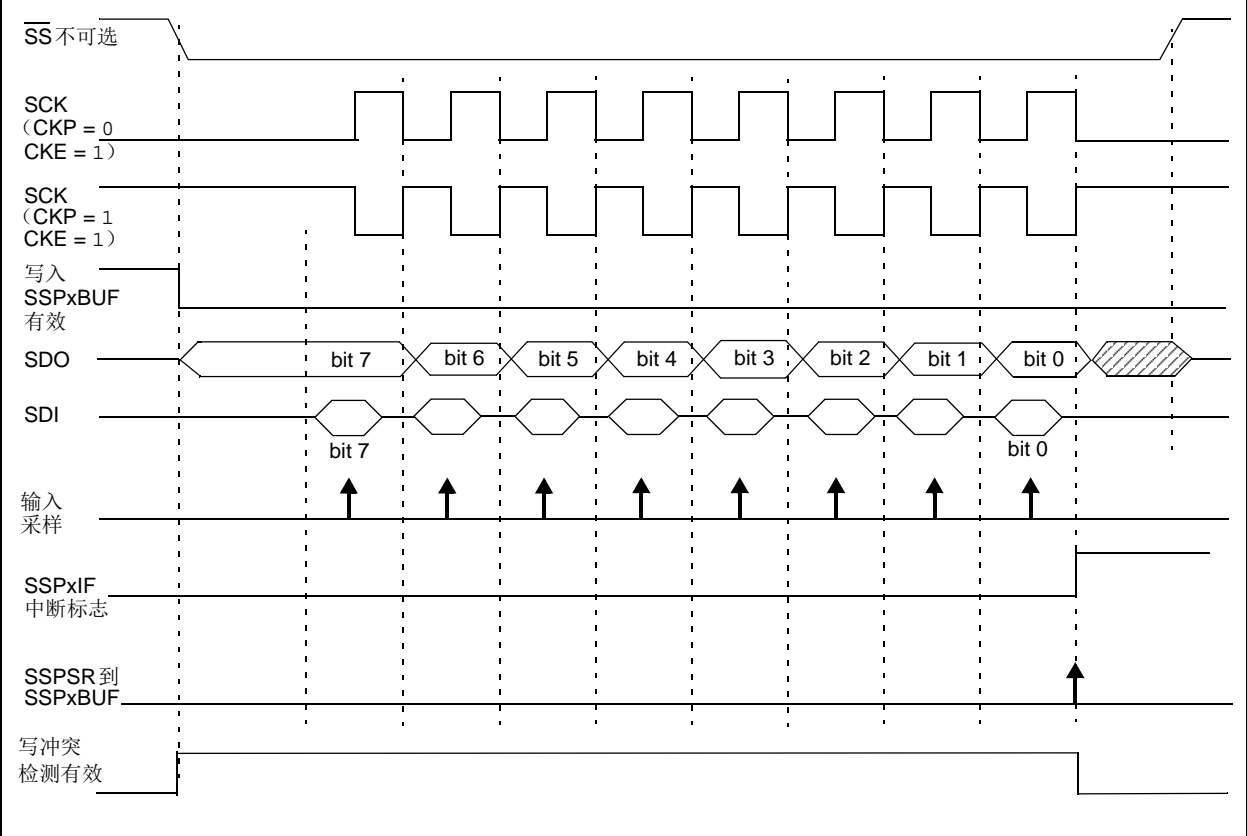


图24-10: SPI模式波形图（从模式，CKE = 1）



# PIC16(L)F1615/9

## 24.2.6 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状态唤醒。

如果不想从休眠模式退出，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24-1: 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	160
ANSELC	ANSC7 <sup>(2)</sup>	ANSC6 <sup>(2)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
INTCON	GIE	PEIE	TMR0IE	INTE	IOIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RxyPPS	—	—	—	RxyPPS<4:0>					180
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					182 和 180
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					182 和 180
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					182 和 180
SSP1BUF	同步串行端口接收缓冲/发送寄存器								271*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				316
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	315
SSP1STAT	SMP	CKE	D/ $\overline{A}$	P	S	R/ $\overline{W}$	UA	BF	315
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159
TRISB <sup>(2)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(2)</sup>	TRISC6 <sup>(2)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173

图注： — = 未实现位，读为0。SPI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: 未实现，读为1。

2: 仅限PIC16(L)F1619。

### 24.3 I<sup>2</sup>C 模式概述

I<sup>2</sup>C 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。

I<sup>2</sup>C 总线规定了两种信号连接：

- 串行时钟（SCL）
- 串行数据（SDA）

图24-11给出了MSSP模块在I<sup>2</sup>C模式下工作时的框图。

SCL和SDA连接都是双向的漏极开路线路，它们都需要使用用于电源电压的上拉电阻。线路下拉为地电压时，信号视为逻辑0；线路保持悬空时，信号视为逻辑1。

图24-11给出了分别配置为主器件和从器件的两个处理器之间的典型连接。

I<sup>2</sup>C 总线工作时可以有一个或多个主器件，以及一个或多个从器件。

对于给定器件，有4种可能的工作模式：

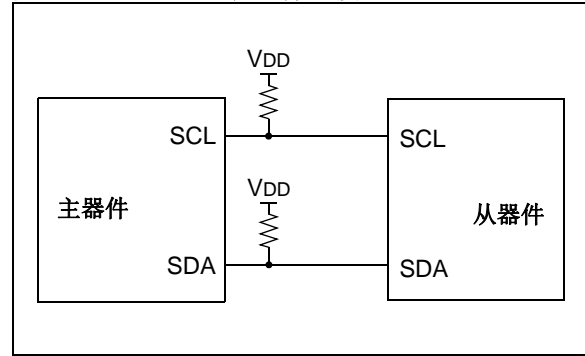
- 主发送模式（主器件向从器件发送数据）
- 主接收模式（主器件从从器件接收数据）
- 从发送模式（从器件向主器件发送数据）
- 从接收模式（从器件从主器件接收数据）

要开始进行通信，主器件需要以主发送模式启动。主器件送出启动位，后面跟随它希望进行通信的从器件的地址字节。后面再跟随单个读/写位，该位决定主器件是向从器件发送数据还是从从器件接收数据。

如果总线上存在所请求的从器件，从器件会使用应答位（也称为ACK）进行响应。然后，主器件会以发送模式或接收模式继续通信，从器件则以互补模式（分别为接收模式或发送模式）继续通信。

启动位由SCL线保持为高电平时SDA线的由高至低跳变来指示。地址和数据字节随后送出，先发送最高有效位（MSb）。在主器件希望从从器件读取数据时，送出的读/写位为逻辑1，在主器件希望向从器件写入数据时，该位为逻辑0。

图24-11： I<sup>2</sup>C 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。

I<sup>2</sup>C 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的组合报文或者启动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。

## 24.3.1 时钟延长

在从器件尚未完成数据处理时，它可以通过时钟延长这一过程来延迟更多数据的传输。寻址到的从器件可以在接收或发送一位数据之后将**SCL**时钟线保持为低电平，指示它尚未准备好继续。与从器件进行通信的主器件将会尝试上拉**SCL**线，以传输下一位数据，但它会检测到时钟线尚未被释放。由于**SCL**连接是漏极开路，所以从器件可以一直将线路保持为低电平，直到它准备好继续通信为止。

通过时钟延长，无法与发送器保持同速的接收器可以控制传入数据流。

## 24.3.2 仲裁

每个主器件都必须监视总线上是否出现启动位和停止位。如果器件检测到总线正忙，则在总线恢复为空闲状态之前，它无法开始新的报文。

但是，可能会有两个主器件尝试同时或几乎同时启动数据发送。发生这种情况时，将会开始仲裁过程。每个发送器会检查**SDA**数据线的电平，并将它与自己期望的电平进行比较。发现两个电平不匹配的发送器会在仲裁中失败，必须停止在**SDA**线上发送数据。

例如，如果一个发送器将**SDA**线保持为逻辑1（保留悬空），而第二个发送器将它保持为逻辑0（下拉为低电平），则结果是**SDA**线将为低电平。那么，第一个发送器会发现线路电平与期望电平不同，并断定有另一个发送器正在进行通信。

发现电平不同的第一个发送器将是仲裁失败的发送器，必须停止驱动**SDA**线。如果该发送器同时也是主器件，则它还必须停止驱动**SCL**线。然后，它可以在尝试重新启动数据发送之前监视线路上是否出现停止条件。与此同时，另一个未发现期望电平与**SDA**线实际电平不同的器件将继续原来的数据发送。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。

24.4 I<sup>2</sup>C 模式操作

所有MSSP I<sup>2</sup>C 通信都是针对字节的，并且会先移出 MSb。6个SFR 寄存器和2个中断标志用作模块与PIC<sup>®</sup> 单片机和用户软件的接口。模块通过两个引脚SDA 和SCL来与其他外部I<sup>2</sup>C 器件进行通信。

24.4.1 字节格式

I<sup>2</sup>C 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。

24.4.2 I<sup>2</sup>C 术语的定义

在I<sup>2</sup>C 通信的描述中存在一些用语和术语，它们具有特定于I<sup>2</sup>C 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 I<sup>2</sup>C 规范改写。

24.4.3 SDA 和 SCL 引脚

在SSPEN位置1的情况下选择任意I<sup>2</sup>C 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I<sup>2</sup>C 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。

24.4.4 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24-2: I<sup>2</sup>C 总线术语

术语	说明
发送器	将数据移送到总线上的器件。
接收器	从总线上移入数据的器件。
主器件	启动数据传输、产生时钟信号和终止数据传输的器件。
从器件	主器件寻址到的器件。
多主器件	有多个器件可以启动数据传输的总线。
仲裁	用于确保每次只有一个主器件控制总线的过程。仲裁获胜可以确保报文不会被损坏。
同步	用于将总线上两个或更多器件的时钟进行同步的过程。
空闲	没有任何主器件在控制总线，并且SDA 和SCL 线均为高电平。
有效	每当有一个或多个主器件在控制总线时。
可寻址的从器件	已接收到匹配地址、并且正在由主器件提供时钟的从器件。
匹配地址	送入从器件中、并与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。

24.4.5 启动条件

I<sup>2</sup>C 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24-12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<sup>2</sup>C 规范，该规范规定启动时不能发生总线冲突。

24.4.6 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。

24.4.7 重复启动条件

重复启动条件在每次停止条件有效的时候有效。如果主器件希望在终止当前传输之后保持总线，主器件可以发出重复启动条件。重复启动对从器件产生的影响与启动条件相同，即复位所有从器件逻辑并使之准备接收一个地址。主器件可以寻址同一个或另一个从器件。图24-13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。

24.4.8 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。

图24-12: I<sup>2</sup>C 启动和停止条件

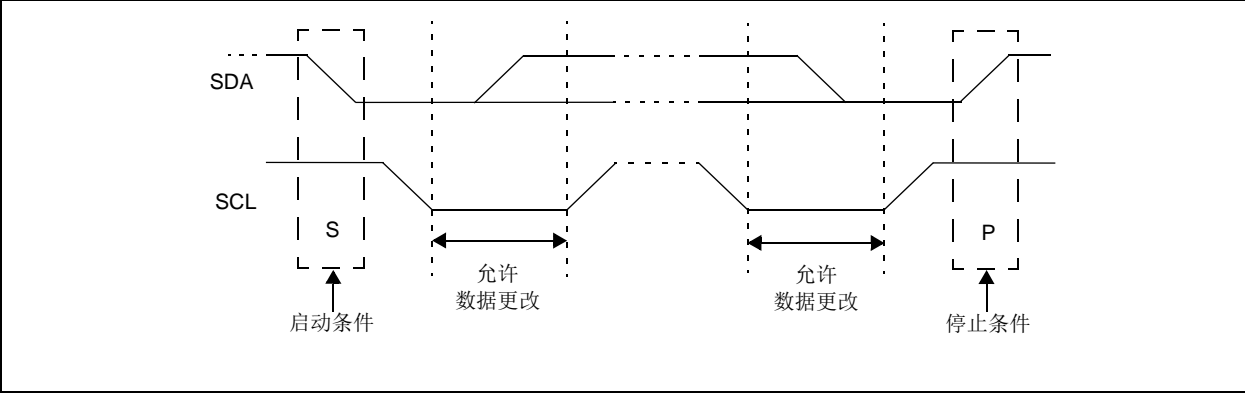
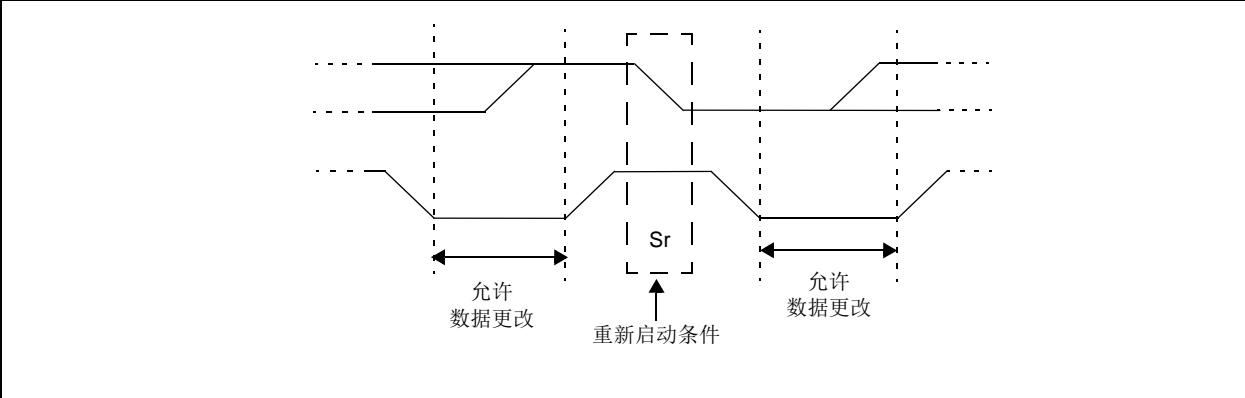


图24-13: I<sup>2</sup>C 重复启动条件



#### 24.4.9 应答序列

在I<sup>2</sup>C中，所有传输字节的第9个SCL脉冲都专门用作应答信号。它使接收器件可以通过将SDA下拉为低电平来响应发送器。发送器在该时间内必须释放对线路的控制，以移入响应信号。应答(ACK)是低电平有效信号，它会将SDA线下拉为低电平，用于指示发送器接收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。

#### 24.5 I<sup>2</sup>C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，会将SSPxIF置1。

##### 24.5.1 从模式地址

SSPxADD寄存器(寄存器24-6)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24-5)会影响地址匹配过程。更多信息，请参见第24.5.8节“SSP掩码寄存器”。

##### 24.5.1.1 I<sup>2</sup>C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。

##### 24.5.1.2 I<sup>2</sup>C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位地址的两个MSb，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。

## 24.5.2 从接收

当接收到的匹配地址字节的 $\overline{R/\overline{W}}$ 位清零时，SSPxSTAT寄存器的 $\overline{R/\overline{W}}$ 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上溢条件时，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24-4。

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24.5.6.2节“10位寻址模式”。

### 24.5.2.1 7位寻址接收

本节介绍在7位寻址模式下，配置为I<sup>2</sup>C从器件的MSSP模块的标准事件序列。图24-14和图24-15用直观的方式对此作了说明。

以下列出了实现I<sup>2</sup>C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 $\overline{R/\overline{W}}$ 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 $\overline{ACK}$ ，并将SSPxIF位置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 $\overline{ACK}$ ，并将SSPxIF位置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。

### 24.5.2.2 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I<sup>2</sup>C通信使用这些选项时，从器件软件需要执行的步骤。图24-16显示了同时使用地址和数据保持功能的模块。图24-17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.  $\overline{R/\overline{W}}$ 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 $\overline{ACK}$ 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. 从器件通过将ACKDT置1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 $\overline{ACK}$ 之后置1，不会在NACK之后置1。
9. 如果SEN = 1，从器件硬件会在 $\overline{ACK}$ 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 $\overline{ACK}$  = 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SSPxSTAT寄存器的P位才能知道停止条件。



图24-14: I<sup>2</sup>C从模式，7位地址，接收 (SEN = 0, AHEN = 0, DHEN = 0)

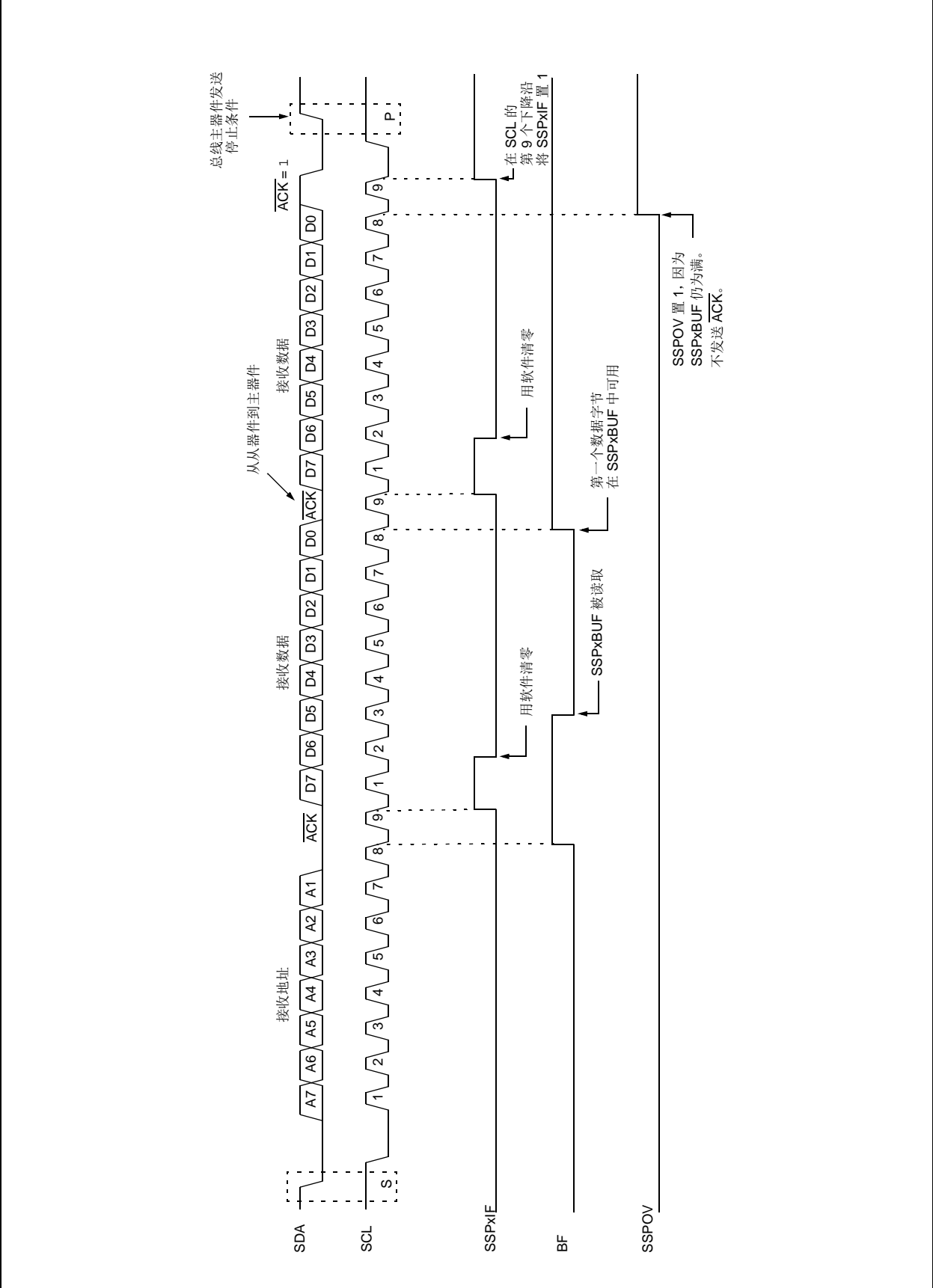


图24-15: I<sup>2</sup>C从模式，7位地址，接收 (SEN = 1, AHEN = 0, DHEN = 0)

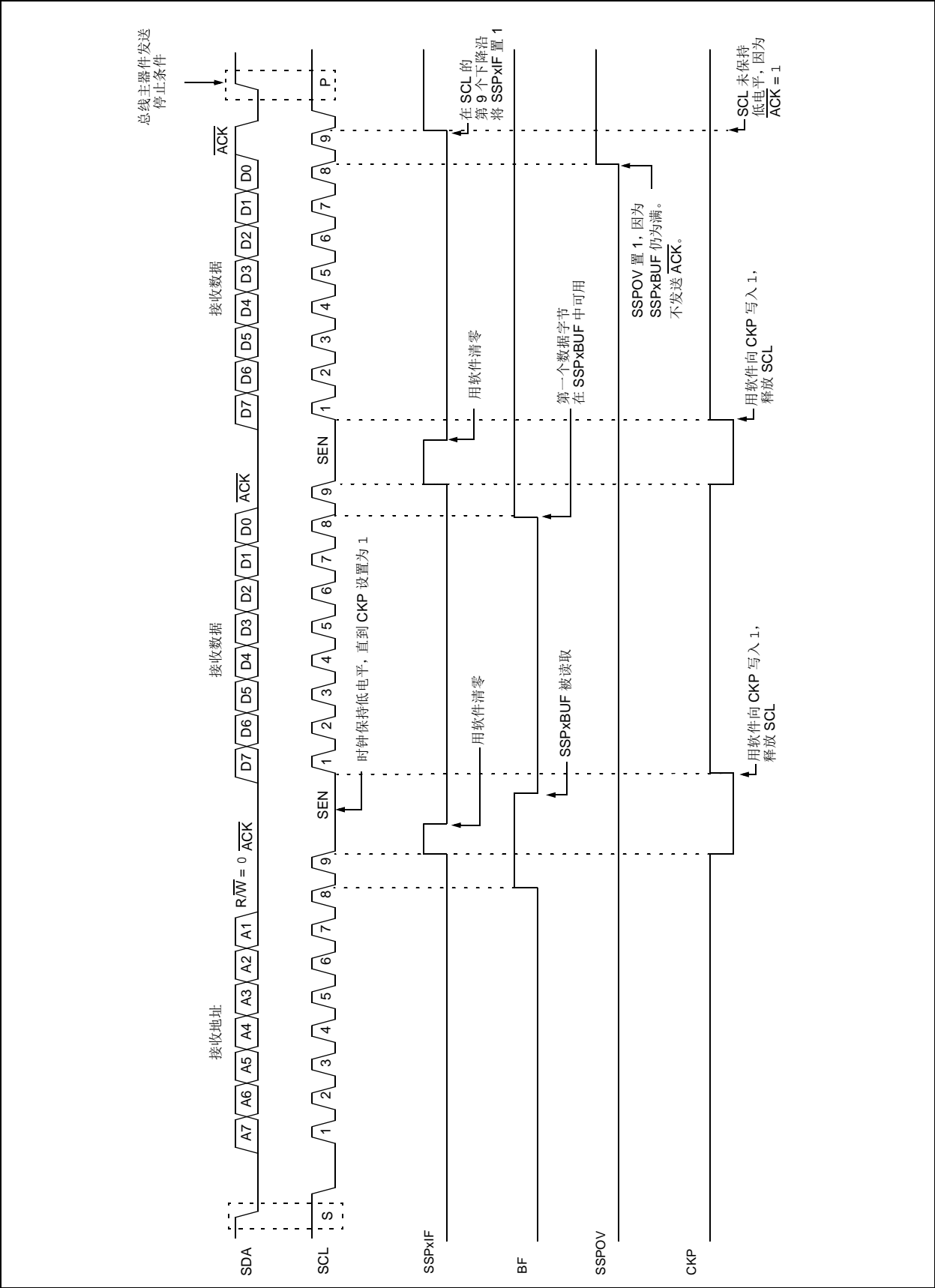


图24-16: I<sup>2</sup>C从模式，7位地址，接收 (SEN = 0, AHEN = 1, DHEN = 1)

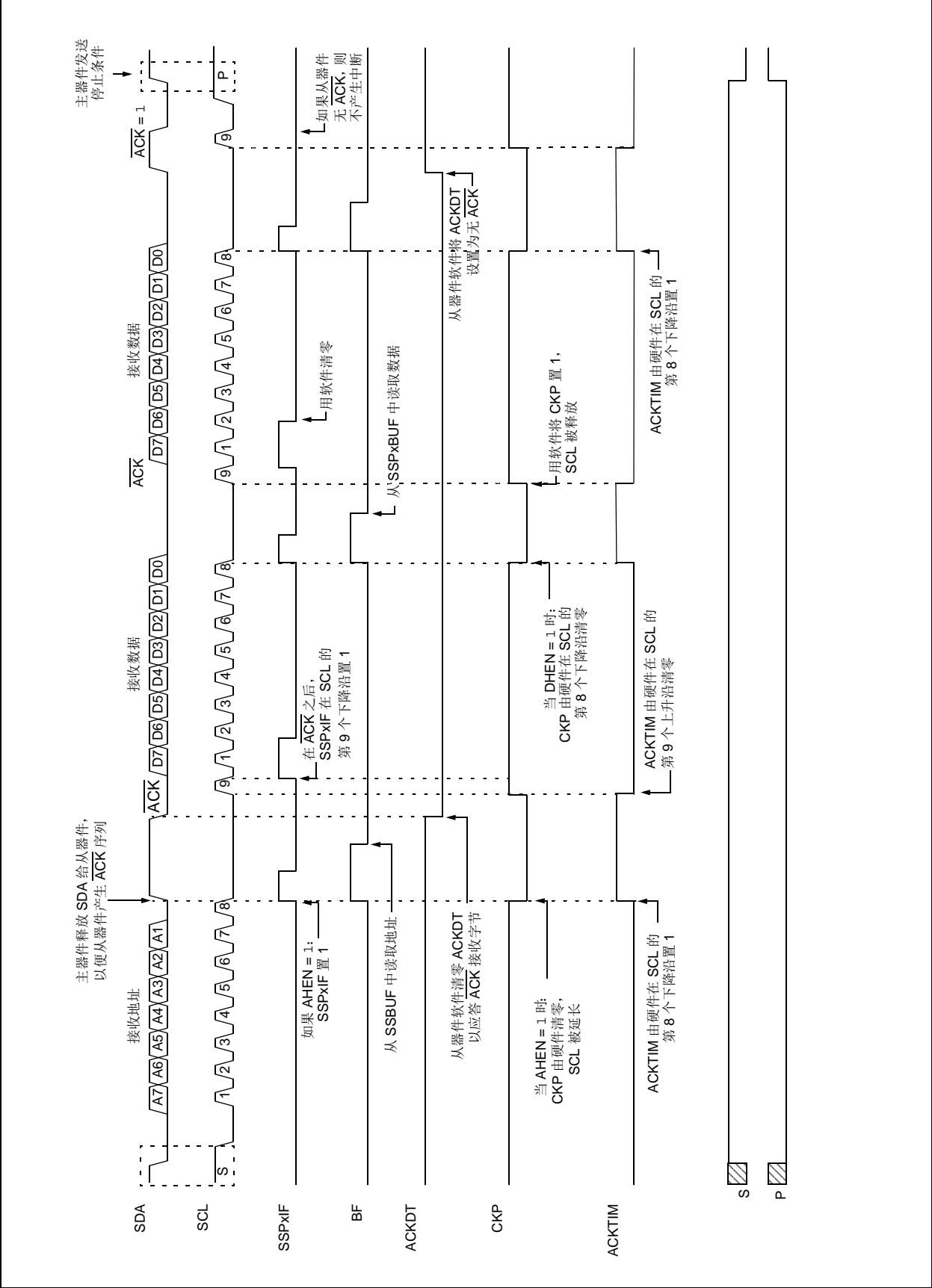
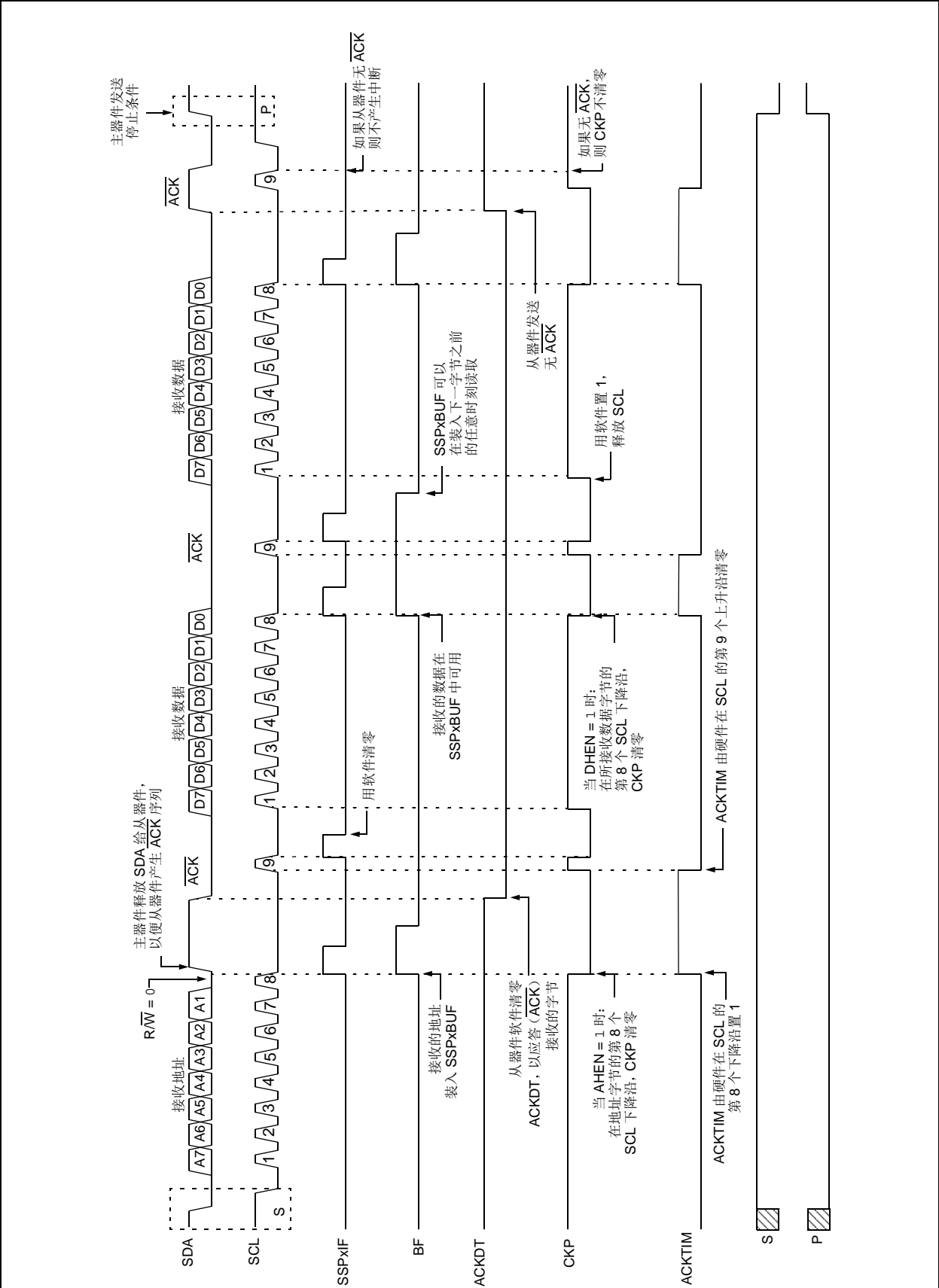


图24-17: I<sup>2</sup>C从模式，7位地址，接收 (SEN = 1, AHEN = 1, DHEN = 1)



## 24.5.3 从发送

当输入地址字节的  $\overline{R/\overline{W}}$  位置 1 并发生地址匹配时，SSPxSTAT 寄存器的  $\overline{R/\overline{W}}$  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  $\overline{ACK}$  脉冲。

在  $\overline{ACK}$  之后，从器件硬件会清零 CKP 位，并且 SCL 引脚保持低电平（更多详细信息，请参见第 24.5.6 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 为高电平期间 SDA 信号是有效的。

来自主接收器的  $\overline{ACK}$  脉冲将在第 9 个 SCL 输入脉冲的上升沿锁存。该  $\overline{ACK}$  值会被复制到 SSPxCON2 寄存器的 ACKSTAT 位中。如果 ACKSTAT 置 1（无  $\overline{ACK}$  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  $\overline{ACK}$  时，从器件会进入空闲状态，并等待出现另一个启动位。如果 SDA 线为低电平（ $\overline{ACK}$ ），则必须将下一个要发送的数据装入 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。

### 24.5.3.1 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，PIR2 寄存器的 BCL1IF 位会置 1。在检测到总线冲突时，从器件会变为空闲状态，等待再次被寻址。用户软件可以通过使用 BCL1IF 位来处理从器件总线冲突。

## 24.5.3.2 7 位发送

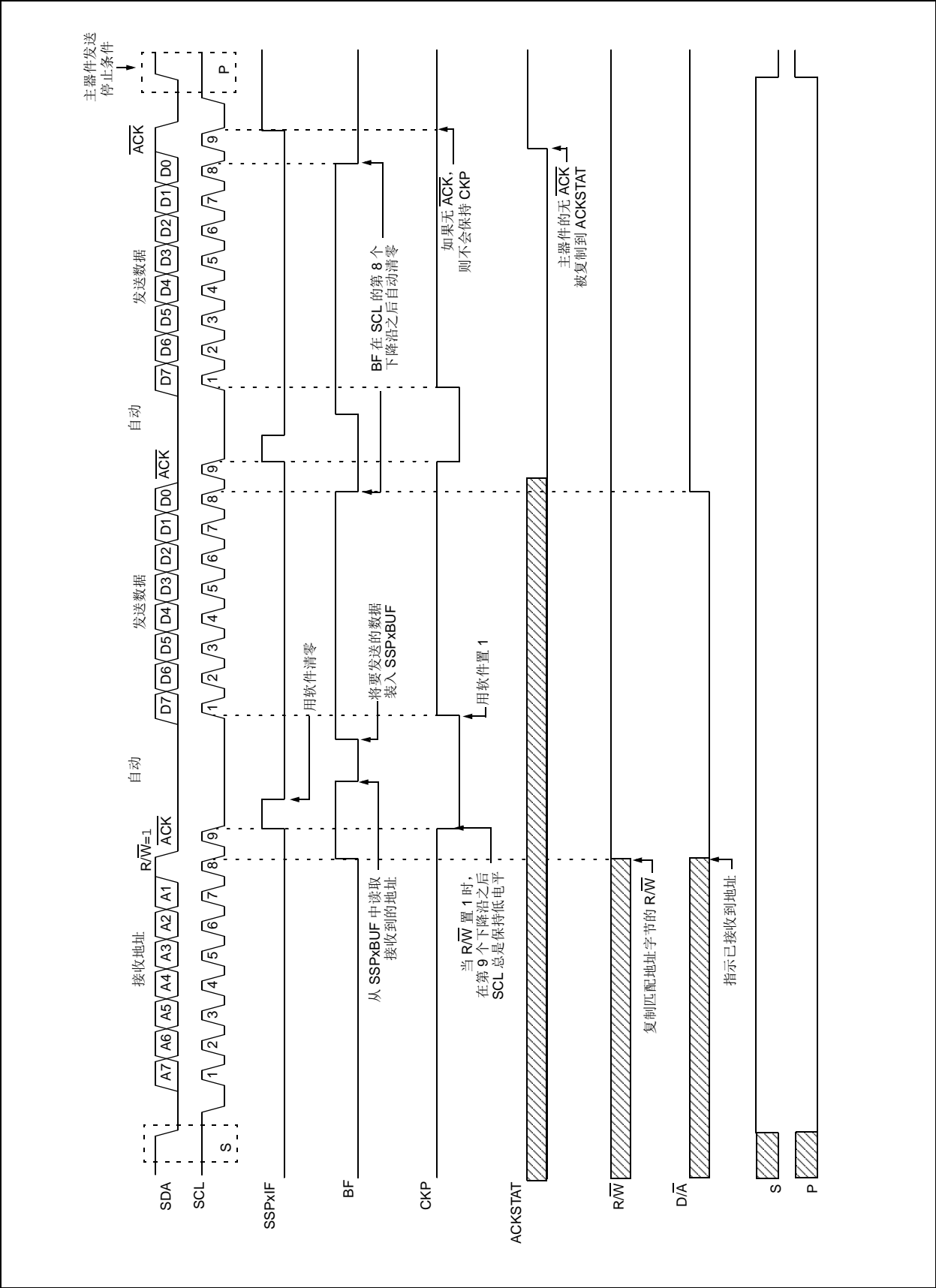
主器件可以向从器件发送读请求，然后从从器件中移出数据。以下列表列出了在实现标准数据发送时，从器件软件需要执行的操作。图 24-18 可用作该列表的参考。

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  $\overline{R/\overline{W}}$  位置 1 的匹配地址，并将 SSPxIF 位置 1。
4. 从器件硬件产生  $\overline{ACK}$ ，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.  $\overline{R/\overline{W}}$  置 1，所以 CKP 会在  $\overline{ACK}$  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  $\overline{ACK}$  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 ( $\overline{ACK}$ )，时钟将被延长。  
**2:** ACKSTAT 是惟一一个在 SCL 上升沿（第 9 个）而不是下降沿发生更新的位。

13. 对于每个发送字节重复步骤 9-13。
14. 如果主器件发送无  $\overline{ACK}$ ，则不会保持时钟，但 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。

图24-18: I<sup>2</sup>C从模式，7位地址，发送 (AHEN = 0)



## 24.5.3.3 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24-19给出了在使能AHEN时7位地址从发送的标准波形图。

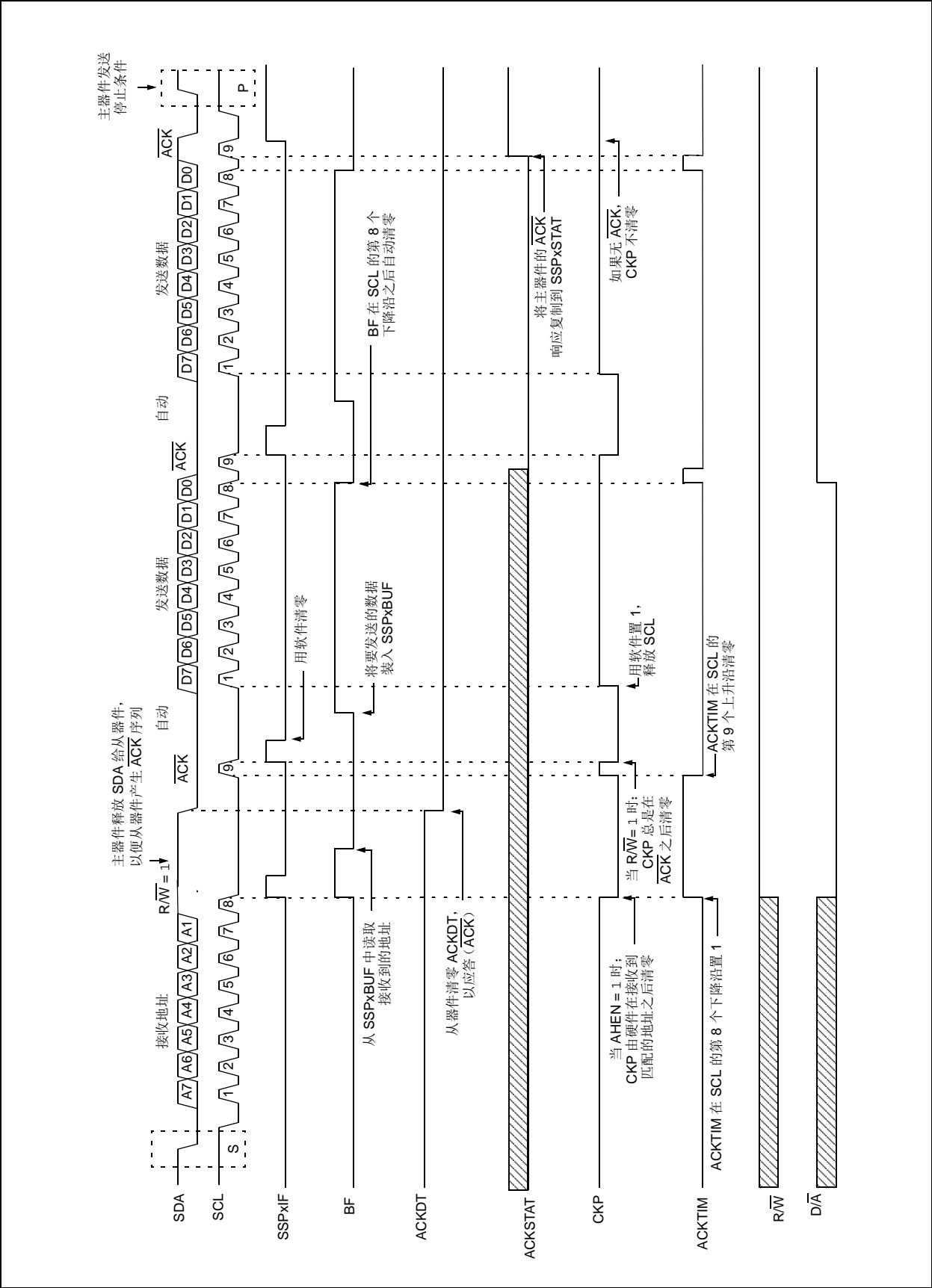
1. 总线启动时为空闲模式。
2. 主器件发送启动条件；SSPxSTAT的S位置1；如果允许在检测到启动条件时产生中断，则SSPxIF会置1。
3. 主器件发送R/W位置1的匹配地址。在SCL线的第8个下降沿之后，CKP位清零，并产生SSPxIF中断。
4. 从器件软件清零SSPxIF。
5. 从器件软件读取SSPxCON3寄存器的ACKTIM位，以及SSPxSTAT寄存器的R/W和D/A位，以确定中断源。
6. 从器件从SSPxBUF寄存器中读取地址值，使BF位清零。
7. 从器件软件根据该信息确定它是产生 $\overline{\text{ACK}}$ 还是产生无 $\overline{\text{ACK}}$ ，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 $\overline{\text{ACK}}$ 值。
10. 如果R/W位置1，则在 $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ 值。
15. 从器件硬件将 $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ ，从器件会释放总线，让主器件可以发送停止条件和结束通信。

**注：** 主器件必须对于最后一个字节发送无 $\overline{\text{ACK}}$ ，以确保从器件释放SCL线来接收停止条件。

图24-19: I<sup>2</sup>C从模式，7位地址，发送 (AHEN = 1)





## 24.5.4 从模式10位地址接收

本节介绍在10位寻址模式下，配置为I<sup>2</sup>C从器件的MSSP模块的标准事件序列。

图24-20用直观的方式对此作了说明。

以下列出了实现I<sup>2</sup>C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置1。
3. 主器件发送R $\overline{W}$ 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。

## 24.5.5 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24-21可以用作AHEN置1时10位寻址模式下从器件的参考图示。

图24-22给出了10位寻址模式下从发送器的标准波形图。

图24-20: I<sup>2</sup>C从模式，10位地址，接收 (SEN = 1, AHEN = 0, DHEN = 0)

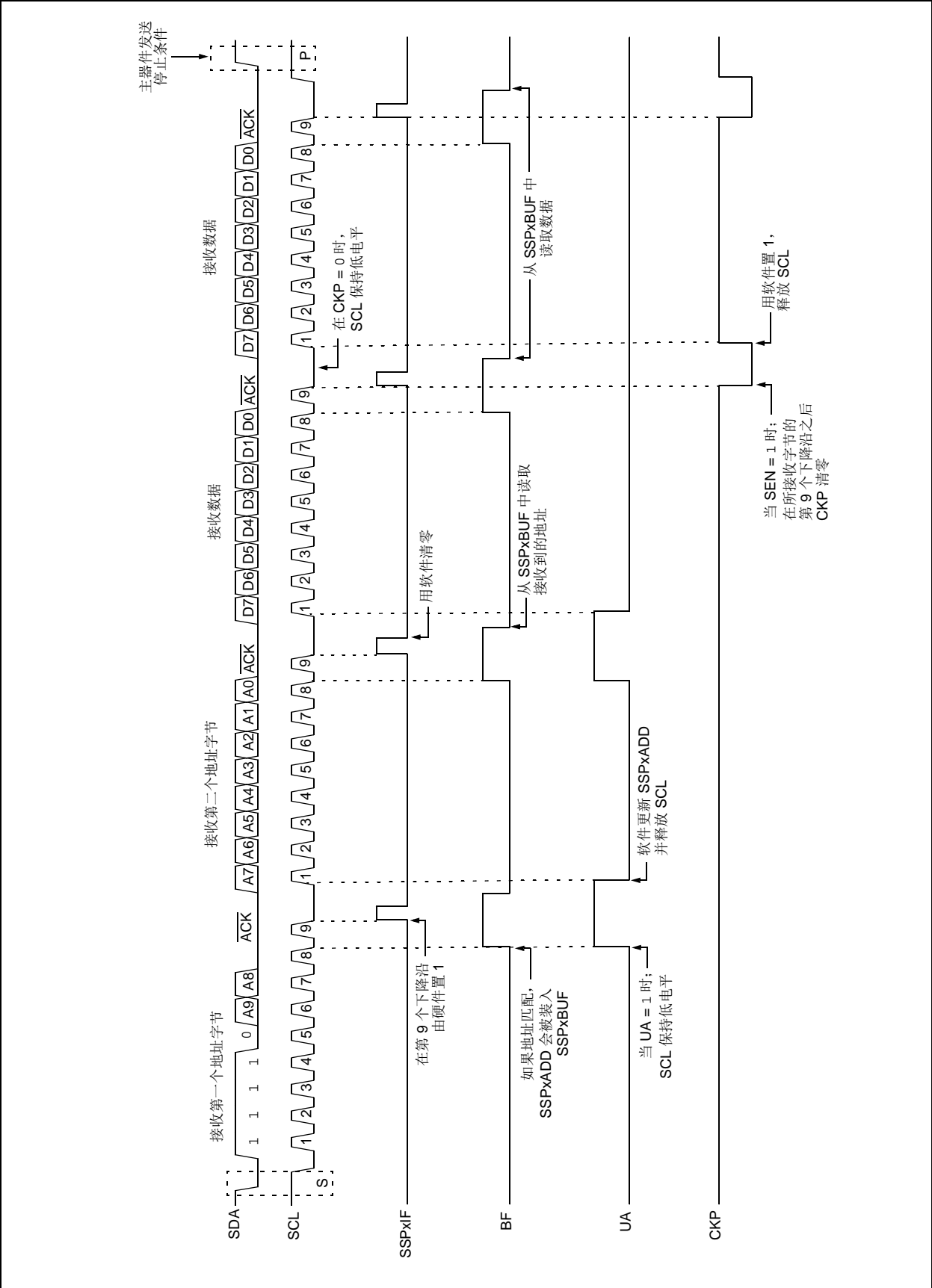


图24-21: I<sup>2</sup>C从模式，10位地址，接收 (SEN = 0, AHEN = 1, DHEN = 0)

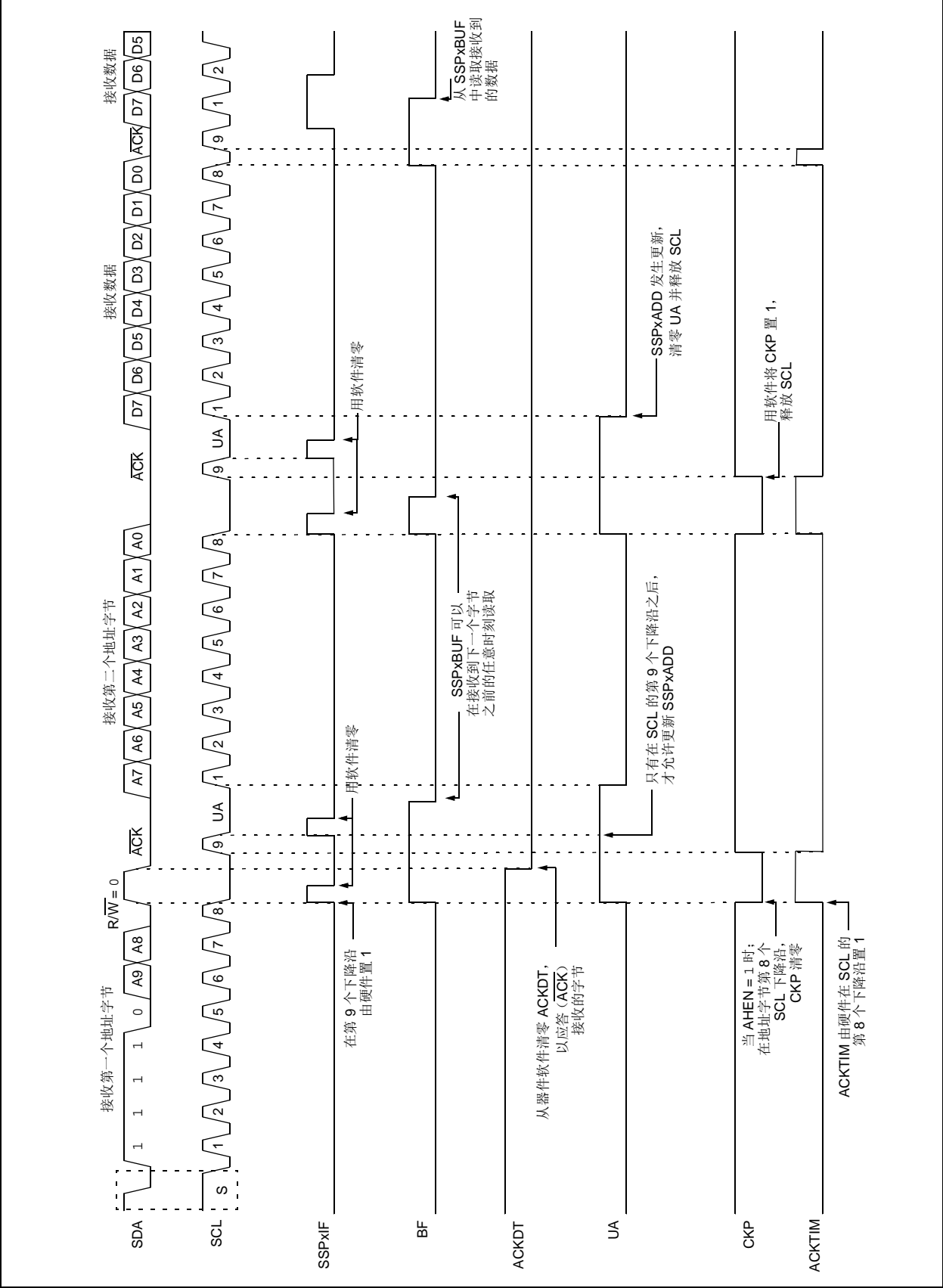
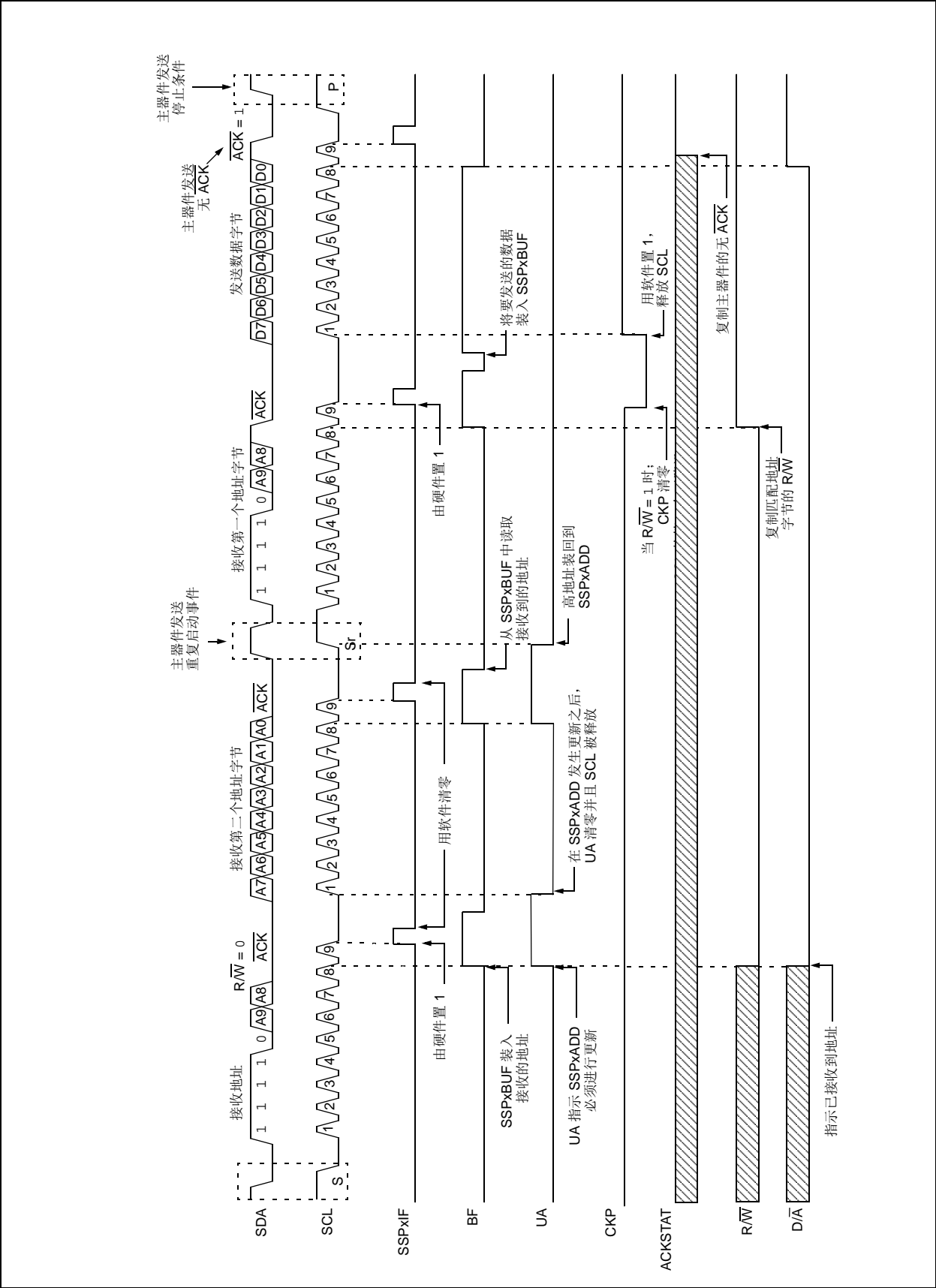


图24-22: I<sup>2</sup>C从模式，10位地址，发送 (SEN = 0, AHEN = 0, DHEN = 0)



24.5.6 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。

24.5.6.1 正常时钟延长

如果SSPxSTAT的R/W位置1（读请求），则在ACK之后，从器件硬件会清零CKP。这让从器件可以有时间使用要传输给主器件的数据更新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，先前版本将不会延长时钟及清零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。现在，对于读请求，总是会将该位清零。

24.5.6.2 10位寻址模式

在10位寻址模式下，当UA位置1时，时钟总是会被延长。这是无需清零CKP就会延长SCL的惟一情形。在写入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。

24.5.6.3 字节无应答

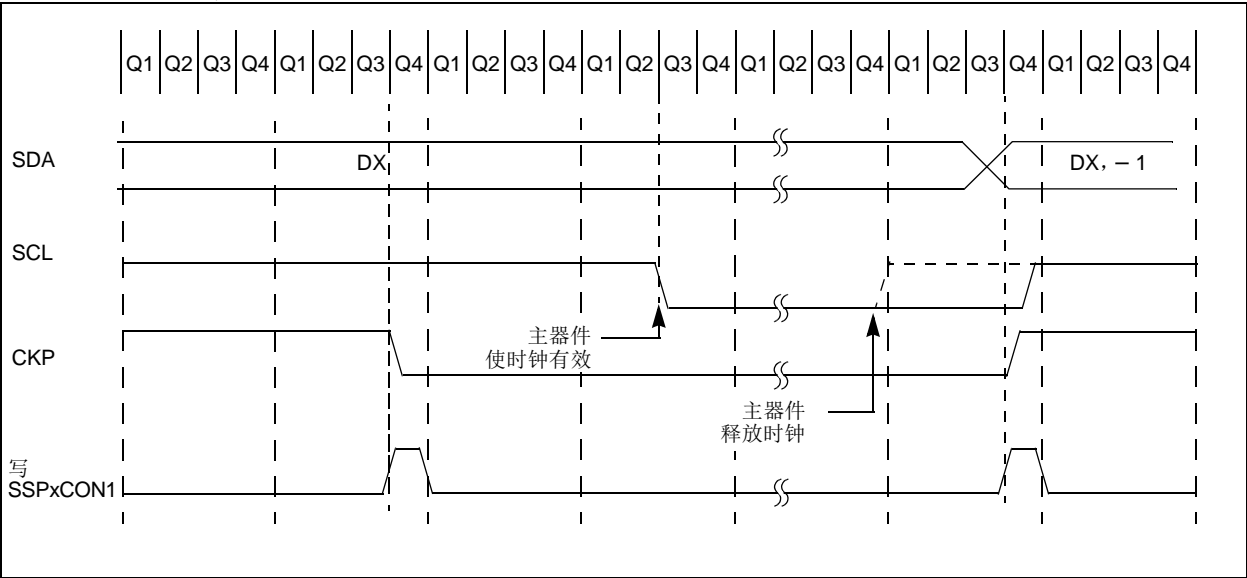
当SSPxCON3的AHEN位置1时，则在所接收匹配地址字节的第8个SCL下降沿之后，硬件会将CKP清零。当SSPxCON3的DHEN位置1时，在所接收数据的第8个SCL下降沿之后，CKP会被清零。

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。

24.5.6.4 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输出保持为低电平。因此，CKP位不会将SCL线拉为低电平，除非外部I<sup>2</sup>C主器件已将SCL线拉为低电平。SCL输出将保持低电平，直到CKP位置1且I<sup>2</sup>C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24-23）。

图24-23: 时钟同步时序



## 24.5.7 广播呼叫地址支持

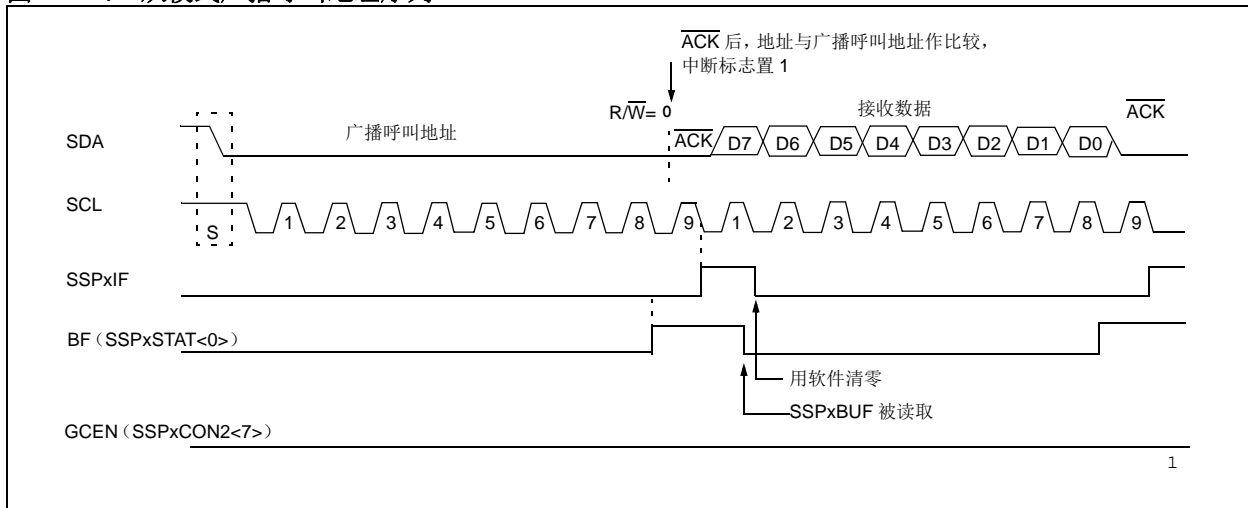
在I<sup>2</sup>C总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有器件都应该发送一个应答信号来响应。

广播呼叫地址是I<sup>2</sup>C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24-24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24-24： 从模式广播呼叫地址序列



## 24.5.8 SSP掩码寄存器

SSP掩码 (SSPxMSK) 寄存器 (寄存器24-5) 在I<sup>2</sup>C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。

24.6 I<sup>2</sup>C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。

通过在检测到启动和停止条件时产生中断来支持主操作模式。停止(P)位和启动(S)位在复位或禁止MSSP模块时清零。当P位置1或总线空闲时，可取得I<sup>2</sup>C总线的控制权。

在固件控制的主模式下，用户代码根据启动位和停止条件检测执行所有的I<sup>2</sup>C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I<sup>2</sup>C主模式时，MSSP模块不允许事件排队。例如，在启动条件结束前，不允许用户发出启动条件以及立即写SSPxBUF寄存器以启动传输。在这种情况下，将不会执行写SSPxBUF，WCOL位将被置1，指示没有发生对SSPxBUF的写操作。

**2:** 处于主模式时，如果SEN/PEN位清零，并且启动/停止条件完成，则会屏蔽启动/停止检测和产生中断。

24.6.1 I<sup>2</sup>C主模式操作

主器件产生所有的串行时钟脉冲、启动条件和停止条件。以停止条件或重复启动条件结束传输过程。因为重复启动条件也是下一次串行传输的开始，因此I<sup>2</sup>C总线不会被释放。

在主发送器模式下，串行数据通过SDA输出，而串行时钟由SCL输出。发送的第一个字节包括接收器件的从器件地址（7位）和读/写（R/W）位。在这种情况下，R/W位将为逻辑0。一次发送8位串行数据。每发送一个字节，都会接收到一个应答位。输出启动和停止条件指示串行传输的开始和结束。

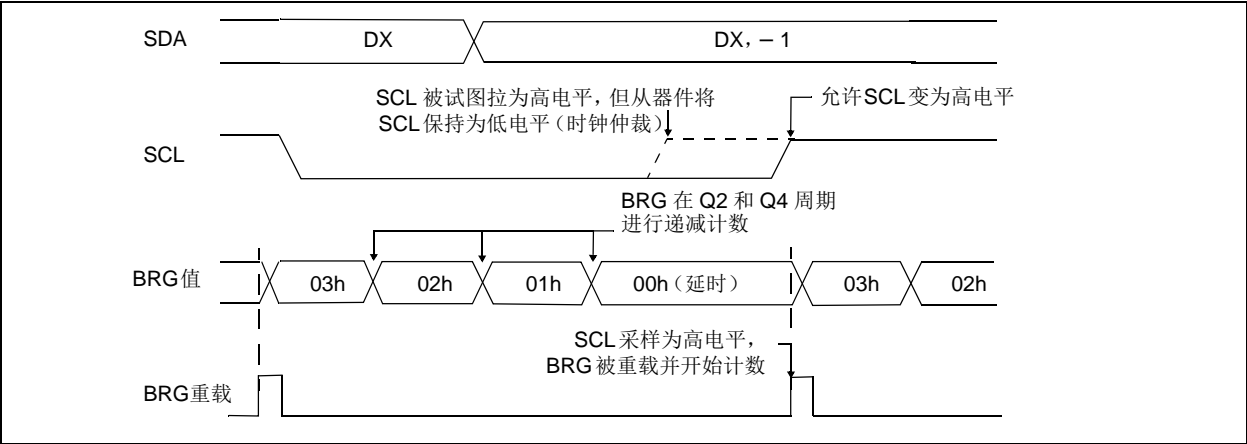
在主接收模式下，发送的第一个字节包括作为发送器件的从地址（7位）和R/W位。在这种情况下，R/W位将为逻辑1。因此，发送的第一个字节是一个7位从地址，后面跟随一个1来指示接收位。串行数据通过SDA接收，而串行时钟由SCL输出。一次接收8位串行数据。每接收到一个字节，都会发送一个应答位。启动和停止条件指示发送的开始和结束。

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24.7节“波特率发生器”。

24.6.2 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24-25）。

图24-25：带有时钟仲裁的波特率发生器时序



24.6.3 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位被置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。



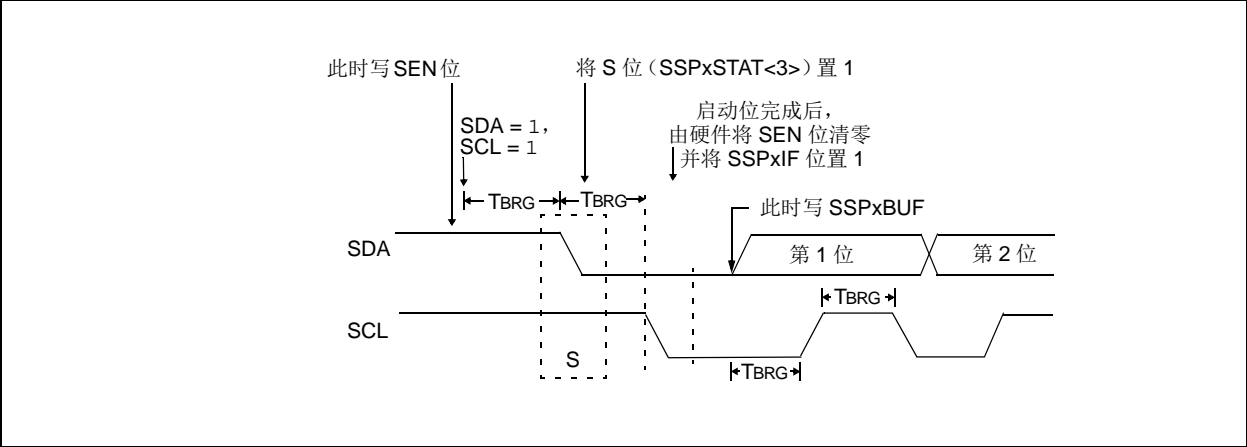
24.6.4 I<sup>2</sup>C主模式启动条件时序

要发出启动条件（图24-26），用户应将SSPxCON2寄存器的启动使能位SEN置1。如果SDA和SCL引脚被采样为高电平，则波特率发生器会重新装入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BCL1IF置1，启动条件中止，I<sup>2</sup>C 模块复位到空闲状态。
- 2： Philips I<sup>2</sup>C规范规定启动时不能发生总线冲突。

图24-26： 第一个启动位时序



# PIC16(L)F1615/9

### 24.6.5 I<sup>2</sup>C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24-27）。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，

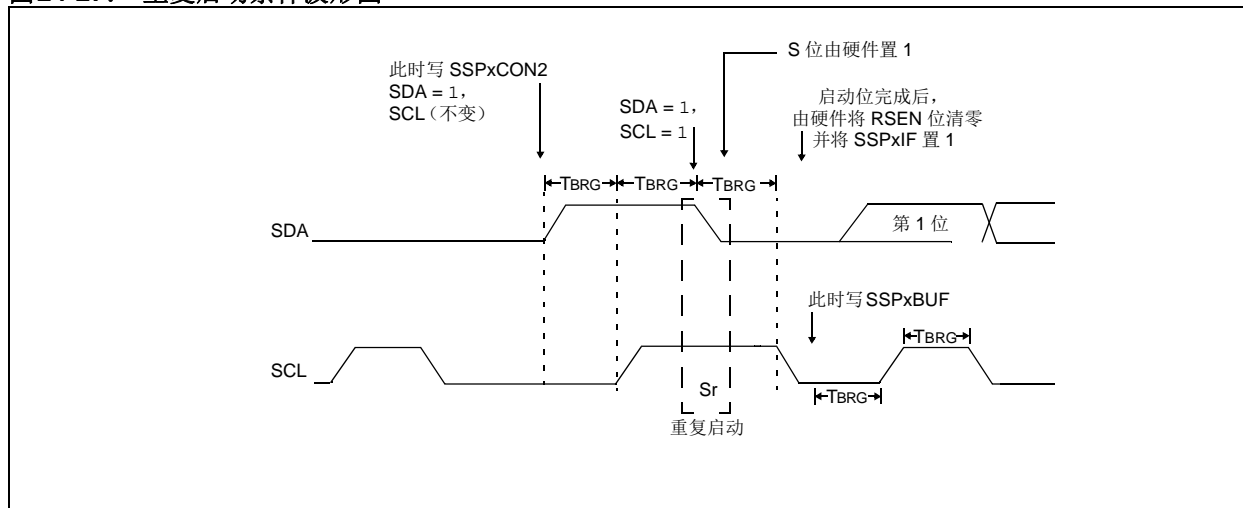
SDA 引脚保持低电平。一旦在 SDA 和 SCL 引脚上检测到启动条件，SSPxSTAT 寄存器的 S 位就会被置 1。SSPxIF 位在波特率发生器超时之前不会被置 1。

**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试图发送一个数据 1。

图24-27: 重复启动条件波形图



### 24.6.6 I<sup>2</sup>C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24-28）。

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。

#### 24.6.6.1 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。

#### 24.6.6.2 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）。

在下次发送前WCOL必须用软件清零。

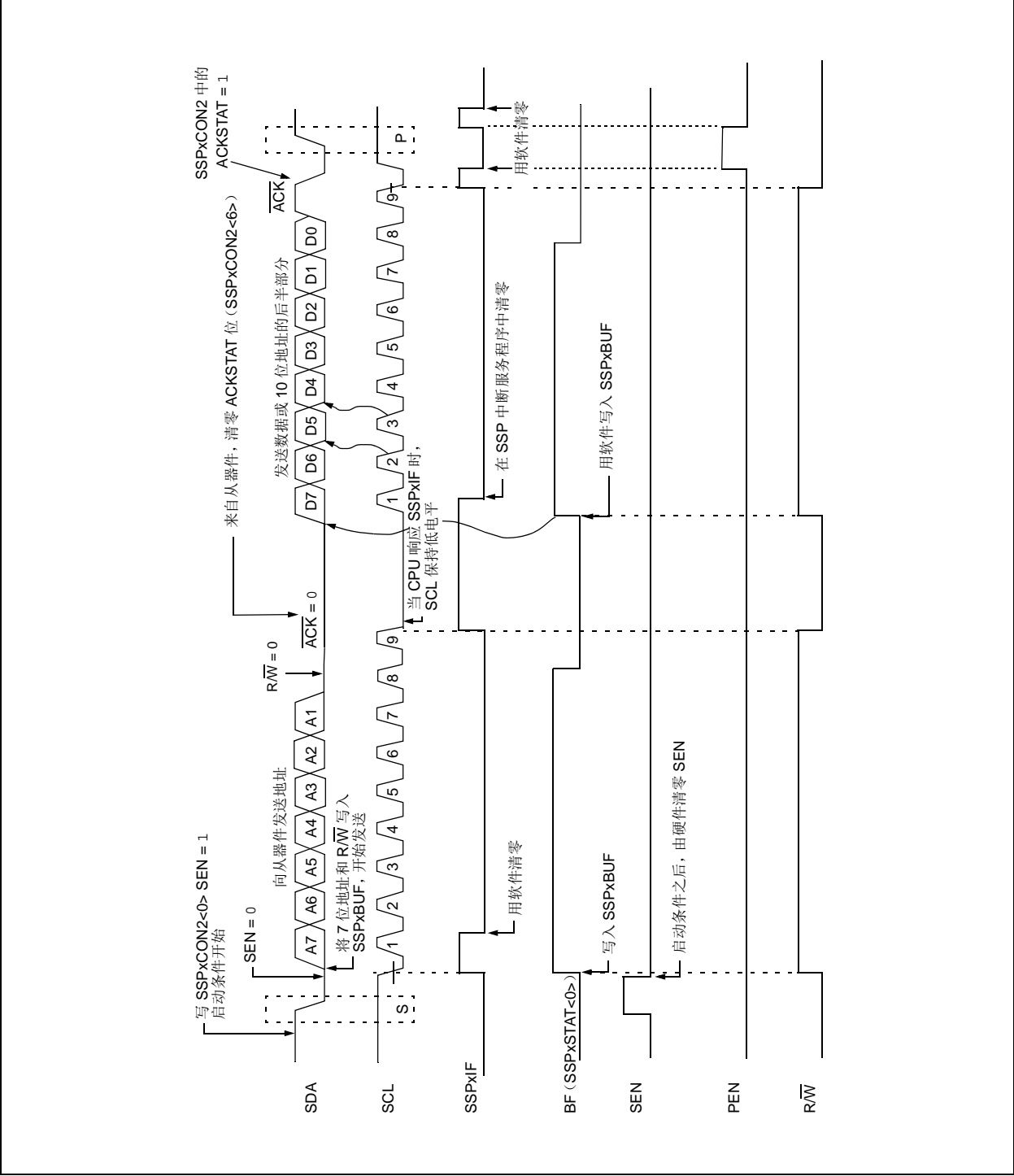
### 24.6.6.3 ACKSTAT状态标志

在发送模式下，当从器件发送应答(ACK = 0)时，SSPxCON2寄存器的ACKSTAT位被清零；当从器件没有应答(ACK = 1)时，该位被置1。从器件在识别出其地址（包括广播呼叫地址）或正确接收数据后，会发送一个应答。

#### 24.6.6.4 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. 器件地址从SDA引脚移出，直到发送完所有8位地址数据。数据发送会在写入SSPxBUF后立刻开始。
7. 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。
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位地址数据。
11. 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。
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。

图24-28: I<sup>2</sup>C主模式波形图 (发送, 7位或10位地址)



### 24.6.7 I<sup>2</sup>C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24-29）。

**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。

#### 24.6.7.1 BF状态标志

在接收操作中，将地址或数据字节从SSPSR装入SSPxBUF时，BF位被置1。在读SSPxBUF寄存器时将其清零。

#### 24.6.7.2 SSPOV状态标志

在接收操作中，当SSPSR接收到8位数据且BF标志位已经在上一次接收中置1时，SSPOV位置1。

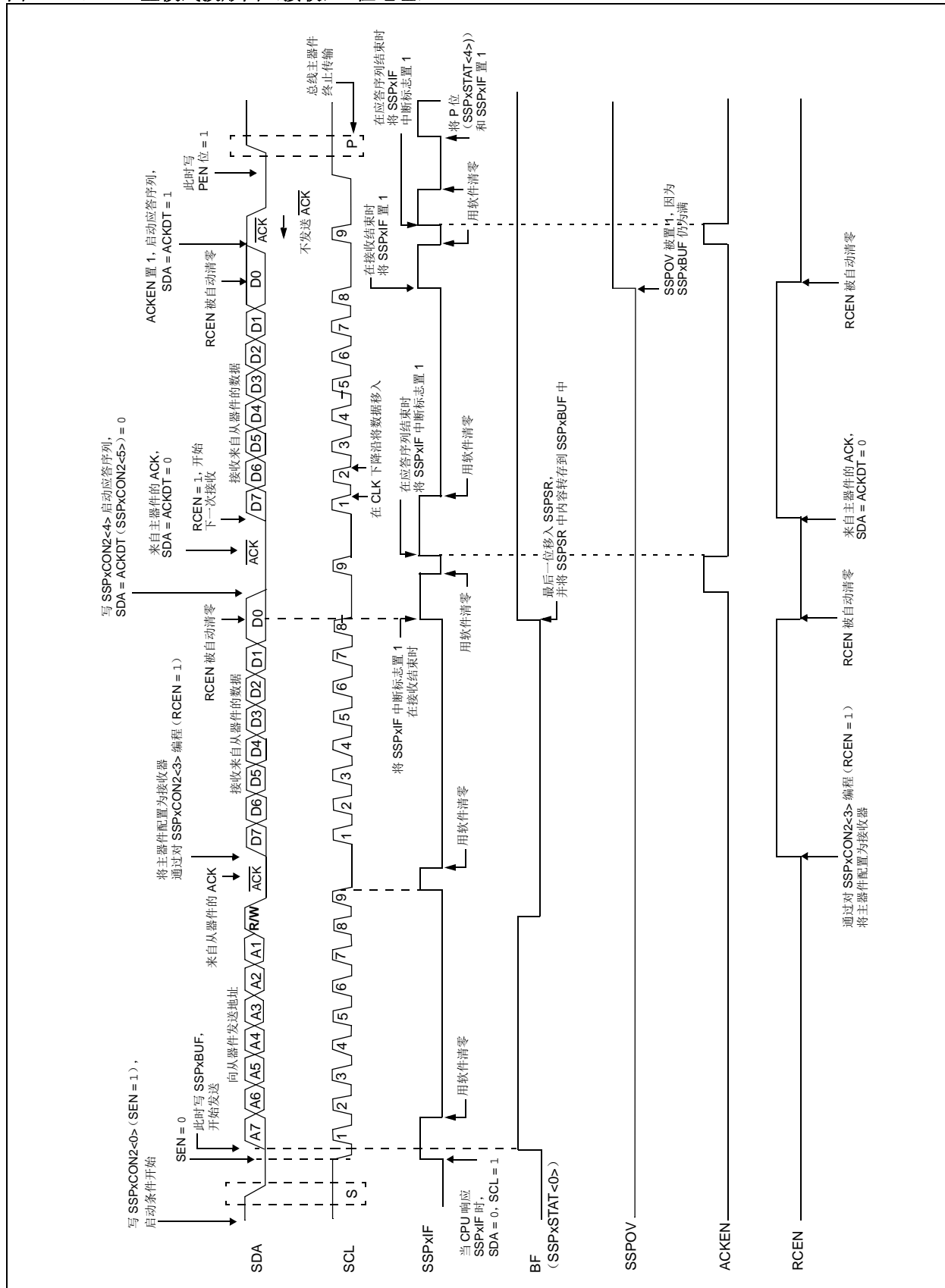
#### 24.6.7.3 WCOL状态标志

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，同时缓冲区内内容不变（未发生写操作）。

### 24.6.7.4 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位地址数据。数据发送会在写入SSPxBUF后立刻开始。
6. 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。
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 $\overline{\text{ACK}}$ ，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 $\overline{\text{ACK}}$ 或停止条件来结束通信。

**图24-29: I<sup>2</sup>C主模式波形图（接收，7位地址）**



24.6.8 应答序列时序

通过将SSPxCON2寄存器的应答序列使能位ACKEN置1，可以使能应答序列。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位置1。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24-30）。

24.6.8.1 WCOL状态标志

如果在应答序列进行过程中用户写SSPxBUF，则WCOL位置1，同时缓冲区内容不变（未发生写操作）。

24.6.9 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线采样为低电平时，波特率发生器被重载并递减计数至0。当波特率发生器发生超时，SCL引脚被拉为高电平，在一个TBRG（波特率发生器计满返回周期）之后，SDA引脚将被拉高。当SDA引脚采样为高电平且SCL也是高电平时，SSPxSTAT寄存器的P位置1。另一个TBRG之后，PEN位被清零，同时SSPxIF位置1（图24-31）。

24.6.9.1 WCOL状态标志

如果在停止序列进行过程中用户写SSPxBUF，则WCOL位置1，同时缓冲区内容不变（未发生写操作）。

图24-30： 应答序列波形图

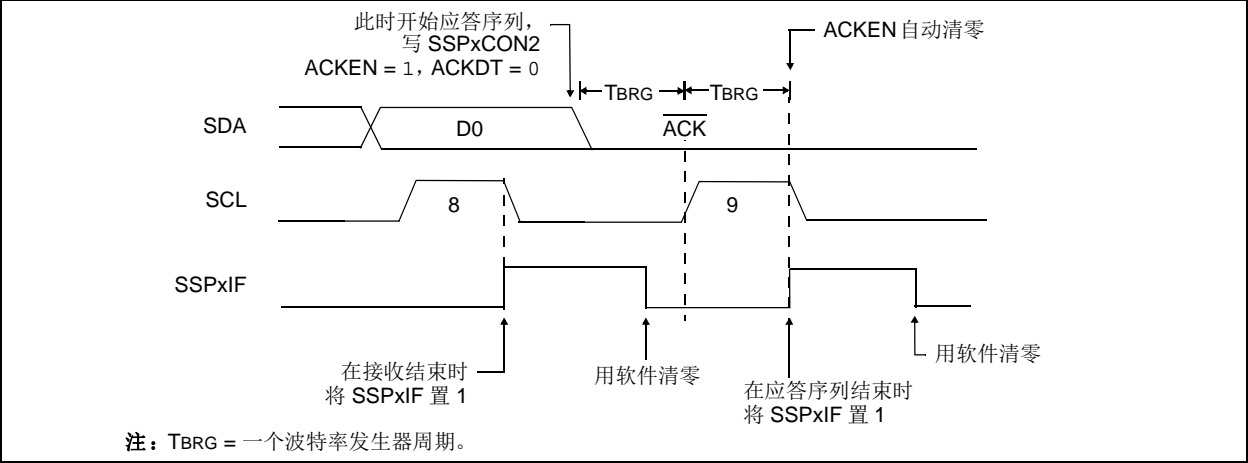
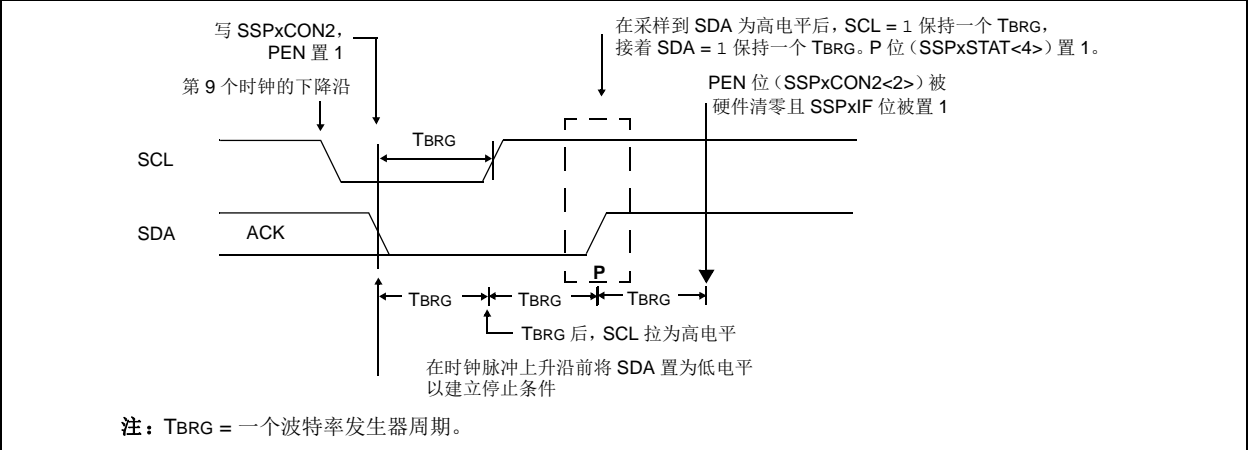


图24-31： 停止条件接收或发送模式



## 24.6.10 休眠模式下的操作

在休眠模式下，I<sup>2</sup>C 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，会将处理器从休眠状态唤醒。

## 24.6.11 复位的影响

复位会禁止 MSSP 模块并终止当前的数据传输。

## 24.6.12 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 I<sup>2</sup>C 总线的控制权；或者，总线处于空闲状态，S 位和 P 位都清零。当总线忙且允许 SSP 中断时，一旦发生停止条件便产生 SSP 中断。

在多主器件操作中，必须监视 SDA 线来进行仲裁，以查看信号电平是否为期望的输出电平。此操作由硬件实现，其结果保存在 BCL1IF 位中。

可能导致仲裁失败的情况是：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

## 24.6.13 多主器件通信、总线冲突和总线仲裁

多主器件模式是通过总线仲裁来支持的。当主器件将地址/数据位输出到 SDA 引脚时，如果一个主器件在 SDA 引脚上输出 1（将 SDA 引脚悬空为高电平），而另一个主器件输出 0，就会发生总线仲裁。当 SCL 引脚悬空为高电平时，数据应是稳定的。如果 SDA 引脚上期望的数据是 1，而实际采样到的数据是 0，则发生了总线冲突。主器件会将总线冲突中断标志 BCL1IF 置 1，并将 I<sup>2</sup>C 端口复位到空闲状态（图 24-32）。

如果在发送过程中发生总线冲突，则发送操作停止，BF 标志被清零，SDA 和 SCL 线被置为无效，并且可写入 SSPxBUF。当执行总线冲突中断服务程序时，如果 I<sup>2</sup>C 总线空闲，用户可通过发出启动条件恢复通信。

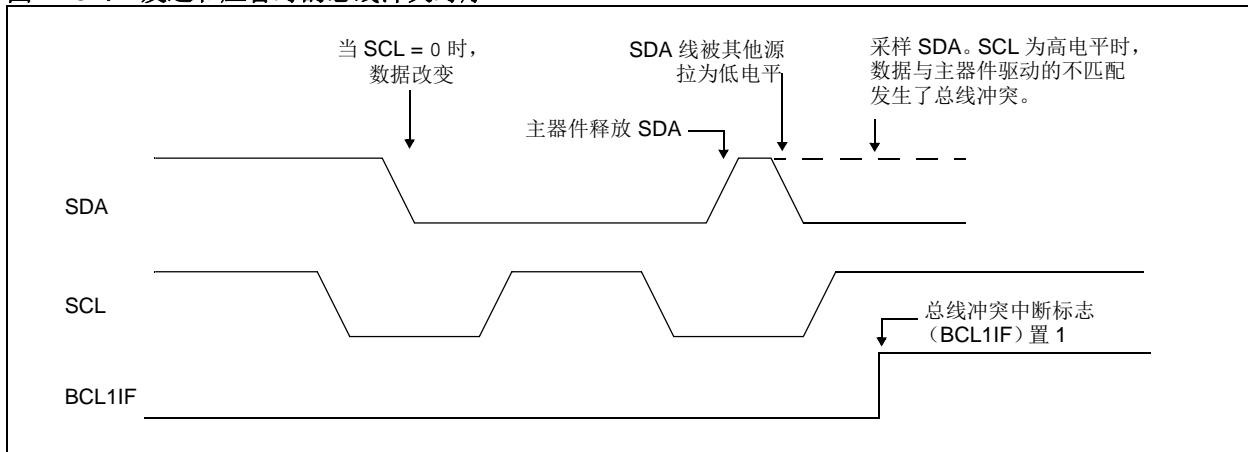
如果在启动、重复启动、停止或应答条件过程中发生总线冲突，则条件被中止，SDA 和 SCL 线被置为无效，SSPxCON2 寄存器中的相应控制位清零。当执行总线冲突中断服务程序时，如果 I<sup>2</sup>C 总线空闲，用户可通过发出启动条件恢复通信。

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，可以取得 I<sup>2</sup>C 总线的控制权；或者，总线处于空闲状态，S 位和 P 位都清零。

图 24-32： 发送和应答时的总线冲突时序





24.6.13.1 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

- a) 在启动条件开始时，SDA 或 SCL 采样为低电平（图24-33）。
- b) SDA置为低电平之前，SCL采样为低电平（图24-34）。

在启动条件期间，SDA和SCL引脚都会被监视。

如果SDA引脚或SCL引脚已经是低电平，则发生以下所有事件：

- 启动条件中止，
- BCL1IF标志置1，并且
- MSSP模块复位为空闲状态（图24-33）。

启动条件从SDA和SCL引脚被置为无效时开始。当SDA引脚采样为高电平时，波特率发生器装入值并递减计数。如果在SDA为高电平时，SCL引脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图驱动一个数据1。

如果SDA引脚在该计数周期内采样为低电平，则BRG复位，且SDA线提前置为高电平（图24-35）。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24-33： 启动条件期间的总线冲突（仅用于SDA）

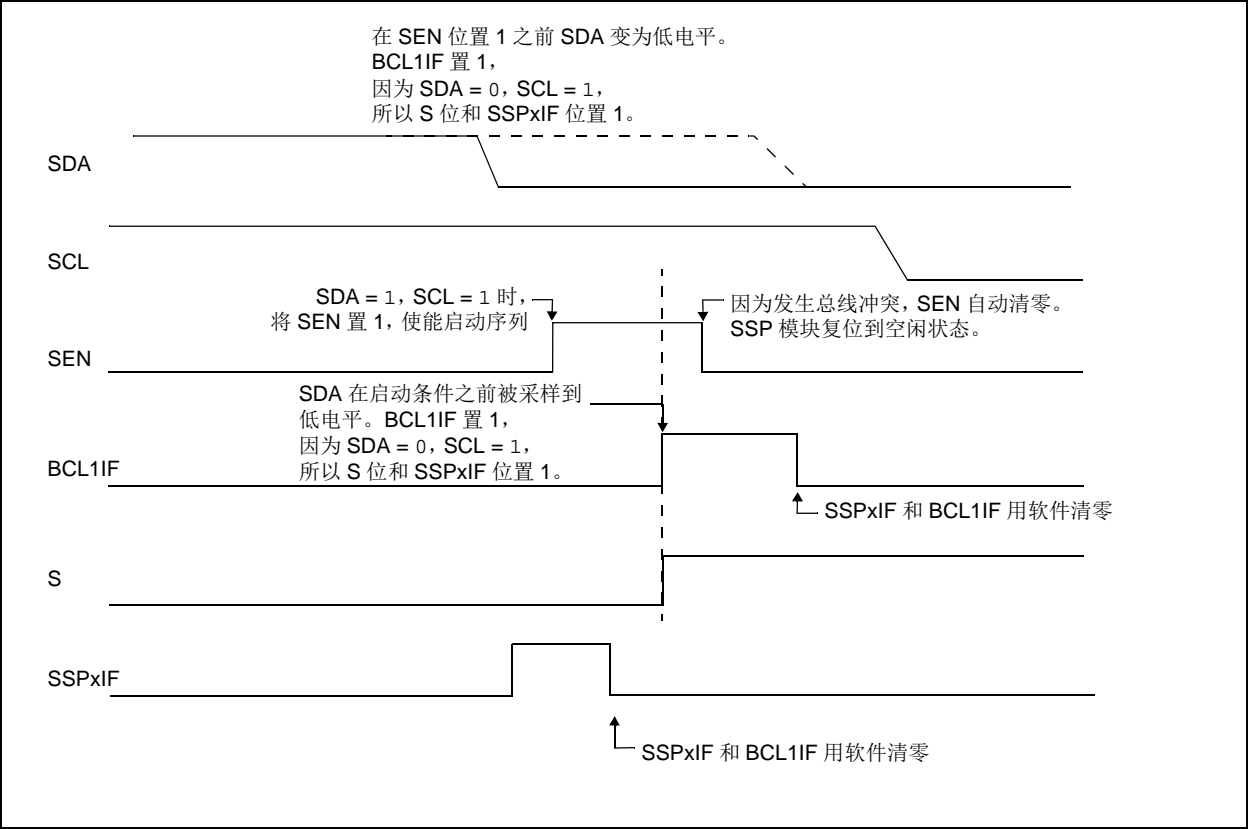


图24-34： 启动条件期间的总线冲突（SCL = 0）

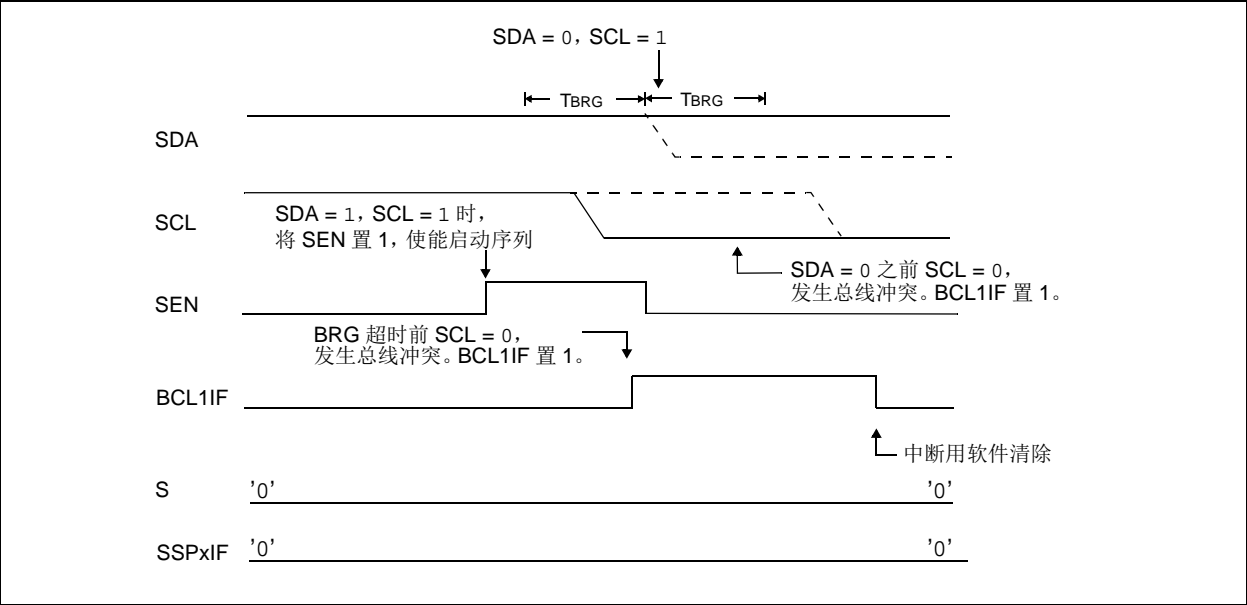
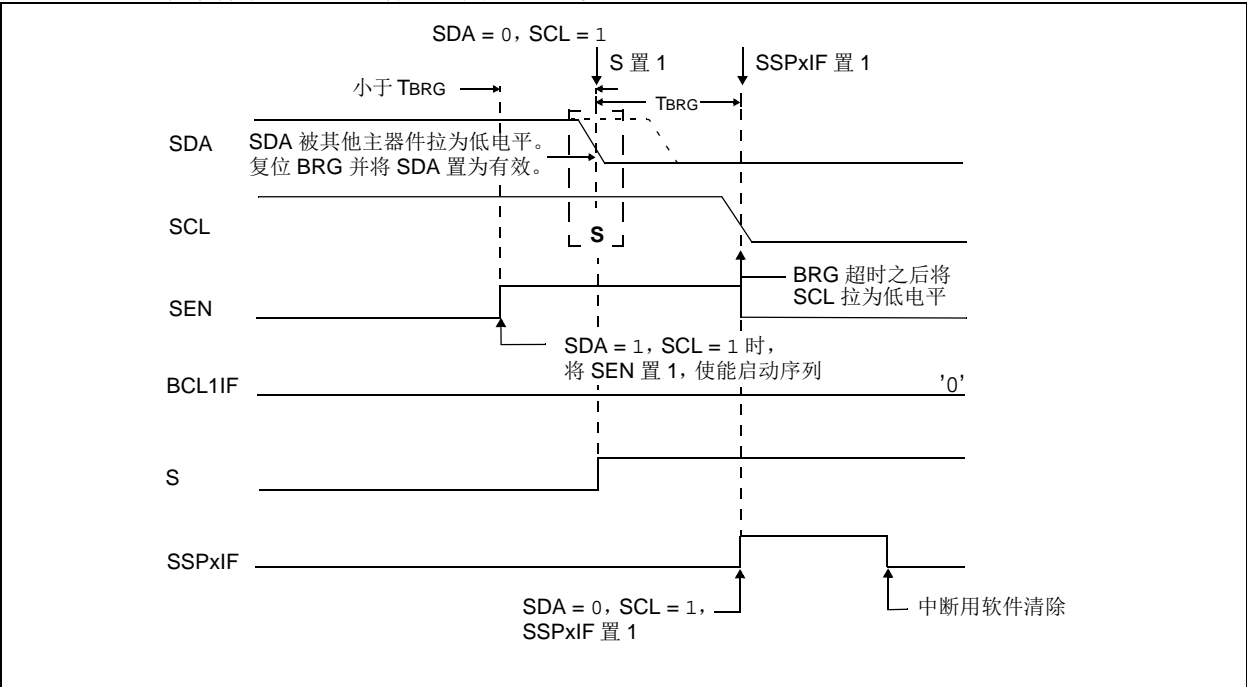


图24-35： 启动条件期间由SDA仲裁引起的BRG复位



24.6.13.2 重复启动条件期间的总线冲突

在重复启动条件期间，如果发生以下情况，则会发生总线冲突：

- a) 在SCL由低电平变为高电平期间，在SDA上采样到低电平（情形1）。
- b) 在SDA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24-36）。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24-37）。

如果在BRG超时结束时SCL和SDA都仍然是高电平，则SDA引脚被驱动为低电平，BRG被重载并开始计数。在计数结束时，不管SCL引脚的状态如何，SCL引脚都被驱动为低电平，重复启动条件结束。

图24-36： 重复启动条件期间的总线冲突（情形1）

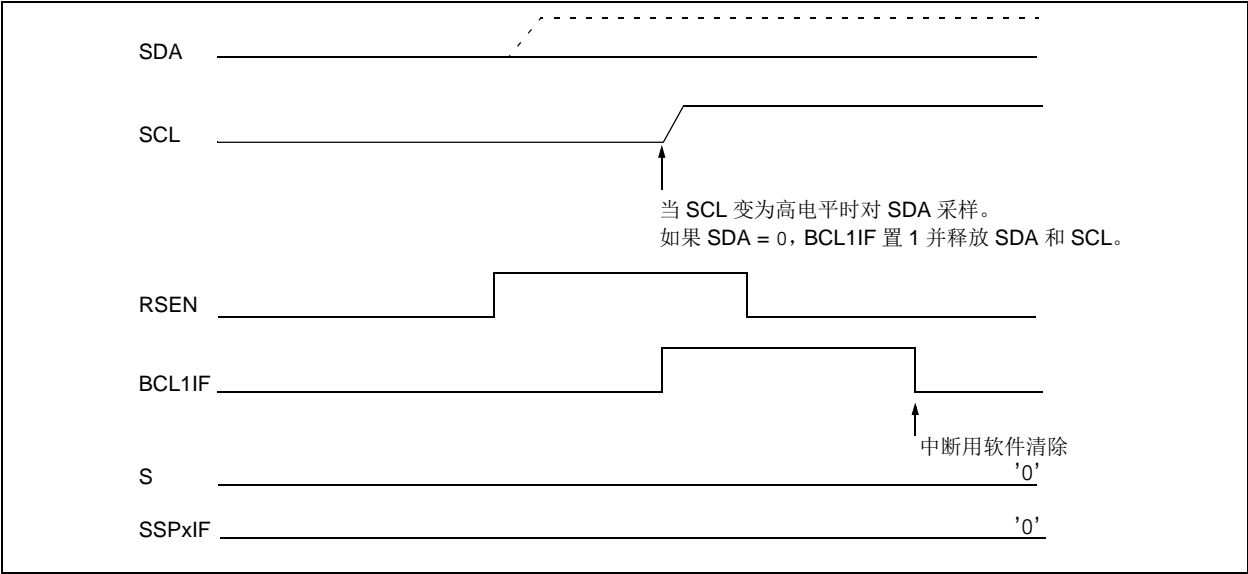
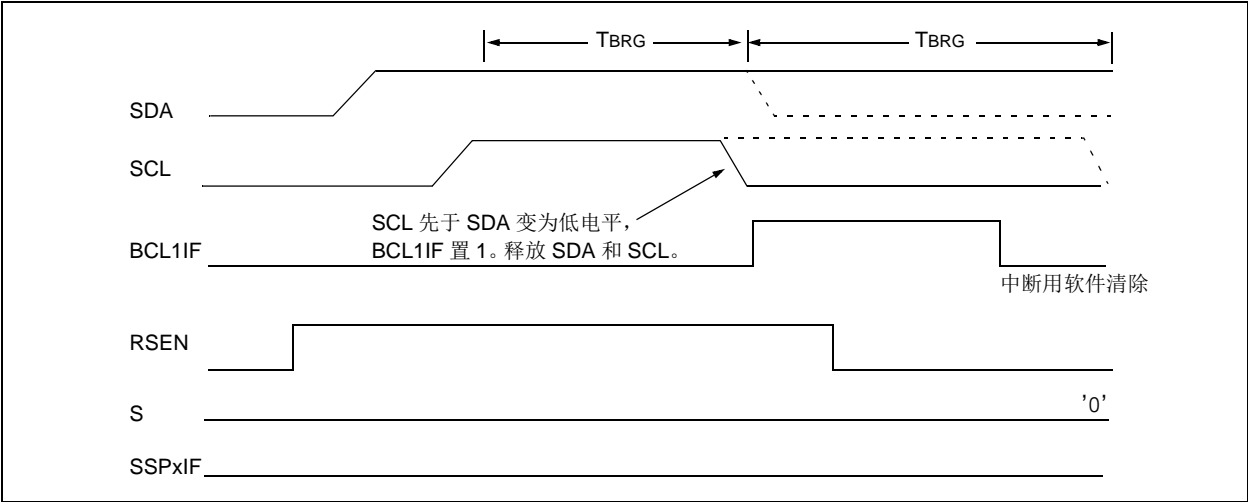


图24-37： 重复启动条件期间的总线冲突（情形2）



24.6.13.3 停止条件期间的总线冲突

以下事件会导致停止条件期间发生总线冲突：

- a) SDA 引脚已被置为无效并允许悬空为高电平之后，SDA 在 BRG 超时后被采样到低电平（情形 1）。
- b) SCL 引脚被置为无效之后，SCL 在 SDA 变成高电平之前被采样到低电平（情形 2）。

停止条件从 SDA 被置为低电平开始。当 SDA 采样为低电平时，允许 SCL 引脚悬空。当引脚被采样到高电平（时钟仲裁）时，波特率发生器装入 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 24-38）。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 24-39）。

图 24-38： 停止条件期间的总线冲突（情形 1）

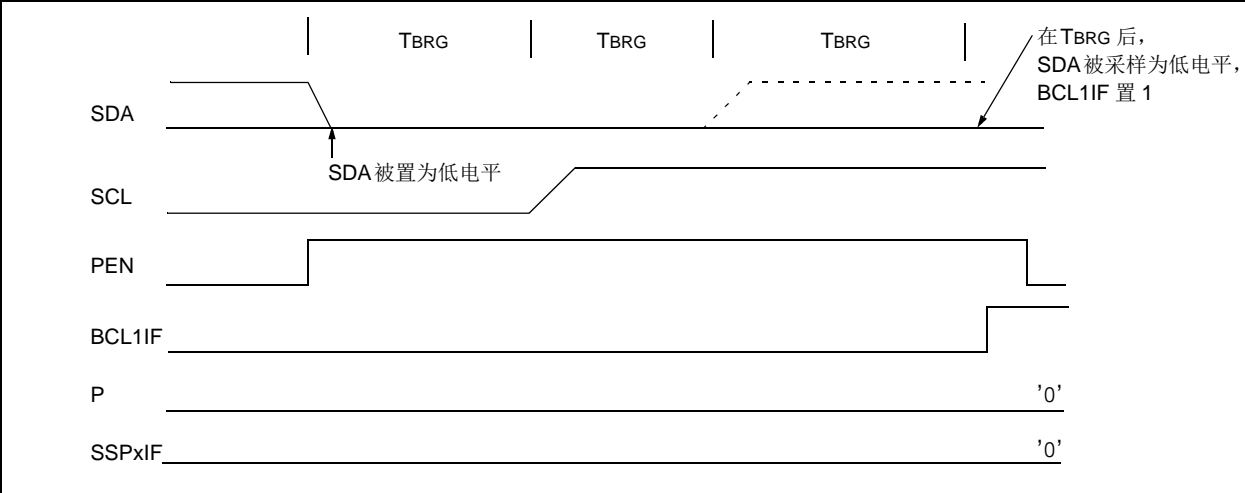
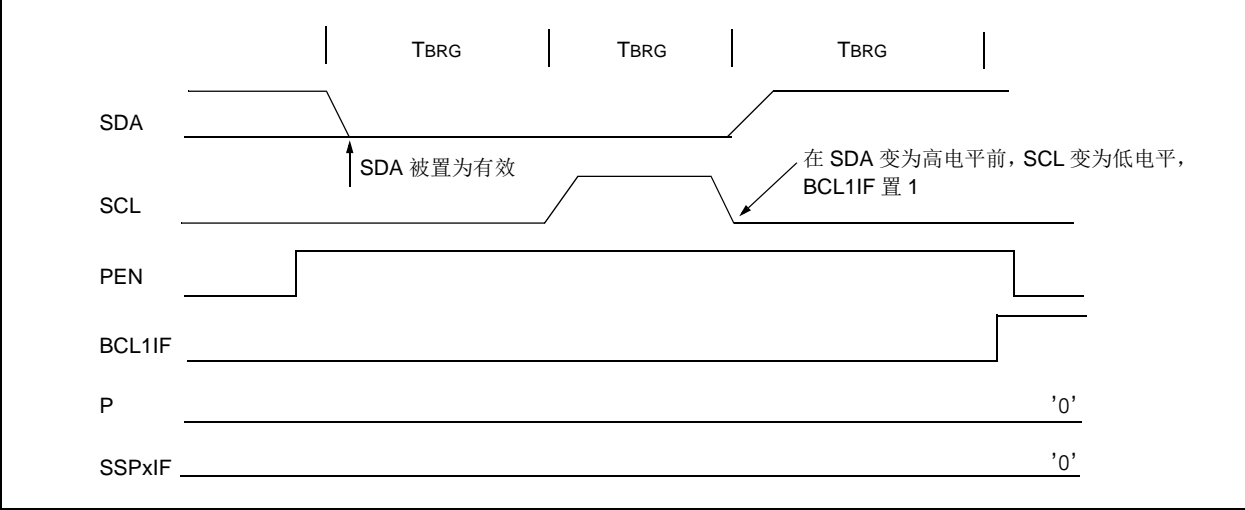


图 24-39： 停止条件期间的总线冲突（情形 2）



**表24-3: 与I<sup>2</sup>C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	复位值 所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	160
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	107
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF	112
RxyPPS	—	—	—	RxyPPS<4:0>					180
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					182 和 180
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					182 和 180
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					182 和 180
SSP1ADD	ADD<7:0>								319
SSP1BUF	同步串行端口接收缓冲/发送寄存器								271*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				316
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	317
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	318
SSP1MSK	MSK<7:0>								319
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	315
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	159
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173

**图注:** — = 未实现位, 读为0。I<sup>2</sup>C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)F1619。

**2:** 未实现, 读为1。

24.7 波特率发生器

MSSP模块具有一个波特率发生器，可用于在I<sup>2</sup>C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 24-6）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24-40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

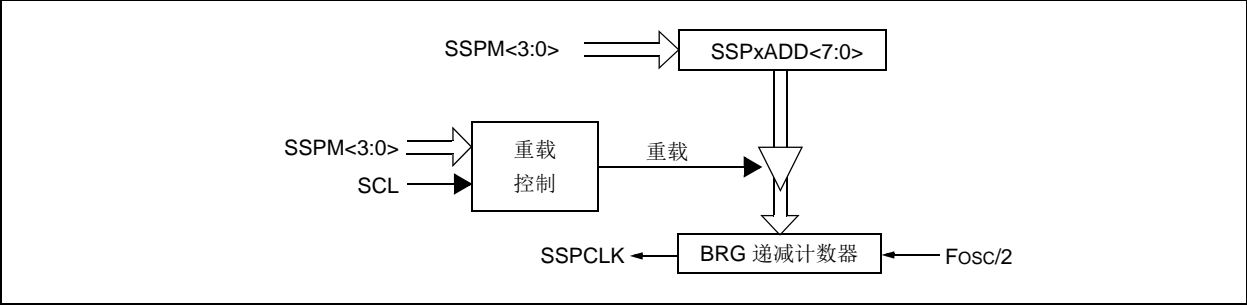
发生两次。指定重载信号何时置为有效的逻辑依赖于 MSSP 当前的工作模式。

表24-4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24-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

图24-40： 波特率发生器框图



注： 在用作 I<sup>2</sup>C 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24-4： 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35-4中的I/O端口电气规范。

## 24.8 寄存器定义：MSSP控制

寄存器 24-1: 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	<b>SMP:</b> SPI 数据输入采样位 <b>SPI 主模式:</b> 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 <b>SPI 从模式:</b> 当 SPI 工作在从模式时, 必须将 SMP 清零 在 I <sup>2</sup> 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	<b>CKE:</b> SPI 时钟边沿选择位 (仅限 SPI 模式) <b>在 SPI 主模式或从模式下:</b> 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 <b>仅在 I<sup>2</sup>C 模式下:</b> 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	<b>D/A:</b> 数据/地址位 (仅限 I <sup>2</sup> 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	<b>P:</b> 停止位 (仅限 I <sup>2</sup> 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	<b>S:</b> 启动位 (仅限 I <sup>2</sup> 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	<b>R/W:</b> 读/写位信息 (仅限 I <sup>2</sup> 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。 <b>在 I<sup>2</sup>C 从模式下:</b> 1 = 读: 0 = 写: <b>在 I<sup>2</sup>C 主模式下:</b> 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	<b>UA:</b> 更新地址位 (仅限 10 位 I <sup>2</sup> 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	<b>BF:</b> 缓冲区满状态位 <b>接收 (SPI 和 I<sup>2</sup>C 模式):</b> 1 = 接收完成, SSP1BUF 已满 0 = 接收未完成, SSP1BUF 为空 <b>发送 (仅限 I<sup>2</sup>C 模式):</b> 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)F1615/9

寄存器 24-2:        **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 <sup>(1)</sup>	SSPEN	CKP	SSPM<3:0>			
bit 7				bit 0			

<b>图注:</b>							
R = 可读位		W = 可写位		U = 未实现位, 读为0			
u = 不变		x = 未知		-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	<b>WCOL:</b> 写冲突检测位 <u>主模式:</u> 1 = 当 I <sup>2</sup> C 不满足启动发送数据的条件时, 试图向 SSP1BUF 寄存器写入数据 0 = 未发生冲突 <u>从模式:</u> 1 = 正在发送前一个字时, 又有数据写入 SSP1BUF 寄存器 (必须用软件清零) 0 = 未发生冲突
bit 6	<b>SSPOV:</b> 接收上溢指示位 <sup>(1)</sup> <u>在 SPI 模式下:</u> 1 = SSP1BUF 寄存器中仍保存前一数据时, 又接收到一个新的字节。如果发生上溢, SSPSR 中的数据会丢失。上溢只会发生在从模式下发生。在从模式下, 即使只是发送数据, 用户也必须读 SSP1BUF, 以避免将上溢位置 1。在主模式下, 上溢位不会被置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSP1BUF 寄存器启动的 (必须用软件清零)。 0 = 无上溢 <u>在 I<sup>2</sup>C 模式下:</u> 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 = 无上溢
bit 5	<b>SSPEN:</b> 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 <u>在 SPI 模式下:</u> 1 = 使能串口并将 SCK、SDO、SDI 和 $\overline{SS}$ 配置为串口引脚源 <sup>(2)</sup> 0 = 禁止串口并将上述引脚配置为 I/O 端口引脚 <u>在 I<sup>2</sup>C 模式下:</u> 1 = 使能串口并将 SDA 和 SCL 引脚配置为串口引脚源 <sup>(3)</sup> 0 = 禁止串口并将上述引脚配置为 I/O 端口引脚
bit 4	<b>CKP:</b> 时钟极性选择位 <u>在 SPI 模式下:</u> 1 = 时钟的空闲状态为高电平 0 = 时钟的空闲状态为低电平 <u>在 I<sup>2</sup>C 从模式下:</u> <b>SCL 释放控制</b> 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。)。 <u>在 I<sup>2</sup>C 主模式下:</u> 在此模式下未使用
bit 3-0	<b>SSPM&lt;3:0&gt;:</b> 同步串口模式选择位 1111 = I <sup>2</sup> C 从模式, 10 位地址, 并允许启动位和停止位中断 1110 = I <sup>2</sup> C 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 = I <sup>2</sup> C 固件控制的主模式 (从器件空闲) 1010 = SPI 主模式, 时钟 = Fosc/(4 * (SSP1ADD+1)) <sup>(5)</sup> 1001 = 保留 1000 = I <sup>2</sup> C 主模式, 时钟 = Fosc / (4 * (SSP1ADD+1)) <sup>(4)</sup> 0111 = I <sup>2</sup> C 从模式, 10 位地址 0110 = I <sup>2</sup> C 从模式, 7 位地址 0101 = SPI 从模式, 时钟 = SCK 引脚, 禁止 $\overline{SS}$ 引脚控制, $\overline{SS}$ 可用作 I/O 引脚 0100 = SPI 从模式, 时钟 = SCK 引脚, 使能 $\overline{SS}$ 引脚控制 0011 = SPI 主模式, 时钟 = T2_match/2 0010 = SPI 主模式, 时钟 = 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, 上溢位不会被置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSP1BUF 寄存器启动的。
- 2: 当使能时, 必须将这些引脚正确地配置为输入或输出。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 I<sup>2</sup>C 模式, 不支持 SSPADD 值 0、1 或 2。
- 5: 不支持 SSP1ADD 值 0。改为使用 SSPM = 0000。



## 寄存器 24-3: SSP1CON2: SSP 控制寄存器 2<sup>(1)</sup>

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 I<sup>2</sup>C 从模式)  
1 = 当 SSPSR 接收到广播呼叫地址 (0x00 或 00h) 时允许中断  
0 = 禁止广播呼叫地址
- bit 6      **ACKSTAT:** 应答状态位 (仅限 I<sup>2</sup>C 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 I<sup>2</sup>C 模式)  
在接收模式下:  
当用户在接收结束时发出一个应答序列时要发送的值  
1 = 无应答  
0 = 应答
- bit 4      **ACKEN:** 应答序列使能位 (仅限 I<sup>2</sup>C 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 I<sup>2</sup>C 主模式)  
1 = 使能 I<sup>2</sup>C 接收模式  
0 = 接收空闲
- bit 2      **PEN:** 停止条件使能位 (仅限 I<sup>2</sup>C 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 I<sup>2</sup>C 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 (已使能时钟延长) 使能时钟延长  
0 = 禁止时钟延长

**注 1:** 对于 ACKEN、RCEN、PEN、RSEN 和 SEN 位: 如果 I<sup>2</sup>C 模块不处于空闲模式, 该位可能不会被置 1 (不支持并行操作), 并且可能不会写入 SSP1BUF (或禁止写入 SSP1BUF)。

# PIC16(L)F1615/9

寄存器 24-4: SSP1CON3: SSP 控制寄存器 3

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACKTIM <sup>(3)</sup>	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 I<sup>2</sup>C 模式) <sup>(3)</sup>  
1 = 指示 I<sup>2</sup>C 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 I<sup>2</sup>C 模式)  
1 = 检测到停止条件时允许中断  
0 = 禁止检测到停止条件时的中断 <sup>(2)</sup>
- bit 5 **SCIE:** 启动条件中断允许位 (仅限 I<sup>2</sup>C 模式)  
1 = 检测到启动或重复启动条件时允许中断  
0 = 禁止检测到启动条件时的中断 <sup>(2)</sup>
- bit 4 **BOEN:** 缓冲区改写使能位  
在 SPI 从模式下: <sup>(1)</sup>  
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 I<sup>2</sup>C 主模式和 SPI 主模式下:  
该位被忽略。  
在 I<sup>2</sup>C 从模式下:  
1 = 仅当 BF 位 = 0 时, 在接收到地址/数据字节时, 更新 SSP1BUF 并产生  $\overline{\text{ACK}}$  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 I<sup>2</sup>C 模式)  
1 = 在 SCL 的下降沿之后, 在 SDA 上最少有 300 ns 的保持时间  
0 = 在 SCL 的下降沿之后, 在 SDA 上最少有 100 ns 的保持时间
- bit 2 **SBCDE:** 从模式总线冲突检测使能位 (仅限 I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C 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 24-5: 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I<sup>2</sup>C模式下地址是否匹配  
0 = 接收到的地址bit n不用于检测I<sup>2</sup>C模式下地址是否匹配
- bit 0

**MSK<0>:** 用于I<sup>2</sup>C从模式, 10位地址的掩码位  
I<sup>2</sup>C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I<sup>2</sup>C模式下地址是否匹配  
0 = 接收到的地址bit 0不用于检测I<sup>2</sup>C模式下地址是否匹配  
I<sup>2</sup>C从模式, 7位地址, 该位被忽略

寄存器 24-6: SSP1ADD: MSSP地址和波特率寄存器 (I<sup>2</sup>C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——高位地址字节:

- bit 7-3

**未使用:** 不使用高位地址字节。该寄存器的位状态为“无关”。主器件发送的位模式由I<sup>2</sup>C规范确定, 必须等于11110。但是, 这些位通过硬件进行比较, 不会受该寄存器中的值影响。
- bit 2-1

**ADD<2:1>:** 10位地址的高2位
- bit 0

**未使用:** 在此模式下未使用。位状态为“无关”。

10位从模式——低位地址字节:

- bit 7-0

**ADD<7:0>:** 10位地址的低8位

7位从模式:

- bit 7-1

**ADD<7:1>:** 7位地址
- bit 0

**未使用:** 在此模式下未使用。位状态为“无关”。

## 25.0 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 (Enhanced Universal Synchronous Asynchronous Receiver Transmitter, EUSART) 模块是一种串行 I/O 通信外设。它包含用来完成与器件程序执行无关的输入或输出串行数据传输所需的所有时钟发生器、移位寄存器和数据缓冲区等。EUSART 也可称为串行通信接口 (Serial 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 25-1 和图 25-2 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 (Data Signal Modulator, DSM)

图 25-1: EUSART 发送框图

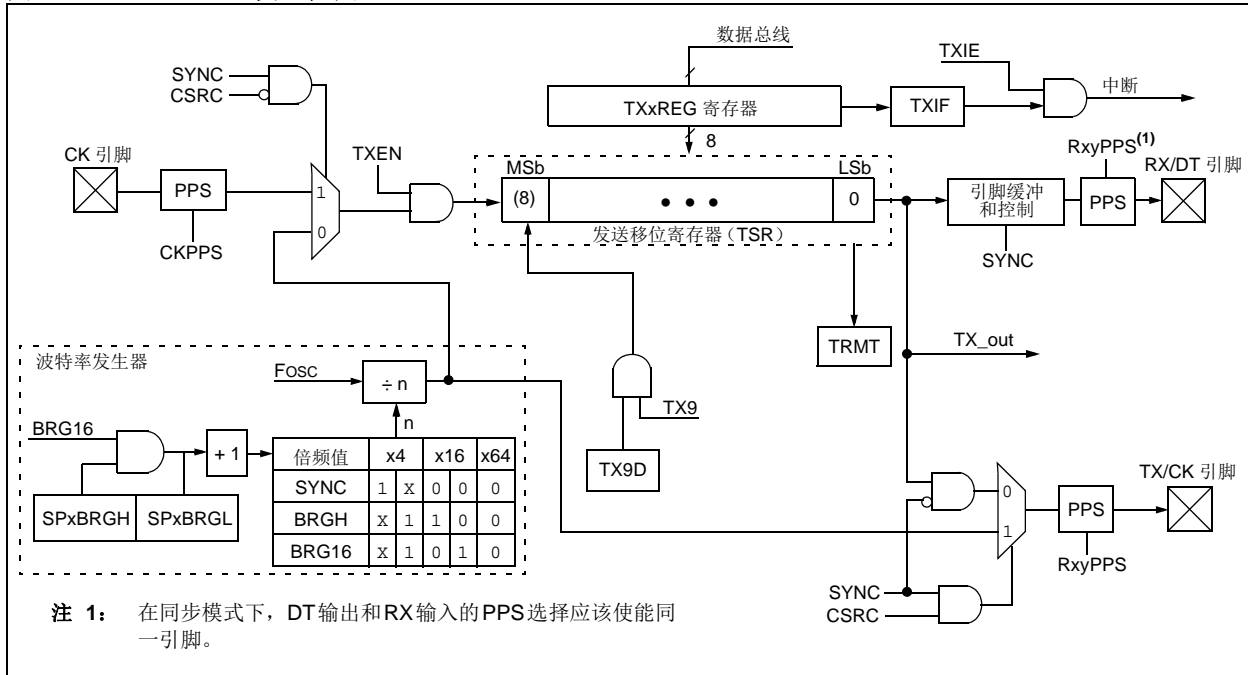
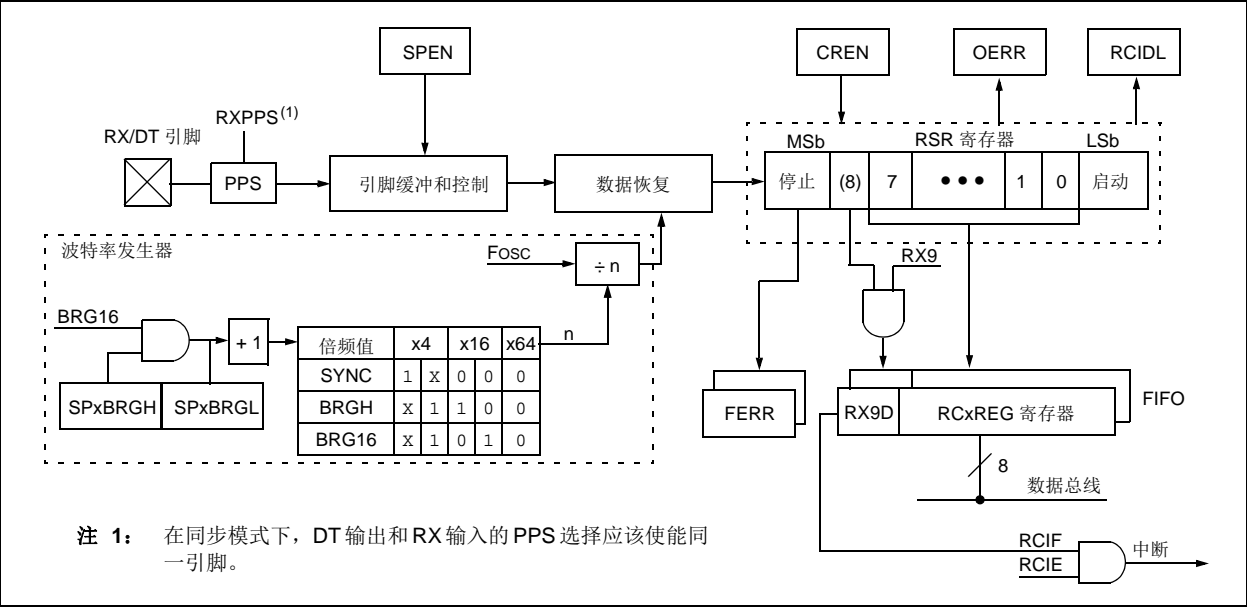


图25-2: EUSART接收框图



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 25-1](#)、[寄存器 25-2](#) 和 [寄存器 25-3](#)。

RX 和 CK 输入引脚分别使用 RXPPS 和 CKPPS 寄存器进行选择。TX、CK 和 DT 输出引脚使用每个引脚的 RxyPPS 寄存器进行选择。由于在同步模式下 RX 输入与 DT 输出配合，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。

## 25.1 EUSART异步模式

EUSART采用标准不归零（non-return-to-zero, NRZ）格式发送和接收数据。NRZ实现为两种电平：VOH标记状态（Mark state）代表“1”数据位，而VOL空格状态（Space state）代表“0”数据位。NRZ指的是连续发送具有相同值的数据位时，输出电平始终保持不变，而不会在发送完每个位之后回到中间电平。NRZ发送端口在标记状态空闲。每个字符发送包含1个启动位及随后的8个或9个数据位，并始终由1个或多个停止位终止。启动位始终是一个空格，停止位始终是标记。最常用的数据格式为8位。每个发送位保持时间为1/(波特率)。使用片上专用8位/16位波特率发生器从系统振荡器产生标准波特率频率。波特率配置示例请参见表25-5。

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。

### 25.1.1 EUSART异步发送器

图25-1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。

#### 25.1.1.1 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。

#### 25.1.1.2 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、数据位和停止位的序列发送立即开始。

#### 25.1.1.3 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25.5.1.2节“时钟极性”。

#### 25.1.1.4 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后应清零TXIE中断允许位。



# PIC16(L)F1615/9

表25-1: 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	160
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	332
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	331
RxyPPS	—	—	—	RxyPPS<4:0>					180
SP1BRGL	BRG<7:0>								333*
SP1BRGH	BRG<15:8>								333*
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	157
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
TX1REG	EUSART发送数据寄存器								322*
TX1STA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	330

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)F1619。

2: 未实现, 读为1。



25.1.2 EUSART 异步接收器

异步模式通常用于 RS-232 系统中。图 25-2 给出了接收器框图。数据在 RX/DT 引脚上接收并驱动数据恢复模块。数据恢复模块实际上是一个高速移位器，工作频率为 16 倍波特率，而串行接收移位寄存器（Receive Shift Register, RSR）工作频率为比特率。所有 8 位或 9 位字符移入后被立即传送到双字符的先进先出（First-In-First-Out, FIFO）存储区中。FIFO 缓冲区允许先接收两个完整字符和第三个字符的开始部分后，再启动软件服务 EUSART 接收器。FIFO 和 RSR 寄存器不能直接用软件访问。通过 RCxREG 寄存器访问接收数据。

25.1.2.1 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。

25.1.2.2 接收数据

接收器的数据恢复电路在第一位的下降沿启动字符接收。第一位也称启动（Start）位，始终为零。数据恢复电路计数传输半个位的时间至启动位的中点并验证该位是否仍为零。如果该位非零，则数据恢复电路中止字符接收，不产生错误，并恢复寻找启动位的下降沿。如果启动位被验证为零，则数据恢复电路计数一整个位时间至下个位的中点。该位被一个择多检测电路采样，其结果（0 或 1）被移入 RSR。重复此过程直到所有数据位均被采样并移入 RSR。最后一个位时间被测量且其电平被采样。它是停止位，总是为 1。如果数据恢复电路在停止位处采样到 0，则置 1 此字符的帧错误标志位，否则清零此字符的帧错误标志位。关于帧错误的更多信息，请参见第 25.1.2.4 节“接收帧错误”。

所有数据位和停止位被接收后，RSR 中的字符就被立即传送到 EUSART 接收 FIFO，且 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 25.1.2.5 节“接收溢出错误”。

25.1.2.3 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。

## 25.1.2.4 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，只用于接收FIFO中顶部的未读字符。帧错误（FERR = 1）并不会禁止接收更多字符。此时不必将FERR位清零。从FIFO缓冲区读出下一个字符将使FIFO进入下一个字符和下一个相应的帧错误。

将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。

## 25.1.2.5 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。

## 25.1.2.6 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。

## 25.1.2.7 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，并将RCIF中断标志位置1。所有其他字符均被忽略。

接收到地址字符后，用户软件可判断地址是否与自身匹配。地址匹配时，发生下一个停止位前，用户软件必须通过清零ADDEN位禁止地址检测。当用户软件根据所使用的报文协议检测到报文的末尾时，软件将ADDEN位置1，将接收器重新置于地址检测模式。

25.1.2.8 异步接收设置

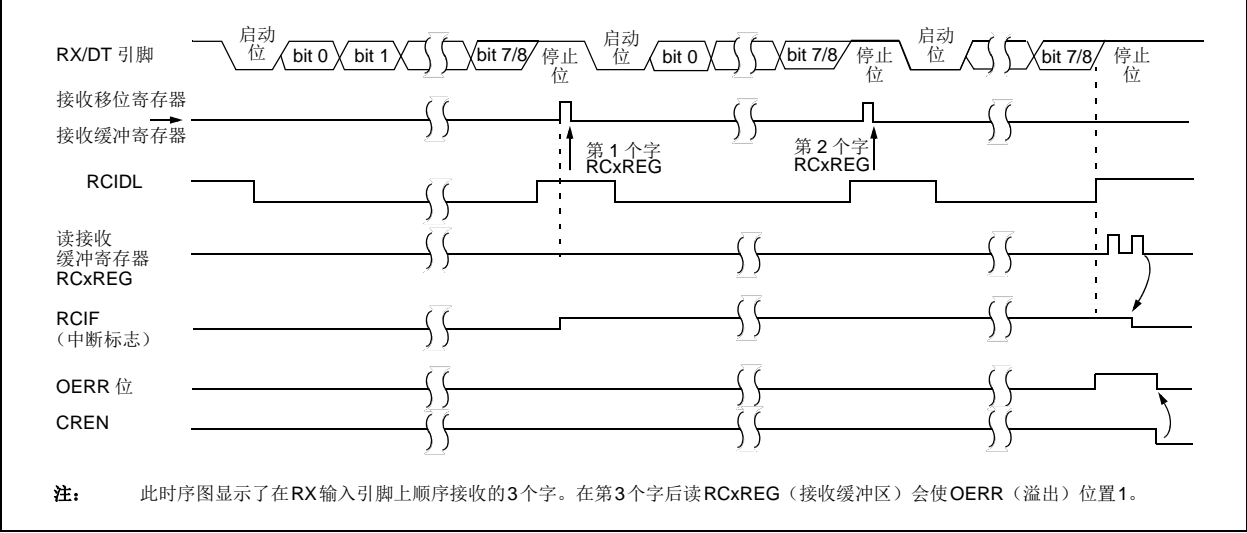
1. 初始化 SPxBRGH 和 SPxBRGL 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率（见第25.4节“EUSART波特率发生器（BRG）”）。
2. 清零RX引脚的ANSEL位（如适用）。
3. 将 SPEN 位置 1 使能串口。SYNC 位必须清零才能进行异步操作。
4. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 如果需要接收 9 位数据，将 RX9 位置 1。
6. 将 CREN 位置 1 使能接收。
7. 当字符从 RSR 被移入接收缓冲区时，RCIF 中断标志位将被置 1。如果 RCIE 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。

25.1.2.9 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25.4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 25-5： 异步接收



# PIC16(L)F1615/9

表25-2: 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	160
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	332
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	105
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RC1REG	EUSART接收数据寄存器								325*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	331
RxyPPS	—	—	—	RxyPPS<4:0>					180
SP1BRGL	BRG<7:0>								333
SP1BRGH	BRG<15:8>								333
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	159
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
TX1STA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	330

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)F1619。

2: 未实现, 读为1。

## 25.2 异步操作的时钟精度

内部振荡器模块输出（INTOSC）在出厂时做了校准。但是，VDD或温度变化时，INTOSC频率有可能漂移，这将直接影响异步波特率。有两种方法可用来调整波特率时钟，但它们都需要某种参考时钟源。

第一种（首选）方法使用OSCTUNE寄存器调整INTOSC输出。调整OSCTUNE寄存器的值可对系统时钟源的分辨率进行微调。更多信息，请参见[第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。自动波特率检测可自动完成这种调整（见[第25.4.1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)F1615/9

## 25.3 寄存器定义：EUSART控制

寄存器 25-1: TX1STA: 发送状态和控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

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

- bit 7

**CSRC:** 时钟源选择位  
异步模式:  
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<sup>(1)</sup>  
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

**SENDB:** 发送间隔字符位  
异步模式:  
1 = 在下一次发送时发送同步间隔字符 (完成后由硬件清零)  
0 = 同步间隔字符发送完成  
同步模式:  
无关位
- 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 25-2: 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	<b>SPEN:</b> 串口使能位 1 = 使能串口 0 = 禁止串口 (保持在复位状态)
bit 6	<b>RX9:</b> 9位接收使能位 1 = 选择9位接收 0 = 选择8位接收
bit 5	<b>SREN:</b> 单字节接收使能位 <u>异步模式:</u> 无关位 <u>同步主模式:</u> 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。 <u>同步从模式:</u> 无关位
bit 4	<b>CREN:</b> 连续接收使能位 <u>异步模式:</u> 1 = 使能接收器 0 = 禁止接收器 <u>同步模式:</u> 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	<b>ADDEN:</b> 地址检测使能位 <u>9位异步模式 (RX9 = 1):</u> 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 <u>8位异步模式 (RX9 = 0):</u> 无关位
bit 2	<b>FERR:</b> 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	<b>OERR:</b> 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	<b>RX9D:</b> 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)F1615/9

## 寄存器 25-3: 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:** 自动波特率检测上溢位

#### 异步模式:

1 = 自动波特率定时器上溢

0 = 自动波特率定时器未上溢

#### 同步模式:

无关位

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 = 数据在时钟上升沿同步

0 = 数据在时钟下降沿同步

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字节将被置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



## 25.4 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TxSTA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25-3提供了确定波特率的公式。例25-1提供了确定波特率和波特率误差的计算示例。

为便于使用，各种异步模式的典型波特率和误差值已经计算出来，如表25-5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无需等待定时器上溢就可以输出新的波特率。

如果系统时钟在有效的接收操作过程中被更改，可能会导致接收错误或数据丢失。为避免此问题，应检查RCIDL位的状态，以确保在改变系统时钟前接收操作处于空闲状态。

### 例25-1: 计算波特率误差

针对工作在异步模式下、Fosc= 16 MHz，目标波特率 = 9600且采用8位BRG的器件：

$$\text{目标波特率} = \frac{F_{OSC}}{64([SPBRGH:SPBRGL] + 1)}$$

求解 SPxBRGH:SPxBRGL:

$$X = \frac{\frac{F_{OSC}}{\text{目标波特率}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{计算波特率} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{误差} = \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

# PIC16(L)F1615/9

表25-3: 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	$F_{OSC}/[64 (n+1)]$
0	0	1	8 位/异步	$F_{OSC}/[16 (n+1)]$
0	1	0	16 位/异步	
0	1	1	16 位/异步	$F_{OSC}/[4 (n+1)]$
1	0	x	8 位/同步	
1	1	x	16 位/同步	

图注: x = 无关位, n = SPxBRGH 和 SPxBRGL 寄存器对的值。

表25-4: 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	<a href="#">332</a>
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	<a href="#">331</a>
SP1BRGL	BRG<7:0>								<a href="#">333</a>
SP1BRGH	BRG<15:8>								<a href="#">333</a>
TX1STA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	<a href="#">330</a>

图注: — = 未实现位, 读为0。波特率发生器模块不使用阴影单元。

\* 提供寄存器信息的页。

表 25-5: 异步模式下的波特率

波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1202	0.16	207	1200	0.00	143
2400	2404	0.16	129	2400	0.00	119	2404	0.16	103	2400	0.00	71
9600	9470	-1.36	32	9600	0.00	29	9615	0.16	25	9600	0.00	17
10417	10417	0.00	29	10286	-1.26	27	10417	0.00	23	10165	-2.42	16
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.23k	0.16	12	19.20k	0.00	8
57.6k	—	—	—	57.60k	0.00	7	—	—	—	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.64k	-1.36	10	115.2k	0.00	9	111.1k	-3.55	8	115.2k	0.00	5

# PIC16(L)F1615/9

表25-5: 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	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	300.0	-0.01	4166	300.0	0.00	3839	300.03	0.01	3332	300.0	0.00	2303
1200	1200	-0.03	1041	1200	0.00	959	1200.5	0.04	832	1200	0.00	575
2400	2399	-0.03	520	2400	0.00	479	2398	-0.08	416	2400	0.00	287
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.818	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.636	-1.36	10	115.2k	0.00	9	111.11k	-3.55	8	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	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

表25-5: 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 = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	13332	300.0	0.00	9215
1200	1200	-0.01	4166	1200	0.00	3839	1200.1	0.01	3332	1200	0.00	2303
2400	2400	0.02	2082	2400	0.00	1919	2399.5	-0.02	1666	2400	0.00	1151
9600	9597	-0.03	520	9600	0.00	479	9592	-0.08	416	9600	0.00	287
10417	10417	0.00	479	10425	0.08	441	10417	0.00	383	10433	0.16	264
19.2k	19.23k	0.16	259	19.20k	0.00	239	19.23k	0.16	207	19.20k	0.00	143
57.6k	57.47k	-0.22	86	57.60k	0.00	79	57.97k	0.64	68	57.60k	0.00	47
115.2k	116.3k	0.94	42	115.2k	0.00	39	114.29k	-0.79	34	115.2k	0.00	23

波特率	SYNC = 0, BRGH = 1, BRG16 = 1或SYNC = 1, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)	实际 波特率	% 误差	SPBRG 值 (十进制)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

25.4.1 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”）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验序列。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 25-6 所示。在第 8 位周期的末尾将在 RX 引脚上出现第 5 个上升沿。此时，累计数据即正确的 BRG 周期总数被保存在 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，用户可通过查询 SPxBRGH 寄存器的值是否为 00h 来验证 SPxBRGL 寄存器是否上溢。

BRG 自动波特率时钟由 BRG16 和 BRGH 位决定，如表 25-6 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。

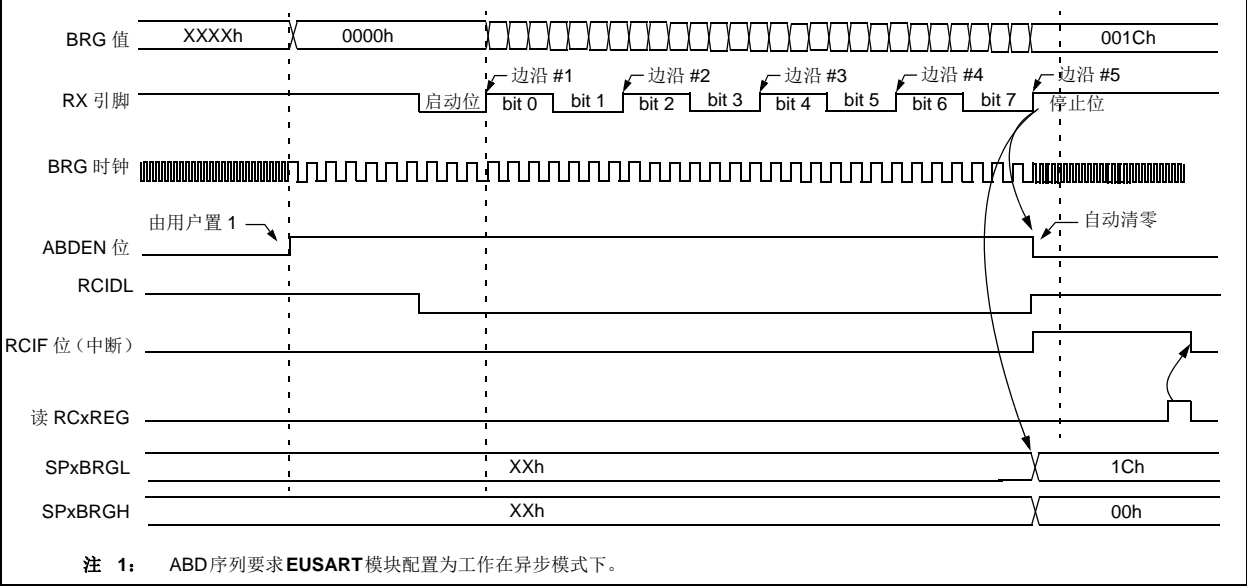
- 注 1: 如果 WUE 位和 ABDEN 位都置 1，自动波特率检测将从间隔字符之后的字节开始（见第 25.4.3 节“接收到间隔字符时自动唤醒”）。
- 2: 需要由用户来判断输入字符的波特率是否处于所选 BRG 时钟源范围内。可能无法实现某些振荡器频率和 EUSART 波特率组合。
- 3: 在自动波特率过程中，自动波特率计数器从 1 开始计数。自动波特率序列完成后，为了得到最准确的结果，应从 SPxBRGH:SPxBRGL 寄存器对的值中减去 1。

表 25-6: 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 25-6: 自动波特率校准



## 25.4.2 自动波特率上溢

在自动波特率检测过程中，如果在RX引脚上检测到第5个上升沿之前波特率计数器上溢，则BAUDxCON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。上溢条件将RCIF标志置1。计数器将继续计数，直到在RX引脚上检测到第5个上升沿为止。RCIDL位保持为假值（0），直到第5个上升沿出现时RCIDL将置1。如果在上溢发生后但在第5个上升沿出现前读取RCREG，则第5个上升沿会再次将RCIF置1。

提前终止自动波特率进程以清除上溢条件，将无法正确检测同步字符的第5个上升沿。如果同步字符的下降沿在ABDEN位清零时还未出现，则它们将被错误地检测为启动位。建议通过以下步骤清除上溢条件：

1. 读取RCREG以清除RCIF。
2. 如果RCIDL为零，则等待RCIF并重复步骤1。
3. 清零ABDOVF位。

## 25.4.3 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BAUDxCON寄存器的WUE位置1来使能。一旦置1，RX/DT上的正常接收序列就被禁止，EUSART保持在空闲状态，监视与CPU模式无关的唤醒事件。唤醒事件包含RX/DT线上电平由高至低的跳变。（这与同步间隔字符或LIN协议的唤醒信号字符的启动条件一致。）

EUSART模块产生的RCIF中断与唤醒事件同步。在正常CPU工作模式下，与Q时钟同步产生中断（图25-7）；在休眠模式下，与Q时钟异步产生中断（图25-8）。通过读RCxREG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。

## 25.4.3.1 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，唤醒字符必须为零。

唤醒被使能时，其工作状况与数据流的低电平时间无关。如果WUE位置1并接收到了有效的非零字符，则从启动位至第一个上升沿的低电平时间将被解读为唤醒事件。字符的其余位将作为碎片字符接收，后续字符有可能产生帧错误或溢出错误。

因此，发送的首字符必须为零。这必须持续10个或更长的位时间，对于LIN总线，建议持续13个位时间，而对于标准RS-232器件，可为任意个位时间。

### 振荡器起振时间

必须考虑振荡器起振时间，特别在使用起振时间较长的振荡器（即，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。然后，通过读取RCxREG寄存器并丢弃其内容，可以由软件清除中断条件。

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。



图 25-7: 正常工作时的自动唤醒位 (WUE) 时序

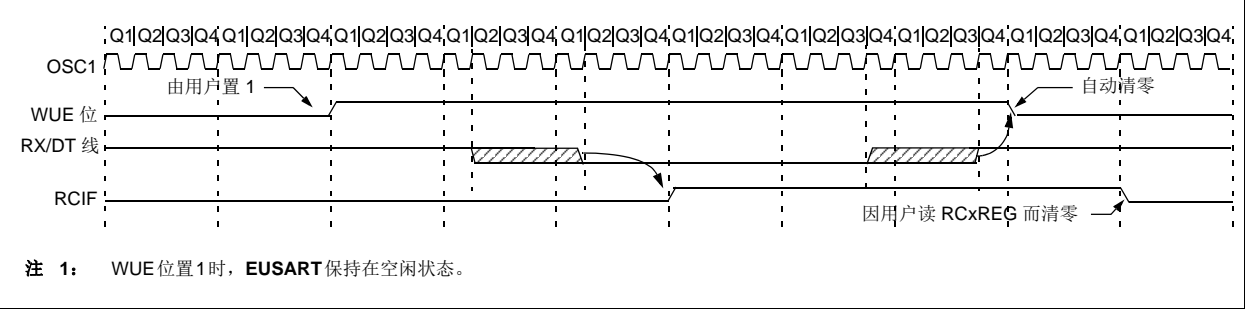
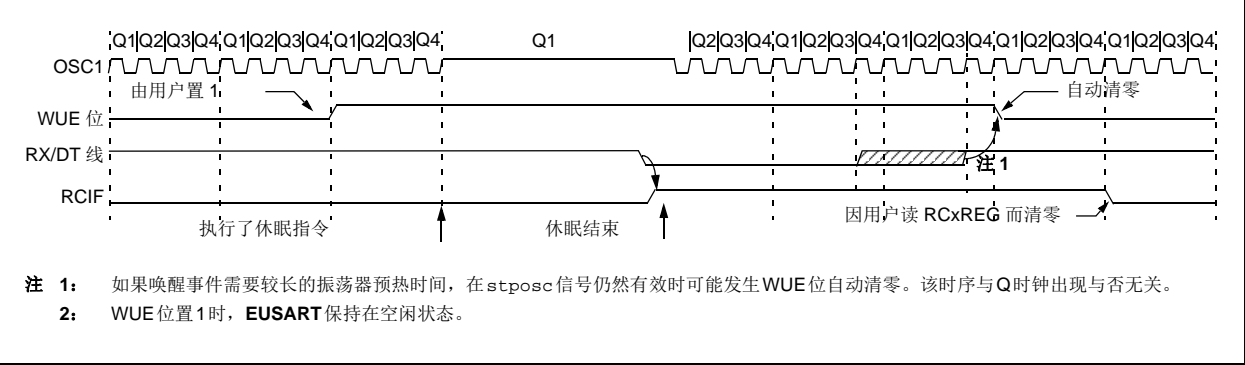


图 25-8: 休眠时的自动唤醒位 (WUE) 时序



25.4.4 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 25-9 给出了发送间隔字符的时序。

25.4.4.1 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 指出），下一个数据字节会写入 TXxREG。

25.4.5 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。

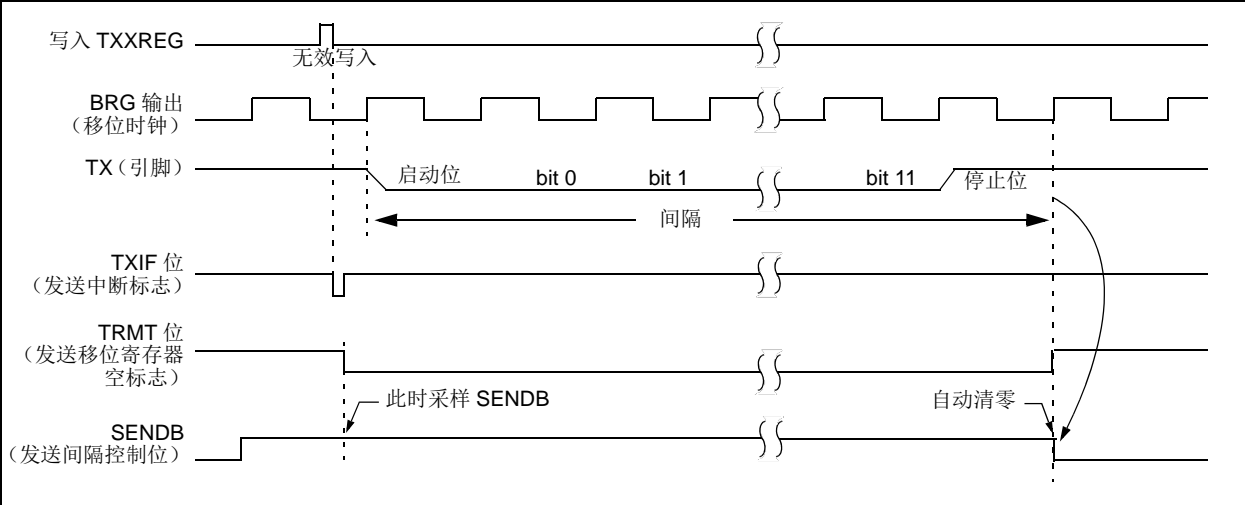
发生以下情况时，表明接收到间隔字符：

- RCIF 位被置 1
- FERR 位被置 1
- RCxREG = 00h

第二种方法采用第 25.4.3 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。

图 25-9： 发送间隔字符序列



## 25.5 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。

### 25.5.1 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。

#### 25.5.1.1 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。

#### 25.5.1.2 时钟极性

提供了与Microwire兼容的时钟极性选项。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。

#### 25.5.1.3 同步主发送

从器件的RX/DT引脚输出数据。EUSART配置为同步主发送操作时，RX/DT和TX/CK引脚的输出驱动器被自动使能。

向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寄存器中，因此用户无法使用。

#### 25.5.1.4 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25.4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25-10： 同步发送

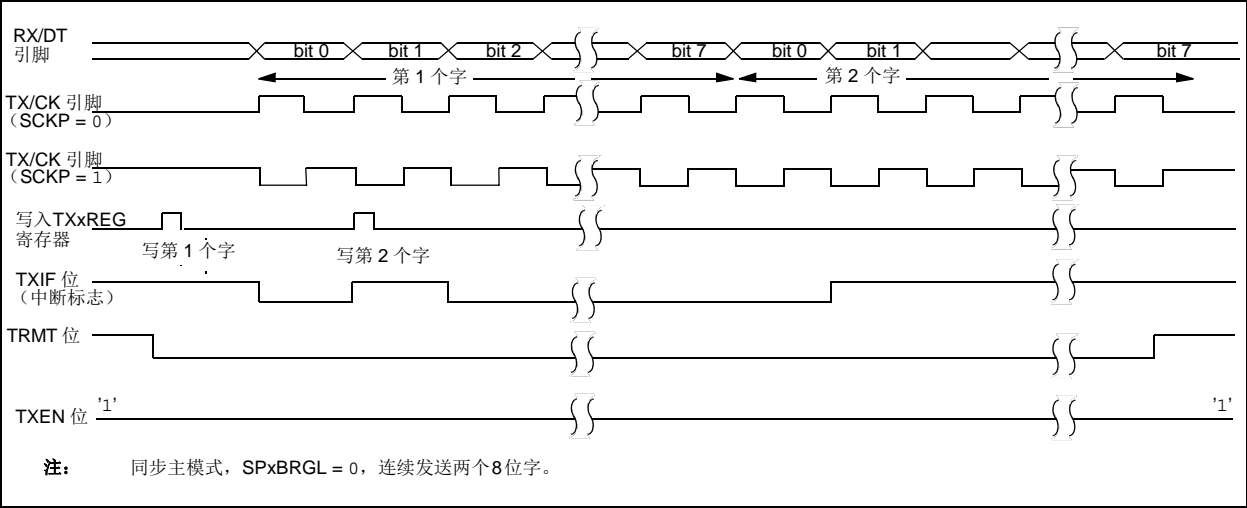
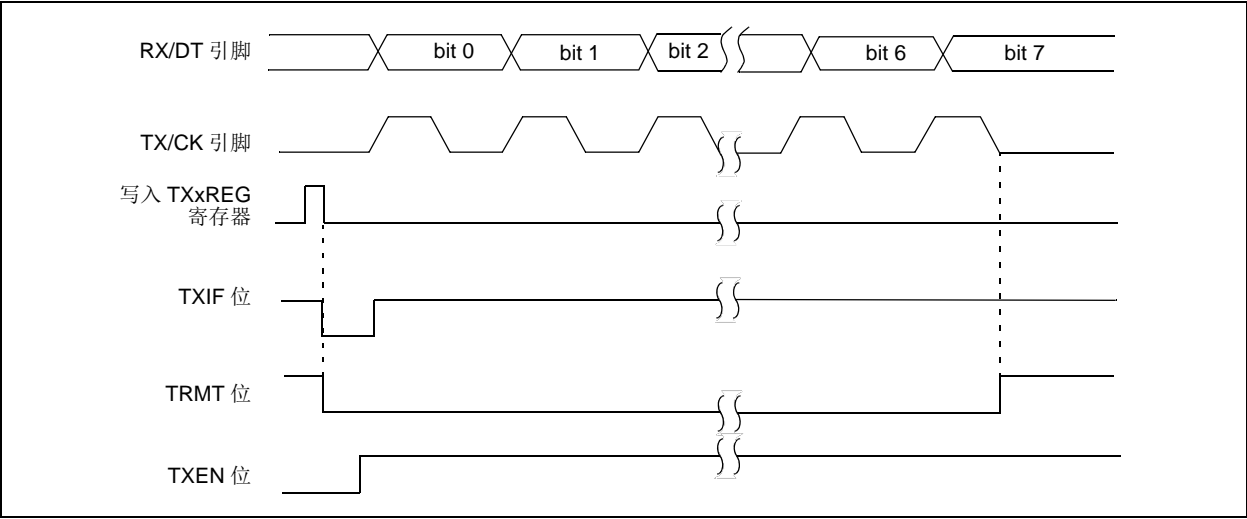


图25-11： 同步发送（由TXEN位控制）



**表25-7: 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	160
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	332
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	331
RxyPPS	—	—	—	RxyPPS<4:0>					180
SP1BRGL	BRG<7:0>								333
SP1BRGH	BRG<15:8>								333
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	159
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
TX1REG	EUSART 发送数据寄存器								322*
TX1STA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	330

**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)F1619。

**2:** 未实现, 读为1。

## 25.5.1.5 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，可通过将单字节接收使能位（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。一个字符接收完成后SREN位被自动清零。CREN置1时，将连续产生时钟直到CREN被清零。如果CREN在字符接收过程中被清零，则CK时钟立即停止，接收到的部分字符被丢弃。如果SREN和CREN同时置1，则首字符接收完成时SREN被清零，CREN优先。

要启动接收，将SREN或CREN置1。在TX/CK时钟引脚的后沿对RX/DT引脚上的数据进行采样，并移入接收移位寄存器（RSR）。在完整的字符被接收进RSR后，RCIF位置1且该字符被自动送入两个字符的接收FIFO。接收FIFO中顶部字符的低8位在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。

## 25.5.1.6 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，就产生多少个接收时钟周期。

**注：** 如果将器件配置为从器件并且TX/CK功能位于模拟引脚上，则必须清零相应的ANSEL位。

## 25.5.1.7 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（该位将EUSART复位），可以清除错误条件。

## 25.5.1.8 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。

## 25.5.1.9 同步主接收设置

1. 初始化SPxBRGH:SPxBRGL寄存器对，获得所需的波特率。按需要将BRGH和BRG16位置1或清零，获得所需的波特率。
2. 清零RX引脚的ANSEL位（如适用）。
3. 将SYNC、SPEN和CSRC位置1，以使能同步串口。
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. 通过读取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（该位将EUSART复位），可以清除错误。

图25-12: 同步接收（主模式，SREN）

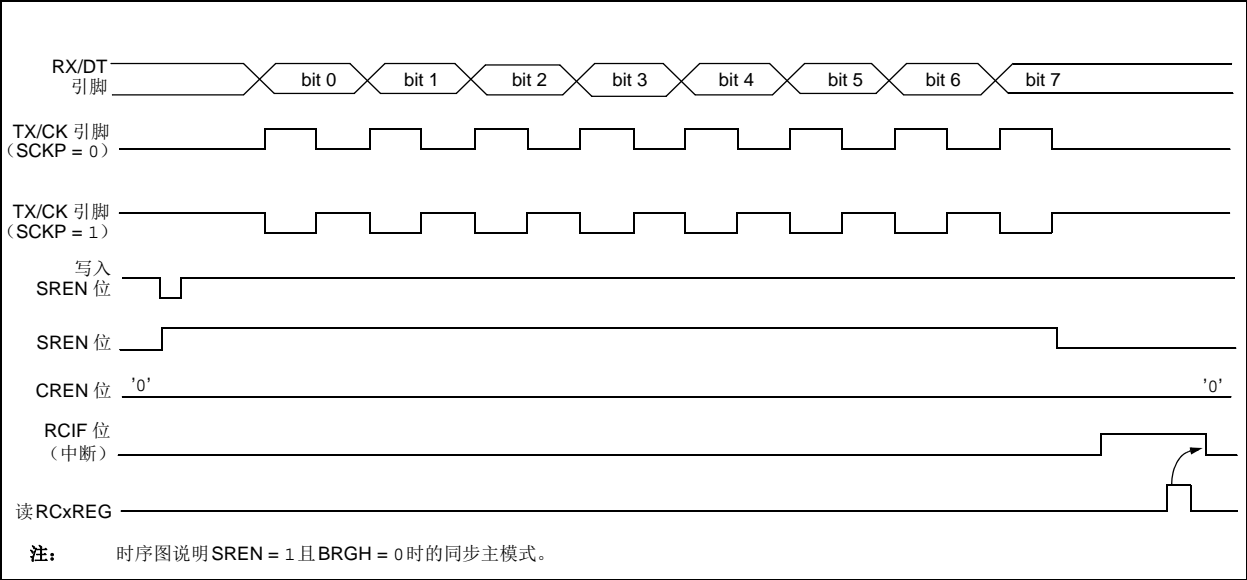


表25-8: 与同步主接收相关的寄存器汇总

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	160
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	332
CKPPS	—	—	—	CKPPS<4:0>					182和180
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RC1REG	EUSART接收数据寄存器								325*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	331
RXPPS	—	—	—	RXPPS<4:0>					182和180
RxyPPS	—	—	—	RxyPPS<4:0>					180
SP1BRGL	BRG<7:0>								333*
SP1BRGH	BRG<15:8>								333*
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	159
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
TX1STA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	330

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)F1619。

2: 未实现，读为1。

## 25.5.2 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。

### 25.5.2.1 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原理是相同的（见第25.5.1.3节“同步主发送”）。

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。

### 25.5.2.2 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。



**表 25-9: 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	160
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	332
CKPPS	—	—	—	CKPPS<4:0>					182 和 180
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	331
RXPPS	—	—	—	RXPPS<4:0>					182 和 180
RxyPPS	—	—	—	RxyPPS<4:0>					180
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	159
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
TX1REG	EUSART 发送数据寄存器								322*
TX1STA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	330

**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)F1619。

**2:** 未实现, 读为 1。

# PIC16(L)F1615/9

## 25.5.2.3 EUSART同步从接收

除下列各项外，同步主模式和从模式的工作原理是相同的（第25.5.1.5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。接收到该字后，RSR寄存器会将数据发送到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。

## 25.5.2.4 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，以使能接收。
6. 接收完成时RCIF位将被置1。如果RCIE位已置1，则产生中断。
7. 如果使能了9位模式，从RCxSTA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（该位将EUSART复位），可以清除错误。

表25-10：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	160
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	332
CKPPS	—	—	—	CKPPS<4:0>					182和180
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	105
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	106
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	111
RC1REG	EUSART接收数据寄存器								325*
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	331
RXPPS	—	—	—	RXPPS<4:0>					182和180
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	159
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	166
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173
TX1STA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	330

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)F1619。

2： 未实现，读为1。

## 25.6 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使发送或接收移位寄存器工作必需的信号。

同步从模式使用外部产生的时钟使发送和接收移位寄存器工作。

### 25.6.1 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25.5.2.4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。数据字从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。

从休眠状态唤醒时，将执行SLEEP指令后紧跟的指令。如果INTCON寄存器的全局中断允许（GIE）位也置1，则调用地址004h处的中断服务程序。

### 25.6.2 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（见第25.5.2.2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。

从休眠状态唤醒时，将执行SLEEP指令后紧跟的指令。如果全局中断允许（GIE）位也置1，则调用地址0004h处的中断服务程序。

26.0 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（Pulse-Width Modulation, PWM）信号的外设。在捕捉模式下，外设允许对事件的持续时间进行计时。当超过预先确定的时间时，比较模式允许用户触发一个外部事件。PWM模式可以产生不同频率和占空比的脉宽调制信号。

该器件系列包含两个标准捕捉/比较/PWM模块（CCP1和CCP2）。

**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。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。

26.1 捕捉模式

本节介绍的捕捉模式功能适用于所有CCP模块而且是相同的。

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）
- 电平变化中断触发（IOC\_interrupt）

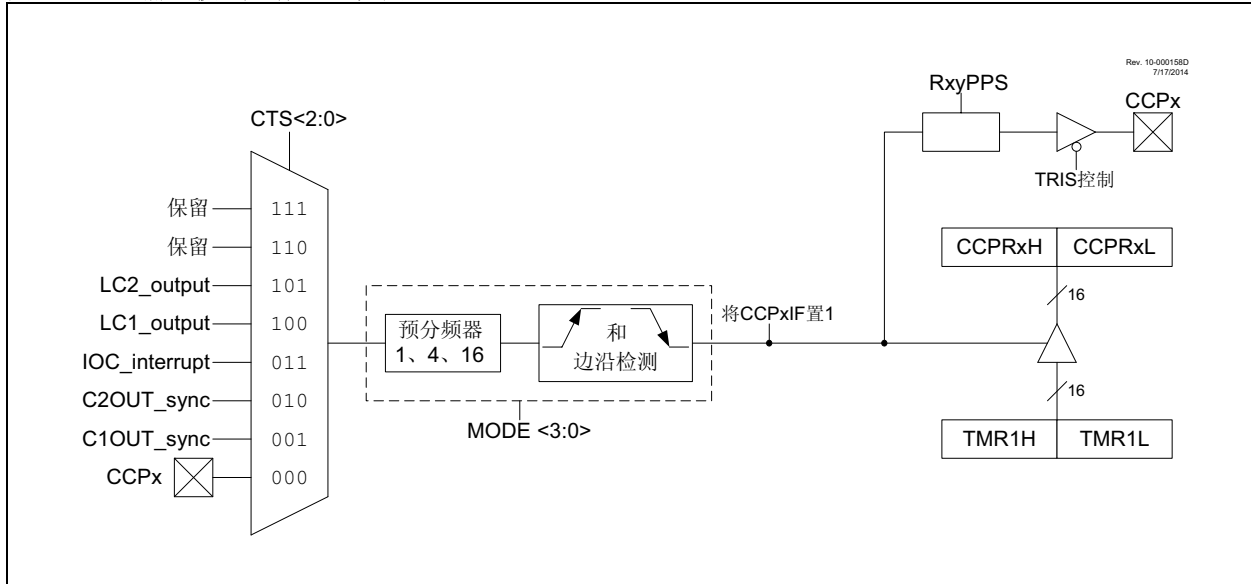
进行捕捉时，PIRx寄存器的中断请求标志位CCPxIF被置1。该中断标志必须用软件清零。如果在CCPRxH和CCPRxL寄存器对中的值被读取之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖。

图26-1给出了捕捉操作的简化框图。

26.1.1 CCP引脚配置

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

图26-1: 捕捉模式工作原理框图



## 26.1.2 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“带门控制的Timer1/3/5模块”。

## 26.1.3 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PIEx寄存器的CCPxIE中断允许位清零以避免错误中断。此外，用户应在工作模式的任何改变之后清零PIRx寄存器的CCPxIF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CCPx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。

## 26.1.4 CCP 预分频器

通过CCPxCON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CCPxCON寄存器的EN位来关闭模块。

## 26.1.5 休眠期间的捕捉操作

捕捉模式能否正常工作取决于Timer1模块。有两个选项可用于在捕捉模式下驱动Timer1模块。它可由指令时钟（Fosc/4）驱动，或由外部时钟源驱动。

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

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。

## 26.1.6 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CWG的输入信号、用作ADC的自动转换触发信号、用作TMR2模块的外部复位信号、用作SMT的窗口输入，以及用作CLC模块的输入。此外，CCPx引脚输出可通过PPS（见第13.2节“PPS输出”）映射到输出引脚。

26.2 比较模式

本节介绍的比较模式功能适用于所有CCP模块而且是相同的。

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
- 可选地复位TMR1

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。

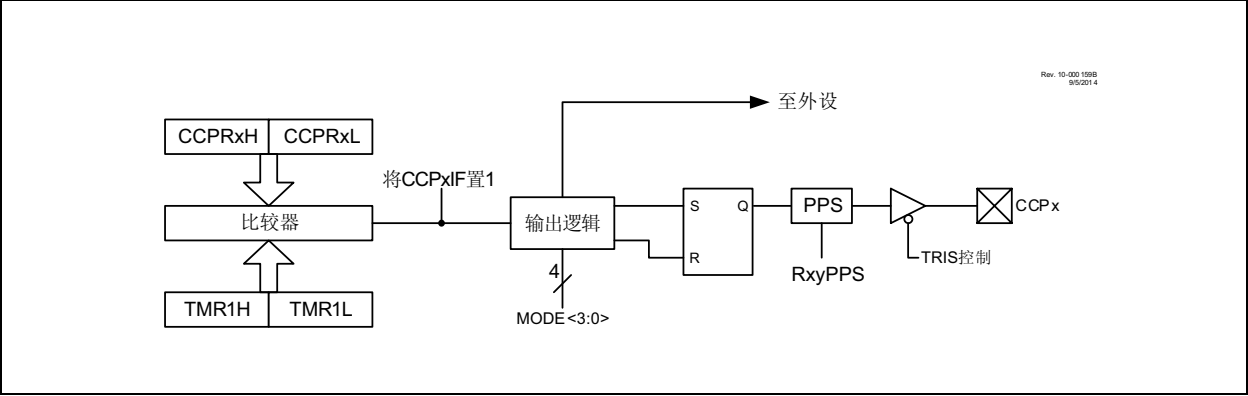
所有比较模式都能产生中断。

图26-2给出了比较操作的简化框图。

26.2.1 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。

图26-2: 比较模式工作原理框图



## 26.2.2 TIMER1 模式资源

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

关于配置Timer1的更多信息，请参见[第22.0节“带门控控制的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（Fosc/4）或外部时钟源提供。

## 26.2.3 软件中断模式

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

## 26.2.4 休眠期间的比较操作

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

## 26.2.5 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 CCP 被配置为的比较模式。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。这意味着脉冲宽度由 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。该输出可用作 CWG 的输入信号、用作 ADC 的自动转换触发信号、用作 TMR2 模块的外部复位信号、用作 SMT 的窗口输入，以及用作 CLC 模块的输入。此外，CCPx 引脚输出可通过 PPS（见[第13.2节“PPS输出”](#)）映射到输出引脚。

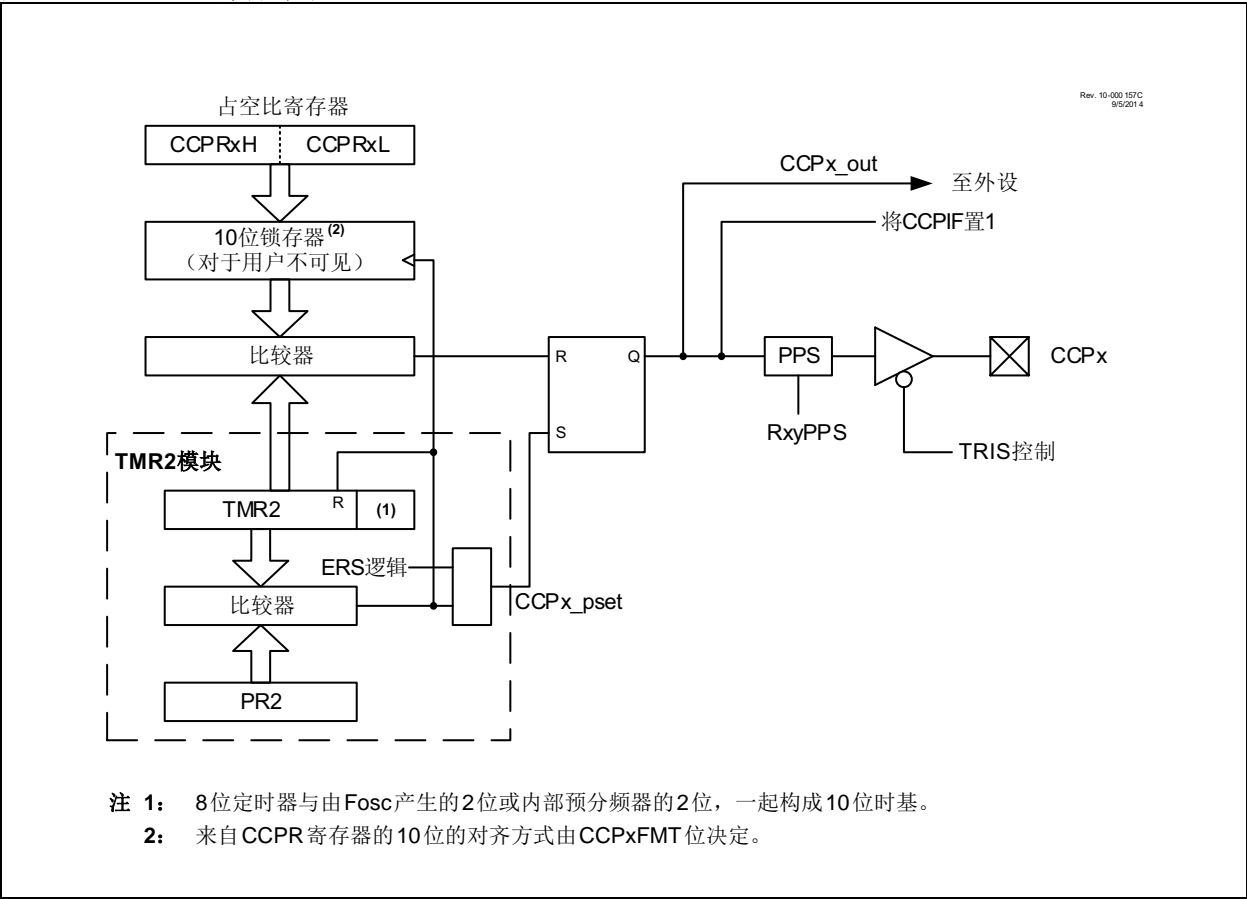
26.3 PWM概述

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

PWM分辨率定义为可以在单个PWM周期中出现的最大步幅数量。分辨率越高，就可以越精确地控制脉宽时间，从而更精确地控制在负载上的供电量。

占空比这一术语描述开启时间与关闭时间之间以百分比形式表示的比例，0%代表完全关闭，100%代表完全开启。占空比越低，对应的供电量就越低；占空比越高，对应的供电量就越高。

图26-3： PWM的简化框图





26.3.1 标准 PWM 操作

本节介绍的标准 PWM 功能适用于所有 CCP 模块而且是相同的。

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：

- PR2/4/6 寄存器
- T2CON/T4CON/T6CON 寄存器
- CCPRxH:CCPRxL 寄存器对

图 26-3 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。

26.3.2 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. 确定使能哪个定时器来按时钟控制 CCP：Timer2/4/6。
3. 将 PWM 周期值装入关联的 PR2/4/6 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 Timer2/4/6：
  - 清零 PIRx 寄存器的 TMR2IF/TMR4IF/TMR6IF 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：
  - 等待直到定时器上溢，PIRx 寄存器的 TMR2IF/TMR4IF/TMR6IF 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。

## 26.4 CCP/PWM时钟选择

PIC16(L)F1615/9 允许每个 CCP 和 PWM 模块选择控制模块的定时器源。每个模块具有独立的选择。

由于最多存在 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一个。

CCPTMRS 寄存器用于选择要使用的定时器。

### 26.4.1 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。

### 26.4.2 PWM 周期

PWM 周期可通过 Timer2/4/6 的 PR2/4/6 寄存器来指定。PWM 周期可由公式 26-1 计算。

#### 公式 26-1: PWM 周期

$$PWM \text{ 周期} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (TMR2 \text{ 预分频值})$$

注 1:  $T_{OSC} = 1/F_{OSC}$

当 TMR2/4/6 中的值与其相应的 PR2/4/6 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 26-1）。

### 26.4.3 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，特定的位序由 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。图 26-4 给出了它的图示。这些位可以

在任意时间写入。在周期结束（即，PR2/4/6 和 TMR2/4/6 寄存器发生匹配）之前，占空比值不会被锁存到内部锁存器中。

公式 26-2 用于计算 PWM 脉冲宽度。公式 26-3 用于计算 PWM 占空比。

#### 公式 26-2: 脉冲宽度

$$\text{脉冲宽度} = CCPRxH:CCPRxL \cdot T_{OSC} \cdot (TMR2 \text{ 预分频值})$$

#### 公式 26-3: 占空比

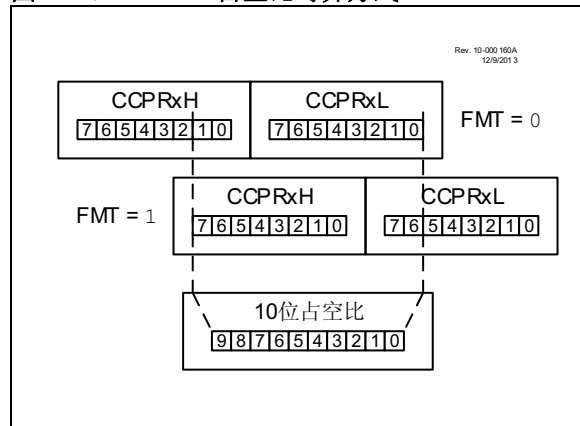
$$\text{占空比} = \frac{CCPRxH:CCPRxL}{4(PR2 + 1)}$$

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 26-3）。

图 26-4: CCPx 占空比对齐方式



### 26.4.4 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 个不连续的占空比，而 8 位分辨率将可得到 256 个不连续的占空比。

当PR2/4/6为255时，PWM最大分辨率为10位。分辨率是PR2/4/6寄存器值的函数，如[公式26-4](#)所示。

公式26-4:     PWM分辨率

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

**注：**     如果脉冲宽度值比周期长，则指定的PWM引脚将保持不变。

# PIC16(L)F1615/9

表26-1: PWM频率和分辨率示例 (Fosc = 20 MHz)

PWM 频率	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频值	16	4	1	1	1	1
PR2 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率 (位)	10	10	10	8	7	6

表26-2: PWM频率和分辨率示例 (Fosc = 8 MHz)

PWM 频率	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频值	16	4	1	1	1	1
PR2 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率 (位)	8	8	8	6	5	5

## 26.4.5 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”](#)。

## 26.4.6 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。

## 26.4.7 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。该输出可用作CWG的输入信号、用作ADC的自动转换触发信号、用作TMR2模块的外部复位信号、用作SMT的窗口输入, 以及用作CLC模块的输入。此外, CCPx引脚输出可通过PPS(见[第13.2节“PPS输出”](#))映射到输出引脚。

26.5 寄存器定义：CCP控制

寄存器 26-1: CCPxCON: CCPx控制寄存器

R/W-0/0	U/U-0/0	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	<b>EN:</b> CCPx模块使能位 1 = 使能CCPx 0 = 禁止CCPx
bit 6	<b>未实现:</b> 读为0
bit 5	<b>OUT:</b> CCPx输出数据位 (只读)
bit 4	<b>FMT:</b> CCPW (脉冲宽度) 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	<b>MODE&lt;3:0&gt;:</b> 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: 发生比较匹配时将输出清零 1000 = 比较模式: 发生比较匹配时将输出置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)F1615/9

## 寄存器 26-2: 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>		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 = 在 PWM 模式下, PWM4 基于 Timer6 产生  
01 = 在 PWM 模式下, PWM4 基于 Timer4 产生  
00 = 在 PWM 模式下, PWM4 基于 Timer2 产生
- 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 产生  
01 = 在 PWM 模式下, PWM3 基于 Timer4 产生  
00 = 在 PWM 模式下, PWM3 基于 Timer2 产生
- 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 产生  
01 = 在 PWM 模式下, CCP2 基于 Timer4 产生  
00 = 在 PWM 模式下, CCP2 基于 Timer2 产生
- 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 产生  
01 = 在 PWM 模式下, CCP1 基于 Timer4 产生  
00 = 在 PWM 模式下, CCP1 基于 Timer2 产生

寄存器 26-3: 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> —— 脉冲宽度低 8 位  
MODE = PWM 模式且 FMT = 1  
**CCPRxL<7:6>**: CCPW<1:0> —— 脉冲宽度低 2 位  
**CCPRxL<5:0>**: 不使用

# PIC16(L)F1615/9

寄存器 26-4: 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>**: 不要使用

**CCPRxH<1:0>**: CCPW<9:8> —— 脉冲宽度高2位

MODE = PWM 模式且 FMT = 1

**CCPRxH<7:0>**: CCPW<9:2> —— 脉冲宽度高8位

寄存器 26-5: 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							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 = LC4\_out

110 = LC3\_out

101 = LC2\_out

100 = LC1\_out

011 = IOC\_interrupt

010 = C2\_OUT\_sync

001 = C1\_OUT\_sync

000 = CCPx 引脚



**表26-3: 与标准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<1:0>		<a href="#">364</a>
CCPxCON	EN	—	OUT	FMT	MODE<3:0>				<a href="#">361</a>
CCPRxL	捕捉/比较/PWM寄存器x（LSB）								<a href="#">363</a>
CCPRxH	捕捉/比较/PWM寄存器x（MSB）								<a href="#">364</a>
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		<a href="#">362</a>
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	<a href="#">105</a>
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	<a href="#">106</a>
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	<a href="#">107</a>
PR2	Timer2周期寄存器								<a href="#">244*</a>
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				<a href="#">263</a>
TMR2	Timer2模块寄存器								<a href="#">244</a>
PR4	Timer4周期寄存器								<a href="#">244*</a>
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				<a href="#">263</a>
TMR4	Timer4模块寄存器								<a href="#">244</a>
PR6	Timer6周期寄存器								<a href="#">244*</a>
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				<a href="#">263</a>
TMR6	Timer6模块寄存器								<a href="#">244</a>
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	<a href="#">159</a>

**图注:** — = 未实现位, 读为0。PWM不使用阴影单元。

\* 提供寄存器信息的页。

**注 1:** 未实现, 读为1。

27.0 脉宽调制（PWM）模块

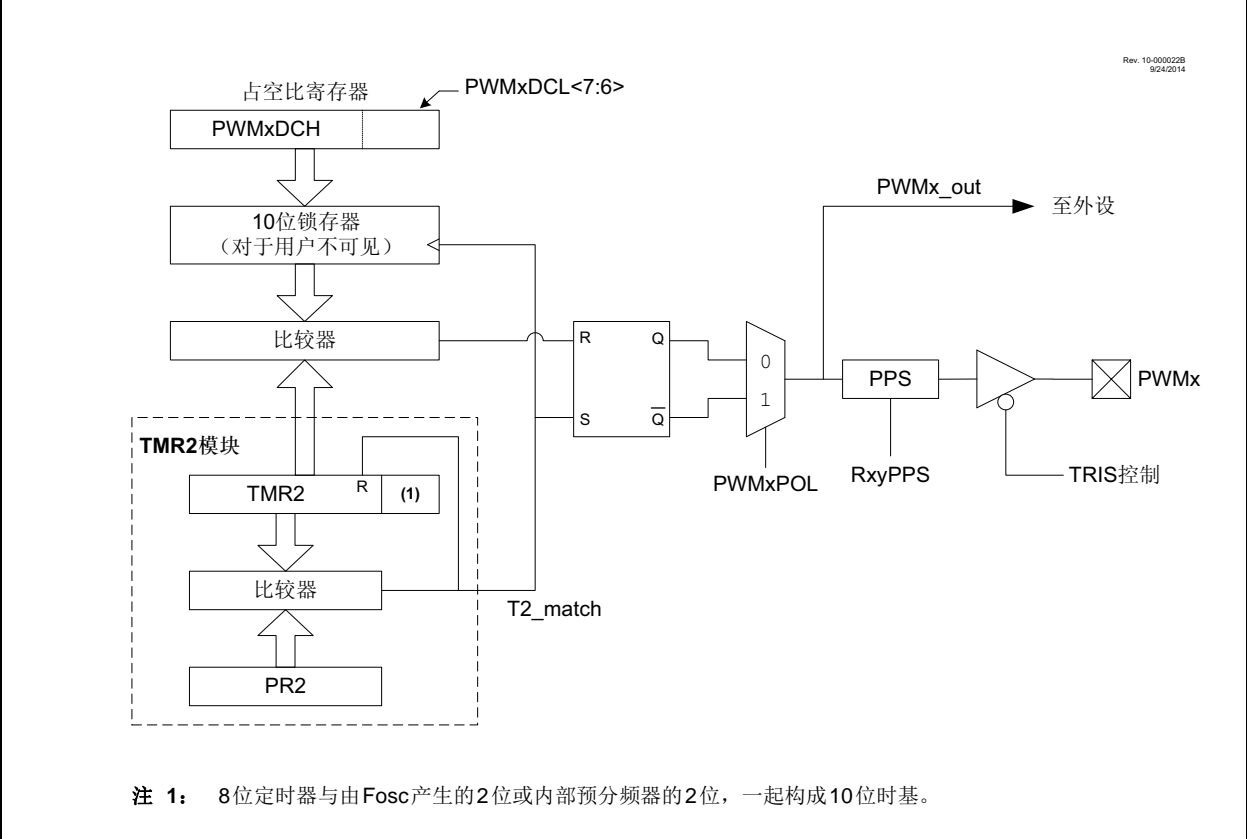
PWM 模块可产生由占空比、周期和分辨率决定的脉宽调制信号，占空比、周期和分辨率则通过以下寄存器进行配置：

- PR2
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

图27-1 给出了PWM操作的简化框图。

关于如何设置该模块使之工作于PWM模式的详细步骤，请参见第27.1.9节“使用PWMx引脚设置PWM操作”。

图27-1： PWM 的简化框图



27.1 PWMx 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。

27.1.1 基本操作

PWM模块可产生一个10位分辨率的输出。Timer2和PR2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（8 MSb）和PWMxDCL<7:6>（2 LSB）寄存器指定的值时，每个PWMx都会清零。当值大于等于PR2时，PWM输出永远不会清零（占空比为100%）。

**注：** PWMxDCH和PWMxDCL寄存器是双重缓冲的。当Timer2与PR2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。

27.1.2 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。

27.1.3 PWM周期

PWM周期可通过Timer2的PR2寄存器来指定。PWM周期可由公式27-1计算。

公式27-1: PWM周期

$$PWM\ 周期 = [(PR2) + 1] \bullet 4 \bullet T_{osc} \bullet$$

(TMR2 预分频值)

**注：**  $T_{osc} = 1/F_{osc}$

当TMR2中的值与PR2中的值相等时，在下一个递增周期将发生以下3个事件：

- TMR2被清零
- 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。

27.1.4 PWM占空比

PWM占空比通过将10位值写入PWMxDCH和PWMxDCL寄存器来指定。PWMxDCH寄存器包含高8位，而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27-2用于计算PWM脉冲宽度。

公式27-3用于计算PWM占空比。

公式27-2: 脉冲宽度

$$脉冲宽度 = (PWMxDCH:PWMxDCL<7:6>) \bullet T_{osc} \bullet (TMR2\ 预分频值)$$

**注：**  $T_{osc} = 1/F_{osc}$

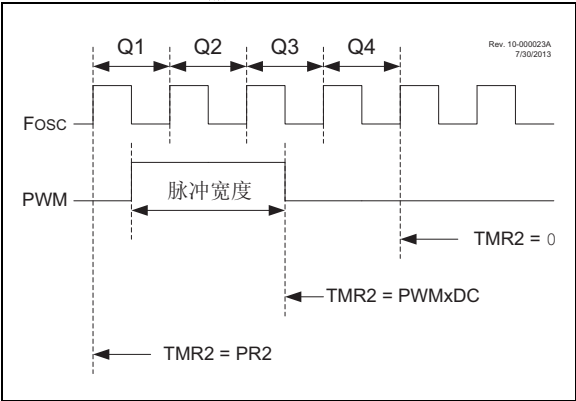
公式27-3: 占空比

$$占空比 = \frac{(PWMxDCH:PWMxDCL<7:6>)}{4(PR2 + 1)}$$

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。

图27-2显示了在占空比设置为产生最小可能脉冲时，PWM信号的波形。

图27-2: PWM输出



# PIC16(L)F1615/9

## 27.1.5 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个不连续的占空比，而8位分辨率将可得到256个不连续的占空比。

当PR2为255时，最大PWM分辨率为10位。分辨率是PR2寄存器值的函数，如公式27-4所示。

公式27-4: PWM分辨率

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

注：如果脉冲宽度值比周期长，则指定的PWM引脚将保持不变。

表27-1: 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
PR2 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率（位）	10	10	10	8	7	6.6

表27-2: 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
PR2 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率（位）	8	8	8	6	5	5

## 27.1.6 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。

## 27.1.7 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”。

## 27.1.8 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。

## 27.1.9 使用PWMx引脚设置PWM操作

当使用PWMx引脚将模块配置为PWM操作时，可采用以下步骤：

1. 通过将相关的TRIS位置1，禁止PWMx引脚输出驱动器。
2. 清零PWMxCON寄存器。
3. 将PWM周期值装入PR2寄存器。
4. 清零PWMxDCH寄存器和PWMxDCL寄存器的bit<7:6>。
5. 配置并启动Timer2：
  - 清零PIR1寄存器的TMR2IF中断标志位。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. 通过将关联的TRIS位清零并将PWMxCON寄存器的PWMxOE位置1，使能PWMx引脚输出驱动器。
8. 通过将相应值装入PWMxCON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。如果并非必须以一个完整PWM信号开始，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PWMx引脚输出。

# PIC16(L)F1615/9

## 27.2 寄存器定义：PWM控制

寄存器 27-1: 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
PWMxEN	—	PWMxOUT	PWMxPOL	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7 **PWMxEN:** PWM模块使能位

1 = 使能PWM模块

0 = 禁止PWM模块

bit 6 **未实现:** 读为0

bit 5 **PWMxOUT:** PWM模块输出值位

bit 4 **PWMxPOL:** PWMx输出极性选择位

1 = PWM输出为低电平有效

0 = PWM输出为高电平有效

bit 3-0 **未实现:** 读为0

寄存器 27-2: PWMxDCH: PWM占空比高位

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PWMxDCH<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 **PWMxDCH<7:0>:** PWM占空比最高有效位

这些位是PWM占空比的高位。低2位位于PWMxDCL寄存器中。

寄存器 27-3: 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
PWMxDCL<7:6>		—	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 **PWMxDCL<7:6>:** PWM占空比最低有效位

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27-3: 与PWM相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
PR2	Timer2模块周期寄存器								368*
PWM3CON	EN	—	OUT	POL	—	—	—	—	370
PWM3DCH	DC<9:2>								370
PWM3DCL	DC<1:0>		—	—	—	—	—	—	370
PWM4CON	EN	—	OUT	POL	—	—	—	—	370
PWM4DCH	DC<9:2>								370
PWM4DCL	DC<1:0>		—	—	—	—	—	—	370
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				263
TMR2	Timer2模块寄存器								244*
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	159
TRISC	TRISC7 <sup>(2)</sup>	TRISC6 <sup>(2)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	173

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

\* 提供寄存器信息的页。

注 1: 未实现, 读为1。

2: 仅限PIC16(L)F1619。

## 28.0 互补波形发生器（CWG）模块

互补波形发生器（CWG）可产生半桥、全桥和转向PWM波形。它与以前的ECCP功能保持向后兼容。

CWG具有以下特性：

- 6种工作模式：
  - 同步转向模式
  - 异步转向模式
  - 全桥模式，正向
  - 全桥模式，反向
  - 半桥模式
  - 推挽模式
- 输出极性控制
- 输出转向
  - 与上升沿事件进行同步
  - 立即生效
- 独立的6位上升沿和下降沿事件死区定时器
  - 按时钟控制的死区
  - 独立的上升沿和下降沿死区使能
- 可使用以下方法启动自动关断控制：
  - 可选关断源
  - 自动重启使能
  - 自动关断引脚改写控制

## 28.1 基本操作

CWG模块可以在6种不同模式下工作，这些模式由CWGxCON0寄存器的MODE指定：

- 半桥模式（图28-9）
- 推挽模式（图28-2）
  - 全桥模式，正向（图28-3）
  - 全桥模式，反向（图28-3）
- 转向模式（图28-10）
- 同步转向模式（图28-11）

可能需要防止电路发生故障、反馈事件太晚送达或根本不送达的可能性。在这种情况下，必须在故障条件造成损坏之前终止有效驱动。因此，所有输出模式均支持自动关断，第28.10节“自动关断”对它进行了介绍。

### 28.1.1 半桥模式

在半桥模式下，将以输入的真值和反相形式生成两个输出信号，如图28-9所示。在两个输出之间插入不重叠（死区）时间，以防止各种电源应用中产生直通电流。

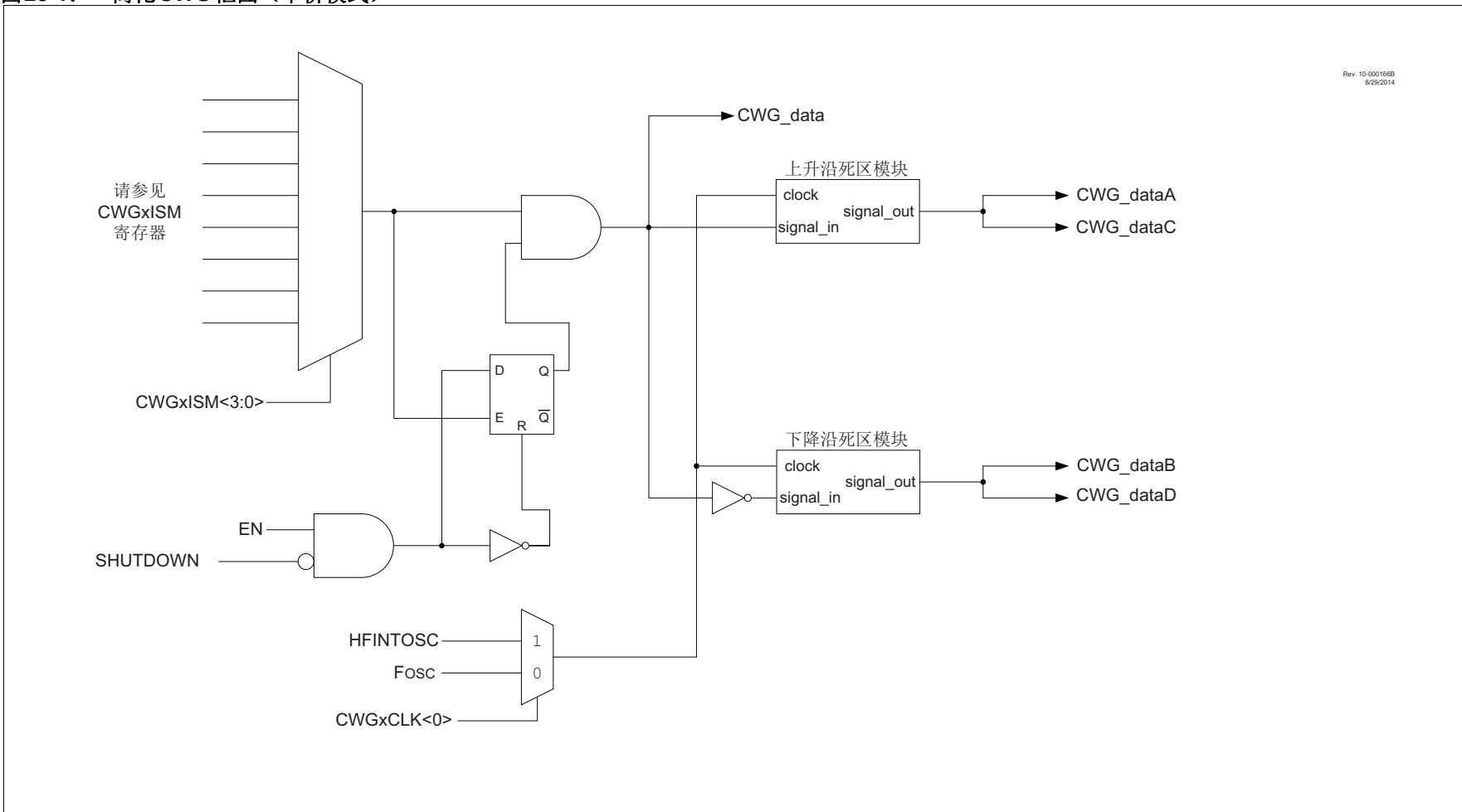
第28.5节“死区控制”对死区控制进行了介绍。

未用输出CWGxC和CWGxD驱动类似的信号，它们的极性分别由CWGxCON1寄存器的POLC和POLD位独立控制。



Rev. 10-000166B  
8/29/2014

图28-1: 简化CWG框图（半桥模式）



## 28.1.2 推挽模式

在推挽模式下，将生成两个输出信号，它们为输入的交替副本，如图28-2所示。这种交替可以产生驱动一些基于变压器的电源设计所需的推挽效应。

推挽排序器在每次EN = 0或发生自动关断事件时复位。该排序器由第一个输入脉冲提供时钟，第一个输出出现在CWGxA上。

未用输出CWGxC和CWGxD分别驱动CWGxA和CWGxB的副本，但它们的极性分别由CWGxCON1寄存器的POLC和POLD位控制。

## 28.1.3 全桥模式

在正向和反向全桥模式下，3个输出驱动静态值，第4个输出则通过输入数据信号进行调制。在正向全桥模式下，CWGxA被驱动为其有效状态，CWGxB和CWGxC被驱动为其无效状态，CWGxD通过输入信号进行调制。在反向全桥模式下，CWGxC被驱动为其有效状态，CWGxA和CWGxD被驱动为其无效状态，CWGxB通过输入信号进行调制。在全桥模式下，在从正向切换为反向（或反之）时会应用死区周期。第28.5节“死区控制”对这种死区控制进行了介绍，第28.6节“上升沿和反向死区”和第28.7节“下降沿和正向死区”提供了更多详细信息。

模式选择可以在正向和反向之间切换，方法是翻转CWGxCON0的MODE<0>位，同时将MODE<2:1>保持静态，而无需禁止CWG模块。

Rev. 10-000167B  
8/28/2014

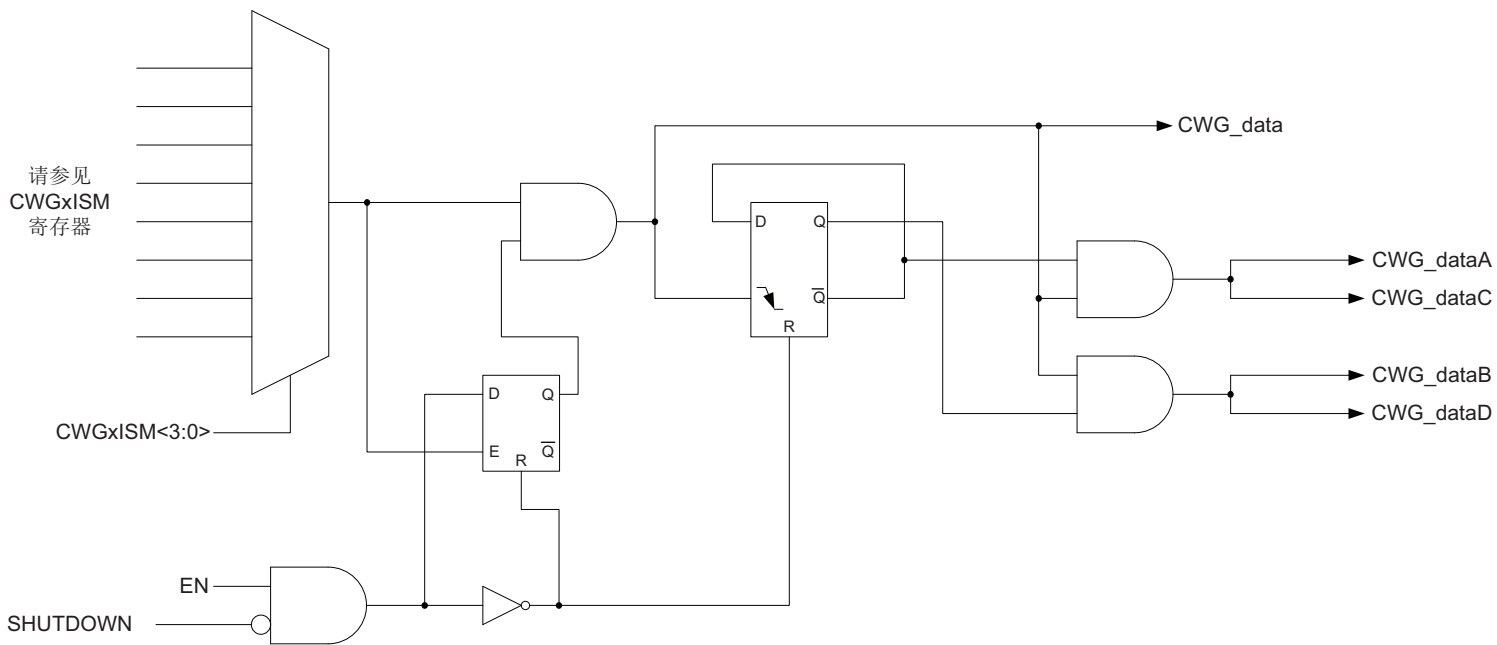
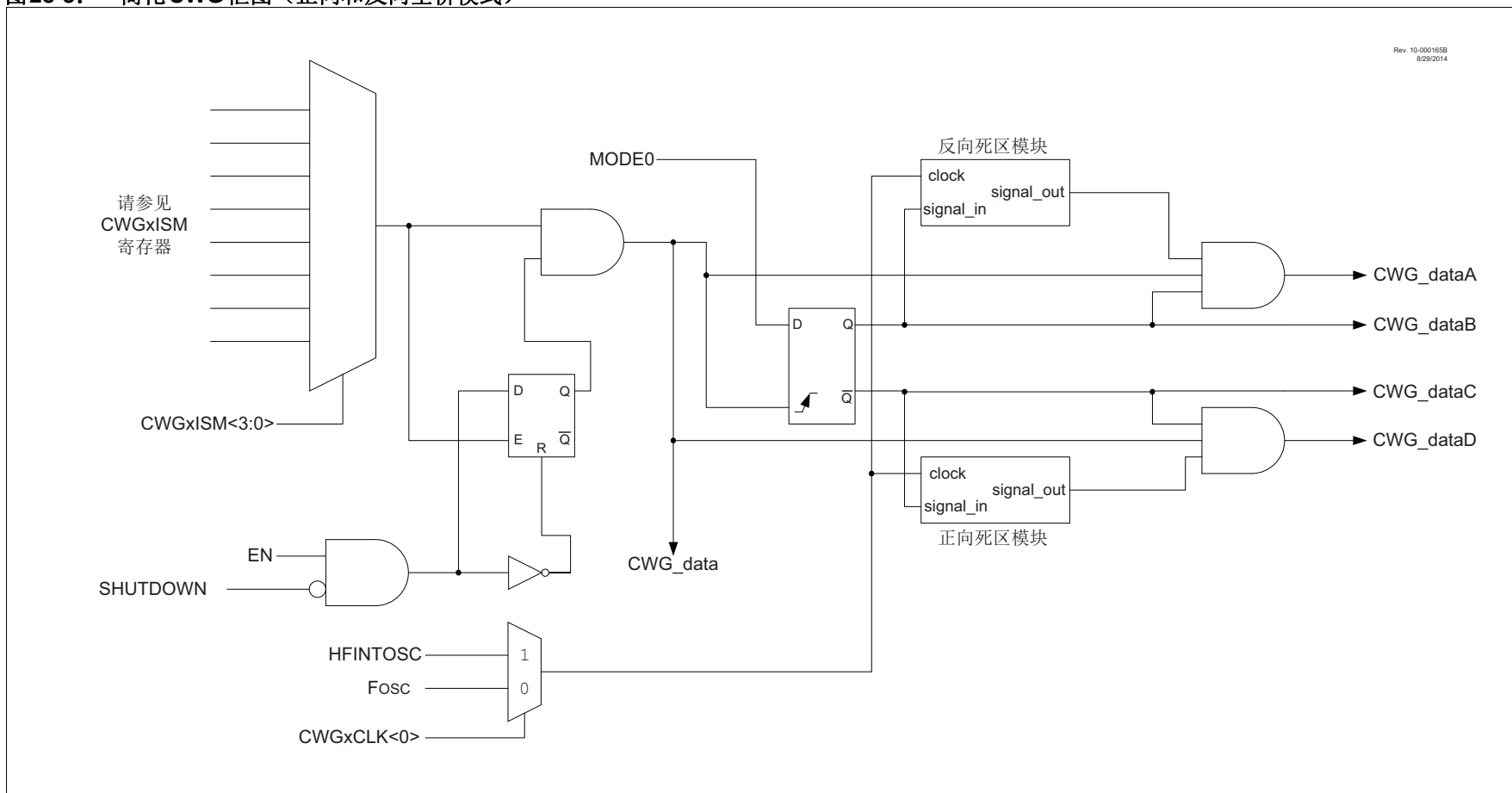


图28-2: 简化CWG框图（推挽模式）

请参见  
CWGxISM  
寄存器

图28-3: 简化CWG框图（正向和反向全桥模式）

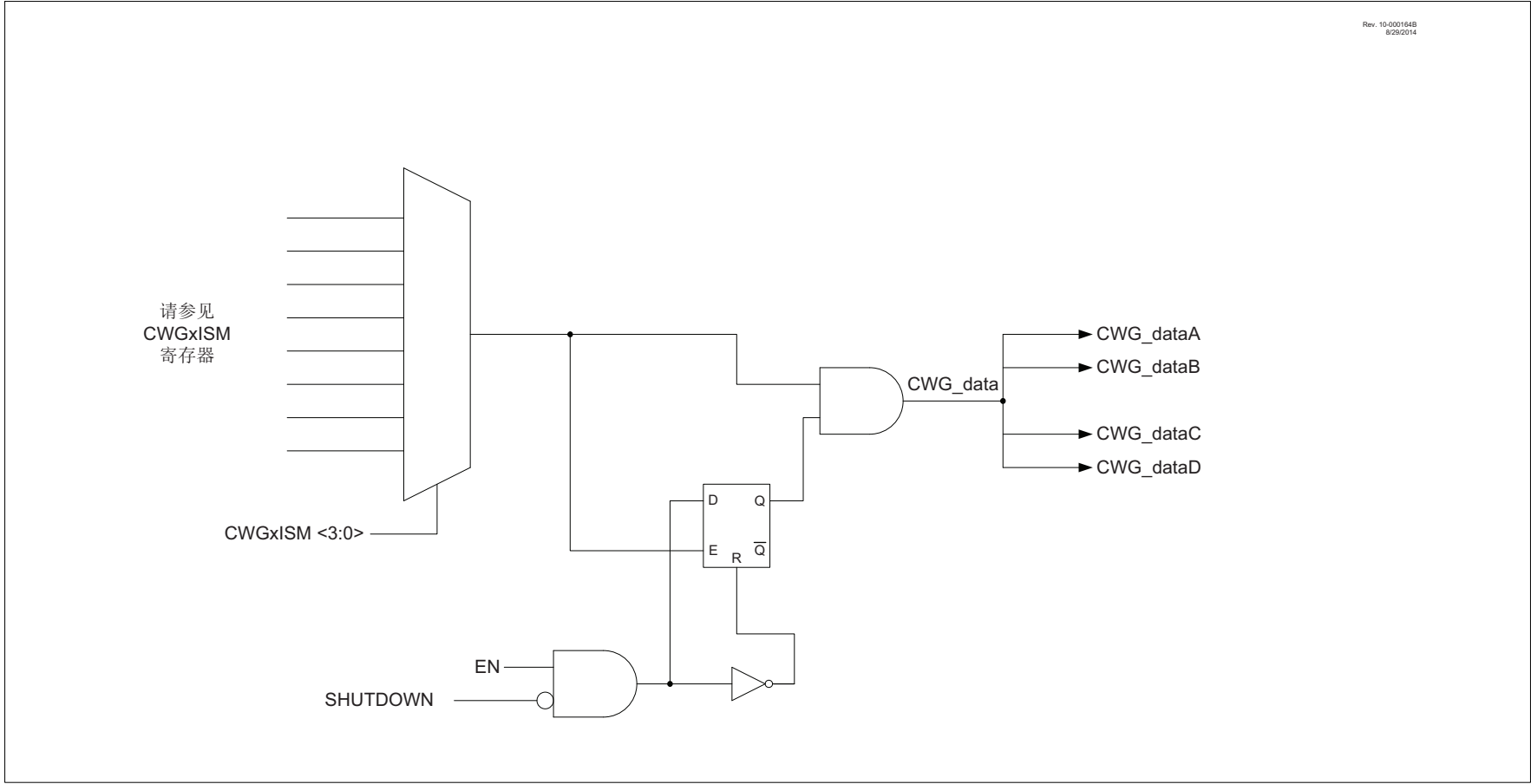


## 28.1.4 转向模式

在转向模式下，可以将数据输入转向到4个CWG输出引脚中的任意或全部引脚。在同步转向模式下，对转向选择寄存器的更改在出现下一个上升沿输入时生效。

在非同步模式下，转向在下一个指令周期生效。更多详细信息，请参见[第28.9节“CWG转向模式”](#)。

图28-4: 简化CWG框图（输出转向模式）



28.2 时钟源

CWG 模块允许选择以下时钟源：

- Fosc（系统时钟）
- HFINTOSC（仅限 16 MHz）

时钟源使用 CWGxCLKCON 寄存器的 CS 位进行选择。

28.3 可选输入源

CWG 基于表 28-1 中的输入源来生成输出波形。

表 28-1： 可选输入源

输入源外设	信号名称
CWG 引脚	CWGxINPPS 选择
比较器 C1	C1_OUT_sync
比较器 C2	C2_OUT_sync
CCP1	CCP1_out
CCP2	CCP2_out
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
CLC4	LC4_out
PWM3	PWM3_out
PWM4	PWM4_out

输入源使用 CWGxISM 寄存器进行选择。

28.4 输出控制

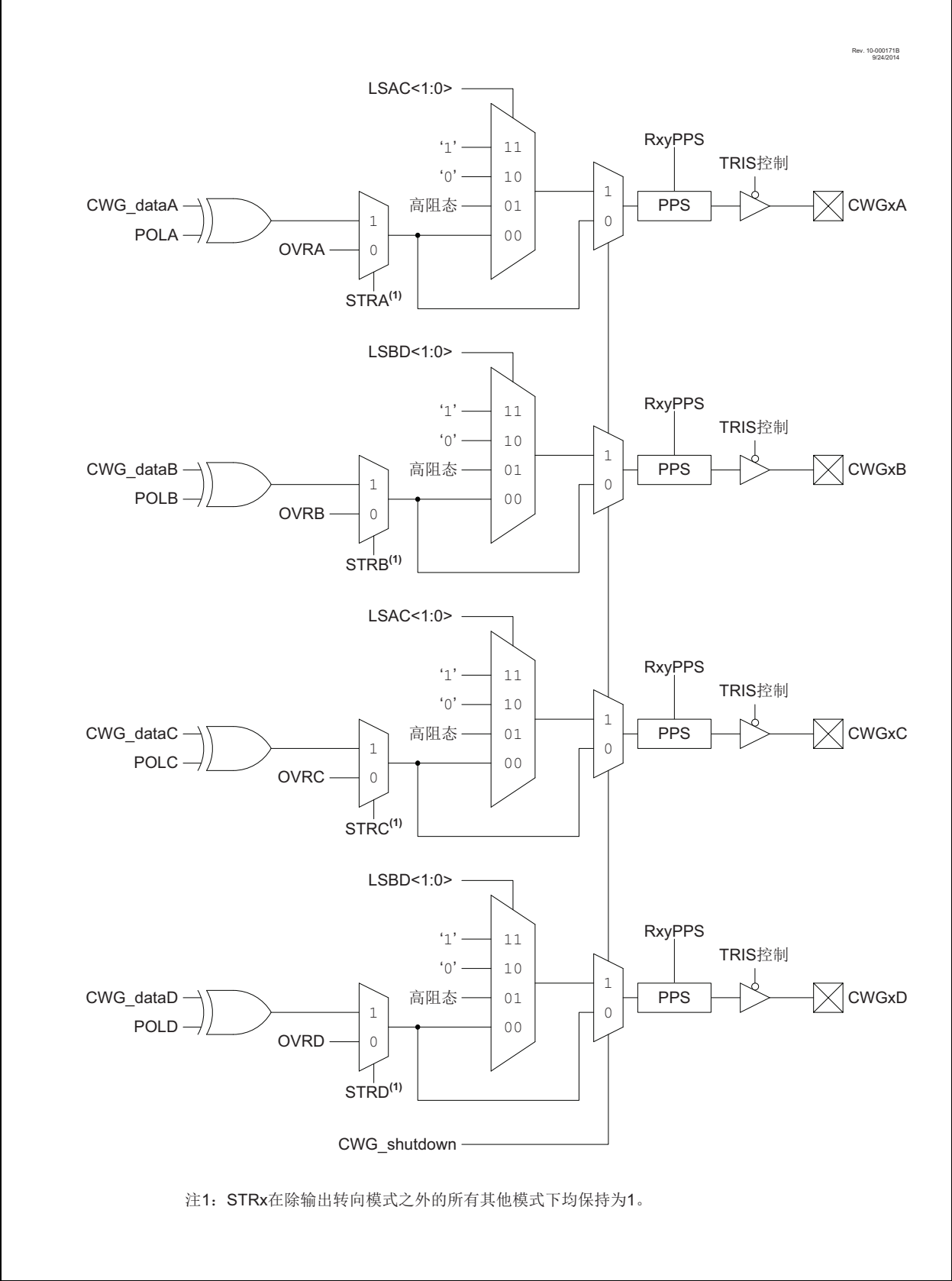
28.4.1 输出使能

每个 CWG 输出引脚都具有独立的输出使能控制。输出使能使用 Gx1OEx<3:0> 位进行选择。当输出使能控制清零时，模块对引脚无控制权。当输出使能置 1 时，每次选择端口极性都将对引脚施加改写值或有效 PWM 波形。输出引脚使能取决于 CWGxCON0 寄存器中的模块使能位 EN。当 EN 清零时，CWG 输出使能和 CWG 驱动电平没有任何作用。

28.4.2 极性控制

每个 CWG 输出的极性可以单独进行选择。当输出极性位置 1 时，相应的输出为高电平有效。清零输出极性位时，相应输出将配置为低电平有效。但是，极性不会影响改写电平。输出极性使用 CWGxCON1 的 POLx 位进行选择。自动关断和转向选项不会受极性影响。

图28-5: CWG输出框图





## 28.5 死区控制

死区控制用于提供不重叠的PWM信号，以防止PWM开关中产生直通电流。半桥和全桥模式会采用死区操作。CWG包含两个6位死区计数器。一个用于半桥模式下的输入源控制的上升沿，或用于全桥模式下的反向死区。另一个用于半桥模式下的输入源控制的下降沿，或用于全桥模式下的正向死区。

死区的计时方式是对CWG时钟周期进行计数，从0开始一直计数至上升沿或下降沿死区计数器寄存器中的值。请分别参见CWGxDBR和CWGxDBF寄存器。

### 28.5.1 半桥模式下的死区功能

在半桥模式下，死区计数器决定正常输出下降沿和反相输出上升沿之间的延时。图28-9中可以看到这一点。

### 28.5.2 全桥模式下的死区功能

在全桥模式下，死区计数器在方向改变时使用。CWGxCON0寄存器的MODE<0>位可以在CWG运行时置1或清零，从而可以从正向变为反向模式。CWGxA和CWGxC信号会在方向改变后出现第一个上升输入沿时立即改变，但调制信号（CWGxB或CWGxD，取决于改变的方向）会遇到由死区计数器决定的延时。如图28-3所示。

## 28.6 上升沿和反向死区

CWGxDBR控制CWGxA前沿（半桥模式）或CWGxB前沿（全桥模式）处的上升沿死区时间。CWGxDBR值是双重缓冲的。当EN = 0时，会在写入CWGxDBR时立即装入CWGxDBR寄存器。当EN = 1时，软件必须先将CWGxCON0寄存器的LD位置1，然后将在CWG输入信号的下一个下降沿装入缓冲器。如果输入源信号出现的时间不足以完成计数，则相应输出上不会产生任何输出。

## 28.7 下降沿和正向死区

CWGxDBF控制CWGxB前沿（半桥模式）或CWGxD前沿（全桥模式）处的死区时间。CWGxDBF值是双重缓冲的。当EN = 0时，会在写入CWGxDBF时立即装入CWGxDBF寄存器。当EN = 1时，软件必须先将CWGxCON0寄存器的LD位置1，然后将在CWG输入信号的下一个下降沿装入缓冲器。如果输入源信号出现的时间不足以完成计数，则相应输出上不会产生任何输出。

示例请参见图28-6和图28-7。

图28-6: 死区操作, CWGXDBR = 0X01, CWGXDBF = 0X02

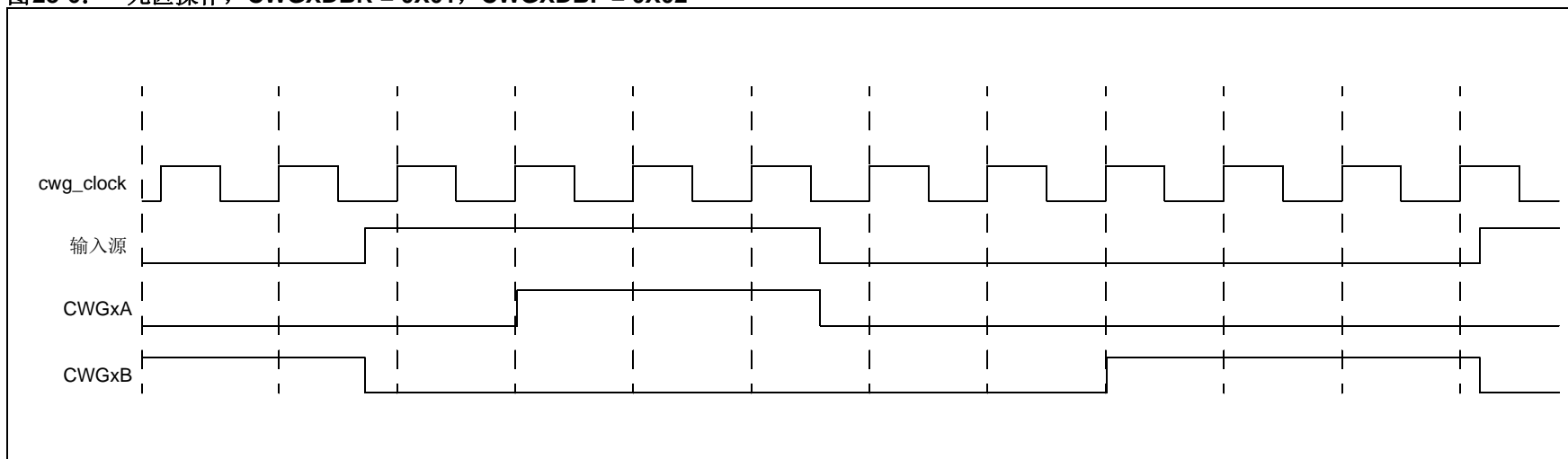
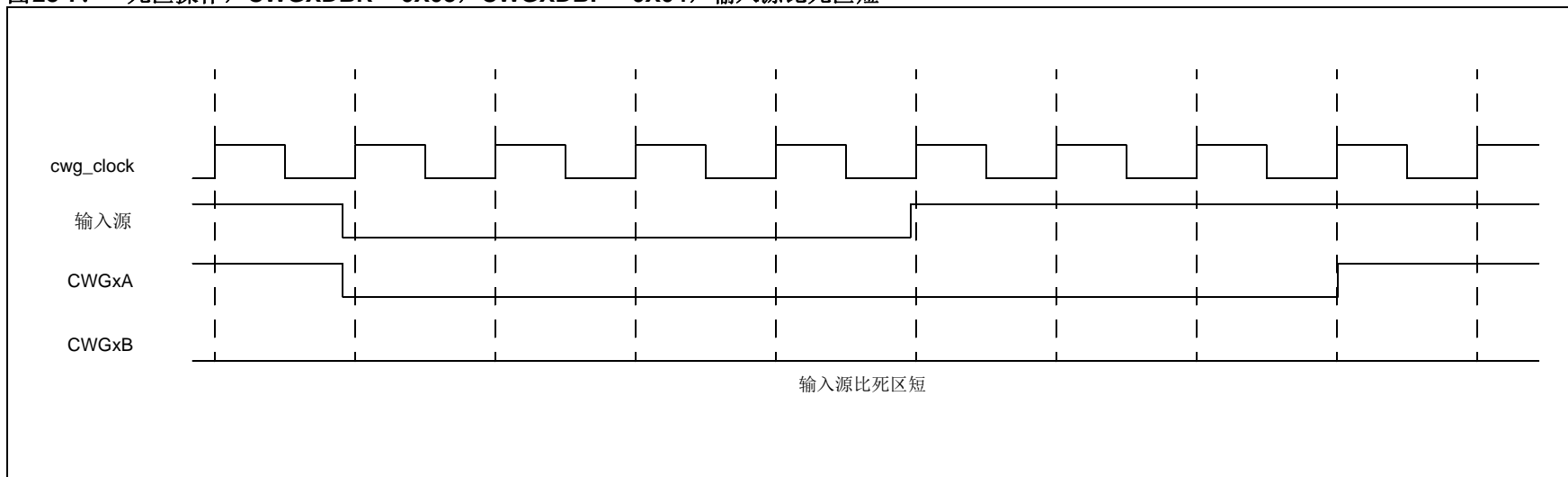


图28-7: 死区操作, CWGXDBR = 0X03, CWGXDBF = 0X04, 输入源比死区短



28.8 死区偏差

当输入源的上升沿和下降沿与CWG时钟异步时，死区延时会产生偏差。最大偏差等于1个CWG时钟周期。更多信息，请参见[公式28-1](#)。

公式28-1： 死区偏差

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{F_{cwg\_clock}}$$

示例：

$$F_{CWG\_CLOCK} = 16\text{ MHz}$$

因此：

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{F_{cwg\_clock}}$$
$$= \frac{1}{16\text{ MHz}}$$
$$= 62.5\text{ ns}$$

图28-8： PWM方向改变的示例

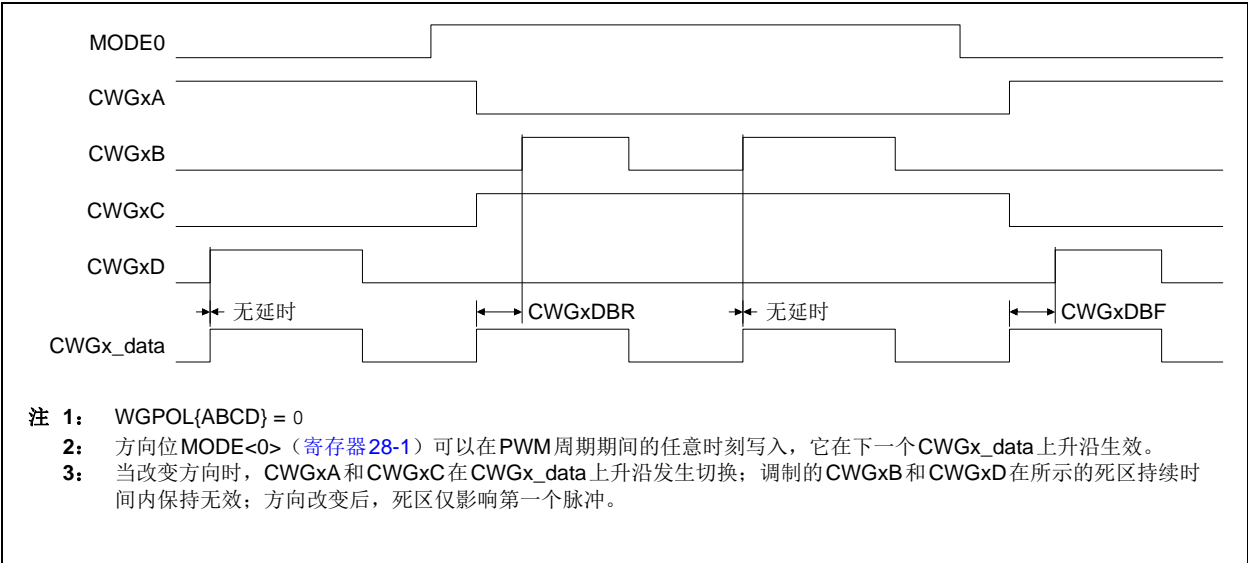
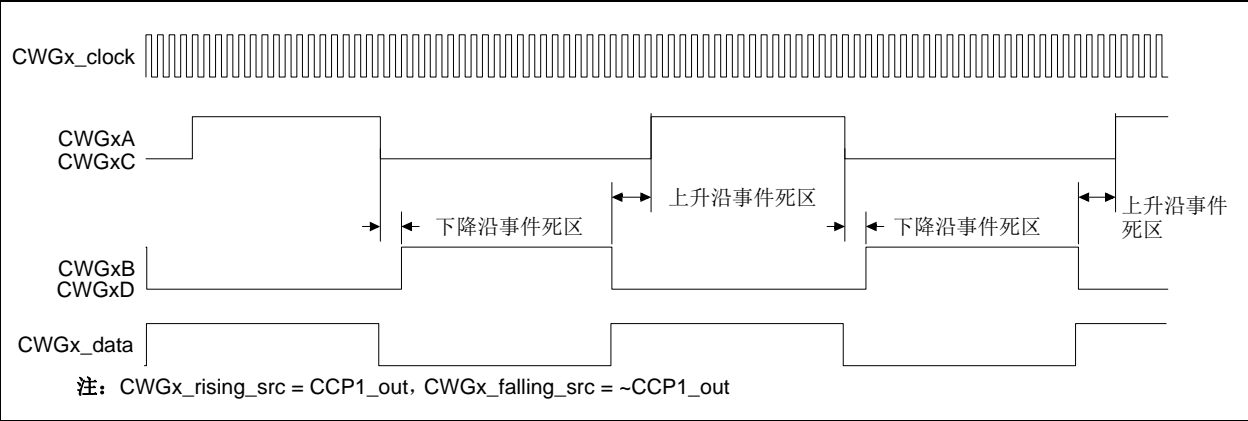


图28-9： CWG半桥模式操作



28.9 CWG转向模式

在转向模式下（MODE = 00x），CWG 允许使用任意 CWGxx 引脚组合来输出调制信号。同一信号可以同时送到多个引脚上，或者也可以送出固定值输出。

当CWGxOCON0相应的STRx位为0时，相应引脚会保持在所定义的电平。当CWGxOCON0相应的STRx位为1时，引脚由输入数据信号驱动。用户可以将输入数据信号分配到1个、2个、3个或全部4个输出引脚。

CWGxCON1 寄存器的POLx 位仅在 STRx = 1 时控制信号的极性。

CWG 自动关断操作也适用于转向模式，如第28.10节“自动关断”中所述。自动关断事件只对STRx = 1 的引脚有影响。

28.9.1 转向同步

更改MODE 位可以设置两种转向模式：同步和异步。

MODE = 000 时，转向事件是异步的，将在写入STRx 的指令结束时（即立刻）发生。在这种情况下，输出引脚的输出信号可能是一个不完整的波形。这可用于立即从引脚上去除信号。

当MODE = 001 时，转向更新是同步的，在输入数据信号下一个上升沿开始时发生。此时，转向开/关输出将总是产生一个完整的波形。

图28-10和图28-11 分别给出了异步和同步转向的时序。

图28-10： 指令结束时发生的转向事件的示例（MODE<2:0> = 000）

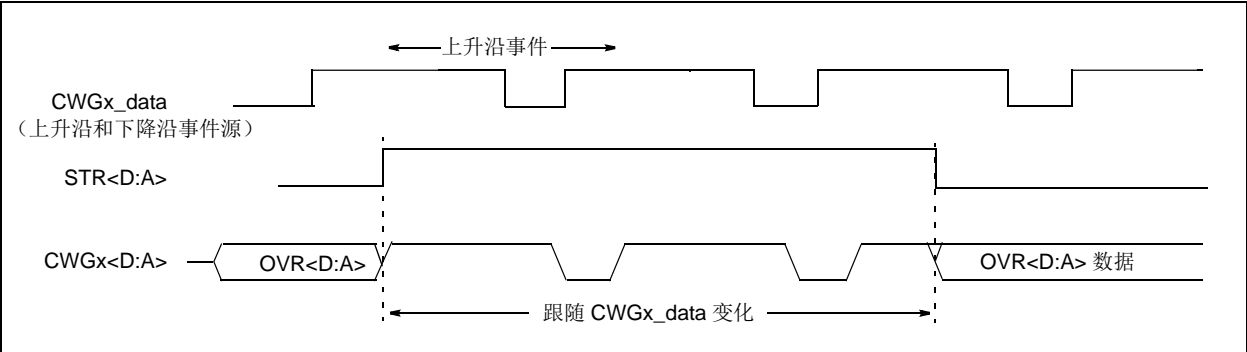
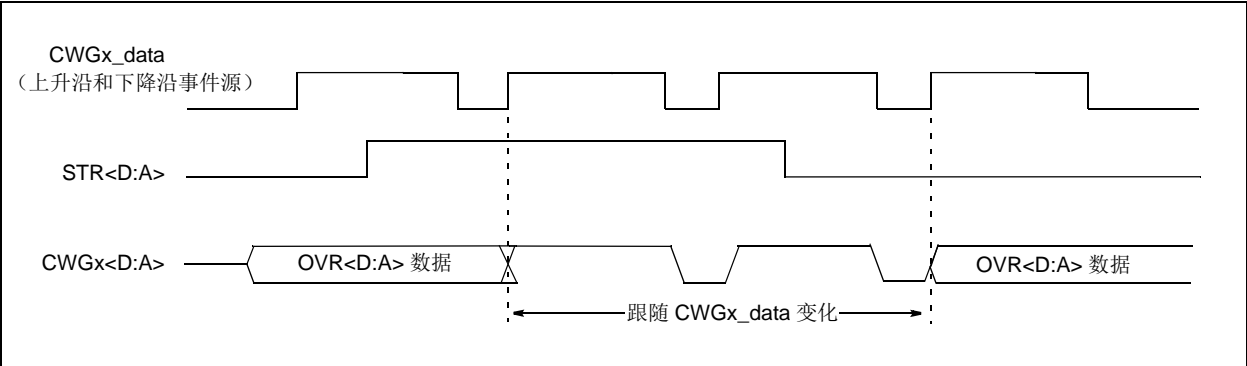


图28-11： 指令开始时发生的转向事件的示例（MODE<2:0> = 001）



28.10 自动关断

自动关断是一种使用特定改写信号立即改写CWG输出电平，从而安全关断电路的方法。关断状态可以自动清除，也可以一直保持，直到用软件清除。图28-12给出了自动关断电路的图示。

28.10.1 关断

关断状态可以通过以下两种方法之一进入：

- 软件生成
- 外部输入

28.10.1.1 由软件产生的关断

将CWGxAS0寄存器的SHUTDOWN位置1可以强制CWG进入关断状态。

在禁止自动重启时，只要SHUTDOWN位置1，就会一直保持关断状态。

在使能自动重启时，SHUTDOWN位会自动清零，并在发生下一个上升沿事件时继续工作。

28.10.2 外部输入源

外部关断输入提供了在出现故障条件时安全地暂停CWG工作的最快办法。当选定的任意关断输入变为有效时，CWG输出会立即变为选定的改写电平，无任何软件延时。可以选择几个输入源来产生关断条件。所有输入源均为低电平有效。这些输入源是：

- 比较器C1\_OUT\_sync
- 比较器C2\_OUT\_sync
- Timer2——TMR2\_postscaled
- Timer4——TMR4\_postscaled
- Timer6——TMR6\_postscaled
- CWGxIN输入引脚

关断输入使用CWGxAS1寄存器（寄存器28-6）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。

28.11 休眠期间的操作

CWG模块独立于系统时钟工作，只要选定的时钟和输入源保持活动状态，它就会继续在休眠期间运行。

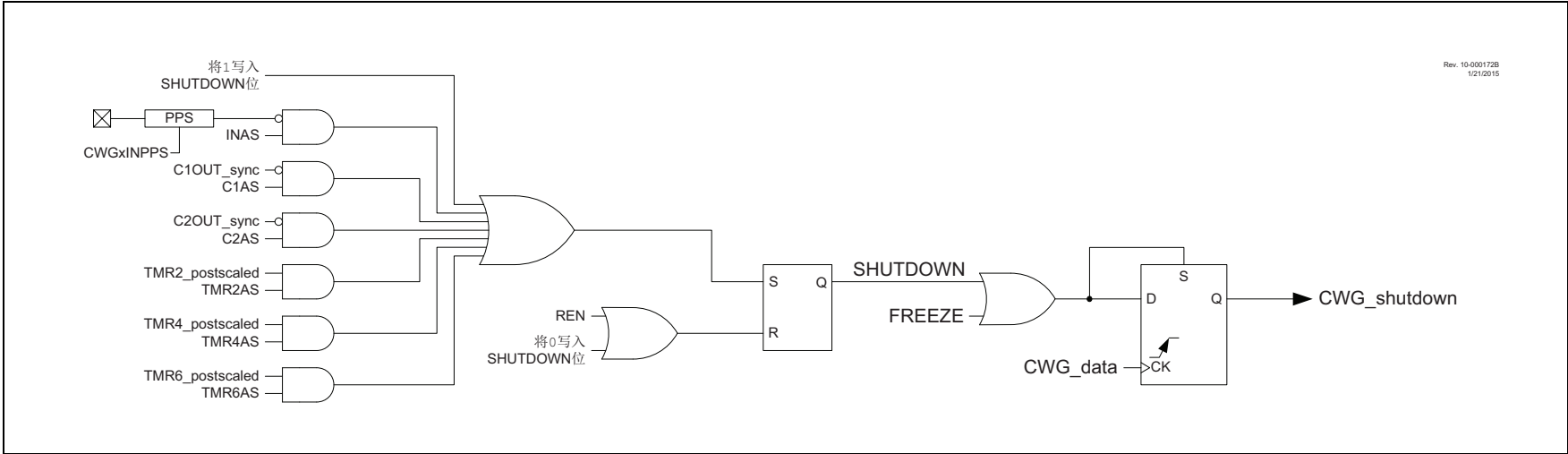
在休眠期间，当满足以下所有条件时，HFINTOSC会保持活动状态：

- 使能了CWG模块
- 输入源处于活动状态
- 选择HFINTOSC作为时钟源，无论选择的系统时钟源如何。

即，如果在CWG使能且输入源处于活动状态时，同时选择HFINTOSC作为系统时钟和CWG时钟源，则CPU在休眠期间将处于空闲状态，而HFINTOSC将保持活动状态，并且CWG会继续工作。这会直接影响休眠模式的电流。

Rev. 10-000172B  
1/21/2015

图28-12: CWG关断框图



## 28.12 配置CWG

以下步骤说明了如何正确配置CWG。

1. 确保对应于应用所需CWG引脚的TRIS控制位置1，从而将这些引脚配置为输入。
2. 清零EN位（如果尚未清零）。
3. 使用MODE位设置所需工作模式。
4. 使用CWGxDBR和CWGxDBF寄存器设置所需的死区时间（如适用于模式）。
5. 在CWGxAS0和CWGxAS1寄存器中设置以下控制。
  - a. 选择所需的关断源。
  - b. 将两个输出改写选择为所需电平（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）。
  - c. 使用CWGxAS1寄存器设置哪些引脚将受自动关断影响。
  - d. 将SHUTDOWN位置1，将REN位清零。
6. 使用CWGxISM寄存器选择所需的输入源。
7. 配置以下控制。
  - a. 使用CWGxCLKCON寄存器选择所需的时钟源。
  - b. 使用CWGxCON1寄存器选择所需的输出极性。
  - c. 将所需输出的输出使能置1。
8. 将EN位置1。
9. 清零对应于所需输出引脚的TRIS控制位，将这些引脚配置为输出。
10. 如果要使用自动重启，则将REN位置1，SHUTDOWN位将会自动清零。否则，通过清零SHUTDOWN位来启动CWG。

### 28.12.1 引脚改写电平

在关断输入为真时驱动到输出引脚上的电平由CWGxAS0寄存器的LSBD和LSAC位控制。LSBD<1:0>控制CWGxB和CWGxD改写电平，LSAC<1:0>控制CWGxA和CWGxC改写电平。控制位逻辑电平对应于处于关断状态时的输出逻辑驱动电平。极性控制不会影响改写电平。

### 28.12.2 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CWGxCON2寄存器的REN位进行选择。[图28-13](#)和[图28-14](#)给出了软件控制重启和自动重启的波形。

#### 28.12.2.1 软件控制重启

当CWGxAS0寄存器的REN位清零时，在自动关断事件之后，必须用软件重启CWG。清除关断状态要求所有选定的关断输入为低电平，否则SHUTDOWN位将保持置1。改写电平将一直保持有效，直到SHUTDOWN位清零之后发生第一个上升沿事件为止。然后，CWG将继续工作。

#### 28.12.2.2 自动重启

当CWGxCON2寄存器的REN位置1时，CWG将从自动关断状态中自动重启。当所有关断源变为低电平时，SHUTDOWN位将自动清零。改写电平将一直保持有效，直到SHUTDOWN位清零之后发生第一个上升沿事件为止。然后，CWG将继续工作。

图28-13: 关断功能, 禁止自动重启 (REN = 0, LSAC = 01, LSB0 = 01)

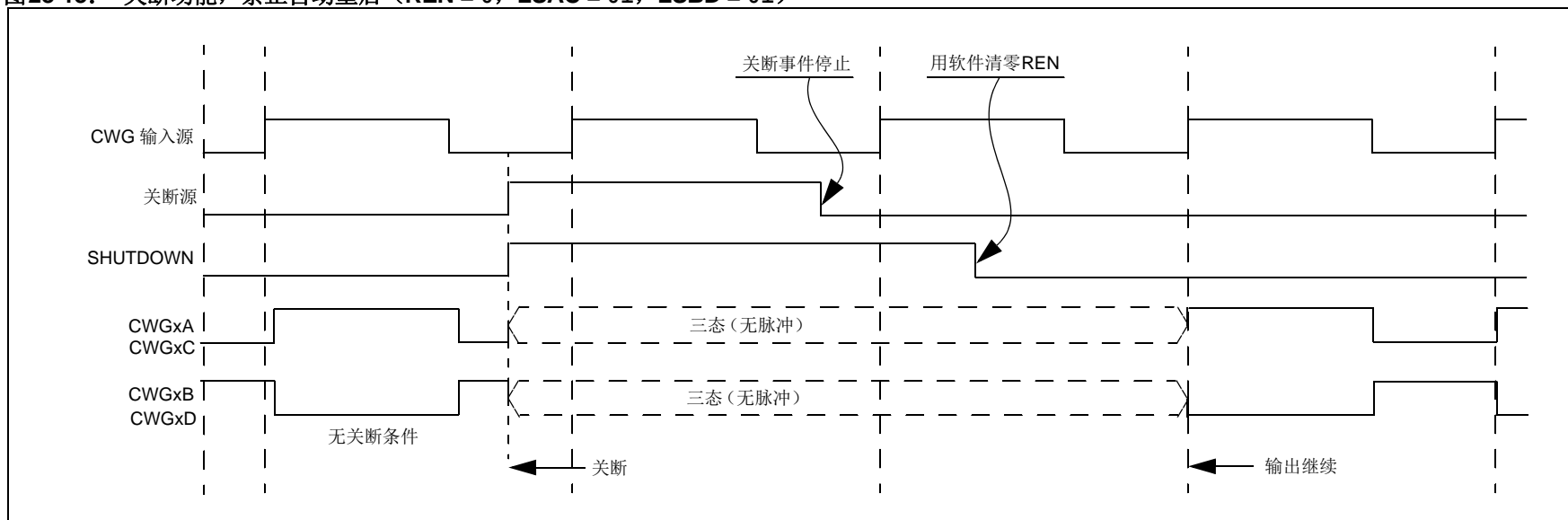
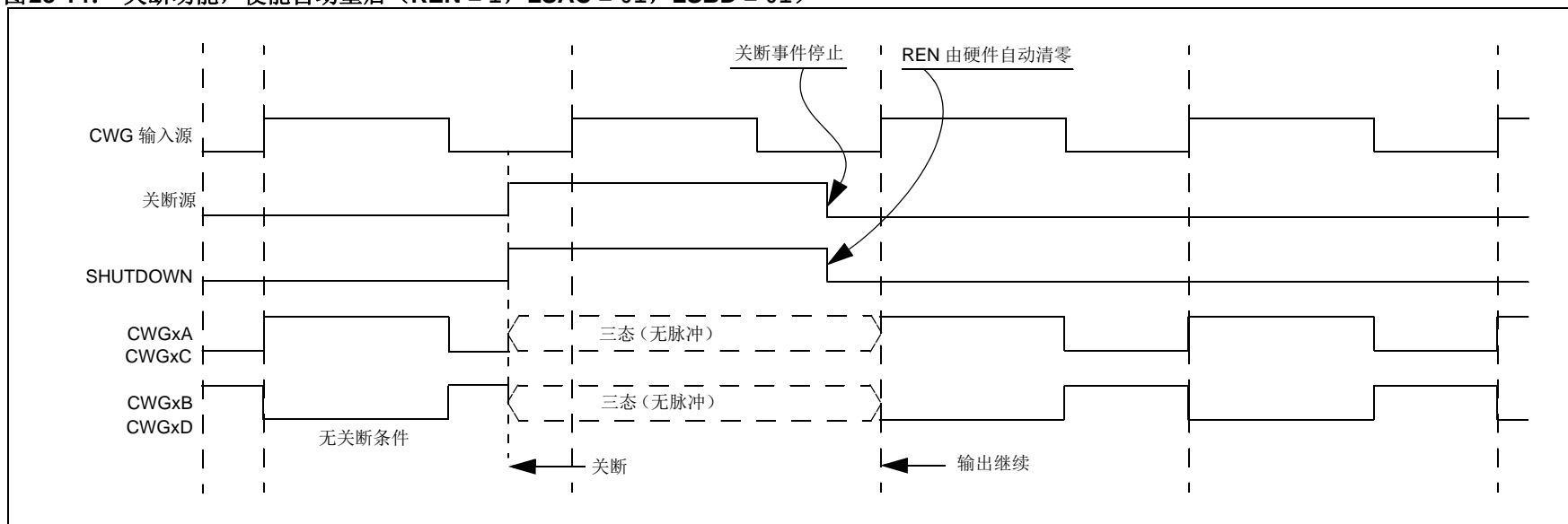


图28-14: 关断功能, 使能自动重启 (REN = 1, LSAC = 01, LSB0 = 01)





28.13 寄存器定义：CWG控制

寄存器 28-1: CWGxCON0: CWGx控制寄存器0

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD <sup>(1)</sup>	—	—	—	MODE<2:0>		
bit 7						bit 0	

<b>图注:</b>		
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	<b>EN:</b> CWGx使能位 1 = 使能模块 0 = 禁止模块
bit 6	<b>LD:</b> CWGx装入缓冲器位 <sup>(1)</sup> 1 = 在发生下一个上升沿/下降沿事件时装入缓冲器 0 = 不装入缓冲器
bit 5-3	<b>未实现:</b> 读为0
bit 2-0	<b>MODE&lt;2:0&gt;:</b> CWGx模式位 111 = 保留 110 = 保留 101 = CWG输出在推挽模式下工作 100 = CWG输出在半桥模式下工作 011 = CWG输出在反向全桥模式下工作 010 = CWG输出在正向全桥模式下工作 001 = CWG输出在同步转向模式下工作 000 = CWG输出在转向模式下工作

注 1: 该位只能在EN = 1后置1，不能在EN置1的同一指令中置1。

# PIC16(L)F1615/9

寄存器 28-2: CWGxCON1: CWGx控制寄存器 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
bit 7							bit 0

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

bit 7-6	未实现: 读为0
bit 5	IN: CWG输入值
bit 4	未实现: 读为0
bit 3	POLD: CWGxD输出极性位 1 = 信号输出极性翻转 0 = 信号输出为正常极性
bit 2	POLC: CWGxC输出极性位 1 = 信号输出极性翻转 0 = 信号输出为正常极性
bit 1	POLB: CWGxB输出极性位 1 = 信号输出极性翻转 0 = 信号输出为正常极性
bit 0	POLA: CWGxA输出极性位 1 = 信号输出极性翻转 0 = 信号输出为正常极性

寄存器 28-3: CWGxDBR: CWGx 上升沿死区计数器寄存器

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
—	—	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器值位

寄存器 28-4: CWGxDBF: CWGx 下降沿死区计数器寄存器

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
—	—	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器值位

# PIC16(L)F1615/9

寄存器 28-5: **CWGxAS0: CWGx 自动关断控制寄存器 0**

R/W/HS-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	U-0	U-0
SHUTDOWN <sup>(1, 2)</sup>	REN	LSBD<1:0>		LSAC<1:0>		—	—
bit 7							bit 0

**图注:**

HC = 硬件清零位

R = 可读位

u = 不变

1 = 置 1

W = 可写位

x = 未知

0 = 清零

HS = 硬件置 1 位

U = 未实现位, 读为 0

-n/n = POR 和 BOR 时的值/所有其他复位时的值

q = 值取决于具体条件

bit 7 **SHUTDOWN:** 自动关断事件状态位<sup>(1, 2)</sup>

1 = 自动关断状态有效

0 = 未发生自动关断事件

bit 6 **REN:** 自动重启使能位

1 = 使能自动重启

0 = 禁止自动重启

bit 5-4 **LSBD<1:0>:** CWGxB 和 CWGxD 自动关断状态控制位

11 = 存在自动关断事件时, 将逻辑 1 放置在 CWGxB/D 上

10 = 存在自动关断事件时, 将逻辑 0 放置在 CWGxB/D 上

01 = 存在自动关断事件时, CWGxB/D 上的引脚处于三态

00 = 在所需的死区时间间隔之后, 将引脚的无效状态 (包括极性) 放置在 CWGxB/D 上

bit 3-2 **LSAC<1:0>:** CWGxA 和 CWGxC 自动关断状态控制位

11 = 存在自动关断事件时, 将逻辑 1 放置在 CWGxA/C 上

10 = 存在自动关断事件时, 将逻辑 0 放置在 CWGxA/C 上

01 = 存在自动关断事件时, CWGxA/C 上的引脚处于三态

00 = 在所需的死区时间间隔之后, 将引脚的无效状态 (包括极性) 放置在 CWGxA/C 上

bit 1-0 **未实现:** 读为 0

**注 1:** 在 EN = 0 (CWGxCON0 寄存器) 时, 可以写入该位, 将输出置为关断配置。

**2:** 输出将一直保持在自动关断状态, 直到该位清零后出现输入信号的下一个上升沿为止。

寄存器 28-6: CWGxAS1: CWGx 自动关断控制寄存器 1

U-1	R/W-0/0	R/W-0/0	R/W-0/0	U-1	R/W-0/0	R/W-0/0	R/W-0/0
—	TMR6AS	TMR4AS	TMR2AS	—	C2AS <sup>(1)</sup>	C1AS	INAS
bit 7				bit 0			

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

bit 7	未实现: 读为1
bit 6	<b>TMR6AS:</b> TMR6 后分频输出位 1 = 使能 TMR6 后分频关断 0 = 禁止 TMR6 后分频关断
bit 5	<b>TMR4AS:</b> TMR4 后分频输出位 1 = 使能 TMR4 后分频关断 0 = 禁止 TMR4 后分频关断
bit 4	<b>TMR2AS:</b> TMR2 后分频输出位 1 = 使能 TMR2 后分频关断 0 = 禁止 TMR2 后分频关断
bit 3	未实现: 读为1
bit 2	<b>C2AS:</b> 比较器 C2 输出位 1 = 使能 C2 输出关断 0 = 禁止 C2 输出关断
bit 1	<b>C1AS:</b> 比较器 C1 输出位 1 = 使能 C1 输出关断 0 = 禁止 C1 输出关断
bit 0	<b>INAS:</b> CWGx 输入引脚位 1 = 使能 CWGxIN 输入引脚关断 0 = 禁止 CWGxIN 输入引脚关断

# PIC16(L)F1615/9

寄存器 28-7: CWGxOCON0: CWGx 转向控制寄存器<sup>(1)</sup>

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OVRD	OVRC	OVRB	OVRA	STRD <sup>(2)</sup>	STRC <sup>(2)</sup>	STRB <sup>(2)</sup>	STRA <sup>(2)</sup>
bit 7							bit 0

图注:		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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	<b>OVRD:</b> 转向数据 D 位
bit 6	<b>OVRC:</b> 转向数据 C 位
bit 5	<b>OVRB:</b> 转向数据 B 位
bit 4	<b>OVRA:</b> 转向数据 A 位
bit 3	<b>STRD:</b> 转向使能 D 位 <sup>(2)</sup> 1 = CWGxD 输出具有 CWGx_data 波形, 其极性由 POLD 位控制 0 = CWGxD 输出指定为 OVRD 位的值
bit 2	<b>STRC:</b> 转向使能 C 位 <sup>(2)</sup> 1 = CWGxC 输出具有 CWGx_data 波形, 其极性由 POLC 位控制 0 = CWGxC 输出指定为 OVRC 位的值
bit 1	<b>STRB:</b> 转向使能 B 位 <sup>(2)</sup> 1 = CWGxB 输出具有 CWGx_data 波形, 其极性由 POLB 位控制 0 = CWGxB 输出指定为 OVRB 位的值
bit 0	<b>STRA:</b> 转向使能 A 位 <sup>(2)</sup> 1 = CWGxA 输出具有 CWGx_data 波形, 其极性由 POLA 位控制 0 = CWGxA 输出指定为 OVRA 位的值

注 1: 该寄存器中的位仅在 MODE<2:0> = 00x 时适用。  
2: MODE<2:0> = 001 时, 该位实际上是双重缓冲的。

寄存器 28-8: CWGxCLKCON: CWGx时钟选择控制寄存器

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

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

bit 7-1 未实现: 读为0  
bit 0 CS: CWGx时钟选择位  
1 = 选择HFINTOSC 16 MHz  
0 = 选择FOSC

寄存器 28-9: CWGxISM: CWGx输入选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	IS<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 IS<3:0>: CWGx输入选择位  
1111 = 保留。不连接任何通道。  
•  
•  
•  
1011 = 保留。不连接任何通道。  
1010 = PWM4\_out  
1001 = PWM3\_out  
1000 = LC4\_out  
0111 = LC3\_out  
0110 = LC2\_out  
0101 = LC1\_out  
0100 = CCP2\_out  
0011 = CCP1\_out  
0010 = C2\_OUT\_sync  
0001 = C1\_OUT\_sync  
0000 = CWGxIN引脚

表28-2: 与CWG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
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	392
CWG1AS1	—	TMR6AS	TMR4AS	TMR2AS	—	C2AS	C1AS	INAS	393
CWG1CLKCON	—	—	—	—	—	—	—	CS	395
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			394
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	390
CWG1DBF	—	—	DBF<5:0>						391
CWG1DBR	—	—	DBR<5:0>						391
CWG1ISM	—	—	—	—	IS<3:0>				395
CWG1OCON0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	394

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



注:

29.0 可配置逻辑单元（CLC）

可配置逻辑单元（CLCx）提供可超越软件执行速度限制而工作的可编程逻辑。该逻辑单元最多可接收16个输入信号，并通过使用可配置门将16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

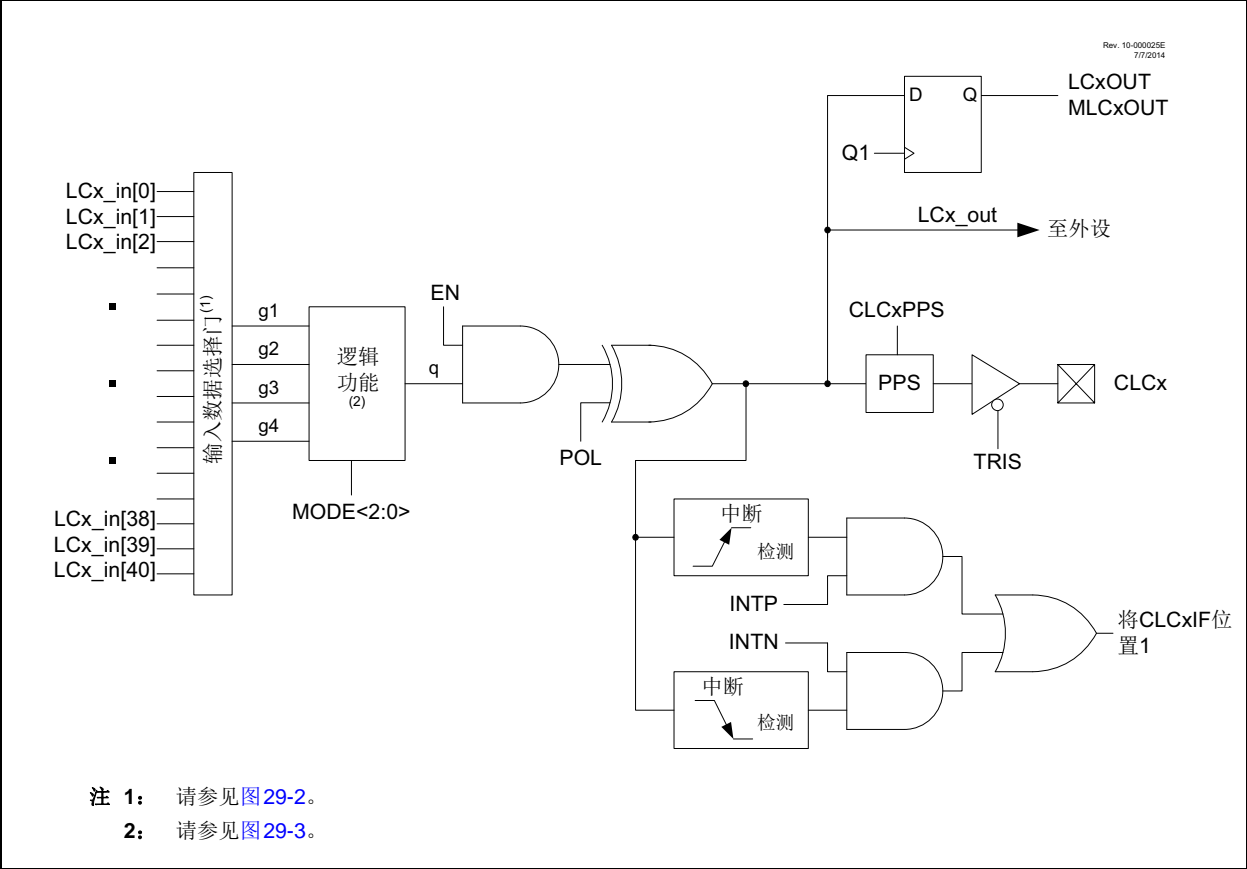
可将输出内部连接到外设和输出引脚。

关于说明通过CLCx的信号流的简化图，请参见图29-1。

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 29-1： 可配置逻辑单元框图



## 29.1 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
- 数据门控
- 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。这样可以在程序执行期间即时执行逻辑重新配置。

### 29.1.1 数据选择

有 41 个信号可用作可配置逻辑的输入。使用 4 个 41 输入多路开关来选择要传递到下一级的输入。这允许将任何可能的输入信号用作 CLC 模块四个输入的任意一个。

数据选择通过图 29-2 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码的输入名称来表示。

表 29-1 列出了每个 CLC 模块中的通用编码的输入名称和实际信号。标记为 CLCxSELY 的列指的是与 4 个多路开关相关的 4 个寄存器（CLCxSEL0 至 CLCxSEL3）中的任意一个的值。

每个多路开关的数据输入通过其各自的 CLCxSELY 寄存器来选择。

**注：** 数据选择在上电时是未定义的。

表 29-1: CLCx 数据输入选择

数据输入	CLCxSELY	CLC 输入信号
LCx_in[0]	000000	CLCIN0
LCx_in[1]	000001	CLCIN1
LCx_in[2]	000010	CLCIN2
LCx_in[3]	000011	CLCIN3
LCx_in[4]	000100	LC1_out
LCx_in[5]	000101	LC2_out
LCx_in[6]	000110	LC3_out
LCx_in[7]	000111	LC4_out
LCx_in[8]	001000	C1OUT_sync
LCx_in[9]	001001	C2OUT_sync
LCx_in[10]	001010	CWGOUTA
LCx_in[11]	001011	CWGOUTB
LCx_in[12]	001100	CCP1_out
LCx_in[13]	001101	CCP2_out
LCx_in[14]	001110	PWM3_out
LCx_in[15]	001111	PWM4_out
LCx_in[16]	010000	AT1_cmp1
LCx_in[17]	010001	AT1_cmp2
LCx_in[18]	010010	AT1_cmp3
LCx_in[19]	010011	SMT1_match
LCx_in[20]	010100	SMT2_match
LCx_in[21]	010101	ZCD1_output
LCx_in[22]	010110	TMR0_overflow
LCx_in[23]	010111	TMR1_overflow
LCx_in[24]	011000	TMR2_postscaled
LCx_in[25]	011001	TMR3_overflow
LCx_in[26]	011010	TMR4_postscaled
LCx_in[27]	011011	TMR5_overflow
LCx_in[28]	011100	TMR6_postscaled
LCx_in[29]	011101	IOC_interrupt
LCx_in[30]	011110	ADC_rc
LCx_in[31]	011111	LFINTOSC
LCx_in[32]	100000	HFINTOSC
LCx_in[33]	100001	FOSC
LCx_in[34]	100010	AT1_missedpulse
LCx_in[35]	100011	AT1_perclk
LCx_in[36]	100100	AT1_phsclk
LCx_in[37]	100101	TX
LCx_in[38]	100110	RX
LCx_in[39]	100111	SCK
LCx_in[40]	101000	SDO

29.2 数据门控

来自输入多路开关的输出将通过数据门控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，数据门控是未定义的。

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。

门控实际上是一个1至4的输入AND/NAND/OR/NOR门。如果将每个输入和输出进行反相，则该门的作用是对所有已使能数据输入进行或运算。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与运算。

表29-2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29-2: 数据门控逻辑

CLCxGLS0	LCxG1POL	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（瞬态电流引起的脉冲）。如果通道的输出必须为0或1，则建议将所有门位设置为0，并使用门极性位来设置所需的电平。

数据门控使用如下逻辑门选择寄存器进行配置：

- 门1: CLCxGLS0（寄存器29-7）
- 门2: CLCxGLS1（寄存器29-8）
- 门3: CLCxGLS2（寄存器29-9）
- 门4: CLCxGLS3（寄存器29-10）

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29-2右侧给出了数据门控的图示。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。

29.2.1 逻辑功能

有以下8种可用的逻辑功能：

- 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29-3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。

29.2.2 输出极性

可配置逻辑单元中的最后一级是输出极性。将CLCxCON寄存器的LCxPOL位置1时，来自逻辑级的输出信号会被反相。如果在允许中断时改变极性会在输出转换时产生中断。

## 29.2.3 CLCx 设置步骤

在设置CLCx时，应遵循以下步骤：

- 通过清零LCxEN位来禁止CLCx。
- 使用CLCxSEL0、CLCxSEL1、CLCxSEL2和CLCxSEL3寄存器选择所需的输入（见表29-1）。
- 清零所有关联的ANSEL位。
- 将与输入关联的所有TRIS位置1。
- 将与输出关联的所有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LCxPOLy位选择门输出极性。
- 使用CLCxCON寄存器的LCxMODE<2:0>位选择所需的逻辑功能。
- 使用CLCxPOL寄存器的LCxPOL位选择所需的逻辑输出极性。（该步骤可与前面的门输出极性步骤结合）。
- 如果驱动某个器件，则设置所需的引脚PPS控制寄存器，并另外清零对应于该输出的TRIS位。
- 如果需要中断，则配置以下位：
  - 上升沿事件时，将CLCxCON寄存器中的LCxINTP位置1。
  - 下降沿事件时，将CLCxCON寄存器中的LCxINTN位置1。
  - 将关联的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LCxEN位置1来使能CLCx。

## 29.3 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，且其关联的使能位置1时，关联PIR寄存器的CLCxIF位会置1。LCxINTP位用于允许上升沿中断，LCxINTN位用于允许下降沿中断。它们都位于CLCxCON寄存器中。

要完全允许中断，需要将以下位置1：

- CLCxCON寄存器的LCxON位
- 关联PIE寄存器的CLCxIE位
- CLCxCON寄存器的LCxINTP位（对于上升沿检测）
- CLCxCON寄存器的LCxINTN位（对于下降沿检测）
- INTCON寄存器的PEIE和GIE位

作为中断服务的一部分，必须用软件将关联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。

## 29.4 输出镜像副本

所有CLCxCON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CLCxOUT位而导致读取差错。

## 29.5 复位的影响

发生复位后，CLCxCON寄存器会清零。所有其他选择和门控值保持不变。

## 29.6 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 29-2： 输入数据选择和门控

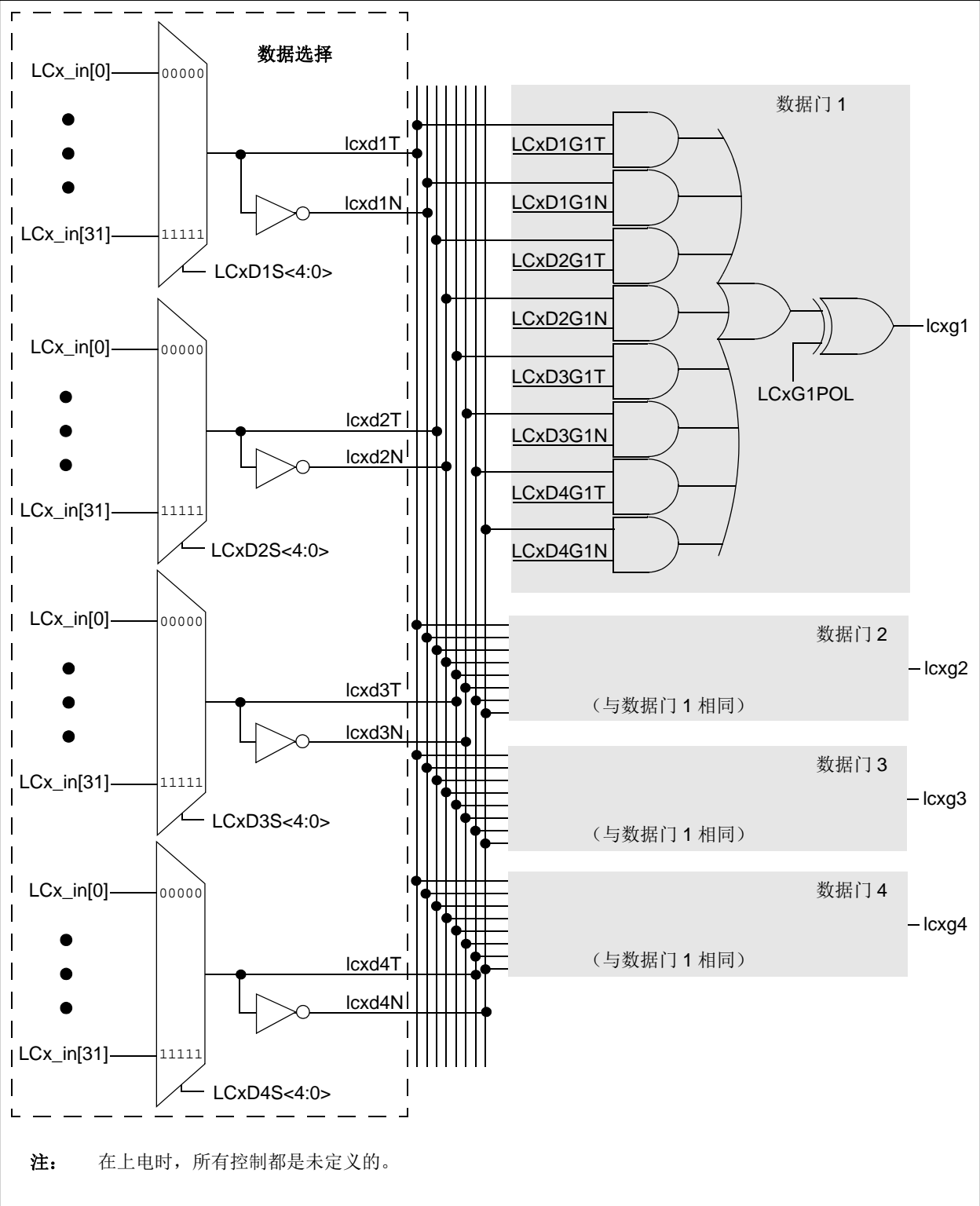
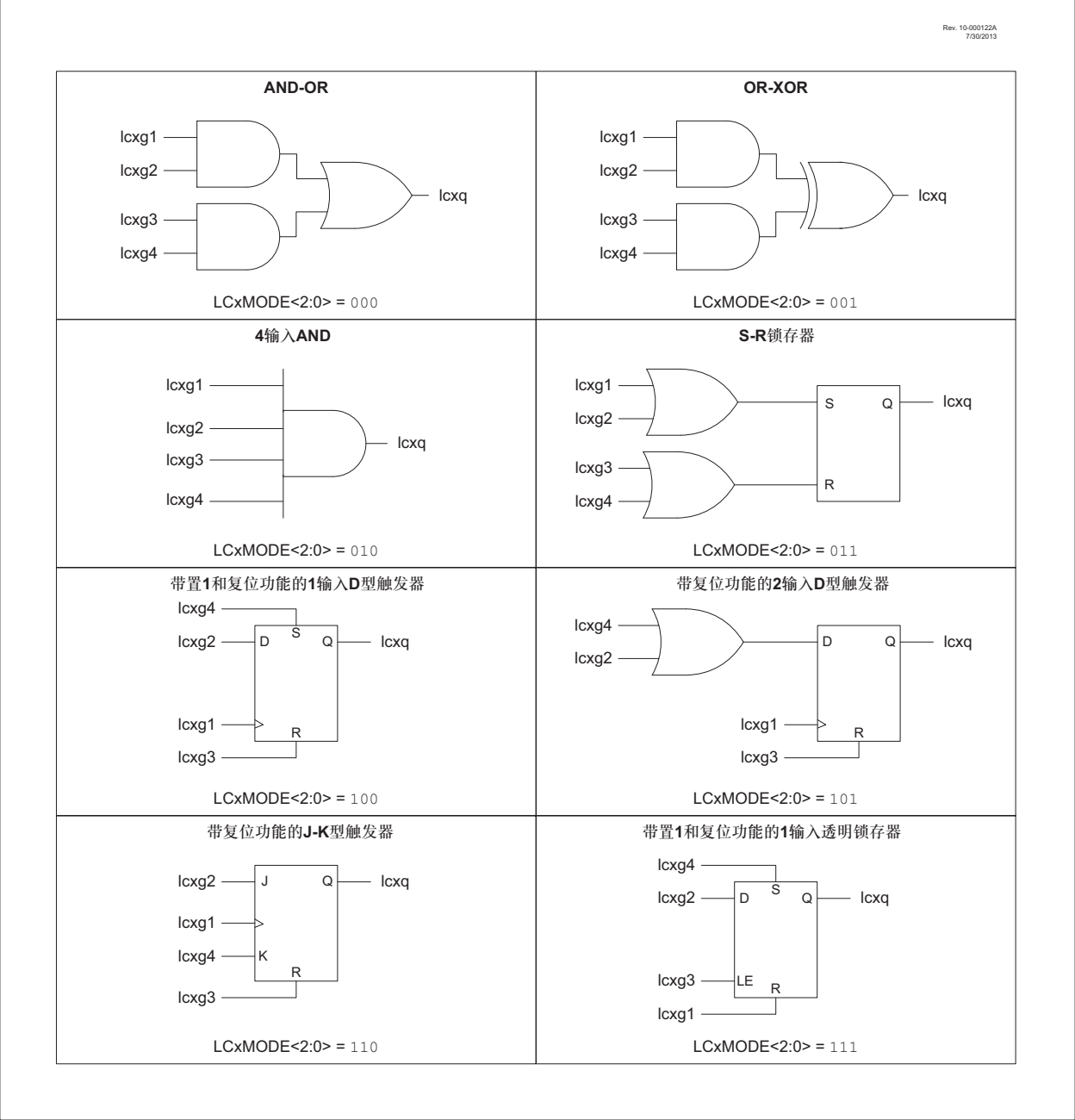


图 29-3： 可编程逻辑功能



# PIC16(L)F1615/9

## 29.7 寄存器定义：CLC控制

寄存器 29-1: CLCxCON: 可配置逻辑单元控制寄存器

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	LCxOE	LCxOUT	LCxINTP	LCxINTN	LCxMODE<2:0>		
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

- bit 7      **LCxEN:** 可配置逻辑单元使能位  
1 = 使能可配置逻辑单元, 并混合输入信号  
0 = 禁止可配置逻辑单元, 并输出逻辑0
- bit 6      **LCxOE:** 可配置逻辑单元输出使能位  
1 = 使能可配置逻辑单元端口引脚输出  
0 = 禁止可配置逻辑单元端口引脚输出
- bit 5      **LCxOUT:** 可配置逻辑单元数据输出位  
只读: 经过LCxPOL之后的逻辑单元输出数据; 从lcx\_out线采样。
- bit 4      **LCxINTP:** 可配置逻辑单元上升边沿中断允许位  
1 = CLCxIF将在lcx\_out上出现上升沿时置1  
0 = CLCxIF不会置1
- bit 3      **LCxINTN:** 可配置逻辑单元下降沿中断允许位  
1 = CLCxIF将在lcx\_out上出现下降沿时置1  
0 = CLCxIF不会置1
- bit 2-0    **LCxMODE<2:0>:** 可配置逻辑单元功能模式位  
111 = 单元是带置1和复位功能的1输入透明锁存器  
110 = 单元是带复位功能的J-K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



寄存器 29-2: CLCxPOL: 信号极性控制寄存器

R/W-0/0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL
bit 7							bit 0

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

bit 7	<b>LCxPOL:</b> LCOUNT 极性控制位 1 = 逻辑单元的输出反相 0 = 逻辑单元的输出同相
bit 6-4	<b>未实现:</b> 读为0
bit 3	<b>LCxG4POL:</b> 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的输出同相
bit 2	<b>LCxG3POL:</b> 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的输出同相
bit 1	<b>LCxG2POL:</b> 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的输出同相
bit 0	<b>LCxG1POL:</b> 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的输出同相

# PIC16(L)F1615/9

寄存器 29-3: CLCxSEL0: 多路开关数据0选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD1S<5:0>					
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-0 LCxD1S<5:0>: 输入数据1选择控制位  
关于与输入关联的信号名称, 请参见表29-1。

寄存器 29-4: CLCxSEL1: 多路开关数据1选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD2S<5:0>					
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-0 LCxD2S<5:0>: 输入数据2选择控制位  
关于与输入关联的信号名称, 请参见表29-1。

寄存器 29-5: CLCxSEL2: 多路开关数据2选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD3S<5:0>					
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和BOR时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-6 未实现: 读为0

bit 5-0 LCxD3S<5:0>: 输入数据3选择控制位  
关于与输入关联的信号名称, 请参见表29-1。

寄存器 29-6: CLCxSEL3: 多路开关数据3选择寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	LCxD4S<5:0>					
bit 7							bit 0

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

bit 7-6        未实现: 读为0

bit 5-0        **LCxD4S<5:0>**: 输入数据4选择控制位  
关于与输入关联的信号名称, 请参见表29-1。

# PIC16(L)F1615/9

寄存器 29-7: CLCxGLS0: 门1逻辑选择寄存器

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
LCxG1D4T	LCxG1D4N	LCxG1D3T	LCxG1D3N	LCxG1D2T	LCxG1D2N	LCxG1D1T	LCxG1D1N
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7	<b>LCxG1D4T:</b> 门1数据4（同相）位 1 = lcx4T通过门输入到lcxg1 0 = lcx4T未通过门输入到lcxg1
bit 6	<b>LCxG1D4N:</b> 门1数据4取反（反相）位 1 = lcx4N通过门输入到lcxg1 0 = lcx4N未通过门输入到lcxg1
bit 5	<b>LCxG1D3T:</b> 门1数据3（同相）位 1 = lcx3T通过门输入到lcxg1 0 = lcx3T未通过门输入到lcxg1
bit 4	<b>LCxG1D3N:</b> 门1数据3取反（反相）位 1 = lcx3N通过门输入到lcxg1 0 = lcx3N未通过门输入到lcxg1
bit 3	<b>LCxG1D2T:</b> 门1数据2（同相）位 1 = lcx2T通过门输入到lcxg1 0 = lcx2T未通过门输入到lcxg1
bit 2	<b>LCxG1D2N:</b> 门1数据2取反（反相）位 1 = lcx2N通过门输入到lcxg1 0 = lcx2N未通过门输入到lcxg1
bit 1	<b>LCxG1D1T:</b> 门1数据1（同相）位 1 = lcx1T通过门输入到lcxg1 0 = lcx1T未通过门输入到lcxg1
bit 0	<b>LCxG1D1N:</b> 门1数据1取反（反相）位 1 = lcx1N通过门输入到lcxg1 0 = lcx1N未通过门输入到lcxg1

## 寄存器 29-8: CLCxGLS1: 门2逻辑选择寄存器

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
LCxG2D4T	LCxG2D4N	LCxG2D3T	LCxG2D3N	LCxG2D2T	LCxG2D2N	LCxG2D1T	LCxG2D1N
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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      **LCxG2D4T:** 门2数据4 (同相) 位  
1 = lcx4T 通过门输入到 lcxg2  
0 = lcx4T 未通过门输入到 lcxg2
- bit 6      **LCxG2D4N:** 门2数据4取反 (反相) 位  
1 = lcx4N 通过门输入到 lcxg2  
0 = lcx4N 未通过门输入到 lcxg2
- bit 5      **LCxG2D3T:** 门2数据3 (同相) 位  
1 = lcx3T 通过门输入到 lcxg2  
0 = lcx3T 未通过门输入到 lcxg2
- bit 4      **LCxG2D3N:** 门2数据3取反 (反相) 位  
1 = lcx3N 通过门输入到 lcxg2  
0 = lcx3N 未通过门输入到 lcxg2
- bit 3      **LCxG2D2T:** 门2数据2 (同相) 位  
1 = lcx2T 通过门输入到 lcxg2  
0 = lcx2T 未通过门输入到 lcxg2
- bit 2      **LCxG2D2N:** 门2数据2取反 (反相) 位  
1 = lcx2N 通过门输入到 lcxg2  
0 = lcx2N 未通过门输入到 lcxg2
- bit 1      **LCxG2D1T:** 门2数据1 (同相) 位  
1 = lcx1T 通过门输入到 lcxg2  
0 = lcx1T 未通过门输入到 lcxg2
- bit 0      **LCxG2D1N:** 门2数据1取反 (反相) 位  
1 = lcx1N 通过门输入到 lcxg2  
0 = lcx1N 未通过门输入到 lcxg2

# PIC16(L)F1615/9

寄存器 29-9: **CLCxGLS2: 门3逻辑选择寄存器**

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
LCxG3D4T	LCxG3D4N	LCxG3D3T	LCxG3D3N	LCxG3D2T	LCxG3D2N	LCxG3D1T	LCxG3D1N
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7	<b>LCxG3D4T:</b> 门3数据4（同相）位 1 = lcx4T通过门输入到lcxg3 0 = lcx4T未通过门输入到lcxg3
bit 6	<b>LCxG3D4N:</b> 门3数据4取反（反相）位 1 = lcx4N通过门输入到lcxg3 0 = lcx4N未通过门输入到lcxg3
bit 5	<b>LCxG3D3T:</b> 门3数据3（同相）位 1 = lcx3T通过门输入到lcxg3 0 = lcx3T未通过门输入到lcxg3
bit 4	<b>LCxG3D3N:</b> 门3数据3取反（反相）位 1 = lcx3N通过门输入到lcxg3 0 = lcx3N未通过门输入到lcxg3
bit 3	<b>LCxG3D2T:</b> 门3数据2（同相）位 1 = lcx2T通过门输入到lcxg3 0 = lcx2T未通过门输入到lcxg3
bit 2	<b>LCxG3D2N:</b> 门3数据2取反（反相）位 1 = lcx2N通过门输入到lcxg3 0 = lcx2N未通过门输入到lcxg3
bit 1	<b>LCxG3D1T:</b> 门3数据1（同相）位 1 = lcx1T通过门输入到lcxg3 0 = lcx1T未通过门输入到lcxg3
bit 0	<b>LCxG3D1N:</b> 门3数据1取反（反相）位 1 = lcx1N通过门输入到lcxg3 0 = lcx1N未通过门输入到lcxg3

寄存器 29-10: CLCxGLS3: 门4逻辑选择寄存器

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
LCxG4D4T	LCxG4D4N	LCxG4D3T	LCxG4D3N	LCxG4D2T	LCxG4D2N	LCxG4D1T	LCxG4D1N
bit 7							bit 0

图注:		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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	<b>LCxG4D4T:</b> 门4数据4 (同相) 位 1 = lcx4T通过门输入到lcxg4 0 = lcx4T未通过门输入到lcxg4
bit 6	<b>LCxG4D4N:</b> 门4数据4取反 (反相) 位 1 = lcx4N通过门输入到lcxg4 0 = lcx4N未通过门输入到lcxg4
bit 5	<b>LCxG4D3T:</b> 门4数据3 (同相) 位 1 = lcx3T通过门输入到lcxg4 0 = lcx3T未通过门输入到lcxg4
bit 4	<b>LCxG4D3N:</b> 门4数据3取反 (反相) 位 1 = lcx3N通过门输入到lcxg4 0 = lcx3N未通过门输入到lcxg4
bit 3	<b>LCxG4D2T:</b> 门4数据2 (同相) 位 1 = lcx2T通过门输入到lcxg4 0 = lcx2T未通过门输入到lcxg4
bit 2	<b>LCxG4D2N:</b> 门4数据2取反 (反相) 位 1 = lcx2N通过门输入到lcxg4 0 = lcx2N未通过门输入到lcxg4
bit 1	<b>LCxG4D1T:</b> 门4数据1 (同相) 位 1 = lcx1T通过门输入到lcxg4 0 = lcx1T未通过门输入到lcxg4
bit 0	<b>LCxG4D1N:</b> 门4数据1取反 (反相) 位 1 = lcx1N通过门输入到lcxg4 0 = lcx1N未通过门输入到lcxg4

# PIC16(L)F1615/9

寄存器 29-11: CLCDATA: CLC 数据输出

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-4 未实现: 读为0

bit 3 **MLC4OUT**: LC4OUT 的镜像副本位

bit 2 **MLC3OUT**: LC3OUT 的镜像副本位

bit 1 **MLC2OUT**: LC2OUT 的镜像副本位

bit 0 **MLC1OUT**: LC1OUT 的镜像副本位

表 29-3: 与 CLCx 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	160
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	167
ANSELC	ANSC7 <sup>(2)</sup>	ANSC6 <sup>(2)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	174
CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			404
CLCDATA	—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT	412
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	408
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	409
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	410
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	411
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	405
CLC1SEL0	—	—	LC1D1S<5:0>						406
CLC1SEL1	—	—	LC1D2S<5:0>						406
CLC1SEL2	—	—	LC1D3S<5:0>						406
CLC1SEL3	—	—	LC1D4S<5:0>						407
CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			404
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	408
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	409
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	410
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	411
CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	405
CLC2SEL0	—	—	LC2D1S<5:0>						406
CLC2SEL1	—	—	LC2D2S<5:0>						406
CLC2SEL2	—	—	LC2D3S<5:0>						406
CLC2SEL3	—	—	LC2D4S<5:0>						407
CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			404
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	408

图注: — = 未实现, 读为0。CLC 模块不使用阴影单元。

注 1: 未实现, 读为1。

2: 仅限 PIC16(L)F1619。



**表29-3: 与CLCx相关的寄存器汇总（续）**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	<a href="#">409</a>
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	<a href="#">410</a>
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	<a href="#">411</a>
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	<a href="#">405</a>
CLC3SEL0	—	—	LC3D1S<5:0>						<a href="#">406</a>
CLC3SEL1	—	—	LC3D2S<5:0>						<a href="#">406</a>
CLC3SEL2	—	—	LC3D3S<5:0>						<a href="#">406</a>
CLC3SEL3	—	—	LC3D4S<5:0>						<a href="#">407</a>
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			<a href="#">404</a>
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	<a href="#">408</a>
CLC4GLS1	LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	<a href="#">409</a>
CLC4GLS2	LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	<a href="#">410</a>
CLC4GLS3	LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	<a href="#">411</a>
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	<a href="#">405</a>
CLC4SEL0	—	—	LC4D1S<5:0>						<a href="#">406</a>
CLC4SEL1	—	—	LC4D2S<5:0>						<a href="#">406</a>
CLC4SEL2	—	—	LC4D3S<5:0>						<a href="#">406</a>
CLC4SEL3	—	—	LC4D4S<5:0>						<a href="#">407</a>
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	<a href="#">105</a>
PIE3	—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	<a href="#">108</a>
PIR3	—	—	CWGIF	ZCDIF	CLC4IF	CLC3IF	CLC2IF	CLC1IF	<a href="#">113</a>
TRISA	—	—	TRISA5	TRISA4	— <sup>(1)</sup>	TRISA2	TRISA1	TRISA0	<a href="#">159</a>
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	<a href="#">166</a>
TRISC	TRISC7 <sup>(2)</sup>	TRISC6 <sup>(2)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	<a href="#">173</a>

**图注:** — = 未实现，读为0。CLC模块不使用阴影单元。

**注 1:** 未实现，读为1。

**2:** 仅限PIC16(L)F1619。

## 30.0 信号测量定时器（SMT）

SMT是一个具有高级时钟和门控逻辑的24位计数器，它可以配置为测量各种数字信号参数，例如脉冲宽度、频率和占空比，以及两个信号上边沿之间的时间差。

SMT的特性包括：

- 24位定时器/计数器
  - 3个8位寄存器（SMTxL/H/U）
  - 可读写
  - 可选的16位工作模式
- 2个24位测量捕捉寄存器
- 1个24位周期匹配寄存器
- 多模式操作，包括相对时间测量
- 发生周期匹配时中断
- 多个时钟、门控和信号源
- 采集完成时产生中断
- 可读取当前输入值

<p><b>注：</b> 这些器件实现了两个SMT模块。对SMTx的所有引用均适用于SMT1和SMT2。</p>
--



## 30.1 SMT工作原理

该模块的核心是一个与复杂数据采集前端相结合的24位计数器SMTxTMR。根据所选的工作模式，SMT可以执行表30-1汇总的各种测量。

### 30.1.1 时钟源

SMT可用的时钟源包括：

- FOSC
- FOSC/4
- HFINTOSC 16 MHz
- LFINTOSC
- MFINTOSC 31.25 kHz

SMT时钟源通过配置SMTxCLK寄存器的CSEL<2:0>位进行选择。时钟源还可以使用SMTxCON0寄存器的PS<1:0>位进行预分频。预分频后的时钟源用于为计数器和该模块使用的任何同步逻辑提供时钟。

### 30.1.2 周期匹配中断

类似于其他定时器，SMT会在SMTxTMR计满返回到0时触发中断。它在SMTxTMR = SMTxPR时发生，无论模式如何。因此，在任何依赖于外部信号或窗口来复位定时器的模式下，需要将SMTxPR设置为大于预期信号或窗口的周期，这样才能正确工作。

## 30.2 基本定时器功能寄存器

SMTxTMR时基和SMTxCPW/SMTxPR/SMTxCPR缓冲寄存器提供了几种功能，可以使用软件手动进行更新。

### 30.2.1 时基

SMTxTMR是一个24位计数器，它是SMT的核心。在SMT的每种模式下，它用作进行测量的基本计数器/定时器。通过将SMTxSTAT寄存器的RST位置1，可以将它复位为值24'h00\_0000。可以用软件读写它，但并不保证原子访问，因此应当仅在GO = 0时读写SMTxTMR，或软件应采取其他措施来以确保SMTxTMR读/写的完整性。

### 30.2.2 脉冲宽度锁存器寄存器

SMTxCPW寄存器为24位SMT脉冲宽度锁存器。它们用于在被各种信号触发时锁存SMTxTMR的值，这些信号由SMT当前所处的模式决定。通过将SMTxSTAT寄存器的CPWUP位置1，也可以使用SMTxTMR的当前值来更新SMTxCPW寄存器。

### 30.2.3 周期锁存器寄存器

SMTxCPR寄存器为24位SMT周期锁存器。它们用于在被各种其他信号触发时锁存SMTxTMR的其他值，这些其他信号由SMT当前所处的模式决定。

通过将SMTxSTAT寄存器的CPRU位置1，也可以使用SMTxTMR的当前值来更新SMTxCPR寄存器。

## 30.3 暂停操作

使用SMTxCON0寄存器中的STP位，可以阻止计数器发生计满返回。当使能暂停时，周期匹配中断会持续发生，直到复位SMTxTMR为止（通过手动复位（第30.2.1节“时基”）或通过用软件清零SMTxCON1寄存器的SMTxGO位并写入SMTxTMR值）。

## 30.4 极性控制

SMT的3个输入信号具有极性控制功能，决定它们是高电平有效/正边沿还是低电平有效/负边沿信号。

以下位适用于极性控制：

- WSEL位（窗口极性）
- SSEL位（信号极性）
- CSEL位（时钟极性）

这些位位于SMTxCON0寄存器中。

## 30.5 状态信息

SMT可以为用户提供输入状态信息，而无需处理传入信号的极性。

### 30.5.1 窗口状态

窗口状态通过SMTxSTAT寄存器的WS位确定。该位仅在窗口测量、门控计数器和门控窗口测量模式下使用，并且仅在TS = 1时有效，在非计数器模式下会被延迟，被延迟的时间量等于同步器延时。

### 30.5.2 信号状态

信号状态通过SMTxSTAT寄存器的AS位确定。该位在除窗口测量、行程时间和捕捉测量之外的所有其他模式下使用，并且仅在TS = 1时有效，在非计数器模式下会被延迟，被延迟的时间量等于同步器延时。

### 30.5.3 运行状态

定时器运行状态通过SMTxSTAT寄存器的TS位确定，在非计数器模式下会被延迟，被延迟的时间量等于同步器延时。

30.6 工作模式

表30-1对工作模式进行了汇总。后面几节提供了关于如何使用这些模式的详细说明和示例。请注意，所有波形均假定WPOL/SPOL/CPOL = 0。当WPOL/SPOL/CPOL = 1时，所有SMTSIGx、SMTWINx和SMT时钟信号的极性将与所示极性相反。对于所有模式，REPEAT位控制是进行重复采集还是单次采集。当REPEAT = 0（单次采集模式）时，则在完成采集时，定时器将停止递增，SMTxGO位将发生复位。否则，定时器将继续工作，允许后续采集覆盖先前的采集，直到用软件停止定时器为止。

30.6.1 定时器模式

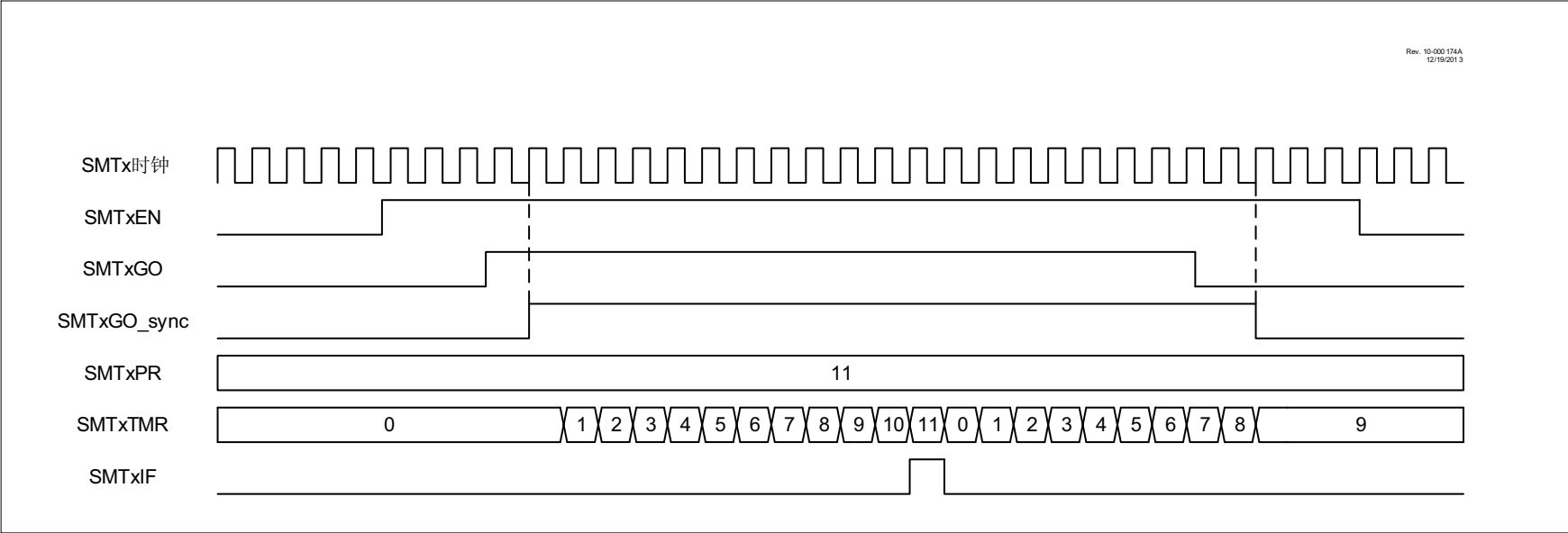
定时器模式是最简单的工作模式，SMTxTMR在该模式下用作16/24位定时器。该模式下不会发生数据采集。只要用软件将SMTxGO位置1，定时器就会一直递增。没有SMT窗口或SMT信号事件会影响SMTxGO位。所有时序都与SMT时钟源进行同步。当定时器发生周期匹配（SMTxTMR = SMTxPR）时，SMTxTMR会发生复位，并触发周期匹配中断。请参见图30-3。

表30-1： 工作模式

MODE	工作模式	同步工作	参考章节
0000	定时器	是	第30.6.1节“定时器模式”
0001	门控定时器	是	第30.6.2节“门控定时器模式”
0010	周期和占空比采集	是	第30.6.3节“周期和占空比模式”
0011	高电平和低电平时间测量	是	第30.6.4节“高电平和低电平测量模式”
0100	窗口测量	是	第30.6.5节“窗口测量模式”
0101	门控窗口测量	是	第30.6.6节“门控窗口测量模式”
0110	行程时间	是	第30.6.7节“行程时间测量模式”
0111	捕捉	是	第30.6.8节“捕捉模式”
1000	计数器	否	第30.6.9节“计数器模式”
1001	门控计数器	否	第30.6.10节“门控计数器模式”
1010	窗口计数器	否	第30.6.11节“窗口计数器模式”
1011 - 1111	保留	—	—

Rev. 10-000 174A  
12/19/2013

图 30-3: 定时器模式时序图

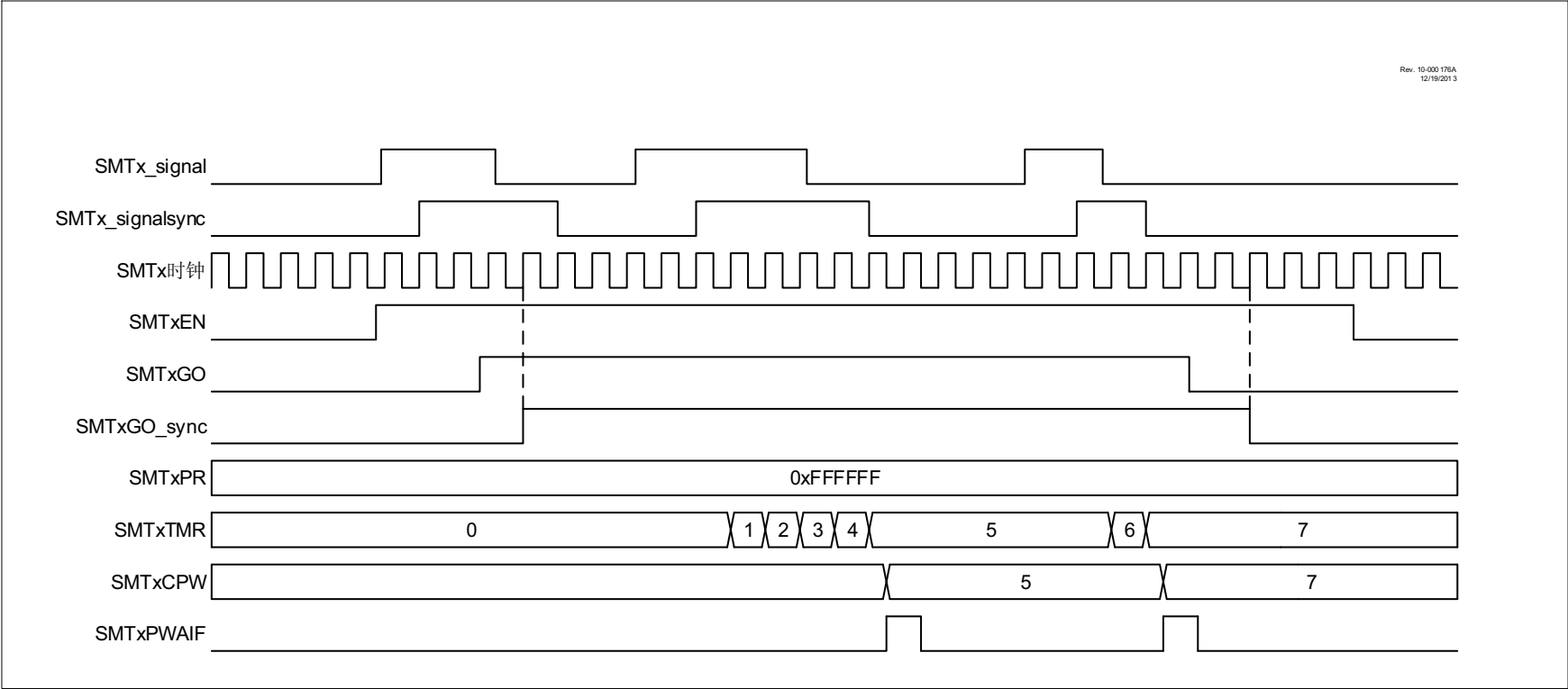


## 30.6.2 门控定时器模式

门控定时器模式使用 **SMTSIGx** 输入来控制 **SMTxTMR** 是否进行递增。在外部信号的下降沿，**SMTxCPW** 寄存器会被更新为 **SMTxTMR** 的当前值。图30-4和图30-5给出了进行重复采集和单次采集的示例波形。

Rev. 10-000 176A  
12/18/2013

图 30-4: 门控定时器模式重复采集时序图





Rev. 10-000 175A  
12/19/2013

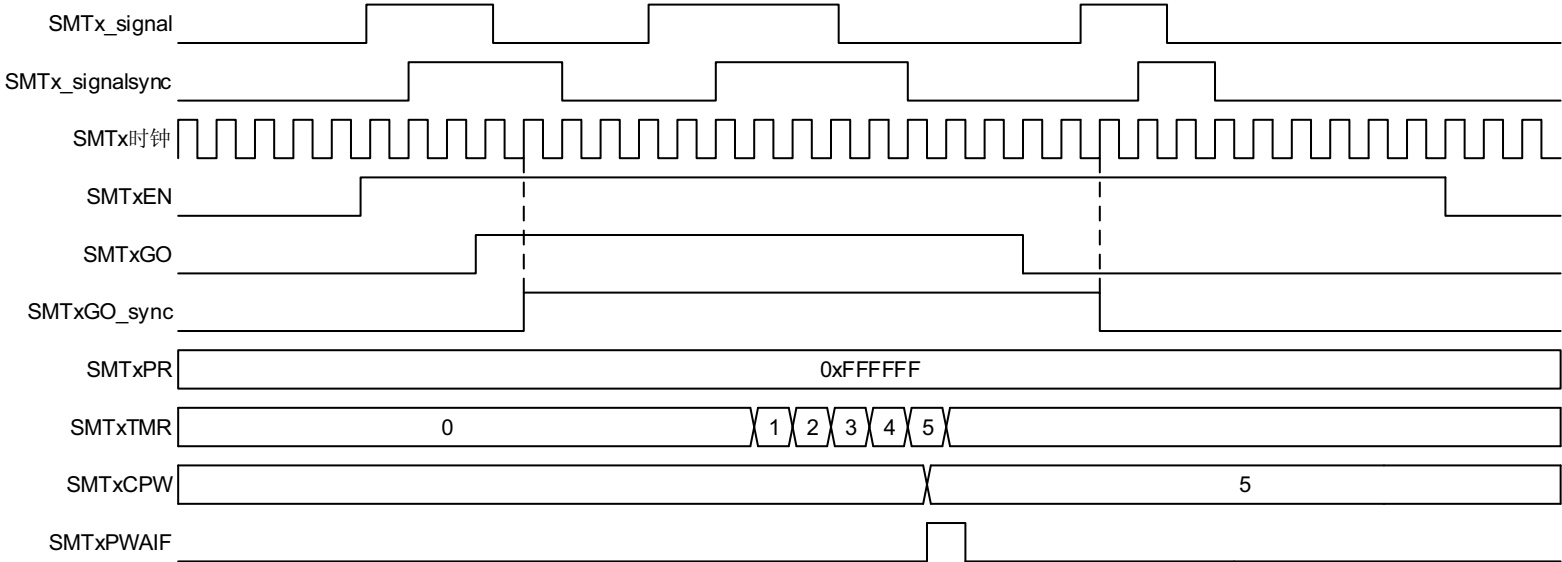


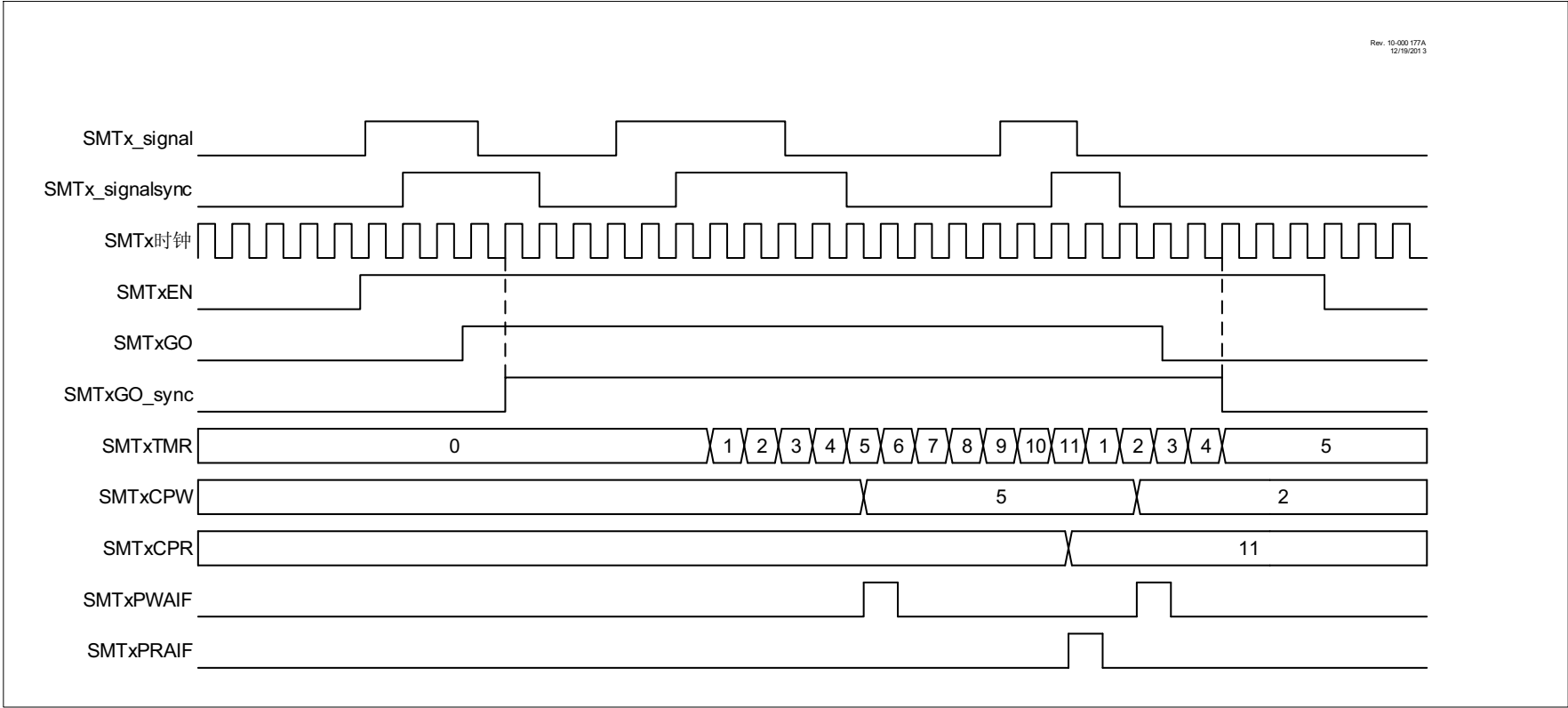
图 30-5: 门控定时器模式单次采集时序图

## 30.6.3 周期和占空比模式

在占空比模式下，可以相对于SMT时钟采集SMTx\_signal的占空比和周期（取决于极性）。CPW寄存器在信号的下降沿更新，CPR寄存器在信号的上升沿更新，并且SMTxTMR复位为0x0001。此外，在SMT处于单次采集模式时，SMTxGO位是在上升沿发生复位。请参见图30-6和图30-7。

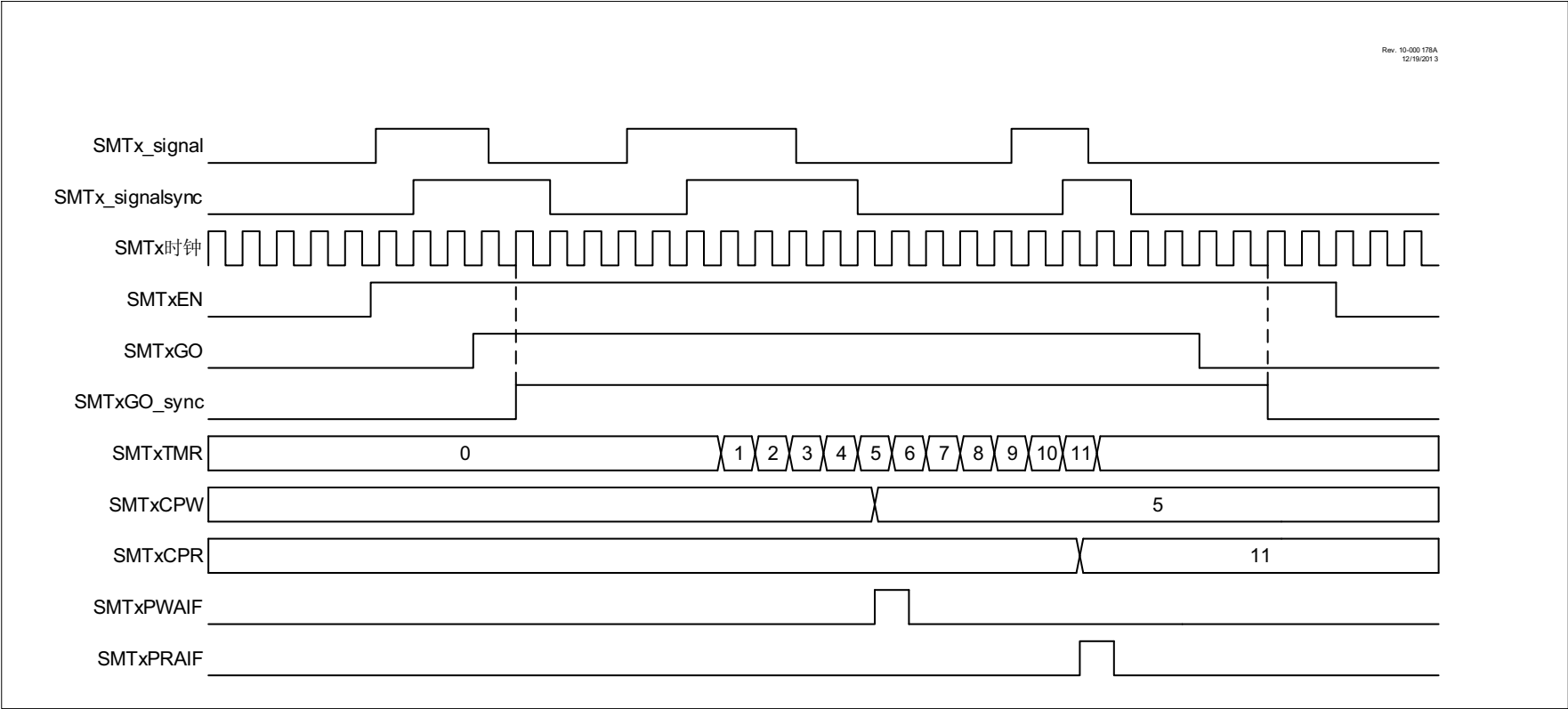
Rev. 10-000 177A  
12/19/2013

图 30-6: 周期和占空比重复采集模式时序图



Rev. 10-000 178A  
12/19/2013

图 30-7: 周期和占空比单次采集时序图

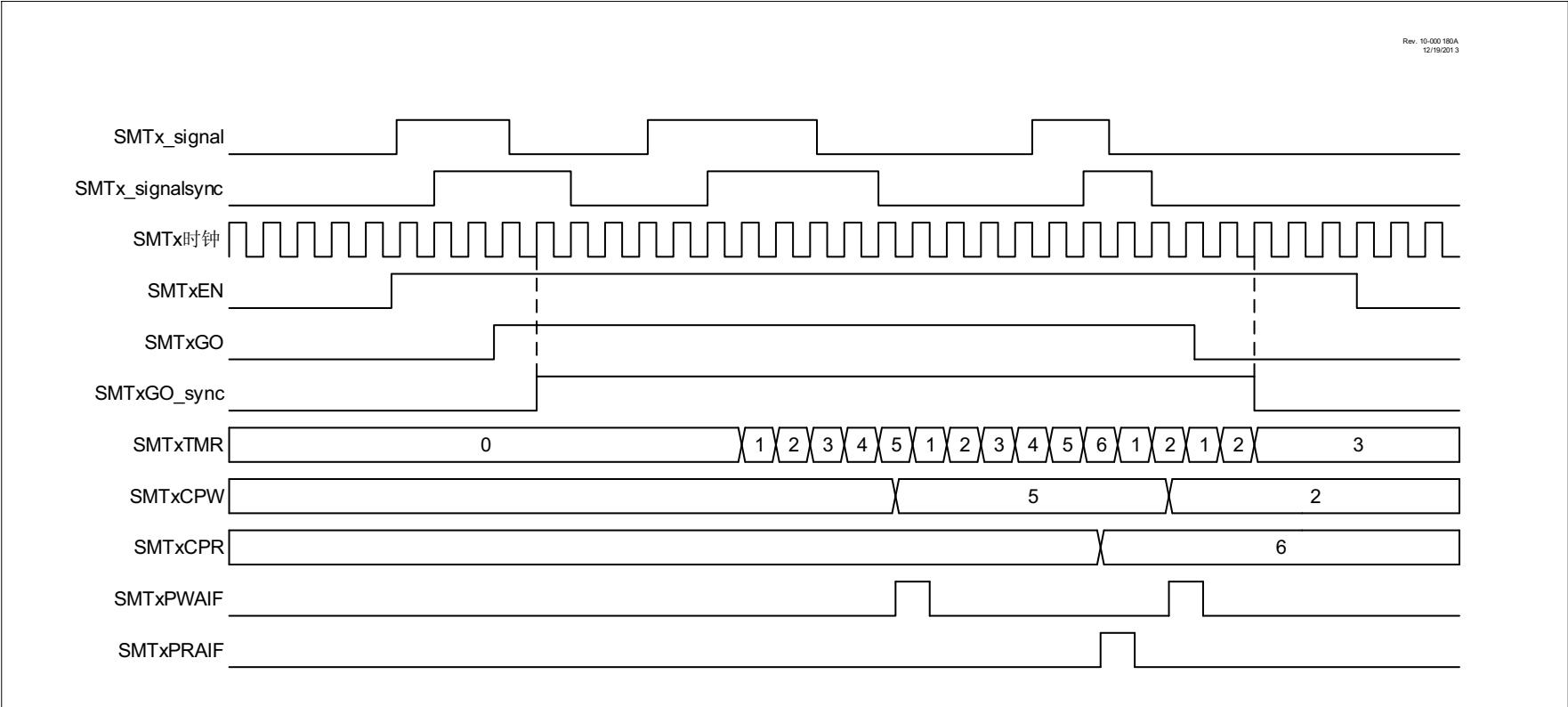


## 30.6.4 高电平和低电平测量模式

该模式相对于 **SMT** 时钟测量 **SMTSIGx** 的高电平和低电平脉冲时间。它在 **SMTSIGx** 输入的上升沿开始递增 **SMTxTMR**，在下降沿使用值更新 **SMTxCPW** 寄存器并复位 **SMTxTMR**，然后重新开始递增。在检测到另一个上升沿时，它会使用其当前值更新 **SMTxCPR** 寄存器并再次复位 **SMTxTMR**，然后重新开始递增。请参见图 30-8 和图 30-9。

Rev. 10-000 180A  
12/19/2013

图 30-8: 高电平和低电平测量模式重复采集时序图



Rev. 10-000 179A  
12/19/2013

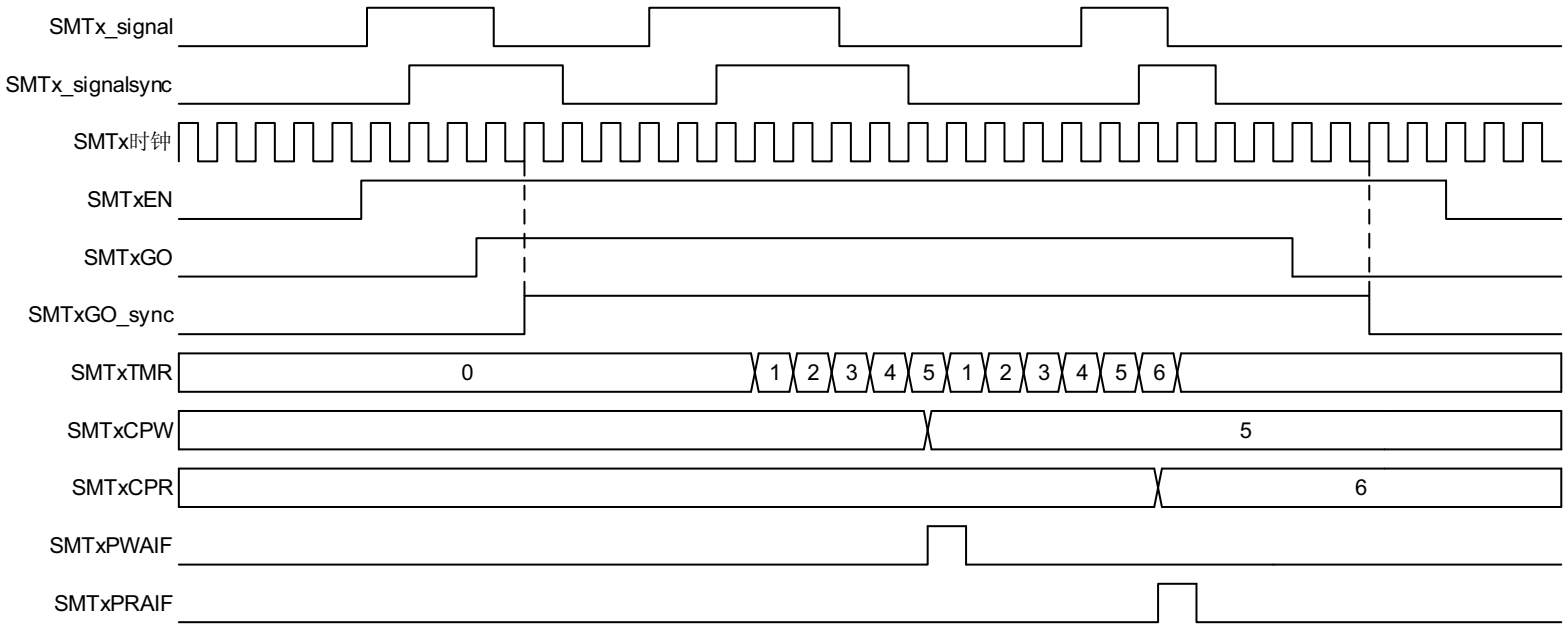


图30-9: 高电平和低电平测量模式单次采集时序图

## 30.6.5 窗口测量模式

该模式测量 SMT 的 SMTWINx 输入的窗口持续时间。它在 SMTWINx 输入的上升沿开始递增定时器，在第二个上升沿使用定时器的值更新 SMTxCPR 寄存器并复位定时器。请参见图 30-10 和图 30-11。



Rev. 10-000 162A  
12/19/2013

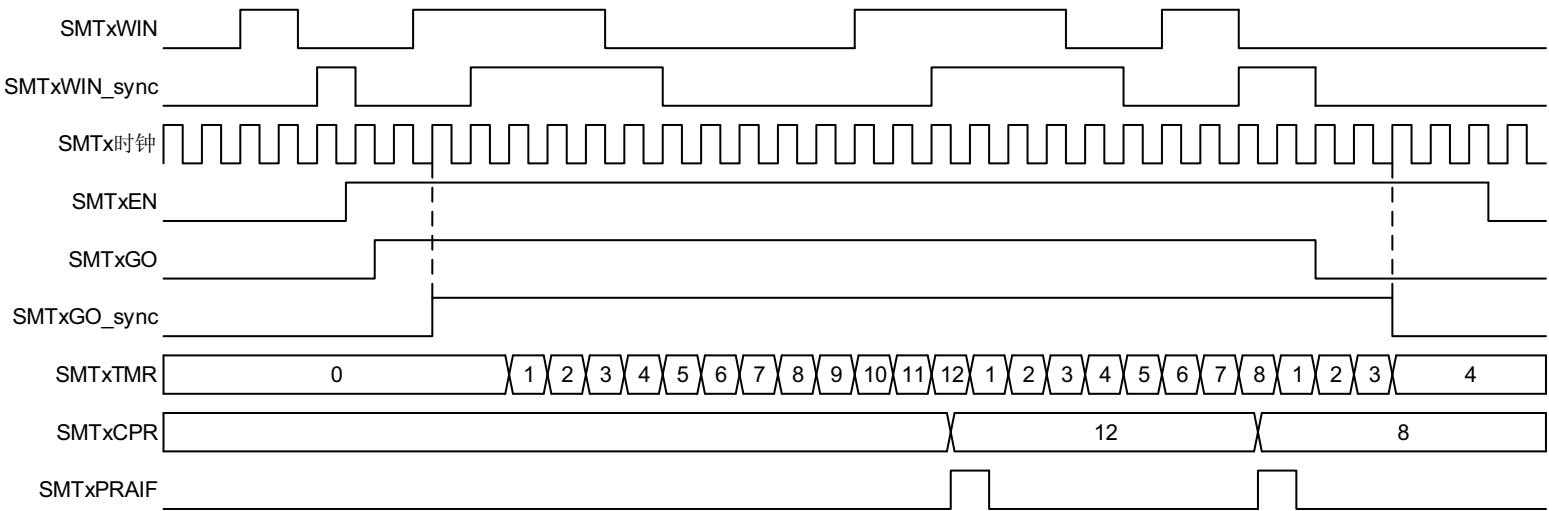


图30-10: 窗口测量模式重复采集时序图

Rev. 10-000-181A  
12/19/2013

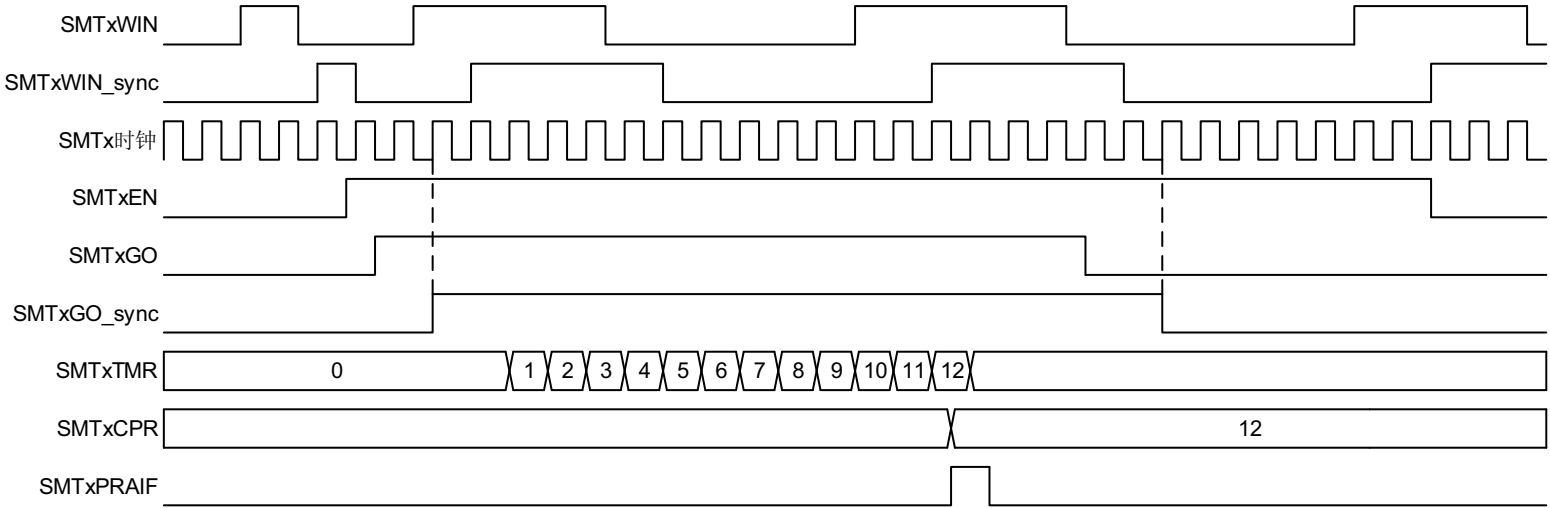


图 30-11: 窗口测量模式单次采集时序图

## 30.6.6 门控窗口测量模式

该模式测量 **SMTx\_signal** 输入在一个已知输入窗口中的占空比。它的实现方式是，在 **SMTx\_signal** 输入为高电平时，对于时钟信号的每个脉冲递增定时器，并在 **SMTWINx** 输入第一个上升沿之后的每个上升沿更新 **SMTxCPR** 寄存器并复位定时器。请参见 [图30-12](#) 和 [图30-13](#)。

Rev. 10-000 184A  
12/19/2013

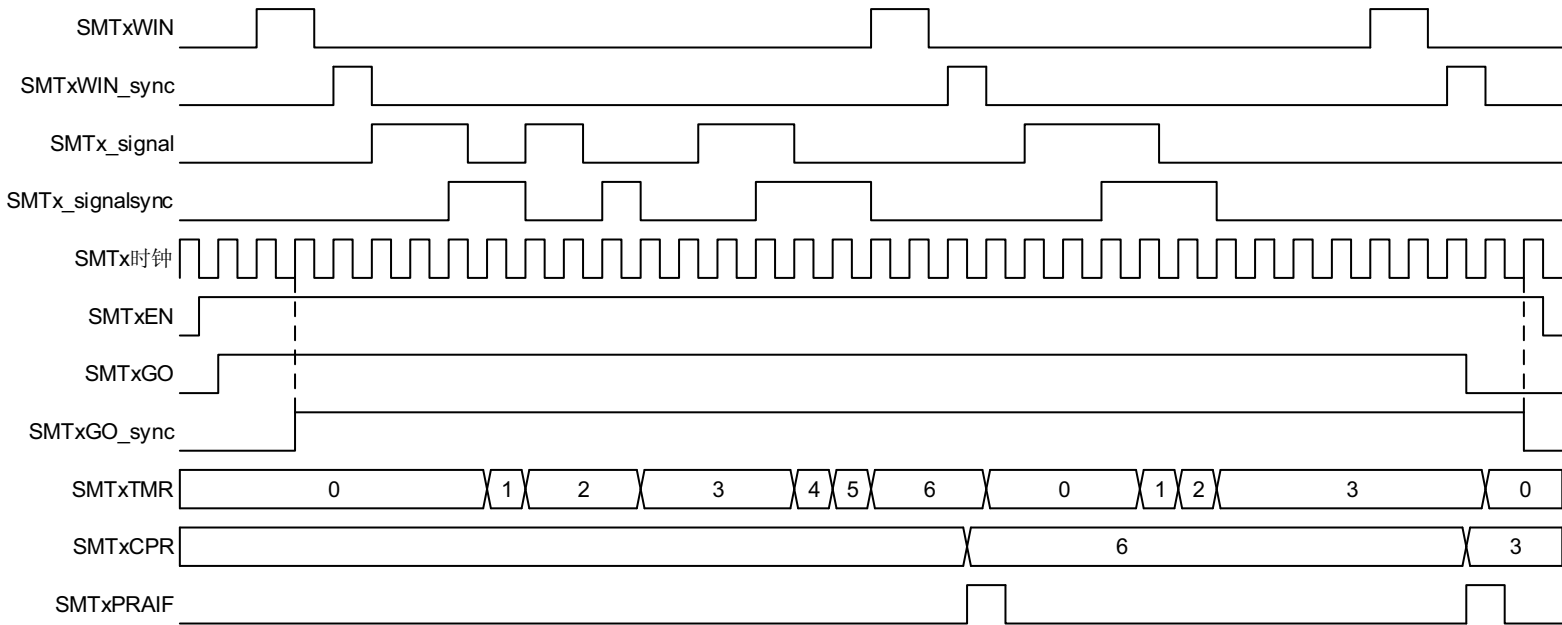
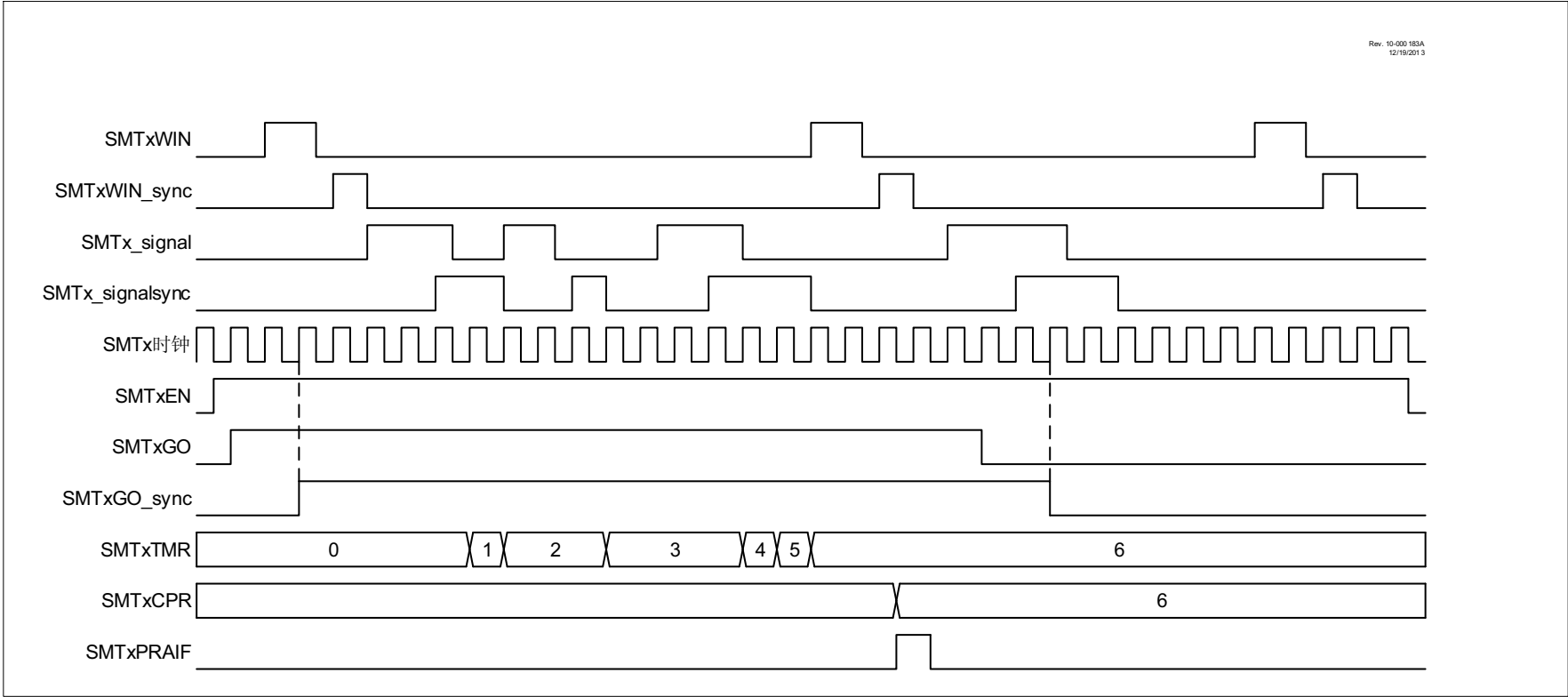


图 30-12: 门控窗口测量模式重复采集时序图

Rev. 10-000 183A  
12/19/2013

图30-13: 门控窗口测量模式单次采集时序图



## 30.6.7 行程时间测量模式

该模式测量 SMTWINx 输入上升沿和 SMTx\_signal 输入上升沿之间的时间间隔，即在 SMTWINx 输入上检测到上升沿时开始递增定时器，在 SMTx\_signal 输入上检测到上升沿时更新 SMTxCPR 寄存器并复位定时器。在出现两个 SMTWINx 上升沿而未出现 SMTx\_signal 上升沿的情况下，它会使用定时器的当前值更新 SMTxCPW 寄存器并复位定时器值。请参见图 30-14 和图 30-15。

Rev. 10-000 186A  
12/19/2013

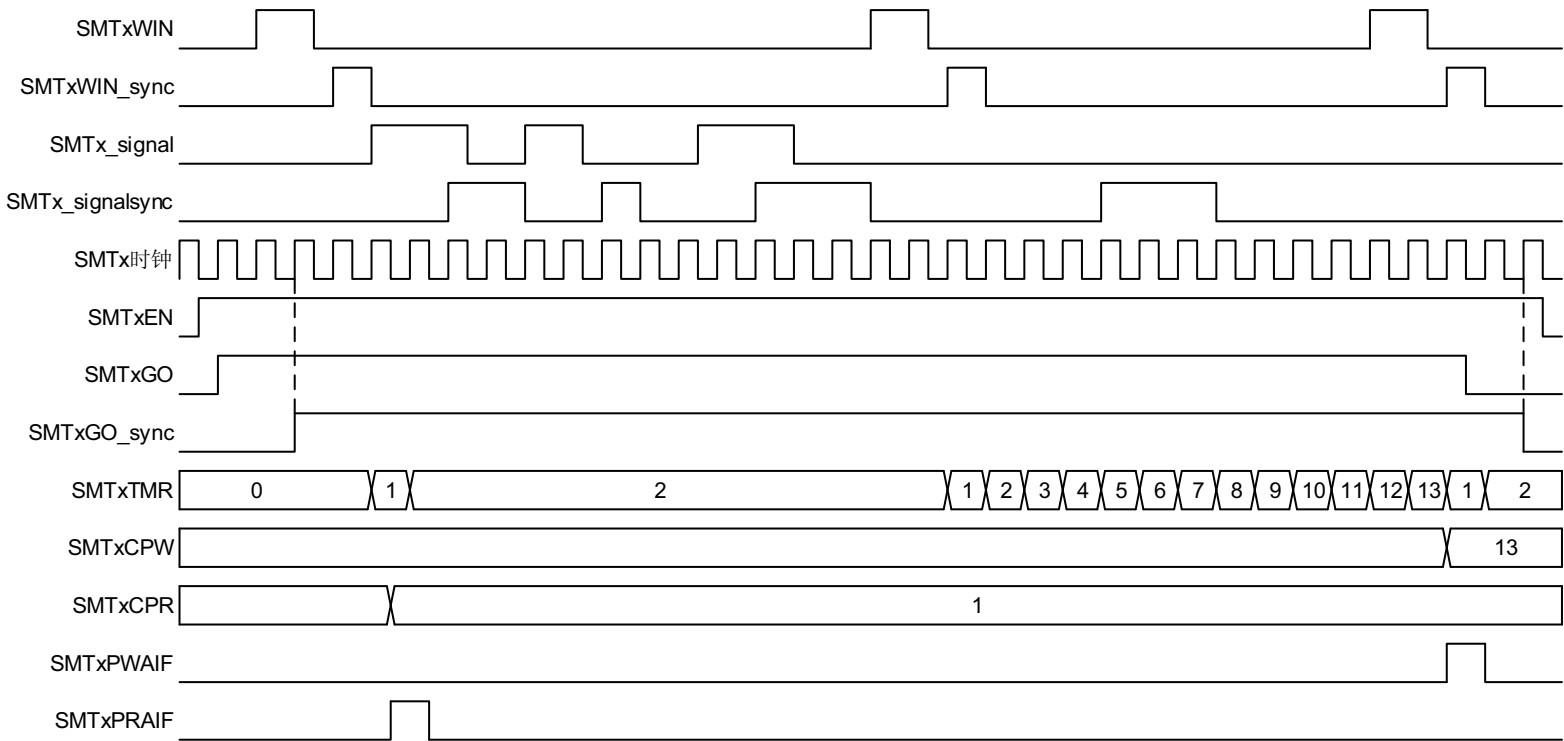


图30-14: 行程时间模式重复采集时序图

Rev. 10-000 185A  
12/19/2013

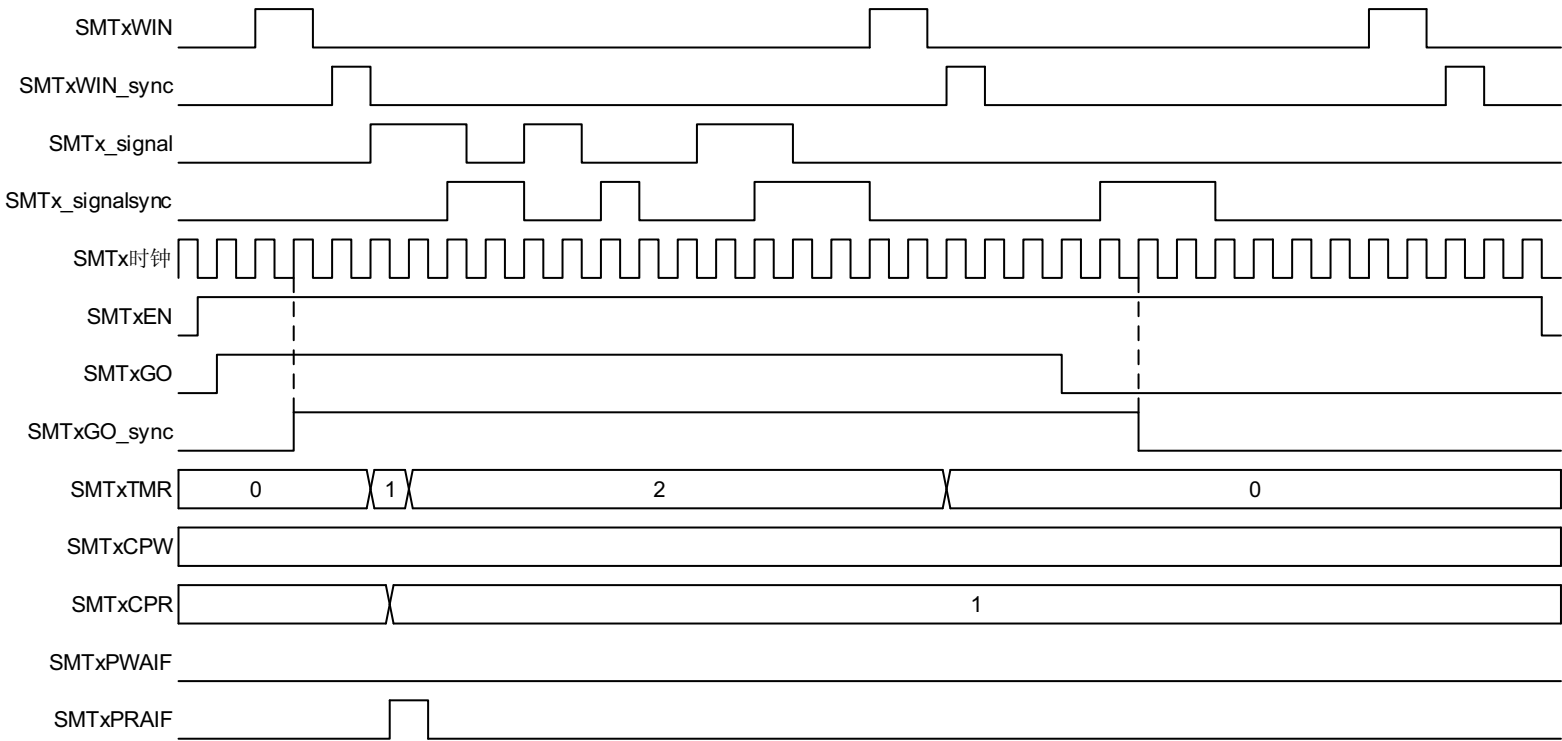


图 30-15: 行程时间模式单次采集时序图



## 30.6.8 捕捉模式

该模式基于 SMTWINx 输入上的上升沿或下降沿捕捉定时器值并触发中断。它模仿了 CCP 模块的捕捉功能。定时器在 SMTxGO 位置 1 时开始递增，并在 SMTWINx 的每个上升沿更新 SMTxCPR 寄存器的值，在 SMTWINx 的每个下降沿更新 CPW 寄存器的值。在该模式下，任何硬件条件都不会复位定时器，如果需要，必须用软件复位。请参见图 30-16 和图 30-17。

Rev. 10-000 186A  
12/19/2013

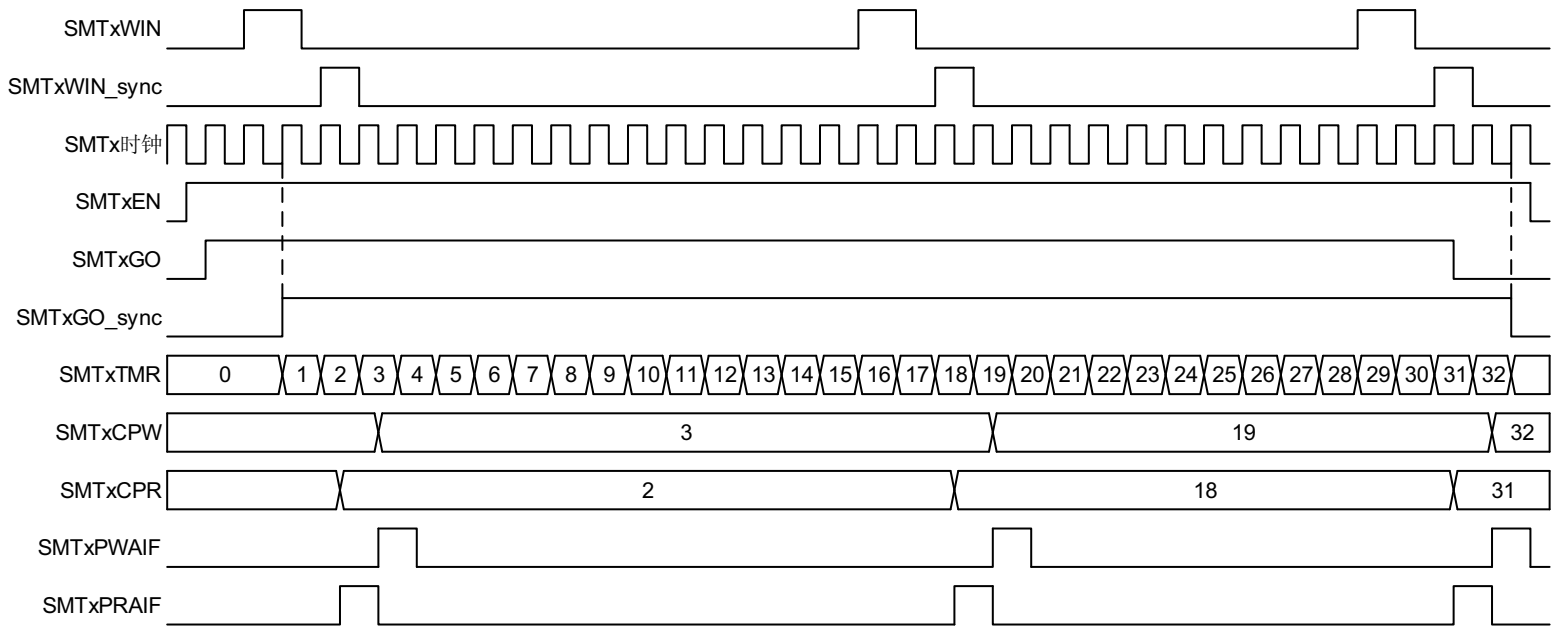


图 30-16: 捕捉模式重复采集时序图

Rev. 10-000 187A  
12/19/2013

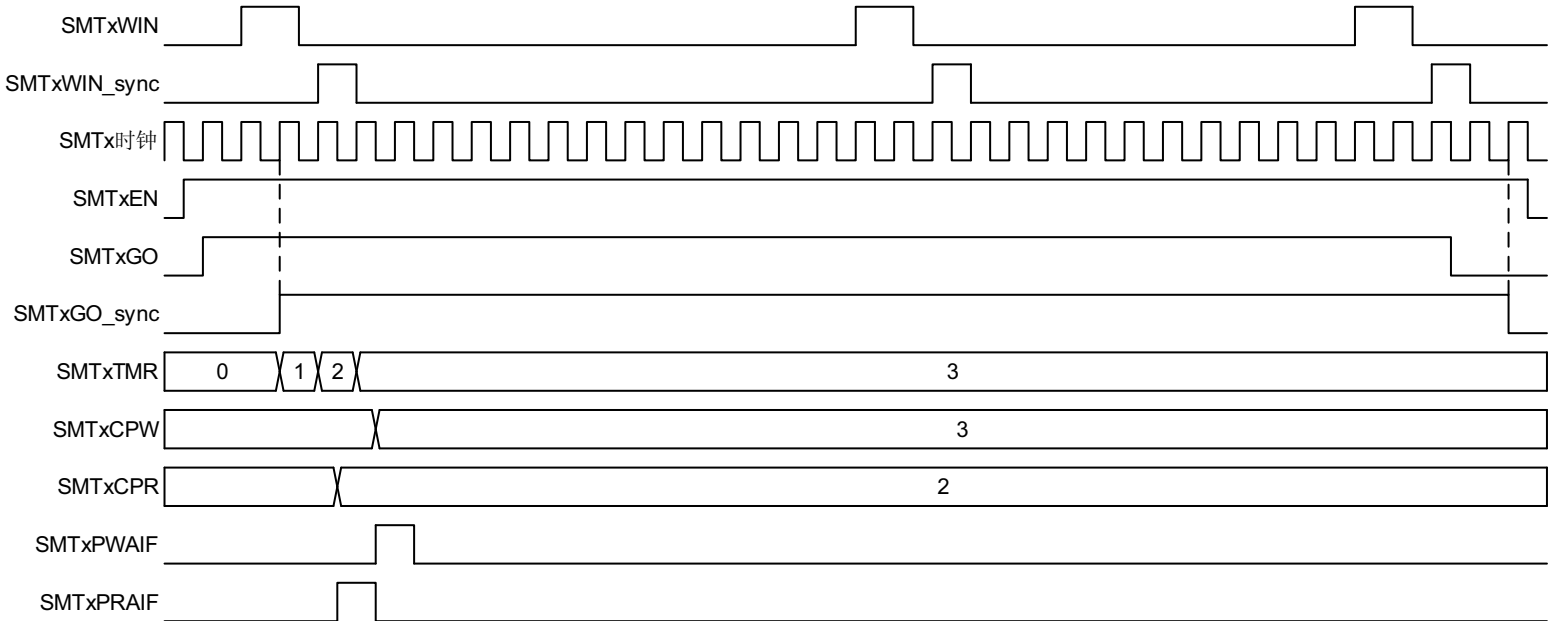


图30-17: 捕捉模式单次采集时序图

## 30.6.9 计数器模式

该模式对于 **SMTx\_signal** 输入的每个脉冲递增定时器。该模式是与 **SMT** 时钟异步的，它使用 **SMTx\_signal** 作为时间源。可以在 **SMTxWIN** 输入的上升沿将 **SMTxCPW** 寄存器更新为当前 **SMTxTMR** 值。请参见图 30-18。

Rev. 10-000188A  
8/25/2014

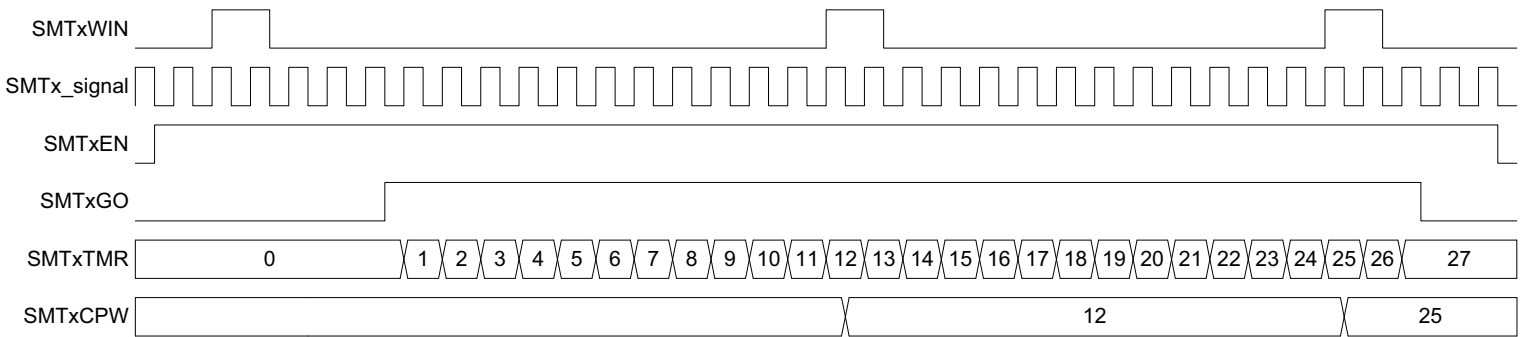


图30-18: 计数器模式时序图

## 30.6.10 门控计数器模式

该模式对SMTx\_signal输入上的脉冲进行计数，并通过SMTxWIN输入进行门控。它在检测到SMTxWIN输入的上升沿时开始递增定时器，并在SMTxWIN输入的下降沿更新SMTxCPW寄存器。请参见图30-19和图30-20。

图30-19: 门控计数器模式重复采集时序图

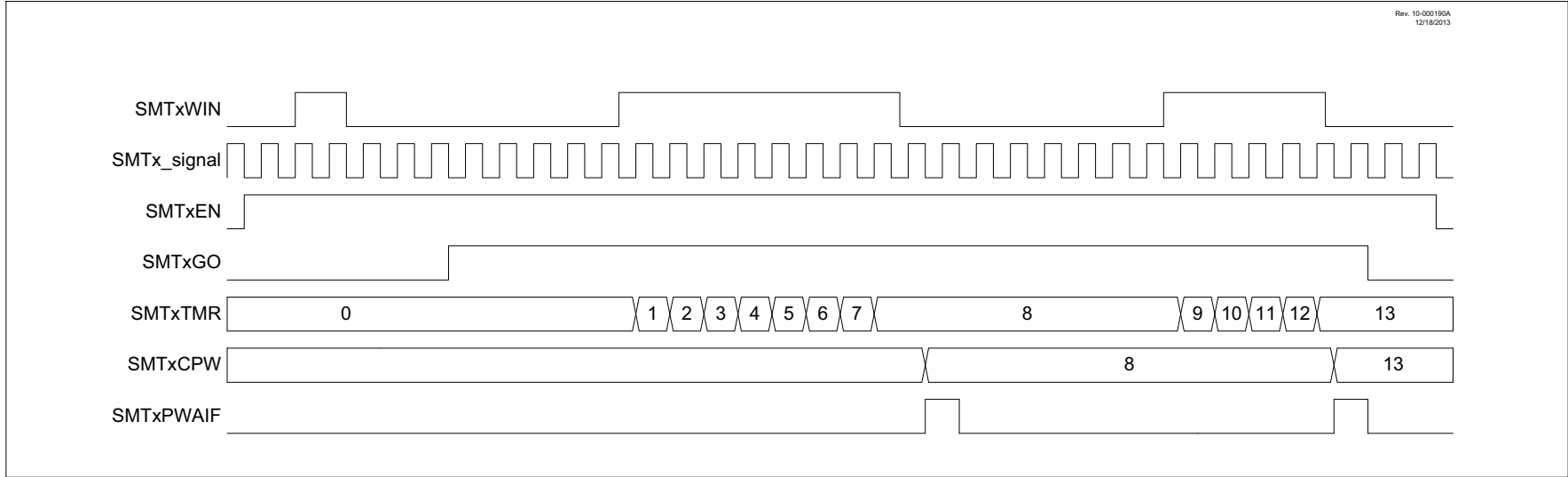
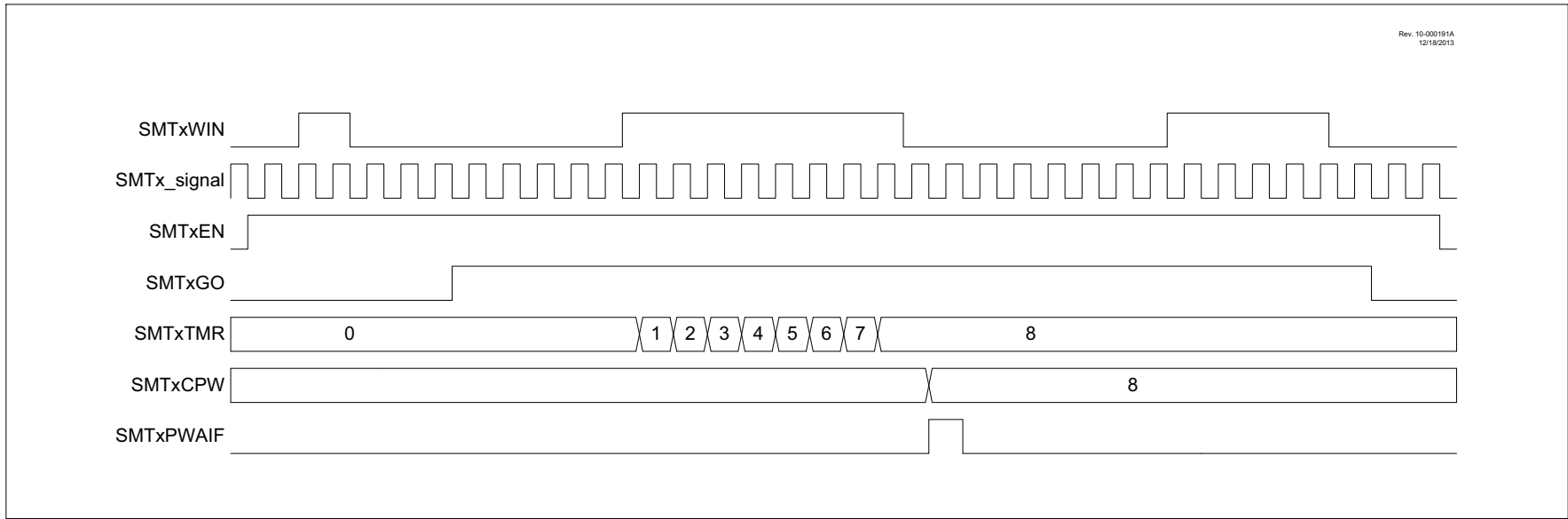


图30-20: 门控计数器模式单次采集时序图



## 30.6.11 窗口计数器模式

该模式在由SMTxWIN输入指定的窗口中，对SMTx\_signal输入上的脉冲进行计数。它在检测到SMTxWIN输入的上升沿时开始计数，在SMTxWIN输入的下降沿更新SMTxCPW寄存器，并在SMTxWIN输入第一个上升沿之外的每个上升沿更新SMTxCPR寄存器。请参见图30-21和图30-22。



图 30-21: 窗口计数器模式重复采集时序图

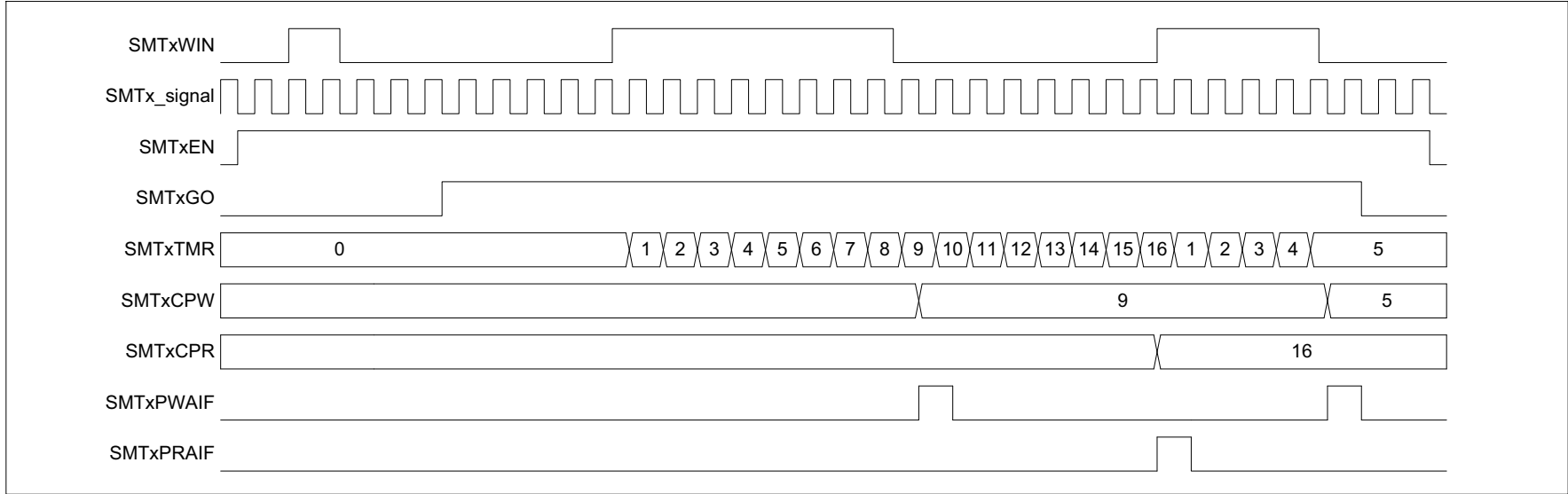
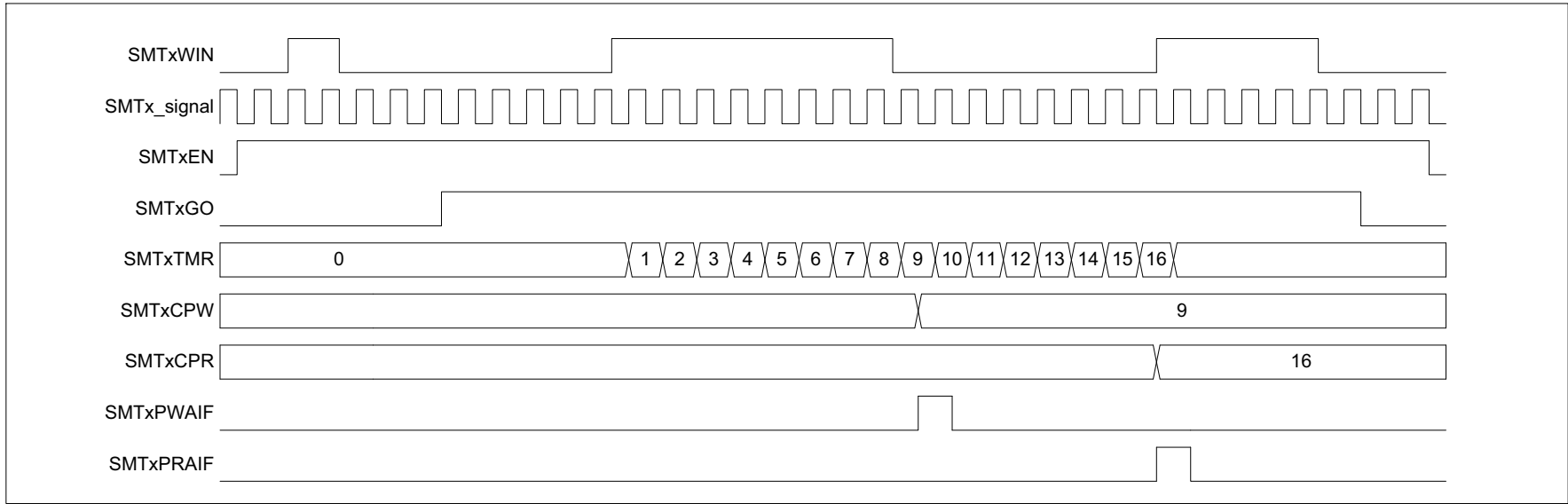


图 30-22: 窗口计数器模式单次采集时序图



## 30.7 中断

SMT可以在3种不同条件下触发中断：

- PW采集完成
- PR采集完成
- 计数器周期匹配

中断由器件的PIR和PIE寄存器控制。

### 30.7.1 PW和PR采集中断

SMT可以在它每次更新SMTxCPW和SMTxCPR寄存器时触发中断，进行这些更新的条件取决于SMT模式，每种模式的具体章节中对此进行了介绍。SMTxCPW中断由SMTxPWAIF和SMTxPWAIE位控制，这两个位分别位于寄存器PIR4和PIE4中。SMTxCPR中断由SMTxPRAIF和SMTxPRAIE位控制，这两个位也分别位于寄存器PIR4和PIE4中。

在同步SMT模式下，中断触发与SMTxCLK进行同步。在异步模式下，中断触发是异步的。在两种模式下，在触发之后，中断都会与CPU时钟进行同步。

### 30.7.2 计数器周期匹配中断

如[第30.1.2节“周期匹配中断”](#)所述，SMT还会在SMTxTMR与SMTxPR匹配时产生中断，其周期匹配限制功能如[第30.3节“暂停操作”](#)所述。周期匹配中断由SMTxIF和SMTxIE控制，这两个位分别位于寄存器PIR4和PIE4中。

### 30.8 寄存器定义：SMT控制

表30-2给出了信号测量定时器外设的长位名称前缀。更多信息，请参见第1.1节“寄存器和位的命名约定”。

表30-2:

外设	位名称前缀
SMT1	SMT1
SMT2	SMT2

#### 寄存器30-1: SMTxCON0: SMT控制寄存器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	R/W-0/0
EN <sup>(1)</sup>	—	STP	WPOL	SPOL	CPOL	SMTxPS<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 **EN: SMT使能位<sup>(1)</sup>**  
 1 = 使能SMT  
 0 = 禁止SMT; 复位内部状态, 禁止时钟请求
- bit 6 **未实现:** 读为0
- bit 5 **STP: SMT计数器暂停使能位**  
 当SMTxTMR = SMTxPR时:  
 1 = 计数器保持为SMTxPR; 在输入时钟时产生周期匹配中断  
 0 = 计数器复位为24'h000000; 在输入时钟时产生周期匹配中断
- bit 4 **WPOL: SMTxWIN输入极性控制位**  
 1 = SMTxWIN信号低电平有效/使能下降沿  
 0 = SMTxWIN信号高电平有效/使能上升沿
- bit 3 **SPOL: SMTxSIG输入极性控制位**  
 1 = SMTx\_signal低电平有效/使能下降沿  
 0 = SMTx\_signal高电平有效/使能上升沿
- bit 2 **CPOL: SMT时钟输入极性控制位**  
 1 = SMTxTMR在选定时钟信号的下降沿递增  
 0 = SMTxTMR在选定时钟信号的上升沿递增
- bit 1-0 **SMTxPS<1:0>: SMT预分频比选择位**  
 11 = 预分频比 = 1:8  
 10 = 预分频比 = 1:4  
 01 = 预分频比 = 1:2  
 00 = 预分频比 = 1:1

**注 1:** 将EN设置为0不会影响寄存器的内容。

# PIC16(L)F1615/9

寄存器 30-2: SMTxCON1: SMT 控制寄存器 1

R/W/HC-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
SMTxGO	REPEAT	—	—	MODE<3:0>			
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	q = 值取决于具体条件

bit 7	<b>SMTxGO:</b> SMT 运行数据采集位 1 = 使能递增、数据采集 0 = 禁止递增、数据采集
bit 6	<b>REPEAT:</b> SMT 重复采集使能位 1 = 使能重复数据采集模式 0 = 使能单次采集模式
bit 5-4	<b>未实现:</b> 读为 0
bit 3-0	<b>MODE&lt;3:0&gt;:</b> SMT 工作模式选择位 1111 = 保留 • • • 1011 = 保留 1010 = 窗口计数器 1001 = 门控计数器 1000 = 计数器 0111 = 捕捉 0110 = 行程时间 0101 = 门控窗口测量 0100 = 窗口测量 0011 = 高电平和低电平时间测量 0010 = 周期和占空比采集 0001 = 门控定时器 0000 = 定时器

寄存器 30-3: SMTxSTAT: SMT 状态寄存器

R/W/HC-0/0	R/W/HC-0/0	R/W/HC-0/0	U-0	U-0	R-0/0	R-0/0	R-0/0
CPRUP	CPWUP	RST	—	—	TS	WS	AS
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

**CPRUP:** SMT 手动周期缓冲器更新位  
1 = 请求更新 SMTxPRx 寄存器  
0 = SMTxPRx 寄存器更新已完成
- bit 6

**CPWUP:** SMT 手动脉冲宽度缓冲器更新位  
1 = 请求更新 SMTxCPW 寄存器  
0 = SMTxCPW 寄存器更新已完成
- bit 5

**RST:** SMT 手动定时器复位位  
1 = 请求复位 SMTxTMR 寄存器  
0 = SMTxTMR 寄存器更新已完成
- bit 4-3

**未实现:** 读为 0
- bit 2

**TS:** SMT 运行值状态位  
1 = SMT 定时器正在递增  
0 = SMT 定时器不在递增
- bit 1

**WS:** SMTxWIN 值状态位  
1 = SMT 窗口已打开  
0 = SMT 窗口已关闭
- bit 0

**AS:** SMT\_signal 值状态位  
1 = 正在进行 SMT 采集  
0 = 未进行 SMT 采集

# PIC16(L)F1615/9

寄存器 30-4: SMTxCLK: SMT时钟选择寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CSEL<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 7-3
- 未实现: 读为0
- bit 2-0
- CSEL<2:0>: SMT时钟选择位
- 111 = 保留
- 110 = AT1\_perclk
- 101 = MFINTOSC
- 100 = MFINTOSC/16
- 011 = LFINTOSC
- 010 = HFINTOSC 16 MHz
- 001 = Fosc/4
- 000 = Fosc

寄存器 30-5: SMT1WIN: SMT1 窗口输入选择寄存器

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
—	—	—	WSEL<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	WSEL<4:0>: SMT1 窗口选择位
	11111 = 保留
	•
	•
	•
	11000 = 保留
	10111 = MFINTOSC/16
	10110 = AT1_perclk
	10101 = LFINTOSC
	10100 = PWM4_out
	10011 = PWM3_out
	10010 = SMT2_match
	10001 = 保留
	10000 = TMR0_overflow
	01111 = TMR5_overflow
	01110 = TMR3_overflow
	01101 = TMR1_overflow
	01100 = LC4_out
	01011 = LC3_out
	01010 = LC2_out
	01001 = LC1_out
	01000 = TMR6_postscaled
	00111 = TMR4_postscaled
	00110 = TMR2_postscaled
	00101 = ZCD1_out
	00100 = CCP2_out
	00011 = CCP1_out
	00010 = C2OUT_sync
	00001 = C1OUT_sync
	00000 = SMTWINx 引脚

# PIC16(L)F1615/9

寄存器 30-6: SMT2WIN: SMT2 窗口输入选择寄存器

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
—	—	—	WSEL<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	WSEL<4:0>: SMT2 窗口选择位
	11111 = 保留
	•
	•
	•
	11000 = 保留
	10111 = MFINTOSC/16
	10110 = AT1_perclk
	10101 = LFINTOSC
	10100 = PWM4_out
	10011 = PWM3_out
	10010 = 保留
	10001 = SMT1_match
	10000 = TMR0_overflow
	01111 = TMR5_overflow
	01110 = TMR3_overflow
	01101 = TMR1_overflow
	01100 = LC4_out
	01011 = LC3_out
	01010 = LC2_out
	01001 = LC1_out
	01000 = TMR6_postscaled
	00111 = TMR4_postscaled
	00110 = TMR2_postscaled
	00101 = ZCD1_out
	00100 = CCP2_out
	00011 = CCP1_out
	00010 = C2OUT_sync
	00001 = C1OUT_sync
	00000 = SMTWINx 引脚



寄存器 30-7: SMT1SIG: SMT1 信号输入选择寄存器

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
—	—	—	SSEL<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

SSEL<4:0>: SMT1 信号选择位

11111 = 保留

- 
- 
-

10101 = 保留

10100 = PWM4\_out

10011 = PWM3\_out

10010 = CCP2\_out

10001 = CCP1\_out

10000 = TMR0\_overflow

01111 = SMT2\_match

01110 = 保留

01101 = TMR5\_overflow

01100 = TMR3\_overflow

01011 = TMR1\_overflow

01010 = LC4\_out

01001 = LC3\_out

01000 = LC2\_out

00111 = LC1\_out

00110 = TMR6\_postscaled

00101 = TMR4\_postscaled

00100 = TMR2\_postscaled

00011 = ZCD1\_out

00010 = C2OUT\_sync

00001 = C1OUT\_sync

00000 = SMTxSIG 引脚

# PIC16(L)F1615/9

寄存器 30-8: SMT2SIG: SMT2信号输入选择寄存器

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
—	—	—	SSEL<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	SSEL<4:0>: SMT2信号选择位
	11111 = 保留
	•
	•
	•
	10101 = 保留
	10100 = PWM4_out
	10011 = PWM3_out
	10010 = CCP2_out
	10001 = CCP1_out
	10000 = TMR0_overflow
	01111 = 保留
	01110 = SMT1_match
	01101 = TMR5_overflow
	01100 = TMR3_overflow
	01011 = TMR1_overflow
	01010 = LC4_out
	01001 = LC3_out
	01000 = LC2_out
	00111 = LC1_out
	00110 = TMR6_postscaled
	00101 = TMR4_postscaled
	00100 = TMR2_postscaled
	00011 = ZCD1_out
	00010 = C2OUT_sync
	00001 = C1OUT_sync
	00000 = SMTxSIG 引脚

寄存器 30-9: SMTxTMRL: SMT 定时器寄存器——低字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SMTxTMR<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 SMTxTMR<7:0>: SMT 计数器的有效位——低字节

寄存器 30-10: SMTxTMRH: SMT 定时器寄存器——高字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SMTxTMR<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 SMTxTMR<15:8>: SMT 计数器的有效位——高字节

寄存器 30-11: SMTxTMRU: SMT 定时器寄存器——高字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SMTxTMR<23:16>							
bit 7				bit 0			

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

bit 7-0 SMTxTMR<23:16>: SMT 计数器的有效位——高字节

# PIC16(L)F1615/9

## 寄存器 30-12: SMTxCPR<sub>L</sub>: SMT 捕捉周期寄存器——低字节

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPR<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 SMTxCPR<7:0>: SMT 周期锁存器的有效位——低字节

## 寄存器 30-13: SMTxCPR<sub>H</sub>: SMT 捕捉周期寄存器——高字节

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPR<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 SMTxCPR<15:8>: SMT 周期锁存器的有效位——高字节

## 寄存器 30-14: SMTxCPR<sub>U</sub>: SMT 捕捉周期寄存器——高字节

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPR<23:16>							
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 SMTxCPR<23:16>: SMT 周期锁存器的有效位——高字节

寄存器 30-15: SMTxCPWL: SMT 捕捉脉冲宽度寄存器——低字节

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPW<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 SMTxCPW<7:0>: SMT PW 锁存器的有效位——低字节

寄存器 30-16: SMTxCPWH: SMT 捕捉脉冲宽度寄存器——高字节

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPW<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 SMTxCPW<15:8>: SMT PW 锁存器的有效位——高字节

寄存器 30-17: SMTxCPWU: SMT 捕捉脉冲宽度寄存器——高字节

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMTxCPW<23:16>							
bit 7				bit 0			

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

bit 7-0 SMTxCPW<23:16>: SMT PW 锁存器的有效位——高字节

# PIC16(L)F1615/9

## 寄存器 30-18: SMTxPRL: SMT 周期寄存器——低字节

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<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 **SMTxPR<7:0>**: 周期匹配时的 SMT 定时器值的有效位——低字节

## 寄存器 30-19: SMTxPRH: SMT 周期寄存器——高字节

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<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 **SMTxPR<15:8>**: 周期匹配时的 SMT 定时器值的有效位——高字节

## 寄存器 30-20: SMTxPRU: SMT 周期寄存器——高字节

R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1	R/W-x/1
SMTxPR<23:16>							
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

bit 7-0 **SMTxPR<23:16>**: 周期匹配时的 SMT 定时器值的有效位——高字节

**表 30-3: 与 SMTx 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
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	109
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	114
SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1PS<1:0>		447
SMT1CON1	SMT1GO	REPEAT	—	—	MODE<3:0>				448
SMT1CPRH	SMT1CPR<15:8>								456
SMT1CPRL	SMT1CPR<7:0>								456
SMT1CPRU	SMT1CPR<23:16>								456
SMT1CPWH	SMT1CPW<15:8>								457
SMT1CPWL	SMT1CPW<7:0>								457
SMT1CPWU	SMT1CPW<23:16>								457
SMT1PRH	SMT1PR<15:8>								458
SMT1PRL	SMT1PR<7:0>								458
SMT1PRU	SMT1PR<23:16>								458
SMT1SIG	—	—	—	SSEL<4:0>					453
SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	449
SMT1TMRH	SMT1TMR<15:8>								455
SMT1TMRL	SMT1TMR<7:0>								455
SMT1TMRU	SMT1TMR<23:16>								455
SMT1WIN	—	—	—	WSEL<4:0>					451
SMT2CLK	—	—	—	—	—	CSEL<2:0>			450
SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT2PS<1:0>		447
SMT2CON1	SMT2GO	REPEAT	—	—	MODE<3:0>				448
SMT2CPRH	SMT2CPR<15:8>								456
SMT2CPRL	SMT2CPR<7:0>								456
SMT2CPRU	SMT2CPR<23:16>								456
SMT2CPWH	SMT2CPW<15:8>								457
SMT2CPWL	SMT2CPW<7:0>								457
SMT2CPWU	SMT2CPW<23:16>								457
SMT2PRH	SMT2PR<15:8>								458
SMT2PRL	SMT2PR<7:0>								458
SMT2PRU	SMT2PR<23:16>								458
SMT2SIG	—	—	—	—	—	SSEL<2:0>			453
SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	449
SMT2TMRH	SMT2TMR<15:8>								455
SMT2TMRL	SMT2TMR<7:0>								455
SMT2TMRU	SMT2TMR<23:16>								455
SMT2WIN	—	—	—	WSEL<4:0>					452

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

31.0 角度定时器（AT）模块

角度定时器（Angular Timer，AT）模块可将周期信号细分为更小的等间隔区间，这些区间的数量保持恒定，不随周期信号频率的变化而变化。计数器在每个周期事件中从零开始跟踪这些区间。可通过将计数器值与用户定义的值进行比较来触发事件，也可通过模块的外部事件来捕捉计数器值。这可用于各种应用，例如，交流市电测量、电机停转检测、无刷电机换相以及内燃机的TDC检测。另一个计数器跟踪周期时间。这可以用于基于预编程的设定值来测量周期误差以及检测信号中的缺失脉冲。角度定时器模块包括以下特性：

- 两种工作模式
  - 每周期单个脉冲
  - 每周期多个脉冲
- 两种缺失脉冲模式
  - 自适应
  - 固定
- 多个可选择的时钟源
- 带极性控制的相位时钟输出
- 带极性控制的周期时钟输出
- 带极性控制的缺失脉冲输出
- 相位和周期时钟生成中断以及缺失脉冲检测中断
- 周期设定值和误差寄存器
- 比较脉冲输出
  - 独立中断
- 捕捉输入
  - 输入极性控制
  - 独立中断

31.1 工作原理

考虑一下公式31-1中的说明：

公式31-1：

如果：	并且：	则：
$P = \frac{F}{R}$	$A = \frac{F}{P}$	$A = R$

在这三个公式中：

- P表示周期计数ATxPER
- A表示角度或相位计数ATxPHS
- R表示所需的分辨率ATxRES
- F表示某个任意换算值

请注意，相位计数等于所需的分辨率，与F的值无关。如果令F等于系统时钟与输入信号频率的比值，则意味着相位计数是一个等于所需分辨率的常量，而与输入频率无关。这有许多特殊的用途，包括：

- 使用相位比较功能在周期内按固定相位角创建某个事件
- 使用捕捉功能捕捉发生某个事件时的相位角
- 使用误差功能监视与用户指定的周期时间的偏差

本章接下来几节将介绍这些功能及更多功能的详细信息。

31.2 角度定时器工作模式

AT模块有两种基本工作模式：

- 单脉冲模式
- 多脉冲模式

这两种模式的工作原理相同：将一个周期输入信号分为多个等间隔区间，并允许通过这些区间来触发事件。这两种模式的主要区别在于如何确定周期。单脉冲模式将周期定义为输入脉冲流中相邻两脉冲间的时间间隔。多脉冲模式将周期定义为输入脉冲流中相邻两缺失脉冲间的时间间隔。

这两种模式的主要参数均为ATxRES寄存器对。该值用于确定相位计数器的粒度以及模块的相位时钟输出频率。



## 31.2.1 单脉冲模式

图31-1说明了单脉冲工作模式。输入信号的计算分几步完成。首先，分频器通过ATxRES寄存器对将模块时钟进行分频，并使用产生的信号来递增周期计数器。公式31-2表达了此操作。该公式与公式31-1略有不同，因为计数器包含计数0。为了补偿这一差异，写入分辨率寄存器ATxRES的数字必须比所需分辨率小1。

### 公式31-2:

$$ATxPER = \frac{F(ATxclk)}{F(ATxsig)} \cdot (ATxRES + 1)$$

公式31-2中的变量如下:

- ATxPER 是通过输入信号锁存的周期计数器的值。
- ATxRES 是用户指定的分辨率。相位计数器将递增计数到该值。
- F(ATxclk) 是ATx时钟频率。
- F(ATxsig) 是输入信号频率。

角度定时器的第二步操作是创建相位时钟，相位时钟也在图31-1中进行了说明。输入时钟通过在上一个步骤中锁存的ATxPER值进行分频，产生的信号用于递增相位计数器。该信号还用作相位时钟输出，以及用于将ATxIR0寄存器的PHSIF中断标志位置1。结果是相位计数器从0计数到公式31-3表达的最终值，计数器每递增一次，就会输出一个脉冲。相位计数器的值可用软件通过读取ATxPHS寄存器对来访问。但是，由于需要同步，为了准确读取该寄存器对，指令时钟（Fosc/4）需要至少为ATx\_phsclock输出频率的3倍。

### 公式31-3:

$$ATxPHS(final) = \frac{\left( \frac{F(ATxclk)}{F(ATxsig)} \right)}{(ATxPER + 1)}$$

公式31-3中的变量如下:

- ATxPHS(final) 是相位计数器在被输入信号复位之前将达到的最大值。根据公式31-1可知，该值等于ATxRES。
- ATxPER 是周期计数器的最大值。
- F(ATxclk) 是ATx时钟频率。
- F(ATxsig) 是输入信号频率。

请注意，分频值为ATxPER + 1。理想情况下，该值应该正好是ATxPER，但分频值包括计数0。在大多数应用中，ATxPER的数值较大，因此加1所引入的误差可以忽略不计。

当输入信号表示旋转（例如，电机或交流市电）时，ATxPHS从0计数到ATxRES会非常有用。在这种情况下，输入信号可以理解为由每360度提供一个周期脉冲。由于相位时钟将信号周期等分为多个区间（数量由ATxRES寄存器对决定），因此相位时钟输出上的每个脉冲在相应的旋转中标记固定的相位角，如公式31-4所示。

### 公式31-4:

$$\text{角度分辨率} = \frac{360^\circ}{ATxRES + 1}$$

之后可以将ATxRES和ATxPHS寄存器对的瞬时值代入公式31-5，计算得出旋转的瞬时角。

### 公式31-5:

$$\text{角度} = 360^\circ \cdot \frac{ATxPHS}{ATxRES + 1}$$

## 31.2.2 多脉冲模式

图31-3说明了多脉冲工作模式。输入信号的计算与单脉冲模式类似，主要区别在于ATxPHS寄存器对何时复位。

周期计数器锁存在ATxPER寄存器对中，并在每次输入脉冲（缺失脉冲后紧跟的脉冲除外）时复位。缺失脉冲后的第一个有效脉冲触发以下各项:

- 周期时钟输出
- PERIF中断
- 相位计数器复位

结果是周期时钟输出的周期长度等于相邻两缺失脉冲（例如，齿轮上缺失的齿）间的时间间隔。这会导致ATxRES与最大相位计数ATxPHS之间的关系发生显著变化，如公式31-6所示。

### 公式31-6:

$$ATxPHS(final) = ATxRES \left( \frac{MissP}{PulseP} \right)$$

# PIC16(L)F1615/9

---

公式31-6中的变量如下：

- MissP 是相邻两缺失脉冲间的时间间隔
- PulseP 是相邻两输入脉冲间的时间间隔
- ATxPHS(final) 是相位计数器的最大值

这将导致相位时钟输出在每次输入脉冲时产生ATxRES+1个脉冲，相位计数器在相邻两缺失脉冲间的整段时间内从0递增至ATxPHS(final)。

与单脉冲模式类似，这允许在单个信号周期中按固定相位角触发事件（其中，周期定义为相邻两缺失脉冲间的时间间隔）。图31-5的时序图中说明了多脉冲工作示例，还演示了输入信号周期变化所导致的结果。

**图31-1: 角度定时器简化框图（单脉冲模式）**

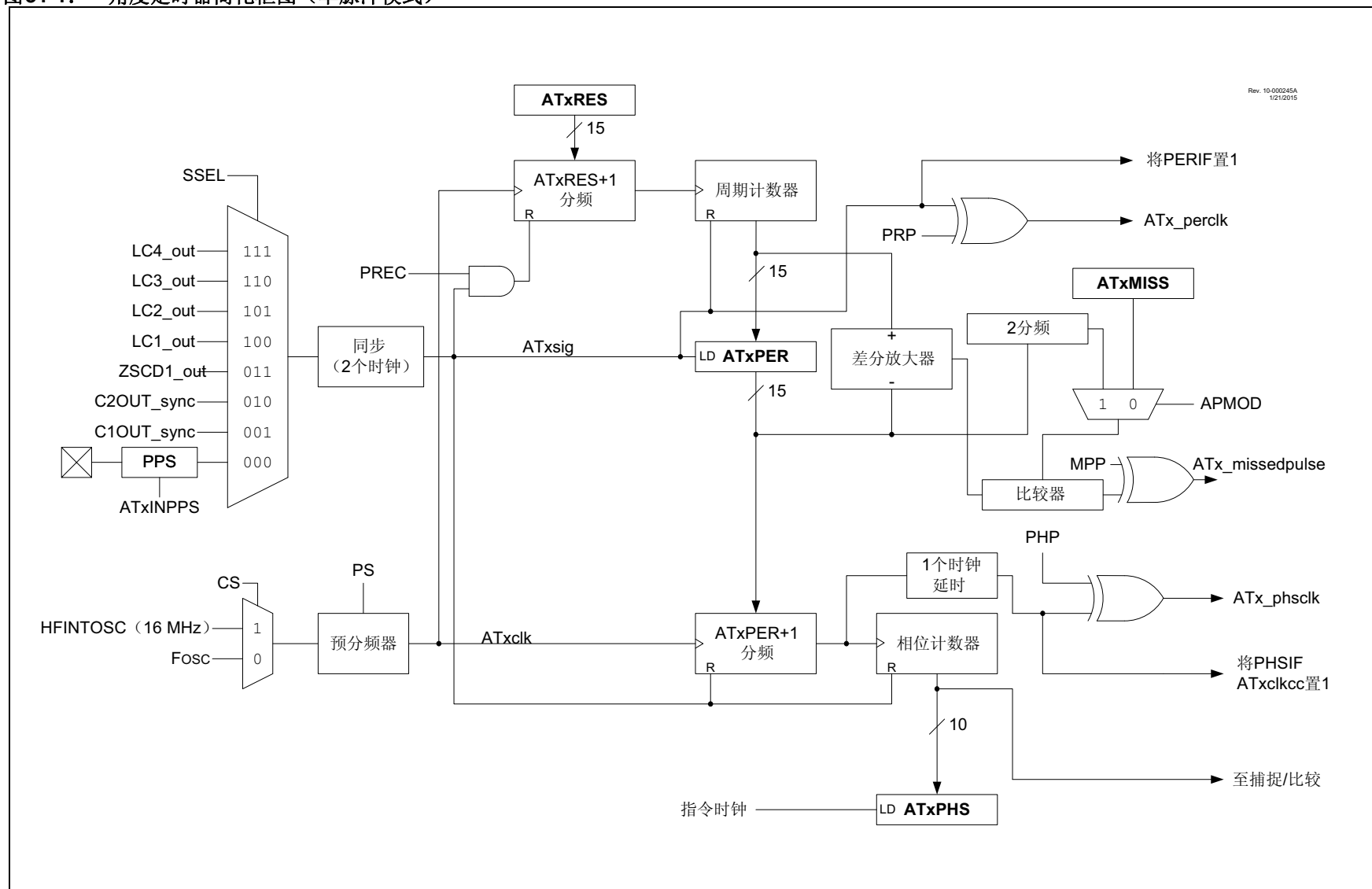
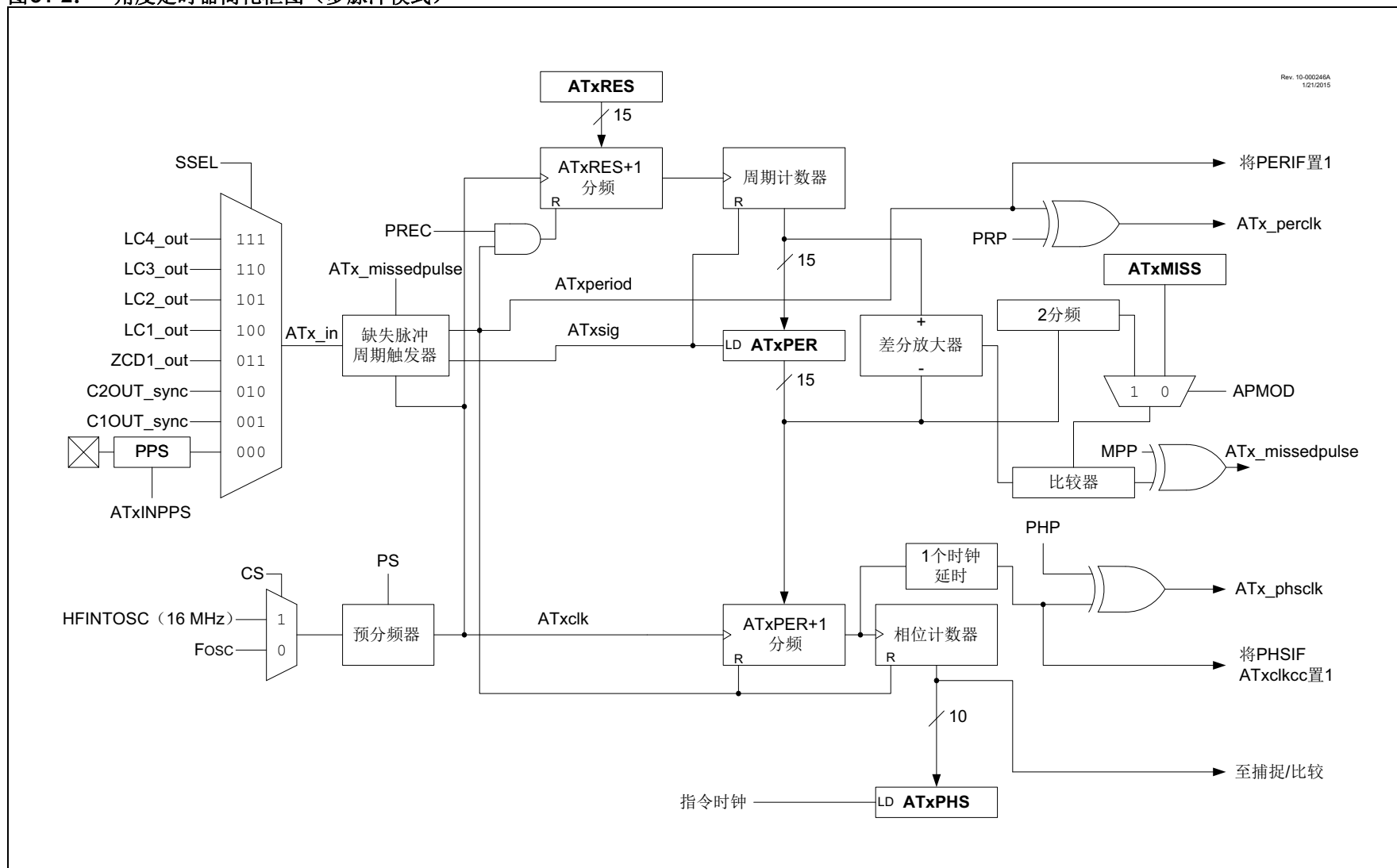


图31-2: 角度定时器简化框图 (多脉冲模式)



Rev. 10-000247A  
7/25/2014

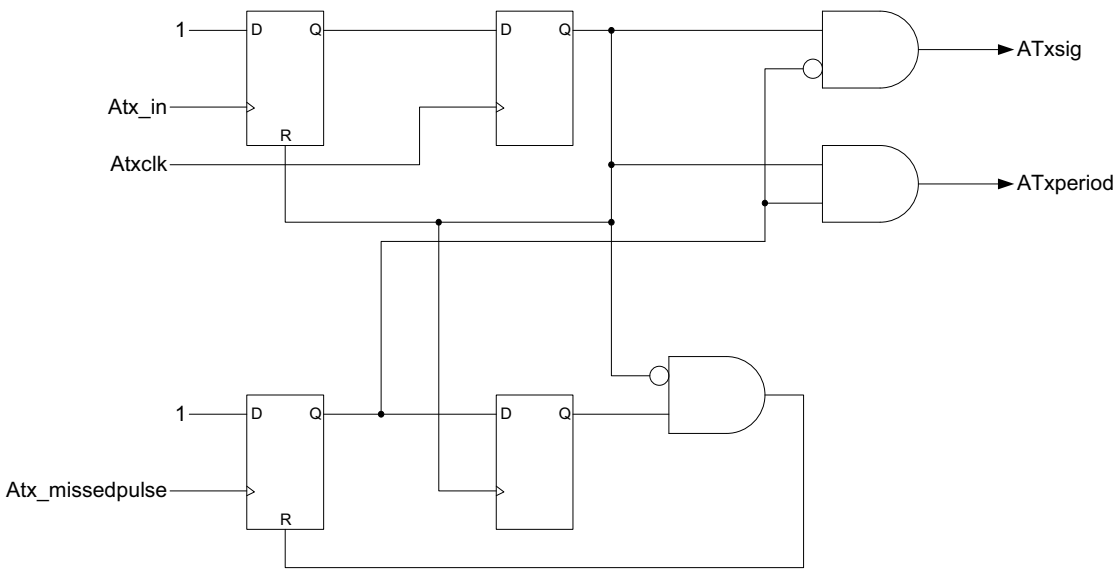


图31-3: 角度定时器多脉冲周期触发器简化框图

## 31.2.3 缺失脉冲检测

在单脉冲模式和多脉冲模式下，AT模块通过以下方式监视缺失脉冲。随着周期计数器递增计数，将不断从周期计数器的值中减去ATxPER寄存器对的锁存值。该减法运算的结果将与第三个值进行比较，比较结果匹配时将产生缺失脉冲事件。

第三个值是ATxMISS寄存器对的值或ATxPER寄存器对经2分频后的值。ATxCON0寄存器（寄存器31-1）的APMOD位选择使用上述两个值中的哪一个。

在单脉冲模式下，缺失脉冲事件会产生模块的缺失脉冲输出并会触发MISSIF中断。

在多脉冲模式下，缺失脉冲事件会产生输出和中断，并且还用于确定周期信号时序。

## 31.2.4 缺失脉冲模式

缺失脉冲检测有两种工作模式，可通过ATxCON0寄存器的APMOD位来选择：

- 自适应
- 固定

### 31.2.4.1 自适应缺失脉冲模式

当APMOD = 1时，缺失脉冲检测处于自适应模式。在自适应模式下，周期计数器与ATxPER锁存值的差值与经2分频后的ATxPER锁存值进行比较。如果在一定时间内（即，前一个脉冲时间间隔的1.5倍）未检测到输入信号脉冲，则将发生缺失脉冲事件。如果信号输入周期发生变化，则缺失脉冲比较会适当调整以适应这一变化，从而使缺失脉冲事件的相对时间保持为前一个脉冲时间间隔的1.5倍。

#### 31.2.4.2 固定缺失脉冲模式

当APMOD = 0时，缺失脉冲检测处于固定模式。在固定模式下，周期计数器与ATxPER锁存值的差值与ATxMISS寄存器对中的值进行比较。这样，用户便可绝对控制何时检测缺失脉冲，但代价是不能适当调整以适应周期的变化。

图31-4： 单脉冲模式的时序图

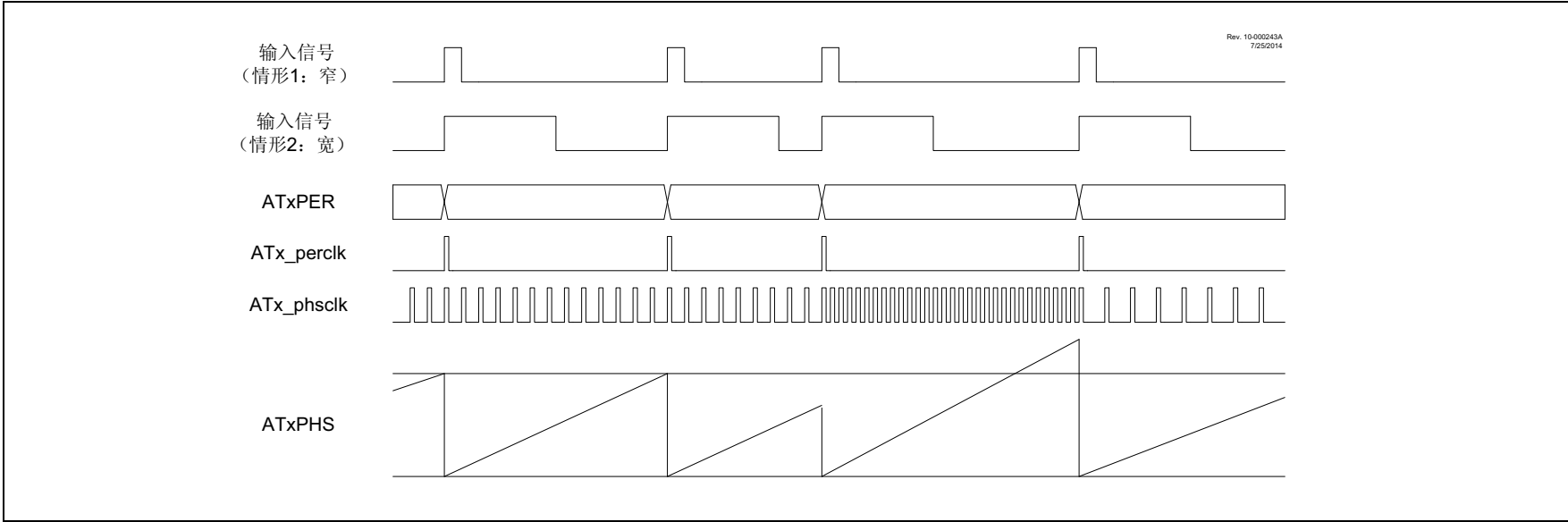
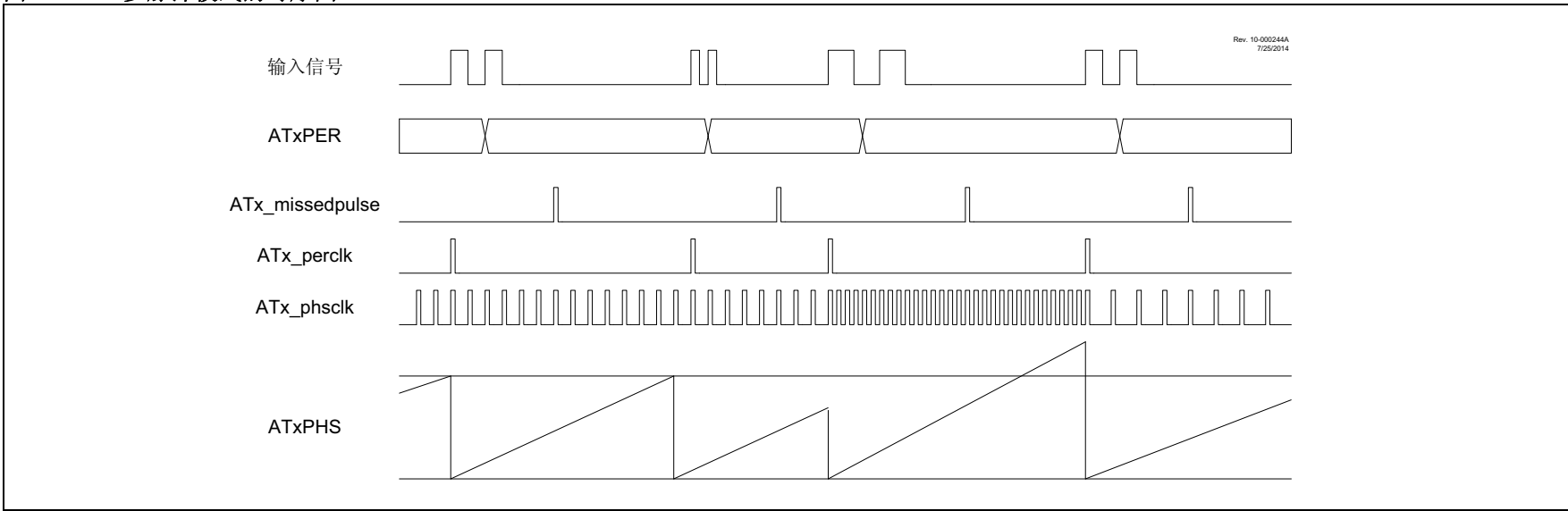


图31-5： 多脉冲模式的时序图



## 31.2.5 VALID 位

AT 模块使用的几个值必须根据外部信号计算。因此，这些值在角度定时器启动后的一段时间内可能不准确。为此，在模块使能后的一段时间内或者存在某些其他会影响输出值准确性的条件下，模块将不会输出信号或触发中断。该输出禁止状态由 ATxCON1 的只读位 VALID 清零来指示。

以下情况下将通过硬件清零 VALID 位：

- 对 ATxRES 寄存器对的任何写操作
- 相位计数器溢出（ATxPHS 寄存器对超过 0x3FF）
- 在线调试器暂停
- EN = 0
- ATxPER 寄存器对 = 0
- 器件复位

VALID 位清零时，将发生以下事件：

- 周期时钟不会输出，相关的中断不会触发。
- 缺失脉冲不会输出，相关的中断不会触发。
- 相位时钟不会输出，相关的中断不会触发。
- 相位计数器不会递增。
- 捕捉逻辑不起作用，相关的中断不会触发。
- 比较逻辑不起作用，相关的中断不会触发。
- 每个 ATxsig 边沿将周期计数器值锁存到 ATxPER 寄存器对，而与模式无关。

在单脉冲模式下，VALID 位在锁存 ATxPER 寄存器对的信号的第 3 个有效输入边沿置 1。在多脉冲模式下，还需要缺失脉冲触发，从而确保至少有 1 个完整的输入周期。

图 31-6 给出了单脉冲模式下 VALID 位的示例。

## 31.2.6 确定精度

ATxRES 寄存器对用于确定周期测量的分辨率，以及相位计数器在每个输入信号周期结束时达到的最大值。就电路的本质而言，用于导出相位计数器值的中间值 ATxPER 是一个整数。通过电路获得的整数与所计算的浮点值之比就是测量的固有误差。当 ATxRES 较小时，取整会产生较大的误差。导致误差较大的因素包括：

- ATxRES 值较大
- ATxclk 频率相对较低
- ATxsig 输入频率相对较高

实际误差可通过公式 31-7 来确定。

### 公式 31-7:

$$\text{周期} = \frac{F(ATxclk)}{F(ATxsig) \cdot (ATxRES + 1)}$$

$$\text{误差}(\%) = 100 \cdot \left( \frac{\text{周期} - \text{int}(\text{周期} + 1)}{\text{周期}} \right)$$

## 31.3 输入和时钟选择

AT 模块的输入时钟可来自 Fosc 系统时钟或 16 MHz HFINTOSC，具体通过 ATxCLK 寄存器的 CS0 位来选择。此外，时钟可进行预分频（/1、/2、/4 或 /8），具体通过 ATxCON0 寄存器的 PS<1:0> 位来配置。该预分频后的时钟随后用于角度定时器的所有时钟操作，因此应当用于上述所有确定角度定时器行为的公式。

AT 模块的输入信号可来自不同的信号源。信号源通过 ATxSIG 寄存器（寄存器 31-4）的 SSEL 位来选择。

## 31.4 模块输出

### 31.4.1 角度/相位时钟输出

角度/相位时钟信号（ATx\_phsclk）可供 CLC 用作组合逻辑的输入信号。该信号的极性通过 ATxCON1 寄存器的 PHP 位来配置。

### 31.4.2 周期时钟输出

周期时钟信号（ATx\_perclk）可用作 Timer2/4/6 和信号测量模块的输入时钟，以及用作组合逻辑中 CLC 的输入信号。该信号的极性通过 ATxCON1 寄存器（寄存器 31-2）的 PRP 位来配置。

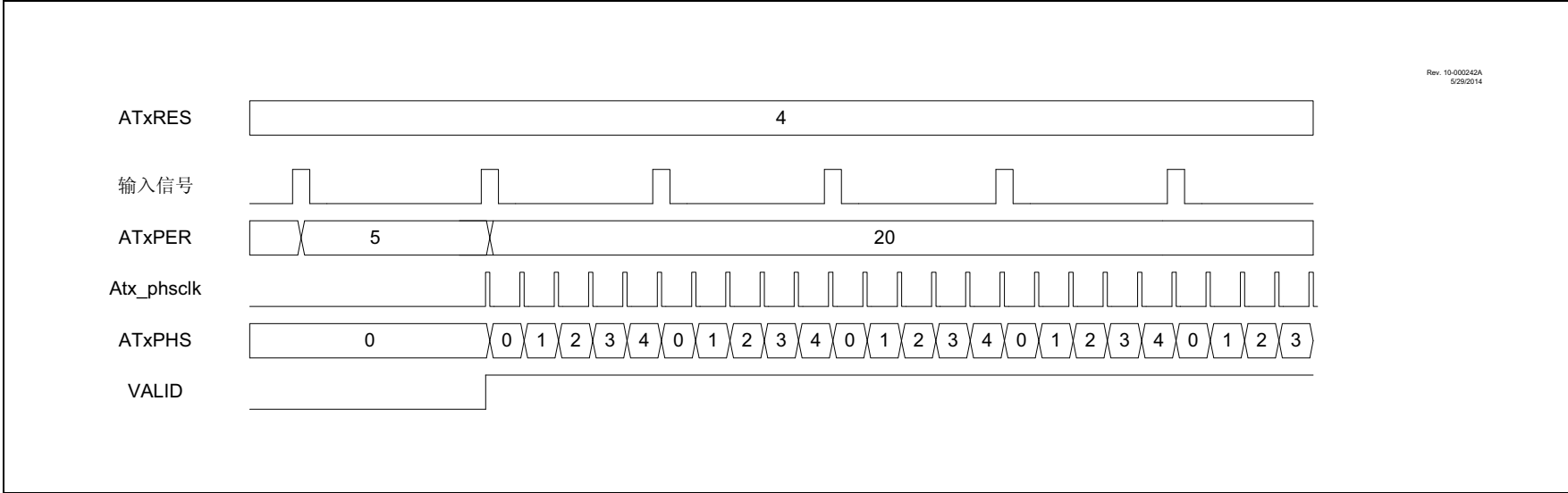
### 31.4.3 缺失脉冲输出

缺失脉冲信号（ATx\_missedpulse）可供 CLC 用作组合逻辑的输入信号。该信号的极性通过 ATxCON1 寄存器的 MPP 位来配置。



Rev. 10-000242A  
5/20/2014

图31-6: 操作示例 (ATxRES = 4)

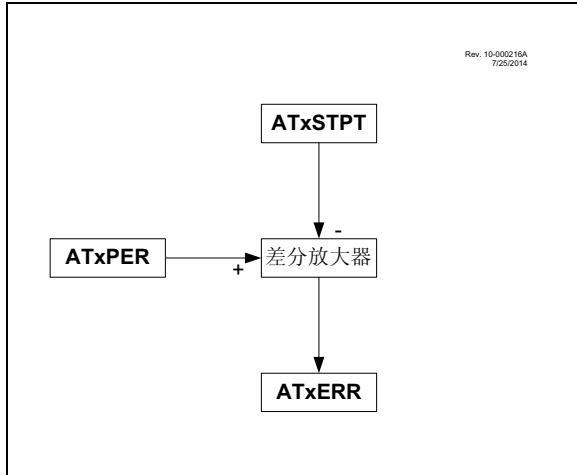


## 31.5 周期设定值和误差测量

ATxSTPT 寄存器对控制 AT 模块的周期设定值。信号周期在每个信号脉冲输入时捕捉到 ATxPER 寄存器对中。从 15 位无符号 ATxPER 值中减去 15 位无符号 ATxSTPT 值，将 16 位有符号结果存放到 ATxERR 寄存器对中。

ATxSTPT 值是双重缓冲的，需要写入 ATxSTPTL 值后才能使 ATxSTPTH 值生效。这样可以同时更新全部 16 位，以便正确计算误差。

图31-7： 角度定时器设定值计算框图



捕捉事件还会产生一个脉冲，该脉冲可用于以下各项：

- 触发 ADC 读操作
- CLC 逻辑输入
- 将 CCyIF 位置 1

有关 AT 模块触发的中断的更多详细信息，请参见第 31.7 节“中断”。

捕捉输入信号源通过捕捉/比较模块的相应 ATxCSELY 寄存器（寄存器 31-22）来选择，其极性通过 ATxCCONy 寄存器（寄存器 31-21）的 ATxCAPyP 位来选择。请注意，当处于捕捉模式时，ATxCCy 寄存器对是只读的。

### 31.6.2 比较模式

当（ATxCCONy 寄存器的）CCyMODE 位 = 0 时，选择比较模式。请参见图 31-9。

在比较模式下，模块会将 ATxCCy 寄存器对中的当前值与相位计数器值进行比较。当这两个值相等时，将产生并输出一个比较事件，以用于以下各项：

- 触发 ADC 读操作
- CLC 逻辑输入
- 将 CCyIF 位置 1

有关 AT 模块触发的中断的更多详细信息，请参见第 31.7 节“中断”。

## 31.6 捕捉和比较功能

角度定时器包含多个内置的捕捉/比较模块。这些模块通过其各自的 ATxCCONy 寄存器来控制，其中“x”是指 AT 实例，“y”是指相应 AT 模块内的捕捉/比较实例。

该特定器件的 AT 模块中包含 3 个捕捉/比较模块。ATxCCONy 寄存器的 CCyMODE 位控制每个特定模块处于捕捉模式还是比较模式。每个模块各自的输出信号的极性通过 ATxCCONy 寄存器（寄存器 31-21）的 CCyPOL 位来控制。捕捉模式和比较模式均使用 ATxclk 信号的边沿检测。

### 31.6.1 捕捉模式

当（ATxCCONy 寄存器的）CCyMODE 位 = 1 时，选择捕捉模式。请参见图 31-8。

在捕捉模式下，相位计数器的值在捕捉输入信号的上升沿写入相应的 ATxCCy 寄存器。

图 31-8： 角度定时器捕捉/比较单元框图：捕捉模式

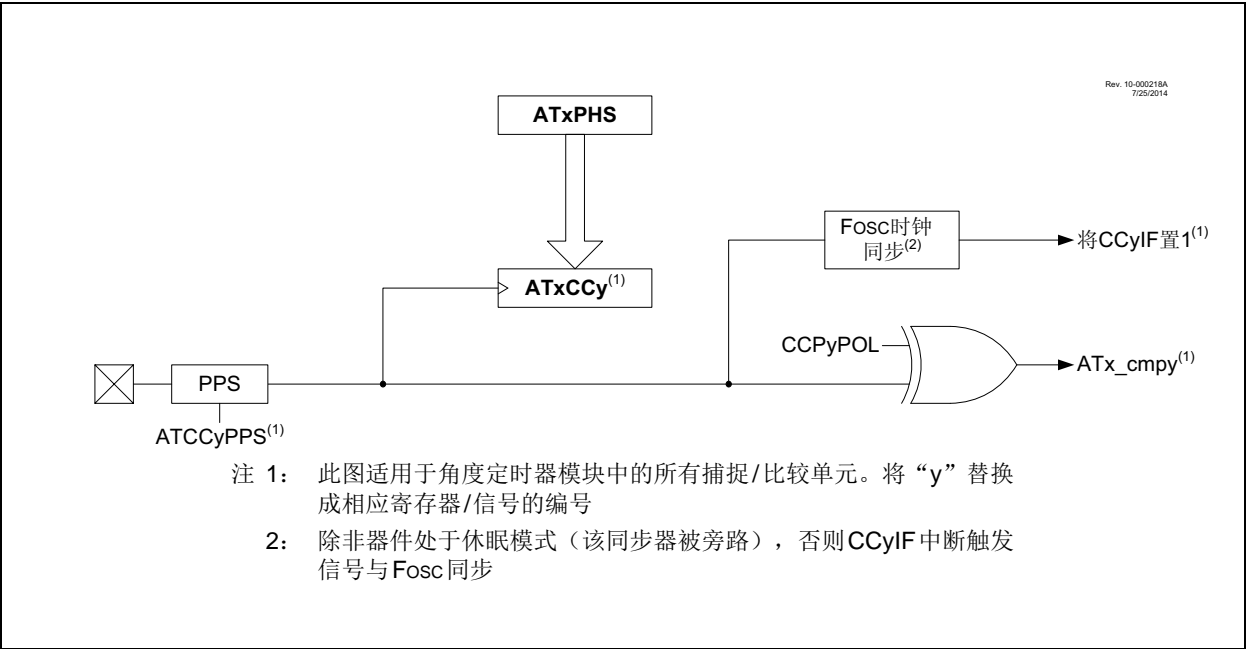
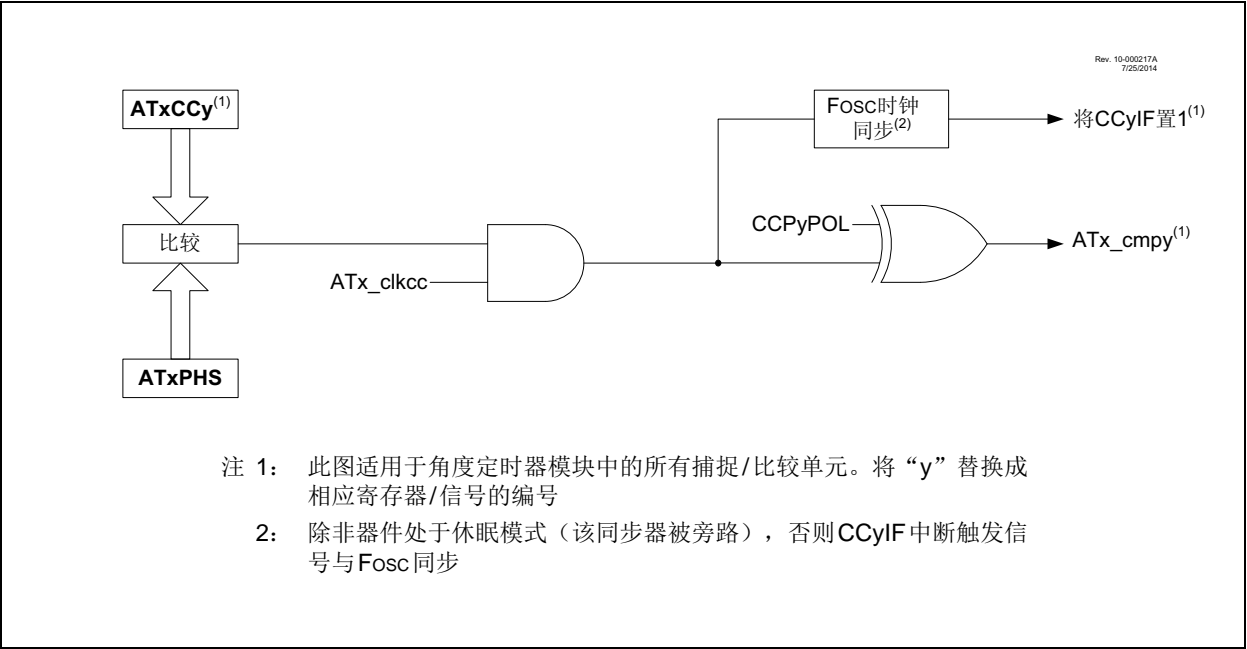


图 31-9： 角度定时器捕捉/比较单元框图：比较模式



## 31.7 中断

角度定时器及其捕捉/比较功能可产生多个中断条件。为适应所有这些中断源，模块拥有自己的中断逻辑结构，类似于单片机的中断逻辑结构。角度定时器中断通过ATxIE0寄存器（寄存器31-13）来允许，其相应的标志位于ATxIR0寄存器（寄存器31-14）中。捕捉/比较中断通过ATxIE1寄存器（寄存器31-15）来允许，其相应的标志位于ATxIR1寄存器（寄存器31-16）中。所有中断源共用一个角度定时器中断标志位，即PIR5寄存器（寄存器7-11）中的ATxIF位。这意味着触发中断后，ATxIR0和ATxIR1寄存器位将指示中断触发源。这也意味着为了通过特定中断产生单片机中断，PIE寄存器的ATxIE位以及ATxIE0或ATxIE1中的所需允许位都必须置1。

**注：** 由于角度定时器中断的性质，PIR5寄存器的ATxIF标志位是只读的。

### 31.7.1 角度定时器周期中断

当AT模块检测到周期事件时触发该中断。在单脉冲模式下，周期事件在每个输入信号边沿发生。在多脉冲模式下，周期事件在缺失脉冲后的输入信号边沿发生。周期中断的产生与定时器周期时钟输出上的脉冲匹配。它通过ATxIE0寄存器的ATPERIE位来允许，其状态由ATxIR0寄存器的PERIF位指示。

### 31.7.2 角度定时器相位时钟中断

该中断在定时器相位时钟输出的每个脉冲上触发。它通过ATxIE0寄存器的ATPHIE位来允许，其状态由ATxIR0寄存器的PHSIF位指示。

### 31.7.3 角度定时器缺失脉冲中断

该中断在输出缺失脉冲检测信号时被触发。更多信息，请参见第31.2.3节“缺失脉冲检测”。该中断通过ATxIE0寄存器的ATMISSIE位来允许，其状态由ATxIR0寄存器的ATMISSIF位指示。

### 31.7.4 角度定时器捕捉/比较中断

捕捉和比较中断通过模块的捕捉/比较功能来触发。如果配置为捕捉模式，则中断将在捕捉信号将相位计数器的值成功锁存到捕捉寄存器中后触发。如果配置为比较模式，则中断将在检测到比较寄存器中存放的值与相位计数器的值匹配时触发。这些中断分别通过ATxIE1寄存器的CC1IE、CC2IE和CC3IE位来控制，同样地，其状态分别由ATxIR1寄存器的CC1IF、CC2IF和CC3IF位指示。

### 31.8 角度定时器控制寄存器

表31-1给出了角度定时器外设的长位名称前缀。更多信息，请参见第1.1节“寄存器和位的命名约定”。

表31-1:

外设	位名称前缀
AT1	AT1

寄存器31-1: **ATxCON0: 角度定时器控制0寄存器**

R/W-0/0	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
EN	PREC	PS<1:0>	POL	—	—	APMOD	MODE
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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	<b>EN:</b> 角度定时器使能位 1 = 使能角度定时器; 内部时钟工作 0 = 禁止角度定时器
bit 6	<b>PREC:</b> 周期精度控制位 1 = 周期预分频比在每个周期开始时复位 0 = 周期预分频比在每个周期开始时不复位; 部分周期会影响下一个周期测量
bit 5-4	<b>PS&lt;1:0&gt;:</b> 时钟预分频比控制位 11 = 分辨率和相位计数器预分频比逻辑的时钟由ATxCLK/8提供 10 = 分辨率和相位计数器预分频比逻辑的时钟由ATxCLK/4提供 01 = 分辨率和相位计数器预分频比逻辑的时钟由ATxCLK/2提供 00 = 分辨率和相位计数器预分频比逻辑的时钟由ATxCLK提供
bit 3	<b>POL:</b> ATxsig有效边沿(极性)选择位 1 = ATxsig的下降沿为有效边沿 0 = ATxsig的上升沿为有效边沿
bit 2	<b>未实现:</b> 读为0
bit 1	<b>APMOD:</b> 自适应缺失脉冲模式选择位 1 = 自适应缺失脉冲模式。当在1.5倍的ATxPER内未检测到脉冲时, 检测到缺失脉冲 0 = 固定缺失脉冲模式。ATxMISS寄存器对确定缺失脉冲事件。
bit 0	<b>MODE:</b> 角度定时器模式选择位 1 = 角度定时器处于多脉冲模式(输入信号的周期由缺失脉冲定义) 0 = 角度定时器处于单脉冲模式(输入信号的周期由输入脉冲定义)

# PIC16(L)F1615/9

寄存器 31-2:      **ATxCON1: 角度定时器控制1 寄存器**

U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R-0/0	R-0/0
—	PHP	—	PRP	—	MPP	ACCS	VALID
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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	<b>未实现:</b> 读为0
bit 6	<b>PHP:</b> 相位时钟输出极性位 1 = 相位时钟输出为低电平有效 0 = 相位时钟输出为高电平有效
bit 5	<b>未实现:</b> 读为0
bit 4	<b>PRP:</b> 周期时钟输出极性位 1 = 周期时钟输出为低电平有效 0 = 周期时钟输出为高电平有效
bit 3	<b>未实现:</b> 读为0
bit 2	<b>MPP:</b> 缺失脉冲输出极性位 1 = 缺失脉冲输出为低电平有效 0 = 缺失脉冲输出为高电平有效
bit 1	<b>ACCS:</b> 加速度符号位 1 = ATxPER中的当前值小于前一个值 0 = ATxPER中的当前值大于或等于前一个值
bit 0	<b>VALID:</b> 有效测量位 1 = 出现了足够多的输入周期，能够使ATxPER和ATxPHS有效。 0 = ATxPER和ATxPHS中的值无效；未出现足够多的输入周期

寄存器 31-3:      ATxCLK: 角度定时器时钟选择寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-x/x
—	—	—	—	—	—	—	CS0
bit 7							bit 0

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

bit 7-1      未实现: 读为0  
bit 0      **CS0:** 角度定时器时钟选择位  
            1 = HFINTOSC 16 MHz  
            0 = FOSC

寄存器 31-4:      ATxSIG: 角度定时器输入信号选择寄存器

U-0	U-0	U-0	U-0	U-0	R/W-x/x	R/W-x/x	R/W-x/x
—	—	—	—	—	SSEL<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 7-3      未实现: 读为0  
bit 2-0      **SSEL<2:0>:** 角度输入信号选择位  
            111 = LC4\_out  
            110 = LC3\_out  
            101 = LC2\_out  
            100 = LC1\_out  
            011 = ZCD1\_out  
            010 = cmp2\_sync  
            001 = cmp1\_sync  
            000 = ATxINPPS

# PIC16(L)F1615/9

## 寄存器 31-5: ATxRESH: 角度定时器分辨率高字节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u
—	—	—	—	—	—	RES<9:8>	
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7-2 未实现: 读为0

bit 1-0 **RES<9:8>**: ATxRES 最高有效位, 相位计数器分辨率

**注 1:** 写入该寄存器将复位 ATxCON1 ([寄存器 31-2](#)) 的 VALID 位; 至少 2 个输入周期内将禁止输出信号。

**2:** 该寄存器不保证原子访问, 只应在定时器未运行时访问。

## 寄存器 31-6: ATxRESL: 角度定时器分辨率低字节寄存器

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
RES<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

q = 值取决于具体条件

bit 7-0 **RES<7:0>**: ATxRES 最低有效位, 相位计数器分辨率

**注 1:** 写入该寄存器将复位 ATxCON1 ([寄存器 31-2](#)) 的 VALID 位; 至少 2 个输入周期内将禁止输出信号。

**2:** 该寄存器不保证原子访问, 只应在定时器未运行时访问。



寄存器 31-7:      **ATxMISSH: 角度定时器缺失脉冲延时高字节寄存器**

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
MISS<15:8> <sup>(1)</sup>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0      **MISS<15:8><sup>(1)</sup>:** ATxMISS 的最高有效位（二进制补码）。ATxMISS 基于当前计数器值与 ATxPER 锁存值的差值来定义缺失脉冲输出变为有效时的周期计数器值。

**注 1:** ATxMISSH 一直保持到 ATxMISSL 被写入。正确的 ATxMISS 写操作应是先写入 ATxMISSH, 再写入 ATxMISSL, 以确保正确写入值。

寄存器 31-8:      **ATxMISSL: 角度定时器缺失脉冲延时低字节寄存器**

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
MISS<7:0>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0      **MISS<7:0>:** ATxMISS 的最低有效位（二进制补码）。ATxMISS 基于当前计数器值与 ATxPER 锁存值的差值来定义缺失脉冲输出变为有效时的周期计数器值。

# PIC16(L)F1615/9

寄存器 31-9: ATxPERH: 角度定时器测量周期高字节寄存器

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
POV	PER<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	q = 值取决于具体条件

bit 7      **POV:** 周期计数器溢出位  
1 = 测量期间计数器计满返回1次或多次  
0 = ATxPER 所示的值有效

bit 6-0      **PER<14:8>:** ATxPER 的最高有效位。ATxPER 是周期计数器的测量周期值。

寄存器 31-10: ATxPERL: 角度定时器测量周期低字节寄存器

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
PER<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	q = 值取决于具体条件

bit 7-0      **PER<7:0>:** ATxPER 的最低有效位。ATxPER 是周期计数器的测量周期值。

寄存器 31-11:     **ATxPHSH: 角度定时器相位计数器高字节寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	R-x/x	R-x/x
—	—	—	—	—	—	PHS<9:8>	
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-2           **未实现:** 读为0  
bit 1-0           **PHS<9:8>:** ATxPHS 的最高有效位。ATxPHS 是相位计数器的瞬时值。

寄存器 31-12:     **ATxPHSL: 角度定时器相位计数器低字节寄存器**

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
PHS<7:0>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0           **PHS<7:0>:** ATxPHS 的最低有效位。ATxPHS 是相位计数器的瞬时值。

# PIC16(L)F1615/9

寄存器 31-13: ATxIE0: 角度定时器允许0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
—	—	—	—	—	PHSIE	MISSIE	PERIE
bit 7					bit 0		

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 7-3 未实现: 读为0

bit 2 **PHSIE**: 相位中断允许位

1 = 允许相位中断

0 = 禁止相位中断

bit 1 **MISSIE**: 缺失脉冲中断允许位

1 = 允许缺失脉冲中断

0 = 禁止缺失脉冲中断

bit 0 **PERIE**: 周期中断允许位

1 = 允许周期中断

0 = 禁止周期中断

寄存器 31-14: ATxIR0: 角度定时器中断标志0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
—	—	—	—	—	PHSIF	MISSIF	PERIF
bit 7					bit 0		

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 7-3 未实现: 读为0

bit 2 **PHSIF**: 相位中断标志位

1 = 发生了相位中断

0 = 未发生相位中断, 或已清除相位中断

bit 1 **MISSIF**: 缺失脉冲中断标志位

1 = 发生了缺失脉冲中断

0 = 未发生缺失脉冲中断, 或已清除缺失脉冲中断

bit 0 **PERIF**: 周期中断标志位

1 = 发生了周期中断

0 = 未发生周期中断, 或已清除周期中断

寄存器 31-15: ATxIE1: 角度定时器允许1 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
—	—	—	—	—	CC3IE	CC2IE	CC1IE
bit 7					bit 0		

图注:		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-3	未实现: 读为0
bit 2	<b>CC3IE:</b> 捕捉/比较中断3 允许位 <u>如果CC3MODE = 1 (捕捉)</u> 1 = 允许捕捉中断3 0 = 禁止捕捉中断3 <u>如果CC3MODE = 0 (比较)</u> 1 = 允许比较中断3 0 = 禁止比较中断3
bit 1	<b>CC2IE:</b> 捕捉/比较中断2 允许位 <u>如果CC2MODE = 1 (捕捉)</u> 1 = 允许捕捉中断2 0 = 禁止捕捉中断2 <u>如果CC2MODE = 0 (比较)</u> 1 = 允许比较中断2 0 = 禁止比较中断2
bit 0	<b>CC1IE:</b> 捕捉/比较中断1 允许位 <u>如果CC1MODE = 1 (捕捉)</u> 1 = 允许捕捉中断1 0 = 禁止捕捉中断1 <u>如果CC1MODE = 0 (比较)</u> 1 = 允许比较中断1 0 = 禁止比较中断1

# PIC16(L)F1615/9

寄存器 31-16: ATxIR1: 角度定时器中断标志 1 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
—	—	—	—	—	CC3IF	CC2IF	CC1IF
bit 7					bit 0		

图注:		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-3	未实现: 读为0
bit 2	<b>CC3IF:</b> 捕捉/比较中断3标志位 <u>如果CC3MODE = 1 (捕捉)</u> 1 = 发生了捕捉中断3; 捕捉的相位值位于ATxCC3中 0 = 未发生捕捉中断3, 或已清除捕捉中断3 <u>如果CC3MODE = 0 (比较)</u> 1 = 发生了比较中断3 0 = 未发生比较中断3, 或已清除比较中断3
bit 1	<b>CC2IF:</b> 捕捉/比较中断2标志位 <u>如果CC2MODE = 1 (捕捉)</u> 1 = 发生了捕捉中断2; 捕捉的相位值位于ATxCC2中 0 = 未发生捕捉中断2, 或已清除捕捉中断2 <u>如果CC2MODE = 0 (比较)</u> 1 = 发生了比较中断2 0 = 未发生比较中断2, 或已清除比较中断2
bit 0	<b>CC1IF:</b> 捕捉/比较中断1标志位 <u>如果CC1MODE = 1 (捕捉)</u> 1 = 发生了捕捉中断1; 捕捉的相位值位于ATxCC1中 0 = 未发生捕捉中断1, 或已清除捕捉中断1 <u>如果CC1MODE = 0 (比较)</u> 1 = 发生了比较中断1 0 = 未发生比较中断1, 或已清除比较中断1

寄存器 31-17:     **ATxSTPTH: 角度定时器设定值高字节寄存器<sup>(1)</sup>**

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	R/W-x/u
—	STPT<14:8>						
bit 7							bit 0

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

bit 7           **未实现:** 读为0  
bit 6-0         **STPT<14:8>:** 设定值最高有效位。ATxSTPT 确定与周期进行比较以计算误差的阈值设置。  
**注 1:** 写入ATxSTPTH的值是双重缓冲的。写入该寄存器的值一直保持到ATxSTPTL被写入后才锁存到寄存器中。

寄存器 31-18:     **ATxSTPTL: 角度定时器设定值低字节寄存器**

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
STPT<7:0>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0         **STPT<7:0>:** 设定值最低有效位。ATxSTPT 确定与周期进行比较以计算误差的阈值设置。

# PIC16(L)F1615/9

寄存器 31-19:     **ATxERRH: 角度定时器设定值误差值高字节寄存器**

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
ERR<15:8>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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 7-0           **ERR<15:8>:** ATxERR的最高有效位。ATxERR是测量周期值相比于阈值设置的误差，定义为 ATxPER-ATxSTPTP。

寄存器 31-20:     **ATxERRL: 角度定时器设定值误差值低字节寄存器**

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
ERR<7:0>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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 7-0           **ERR<7:0>:** ATxERR的最低有效位。ATxERR是测量周期值相比于阈值设置的误差，定义为 ATxPER-ATxSTPTP。



寄存器31-21: ATxCCONy: 角度定时器捕捉/比较控制1 寄存器

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
CCyEN	—	—	CCPyPOL	CAPyP	—	—	CCyMODE
bit 7							bit 0

图注:		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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	<b>CCyEN:</b> 捕捉/比较使能位 1 = 使能捕捉/比较逻辑 0 = 禁止捕捉/比较逻辑
bit 6-5	<b>未实现:</b> 读为0
bit 4	<b>CCyPOL:</b> 捕捉/比较输出极性位 <u>在捕捉模式下 (CCyMODE = 1) :</u> 1 = 当ATxCCy更新时, ATxCCOUT1 低电平有效 0 = 当ATxCCy更新时, ATxCCOUT1 高电平有效 <u>在比较模式下 (CCyMODE = 0) :</u> 1 = 当ATxPHS = ATxCCy时, ATxCCOUT1 低电平有效 0 = 当ATxPHS = ATxCCy时, ATxCCOUT1 高电平有效
bit 3	<b>CAPyP:</b> 捕捉输入极性位: <u>在捕捉模式下 (CCyMODE = 1) :</u> 1 = 在捕捉输入的下降沿 (通过ATxCSELy选择), 将相位计数器的值捕捉到ATxCC1中 0 = 在捕捉输入的上升沿 (通过ATxCSELy选择), 将相位计数器的值捕捉到ATxCC1中 <u>在比较模式下 (CCyMODE = 0) :</u> 该位被忽略。
bit 2-1	<b>未实现:</b> 读为0
bit 0	<b>CCyMODE:</b> 捕捉/比较模式选择位 1 = 捕捉/比较逻辑处于捕捉模式 0 = 捕捉/比较逻辑处于比较模式

# PIC16(L)F1615/9

寄存器 31-22:     ATxCSELy: 角度定时器捕捉输入选择y 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
—	—	—	—	—	CPyS<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 7-3     未实现: 读为0
- bit 2-0     **CPyS<2:0>**: 捕捉输入源选择位
- 111 = CWG\_interrupt
- 110 = LC4\_out
- 101 = LC3\_out
- 100 = LC2\_out
- 111 = LC1\_out
- 010 = cmp2\_sync
- 001 = cmp1\_sync
- 000 = ATxCAP 引脚

寄存器 31-23:     **ATxCCyH: 角度定时器捕捉/比较y 高字节寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	R/q-0/0	R/q-0/0
—	—	—	—	—	—	CCy<9:8>	
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-2       **未实现:** 读为0  
bit 1-0       **CCy<9:8>:** ATxCCy 最高有效位  
              在捕捉模式下 (CCyMODE = 1) (只读):  
              当发出捕捉输入信号时, ATxCCy 是 ATxPHS 的捕捉值。  
              在比较模式下 (CCyMODE = 0):  
              ATxCCy 是与 ATxPHS 的当前值进行比较以触发中断/输出脉冲的值。

**注 1:** 写入 ATxCCyH 的值是双重缓冲的。写入该寄存器的值一直保持到 ATxCCyL 被写入后才锁存到寄存器中。

寄存器 31-24:     **ATxCCyL: 角度定时器捕捉/比较y 低字节寄存器**

R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0	R/q-0/0
CCy<7:0>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0       **CCy<7:0>:** ATxCCy 最低有效位  
              在捕捉模式下 (CCyMODE = 1) (只读):  
              当发出捕捉输入信号时, ATxCCy 是 ATxPHS 的捕捉值。  
              在比较模式下 (CCyMODE = 0):  
              ATxCCy 是与 ATxPHS 的当前值进行比较以触发中断/输出脉冲的值。

# PIC16(L)F1615/9

表31-2: 与角度定时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
AT1CC1H	—	—	—	—	—	—	CC1<9:8>		487
AT1CC1L	CC1<7:0>								487
AT1CCON1	CC1EN	—	—	CC1POL	CAP1P	—	—	CC1MODE	485
AT1CCON2	CC2EN	—	—	CC2POL	CAP2P	—	—	CC2MODE	485
AT1CCON3	CC3EN	—	—	CC3POL	CAP3P	—	—	CC3MODE	485
AT1CLK	—	—	—	—	—	—	—	CS0	475
AT1CON0	EN	PREC	PS<1:0>		POL	—	APMOD	MODE	473
AT1CON1	—	PHP	—	PRP	—	MPP	ACCS	VALID	474
AT1CSEL1	—	—	—	—	—	CP1S<2:0>			486
AT1CSEL2	—	—	—	—	—	CP2S<2:0>			486
AT1CSEL3	—	—	—	—	—	CP3S<2:0>			486
AT1ERRH	ERR<15:8>								484
AT1ERRL	ERR<7:0>								484
AT1IE0	—	—	—	—	—	PHSIE	MISSIE	PERIE	480
AT1IR0	—	—	—	—	—	PHSIF	MISSIF	PERIF	480
AT1IE1	—	—	—	—	—	CC3IE	CC2IE	CC1IE	481
AT1IR1	—	—	—	—	—	CC3IF	CC2IF	CC1IF	482
AT1MISSH	MISS<15:8>								477
AT1MISSL	MISS<7:0>								477
AT1PERH	POV	PER<14:8>							478
AT1PERL	PER<7:0>								478
AT1PHSH	—	—	—	—	—	—	PHS<9:8>		479
AT1PHSL	PHS<7:0>								479
AT1RESH	—	—	—	—	—	—	RES<9:8>		476
AT1RESL	RES<7:0>								476
AT1SIG	—	—	—	—	—	SSEL<2:0>			475
AT1STPTH	—	STPT<14:8>							483
AT1STPTL	STPT<7:0>								483
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE	110
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF	115

图注: — = 未实现, 读为0。AT 模块不使用阴影单元。

## 32.0 带比例-积分-微分（PID）模块的数学加速器

数学加速器是一个数学模块，可以执行多种运算，最突出的是用作PID（比例-积分-微分）控制器。PID控制器是一种在系统中利用当前误差（比例）、当前误差与所有先前误差之和（积分）以及当前变化与先前变化之差（微分）来修正误差并提供稳定性的算法。它通过一系列迭代向系统提供反馈，利用当前误差以及先前误差来计算控制器的新输入。图32-1显示了两种PID模式的数据流。

该模块利用用户提供的系数以及乘法器和累加器来完成计算PID算法的任务。因此，也可以将此乘法器和累加器配置为快速高效地执行有符号和无符号乘加计算（无论是否有累加）。图32-2显示了这些模式的数据流。

此模块的特性包括：

- 有符号乘法器
- 35位有符号累加器
- PID控制器支持K1、K2、K3、系统误差和所需设定值的用户输入
- 完成和误差中断
- 多种用户模式，支持有或没有累加的PID以及多种乘法运算

## 32.1 PID模块设置汇总

可以将PID模块配置为PID控制器或相乘并累加模块。可以在4种模式下执行相乘并累加：

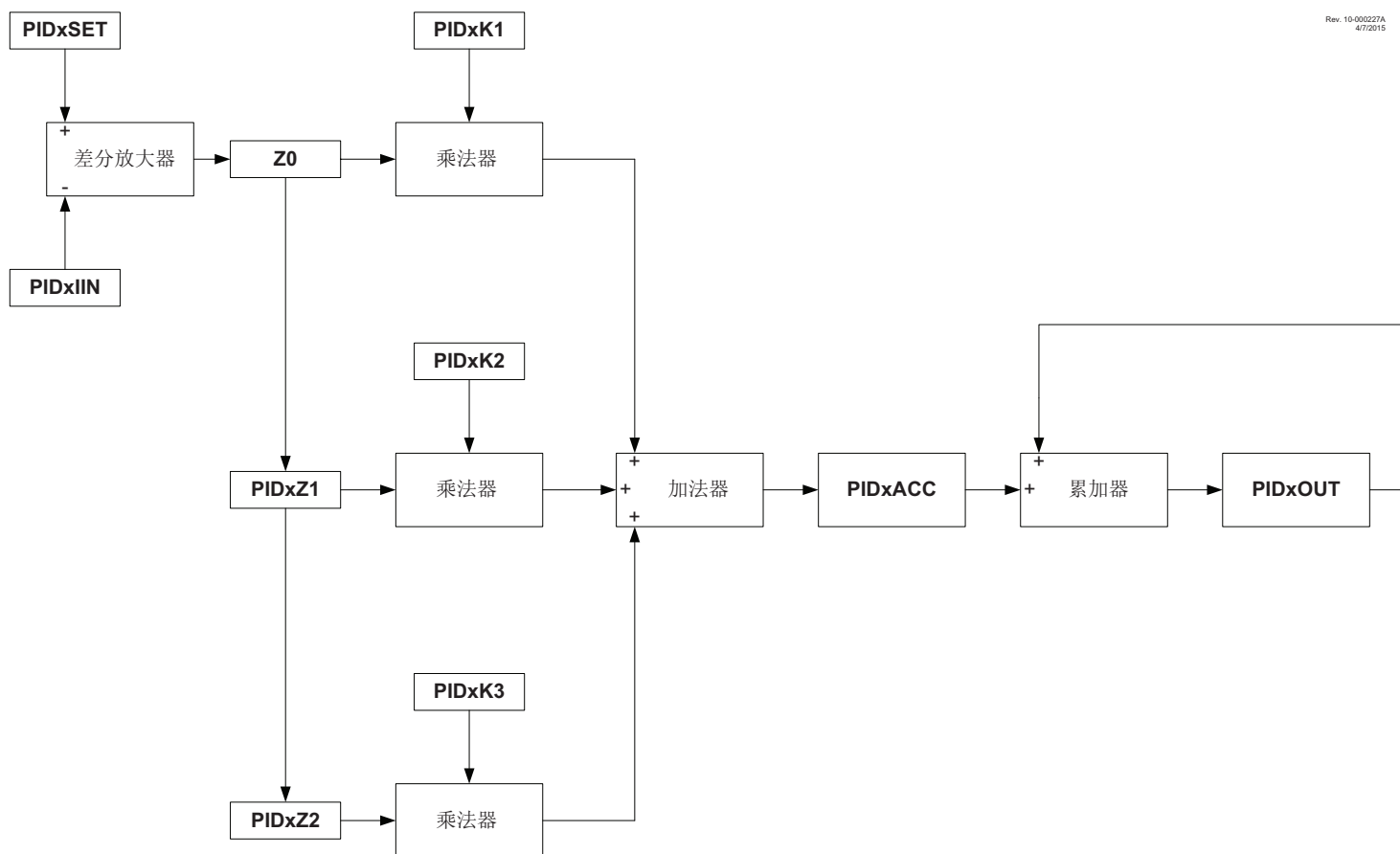
- 无符号相乘并相加，无累加
- 无符号相乘并累加
- 有符号相乘并相加，无累加
- 有符号相乘并累加

所有模式均由PIDxCON寄存器的MODE<2:0>位进行选择。

### 32.1.1 PID模式设置和操作

PIDxCON寄存器的MODE<2:0>位等于101时，模块处于PID控制器模式。模块在PID控制器模式下的操作通常循环执行。来自外部系统的输入将馈入控制器，控制器的输出将反馈给外部系统。这将从系统中产生新的响应，该响应随后会循环回PID控制器中。图32-1显示了PID操作的数据流。

图 32-1: PID 模块的基本数据流框图, PID 模式



注 1: 当 PIDxZ2 的结果与 PIDxK3 相乘之积添加到累加器之后, PIDxZ1 的当前值装载到 PIDxZ2 中。这同样适用于 PIDxZ1 和当前的 SET-IN 值。

在控制器内，从预编程的设定值中减去输入可得到误差值。该误差值与先前两个误差值（如果存在）分别乘以用户输入系数，这些乘法运算的结果加在一起构成输出。如果PIDxCON寄存器的MODE<2:0>位 = 101，则PID输出等于当前输出加上任意先前输出。

三个用户输入系数（K1、K2和K3）由三个经典PID系数Kp、Ki和Kd得出，并且必须在使用PID模块前进行计算。

1. 系数K1乘以当前误差（SET-IN）。它由如下公式定义：

**公式 32-1:**

$$K1 = Kp + Ki \cdot T + \frac{Kd}{T}$$

**注：** T为采样周期。

2. 系数K2乘以先前迭代的误差（Z1）。其中，T为采样周期，由如下公式定义：

**公式 32-2:**

$$K2 = -\left(Kp + \frac{2Kd}{T}\right)$$

**注：** T为采样周期。

3. 系数K3乘以在当前迭代（Z2）之前发生两次迭代的误差。它由如下公式定义：

**公式 32-3:**

$$K3 = \frac{Kd}{T}$$

**注：** T为采样周期。

要使模块在PID控制器模式下工作，请执行以下步骤：

1. 将PIDxCON寄存器的MODE<2:0>位设置为101，然后将PIDxCON寄存器的EN位置1。
2. 将先前计算的K1、K2和K3的值分别写入PIDxK1、PIDxK2和PIDxK3寄存器。
3. 将所需设定值写入PIDxSET寄存器，输入将与该设定值进行比较。
4. 将来自外部系统的值的高字节写入PIDxINH。然后将来自外部系统的值的低字节写入PIDxINL。这将开始计算并将PIDxCON寄存器的BUSY位置1。
5. 查询PIDxCON寄存器的BUSY位以检查其是否清零，或等待PIDxDIF中断触发（表示操作已完成）。
6. 读取PIDxOUT寄存器中的输出值。如果PID处于累加模式下，PIDxOUT将包含输出加上先前输出的累加，否则将只包含最新输出。
7. 为使PID正确工作，在将下一个输入应用到PID之前，需要将该输出应用到外部系统。这是为了确保在进行下次计算前，系统可以根据PID控制器的反馈进行调节。

**注：** 写PIDxINL后PIDxCON寄存器的BUSY位将立即变为高电平并一直保持此状态，直到完成所有计算。在BUSY位变为低电平前，PIDxOUT值一直无效，此时不应写与PID模块相关的任何寄存器，因为任何写操作都将中断计算。

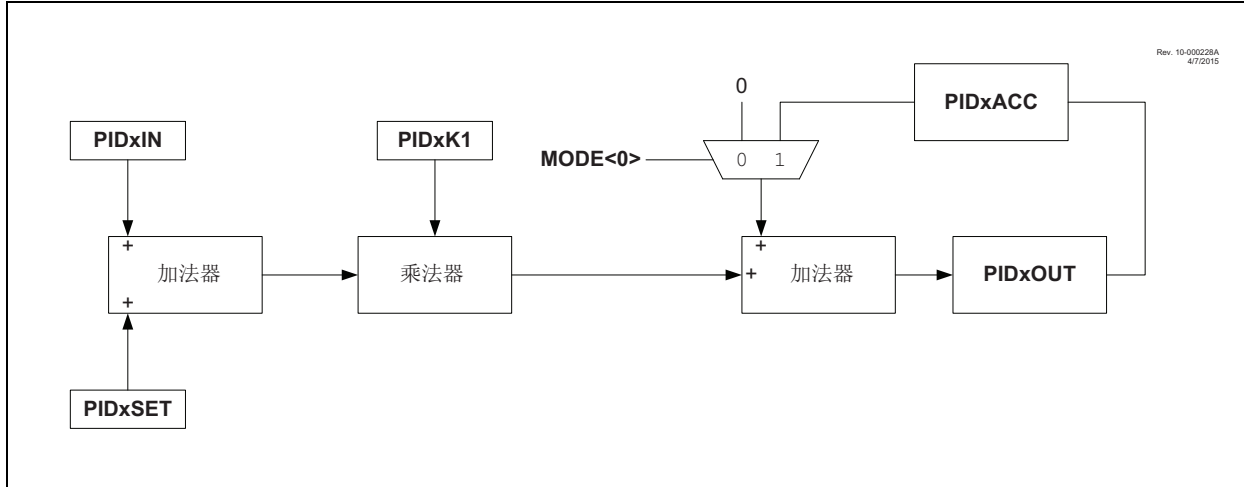
## 32.1.2 现场保护

可以在软件中保存PID控制器的当前状态并稍后将其恢复。要执行现场保护，当前不得存在活动的计算（BUSY = 0）。在存储器中的其他位置保存PIDxOUT、PIDxZ1和PIDxZ2，将保存PID控制器的当前状态，虽然也可能需要保存PIDxK1、PIDxK2、PIDxK3和/或PIDxSET，具体取决于应用。稍后，可在需要时将这些值写回各自对应的寄存器，最后写入PIDxINL，而PID将从先前状态继续。

## 32.2 相加并相乘模式设置和操作

PID 模块也可以用于执行 16 位相加并相乘计算。当 PIDxCON 寄存器的 MODE<2:0> 位等于 000、001、010 或 011 时，模块处于相加并相乘模式。图 32-2 显示了相加并相乘操作的数据流。

图 32-2: PID 模块的基本数据流框图，相加并相乘模式



所有相加并相乘模式均执行以下形式的操作。

### 公式 32-4:

$$\text{输出} = (A + B) \cdot C$$

注: A = PIDxIN、B = PICxSET 且 C = PIDxK1。

四种不同的相加并相乘模式包括:

- MODE<2:0> = 000: 输入为无符号，输出不累加
- MODE<2:0> = 001: 输入为无符号，输出累加先前输出
- MODE<2:0> = 010: 输入为有符号，输出不累加
- MODE<2:0> = 011: 输入为有符号，输出累加先前输出

要执行相加并相乘操作，请执行以下步骤:

1. 根据所需的计算形式，将 PIDxCON 寄存器的 MODE<2:0> 位设置为四种相加/相乘模式之一，然后将 PIDxCON 寄存器的 EN 位置 1。
2. 将 C 的值写入 PIDxK1H/L 寄存器对、B 的值写入 PIDxSETH/L 寄存器对，以及将 A 的高字节写入 PIDxINH 寄存器。
3. 最后，将 A 的低字节写入 PIDxINL 寄存器。这将开始数学运算并将 PIDxCON 寄存器的 BUSY 位置 1。
4. 查询 PIDxCON 寄存器的 BUSY 位以检查其是否清零，或等待 PIDxDIF 中断触发（表示操作已完成）。
5. 读取 PIDxOUT 寄存器中的计算结果。在累加模式下，PIDxOUT 寄存器将保留所有先前值加上当前计算值。在无累加模式下，PIDxOUT 寄存器将仅保留当前计算值。

这些模式也可以用于执行 16 位加法（通过将上述公式中的 C 项设置为 1）或 16 位乘法（通过将 A 或 B 设置为 0）。



## 32.3 中断

PID 模块具有 2 个中断，由 PIR5 寄存器中的中断标志 PIDxDIF 和 PIDxEIF 指示，并分别由 PIE5 寄存器中的中断控制位 PIDxDIE 和 PIDxEIE 控制。

当 PIDxCON 寄存器的 BUSY 位变为低电平时，将在成功完成计算后触发 PIDxDIF 中断。

当 PID 或相乘并相加计算中存在错误，尤其是输出值发生溢出错误时，会触发 PIDxEIF 中断。

## 32.4 处理错误溢出

如果计算导致 OUT 寄存器中的值溢出，上述寄存器中的该值将计满返回并触发 PIDxEIF 中断。对于 PID 计算，这表示该错误已超出 PID 修正系统误差的能力。在这种情况下，建议只要 PIDxEIF 中断设置为中断服务程序（ISR）的一部分，就用软件对 OUT 寄存器进行“饱和处理”，如例 32-1 所示。

### 例 32-1： 处理 PID 溢出

```
//Interrupt service routine
void interrupt ISR(void)

{
    IF (PIR5BITS.PID1EIF==1&&PIE5BITS.PID1EIE==1)
    {
        //saturate the PID1OUT registers
        PID1OUTH=0xFF;
        PID1OUTL=0xFF;
        PID1OUTLH=0xFF;
        PID1OUTLL=0xFF;
        PID1OUTH=0xFF;
        //clear the interrupt flag
        PIR5bits.PID1EIF=0;
    }
}
```

32.5 PID控制寄存器

表32-1给出了16位PID外设的长位名称前缀。更多信息，请参见第1.1节“寄存器和位的命名约定”。

表32-1:

外设	位名称前缀
PID1	PID1

寄存器32-1: PIDxCON: PID配置寄存器

R/W-0/0	R/HS/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	BUSY	—	—	—	MODE<2:0>		
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	-m/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	<b>EN:</b> PID模块使能位 1 = 使能PID模块 0 = 禁止PID模块
bit 6	<b>BUSY:</b> PID模块当前正在计算
bit 5-3	<b>未实现:</b> 读为0
bit 2-0	<b>MODE&lt;2:0&gt;:</b> PID模式控制位 11x = 保留。不要使用。 101 = PID输出是以二进制补码计数法表示的计算输出（当前误差加上累加的先前误差） 100 = 保留。不要使用。 011 = (IN<15:0>+SET<15:0>)*K1<15:0>二进制补码有符号输入，有累加 010 = (IN<15:0>+SET<15:0>)*K1<15:0>二进制补码有符号输入，无累加 001 = (IN<15:0>+SET<15:0>)*K1<15:0>无符号输入，有累加 000 = (IN<15:0>+SET<15:0>)*K1<15:0>无符号输入，无累加

寄存器 32-2:      **PIDxINH: PID 输入高字节寄存器**

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
IN<15:8>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-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 7-0      **IN<15:8>**: IN 的高 8 位。IN 是从控制系统到 PID 模块的 16 位输入。

寄存器 32-3:      **PIDxINL: PID 输入低字节寄存器**

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
IN<7:0>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-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 7-0      **IN<7:0>**: IN 的低 8 位。IN 是从控制系统到 PID 模块的 16 位输入。

寄存器 32-4:      **PIDxSETH: PID 设定值高字节寄存器**

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
SET<15:8>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-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 7-0      **SET<15:8>**: SET 的高 8 位。SET 是 16 位用户控制变量, 用于与来自控制系统的输入进行比较以确定系统中的误差

寄存器 32-5:      **PIDxSETL: PID 设定值低字节寄存器**

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
SET<7:0>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-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 7-0      **SET<7:0>**: SET 的低 8 位。SET 是 16 位用户控制变量, 用于与来自控制系统的输入进行比较以确定系统中的误差

# PIC16(L)F1615/9

## 寄存器 32-6:      PIDxK1H: PID K1 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
K1<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	q = 值取决于具体条件

bit 7-0      **K1<15:8>**: K1的高8位。K1是根据Kp + Ki + Kd计算得出的16位用户控制系数

## 寄存器 32-7:      PIDxK1L: PID K1 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
K1<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	q = 值取决于具体条件

bit 7-0      **K1<7:0>**: K1的低8位。K1是根据Kp + Ki + Kd计算得出的16位用户控制系数

## 寄存器 32-8:      PIDxK2H: PID K2 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
K2<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	q = 值取决于具体条件

bit 7-0      **K2<15:8>**: K2的高8位。K2是根据-(Kp + 2Kd)计算得出的16位用户控制系数

## 寄存器 32-9:      PIDxK2L: PID K2 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
K2<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	q = 值取决于具体条件

bit 7-0      **K2<7:0>**: K2的低8位。K2是根据-(Kp + 2Kd)计算得出的16位用户控制系数

寄存器 32-10:     **PIDxK3H: PID K3 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
K3<15:8>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0           **K3<15:8>**: K3 的高 8 位。K3 是根据 Kd 计算得出的 16 位用户控制系数

寄存器 32-11:     **PIDxK3L: PID K3 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
K3<7:0>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0           **K3<7:0>**: K3 的低 8 位。K3 是根据 Kd 计算得出的 16 位用户控制系数

# PIC16(L)F1615/9

寄存器 32-12:     **PIDxOUTU: PID 输出高字节寄存器**

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	OUT<35:32>			
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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           **OUT<35:32>:** OUT 的 bit <35:32>。OUT 是 PID 对指定输入完成指定计算后得到的输出值。

寄存器 32-13:     **PIDxOUTH: PID 输出高电平高字节寄存器**

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
OUT<31:24>							
bit 7				bit 0			

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位，读为0
u = 不变	x = 未知	-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 7-0           **OUT<31:24>:** OUT 的 bit <31:24>。OUT 是 PID 对指定输入完成指定计算后得到的输出值。

寄存器 32-14:     **PIDxOUTH: PID 输出高电平低字节寄存器**

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
OUT<23:16>							
bit 7				bit 0			

<b>图注:</b>							
R = 可读位	W = 可写位			U = 未实现位, 读为0			
u = 不变	x = 未知			-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 7-0           **OUT<23:16>:** OUT的bit <23:16>。OUT是PID对指定输入完成指定计算后得到的输出值。

寄存器 32-15:     **PIDxOUTLH: PID 输出低电平高字节寄存器**

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
OUT<15:8>							
bit 7				bit 0			

<b>图注:</b>							
R = 可读位	W = 可写位			U = 未实现位, 读为0			
u = 不变	x = 未知			-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 7-0           **OUT<15:8>:** OUT的bit <15:8>。OUT是PID对指定输入完成指定计算后得到的输出值。

寄存器 32-16:     **PIDxOUTLL: PID 输出低电平低字节寄存器**

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
OUT<7:0>							
bit 7				bit 0			

<b>图注:</b>							
R = 可读位	W = 可写位			U = 未实现位, 读为0			
u = 不变	x = 未知			-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 7-0           **OUT<7:0>:** OUT的bit <7:0>。OUT是PID对指定输入完成指定计算后得到的输出值。

# PIC16(L)F1615/9

寄存器 32-17: PIDxZ1U: PID Z1 高字节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	Z116
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7-1 未实现: 读为0

bit 0 Z116: Z1 的 bit 16。在 PID 模式下, Z1 是 PID 控制环先前迭代的误差 (IN 减去 SET) 值。

寄存器 32-18: PIDxZ1H: PID Z1 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
Z1<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

q = 值取决于具体条件

bit 7-0 Z1<15:8>: Z1 的 bit <15:8>。在 PID 模式下, Z1 是 PID 控制环先前迭代的误差 (IN 减去 SET) 值。

寄存器 32-19: PIDxZ1L: PID Z1 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
Z1<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

q = 值取决于具体条件

bit 7-0 Z1<7:0>: Z1 的 bit <7:0>。在 PID 模式下, Z1 是 PID 控制环先前迭代的误差 (IN 减去 SET) 值。



寄存器 32-20:     **PIDxZ2U: PID Z2 高字节寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	Z216
bit 7							bit 0

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

bit 7-1           **未实现:** 读为0  
bit 0           **Z216:** Z2的bit 16。在PID模式下, Z2是PID控制环先前迭代的误差(IN减去SET)值。

寄存器 32-21:     **PIDxZ2H: PID Z2 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
Z2<15:8>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0           **Z2<15:8>:** Z2的bit <15:8>。在PID模式下, Z2是PID控制环先前迭代的误差(IN减去SET)值。

寄存器 32-22:     **PIDxZ2L: PID Z2 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
Z2<7:0>							
bit 7							bit 0

<b>图注:</b>		
R = 可读位	W = 可写位	U = 未实现位, 读为0
u = 不变	x = 未知	-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 7-0           **Z2<7:0>:** Z2的bit <7:0>。在PID模式下, Z2是PID控制环先前迭代的误差(IN减去SET)值。

# PIC16(L)F1615/9

## 寄存器 32-23: PIDxACCU: PID 累加器最高字节寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	ACC<34:32>		
bit 7					bit 0		

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 7-3

未实现: 读为0

bit 2-0

**ACC<34:32>**: ACC 的 bit <34:32>。ACC 是累加器寄存器, 其中 PID 的所有乘法器结果都会在写入输出前进行累加。

## 寄存器 32-24: PIDxACCHH: PID 累加器高字节高位寄存器

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
ACC<31:24>							
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-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 7-0

**ACC<31:24>**: ACC 的 bit <31:24>。ACC 是累加器寄存器, 其中 PID 的所有乘法器结果都会在写入输出前进行累加。

## 寄存器 32-25: PIDxACCHL: PID 累加器高字节低位寄存器

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
ACC<23:16>							
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值/所有其他复位时的值

1 = 置1

0 = 清零

q = 值取决于具体条件

bit 7-0

**ACC<23:16>**: ACC 的 bit <23:16>。ACC 是累加器寄存器, 其中 PID 的所有乘法器结果都会在写入输出前进行累加。

寄存器 32-26:     **PIDxACCLH: PID 累加器低字节高位寄存器**

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
ACC<15:8>							
bit 7				bit 0			

<b>图注:</b>							
R = 可读位	W = 可写位			U = 未实现位, 读为0			
u = 不变	x = 未知			-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 7-0     **ACC<15:8>**: ACC的bit <15:8>。ACC是累加器寄存器, 其中PID的所有乘法器结果都会在写入输出前进行累加。

寄存器 32-27:     **PIDxACCLL: PID 累加器低字节低位寄存器**

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
ACC<7:0>							
bit 7				bit 0			

<b>图注:</b>							
R = 可读位	W = 可写位			U = 未实现位, 读为0			
u = 不变	x = 未知			-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 7-0     **ACC<7:0>**: ACC的bit <7:0>。ACC是累加器寄存器, 其中PID的所有乘法器结果都会在写入输出前进行累加。

# PIC16(L)F1615/9

表 32-2: 与 PID 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
PID1ACCU	—	—	—	—	—	ACC<34:32>			502
PID1ACCHH	ACC<31:24>								502
PID1ACCHL	ACC<23:16>								502
PID1ACCLH	ACC<15:8>								503
PID1ACCLL	ACC<7:0>								503
PID1CON	EN	BUSY	—	—	—	MODE<2:0>			494
PID1INH	IN<15:8>								495
PID1INL	IN<7:0>								495
PID1K1H	K1<15:8>								496
PID1K1L	K1<7:0>								496
PID1K2H	K2<15:8>								496
PID1K2L	K2<7:0>								496
PID1K3H	K3<15:8>								497
PID1K3L	K3<7:0>								497
PID1OUTU	—	—	—	—	OUT<35:32>				498
PID1OUTHH	OUT<31:24>								498
PID1OUTHL	OUT<23:16>								499
PID1OUTLH	OUT<15:8>								499
PID1OUTLL	OUT<7:0>								499
PID1SETH	SET<15:8>								495
PID1SETL	SET<7:0>								495
PID1Z1U	—	—	—	—	—	—	—	Z116	500
PID1Z1H	Z1<15:8>								500
PID1Z1L	Z1<7:0>								500
PID1Z2U	—	—	—	—	—	—	—	Z216	501
PID1Z2H	Z2<15:8>								501
PID1Z2L	Z2<7:0>								501
PIE5	TMR3GIE	TMR3IE	TMR5GIE	TMR5IE	—	AT1IE	PID1EIE	PID1DIE	110
PIR5	TMR3GIF	TMR3IF	TMR5GIF	TMR5IF	—	AT1IF	PID1EIF	PID1DIF	115

图注: — = 未实现, 读为 0。PID 模块不使用阴影单元。

## 33.0 在线串行编程（ICSP™）

ICSP 编程允许用户在生产电路板时使用未编程器件。编程可以在组装流程之后完成，从而可以使用最新版本的固件或者定制固件对器件编程。ICSP 编程需要5个引脚：

- ICSPCLK
- ICSPDAT
- 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“*PIC12(L)F1612/PIC16(L)F161X Memory Programming Specification*”（DS40001720）。

### 33.1 高电压编程进入模式

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进入模式。

### 33.2 低电压编程进入模式

通过低电压编程进入模式，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 编程进入模式。要禁止低电压 ICSP 模式，LVP 位必须编程为 0。

进入低电压编程进入模式需要执行以下步骤：

1.  $\overline{\text{MCLR}}$  电压设置为  $V_{IL}$ 。
2. 在提供 ICSPCLK 时钟的同时，在 ICSPDAT 上送出 32 位密钥序列。

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  $\overline{\text{MCLR}}$  保持为  $V_{IL}$ 。

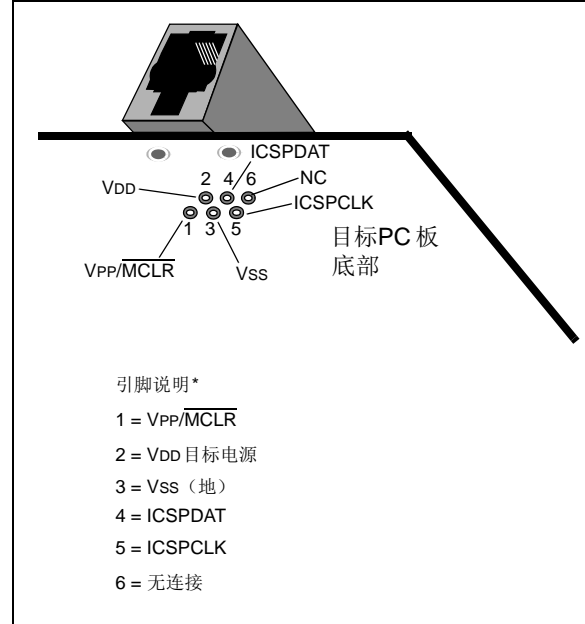
如果使能了低电压编程（LVP = 1），则  $\overline{\text{MCLR}}$  复位功能会被自动使能，无法禁止。更多信息，请参见第 6.5 节“**MCLR**”。

LVP 位只能通过使用高电压编程模式重新设定为 0。

## 33.3 常用编程接口

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 33-1。

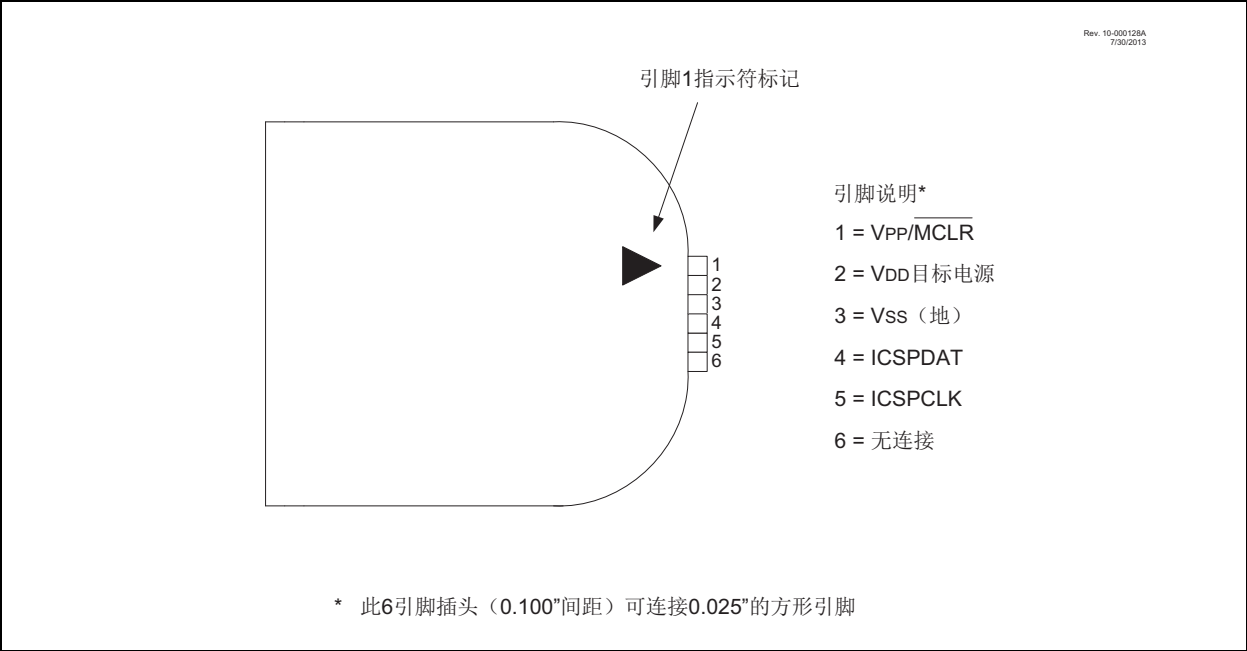
图 33-1： ICD RJ-11 型连接器接口



另一种常用于 PICkit™ 编程器的连接器是间距为 0.1 英寸的标准 6 引脚插头。请参见图 33-2。

# PIC16(L)F1615/9

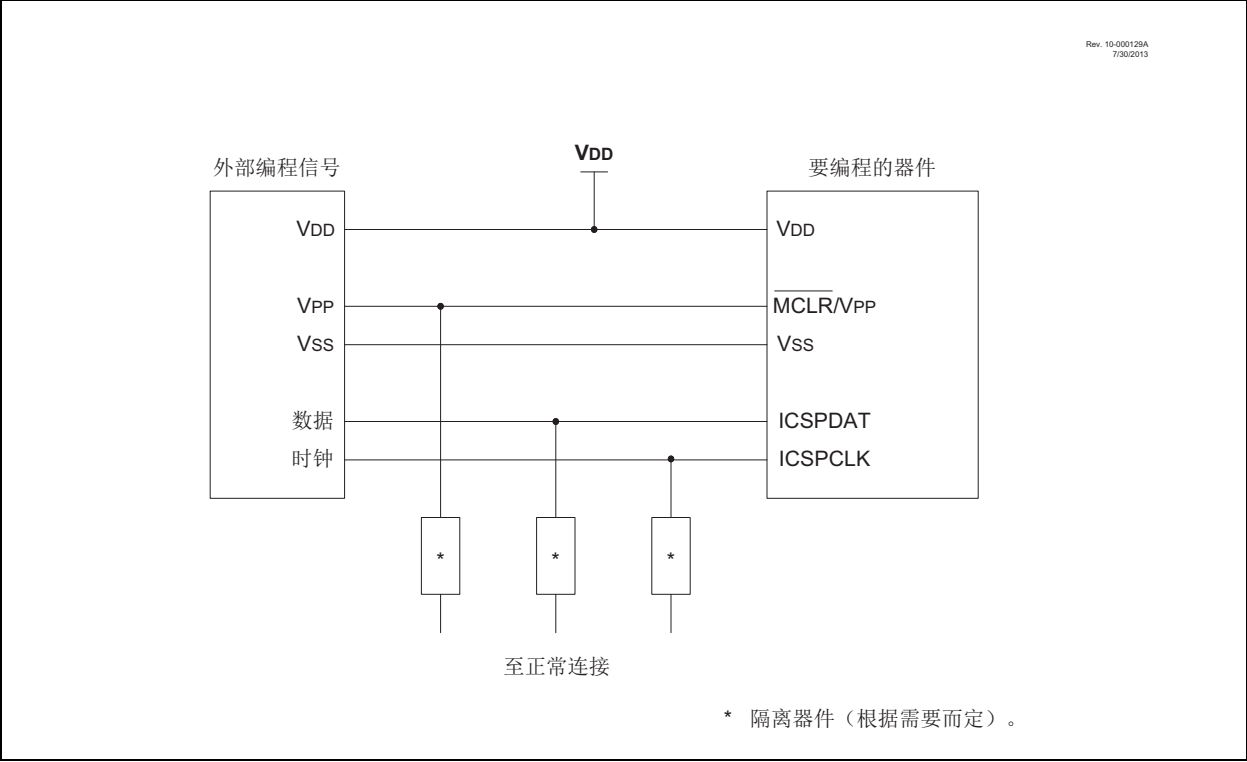
图33-2: PICKit™ 编程器型连接器接口



关于其他接口建议，请在进行PCB设计之前参见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33-3。

图33-3: ICSP™ 编程的典型连接



34.0 指令集汇总

每条指令都是一个包含操作码和所有必需操作数的14位字。操作码可以分为三大类。

- 面向字节的操作类指令
- 面向位的操作类指令
- 立即数和控制操作类指令

立即数和控制类指令字格式最为丰富。

表34-3列出了MPASM™汇编器可识别的指令。

除了以下指令（可能需要2或3个周期），所有指令都在单个指令周期内执行：

- 子程序指令需要两个周期（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、BTFSS、BTFSC、DECFSZ和INCSFZ）

- 当任意指令引用某个间接文件寄存器，并且文件选择寄存器指向程序存储器时，将需要使用一个额外的指令周期。

一个指令周期包含4个振荡器周期；振荡器频率为4 MHz时，得到的标称指令执行速率为1 MHz。

所有指令示例均使用格式“0xhh”来表示一个十六进制数，其中“h”表示一个十六进制数字。

34.1 读-修改-写操作

任何一条指定文件寄存器作为指令一部分的指令都进行读-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。即使指令写入该寄存器，还是会执行对寄存器的读操作。

表34-1： 操作码字段说明

字段	说明
f	文件寄存器地址（0x00至0x7F）
W	工作寄存器（累加器）
b	8位文件寄存器内的位地址
k	立即数字段、常数或标号
x	无关位（= 0或1）。 汇编器将产生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34-2： 缩写说明

字段	说明
PC	程序计数器
$\overline{TO}$	超时位
C	进位位
DC	半进位位
Z	全零标志位
$\overline{PD}$	掉电位

图34-1： 指令的通用格式

**面向字节的文件寄存器操作类指令**

138760

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

**面向位的文件寄存器操作类指令**

13109760

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

13870

操作码	k(立即数)
-----	--------

k = 8 位立即数的值

仅限 CALL 和 GOTO 指令

1311100

操作码	k(立即数)
-----	--------

k = 11 位立即数的值

仅限 MOVL P 指令

13760

操作码	k(立即数)
-----	--------

k = 7 位立即数的值

仅限 MOVLB 指令

13540

操作码	k(立即数)
-----	--------

k = 5 位立即数的值

仅限 BRA 指令

13980

操作码	k(立即数)
-----	--------

k = 9 位立即数的值

**FSR 偏移指令**

137650

操作码	n	k(立即数)
-----	---	--------

n = 相应的 FSR  
k = 6 位立即数的值

**FSR 递增指令**

133210

操作码	n	m(模式)
-----	---	-------

n = 相应的 FSR  
M = 2 位模式值

**仅限操作码**

130

操作码
-----



表34-3: 增强型中档指令集

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			
面向字节的文件寄存器操作类指令									
ADDWF	f, d	W和f相加	1	00	0111	dfff	ffff	C、DC和Z	2
ADDWFC	f, d	W和f相加（带进位）	1	11	1101	dfff	ffff	C、DC和Z	2
ANDWF	f, d	W和f作逻辑与运算	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	算术右移	1	11	0111	dfff	ffff	C和Z	2
LSLF	f, d	逻辑左移	1	11	0101	dfff	ffff	C和Z	2
LSRF	f, d	逻辑右移	1	11	0110	dfff	ffff	C和Z	2
CLRF	f	将f清零	1	00	0001	1fff	ffff	Z	2
CLRW	—	将W清零	1	00	0001	0000	00xx	Z	2
COMF	f, d	对f取反	1	00	1001	dfff	ffff	Z	2
DECf	f, d	f递减1	1	00	0011	dfff	ffff	Z	2
INCF	f, d	f递增1	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	W和f做逻辑或运算	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	传送f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	将W的内容传送到f	1	00	0000	1fff	ffff	Z	2
RLF	f, d	f带进位循环左移	1	00	1101	dfff	ffff	C	2
RRF	f, d	f带进位循环右移	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	f减去W	1	00	0010	dfff	ffff	C、DC和Z	2
SUBWFB	f, d	f减去W（带借位）	1	11	1011	dfff	ffff	C、DC和Z	2
SWAPF	f, d	将f中的两个半字节进行交换	1	00	1110	dfff	ffff	Z	2
XORWF	f, d	W和f作逻辑异或运算	1	00	0110	dfff	ffff	Z	2
面向字节的跳过操作类指令									
DECFSZ	f, d	f递减1，为0则跳过	1（2）	00	1011	dfff	ffff		1，2
INCFSZ	f, d	f递增1，为0则跳过	1（2）	00	1111	dfff	ffff		1，2
面向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
面向位的跳过操作类指令									
BTFSC	f, b	测试f中的某位，为0则跳过	1（2）	01	10bb	bfff	ffff		1，2
BTFSS	f, b	测试f中的某位，为1则跳过	1（2）	01	11bb	bfff	ffff		1，2
立即数操作类指令									
ADDLW	k	立即数和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	将立即数传送到BSR	1	00	0000	001k	kkkk		
MOVLW	k	将立即数传送到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)F1615/9

表 34-3: 增强型中档指令集（续）

助记符, 操作数		说明	周期数	14位操作码				受影响的状态位	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	返回并将立即数传送到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	$\overline{TO}$ 和 $\overline{PD}$	
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	$\overline{TO}$ 和 $\overline{PD}$	
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		
优化的C编译器指令									
ADDFSR	n, k	立即数k和FSRn相加	1	11	0001	0nkk	kkkk	Z	2, 3
MOVIW	n mm	将间接寄存器FSRn传送到W, 带有预/后递增/递减修改变量mm	1	00	0000	0001	0nmm kkkk		
MOVWI	k[n]	将INDFn送入W, 间接变址寻址。	1	11	1111	0nkk	1nmm		
	n mm	将W传送到间接寄存器FSRn, 带有预/后递增/递减修改变量mm	1	00	0000	0001	kkkk		
	k[n]	将W送入INDFn, 间接变址寻址。	1	11	1111	1nkk			

- 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。

34.2 指令说明

ADDFSR	立即数和FSRn相加
语法:	[标号] ADDFSR FSRn, k
操作数:	$-32 \leq k \leq 31$ $n \in [0, 1]$
操作:	$FSR(n) + k \rightarrow FSR(n)$
受影响的状态位:	无
说明:	将有符号6位立即数k与FSRnH:FSRnL寄存器对的内容相加。  FSRn地址范围限制为0000h-FFFFh。传送地址超出该边界时，FSR会发生折回。

ANDLW	立即数和W作逻辑与运算
语法:	[标号] ANDLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) .AND. (k) \rightarrow (W)$
受影响的状态位:	Z
说明:	将W寄存器的内容与8位立即数k进行逻辑与运算。结果存入W寄存器。

ADDLW	立即数和W相加
语法:	[标号] ADDLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) + k \rightarrow (W)$
受影响的状态位:	C、DC和Z
说明:	将W寄存器的内容与8位立即数k相加，结果存入W寄存器。

ANDWF	W和f作逻辑与运算
语法:	[标号] ANDWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) .AND. (f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将W寄存器的内容与寄存器f的内容进行逻辑与运算。如果d为0，结果存入W寄存器。如果d为1，结果存回寄存器f。

ADDWF	W和f相加
语法:	[标号] ADDWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) + (f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	C、DC和Z
说明:	将W寄存器的内容与寄存器f的内容相加。如果d为0，结果存入W寄存器。如果d为1，结果存回寄存器f。

ASRF	算术右移
语法:	[标号] ASRF f{,d}
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f<7>) \rightarrow \text{dest}<7>$ $(f<7:1>) \rightarrow \text{dest}<6:0>$ , $(f<0>) \rightarrow C$
受影响的状态位:	C和Z
说明:	将寄存器f的内容连同进位标志位一起右移1位。MSb保持不变。如果d为0，结果放入W寄存器。如果d为1，结果存回寄存器f。



ADDWFC	W和f相加（带进位）
语法:	[标号] ADDWFC f{,d}
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) + (f) + (C) \rightarrow \text{dest}$
受影响的状态位:	C、DC和Z
说明:	将W的内容、进位标志位与数据存储单元f的内容相加。如果d为0，结果放入W寄存器。如果d为1，结果放入数据存储单元f。

# PIC16(L)F1615/9

## 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:  $0 \rightarrow (f<b>)$   
受影响的状态位: 无  
说明: 将寄存器f中的位b清零。

## BTFSC 测试f中的某位, 为0则跳过

语法: [标号] BTFSC f,b  
操作数:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
操作: 如果  $(f<b>) = 0$ , 则跳过  
受影响的状态位: 无  
说明: 如果寄存器f的位b为1, 则执行下一条指令。  
如果寄存器f的位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与PC相加。由于PC将递增以便取出下一条指令, 所以新地址将为  $\text{PC} + 1 + k$ 。该指令为一条双周期指令。该跳转的地址范围存在限制。

## BTFSS 测试f中的某位, 为1则跳过

语法: [标号] BTFSS f,b  
操作数:  $0 \leq f \leq 127$   
 $0 \leq b < 7$   
操作: 如果  $(f<b>) = 1$ , 则跳过  
受影响的状态位: 无  
说明: 如果寄存器f的位b为0, 则执行下一条指令。  
如果位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)与PC相加。由于PC将递增以便取出下一条指令, 所以新地址将为  $\text{PC} + 1 + (W)$ 。该指令为一条双周期指令。

## CALL 调用子程序

语法: [标号] CALL k  
操作数:  $0 \leq k \leq 2047$   
操作:  $(\text{PC}) + 1 \rightarrow \text{TOS}$ ,  
 $k \rightarrow \text{PC}<10:0>$ ,  
 $(\text{PCLATH}<6:3>) \rightarrow \text{PC}<14:11>$   
受影响的状态位: 无  
说明: 调用子程序。首先, 将返回地址  $(\text{PC} + 1)$  压入堆栈。11位直接地址值被装入PC的<10:0>位。PC的高位值从PCLATH装入。CALL是一条双周期指令。

## 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:  $1 \rightarrow (f<b>)$   
受影响的状态位: 无  
说明: 将寄存器f中的位b置1。

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:	$00h \rightarrow WDT$ $0 \rightarrow WDT$ 分频器, $1 \rightarrow \overline{TO}$ $1 \rightarrow \overline{PD}$
受影响的状态位:	$\overline{TO}$ 和 $\overline{PD}$
说明:	CLRWDT 指令复位看门狗定时器及其预分频器。 状态位 $\overline{TO}$ 和 $\overline{PD}$ 均被置1。

COMF	对f取反
语法:	[标号] COMF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(\bar{f}) \rightarrow (\text{目标寄存器})$
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, 则执行下一条指令。如果结果为0, 代之执行一条NOP指令, 使之成为一条双周期指令。

# PIC16(L)F1615/9

GOTO	无条件跳转
语法:	[标号] GOTO k
操作数:	$0 \leq k \leq 2047$
操作:	$k \rightarrow PC<10:0>$ $PCLATH<6:3> \rightarrow PC<14:11>$
受影响的状态位:	无
说明:	GOTO 是一条无条件跳转指令。11 位立即数值被装入 PC 的 <10:0> 位。PC 的高位从 PCLATH<4:3> 装入。GOTO 是一条双周期指令。

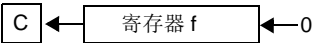
INCF	f 递增 1
语法:	[标号] INCF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) + 1 \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将寄存器 f 的内容递增 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

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，则执行下一条指令。如果结果为 0，代之执行一条 NOP 指令，使之成为一条双周期指令。

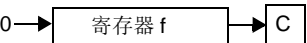
IORLW	立即数和 W 作逻辑或运算
语法:	[标号] IORLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) .OR. k \rightarrow (W)$
受影响的状态位:	Z
说明:	将 W 寄存器的内容与 8 位立即数 k 进行逻辑或运算。结果存入 W 寄存器。

IORWF	W 和 f 做逻辑或运算
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:	$(f7) \rightarrow C$ $(f<6:0>) \rightarrow \text{dest}<7:1>$ $0 \rightarrow \text{dest}<0>$
受影响的状态位:	C 和 Z
说明:	将寄存器 f 的内容连同进位标志位一起左移 1 位。0 移入 LSB。如果 d 为 0，结果放入 W 寄存器。如果 d 为 1，结果存回寄存器 f。



LSRF	逻辑右移
语法:	[标号] LSRF f,{d}
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$0 \rightarrow \text{dest}<7>$ $(f<7:1>) \rightarrow \text{dest}<6:0>$ ， $(f0) \rightarrow C$
受影响的状态位:	C 和 Z
说明:	将寄存器 f 的内容连同进位标志位一起右移 1 位。0 移入 MSB。如果 d 为 0，结果放入 W 寄存器。如果 d 为 1，结果存回寄存器 f。



MOVWF	传送f
语法:	[标号] MOVWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	(f) → (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对文件寄存器进行检测。
指令字数:	1
指令周期数:	1
示例:	MOVWF FSR, 0 执行指令后 W = FSR寄存器的值 Z = 1

MOVLW	将INDFn的内容传送到W
语法:	[标号] MOVLW ++FSRn [标号] MOVLW --FSRn [标号] MOVLW FSRn++ [标号] MOVLW FSRn-- [标号] MOVLW k[FSRn]
操作数:	$n \in [0,1]$ $mm \in [00,01,10,11]$ $-32 \leq k \leq 31$
操作:	INDFn → W 有效地址通过以下方式确定 <ul style="list-style-type: none"><li>FSR + 1 (预递增)</li><li>FSR - 1 (预递减)</li><li>FSR + k (相对偏移)</li></ul> 在传送之后, FSR值将为以下之一: <ul style="list-style-type: none"><li>FSR + 1 (全部递增)</li><li>FSR - 1 (全部递减)</li><li>不变</li></ul>
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	将立即数传送到BSR
语法:	[标号] MOVLB k
操作数:	$0 \leq k \leq 31$
操作:	k → 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)。

# PIC16(L)F1615/9

<b>MOVLW</b>	<b>将立即数传送到PCLATH</b>
语法:	[标号] MOVLW k
操作数:	$0 \leq k \leq 127$
操作:	$k \rightarrow \text{PCLATH}$
受影响的状态位:	无
说明:	将7位立即数k装入PCLATH寄存器。

<b>MOVLW</b>	<b>将立即数传送到W</b>
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:	<pre>MOVLW    0x5A 执行指令后 W        =    0x5A</pre>

<b>MOVWF</b>	<b>将W的内容传送到f</b>
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:	<pre>MOVWF    OPTION_REG 执行指令前 OPTION_REG = 0xFF W          = 0x4F 执行指令后 OPTION_REG = 0x4F W          = 0x4F</pre>

<b>MOVWI</b>	<b>将W的内容传送到INDFn</b>
语法:	[标号] MOVWI ++FSRn [标号] MOVWI --FSRn [标号] MOVWI FSRn++ [标号] MOVWI FSRn-- [标号] MOVWI k[FSRn]
操作数:	$n \in [0,1]$ $mm \in [00,01,10,11]$ $-32 \leq k \leq 31$
操作:	$W \rightarrow \text{INDFn}$ 有效地址通过以下方式确定 • FSR + 1（预递增） • FSR - 1（预递减） • FSR + k（相对偏移） 在传送之后，FSR值将为以下之一： • FSR + 1（全部递增） • FSR - 1（全部递减） 不变
受影响的状态位:	无

模式	语法	mm
预递增	++FSRn	00
预递减	--FSRn	01
后递增	FSRn++	10
后递减	FSRn--	11

说明:

该指令用于在W和一个间接寄存器（INDFn）之间传送数据。在该传送操作之前/之后，将通过预/后递增/递减指针来更新指针（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。

对于FSRn的递增/递减操作不会影响任何状态位。



NOP	空操作
语法:	[标号] NOP
操作数:	无
操作:	空操作
受影响的状态位:	无
说明:	不执行任何操作。
指令字数:	1
指令周期数:	1
示例:	NOP

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) → 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。

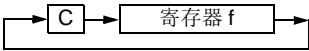
RESET	软件复位
语法:	[标号] RESET
操作数:	无
操作:	执行器件复位。复位PCON寄存器的RI标志。
受影响的状态位:	无
说明:	此指令可实现用软件执行硬件复位。

RETFIE	从中断返回
语法:	[标号] RETFIE
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 (Top-of-Stack, TOS) 的内容装入PC。通过将全局中断允许位GIE (INTCON<7>) 置1, 来允许中断。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	RETFIE
中断后	PC = TOS GIE = 1

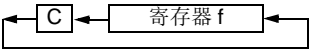
RETLW	返回并将立即数传送到W
语法:	[标号] RETLW k
操作数:	0 ≤ k ≤ 255
操作:	k → (W); TOS → PC
受影响的状态位:	无
说明:	将8位立即数k装入W寄存器。将栈顶内容 (返回地址) 装入程序计数器。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	CALL TABLE;W contains table ;offset value • ;W now has table value • • ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • • RETLW kn ; End of table
TABLE	
执行指令前	W = 0x07
执行指令后	W = k8的值

RETURN	从子程序返回
语法:	[标号] RETURN
操作数:	无
操作:	TOS → 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。

RRF	f带进位循环右移
语法:	[标号] RRF 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。



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:	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

SLEEP	进入休眠模式
语法:	[标号] SLEEP
操作数:	无
操作:	00h → WDT, 0 → WDT分频器, 1 → $\overline{TO}$ , 0 → $\overline{PD}$
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 $\overline{TO}$ 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。

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	$W<3:0> \leq k<3:0>$

SUBWF f减去W

语法: [标号] SUBWF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

操作:  $(f) - (W) \rightarrow (\text{目标寄存器})$

受影响的状态位: C、DC和Z

说明: 用寄存器f的内容减去W寄存器的内容（通过二进制补码方式进行运算）。如果d为0，结果存入W寄存器。如果d为1，结果存回寄存器f。

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W<3:0> > f<3:0>$
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:  $(f) - (W) - (\overline{B}) \rightarrow \text{dest}$

受影响的状态位: C、DC和Z

说明: 用寄存器f的内容减去W的内容和借位标志（进位）（通过二进制补码方式进行运算）。如果d为0，结果存入W寄存器。如果d为1，结果存回寄存器f。

SWAPF 将f中的两个半字节进行交换

语法: [标号] SWAPF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

操作:  $(f<3:0>) \rightarrow (\text{目标寄存器}<7:4>)$ ,  
 $(f<7:4>) \rightarrow (\text{目标寄存器}<3:0>)$

受影响的状态位: 无

说明: 寄存器f的高半字节和低半字节相互交换。如果d为0，结果存入W寄存器。如果d为1，结果存入寄存器f。

TRIS 将W的内容装入TRIS寄存器

语法: [标号] TRIS f

操作数:  $5 \leq f \leq 7$

操作:  $(W) \rightarrow \text{TRIS寄存器}f$

受影响的状态位: 无

说明: 将W寄存器的数据传送到TRIS寄存器。  
当f = 5时，装入TRISA。  
当f = 6时，装入TRISB。  
当f = 7时，装入TRISC。

# PIC16(L)F1615/9

---

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进行逻辑或运算。结果存入W寄存器。

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) $\rightarrow$ (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。

35.0 电气规范

35.1 绝对最大值†)

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	
PIC16F1615/9 .....	-0.3V 至 +6.5V
PIC16LF1615/9 .....	-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 <sup>(1)</sup>	
-40°C ≤ TA ≤ +85°C .....	250 mA
+85°C ≤ TA ≤ +125°C .....	85 mA
VDD 引脚 <sup>(1)</sup>	
-40°C ≤ TA ≤ +85°C .....	250 mA
+85°C ≤ TA ≤ +125°C .....	85 mA
任一标准 I/O 引脚的最大灌电流 .....	50 mA
任一标准 I/O 引脚的最大拉电流 .....	50 mA
任一高电流 I/O 引脚的最大灌电流 .....	100 mA
任一高电流 I/O 引脚的最大拉电流 .....	100 mA
钳位电流, IK (VPIN < 0 或 VPIN > VDD) .....	±20 mA
总功耗 <sup>(2)</sup> .....	800 mW

**注 1:** 最大电流要求 I/O 引脚上具有均匀的负载分布。最大电流可以通过器件封装功率耗散特性进行限制, 请参见表 35-6 来计算器件规范值。

**2:** 功耗按如下公式计算:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ 。

**†注:** 如果器件工作条件超过上述“绝对最大值”, 可能对器件造成永久性损坏。上述数值仅是工作条件最大值, 我们建议不要使器件工作在最大值甚至超过最大值的条件下。器件长时间工作在最大值条件下, 其可靠性可能受到影响。

# PIC16(L)F1615/9

## 35.2 标准工作条件

所有器件的标准工作条件定义如下：

工作电压： $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$   
工作温度： $T_{A\_MIN} \leq T_A \leq T_{A\_MAX}$

**V<sub>DD</sub>**—— 供电电压工作范围<sup>(1)</sup>

PIC16LF1615/9	
V <sub>DDMIN</sub> (F <sub>osc</sub> ≤ 16 MHz).....	+1.8V
V <sub>DDMIN</sub> (F <sub>osc</sub> ≤ 32 MHz).....	+2.5V
V <sub>DDMAX</sub> .....	+3.6V
PIC16F1615/9	
V <sub>DDMIN</sub> (F <sub>osc</sub> ≤ 16 MHz).....	+2.3V
V <sub>DDMIN</sub> (F <sub>osc</sub> ≤ 32 MHz).....	+2.5V
V <sub>DDMAX</sub> .....	+5.5V

**T<sub>A</sub>**—— 工作环境温度范围

工业级温度	
T <sub>A\_MIN</sub> .....	-40°C
T <sub>A\_MAX</sub> .....	+85°C
扩展级温度	
T <sub>A\_MIN</sub> .....	-40°C
T <sub>A\_MAX</sub> .....	+125°C

**注 1：** 请参见参数D001，DS特性：供电电压。

图 35-1: 电压频率关系图,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ , 仅限 PIC16F1615/9

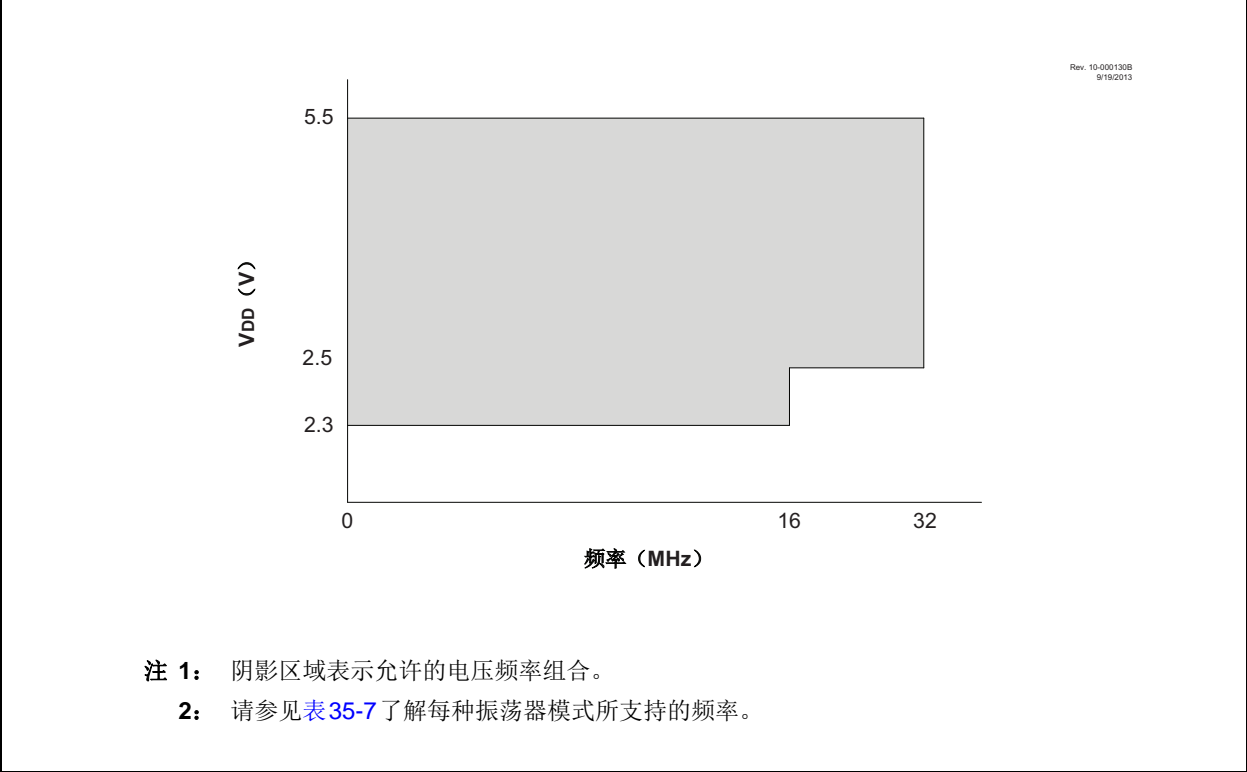
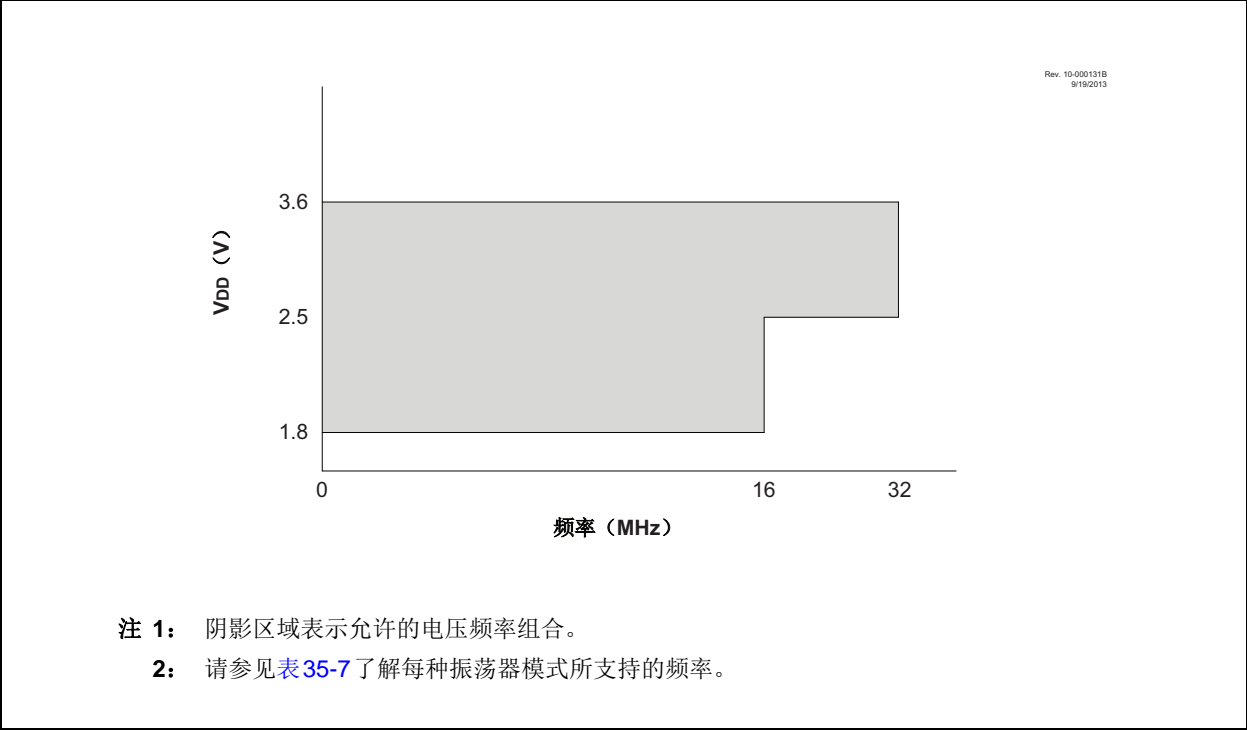


图 35-2: 电压频率关系图,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ , 仅限 PIC16LF1615/9



# PIC16(L)F1615/9

## 35.3 直流特性

表 35-1: 供电电压

PIC16LF1615/9		标准工作条件（除非另外声明）					
PIC16F1615/9							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
D001	VDD	供电电压					
			VDDMIN		VDDMAX		
			1.8 2.5	— —	3.6 3.6	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz
D001			2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz
D002*	VDR	RAM 数据保持电压 <sup>(1)</sup>					
			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 <sup>(2)</sup>					
			—	1.6	—	V	
D002A*			—	1.6	—	V	
D002B*	VPORR*	上电复位重新激活电压 <sup>(2)</sup>					
			—	0.8	—	V	
D002B*			—	1.5	—	V	
D003	VFVR	固定参考电压					
			—	1.024	—	V	-40°C ≤ TA ≤ +85°C
D003			—	1.024	—	V	-40°C ≤ TA ≤ +85°C
D003A	VADFVR	ADC 的 FVR 增益电压精度					
			-4	—	+4	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V
D003A			-4	—	+4	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V 4x VFVR, VDD ≥ 4.75V
D003B	VCDAFVR	比较器的 FVR 增益电压精度					
			-4	—	+4	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V
D003B			-4	—	+4	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V 4x VFVR, VDD ≥ 4.75V
D004*	SVDD	VDD 上升速率 <sup>(2)</sup>					
			0.05	—	—	V/ms	请确保正确释放上电复位信号。
D004*			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: 请参见图 35-3，可以看到 VDD 缓慢上升时，POR 和 POR 重新激活。



图35-3: VDD缓慢上升时, POR和POR重新激活

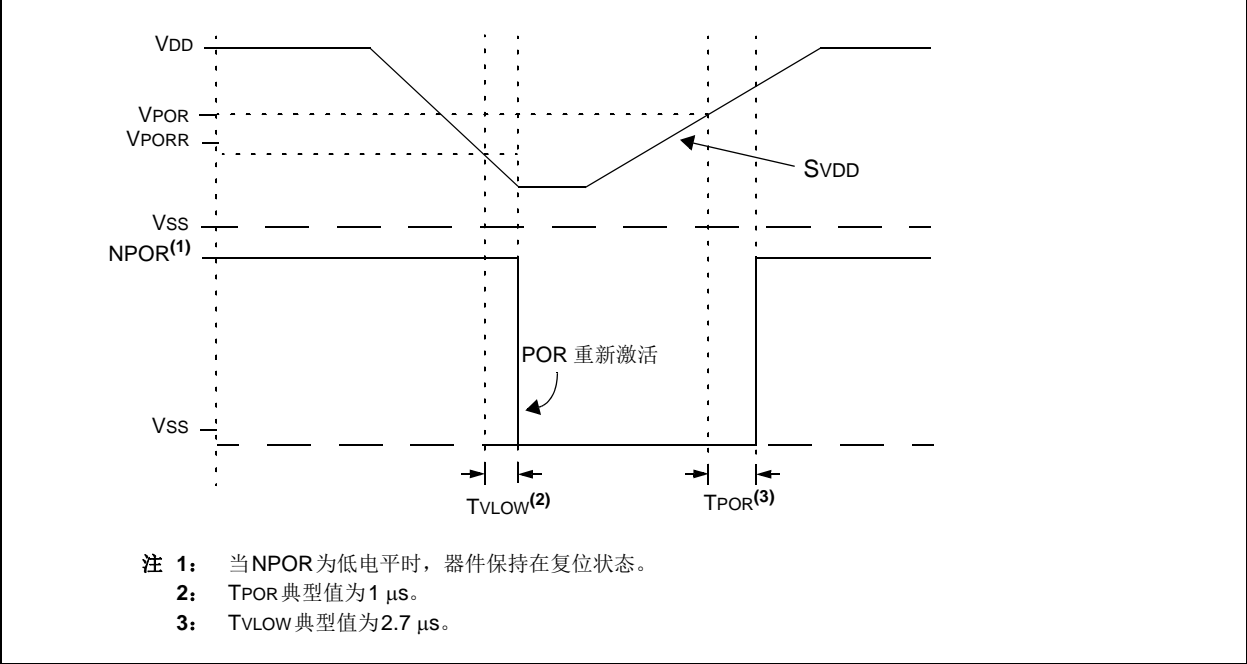


表35-2: 供电电流 (IDD) (1,2)

PIC16LF1615/9		标准工作条件 (除非另外声明)					
PIC16F1615/9							
参数编号	器件特性	最小值	典型值†	最大值	单位	条件	
						VDD	注
D013		—	30	90	μA	1.8	FOSC = 1 MHz, 外部时钟 (ECM), 中等功耗模式
		—	55	110	μA	3.0	
D013		—	65	120	μA	2.3	FOSC = 1 MHz, 外部时钟 (ECM), 中等功耗模式
		—	85	150	μA	3.0	
		—	115	200	μA	5.0	
D014		—	115	260	μA	1.8	FOSC = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	210	380	μA	3.0	
D014		—	180	310	μA	2.3	FOSC = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	240	410	μA	3.0	
		—	295	520	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: 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VSS; MCLR = VDD; WDT禁止。  
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。

# PIC16(L)F1615/9

表 35-2: 供电电流 (IDD) (1,2) (续)

PIC16LF1615/9		标准工作条件 (除非另外声明)					
PIC16F1615/9							
参数编号	器件特性	最小值	典型值†	最大值	单位	条件	
						VDD	注
D015		—	9.6	36	μA	1.8	Fosc = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	16.2	60	μA	3.0	
D015		—	39	84	μA	2.3	Fosc = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	45	90	μA	3.0	
		—	51	108	μA	5.0	
D016		—	215	360	μA	1.8	Fosc = 500 kHz, HFINTOSC
		—	275	480	μA	3.0	
D016		—	270	450	μA	2.3	Fosc = 500 kHz, HFINTOSC
		—	300	500	μA	3.0	
		—	350	620	μA	5.0	
D017*		—	410	800	μA	1.8	Fosc = 8 MHz, HFINTOSC
		—	630	1200	μA	3.0	
D017*		—	530	950	μA	2.3	Fosc = 8 MHz, HFINTOSC
		—	660	1300	μA	3.0	
		—	730	1400	μA	5.0	
D018		—	600	1200	μA	1.8	Fosc = 16 MHz, HFINTOSC
		—	970	1850	μA	3.0	
D018		—	780	1500	μA	2.3	Fosc = 16 MHz, HFINTOSC
		—	1000	1900	μA	3.0	
		—	1090	2100	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: 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VSS；MCLR = VDD；WDT禁止。

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。

表35-2: 供电电流 (IDD) (1,2) (续)

PIC16LF1615/9		标准工作条件 (除非另外声明)					
PIC16F1615/9							
参数编号	器件特性	最小值	典型值†	最大值	单位	条件	
						VDD	注
D019		—	1.6	5.0	mA	3.0	Fosc = 32 MHz, HFINTOSC
		—	1.9	6.0	mA	3.6	
D019		—	1.6	5.0	mA	3.0	Fosc = 32 MHz, HFINTOSC
		—	1.9	6.0	mA	5.0	
D020A		—	1.6	5.0	mA	3.0	Fosc = 32 MHz, 外部时钟 (ECH), 高功耗模式
		—	1.9	6.0	mA	3.6	
D020A		—	1.6	5.0	mA	3.0	Fosc = 32 MHz, 外部时钟 (ECH), 高功耗模式
		—	1.9	6.0	mA	5.0	
D020B		—	6	16	μA	1.8	Fosc = 32 kHz, 外部时钟 (ECL), 低功耗模式
		—	8	22	μA	3.0	
D020B		—	13	43	μA	2.3	Fosc = 32 kHz, 外部时钟 (ECL), 低功耗模式
		—	15	55	μA	3.0	
		—	16	57	μA	5.0	
D020C		—	19	40	μA	1.8	Fosc = 500 kHz, 外部时钟 (ECL), 低功耗模式
		—	32	60	μA	3.0	
D020C		—	31	60	μA	2.3	Fosc = 500 kHz, 外部时钟 (ECL), 低功耗模式
		—	38	90	μA	3.0	
		—	44	100	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: 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VSS; MCLR = VDD; WDT禁止。

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。

# PIC16(L)F1615/9

表35-3: 掉电电流 (IPD) (1,2)

PIC16LF1615/9		工作条件: (除非另外声明) 低功耗休眠模式						
PIC16F1615/9		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
D022	IPD 基本电流	—	0.020	1.0	8.0	μA	1.8	禁止 WDT、BOR 和 FVR, 所有外设不工作
		—	0.025	2.0	9.0	μA	3.0	
D022	IPD 基本电流	—	0.25	3.0	10	μA	2.3	禁止 WDT、BOR 和 FVR, 所有外设不工作, 低功耗休眠模式
		—	0.30	4.0	12	μA	3.0	
		—	0.40	6.0	15	μA	5.0	
D022A	IPD 基本电流	—	9.8	16	18	μA	2.3	禁止 WDT、BOR 和 FVR, 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	
D023		—	0.26	2.0	9.0	μA	1.8	WDT 电流
		—	0.44	3.0	10	μA	3.0	
D023		—	0.43	6.0	15	μA	2.3	WDT 电流
		—	0.53	7.0	20	μA	3.0	
		—	0.64	8.0	22	μA	5.0	
D023A		—	15	28	30	μA	1.8	FVR 电流
		—	18	30	33	μA	3.0	
D023A		—	18	33	35	μA	2.3	FVR 电流
		—	19	35	37	μA	3.0	
		—	20	37	39	μA	5.0	
D024		—	6.0	17	20	μA	3.0	BOR 电流
D024		—	7.0	17	30	μA	3.0	BOR 电流
		—	8.0	20	40	μA	5.0	
D24A		—	0.1	4.0	10	μA	3.0	LPBOR 电流
D24A		—	0.35	5.0	14	μA	3.0	LPBOR 电流
		—	0.45	8.0	17	μA	5.0	
D026		—	0.11	1.5	9.0	μA	1.8	ADC 电流 (注3), 无转换
		—	0.12	2.7	10	μA	3.0	
D026		—	0.30	4.0	11	μA	2.3	ADC 电流 (注3), 无转换
		—	0.35	5.0	13	μA	3.0	
		—	0.45	8.0	16	μA	5.0	
D026A*		—	250	—	—	μA	1.8	ADC 电流 (注3), 转换正在进行
		—	250	—	—	μA	3.0	
D026A*		—	280	—	—	μA	2.3	ADC 电流 (注3), 转换正在进行
		—	280	—	—	μA	3.0	
		—	280	—	—	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。

图注: TBD = 待定

注 1: 可通过从该参数值中减去基本 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 35-3: 掉电电流 (IPD) (1,2) (续)

PIC16LF1615/9		工作条件: (除非另外声明) 低功耗休眠模式						
PIC16F1615/9		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
D027		—	7	22	25	μA	1.8	比较器, CxSP = 0
		—	8	23	27	μA	3.0	
D027		—	17	35	37	μA	2.3	比较器, CxSP = 0
		—	18	37	38	μA	3.0	
		—	19	38	40	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。
- 图注: TBD = 待定
- 注 1: 可通过从该参数值中减去基本 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)F1615/9

表 35-4: 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
D030 D030A D031 D032	V <sub>IL</sub>	输入低电压					
		I/O 端口:					
		带 TTL 缓冲器	—	—	0.8	V	4.5V ≤ V <sub>DD</sub> ≤ 5.5V
			—	—	0.15 V <sub>DD</sub>	V	1.8V ≤ V <sub>DD</sub> ≤ 4.5V
		带施密特触发器缓冲器	—	—	0.2 V <sub>DD</sub>	V	2.0V ≤ V <sub>DD</sub> ≤ 5.5V
		MCLR	—	—	0.2 V <sub>DD</sub>	V	
D040 D040A D041 D042	V <sub>IH</sub>	输入高电压					
		I/O 端口:					
		带 TTL 缓冲器	2.0	—	—	V	4.5V ≤ V <sub>DD</sub> ≤ 5.5V
			0.25 V <sub>DD</sub> + 0.8	—	—	V	1.8V ≤ V <sub>DD</sub> ≤ 4.5V
		带施密特触发器缓冲器	0.8 V <sub>DD</sub>	—	—	V	2.0V ≤ V <sub>DD</sub> ≤ 5.5V
		MCLR	0.8 V <sub>DD</sub>	—	—	V	
D060  D061	I <sub>IL</sub>	输入泄漏电流 <sup>(1)</sup>					
		I/O 端口	—	± 5	± 125	nA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态, 85°C
			—	± 5	± 1000	nA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态, 125°C
		MCLR <sup>(3)</sup>	—	± 50	± 200	nA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态, 85°C
D070*	I <sub>PUR</sub>	弱上拉电流					
			25	100	200	μA	V <sub>DD</sub> = 3.3V, V <sub>PIN</sub> = V <sub>SS</sub>
			25	140	300	μA	V <sub>DD</sub> = 5.0V, V <sub>PIN</sub> = V <sub>SS</sub>
D080  D080A	V <sub>OL</sub>	输出低电压 <sup>(3)</sup>					
		I/O 端口	—	—	0.6	V	I <sub>OL</sub> = 8.0 mA, V <sub>DD</sub> = 5.0V I <sub>OL</sub> = 6.0 mA, V <sub>DD</sub> = 3.3V I <sub>OL</sub> = 1.8 mA, V <sub>DD</sub> = 1.8V
		高驱动能力 I/O <sup>(1)</sup>	—	2.5V	—	V	I <sub>OL</sub> = 100 mA, V <sub>DD</sub> = 5.0V
D090  D090A	V <sub>OH</sub>	输出高电压 <sup>(3)</sup>					
		I/O 端口	V <sub>DD</sub> - 0.7	—	—	V	I <sub>OH</sub> = 3.5 mA, V <sub>DD</sub> = 5.0V I <sub>OH</sub> = 3.0 mA, V <sub>DD</sub> = 3.3V I <sub>OH</sub> = 1.0 mA, V <sub>DD</sub> = 1.8V
		高驱动能力 I/O <sup>(1)</sup>	—	2.5V	—	V	I <sub>OL</sub> = 100 mA, V <sub>DD</sub> = 5.0V
D101A*	C <sub>IO</sub>	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: 负电流定义为引脚的拉电流。

2: MCLR 引脚上的泄漏电流主要取决于所施加的电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。

3: 在 CLKOUT 模式下不包括 OSC2。

表 35-5: 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					
D111	IDDP	MCLR/VPP 引脚上的电压	8.0	—	9.0	V	(注2)
D112	VBE	编程时的供电电流	—	—	10	mA	
D113	VPEW	批量擦除时的VDD	2.7	—	VDDMAX	V	
D114	IPEW	写或行擦除时的VDD	VDDMIN	—	VDDMAX	V	
D115	IPPPGM	擦除/写操作时 MCLR/VPP 上的电流	—	1.0	—	mA	
D116	IDDPGM	擦除/写操作时 VDD 上的电流	—	5.0	—	mA	
D121	EP	闪存程序存储器					
D122	VPRW	单元耐擦写能力	10K	—	—	E/W	-40°C ≤ TA ≤ +85°C (注1)
D123	TiW	读/写操作时的VDD	VDDMIN	—	VDDMAX	V	
D124	TRET	自定时写周期时间	—	2	2.5	ms	
D125	TRETD	特性保持时间	—	40	—	年	假设没有违反其他规范
D126	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。

# PIC16(L)F1615/9

表 35-6: 温度特性

标准工作条件 (除非另外声明)

参数编号	符号	特性	典型值	单位	条件
TH01	$\theta_{JA}$	热阻 (结点到环境)	62.2	°C/W	20 引脚 DIP 封装
			77.7	°C/W	20 引脚 SOIC 封装
			87.3	°C/W	20 引脚 SSOP 封装
			43	°C/W	20 引脚 QFN 4x4 mm 封装
TH02	$\theta_{JC}$	热阻 (结点到管壳)	27.5	°C/W	20 引脚 DIP 封装
			23.1	°C/W	20 引脚 SOIC 封装
			31.1	°C/W	20 引脚 SSOP 封装
			5.3	°C/W	20 引脚 QFN 4x4 mm 封装
TH03	$T_{JMAX}$	最高结温	150	°C	
TH04	PD	功耗	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	内部功耗	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}^{(1)}$
TH06	$P_{I/O}$	I/O 功耗	—	W	$P_{I/O} = \Sigma (I_{OL} \times V_{OL}) + \Sigma (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	$P_{DER}$	降额功耗	—	W	$P_{DER} = P_{D_{MAX}} (T_J - T_A) / \theta_{JA}^{(2)}$

注 1:  $I_{DD}$  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:  $T_A$  = 环境温度,  $T_J$  = 结温。



35.4 交流特性

可根据以下一种格式来创建时序参数符号:

- 1. TppS2ppS
- 2. TppS

<b>T</b>		
F	频率	T 时间

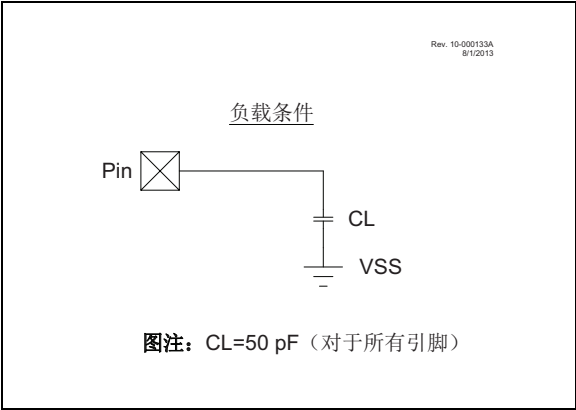
小写字母 (pp) 及其含义:

<b>pp</b>		
cc	CCP1	osc CLKIN
ck	CLKOUT	rd $\overline{RD}$
cs	$\overline{CS}$	rw $\overline{RD}$ 或 $\overline{WR}$
di	SDIx	sc SCKx
do	SDO	ss $\overline{SS}$
dt	数据输入	t0 T0CKI
io	I/O 端口	t1 T1CKI
mc	$\overline{MCLR}$	wr $\overline{WR}$

大写字母及其含义:

<b>S</b>		
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35-4: 负载条件



# PIC16(L)F1615/9

图35-5： 时钟时序

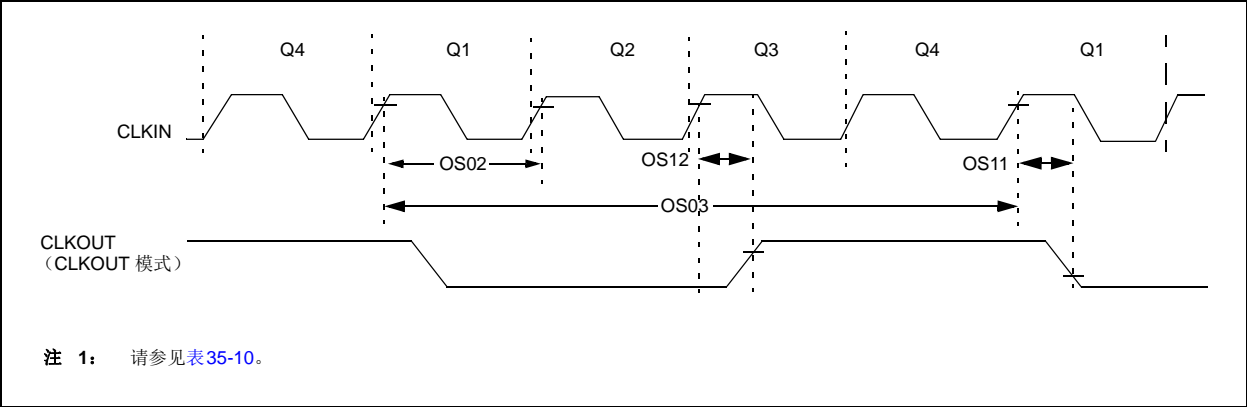


表35-7： 时钟振荡器时序要求

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 <sup>(1)</sup>	DC	—	0.5	MHz	外部时钟（ECL）
			DC	—	4	MHz	外部时钟（ECM）
			DC	—	32	MHz	外部时钟（ECH）
OS02	Tosc	外部CLKIN周期 <sup>(1)</sup>	31.25	—	∞	ns	外部时钟（EC）
OS03	Tcy	指令周期 <sup>(1)</sup>	200	Tcy	DC	ns	Tcy = 4/Fosc

\* 这些参数为特性值，但未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为3.0V和25°C条件下的值。这些参数仅供设计参考，未经测试。

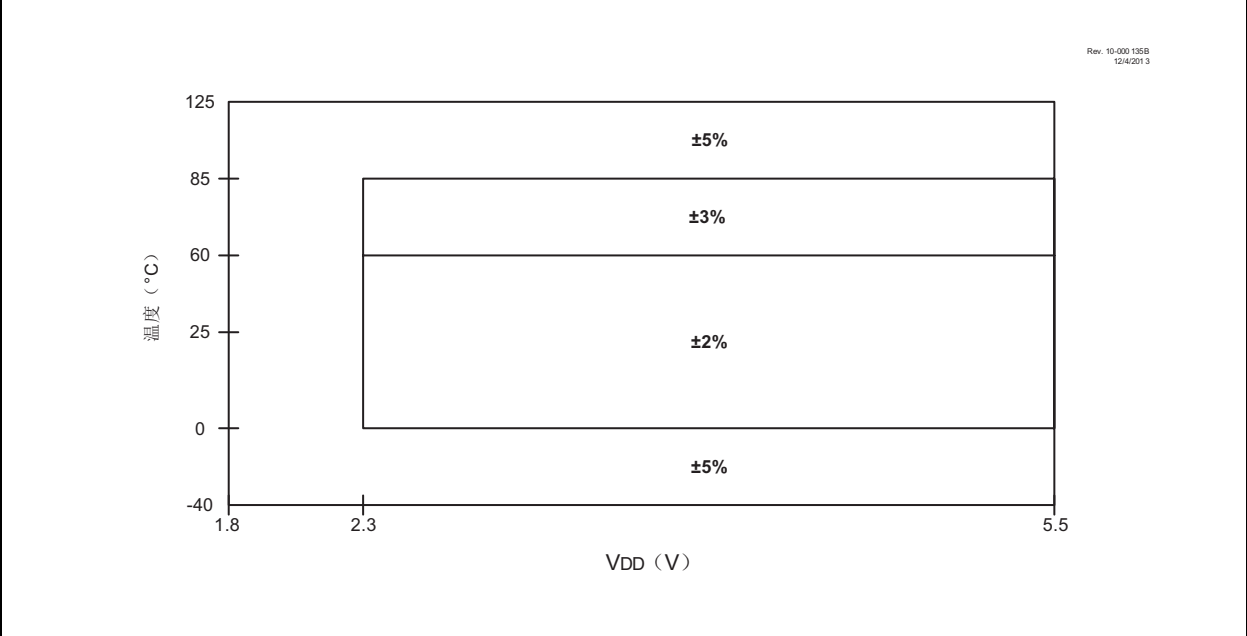
注 1： 指令周期（Tcy）等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值，可能导致振荡器运行不稳定和/或导致电流消耗超出预期值。所有器件在测试“最小”值时，都在CLKIN引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的“最大”周期时间限制为“DC”（无时钟）。

表 35-8： 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 <sup>(1)</sup>	—	—	16.0	—	MHz	(注2)
OS09	LFOSC	内部 LFINTOSC 频率	—	—	31	—	kHz	(注3)
OS10*	TIOSC ST	HFINTOSC 从休眠模式唤醒的启动时间	—	—	5	15	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35-6。
- 3： 请参见图36-27和图36-28。

图 35-6： 在整个VDD和温度范围内的HFINTOSC频率精度



# PIC16(L)F1615/9

表 35-9: PLL 时钟时序规范

标准工作条件（除非另外声明）							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
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。

图35-7: CLKOUT和I/O时序

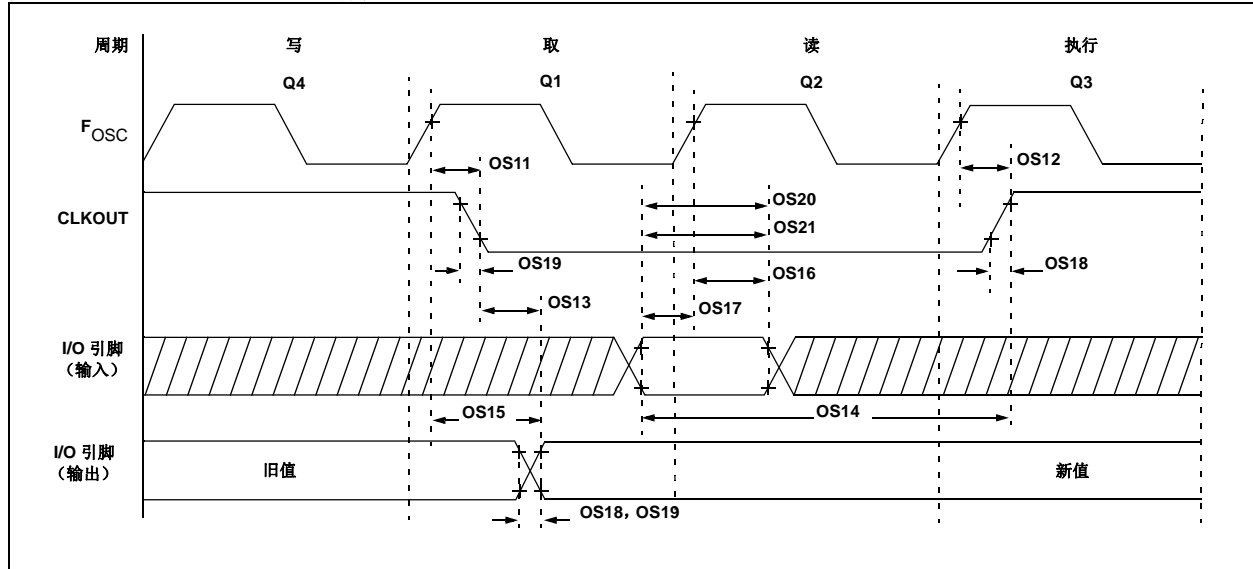


表35-10: CLKOUT和I/O时序参数

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 <sup>(1)</sup>	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	Fosc↑到CLKOUT↑的时间 <sup>(1)</sup>	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓到端口输出有效的的时间 <sup>(1)</sup>	—	—	20	ns	
OS14	TioV2ckH	CLKOUT↑之前端口输入有效的的时间 <sup>(1)</sup>	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑（Q1周期）到端口输出有效的的时间	—	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输入建立时间）	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 x TOSC。

图35-8： 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序

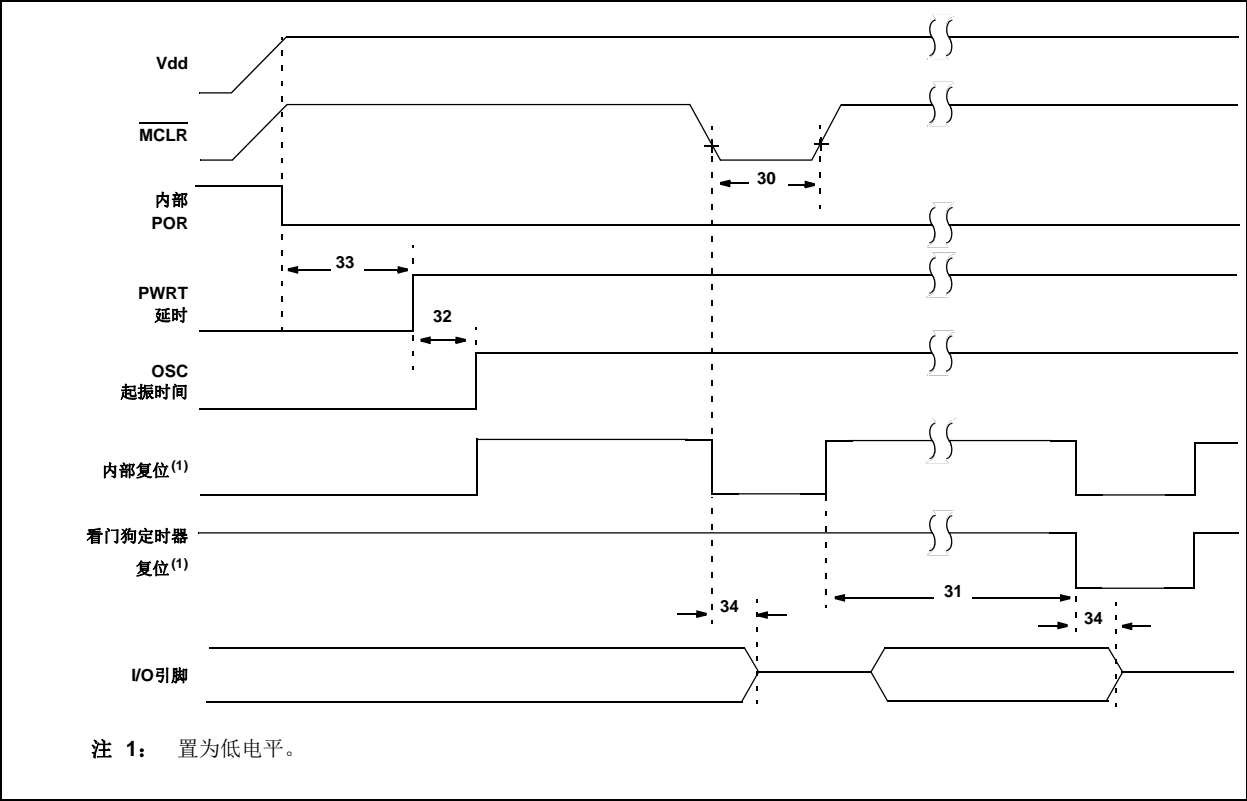


表35-11： 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位参数

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 <sup>(1)</sup>	—	1024	—	TOSC	
33*	TPWRT	上电延时定时器周期	40	65	140	ms	PWRT = 0
34*	TIOZ	自 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	欠压复位电压 <sup>(2)</sup>	2.55	2.70	2.85	V	BORV = 0
			2.35	2.45	2.58	V	BORV = 1（PIC16F1615/9）
			1.80	1.90	2.05	V	BORV = 1（PIC16LF1615/9）
36*	VHYST	欠压复位滞后电压	0	25	60	mV	-40°C ≤ TA ≤ +85°C
37*	TBORDC	欠压复位直流响应时间	1	16	35	μs	VDD ≤ VBOR
38	VLPBOR	低功耗欠压复位电压	1.8	2.1	2.5	V	LPBOR = 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： 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35-9： 欠压复位时序和特性

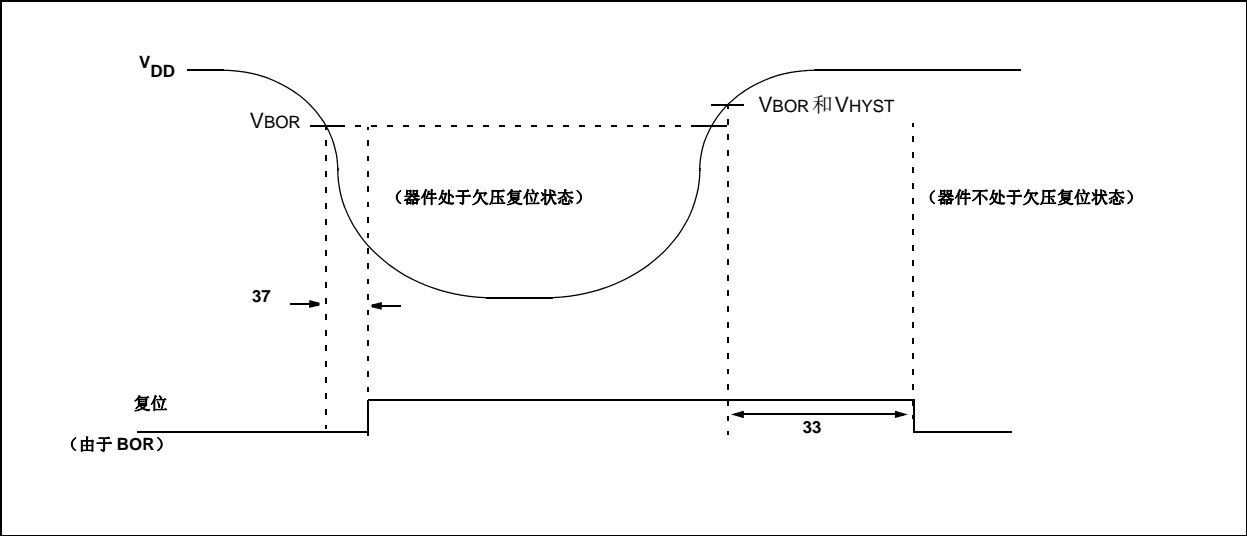


图35-10: TIMER0和TIMER1外部时钟时序

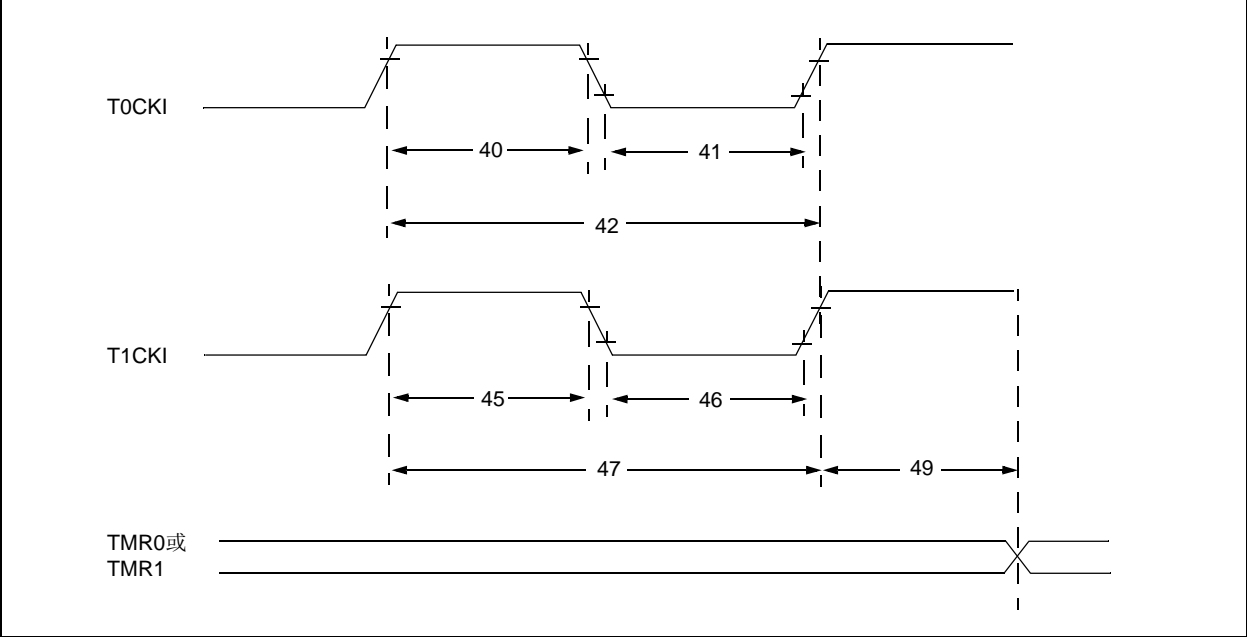


表35-12: TIMER0和TIMER1外部时钟要求

标准工作条件（除非另外声明）								
参数编号	符号	特性		最小值	典型值†	最大值	单位	条件
40*	Tt0H	T0CKI 高电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			带预分频器	10	—	—	ns	
41*	Tt0L	T0CKI 低电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			带预分频器	10	—	—	ns	
42*	Tt0P	T0CKI 周期		取如下二者中较大值: $20$ 或 $\frac{T_{CY} + 40}{N}$	—	—	ns	N = 预分频值
45*	Tt1H	T1CKI 高电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			同步, 有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
46*	Tt1L	T1CKI 低电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			同步, 有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
47*	Tt1P	T1CKI 输入周期	同步	取如下二者中较大值: $30$ 或 $\frac{T_{CY} + 40}{N}$	—	—	ns	N = 预分频值
			异步	60	—	—	ns	
49*	TCKEZTMR1	从外部时钟边沿到定时器递增的延时		$2 T_{OSC}$	—	$7 T_{OSC}$	—	同步模式下的定时器

\* 这些参数为特性值, 但未经测试。  
† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。



表 35-13: 模数转换器 (ADC) 特性<sup>(1,2,3)</sup>

工作条件 (除非另外声明) VDD = 3.0V, TA = 25°C							
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
AD01	NR	分辨率	—	—	10	位	
AD02	EIL	积分误差	—	±1	±1.7	LSb	VREF = 3.0V
AD03	EDL	微分误差	—	±1	±1	LSb	无丢失编码 VREF = 3.0V
AD04	EOFF	失调误差	—	±1	±2.5	LSb	VREF = 3.0V
AD05	EGN	增益误差	—	±1	±2.0	LSb	VREF = 3.0V
AD06	VREF	参考电压	1.8	—	VDD	V	VREF = (VRPOS - VRNEG) (注4)
AD07	VAIN	满量程	VSS	—	VREF	V	
AD08	ZAIN	模拟电压源的推荐阻抗	—	—	10	kΩ	如果输入引脚上接有 0.01 μF 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°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: 关于工作特性, 请参见第 36.0 节“直流和交流特性图表”。
- 4: ADC VREF 通过 ADPREF<0> 位进行选择。

图35-11： ADC 转换时序（ADC时钟基于Fosc）

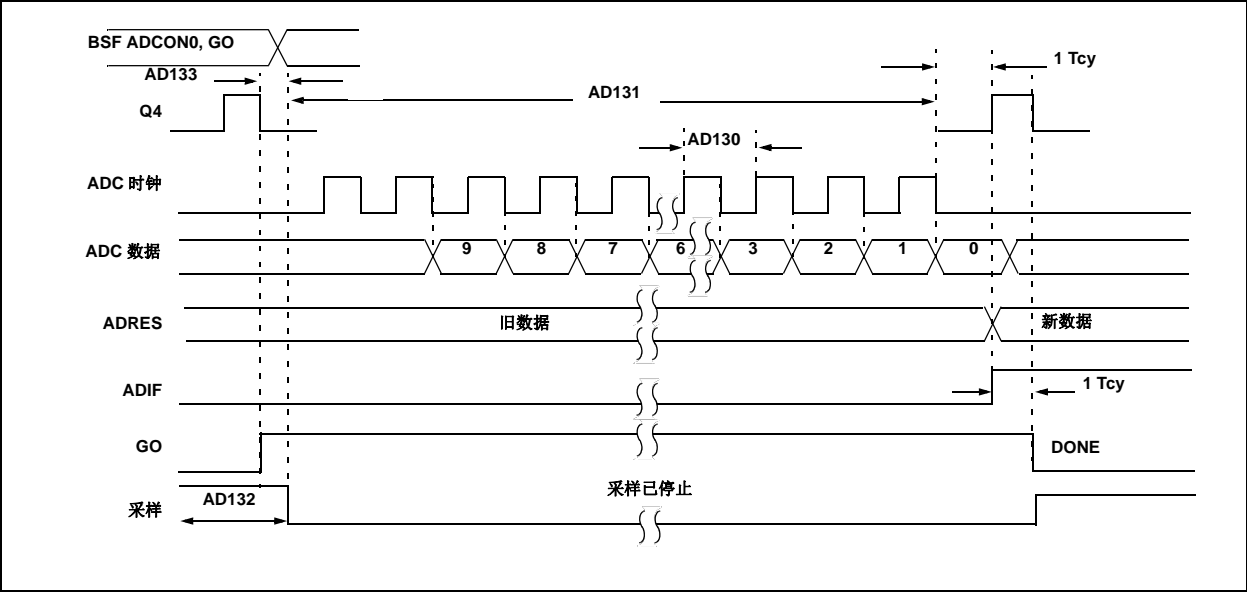


图35-12： ADC 转换时序（ADC时钟来自FRC）

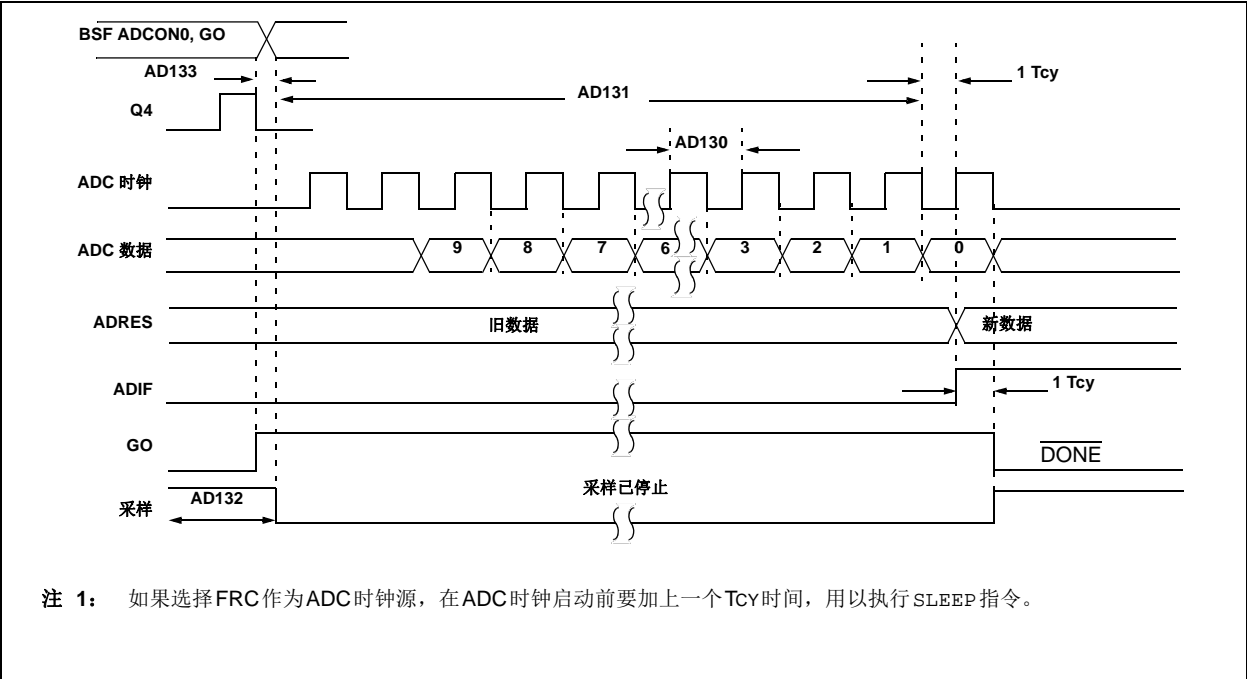


表35-14： ADC转换要求

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）	1.0	—	6.0	μs	基于Fosc
		ADC内部FRC振荡器周期（TFRC）	1.0	2.0	6.0	μs	ADCS<2:0> = x11（ADC FRC模式）
AD131	TCNV	转换时间（不包括采集时间） <sup>(1)</sup>	—	11	—	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	—		基于Fosc
			—	1/2 TAD + 1Tcy	—		ADCS<2:0> = x11（ADC FRC模式）

\* 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： ADRES寄存器可在下一个Tcy周期被读取。

表35-15： 比较器规范<sup>(1)</sup>

工作条件（除非另外声明） VDD = 3.0V，TA = 25°C							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
CM01	Vioff	输入失调电压	—	±7.5	±60	mV	CxSP = 1， Vicm = VDD/2
CM02	Vicm	输入共模电压	0	—	VDD	V	
CM03	CMRR	共模抑制比	—	50	—	dB	
CM04A	Tresp(2)	响应时间上升沿	—	400	800	ns	CxSP = 1
CM04B		响应时间下降沿	—	200	400	ns	CxSP = 1
CM04C		响应时间上升沿	—	1200	—	ns	CxSP = 0
CM04D		响应时间下降沿	—	550	—	ns	CxSP = 0
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	—	25	—	mV	CxHYS = 1， CxSP = 1

\* 这些参数为特性值，但未经测试。  
注 1： 关于工作特性，请参见第36.0节“直流和交流特性图表”。  
2： 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)F1615/9

表35-16：数模转换器（DAC）规范<sup>(1)</sup>

工作条件（除非另外声明） VDD = 3.0V，TA = 25°C							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
DAC01*	CLSB	步长	—	VDD/256	—	V	
DAC02*	CACC	绝对精度	—	—	± 1.5	LSb	
DAC03*	CR	单位电阻值（R）	—	—	—	Ω	
DAC04*	CST	稳定时间 <sup>(2)</sup>	—	—	10	μs	

\* 这些参数为特性值，但未经测试。

注 1：关于工作特性，请参见第36.0节“直流和交流特性图表”。

2：稳定时间是在DACR<4:0>从0000跳变到1111时测得的。

表35-17：过零引脚规范

工作条件（除非另外声明） VDD = 3.0V，TA = 25°C							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
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。

36.0 直流和交流特性图表

本节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VDD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。

**注：** 以下图表为基于有限数量的统计结果，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”代表25°C时的分布的平均值。“最大值”和“最小值”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。

图36-1： 在整个温度范围内的VoH—IoH曲线，VDD = 5.5V，仅限PIC16F1615/9

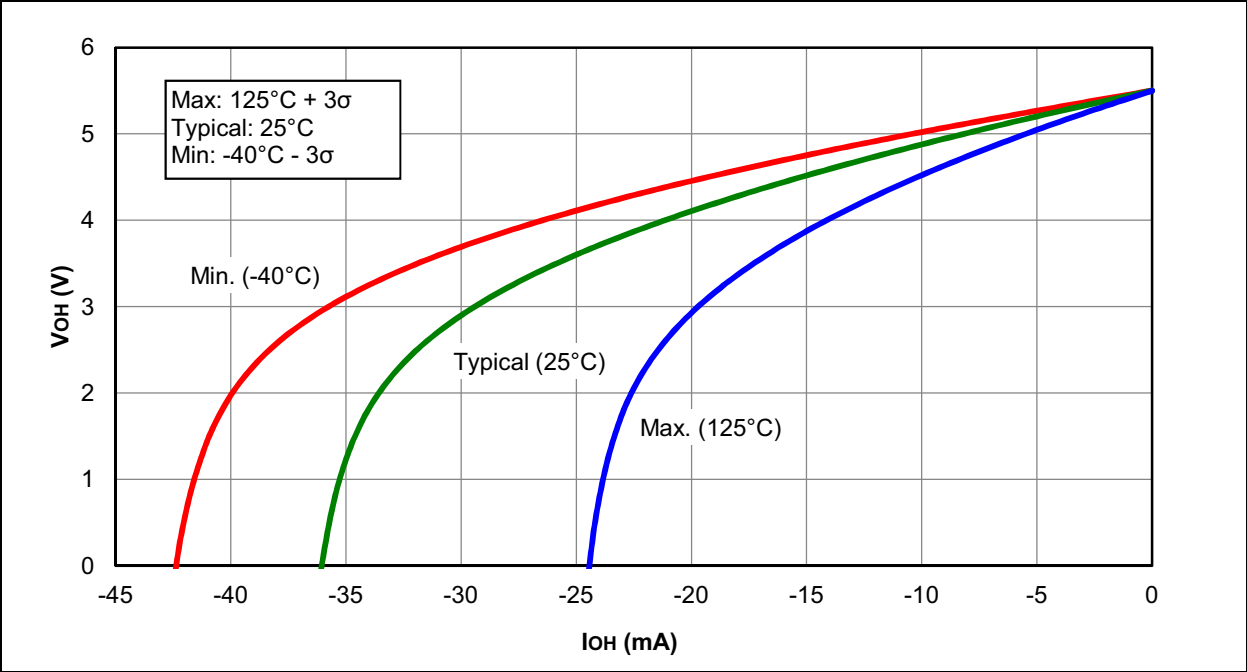


图36-2： 在整个温度范围内的VoL—IoL曲线，VDD = 5.5V，仅限PIC16F1615/9

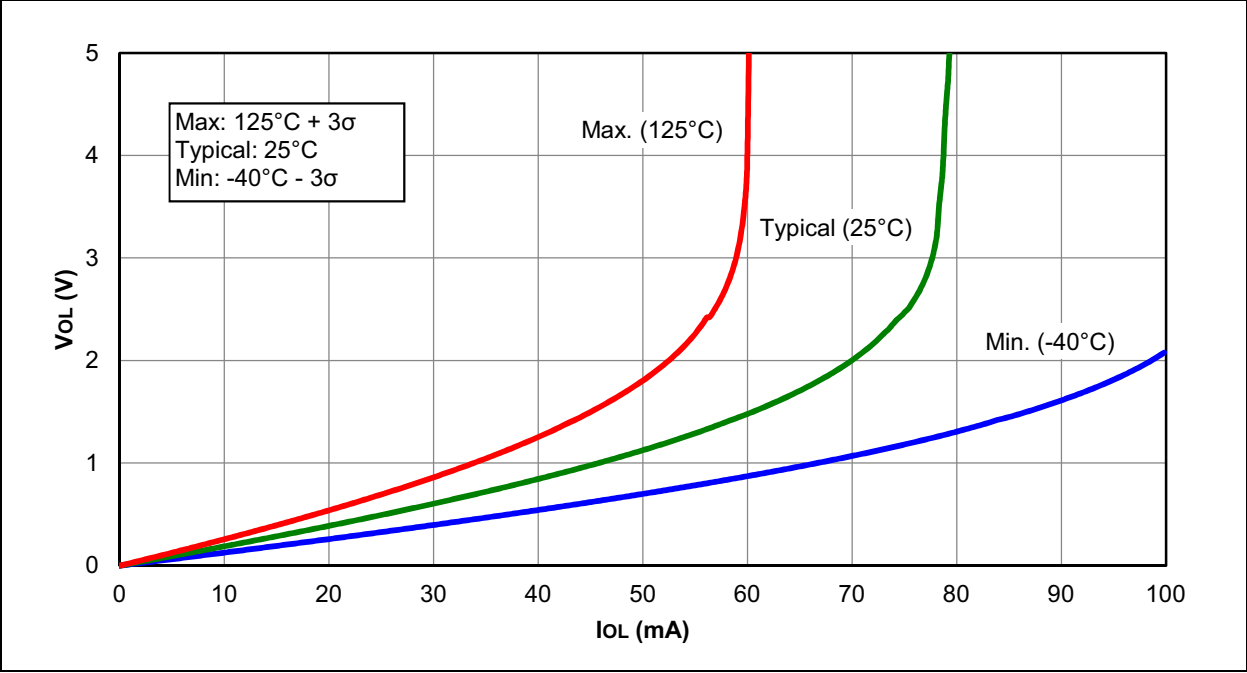


图36-3: 在整个温度范围内的  $V_{OH}$ — $I_{OH}$  曲线,  $V_{DD} = 3.0V$

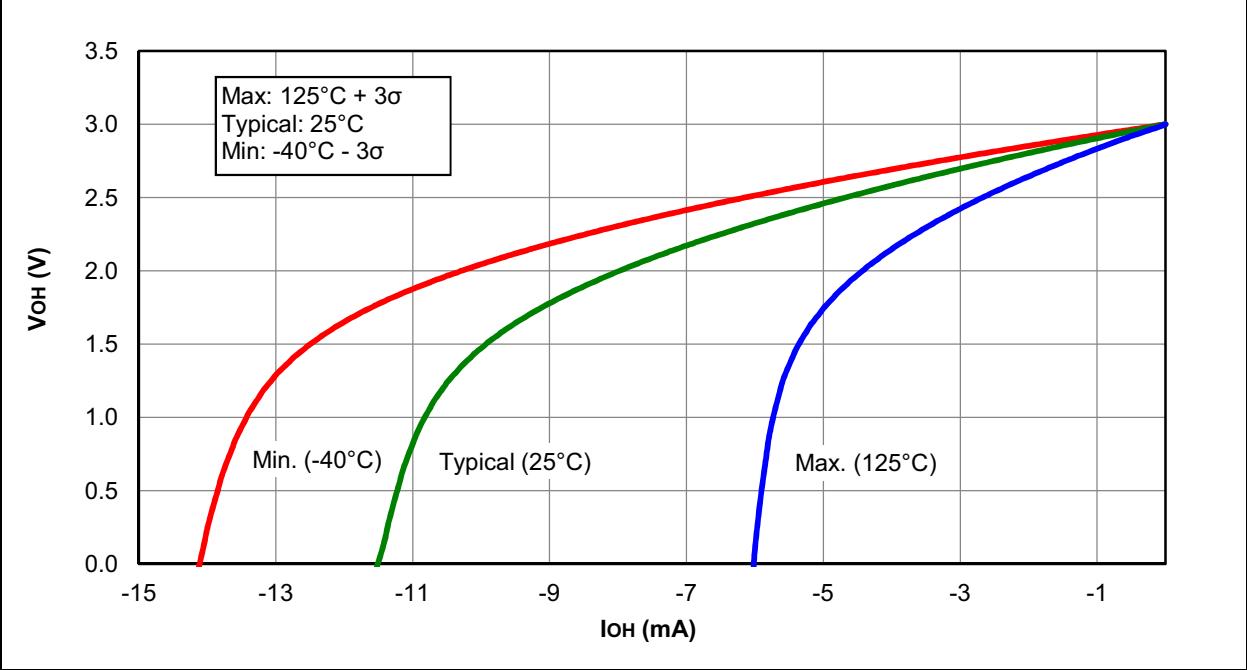


图36-4: 在整个温度范围内的  $V_{OL}$ — $I_{OL}$  曲线,  $V_{DD} = 3.0V$

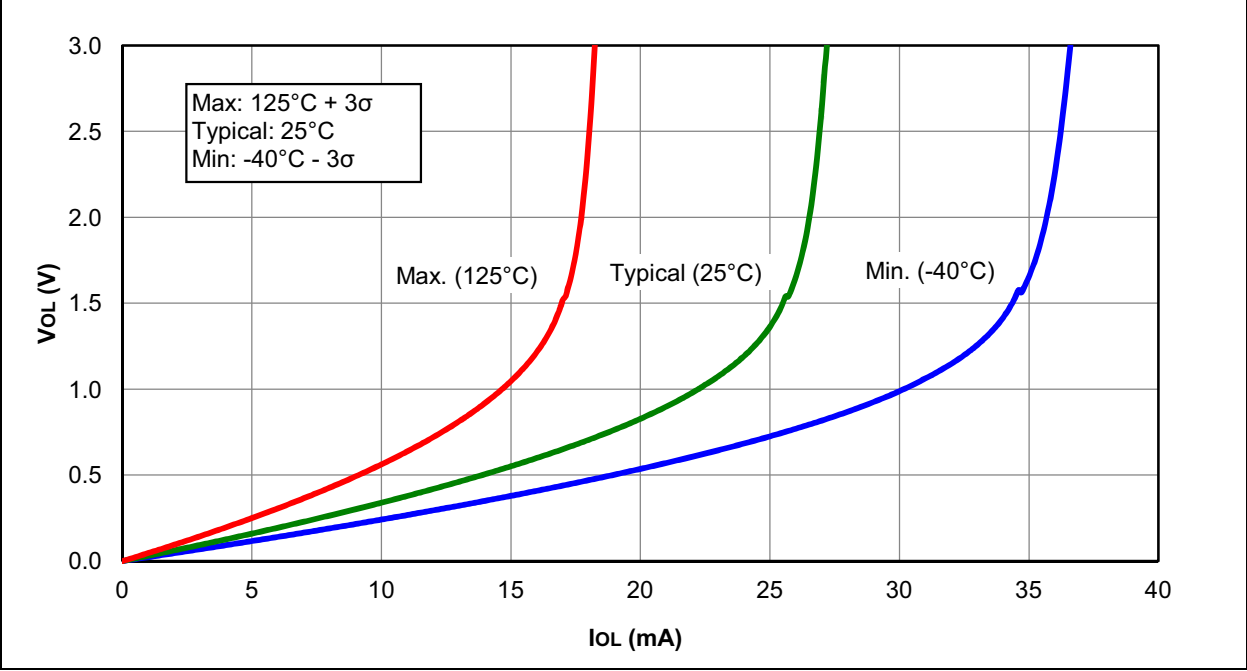


图36-5: 在整个温度范围内的VoH—IoH曲线, VDD = 1.8V, 仅限PIC16LF1615/9

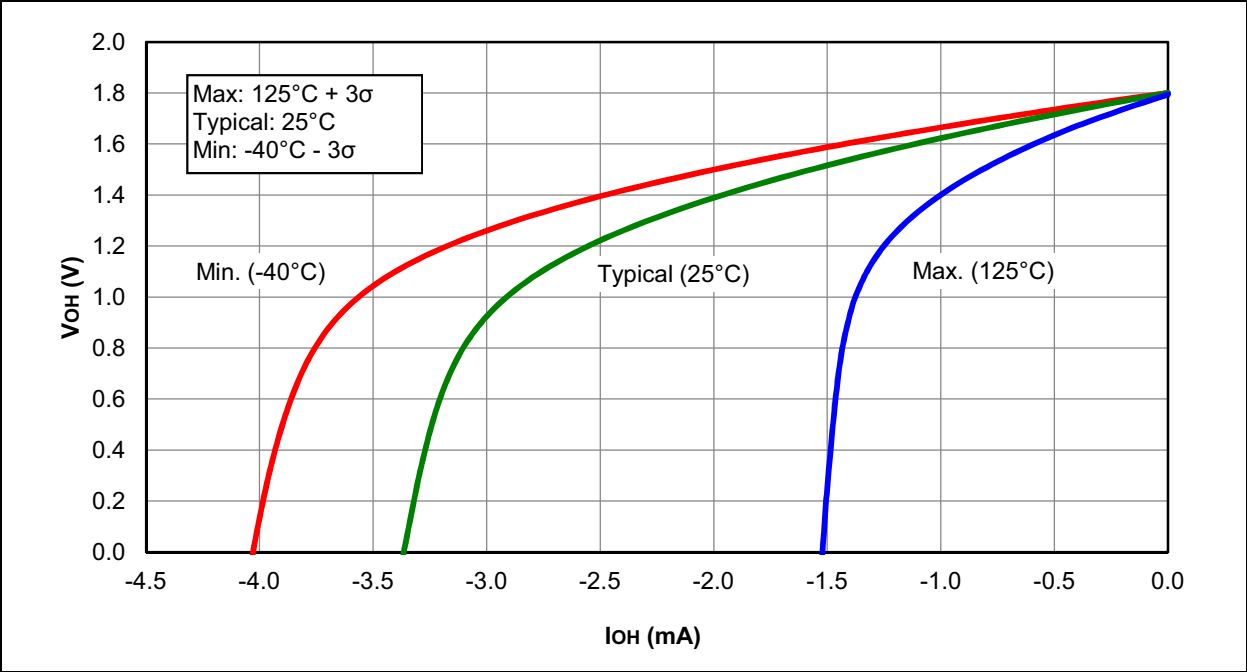


图36-6: 在整个温度范围内的VoL—IoL曲线, VDD = 1.8V, 仅限PIC16LF1615/9

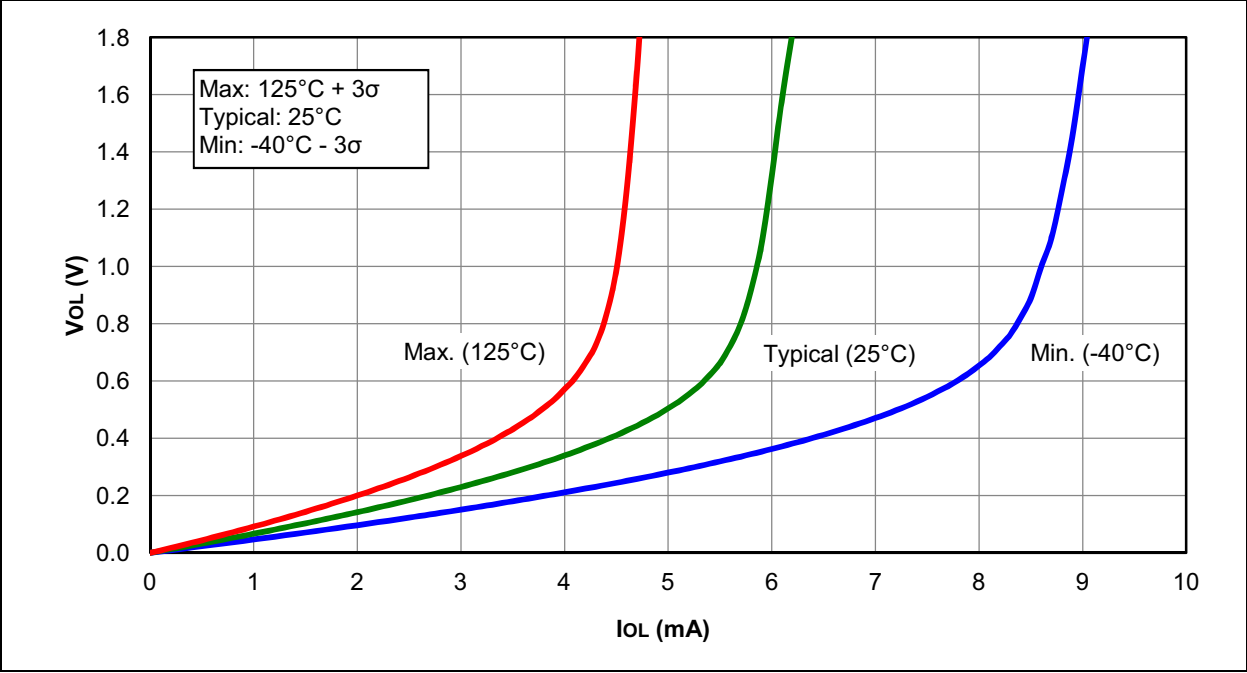




图 36-7: 在整个温度范围内高驱动能力引脚的  $V_{OH}$ — $I_{OH}$  曲线,  $V_{DD} = 5.0V$

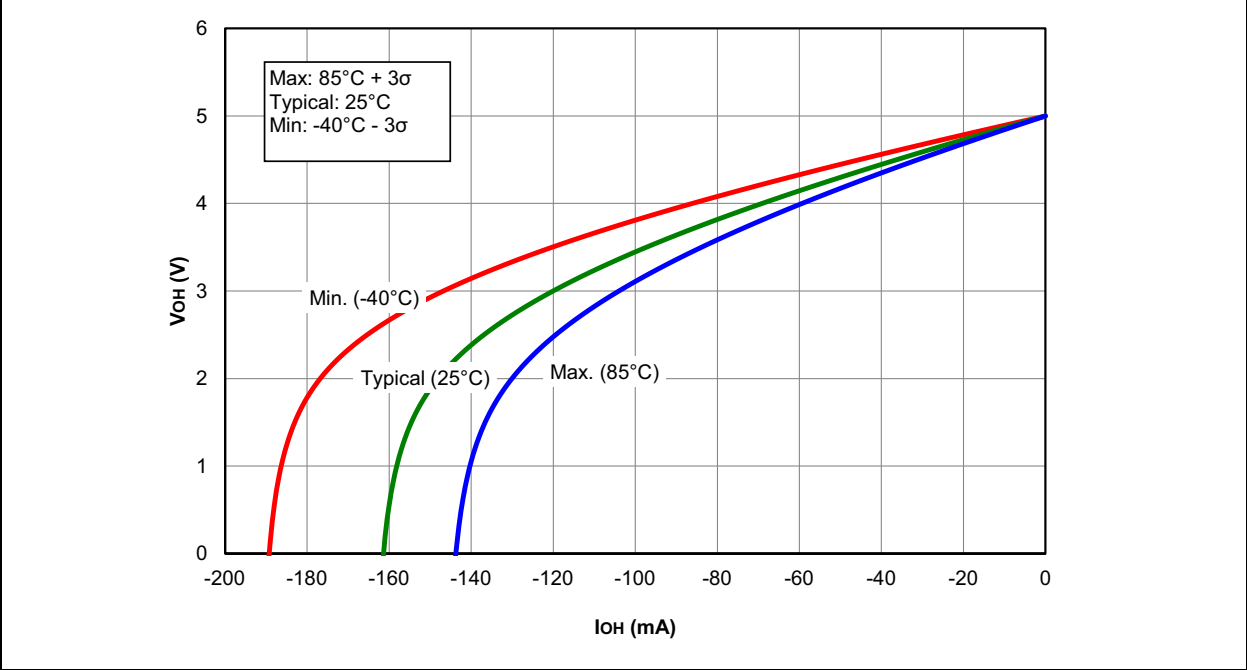


图 36-8: 在整个温度范围内高驱动能力引脚的  $V_{OL}$ — $I_{OL}$  曲线,  $V_{DD} = 5.0V$

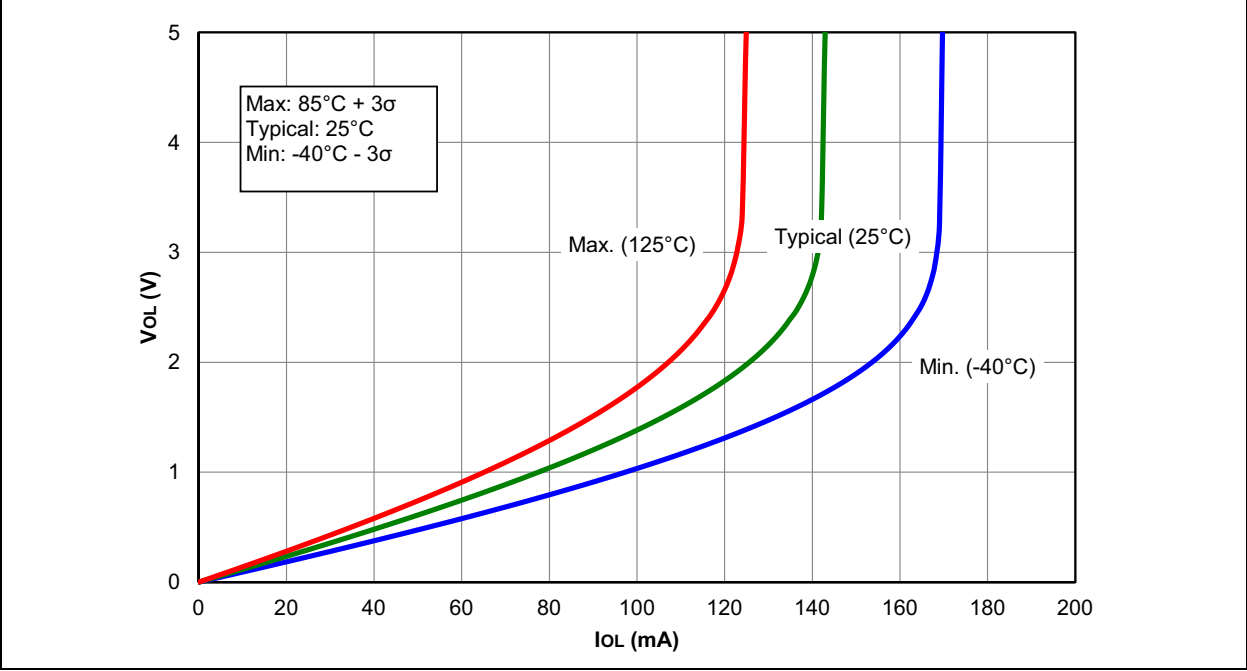


图36-9: POR 释放电压

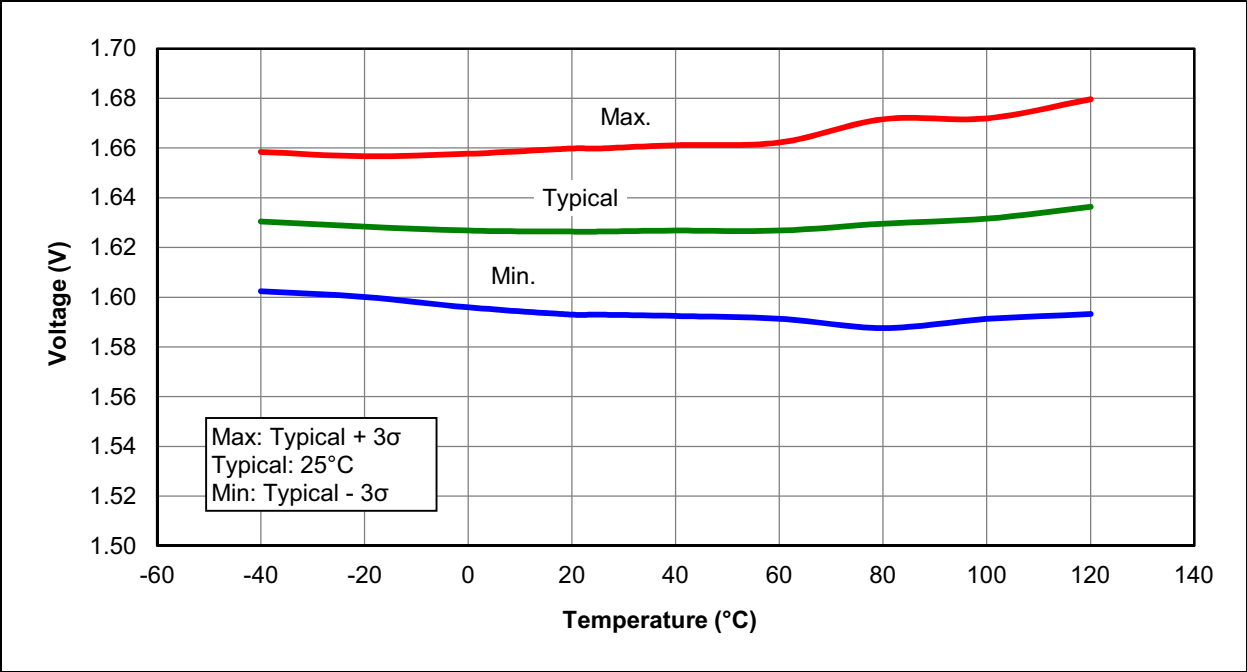


图36-10: POR 重新激活电压, 仅限 PIC16F1615/9

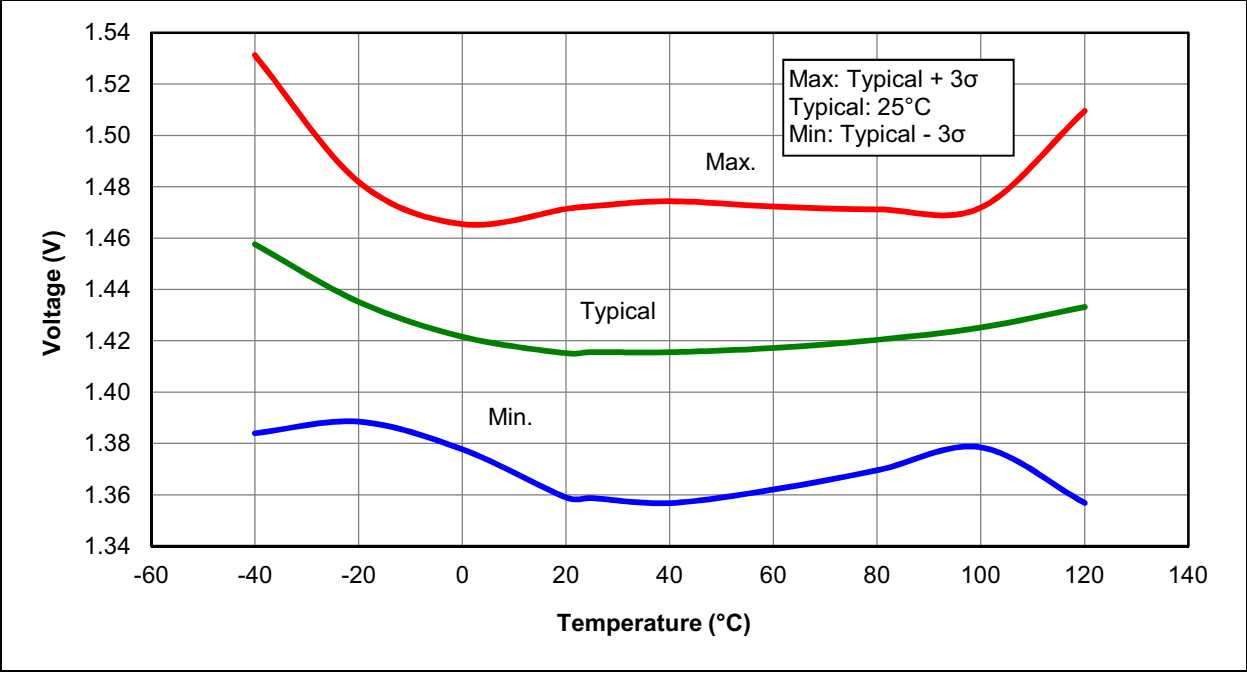


图36-11： 欠压复位电压，BORV = 1，仅限PIC16LF1615/9

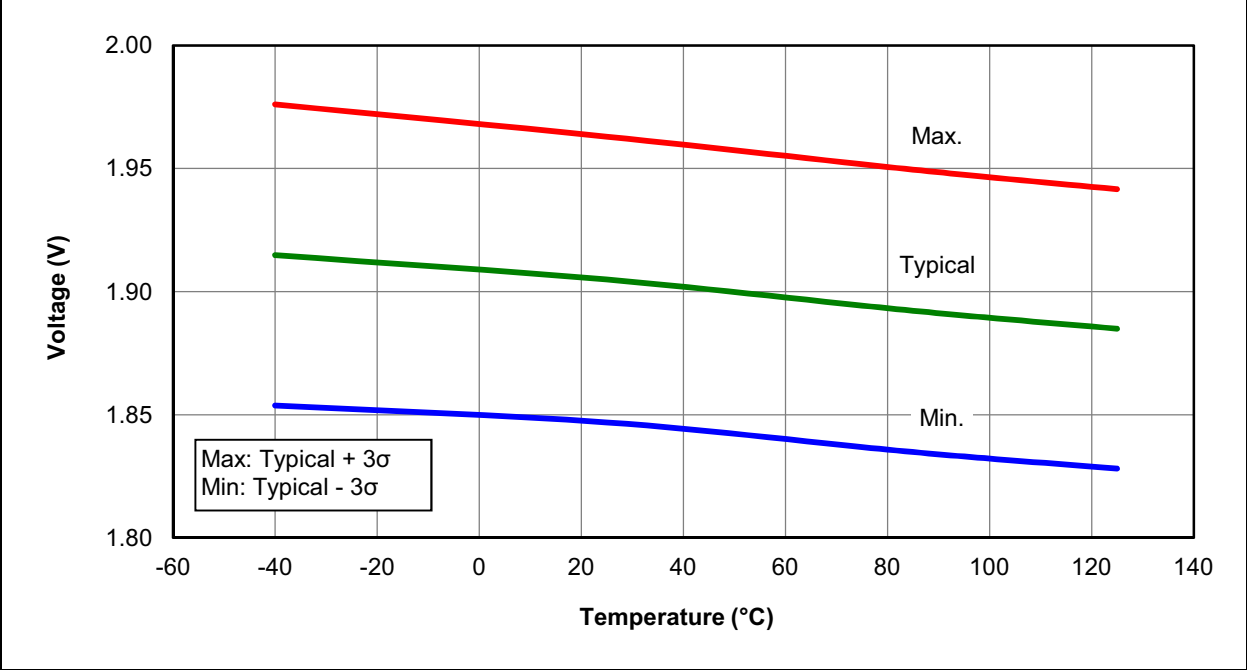


图36-12： 欠压复位滞后电压，BORV = 1，仅限PIC16LF1615/9

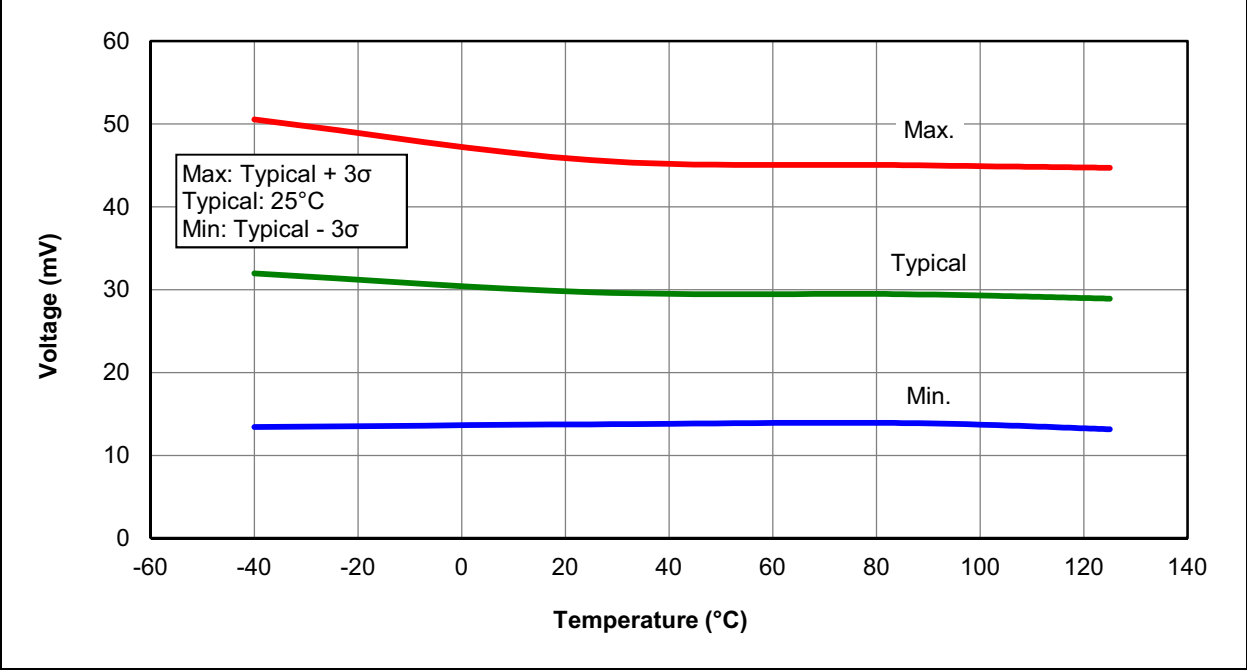


图36-13: 欠压复位电压, BORV = 1, 仅限PIC16F1615/9

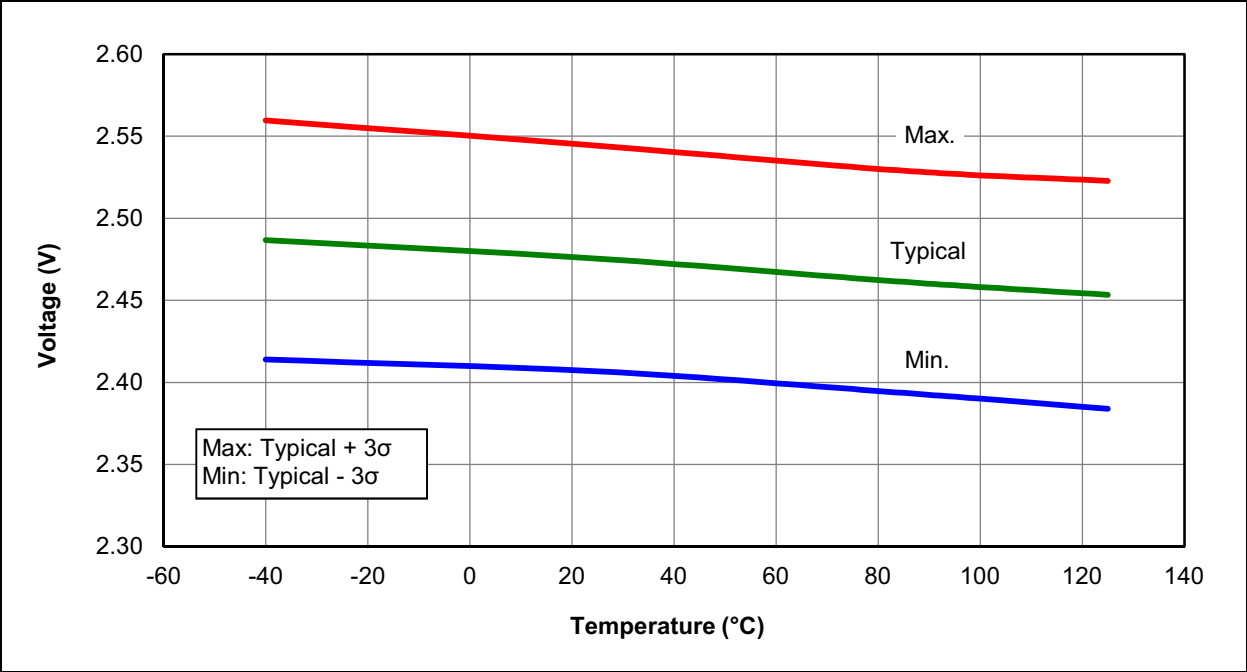


图36-14: 欠压复位滞后电压, BORV = 1, 仅限PIC16F1615/9

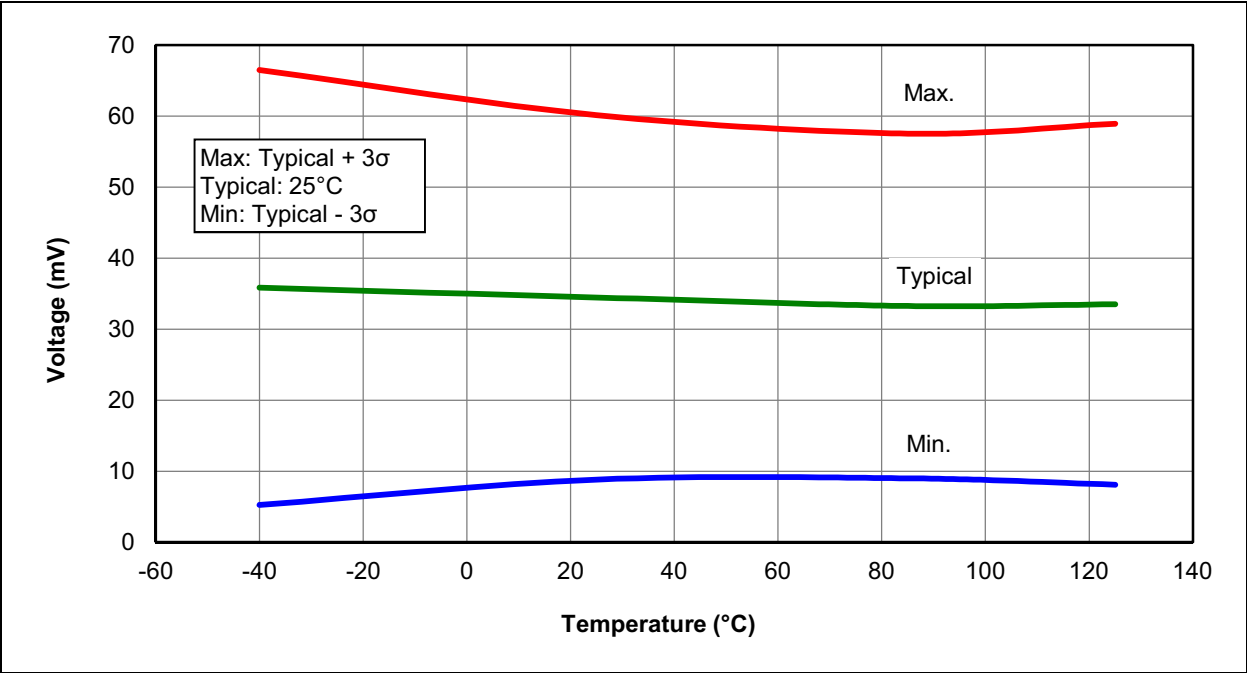


图36-15: 欠压复位电压, BORV = 0

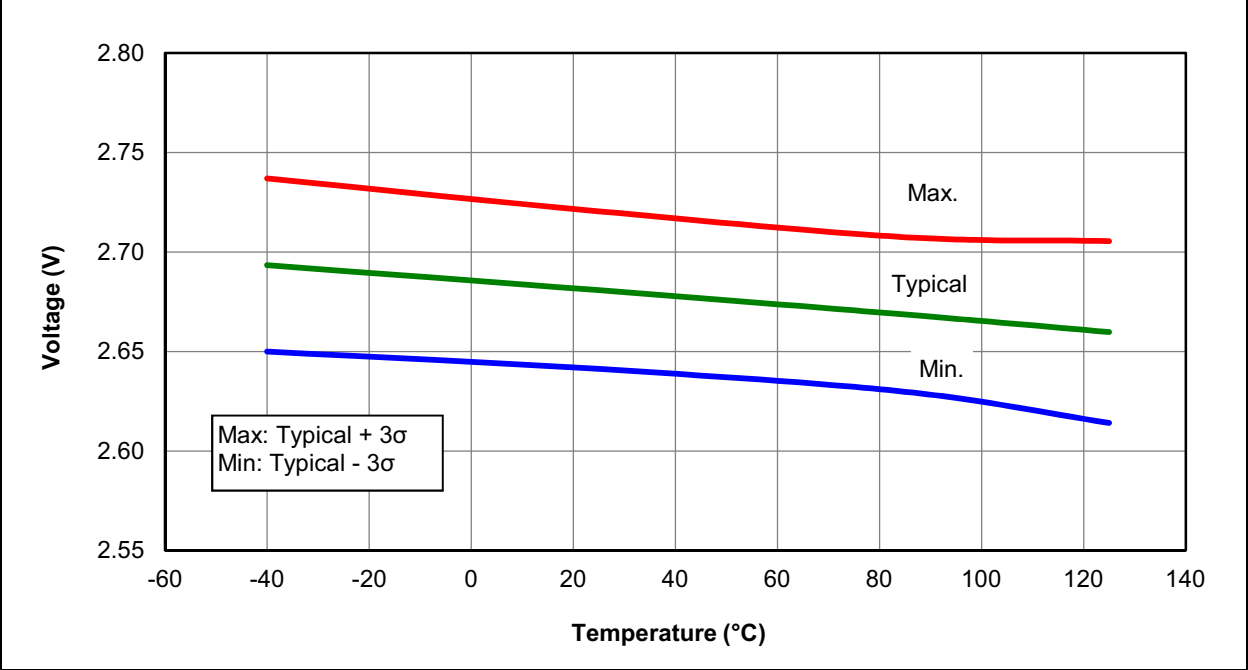


图36-16: 欠压复位滞后电压, BORV = 0

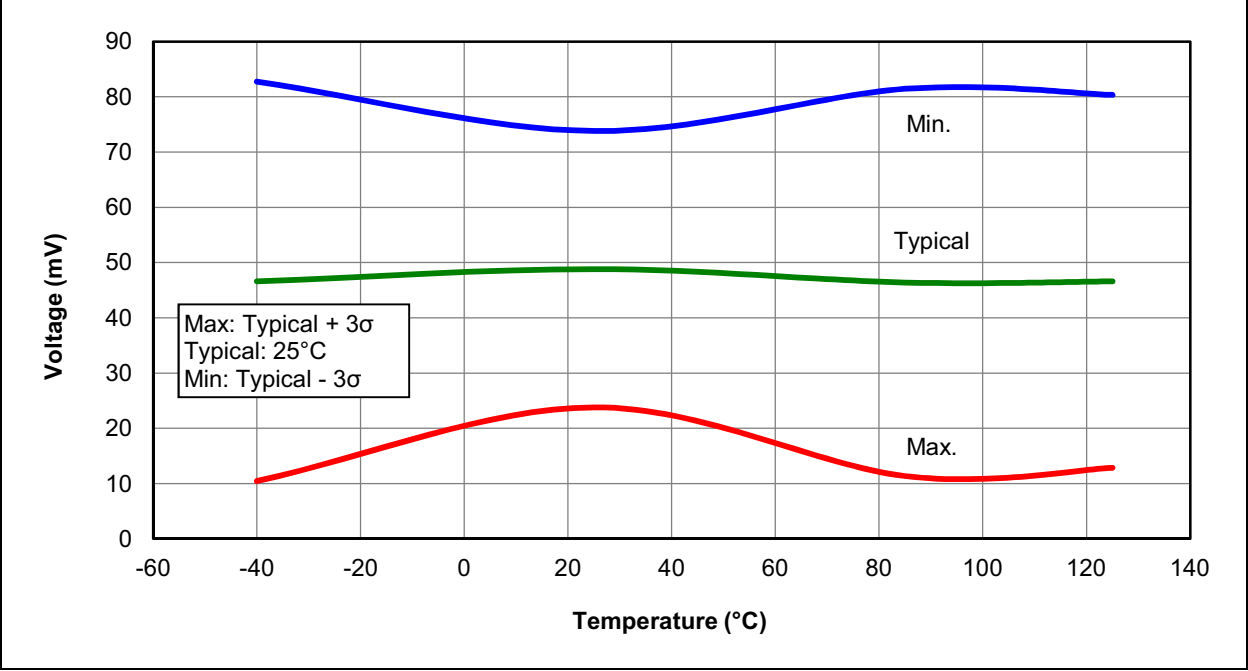


图36-17： 低功耗欠压复位电压，LPBOR = 0

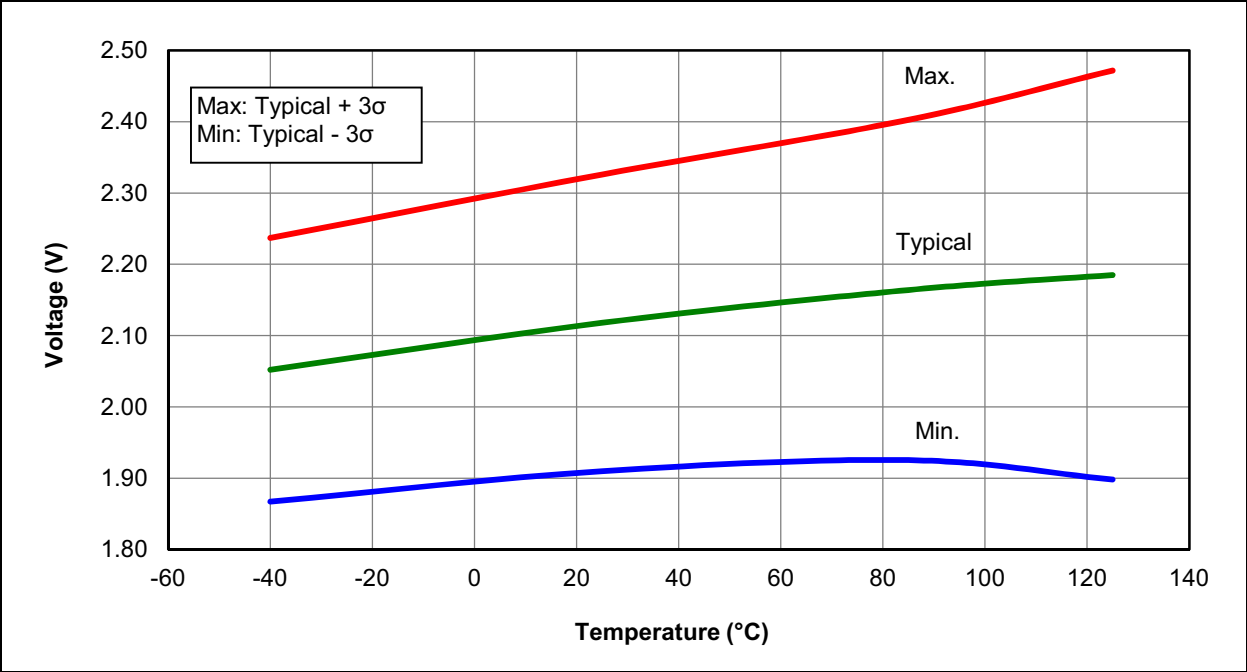


图36-18： 低功耗欠压复位滞后电压，LPBOR = 0

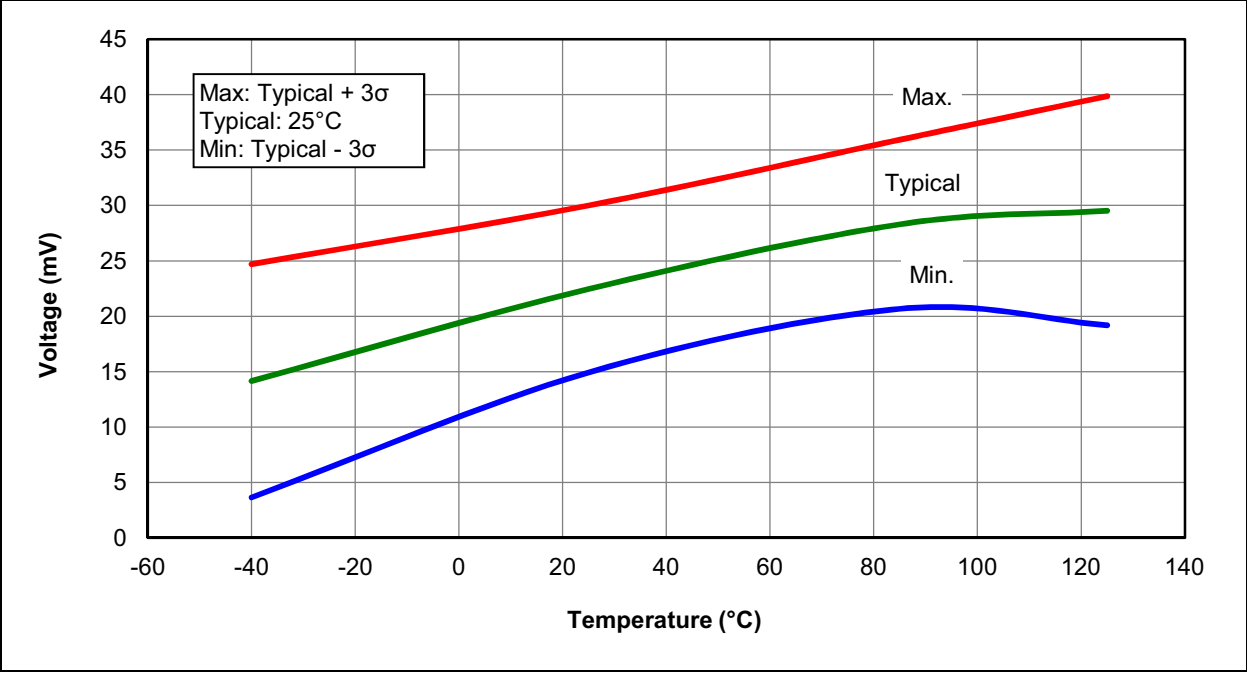


图36-19: WDT 超时周期

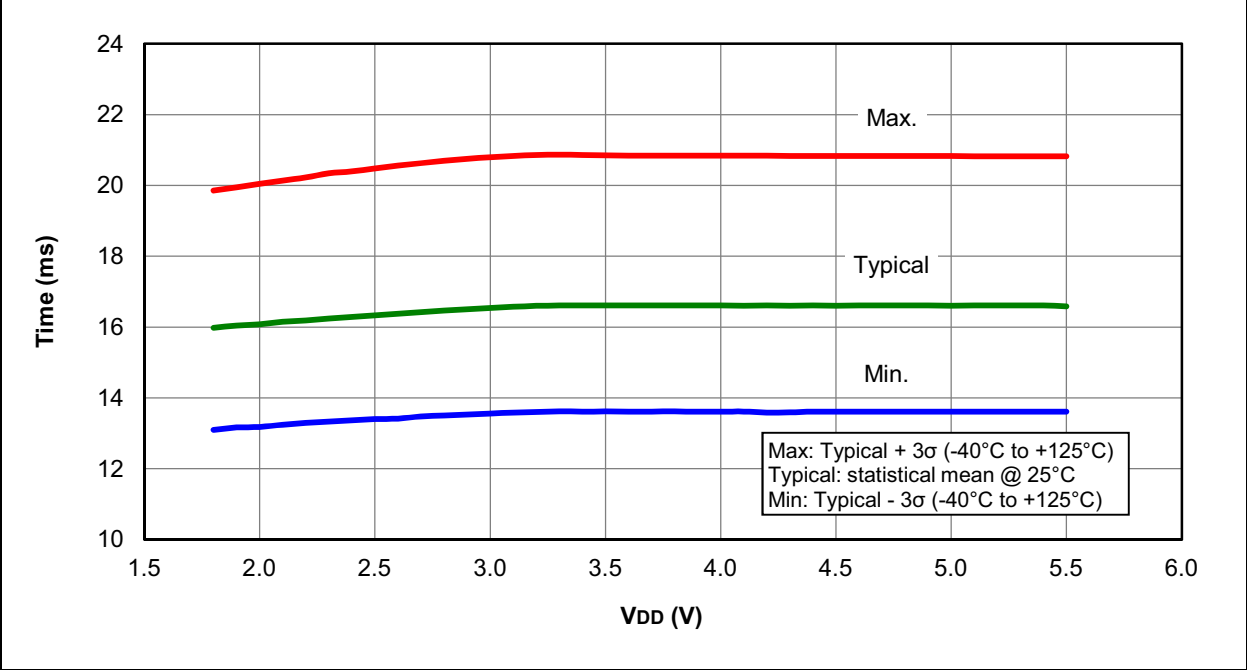


图36-20: PWRT 周期

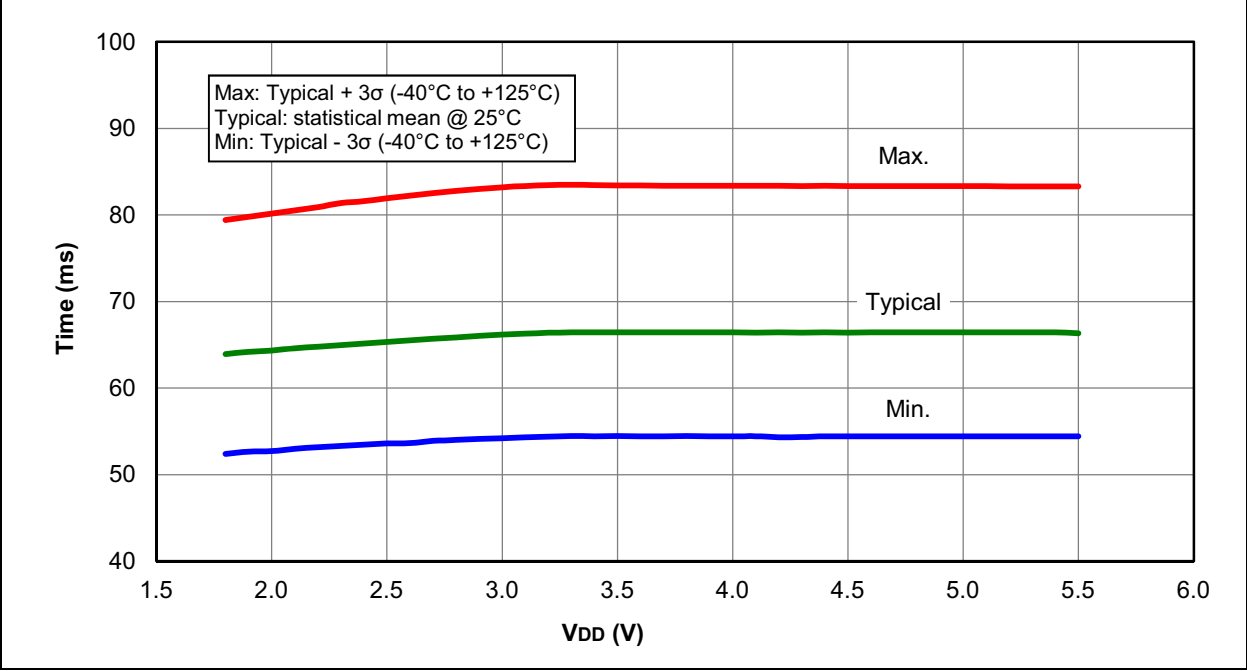


图36-21： FVR 稳定周期

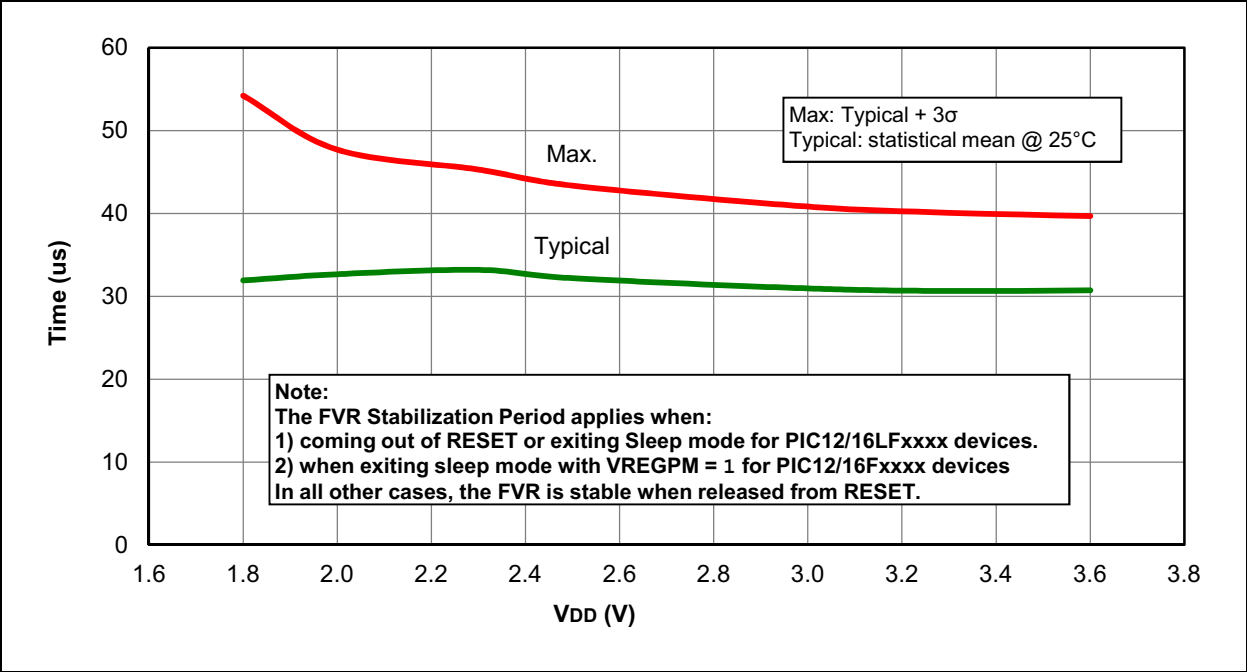




图36-22: 比较器滞后电压, 正常功耗模式 (CxSP = 1, CxHYS = 1)

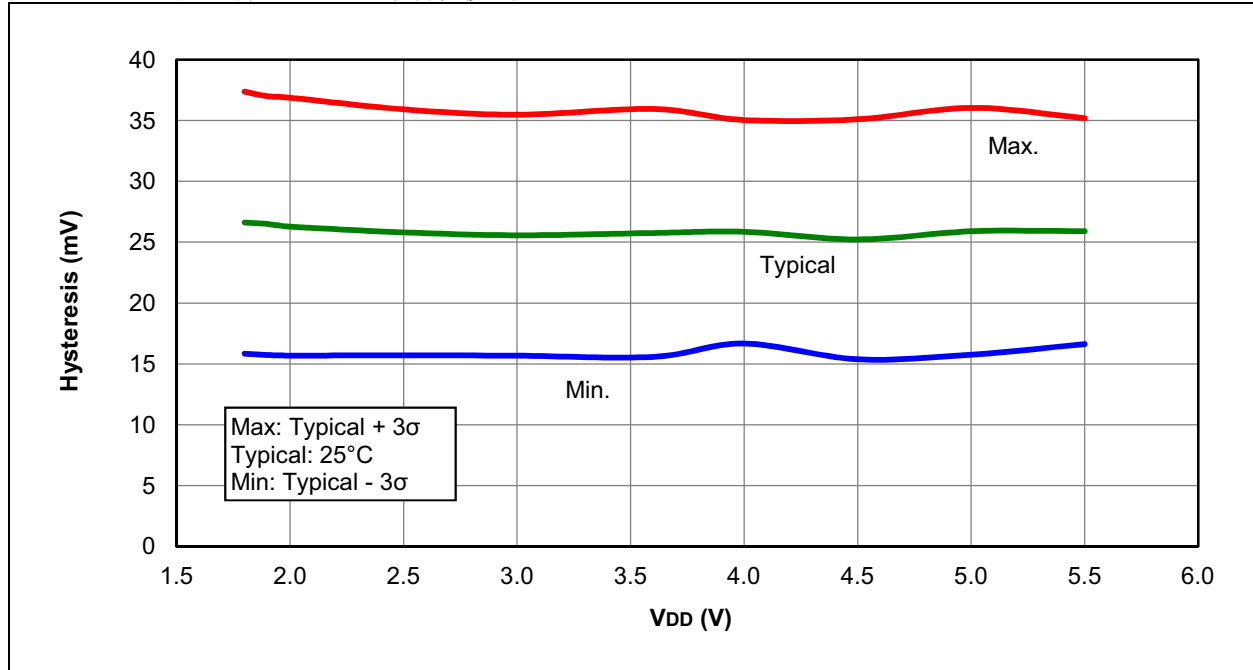


图36-23: 比较器滞后电压, 低功耗模式 (CxSP = 0, CxHYS = 1)

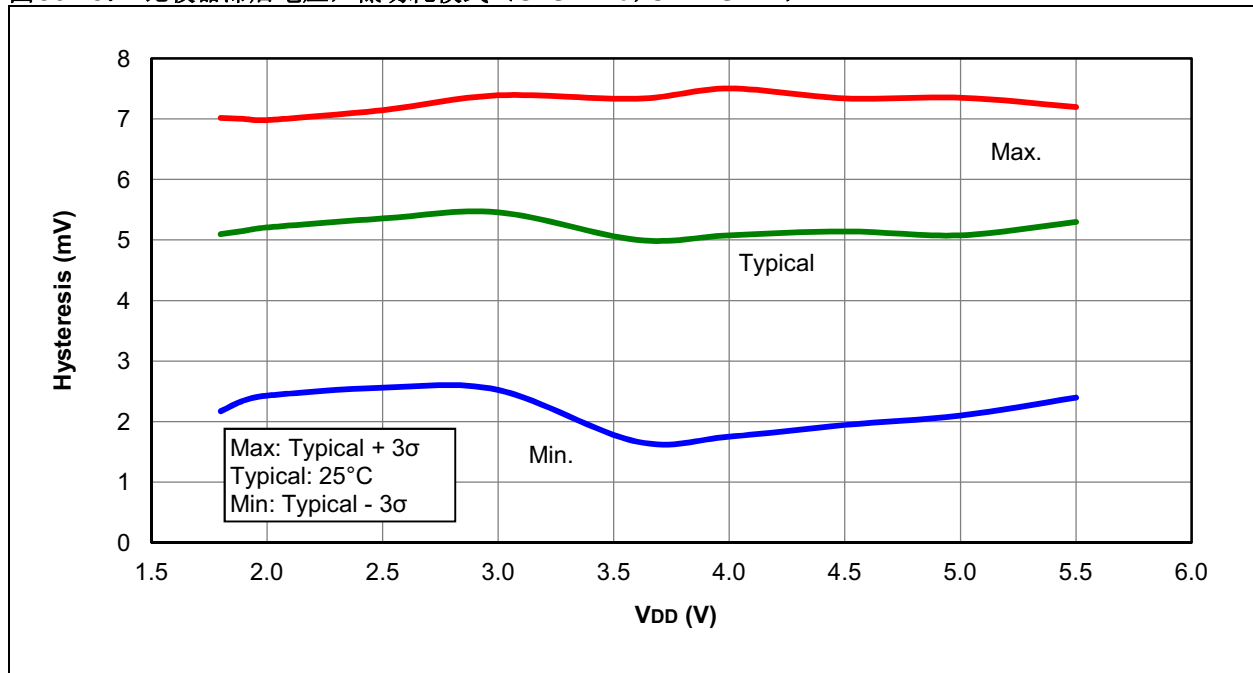


图36-24: 比较器响应时间, 正常功耗模式 (CxSP = 1)

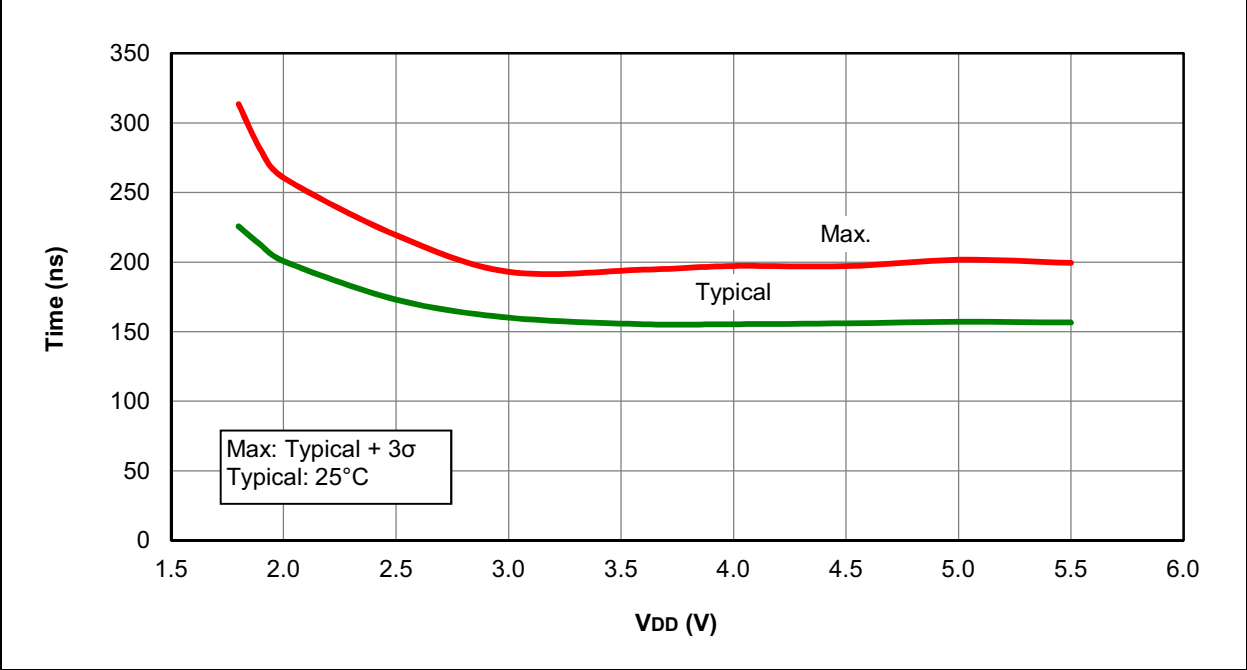


图36-25: 在整个温度范围内的比较器响应时间, 正常功耗模式 (CxSP = 1)

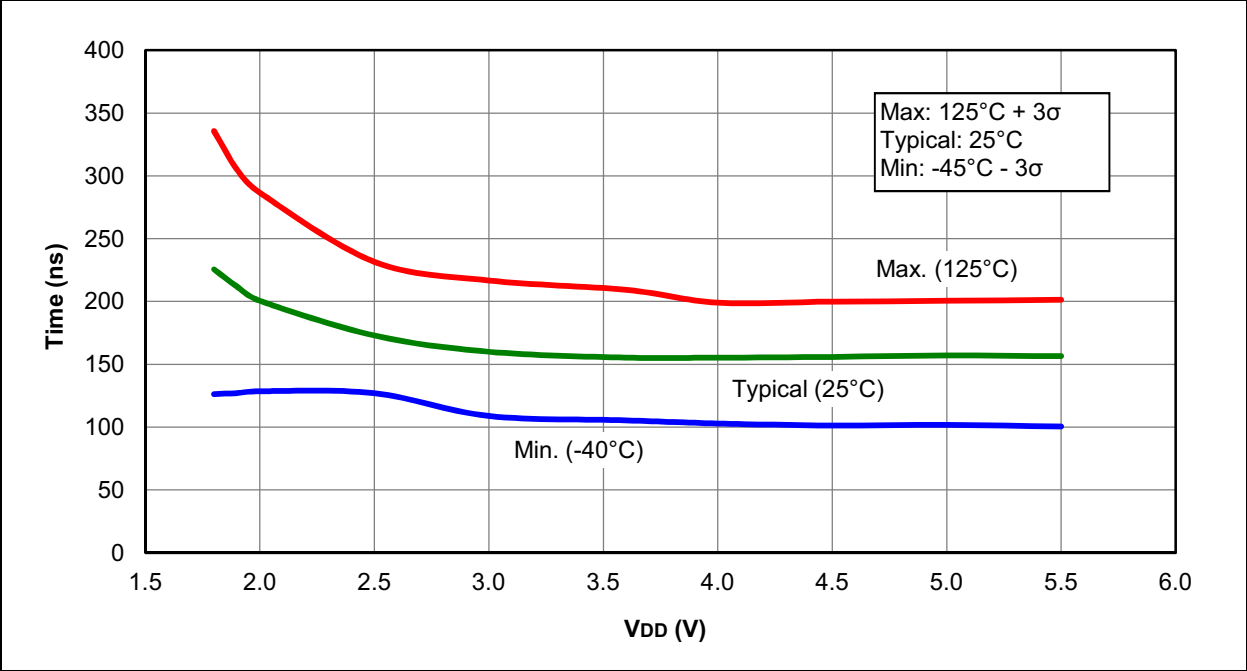
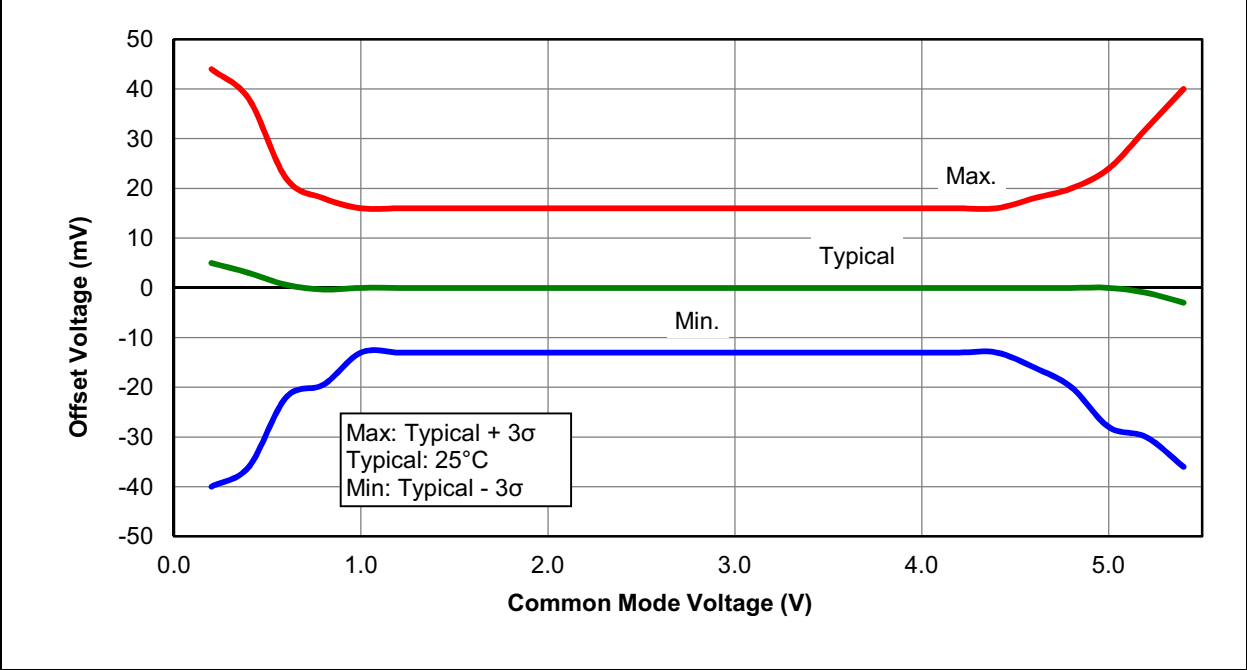


图 36-26: 25°C 下的比较器输入失调电压, 正常功耗模式 (CxSP = 1), 仅限 PIC16F1615/9



# PIC16(L)F1615/9

图36-27: 在整个VDD和温度范围内的LFINTOSC频率, 仅限PIC16LF1615/9

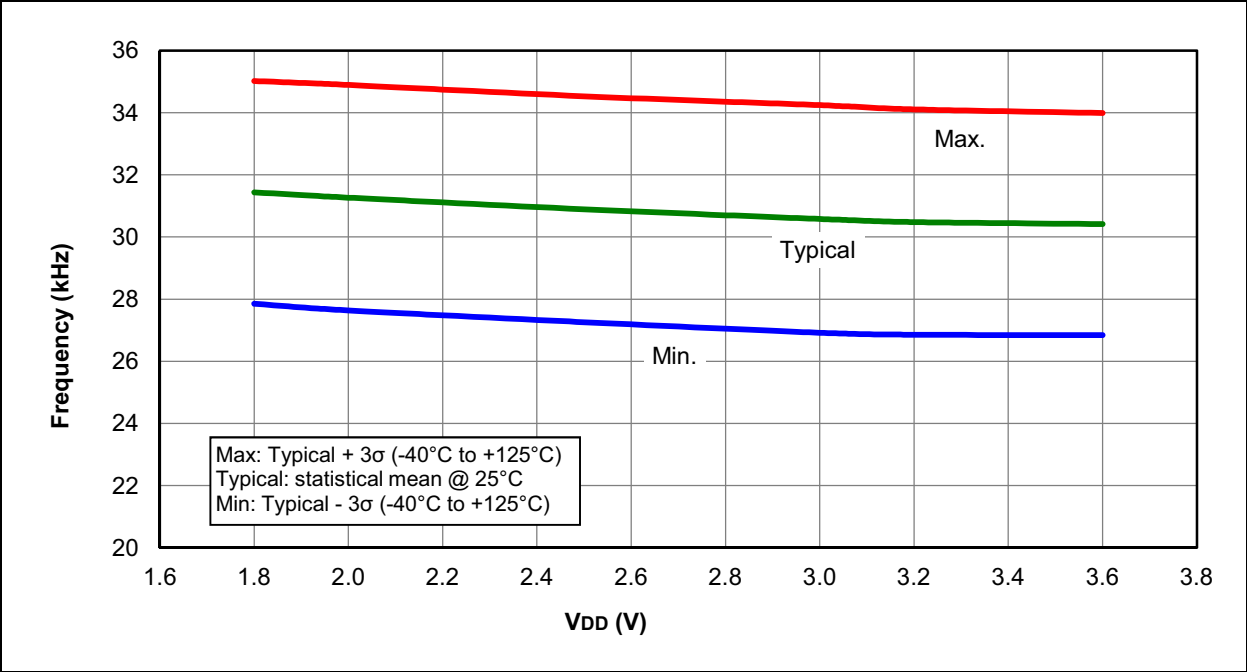


图36-28: 在整个VDD和温度范围内的LFINTOSC频率, 仅限PIC16F1615/9

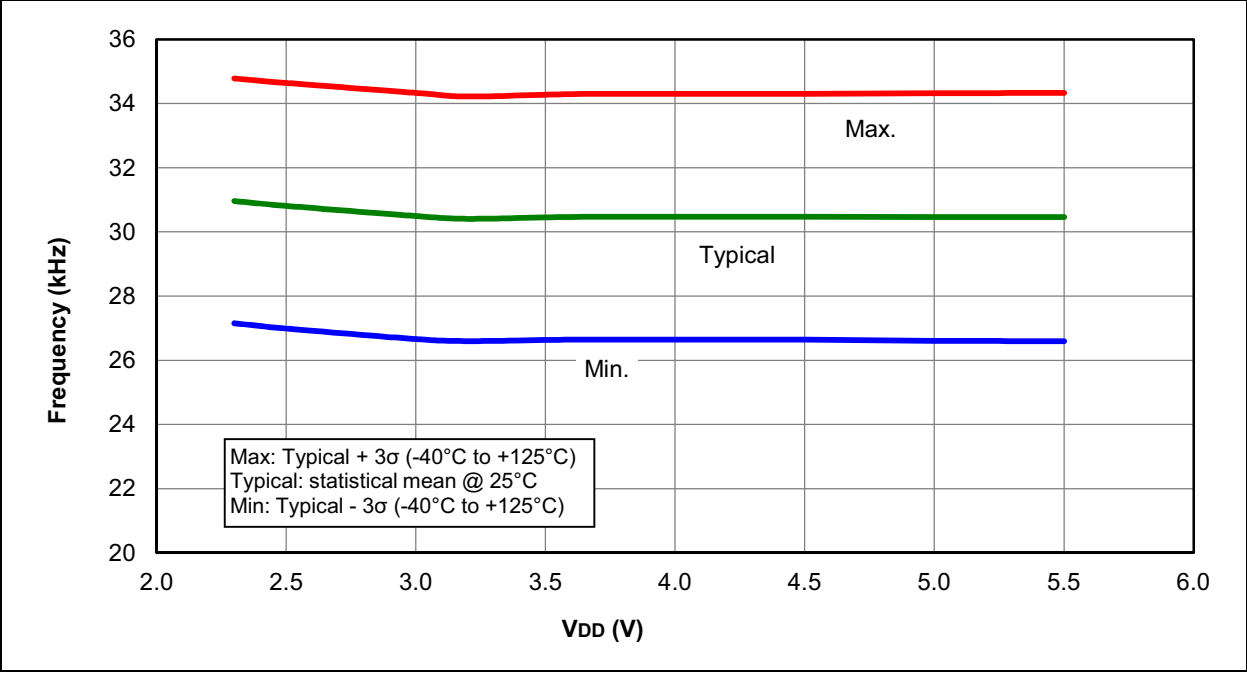


图36-29： 休眠模式，唤醒周期与 HFINTOSC 源，仅限 PIC16LF1615/9

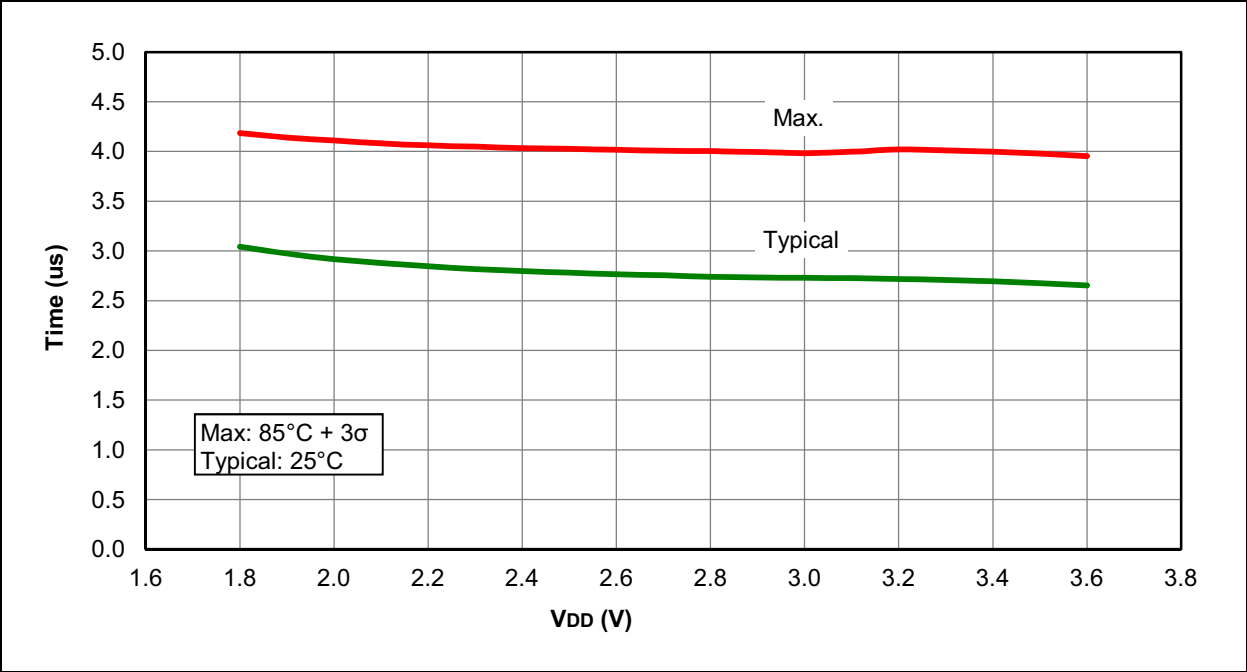


图36-30： 低功耗休眠模式，唤醒周期与 HFINTOSC 源，VREGPM = 1，仅限 PIC16F1615/9

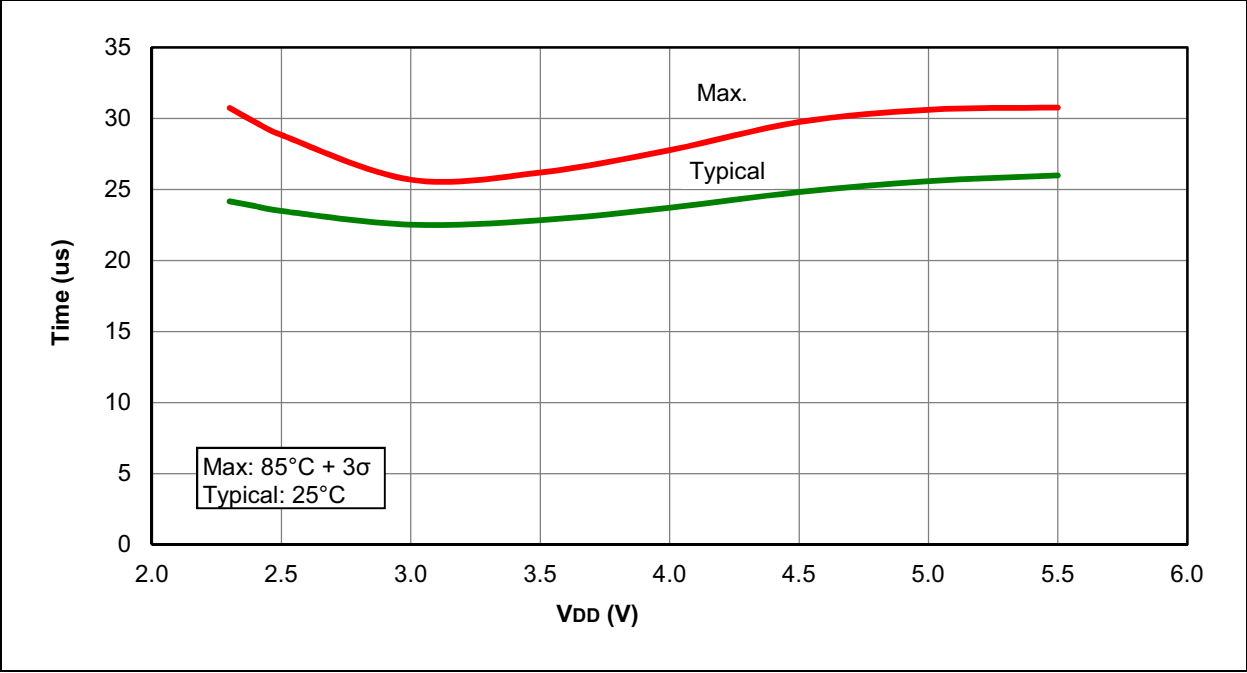
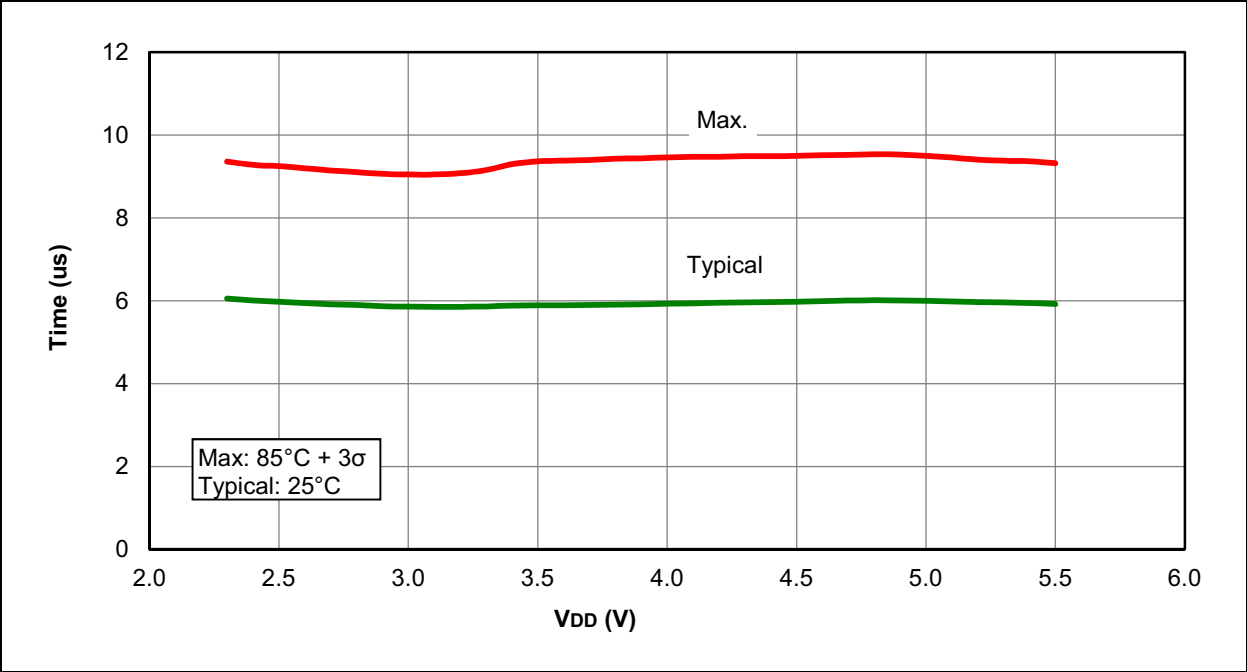


图36-31： 休眠模式，唤醒周期与HFINTOSC源，VREGPM = 0，仅限PIC16F1615/9



## 37.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机（MCU）和 dsPIC® 数字信号控制器（DSC）提供支持：

- 集成开发环境
  - MPLAB® X IDE 软件
- 编译器 / 汇编器 / 链接器
  - MPLAB XC 编译器
  - MPASM™ 汇编器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - 适用于各种器件系列的 MPLAB 汇编器 / 链接器 / 库管理器
- 模拟器
  - MPLAB X SIM 软件模拟器
- 仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器 / 编程器
  - MPLAB ICD 3
  - PICKit™ 3
- 器件编程器
  - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包
- 第三方开发工具

## 37.1 MPLAB X 集成开发环境软件

MPLAB X IDE 是适用于 Microchip 和第三方软硬件开发工具统一的通用图形用户界面，可以在 Windows®、Linux 和 Mac OS® X 上运行。MPLAB X IDE 是一款全新的 IDE，它基于 NetBeans IDE，包含许多免费的软件组件和插件，适用于高性能的应用程序开发和调试。通过这一无缝交互的用户界面，在不同工具之间的迁移以及从软件模拟器到硬件调试和编程工具的升级都变得极为简便。

MPLAB X IDE 具有完善的项目管理、可视化的调用图、可配置的观察窗口以及包含代码补全功能和上下文菜单的功能丰富编辑器，因此对于新用户来说非常灵活和友好。MPLAB X IDE 支持对多个项目使用多个工具和同时调试，因此也完全可以满足经验丰富用户的需求。

功能丰富的编辑器：

- 彩色高亮显示语法
- 智能代码补全功能，在输入代码时提供建议和提示
- 基于用户定义规则，代码自动格式化
- 即时解析

用户友好的可定制界面：

- 完全可定制界面：工具栏、工具栏图标、窗口和窗口放置等
- 调用图窗口

基于项目的工作空间：

- 多个项目
- 多个工具
- 多种配置
- 同时调试会话

文件历史和错误跟踪：

- 本地文件历史功能
- 内建对 Bugzilla 缺陷跟踪系统的支持

## 37.2 MPLAB XC 编译器

MPLAB XC编译器是适用于Microchip所有8位、16位和32位MCU以及DSC器件的完全ANSI C编译器。这些编译器提供强大的集成功能以及出色的代码优化功能，且易于使用。MPLAB XC编译器可在Windows、Linux或Mac OS X上运行。

为方便进行源代码级调试，编译器提供了已针对MPLAB X IDE优化的调试信息。

MPLAB XC编译器的免费版支持所有器件和命令，没有时间或存储容量限制，且为大多数应用程序提供了充分的代码优化。

MPLAB XC编译器包含汇编器、链接器和实用程序。汇编器生成可重定位目标文件，然后通过链接器将生成的可重定位目标文件与其他可重定位目标文件或归档文件归档或链接在一起，进而生成可执行文件。MPLAB XC编译器使用汇编器来生成目标文件。汇编器具有如下突出特性：

- 支持全部器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的伪指令集
- 灵活的宏语言
- 与MPLAB X IDE兼容

## 37.3 MPASM 汇编器

MPASM汇编器是全功能通用宏汇编器，适用于PIC10/12/16/18 MCU。

MPASM汇编器可生成用于MPLINK目标链接器的可重定位目标文件、Intel®标准HEX文件、详细描述存储器使用状况和符号参考的MAP文件、包含源代码行及生成机器码的绝对LST文件以及用于调试的COFF文件。

MPASM汇编器具有如下特性：

- 集成在MPLAB X IDE项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

## 37.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK目标链接器组合由MPASM汇编器生成的可重定位目标文件。通过使用链接器脚本中的伪指令，它还可链接预编译库中的可重定位目标文件。

MPLIB目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

## 37.5 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB汇编器为PIC24和PIC32 MCU以及dsPIC DSC器件从符号汇编语言生成可重定位机器码。MPLAB XC编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特性：

- 支持整个器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的指令集
- 与MPLAB X IDE兼容



## 37.6 MPLAB X SIM 软件模拟器

MPLAB X SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC DSC 进行模拟,可在 PC 主机环境下进行代码开发。对于任何给定的指令,都可以对数据区进行检查或修改,并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中,以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器。

MPLAB X SIM 软件模拟器完全支持使用 MPLAB XC 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码,是一款完美且经济的软件开发工具。

## 37.7 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件推出的新一代高速仿真器。结合 MPLAB X IDE 易于使用且功能强大的图形用户界面,该仿真器可对所有 8 位、16 位和 32 位 MCU 及 DSC 器件进行调试和编程。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连,并利用与在线调试器系统兼容的连接器和 (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。

## 37.8 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器,适用于 Microchip 的闪存 DSC 和 MCU 器件。结合 MPLAB X IDE 功能强大但易于使用的图形用户界面,该调试器可对 PIC 闪存单片机和 dsPIC DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连,并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器和 (RJ-11) 与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 连接器。

## 37.9 PICKit 3 在线调试器 / 编程器

结合 MPLAB X IDE 功能强大的图形用户界面, MPLAB PICKit 3 可对 PIC 闪存单片机和 dsPIC 数字信号控制器进行调试和编程,且价位较低。MPLAB PICKit 3 通过全速 USB 接口与设计工程师的 PC 相连,并利用 Microchip 调试连接器 (RJ-11) (与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容) 与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程 (In-Circuit Serial Programming™, ICSP™)。

PICKit 3 Debug Express 包括 PICKit 3、演示板和单片机、连接电缆和光盘 (内含用户指南、课程、教程、编译器、MPLAB IDE 软件)。

## 37.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器,在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器 (128 x 64), 以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP 电缆。在单机模式下, MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC MCU 和 dsPIC DSC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法,可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡,用于文件存储及数据应用。

## 37.11 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVAL® 评估系统、 $\Sigma\text{-}\Delta$  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](http://www.microchip.com))。

## 37.12 第三方开发工具

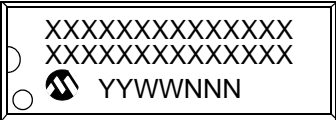
Microchip 还提供一些来自第三方供应商的优秀开发工具。这些工具均经过精心挑选，功能独特，物有所值。

- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 ILogika® 等公司提供的嵌入式以太网解决方案

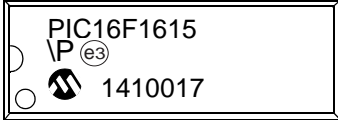
38.0 封装信息

38.1 封装标识信息

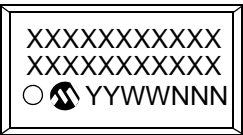
14 引脚 PDIP



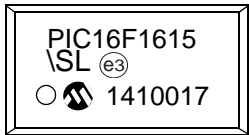
示例



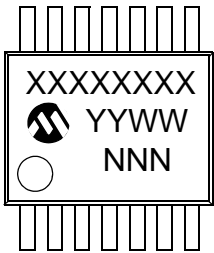
14 引脚 SOIC (0.150 英寸)



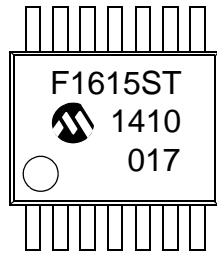
示例



14 引脚 TSSOP



示例



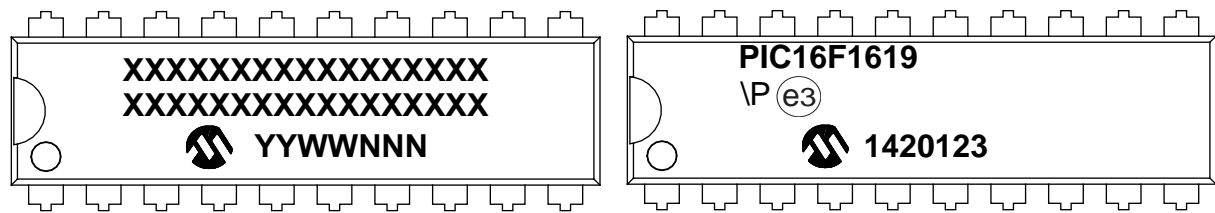
图注:	XX...X	客户指定信息
	Y	年份代码 (日历年的最后一位数字)
	YY	年份代码 (日历年的最后两位数字)
	WW	星期代码 (一月一日的星期代码为“01”)
	NNN	以字母数字排序的追踪代码
	(e3)	雾锡 (Matte Tin, Sn) 的 JEDEC 无铅标志
	*	表示无铅封装。JEDEC 无铅标志 ((e3)) 标示于此种封装的外包装上。
注:	Microchip 部件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户指定信息的字符数。	

# PIC16(L)F1615/9

## 38.1 封装标识信息（续）

20 引脚 PDIP (300 mil)

示例



20 引脚 SOIC (7.50 mm)

示例



20 引脚 SSOP (5.30 mm)

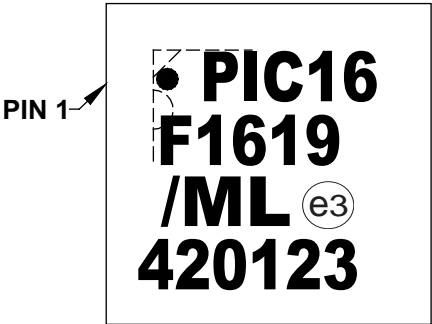
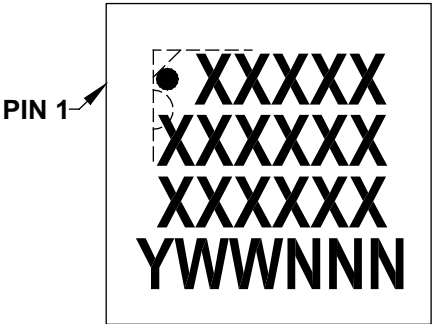
示例



38.1 封装标识信息（续）

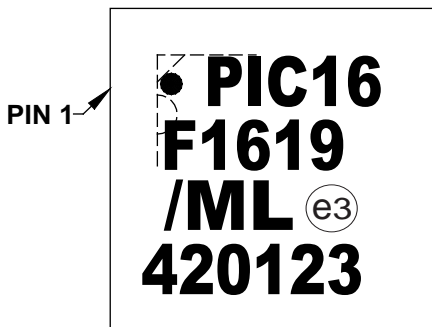
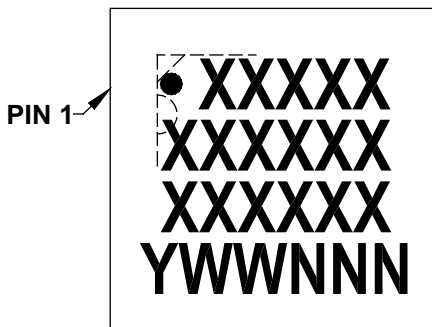
16 引脚 QFN (4x4x0.5 mm)

示例



20 引脚 QFN/UQFN (4x4x0.5 mm)

示例



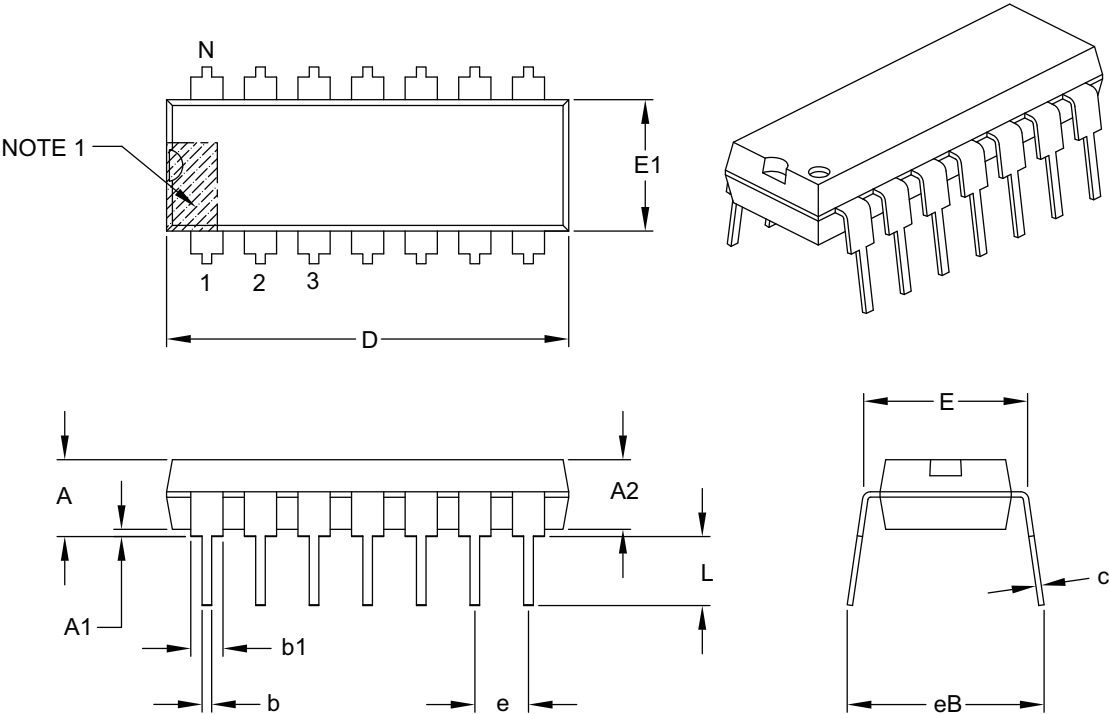
# PIC16(L)F1615/9

## 38.2 封装详细信息

以下部分将介绍各种封装的技术细节。

### 14引脚塑封双列直插式封装（P）—— 主体300 mil [PDIP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



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:**

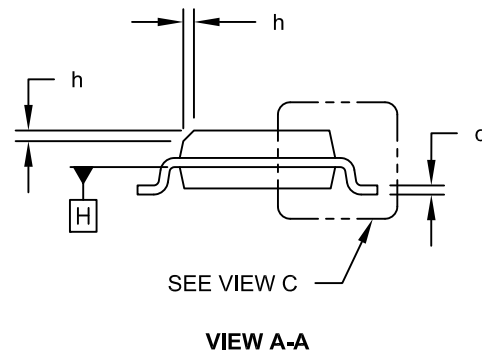
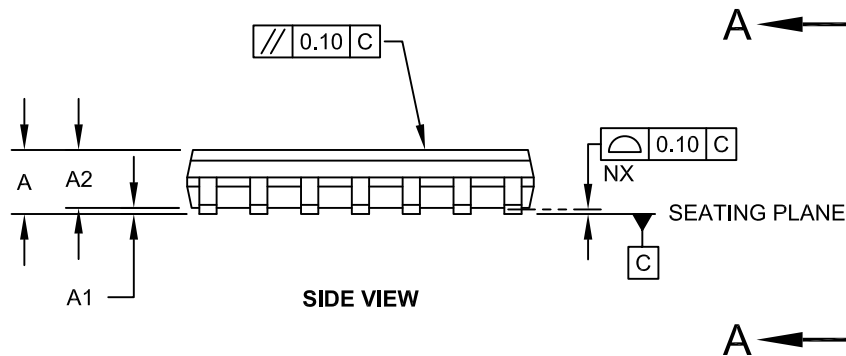
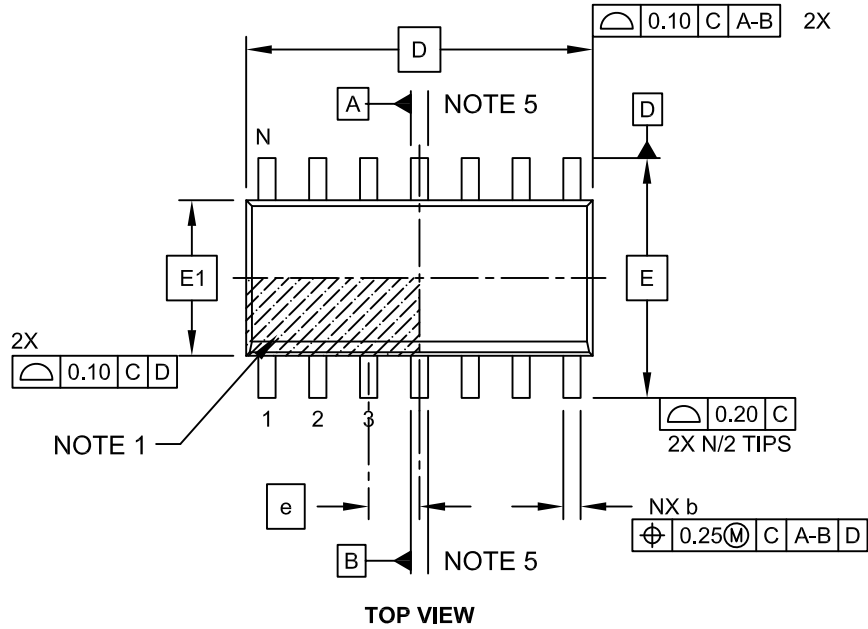
- 1. Pin 1 visual index feature may vary, but must be located with the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

## 14 引脚塑封窄条小外形封装 (SL) —— 主体 3.90 mm [SOIC]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。

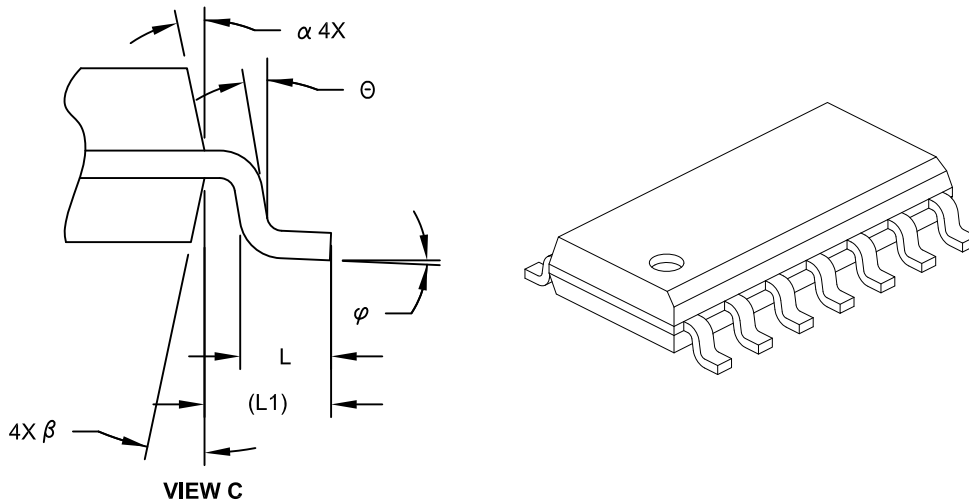


Microchip Technology Drawing No. C04-065C Sheet 1 of 2

# PIC16(L)F1615/9

## 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。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	8.65 BSC		
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1	1.04 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.10	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

### Notes:

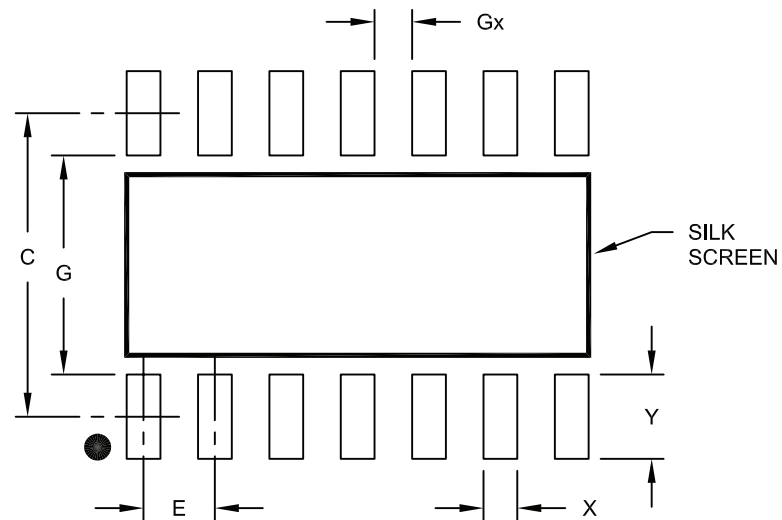
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
  - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
  - REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2



## 14 引脚塑封窄条小外形封装 (SL) —— 主体 3.90 mm [SOIC]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		5.40	
Contact Pad Width	X			0.60
Contact Pad Length	Y			1.50
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	3.90		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

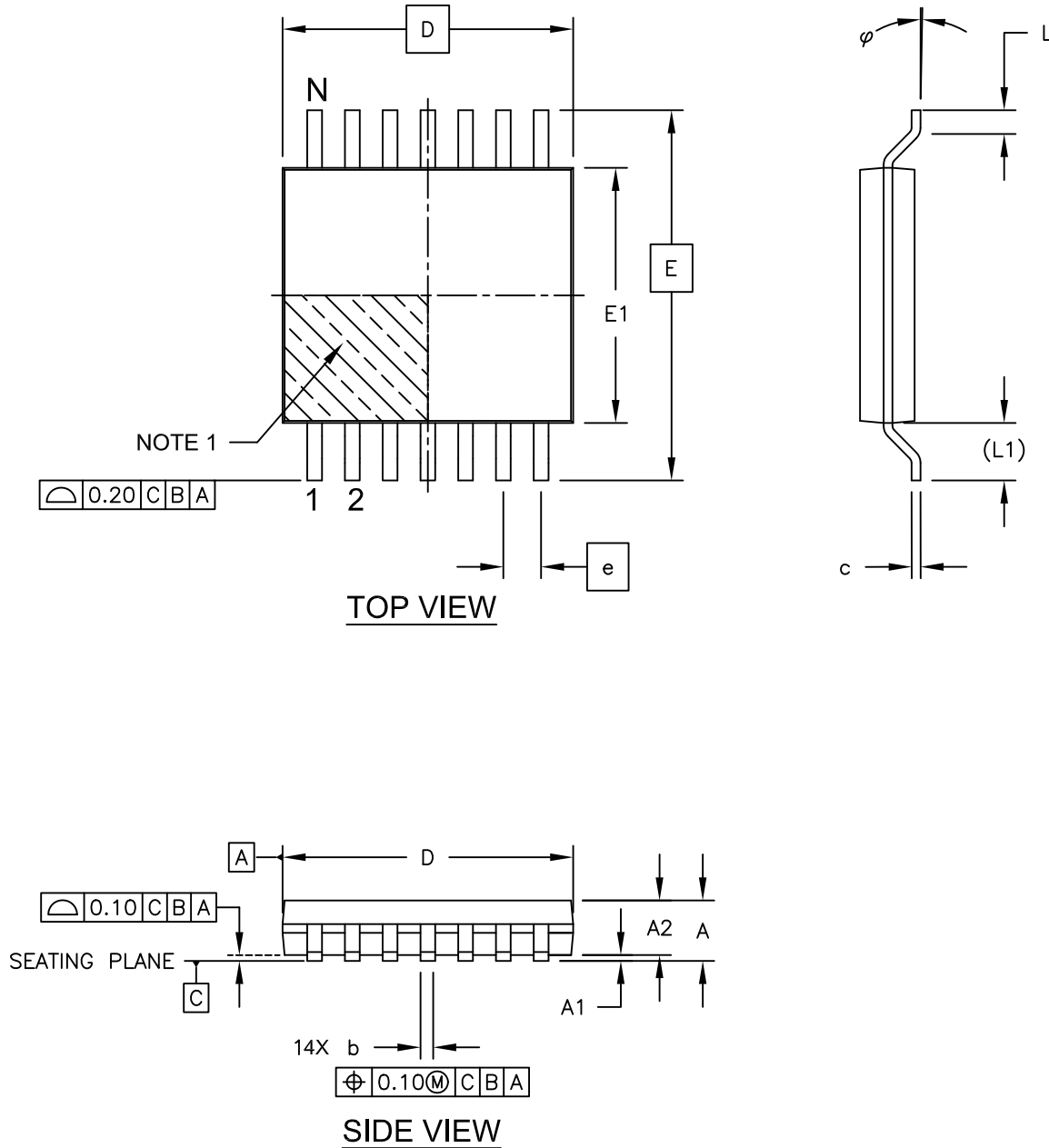
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

# PIC16(L)F1615/9

## 14引脚塑封薄型紧缩小外形封装（ST）——主体4.4 mm [TSSOP]

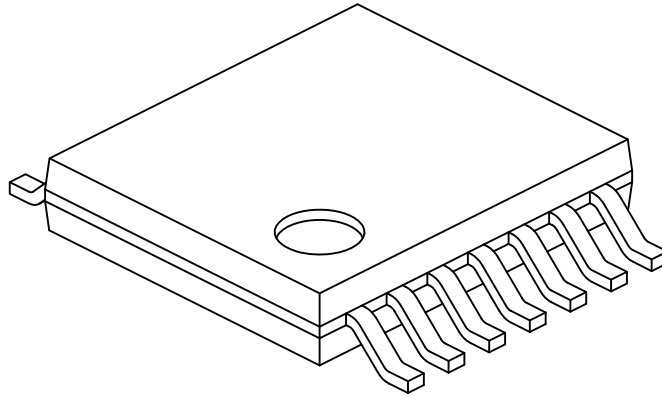
注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Microchip Technology Drawing C04-087C Sheet 1 of 2

## 14引脚塑封薄型紧缩小外形封装（ST）——主体4.4 mm [TSSOP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	0.65 BSC		
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	-	0.15
Overall Width	E	6.40 BSC		
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	(L1)	1.00 REF		
Foot Angle	$\varphi$	0°	-	8°
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.19	-	0.30

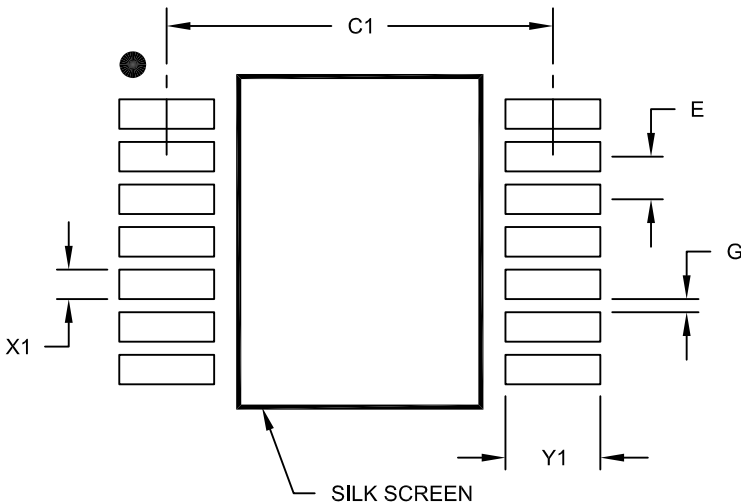
### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
  - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
  - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

14 引脚塑封薄型紧缩小外形封装（ST）—— 主体4.4 mm [TSSOP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C1		5.90	
Contact Pad Width (X14)	X1			0.45
Contact Pad Length (X14)	Y1			1.45
Distance Between Pads	G	0.20		

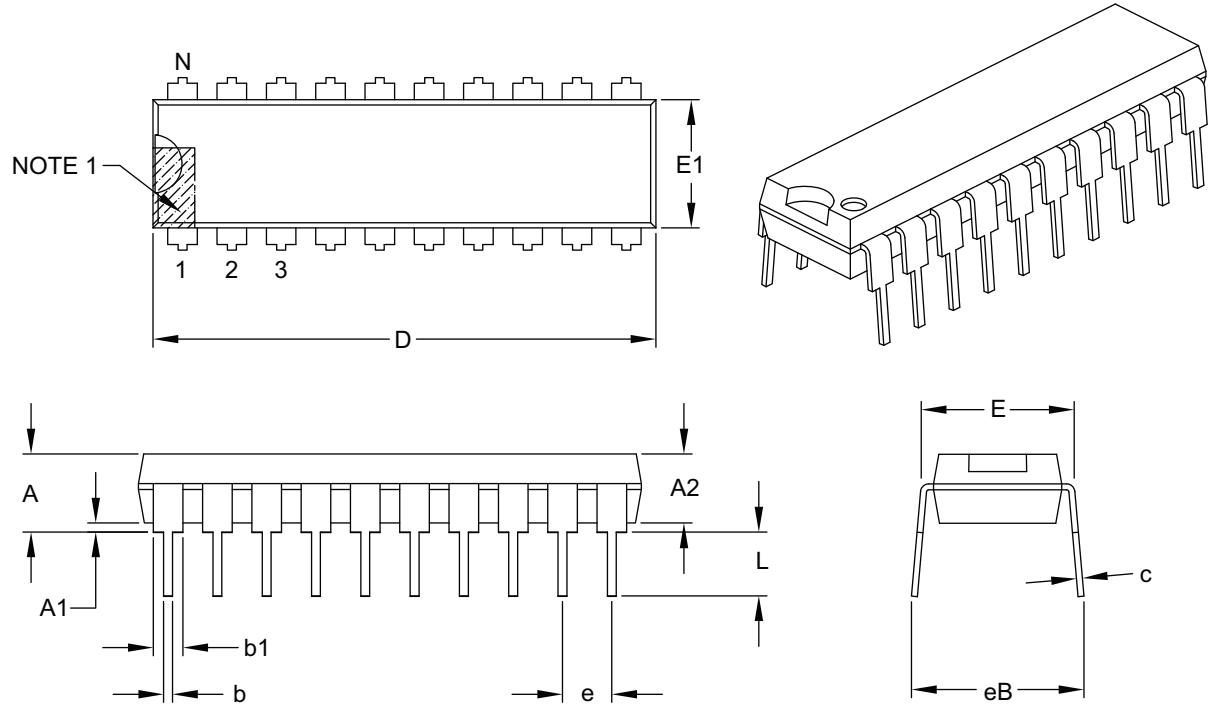
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

## 20 引脚塑封双列直插式封装（P）—— 主体 300 mil [PDIP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	.100 BSC		
Top to Seating Plane	A	—	—	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	—	—
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	—	—	.430

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

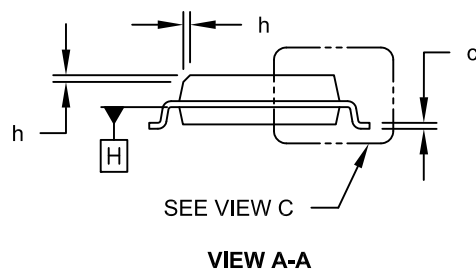
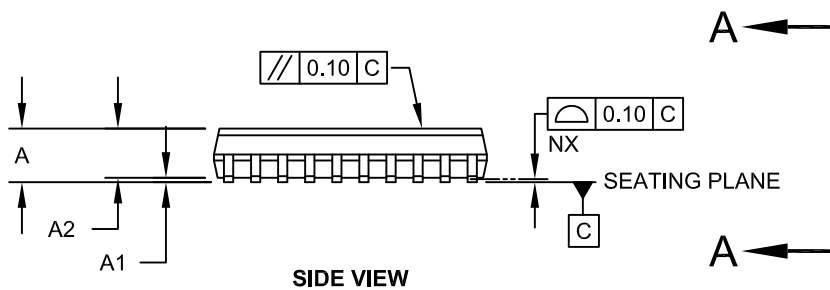
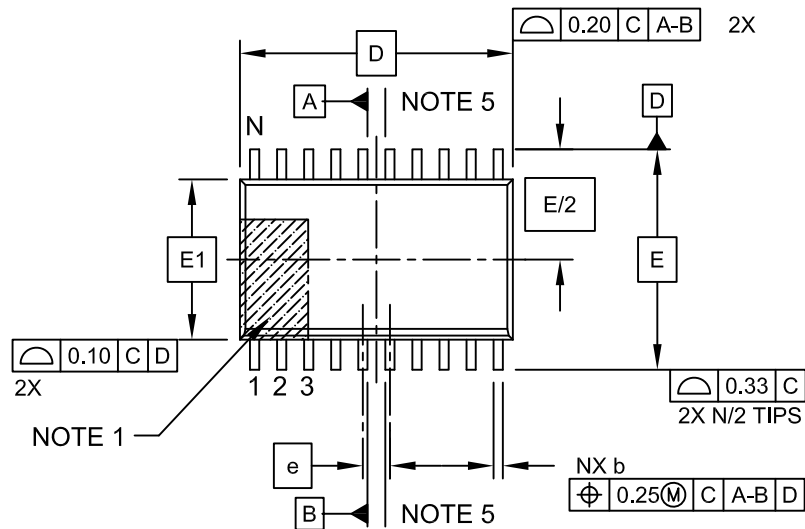
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B

# PIC16(L)F1615/9

## 20引脚塑封宽条小外形封装（SO）——主体7.50 mm [SOIC]

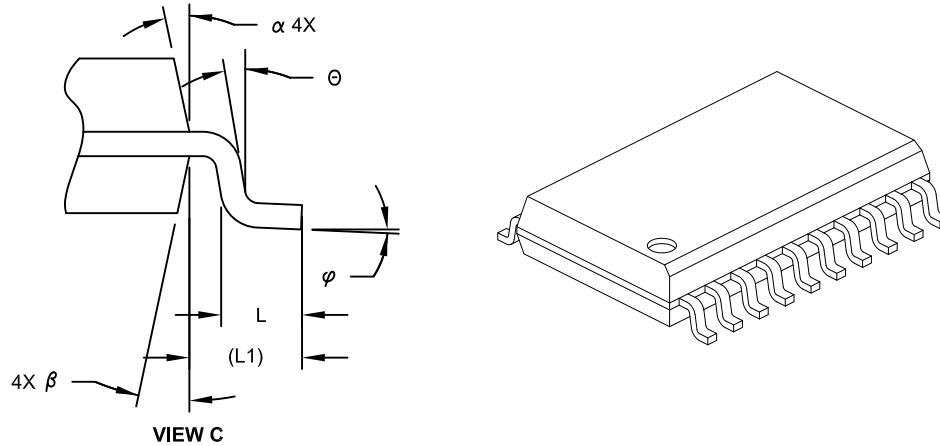
注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



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:

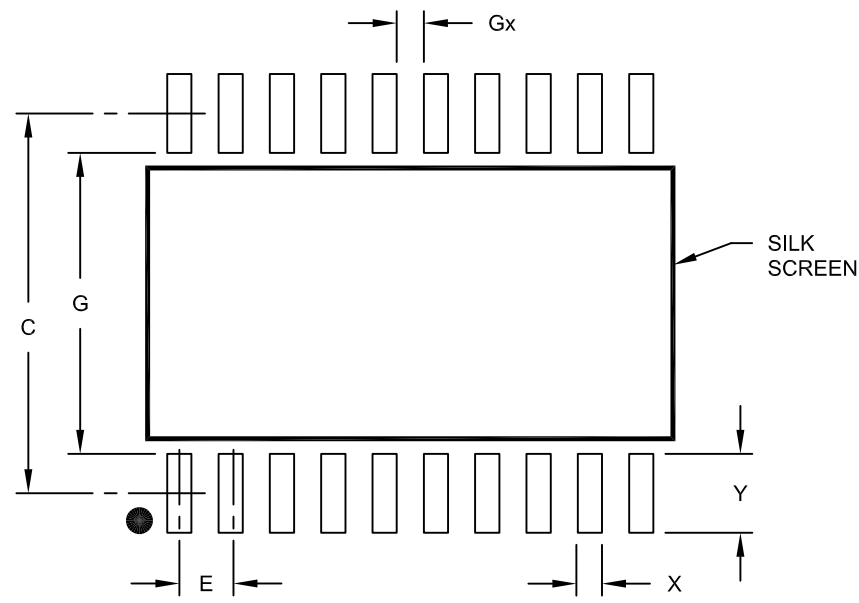
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

# PIC16(L)F1615/9

## 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:

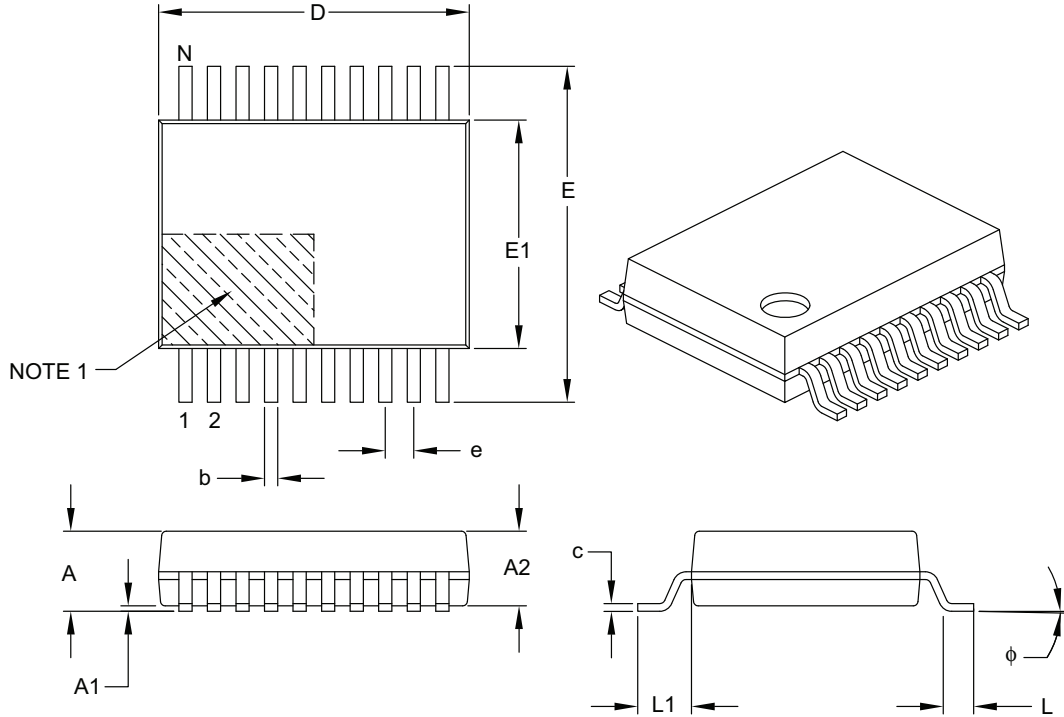
1. Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A



## 20 引脚塑封紧缩小外形封装（SS）—— 主体 5.30 mm [SSOP]

注：最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	0.65 BSC		
Overall Height	A	—	—	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	—	—
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	—	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	—	0.38

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

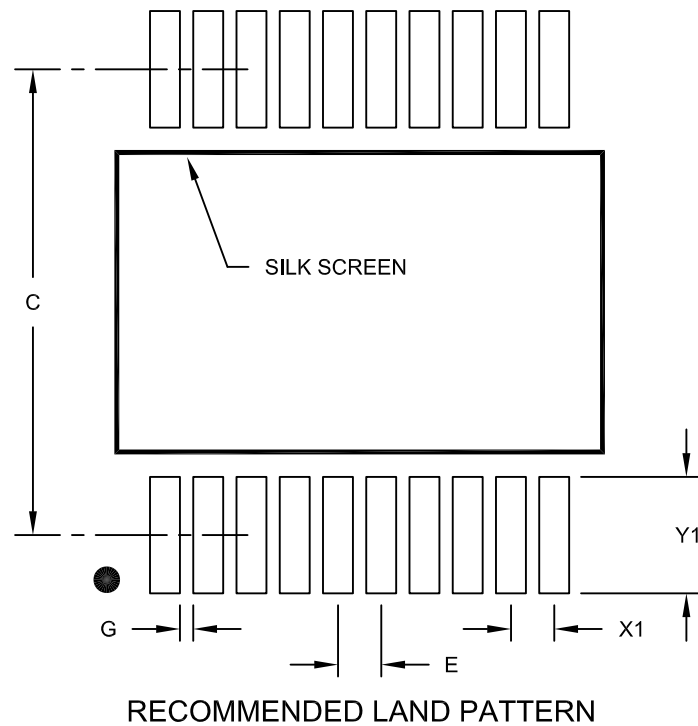
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

# PIC16(L)F1615/9

## 20 引脚塑封紧缩小外形封装（SS）—— 主体 5.30 mm [SSOP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

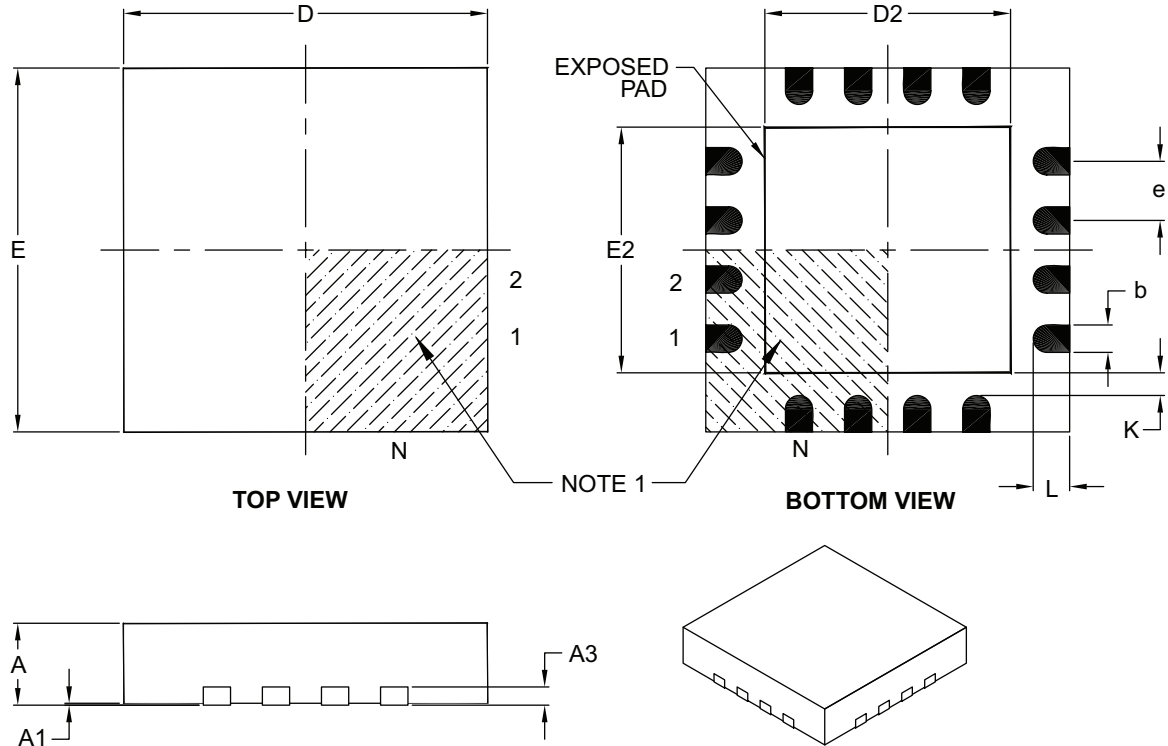
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

## 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。



Dimension	Units	MILLIMETERS		
		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.

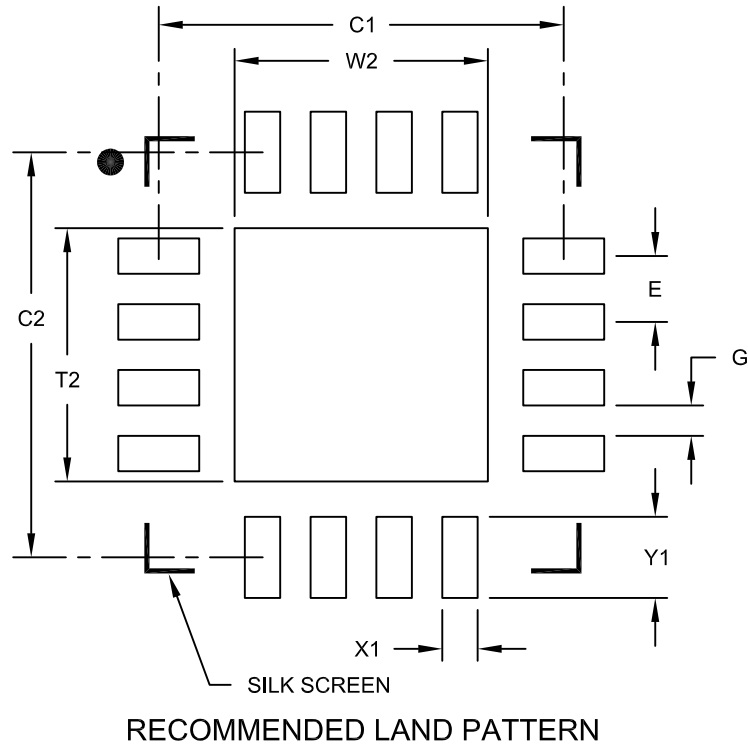
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127B

# PIC16(L)F1615/9

## 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:

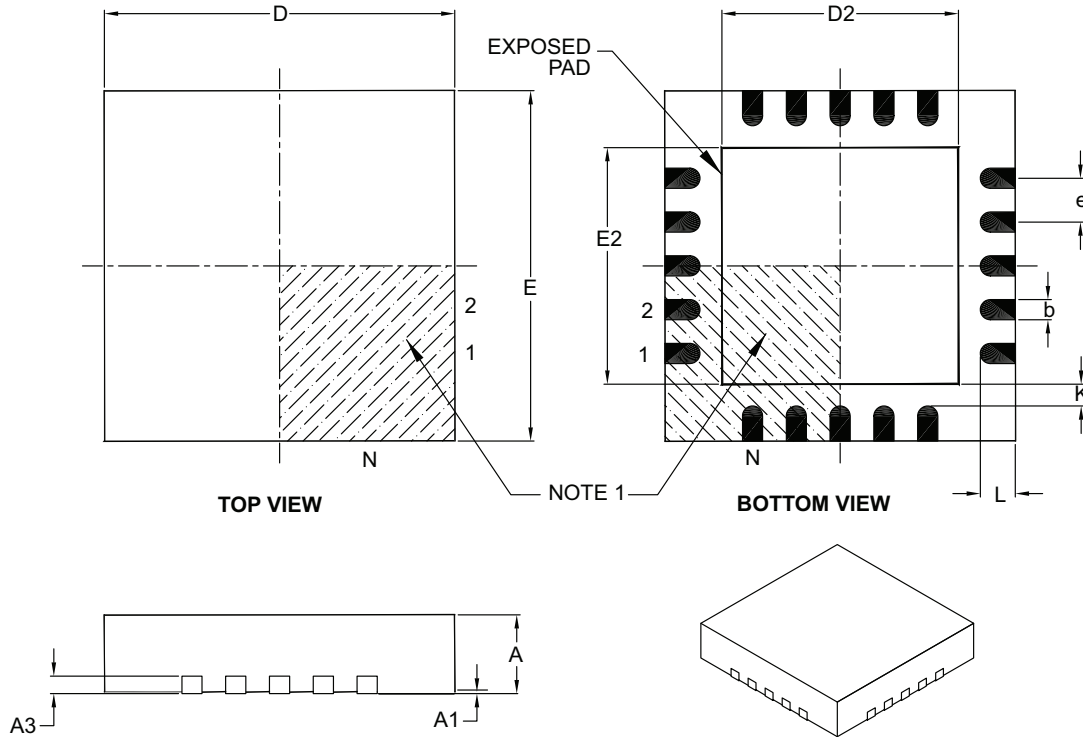
1. Dimensioning and tolerancing per ASME Y14.5M

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	20		
Pitch	e	0.50 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	2.60	2.70	2.80
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.60	2.70	2.80
Contact Width	b	0.18	0.25	0.30
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.

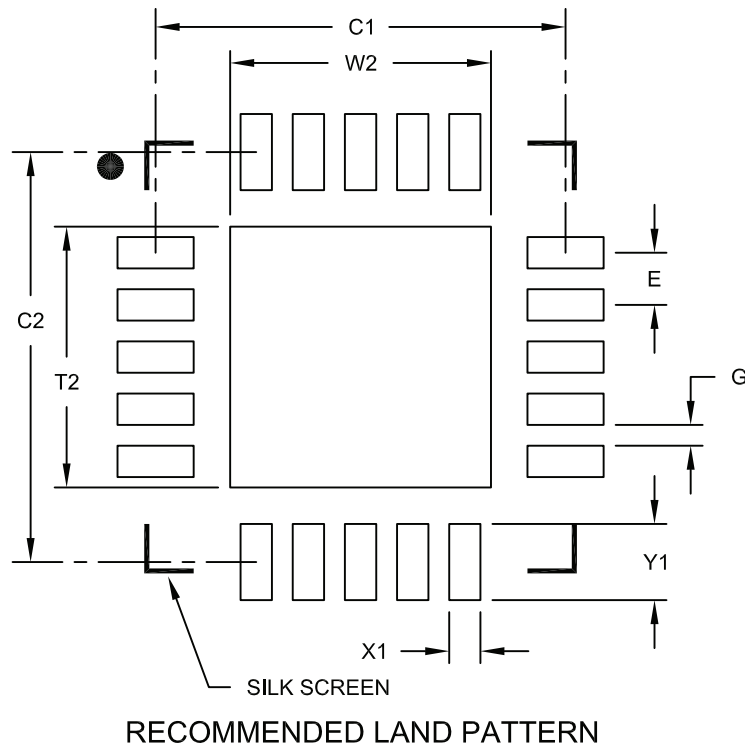
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-126B

# PIC16(L)F1615/9

20 引脚塑封正方扁平无脚封装（ML）—— 主体 4x4 mm [QFN]，触点长度 0.40 mm

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
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:

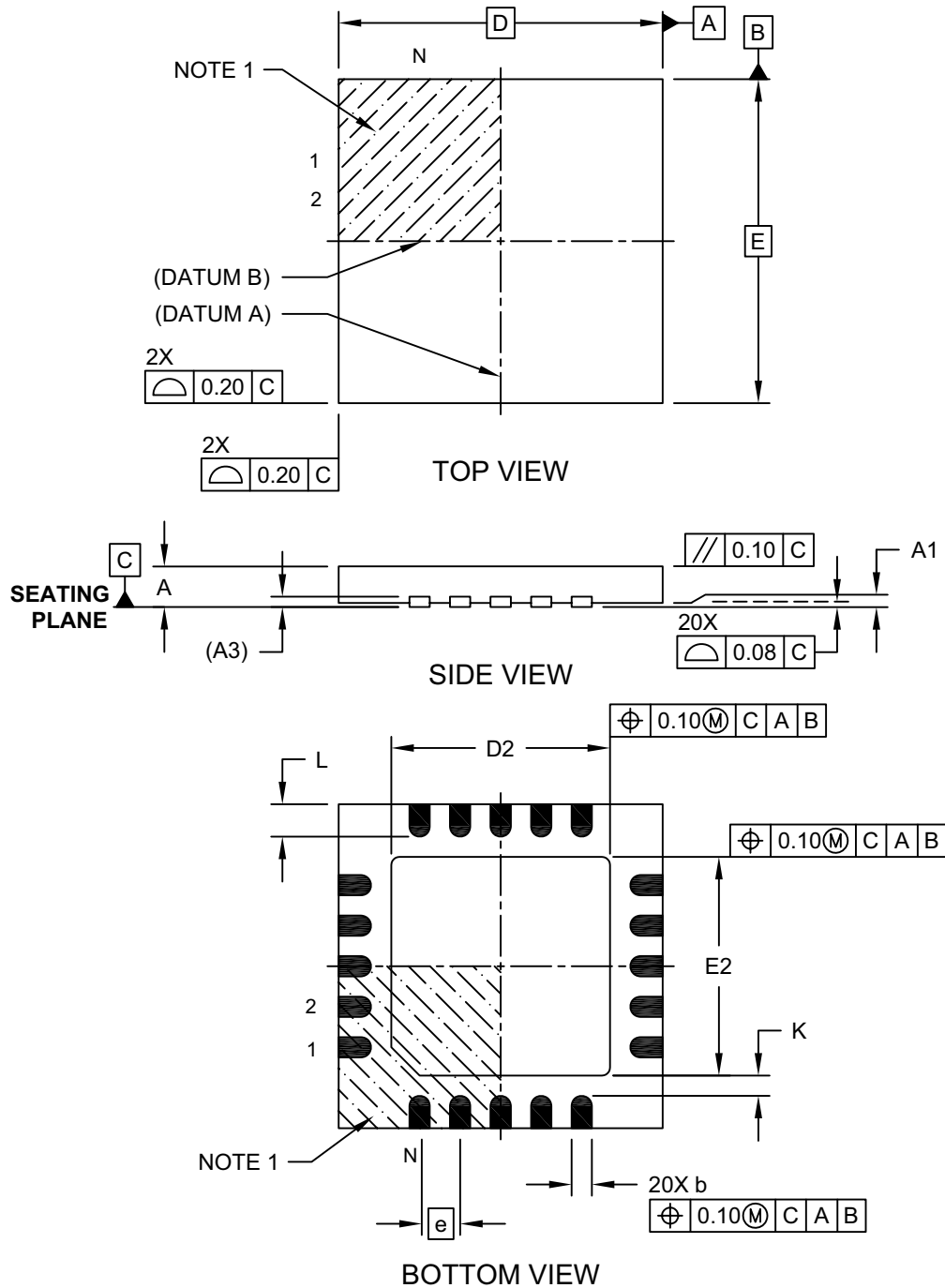
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2126A

## 20 引脚超薄塑封四方扁平无脚封装（GZ）—— 主体 4x4x0.5 mm [UQFN]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。

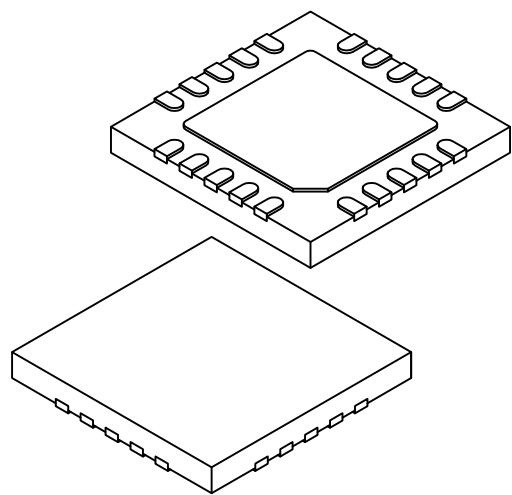


Microchip Technology Drawing C04-255A Sheet 1 of 2

# PIC16(L)F1615/9

## 20 引脚超薄塑封四方扁平无脚封装（GZ）—— 主体 4x4x0.5 mm [UQFN]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Terminals	N		20		
Pitch	e		0.50 BSC		
Overall Height	A		0.45	0.50	0.55
Standoff	A1		0.00	0.02	0.05
Terminal Thickness	A3		0.127 REF		
Overall Width	E		4.00 BSC		
Exposed Pad Width	E2		2.60	2.70	2.80
Overall Length	D		4.00 BSC		
Exposed Pad Length	D2		2.60	2.70	2.80
Terminal Width	b		0.20	0.25	0.30
Terminal Length	L		0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		0.20	-	-

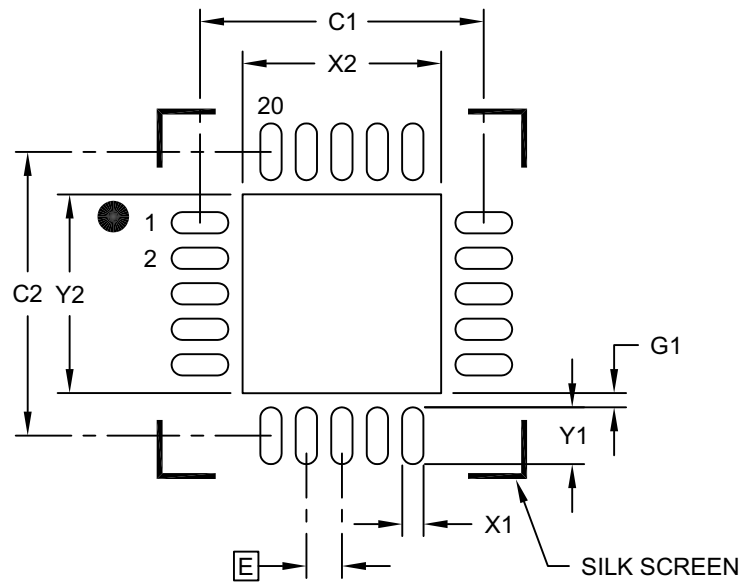
Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Package is saw singulated
- 3. Dimensioning and tolerancing per ASME Y14.5M
  - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
  - REF: Reference Dimension, usually without tolerance, for information purposes only.



## 20 引脚超薄塑封四方扁平无脚封装（GZ）—— 主体 4x4x0.5 mm [UQFN]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



### RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	X2			2.80
Optional Center Pad Length	Y2			2.80
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X20)	X1			0.30
Contact Pad Length (X20)	Y1			0.80
Contact Pad to Center Pad (X20)	G1	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2255A

# PIC16(L)F1615/9

---

## 附录A： 版本历史

### 版本A（2014年10月）

本文档的初始版本。

### 版本B（2015年4月）

增加了高电流引脚。

更新了PIC12/16(L)F161X系列类型表和封装表。

增加了适用于高驱动能力引脚的 $V_{OH}$ — $I_{OH}$ 曲线的图36-7和图36-8。

删除了图36-27和图36-28。

更新了例3-2。

更新了图5-1、图23-1、图23-2、图28-12、图31-1、图31-2、图31-4、图32-1和图35-6。

更新了寄存器23-4、寄存器27-2、寄存器31-4和寄存器31-7。

更新了第26.3节、第25.4.2节、第27.1节、第30.0节、第35.0节和第35.1节。

更新了表1-2、表1-3、表3-5、表3-16、表13-2、表23-3、表23-4、表23-7、表28-1、表35-4、表35-8和表35-17。

更新了第23节——增加了丢失模式/模式汇总表，改写了文本以使表达更加明确/更具描述性。

更正了少量打字错误。

## MICROCHIP 网站

Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息：

- **产品支持** —— 数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持** —— 常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务** —— 产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 [www.microchip.com](http://www.microchip.com)。在“支持”(Support)下，点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

# PIC16(L)F1615/9

## 产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

器件编号	[X] <sup>(1)</sup>	-	X	/XX	XXX
器件	卷带式选项		温度范围	封装	定制编号
器件:	PIC16LF1615、PIC16F1615、 PIC16LF1619和PIC16F1619				
卷带式选项:	空白 = 标准封装（料管或托盘封装） T = 卷带式 <sup>(1)</sup>				
温度范围:	I = -40°C到+85°C（工业级） E = -40°C到+125°C（扩展级）				
封装: <sup>(2)</sup>	MF = DFN（8引脚） ML = QFN（16引脚和20引脚） P = 塑封DIP SL = SOIC（14引脚） SN = SOIC（8引脚） ST = TSSOP GZ = UQFN（20引脚）				
定制编号:	QTP、SQTP、代码或特殊要求 （其他情况空白）				

示例:

a) PIC16LF1615T - I/SN  
卷带式，  
工业级温度，  
SOIC封装

b) PIC16F1619 - I/P  
工业级温度，  
PDIP封装

c) PIC16F1619 - E/ML 298  
扩展级温度，  
QFN封装，  
QTP模式#298

注 1: 卷带式标识符仅出现在产品目录的器件编号描述中。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: 关于其他可用小型封装的供货和标识信息，请访问[www.microchip.com/packaging](http://www.microchip.com/packaging)，或联系当地的销售办事处。

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 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，不得暗或以其他方式转让任何许可证。

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、FlashFlex、flexPWR、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、MediaLB、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、PICSTART、PIC<sup>32</sup> 徽标、RightTouch、SpyNIC、SST、SST 徽标、SuperFlash 及 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

The Embedded Control Solutions Company 和 mTouch 为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、ECAN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、KleerNet、KleerNet 徽标、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、RightTouch 徽标、REAL ICE、SQI、Serial Quad I/O、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2015, Microchip Technology Inc. 版权所有。

ISBN: 978-1-63277-708-9

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器及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。

## 全球销售及服务中心

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:

[http://www.microchip.com/  
support](http://www.microchip.com/support)

网址: [www.microchip.com](http://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

**德国 Germany - Karlsruhe**  
Tel: 49-721-625370

**德国 Germany - Munich**  
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

**荷兰 Netherlands - Drunen**  
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

07/14/15