

8/14/16 引脚 8 位闪存单片机

说明:

PIC12(L)F1612/16(L)F1613 单片机以 8 和 14 引脚封装提供了一些用于小型电机和通用应用的嵌入式控制设计的独特片上功能。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。CRC 和窗口 WDT 可支持家用电器和白色家电中的安全关键型应用。

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:
 - 0-32 MHz 时钟输入
 - 最小指令周期为 125 ns
- 带有自动现场保护的中断功能
- 带有可选上溢 / 下溢复位的 16 级深硬件堆栈
- 直接、间接和相对寻址模式:
 - 两个完全 16 位文件选择寄存器 (File Select Register, FSR)
 - FSR 可以读取程序和数据存储器
- 上电复位 (Power-on Reset, POR)
- 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源

存储器:

- 最大 2K 字的闪存程序存储器
- 最大 256 字节的数据 SRAM 存储器
- 直接、间接和相对寻址模式

工作特性:

- 通过两个引脚进行在线串行编程 (In-Circuit Serial Programming™, ICSP™)
- 增强型低电压编程 (Low-Voltage Programming, LVP)
- 工作电压范围:
 - 1.8V 至 3.6V (PIC12LF1612/16LF1613)
 - 2.3V 至 5.5V (PIC12F1612/16F1613)
- 可编程代码保护
- 可在软件控制下自编程

时钟结构:

- 精确的内部振荡器:
 - 出厂时精度已校准到 $\pm 1\%$, 典型值
 - 可通过软件选择频率范围: 从 31 kHz 至 32 MHz
- 31 kHz 低功耗内部振荡器
- 外部振荡器模块具有:
 - 3 种外部时钟模式, 频率最高为 32 MHz
- 4 倍频锁相环 (Phase-Locked Loop, PLL)

数字外设:

- 最多 11 个 I/O 引脚和 1 个仅用作输入的引脚:
 - 可单独编程的电平变化中断引脚
 - 可单独编程的弱上拉
 - 可单独编程的数字端口控制 (输入电平选择、漏极开路和压摆率控制)
- Timer0: 带有 8 位可编程预分频器的 8 位定时器 / 计数器
- 增强型 Timer1:
 - 带有预分频器的 16 位定时器 / 计数器
 - 外部门控输入模式
 - 定时器时钟输入 (T1CKI)
- 增强型 Timer2/4/6:
 - 带有 8 位周期寄存器的 8 位定时器 / 计数器
 - 1:1 至最高 1:16 线性后分频比
 - 1:1 至最高 128:1 预分频比
 - 异步时钟源功能
 - 外部复位 / 门控源
 - 单事件计数操作
- 两个捕捉 / 比较 / PWM 模块:
 - 16 位捕捉 / 比较
 - 10 位 PWM
- 两个信号测量定时器 (Signal Measurement Timer, SMT):
 - 24 位信号测量定时器
 - 最多 12 种不同的采集模式
 - 两个 24 位结果缓冲寄存器
 - 输入极性控制
- 16 位 CRC:
 - 可通过软件选择多项式
 - 可通过软件选择数据宽度
 - 用于存储器完整性检查的集成 CCPR 存储器扫描功能

PIC12(L)F1612/16(L)F1613

数字外设（续）：

- 互补波形发生器（Complementary Waveform Generator, CWG）：
 - 多个信号源
 - 来自任意源的真值和互补值
 - 可编程的 1 至 4 个交越
 - 可编程死区
 - 故障关断输入

模拟外设：

- 模数转换器（Analog-to-Digital Converter, ADC）：
 - 10 位分辨率
 - 最多 8 个通道
- 参考电压模块：
 - 具有 1.024V、2.048V 和 4.096V 输出的固定参考电压（Fixed Voltage Reference, FVR）
 - 带有正参考电压选择的 8 位轨到轨电阻式 DAC
- 过零检测器：
 - 检测引脚上的交流信号何时越过地电位
- 最多两个比较器：
 - 轨到轨输入
 - 功耗模式控制
 - 可通过软件控制滞后

封装：

- PIC12(L)F1612：
 - 8 引脚：PDIP、SOIC 和 DFN
- PIC16(L)F1613：
 - 14 引脚：PDIP、SOIC 和 TSSOP
 - 16 引脚：QFN（4x4x0.9）

调试功能：

- 在线调试（In-Circuit Debug, ICD）：
 - 集成：支持所有服务
 - 转接头：无需
- 仿真：
 - 转接头：支持所有器件

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)	数据 SRAM (字节)	I/O 引脚	8 位 /16 位定时器	比较器	10 位 ADC (通道)	过零检测	CCP/10 位 PWM	CWG	CLC	SMT/HLT	角度定时器	窗口看门狗定时器	CRC 与存储器扫描	数学加速器	EUSART	I ² C™/SPI	调试(1)
PIC12(L)F1612	(A)	2048	256	6	1/1	1	4	1	2/0	1	0	2/3	0	有	有	0	0	0	I/H
PIC16(L)F1613	(A)	2048	256	12	1/1	2	8	1	2/0	1	0	2/3	0	有	有	0	0	0	I/H
PIC16(L)F1614	(B)	4096	512	12	1/3	2	8	1	2/2	1	2	2/3	1	有	有	1	1	1	I/H
PIC16(L)F1615	(C)	8192	1024	12	1/3	2	8	1	2/2	1	4	2/3	1	有	有	1	1	1	I/H
PIC16(L)F1618	(B)	4096	512	18	1/3	2	12	1	2/2	1	2	2/3	1	有	有	1	1	1	I/H
PIC16(L)F1619	(C)	8192	1024	18	1/3	2	12	1	2/2	1	4	2/3	1	有	有	1	1	1	I/H

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：

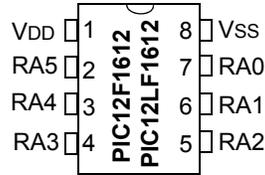
- A. DS40001737A_CN PIC12(L)F1612/16(L)F1613 数据手册， 8/14 引脚 8 位闪存单片机
- B. 未来发布 PIC16(L)F1614/8 数据手册， 14/20 引脚 8 位闪存单片机
- C. 未来发布 PIC16(L)F1615/9 数据手册， 14/20 引脚 8 位闪存单片机

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> 或当地的销售办事处。

PIC12(L)F1612/16(L)F1613

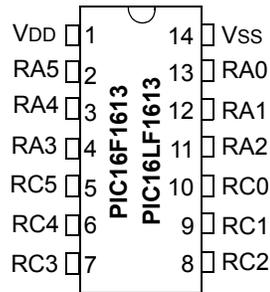
引脚图

引脚图——8 引脚 PDIP、SOIC、DFN 和 UDFN



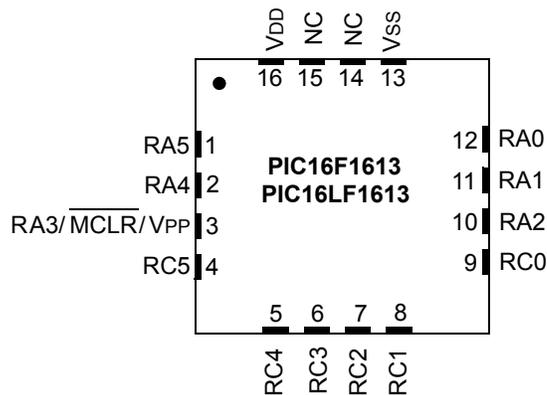
注：关于所有外设功能的位置，请参见表 2。

引脚图——14 引脚 PDIP、SOIC 和 TSSOP



注：关于所有外设功能的位置，请参见表 3。

引脚图——16 引脚 QFN



注 1：关于所有外设功能的位置，请参见表 3。

注 2：对于 QFN 封装，建议将底部焊盘连接到 VSS。

PIC12(L)F1612/16(L)F1613

引脚分配表

表 2: PIC12(L)F1612 的 8 引脚分配表

I/O	8 引脚 PDIP/SOIC/DFN	ADC	参考电压	比较器	定时器	CCP	CWG	ZCD	中断	SMT	上拉	基本功能
RA0	7	AN0	DAC1OUT1	C1IN+	—	CCP2	CWG1B	—	IOC	—	有	ICSPDAT
RA1	6	AN1	VREF+	C1IN0-	—	—	—	ZCD1OUT	IOC	—	有	ICSPCLK
RA2	5	AN2	—	C1OUT	T0CKI T4IN	CCP1 ⁽¹⁾	CWG1A ⁽¹⁾ CWG1IN	ZCD1IN	INT IOC	SMTSIG2	有	—
RA3	4	—	—	—	T1G ⁽¹⁾ T6IN	—	—	—	IOC	SMTWIN2	有	MCLR VPP
RA4	3	AN3	—	C1IN1-	T1G	—	CWG1B	—	IOC	SMTSIG1	有	CLKOUT
RA5	2	—	—	—	T1CKI T2IN	CCP1	CWG1A	—	IOC	SMTWIN1	有	CLKIN
VDD	1	—	—	—	—	—	—	—	—	—	—	VDD
Vss	8	—	—	—	—	—	—	—	—	—	—	Vss

注 1: 使用 APFCON 寄存器 (寄存器 12-1) 选择的备用引脚功能。

表 3: PIC16(L)F1613 的 14 引脚和 16 引脚分配表

I/O	14 引脚 PDIP/SOIC/TSSOP	16 引脚 QFN	ADC	参考电压	比较器	定时器	CCP	CWG	ZCD	中断	SMT	上拉	基本功能
RA0	13	12	AN0	DAC1OUT1	C1IN+	—	—	—	—	IOC	—	有	ICSPDAT
RA1	12	11	AN1	VREF+	C1IN0- C2IN0-	—	—	—	ZCD1OUT	IOC	—	有	ICSPCLK
RA2	11	10	AN2	—	C1OUT	T0CKI	—	CWG1IN	ZCD1IN	INT IOC	—	有	—
RA3	4	3	—	—	—	T1G ⁽¹⁾ T6IN	—	—	—	IOC	SMTWIN2	有	MCLR VPP
RA4	3	2	AN3	—	—	T1G	—	—	—	IOC	SMTSIG1	有	CLKOUT
RA5	2	1	—	—	—	T1CKI T2IN	CCP2 ⁽¹⁾	—	—	IOC	SMTWIN1	有	CLKIN
RC0	10	9	AN4	—	C2IN+	—	—	—	—	IOC	—	有	—
RC1	9	8	AN5	—	C1IN1- C2IN1-	T4IN	—	—	—	IOC	SMTSIG2	有	—
RC2	8	7	AN6	—	C1IN2- C2IN2-	—	—	CWG1D	—	IOC	—	有	—
RC3	7	6	AN7	—	C1IN3- C2IN3-	—	CCP2	CWG1C	—	IOC	—	有	—
RC4	6	5	—	—	C2OUT	—	—	CWG1B	—	IOC	—	有	—
RC5	5	4	—	—	—	—	CCP1	CWG1A	—	IOC	—	有	—
VDD	1	16	—	—	—	—	—	—	—	—	—	—	VDD
Vss	14	13	—	—	—	—	—	—	—	—	—	—	Vss

注 1: 使用 APFCON 寄存器 (寄存器 12-1) 选择的备用引脚功能。

目录

1.0 器件概述.....	7
2.0 增强型中档 CPU.....	13
3.0 存储器构成.....	15
4.0 器件配置.....	47
5.0 振荡器模块.....	54
6.0 复位.....	65
7.0 中断.....	73
8.0 掉电模式（休眠）.....	88
9.0 窗口看门狗定时器（WDT）.....	91
10.0 闪存程序存储器控制.....	99
11.0 循环冗余校验（CRC）模块.....	115
12.0 I/O 端口.....	127
13.0 电平变化中断.....	141
14.0 固定参考电压（FVR）.....	146
15.0 温度指示器模块.....	149
16.0 模数转换器（ADC）模块.....	151
17.0 8 位数模转换器（DAC1）模块.....	165
18.0 比较器模块.....	169
19.0 过零检测（ZCD）模块.....	177
20.0 Timer0 模块.....	181
21.0 带门控控制的 Timer1 模块.....	184
22.0 Timer2/4/6 模块.....	195
23.0 捕捉 / 比较 / PWM 模块.....	214
24.0 互补波形发生器（CWG）模块.....	228
25.0 信号测量定时器（SMTx）.....	255
26.0 在线串行编程（ICSP™）.....	299
27.0 指令集汇总.....	301
28.0 电气规范.....	315
29.0 直流和交流特性图表.....	339
30.0 开发支持.....	357
31.0 封装信息.....	361
附录 A: 版本历史.....	380
Microchip 网站.....	381
变更通知客户服务.....	381
客户支持.....	381
产品标识体系.....	382

致 客 户

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

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

最新数据手册

欲获得本数据手册的最新版本，请访问我公司网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中紧跟数字串后的字母是版本号，例如：DS300000000A_CN 是文档的 A 版本。

勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

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

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

客户通知系统

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

1.0 器件概述

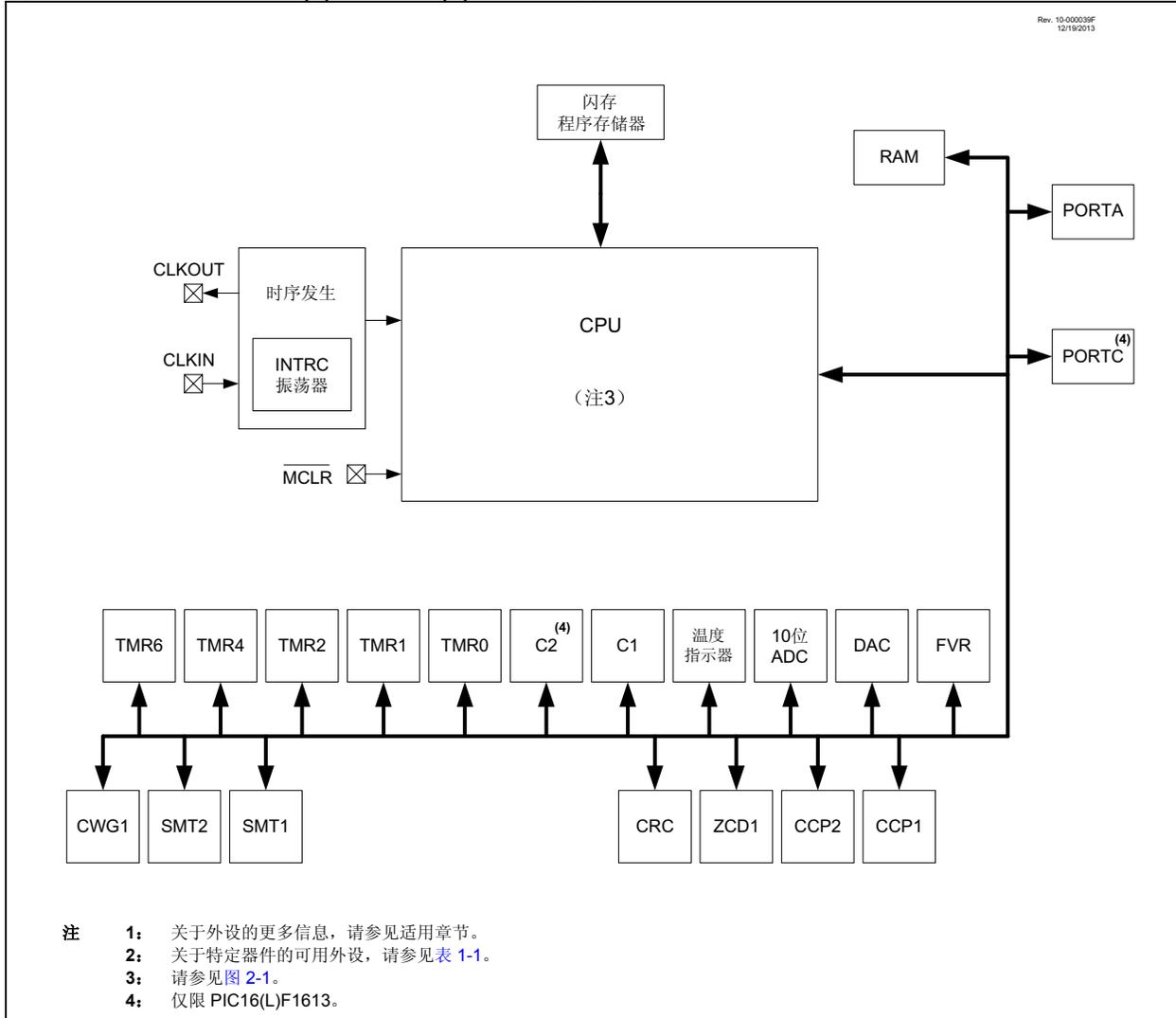
本数据手册介绍了 PIC12(L)F1612/16(L)F1613 器件。这些器件的框图如图 1-1 所示，可用的外设如表 1-1 所示，引脚说明如表 1-2 和表 1-3 所示。

表 1-1: 器件外设汇总

外设		PIC12(L)F1612	PIC16(L)F1613
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 (Digital-to-Analog Converter, DAC)		•	•
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)		•	•
捕获 / 比较 / PWM (Capture/Compare/PWM, CCP) 模块			
	CCP1	•	•
	CCP2	•	•
比较器			
	C1	•	•
	C2		•
信号测量定时器 (SMT)			
	SMT1	•	•
	SMT2	•	•
定时器			
	Timer0	•	•
	Timer1	•	•
	Timer2	•	•
	Timer4	•	•
	Timer6	•	•

PIC12(L)F1612/16(L)F1613

图 1-1: PIC12(L)F1612/16(L)F1613 框图



PIC12(L)F1612/16(L)F1613

表 1-2: PIC12(L)F1612 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/DAC1OUT1/ CCP2/CWG1B ⁽¹⁾ / ICSPDAT	RA0	TTL/ST	CMOS/OD	通用 I/O。
	AN0	AN	—	ADC 通道输入。
	C1IN+	AN	—	比较器的正输入。
	DAC1OUT1	—	AN	数模转换器输出。
	CCP2	TTL/ST	—	捕捉 / 比较 / PWM2。
	CWG1B	TTL/ST	—	CWG 互补输出 B。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RA1/AN1/VREF+/C1IN0-/ ZCD1OUT/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	—	比较器的负输入。
	ZCD1OUT	—	CMOS	过零检测输出。
	ICSPCLK	ST	—	ICSP 编程时钟。
RA2/AN2/C1OUT/T0CKI/T4IN/ CCP1 ⁽¹⁾ /CWG1A ⁽¹⁾ / CWG1IN/ZCD1IN/INT/SMTSIG2	RA2	TTL/ST	CMOS/OD	通用 I/O。
	AN2	AN	—	ADC 通道输入。
	C1OUT	—	CMOS/OD	比较器输出。
	T0CKI	TTL/ST	—	Timer0 时钟输入。
	T4IN	TTL/ST	—	Timer4 输入。
	CCP1	TTL/ST	CMOS/OD	捕捉 / 比较 / PWM1。
	CWG1A	—	—	CWG 互补输出 A。
	CWG1IN	TTL/ST	—	CWG 互补输入。
	ZCD1IN	AN	—	过零检测输入。
	INT	TTL/ST	—	外部中断。
SMTSIG2	TTL/ST	—	SMT2 信号输入。	
RA3/VPP/T1G ⁽¹⁾ /T6IN/ SMTWIN2/MCLR	RA3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	VPP	HV	—	编程电压。
	T1G	TTL/ST	—	Timer1 门控输入。
	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/C1IN1-/T1G ⁽¹⁾ / CWG1B ⁽¹⁾ /SMTSIG1/ CLKOUT	RA4	TTL/ST	CMOS/OD	通用 I/O。
	AN3	AN	—	ADC 通道输入。
	C1IN1-	AN	—	比较器的负输入。
	T1G	TTL/ST	—	Timer1 门控输入。
	CWG1B	—	CMOS/OD	CWG 互补输出 A。
	SMTSIG1	TTL/ST	—	SMT1 信号输入。
	CLKOUT	—	CMOS	Fosc/4 输出。

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

注 1: 使用 APFCON 寄存器 (寄存器 12-1) 选择的备用引脚功能。

PIC12(L)F1612/16(L)F1613

表 1-2: PIC12(L)F1612 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RA5/CLKIN/T1CKI/T2IN/ CCP1 ⁽¹⁾ /CWG1A ⁽¹⁾ / 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。
	CCP1	TTL/ST	CMOS/OD	捕捉 / 比较 / PWM1。
	CWG1A	—	CMOS/OD	CWG 互补输出 A。
	SMTWIN1	TTL/ST	—	SMT1 窗口输入。
VDD	VDD	电源	—	正电源。
VSS	VSS	电源	—	参考地。

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

注 1: 使用 APFCON 寄存器 (寄存器 12-1) 选择的备用引脚功能。

PIC12(L)F1612/16(L)F1613

表 1-3: PIC16(L)F1613 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	—	比较器的正输入。
	DAC1OUT1	—	AN	数模转换器输出。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RA1/AN1/VREF+/C1IN0-/C2IN0-/ZCD1OUT/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	—	比较器的负输入。
	C2IN0-	AN	CMOS/OD	比较器的负输入。
	ZCD1OUT	—	—	过零检测输出。
	ICSPCLK	ST	—	ICSP 编程时钟。
RA2/AN2/C1OUT/T0CKI/CWG1IN/ZCD1IN/INT	RA2	TTL/ST	CMOS/OD	通用 I/O。
	AN2	AN	—	ADC 通道输入。
	C1OUT	—	CMOS/OD	比较器输出。
	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/T1G ⁽¹⁾ /T6IN/SMTWIN2/MCLR	RA3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	VPP	HV	—	编程电压。
	T1G	TTL/ST	—	Timer1 门控输入。
	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 ⁽¹⁾ /SMTSIG1/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/T2IN/CCP2 ⁽¹⁾ /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。
	CCP2	TTL/ST	CMOS/OD	捕捉 / 比较 / PWM2。
SMTWIN1	TTL/ST	—	SMT1 窗口输入。	
RC0/AN4/C2IN+	RC0	TTL/ST	CMOS/OD	通用 I/O。
	AN4	AN	—	ADC 通道输入。
	C2IN+	AN	—	比较器的正输入。

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 I²C™ = 带 I²C™ 电平的施密特触发器输入
HV = 高电压 XTAL = 晶振

注 1: 使用 APFCON 寄存器 (寄存器 12-1) 选择的备用引脚功能。

PIC12(L)F1612/16(L)F1613

表 1-3: PIC16(L)F1613 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RC1/AN5/C1IN1-/C2IN1-/T4IN/ SMTSIG2	RC1	TTL/ST	CMOS/OD	通用 I/O。
	AN5	AN	—	ADC 通道输入。
	C1IN1-	AN	—	比较器的负输入。
	C2IN1-	AN	—	比较器的负输入。
	T4IN	TTL/ST	—	Timer4 输入。
RC2/AN6/C1IN2-/C2IN2-/ CWG1D	RC2	TTL/ST	CMOS/OD	通用 I/O。
	AN6	AN	—	ADC 通道输入。
	C1IN2-	AN	—	比较器的负输入。
	C2IN2-	AN	—	比较器的负输入。
	CWG1D	—	CMOS/OD	CWG 互补输出 D。
RC3/AN7/C1IN3-/C2IN3-/ CCP2 ⁽¹⁾ /CWG1C	RC3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	AN7	AN	—	ADC 通道输入。
	C1IN3-	AN	—	比较器的负输入。
	C2IN3-	AN	—	比较器的负输入。
	CCP2	TTL/ST	CMOS/OD	捕捉 / 比较 / PWM2。
RC4/C2OUT/CWG1B	CWG1C	—	CMOS/OD	CWG 互补输出 C。
	RC4	TTL/ST	CMOS/OD	通用 I/O。
	C2OUT	—	CMOS/OD	比较器输出。
RC5/CCP1/CWG1A	CWG1B	—	CMOS/OD	CWG 互补输出 B。
	RC5	TTL/ST	CMOS/OD	通用 I/O。
	CCP1	TTL/ST	CMOS/OD	捕捉 / 比较 / PWM1。
VDD	CWG1A	—	CMOS/OD	CWG 互补输出 A。
	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 I²C™ = 带 I²C™ 电平的施密特触发器输入
HV = 高电压 XTAL = 晶振

注 1: 使用 APFCON 寄存器 (寄存器 12-1) 选择的备用引脚功能。

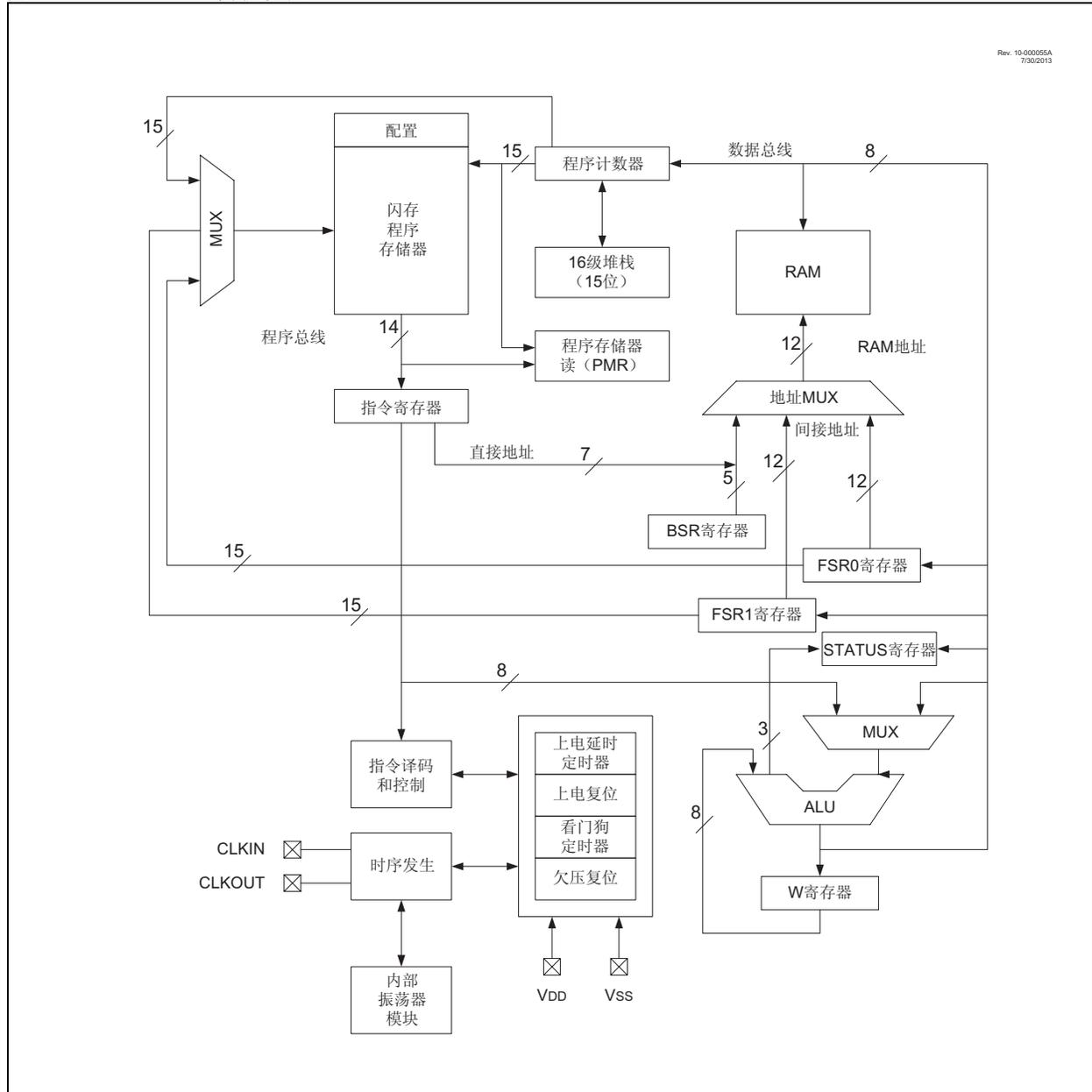
PIC12(L)F1612/16(L)F1613

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（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



PIC12(L)F1612/16(L)F1613

2.1 自动中断现场保护

在中断期间，器件会自动将一些寄存器保存到影子寄存器中，从中断返回时则会恢复这些寄存器。这可以节省堆栈空间和用户代码。更多信息，请参见[第 7.5 节“自动现场保护”](#)。

2.2 带有上溢和下溢的 16 级堆栈

这些器件具有 15 位宽、16 字深的硬件堆栈存储器。在发生堆栈上溢或下溢时，PCON 寄存器中的相应位（STKOVF 或 STKUNF）会置 1，如果使能复位，则会导致软件复位。更多详细信息，请参见[第 3.4 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 3.5 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 27.0 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

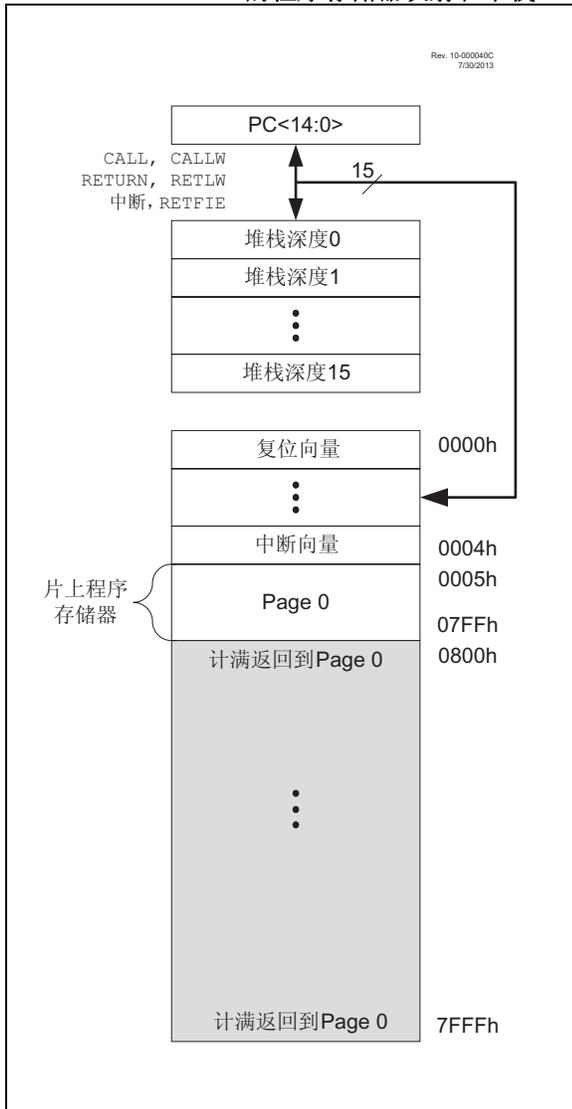
增强型中档内核具有一个 15 位程序计数器，能够寻址 32K x 14 的程序存储空间。表 3-1 列出了已实现的存储器大小。访问超出上述边界的存储单元，将返回到已实现的存储空间内。复位向量位于 0000h，而中断向量位于 0004h（见图 3-1）。

表 3-1: 器件大小和地址

器件	程序存储空间（字）	最后的程序存储器地址
PIC12(L)F1612/16(L)F1613	2,048	07FFh

PIC12(L)F1612/16(L)F1613

图 3-1: PIC12(L)F1612/16(L)F1613 的程序存储器映射和堆栈



3.1.1 将程序存储器作为数据进行读取

有两种方法来访问程序存储器中的常量。第一种方法是使用 RETLW 指令表。第二种方法是设置某个 FSR，使之指向程序存储器。

3.1.1.1 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。

3.1.1.2 通过 FSR 间接读取

通过将 FSRxH 寄存器的 bit 7 置 1，并读取匹配的 INDFx 寄存器，可以将程序存储器作为数据进行访问。MOVIW 指令会将所寻址字的低 8 位放入 W 寄存器。对程序存储器的写操作不能通过 INDF 寄存器执行。通过 FSR 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
    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;... LOTS OF CODE...
    MOVLW LOW constants
    MOVWF FSR1L
    MOVLW HIGH constants
    MOVWF FSR1H
    MOVIW 0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

PIC12(L)F1612/16(L)F1613

3.2 数据存储器的构成

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

工作存储区的选择通过向存储区选择寄存器（Bank Select Register, BSR）写入存储区编号来进行。未实现的存储器将读为 0。所有数据存储器可以直接访问（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 3.5 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。

3.2.1 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 3-2 列出了这些寄存器。详细信息，请参见表 3-9。

表 3-2: 内核寄存器

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

PIC12(L)F1612/16(L)F1613

3.2.1.1 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。

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 27.0 节“指令集汇总”。

注 1: 在减法运算中，C 和 DC 位分别作为借位位和半借位位。

寄存器 3-1: STATUS: 状态寄存器

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

图注:

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

- bit 7-5 **未实现:** 读为 0
- bit 4 **$\overline{\text{TO}}$:** 超时位
1 = 在上电或执行 CLRWDW 指令或 SLEEP 指令后
0 = 发生了 WDT 超时
- bit 3 **$\overline{\text{PD}}$:** 掉电位
1 = 在上电或执行 CLRWDW 指令后
0 = 执行 SLEEP 指令
- bit 2 **Z:** 全零标志位
1 = 算术运算或逻辑运算的结果为零
0 = 算术运算或逻辑运算的结果不为零
- bit 1 **DC:** 半进位 / 半借位位 (ADDWF、ADDLW、SUBLW 和 SUBWF 指令) ⁽¹⁾
1 = 结果的第 4 个低位发生了进位
0 = 结果的第 4 个低位未发生进位
- bit 0 **C:** 进位 / 借位位 (ADDWF、ADDLW、SUBLW 和 SUBWF 指令) ⁽¹⁾
1 = 结果的最高有效位发生了进位
0 = 结果的最高有效位未发生进位

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

PIC12(L)F1612/16(L)F1613

3.2.2 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。

3.2.3 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 x20h/xA0h 至 x6Fh/xEFh）。

3.2.3.1 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 3.5.2 节“线性数据存储器”。

3.2.4 公共 RAM

有 16 字节的公共 RAM 可以从所有存储区中进行访问。

3.2.5 器件存储器映射

表 3-5 至表 3-8 给出了 PIC12(L)F1612/16(L)F1613 的存储器映射。

图 3-2: 存储区分区

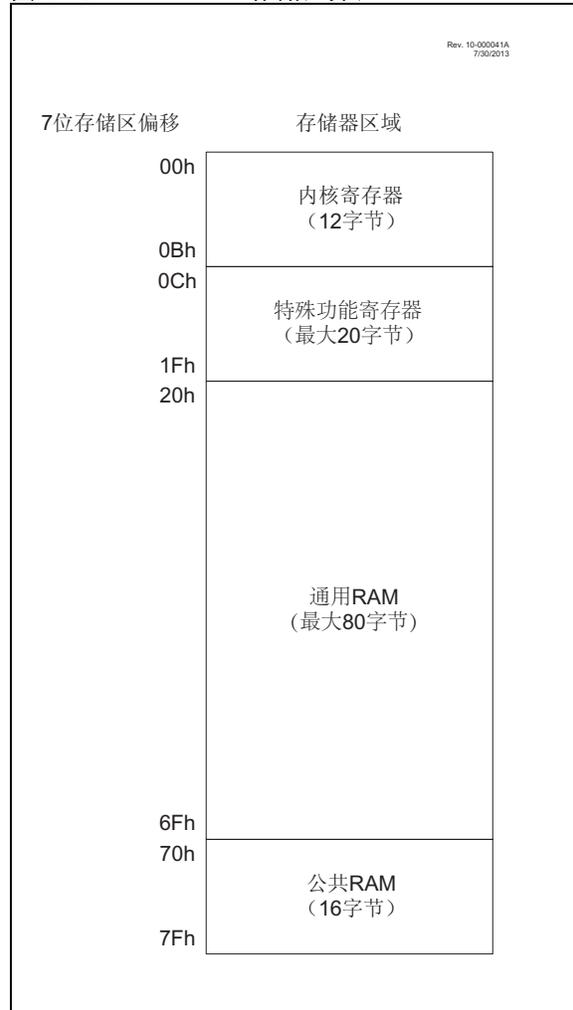


表 3-3: PIC12(L)F1612 存储器映射, BANK 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	内核寄存器 (表 3-2)	080h	内核寄存器 (表 3-2)	100h	内核寄存器 (表 3-2)	180h	内核寄存器 (表 3-2)	200h	内核寄存器 (表 3-2)	280h	内核寄存器 (表 3-2)	300h	内核寄存器 (表 3-2)	380h	内核寄存器 (表 3-2)
00Bh	—	08Bh	—	10Bh	—	18Bh	—	20Bh	—	28Bh	—	30Bh	—	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	—	08Eh	—	10Eh	—	18Eh	—	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	—	291h	CCP1RL	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	CCP1RH	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	—	193h	PMDATL	213h	—	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR4	094h	PIE4	114h	—	194h	PMDATH	214h	—	294h	CCP1CAP	314h	—	394h	—
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	—	295h	—	315h	—	395h	—
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	—	296h	—	316h	—	396h	—
017h	TMR1H	097h	—	117h	FVRCON	197h	VREGCON	217h	—	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	—	218h	—	298h	CCP2RL	318h	—	398h	—
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	—	219h	—	299h	CCP2RH	319h	—	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	—	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	—	21Bh	—	29Bh	CCP2CAP	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	—	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh	APFCON	19Dh	—	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	—	21Eh	—	29Eh	CCPTMRS	31Eh	—	39Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	—	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h	通用寄存器 80 字节	0A0h	通用寄存器 80 字节	120h	通用寄存器 80 字节	1A0h	未实现 读为 0	220h	未实现 读为 0	2A0h	未实现 读为 0	320h	未实现 读为 0	3A0h	未实现 读为 0
06Fh	公共 RAM	0EFh	公共 RAM (快速操作存储区 70h – 7Fh)	16Fh	公共 RAM (快速操作存储区 70h – 7Fh)	1EFh	公共 RAM (快速操作存储区 70h – 7Fh)	26Fh	公共 RAM (快速操作存储区 70h – 7Fh)	2EFh	公共 RAM (快速操作存储区 70h – 7Fh)	36Fh	公共 RAM (快速操作存储区 70h – 7Fh)	3EFh	公共 RAM (快速操作存储区 70h – 7Fh)
070h	公共 RAM	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	公共 RAM	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)F1613 存储器映射, BANK 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	内核寄存器 (表 3-2)	080h	内核寄存器 (表 3-2)	100h	内核寄存器 (表 3-2)	180h	内核寄存器 (表 3-2)	200h	内核寄存器 (表 3-2)	280h	内核寄存器 (表 3-2)	300h	内核寄存器 (表 3-2)	380h	内核寄存器 (表 3-2)
00Bh	—	08Bh	—	10Bh	—	18Bh	—	20Bh	—	28Bh	—	30Bh	—	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	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h	—	291h	CCPR1L	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h	—	292h	CCPR1H	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h	—	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR4	094h	PIE4	114h	CM2CON1	194h	PMDATH	214h	—	294h	CCP1CAP	314h	—	394h	—
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	—	295h	—	315h	—	395h	—
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	—	296h	—	316h	—	396h	—
017h	TMR1H	097h	—	117h	FVRCON	197h	VREGCON	217h	—	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	—	218h	—	298h	CCPR2L	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	—	219h	—	299h	CCPR2H	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	—	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	—	21Bh	—	29Bh	CCP2CAP	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	—	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh	APFCON	19Dh	—	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	—	21Eh	—	29Eh	CCPTMRS	31Eh	—	39Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	—	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h	通用寄存器 80 字节	0A0h	通用寄存器 80 字节	120h	通用寄存器 80 字节	1A0h	未实现 读为 0	220h	未实现 读为 0	2A0h	未实现 读为 0	320h	未实现 读为 0	3A0h	未实现 读为 0
06Fh	—	0EFh	—	16Fh	—	1EFh	—	26Fh	—	2EFh	—	36Fh	—	3EFh	—
070h	公共 RAM	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 3-5: PIC12(L)F1612/16(L)F1613 存储器映射, BANK 8-23

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15			
400h	内核寄存器 (表 3-2)	480h	内核寄存器 (表 3-2)	500h	内核寄存器 (表 3-2)	580h	内核寄存器 (表 3-2)	600h	内核寄存器 (表 3-2)	680h	内核寄存器 (表 3-2)	700h	内核寄存器 (表 3-2)	780h	内核寄存器 (表 3-2)		
40Bh	—	48Bh	—	50Bh	—	58Bh	—	60Bh	—	68Bh	—	70Bh	—	78Bh	—		
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—		
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—		
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—		
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—		
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—		
411h	—	491h	—	511h	—	591h	—	611h	—	691h	CWG1DBR	711h	WDTCON0	791h	CRCDATL		
412h	—	492h	—	512h	—	592h	—	612h	—	692h	CWG1DBF	712h	WDTCON1	792h	CRCDATH		
413h	TMR4	493h	—	513h	—	593h	—	613h	—	693h	CWG1AS0	713h	WDTPSL	793h	CRCACCL		
414h	PR4	494h	—	514h	—	594h	—	614h	—	694h	CWG1AS1	714h	WDTPSH	794h	CRCACCH		
415h	T4CON	495h	—	515h	—	595h	—	615h	—	695h	CWG1OCON0	715h	WDTTMR	795h	CRCSHIFTL		
416h	T4HLT	496h	—	516h	—	596h	—	616h	—	696h	CWG1CON0	716h	—	796h	CRCSHIFTH		
417h	T4CLKCON	497h	—	517h	—	597h	—	617h	—	697h	CWG1CON1	717h	—	797h	CRCXORL		
418h	T4RST	498h	—	518h	—	598h	—	618h	—	698h	CWG1OCON1	718h	SCANLADRL	798h	CRCXORH		
419h	—	499h	—	519h	—	599h	—	619h	—	699h	CWG1CLKCON	719h	SCANLADRH	799h	CRCCON0		
41Ah	TMR6	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	CWG1ISM	71Ah	SCANHADRL	79Ah	CRCCON1		
41Bh	PR6	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	SCANHADRH	79Bh	—		
41Ch	T6CON	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	SCANCON0	79Ch	—		
41Dh	T6HLT	49Dh	—	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	未实现 读为 0	4A0h	未实现 读为 0	520h	未实现 读为 0	5A0h	未实现 读为 0	620h	未实现 读为 0	6A0h	未实现 读为 0	720h	未实现 读为 0	7A0h	未实现 读为 0		
46Fh	快速操作存储区 70h – 7Fh	4EFh	快速操作存储区 70h – 7Fh	56Fh	快速操作存储区 70h – 7Fh	5EFh	快速操作存储区 70h – 7Fh	66Fh	快速操作存储区 70h – 7Fh	66Fh	快速操作存储区 70h – 7Fh	76Fh	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	快速操作存储区 70h – 7Fh	4F0h	快速操作存储区 70h – 7Fh	570h	快速操作存储区 70h – 7Fh	5F0h	快速操作存储区 70h – 7Fh	670h	快速操作存储区 70h – 7Fh	670h	快速操作存储区 70h – 7Fh	770h	快速操作存储区 70h – 7Fh	770h	快速操作存储区 70h – 7Fh	7F0h	快速操作存储区 70h – 7Fh
47Fh	快速操作存储区 70h – 7Fh	4FFh	快速操作存储区 70h – 7Fh	57Fh	快速操作存储区 70h – 7Fh	5FFh	快速操作存储区 70h – 7Fh	67Fh	快速操作存储区 70h – 7Fh	67Fh	快速操作存储区 70h – 7Fh	77Fh	快速操作存储区 70h – 7Fh	77Fh	快速操作存储区 70h – 7Fh	7FFh	快速操作存储区 70h – 7Fh
BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23			
800h	内核寄存器 (表 3-2)	880h	内核寄存器 (表 3-2)	900h	内核寄存器 (表 3-2)	980h	内核寄存器 (表 3-2)	A00h	内核寄存器 (表 3-2)	A80h	内核寄存器 (表 3-2)	B00h	内核寄存器 (表 3-2)	B80h	内核寄存器 (表 3-2)		
80Bh	—	88Bh	—	90Bh	—	98Bh	—	A0Bh	—	A8Bh	—	B0Bh	—	B8Bh	—		
80Ch	未实现 读为 0	88Ch	未实现 读为 0	90Ch	未实现 读为 0	98Ch	未实现 读为 0	A0Ch	未实现 读为 0	A8Ch	未实现 读为 0	B0Ch	未实现 读为 0	B8Ch	未实现 读为 0		
86Fh	快速操作存储区 70h – 7Fh	8EFh	快速操作存储区 70h – 7Fh	96Fh	快速操作存储区 70h – 7Fh	9EFh	快速操作存储区 70h – 7Fh	A6Fh	快速操作存储区 70h – 7Fh	A6Fh	快速操作存储区 70h – 7Fh	B6Fh	快速操作存储区 70h – 7Fh	BEFh	快速操作存储区 70h – 7Fh		
870h	快速操作存储区 70h – 7Fh	8F0h	快速操作存储区 70h – 7Fh	970h	快速操作存储区 70h – 7Fh	9F0h	快速操作存储区 70h – 7Fh	A70h	快速操作存储区 70h – 7Fh	A70h	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	快速操作存储区 70h – 7Fh	8FFh	快速操作存储区 70h – 7Fh	97Fh	快速操作存储区 70h – 7Fh	9FFh	快速操作存储区 70h – 7Fh	A7Fh	快速操作存储区 70h – 7Fh	A7Fh	快速操作存储区 70h – 7Fh	B7Fh	快速操作存储区 70h – 7Fh	B7Fh	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 3-6: PIC12(L)F1612/16(L)F1613 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 3-2)	C80h	内核寄存器 (表 3-2)	D00h	内核寄存器 (表 3-2)	D80h	内核寄存器 (表 3-2)	E00h	内核寄存器 (表 3-2)	E80h	内核寄存器 (表 3-2)	F00h	内核寄存器 (表 3-2)	F80h	内核寄存器 (表 3-2)
C0Bh	—	C8Bh	—	D0Bh	—	D8Bh	请参见表 3-7 了解 寄存器映射详细 信息	E0Bh	—	E8Bh	—	F0Bh	—	F8Bh	请参见表 3-8 了解 寄存器映射详细 信息
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		E18h	—	E98h	—	F18h	—	F98h	
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	未实现 读为 0	EA0h	未实现 读为 0	F20h	未实现 读为 0	FA0h		
C6Fh	快速操作存储区 70h – 7Fh	CEFh	快速操作存储区 70h – 7Fh	D6Fh	快速操作存储区 70h – 7Fh	DEFh	快速操作存储区 70h – 7Fh	E6Fh	快速操作存储区 70h – 7Fh	EEFh	快速操作存储区 70h – 7Fh	F6Fh	快速操作存储区 70h – 7Fh	FEFh	快速操作存储区 70h – 7Fh
C70h		CF0h		D70h		DF0h		E70h		EF0h		F70h		FF0h	
CFFh	—	CFFh	—	D7Fh	—	DFFh	—	E7Fh	—	EFFh	—	F7Fh	—	FFFh	—

图注: = 未实现的数据存储单元, 读为 0。

PIC12(L)F1612/16(L)F1613

表 3-7: PIC12(L)F1612/16(L)F1613
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: PIC12(L)F1612/16(L)F1613
存储器映射, BANK 31

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

图注:  = 未实现的数据存储单元, 读为 0。

PIC12(L)F1612/16(L)F1613

3.2.6 内核功能寄存器汇总

表 3-9 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 3-9: 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	IOCFIE	TMR0IF	INTF	IOCFIF	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 3-10: 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	—	未实现									—	—
00Eh	PORTC ⁽⁴⁾	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx xxxx	--xx xxxx	
00Fh	—	未实现									—	—
010h	—	未实现									—	—
011h	PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	00-- -000	00-- -000	
012h	PIR2	—	C2IF ⁽⁴⁾	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	-00- -000	-00- -000	
013h	PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	--00 ----	--00 ----	
014h	PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	0000 0000	0000 0000	
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<3:0>			000- 0000	000- 0000		
01Eh	T2CLKCON	—	—	—	—	T2CS<2:0>			---- -000	---- -000		
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 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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	— ⁽²⁾	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111	
08Dh	—	未实现									—	—
08Eh	TRISC ⁽⁴⁾	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111	
08Fh	—	未实现									—	—
090h	—	未实现									—	—
091h	PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	00-- -000	00-- -000	
092h	PIE2	—	C2IE ⁽⁴⁾	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	-00- -000	-00- -000	
093h	PIE3	—	—	CWGIE	ZCDIE	—	—	—	—	--00 ----	--00 ----	
094h	PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	0000 0000	0000 0000	
095h	OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111	1111 1111	
096h	PCON	STKOVF	STKUNF	WDTWV	RWD \overline{T}	R \overline{M} CLR	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}	—	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	-0-0 0000	-q-p qqpp	
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{D} ONE	ADON		-000 0000	-000 0000	
09Eh	ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>			0000 --00	0000 --00
09Fh	ADCON2	TRIGSEL<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 PIC12F1612/16F1613。

注 2: 未实现, 读为 1。

注 3: 仅限 PIC12(L)F1612。

注 4: 仅限 PIC16(L)F1613。

表 3-10: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
Bank 2												
10Ch	LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx	--uu -uuu	
10Dh	—	未实现									—	—
10Eh	LATC ⁽⁴⁾	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	--xx xxxx	--uu uuuu	
10Fh	—	未实现									—	—
110h	—	未实现									—	—
111h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	0000 -100	0000 -100	
112h	CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			0000 -000	0000 -000	
113h	CM2CON0 ⁽⁴⁾	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	0000 -100	0000 -100	
114h	CM2CON1 ⁽⁴⁾	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	ZCD1OE	ZCD1OUT	ZCD1POL	—	—	ZCD1INTP	ZCD1INTN	0000 --00	0000 --00	
11Dh	APFCON	—	CWGASEL ⁽³⁾	CWGBSEL ⁽³⁾	—	T1GSEL	—	CCP2SEL ⁽⁴⁾	CCP1SEL ⁽³⁾	-00- 0-00	-00- 0-00	
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 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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	—	未实现											
18Eh	ANSELC ⁽⁴⁾	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	---- 1111	---- 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	— ⁽²⁾	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	— ⁽²⁾	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 ⁽¹⁾	—	—	—	—	—	—	VREGPM	保留	---- --01	---- --01		
198h 至 19Fh	—	未实现											
Bank 4													
20Ch	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--11 1111	--11 1111		
20Dh	—	未实现											
20Eh	WPUC ⁽⁴⁾	—	—	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	--11 1111	--11 1111		
20Fh 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 PIC12F1612/16F1613。
 注 2: 未实现, 读为 1。
 注 3: 仅限 PIC12(L)F1612。
 注 4: 仅限 PIC16(L)F1613。

表 3-10: 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	—	未实现									—	—
28Eh	ODCONC ⁽⁴⁾	—	—	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	--00 0000	--00 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	OE	OUT	FMT	MODE<3:0>				0000 0000	0000 0000	
294h	CCP1CAP	—	—	—	—	—	—	CTS<1:0>		---- --00	---- --00	
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	OE	OUT	FMT	MODE<3:0>				0000 0000	0000 0000	
29Bh	CCP2CAP	—	—	—	—	—	—	CTS<1:0>		---- --00	---- --00	
29Ch	—	未实现									—	—
29Dh	—	未实现									—	—
29Eh	CCPTMRS	—	—	—	—	C2TSEL<1:0>		C1TSEL<1:0>		---- 0000	---- 0000	
29Fh	—	未实现									—	—
Bank 6												
30Ch	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	--00 -000	--00 -000	
30Dh	—	未实现									—	—
30Eh	SLRCONC ⁽⁴⁾	—	—	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	--00 0000	--00 0000	
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 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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	—	未实现									—	—
38Eh	INLVLC ⁽⁴⁾	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	--11 1111	--11 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	—	未实现									—	—
395h	—	未实现									—	—
396h	—	未实现									—	—
397h	IOCCP ⁽⁴⁾	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	--00 0000	--00 0000	
398h	IOCCN ⁽⁴⁾	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	--00 0000	--00 0000	
399h	IOCCF ⁽⁴⁾	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	--00 0000	--00 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 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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 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<3:0>			000- 0000		000- 0000	
417h	T4CLKCON	—	—	—	—	—	T4CS<2:0>			---- -000	---- -000	
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<3:0>			000- 0000		000- 0000	
41Eh	T6CLKCON	—	—	—	—	—	T6CS<2:0>			---- -000	---- -000	
41Fh	T6RST	—	—	—	—	RSEL<3:0>			---- 0000		---- 0000	
Bank 9												
48Ch 至 49Fh	—	未实现								—	—	
Bank 10												
50Ch 至 51Fh	—	未实现								—	—	
Bank 11												
58Ch 至 59Fh	—	未实现								—	—	

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

- 注
- 1: 仅限 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 12											
60Ch 至 61Fh	—	未实现								—	—
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>		—	—	0000 00--	0000 00--
694h	CWG1AS1	—	TMR6AS	TMR4AS	TMR2AS	—	C2AS ⁽⁴⁾	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	CWG1OCON1	—	—	—	—	OED	OEC	OEB	OEA	---- 0000	---- 0000
699h	CWG1CLKCON	—	—	—	—	—	—	—	CS	---- ---0	---- ---0
69Ah	CWG1ISM	—	—	—	—	—	IS<2:0>		—	---- -000	---- -000
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 PIC12F1612/16F1613。

注 2: 未实现, 读为 1。

注 3: 仅限 PIC12(L)F1612。

注 4: 仅限 PIC16(L)F1613。

表 3-10: 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<1:0>		---- --00	---- --00	
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 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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	DATA<7:0>								xxxx xxxx	xxxx xxxx
792h	CRCDATH	DATA<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	X<7:1>						—	—	xxxx xxx-	xxxx xxx-
798h	CRCXORH	X<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								—	—
Bank 16-26											
x0Ch/ x8Ch — x1Fh/ x9Fh	—	未实现								—	—

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

- 注
- 1: 仅限 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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	SMTxPS<1:0>		0-00 0000	0-00 0000
D99h	SMT1CON1	SMTxGO	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<3:0>			---- 0000	---- 0000	
D9Dh	SMT1WIN	—	—	—	—	—	WSEL<2:0>		---- -000	---- -000	
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 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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	SMTxCPW<15:8>								xxxx xxxx	xxxx xxxx
DA6h	SMT2CPWU	SMTxCPW<23:16>								xxxx xxxx	xxxx xxxx
DA7h	SMT2PRL	SMTxPR<7:0>								xxxx xxxx	xxxx xxxx
DA8h	SMT2PRH	SMTxPR<15:8>								xxxx xxxx	xxxx xxxx
DA9h	SMT2PRU	SMTxPR<23:16>								xxxx xxxx	xxxx xxxx
DAAh	SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMTxPS<1:0>		0-00 0000	0-00 0000
DABh	SMT2CON1	SMTxGO	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<3:0>			---- 0000	---- 0000	
DAFh	SMT2WIN	—	—	—	—	—	WSEL<2:0>		---- -000	---- -000	
Bank 28-30											
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 PIC12F1612/16F1613。
 - 2: 未实现, 读为 1。
 - 3: 仅限 PIC12(L)F1612。
 - 4: 仅限 PIC16(L)F1613。

表 3-10: 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	
FEEh	TOSL	栈顶低字节								xxxx xxxx	uuuu uuuu	
FEFh	TOSH	—	栈顶高字节								-xxx xxxx	-uuu uuuu

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

注 1: 仅限 PIC12F1612/16F1613。

注 2: 未实现, 读为 1。

注 3: 仅限 PIC12(L)F1612。

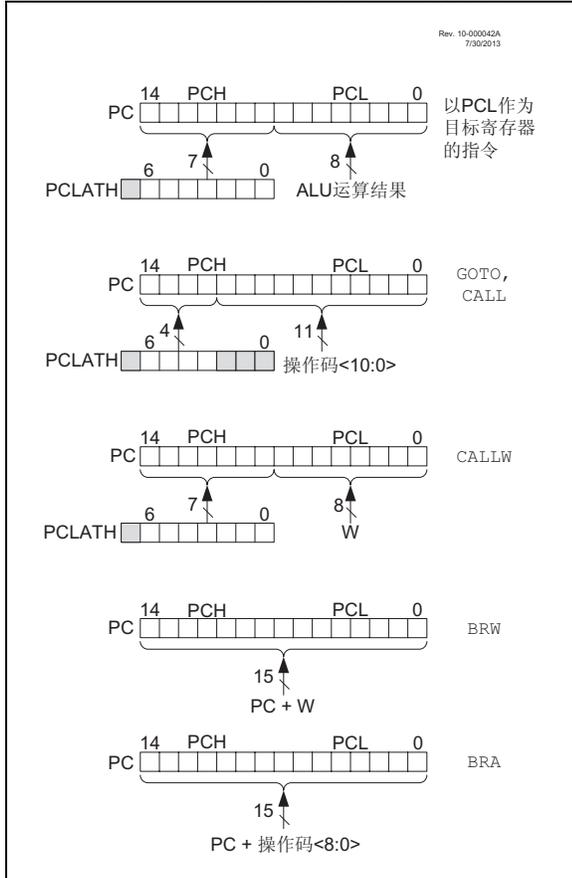
注 4: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

3.3 PCL 和 PCLATH

程序计数器 (PC) 为 15 位宽。其低字节来自可读写 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



3.3.1 修改 PCL

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

3.3.2 计算 GOTO

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

3.3.3 计算函数调用

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

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

CALLW 指令通过将 PCLATH 和 W 组合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。

3.3.4 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。

PIC12(L)F1612/16(L)F1613

3.4 堆栈

所有器件都具有 16 级 x 15 位宽的硬件堆栈（见图 3-4 至图 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。

3.4.1 访问堆栈

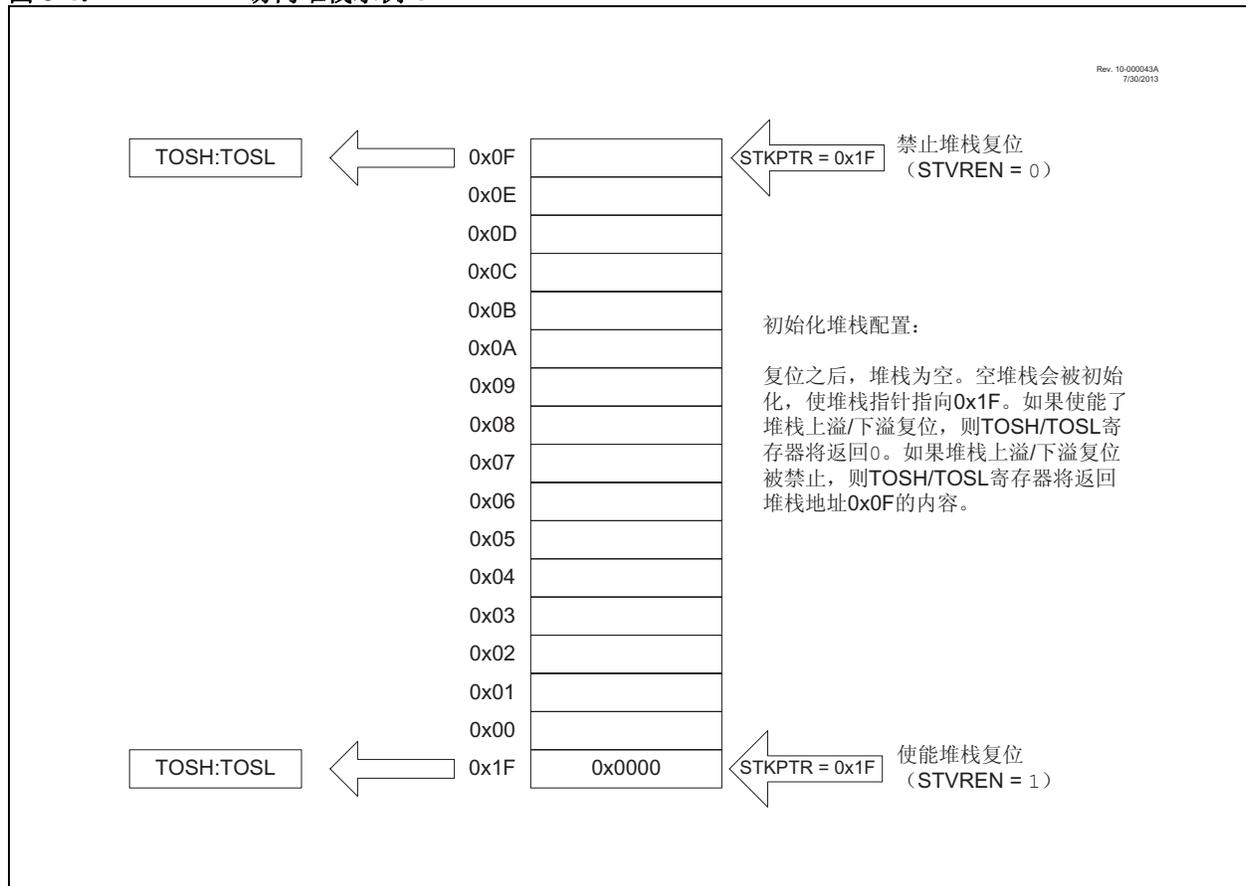
通过 TOSH、TOSL 和 STKPTR 寄存器可以使用堆栈。STKPTR 是堆栈指针的当前值。TOSH:TOSL 寄存器对指向栈顶。两个寄存器都是可读写的。由于 PC 的大小为 15 位，所以 TOS 拆分为 TOSH 和 TOSL。要访问堆栈，可以调整 STKPTR 的值（它会决定 TOSH:TOSL 位置），然后读 / 写 TOSH:TOSL。STKPTR 的宽度为 5 位，以允许检测上溢和下溢。

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



PIC12(L)F1612/16(L)F1613

图 3-5: 访问堆栈示例 2

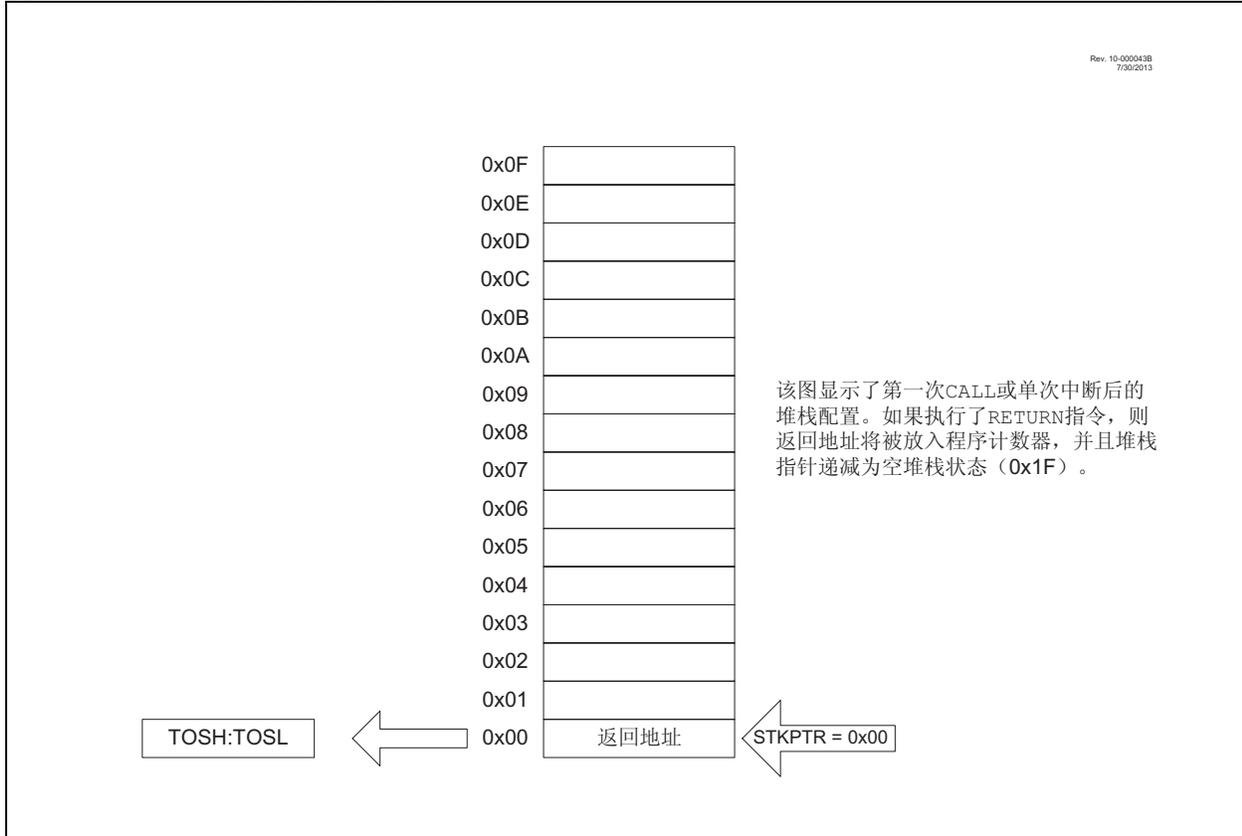


图 3-6: 访问堆栈示例 3

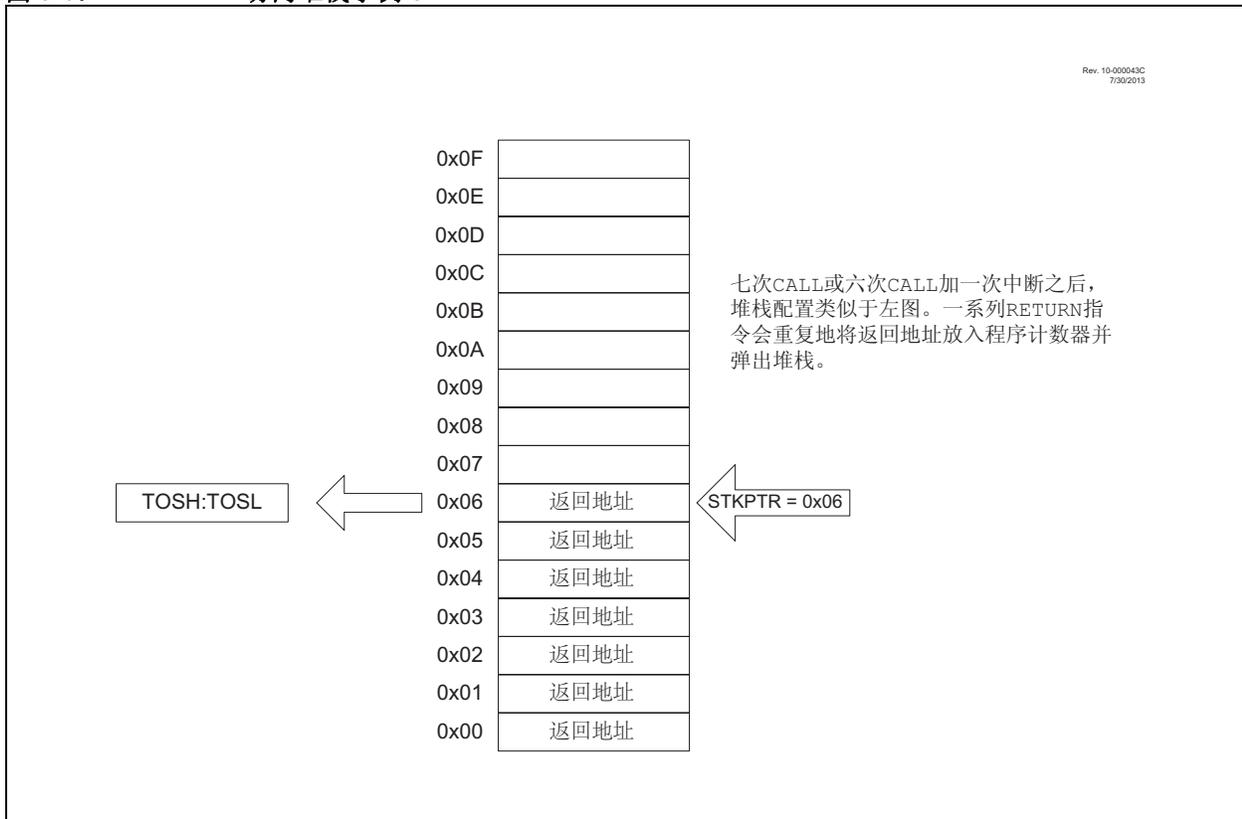
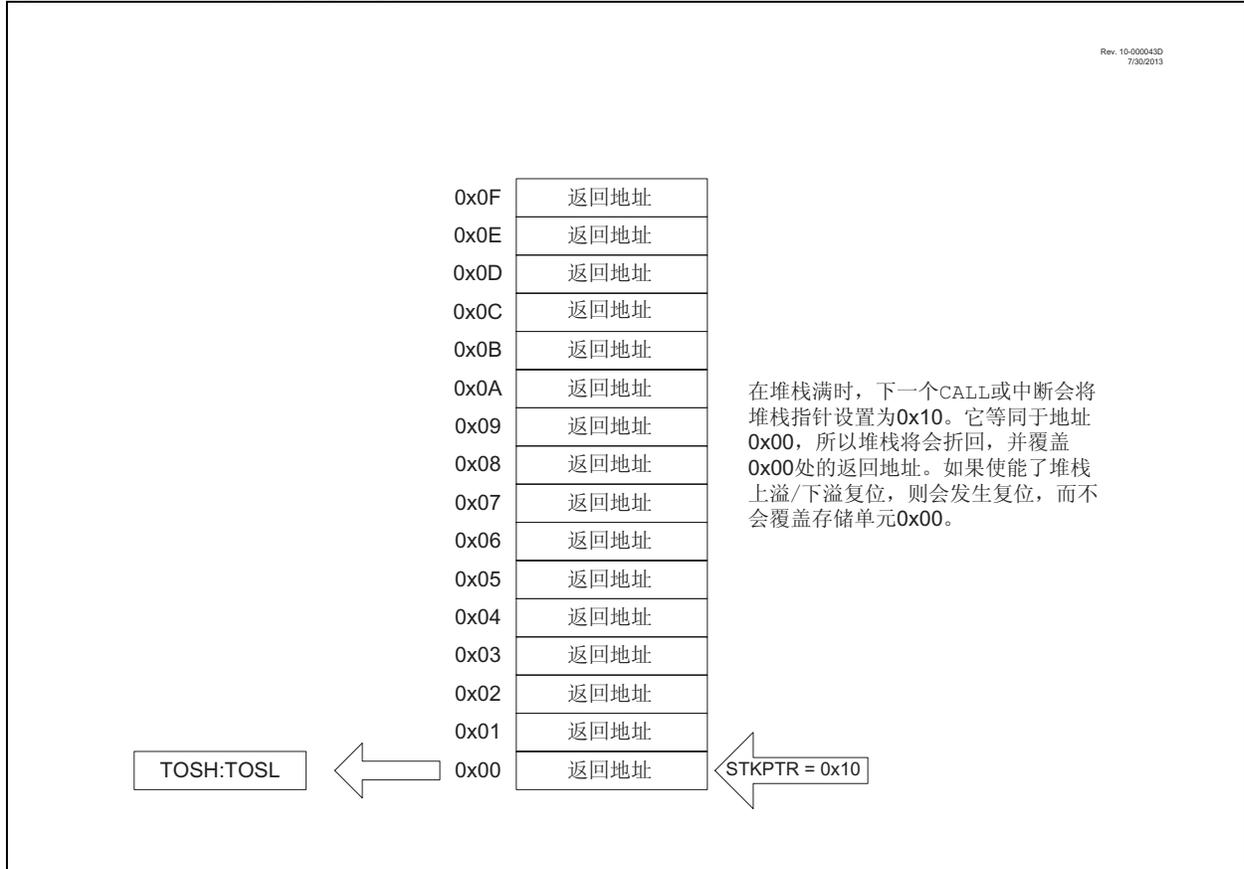


图 3-7: 访问堆栈示例 4



3.4.2 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。

3.5 间接寻址

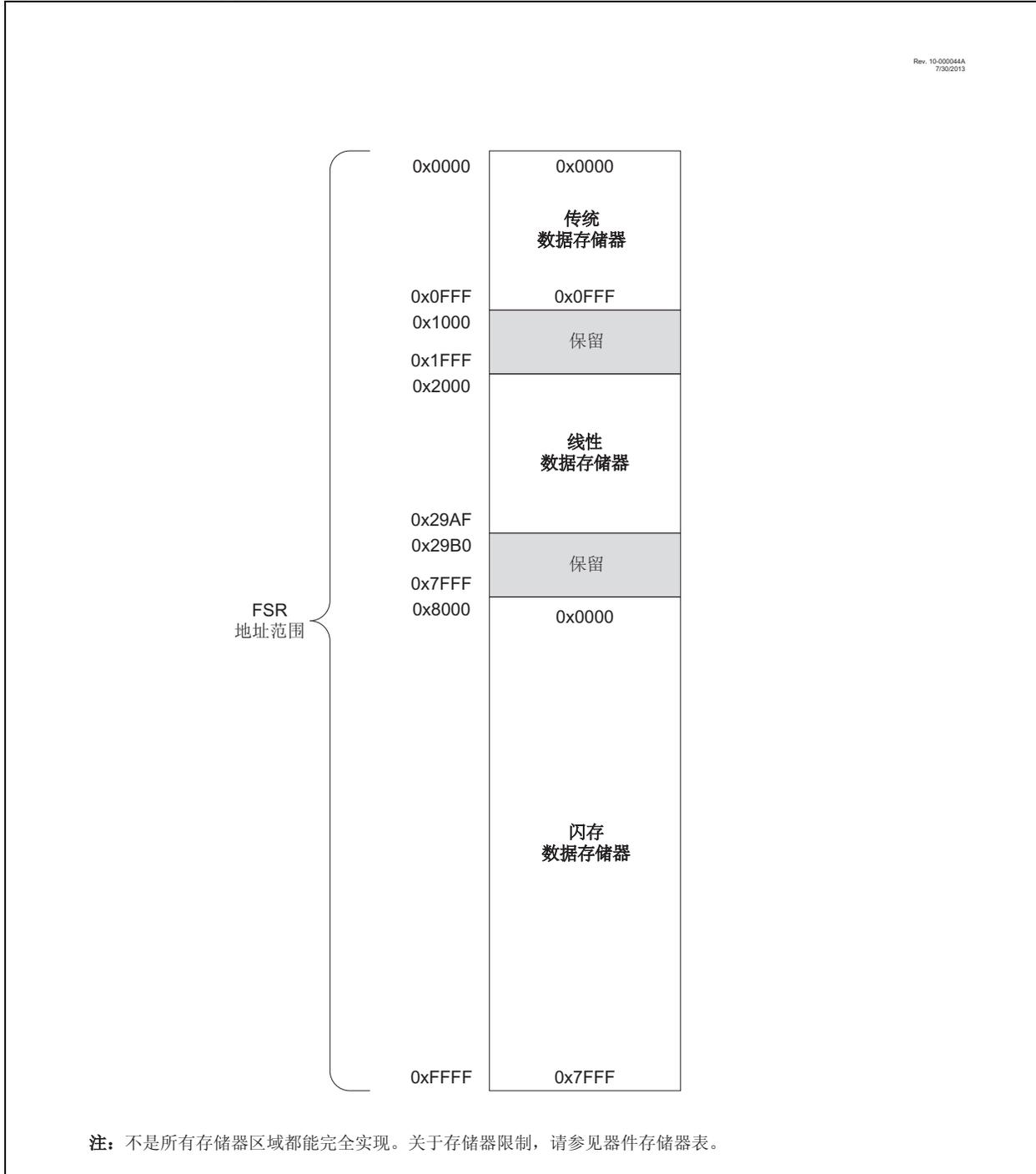
INDFn 寄存器不是物理寄存器。访问 **INDFn** 寄存器的所有指令实际上访问的是由文件选择寄存器（**FSR**）指定的地址处的寄存器。如果 **FSRn** 地址指定了两个 **INDFn** 寄存器中的一个，则读操作将返回 0，写操作将不会发生（虽然状态位可能会受影响）。**FSRn** 寄存器值由 **FSRnH** 和 **FSRnL** 对构成。

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

PIC12(L)F1612/16(L)F1613

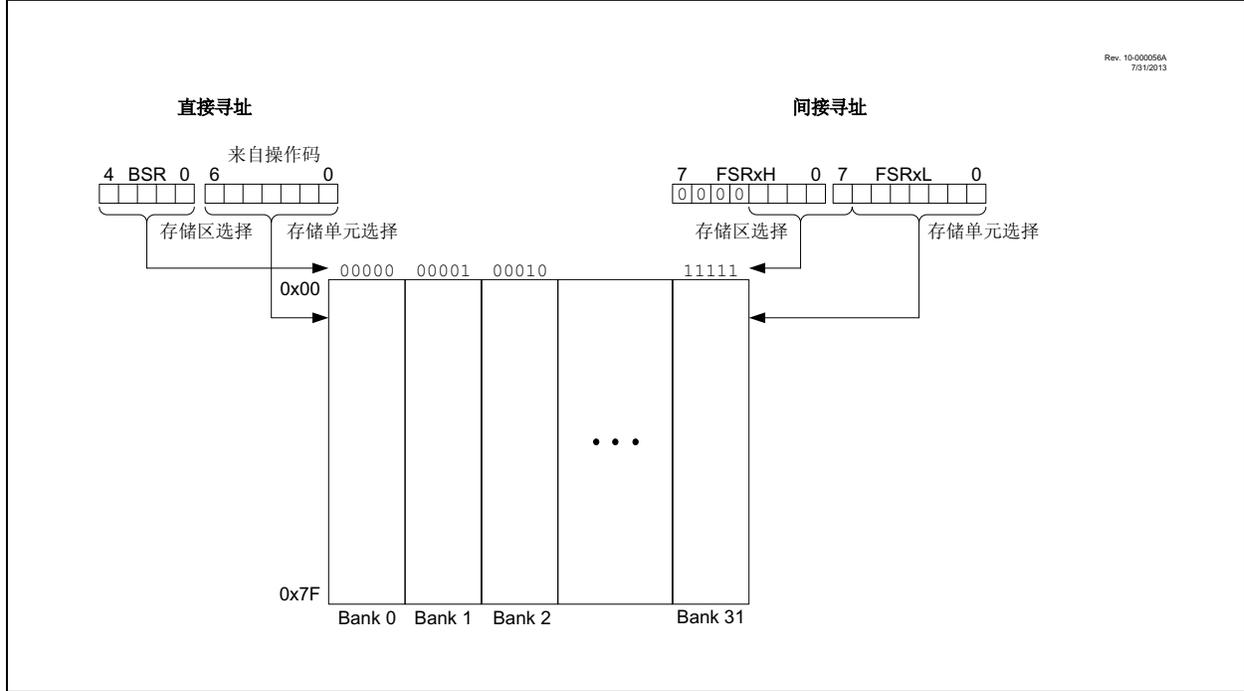
图 3-8: 间接寻址



3.5.1 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



PIC12(L)F1612/16(L)F1613

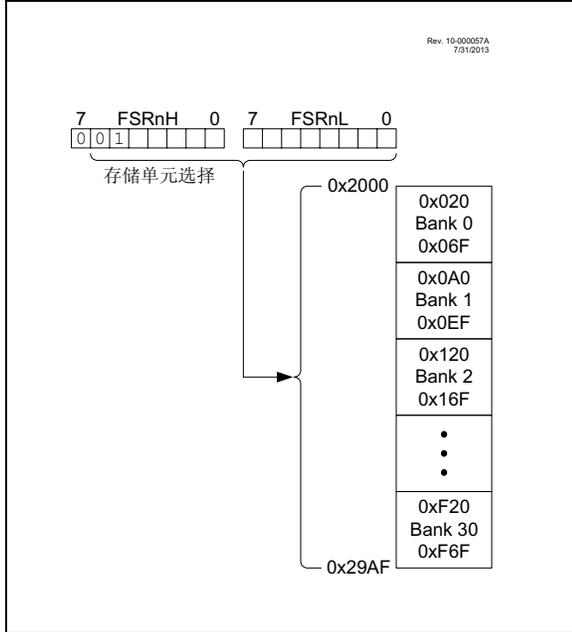
3.5.2 线性数据存储器

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

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

16 字节的公共存储器不包含在线性数据存储器区域中。

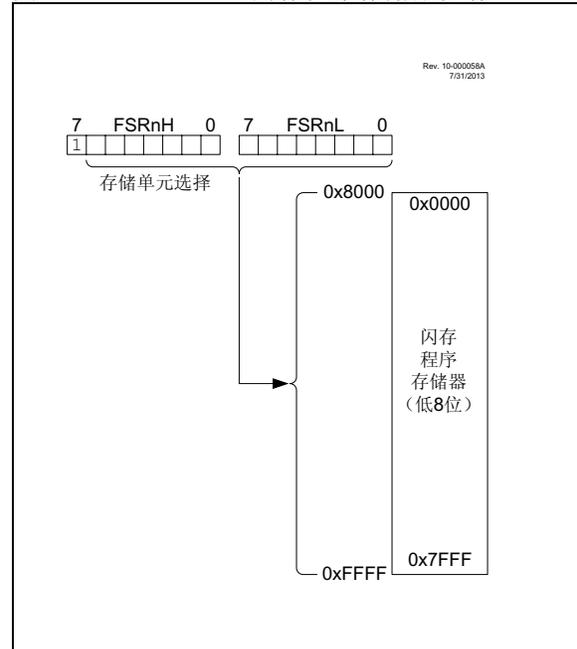
图 3-10: 线性数据存储器映射



3.5.3 闪存程序存储器

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

图 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。

注：	配置字中的 <u>DEBUG</u> 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。
-----------	---

PIC12(L)F1612/16(L)F1613

4.2 寄存器定义：配置字

寄存器 4-1: CONFIG1: 配置字 1

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

R/P-1	R/P-1	R/P-1	U-1	U-1	U-1	R/P-1	R/P-1
\overline{CP} ⁽²⁾	MCLRE	\overline{PWRTE}	—	—	—	FOSC<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 13-12 **未实现:** 读为 1
- 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 = 在 CLKOUT 引脚使能 CLKOUT 功能
- bit 10-9 **BOREN<1:0>:** 欠压复位使能位 ⁽¹⁾
 11 = 使能 BOR
 10 = 在工作期间使能 BOR, 在休眠期间禁止 BOR
 01 = BOR 由 BORCON 寄存器的 SBOREN 位控制
 00 = 禁止 BOR
- bit 8 **未实现:** 读为 1
- bit 7 **CP:** 代码保护位 ⁽²⁾
 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 **MCLRE:** 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 \overline{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 **PWRTE:** 上电延时定时器使能位
 1 = 禁止 PWRT
 0 = 使能 PWRT
- bit 4-2 **未实现:** 读为 1
- bit 1-0 **FOSC<1:0>:** 振荡器选择位
 11 = ECH: 外部时钟, 高功耗模式: 在 CLKIN 引脚上
 10 = ECM: 外部时钟, 中等功耗模式: 在 CLKIN 引脚上
 01 = ECL: 外部时钟, 低功耗模式: 在 CLKIN 引脚上
 00 = INTOSC 振荡器: CLKIN 引脚为 I/O 功能

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。

PIC12(L)F1612/16(L)F1613

寄存器 4-2: CONFIG2: 配置字 2

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

R/P-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
ZCDDIS	—	—	—	—	—	WRT<1:0>	
bit 7						bit 0	

图注:

R = 可读位 P = 可编程位 U = 未实现位, 读为 1
 0 = 清零 1 = 置 1 -n = 空白时或批量擦除后的值

- bit 13 **LVP:** 低电压编程使能位 ⁽¹⁾
 1 = 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 ⁽³⁾
 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 ⁽²⁾
 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 = 堆栈上溢或下溢不会导致复位
- bit 8 **PLLEN:** PLL 使能位
 1 = 使能 4xPLL
 0 = 禁止 4xPLL
- bit 7 **ZCDDIS:** 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 6-2 **未实现:** 读为 1
- bit 1-0 **WRT<1:0>:** 闪存自写保护位
2 kW 闪存 (PIC12(L)F1612/16(L)F1613):
 11 = OFF — 关闭写保护
 10 = BOOT — 000h 至 1FFh 被写保护, 200h 至 7FFh 可由 PMCON 控制修改
 01 = HALF — 000h 至 3FFh 被写保护, 400h 至 7FFh 可由 PMCON 控制修改
 00 = ALL — 000h 至 7FFh 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。

PIC12(L)F1612/16(L)F1613

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>		WDTCPS<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 ⁽¹⁾		

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

PIC12(L)F1612/16(L)F1613

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 _{IN} = 31 kHz)	
11111	01011	1:65536	2 ¹⁶	2 s	是
10011	10011	1:32	2 ⁵	1 ms	否
...	...				
11110	11110				
10010	10010	1:8388608	2 ²³	256 s	否
10001	10001	1:4194304	2 ²²	128 s	
10000	10000	1:2097152	2 ²¹	64 s	
01111	01111	1:1048576	2 ²⁰	32 s	
01110	01110	1:524299	2 ¹⁹	16 s	
01101	01101	1:262144	2 ¹⁸	8 s	
01100	01100	1:131072	2 ¹⁷	4 s	
01011	01011	1:65536	2 ¹⁶	2 s	
01010	01010	1:32768	2 ¹⁵	1 s	
01001	01001	1:16384	2 ¹⁴	512 ms	
01000	01000	1:8192	2 ¹³	256 ms	
00111	00111	1:4096	2 ¹²	128 ms	
00110	00110	1:2048	2 ¹¹	64 ms	
00101	00101	1:1024	2 ¹⁰	32 ms	
00100	00100	1:512	2 ⁹	16 ms	
00011	00011	1:256	2 ⁸	8 ms	
00010	00010	1:128	2 ⁷	4 ms	
00001	00001	1:64	2 ⁶	2 ms	
00000	00000	1:32	2 ⁵	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。

PIC12(L)F1612/16(L)F1613

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 $\text{WRT}\langle 1:0 \rangle$ 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/PIC16(L)F1613 Memory Programming Specification”（DS40001720）。

PIC12(L)F1612/16(L)F1613

4.6 器件 ID 和版本 ID

存储单元 8006h 是存储器件 ID 和版本 ID 的位置。高 9 位保存器件 ID。低 5 位保存版本 ID。关于访问这些存储单元的更多信息，请参见第 10.4 节“用户 ID、器件 ID 和配置字访问”。

开发工具（如器件编程器和调试器）可用于读取器件 ID 和版本 ID。

4.7 寄存器定义：器件 ID

寄存器 4-4: DEVID: 器件 ID 寄存器

R	R	R	R	R	R	R	R
DEV<13:8>							
bit 13							bit 8
R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7							bit 0

图注:

R = 可读位

1 = 置 1

0 = 清零

bit 13-0

DEVID<13:0>: 器件 ID 位

器件	DEVID<13:0> 值
PIC12F1612	11 0000 0101 1000 (3058h)
PIC12LF1612	11 0000 0101 1001 (3059 h)
PIC16F1613	11 0000 0100 1100 (304Ch)
PIC16LF1613	11 0000 0100 1101 (304Dh)

寄存器 4-5: REVID: 版本 ID 寄存器

R	R	R	R	R	R	R	R
REV<13:8>							
bit 13							bit 8
R	R	R	R	R	R	R	R
REV<7:0>							
bit 7							bit 0

图注:

R = 可读位

1 = 置 1

0 = 清零

bit 13-0

REVID<13:0>: 版本 ID 位

PIC12(L)F1612/16(L)F1613

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，同时最大限度地发挥性能并降低功耗。图5-1给出了振荡器模块的框图。

时钟源可由外部振荡器提供。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。

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）

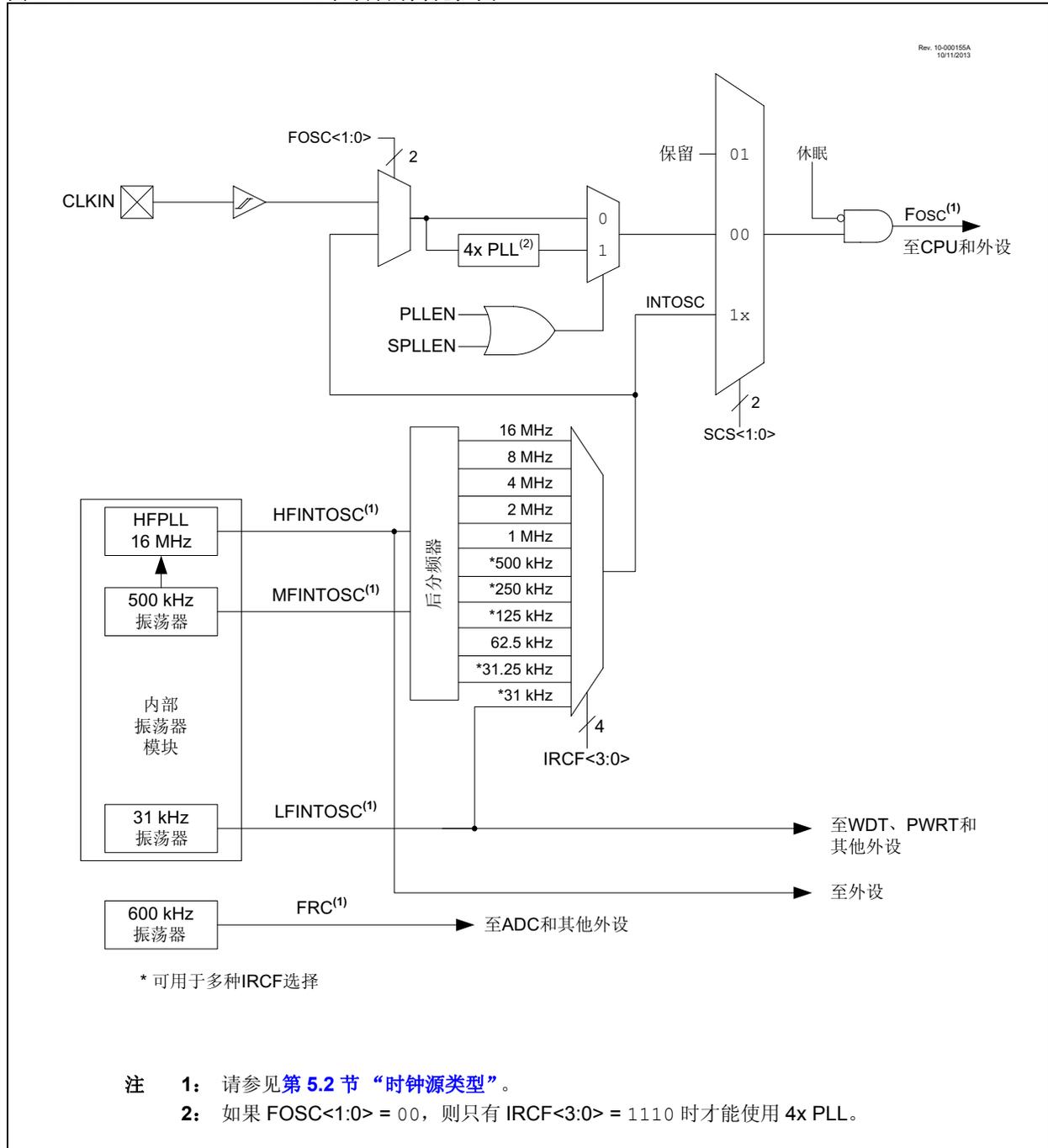
时钟源模式通过配置字中的 FOSC<1:0> 位进行选择。FOSC 位决定在器件初次上电时使用的振荡器类型。

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。

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。

PIC12(L)F1612/16(L)F1613

图 5-1: PIC® MCU 时钟源的简化框图



PIC12(L)F1612/16(L)F1613

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。

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

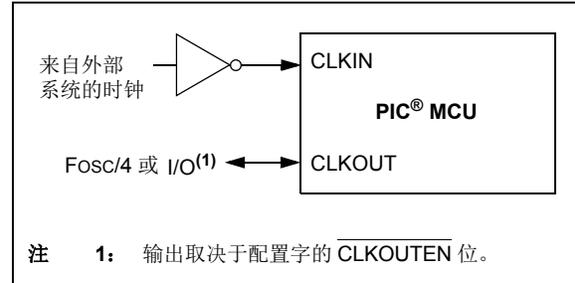
外部时钟（External Clock, EC）模式允许外部产生的逻辑电平信号作为系统时钟源。工作在该模式下时，外部时钟源连接到 CLKIN 输入。CLKOUT 可用作通用 I/O 或 CLKOUT。图 5-2 给出了 EC 模式的引脚连接图。

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



5.2.2 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。

PIC12(L)F1612/16(L)F1613

5.2.2.3 内部振荡器频率调整

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

OSCTUNE 寄存器的默认值为 0。该值是一个 6 位的二进制补码。值为 1Fh 时，将调整为最高频率。值为 20h 时，将调整为最低频率。

当 OSCTUNE 寄存器被修改时，振荡器频率将开始转变为新频率。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，其工作不受频率改变的影响。

5.2.2.4 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准的 31 kHz 内部时钟源。

LFINTOSC 的输出连接到多路开关（见图 5-1）。使用 OSCCON 寄存器的 IRCF<3:0> 位，通过软件选择 31 kHz。更多信息，请参见第 5.2.2.8 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）

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> 位都被设置为 0111，频率选择被设置为 500 kHz。用户可以修改 IRCF 位来选择其他频率。

通过 OSCCON 寄存器的 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，以使用 4x PLL，或者配置字的 PLEN 位必须编程为 1。

注： 使用配置字的 PLEN 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 将不能用于内部振荡器。要将 4x PLL 用于内部振荡器，SCS 位必须设置为 00。

5.2.2.8 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 5-3）。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。

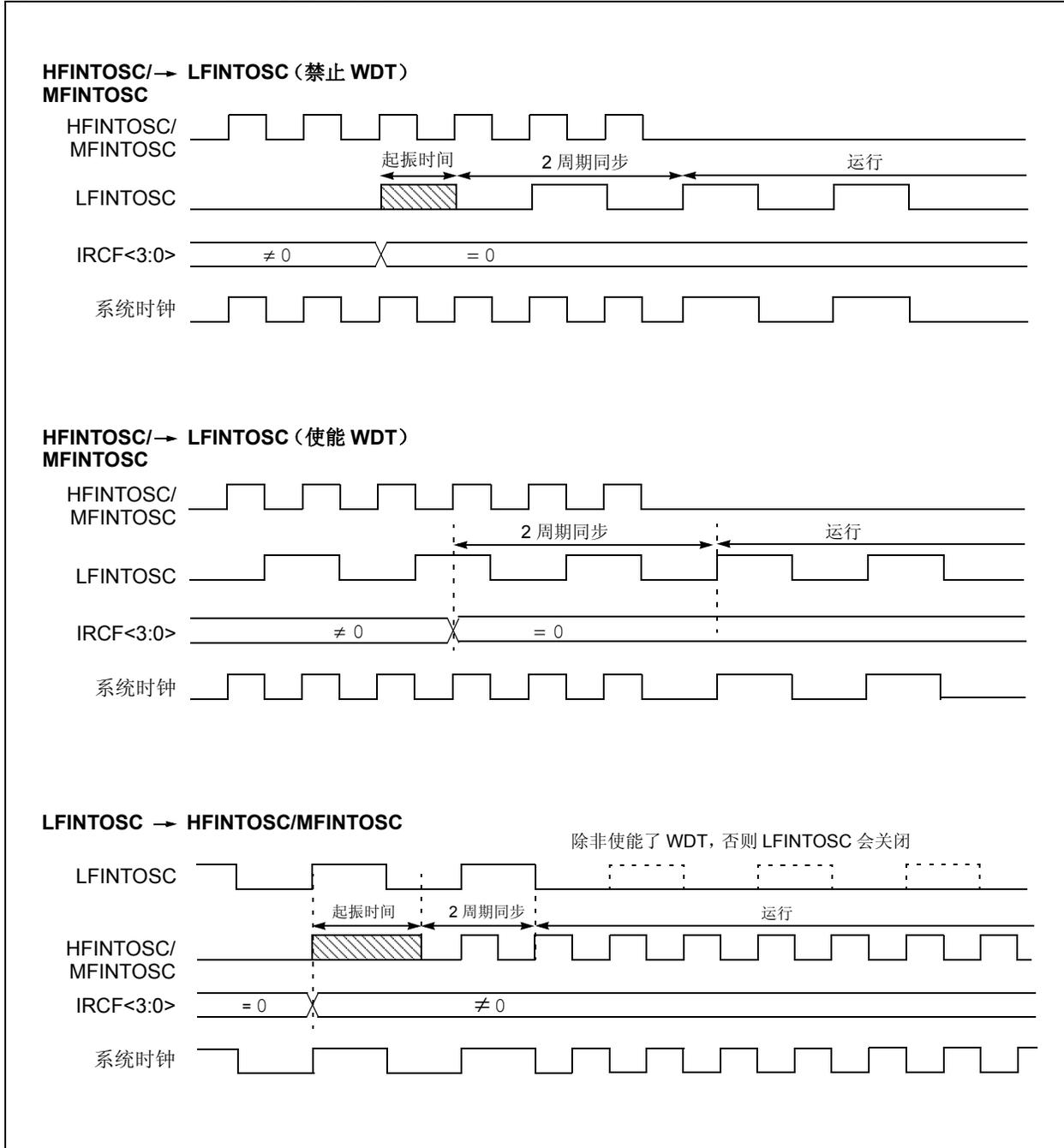
更多信息，请参见图 5-3。

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 28.0 节“电气规范”中的振荡器表。

PIC12(L)F1612/16(L)F1613

图 5-3: 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)

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

5.3.1 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。

表 5-1: 振荡器切换延时

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

注 1: PLL 无效。

PIC12(L)F1612/16(L)F1613

5.4 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 PLEN = 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 PLEN = 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⁽¹⁾

1001 = 250 kHz HF⁽¹⁾

1000 = 125 kHz HF⁽¹⁾

0111 = 500 kHz MF (复位时的默认值)

0110 = 250 kHz MF

0101 = 125 kHz MF

0100 = 62.5 kHz MF

0011 = 31.25 kHz HF⁽¹⁾

0010 = 31.25 kHz MF

000x = 31 kHz LF

bit 2

未实现: 读为 0

bit 1-0

SCS<1:0>: 系统时钟选择位

1x = 内部振荡器模块

01 = 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。

PIC12(L)F1612/16(L)F1613

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	—	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

q = 条件值

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

PIC12(L)F1612/16(L)F1613

寄存器 5-3: OSCTUNE: 振荡器调节寄存器

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

图注:

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

bit 7-6 未实现: 读为 0
bit 5-0 **TUN<5:0>**: 频率调节位
 100000 = 最低频率
 ·
 ·
 ·
 111111 =
 000000 = 振荡器模块以出厂时校准的频率运行
 000001 =
 ·
 ·
 ·
 011110 =
 011111 = 最高频率

表 5-2: 与时钟源相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		62
OSCSTAT	—	PLLRC	—	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	63
OSCTUNE	—	—	TUN<5:0>						64

图注: — = 未实现位, 读为 0。时钟源不使用阴影单元。

表 5-3: 与时钟源相关的配置字汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	48
	7:0	CP	MCLRE	PWRTE	—	—	—	FOSC<1:0>		

图注: — = 未实现位, 读为 0。时钟源不使用阴影单元。

6.0 复位

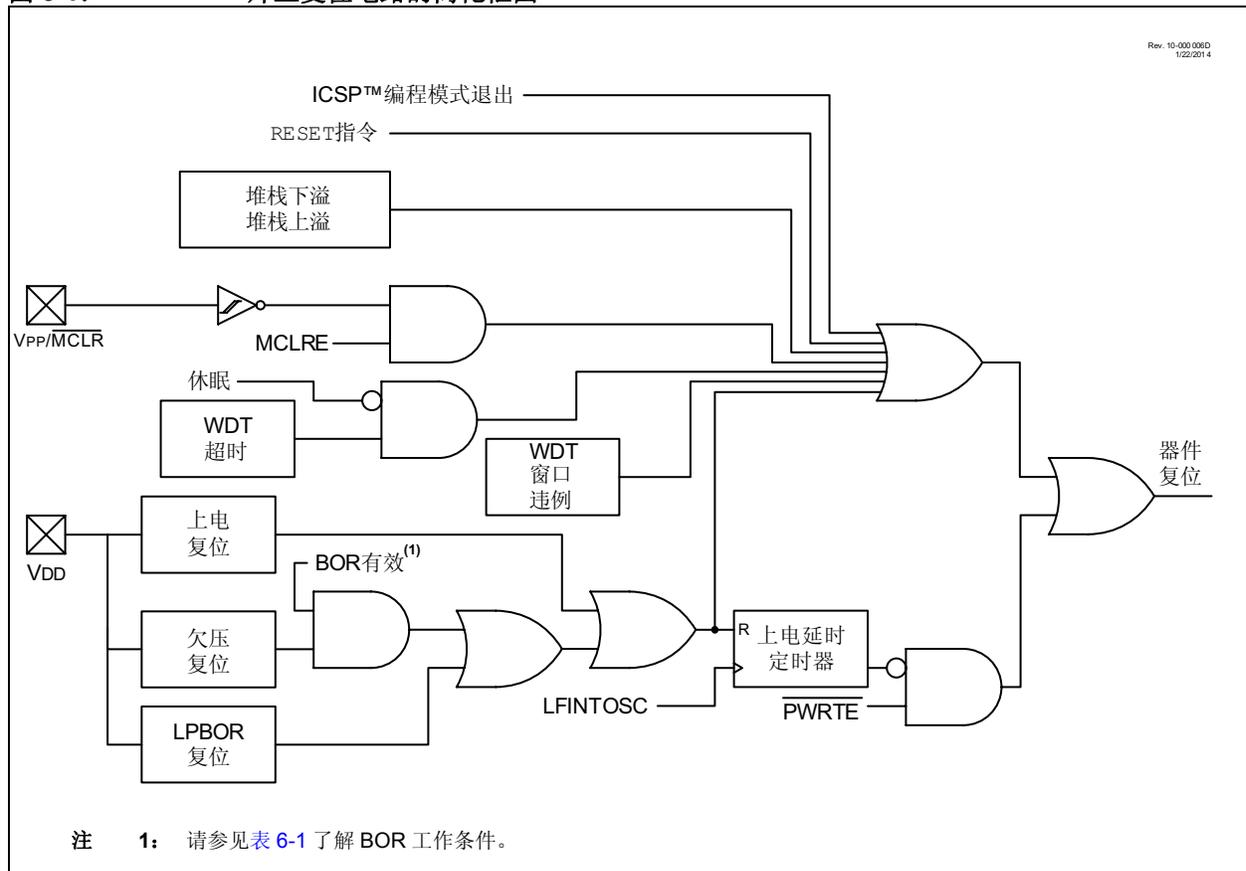
该器件的复位有几种方式：

- 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



PIC12(L)F1612/16(L)F1613

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 PWRT_{EN} 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 V_{BOR} 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 ⁽¹⁾ (BORRDY = 1)
10	X	唤醒	有效	等待 BOR 就绪 (BORRDY = 1)
		休眠	禁止	
01	1	X	有效	等待 BOR 就绪 ⁽¹⁾ (BORRDY = 1)
	0	X	禁止	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”的特殊情况下，启动时没有任何延时。在 CPU 准备好执行指令之前，BOR 就绪标志会置 1 (BORRDY = 1)，这是因为 BOR 电路通过 BOREN<1:0> 位被强制开启。

6.2.1 BOR 总是开启

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

BOR 保护在休眠期间有效。BOR 不会延迟从休眠中唤醒。

6.2.2 BOR 在休眠模式下关闭

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

BOR 保护在休眠期间无效。器件唤醒会被延迟，直到 BOR 就绪为止。

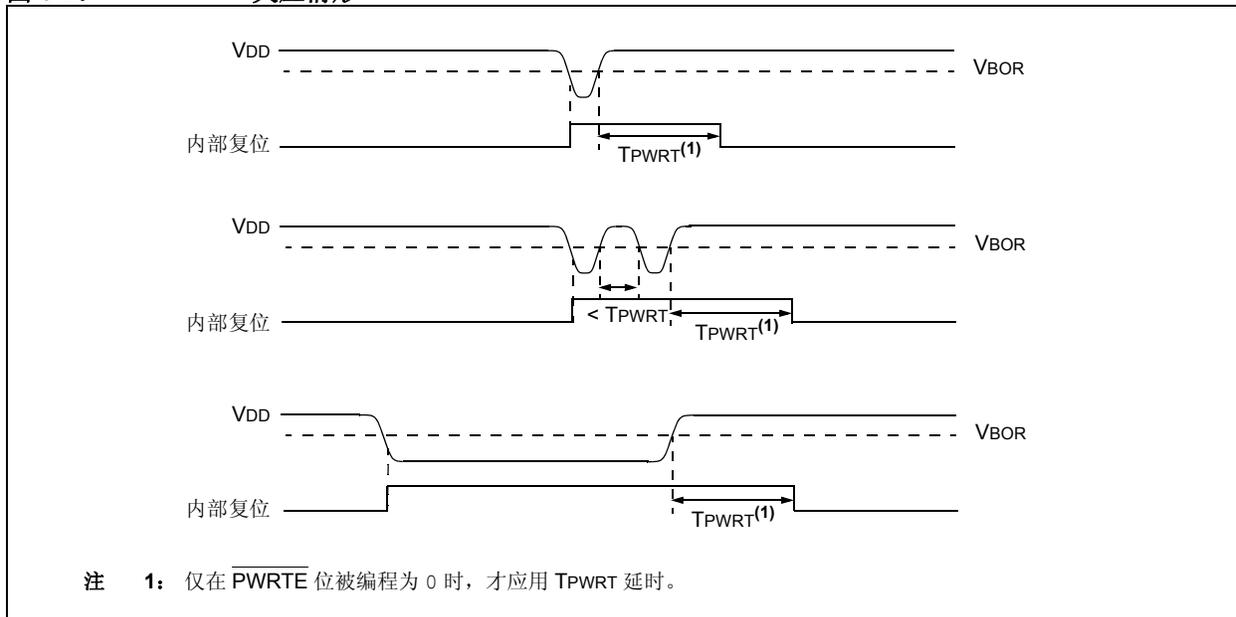
6.2.3 通过软件对 BOR 进行控制

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 $\text{BOREN}\langle 1:0 \rangle = 01$:

1 = 使能 BOR

0 = 禁止 BOR

如果配置字中的 $\text{BOREN}\langle 1:0 \rangle \neq 01$:

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 ⁽¹⁾

如果 $\text{BOREN}\langle 1:0 \rangle = 10$ (在休眠模式下禁止) 或 $\text{BOREN}\langle 1:0 \rangle = 01$ (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 $\text{BOREN}\langle 1:0 \rangle = 11$ (总是开启) 或 $\text{BOREN}\langle 1:0 \rangle = 00$ (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: $\text{BOREN}\langle 1:0 \rangle$ 位位于配置字中。

PIC12(L)F1612/16(L)F1613

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 $\overline{\text{BOR}}$ 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 $\overline{\text{LPBOR}}$ 位控制。在器件被擦除后，LPBOR 模块默认设为禁止。

6.5 $\overline{\text{MCLR}}$

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

表 6-2: $\overline{\text{MCLR}}$ 配置

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

6.5.1 $\overline{\text{MCLR}}$ 使能

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

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

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

6.5.2 $\overline{\text{MCLR}}$ 禁止

当 $\overline{\text{MCLR}}$ 被禁止时，引脚将用作通用输入，内部弱上拉由软件控制。更多信息，请参见第 12.3 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 CLRWDI 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 $\overline{\text{WDTWV}}$ 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 $\overline{\text{RI}}$ 位将设置为 0。关于发生 RESET 指令之后的默认条件，请参见表 6-4。

6.8 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 3.4.2 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 $\overline{\text{PWRTE}}$ 位控制。

6.11 启动序列

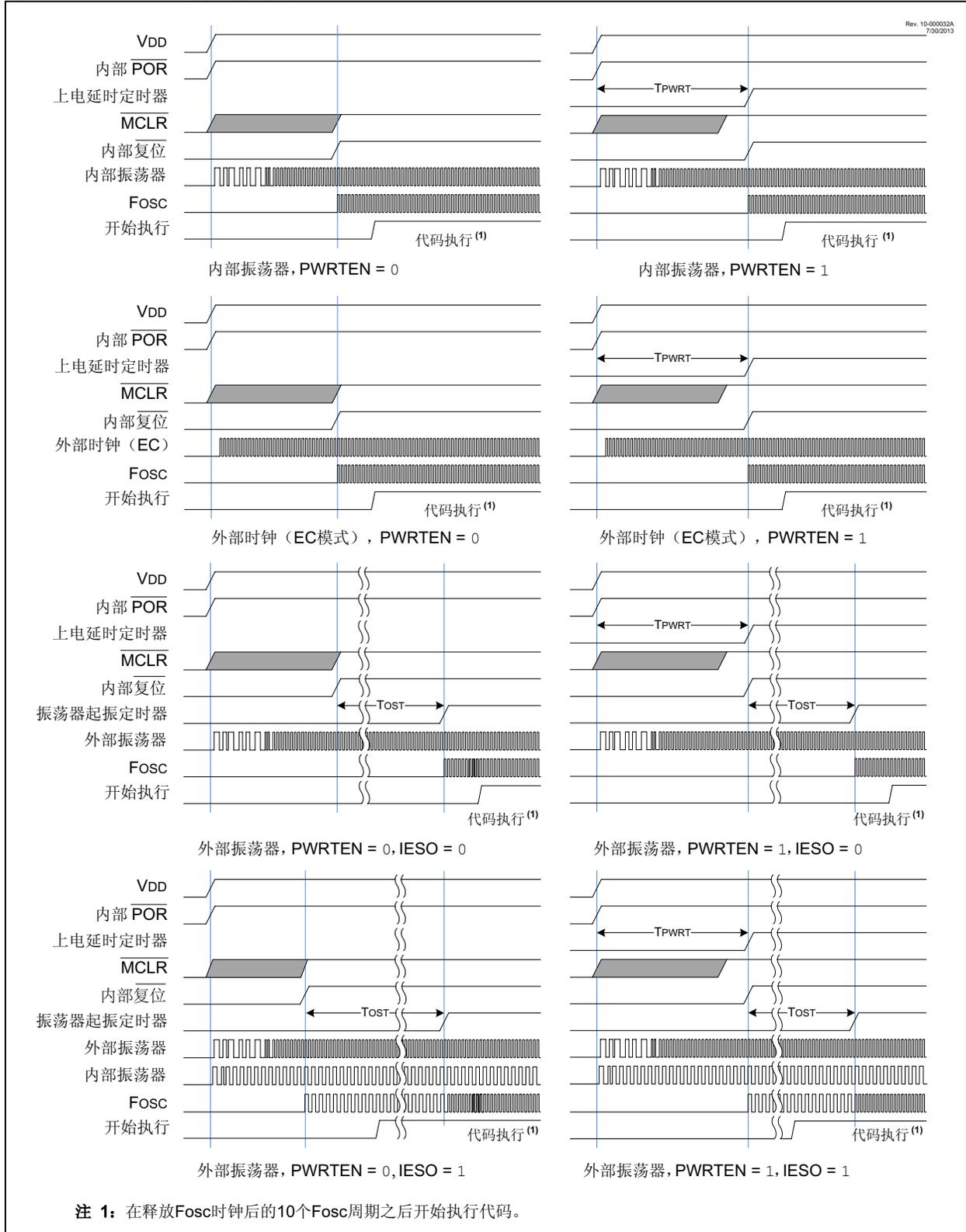
在 POR 或 BOR 释放时，只有先发生以下事件，器件才会开始执行：

1. 上电延时定时器运行完毕 (如果使能)。
2. $\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ 复位无关。如果 $\overline{\text{MCLR}}$ 保持低电平的时间足够长，上电延时定时器将会延时结束。当 $\overline{\text{MCLR}}$ 变为高电平时，器件将在 10 个 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



PIC12(L)F1612/16(L)F1613

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	RWDT	RMCLR	RI	POR	BOR	TO	PD	条件
0	0	1	1	1	0	x	1	1	上电复位
0	0	1	1	1	0	x	0	x	非法的，POR 时 TO 被置 1
0	0	1	1	1	0	x	x	0	非法的，POR 时 PD 被置 1
0	0	u	1	1	u	0	1	1	欠压复位
u	u	0	u	u	u	u	0	u	WDT 复位
u	u	u	u	u	u	u	0	0	被 WDT 从休眠状态唤醒
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 MCLR 复位
u	u	u	0	u	u	u	1	0	休眠期间的 MCLR 复位
u	u	u	u	0	u	u	u	u	执行了 RESET 指令
1	u	u	u	u	u	u	u	u	堆栈上溢复位 (STVREN = 1)
u	1	u	u	u	u	u	u	u	堆栈下溢复位 (STVREN = 1)

表 6-4: 特殊寄存器的复位条件

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

图注: u = 不变, x = 未知, - = 未实现位, 读为 0。

注 1: 当器件被中断唤醒且全局中断允许位 GIE 被置 1 时, 返回地址被压入堆栈, 并且在执行 PC + 1 后, PC 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（ $\overline{\text{RWDT}}$ ）
- 堆栈下溢复位（ $\overline{\text{STKUNF}}$ ）
- 堆栈上溢复位（ $\overline{\text{STKOVF}}$ ）

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
$\overline{\text{STKOVF}}$	$\overline{\text{STKUNF}}$	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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 **STKOVF:** 堆栈上溢标志位
 1 = 发生了堆栈上溢
 0 = 未发生堆栈上溢或由固件清零
- bit 6 **STKUNF:** 堆栈下溢标志位
 1 = 发生了堆栈下溢
 0 = 未发生堆栈下溢或由固件清零
- bit 5 **WDTWV:** 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 CLRWDT 指令（由硬件清零））
- bit 4 **RWDT:** 看门狗定时器复位标志位
 1 = 未发生看门狗定时器复位或由固件置 1
 0 = 发生了看门狗定时器复位（由硬件清零）
- bit 3 **RMCLR:** 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 **RI:** RESET 指令标志位
 1 = 未执行 RESET 指令或由固件置 1
 0 = 执行了 RESET 指令（由硬件清零）
- bit 1 **POR:** 上电复位状态位
 1 = 未发生上电复位
 0 = 发生了上电复位（发生上电复位后必须用软件置 1）

PIC12(L)F1612/16(L)F1613

寄存器 6-2: PCON: 电源控制寄存器 (续)

bit 0 **BOR**: 欠压复位状态位
 1 = 未发生欠压复位
 0 = 发生了欠压复位 (发生上电复位或欠压复位后必须用软件置 1)

表 6-5: 与复位相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	67
PCON	STKOVF	STKUNF	WDTWV	RWD \bar{T}	RMCLR	RI	POR	BOR	71
STATUS	—	—	—	TO	PD	Z	DC	C	19
WDTCON0	—	—	WDTPS<4:0>					SEN	95

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	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	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	48
	7:0	CP	MCLRE	PWRTE	—	—	—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	49
	7:0	ZCDDIS	—	—	—	—	WRT<1:0>			
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			50
	7:0	—	WDTE<1:0>		WDTCPSC<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。

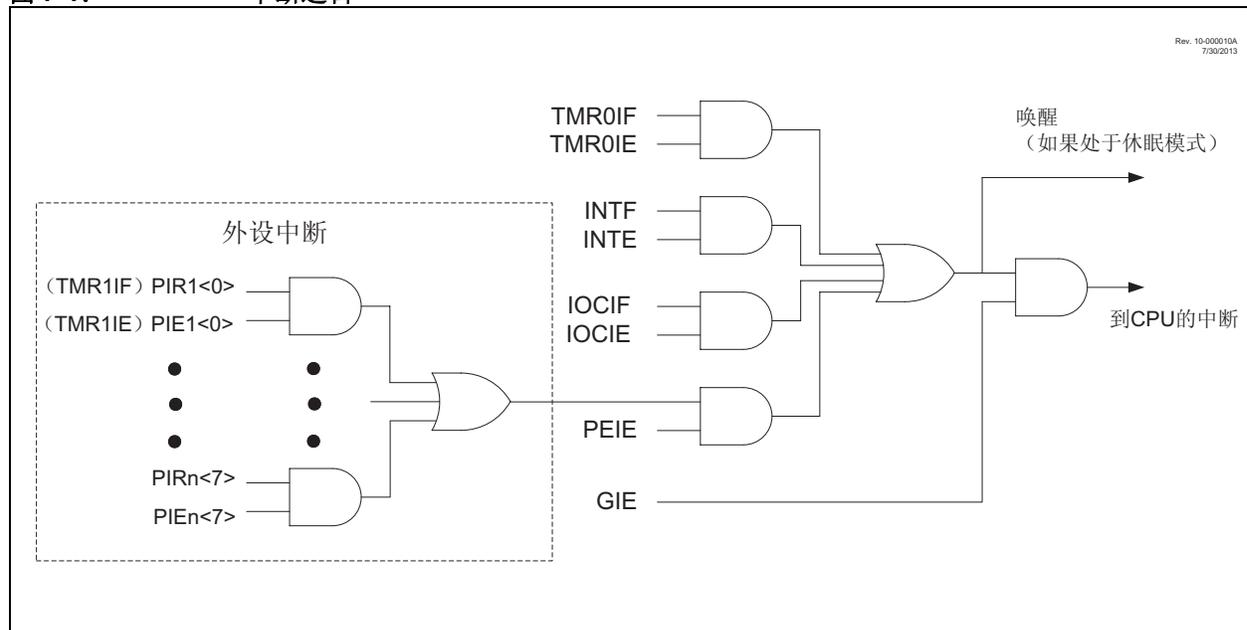
本章包含了关于中断的以下信息：

- 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



PIC12(L)F1612/16(L)F1613

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（Interrupt Service Routine, ISR）中的固件应通过查询中断标志位来确定中断源。退出 ISR 前必须清零中断标志位，以避免重复中断。由于 GIE 位清零，执行 ISR 期间发生的任何中断都会通过其中断标志位记录下来，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。

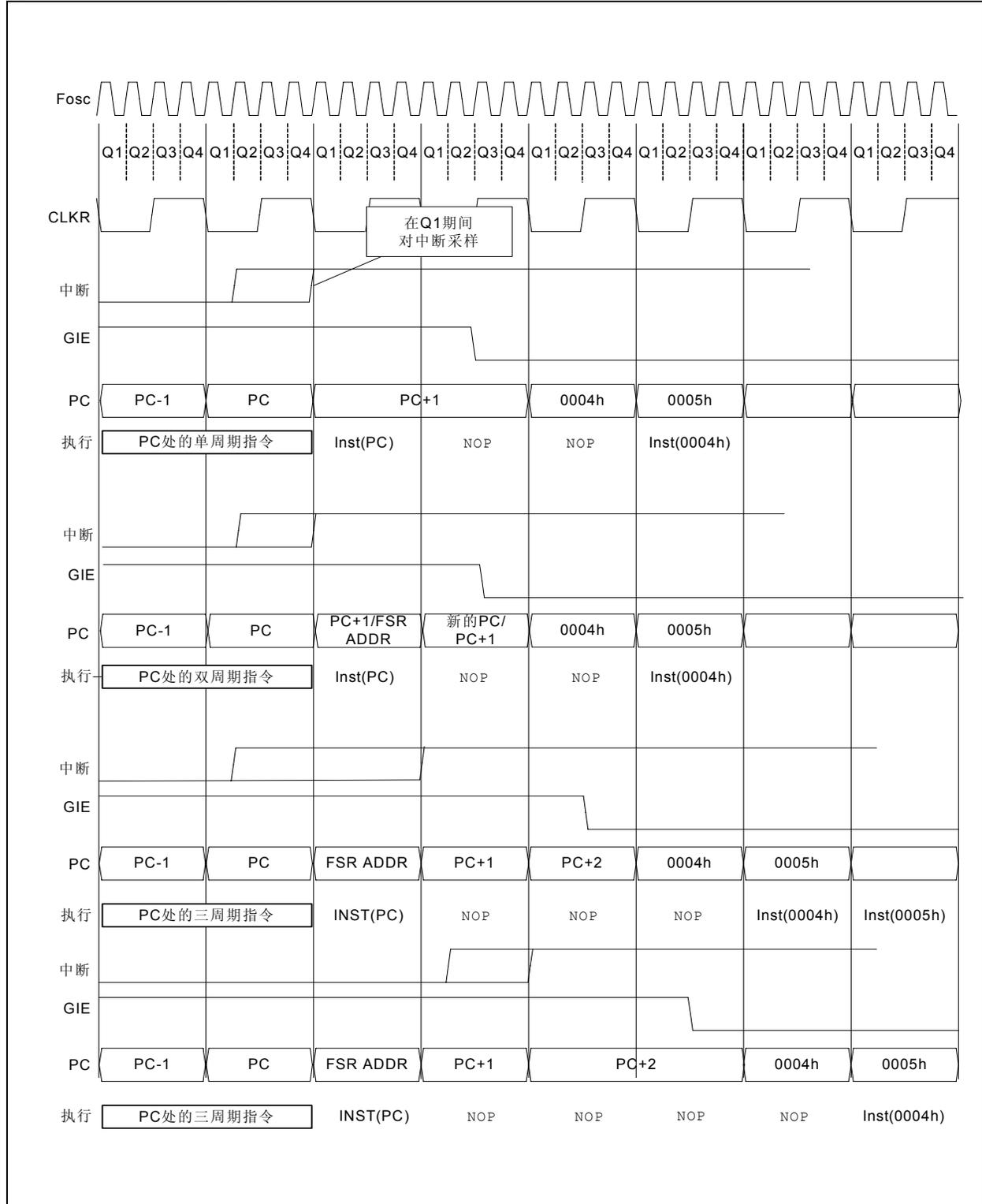
- 注 1：** 无论中断允许位状态如何，各中断标志位都会在中断发生时置 1。
- 2：** GIE 位清零时，将忽略所有中断。GIE 位清零期间发生的任何中断都会在 GIE 位再次置 1 时得到处理。

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。

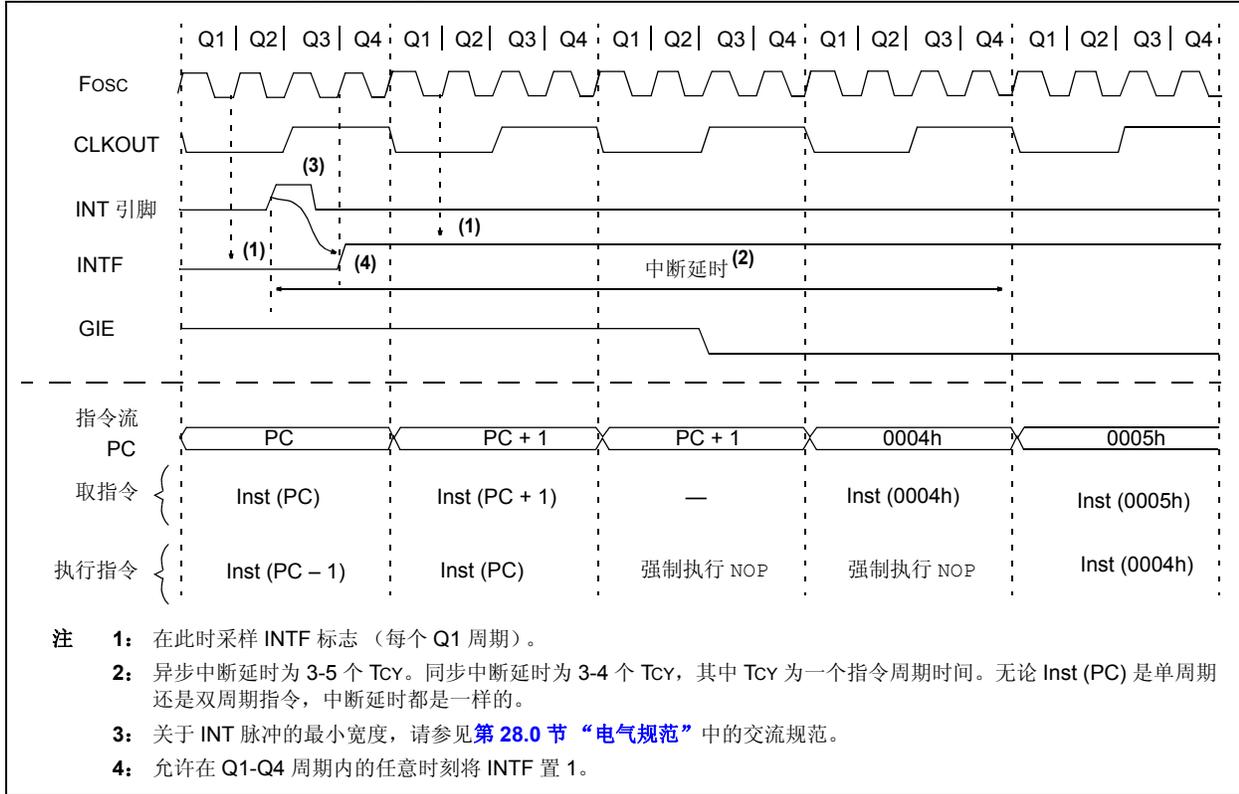
PIC12(L)F1612/16(L)F1613

图 7-2: 中断延时



PIC12(L)F1612/16(L)F1613

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

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

PIC12(L)F1612/16(L)F1613

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 ⁽¹⁾	PEIE ⁽²⁾	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCF ⁽³⁾
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7 **GIE:** 全局中断允许位 ⁽¹⁾

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 ⁽²⁾

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 **IOCFIE:** 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 **IOCF:** 电平变化中断标志位 ⁽³⁾

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: IOCF 标志位是只读位, 它在 IOCFx 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。

PIC12(L)F1612/16(L)F1613

寄存器 7-2: PIE1: 外设中断允许寄存器 1

R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	—	—	—	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 5-3 **未实现:** 读为 0
- 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。

PIC12(L)F1612/16(L)F1613

寄存器 7-3: **PIE2: 外设中断允许寄存器 2**

U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	C2IE ⁽¹⁾	C1IE	—	—	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 **未实现:** 读为 0
- bit 6 **C2IE:** 比较器 C2 中断允许位 ⁽¹⁾
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 4-3 **未实现:** 读为 0
- 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: 仅限 PIC16(L)F1613。

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

PIC12(L)F1612/16(L)F1613

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	U-0	U-0	U-0	U-0
—	—	CWGIE	ZCDIE	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 3-0 **未实现:** 读为 0

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

PIC12(L)F1612/16(L)F1613

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 **SCANIE:** 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 **CRCIE:** CRC 中断允许位
1 = 允许 CRC 中断
0 = 禁止 CRC 中断
- bit 5 **SMT2PWAIE:** 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 **SMT2PRAIE:** 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 **SMT2IE:** SMT2 上溢中断允许位
1 = 允许 SMT 上溢中断
0 = 禁止 SMT 上溢中断
- bit 2 **SMT1PWAIE:** 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 **SMT1PRAIE:** 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 **SMT1IE:** SMT1 上溢中断允许位
1 = 允许 SMT 上溢中断
0 = 禁止 SMT 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。

PIC12(L)F1612/16(L)F1613

寄存器 7-6: **PIR1: 外设中断请求寄存器 1**

R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7 **TMR1GIF:** Timer1 门控中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态
- bit 6 **ADIF:** 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 5-3 **未实现:** 读为 0
- bit 2 **CCP1IF:** CCP1 中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态
- bit 1 **TMR2IF:** Timer2 与 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 **TMR1IF:** Timer1 上溢中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态

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。

PIC12(L)F1612/16(L)F1613

寄存器 7-7: PIR2: 外设中断请求寄存器 2

U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	C2IF ⁽¹⁾	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7 **未实现:** 读为 0
- bit 6 **C2IF:** 比较器 C2 中断标志位 ⁽¹⁾
1 = 中断处于待处理状态
0 = 中断不处于待处理状态
- bit 5 **C1IF:** 比较器 C1 中断标志位
1 = 中断处于待处理状态
0 = 中断不处于待处理状态
- bit 4-3 **未实现:** 读为 0
- bit 2 **TMR6IF:** 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 **TMR4IF:** 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 **CCP2IF:** 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 1: 仅限 PIC16(L)F1613。

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。

PIC12(L)F1612/16(L)F1613

寄存器 7-8: PIR3: 外设中断请求寄存器 3

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
—	—	CWGIF	ZCDIF	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 **CWGIF:** 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 **ZCDIF:** 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 3-0 **未实现:** 读为 0

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。

PIC12(L)F1612/16(L)F1613

寄存器 7-9: 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 **SCANIF:** 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 **CRCIF:** 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 **SMT2PWAIF:** 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 **SMT2PRAIF:** 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 **SMT2IF:** SMT2 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 2 **SMT1PWAIF:** 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 **SMT1PRAIF:** 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 **SMT1IF:** SMT1 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。

PIC12(L)F1612/16(L)F1613

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	78
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			183
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	79
PIE2	—	C2IE ⁽¹⁾	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	80
PIE3	—	—	CWGIE	ZCDIE	—	—	—	—	81
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	82
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	83
PIR2	—	C2IF ⁽¹⁾	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	84
PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	85
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	86

图注: — = 未实现位, 读为 0。中断不使用阴影单元。

注 1: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

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{PD} 位被清零。
3. STATUS 寄存器的 \overline{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 14.0 节“固定参考电压 (FVR)”。

8.1 从休眠状态唤醒

发生以下任一事件将器件从休眠状态唤醒：

1. \overline{MCLR} 引脚上的外部复位输入（如果使能）
2. BOR 复位（如果使能）
3. POR 复位
4. 看门狗定时器（如果使能）
5. 任何外部中断

6. 可以在休眠期间运行的外设产生的中断（更多信息，请参见各个外设）

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

当执行 SLEEP 指令时，下一条指令 (PC + 1) 被预先取出。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。

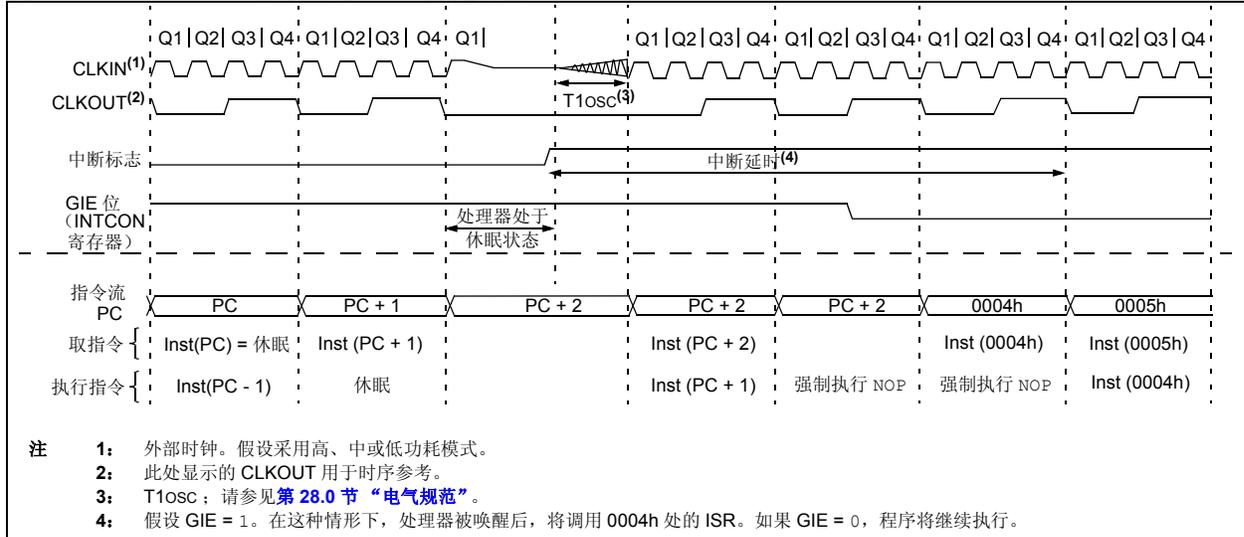
8.1.1 使用中断唤醒

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{TO} 位不会被置 1
 - STATUS 寄存器的 \overline{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{TO} 位将被置 1
 - STATUS 寄存器的 \overline{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是在 SLEEP 指令执行完毕之前被置 1。要确定是否执行了 SLEEP 指令，可测试 \overline{PD} 位。如果 \overline{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, 所以器件能够快速地退出休眠模式。在低功耗休眠模式下, 从休眠模式中唤醒时, 这些电路需要一个额外的延时, 然后才会恢复为正常配置并稳定下来。

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

8.2.2 休眠模式下的外设使用

选择低功耗休眠模式时, 一些可以在休眠模式下工作的外设将无法正常工作。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 24.11 节“休眠期间的操作”。

注: PIC12LF1612/16LF1613 不具有可配置的低功耗休眠模式。PIC12LF1612/16LF1613 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 V_{DD} 和 I/O 电压低于 PIC12F1612/16F1613。更多信息, 请参见第 28.0 节“电气规范”。

PIC12(L)F1612/16(L)F1613

8.3 寄存器定义：稳压器控制

寄存器 8-1: **VREGCON**: 稳压器控制寄存器 ⁽¹⁾

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

图注:

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

bit 7-2 **未实现:** 读为 0
bit 1 **VREGPM:** 稳压器功耗模式选择位
 1 = 休眠时使能低功耗休眠模式 ⁽²⁾
 休眠时消耗的电流最低, 唤醒速度较慢
 0 = 休眠时使能正常功耗模式 ⁽²⁾
 休眠时消耗的电流较高, 唤醒速度较快
bit 0 **保留:** 读为 1。保持该位置 1。

注 1: 仅限 PIC12F1612/16F1613。
注 2: 请参见第 28.0 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	78
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	143
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	143
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	143
IOCCP	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	144
IOCCN	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	144
IOCCF	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	144
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	79
PIE2	—	C2IE ⁽¹⁾	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	80
PIE3	—	—	CWGIE	ZCDIE	—	—	—	—	81
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	82
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	83
PIR2	—	C2IF ⁽¹⁾	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	84
PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	85
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	86
STATUS	—	—	—	T _O	P _D	Z	DC	C	19
WDTCON0	—	—	WDTPS<4:0>					SEN	95

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)F1613。

9.0 窗口看门狗定时器 (WDT)

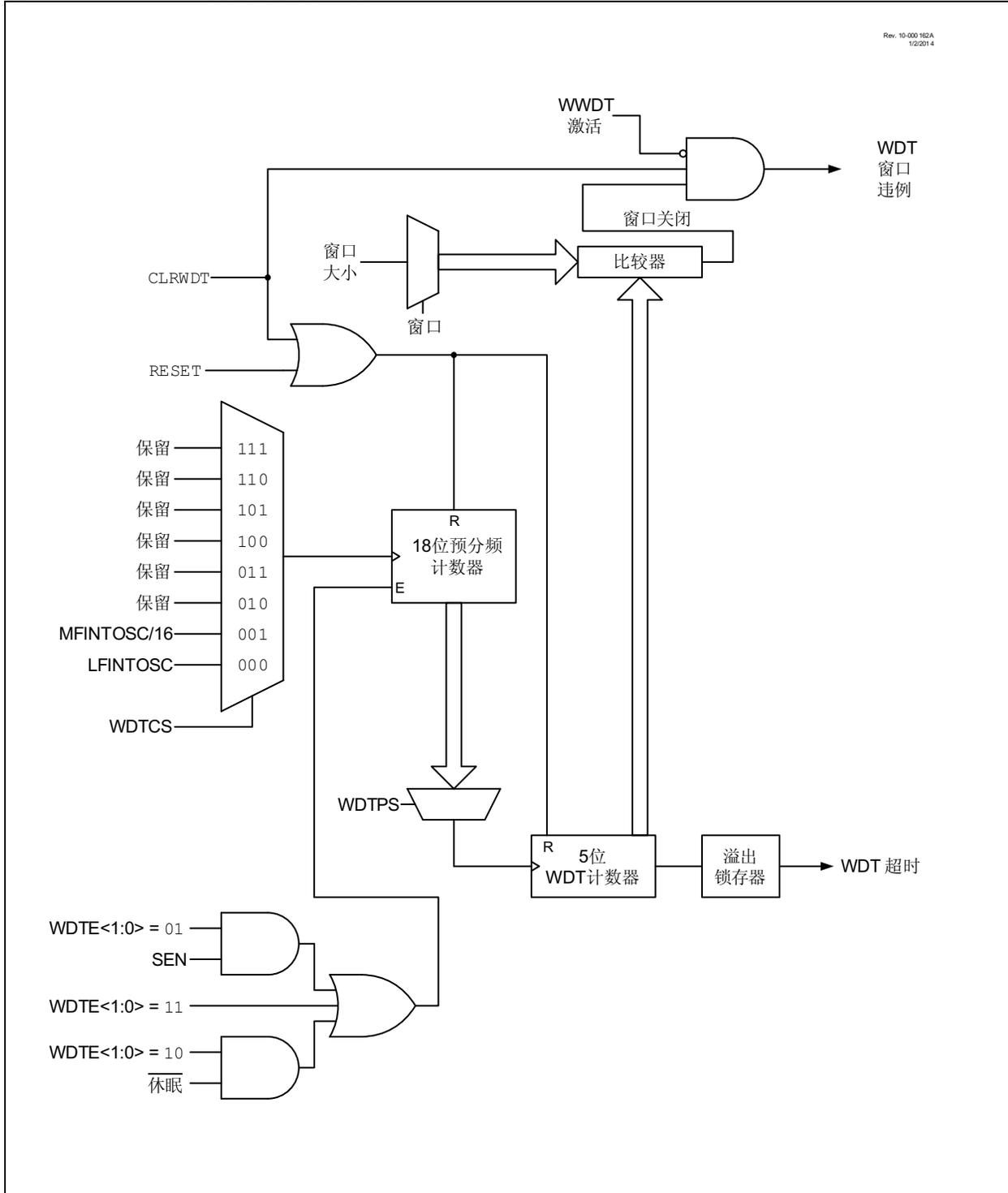
看门狗定时器 (WDT) 是一个系统定时器, 如果固件未在超时周期内发出 CLRWDT 指令, 看门狗定时器会产生复位。看门狗定时器通常用于使系统从意外事件中恢复。窗口看门狗定时器 (WDT) 的不同之处在于, 只有在超时周期期间的特定窗口内执行 CLRWDT 指令时才会接受该指令。

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

PIC12(L)F1612/16(L)F1613

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 WDTCON0 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 28.0 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 将总是开启。

WDT 保护在休眠期间有效。

9.2.2 WDT 在休眠模式下关闭

当配置字的 WDTE 位设置为 10 时，除非处于休眠模式，否则 WDT 将开启。

WDT 保护在休眠期间无效。

9.2.3 WDT 通过软件进行控制

当配置字的 WDTE 位设置为 01 时，WDT 将通过 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，CLRWDWT 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 CLRWDWT 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 CLRWDWT 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

9.5.1 CLRWDWT 注意事项（窗口模式）

在窗口模式下，必须先激活 WDT，之后的 CLRWDWT 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 CLRWDWT 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”。

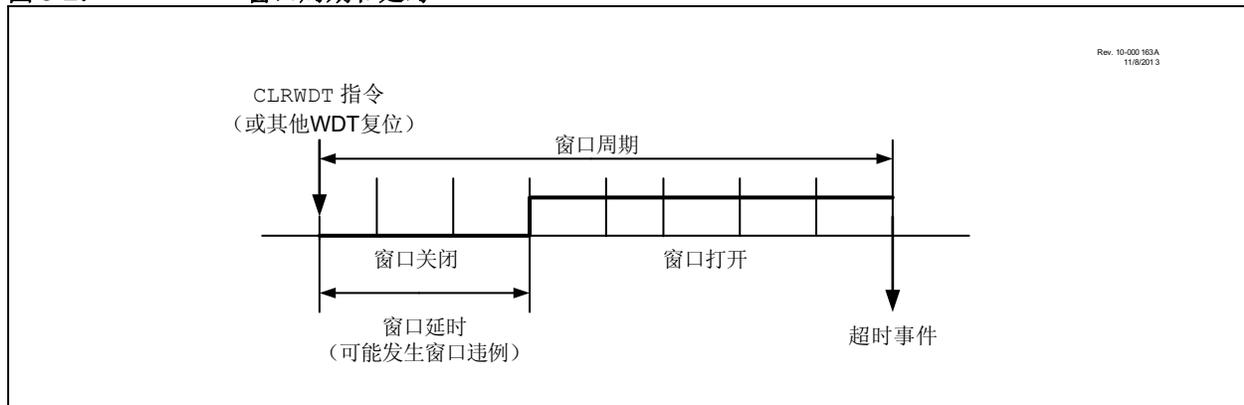
在器件处于休眠模式的情况下发生 WDT 超时，不会产生复位。器件将会唤醒并继续工作。STATUS 寄存器中的 TO 和 PD 位会发生改变，指示发生的事件。也可以使用 PCON 寄存器中的 RWDT 位。更多信息，请参见第 3.0 节“存储器构成”。

PIC12(L)F1612/16(L)F1613

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	
退出休眠 + 系统时钟 = XT、HS 或 LP	清零, 直到 OST 延时结束
更改 INTOSC 分频比 (IRCF 位)	不受影响

图 9-2: 窗口周期和延时



PIC12(L)F1612/16(L)F1613

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

位值 = 预分频比

11111 = 保留。产生最小的时间间隔 (1:32)

·
·
·

10011 = 保留。产生最小的时间间隔 (1:32)

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

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

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

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

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

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

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

01011 = 1:65536 (时间间隔标称值为 2s) (复位值)

01010 = 1:32768 (时间间隔标称值为 1s)

01001 = 1:16384 (时间间隔标称值为 512 ms)

01000 = 1:8192 (时间间隔标称值为 256 ms)

00111 = 1:4096 (时间间隔标称值为 128 ms)

00110 = 1:2048 (时间间隔标称值为 64 ms)

00101 = 1:1024 (时间间隔标称值为 32 ms)

00100 = 1:512 (时间间隔标称值为 16 ms)

00011 = 1:256 (时间间隔标称值为 8 ms)

00010 = 1:128 (时间间隔标称值为 4 ms)

00001 = 1:64 (时间间隔标称值为 2 ms)

00000 = 1:32 (时间间隔标称值为 1 ms)

bit 0 **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。

PIC12(L)F1612/16(L)F1613

寄存器 9-2: WDTCON1: 看门狗定时器控制寄存器 1

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

PIC12(L)F1612/16(L)F1613

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> ⁽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 7-0 **PSCNT<7:0>**: 预分频比选择低字节位⁽¹⁾

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> ⁽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 7-0 **PSCNT<15:8>**: 预分频比选择高字节位⁽¹⁾

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> ⁽¹⁾		
bit 7							bit 0

图注:

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

bit 7-3 **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⁽¹⁾

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。

PIC12(L)F1612/16(L)F1613

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>			62
PCON	STKOVF	STKUNF	WDTWV	RWD \bar{T}	RMCLR	RI	POR	BOR	71
STATUS	—	—	—	TO	PD	Z	DC	C	19
WDTCON0	—	—	WDTPS<4:0>				SEN		95
WDTCON1	—	WDTCS<2:0>		—	WINDOW<2:0>				96
WDTPSL	PSCNT<7:0>								97
WDTPSH	PSCNT<15:8>								97
WDTTMR	—	WDTTMR<4:0>				STATE	PSCNT<17:16>		97

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

表 9-4: 与看门狗定时器相关的配置字汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	48
	7:0	CP	MCLRE	PWRTE	—	—	FOSC<1:0>		—	
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			50
	7:0	—	WDTE<1:0>		WDTCPSS<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 ($\overline{CP} = 0$) ⁽¹⁾ 会禁止通过外部器件编程器对闪存程序存储器进行访问（读写操作）。代码保护不会影响自写和擦除功能。代码保护只能通过器件编程器对器件执行批量擦除操作，从而清除所有闪存程序存储器、配置位和用户 ID 而复位。

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

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

10.1 PMADRL 和 PMADRH 寄存器

PMADRH:PMADRL 寄存器对能寻址最大 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。

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

10.2 闪存程序存储器概述

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

在擦除某行之后，用户可以对该行的全部或部分内容进行再编程。要写入程序存储器行的数据将写入 14 位宽的数据写锁存器中。用户不能直接访问这些写锁存器，但可以通过连续写入 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。

PIC12(L)F1612/16(L)F1613

表 10-1: 闪存构成 (按器件)

器件	行擦除 (字)	写锁存器 (字)
PIC12(L)F1612	16	16
PIC16(L)F1613		

10.2.1 读取闪存程序存储器

要读取程序存储单元, 用户必须:

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

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

PMDATH:PMDATL 寄存器对将保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

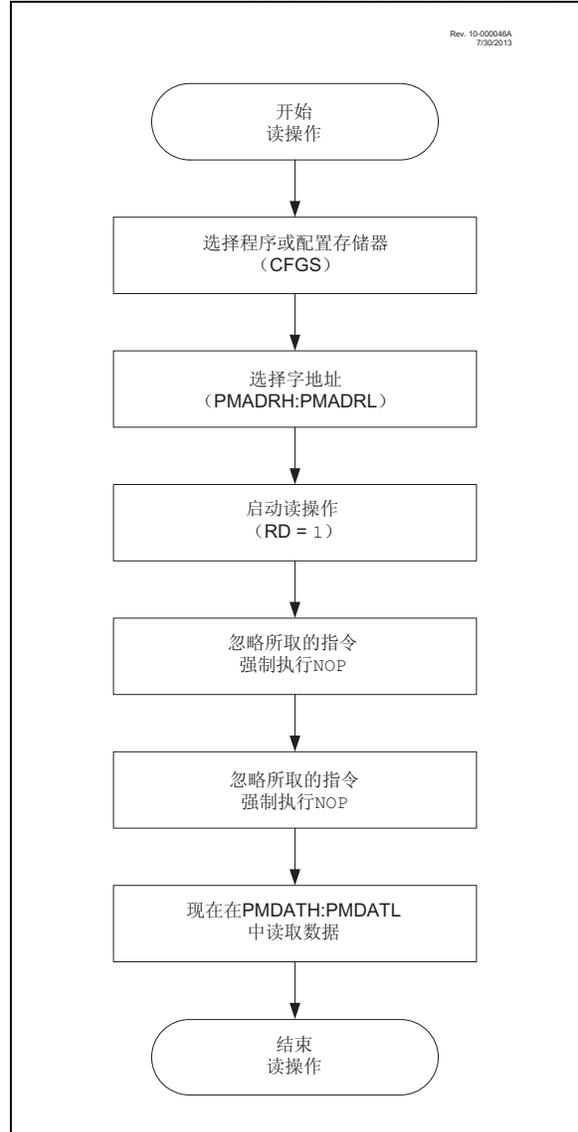
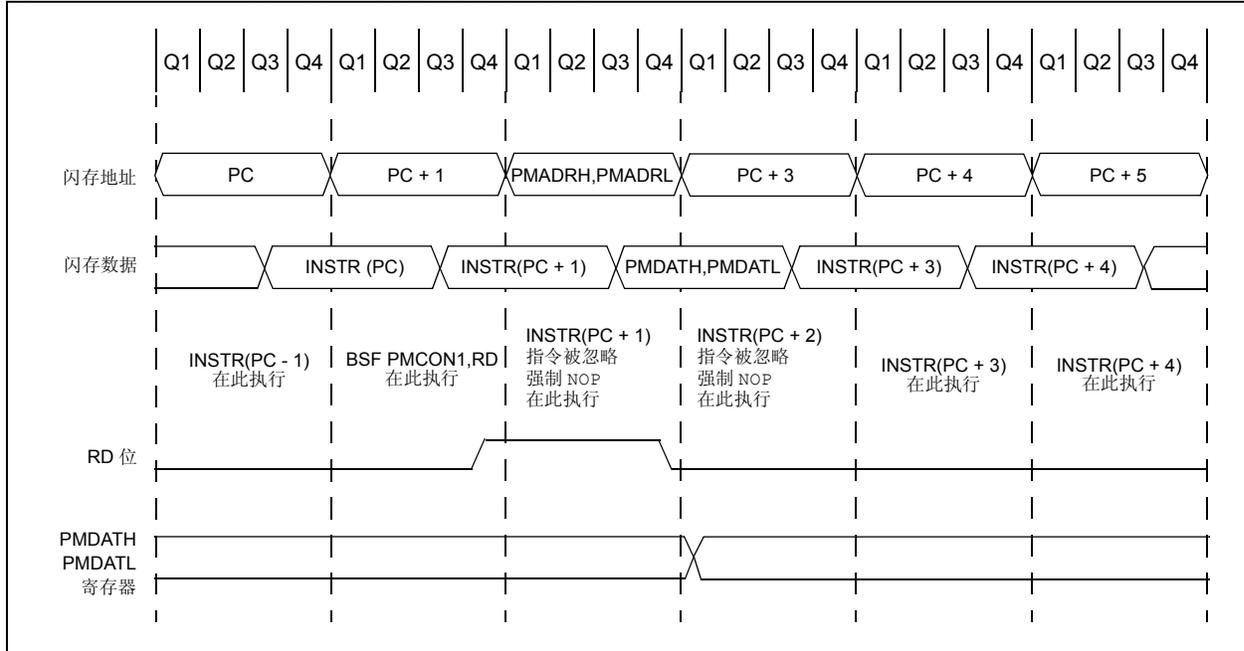


图 10-2: 闪存程序存储器读周期执行时序



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 PCON registers
  MOVLW   PROG_ADDR_LO    ;
  MOVWF   PMADRL          ; Store LSB of address
  MOVLW   PROG_ADDR_HI    ;
  MOVWF   PMADRH          ; Store MSB of address

  BCF     PMCON1,CFGSS    ; 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
    
```

PIC12(L)F1612/16(L)F1613

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

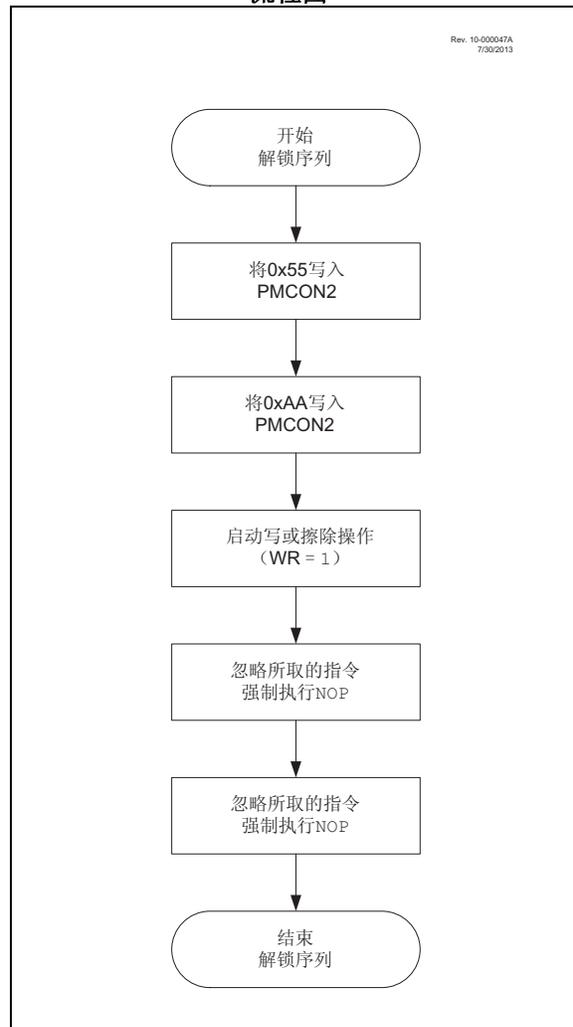
解锁序列包含以下步骤：

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寄存器装入数据时，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

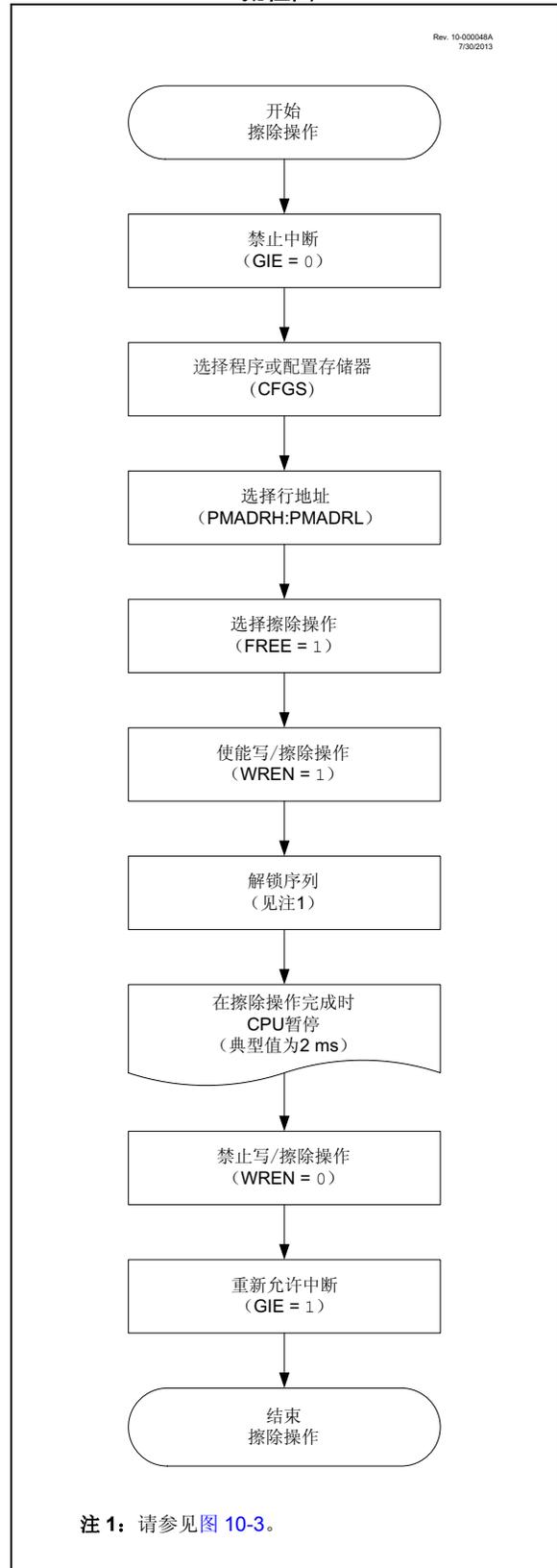
在执行代码时，程序存储器只能按行进行擦除。要擦除某行，请执行以下步骤：

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 CFGS 位清零。
3. 将 PMCON1 寄存器的 FREE 和 WREN 位置 1。
4. 向 PMCON2 中先写入 55h，然后写入 AAh（闪存编程解锁序列）。
5. 将 PMCON1 寄存器的控制位 WR 置 1，以开始擦除操作。

请参见例 10-2。

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

图 10-4: 闪存程序存储器擦除操作流程



PIC12(L)F1612/16(L)F1613

例 10-2: 擦除程序存储器的一行

```
; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. ADDRH and 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,FRE     ; 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                     ; NOP instructions are forced as processor starts
                                   ; 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
```

必需的序列

10.2.4 写入闪存程序存储器

要编程序存储器，请执行以下步骤：

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

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

程序存储器每次可以写入一个或多个字。每次可以写入的最多字数等于写锁存器的数量。更多详细信息，请参见图 10-5（使用 16 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 CFGS 位清零。
3. 将 PMCON1 寄存器的 LWLO 位置 1。当 PMCON1 寄存器的 LWLO 位为 1 时，写序列之后只向写锁存器装入数据，而不会启动对闪存程序存储器的写操作。
4. 将要写入的存储单元的地址装入 PMADRH:PMADRL 寄存器对。
5. 将要写入的程序存储器数据装入 PMDATH:PMDATL 寄存器对。
6. 执行解锁序列（第 10.2.2 节“闪存解锁序列”）。此时，将数据装入写锁存器。
7. 递增 PMADRH:PMADRL 寄存器对，使之指向下一个存储单元。
8. 重复步骤 5 至步骤 7，直到除了最后一个写锁存器之外的所有写锁存器中都装入数据为止。
9. 将 PMCON1 寄存器的 LWLO 位清零。当 PMCON1 寄存器的 LWLO 位为 0 时，写序列会启动对闪存程序存储器的写操作。
10. 将要写入的程序存储器数据装入 PMDATH:PMDATL 寄存器对。
11. 执行解锁序列（第 10.2.2 节“闪存解锁序列”）。整个程序存储器锁存器的内容现在会被写入闪存程序存储器中。

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

例 10-3 给出了一个完整写序列的示例。初始地址装入 PMADRH:PMADRL 寄存器对；数据使用间接寻址方式装入。

图 10-5: 使用 16 个写锁寄存器对闪存程序存储器进行块写操作

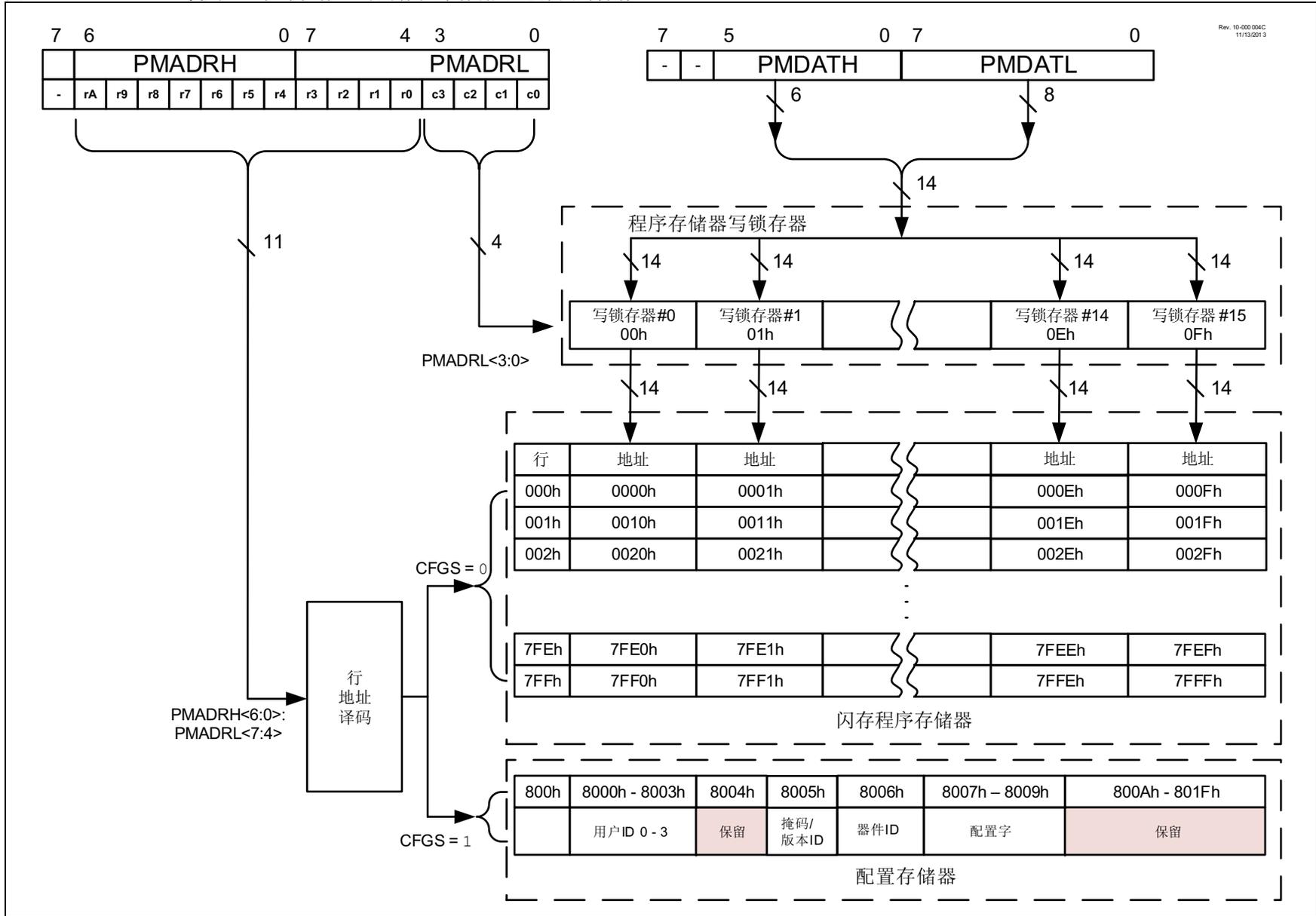
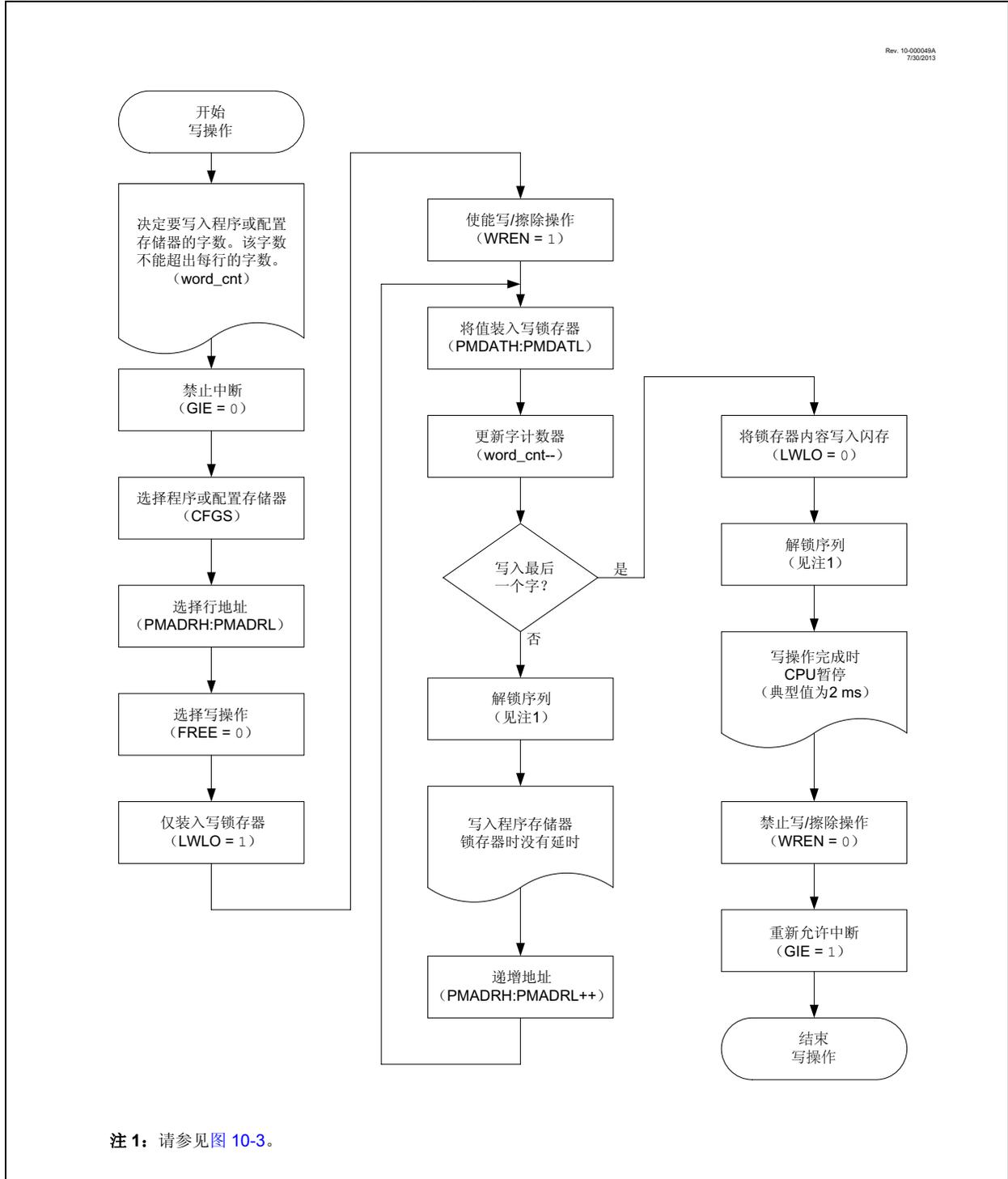


图 10-6: 闪存程序存储器写操作流程



PIC12(L)F1612/16(L)F1613

例 10-3: 写入闪存程序存储器 (16 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    ADDRHL,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,CFGSS   ; 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   PMDATH        ;
    MOVIW   FSR0++        ; Load second data byte into upper
    MOVWF   PMDATH        ;

    MOVF    PMADRL,W      ; Check if lower bits of address are '00000'
    XORLW   0x0F          ; Check if we're on the last of 16 addresses
    ANDLW   0x0F          ;
    BTFSC   STATUS,Z      ; Exit if last of 16 words,
    GOTO    START_WRITE   ;

    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
    ; loads program memory write latches
    NOP     ;

    INCF    PMADRL,F      ; Still loading latches Increment address
    GOTO    LOOP          ; Write next latches

START_WRITE
    BCF     PMCON1,LWLO    ; No more loading latches - Actually start Flash program
    ; memory write

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

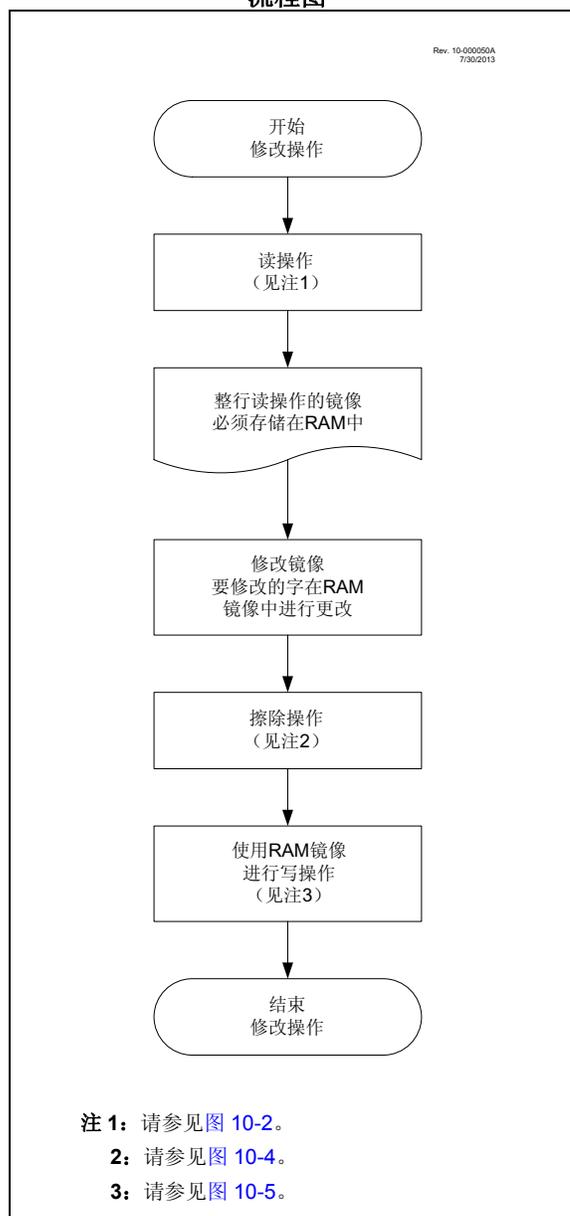
    BCF     PMCON1,WREN    ; Disable writes
    BSF     INTCON,GIE     ; Enable interrupts
```

10.3 修改闪存程序存储器

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

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

图 10-7: 闪存程序存储器修改操作流程图



PIC12(L)F1612/16(L)F1613

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

表 10-2: 用户 ID、器件 ID 和配置字访问 (CFGS = 1)

地址	功能	读访问	写访问
8000h-8003h	用户 ID	是	是
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

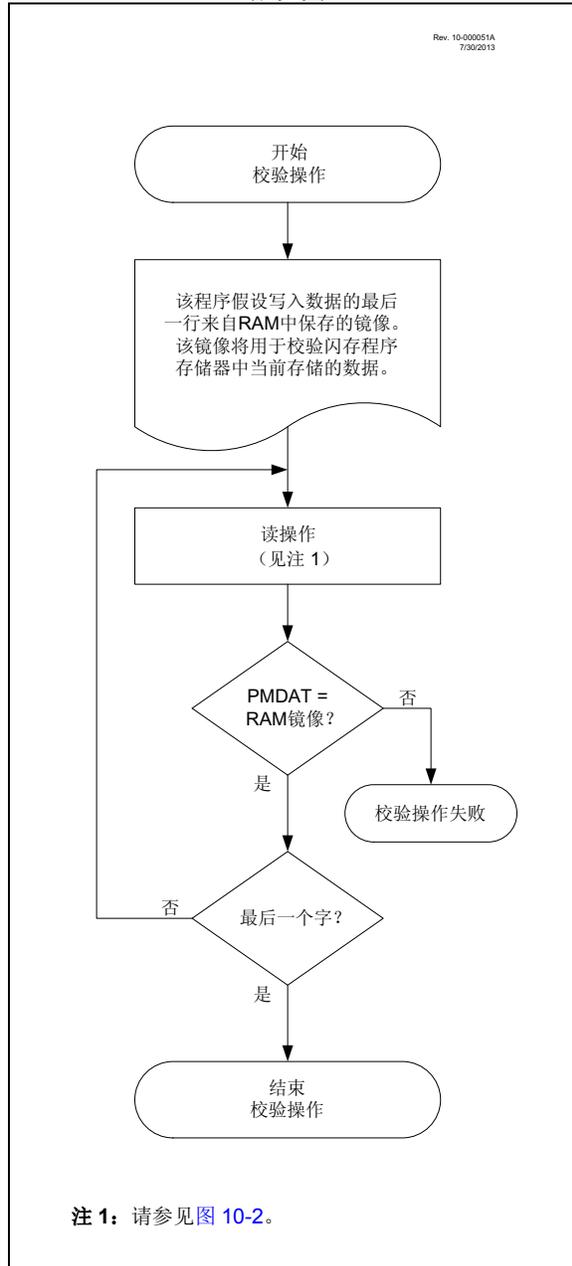
BSF      PMCON1,CFGS     ; Select Configuration Space
BCF      INTCON,GIE      ; Disable interrupts
BSF      PMCON1,RD       ; Initiate read
NOP      ; Executed (See Figure 10-2)
NOP      ; Ignored (See Figure 10-2)
BSF      INTCON,GIE      ; Restore interrupts

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



PIC12(L)F1612/16(L)F1613

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
—(1)	PMADR<14:8>						
bit 7							bit 0

图注:

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

bit 7 **未实现**: 读为 1

bit 6-0 **PMADR<14:8>**: 指定程序存储器地址的最高有效位

注 1: 未实现, 读为 1。

PIC12(L)F1612/16(L)F1613

寄存器 10-5: PMCON1: 程序存储器控制 1 寄存器

U-1	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q ⁽²⁾	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
— ⁽¹⁾	CFG5	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	未实现: 读为 1
bit 6	CFG5: 配置选择位 1 = 访问配置、用户 ID 和器件 ID 寄存器 0 = 访问闪存程序存储器
bit 5	LWLO: 仅装入写锁存器位 ⁽³⁾ 1 = 在下一条 WR 命令时装入 / 更新所寻址的程序存储器写锁存器 0 = 在下一条 WR 命令时装入 / 更新所寻址的程序存储器写锁存器, 并启动对于所有程序存储器写锁存器的写操作
bit 4	FREE: 程序闪存擦除使能位 1 = 在下一条 WR 命令时执行擦除操作 (完成后由硬件清零) 0 = 在下一条 WR 命令时执行写操作
bit 3	WRERR: 编程 / 擦除错误标志位 1 = 条件指示试图 / 终止执行不合法的编程或擦除序列 (试图将 WR 位置 1 (写入 1) 时自动将该位置 1) 0 = 编程或擦除操作正常完成
bit 2	WREN: 编程 / 擦除使能位 1 = 允许编程 / 擦除周期 0 = 禁止对程序闪存的编程 / 擦除操作
bit 1	WR: 写控制位 1 = 启动程序闪存的编程或擦除操作。 操作是自定时的, 一旦该操作完成, 该位即由硬件清零。 用软件只能将 WR 位置 1 (不能清零)。 0 = 对闪存的编程 / 擦除操作已完成并且变为无效。
bit 0	RD: 读控制位 1 = 启动程序闪存的读操作。读操作需要一个周期。RD 由硬件清零。用软件只能将 RD 位置 1 (不能清零)。 0 = 不启动程序闪存的读操作。

- 注
- 1: 未实现位, 读为 1。
 - 2: 在程序存储器写操作或擦除操作启动 (WR = 1) 时, 硬件会自动将 WRERR 位置 1。
 - 3: 在程序存储器擦除操作期间 (FREE = 1), LWLO 位会被忽略。

PIC12(L)F1612/16(L)F1613

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	78
PMCON1	—(1)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	113
PMCON2	程序存储器控制寄存器 2								114
PMADRL	PMADRL<7:0>								112
PMADRH	—(1)	PMADRH<6:0>							112
PMDATL	PMDATL<7:0>								112
PMDATH	—	—	PMDATH<5:0>						112

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	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	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	48
	7:0	CP	MCLRE	PWRTE	—	—	—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	49
	7:0	ZCDDIS	—	—	—	—	—	WRT<1:0>		
CONFIG3	13:8	—	—	WDTCCS<2:0>			WDTCWS<2:0>			50
	7:0	—	WDTE<1:0>	WDTCPSC<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
- 用于通信 CRC 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或 EEPROM 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

$x^{16} + x^{15} + x^2 + 1$ (17位)

标准16位表示 = 0x8005

CRCXORH = 0b10000000
CRCXORL = 0b0000010- (1)

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

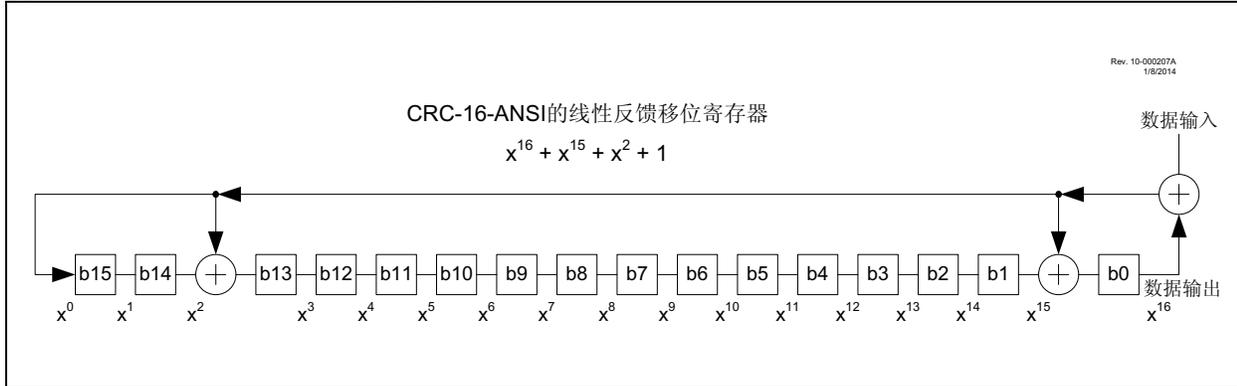
注 1: bit 0未实现。所有CRC多项式的LSb总是为1，CRC在计算CRC校验值时总是将它视为1。该位将在软件中读为0。

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，因为多项式 CRC 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。

PIC12(L)F1612/16(L)F1613

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. 使用所需的生成器多项式设定 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 CRCDATH/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 CRCDATH，因为移位器会在写入 CRCDATL 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 CRCDATH/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 以启用模块。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。

PIC12(L)F1612/16(L)F1613

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。

PIC12(L)F1612/16(L)F1613

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	如果扫描器要窥探不被执行的指令（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。	
		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。	
PC 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。	
数据断点		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。		
单步执行		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。		请注意，突发操作会被外部暂停中断。
SWBP 和 ICDINST		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。

PIC12(L)F1612/16(L)F1613

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)

PIC12(L)F1612/16(L)F1613

寄存器 11-3: CRCDAT_H: CRC 数据高字节寄存器

R/W-xx	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: CRCDAT_L: CRC 数据低字节寄存器

R/W-xx	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。

PIC12(L)F1612/16(L)F1613

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

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X<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 **X<15:8>**: 多项式的项 X_N 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	U-0						
X<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 **X<7:1>**: 多项式的项 X_N 的异或使能位
bit 0 **未实现**: 读为 0

PIC12(L)F1612/16(L)F1613

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 ⁽¹⁾	SCANGO ^(2,3)	BUSY ⁽⁴⁾	INVALID	INTM	—	MODE<1:0> ⁽⁵⁾	
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-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 ⁽¹⁾

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 ^(2,3)

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 ⁽⁴⁾

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 ⁽⁶⁾

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 ⁽⁵⁾

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。

PIC12(L)F1612/16(L)F1613

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> ^(1, 2)							
bit 7							bit 0

图注:

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

bit 7-0 **LADR<15:8>**: 扫描起始 / 当前地址位 ^(1, 2)
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> ^(1, 2)							
bit 7							bit 0

图注:

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

bit 7-0 **LADR<7:0>**: 扫描起始 / 当前地址位 ^(1, 2)
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。

PIC12(L)F1612/16(L)F1613

寄存器 11-14: SCANHADR_H: 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> ^(1, 2)							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **HADR<15:8>**: 扫描结束地址位 ^(1, 2)
所指定扫描的结束地址的高位

- 注 1: 寄存器 SCANHADR_H/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> ^(1, 2)							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **HADR<7:0>**: 扫描结束地址位 ^(1, 2)
所指定扫描的结束地址的低位

- 注 1: 寄存器 SCANHADR_H/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。

PIC12(L)F1612/16(L)F1613

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<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 7-2 **未实现:** 读为 0
bit 1-0 **TSEL<1:0>**: 扫描器数据触发输入选择位
11 = TMR6_postscaled
10 = TMR4_postscaled
01 = TMR2_postscaled
00 = 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>								121
CRCACCL	ACC<7:0>								121
CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	120
CRCCON1	DLEN<3:0>				PLEN<3:0>				120
CRCDATH	DATA<15:8>								121
CRCDATL	DATA<7:0>								121
CRCSHIFTH	SHIFT<15:8>								122
CRCSHIFTL	SHIFT<7:0>								122
CRCXORH	X<15:8>								122
CRCXORL	X<7:1>								122
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
PIR4	SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF	86
PIE4	SCANIE	CRCIE	SMT2PWAIE	SMT2PRAIE	SMT2IE	SMT1PWAIE	SMT1PRAIE	SMT1IE	82
SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		123
SCANHADRH	HADR<15:8>								125
SCANHADRL	HADR<7:0>								125
SCANLADRH	LADR<15:8>								124
SCANLADRL	LADR<7:0>								124
SCANTRIG	—	—	—	—	—	—	TSEL<1:0>		126

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

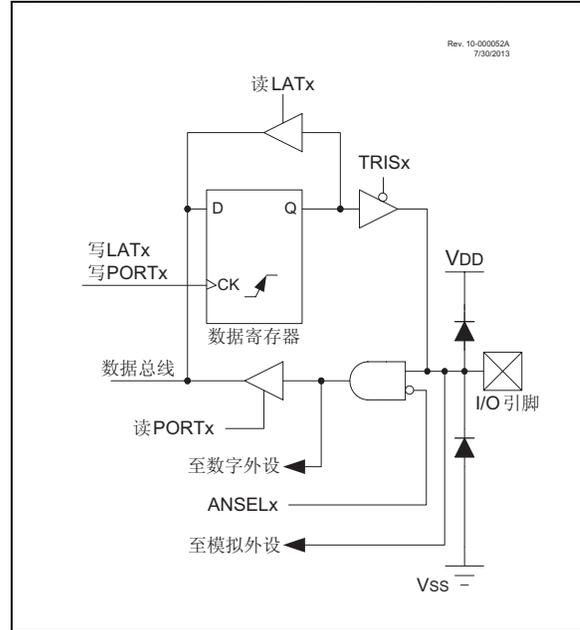
器件	PORTA	PORTC
PIC16(L)F1613	•	•
PIC12(L)F1612	•	

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



PIC12(L)F1612/16(L)F1613

12.1 备用引脚功能

备用引脚功能控制 (APFCON) 寄存器用于将特定的外设输入和输出功能配置到不同的引脚上。APFCON 寄存器如寄存器 12-1 所示。对于本器件系列, 以下功能可以配置到不同的引脚上。

- CWGA
- CWGB
- T1G
- CCP1
- CCP2

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

12.2 寄存器定义: 备用引脚功能控制

寄存器 12-1: APFCON: 备用引脚功能控制寄存器

U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
—	CWGASEL ⁽¹⁾	CWGBSEL ⁽¹⁾	—	T1GSEL	—	CCP2SEL ⁽²⁾	CCP1SEL ⁽¹⁾
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7 **未实现:** 读为 0
- bit 6 **CWGASEL:** 引脚选择位⁽¹⁾
1 = RA5 上具有 CWGA 功能
0 = RA2 上具有 CWGA 功能
- bit 5 **CWGBSEL:** 引脚选择位⁽¹⁾
1 = RA4 上具有 CWGB 功能
0 = RA0 上具有 CWGB 功能
- bit 4 **未实现:** 读为 0
- bit 3 **T1GSEL:** 引脚选择位
1 = RA3 上具有 T1G 功能
0 = RA4 上具有 T1G 功能
- bit 2 **未实现:** 读为 0
- bit 1 **CCP2SEL:** 引脚选择位⁽²⁾
1 = RA5 上具有 CCP2 功能
0 = RC3 上具有 CCP2 功能
- bit 0 **CCP1SEL:** 引脚选择位⁽¹⁾
1 = RA5 上具有 CCP1 功能
0 = RA2 上具有 CCP1 功能

注 1: 仅限 PIC12(L)F1612。

注 2: 仅限 PIC16(L)F1613。

12.3 PORTA 寄存器

12.3.1 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 12-3）。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，它仅可作为输入引脚，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 12-2）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）。

12.3.2 方向控制

TRISA 寄存器（寄存器 12-3）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。

12.3.3 漏极开路控制

ODCONA 寄存器（寄存器 12-7）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。

12.3.4 压摆率控制

SLRCONA 寄存器（寄存器 12-8）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。

12.3.5 输入阈值控制

INLVLA 寄存器（寄存器 12-9）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。输入阈值对于确定 PORTA 寄存器的读取值很重要，同时它也是发生电平变化中断的临界电压（如果使能该功能）。关于阈值电压的更多信息，请参见第 28.3 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

ANSELA 寄存器（寄存器 12-5）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
```

PIC12(L)F1612/16(L)F1613

12.3.7 PORTA 功能和输出优先级

每个 PORTA 引脚都与其他功能复用。表 12-2 列出了引脚及其复用功能和输出优先级。

当使能多个输出时，实际引脚控制权将属于优先级最高的外设。

优先级列表中未列出模拟输入功能，例如 ADC 和比较器输入。这些输入在使用 ANSELx 寄存器将 I/O 引脚设置为模拟模式时有效。当引脚处于模拟模式时，数字输出功能可以按照下面的表 12-2 中列出的优先级控制引脚。

**表 12-2: PORTA 输出优先级
(仅限 PIC12(L)F1612)**

引脚名称	功能优先级 ⁽¹⁾
RA0	DAC1OUT1 CWG1B ⁽²⁾ CCP2 RA0
RA1	ZCD1OUT RA1
RA2	CWG1A ⁽²⁾ C1OUT CCP1 RA2 ⁽²⁾
RA3	RA3
RA4	CLKOUT CWG1B ⁽³⁾ RA4
RA5	CWG1A ⁽³⁾ CCP1 ⁽³⁾ RA5

- 注 1: 优先级按从最高到最低排列。
2: 默认引脚 (见 APFCON 寄存器)。
3: 备用引脚 (见 APFCON 寄存器)。

**表 12-3: PORTA 输出优先级
(仅限 PIC16(L)F1613)**

引脚名称	功能优先级 ⁽¹⁾
RA0	DAC1OUT1 RA0
RA1	ZCD1OUT RA1
RA2	C1OUT RA2 ⁽²⁾
RA3	RA3
RA4	CLKOUT RA4
RA5	CCP2 ⁽³⁾ RA5

- 注 1: 优先级按从最高到最低排列。
2: 默认引脚 (见 APFCON 寄存器)。
3: 备用引脚 (见 APFCON 寄存器)。

PIC12(L)F1612/16(L)F1613

12.4 寄存器定义: PORTA

寄存器 12-2: 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⁽¹⁾

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 12-3: TRISA: PORTA 三态寄存器

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

图注:

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

PIC12(L)F1612/16(L)F1613

寄存器 12-4: LATA: PORTA 数据锁存器寄存器

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

图注:

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

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

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

寄存器 12-5: 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 ⁽¹⁾。数字输入缓冲器被禁止。
 0 = 数字 I/O。引脚被配置为端口或数字特殊功能。
bit 3 未实现: 读为 0
bit 2-0 **ANSA<2:0>**: 将 RA<2:0> 引脚选择为模拟或数字功能
 1 = 模拟输入。引脚被配置为模拟输入 ⁽¹⁾。数字输入缓冲器被禁止。
 0 = 数字 I/O。引脚被配置为端口或数字特殊功能。

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

PIC12(L)F1612/16(L)F1613

寄存器 12-6: 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 (3)
 1 = 使能上拉
 0 = 禁止上拉

- 注 1:** 必须清零 OPTION_REG 寄存器的全局 $\overline{\text{WPUEN}}$ 位, 从而使能各个上拉功能。
注 2: 如果引脚被配置为输出, 则自动禁止弱上拉器件。
注 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 12-7: 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)

PIC12(L)F1612/16(L)F1613

寄存器 12-8: 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 12-9: 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

PIC12(L)F1612/16(L)F1613

表 12-4: 与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	132
APFCON	—	CWGASEL ⁽²⁾	CWGBSEL ⁽²⁾	—	T1GSEL	—	CCP2SEL ⁽³⁾	CCP1SEL ⁽²⁾	128
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	134
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	132
ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	133
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			183
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	131
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	134
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	131
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	133

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。
 2: 仅限 PIC12(L)F1612。
 3: 仅限 PIC16(L)F1613。

表 12-5: 与 PORTA 相关的配置字汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		—	48
	7:0	CP	MCLRE	PWRTE	—	—	—	FOSC<1:0>		

图注: — = 未实现位, 读为 0。PORTA 不使用阴影单元。

PIC12(L)F1612/16(L)F1613

12.5 PORTC 寄存器 (仅限 PIC16(L)F1613)

12.5.1 数据寄存器

PORTC 是一个 6 位宽的双向端口。对应的数据方向寄存器是 TRISC (寄存器 12-11)。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 12-10) 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 12-11) 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 12-14) 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 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 12-15) 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 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 12-16) 用于控制每个可用 PORTC 输入引脚的输入阈值电压。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。输入阈值对于确定 PORTC 寄存器的读取值很重要, 同时它也是发生电平变化中断的临界电压 (如果使能该功能)。关于阈值电压的更多信息, 请参见第 28.3 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 12-13) 用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSEL 位设置为高电平将使引脚上的所有数字读操作都读为 0, 并允许引脚上的模拟功能正常工作。

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

注: 在发生复位之后, ANSEL 位默认为模拟模式。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 PORTC 功能和输出优先级

每个 PORTC 引脚都与其他功能复用。表 12-6 列出了引脚及其复用功能和输出优先级。

当使能多个输出时，实际引脚控制权将属于优先级最高的外设。

输出优先级列表中未包含模拟输入和一些数字输入功能。这些输入功能会在引脚配置为输出时保持有效。一些数字输入功能会改写一些其他端口功能，输出优先级列表中包含了这些功能。

表 12-6: PORTC 输出优先级

引脚名称	功能优先级 ⁽¹⁾
RC0	RC0
RC1	RC1
RC2	CWG1D RC2
RC3	CWG1C CCP2 ⁽²⁾ RC3
RC4	CWG1B C2OUT RC4
RC5	CWG1A CCP1 RC5

- 注 1: 优先级按从最高到最低排列。
注 2: 默认引脚（见 APFCON 寄存器）。

PIC12(L)F1612/16(L)F1613

12.6 寄存器定义: PORTC (仅限 PIC16(L)F1613)

寄存器 12-10: PORTC: PORTC 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
—	—	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

图注:

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

bit 7-6 未实现: 读为 0
bit 5-0 **RC<5:0>**: PORTC 通用 I/O 引脚位
1 = 端口引脚电平 $\geq V_{IH}$
0 = 端口引脚电平 $\leq V_{IL}$

寄存器 12-11: TRISC: PORTC 三态寄存器

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
—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

图注:

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

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

寄存器 12-12: LATC: PORTC 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
—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
bit 7							bit 0

图注:

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

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

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

PIC12(L)F1612/16(L)F1613

寄存器 12-13: ANSELC: PORTC 模拟选择寄存器

U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	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 7-4 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。引脚被配置为模拟输入 ⁽¹⁾。数字输入缓冲器被禁止。
 0 = 数字 I/O。引脚被配置为端口或数字特殊功能。

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

寄存器 12-14: ODCONC: PORTC 漏极开路控制寄存器

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
—	—	ODC5	ODC4	ODC4	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 7-6 未实现: 读为 0

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

PIC12(L)F1612/16(L)F1613

寄存器 12-15: SLRCONC: PORTC 压摆率控制寄存器

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
bit 7							bit 0

图注:

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

bit 7-6 **未实现:** 读为 0
bit 5-0 **SLRC<5:0>:** PORTC 压摆率使能位
对于各个 RC<5:0> 引脚
1 = 端口引脚的压摆率受到限制
0 = 端口引脚的压摆率将为最大值

寄存器 12-16: INLVLC: PORTC 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
—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0
bit 7							bit 0

图注:

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

bit 7-6 **未实现:** 读为 0
bit 5-0 **INLVLC<5:0>:** PORTC 输入电平选择位
对于各个 RC<5:0> 引脚
1 = 对于端口读操作和电平变化中断, 使用 ST 输入
0 = 对于端口读操作和电平变化中断, 使用 TTL 输入

表 12-7: 与 PORTC 相关的寄存器汇总 (2)

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	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	139
APFCON	—	CWGASEL ⁽¹⁾	CWGBSEL ⁽¹⁾	—	T1GSEL	—	CCP2SEL ⁽²⁾	CCP1SEL ⁽¹⁾	128
INLVLC	—	—	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	140
LATC	—	—	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	138
ODCONC	—	—	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	139
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			183
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	138
SLRCONC	—	—	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	140
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	138
WPUC	—	—	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	133

图注: x = 未知, u = 不变, - = 未实现位, 读为 0。PORTC 不使用阴影单元。
注 1: 仅限 PIC12(L)F1612。
2: 仅限 PIC16(L)F1613。

13.0 电平变化中断

PORTA 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 13-1 给出了 IOC 模块的框图。

13.1 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。

13.2 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。

13.3 中断标志

分别位于 IOCAF 和 IOCCF 寄存器中的 IOCAFx 和 IOCCFxF 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 IOCAFx 和 IOCCFxF 位的状态。

13.4 清零中断标志

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

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

例 13-1: 清零中断标志 (以 PORTA 为例)

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

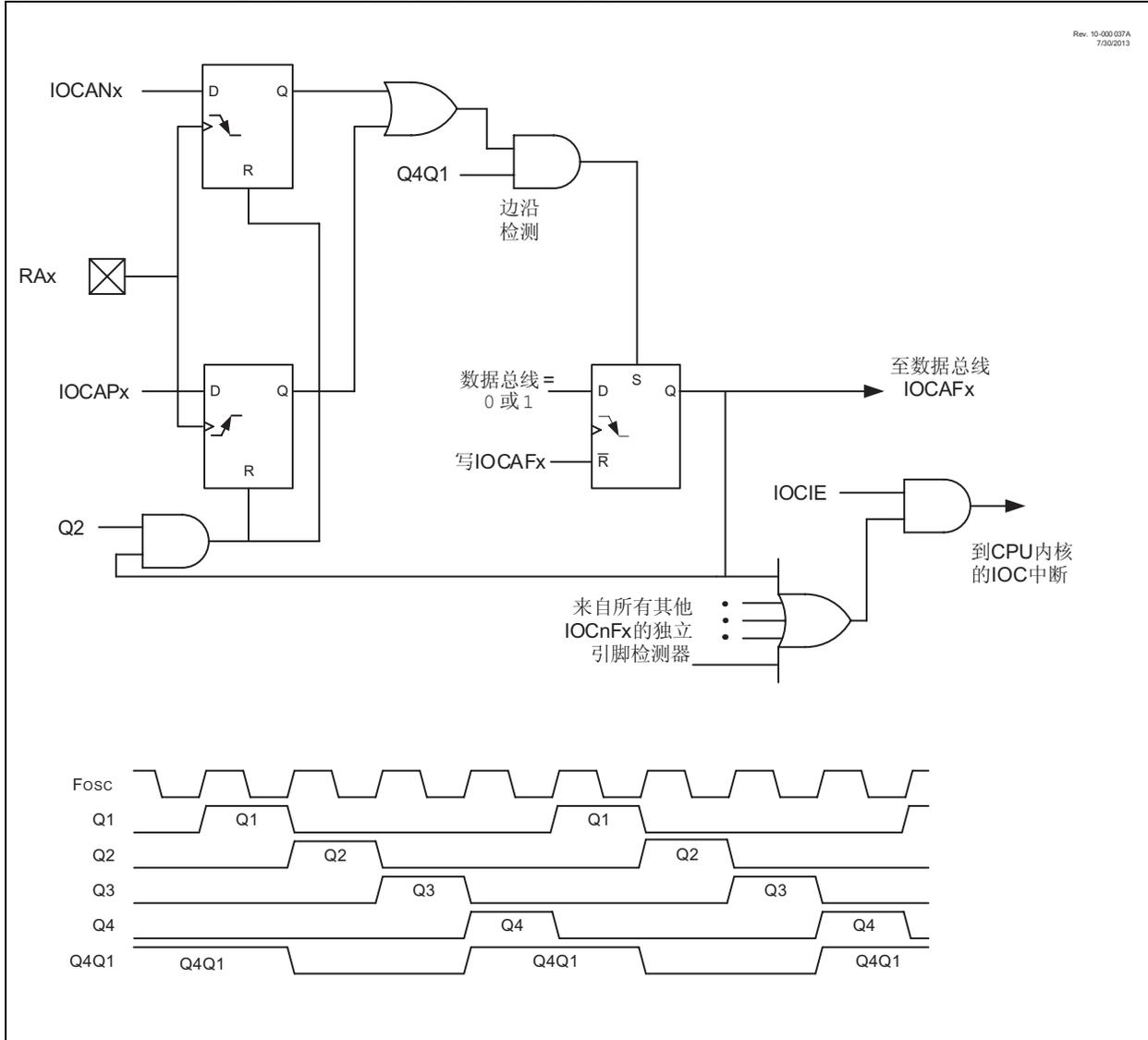
13.5 休眠模式下的操作

如果 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。

PIC12(L)F1612/16(L)F1613

图 13-1: 电平变化中断框图 (以 PORTA 为例)



PIC12(L)F1612/16(L)F1613

13.6 寄存器定义：电平变化中断控制

寄存器 13-1: IOCAP: 电平变化中断 PORTA 正边沿寄存器

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

图注:

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

bit 7-6 **未实现:** 读为 0

bit 5-0 **IOCAP<5:0>:** 电平变化中断 PORTA 正边沿使能位
1 = 在引脚上对于正向边沿允许电平变化中断。 IOCAF_x 位和 IOCIF 标志将在检测到边沿时置 1。
0 = 禁止关联引脚的电平变化中断。

寄存器 13-2: IOCAN: 电平变化中断 PORTA 负边沿寄存器

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

图注:

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

bit 7-6 **未实现:** 读为 0

bit 5-0 **IOCAN<5:0>:** 电平变化中断 PORTA 负边沿使能位
1 = 在引脚上对于负向边沿允许电平变化中断。 IOCAF_x 位和 IOCIF 标志将在检测到边沿时置 1。
0 = 禁止关联引脚的电平变化中断。

寄存器 13-3: IOCAF: 电平变化中断 PORTA 标志寄存器

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

图注:

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

bit 7-6 **未实现:** 读为 0

bit 5-0 **IOCAF<5:0>:** 电平变化中断 PORTA 标志位
1 = 在关联引脚上检测到使能的电平变化。
在 IOCAP_x = 1, 并在 RA_x 上检测到上升沿时置 1, 或者在 IOCAN_x = 1, 并在 RA_x 上检测到下降沿时置 1。
0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

PIC12(L)F1612/16(L)F1613

寄存器 13-4: IOCCP: 电平变化中断 PORTC 正边沿寄存器 (1)

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
—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
bit 7							bit 0

图注:

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

bit 7-6 **未实现:** 读为 0
bit 5-0 **IOCCP<5:0>:** 电平变化中断 PORTC 正边沿使能位
1 = 在引脚上对于正向边沿允许电平变化中断。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: 仅限 PIC16(L)F1613。

寄存器 13-5: IOCCN: 电平变化中断 PORTC 负边沿寄存器 (1)

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
—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
bit 7							bit 0

图注:

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

bit 7-6 **未实现:** 读为 0
bit 5-0 **IOCCN<5:0>:** 电平变化中断 PORTC 负边沿使能位
1 = 在引脚上对于负向边沿允许电平变化中断。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: 仅限 PIC16(L)F1613。

寄存器 13-6: IOCCF: 电平变化中断 PORTC 标志寄存器 (1)

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
—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
bit 7							bit 0

图注:

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

bit 7-6 **未实现:** 读为 0
bit 5-0 **IOCCF<5:0>:** 电平变化中断 PORTC 标志位
1 = 在关联引脚上检测到使能的电平变化。
在 IOCCPx = 1, 并在 RCx 上检测到上升沿时置 1, 或者在 IOCCNx = 1, 并在 RCx 上检测到下降沿时置 1。
0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

注 1: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

表 13-1: 与电平变化中断相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	132
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	143
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	143
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	143
IOCCF ⁽²⁾	—	—	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	144
IOCCN ⁽²⁾	—	—	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	144
IOCCP ⁽²⁾	—	—	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	144
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	131
TRISC ⁽²⁾	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	138

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。

注 2: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

14.0 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 (VFVR) 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
- 比较器的正输入
- 比较器的负输入

FVR 可以通过将 FVRCON 寄存器的 FVREN 位置 1 来使能。

14.1 独立的增益放大器

送到外设 (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 16.0 节“模数转换器 (ADC) 模块”。

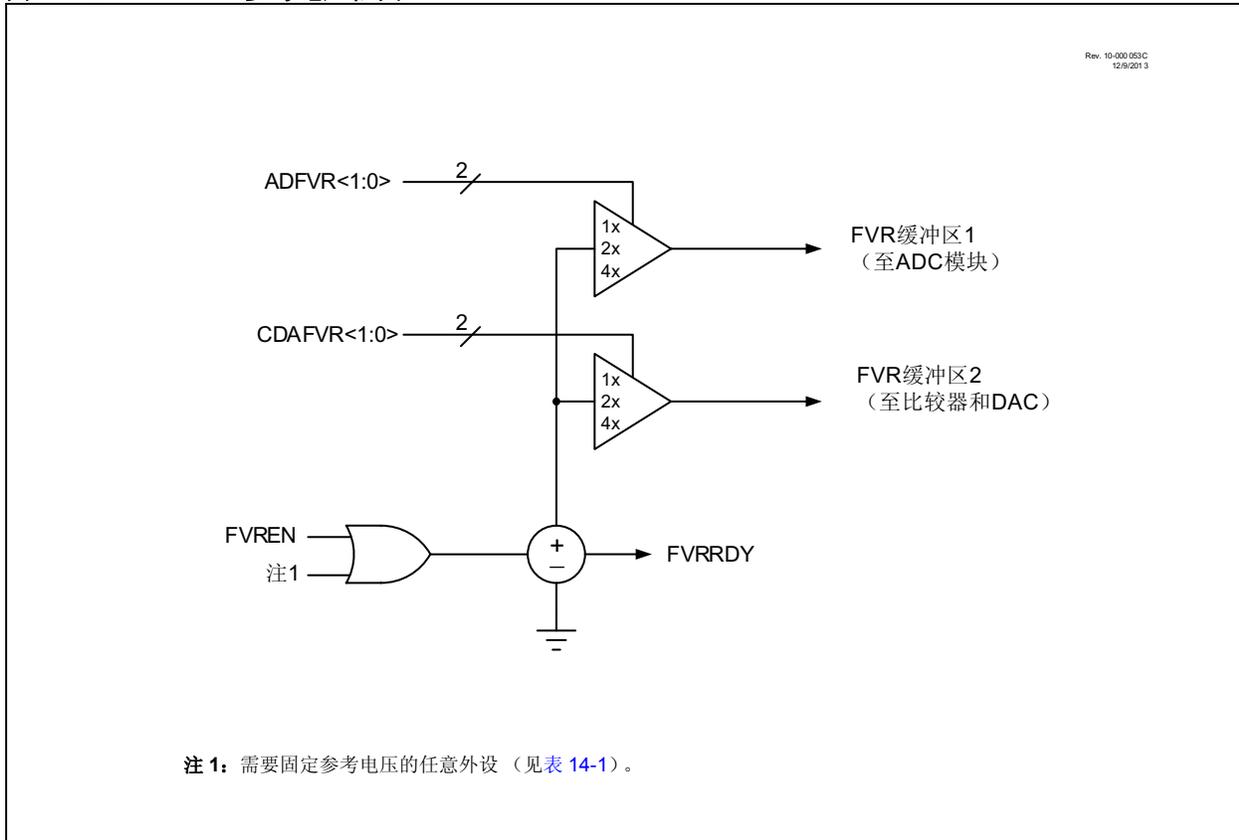
FVRCON 寄存器的 CDAFVR<1:0> 位用于使能和配置送到比较器模块的参考电压的增益放大器设置。更多信息，请参见第 18.0 节“比较器模块”。

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。

14.2 FVR 稳定周期

当固定参考电压模块使能时，参考电压和放大电路需要一段时间才能达到稳定。在电路稳定下来、可供使用时，FVRCON 寄存器的 FVRRDY 位将会置 1。请参见图 29-19。

图 14-1: 参考电压框图



PIC12(L)F1612/16(L)F1613

表 14-1: 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 PIC12F1612/16F1613 器件	处于休眠模式时, 器件会运行低功耗稳压器。

PIC12(L)F1612/16(L)F1613

14.3 寄存器定义：FVR 控制

寄存器 14-1: **FVRCON**: 固定参考电压控制寄存器

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN ⁽¹⁾	FVRRDY ⁽²⁾	TSEN ⁽³⁾	TSRNG ⁽³⁾	CDAFVR<1:0> ⁽¹⁾		ADFVR<1:0> ⁽¹⁾	
bit 7						bit 0	

图注:

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

- bit 7 **FVREN**: 固定参考电压使能位 ⁽¹⁾
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 ⁽²⁾
1 = 固定参考电压输出就绪备用
0 = 固定参考电压输出未就绪或未使能
- bit 5 **TSEN**: 温度指示器使能位 ⁽³⁾
1 = 使能温度指示器
0 = 禁止温度指示器
- bit 4 **TSRNG**: 温度指示器范围选择位 ⁽³⁾
1 = $V_{OUT} = V_{DD} - 4V_T$ (高电压范围)
0 = $V_{OUT} = V_{DD} - 2V_T$ (低电压范围)
- bit 3-2 **CDAFVR<1:0>**: 比较器 FVR 缓冲器增益选择位 ⁽¹⁾
11 = 比较器 FVR 缓冲器增益为 4x, 输出 $V_{CDAFVR} = 4x V_{FVR}$ ⁽⁴⁾
10 = 比较器 FVR 缓冲器增益为 2x, 输出 $V_{CDAFVR} = 2x V_{FVR}$ ⁽⁴⁾
01 = 比较器 FVR 缓冲器增益为 1x, 输出 $V_{CDAFVR} = 1x V_{FVR}$
00 = 比较器 FVR 缓冲器关闭
- bit 1-0 **ADFVR<1:0>**: ADC FVR 缓冲器增益选择位 ⁽¹⁾
11 = ADC FVR 缓冲器增益为 4x, 输出 $V_{ADFVR} = 4x V_{FVR}$ ⁽⁴⁾
10 = ADC FVR 缓冲器增益为 2x, 输出 $V_{ADFVR} = 2x V_{FVR}$ ⁽⁴⁾
01 = ADC FVR 缓冲器增益为 1x, 输出 $V_{ADFVR} = 1x V_{FVR}$
00 = ADC FVR 缓冲器关闭

- 注
- 1: 要最大程度降低禁止 FVR 时的电流消耗, 应通过清零缓冲器增益选择位来关闭 FVR 缓冲器。
 - 2: 对于 PIC12F1612/16F1613 器件, FVRRDY 总是为 1。
 - 3: 更多信息, 请参见第 15.0 节“温度指示器模块”。
 - 4: 固定参考电压输出不能超出 V_{DD} 。

表 14-2: 与固定参考电压相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR>1:0>		ADFVR<1:0>		148

图注: 固定参考电压模块不使用阴影单元。

15.0 温度指示器模块

本器件系列配备了用于测量硅裸片工作温度的温度电路。电路的工作温度量程介于 -40°C 和 $+85^{\circ}\text{C}$ 之间。其输出是与器件温度成正比的电压。温度指示器的输出在内部与器件 ADC 连接。

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

15.1 电路工作原理

图 15-1 给出了温度电路的简化框图。与温度成正比的电压输出通过测量多个硅结的正向压降而得到。

公式 15-1 描述了温度指示器的输出特性。

公式 15-1: V_{OUT} 范围

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

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

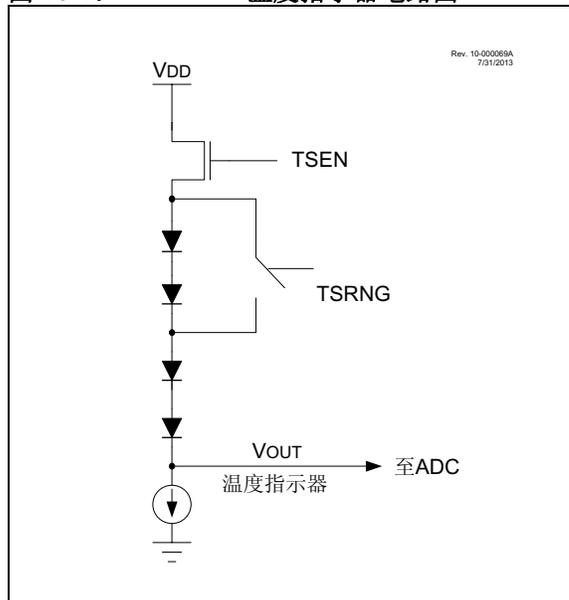
温度检测电路集成了固定参考电压 (FVR) 模块。更多信息，请参见第 14.0 节“固定参考电压 (FVR)”。

可以通过将 FVRCON 寄存器的 TSEN 位置 1 来使能该电路。在禁止时，电路不会消耗任何电流。

电路可以工作于高电压范围或低电压范围。高电压范围的选择方式是将 FVRCON 寄存器的 TSRNG 位置 1，它可提供较宽的输出电压。这可以在整个温度量程中提供更高的分辨率，但各部分之间的一致性较低。该电压范围需要较高的偏置电压才能工作，所以需要较高的 V_{DD} 。

低电压范围的选择方式是将 FVRCON 寄存器的 TSRNG 位清零。低电压范围产生的压降较小，所以只需较低的偏置电压就可以让电路工作。低电压范围旨在用于进行低电压操作。

图 15-1: 温度指示器电路图



15.2 最小工作电压 V_{DD}

当温度电路工作于低电压范围时，器件可以在规范范围内的任意工作电压下工作。

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

表 15-1 给出了建议的最小 V_{DD} 与范围设置。

表 15-1: 建议的 V_{DD} 与范围

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

15.3 温度输出

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

15.4 ADC 采集时间

为了确保精确的温度测量，用户必须在 ADC 输入多路开关连接到温度指示器输出之后至少等待 $200\ \mu\text{s}$ ，然后再执行转换。此外，用户必须在温度指示器输出的连续两次转换之间等待 $200\ \mu\text{s}$ 。

PIC12(L)F1612/16(L)F1613

表 15-2: 与温度指示器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		148

图注: 温度指示器模块不使用阴影单元。

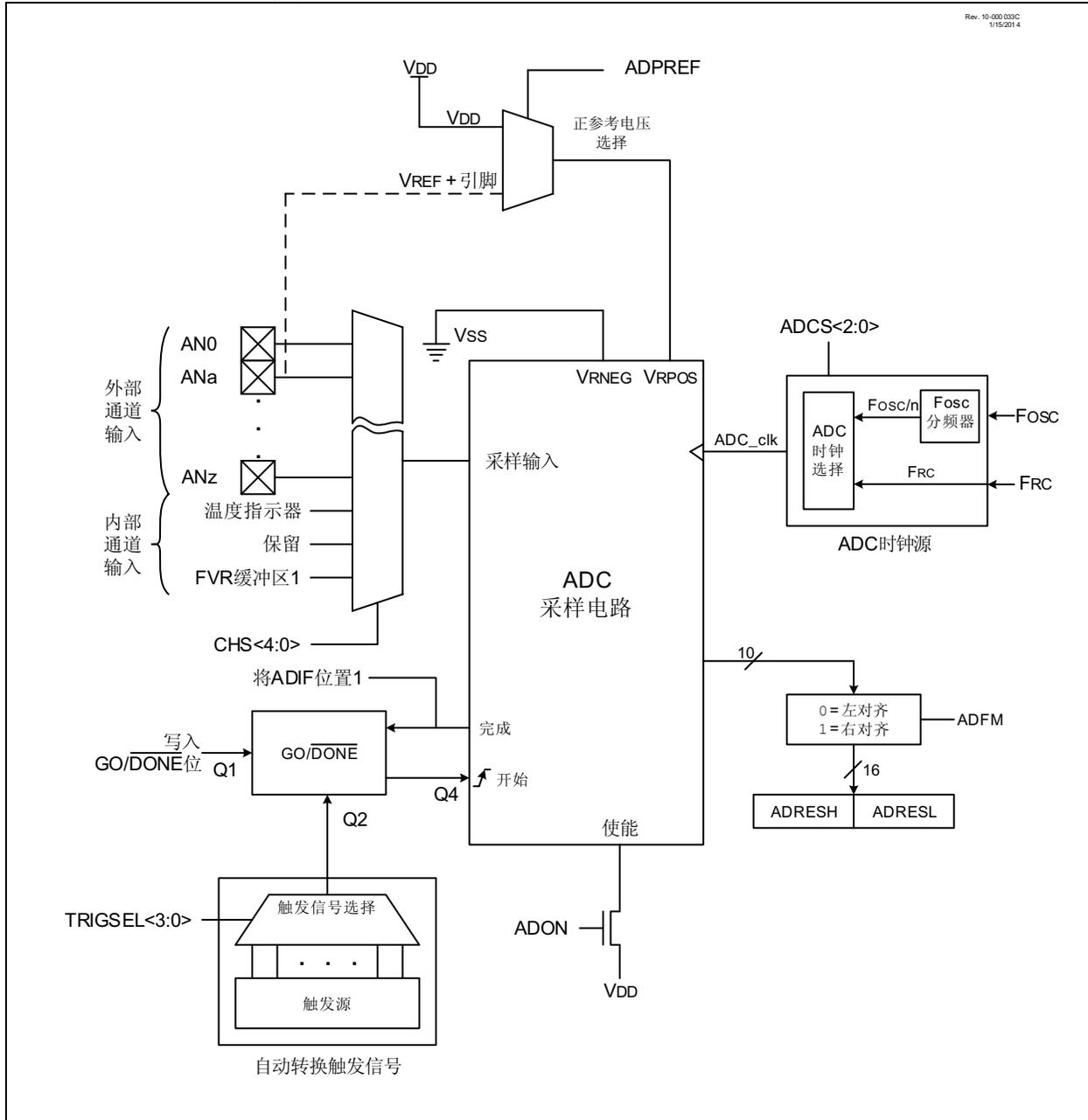
16.0 模数转换器 (ADC) 模块

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

可通过软件方式选择内部产生的电压或外部提供的电压作为 ADC 参考电压。

ADC 可在转换完成时产生中断。该中断可用于将器件从休眠状态唤醒。

图 16-1: ADC 框图



PIC12(L)F1612/16(L)F1613

16.1 ADC 配置

配置和使用 ADC 时必须考虑以下功能：

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

16.1.1 端口配置

ADC 可用于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。

16.1.2 通道选择

有最多 11 个通道选择可供使用：

- AN<7:0> 引脚（仅限 PIC16(L)F1613）
- AN<3:0> 引脚（仅限 PIC12(L)F1612）
- 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 16.2.6 节“[ADC 转换步骤](#)”。

16.1.3 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

16.1.4 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 16-2 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 28.0 节“[电气规范](#)”中的 ADC 转换要求。表 16-1 给出了适当的 ADC 时钟选择的示例。

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

PIC12(L)F1612/16(L)F1613

表 16-1: ADC 时钟周期 (TAD) 与器件工作频率关系表

ADC 时钟周期 (TAD)		器件频率 (Fosc)				
ADC 时钟源	ADCS<2:0>	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

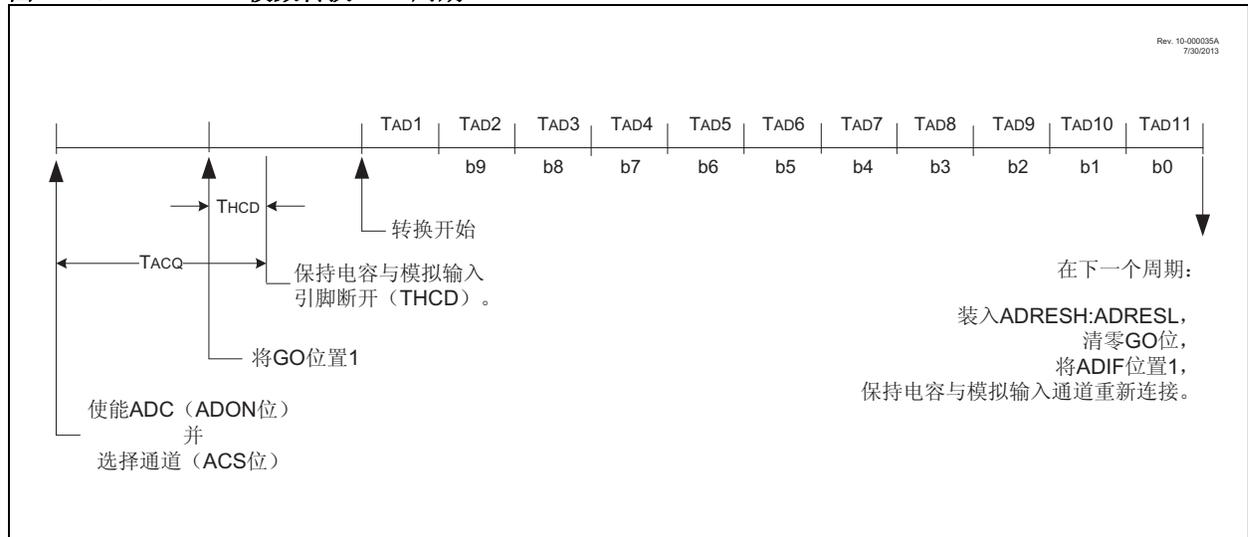
图注: 阴影单元表示超出了建议范围。

注 1: FRC 时钟源的典型 TAD 时间为 1.7 μs。

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 16-2: 模数转换 TAD 周期



PIC12(L)F1612/16(L)F1613

16.1.5 中断

ADC 模块可在模数转换完成时产生中断。ADC 中断标志位是 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。

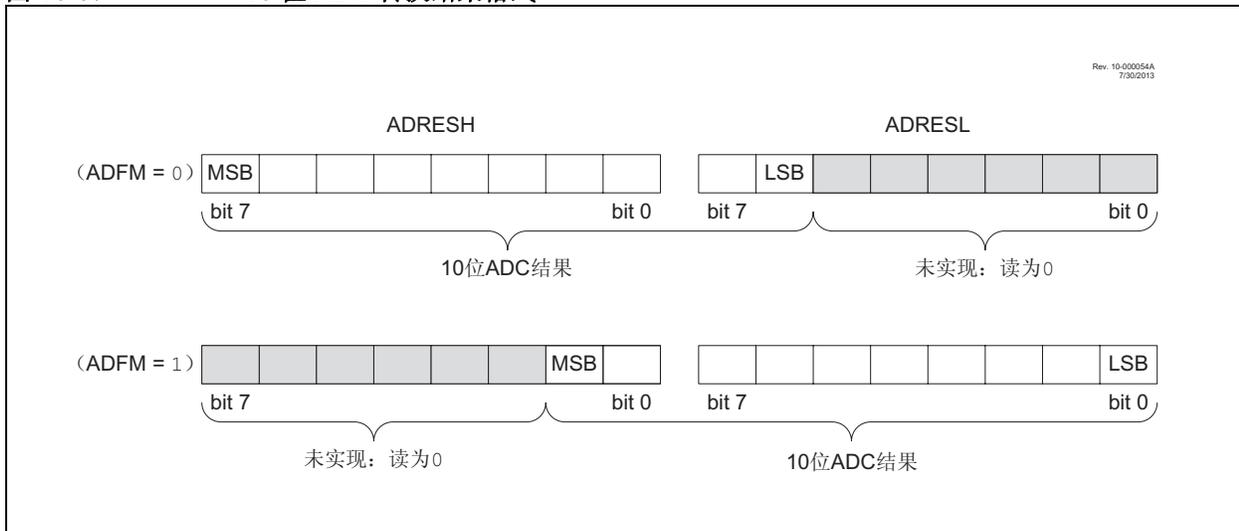
器件工作或休眠时都可产生该中断。如果器件处于休眠状态，该中断会唤醒器件。从休眠状态唤醒时，总是执行紧跟 SLEEP 指令后的下一条指令。如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止 INTCON 寄存器的 GIE 和 PEIE 位。如果使能了 INTCON 寄存器的 GIE 和 PEIE 位，执行将切换到中断服务程序。

16.1.6 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 16-3 给出了两种输出格式。

图 16-3: 10 位 ADC 转换结果格式



16.2 ADC 工作原理

16.2.1 启动转换

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

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

16.2.2 转换完成

转换完成时，ADC 模块将：

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

16.2.3 终止转换

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

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

16.2.4 休眠期间的 ADC 操作

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

16.2.5 自动转换触发器

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

自动转换触发源使用 ADCON2 寄存器的 TRIGSEL<3:0> 位进行选择。

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

关于自动转换源，请参见表 16-2。

表 16-2: 自动转换源

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 ⁽¹⁾	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

注 1: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

16.2.6 ADC 转换步骤

以下是用 ADC 执行模数转换的示例步骤：

1. 配置端口：
 - 禁止引脚输出驱动器（见 TRIS 寄存器）
 - 将引脚配置为模拟功能（见 ANSEL 寄存器）
2. 配置 ADC 模块：
 - 选择 ADC 转换时钟
 - 配置参考电压
 - 选择 ADC 输入通道
 - 开启 ADC 模块
3. 配置 ADC 中断（可选）：
 - 清零 ADC 中断标志
 - 允许 ADC 中断
 - 允许外设中断
 - 允许全局中断⁽¹⁾
4. 等待所需采集时间⁽²⁾。
5. 通过将 $\overline{\text{GO/DONE}}$ 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 $\overline{\text{GO/DONE}}$ 位
 - 等待 ADC 中断（已允许中断）
7. 读取 ADC 结果。
8. 清零 ADC 中断标志（如果已允许中断则需要此操作）。

例 16-1: ADC 转换

```
;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL    ADCON1        ;
MOVLW     B'11110000'    ;Right justify, FRC
                                ;oscillator
MOVWF     ADCON1        ;Vdd and Vss Vref+
BANKSEL    TRISA         ;
BSF       TRISA,0       ;Set RA0 to input
BANKSEL    ANSEL        ;
BSF       ANSEL,0       ;Set RA0 to analog
BANKSEL    ADCON0       ;
MOVLW     B'00000001'    ;Select channel AN0
MOVWF     ADCON0        ;Turn ADC On
CALL     SampleTime     ;Acquisition delay
BSF      ADCON0,ADGO    ;Start conversion
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
```

- 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 16.4 节“ADC 采集要求”。

PIC12(L)F1612/16(L)F1613

16.3 寄存器定义：ADC 控制

寄存器 16-1: **ADCON0: ADC 控制寄存器 0**

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CHS<4:0>					GO/DONE	ADON
bit 7							bit 0

图注:

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

bit 7 **未实现:** 读为 0

bit 6-2 **CHS<4:0>:** 模拟通道选择位
 11111 = FVR (固定参考电压) 缓冲区 1 输出 ⁽³⁾
 11110 = DAC (数模转换器) ⁽²⁾
 11101 = 温度指示器 ⁽¹⁾
 11100 = 保留。不连接任何通道
 ·
 ·
 ·
 01000 = 保留。不连接任何通道
 00111 = AN7⁽⁴⁾
 00110 = AN6⁽⁴⁾
 00101 = AN5⁽⁴⁾
 00100 = AN4⁽⁴⁾
 00011 = AN3
 00010 = AN2
 00001 = AN1
 00000 = AN0

bit 1 **GO/DONE:** ADC 转换状态位
 1 = ADC 转换正在进行。将该位置 1 可启动 ADC 转换周期。
 ADC 转换完成后, 该位由硬件自动清零。
 0 = ADC 转换已完成 / 未进行

bit 0 **ADON:** ADC 使能位
 1 = 使能 ADC
 0 = 禁止 ADC, 不消耗工作电流

- 注 1: 请参见第 15.0 节 “温度指示器模块”。
- 2: 更多信息, 请参见第 17.0 节 “8 位数模转换器 (DAC1) 模块”。
- 3: 更多信息, 请参见第 14.0 节 “固定参考电压 (FVR)”。
- 4: AN<7:4> 仅在 PIC16(L)F1613 上可用。

PIC12(L)F1612/16(L)F1613

寄存器 16-2: **ADCON1: ADC 控制寄存器 1**

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7

ADFM: ADC 结果格式选择位

1 = 右对齐。当装入转换结果时, ADRESH 的高 6 位设置为 0。

0 = 左对齐。当装入转换结果时, ADRESL 的低 6 位设置为 0。

bit 6-4

ADCS<2:0>: ADC 转换时钟选择位

111 = FRC (由内部 RC 振荡器提供的时钟)

110 = Fosc/64

101 = Fosc/16

100 = Fosc/4

011 = FRC (由内部 RC 振荡器提供的时钟)

010 = Fosc/32

001 = Fosc/8

000 = Fosc/2

bit 3-2

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⁽¹⁾

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。详情请参见第 28.0 节“电气规范”。

PIC12(L)F1612/16(L)F1613

寄存器 16-3: ADCON2: ADC 控制寄存器 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
TRIGSEL<3:0> ⁽¹⁾				—	—	—	—
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 **TRIGSEL<3:0>**: 自动转换触发源选择位 ⁽¹⁾

1111 = SMT2_PR
 1110 = SMT1_PR
 1101 = TMR6_postscaled
 1100 = TMR4_postscaled
 1011 = SMT2_CPR
 1010 = SMT2_CPW
 1001 = SMT1_CPR
 1000 = SMT1_CPW
 0111 = C2_OUT_sync⁽³⁾
 0110 = C1_OUT_sync
 0101 = TMR2_postscaled
 0100 = T1_overflow⁽²⁾
 0011 = T0_overflow⁽²⁾
 0010 = CCP2_out
 0001 = CCP1_out
 0000 = 未选择任何自动转换触发源

bit 3-0 **未实现**: 读为 0

- 注
- 1: 这是所有触发源的上升沿敏感输入。
 - 2: 信号还会将其相应的中断标志置 1。
 - 3: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

寄存器 16-4: **ADRESH: ADC 结果寄存器高字节 (ADRESH) ADFM = 0**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
bit 7							bit 0

图注:

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

bit 7-0 **ADRES<9:2>**: ADC 结果寄存器位
10 位转换结果的高 8 位

寄存器 16-5: **ADRESL: ADC 结果寄存器低字节 (ADRESL) ADFM = 0**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>							
bit 7							bit 0

图注:

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

bit 7-6 **ADRES<1:0>**: ADC 结果寄存器位
10 位转换结果的低 2 位

bit 5-0 **保留**: 不要使用。

PIC12(L)F1612/16(L)F1613

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

R/W-x/u	R/W-x/u						
—	—	—	—	—	—	ADRES<9:8>	
bit 7							bit 0

图注:

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

bit 7-2 **保留: 不要使用。**
bit 1-0 **ADRES<9:8>: ADC 结果寄存器位**
 10 位转换结果的高 2 位

寄存器 16-7: ADRESL: ADC 结果寄存器低字节 (ADRESL) ADFM = 1

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

图注:

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

PIC12(L)F1612/16(L)F1613

16.4 ADC 采集要求

为了使 ADC 达到规定的精度，必须使充电保持电容（CHOLD）完全充电至输入通道的电平。模拟输入模型如图 16-4 所示。模拟信号源阻抗（Rs）和内部采样开关阻抗（Rss）直接影响电容 CHOLD 的充电时间。采样开关阻抗（Rss）随器件电压（VDD）的变化而变化，参见图 16-4。模拟信号源的最大阻抗推荐值为 10 kΩ。

采集时间可能随着源阻抗的降低而缩短。在选择（或改变）模拟输入通道后，必须在启动转换前完成 ADC 采集。可以使用公式 16-1 来计算最小采集时间。该公式假设误差为 1/2 LSB（ADC 转换需要 1,024 步）。1/2 LSB 误差是 ADC 达到规定分辨率所能允许的最大误差。

公式 16-1: 采集时间示例

假设: 温度 = 50°C, 外部阻抗为 10 kΩ, VDD 为 5.0V

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

Tc 值可以用以下公式近似计算:

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

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

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

注: 其中 n = ADC 的位数。

求解 Tc:

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

因此:

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

- 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 16-4: 模拟输入模型

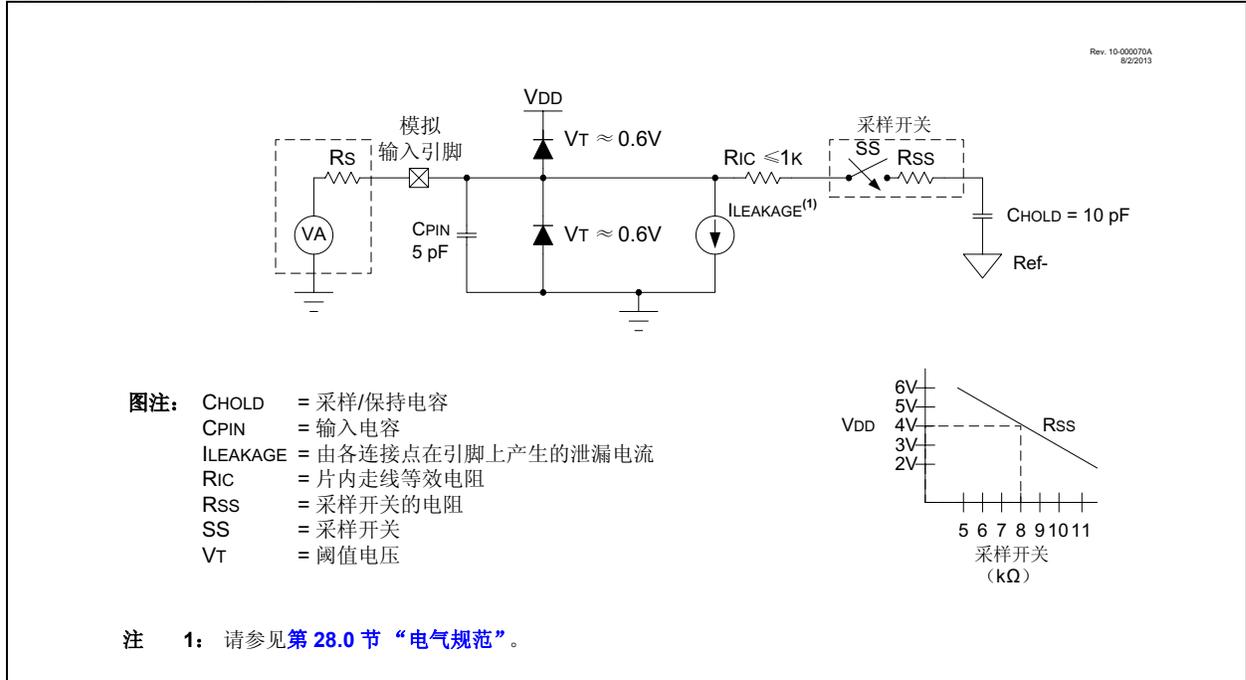
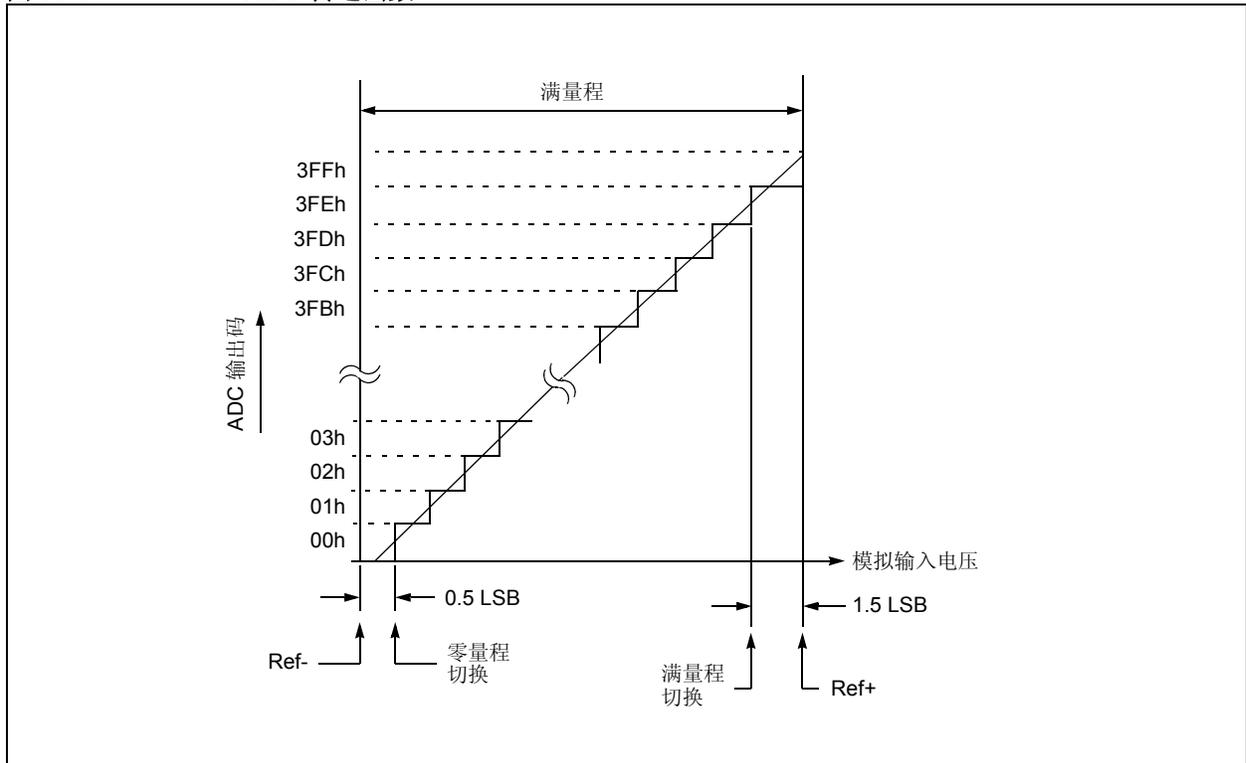


图 16-5: ADC 传递函数



PIC12(L)F1612/16(L)F1613

表 16-3: 与 ADC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ADCON0	—	CHS<4:0>					GO/DONE	ADON	157
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		158
ADCON2	TRIGSEL<3:0>				—	—	—	—	159
ADRESH	ADC 结果寄存器的高字节								160, 161
ADRESL	ADC 结果寄存器的低字节								160, 161
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	132
ANSELC ⁽²⁾	—	—	—	—	ANSC3	ANSC2	ANSC1	ANSC0	139
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	79
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	83
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	131
TRISC ⁽²⁾	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	138
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		148

图注: x = 未知, u = 不变, — = 未实现, 读为 0, α = 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。

注 2: 仅限 PIC16(L)F1613。

17.0 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
- 固定参考电压 (FVR)

DAC 的输出可以配置为向以下对象提供参考电压：

- 比较器的正输入
- 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。

17.1 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 17-1 确定：

公式 17-1: DAC 输出电压

如果 $DAC1EN = 1$

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

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

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

17.2 比例输出电压

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

第 28.0 节“电气规范”中给出了梯形电阻网络中各个电阻的阻值。

17.3 DAC 参考电压输出

通过将 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 17-2 举例说明了这一缓冲技术。

PIC12(L)F1612/16(L)F1613

图 17-1: 数模转换器框图

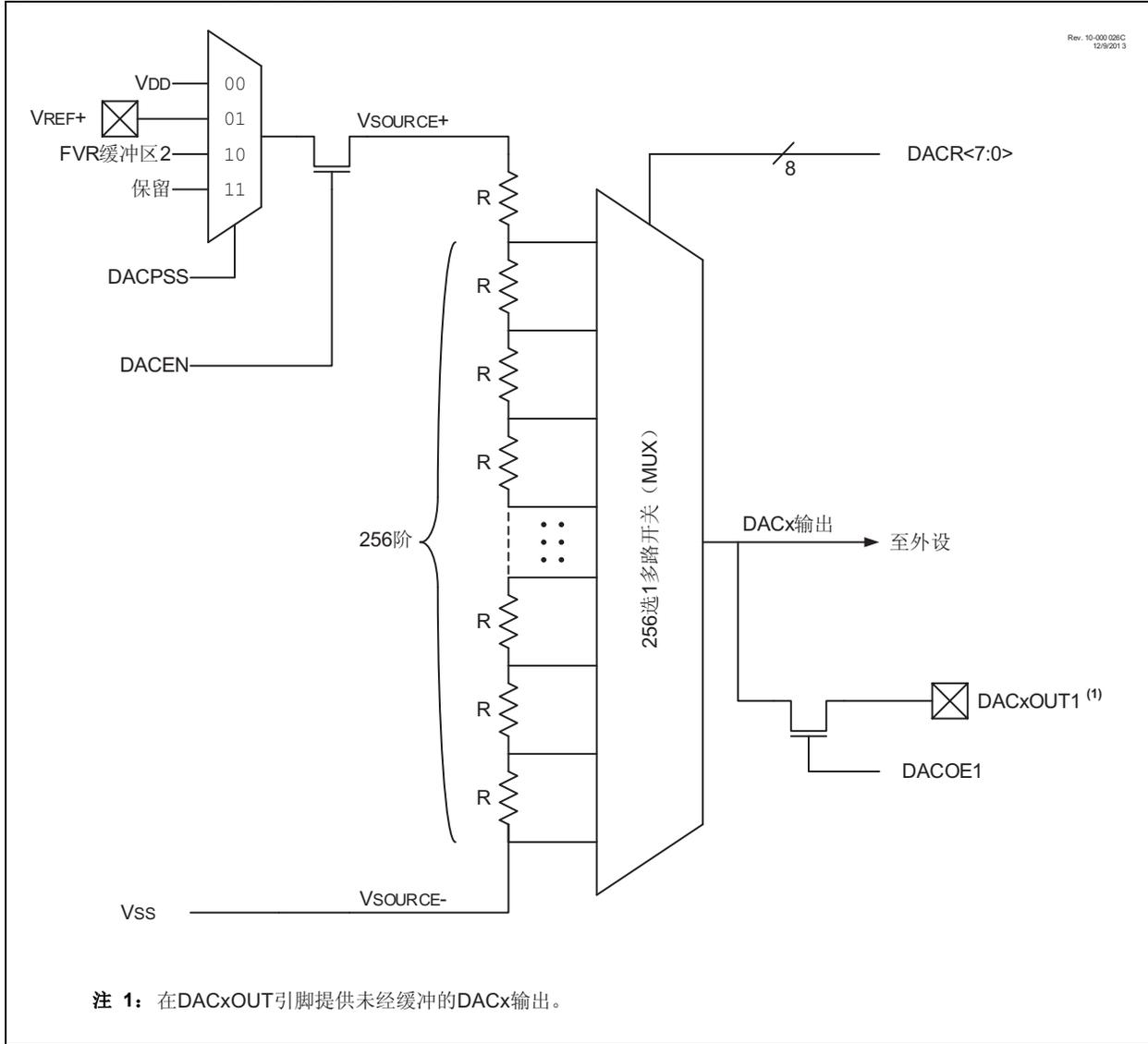
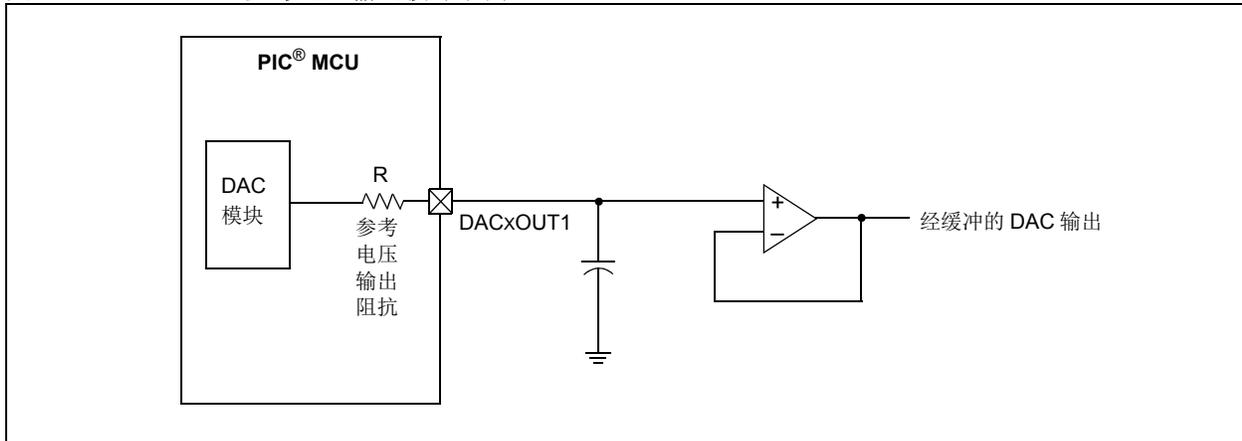


图 17-2: 参考电压输出缓冲示例



17.4 休眠期间的操作

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。

17.5 复位的影响

器件复位会产生以下影响：

- DAC 被禁止。
- DAC 输出电压从 DACXOUT1 引脚上被移除。
- DAC1R<7:0> 范围选择位被清零。

PIC12(L)F1612/16(L)F1613

17.6 寄存器定义：DAC 控制

寄存器 17-1: 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 17-2: 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 17-1: 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>		148
DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	168
DAC1CON1	DAC1R<7:0>								168

图注: — = 未实现位, 读为 0。DAC 模块不使用阴影单元。

18.0 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

18.1 比较器概述

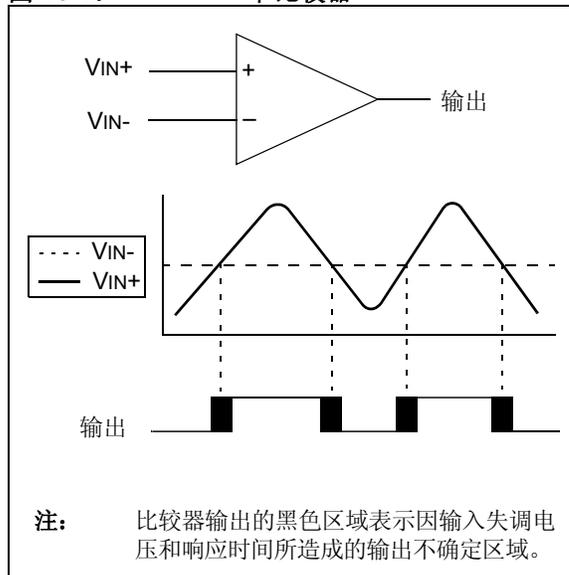
图 18-1 所示为单比较器以及模拟输入电平与数字输出之间的关系。当 V_{IN+} 上的模拟电压小于 V_{IN-} 上的模拟电压时，比较器输出为数字低电平。当 V_{IN+} 上的模拟电压大于 V_{IN-} 上的模拟电压时，比较器输出为数字高电平。

表 18-1 列出了该器件可用的比较器。

表 18-1: 每款器件可用的比较器

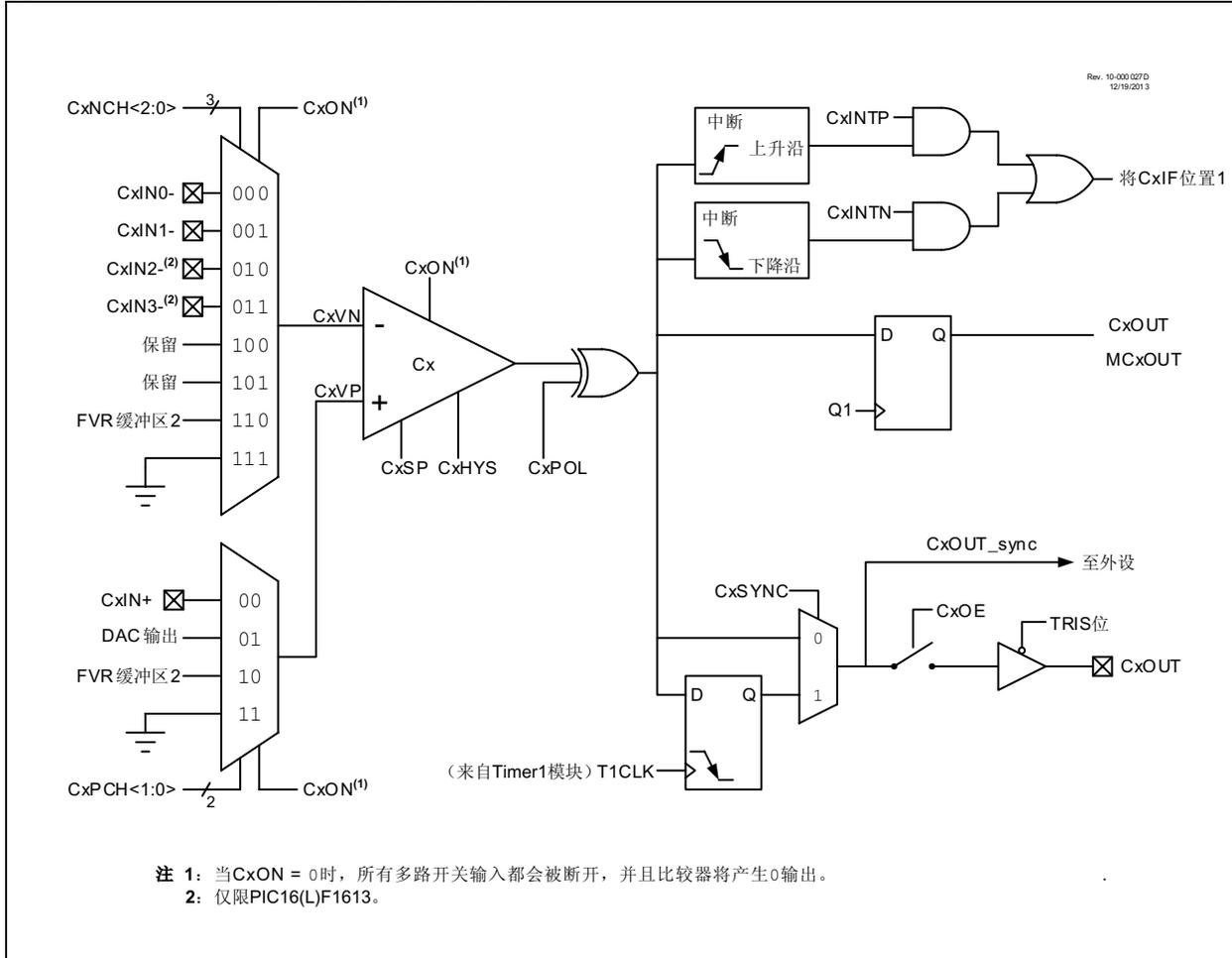
器件	C1	C2
PIC16(L)F1613	•	•
PIC12(L)F1612	•	

图 18-1: 单比较器



PIC12(L)F1612/16(L)F1613

图 18-2: 比较器模块的简化框图



18.2 比较器控制

每个比较器都具有 2 个控制寄存器：CMxCON0 和 CMxCON1。

CMxCON0 寄存器（见寄存器 18-1）包含以下控制和状态位：

- 使能
- 输出选择
- 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 18-2）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

18.2.1 比较器使能

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

18.2.2 比较器输出选择

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。

18.2.3 比较器输出极性

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

表 18-2 给出了输出状态与输入条件的关系（包括极性控制）。

表 18-2: 比较器输出状态与输入条件

输入条件	CxPOL	CxOUT
$CxVN > CxVP$	0	0
$CxVN < CxVP$	0	1
$CxVN > CxVP$	1	1
$CxVN < CxVP$	1	0

18.2.4 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。

PIC12(L)F1612/16(L)F1613

18.3 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，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 28.0 节“电气规范”。

18.4 Timer1 门控操作

比较器操作产生的输出可以用作 Timer1 的门控源。更多信息，请参见第 21.5 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。

18.4.1 比较器输出同步

通过将 CMxCON0 寄存器的 CxSYNC 位置 1，可以使比较器的输出与 Timer1 保持同步。

使能比较器的输出时，比较器的输出在 Timer1 时钟源的下降沿被锁存。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 18-2）和 Timer1 框图（图 21-1）。

18.5 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。

18.6 比较器同相输入选择

通过配置 CMxCON1 寄存器的 CxPCH<2:0> 位，将内部参考电压或模拟引脚连接到比较器的同相输入。

- CxIN+ 模拟引脚
- DAC 输出
- 固定参考电压（FVR）
- Vss（地）

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

关于 DAC 输入信号的更多信息，请参见第 17.0 节“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。

18.7 比较器反相输入选择

CMxCON1 寄存器的 CxNCH<2:0> 位指示模拟输入引脚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。

18.8 比较器响应时间

在改变输入源或选择新的参考电压后，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 28.0 节“电气规范”中的比较器和参考电压规范。

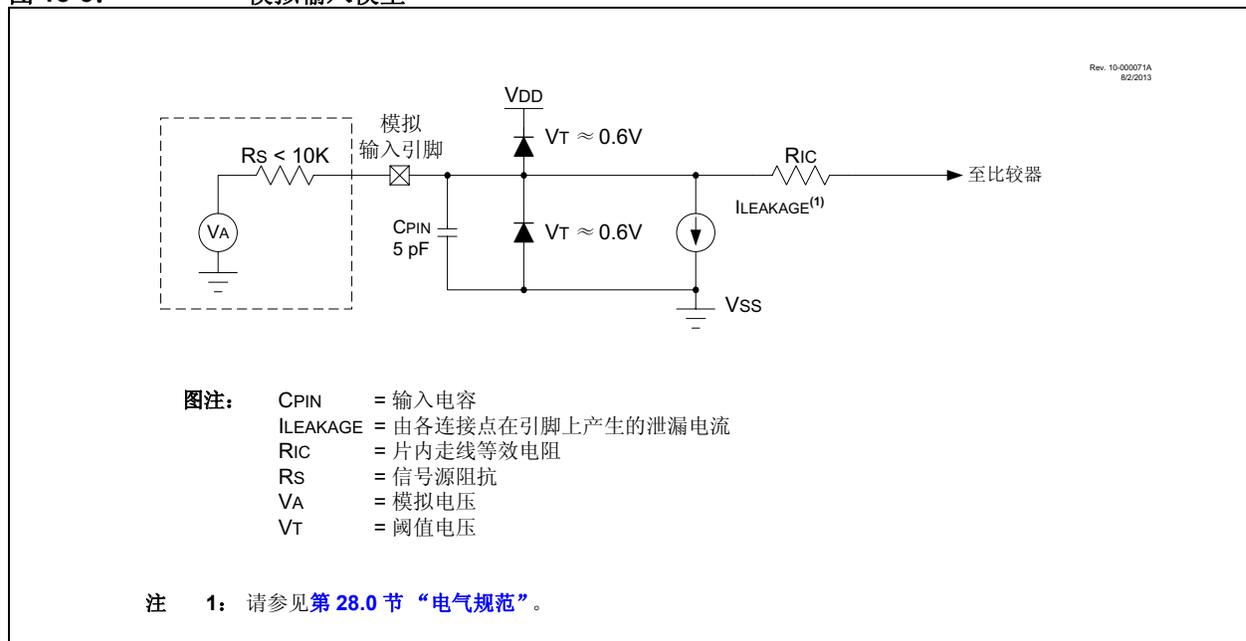
模拟信号源的最大阻抗推荐值为 10 kΩ。任何连接到模拟输入引脚的外部元件（如电容或齐纳二极管），应保证其泄漏电流极小以使引入的误差降至最低。

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

18.9 模拟输入连接注意事项

模拟输入的简化电路如图 18-3 所示。由于模拟输入引脚与数字输入共用连接，它们在 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。

图 18-3: 模拟输入模型



PIC12(L)F1612/16(L)F1613

18.10 寄存器定义：比较器控制

寄存器 18-1: **CMxCON0: 比较器 Cx 控制寄存器 0**

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7 **CxON:** 比较器使能位
1 = 使能比较器
0 = 禁止比较器, 不消耗有功功率
- bit 6 **CxOUT:** 比较器输出位
如果 CxPOL = 1 (极性反相):
1 = CxVP < CxVN
0 = CxVP > CxVN
如果 CxPOL = 0 (极性不反相):
1 = CxVP > CxVN
0 = CxVP < CxVN
- bit 5 **CxOE:** 比较器输出使能位
1 = CxOUT 出现在 CxOUT 引脚。只有关联的 TRIS 位清零时, 才能实际驱动引脚。不受 CxON 影响。
0 = CxOUT 仅在内部有效
- bit 4 **CxPOL:** 比较器输出极性选择位
1 = 比较器输出反相
0 = 比较器输出同相
- bit 3 **未实现:** 读为 0
- bit 2 **CxSP:** 比较器速度 / 功耗选择位
1 = 比较器工作在正常功耗、高速模式下
0 = 比较器工作在低功耗、低速模式下
- bit 1 **CxHYS:** 比较器滞后使能位
1 = 使能比较器滞后
0 = 禁止比较器滞后
- bit 0 **CxSYNC:** 比较器输出同步模式位
1 = 送到 Timer1 和 I/O 引脚的比较器输出与 Timer1 时钟源的变化进行同步。输出在 Timer1 时钟源的下沿进行更新。
0 = 送到 Timer1 和 I/O 引脚的比较器输出是异步的

PIC12(L)F1612/16(L)F1613

寄存器 18-2: **CMxCON1: 比较器 Cx 控制寄存器 1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<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 ⁽¹⁾
 010 = CxVN 连接到 CxIN2- 引脚 ⁽¹⁾
 001 = CxVN 连接到 CxIN1- 引脚
 000 = CxVN 连接到 CxIN0- 引脚

注 1: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

寄存器 18-3: CMOUT: 比较器输出寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT ⁽¹⁾	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 ⁽¹⁾
bit 0 **MC1OUT**: C1OUT 的镜像副本位

注 1: 仅限 PIC16(L)F1613。PIC12(L)F1612 上未实现。

表 18-3: 与比较器模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	132
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	174
CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			175
CM2CON0 ⁽²⁾	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	174
CM2CON1 ⁽²⁾	C2INTP	C2INTN	C2PCH<1:0>		—	C2NCH<2:0>			175
CMOUT	—	—	—	—	—	—	MC2OUT ⁽²⁾	MC1OUT	176
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		148
DAC1CON0	DAC1EN	—	DAC1OE1	—	DAC1PSS<1:0>		—	—	168
DAC1CON1	DAC1R<7:0>								168
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
PIE2	—	C2IE ⁽²⁾	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	80
PIR2	—	C2IF ⁽²⁾	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	84
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	131
TRISC ⁽²⁾	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	138

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。
注 2: 仅限 PIC16(L)F1613。

19.0 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 19-2 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

19.1 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。在通过电阻的标称电流为 $300\mu\text{A}$ 时，选择的阻值产生的压降需要等于全部峰值电压。请参见公式 19-1 和图 19-1。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 19-1: 外部电阻

$$R_{SERIES} = \frac{V_{PEAK}}{3 \times 10^{-4}}$$

图 19-1: 外部电压

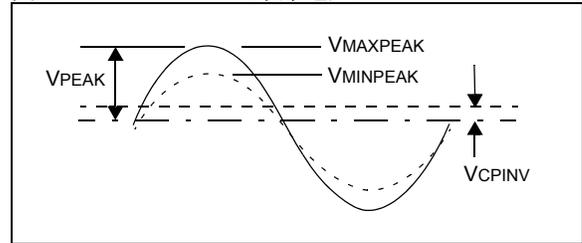
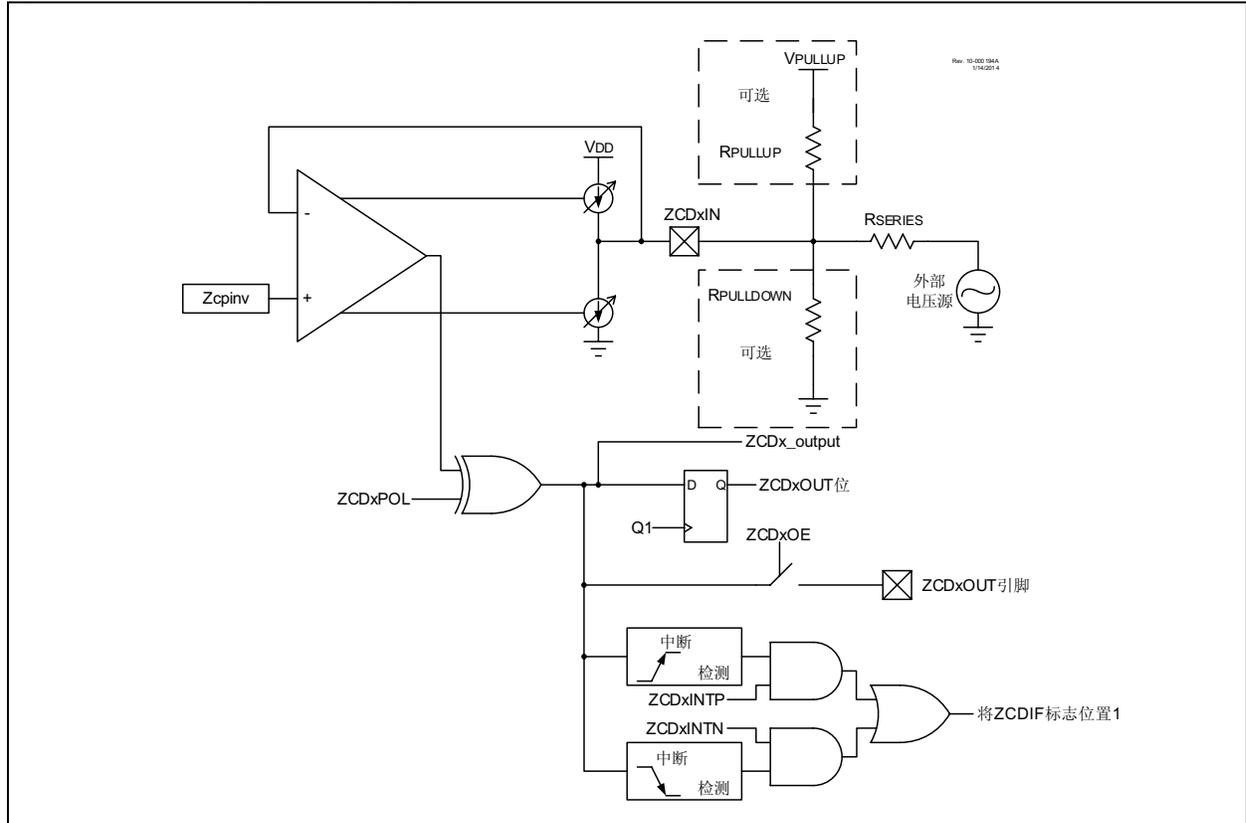


图 19-2: 简化 ZCD 框图



PIC12(L)F1612/16(L)F1613

19.2 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。

19.3 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 19.4 节“ZCD 中断”。

19.4 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。

19.5 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 19-2 所示的相应公式，可以确定正弦波形的实际偏移时间。

公式 19-2: ZCD 事件偏移

当外部电压源相对于 Vss 时：

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

当外部电压源相对于 VDD 时：

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

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 19-3 或公式 19-4 所示的公式确定。

公式 19-3: 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 19-2 和公式 19-3，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 ΔT 等于 $4 * T_{OFFSET}$ 。公式 19-4 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 19-4:

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

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)。

ΔT 为 ZCDxOUT 高电平和低电平周期的差值。

19.6 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 19-5 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 19-3 确定，因为上拉值与峰值电压无关。

公式 19-5: 对应于 V 范围的串联电阻 R

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

19.7 休眠期间的操作

ZCD 电流源和中断不会受休眠影响。

19.8 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 ZCDDIS 配置位清零时，ZCD 电路将在 POR 时处于工作状态。当 ZCDDIS 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。

PIC12(L)F1612/16(L)F1613

19.9 寄存器定义：ZCD 控制

寄存器 19-1: ZCDxCON: 过零检测控制寄存器

R/W-q/q	R/W-0/0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ZCDxEN	ZCDxOE	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 **ZCDxEN:** 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 **ZCDxOE:** 过零检测输出使能位
 1 = 使能 ZCD 引脚输出
 0 = 禁止 ZCD 引脚输出
- bit 5 **ZCDxOUT:** 过零检测逻辑电平位
ZCDxPOL 位 = 0:
 1 = ZCD 引脚产生灌电流
 0 = ZCD 引脚产生拉电流
ZCDxPOL 位 = 1:
 1 = ZCD 引脚产生拉电流
 0 = ZCD 引脚产生灌电流
- bit 4 **ZCDxPOL:** 过零检测逻辑输出极性位
 1 = 对 ZCD 逻辑输出进行反相
 0 = 不对 ZCD 逻辑输出进行反相
- bit 3-2 **未实现:** 读为 0
- bit 1 **ZCDxINTP:** 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 **ZCDxINTN:** 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 19-1: 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	—	—	—	—	81
PIR3	—	—	CWGIF	ZCDIF	—	—	—	—	85
ZCD1CON	ZCD1EN	ZCD1OE	ZCD1OUT	ZCD1POL	—	—	ZCD1INTP	ZCD1INTN	180

图注: — = 未实现位, 读为 0。ZCD 模块不使用阴影单元。

表 19-2: 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	49
	7:0	ZCDDIS	—	—	—	—	—	WRT<1:0>		

图注: — = 未实现位, 读为 0。ZCD 模块不使用阴影单元。

20.0 TIMER0 模块

Timer0 模块是 8 位定时器 / 计数器，具有以下特性：

- 8 位定时器 / 计数器寄存器 (TMR0)
- 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 20-1 给出了 Timer0 模块的框图。

20.1 Timer0 工作原理

Timer0 模块可被用作 8 位定时器或 8 位计数器。

20.1.1 8 位定时器模式

如果 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。

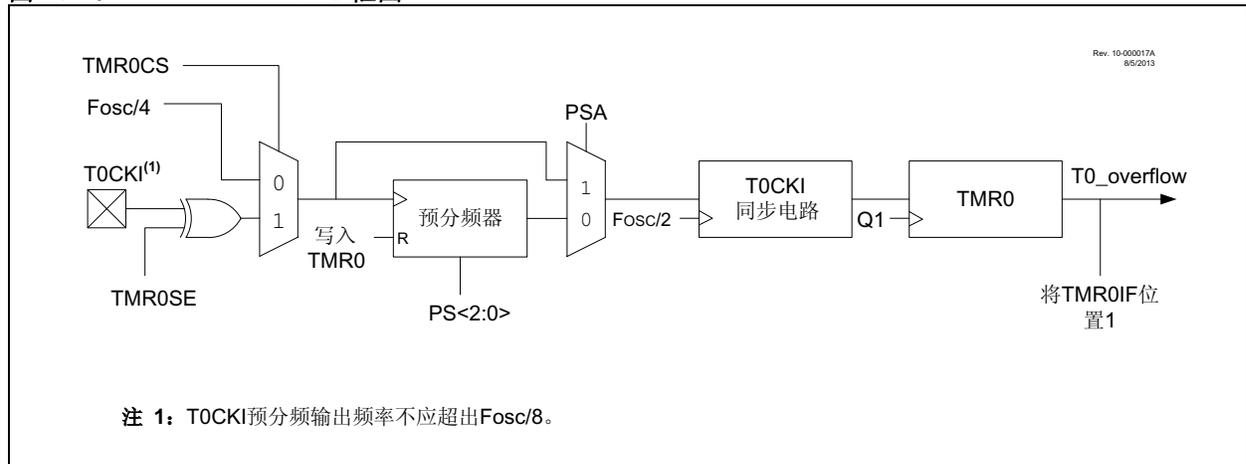
20.1.2 8 位计数器模式

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 20-1: TIMER0 框图



PIC12(L)F1612/16(L)F1613

20.1.3 软件可编程的预分频器

软件可编程的预分频器只能用于 Timer0。可通过清零 OPTION_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。

20.1.4 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。

20.1.5 8 位同步计数器模式

在 8 位计数器模式下，T0CKI 引脚的递增边沿必须与指令时钟保持同步。同步可通过在指令时钟的 Q2 和 Q4 周期对预分频器的输出进行采样实现。外部时钟源的高低电平周期必须满足第 28.0 节“电气规范”中所示的时序要求。

20.1.6 休眠期间的操作

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

PIC12(L)F1612/16(L)F1613

20.2 寄存器定义：选项寄存器

寄存器 20-1: OPTION_REG: 选项寄存器

R/W-1/1							
WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>		
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7 **WPUEN:** 弱上拉使能位
1 = 禁止所有弱上拉 (MCLR 除外, 如果已使能)
0 = 通过各个 WPUx 锁存值使能弱上拉
- bit 6 **INTEDG:** 中断边沿选择位
1 = INT 引脚的上升沿触发中断
0 = INT 引脚的下降沿触发中断
- bit 5 **TMR0CS:** Timer0 时钟源选择位
1 = T0CKI 引脚上的电平跳变
0 = 内部指令周期时钟 (Fosc/4)
- bit 4 **TMR0SE:** Timer0 时钟源边沿选择位
1 = 在 T0CKI 引脚信号从高至低跳变时, 递增计数
0 = 在 T0CKI 引脚信号从低至高跳变时, 递增计数
- bit 3 **PSA:** 预分频器分配位
1 = 预分频器未分配给 Timer0 模块
0 = 预分频器分配给 Timer0 模块
- bit 2-0 **PS<2:0>:** 预分频比选择位

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

表 20-1: 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<3:0>				—	—	—	—	159
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			183
TMR0	8 位 Timer0 计数的保持寄存器								181*
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	131

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。

PIC12(L)F1612/16(L)F1613

21.1 Timer1 工作原理

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

Timer1 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 21-1 显示了 Timer1 使能选择。

表 21-1: 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

21.2 时钟源选择

T1CON 寄存器的 TMR1CS<1:0> 位用于选择 Timer1 的时钟源。表 21-2 显示了时钟源选择。

21.2.1 内部时钟源

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

选择 Fosc 内部时钟源时，Timer1 寄存器的值将在每个指令时钟周期中递增 4 次。由于这个原因，在读取 Timer1 值时，分辨率将会出现 2 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（仅限 PIC16(L)F1613）比较器输入用于进行 Timer1 门控

21.2.2 外部时钟源

当选择外部时钟源时，Timer1 模块可以作为定时器或计数器工作。

Timer1 使能计数时，在外部时钟输入 T1CKI 的上升沿递增。外部时钟源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 21-2: 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）

PIC12(L)F1612/16(L)F1613

21.3 Timer1 预分频器

Timer1 有 4 个预分频比选项，允许对时钟输入进行 1、2、4 或 8 分频。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。

21.4 异步计数器模式下的 Timer1 操作

如果 T1CON 寄存器的控制位 $\overline{T1SYNC}$ 置 1，外部时钟输入将不同步。定时器异步于内部相位时钟进行递增计数。如果选择了外部时钟源，在休眠期间定时器将继续运行，并在上溢时产生中断以唤醒处理器。但是，用软件对定时器进行读/写操作时，要特别当心（见第 21.4.1 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。

21.4.1 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。

21.5 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。

21.5.1 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 21-3。

表 21-3: TIMER1 门控使能选择

T1CLK	T1GPOL	T1G	Timer1 工作状态
↑	0	0	计数
↑	0	1	保持计数
↑	1	0	保持计数
↑	1	1	计数

21.5.2 TIMER1 门控源选择

表 21-4 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 21-4: 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) (1)
11	比较器 2 输出 (C2_OUT_sync) (1,2)

注 1: 可选的同步比较器输出。
2: 仅限 PIC16(L)F1613。

21.5.2.1 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。

21.5.2.2 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。

21.5.3 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，该触发器在信号的每个递增边沿改变状态。时序详细信息请参见图 21-4。

Timer1 门控翻转模式通过将 T1GCON 寄存器的 T1GTM 位置 1 使能。T1GTM 位清零时，将清除触发器并保持清零。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。

21.5.4 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 21-5。

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 21-6。

21.5.5 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。

21.5.6 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。

PIC12(L)F1612/16(L)F1613

21.6 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。

21.7 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。

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

21.7.1 备用引脚位置

该模块具有以下 I/O 引脚: 通过使用备用引脚功能寄存器 APFCON, 可将 I/O 引脚转移到其他位置。要确定哪些引脚可转移以及其在复位时的默认位置, 请参见第 12.1 节 “备用引脚功能” 了解更多信息。

图 21-2: TIMER1 递增边沿

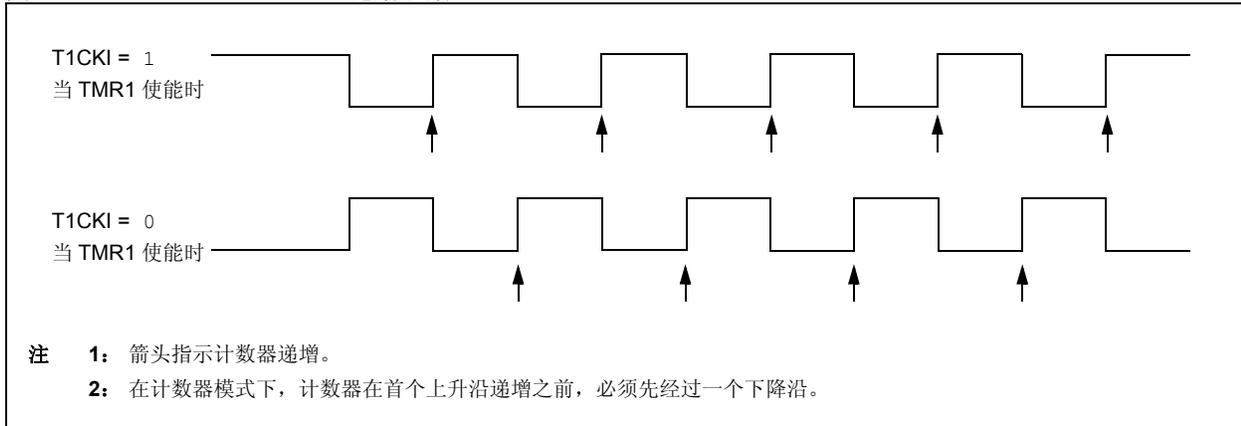


图 21-3: TIMER1 门控使能模式

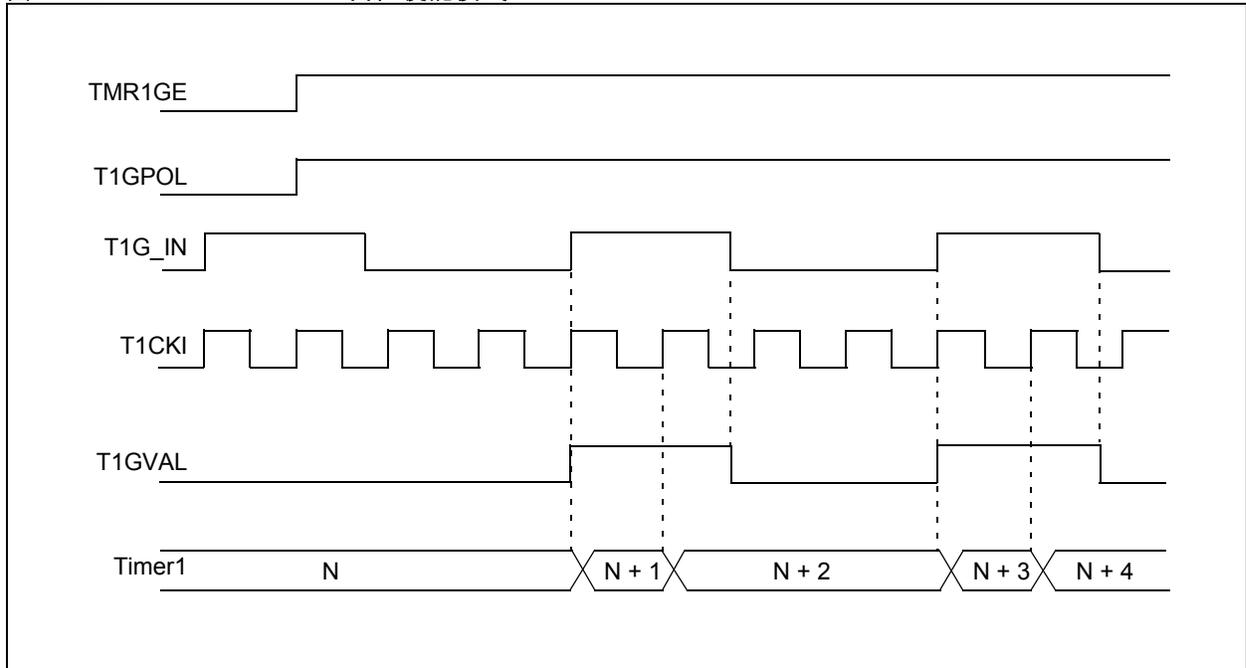
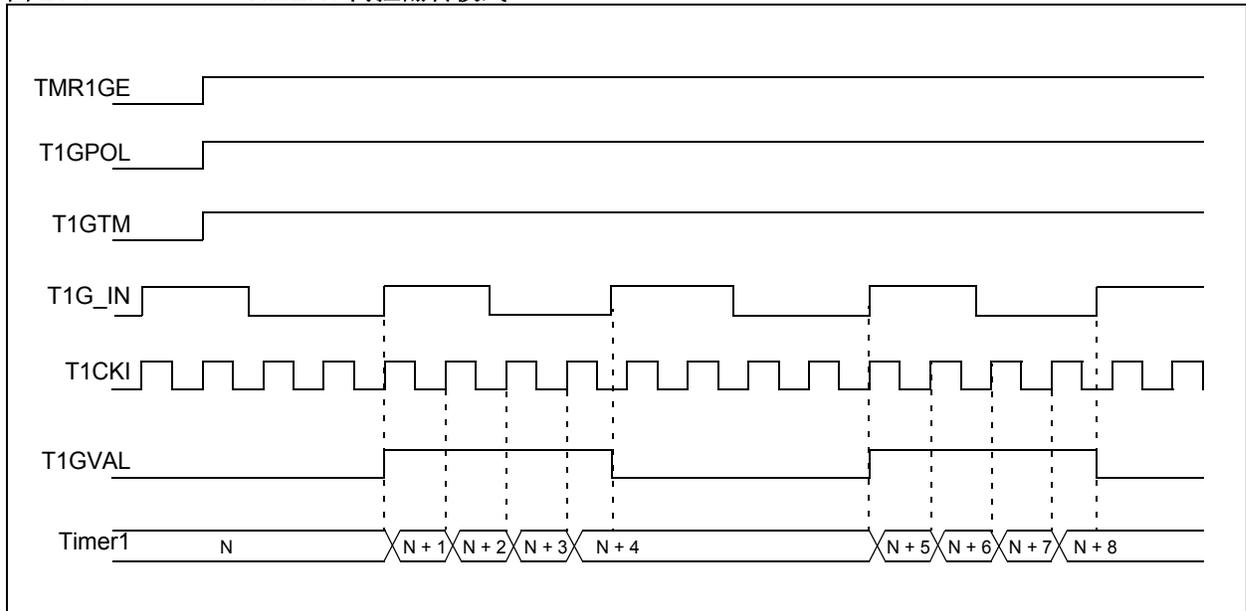


图 21-4: TIMER1 门控翻转模式



PIC12(L)F1612/16(L)F1613

图 21-5: TIMER1 门控单脉冲模式

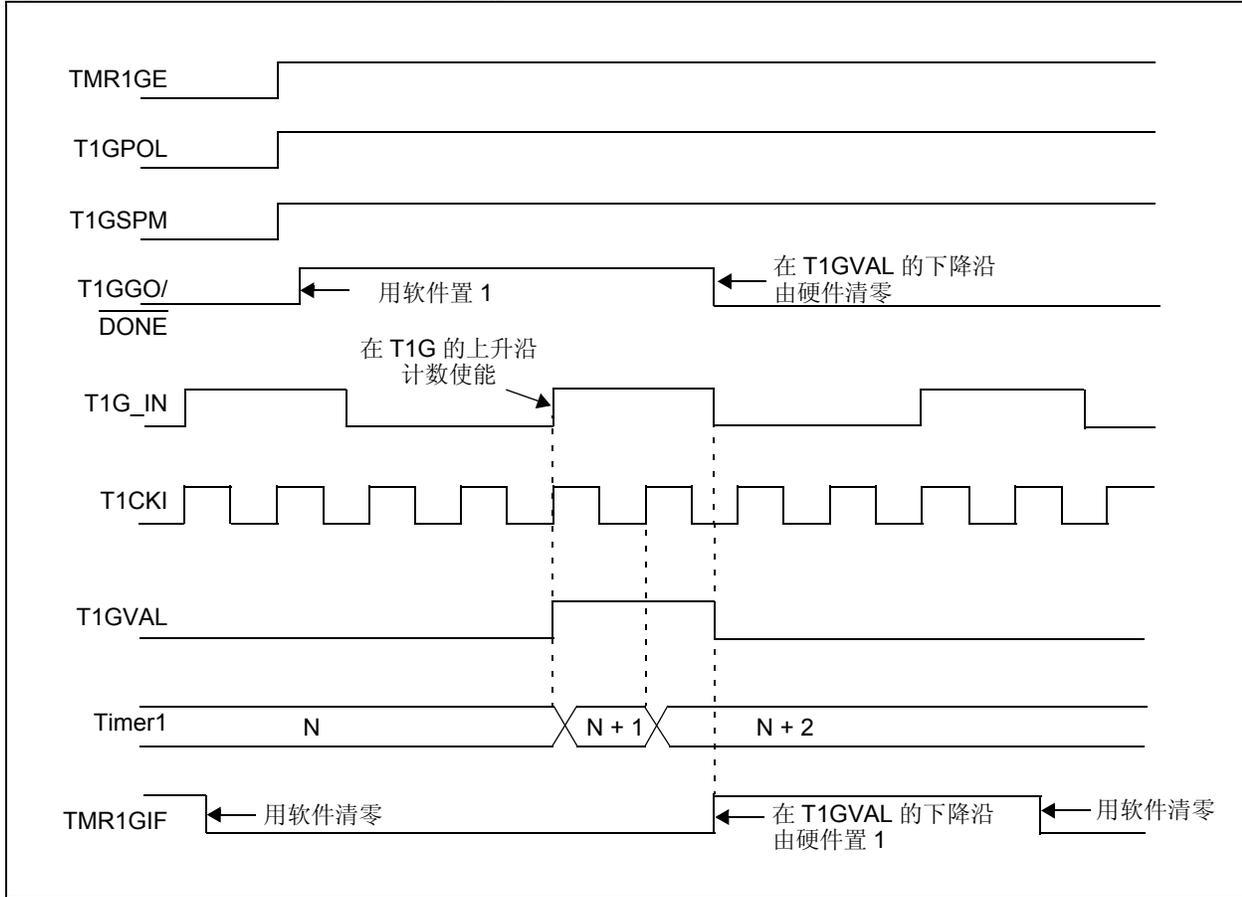
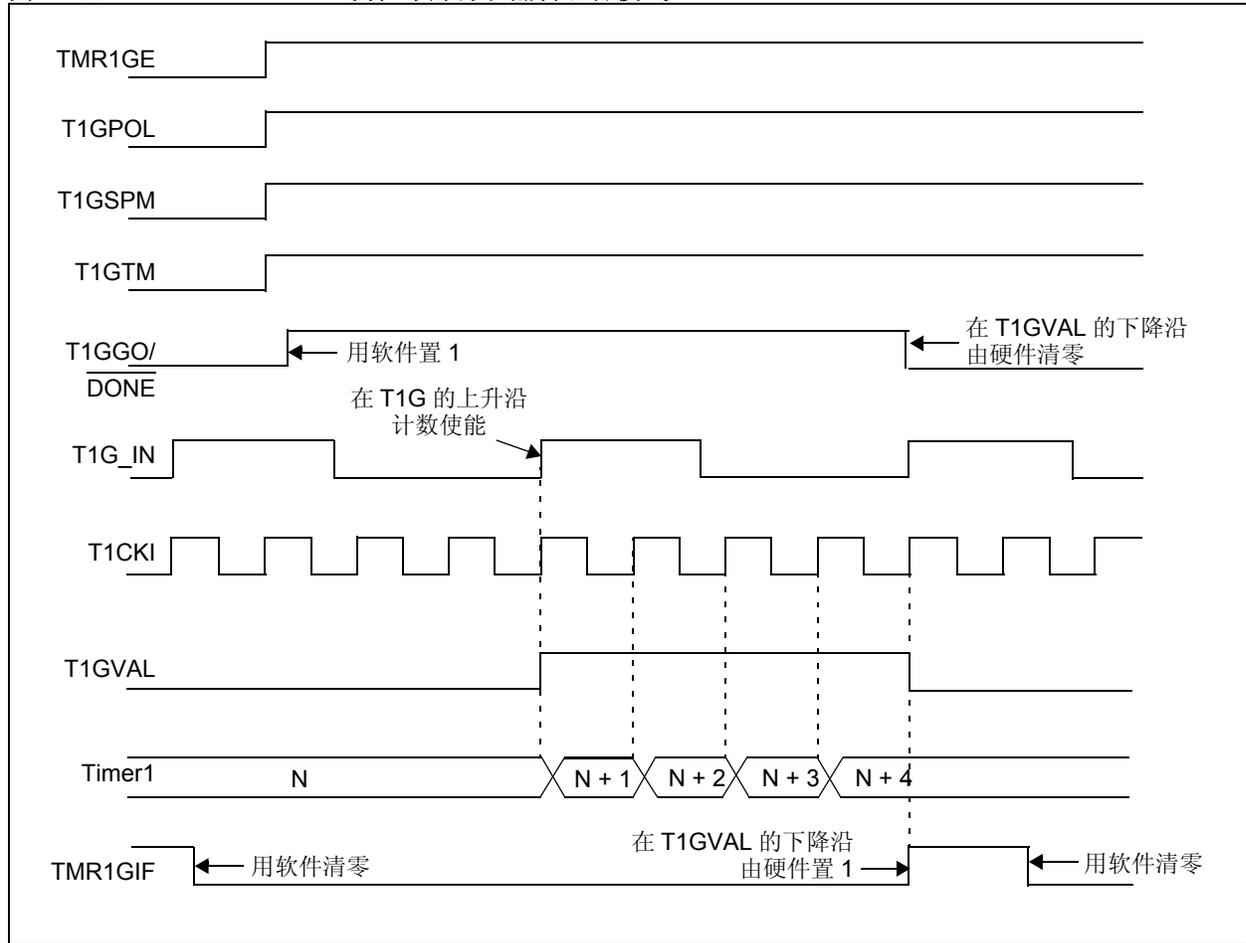


图 21-6: TIMER1 门控单脉冲和翻转组合模式



PIC12(L)F1612/16(L)F1613

21.8 寄存器定义: Timer1 控制

寄存器 21-1: **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>		—	$\overline{T1SYNC}$	—	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 **$\overline{T1SYNC}$** : 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

PIC12(L)F1612/16(L)F1613

寄存器 21-2: 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 1-0 **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) ⁽¹⁾
 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)

注 1: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

表 21-5: 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	132
APFCON	—	CWGASEL ⁽²⁾	CWGBSEL ⁽²⁾	—	T1GSEL	—	CCP2SEL ⁽³⁾	CCP1SEL ⁽²⁾	128
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	79
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	83
TMR1H	16 位 TMR1 计数最高有效字节的保持寄存器								188*
TMR1L	16 位 TMR1 计数最低有效字节的保持寄存器								188*
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	131
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	T1SYNC	—	TMR1ON	192
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		193

图注: — = 未实现位, 读为 0。Timer1 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 PIC12(L)F1612。

3: 仅限 PIC16(L)F1613。

22.0 TIMER2/4/6 模块

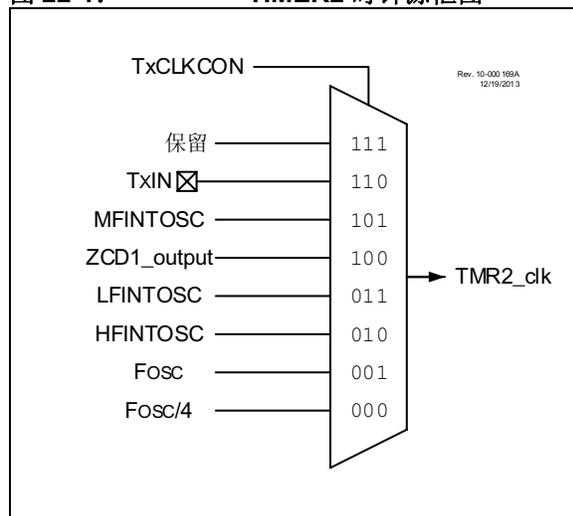
Timer2/4/6 模块是 8 位定时器，具有以下特性：

- 8 位定时器和周期寄存器（分别为 TMR2 和 PR2）
- 可读写（以上两个寄存器）
- 可软件编程的预分频器（分频比为 1:1 至 1:128）
- 软件可编程的后分频器（分频比为 1:1 至 1:16）
- TMR2 与 PR2 匹配时产生中断
- 单事件操作
- 全异步操作
- 包含硬件限制定时器（Hardware Limit Timer, HLT）扩展
- 备用时钟源
- 外部定时器复位信号源
- 可配置定时器复位操作

关于 Timer2 时钟源，请参见图 22-1。关于带 HLT 的 Timer2 的框图，请参见图 22-2。

注： 该器件上实现了 3 个相同的 Timer2 模块。这些定时器命名为 Timer2、Timer4 和 Timer6。对 Timer2 的所有引用均适用于 Timer4 和 Timer6。对 PR2 的所有引用均适用于 PR4 和 PR6。

图 22-1: TIMER2 时钟源框图



22.1 Timer2 工作原理

时钟输入上的 7 位计数器 / 预分频器支持多种预分频选项：从直接输入到 128 分频。这些选项通过 TxCON 寄存器的预分频控制位 CKPS<2:0> 进行选择。

在每个时钟周期，TMR2 的值都会与周期寄存器 PR2 中的值进行比较。当两个值匹配时，比较器会将 TMR2 的值在下一个周期复位为 00h，并驱动输出计数器 / 后分频器（见第 22.2 节“Timer2 中断”）。此外，还可以通过使用外部复位信号来复位定时器，如第 22.4 节“外部复位源”所述。

TMR2 和 PR2 寄存器均可直接读写。在任何器件复位时，TMR2 寄存器都会清零，而 PR2 寄存器则初始化为 FFh。发生以下事件时，预分频器和后分频器计数器都会清零：

- 对 TMR2 寄存器进行写操作
- 对 T2CON 寄存器进行写操作
- 上电复位（POR）

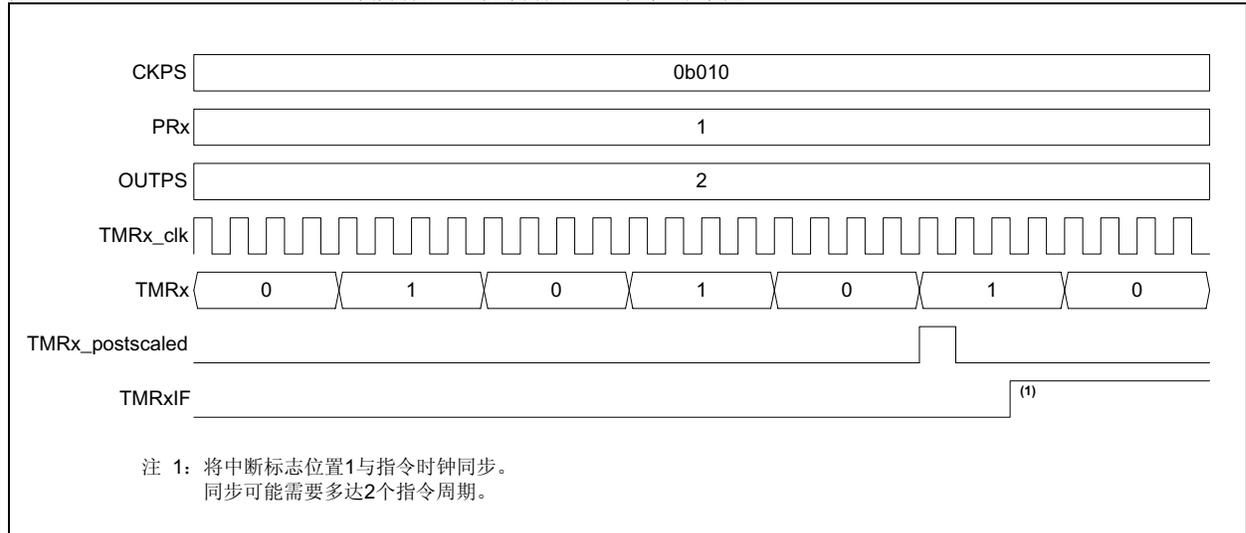
- 欠压复位（BOR）
- MCLR 复位
- 看门狗定时器（WDT）复位
- 堆栈上溢复位
- 堆栈下溢复位
- RESET 指令
- 外部复位源事件，它们会复位定时器。

注： 写 T2CON 时 TMR2 不会清零。

22.2 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 TMR2 中断允许位 TMR2IE 置 1 来允许该中断。中断时序如图 22-3 所示。

图 22-3: TIMER2 预分频器、后分频器和中断时序图



PIC12(L)F1612/16(L)F1613

22.3 Timer2 输出

Timer2 模块的主输出为 TMR2_postscaled，它在每次后分频器计数器与 OUTPS TMR2xCON 匹配时产生单个 TMR2_clk 周期的脉冲。PR2 后分频器在每次 TMR2 值与 PR2 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：

- CRC 存储器扫描器，作为触发模式的触发信号
- ADC 模块，作为自动转换触发信号
- 两个 SMT 模块，作为窗口和 / 或信号输入
- CWG，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 23.4 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 22.5 节“操作示例”中的时序图。

22.4 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<3:0> 位控制。

22.4.1 单事件模式

TMRxHLT 寄存器的 MODE<3> 位控制定时器是处于单事件模式还是最初的正常周期模式。当该位置 1 时，定时器以单事件模式工作，这意味着在定时器寄存器与 PRx 周期寄存器匹配时，定时器将停止递增，直到重新手动启动定时器为止。

22.5 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。
- ON 和 Timer2_ers 进行了一定程度的一般化，时钟同步延时产生的结果可能与图中所示略有不同。
- 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 23.4 节“CCP/PWM 时钟选择”所述。这些信号不属于 Timer2 模块的一部分。

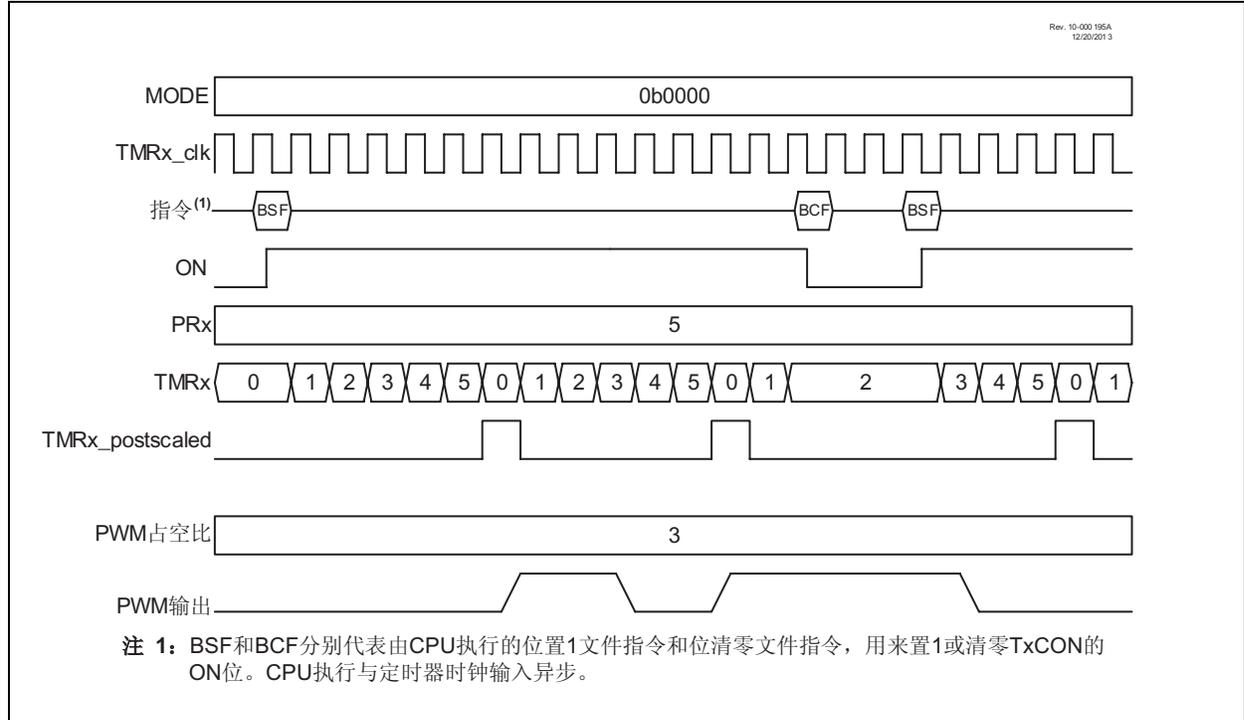
PIC12(L)F1612/16(L)F1613

22.5.1 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 22-4 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 22-4: TIMER2 模式 = 0000 时的时序图



PIC12(L)F1612/16(L)F1613

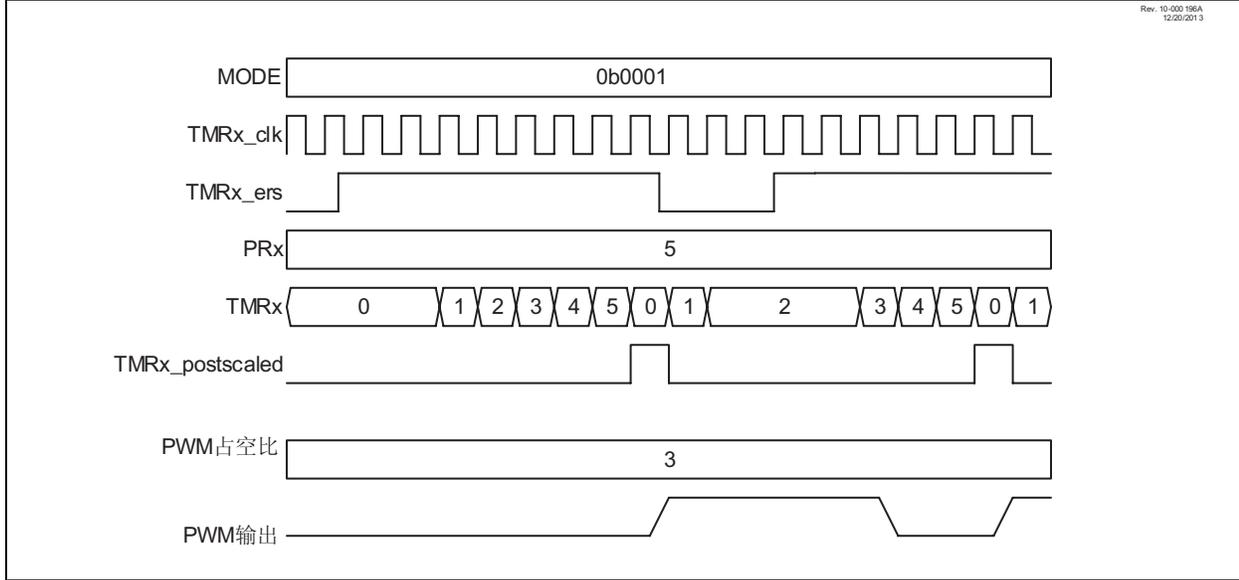
22.5.2 硬件门控模式

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<3:0> = 0001 时，定时器在外部信号为高电平时停止。当 MODE<3:0> = 0010 时，定时器在外部信号为低电平时停止。

图22-5显示了MODE<3:0> = 0001时的硬件门控模式，在该模式下输入高电平会停止计数器。

图 22-5: TIMER2 模式 = 0001 时的时序图



PIC12(L)F1612/16(L)F1613

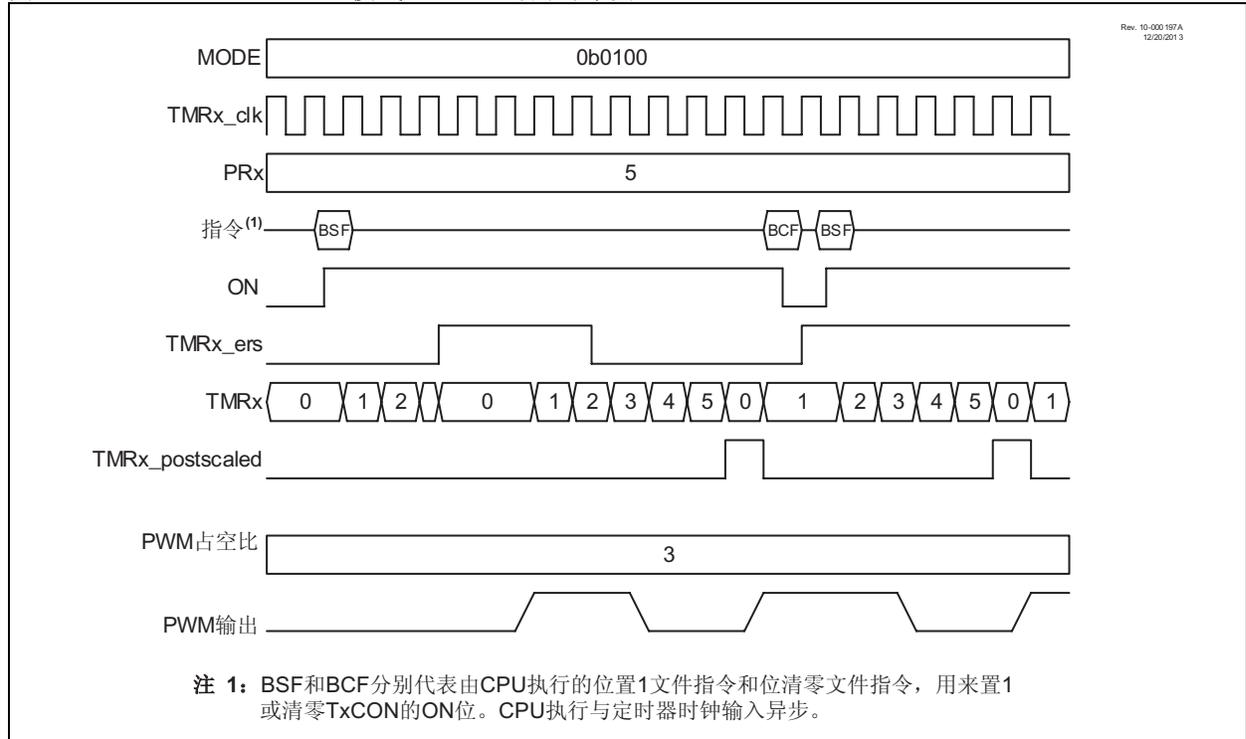
22.5.3 边沿触发硬件限制模式

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<3:0> = 0011)
- 在上升沿复位 (MODE<3:0> = 0100)
- 在下降沿复位 (MODE<3:0> = 0101)

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 22-6。

图 22-6: TIMER2 模式 = 0100 时的时序图



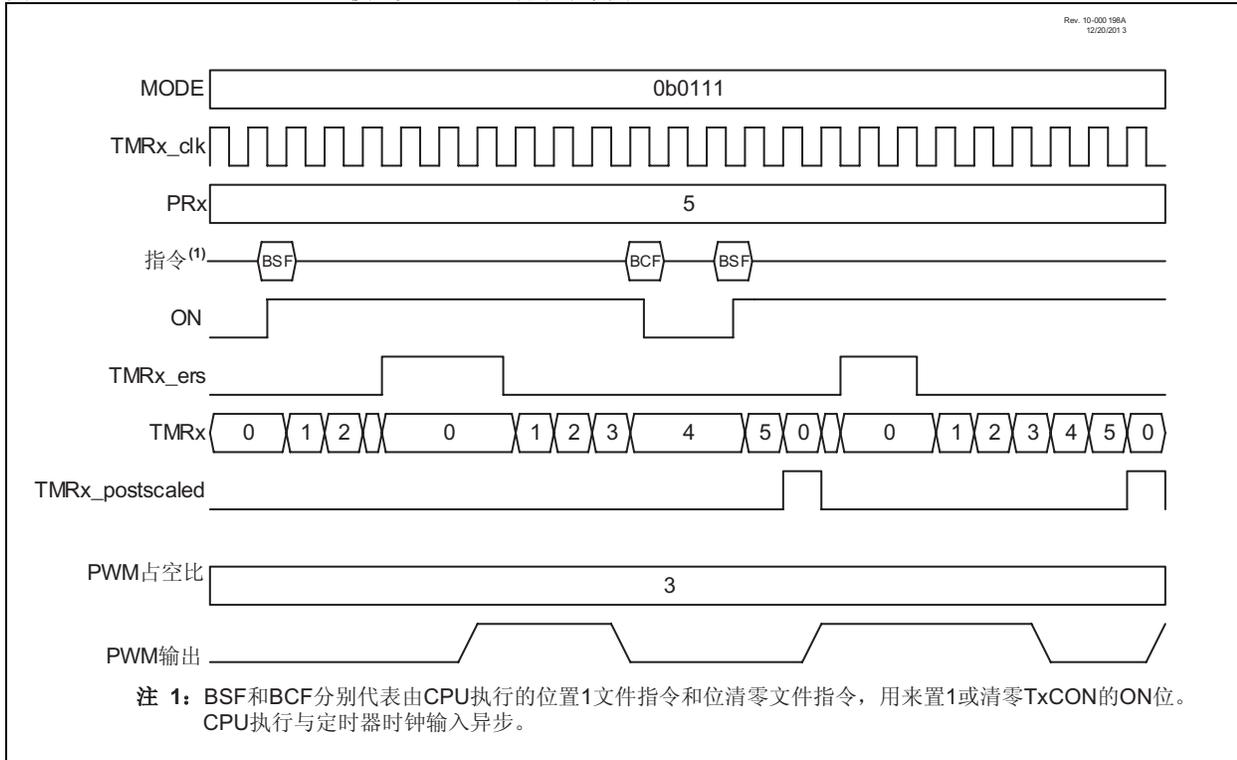
PIC12(L)F1612/16(L)F1613

22.5.4 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 22-7 所示。选择 MODE<3:0> = 0110 会导致定时器在出现低电平外部信号时复位。选择 MODE<3:0> = 0111 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，外部信号复位会在两个时钟的同步延时之后或定时器与 PRx 周期值匹配时将 PWM 输出设置为高电平。PWM 输出将一直保持高电平，直到外部信号被释放和定时器计数至与 CCPRx 脉冲宽度值匹配为止。

图 22-7: TIMER2 模式 = 0111 时的时序图



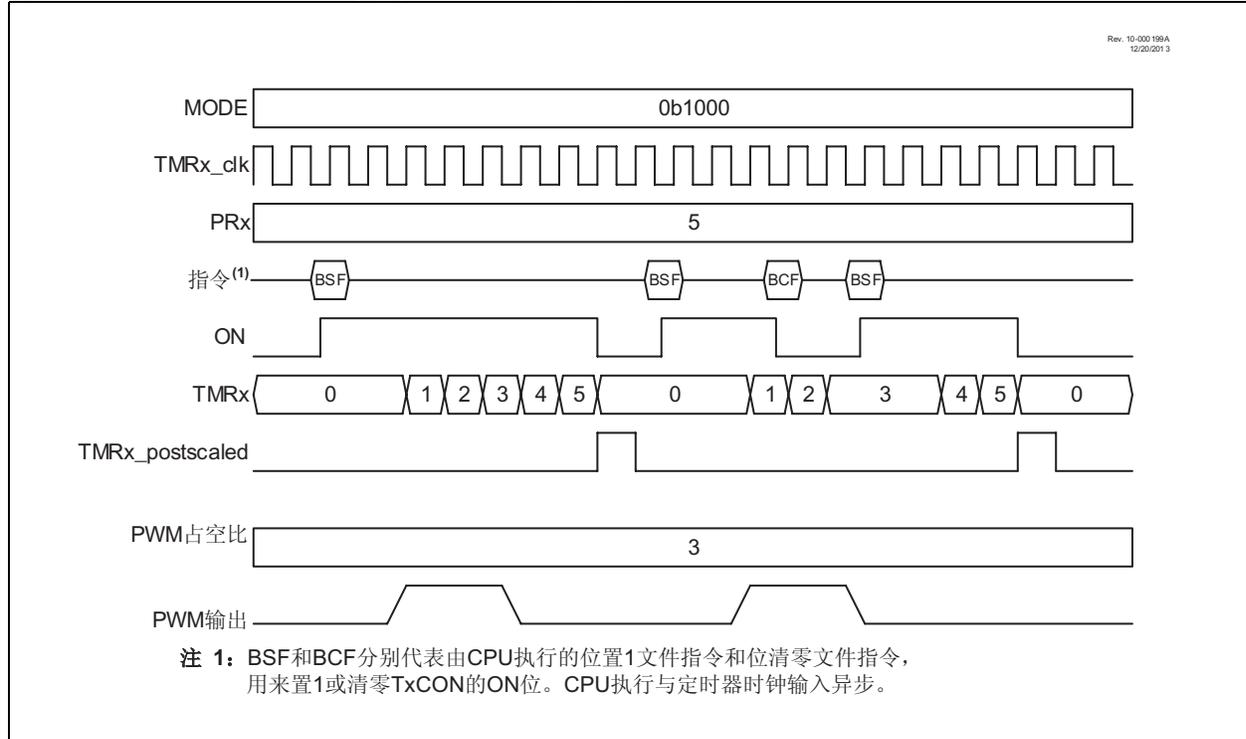
PIC12(L)F1612/16(L)F1613

22.5.5 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<3:0> = 1000 将选择单事件模式，如图 22-8 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 22-8: TIMER2 模式 = 1000 时的时序图



PIC12(L)F1612/16(L)F1613

22.5.6 边沿触发单事件模式

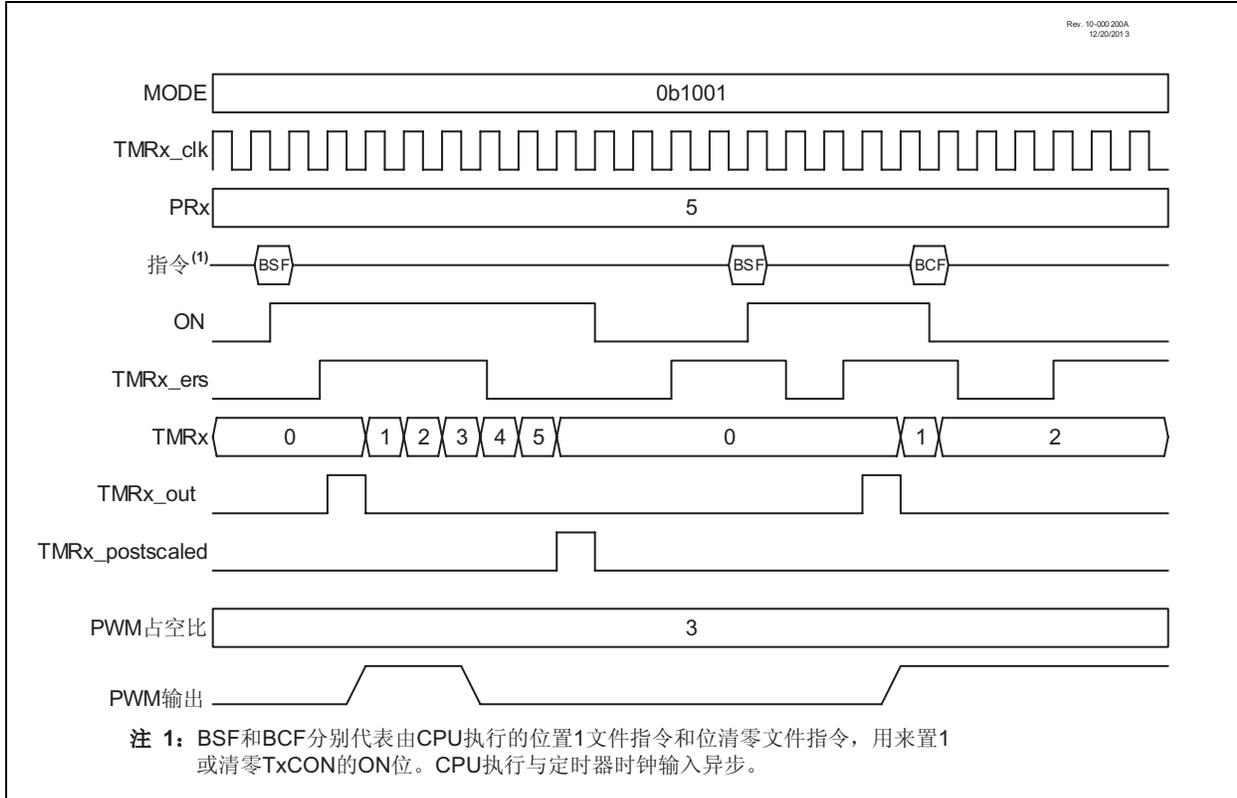
在 ON 位置 1 之后，边沿触发单事件模式会在外部信号输入出现边沿时启动定时器，在定时器与 PRx 周期值匹配时清零 ON 位。以下边沿可以启动定时器：

- 上升沿 (MODE<3:0> = 1001)
- 下降沿 (MODE<3:0> = 1010)
- 上升沿或下降沿 (MODE<3:0> = 1011)

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 22-9 显示了上升沿单事件模式下的操作。

当该模式与 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 22-9: TIMER2 模式 = 1001 时的时序图



22.5.7 边沿触发硬件限制单事件模式

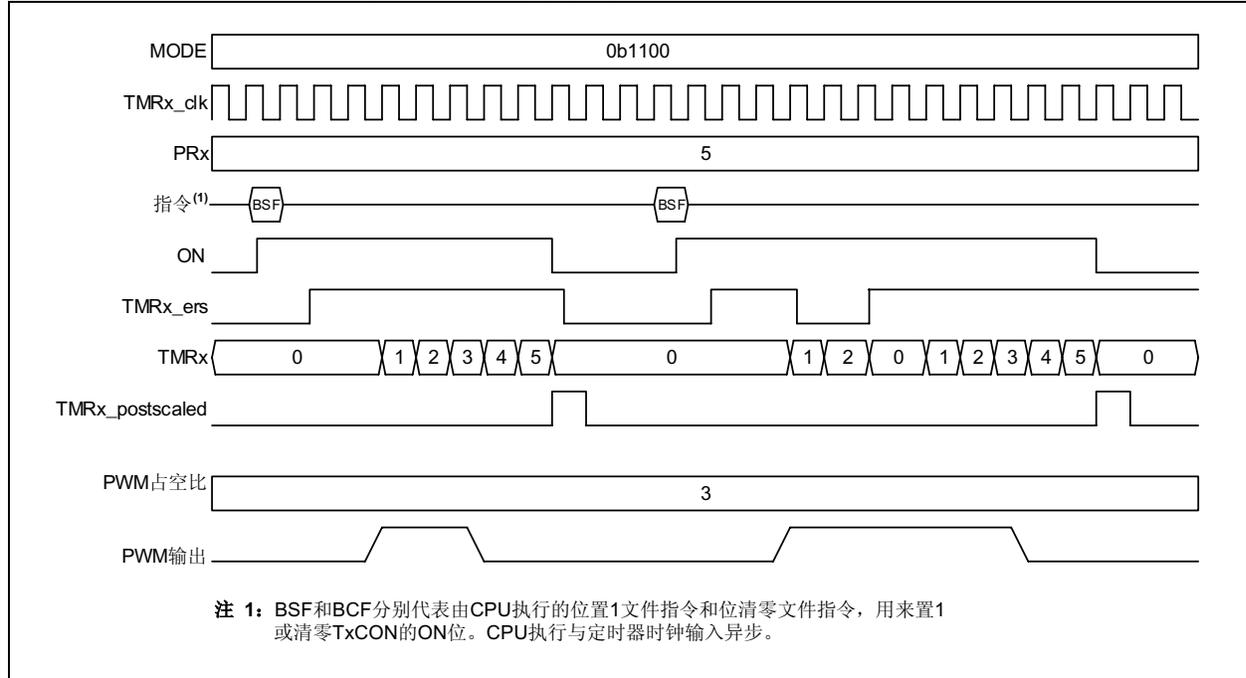
在边沿触发硬件限制单事件模式下，定时器在 ON 位置 1 之后出现第一个外部信号边沿时启动，并在出现所有后续边沿时复位。在 ON 位置 1 后，只需通过第一个边沿来启动定时器。计数器将在所有后续外部复位边沿的两个时钟后自动恢复计数。边沿触发信号如下：

- 上升沿启动和复位 (MODE<3:0> = 1100)
- 下降沿启动和复位 (MODE<3:0> = 1101)

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 22-10 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，第一个启动边沿触发器和所有后续复位边沿都会激活 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 22-10: TIMER2 模式 = 1100 时的时序图



PIC12(L)F1612/16(L)F1613

22.5.8 电平触发硬件限制单事件模式

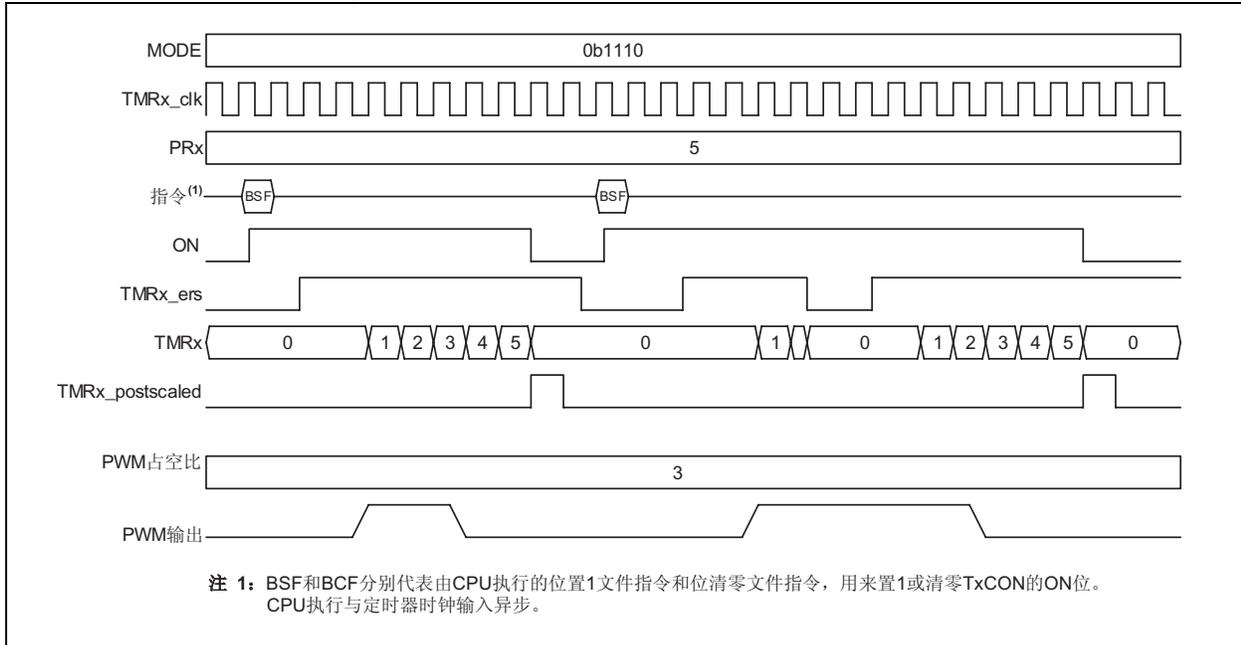
在电平触发单事件模式下，定时器计数在出现外部信号电平时复位，并在外部信号电平释放复位时开始计数。复位电平可选择如下：

- 高复位电平（MODE<3:0> = 1110）
- 低复位电平（MODE<3:0> = 1111）

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驱动不会有效。

图 22-11: TIMER2 模式 = 1110 时的时序图



22.6 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PR2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。

PIC12(L)F1612/16(L)F1613

22.7 寄存器定义: Timer2/4/6 控制

寄存器 22-1: **TxCLKCON: TIMERx 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
—	—	—	—	—	TxCS<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 **TxCS:** Timerx 时钟选择位

111 = 保留

110 = TxIN

101 = MFINTOSC 31.25 kHz

100 = ZCD_output

011 = LFINTOSC

010 = HFINTOSC 16 MHz

001 = FOSC

000 = FOSC/4

PIC12(L)F1612/16(L)F1613

寄存器 22-2: TxCON: TIMERx 控制寄存器

R/W/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ON ⁽¹⁾	CKPS<2:0>			OUTPS<3:0>			
bit 7							bit 0

图注:

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

bit 7 **ON:** Timerx 使能位

- 1 = 开启 Timerx
- 0 = 关闭 Timerx: 复位所有计数器和状态机

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

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

bit 3-0 **OUTPS<3:0>:** Timerx 输出后分频比选择位

- 1111 = 1:16 后分频比
- 1110 = 1:15 后分频比
- 1101 = 1:14 后分频比
- 1100 = 1:13 后分频比
- 1011 = 1:12 后分频比
- 1010 = 1:11 后分频比
- 1001 = 1:10 后分频比
- 1000 = 1:9 后分频比
- 0111 = 1:8 后分频比
- 0110 = 1:7 后分频比
- 0101 = 1:6 后分频比
- 0100 = 1:5 后分频比
- 0011 = 1:4 后分频比
- 0010 = 1:3 后分频比
- 0001 = 1:2 后分频比
- 0000 = 1:1 后分频比

注 1: 在某些模式下, ON 位将由硬件自动清零。请参见第 22.4.1 节“单事件模式”。

PIC12(L)F1612/16(L)F1613

寄存器 22-3: TxHLT: TIMERx 时钟选择寄存器

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
PSYNC ^(1, 2)	CKPOL ⁽³⁾	CKSYNC ^(4, 5)	—	MODE<3:0> ^(6, 7, 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 **PSYNC:** Timerx 预分频器同步使能位 ^(1, 2)
 1 = TMRx 预分频器输出与 Fosc/4 进行同步
 0 = TMRx 预分频器输出不与 Fosc/4 进行同步
- bit 6 **CKPOL:** Timerx 时钟极性选择位 ⁽³⁾
 1 = 定时器 / 预分频器基于输入时钟的下降沿进行计数
 0 = 定时器 / 预分频器基于输入时钟的上升沿进行计数
- bit 5 **CKSYNC:** Timerx 时钟同步使能位 ^(4, 5)
 1 = ON 寄存器位与 TMR2_clk 输入进行同步
 0 = ON 寄存器位不与 TMR2_clk 输入进行同步
- bit 4 **未实现:** 读为 0
- bit 3-0 **MODE<3:0>:** Timerx 控制模式选择位 ^(6, 7, 8)
MODE<3> = 1 的所有模式均为“单事件”模式 (ON 位清零, 时钟停止, TMRx 在 TMRx = PRx 后立即清零)。只有当软件将 ON 位置 1 时, 才会使能下一个单事件操作。ON 位不会受 TMRx_ers 影响。
 1111 = 在 TMRx_ers 上检测到下降沿时, 定时器启动; 在 TMRx_ers = 1 时, 定时器复位
 1110 = 在 TMRx_ers 上检测到上升沿时, 定时器启动; 在 TMRx_ers = 0 时, 定时器复位
 1101 = 定时器在出现第一个 TMRx_ers 下降沿时启动, 在出现所有后续 TMRx_ers 下降沿时重新启动
 1100 = 定时器在出现第一个 TMRx_ers 上升沿时启动, 在出现所有后续 TMRx_ers 上升沿时复位
 1011 = 在 TMRx_ers 上检测到上升沿或下降沿时, 定时器启动
 1010 = 在 TMRx_ers 上检测到下降沿时, 定时器启动
 1001 = 在 TMRx_ers 上检测到上升沿时, 定时器启动
 1000 = 定时器在 ON = 1 时立即启动 (软件控制)
MODE<3> = 0 的所有模式均为计满返回脉冲模式 (在 TMRx = PRx 时清零 TMRx, 然后继续运行)。ON 位必须置 1, 它不受复位影响。
 0111 = 定时器在 TMRx_ers = 1 时复位
 0110 = 定时器在 TMRx_ers = 0 时复位
 0101 = 定时器在 TMRx_ers 的下降沿复位
 0100 = 定时器在 TMRx_ers 的上升沿复位
 0011 = 定时器在 TMRx_ers 的上升沿或下降沿复位
 0010 = 定时器在 ON = 1 且 TMRx_ers = 0 时启动。在 TMRx_ers = 1 时停止
 0001 = 定时器在 ON = 1 且 TMRx_ers = 1 时启动。在 TMRx_ers = 0 时停止
 0000 = 定时器在 ON = 1 时立即启动 (软件控制)

- 注**
- 1: 置 1 该位可确保读取 TMRx 将返回一个有效的数据值。
 - 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: 置 1 该位将“消耗”两个 TMRx 输入时钟来使能定时器。
 - 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。
 - 8: 在边沿触发“单事件”模式下, 触发启动机制在 ON = 0 时复位并重新激活; 直到出现一个输入边沿时, 计数器才会重新启动。

PIC12(L)F1612/16(L)F1613

寄存器 22-4: T2RST: TIMER2 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<3:0>**: Timer2 外部复位信号源选择位

- 1111 = 保留
- 1110 = 保留
- 1101 = 保留
- 1100 = CWGD
- 1011 = CWGC
- 1010 = CWGB
- 1001 = CWGA
- 1000 = ZCD1_output
- 0111 = TMR6_postscaled
- 0110 = TMR4_postscaled
- 0101 = 保留
- 0100 = CCP2_out
- 0011 = CCP1_out
- 0010 = C2_OUT_sync⁽¹⁾
- 0001 = C1_OUT_sync
- 0000 = T2IN

注 1: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

寄存器 22-5: T4RST: TIMER4 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<3:0>:** Timer4 外部复位信号源选择位

- 1111 = 保留
- 1110 = 保留
- 1101 = 保留
- 1100 = CWGD
- 1011 = CWGC
- 1010 = CWGB
- 1001 = CWGA
- 1000 = ZCD1_output
- 0111 = TMR6_postscaled
- 0110 = 保留
- 0101 = TMR4_postscaled
- 0100 = CCP2_out
- 0011 = CCP1_out
- 0010 = C2_OUT_sync⁽¹⁾
- 0001 = C1_OUT_sync
- 0000 = T4IN

注 1: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

寄存器 22-6: T6RST: TIMER6 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<3:0>**: Timer6 外部复位信号源选择位

1111 = 保留
1110 = 保留
1101 = 保留
1100 = CWGD
1011 = CWGC
1010 = CWGB
1001 = CWGA
1000 = ZCD1_output
0111 = 保留
0110 = TMR4_postscaled
0101 = TMR2_postscaled
0100 = CCP2_out
0011 = CCP1_out
0010 = C2_OUT_sync⁽¹⁾
0001 = C1_OUT_sync
0000 = T6IN

注 1: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

表 22-1: 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	OE	OUT	FMT	CCP1M<3:0>				223
CCP2CON	EN	OE	OUT	FMT	CCP2M<3:0>				223
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	79
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	83
PR2	Timer2 模块周期寄存器								197*
TMR2	8 位 TMR2 寄存器的保持寄存器								197*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				208
T2CLKCON	—	—	—	—	—	T2CS<2:0>			207
T2RST	—	—	—	—	RSEL<3:0>				210
T2HLT	PSYNC	CKPOL	CKSYNC	—	MODE<3:0>				209
PR4	Timer4 模块周期寄存器								197*
TMR4	8 位 TMR4 寄存器的保持寄存器								197*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				208
T4CLKCON	—	—	—	—	—	T4CS<2:0>			207
T4RST	—	—	—	—	RSEL<3:0>				211
T4HLT	PSYNC	CKPOL	CKSYNC	—	MODE<3:0>				209
PR6	Timer6 模块周期寄存器								197*
TMR6	8 位 TMR6 寄存器的保持寄存器								197*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				208
T6CLKCON	—	—	—	—	—	T6CS<2:0>			207
T6RST	—	—	—	—	RSEL<3:0>				212
T6HLT	PSYNC	CKPOL	CKSYNC	—	MODE<3:0>				209

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。

PIC12(L)F1612/16(L)F1613

23.0 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。

23.1 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）
（仅限 PIC16(L)F1613）
- 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 23-1 给出了捕捉操作的简化框图。

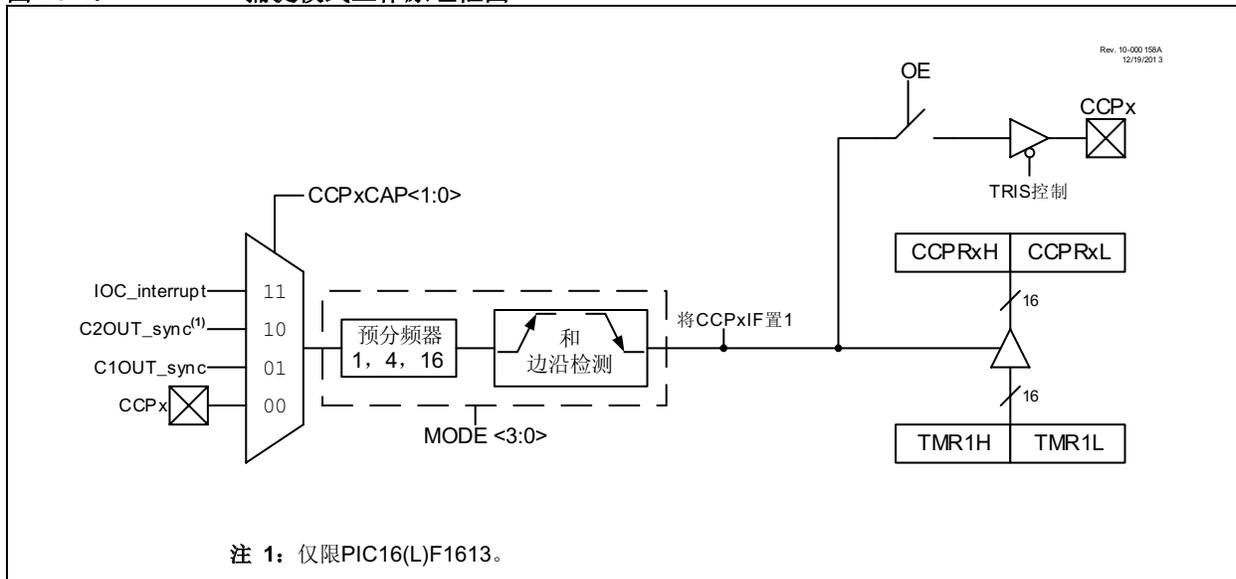
23.1.1 CCP 引脚配置

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。

此外，还可以通过使用 APFCON 寄存器将 CCP2 引脚功能转移到备用引脚上。更多详细信息，请参见第 12.1 节“备用引脚功能”。

注： 如果 CCPx 引脚被配置为输出，则写端口将产生一次捕捉条件。

图 23-1: 捕捉模式工作原理框图



23.1.2 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 21.0 节“带门控控制的 Timer1 模块”。

23.1.3 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 PIRx 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。

23.1.4 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。

23.1.5 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。

23.1.6 备用引脚位置

该模块具有以下 I/O 引脚：通过使用备用引脚功能寄存器 APFCON，可将 I/O 引脚转移到其他位置。要确定哪些引脚可转移以及其在复位时的默认位置，请参见第 12.1 节“备用引脚功能”了解更多信息。

23.1.7 捕捉输出

每当发生捕捉时，CCP 的输出会变为高电平，并保持 1 个系统时钟周期（1/Fosc）。该输出可用作 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，以及用作 SMT 的窗口输入。此外，如果 CCPx 引脚不用作捕捉的输入，可以通过 CCPxCON 寄存器的 OE 位让 CCP 的输出控制 CCPx 引脚。

PIC12(L)F1612/16(L)F1613

23.2 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 23-2 给出了比较操作的简化框图。

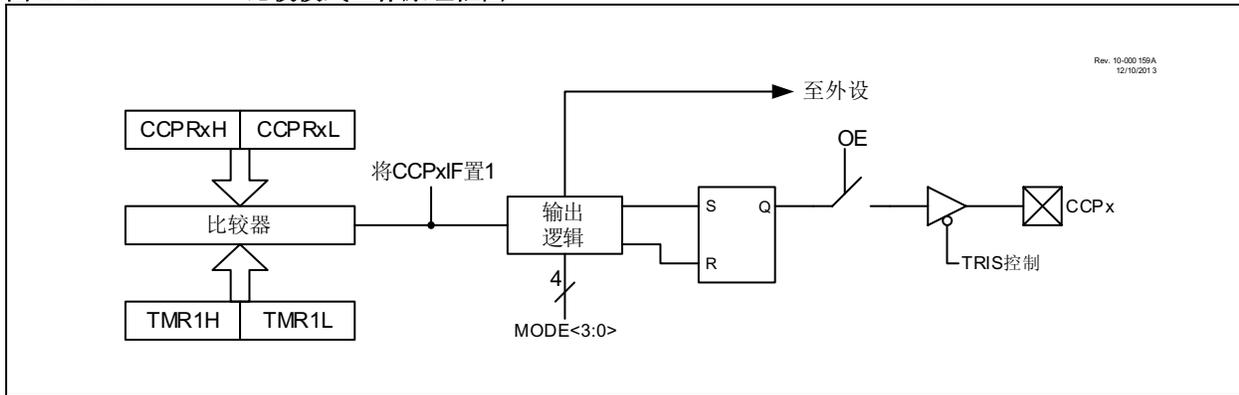
23.2.1 CCPx 引脚配置

用户必须通过将相关的 TRIS 位清零，将 CCPx 引脚配置为输出。

此外，还可以通过使用 APFCON 寄存器（寄存器 12-1）将 CCPx 引脚功能转移到备用引脚上。更多详细信息，请参见第 12.1 节“备用引脚功能”。

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

图 23-2: 比较模式工作原理框图



23.2.2 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 21.0 节“带门控控制的 Timer1 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。

23.2.3 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）。

23.2.4 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。

23.2.5 备用引脚位置

该模块具有以下 I/O 引脚：通过使用备用引脚功能寄存器 APFCON，可将 I/O 引脚转移到其他位置。要确定哪些引脚可转移以及其在复位时的默认位置，请参见第 12.1 节“备用引脚功能”了解更多信息。

23.2.6 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 被配置为哪种比较模式。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，以及用作 SMT 的窗口输入。此外，通过设置 CCPxCON 寄存器的 OE 位，可以让 CCP 的输出控制 CCPx 引脚。

PIC12(L)F1612/16(L)F1613

23.3 PWM 概述

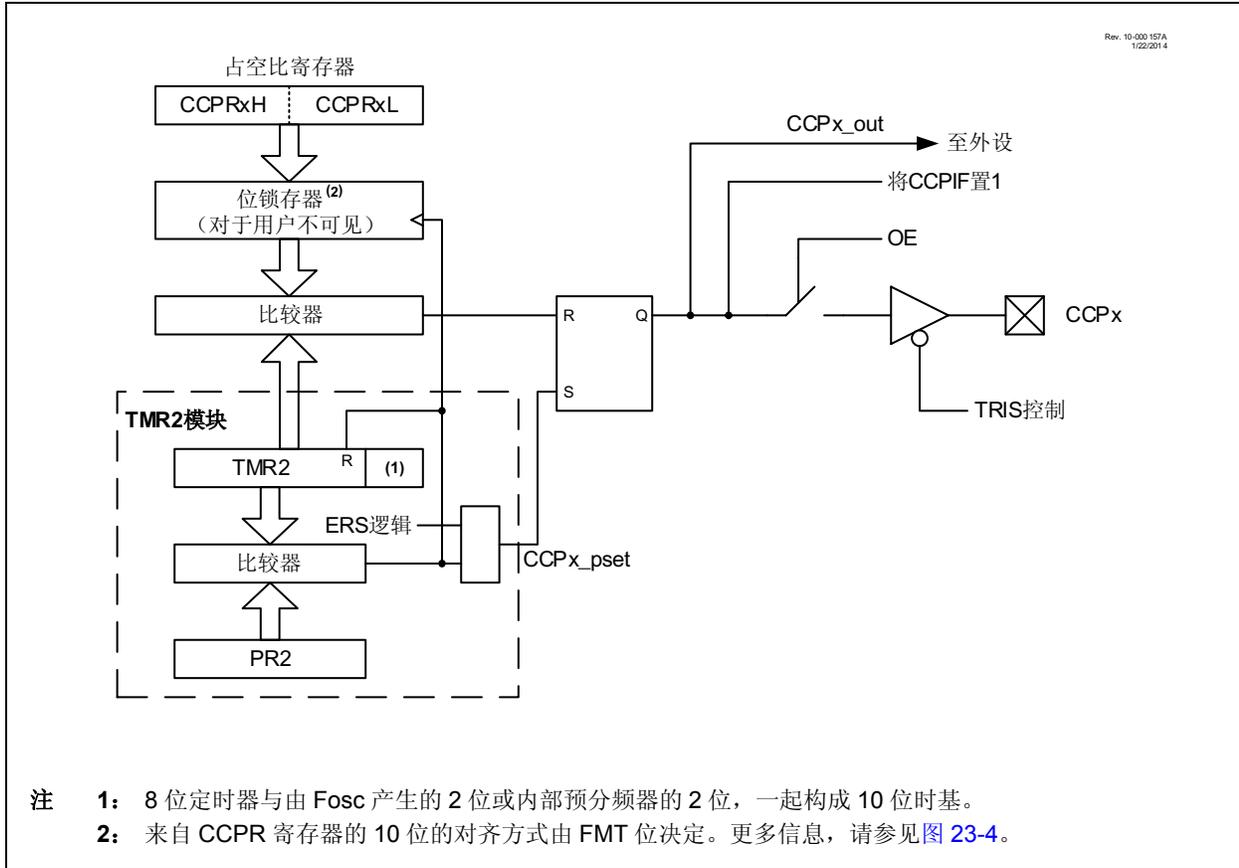
脉宽调制 (PWM) 是一种通过在完全开启和完全关闭状态之间进行快速切换而为负载供电的方案。PWM 信号类似于方波, 信号的高电平部分视为开启状态, 信号的低电平部分视为关闭状态。高电平部分 (也称为脉宽) 可以随时间而变, 并以步幅为单位进行定义。施加的步幅数量越多 (这会增大脉宽), 为负载提供的电量就越多。施加的步幅数量降低时 (这会缩短脉宽), 提供的电量就会下降。PWM 周期定义为一个完整周期的持续时间, 或者开启和关闭时间相加的总时间。

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 23-3 给出了 PWM 信号的典型波形图。

图 23-3: PWM 的简化框图



23.3.1 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 23-3 给出了 PWM 操作的简化框图。

- | |
|---|
| <p>注 1: 要能使 CCPx 引脚上的 PWM 输出，必须清零相应的 TRIS 位。</p> <p>注 2: 清零 CCPxCON 寄存器会放弃对 CCPx 引脚的控制。</p> |
|---|

23.3.2 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。

- | |
|---|
| <p>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。</p> |
|---|

PIC12(L)F1612/16(L)F1613

23.4 CCP/PWM 时钟选择

PIC12(L)F1612/16(L)F1613 允许每个 CCP 和 PWM 模块选择控制模块的定时源。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。

23.4.1 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 22.5 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。

23.4.2 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 23-1 计算。

公式 23-1: 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 22-1）。

23.4.3 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 23-4 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 23-2 用于计算 PWM 脉冲宽度。公式 23-3 用于计算 PWM 占空比。

公式 23-2: 脉冲宽度

$$脉冲宽度 = CCPRxH:CCPRxL \cdot T_{osc} \cdot (TMR2 \text{ 预分频值})$$

公式 23-3: 占空比

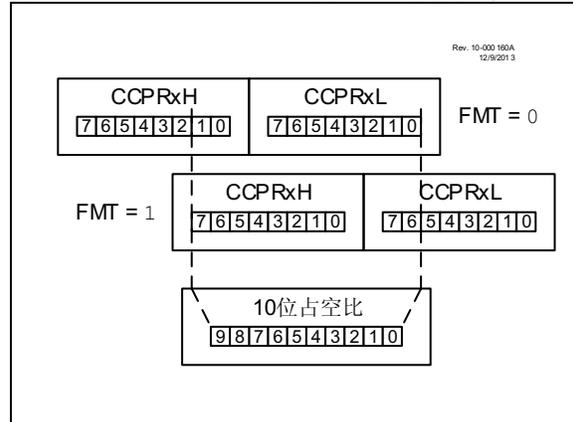
$$占空比 = \frac{CCPRxH:CCPRxL}{4(PR_x + 1)}$$

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 23-3）。

图 23-4: CCPx 占空比对齐方式



23.4.4 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 23-4 所示。

公式 23-4: 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。

PIC12(L)F1612/16(L)F1613

表 23-1: 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 23-2: 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

23.4.5 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”。

23.4.6 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。

23.4.7 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，以及用作 SMT 的窗口输入。此外，通过设置 CCPxCON 寄存器的 OE 位，可以让 CCP 的输出控制 CCPx 引脚。

PIC12(L)F1612/16(L)F1613

23.5 寄存器定义：CCP 控制

寄存器 23-1: CCPxCON: CCPx 控制寄存器

R/W-0/0	R/W-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	OE	OUT	FMT	MODE<3:0>			
bit 7							bit 0

图注:

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

- bit 7 **EN:** CCPx 模块使能位
 1 = 使能 CCPx
 0 = 禁止 CCPx
- bit 6 **OE:** CCPx 输出使能位
 1 = 使能 CCPx 端口引脚输出
 0 = 禁止 CCPx 端口引脚输出
- bit 5 **OUT:** CCPx 输出数据位 (只读)
- bit 4 **FMT:** 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 **MODE:** CCPx 模式选择位
 11xx = PWM 模式
- 1011 = 比较模式: 脉冲输出, 清零 TMR1
 1010 = 比较模式: 脉冲输出 (0 - 1 - 0)
 1001 = 比较模式: 发生比较匹配时清零输出
 1000 = 比较模式: 发生比较匹配时置 1 输出
- 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)

PIC12(L)F1612/16(L)F1613

寄存器 23-2: CCPTMRS: PWM 定时器选择控制寄存器 0

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	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 7-4 **未实现:** 读为 0

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

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

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

PIC12(L)F1612/16(L)F1613

寄存器 23-3: 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

寄存器 23-4: CCPRxH: CCPx 高字节寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPR<15:8>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

MODE = 捕捉模式

CCPRxH<7:0>: 捕捉的 TMR1 值的 MSB

MODE = 比较模式

CCPRxH<7:0>: MSB 与 TMR1 值进行比较

MODE = PWM 模式且 FMT = 0

CCPRxH<7:2>: 不使用

CCPRxH<1:0>: CCPW<9:8>——脉冲宽度高 2 位

MODE = PWM 模式且 FMT = 1

CCPRxH<7:0>: CCPW<9:2>——脉冲宽度高 8 位

PIC12(L)F1612/16(L)F1613

寄存器 23-5: CCPxCAP: CCPx 捕捉输入选择寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	CTS<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 7-2 未实现: 读为 0

bit 1-0 **CTS<1:0>**: 捕捉触发输入选择位

11 = IOC_interrupt

10 = C2_OUT_sync⁽¹⁾

01 = C1_OUT_sync

00 = CCPx 引脚

注 1: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

表 23-3: 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
APFCON	—	CWGASEL ⁽²⁾	CWGBSEL ⁽²⁾	—	T1GSEL	—	CCP2SEL ⁽³⁾	CCP1SEL ⁽²⁾	128
CCPxCAP	—	—	—	—	—	—	CTS<1:0>		226
CCPxCON	EN	OE	OUT	FMT	MODE<3:0>				223
CCPRxL	捕捉 / 比较 / PWM 寄存器 x (LSB)								225
CCPRxH	捕捉 / 比较 / PWM 寄存器 x (MSB)								225
CCPTMRS	—	—	—	—	C2TSEL<1:0>		C1TSEL<1:0>		224
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	78
PIE1	TMR1GIE	ADIE	—	—	—	CCP1IE	TMR2IE	TMR1IE	79
PIE2	—	C2IE ⁽³⁾	C1IE	—	—	TMR6IE	TMR4IE	CCP2IE	80
PIR1	TMR1GIF	ADIF	—	—	—	CCP1IF	TMR2IF	TMR1IF	83
PIR2	—	C2IF ⁽³⁾	C1IF	—	—	TMR6IF	TMR4IF	CCP2IF	84
PR2	Timer2 周期寄存器								197*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				208
TMR2	Timer2 模块寄存器								197
PR4	Timer4 周期寄存器								197*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				208
TMR4	Timer4 模块寄存器								197
PR6	Timer6 周期寄存器								197*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				208
TMR6	Timer6 模块寄存器								197
TRISA	—	—	TRISA5	TRISA4	— ⁽²⁾	TRISA2	TRISA1	TRISA0	131

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。
2: 仅限 PIC12(L)F1612。
3: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

24.0 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 (仅限 PIC16(L)F1613)
 - 全桥模式, 反向 (仅限 PIC16(L)F1613)
 - 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

24.1 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 24-9)
- 推挽模式 (图 24-1)
 - 全桥模式, 正向 (图 24-2) (仅限 PIC16(L)F1613)
 - 全桥模式, 反向 (图 24-2) (仅限 PIC16(L)F1613)
- 转向模式 (图 24-4)
- 同步转向模式 (图 24-4)

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 24.10 节 “自动关断” 对它进行了介绍。

24.1.1 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 24-9 所示。在两个输出之间插入不重叠

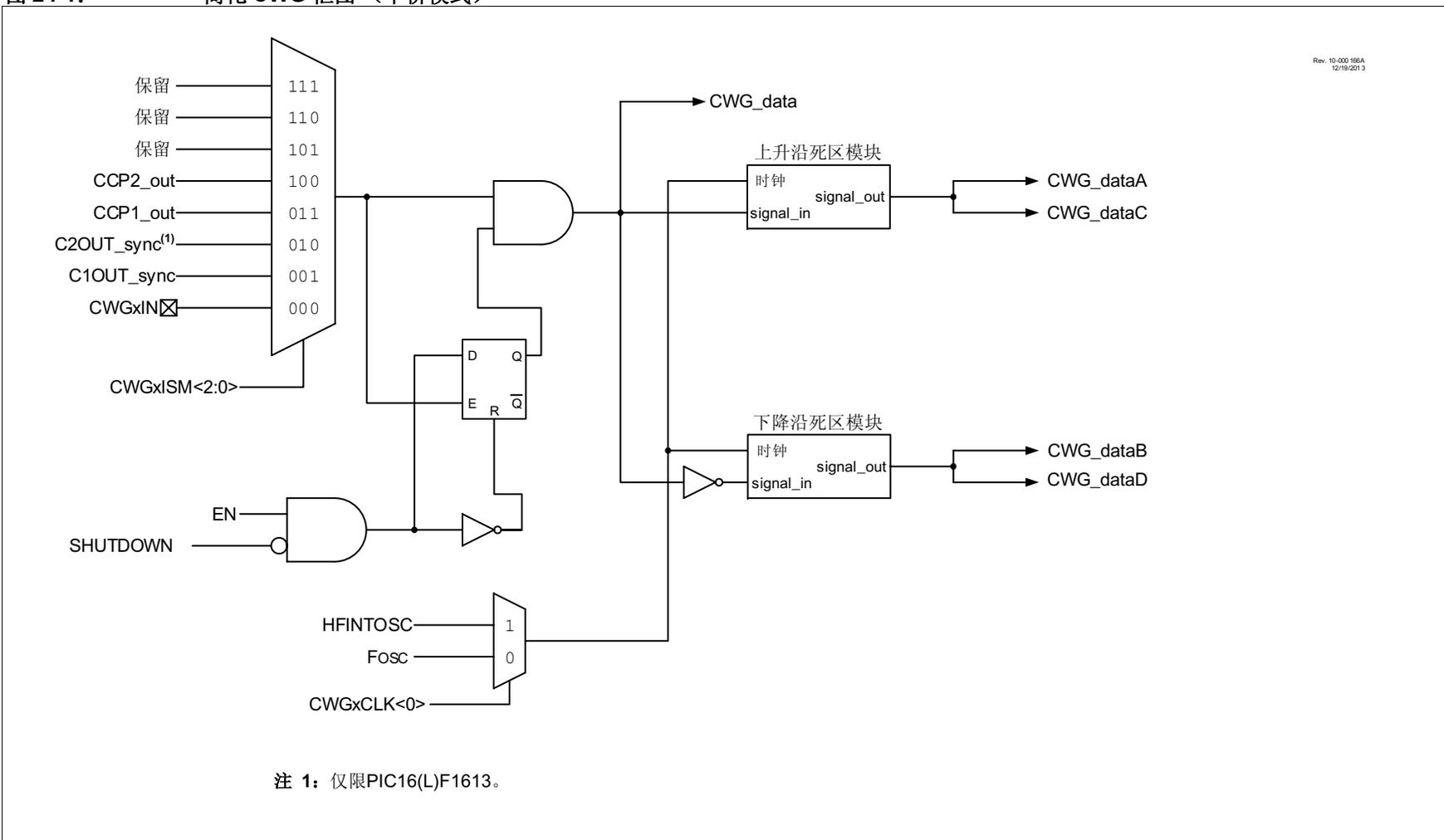
(死区) 时间, 以防止各种电源应用中产生直通电流。

第 24.5 节 “死区控制” 对死区控制进行了介绍。

未用的输出 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. 10-000 165A
12/15/2013

图 24-1: 简化 CWG 框图 (半桥模式)



PIC12(L)F1612/16(L)F1613

24.1.2 推挽模式

在推挽模式下，将生成两个输出信号，它们为输入的交流副本，如图 24-2 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。

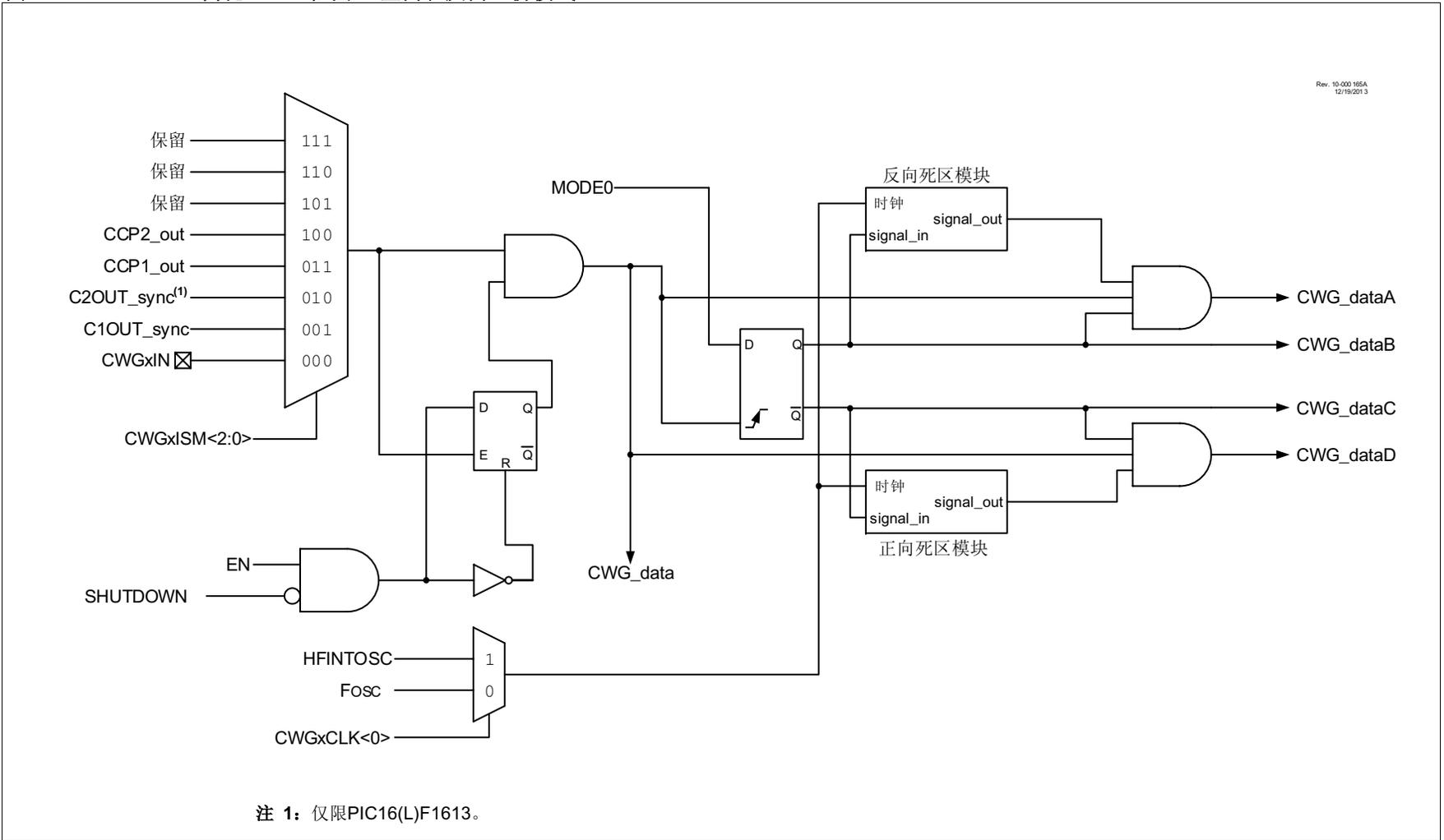
未用的输出 $CWGxC$ 和 $CWGxD$ 分别驱动 $CWGxA$ 和 $CWGxB$ 的副本，但它们的极性分别由 $CWGxCON1$ 寄存器的 $POLC$ 和 $POLD$ 位控制。

24.1.3 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 24.5 节“死区控制”对这种死区控制进行了介绍，第 24.6 节“上升沿和反向死区”和第 24.7 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。

图 24-3: 简化 CWG 框图 (正向和反向全桥模式)

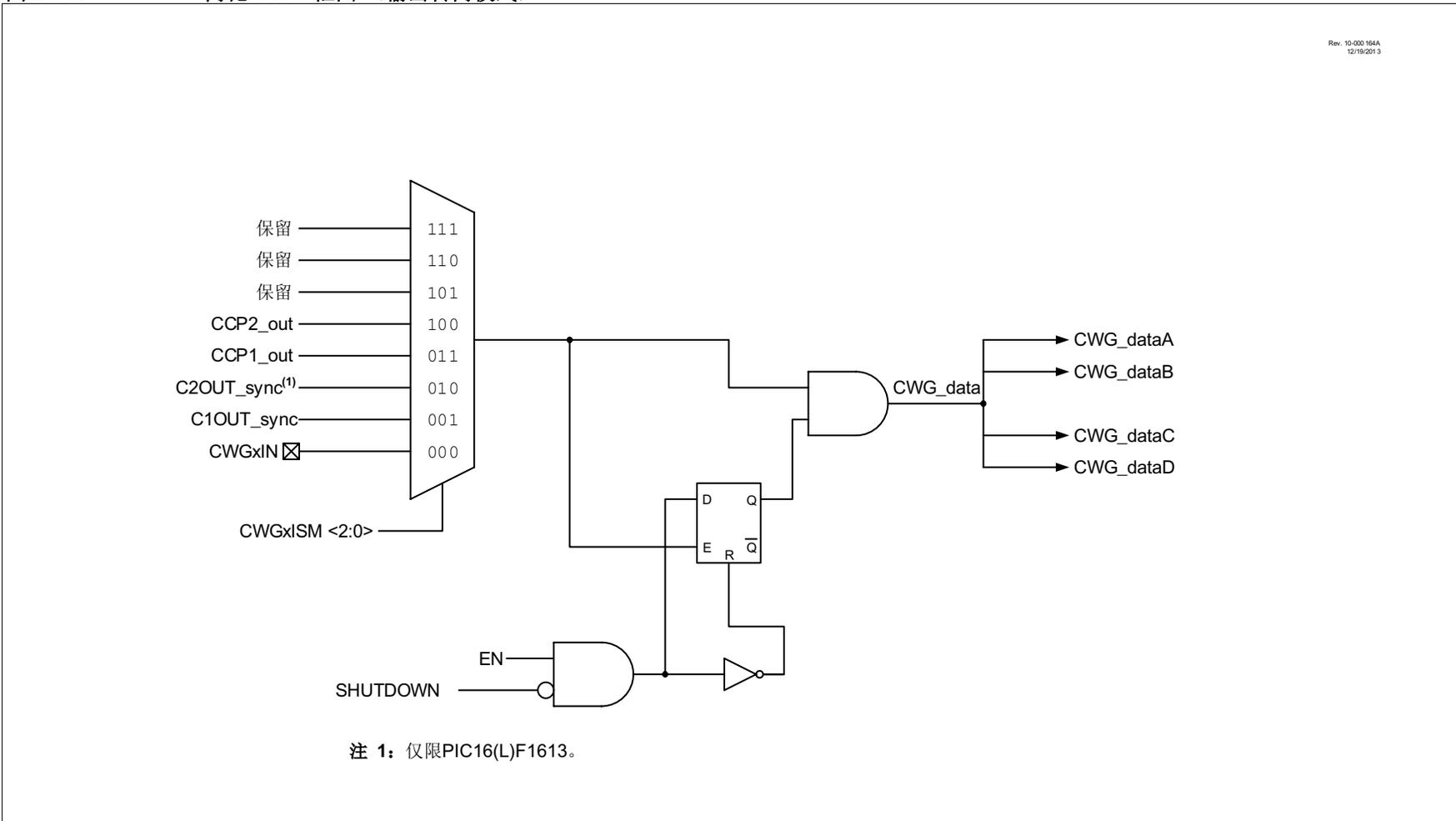


24.1.4 转向模式

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 24.9 节“CWG 转向模式”。

图 24-4: 简化 CWG 框图 (输出转向模式)



24.2 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。

24.3 可选输入源

CWG 基于表 24-1 中的输入源来生成输出波形。

表 24-1: 可选输入源

输入源外设	信号名称
CWG 输入引脚	CWGxIN 引脚
比较器 C1	C1_OUT_sync
比较器 C2 ⁽¹⁾	C2_OUT_sync
CCP1	CCP1_out
CCP2	CCP2_out

注 1: 仅限 PIC16(L)F1613。

输入源使用 CWGxISM 寄存器进行选择。

24.4 输出控制

24.4.1 输出使能

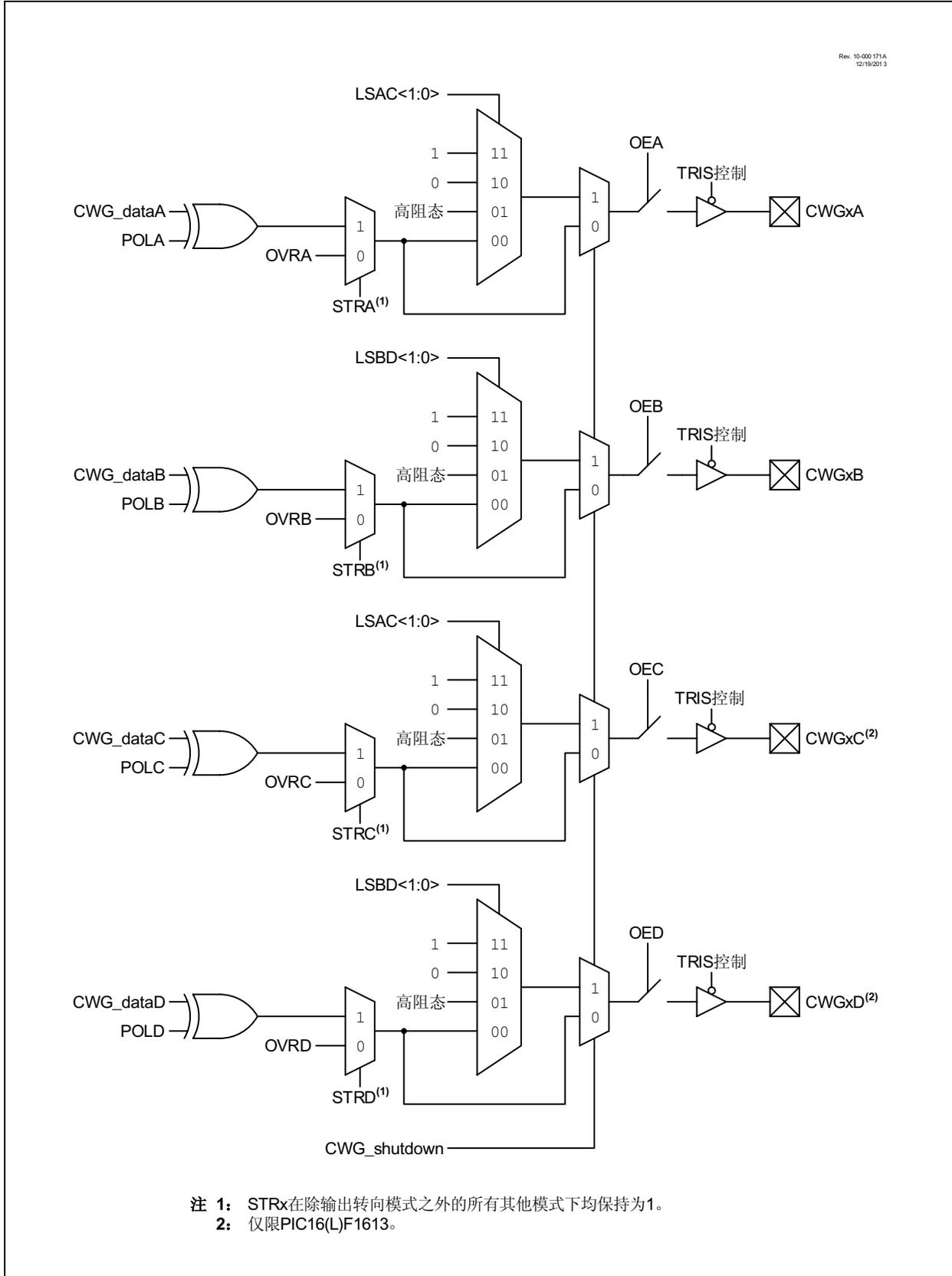
每个 CWG 输出引脚都具有独立的输出使能控制。输出使能使用 GxIOEx<3:0> 位进行选择。当输出使能控制清零时，模块对引脚无控制权。当输出使能置 1 时，每次选择端口极性都将对引脚施加改写值或有效 PWM 波形。输出引脚使能取决于 CWGxCON0 寄存器中的模块使能位 EN。当 EN 清零时，CWG 输出使能和 CWG 驱动电平没有任何作用。

24.4.2 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。

PIC12(L)F1612/16(L)F1613

图 24-5: CWG 输出框图



24.5 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。

24.5.1 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 24-9 中可以看到这一点。

24.5.2 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 24-3 所示。

24.6 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。

24.7 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 24-6 和图 24-7。

图 24-6: 死区操作, CWGxDBR = 0X01, CWGxDBF = 0X02

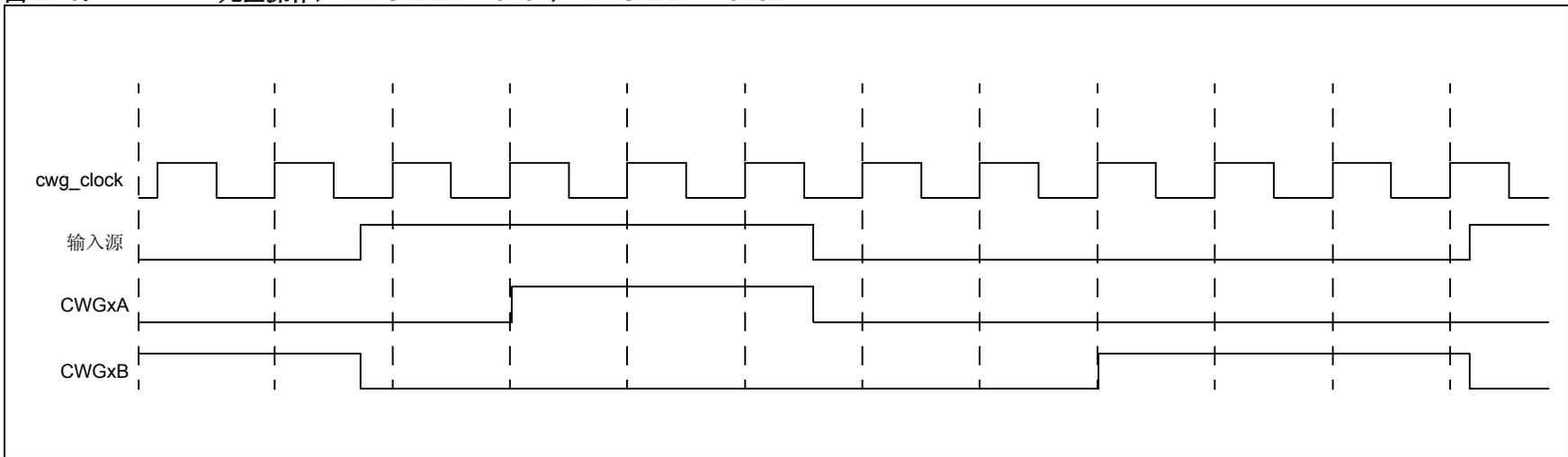
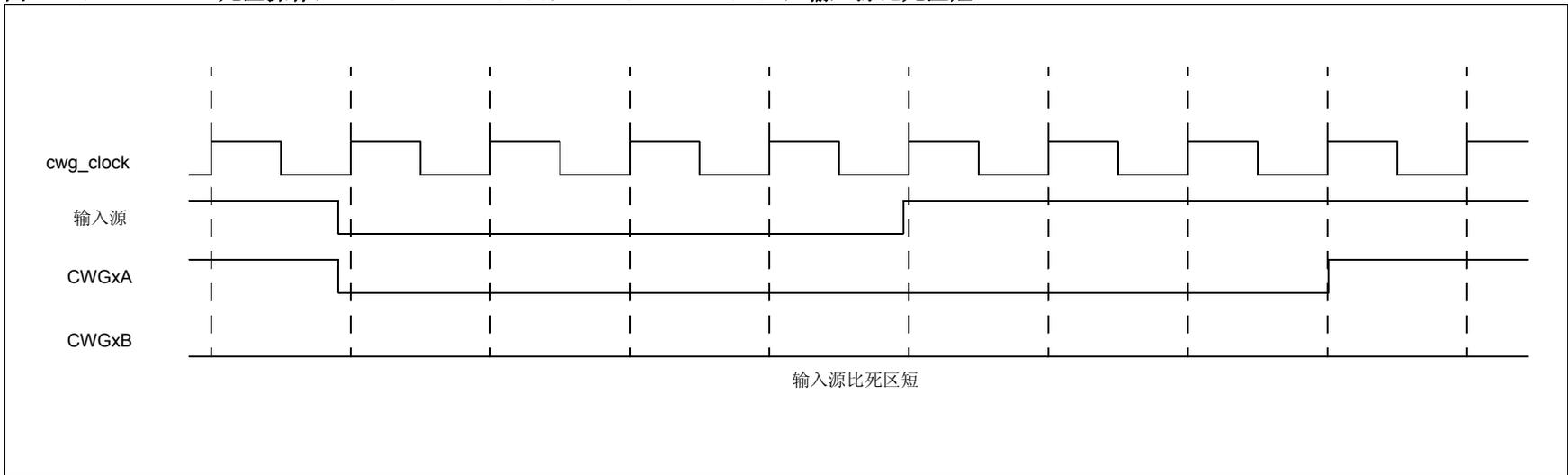


图 24-7: 死区操作, CWGxDBR = 0X03, CWGxDBF = 0X04, 输入源比死区短



PIC12(L)F1612/16(L)F1613

24.8 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。更多详细信息，请参见公式 24-1。

公式 24-1: 死区偏差

$$T_{DEADBAND_UNCERTAINTY} = \frac{1}{F_{cwg_clock}}$$

示例:

$$F_{CWG_CLOCK} = 16 \text{ MHz}$$

因此:

$$\begin{aligned} T_{DEADBAND_UNCERTAINTY} &= \frac{1}{F_{cwg_clock}} \\ &= \frac{1}{16 \text{ MHz}} \\ &= 62.5 \text{ ns} \end{aligned}$$

图 24-8: PWM 方向改变的示例

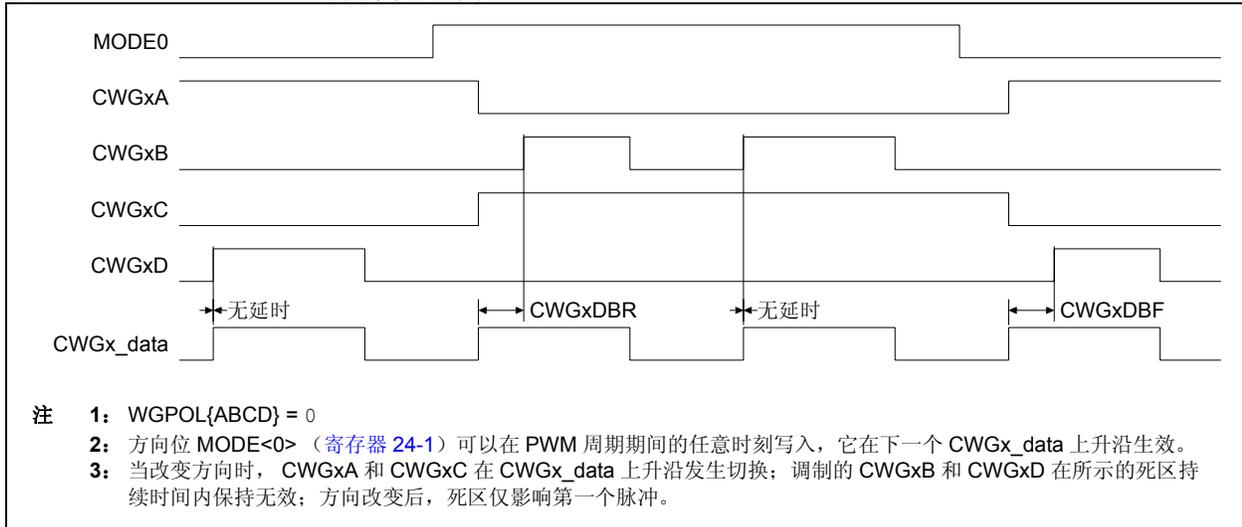
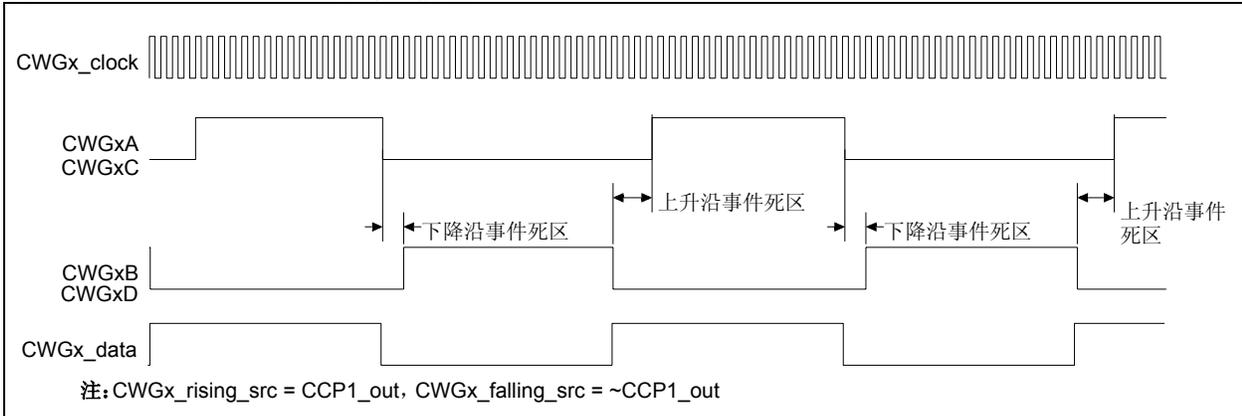


图 24-9: CWG 半桥模式操作



24.9 CWG 转向模式

在转向模式下 ($MODE = 00x$)，CWG 允许使用任意 CWG_{xx} 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 CWG_{xOCON0} 相应的 STR_x 位为 0 时，相应引脚会保持在所定义的电平。当 CWG_{xOCON0} 相应的 STR_x 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。

CWG_{xCON1} 寄存器的 POL_x 位仅在 $STR_x = 1$ 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 24.10 节“自动关断”中所述。自动关断事件只对 $STR_x = 1$ 的引脚有影响。

24.9.1 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 STR_x 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 24-10 和图 24-11 分别给出了异步和同步转向的时序。

图 24-10: 指令结束时发生的转向事件的示例 ($MODE_{<2:0>} = 000$)

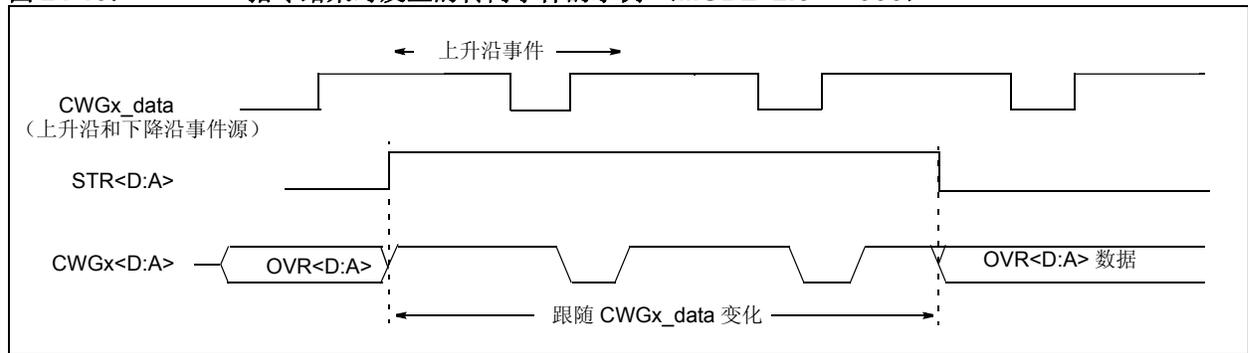
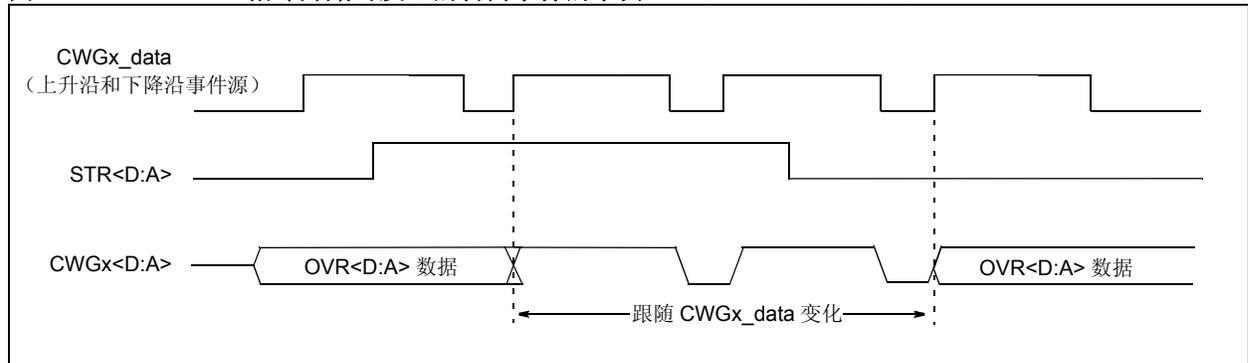


图 24-11: 指令开始时发生的转向事件的示例 ($MODE_{<2:0>} = 001$)



PIC12(L)F1612/16(L)F1613

24.10 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 24-12 给出了自动关断电路的图示。

24.10.1 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

24.10.1.1 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。

24.10.2 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（仅限 PIC16(L)F1613）
- 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 24-6）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。

24.11 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：

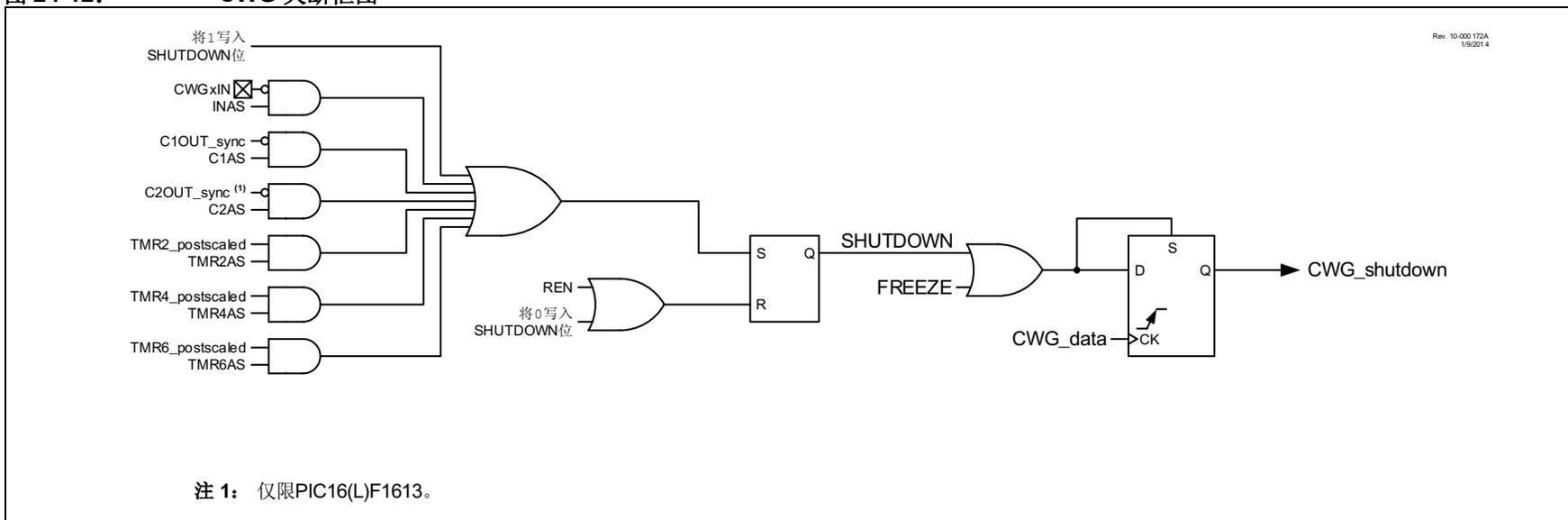
- 使能了 CWG 模块
- 输入源处于活动状态
- 选择 HFINTOSC 作为时钟源，无论选择的系统时钟源如何。

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. 10-000 172A

1/9/2014

图 24-12: CWG 关断框图



PIC12(L)F1612/16(L)F1613

24.12 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。

24.12.1 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 LSB0 和 LSAC 位控制。LSB0<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。

24.12.2 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 24-13 和图 24-14 给出了软件控制重启和自动重启的波形。

24.12.2.1 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。

24.12.2.2 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。

24.12.3 备用输出引脚

该模块具有这种输出：通过使用备用引脚功能寄存器 APFCON，可将输出转移到备用引脚。要确定可转移哪些输出以及其在复位时的默认引脚，请参见第 12.1 节“备用引脚功能”了解更多信息。

图 24-13: 关断功能, 禁止自动重启 (REN = 0, LSAC = 01, LSB D = 01)

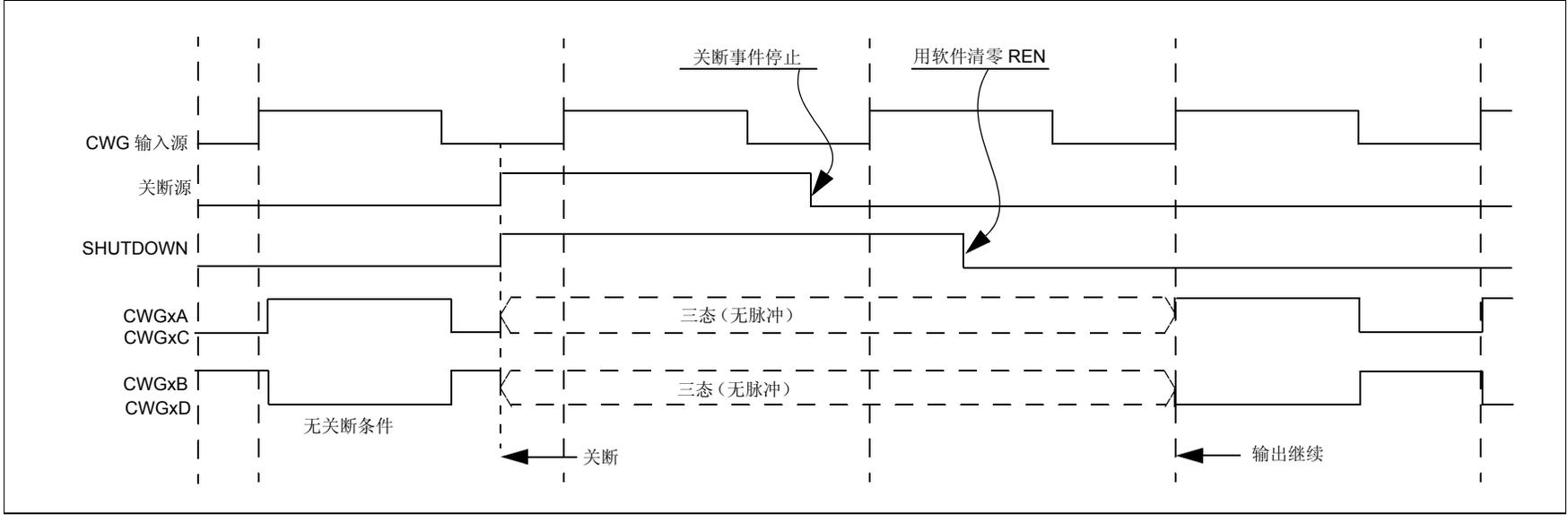
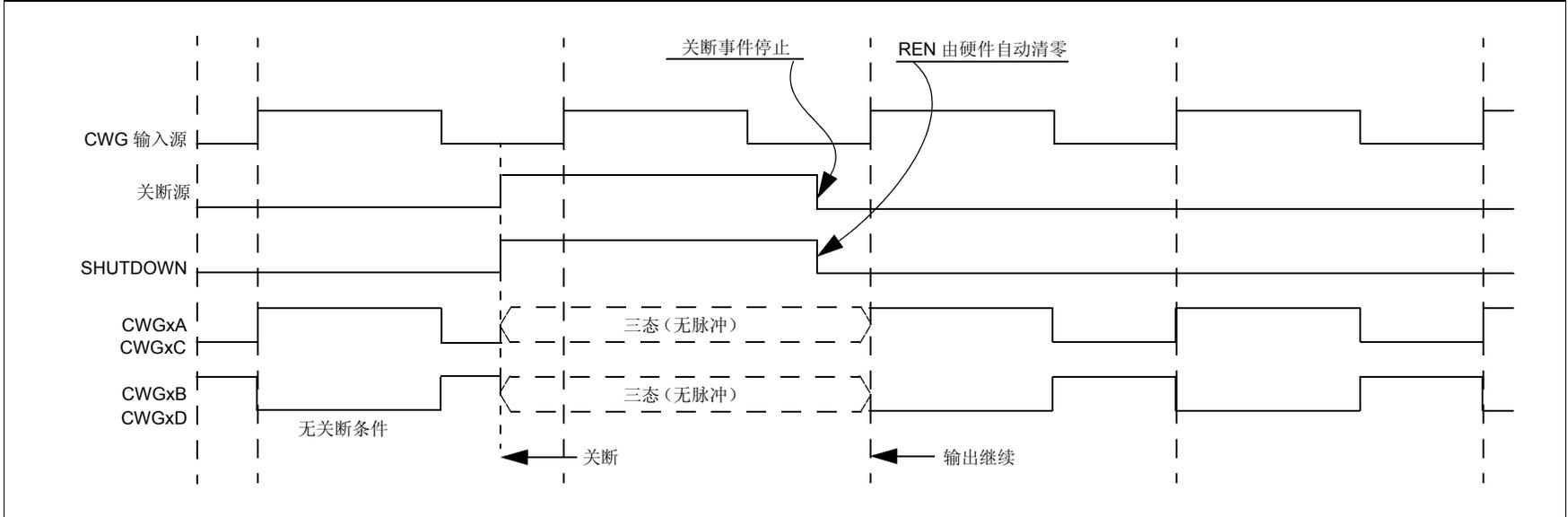


图 24-14: 关断功能, 使能自动重启 (REN = 1, LSAC = 01, LSB D = 01)



PIC12(L)F1612/16(L)F1613

24.13 寄存器定义: CWG 控制

寄存器 24-1: **CWGxCON0: CWGx 控制寄存器 0**

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD ⁽¹⁾	—	—	—	MODE<2:0>		
bit 7							bit 0

图注:

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 **EN:** CWGx 使能位

1 = 使能模块

0 = 禁止模块

bit 6 **LD:** CWGx 装入缓冲器位⁽¹⁾

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 **未实现:** 读为 0

bit 2-0 **MODE<2:0>:** CWGx 模式位

111 = 保留

110 = 保留

101 = CWG 输出在推挽模式下工作

100 = CWG 输出在半桥模式下工作

011 = CWG 输出在反向全桥模式下工作

010 = CWG 输出在正向全桥模式下工作

001 = CWG 输出在同步转向模式下工作

000 = CWG 输出在转向模式下工作

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。

PIC12(L)F1612/16(L)F1613

寄存器 24-2: 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

PIC12(L)F1612/16(L)F1613

寄存器 24-3: **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:** 上升沿事件死区计数器值位

寄存器 24-4: **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:** 下降沿事件死区计数器值位

PIC12(L)F1612/16(L)F1613

寄存器 24-5: 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 ^(1, 2)	REN	LSBD<1:0>		LSAC<1:0>		—	—
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 **SHUTDOWN:** 自动关断事件状态位 ^(1, 2)
 1 = 自动关断状态有效
 0 = 未发生自动关断事件
- bit 6 **REN:** 自动重启使能位
 1 = 使能自动重启
 0 = 禁止自动重启
- bit 5-4 **LSBD<1:0>:** CWGxB 和 CWGxD 自动关断状态控制位
 11 = 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。

PIC12(L)F1612/16(L)F1613

寄存器 24-6: **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 ⁽¹⁾	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 **TMR6AS:** 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 **TMR4AS:** 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 **TMR2AS:** 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 **C2AS:** 比较器 C2 输出位 ⁽¹⁾
1 = 使能 C2 输出关断
0 = 禁止 C2 输出关断
- bit 1 **C1AS:** 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 **INAS:** 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

注 1: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

寄存器 24-7: CWGxOCON0: CWGx 转向控制寄存器 (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
OVRD	OVRC	OVRB	OVRA	STRD ⁽²⁾	STRC ⁽²⁾	STRB ⁽²⁾	STRA ⁽²⁾
bit 7							bit 0

图注:

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

- bit 7 **OVRD:** 转向数据 D 位
- bit 6 **OVRC:** 转向数据 C 位
- bit 5 **OVRB:** 转向数据 B 位
- bit 4 **OVRA:** 转向数据 A 位
- bit 3 **STRD:** 转向使能 D 位 ⁽²⁾
 1 = CWGxD 输出具有 CWGx_data 波形, 其极性由 POLD 位控制
 0 = 将 CWGxD 输出指定为 OVRD 位的值
- bit 2 **STRC:** 转向使能 C 位 ⁽²⁾
 1 = CWGxC 输出具有 CWGx_data 波形, 其极性由 POLC 位控制
 0 = 将 CWGxC 输出指定为 OVRC 位的值
- bit 1 **STRB:** 转向使能 B 位 ⁽²⁾
 1 = CWGxB 输出具有 CWGx_data 波形, 其极性由 POLB 位控制
 0 = 将 CWGxB 输出指定为 OVRB 位的值
- bit 0 **STRA:** 转向使能 A 位 ⁽²⁾
 1 = CWGxA 输出具有 CWGx_data 波形, 其极性由 POLA 位控制
 0 = 将 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。
注 2: MODE<2:0> = 001 时, 该位实际上是双重缓冲的。

PIC12(L)F1612/16(L)F1613

寄存器 24-8: CWGxOCON1: CWGx 输出使能寄存器 1

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	OED	OEC	OEB	OEA
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-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 **OED:** CWGx D 输出引脚使能位
1 = 使能 CWGx D 输出引脚
0 = 禁止 CWGx D 输出引脚
- bit 2 **OEC:** CWGx C 输出引脚使能位
1 = 使能 CWGx C 输出引脚
0 = 禁止 CWGx C 输出引脚
- bit 1 **OEB:** CWGx B 输出引脚使能位
1 = 使能 CWGx B 输出引脚
0 = 禁止 CWGx B 输出引脚
- bit 0 **OEA:** CWGx A 输出引脚使能位
1 = 使能 CWGx A 输出引脚
0 = 禁止 CWGx A 输出引脚

PIC12(L)F1612/16(L)F1613

寄存器 24-9: 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 24-10: 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<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 **GxIS<2:0>:** CWGx 输入选择位
 111 = 保留, 不要使用
 110 = 保留, 不要使用
 101 = 保留, 不要使用
 100 = CCP2_out
 011 = CCP1_out
 010 = C2_OUT_sync⁽¹⁾
 001 = C1_OUT_sync
 000 = CWGxIN 引脚

注 1: 仅限 PIC16(L)F1613。

PIC12(L)F1612/16(L)F1613

表 24-2: 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
APFCON	—	CWGASEL ⁽²⁾	CWGBSEL ⁽²⁾	—	T1GSEL	—	CCP2SEL ⁽²⁾	CCP1SEL ⁽¹⁾	128
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	249
CWG1AS1	—	TMR6AS	TMR4AS	TMR2AS	—	C2AS ⁽²⁾	C1AS	INAS	250
CWG1CLKCON	—	—	—	—	—	—	—	CS	253
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			251
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	247
CWG1DBF	—	—	DBF<5:0>						248
CWG1DBR	—	—	DBR<5:0>						248
CWG1ISM	—	—	—	—	—	IS<2:0>			253
CWG1OCON0	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	251
CWG1OCON1	—	—	—	—	OED	OEC	OEB	OEA	252

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

- 注 1: 仅限 PIC12(L)F1612。
 注 2: 仅限 PIC16(L)F1613。

25.0 信号测量定时器 (SMTx)

SMTx 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。

SMTx 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
- 上溢时产生中断
- 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>注： 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>

PIC12(L)F1612/16(L)F1613

图 25-1: SMTx 框图

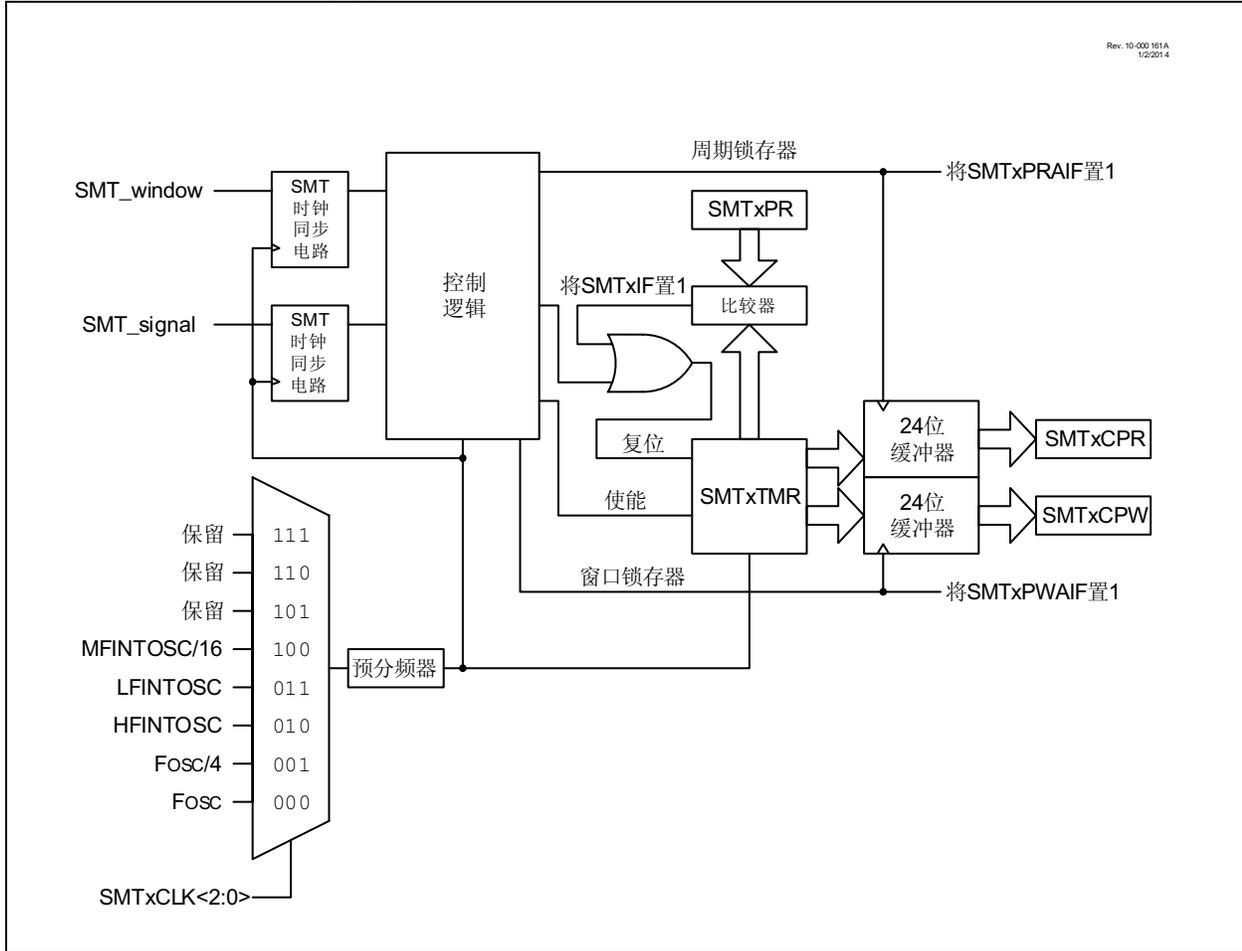
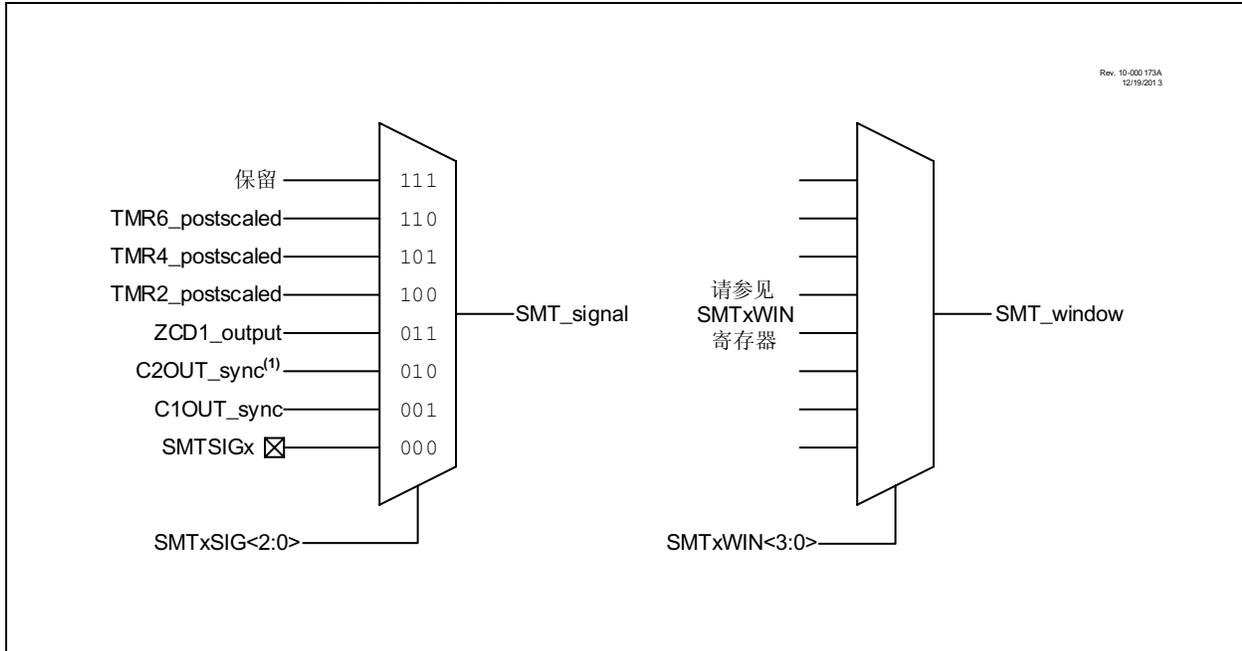


图 25-2: SMTx 信号和窗口框图



25.1 SMTx 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，SMTx 可以执行表 25-1 汇总的各种测量。

25.1.1 时钟源

SMTx 可用的时钟源包括：

- FOSC
- FOSC/4
- HFINTOSC 16 MHz
- LFINTOSC
- MFINTOSC 31.25 kHz

SMTx 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。

25.1.2 上溢中断

类似于其他定时器，SMTx 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。

25.2 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。

25.2.1 时基

SMTxTMR 是一个 24 位计数器，它是 SMTx 的核心。在 SMTx 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确保 SMTxTMR 读 / 写的完整性。

25.2.2 脉冲宽度锁存器寄存器

SMTxCPW 寄存器为 24 位 SMTx 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。

25.2.3 周期锁存器寄存器

SMTxCPR 寄存器为 24 位 SMTx 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 SMTx 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。

25.3 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，上溢中断会持续发生，直到复位 SMTxTMR 为止（通过手动复位（第 25.2.1 节“时基”）或通过用软件清零 SMTxCON1 寄存器的 SMTxGO 位并写入 SMTxTMR 值）。

25.4 极性控制

SMTx 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。

25.5 状态信息

SMTx 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。

25.5.1 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。

25.5.2 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、行程时间和捕捉模式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，延迟时间等于同步器延时。

PIC12(L)F1612/16(L)F1613

25.5.3 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，延迟时间等于同步器延时。

25.6 工作模式

表 25-1 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。

25.6.1 定时器模式

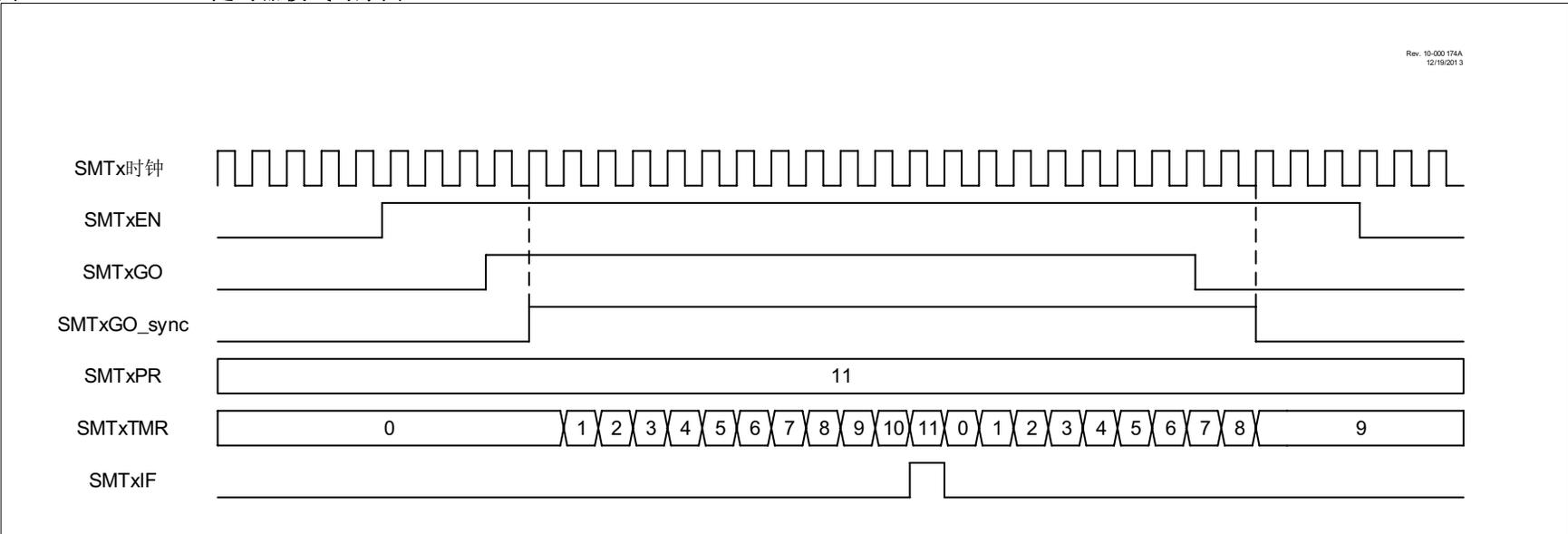
定时器模式是最简单的工作模式，SMTxTMR 在该模式下用作 16/24 位定时器。该模式下不会发生数据采集。只要用软件将 SMTxGO 位置 1，定时器就会一直递增。没有 SMTx 窗口或 SMTx 信号事件会影响 SMTxGO 位。所有时序都与 SMTx 时钟源进行同步。当定时器上溢（SMTxTMR = SMTxPR）时，SMTxTMR 会发生复位，并触发上溢中断。请参见图 25-3。

表 25-1: 工作模式

MODE	工作模式	同步操作	参考章节
0000	定时器	是	第 25.6.1 节 “定时器模式”
0001	门控定时器	是	第 25.6.2 节 “门控定时器模式”
0010	周期和占空比采集	是	第 25.6.3 节 “周期和占空比模式”
0011	高电平和低电平时间测量	是	第 25.6.4 节 “高电平和低电平测量模式”
0100	窗口测量	是	第 25.6.5 节 “窗口测量模式”
0101	门控窗口测量	是	第 25.6.6 节 “门控窗口测量模式”
0110	行程时间	是	第 25.6.7 节 “行程时间测量模式”
0111	捕捉	是	第 25.6.8 节 “捕捉模式”
1000	计数器	否	第 25.6.9 节 “计数器模式”
1001	门控计数器	否	第 25.6.10 节 “门控计数器模式”
1010	窗口计数器	否	第 25.6.11 节 “窗口计数器模式”
1011 - 1111	保留	—	—

Rev. 10-000 174A
12/19/2013

图 25-3: 定时器模式时序图



PIC12(L)F1612/16(L)F1613

25.6.2 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 25-4 和图 25-5 给出了进行重复采集和单次采集的示例波形。

Rev. 10-000 178A
12/19/2013

图 25-4: 门控定时器模式重复采集时序图

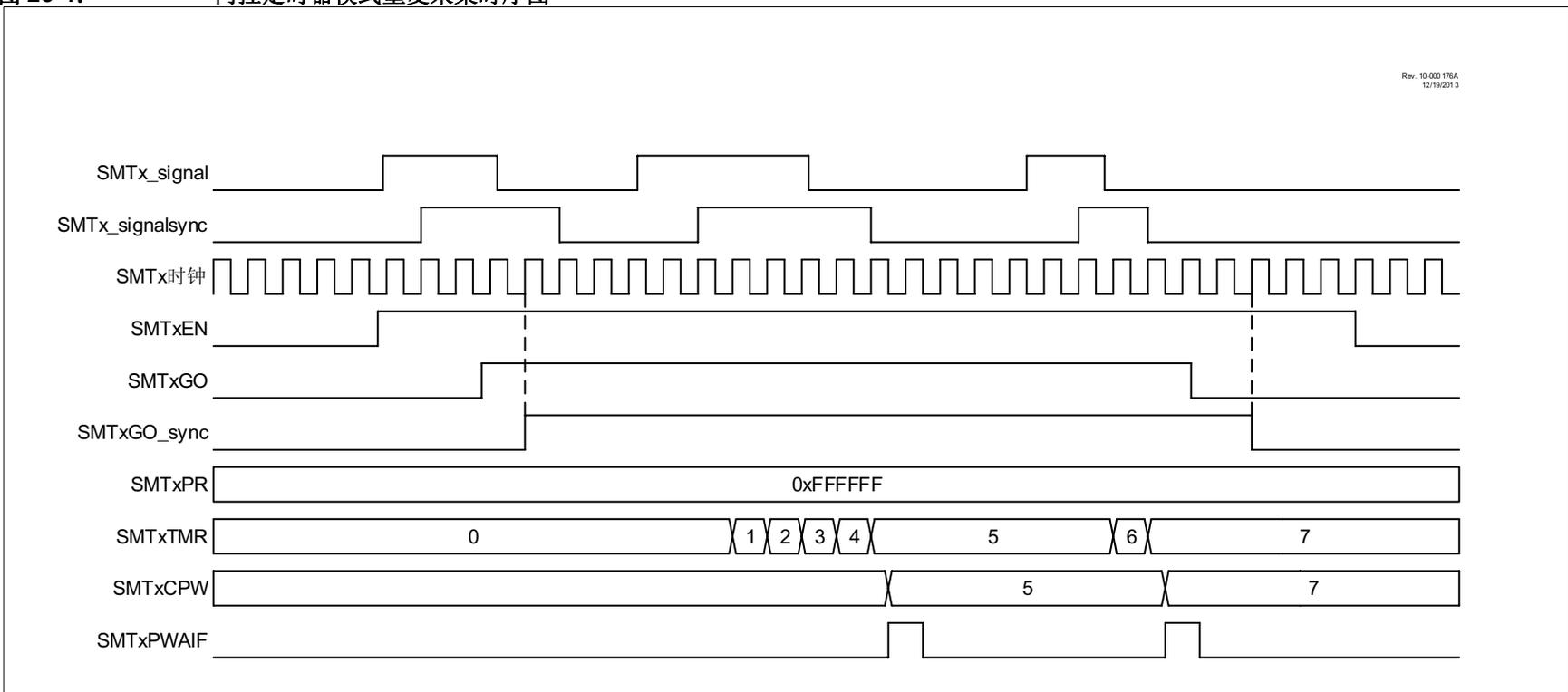
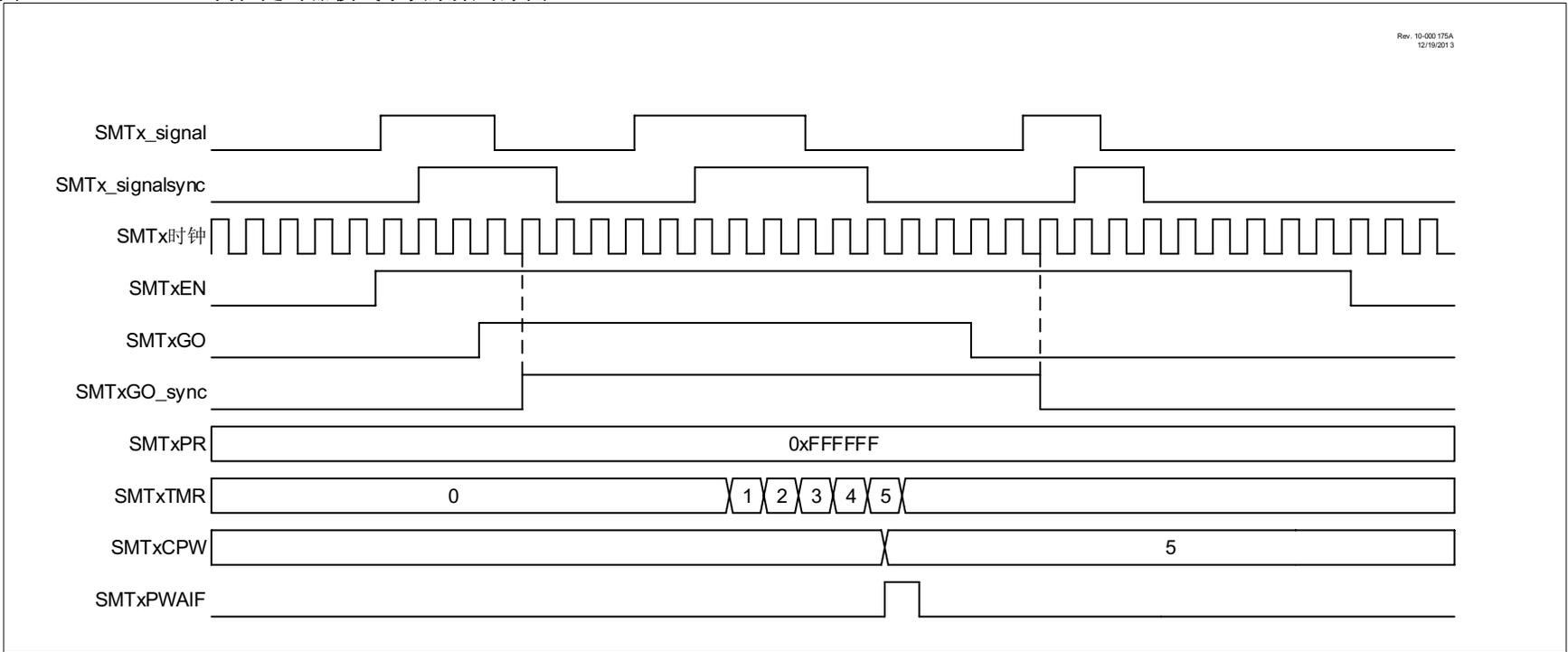


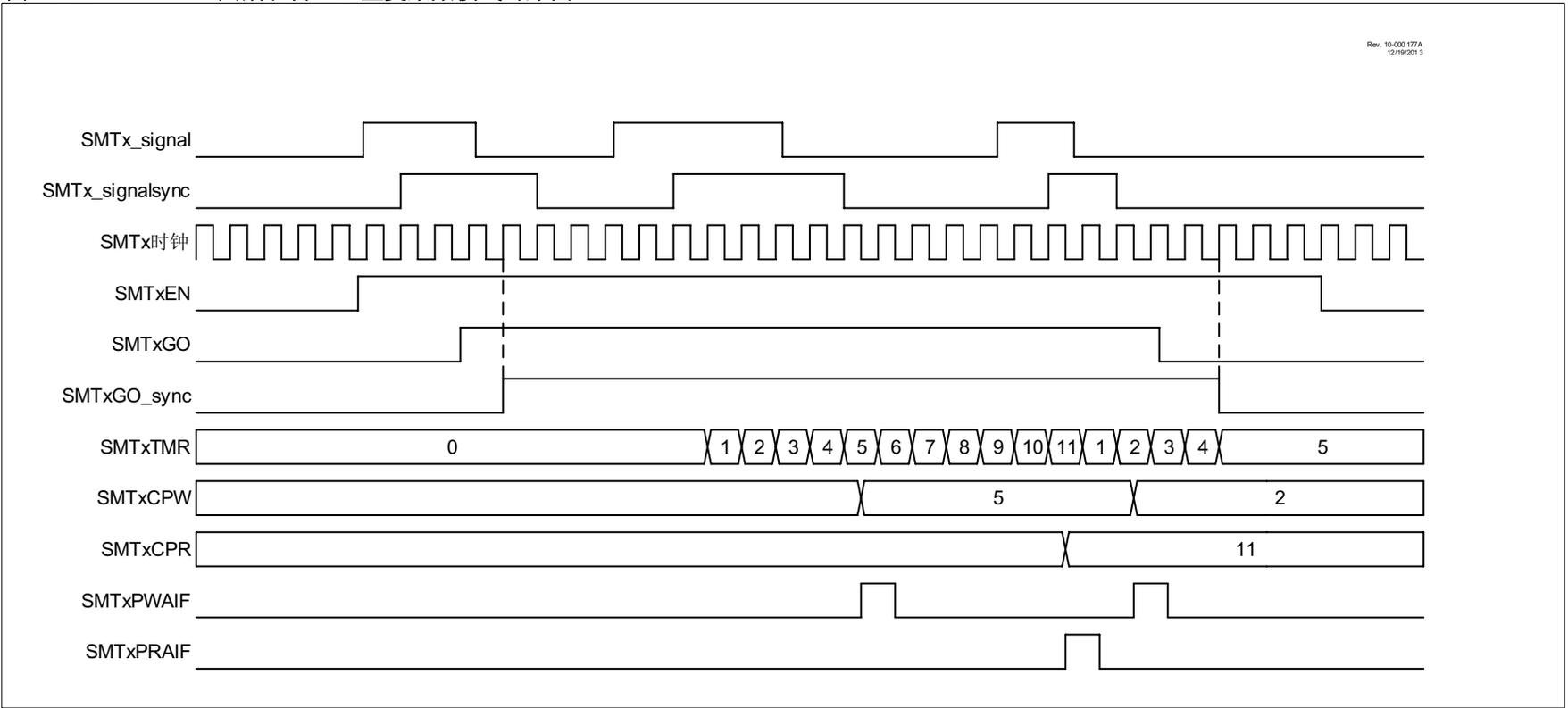
图 25-5: 门控定时器模式单次采集时序图



25.6.3 周期和占空比模式

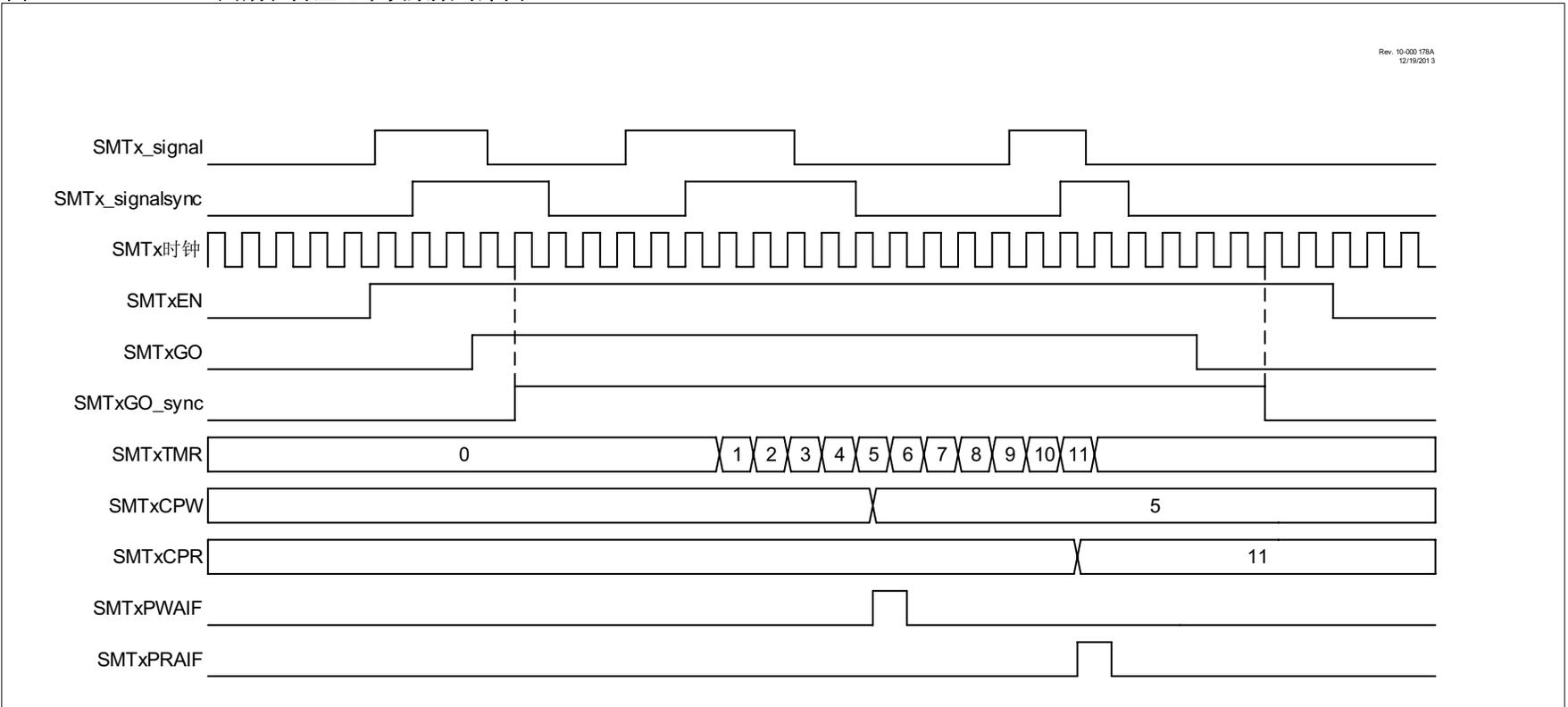
在占空比模式下，可以相对于 SMT 时钟采集 SMTx_signal 的占空比和周期（取决于极性）。CPW 寄存器在信号的下降沿更新，CPR 寄存器在信号的上升沿更新，并且 SMTxTMR 复位为 0x0001。此外，在 SMT 处于单次采集模式时，SMTxGO 位在上升沿发生复位。请参见图 25-6 和图 25-7。

图 25-6: 周期和占空比重复采集模式时序图



Rev. 10-000 178A
12/19/2013

图 25-7: 周期和占空比单次采集时序图



PIC12(L)F1612/16(L)F1613

25.6.4 高电平和低电平测量模式

该模式相对于 SMTx 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 25-8 和图 25-9。

Rev. 10-00 180A
12/19/2013

图 25-8: 高电平和低电平测量模式重复采集时序图

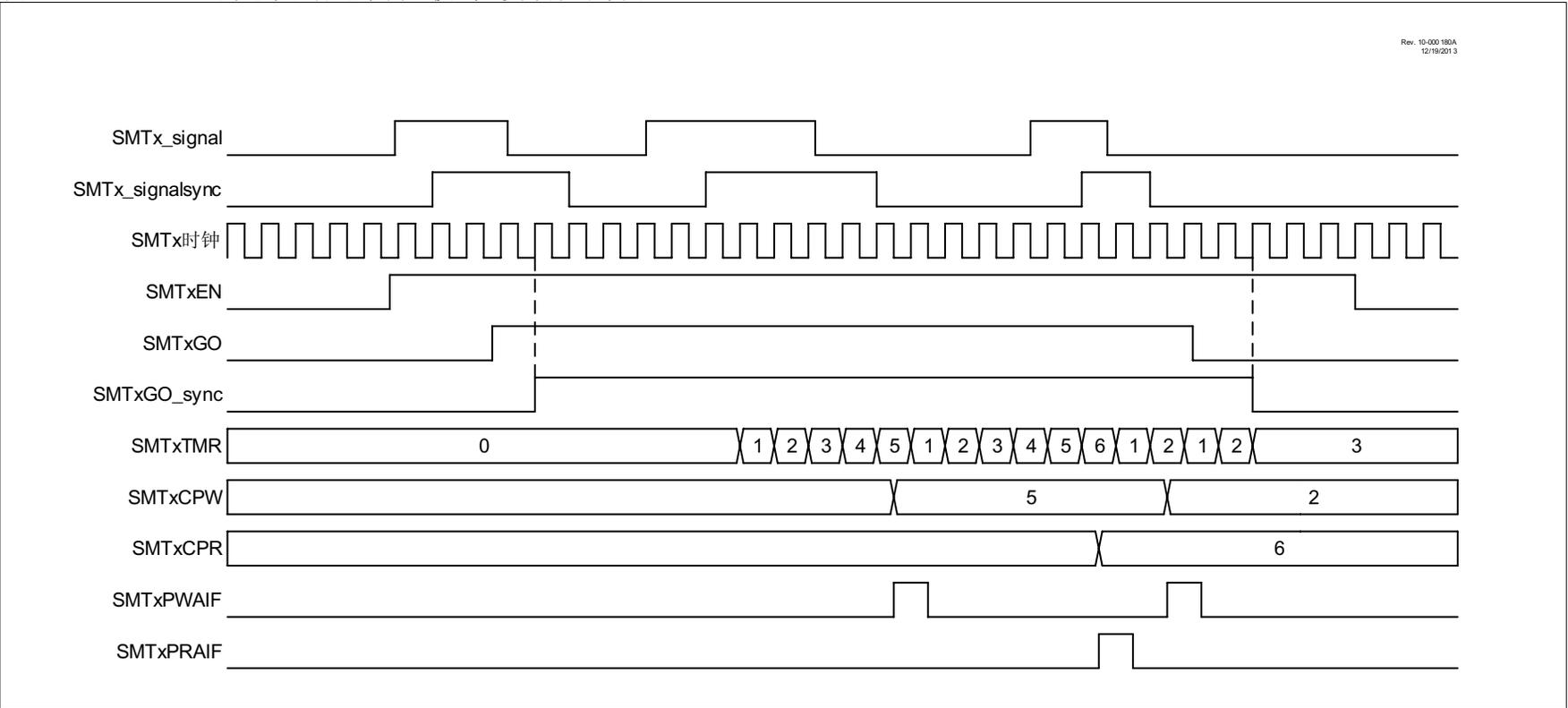
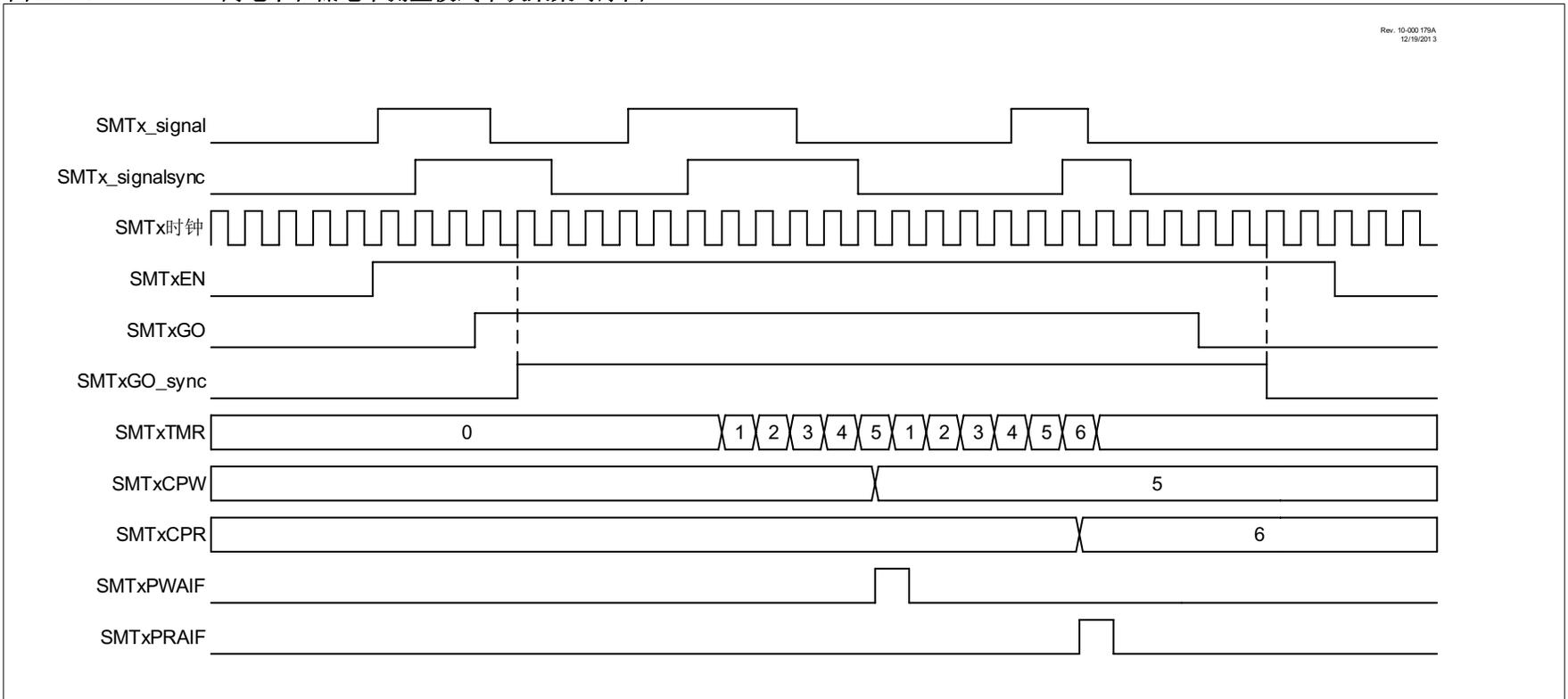


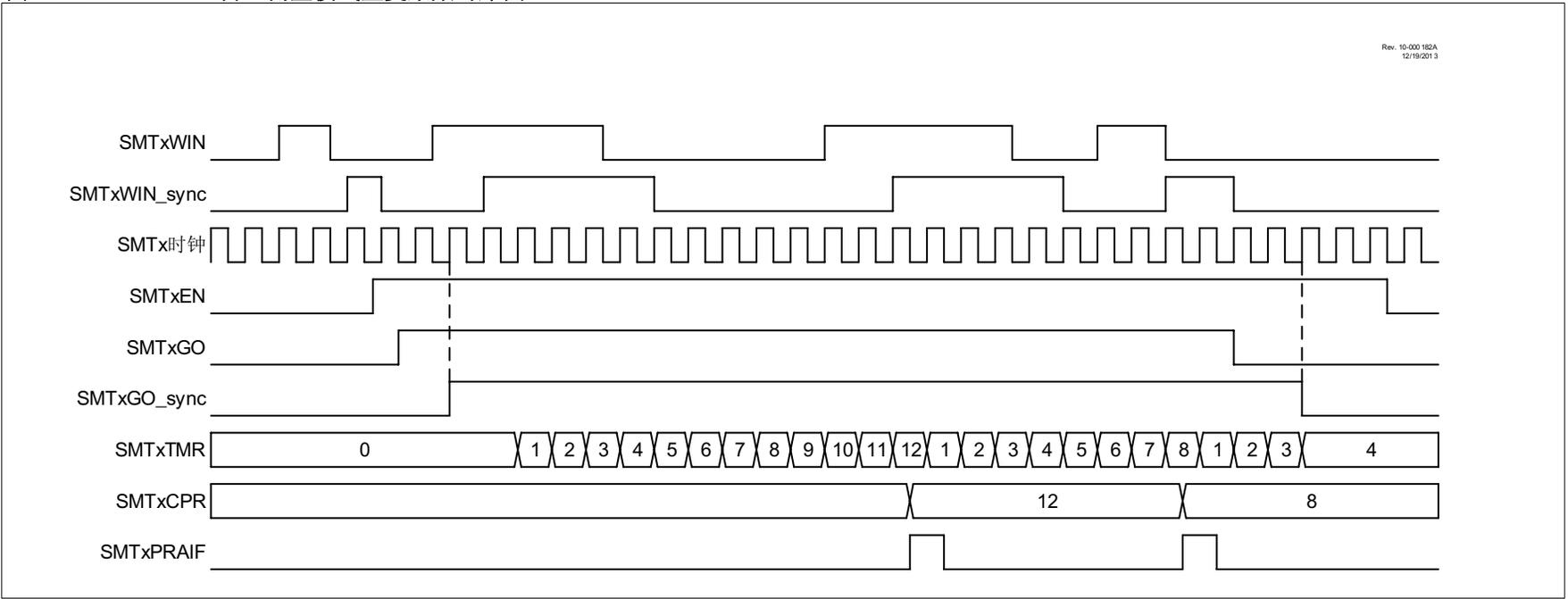
图 25-9: 高电平和低电平测量模式单次采集时序图



25.6.5 窗口测量模式

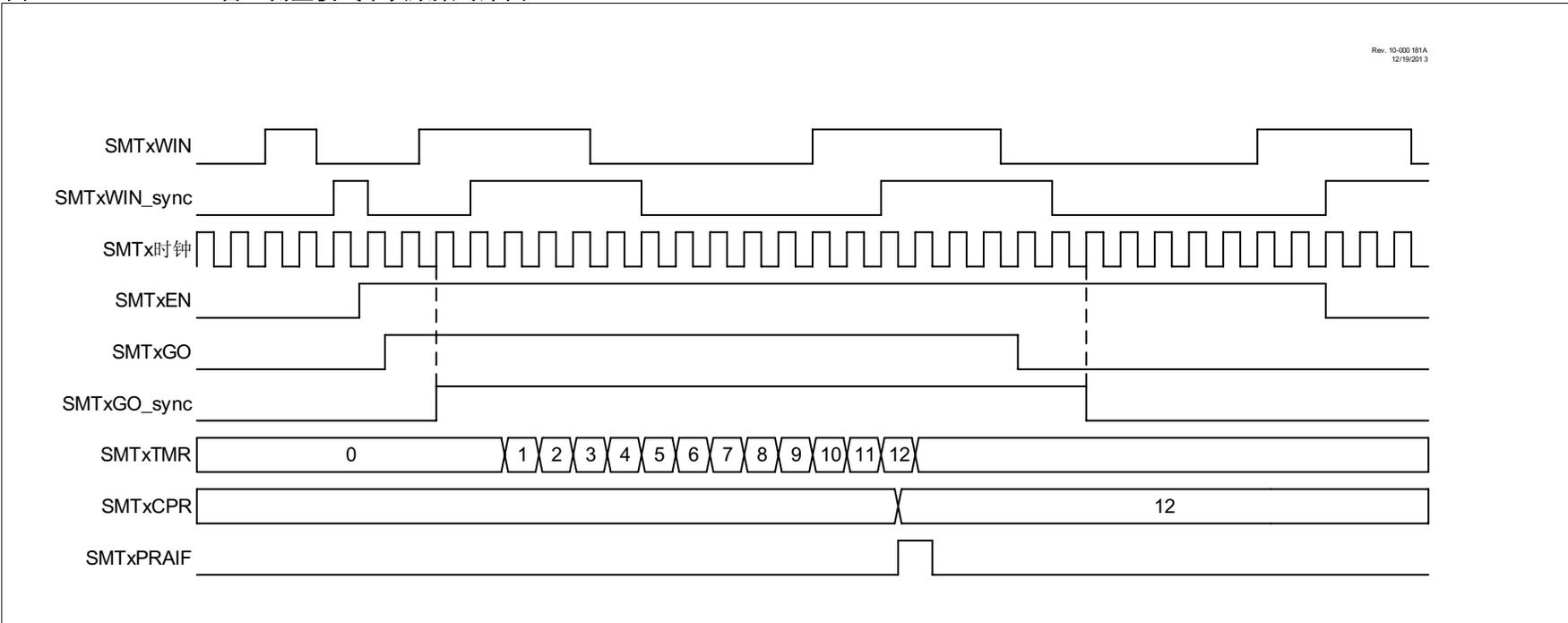
该模式测量 SMTx 的 SMTWINx 输入的窗口持续时间。它在 SMTWINx 输入的上升沿开始递增定时器，在第二个上升沿使用定时器的值更新 SMTxCPR 寄存器并复位定时器。请参见图 25-10 和图 25-11。

图 25-10: 窗口测量模式重复采集时序图



Rev. 10-000 181A
12/19/2013

图 25-11: 窗口测量模式单次采集时序图



PIC12(L)F1612/16(L)F1613

25.6.6 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 25-12 和图 25-13。

Rev. 10-000 184A
12/19/2013

图 25-12: 门控窗口测量模式重复采集时序图

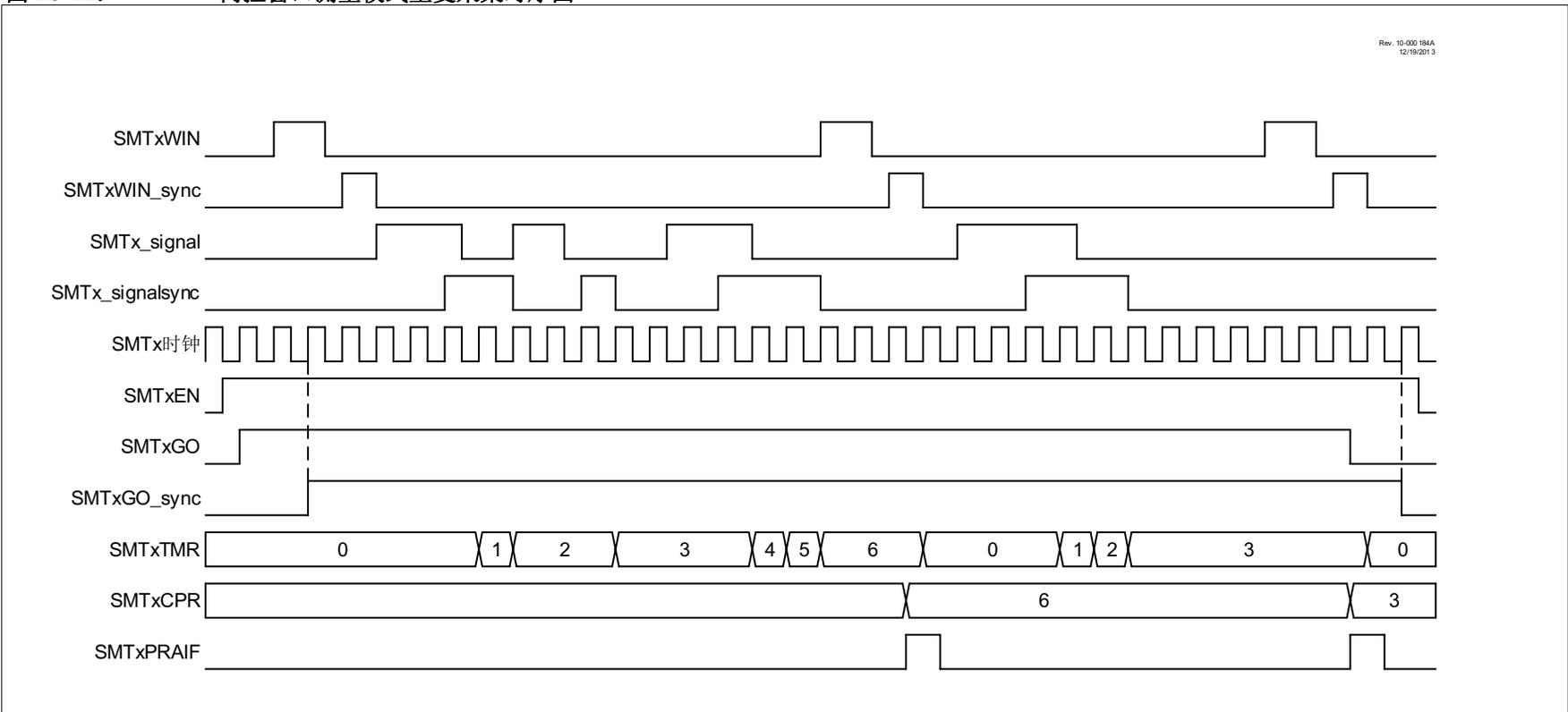
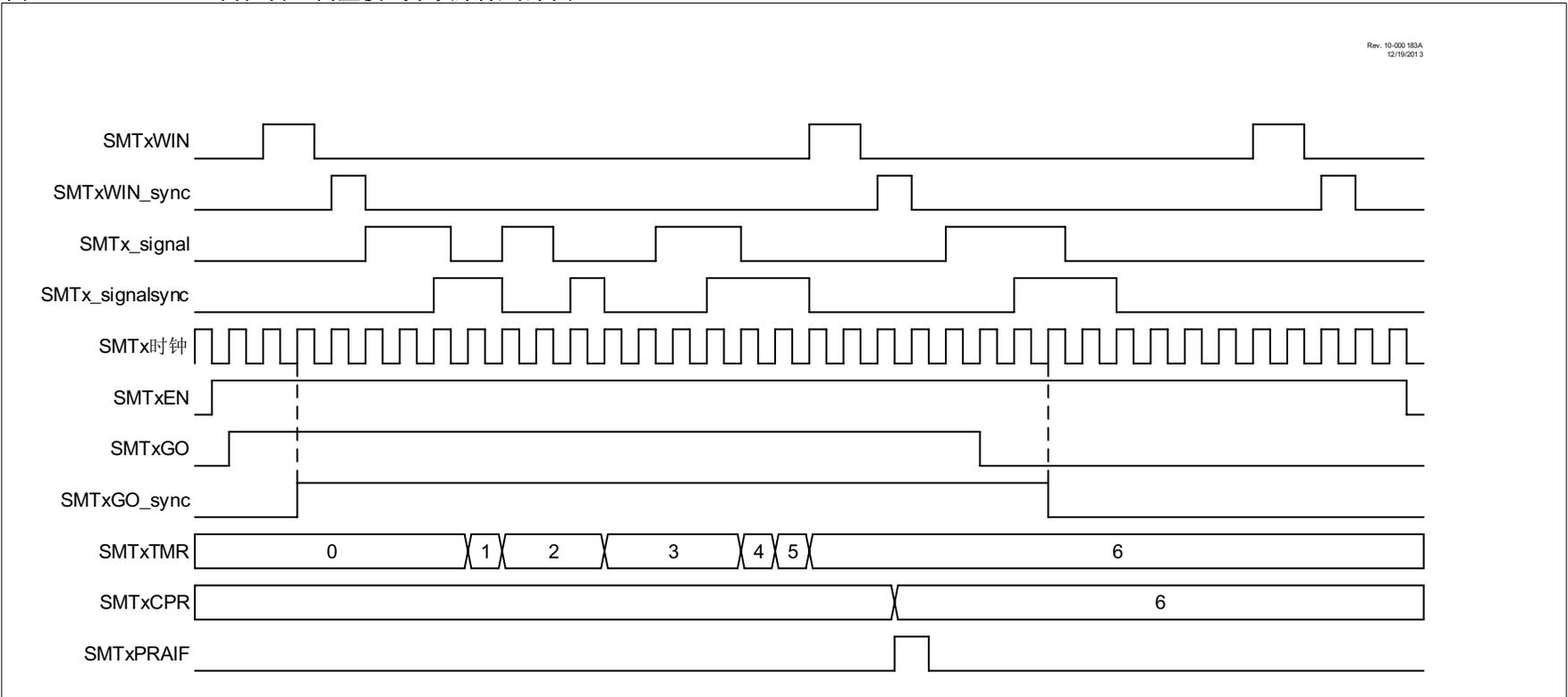


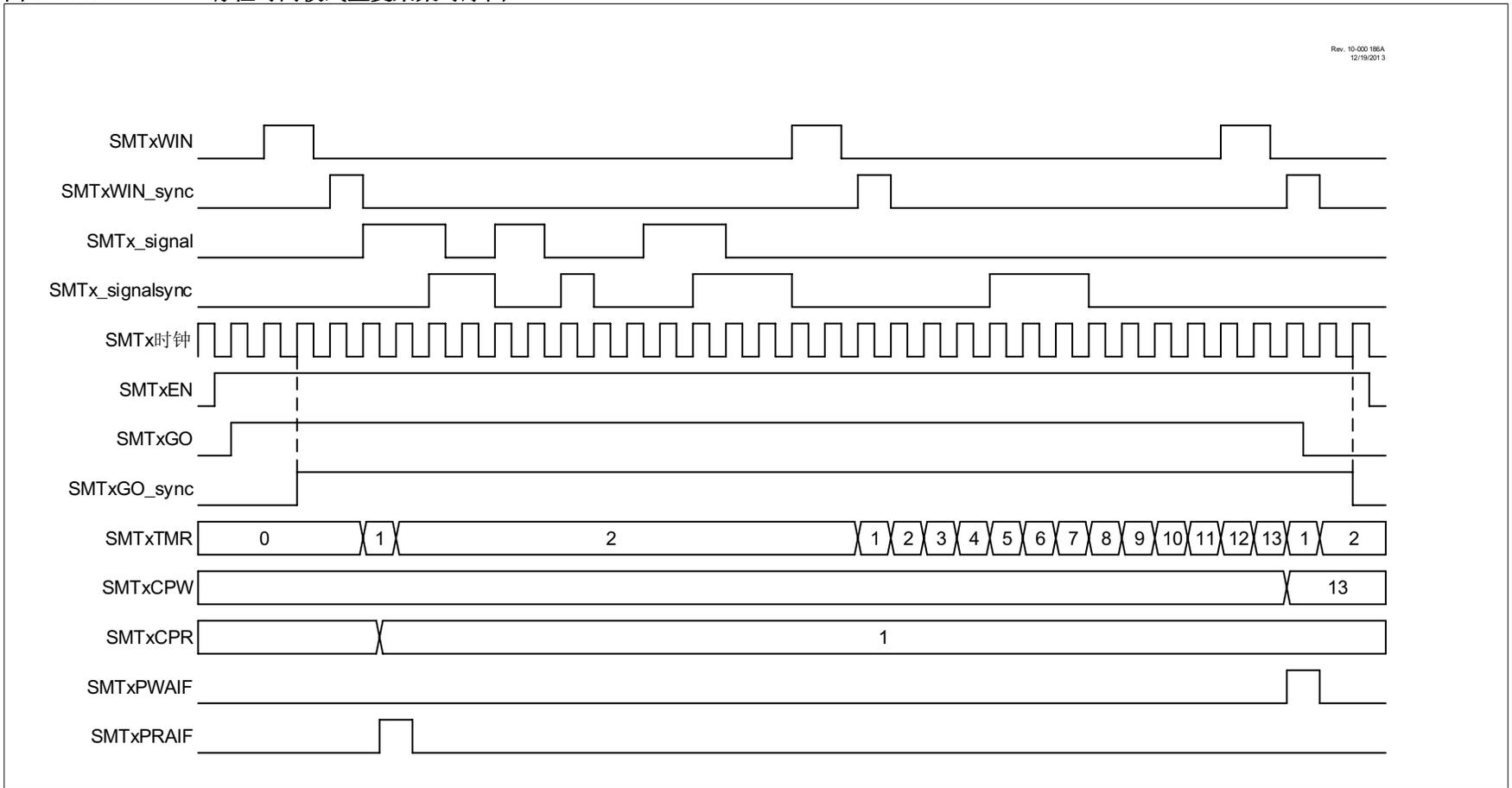
图 25-13: 门控窗口测量模式单次采集时序图



25.6.7 行程时间测量模式

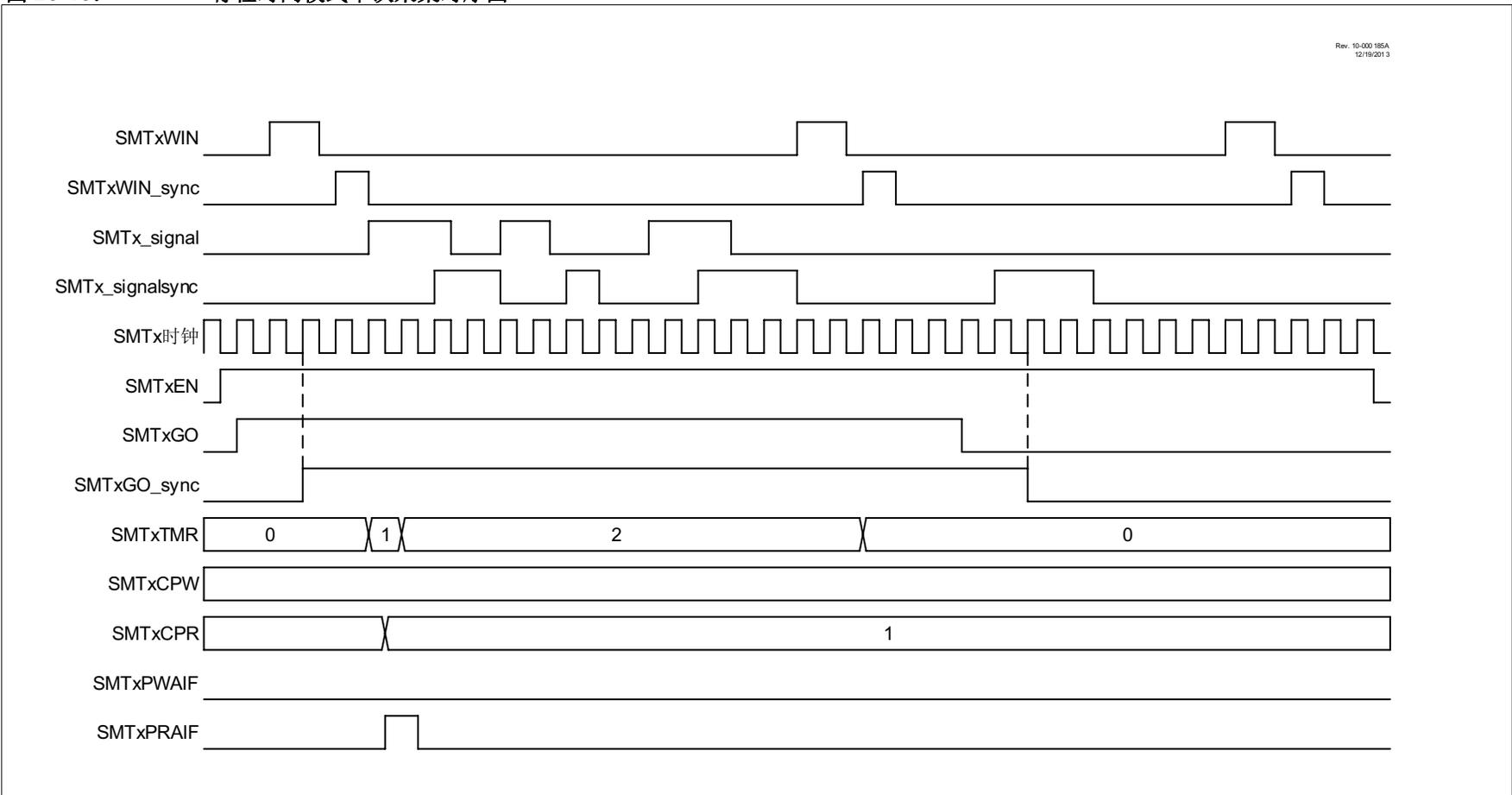
该模式测量 SMTWINx 输入上升沿和 SMTx_signal 输入上升沿之间的时间间隔，即在 SMTWINx 输入上检测到上升沿时开始递增定时器，在 SMTx_signal 输入上检测到上升沿时更新 SMTxCPR 寄存器并复位定时器。在出现两个 SMTWINx 上升沿而未出现 SMTx_signal 上升沿的情况下，它会使用定时器的当前值更新 SMTxCPW 寄存器并复位定时器值。请参见图 25-14 和图 25-15。

图 25-14: 行程时间模式重复采集时序图



Rev. 10-000 185A
12/19/2013

图 25-15: 行程时间模式单次采集时序图



PIC12(L)F1612/16(L)F1613

25.6.8 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 25-16 和图 25-17。

Rev: 10-000-186A
12/18/2013

图 25-16: 捕捉模式重复采集时序图

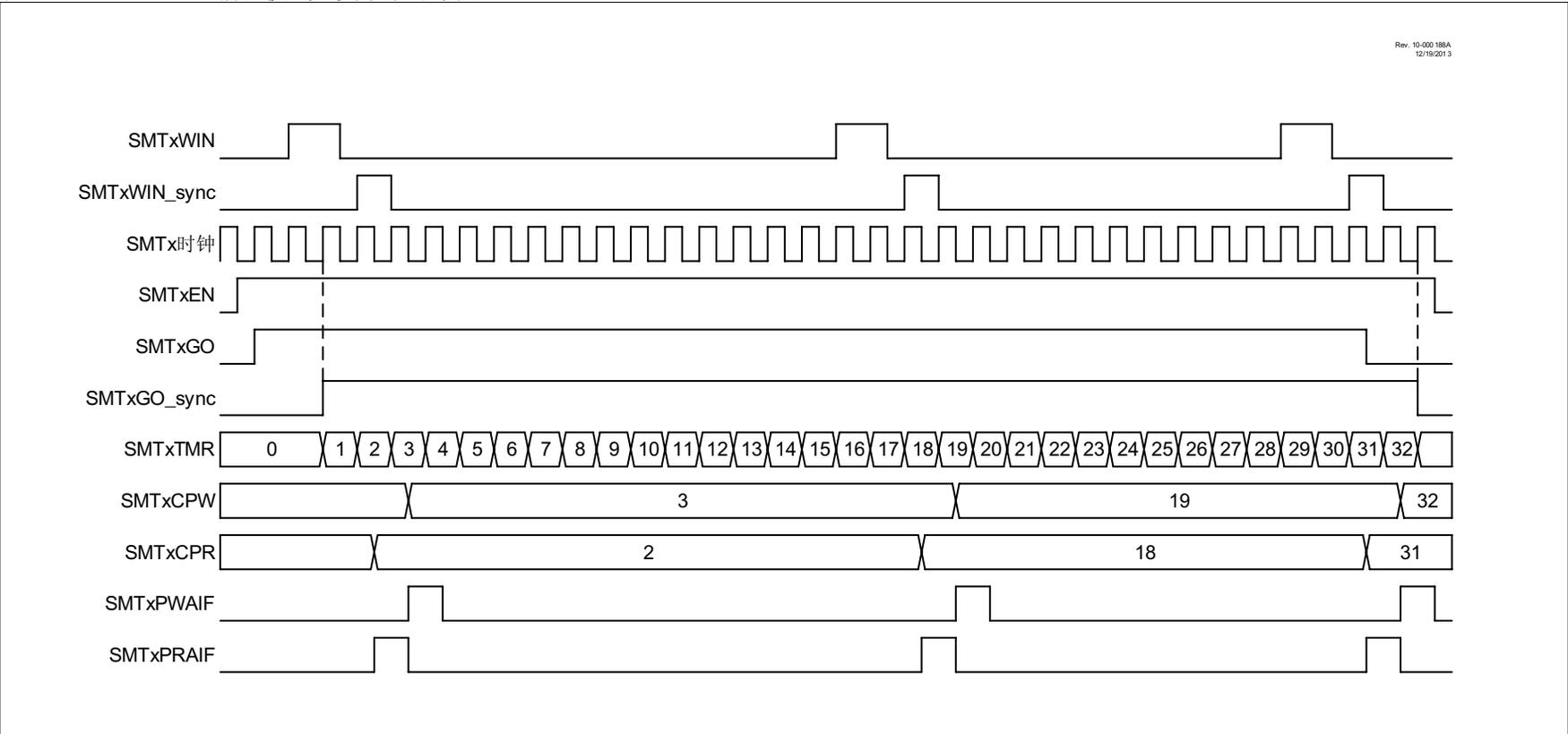
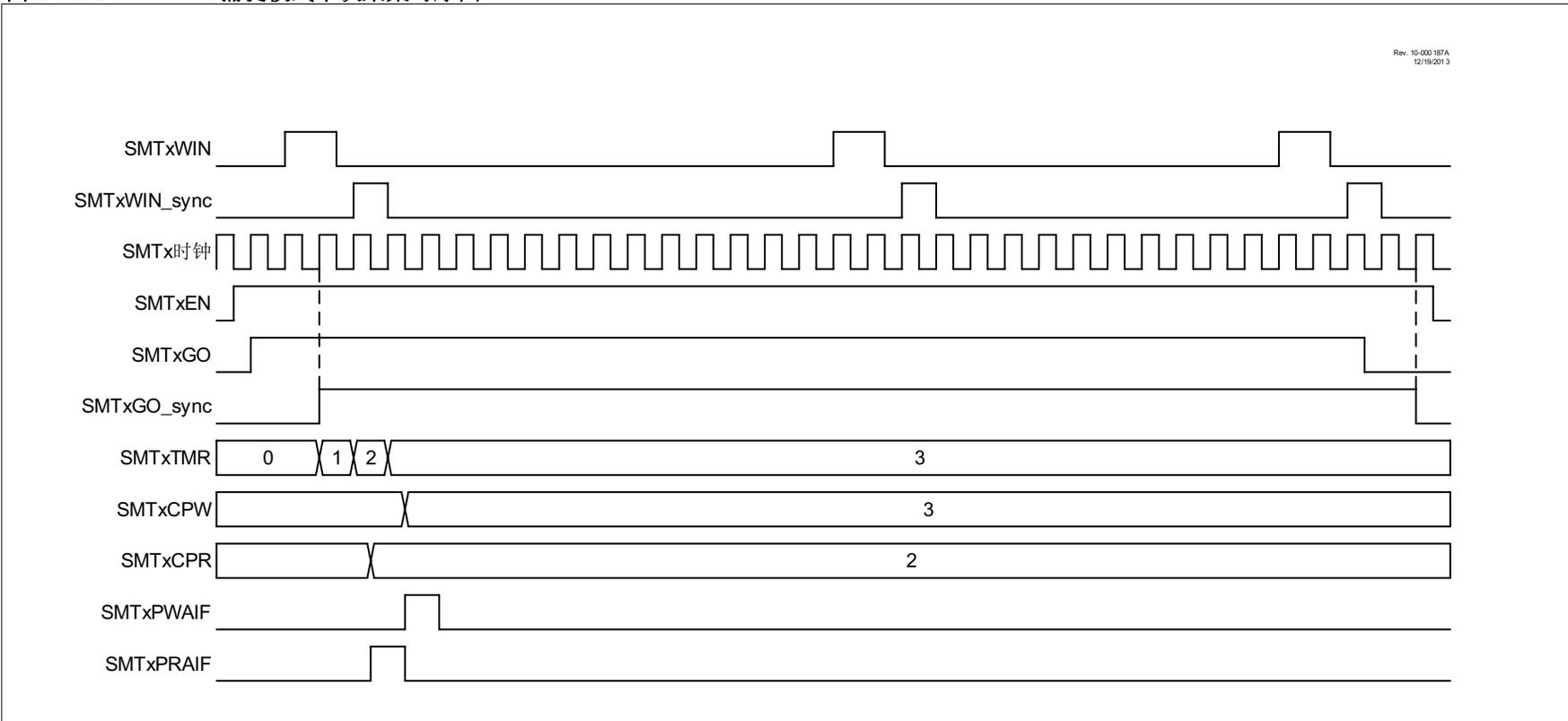


图 25-17: 捕捉模式单次采集时序图

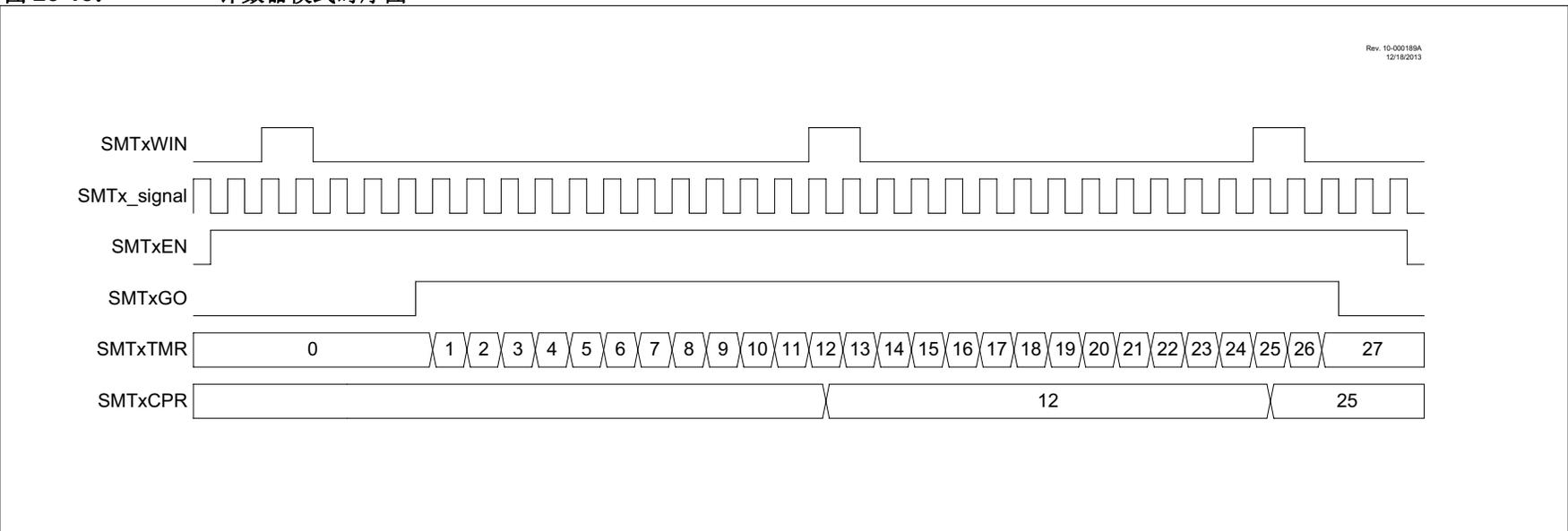


25.6.9 计数器模式

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。可以通过 SMTxWIN 输入的上升沿将 SMTxCPR 寄存器更新为当前时钟值。请参见图 25-18。

Rev. 10-2007/8A
12/18/2013

图 25-18: 计数器模式时序图



25.6.10 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 25-19 和图 25-20。

图 25-19: 门控计数器模式重复采集时序图

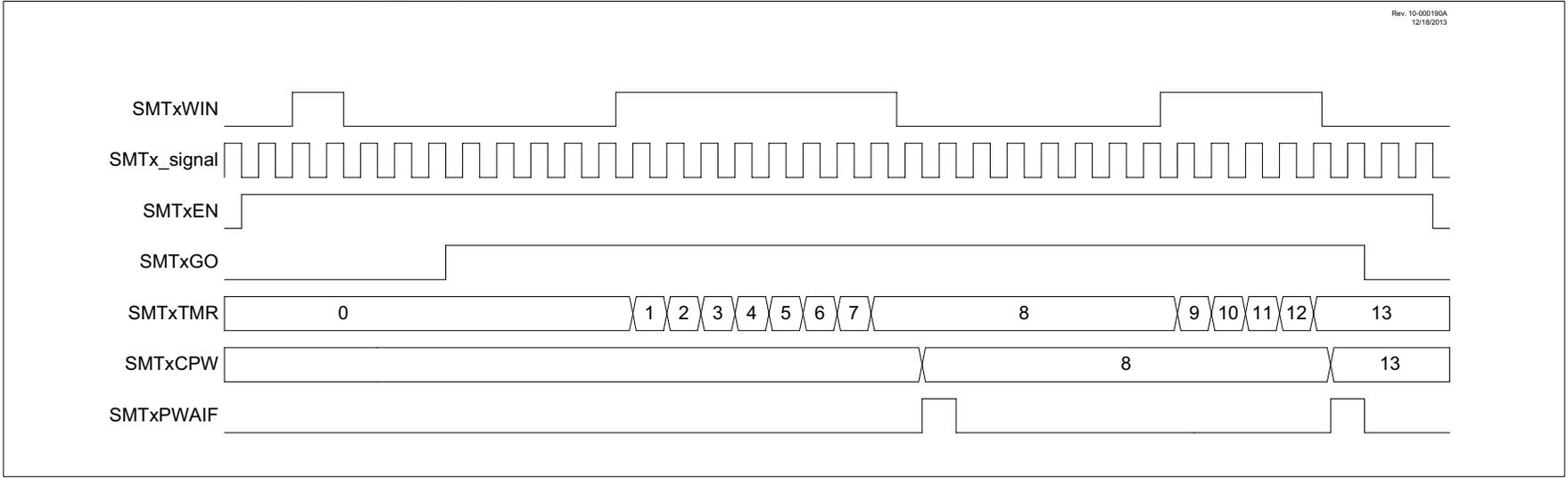
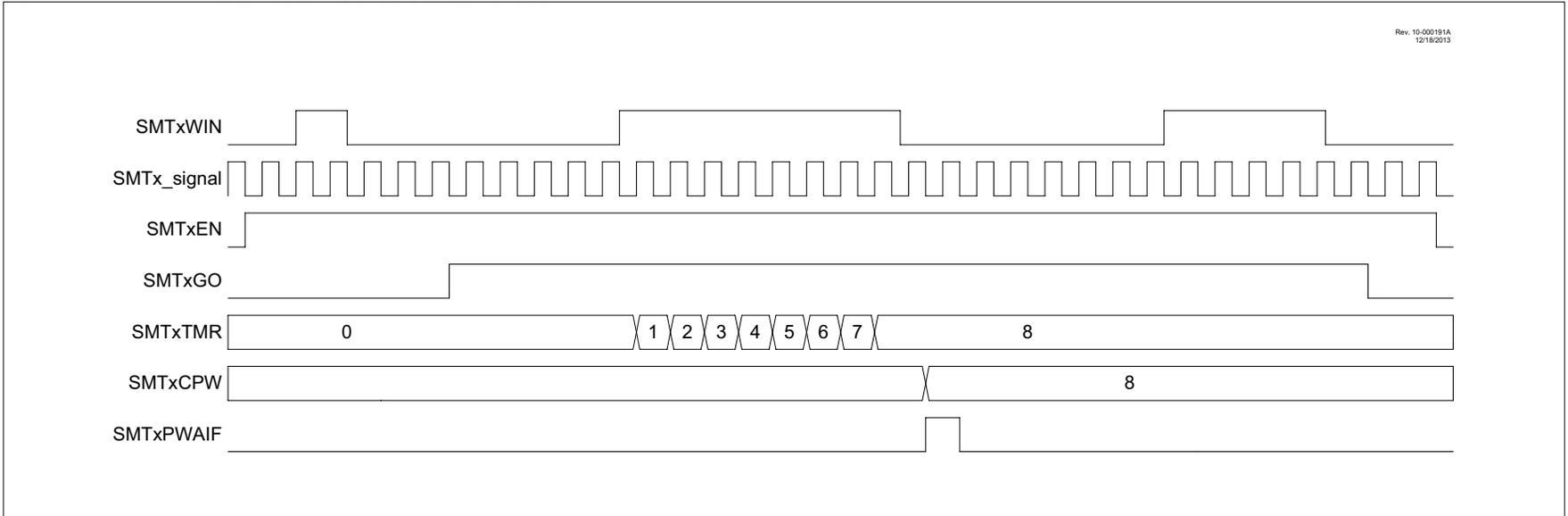


图 25-20: 门控计数器模式单次采集时序图



25.6.11 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 25-21 和图 25-22。

图 25-21: 窗口计数器模式重复采集时序图

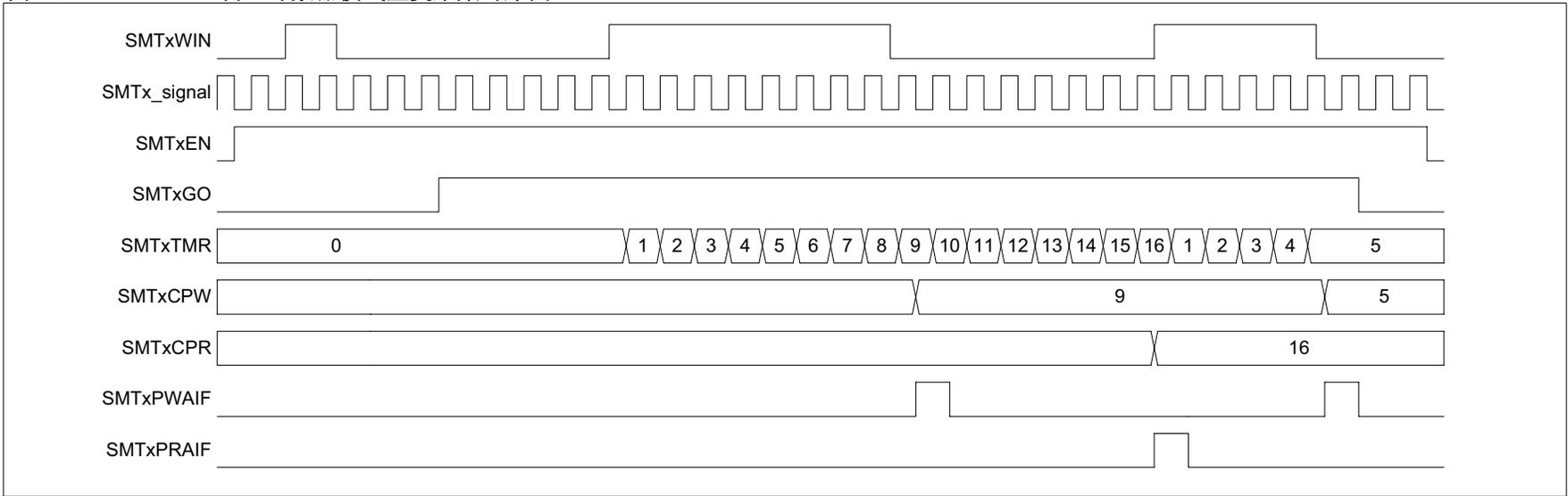
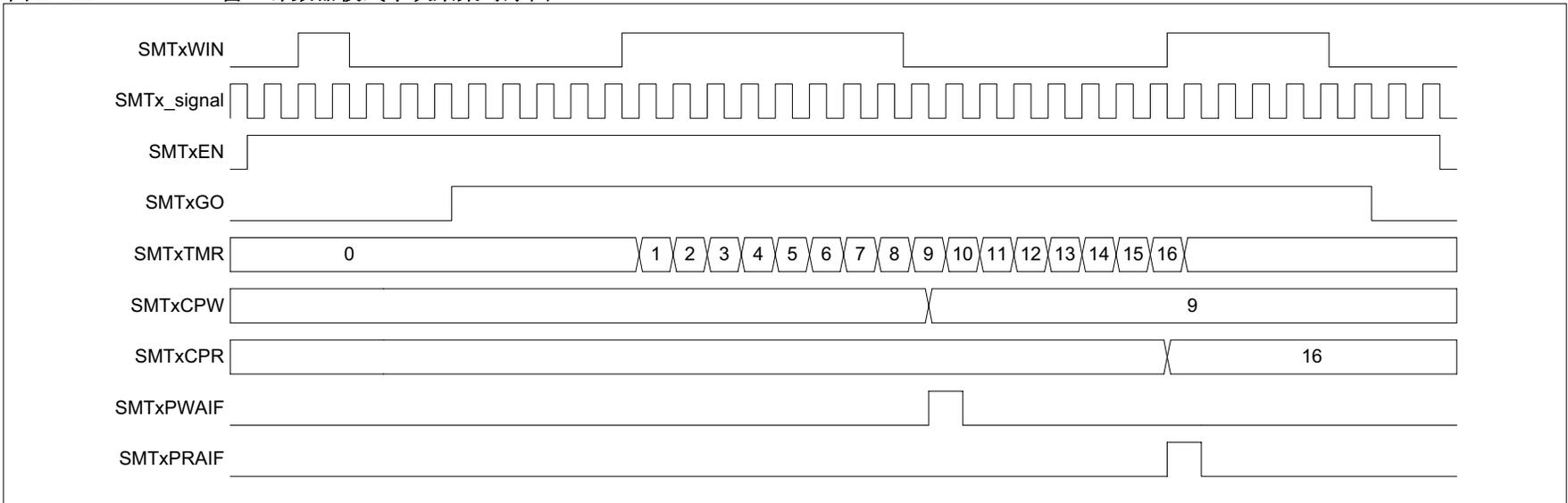


图 25-22: 窗口计数器模式单次采集时序图



25.7 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
- 计数器上溢

中断由器件的 PIR 和 PIE 寄存器控制。

25.7.1 PW 和 PR 采集中断

SMTx 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。

25.7.2 计数器上溢中断

如第25.1.2节“上溢中断”所述，SMT还会在SMTxTMR与SMTxPR匹配时产生中断，其上溢限制功能如第25.3节“暂停操作”所述。上溢中断由 SMTxIF 和 SMTxIE 控制，这两个位分别位于寄存器 PIR4 和 PIE4 中。

PIC12(L)F1612/16(L)F1613

25.8 寄存器定义：SMT 控制

寄存器 25-1: SMTxCON0: SMTx 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 ⁽¹⁾	—	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:** SMTx 使能位 ⁽¹⁾
1 = 使能 SMTx
0 = 禁止 SMTx; 复位内部状态, 禁止时钟请求
- bit 6 **未实现:** 读为 0
- bit 5 **STP:** SMTx 计数器暂停使能位
当 SMTxTMR = SMTxPR 时:
1 = 计数器保持为 SMTxPR; 在输入时钟时产生上溢中断
0 = 计数器计满返回到 24'h000000; 在输入时钟时产生上溢中断
- 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:** SMTx 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>:** SMTx 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。

PIC12(L)F1612/16(L)F1613

寄存器 25-2: SMTxCON1: SMTx 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

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 **SMTxGO:** SMTx 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 **REPEAT:** SMTx 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 **未实现:** 读为 0
- bit 3-0 **MODE<3:0>:** SMTx 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

PIC12(L)F1612/16(L)F1613

寄存器 25-3: SMTxSTAT: SMTx 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

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 **CPRUP:** SMTx 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:** SMTx 手动脉冲宽度缓冲器更新位

1 = 请求更新 SMTxCPWx 寄存器

0 = SMTxCPWx 寄存器更新已完成

bit 5 **RST:** SMTx 手动定时器复位位

1 = 请求复位 SMTxTMRx 寄存器

0 = SMTxTMRx 寄存器更新已完成

bit 4-3 **未实现:** 读为 0

bit 2 **TS:** SMTx 运行值状态位

1 = SMTx 定时器正在递增

0 = SMTx 定时器不在递增

bit 1 **WS:** SMTWINx 值状态位

1 = SMTx 窗口已打开

0 = SMTx 窗口已关闭

bit 0 **AS:** SMTx_signal 值状态位

1 = 正在进行 SMT 采集

0 = 不在进行 SMT 采集

PIC12(L)F1612/16(L)F1613

寄存器 25-4: SMTxCLK: SMTx 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>**: SMTx 时钟选择位

111 = 保留

110 = 保留

101 = 保留

100 = MFINTOSC/16

011 = LFINTOSC

010 = HFINTOSC 16 MHz

001 = Fosc/4

000 = Fosc

PIC12(L)F1612/16(L)F1613

寄存器 25-5: SMTxWIN: SMTx 窗口输入选择寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	WSEL<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 **WSEL<3:0>**: SMTx 窗口选择位
- 1111 = 保留
 -
 -
 -
 - 1001 = 保留
 - 1000 = TMR6_postscaled
 - 0111 = TMR4_postscaled
 - 0110 = TMR2_postscaled
 - 0101 = ZCD1_out
 - 0100 = CCP2_out
 - 0011 = CCP1_out
 - 0010 = C2_OUT_sync⁽¹⁾
 - 0001 = C1_OUT_sync
 - 0000 = SMTWINx 引脚

注 1: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

寄存器 25-6: SMTxSIG: SMTx 信号输入选择寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	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>**: SMTx 信号选择位

1111 = 保留

•

•

•

111 = 保留

110 = TMR6_postscaled

101 = TMR4_postscaled

100 = TMR2_postscaled

011 = ZCD1_out

010 = C2_OUT_sync⁽¹⁾

001 = C1_OUT_sync

000 = SMTxSIG 引脚

注 1: 仅限 PIC16(L)F1613。在 PIC12(L)F1612 上保留。

PIC12(L)F1612/16(L)F1613

寄存器 25-7: SMTxTMRL: SMTx 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 25-8: SMTxTMRH: SMTx 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 25-9: SMTxTMRU: SMTx 定时器寄存器——最高字节

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——最高字节

PIC12(L)F1612/16(L)F1613

寄存器 25-10: SMTxCPRL: SMTx 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 25-11: SMTxCPRH: SMTx 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 25-12: SMTxCPRU: SMTx 捕捉周期寄存器——最高字节

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——最高字节

PIC12(L)F1612/16(L)F1613

寄存器 25-13: SMTxCPWL: SMTx 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 25-14: SMTxCPWH: SMTx 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 25-15: SMTxCPWU: SMTx 捕捉脉冲宽度寄存器——最高字节

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——最高字节

PIC12(L)F1612/16(L)F1613

寄存器 25-16: SMTxPRL: SMTx 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>**: 匹配产生上溢的 SMT 定时器值的有效位——低字节

寄存器 25-17: SMTxPRH: SMTx 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>**: 匹配产生上溢的 SMT 定时器值的有效位——高字节

寄存器 25-18: SMTxPRU: SMTx 周期寄存器——最高字节

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>**: 匹配产生上溢的 SMT 定时器值的有效位——最高字节

PIC12(L)F1612/16(L)F1613

表 25-2: 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
SMT1CLK	—	—	—	—	—	CSEL<2:0>			291
SMT1CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT1PS<1:0>		288
SMT1CON1	SMT1GO	REPEAT	—	—	MODE<3:0>				289
SMT1CPRH	SMT1CPR<15:8>								295
SMT1CPRL	SMT1CPR<7:0>								295
SMT1CPRU	SMT1CPR<23:16>								295
SMT1CPWH	SMT1CPW<15:8>								296
SMT1CPWL	SMT1CPW<7:0>								296
SMT1CPWU	SMT1CPW<23:16>								296
SMT1PRH	SMT1PR<15:8>								297
SMT1PRL	SMT1PR<7:0>								297
SMT1PRU	SMT1PR<23:16>								297
SMT1SIG	—	—	—	—	—	SSEL<2:0>			293
SMT1STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	290
SMT1TMRH	SMT1TMR<15:8>								294
SMT1TMRL	SMT1TMR<7:0>								294
SMT1TMRU	SMT1TMR<23:16>								294
SMT1WIN	—	—	—	—	WSEL<3:0>				292
SMT2CLK	—	—	—	—	—	CSEL<2:0>			291
SMT2CON0	EN	—	STP	WPOL	SPOL	CPOL	SMT2PS<1:0>		288
SMT2CON1	SMT2GO	REPEAT	—	—	MODE<3:0>				289
SMT2CPRH	SMT2CPR<15:8>								295
SMT2CPRL	SMT2CPR<7:0>								295
SMT2CPRU	SMT2CPR<23:16>								295
SMT2CPWH	SMT2CPW<15:8>								296
SMT2CPWL	SMT2CPW<7:0>								296
SMT2CPWU	SMT2CPW<23:16>								296
SMT2PRH	SMT2PR<15:8>								297
SMT2PRL	SMT2PR<7:0>								297
SMT2PRU	SMT2PR<23:16>								297
SMT2SIG	—	—	—	—	—	SSEL<2:0>			293
SMT2STAT	CPRUP	CPWUP	RST	—	—	TS	WS	AS	290
SMT2TMRH	SMT2TMR<15:8>								294
SMT2TMRL	SMT2TMR<7:0>								294
SMT2TMRU	SMT2TMR<23:16>								294
SMT2WIN	—	—	—	—	WSEL<3:0>				292

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

26.0 在线串行编程 (ICSP™)

ICSP™ 编程允许用户在生产电路板时使用未编程器件。编程可以在组装流程之后完成，从而可以使用最新版本的固件或者定制固件对器件编程。ICSP™ 编程需要 5 个引脚：

- ICSPCLK
- ICSPDAT
- $\overline{\text{MCLR}}/\text{VPP}$
- VDD
- VSS

在编程 / 校验模式下，通过串行通信对程序存储器、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)F1501/PIC16(L)F150X Memory Programming Specification” (DS41573)。

26.1 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 $\overline{\text{MCLR}}/\text{VPP}$ 上的电压升至 V_{IH}^{H} ，将器件置于高电压编程进入模式。

26.2 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} 。

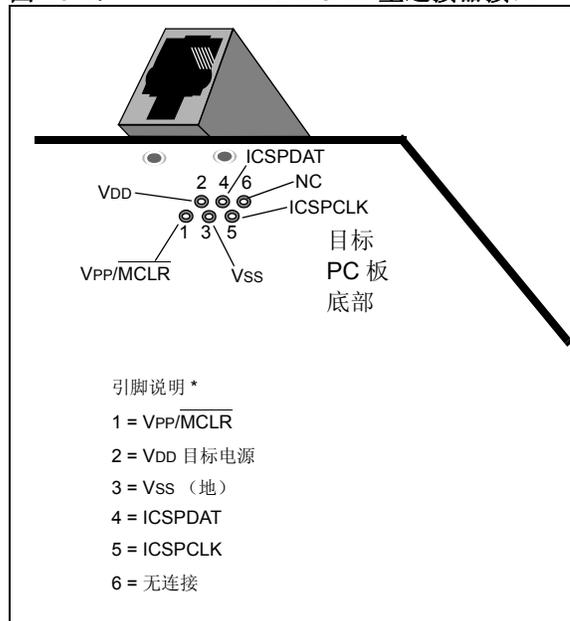
如果使能了低电压编程 (LVP = 1)，则 $\overline{\text{MCLR}}$ 复位功能会被自动使能，无法禁止。更多信息，请参见第 6.5 节“MCLR”。

LVP 位只能通过使用高电压编程模式重新设定为 0。

26.3 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 26-1。

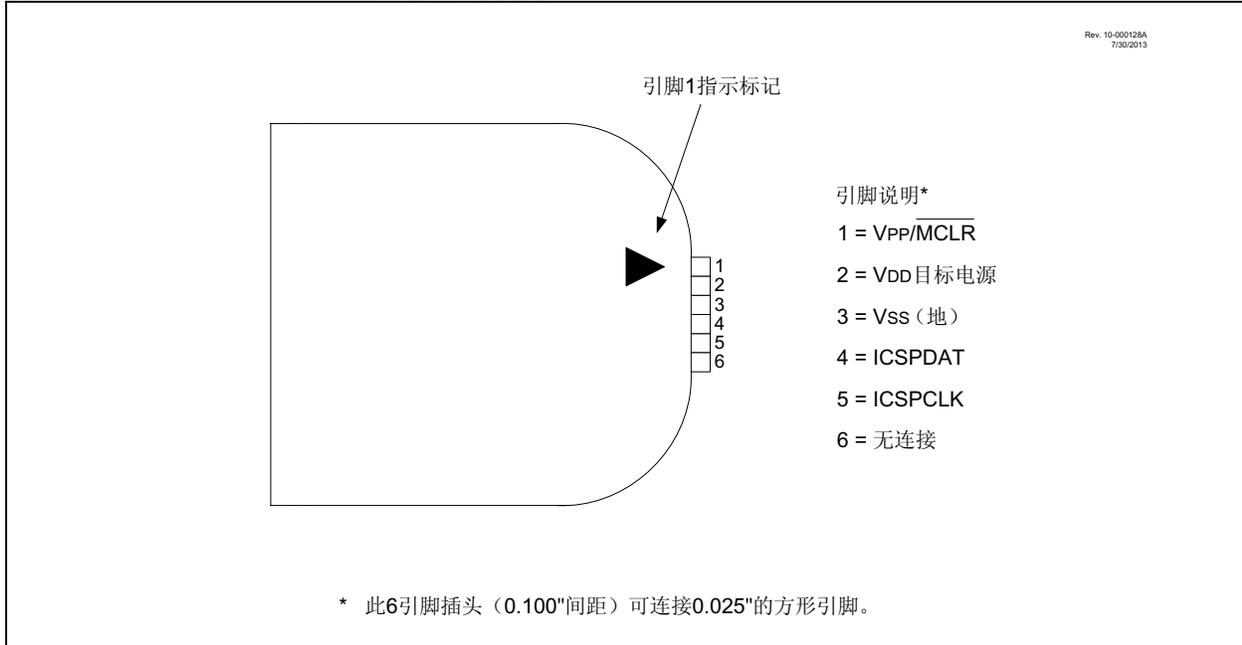
图 26-1: ICD RJ-11 型连接器接口



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 26-2。

PIC12(L)F1612/16(L)F1613

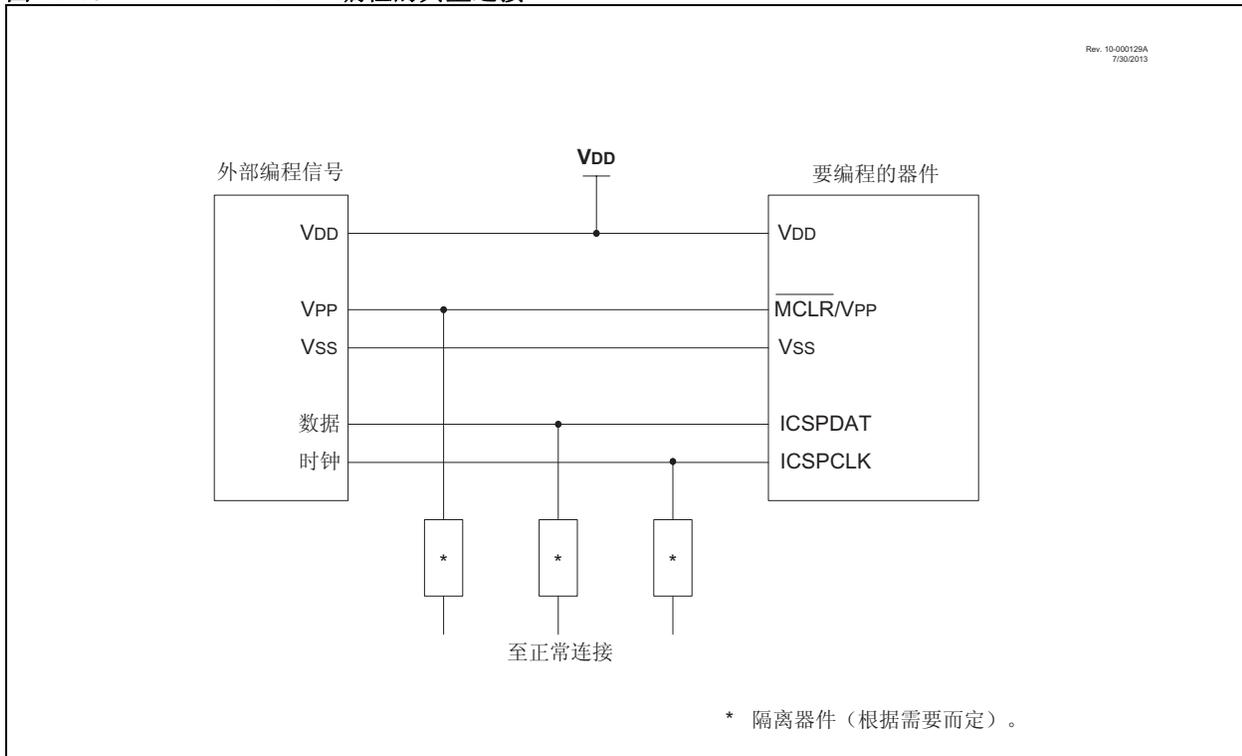
图 26-2: PICkit™ 编程器型连接器接口



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 26-3。

图 26-3: ICSP™ 编程的典型连接



PIC12(L)F1612/16(L)F1613

27.0 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 27-3 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。

27.1 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 27-1: 操作码字段说明

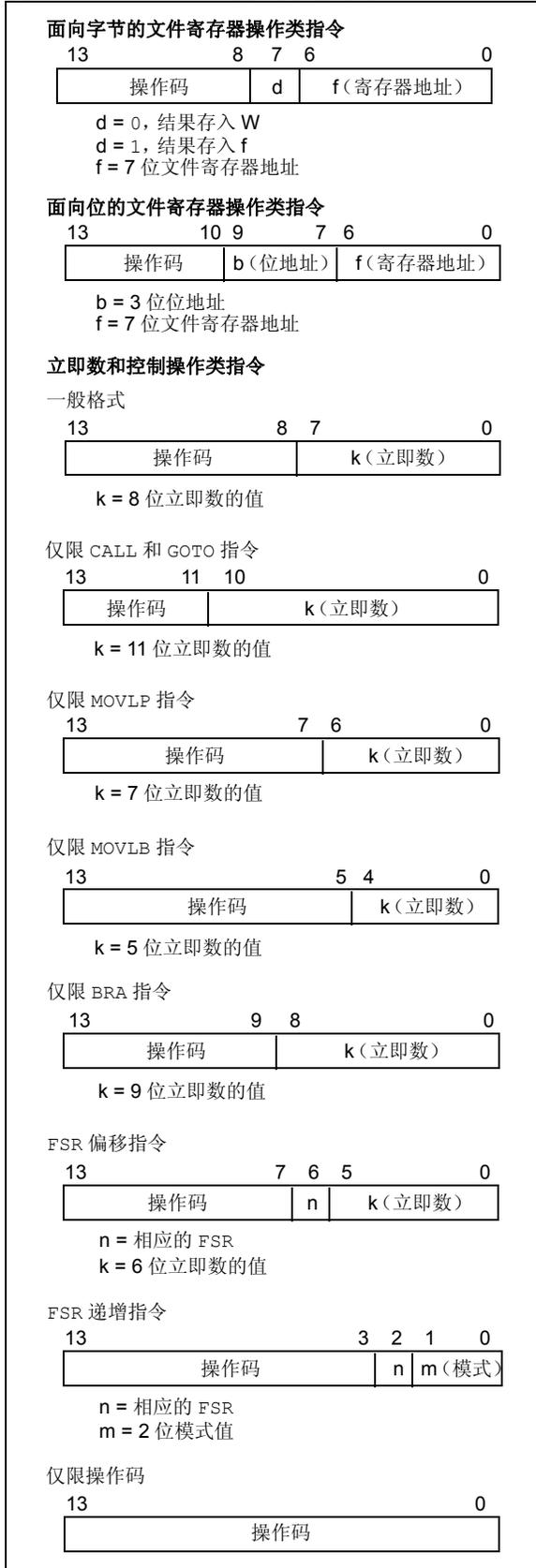
字段	说明
f	文件寄存器地址（0x00 至 0x7F）
W	工作寄存器（累加器）
b	8 位文件寄存器内的位地址
k	立即数字段、常数或标号
x	无关位（= 0 或 1）。 汇编器将产生 x = 0 的代码。为了与所有的 Microchip 软件工具兼容，建议使用这种形式。
d	目标寄存器选择；d = 0：结果存入 W， d = 1：结果存入文件寄存器 f。 默认值 d = 1。
n	FSR 或 INDF 编号。（0-1）
mm	预 / 后递增 / 递减模式选择

表 27-2: 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

PIC12(L)F1612/16(L)F1613

图 27-1: 指令的通用格式



PIC12(L)F1612/16(L)F1613

表 27-3: 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 与 f 相加	1	00	0111	dfff	ffff	C、DC 和 Z	2
ADDWFC	f, d	W 与 f 相加 (带进位)	1	11	1101	dfff	ffff	C、DC 和 Z	2
ANDWF	f, d	W 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
CLRWF	—	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 作逻辑或运算	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	传送 f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	将 W 的内容传送到 f	1	00	0000	1fff	ffff	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	立即数与 W 相加	1	11	1110	kkkk	kkkk	C、DC 和 Z	
ANDLW	k	立即数和 W 作逻辑与运算	1	11	1001	kkkk	kkkk	Z	
IORLW	k	立即数和 W 作逻辑或运算	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	将立即数传送到 BSR	1	00	0000	001k	kkkk		
MOVLP	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。

PIC12(L)F1612/16(L)F1613

表 27-3: 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	2	11	001k	kkkk	kkkk		
BRW	-	2	00	0000	0000	1011		
CALL	k	2	10	0kkk	kkkk	kkkk		
CALLW	-	2	00	0000	0000	1010		
GOTO	k	2	10	1kkk	kkkk	kkkk		
RETFIE	k	2	00	0000	0000	1001		
RETLW	k	2	11	0100	kkkk	kkkk		
RETURN	-	2	00	0000	0000	1000		
固有操作类指令								
CLRWDT	-	1	00	0000	0110	0100	\overline{TO} 和 \overline{PD}	
NOP	-	1	00	0000	0000	0000		
OPTION	-	1	00	0000	0110	0010		
RESET	-	1	00	0000	0000	0001		
SLEEP	-	1	00	0000	0110	0011	\overline{TO} 和 \overline{PD}	
TRIS	f	1	00	0000	0110	0fff		
优化的 C 编译器指令								
ADDFSR	n, k	1	11	0001	0nkk	kkkk		
MOVIW	n mm	1	00	0000	0001	0nmm	Z	2, 3
	k[n]	1	11	1111	0nkk	1nmm	Z	2
MOVWI	n mm	1	00	0000	0001	kkkk		2, 3
	k[n]	1	11	1111	1nkk			2

- 注 1: 如果程序计数器 (PC) 被修改或条件测试结果为真, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。
 注 2: 如果该指令寻址的是 INDF 寄存器, 并且相应 FSR 的 MSb 置 1, 则该指令将需要一个额外的指令周期。
 注 3: 请参见 MOVIW 和 MOVWI 指令说明表。

27.2 指令说明

ADDFSR	将立即数与 FSRn 相加
语法:	[标号] ADDFSR FSRn, k
操作数:	-32 ≤ k ≤ 31 n ∈ [0, 1]
操作:	FSR(n) + k → 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。

ADDLW	立即数与 W 相加
语法:	[标号] ADDLW k
操作数:	0 ≤ k ≤ 255
操作:	(W) + k → (W)
受影响的状态位:	C、DC 和 Z
说明:	将 W 寄存器的内容与 8 位立即数 k 相加，结果存入 W 寄存器。

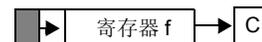
ADDWF	W 和 f 相加
语法:	[标号] ADDWF f,d
操作数:	0 ≤ f ≤ 127 d ∈ [0,1]
操作:	(W) + (f) → (目标寄存器)
受影响的状态位:	C、DC 和 Z
说明:	将 W 寄存器的内容与寄存器 f 的内容相加。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ADDWFC	W 与 f 相加 (带进位)
语法:	[标号] ADDWFC f{,d}
操作数:	0 ≤ f ≤ 127 d ∈ [0,1]
操作:	(W) + (f) + (C) → dest
受影响的状态位:	C、DC 和 Z
说明:	将 W 的内容、进位标志位与数据存储单元 f 的内容相加。如果 d 为 0，结果放入 W 寄存器。如果 d 为 1，结果放入数据存储单元 f。

ANDLW	立即数和 W 作逻辑与运算
语法:	[标号] ANDLW k
操作数:	0 ≤ k ≤ 255
操作:	(W) .AND.(k) → (W)
受影响的状态位:	Z
说明:	将 W 寄存器的内容与 8 位立即数 k 进行逻辑与运算。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 ≤ f ≤ 127 d ∈ [0,1]
操作:	(W) .AND.(f) → (目标寄存器)
受影响的状态位:	Z
说明:	将 W 寄存器的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ASRF	算术右移
语法:	[标号] ASRF f{,d}
操作数:	0 ≤ f ≤ 127 d ∈ [0,1]
操作:	(f<7>) → dest<7> (f<7:0>) → dest<6:0> (f<0>) → 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，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。



PIC12(L)F1612/16(L)F1613

BCF 将 f 中的某位清零

语法: [标号] BCF f,b
操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
操作: $0 \rightarrow (f)$
受影响的状态位: 无
说明: 将寄存器 f 中的位 b 清零。

BTFSF 测试 f 中的某位, 为 0 则跳过

语法: [标号] BTFSF f,b
操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
操作: 如果 $(f) = 0$, 则跳过
受影响的状态位: 无
说明: 如果寄存器 f 的位 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。

BTFSF 测试 f 中的某位, 为 1 则跳过

语法: [标号] BTFSF f,b
操作数: $0 \leq f \leq 127$
 $0 \leq b < 7$
操作: 如果 $(f) = 1$, 则跳过
受影响的状态位: 无
说明: 如果寄存器 f 的位 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。

BSF 将 f 中的某位置 1

语法: [标号] BSF f,b
操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
操作: $1 \rightarrow (f)$
受影响的状态位: 无
说明: 将寄存器 f 的位 b 置 1。

PIC12(L)F1612/16(L)F1613

CALL 调用子程序

语法: [标号] CALL k
操作数: $0 \leq k \leq 2047$
操作: $(PC) + 1 \rightarrow TOS,$
 $k \rightarrow PC<10:0>,$
 $(PCLATH<6:3>) \rightarrow PC<14:11>$
受影响的状态位: 无
说明: 调用子程序。首先, 将返回地址 (PC + 1) 压入堆栈。11 位直接地址值被装入 PC 的 <10:0> 位。PC 的高位值从 PCLATH 装入。CALL 是一条双周期指令。

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: $00h \rightarrow WDT$
 $0 \rightarrow \overline{WDT}$ 预分频器
 $1 \rightarrow \overline{TO}$
 $1 \rightarrow \overline{PD}$
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 TO 和 PD 均被置 1。

COMF 对 f 取反

语法: [标号] COMF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $\overline{(f)} \rightarrow (\text{目标寄存器})$
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。

PIC12(L)F1612/16(L)F1613

DECFSZ f 递减 1, 为 0 则跳过

语法: [标号] DECFSZ f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) - 1 \rightarrow$ (目标寄存器);
如果结果 = 0 则跳过
受影响的状态位: 无
说明: 将寄存器 f 的内容递减 1。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。
如果结果为 1, 则执行下一条指令。如果结果为 0, 代之执行一条 NOP 指令, 使之成为一条双周期指令。

INCFSZ f 递增 1, 为 0 则跳过

语法: [标号] INCFSZ f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) + 1 \rightarrow$ (目标寄存器),
如果结果 = 0 则跳过
受影响的状态位: 无
说明: 将寄存器 f 的内容递增 1。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。
如果结果为 1, 则执行下一条指令。如果结果为 0, 代之执行一条 NOP 指令, 使之成为一条双周期指令。

GOTO 无条件跳转

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。

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。

INCF f 递增 1

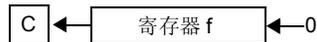
语法: [标号] INCF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) + 1 \rightarrow$ (目标寄存器)
受影响的状态位: Z
说明: 将寄存器 f 的内容递增 1。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

IORWF W 和 f 作逻辑或运算

语法: [标号] IORWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(W) .OR. (f) \rightarrow$ (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。

PIC12(L)F1612/16(L)F1613

LSLF	逻辑左移
语法:	[标号] LSLF f {,d}
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f<7>) \rightarrow C$ $(f<6:0>) \rightarrow \text{dest}<7:1>$ $0 \rightarrow \text{dest}<0>$
受影响的状态位:	C 和 Z
说明:	将寄存器 f 的内容连同进位标志位一起左移 1 位。0 移入 LSb。如果 d 为 0，结果存入 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:	$0 \rightarrow \text{dest}<7>$ $(f<7:0>) \rightarrow \text{dest}<6:0>$ $(f<0>) \rightarrow C$
受影响的状态位:	C 和 Z
说明:	将寄存器 f 的内容连同进位标志位一起右移 1 位。0 移入 MSb。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。



MOVF	传送 f
语法:	[标号] MOVF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	根据 d 的状态，将寄存器 f 的内容传送到目标寄存器。如果 d = 0，目标寄存器为 W 寄存器。如果 d = 1，目标寄存器为文件寄存器 f 本身。由于状态标志位 Z 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:	MOVF FSR, 0

执行指令后

W = FSR 寄存器的值

Z = 1

PIC12(L)F1612/16(L)F1613

MOVIW 将 INDFn 的内容传送到 W

语法: [标号] MOVIW ++FSRn
 [标号] MOVIW --FSRn
 [标号] MOVIW FSRn++
 [标号] MOVIW FSRn--
 [标号] MOVIW k[FSRn]

操作数: $n \in [0,1]$
 $mm \in [00,01, 10, 11]$
 $-32 \leq k \leq 31$

操作: INDFn \rightarrow W
 有效地址通过以下方式确定

- FSR + 1 (预递增)
- FSR - 1 (预递减)
- FSR + k (相对偏移)

在传送之后, FSR 值将为以下之一:

- FSR + 1 (全部递增)
- FSR - 1 (全部递减)
- 不变

受影响的状态位: Z

模式	语法	mm
预递增	++FSRn	00
预递减	--FSRn	01
后递增	FSRn++	10
后递减	FSRn--	11

说明: 该指令用于在 W 和一个间接寄存器 (INDFn) 之间传送数据。在该传送操作之前 / 之后, 将通过预 / 后递增 / 递减指针来更新指针 (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: $k \rightarrow$ BSR

受影响的状态位: 无

说明: 将 5 位立即数 k 装入存储区选择寄存器 (BSR)。

MOVLW 将立即数传送到 PCLATH

语法: [标号] MOVLW k

操作数: $0 \leq k \leq 127$

操作: $k \rightarrow$ PCLATH

受影响的状态位: 无

说明: 将 7 位立即数 k 装入 PCLATH 寄存器。

MOVLW 将立即数传送到 W

语法: [标号] MOVLW k

操作数: $0 \leq k \leq 255$

操作: $k \rightarrow$ (W)

受影响的状态位: 无

说明: 将 8 位立即数 k 装入 W 寄存器。其余无关位均汇编为 0。

指令字数: 1

指令周期数: 1

示例: `MOVLW 0x5A`
 执行指令后
 $W = 0x5A$

MOVWF 将 W 的内容传送到 f

语法: [标号] MOVWF f

操作数: $0 \leq f \leq 127$

操作: (W) \rightarrow (f)

受影响的状态位: 无

说明: 将 W 寄存器的数据传送到寄存器 f。

指令字数: 1

指令周期数: 1

示例: `MOVWF OPTION_REG`
 执行指令前
`OPTION_REG = 0xFF`
`W = 0x4F`
 执行指令后
`OPTION_REG = 0x4F`
`W = 0x4F`

PIC12(L)F1612/16(L)F1613

MOVWI 将 W 的内容传送到 INDFn

语法: [标号] MOVWI ++FSRn
 [标号] MOVWI --FSRn
 [标号] MOVWI FSRn++
 [标号] MOVWI FSRn--
 [标号] MOVWI k[FSRn]

操作数: $n \in [0,1]$
 $mm \in [00,01, 10, 11]$
 $-32 \leq k \leq 31$

操作: $W \rightarrow \text{INDFn}$
 有效地址通过以下方式确定

- 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: $(W) \rightarrow \text{OPTION_REG}$

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。

PIC12(L)F1612/16(L)F1613

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:	<pre> RETFIE 中断后 PC = TOS GIE = 1 </pre>

RETLW	返回并将立即数送入 W
语法:	[标号] RETLW k
操作数:	0 ≤ k ≤ 255
操作:	k → (W); TOS → PC
受影响的状态位:	无
说明:	将 8 位立即数 k 装入 W 寄存器。将栈顶内容 (返回地址) 装入程序计数器。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	<pre> CALL TABLE;W contains table ;offset value • ;W now has table value • • ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • • RETLW kn ; End of table </pre>
	<p>执行指令前</p> <p>W = 0x07</p> <p>执行指令后</p> <p>W = k8 的值</p>

RETURN	从子程序返回
语法:	[标号] RETURN
操作数:	无
操作:	TOS → PC
受影响的状态位:	无
说明:	从子程序返回。执行出栈操作, 将栈顶 (TOS) 内容装入程序计数器。这是一条双周期指令。

RLF	f 带进位循环左移															
语法:	[标号] RLF f,d															
操作数:	0 ≤ f ≤ 127 d ∈ [0,1]															
操作:	参见如下说明															
受影响的状态位:	C															
说明:	将寄存器 f 的内容连同进位标志位一起循环左移 1 位。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。															
指令字数:	1															
指令周期数:	1															
示例:	<pre> RLF REG1,0 </pre> <p>执行指令前</p> <table> <tr><td>REG1</td><td>=</td><td>1110 0110</td></tr> <tr><td>C</td><td>=</td><td>0</td></tr> </table> <p>执行指令后</p> <table> <tr><td>REG1</td><td>=</td><td>1110 0110</td></tr> <tr><td>W</td><td>=</td><td>1100 1100</td></tr> <tr><td>C</td><td>=</td><td>1</td></tr> </table>	REG1	=	1110 0110	C	=	0	REG1	=	1110 0110	W	=	1100 1100	C	=	1
REG1	=	1110 0110														
C	=	0														
REG1	=	1110 0110														
W	=	1100 1100														
C	=	1														

PIC12(L)F1612/16(L)F1613

RRF f 带进位循环右移

语法: [标号] RRF f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
 操作: 参见如下说明
 受影响的状态位: C
 说明: 将寄存器 f 的内容连同进位标志位一起循环右移 1 位。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。



SLEEP 进入休眠模式

语法: [标号] SLEEP
 操作数: 无
 操作: $00h \rightarrow WDT$,
 $0 \rightarrow \overline{WDT}$ 预分频器,
 $1 \rightarrow \overline{TO}$,
 $0 \rightarrow \overline{PD}$
 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	$W\langle 3:0 \rangle > k\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq k\langle 3:0 \rangle$

SUBWF f 减去 W

语法: [标号] SUBWF f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
 操作: $(f) - (W) \rightarrow (\text{目标寄存器})$
 受影响的状态位: C、DC 和 Z
 说明: 用寄存器 f 的内容减去 W 寄存器的内容 (通过二进制补码方式进行运算)。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W\langle 3:0 \rangle > f\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq f\langle 3:0 \rangle$

SUBWFB f 减去 W (带借位)

语法: SUBWFB f {,d}
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
 操作: $(f) - (W) - (\overline{B}) \rightarrow \text{dest}$
 受影响的状态位: C、DC 和 Z
 说明: 用 f 寄存器的内容减去 W 的内容和借位标志 (进位) (通过二进制补码方式进行运算)。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

PIC12(L)F1612/16(L)F1613

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>) → (目标寄存器 <7:4>),
(f<7:4>) → (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, 结果存回寄存器 f。

TRIS 将 W 的内容装入 TRIS 寄存器

语法: [标号] TRIS f
操作数: $5 \leq f \leq 7$
操作: (W) → TRIS 寄存器 f
受影响的状态位: 无
说明: 将 W 寄存器的数据传送到 TRIS 寄存器。
当 f = 5 时, 装入 TRISA。
当 f = 6 时, 装入 TRISB。
当 f = 7 时, 装入 TRISC。

XORLW 立即数和 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 → (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) → (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。

PIC12(L)F1612/16(L)F1613

28.0 电气规范

28.1 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 V _{SS} 的电压	
VDD 引脚	
PIC12F1612/16F1613	-0.3V 至 +6.5V
PIC12LF1612/16LF1613	-0.3V 至 +4.0V
MCLR 引脚.....	-0.3V 至 +9.0V
所有其他引脚.....	-0.3V 至 (V _{DD} + 0.3V)
最大电流	
V _{SS} 引脚 (1)	
-40°C ≤ T _A ≤ +85°C	170 mA
-40°C ≤ T _A ≤ +125°C.....	70 mA
VDD 引脚 (1)	
-40°C ≤ T _A ≤ +85°C	170 mA
-40°C ≤ T _A ≤ +125°C	70 mA
任意 I/O 引脚	±25 mA
钳位电流, I _K (V _{PIN} < 0 或 V _{PIN} > V _{DD})	±20 mA

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 28-6: “温度特性” 来计算器件规范值。

† 注: 如果器件运行条件超过上述各项绝对最大值, 可能对器件造成永久性损坏。上述值仅为运行条件极大值, 我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下, 其可靠性会受到影响。

PIC12(L)F1612/16(L)F1613

28.2 标准工作条件

所有器件的标准工作条件定义如下：

工作电压： $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$

工作温度： $T_{A_MIN} \leq T_A \leq T_{A_MAX}$

V_{DD}——供电电压工作范围⁽¹⁾

PIC12LF1612/16LF1613

V_{DDMIN} (Fosc ≤ 16 MHz) +1.8V

V_{DDMIN} (Fosc ≤ 32 MHz) +2.5V

V_{DDMAX} +3.6V

PIC12F1612/16F1613

V_{DDMIN} (Fosc ≤ 16 MHz) +2.3V

V_{DDMIN} (Fosc ≤ 32 MHz) +2.5V

V_{DDMAX} +5.5V

T_A——工作环境温度范围

工业级温度

T_{A_MIN} -40°C

T_{A_MAX} +85°C

扩展级温度

T_{A_MIN} -40°C

T_{A_MAX} +125°C

注 1: 请参见参数 [D001](#)，DS 特性：供电电压。

PIC12(L)F1612/16(L)F1613

图 28-1: 电压频率关系图, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$, 仅限 PIC12F1612/16F1613

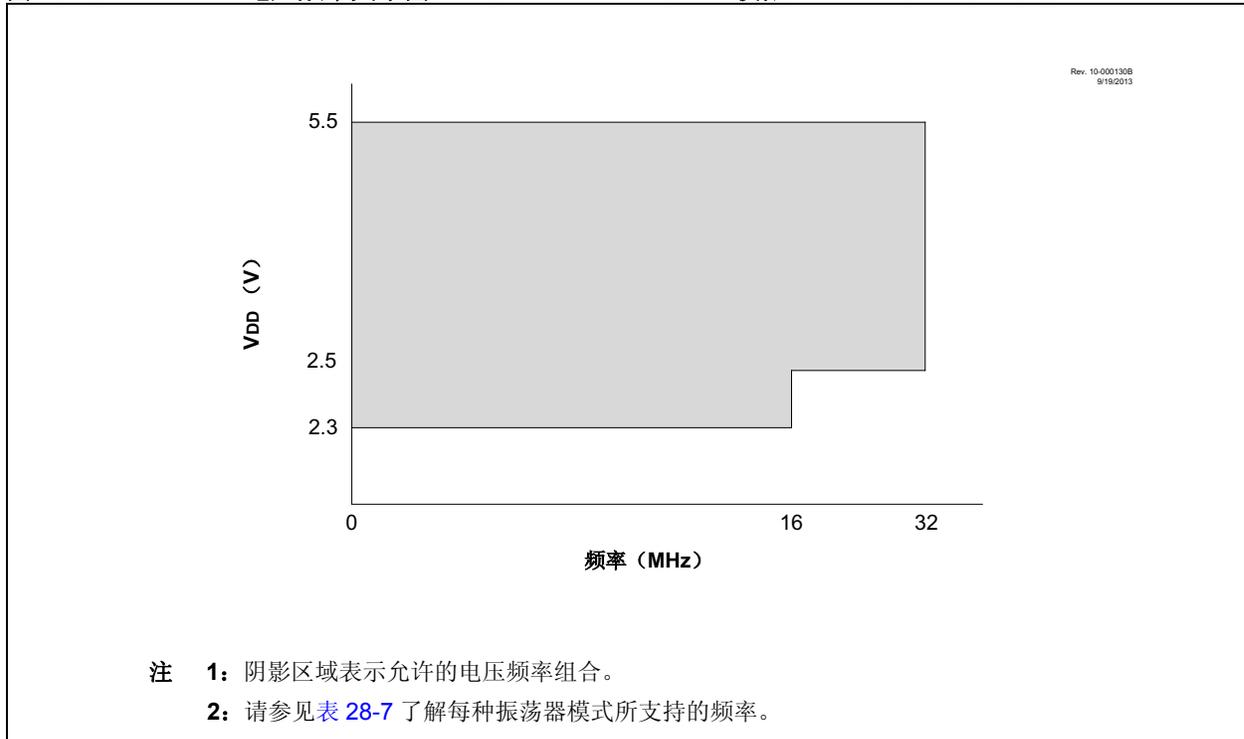
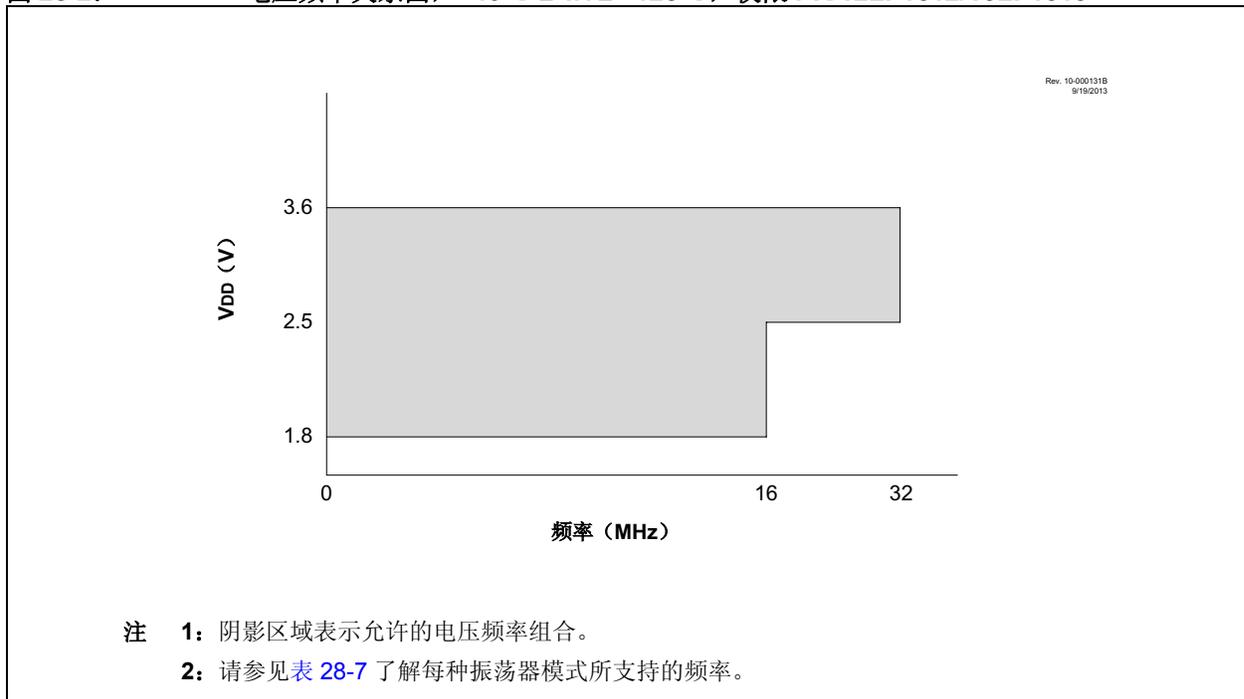


图 28-2: 电压频率关系图, $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$, 仅限 PIC12LF1612/16LF1613



PIC12(L)F1612/16(L)F1613

28.3 直流特性

表 28-1: 供电电压

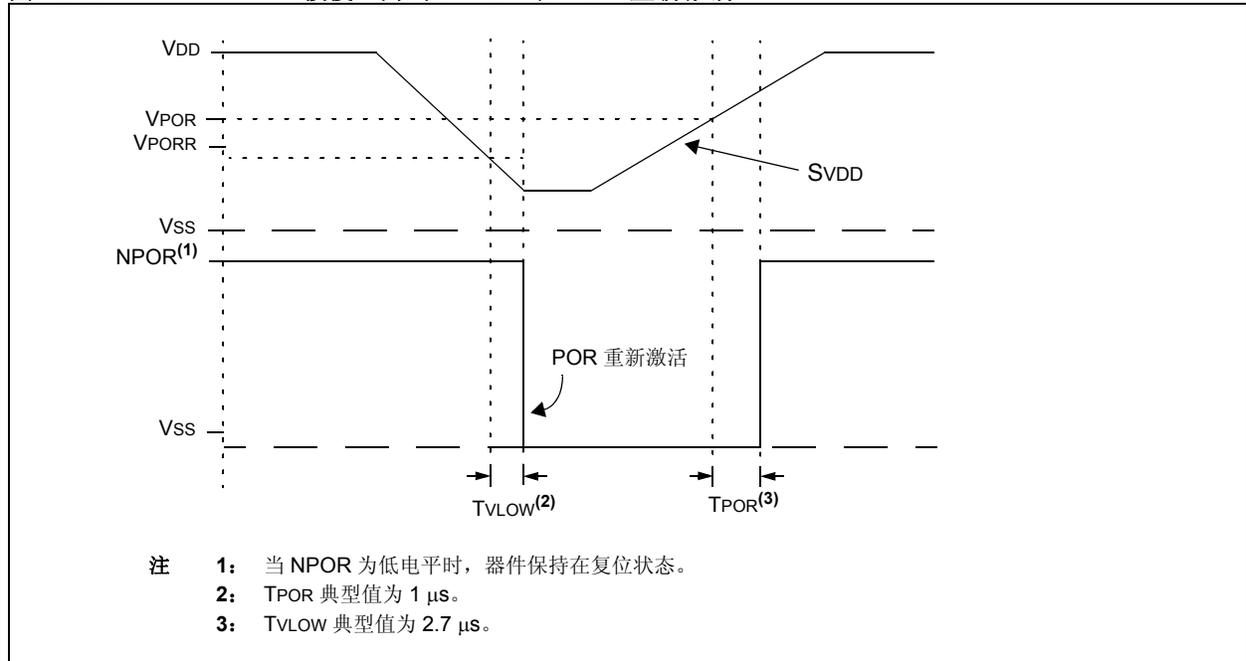
PIC12LF1612/16LF1613		标准工作条件 (除非另外声明)					
PIC12F1612/16F1613							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D001	VDD	供电电压	VDDMIN	—	VDDMAX		
			1.8	—	3.6	V	Fosc ≤ 16 MHz
			2.5	—	3.6	V	Fosc ≤ 32 MHz
D001			2.3	—	5.5	V	Fosc ≤ 16 MHz
			2.5	—	5.5	V	Fosc ≤ 32 MHz
D002*	VDR	RAM 数据保持电压 ⁽¹⁾	1.5	—	—	V	器件处于休眠模式
			1.7	—	—	V	器件处于休眠模式
D002A*	VPOR	上电复位释放电压 ⁽²⁾	—	1.6	—	V	
			—	1.6	—	V	
D002B*	VPORR*	上电复位重新激活电压 ⁽²⁾	—	0.8	—	V	
			—	1.5	—	V	
D003	VFVR	固定参考电压	—	1.024	—	V	-40°C ≤ Ta ≤ +85°C
			—	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
			-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
			-4	—	+4	%	1x VFVR, VDD ≥ 2.5V 2x VFVR, VDD ≥ 2.5V 4x VFVR, VDD ≥ 4.75V
D004*	SVDD	VDD 上升速率 ⁽²⁾	0.05	—	—	V/ms	请确保正确释放上电复位信号。
			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 28-3, 可以看到 VDD 缓慢上升时, POR 和 POR 重新激活。

图 28-3: VDD 缓慢上升时, POR 和 POR 重新激活



PIC12(L)F1612/16(L)F1613

表 28-2: 供电电流 (I_{DD}) (1,2)

PIC12LF1612/16LF1613		标准工作条件 (除非另外声明)					
PIC12F1612/16F1613							
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						V _{DD}	注
D013		—	30	65	μA	1.8	FOSC = 1 MHz, 外部时钟 (ECM), 中等功耗模式
		—	55	100	μA	3.0	
D013		—	65	110	μA	2.3	FOSC = 1 MHz, 外部时钟 (ECM), 中等功耗模式
		—	85	140	μA	3.0	
		—	115	190	μA	5.0	
D014		—	115	190	μA	1.8	FOSC = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	210	310	μA	3.0	
D014		—	180	270	μA	2.3	FOSC = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	240	365	μA	3.0	
		—	295	460	μA	5.0	
D015		—	9.6	36	μA	1.8	FOSC = 31 kHz, LFINTOSC, -40°C ≤ T _A ≤ +85°C
		—	16.2	60	μA	3.0	
D015		—	39	84	μA	2.3	FOSC = 31 kHz, LFINTOSC, -40°C ≤ T _A ≤ +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	660	μA	1.8	FOSC = 8 MHz, HFINTOSC
		—	630	970	μA	3.0	
D017*		—	530	750	μA	2.3	FOSC = 8 MHz, HFINTOSC
		—	660	1100	μA	3.0	
		—	730	1200	μA	5.0	
D018		—	600	940	μA	1.8	FOSC = 16 MHz, HFINTOSC
		—	970	1400	μA	3.0	
D018		—	780	1200	μA	2.3	FOSC = 16 MHz, HFINTOSC
		—	1000	1550	μA	3.0	
		—	1090	1700	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 I_{DD} 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 V_{SS}; MCLR = V_{DD}; 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。

PIC12(L)F1612/16(L)F1613

表 28-2: 供电电流 (IDD) (1,2) (续)

PIC12LF1612/16LF1613		标准工作条件 (除非另外声明)					
PIC12F1612/16F1613							
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	28	μA	2.3	Fosc = 32 kHz, 外部时钟 (ECL), 低功耗模式
		—	15	31	μA	3.0	
		—	16	36	μA	5.0	
D020C		—	19	35	μA	1.8	Fosc = 500 kHz, 外部时钟 (ECL), 低功耗模式
		—	32	55	μA	3.0	
D020C		—	31	52	μA	2.3	Fosc = 500 kHz, 外部时钟 (ECL), 低功耗模式
		—	38	65	μA	3.0	
		—	44	74	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。

PIC12(L)F1612/16(L)F1613

表 28-3: 掉电电流 (IPD) (1,2)

PIC12LF1612/16LF1613		工作条件: (除非另外声明) 低功耗休眠模式						
PIC12F1612/16F1613		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。

PIC12(L)F1612/16(L)F1613

表 28-3: 掉电电流 (IPD) (1,2) (续)

PIC12LF1612/16LF1613		工作条件: (除非另外声明) 低功耗休眠模式						
PIC12F1612/16F1613		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。

PIC12(L)F1612/16(L)F1613

表 28-4: 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 _{IL}	输入低电压					
		I/O 端口: 带 TTL 缓冲器	—	—	0.8	V	4.5V ≤ V _{DD} ≤ 5.5V
			—	—	0.15 V _{DD}	V	1.8V ≤ V _{DD} ≤ 4.5V
		带施密特触发器缓冲器	—	—	0.2 V _{DD}	V	2.0V ≤ V _{DD} ≤ 5.5V
		MCLR	—	—	0.2 V _{DD}	V	
D040 D040A D041 D042	V _{IH}	输入高电压					
		I/O 端口: 带 TTL 缓冲器	2.0	—	—	V	4.5V ≤ V _{DD} ≤ 5.5V
			0.25 V _{DD} + 0.8	—	—	V	1.8V ≤ V _{DD} ≤ 4.5V
		带施密特触发器缓冲器	0.8 V _{DD}	—	—	V	2.0V ≤ V _{DD} ≤ 5.5V
		MCLR	0.8 V _{DD}	—	—	V	
D060 D061	I _{IL}	输入泄漏电流 ⁽¹⁾					
		I/O 端口	—	± 5	± 125	nA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, 85°C
			—	± 5	± 1000	nA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, 125°C
		MCLR ⁽³⁾	—	± 50	± 200	nA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, 85°C
D070*	I _{PUR}	弱上拉电流					
			25	100	200	μA	V _{DD} = 3.3V, V _{PIN} = V _{SS}
			25	140	300	μA	V _{DD} = 5.0V, V _{PIN} = V _{SS}
D080	V _{OL}	输出低电压 ⁽³⁾					
		I/O 端口	—	—	0.6	V	I _{OL} = 8 mA, V _{DD} = 5V I _{OL} = 6 mA, V _{DD} = 3.3V I _{OL} = 1.8 mA, V _{DD} = 1.8V
D090	V _{OH}	输出高电压 ⁽³⁾					
		I/O 端口	V _{DD} - 0.7	—	—	V	I _{OH} = 3.5 mA, V _{DD} = 5V I _{OH} = 3 mA, V _{DD} = 3.3V I _{OH} = 1 mA, V _{DD} = 1.8V
D101A*	CIO	所有 I/O 引脚	—	—	50	pF	

* 这些参数为特性值, 但未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

- 注
- 1: 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。

PIC12(L)F1612/16(L)F1613

表 28-5: 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 MCLR/VPP 引脚上的电压	8.0	—	9.0	V	(注 2)
D111	IDDP	编程时的供电电流	—	—	10	mA	
D112	VBE	批量擦除时的 VDD	2.7	—	VDDMAX	V	
D113	VPEW	写或行擦除时的 VDD	VDDMIN	—	VDDMAX	V	
D114	IPPPGM	擦除 / 写操作时 MCLR/VPP 上的电流	—	1.0	—	mA	
D115	IDDPGM	擦除 / 写操作时 VDD 上的电流	—	5.0	—	mA	
D121	EP	闪存程序存储器 单元耐擦写能力	10K	—	—	E/W	-40°C ≤ Ta ≤ + 85°C (注 1)
D122	VPRW	读 / 写操作时的 VDD	VDDMIN	—	VDDMAX	V	
D123	TIW	自定时写周期时间	—	2	2.5	ms	
D124	TRETD	特性保持时间	—	40	—	年	假设没有违反其他规范
D125	EHEFC	高耐用性闪存单元	100K	—	—	E/W	0°C ≤ Ta ≤ + 60°C, 最后 128 个地址的低字节

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 自写和块擦除。

注 2: 仅当禁止单电源编程时才需要。

PIC12(L)F1612/16(L)F1613

表 28-6: 温度特性

标准工作条件 (除非另外声明)

参数编号	符号	特性	典型值	单位	条件
TH01	θ_{JA}	热阻 (结点到环境)	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	θ_{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	TJMAX	最高结温	150	°C	
TH04	PD	功耗	—	W	PD = PINTERNAL + Pi/o
TH05	PINTERNAL	内部功耗	—	W	PINTERNAL = IDD x VDD ⁽¹⁾
TH06	Pi/o	I/O 功耗	—	W	Pi/o = $\Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	PDER	降额功耗	—	W	PDER = PDMAX (TJ - TA)/ θ_{JA} ⁽²⁾

注 1: IDD 为不驱动输出引脚上任何负载时使芯片独立运行的电流。

注 2: TA = 环境温度, TJ = 结温

28.4 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

T		
F	频率	T
		时间

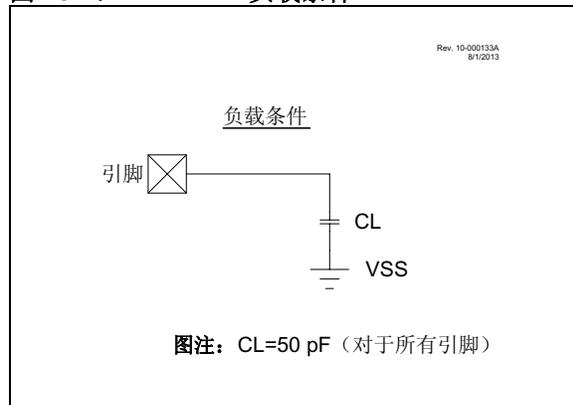
小写字母 (pp) 及其含义：

pp		
cc	CCP1	osc
ck	CLKOUT	rd
cs	\overline{CS}	rw
di	SDIx	sc
do	SDO	ss
dt	数据输入	t0
io	I/O 端口	t1
mc	\overline{MCLR}	wr
		CLKIN
		\overline{RD}
		\overline{RD} 或 \overline{WR}
		SCKx
		\overline{SS}
		T0CKI
		T1CKI
		\overline{WR}

大写字母及其含义：

S		
F	下降	P
H	高	R
I	无效 (高阻)	V
L	低	Z
		周期
		上升
		有效
		高阻

图 28-4: 负载条件



PIC12(L)F1612/16(L)F1613

图 28-5: 时钟时序

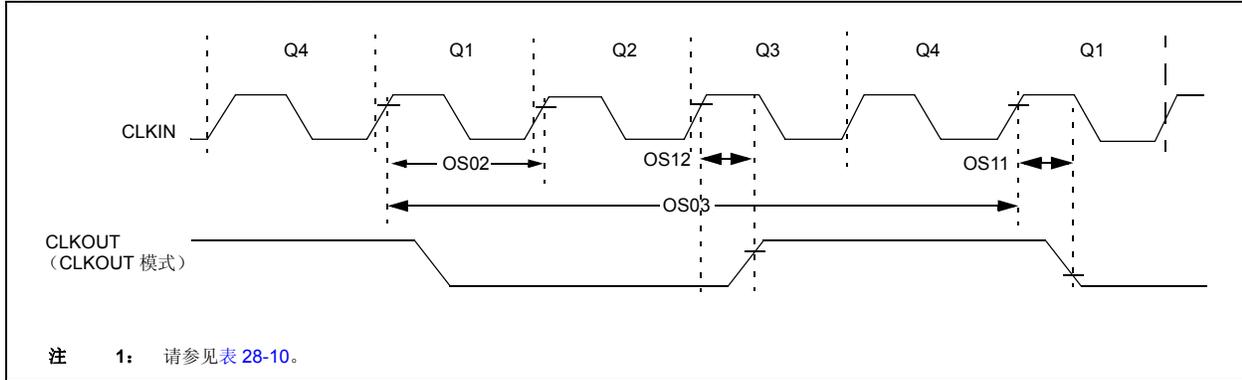


表 28-7: 时钟振荡器时序要求

标准工作条件 (除非另外声明)

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS01	Fosc	外部 CLKIN 频率 ⁽¹⁾	DC	—	0.5	MHz	外部时钟 (ECL)
			DC	—	4	MHz	外部时钟 (ECM)
			DC	—	32	MHz	外部时钟 (ECH)
OS02	Tosc	外部 CLKIN 周期 ⁽¹⁾	31.25	—	∞	ns	外部时钟 (EC)
OS03	Tcy	指令周期 ⁽¹⁾	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)。

PIC12(L)F1612/16(L)F1613

表 28-8: 振荡器参数

标准工作条件 (除非另外声明)								
参数编号	符号	特性	频率容差	最小值	典型值 †	最大值	单位	条件
OS08	HFosc	内部已校准的 HFINTOSC 频率 ⁽¹⁾	±2%	—	16.0	—	MHz	V _{DD} = 3.0V, T _A = 25°C (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 ≤ T _A ≤ +125°C

* 这些参数为特性值, 但未经测试。

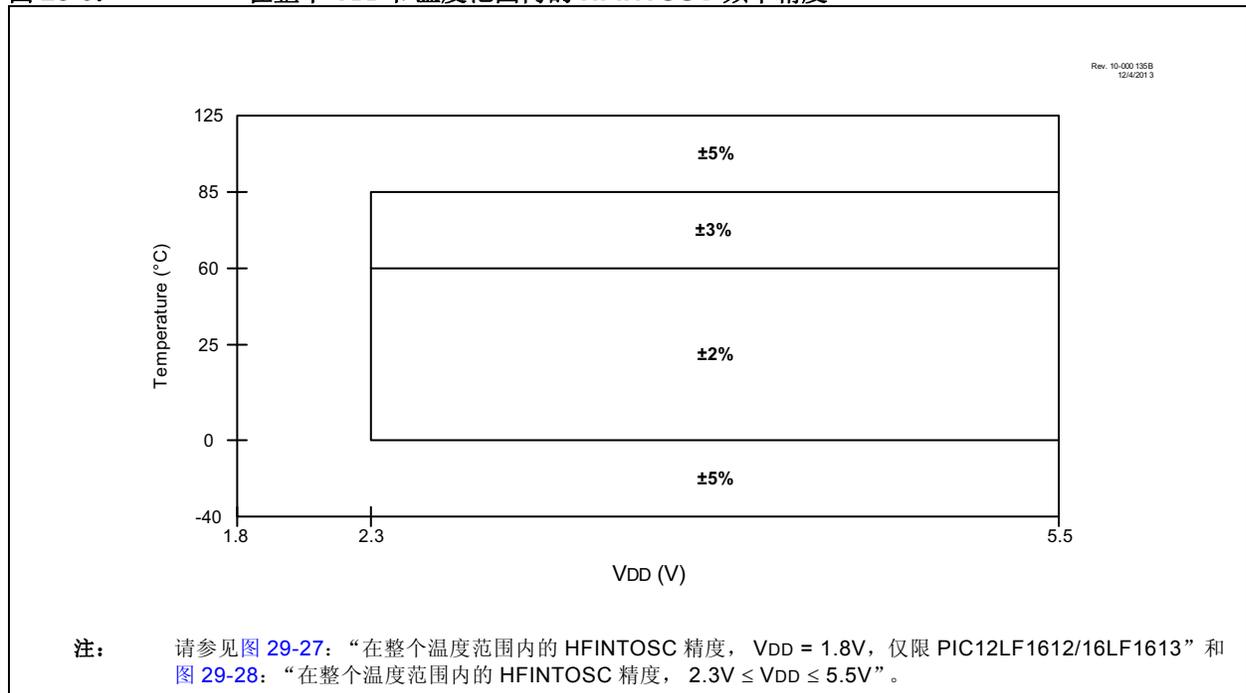
† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

注 1: 为了确保振荡器频率容差, 必须尽可能靠近器件, 在 V_{DD} 和 V_{SS} 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 28-6: “在整个器件 V_{DD} 和温度范围内的 HFINTOSC 频率精度”、图 29-27: “在整个温度范围内的 HFINTOSC 精度, V_{DD} = 1.8V, 仅限 PIC12LF1612/16LF1613”和图 29-28: “在整个温度范围内的 HFINTOSC 精度, 2.3V ≤ V_{DD} ≤ 5.5V”。

3: 请参见图 29-25: “在整个 V_{DD} 和温度范围内的 LFINTOSC 频率, 仅限 PIC12LF1612/16LF1613”和图 29-26: “在整个 V_{DD} 和温度范围内的 LFINTOSC 频率, 仅限 PIC12F1612/16F1613”。

图 28-6: 在整个 V_{DD} 和温度范围内的 HFINTOSC 频率精度



注: 请参见图 29-27: “在整个温度范围内的 HFINTOSC 精度, V_{DD} = 1.8V, 仅限 PIC12LF1612/16LF1613”和图 29-28: “在整个温度范围内的 HFINTOSC 精度, 2.3V ≤ V_{DD} ≤ 5.5V”。

PIC12(L)F1612/16(L)F1613

表 28-9: 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。

PIC12(L)F1612/16(L)F1613

图 28-7: CLKOUT 和 I/O 时序

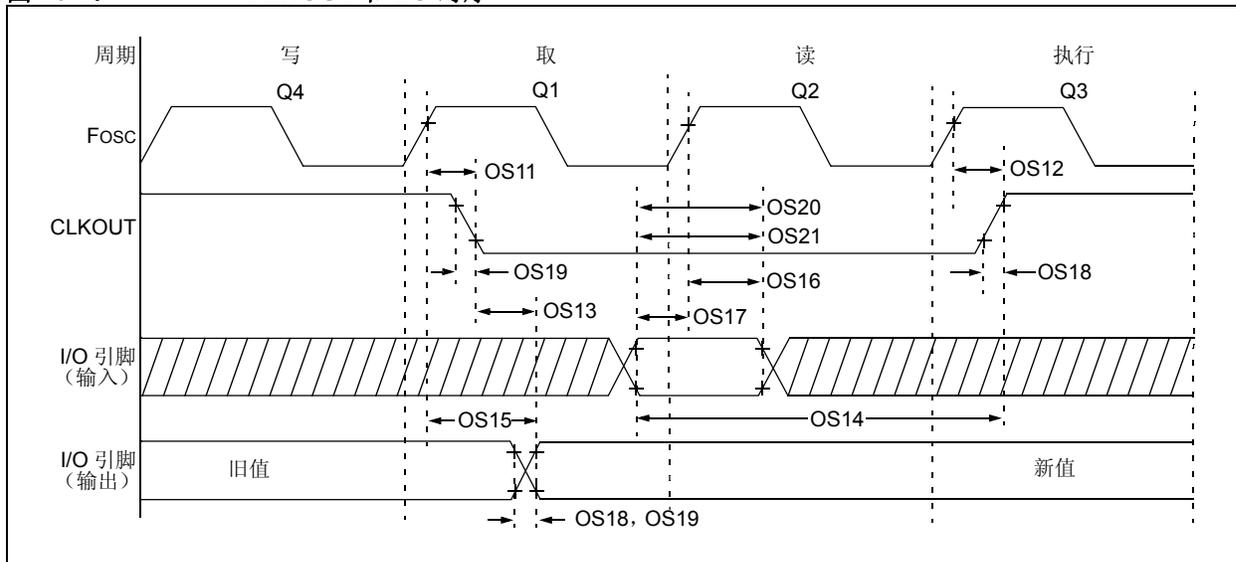


表 28-10: CLKOUT 和 I/O 时序参数

标准工作条件 (除非另外声明)							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS11	TosH2ckL	Fosc↑ 到 CLKOUT↓ 的时间 (1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	Fosc↑ 到 CLKOUT↑ 的时间 (1)	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓ 到端口输出有效的时间 (1)	—	—	20	ns	
OS14	TioV2ckH	CLKOUT↑ 之前端口输入有效的时间 (1)	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 周期) 到端口输出有效的时间	—	50	70*	ns	3.3V ≤ VDD ≤ 5.0V
OS16	TosH2ioI	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	

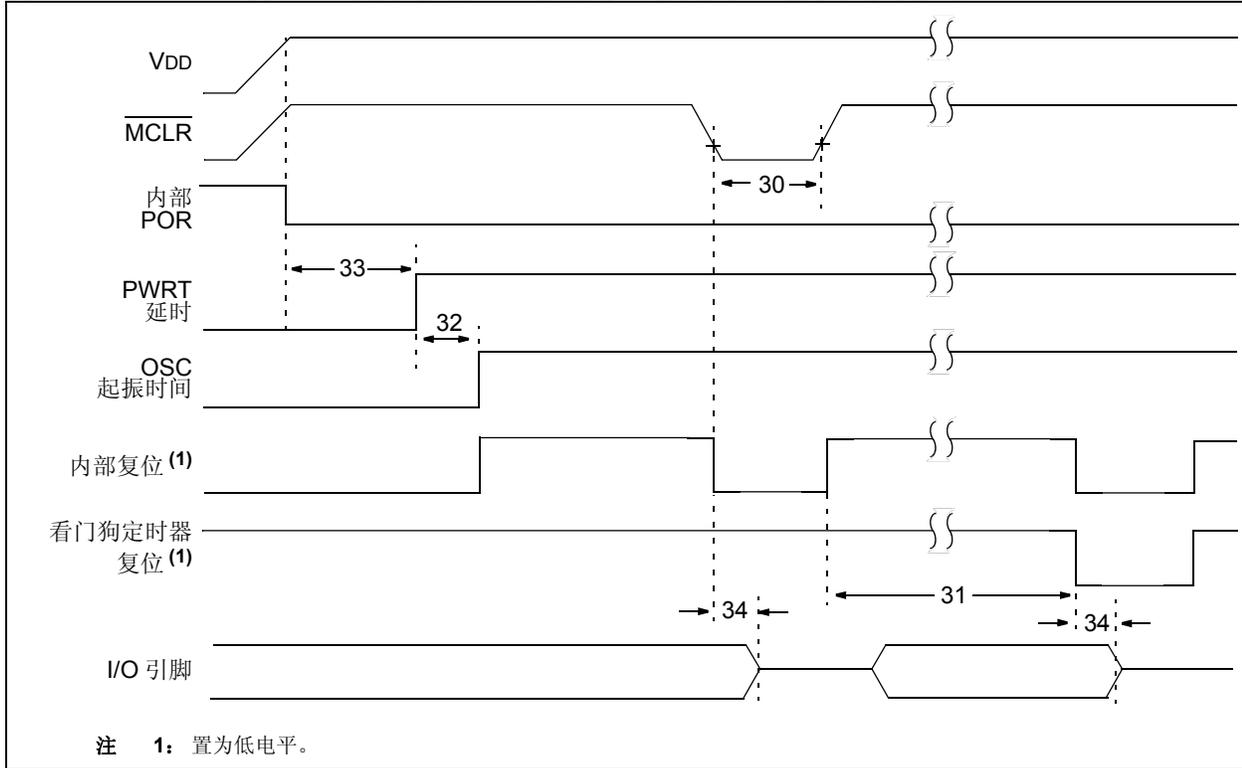
* 这些参数为特性值, 但未经测试。

† 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。

PIC12(L)F1612/16(L)F1613

图 28-8: 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



PIC12(L)F1612/16(L)F1613

表 28-11: 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	V _{DD} = 3.3V-5V, 使用 1:16 预分频比
32	TOST	振荡器起振定时器周期 (1)	—	1024	—	TOSC	
33*	TPWRT	上电延时定时器周期	40	65	140	ms	PWRTE = 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 (2)	2.55	2.70	2.85	V	BORV = 0
			2.35	2.45	2.58	V	BORV = 1
			1.80	1.90	2.05	V	(PIC12F1612/16F1613) BORV = 1 (PIC12LF1612/16LF1613)
36*	VHYST	欠压复位滞后电压	0	25	60	mV	-40°C ≤ T _A ≤ +85°C
37*	TBORDC	欠压复位直流响应时间	1	16	35	μs	V _{DD} ≤ V _{BOR}
38	VLPBOR	低功耗欠压复位电压	1.8	2.1	2.5	V	LPBOR = 1

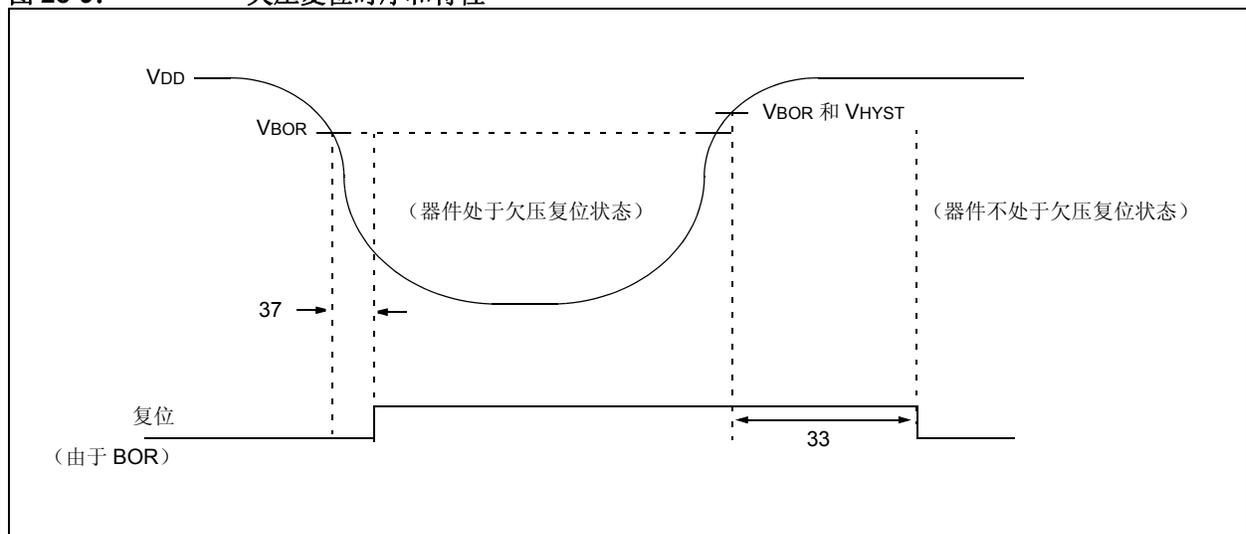
* 这些参数为特性值，但未经测试。

† 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 V_{DD} 和 V_{SS} 之间接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

图 28-9: 欠压复位时序和特性



PIC12(L)F1612/16(L)F1613

图 28-10: TIMER0 和 TIMER1 外部时钟时序

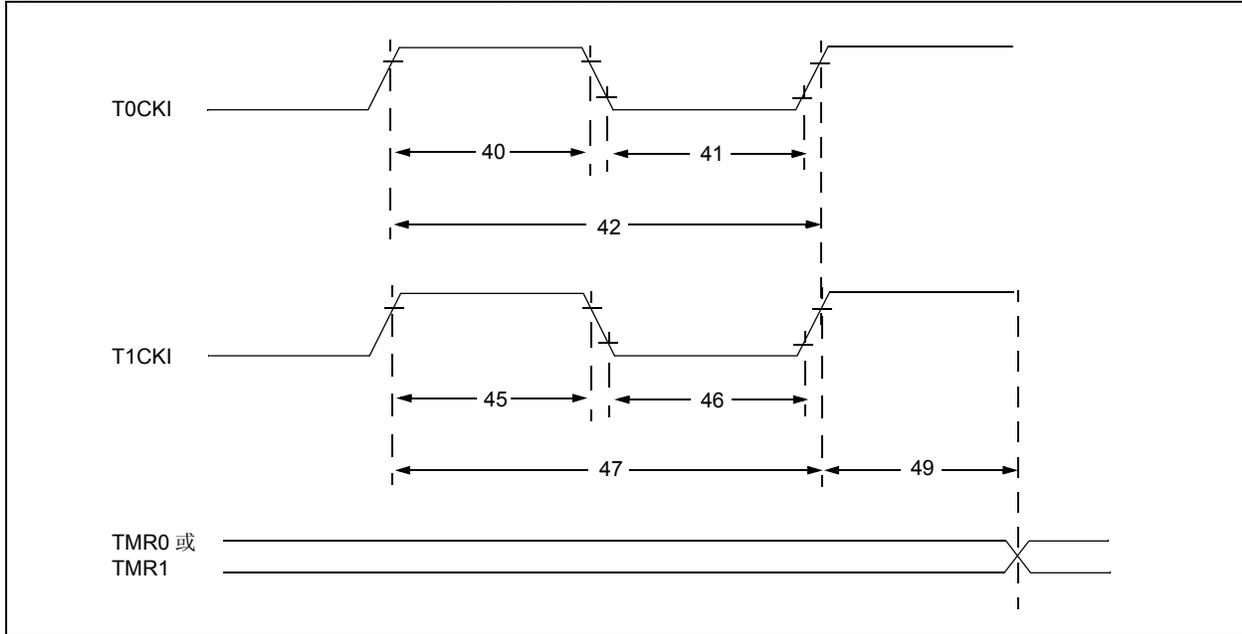


表 28-12: TIMER0 和 TIMER1 外部时钟要求

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 $T_{CY} + 40$ 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, 带预分频器	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, 带预分频器	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 $T_{CY} + 40$ 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。

PIC12(L)F1612/16(L)F1613

表 28-13: 模数转换器 (ADC) 特性 (1,2,3)

工作条件 (除非另外声明) V _{DD} = 3.0V, T _A = 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	V _{REF} = 3.0V
AD03	EDL	微分误差	—	±1	±1	LSb	无丢失编码 V _{REF} = 3.0V
AD04	EOFF	失调误差	—	±1	±2.5	LSb	V _{REF} = 3.0V
AD05	EGN	增益误差	—	±1	±2.0	LSb	V _{REF} = 3.0V
AD06	V _{REF}	参考电压	1.8	—	V _{DD}	V	V _{REF} = (V _{RPOS} - V _{RNEG}) (注 4)
AD07	V _{AIN}	满量程	V _{SS}	—	V _{REF}	V	
AD08	Z _{AIN}	模拟信号源的推荐阻抗	—	—	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 29.0 节“直流和交流特性图表”。
 - 4: ADC V_{REF} 通过 ADPREF<0> 位进行选择。

PIC12(L)F1612/16(L)F1613

图 28-11: ADC 转换时序 (ADC 时钟基于 Fosc)

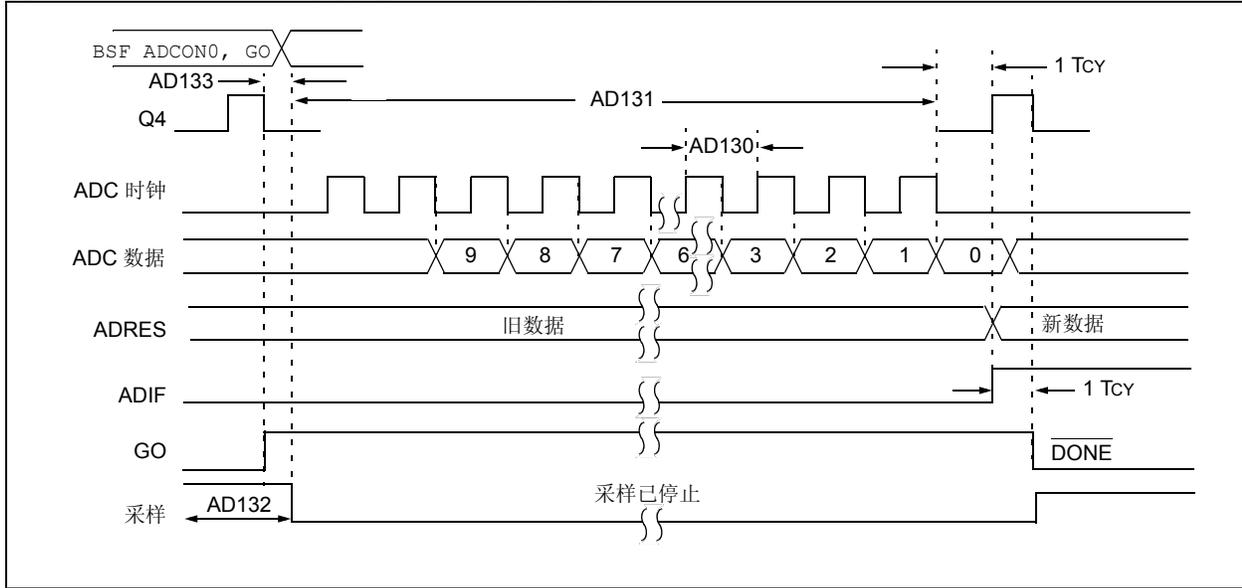
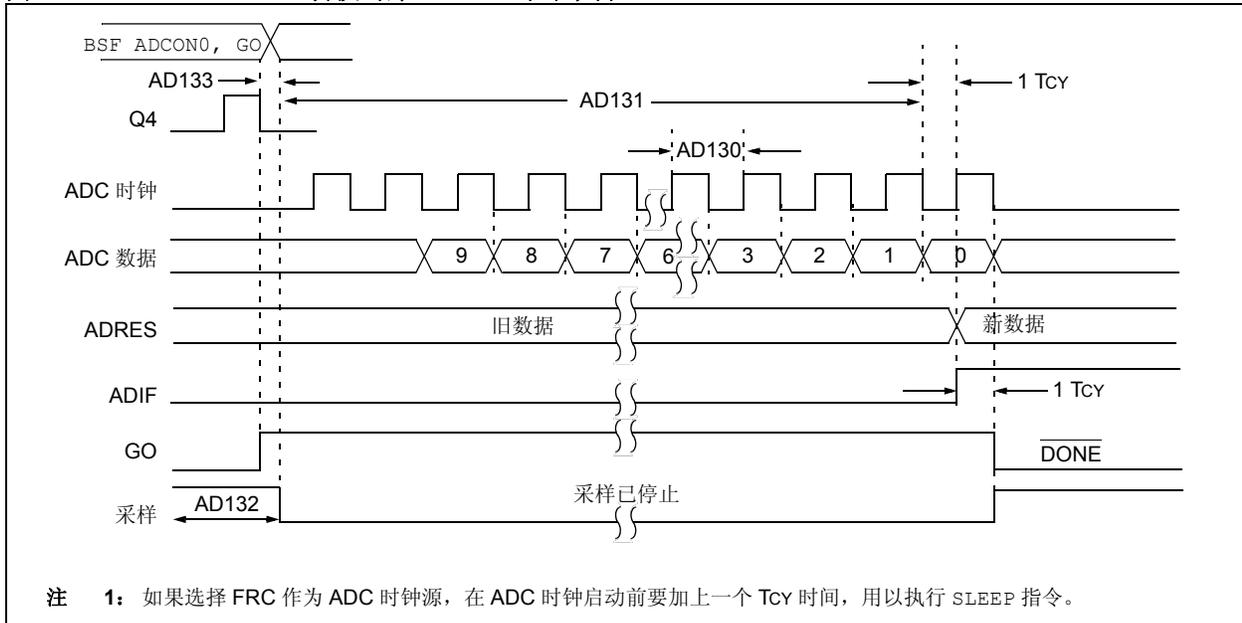


图 28-12: ADC 转换时序 (ADC 时钟来自 FRC)



注 1: 如果选择 FRC 作为 ADC 时钟源, 在 ADC 时钟启动前要加上一个 Tcy 时间, 用以执行 SLEEP 指令。

PIC12(L)F1612/16(L)F1613

表 28-14: 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) (1)	—	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 28-15: 比较器规范 (1)

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 29.0 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。

PIC12(L)F1612/16(L)F1613

表 28-16: 数模转换器 (DAC) 规范⁽¹⁾

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 ⁽²⁾	—	—	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 29.0 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 28-17: 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	—	μA	
ZC03	ZCSNK	灌电流	—	300	—	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。

29.0 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”或“最小值”分别代表（平均值 + 3 σ ）或（平均值 - 3 σ ），其中 σ 是每个温度范围内的标准差。

PIC12(L)F1612/16(L)F1613

图 29-1: 在整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 5.5V$, 仅限 PIC12F1612/16F1613

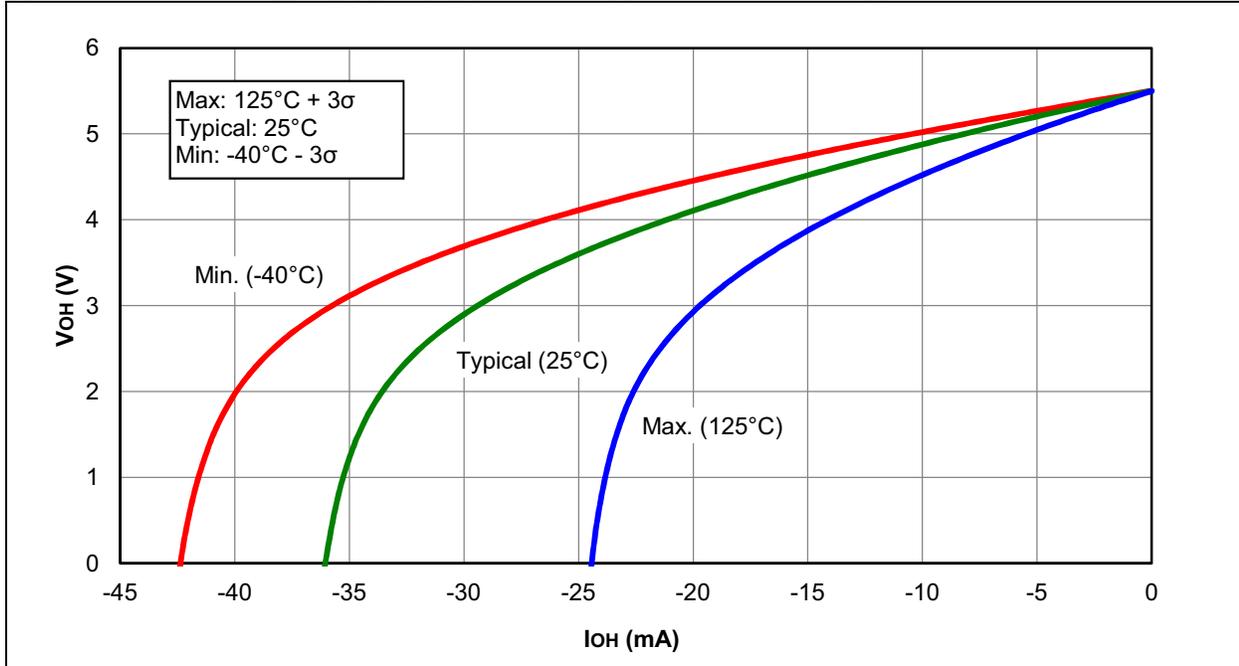
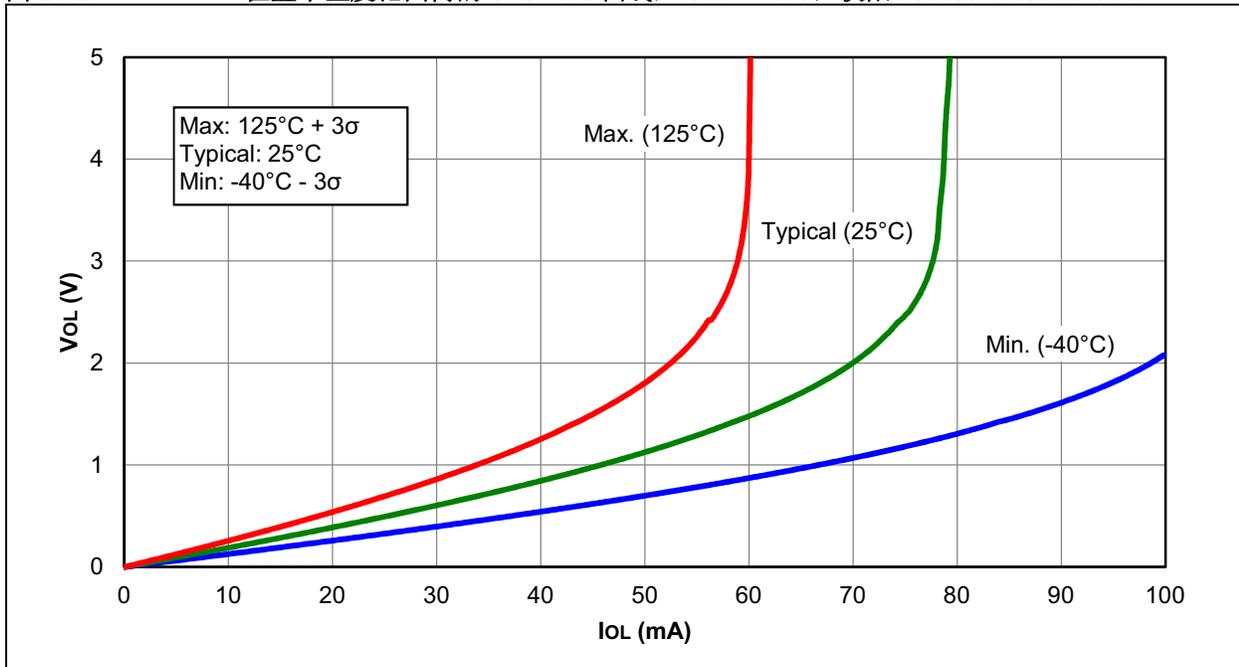


图 29-2: 在整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 5.5V$, 仅限 PIC12F1612/16F1613



PIC12(L)F1612/16(L)F1613

图 29-3: 在整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 3.0V$

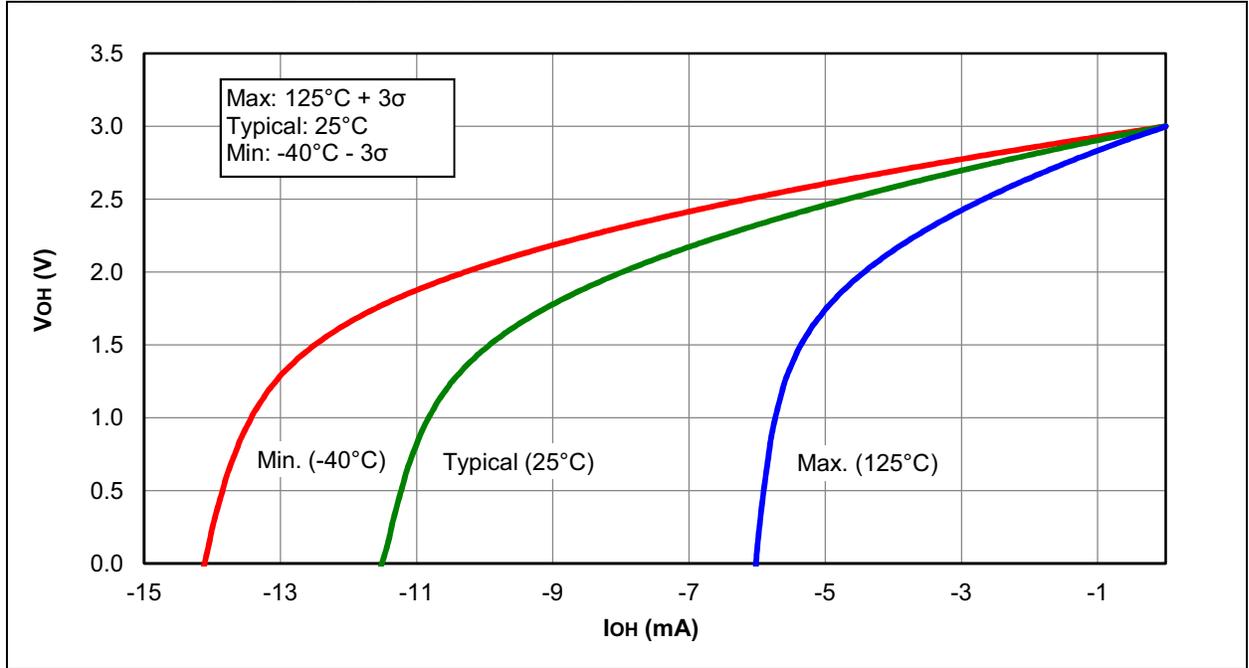
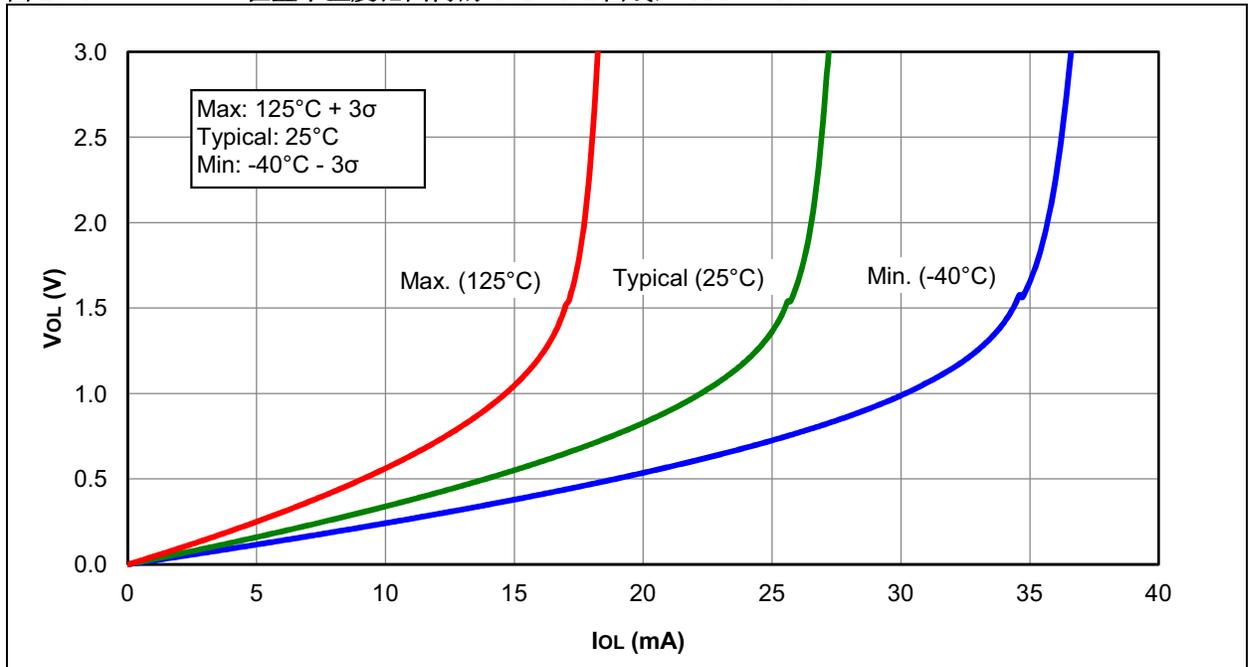


图 29-4: 在整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 3.0V$



PIC12(L)F1612/16(L)F1613

图 29-5: 在整个温度范围内的 V_{OH} — I_{OH} 曲线, $V_{DD} = 1.8V$, 仅限 PIC12LF1612/16LF1613

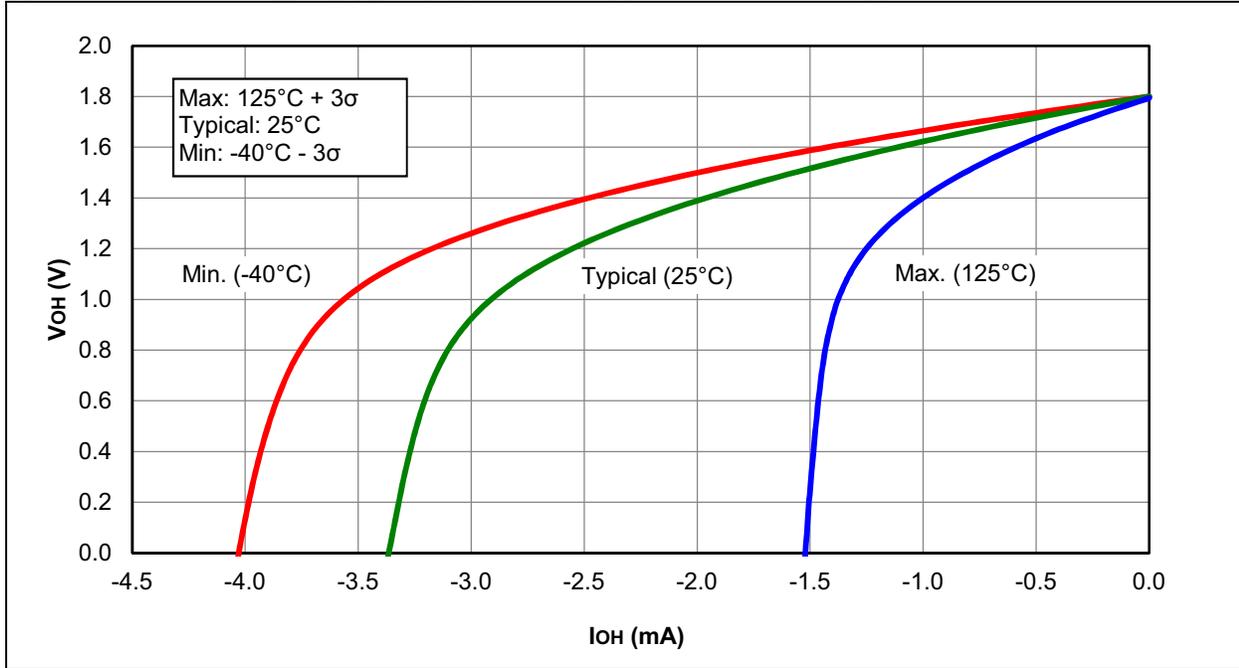
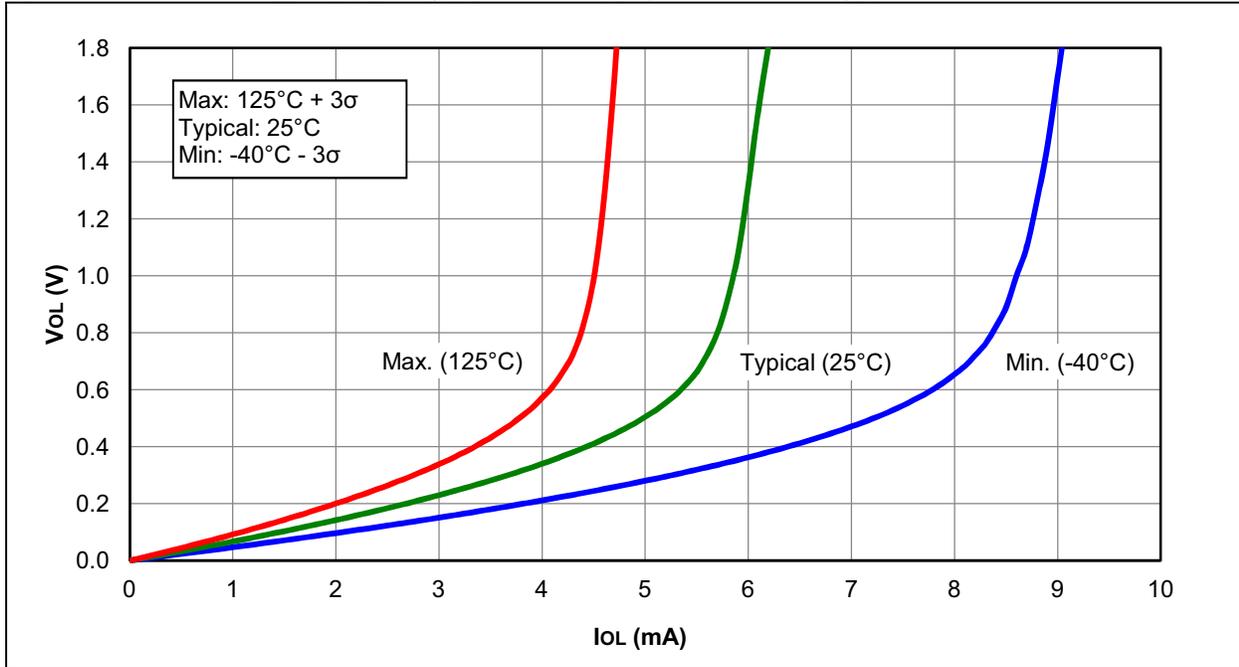


图 29-6: 在整个温度范围内的 V_{OL} — I_{OL} 曲线, $V_{DD} = 1.8V$, 仅限 PIC12LF1612/16LF1613



PIC12(L)F1612/16(L)F1613

图 29-7: POR 释放电压

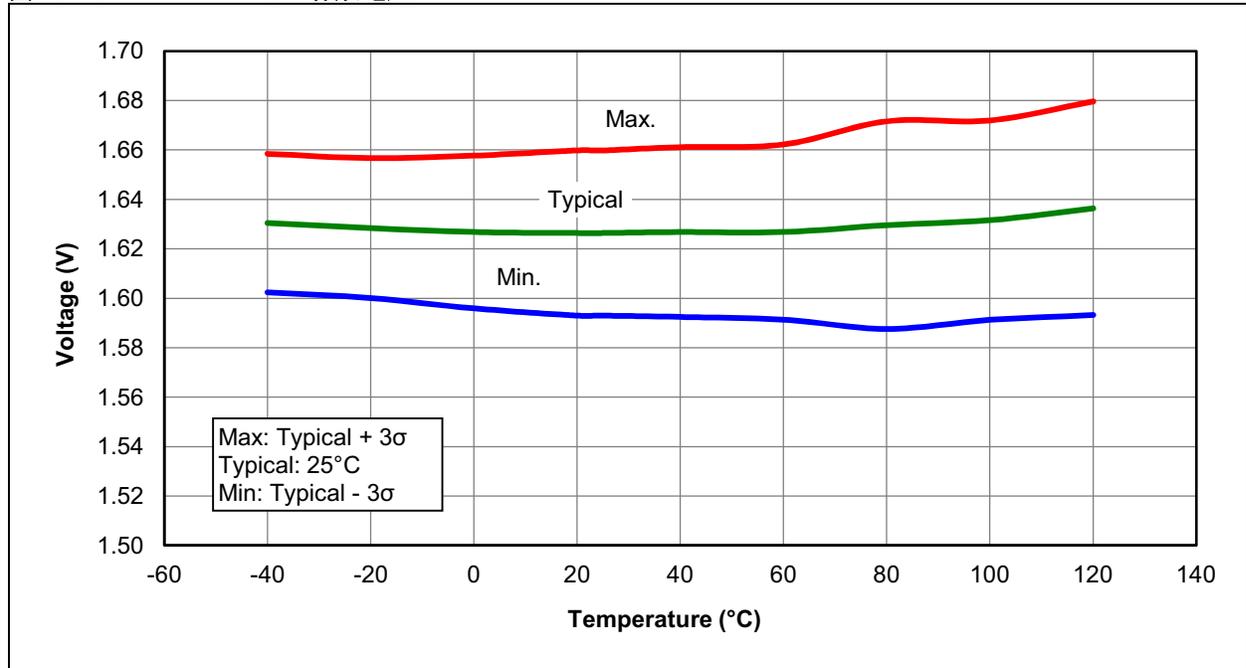
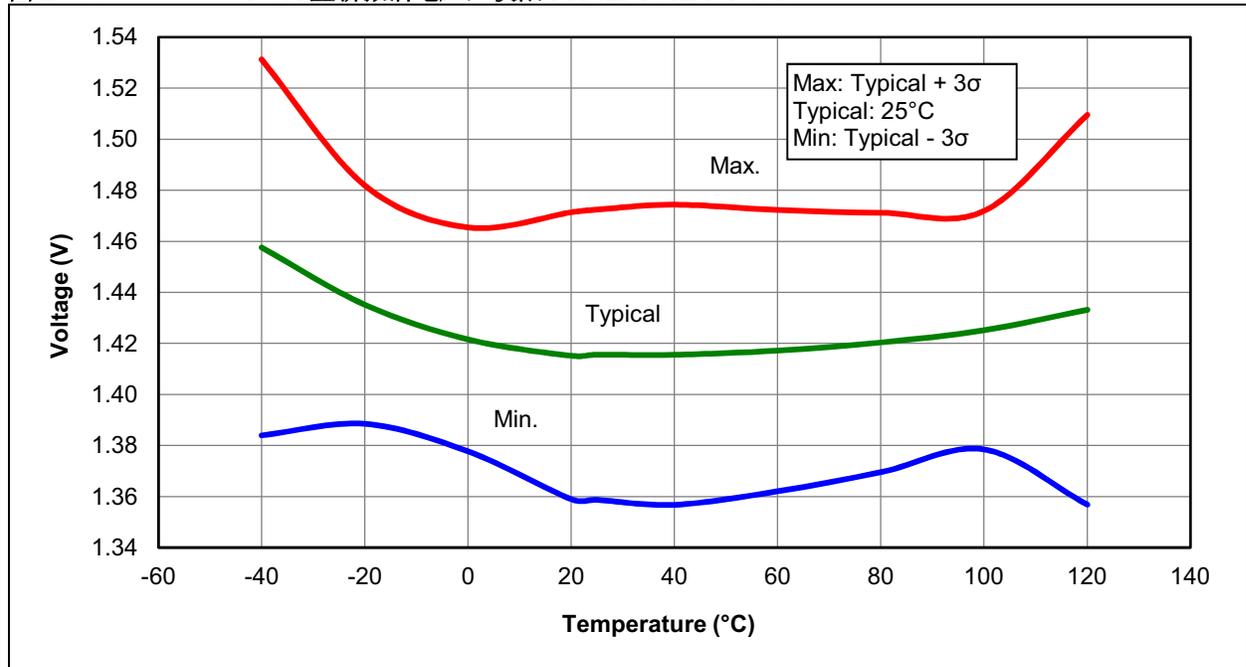


图 29-8: POR 重新激活电压, 仅限 PIC12F1612/16F1613



PIC12(L)F1612/16(L)F1613

图 29-9: 欠压复位电压, BORV = 1, 仅限 PIC12LF1612/16LF1613

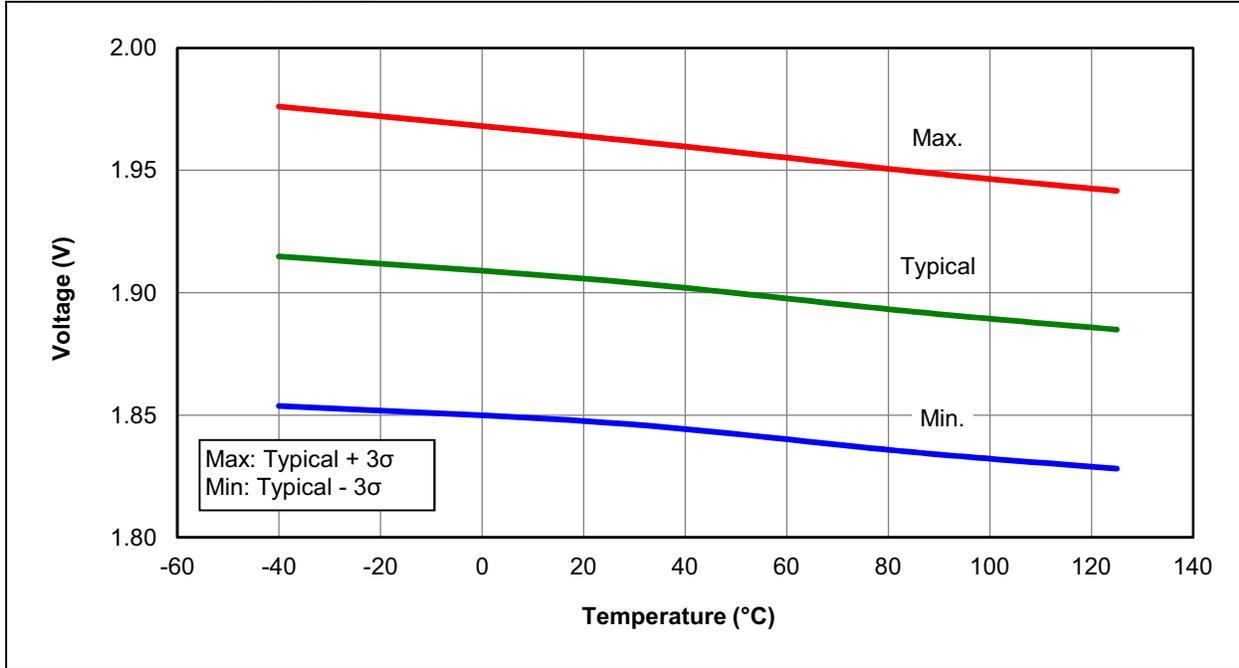
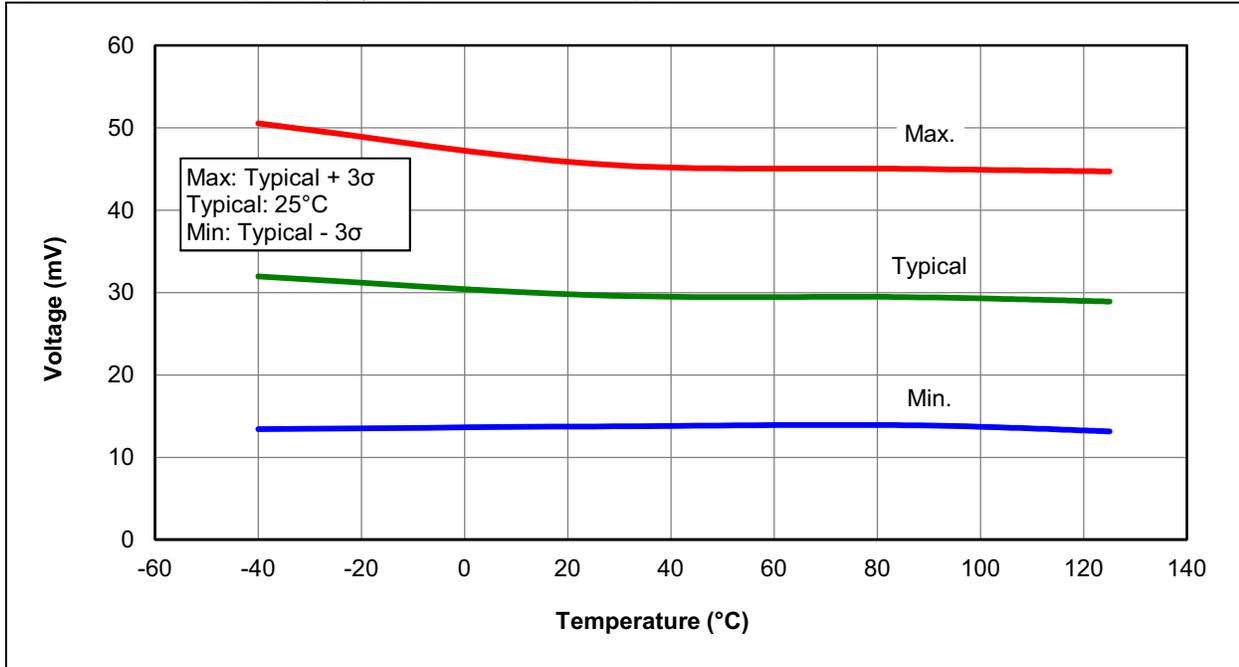


图 29-10: 欠压复位滞后电压, BORV = 1, 仅限 PIC12LF1612/16LF1613



PIC12(L)F1612/16(L)F1613

图 29-11: 欠压复位电压, BORV = 1, 仅限 PIC12F1612/16F1613

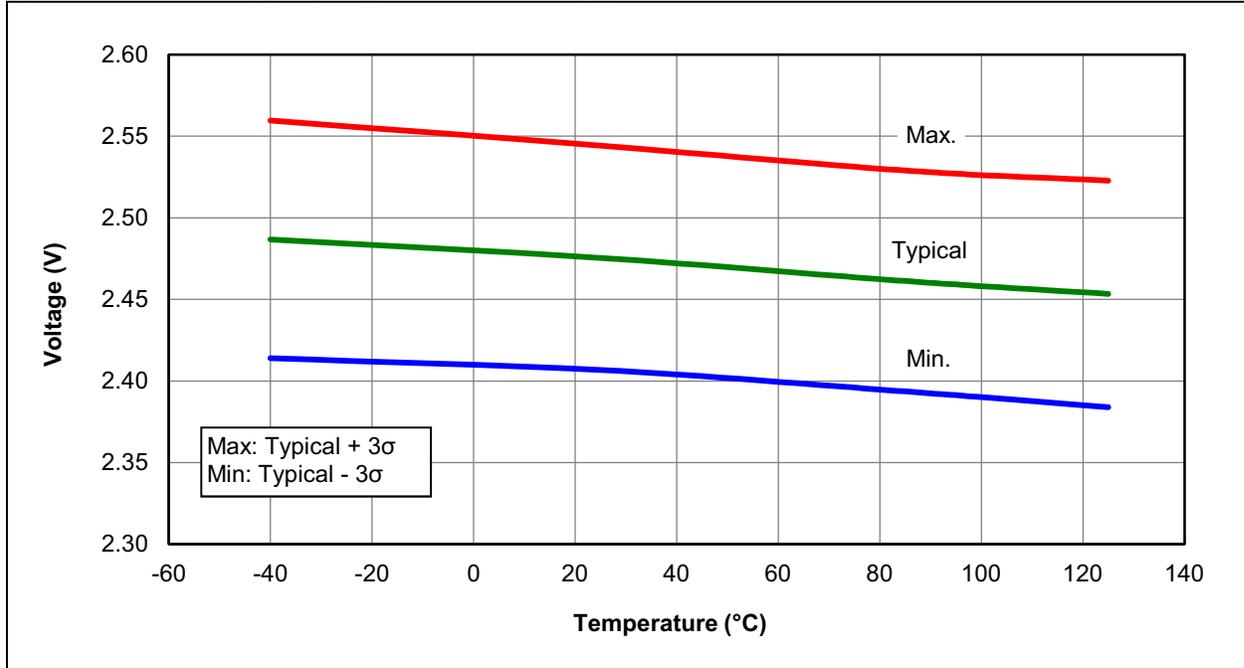
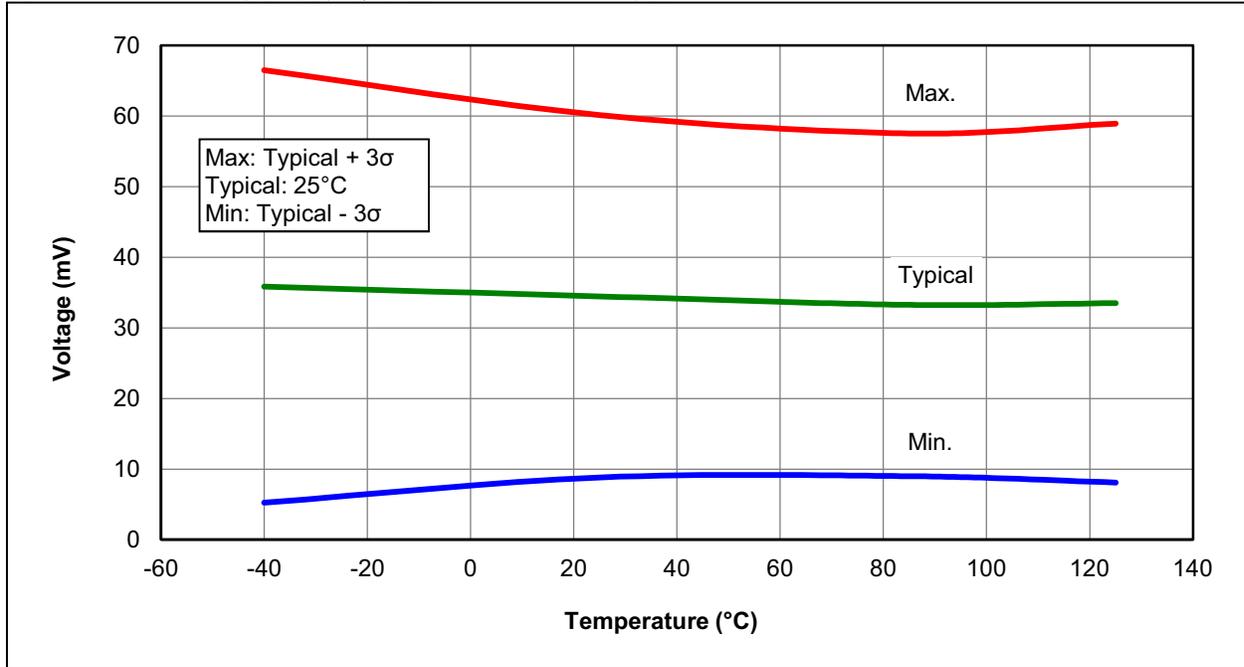


图 29-12: 欠压复位滞后电压, BORV = 1, 仅限 PIC12F1612/16F1613



PIC12(L)F1612/16(L)F1613

图 29-13: 欠压复位电压, BORV = 0

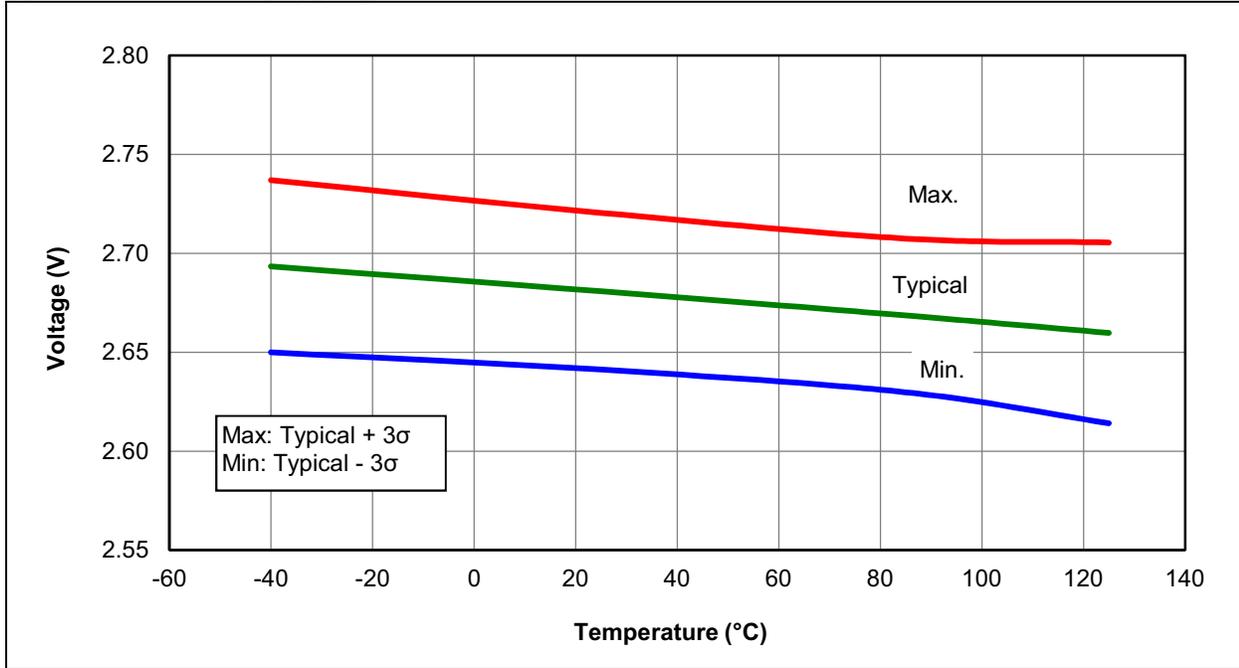
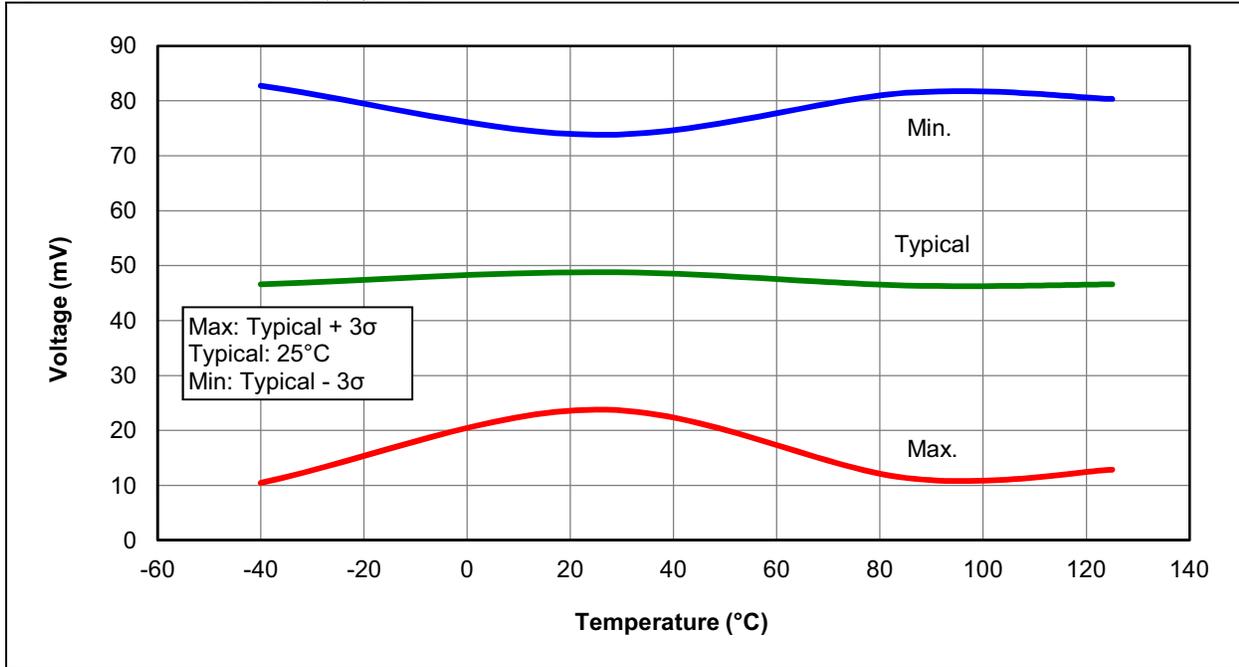


图 29-14: 欠压复位滞后电压, BORV = 0



PIC12(L)F1612/16(L)F1613

图 29-15: 低功耗欠压复位电压, LPBOR = 0

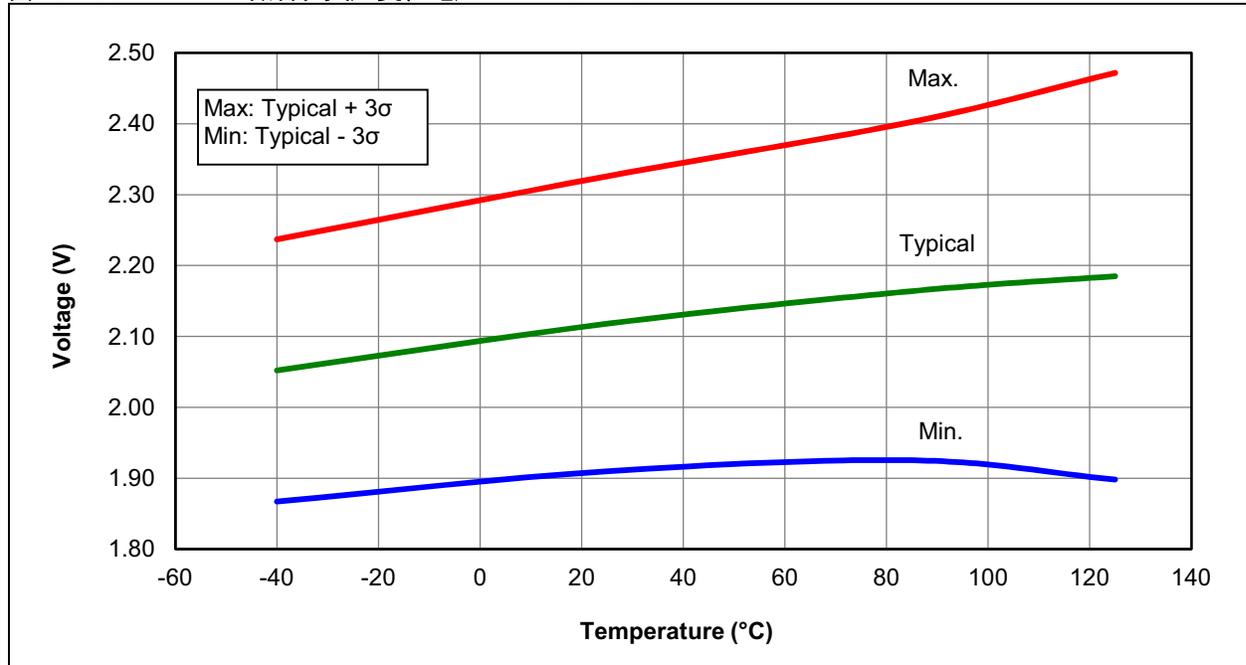
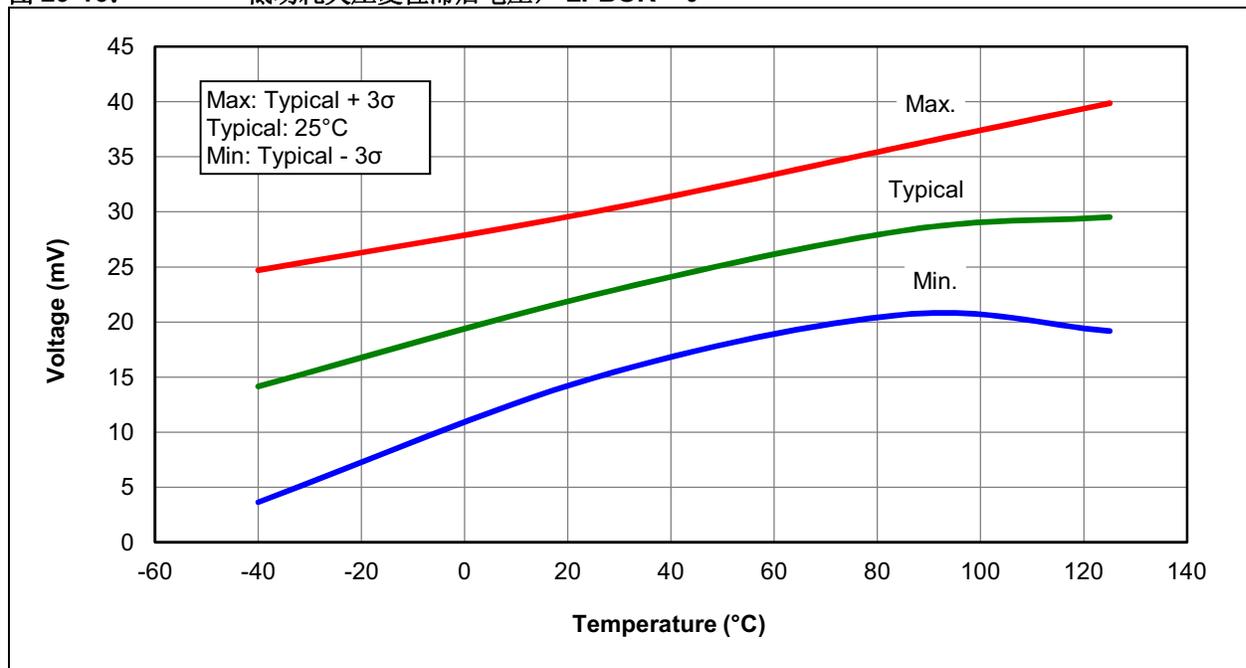


图 29-16: 低功耗欠压复位滞后电压, LPBOR = 0



PIC12(L)F1612/16(L)F1613

图 29-17: WDT 超时周期

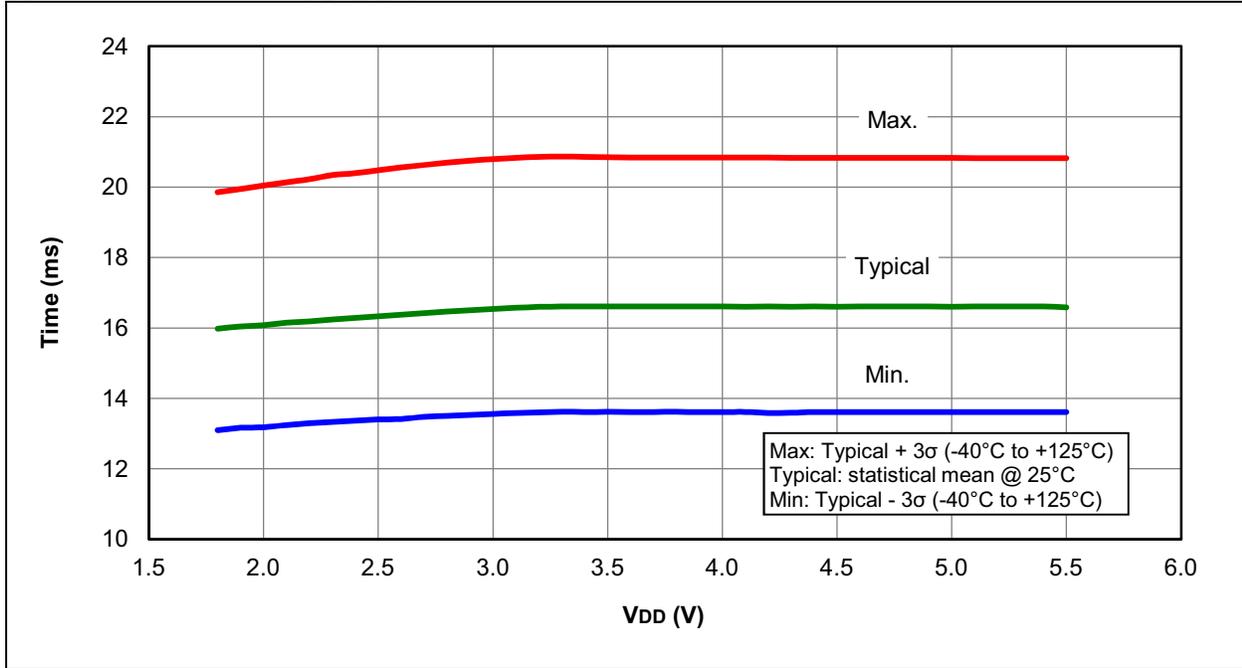


图 29-18: PWRT 周期

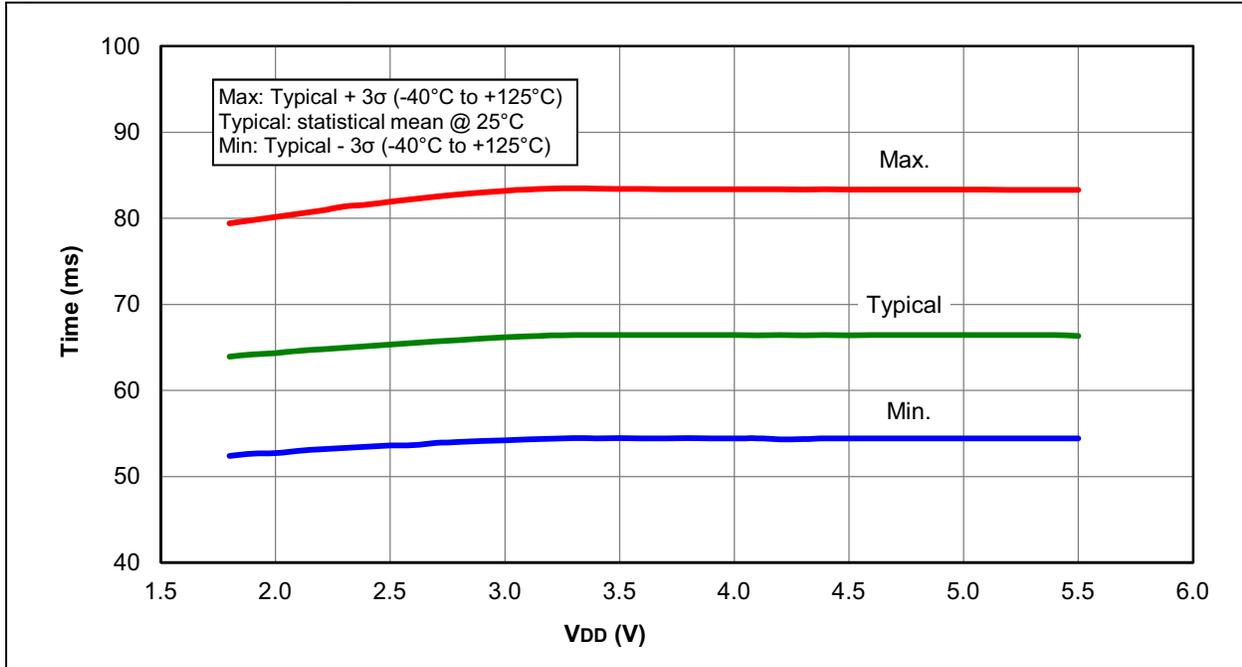
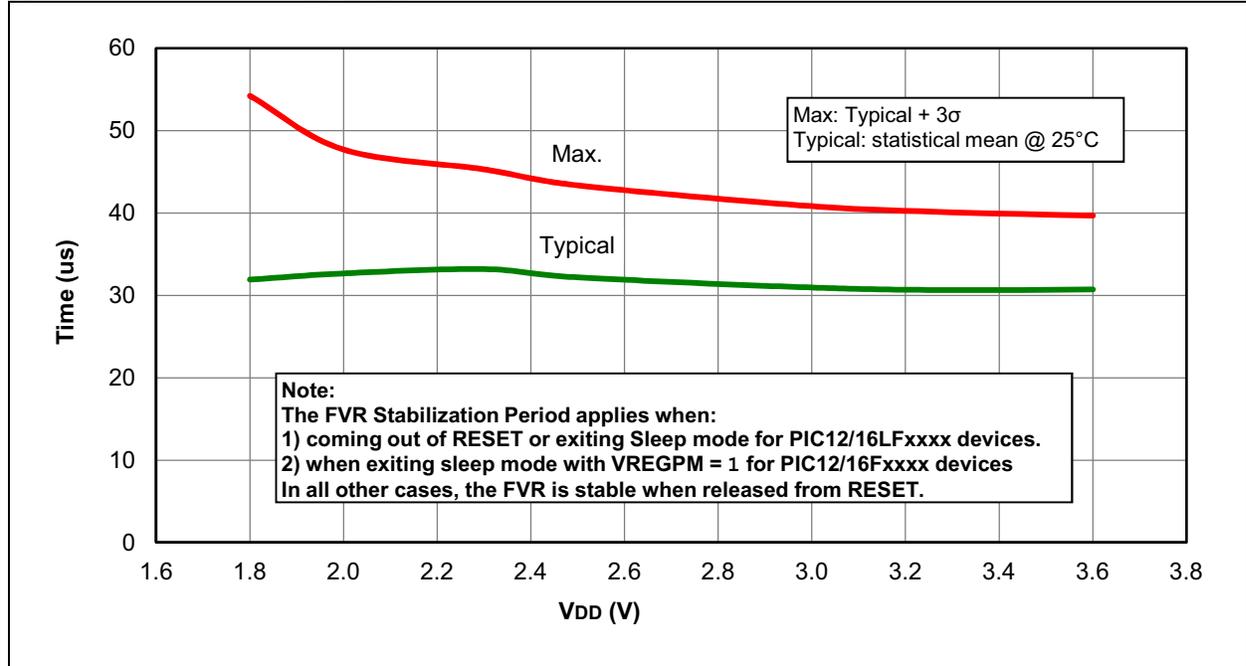


图 29-19: FVR 稳定周期



PIC12(L)F1612/16(L)F1613

图 29-20: 比较器滞后电压, 正常功耗模式 (CxSP = 1, CxHYS = 1)

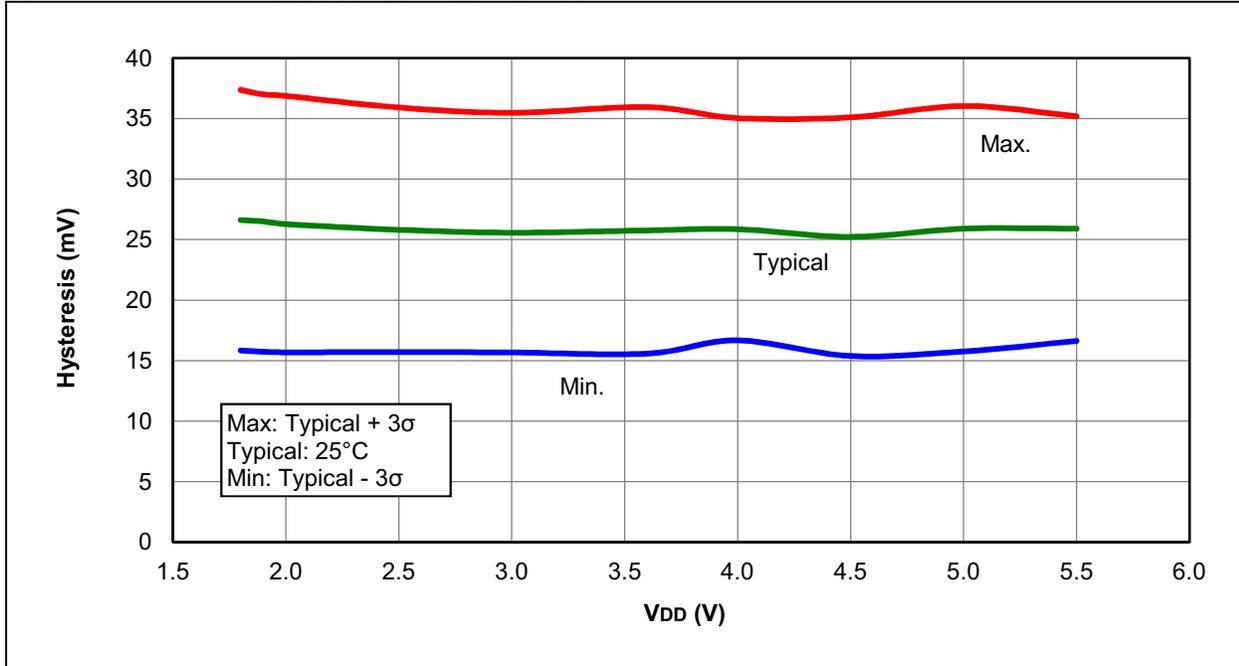
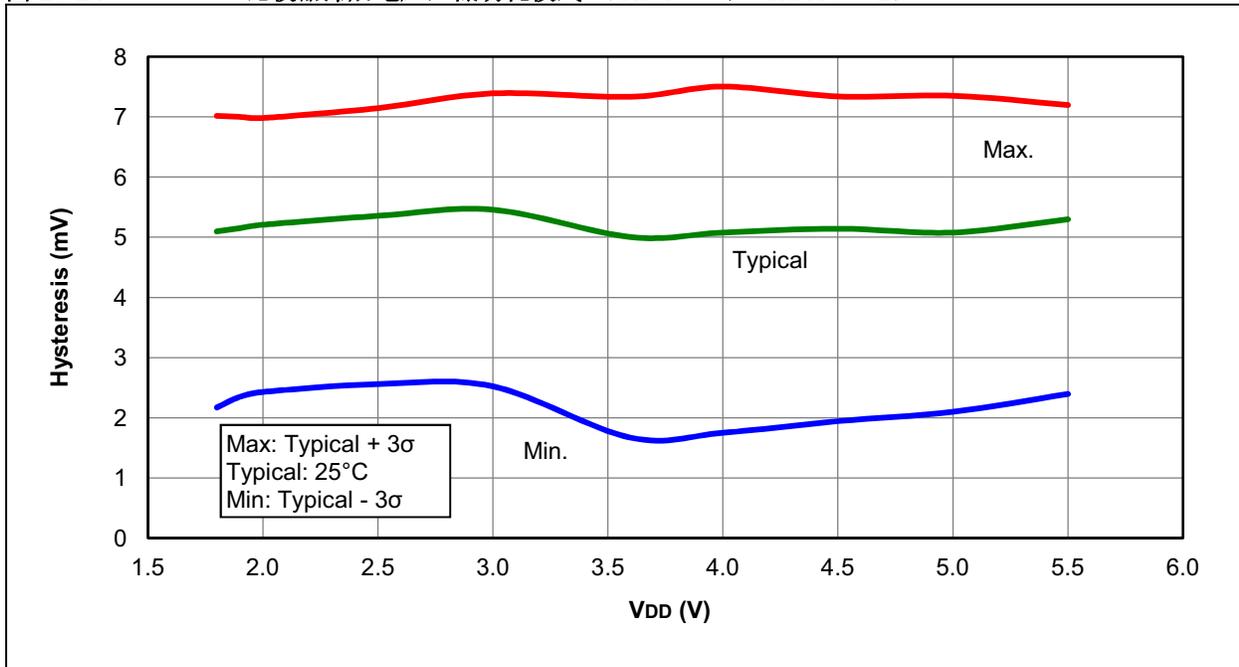


图 29-21: 比较器滞后电压, 低功耗模式 (CxSP = 0, CxHYS = 1)



PIC12(L)F1612/16(L)F1613

图 29-22: 比较器响应时间, 正常功耗模式 (CxSP = 1)

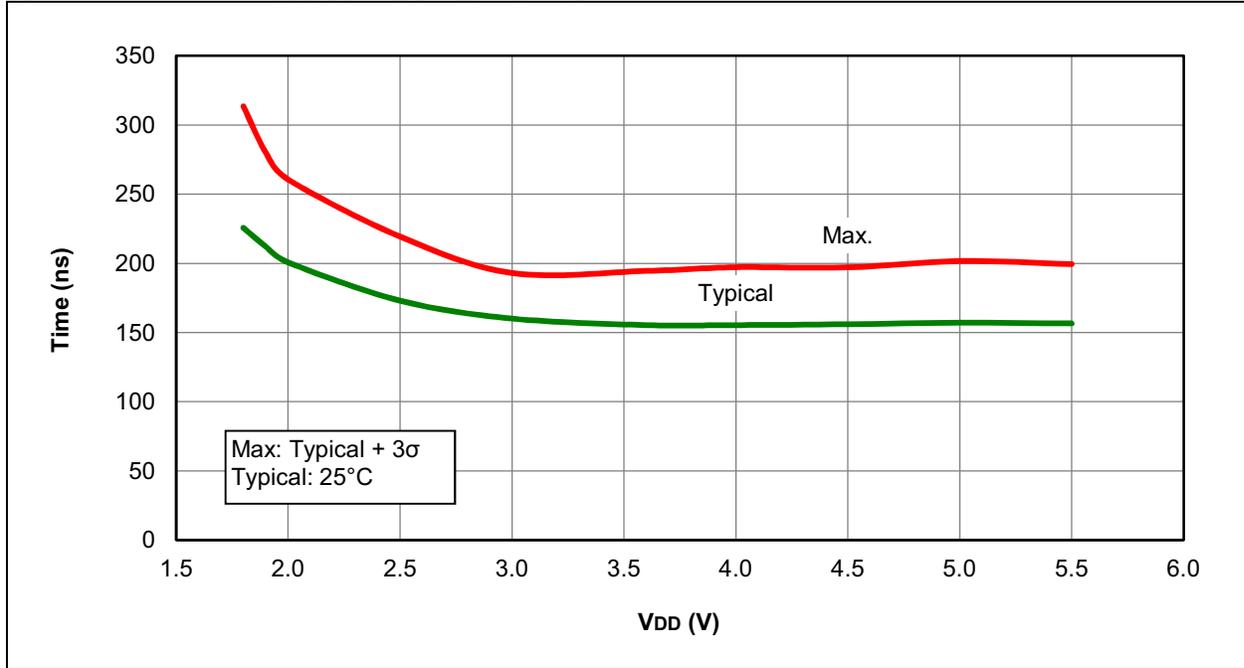
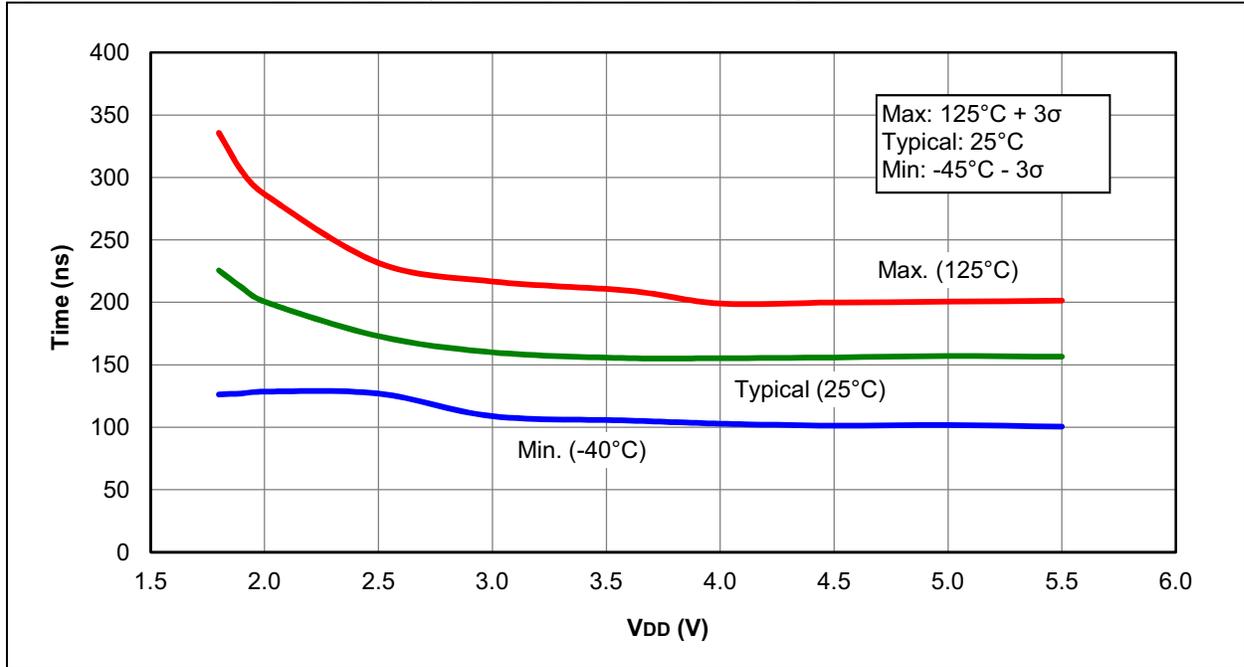
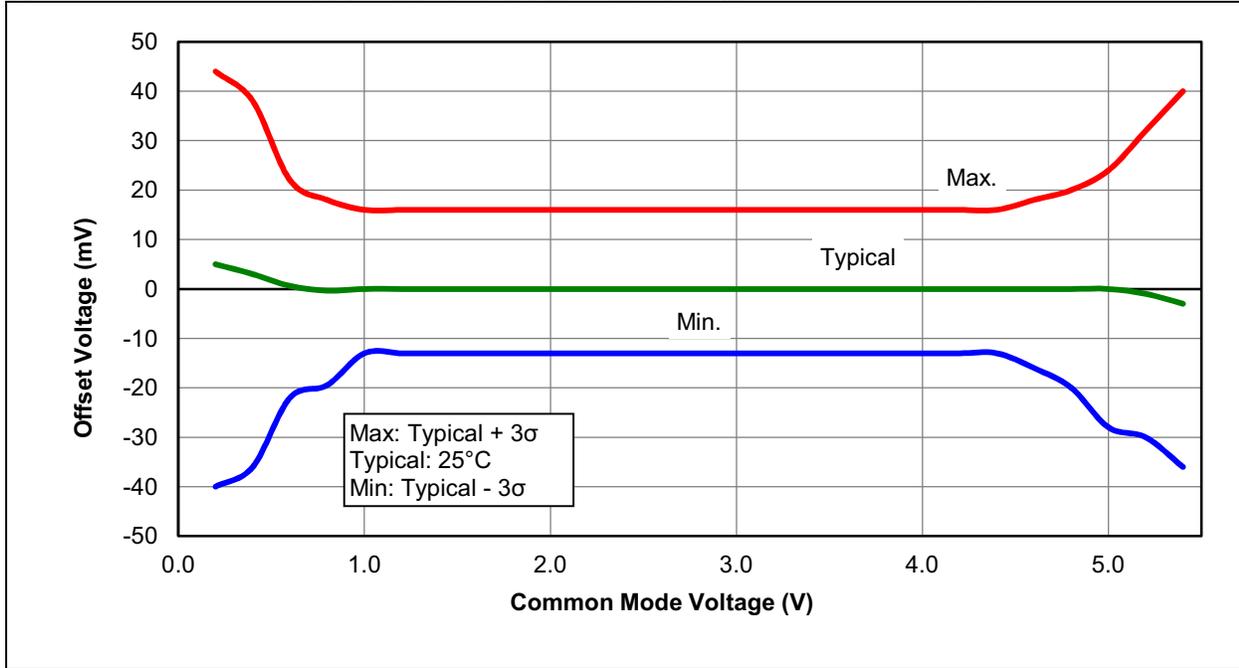


图 29-23: 在整个温度范围内的比较器响应时间, 正常功耗模式 (CxSP = 1)



PIC12(L)F1612/16(L)F1613

图 29-24: 25°C 下的比较器输入失调电压, 正常功耗模式 (CxSP = 1), 仅限 PIC12F1612/16F1613



PIC12(L)F1612/16(L)F1613

图 29-25: 在整个 VDD 和温度范围内的 LFINTOSC 频率, 仅限 PIC12LF1612/16LF1613

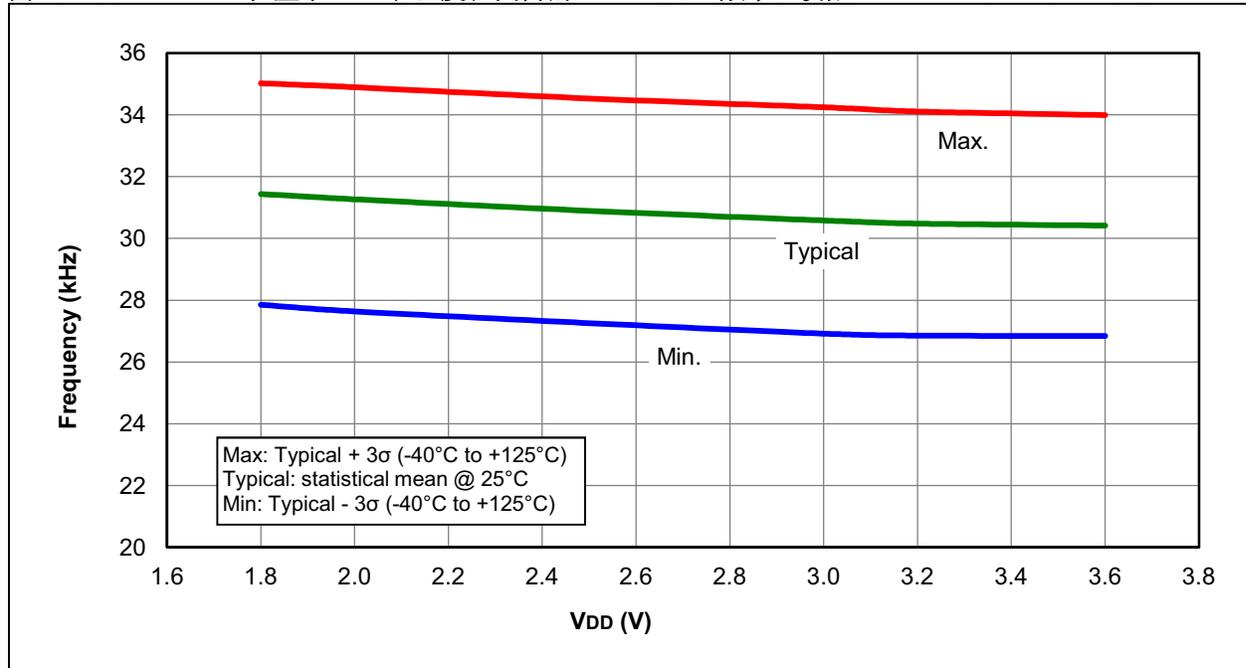
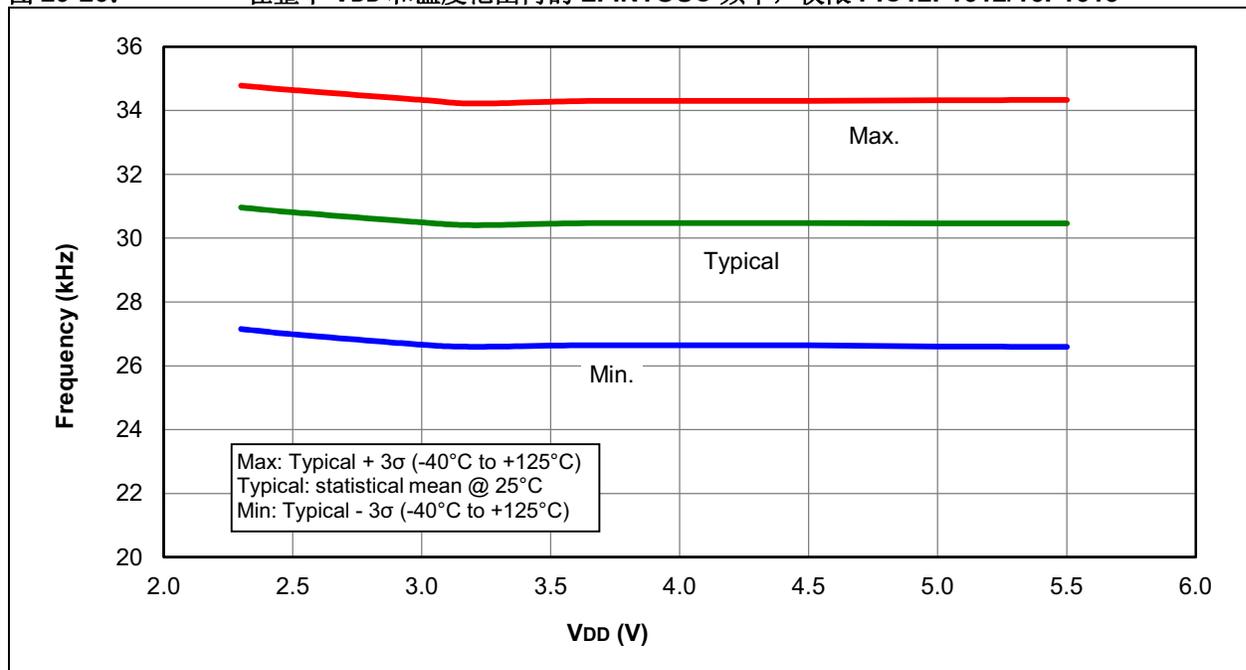


图 29-26: 在整个 VDD 和温度范围内的 LFINTOSC 频率, 仅限 PIC12F1612/16F1613



PIC12(L)F1612/16(L)F1613

图 29-27: 在整个温度范围内的 HFINTOSC 精度, $V_{DD} = 1.8V$, 仅限 PIC12LF1612/16LF1613

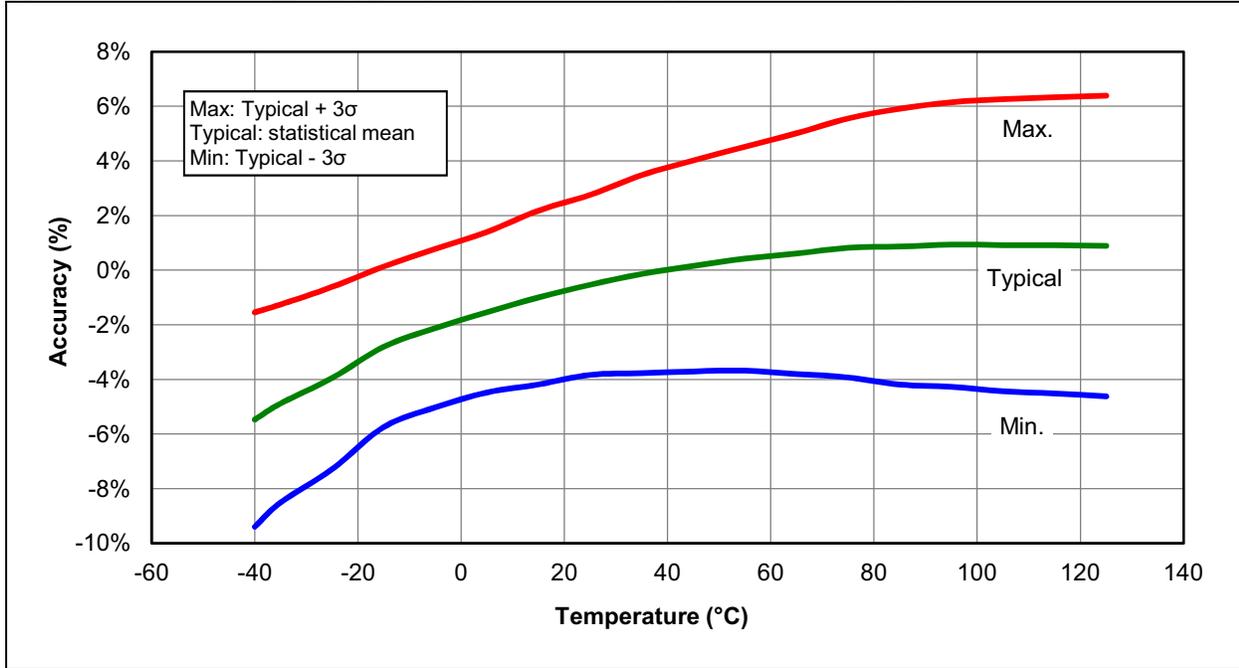
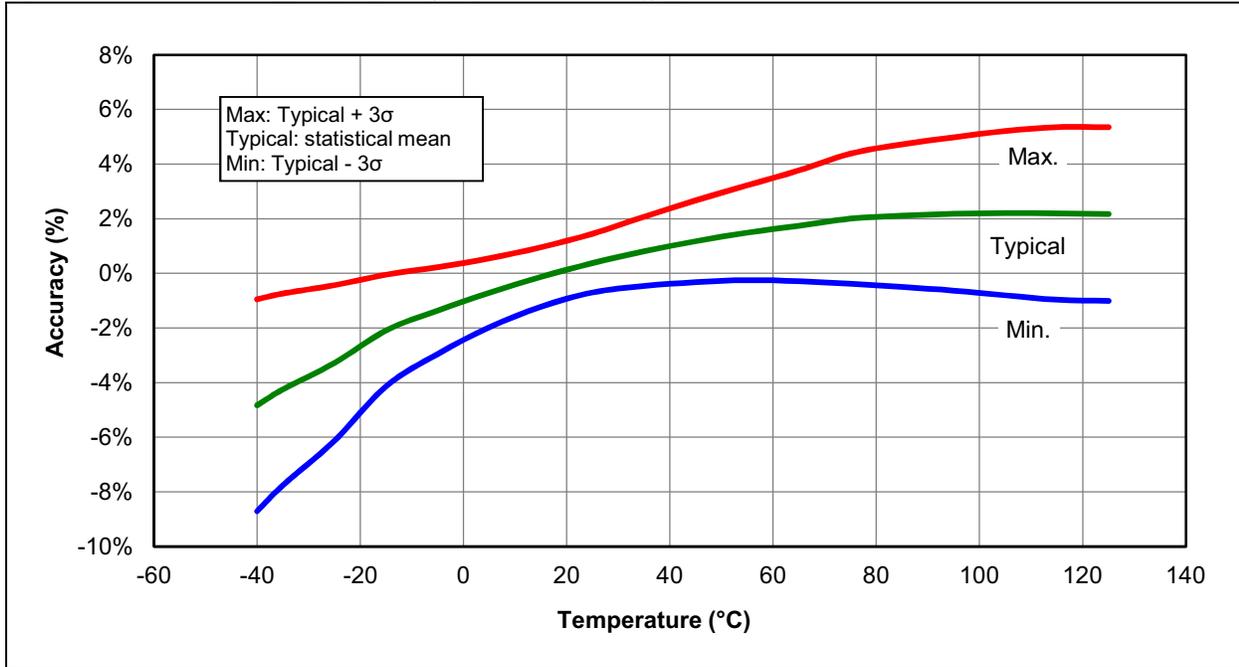
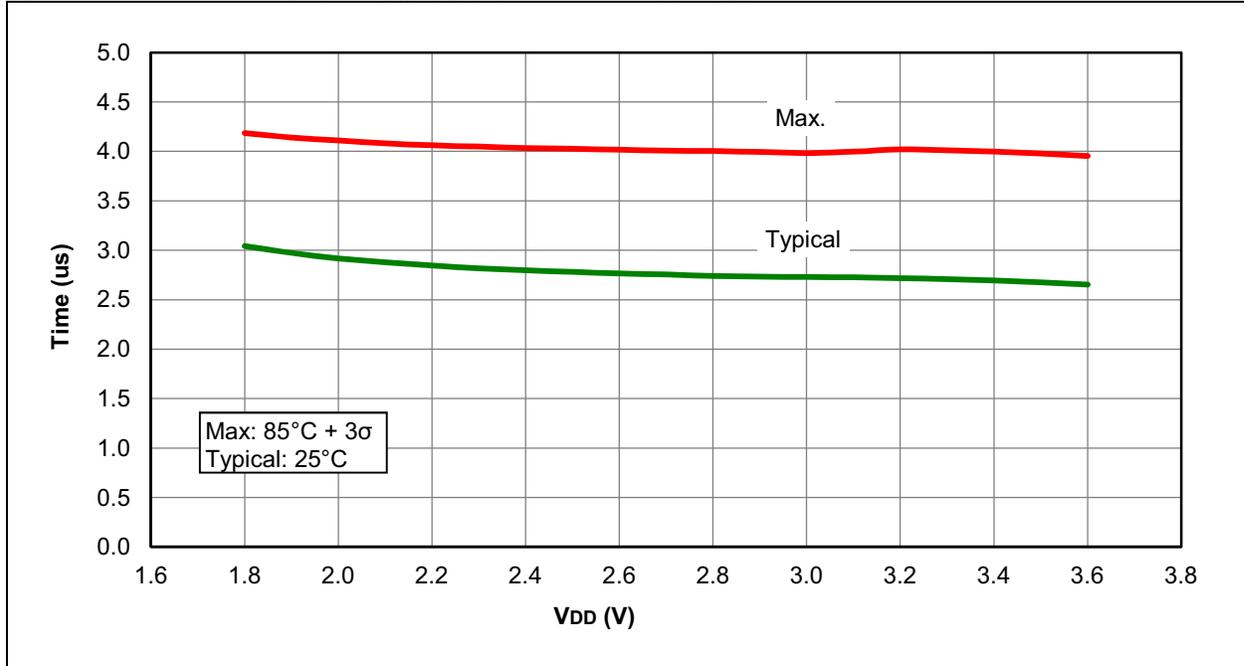


图 29-28: 在整个温度范围内的 HFINTOSC 精度, $2.3V \leq V_{DD} \leq 5.5V$



PIC12(L)F1612/16(L)F1613

图 29-29: 休眠模式, 唤醒周期与 HFINTOSC 源, 仅限 PIC12LF1612/16LF1613



PIC12(L)F1612/16(L)F1613

图 29-30: 低功耗休眠模式, 唤醒周期与 HFINTOSC 源, VREGPM = 1, 仅限 PIC12F1612/16F1613

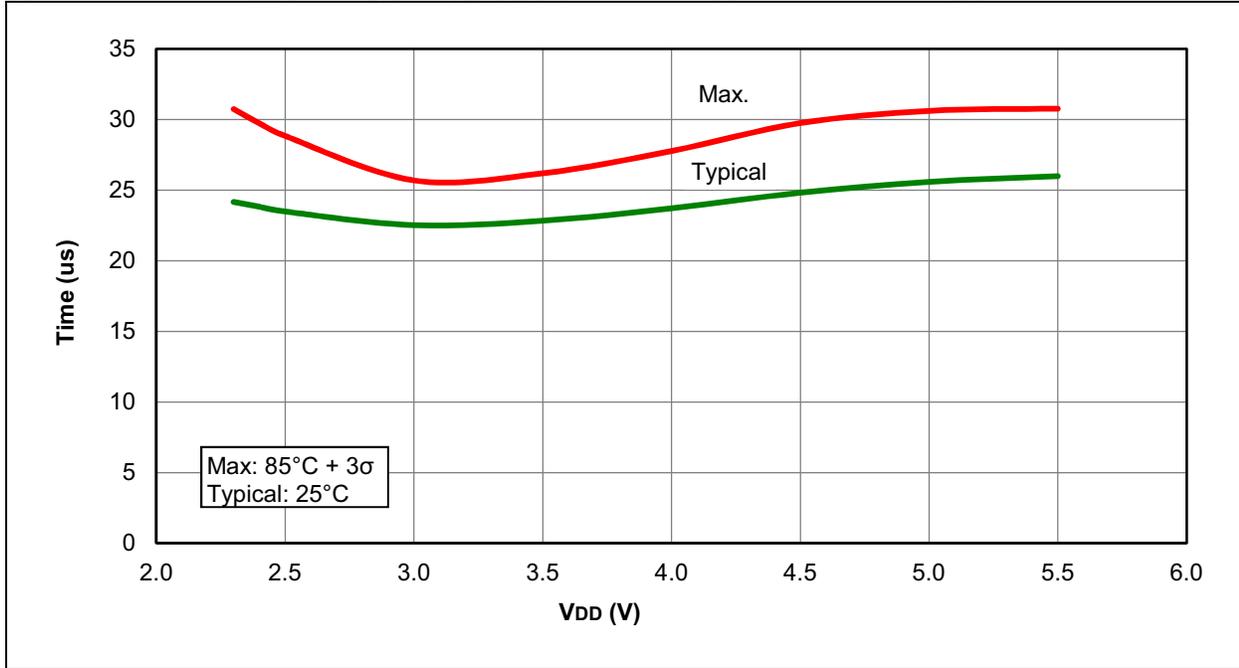
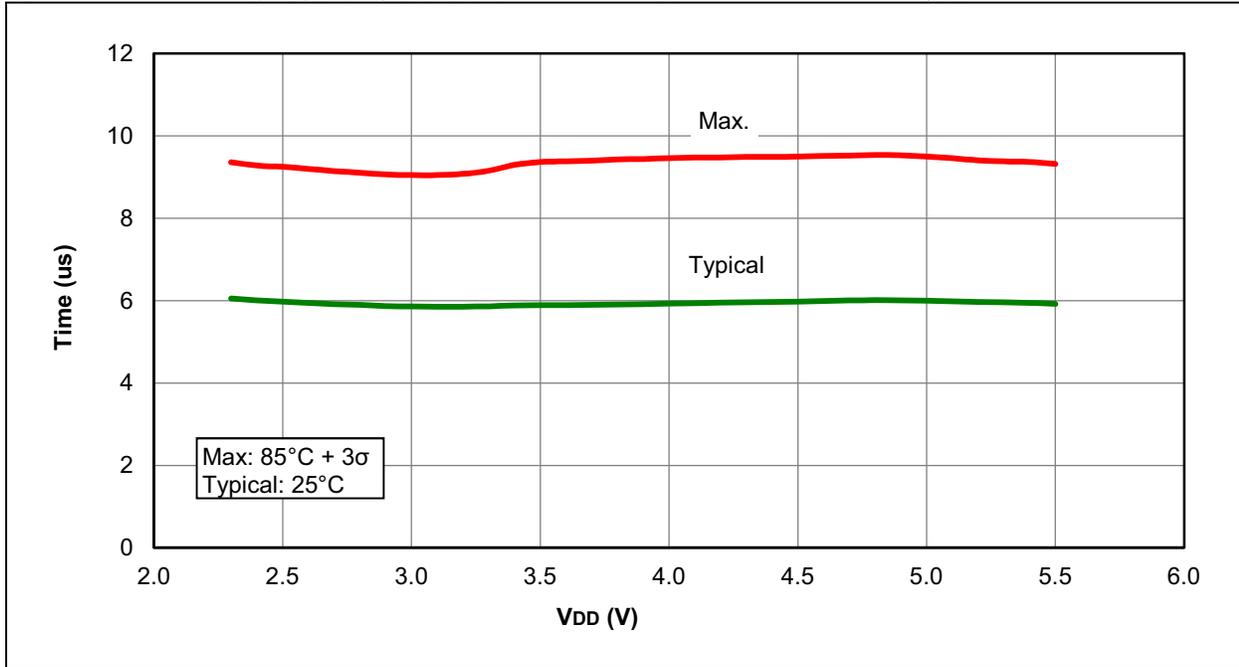


图 29-31: 休眠模式, 唤醒周期与 HFINTOSC 源, VREGPM = 0, 仅限 PIC12F1612/16F1613



30.0 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

30.1 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

PIC12(L)F1612/16(L)F1613

30.2 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

30.3 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

30.4 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

30.5 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB 汇编器为 PIC24 和 PIC32 MCU 以及 dsPIC DSC 器件从符号汇编语言生成可重定位机器码。MPLAB XC 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特性：

- 支持整个器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的指令集
- 与 MPLAB X IDE 兼容

30.6 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。

30.7 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件推出的新一代高速仿真器。结合 MPLAB X IDE 易于使用且功能强大的图形用户界面，该仿真器可对所有 8 位、16 位和 32 位 MCU 及 DSC 器件进行调试和编程。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器和新型抗噪声、高速低压差分信号 (LVDS) 互连电缆 (CAT5) 与目标板相连。

可通过 MPLAB X IDE 下载将来版本的固件，对该仿真器进行现场升级。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：全速仿真、运行时变量观察、跟踪分析、复杂断点、逻辑探针、耐用的探针接口及较长 (长达 3 米) 的互连电缆。

30.8 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 的闪存 DSC 和 MCU 器件。结合 MPLAB X IDE 功能强大但易于使用的图形用户界面，该调试器可对 PIC 闪存单片机和 dsPIC DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器和目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 连接器。

30.9 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 and MPLAB IDE 软件)。

30.10 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。

PIC12(L)F1612/16(L)F1613

30.11 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVAL® 评估系统、 Σ - Δ ADC、流速传感器，等等。

同时还提供入门工具包，其中包含体验指定器件功能所需的所有软硬件。通常提供单个应用以及调试功能，都包含在一块电路板上。

有关演示、开发和评估工具包的完整列表，请访问 Microchip 网站 (www.microchip.com)。

30.12 第三方开发工具

Microchip 还提供一些来自第三方供应商的优秀开发工具。这些工具均经过精心挑选，功能独特，物有所值。

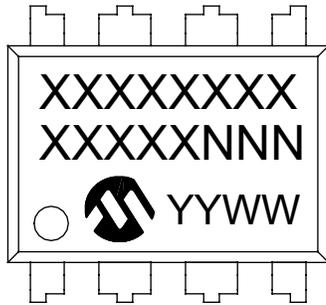
- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 IPLogika® 等公司提供的嵌入式以太网解决方案

PIC12(L)F1612/16(L)F1613

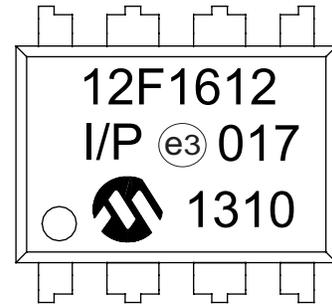
31.0 封装信息

31.1 封装标识信息

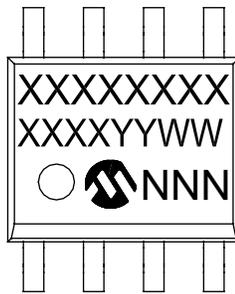
8 引脚 PDIP (300 mil)



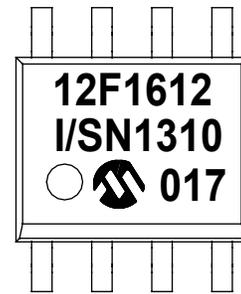
示例



8 引脚 SOIC (3.90 mm)



示例



图注:	XX...X	客户指定信息
	Y	年份代码 (日历年的最后一位数字)
	YY	年份代码 (日历年的最后两位数字)
	WW	星期代码 (一月一日的星期代码为“01”)
	NNN	以字母数字排序的追踪代码
	(e3)	雾锡 (Matte Tin, Sn) 的 JEDEC® 无铅标志
	*	本封装为无铅封装。JEDEC 无铅标志 ((e3)) 标示于此种封装的外包装上。

注: Microchip 部件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户指定信息的字符数。

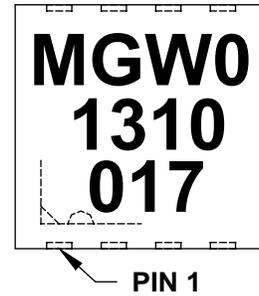
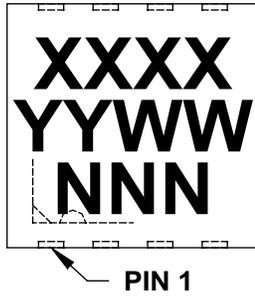
* 标准 PIC® 器件标识由 Microchip 部件编号、年份代码、星期代码和追踪代码组成。若 PIC 器件标识超出上述内容, 需支付一定的附加费用。请向当地的 Microchip 销售办事处咨询确认。对于 QTP 器件, 任何特殊标记的费用都已包含在 QTP 价格中。

PIC12(L)F1612/16(L)F1613

31.1 封装标识信息（仅限 PIC12(L)F1612）（续）

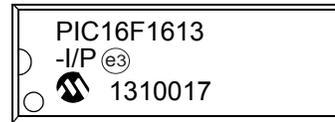
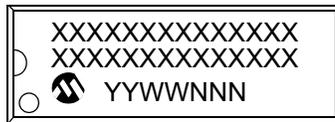
8 引脚 DFN (3x3x0.9 mm)

示例



14 引脚 PDIP

示例



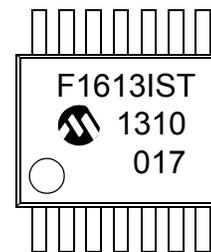
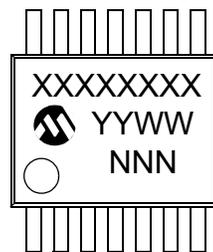
14 引脚 SOIC (0.150 英寸)

示例



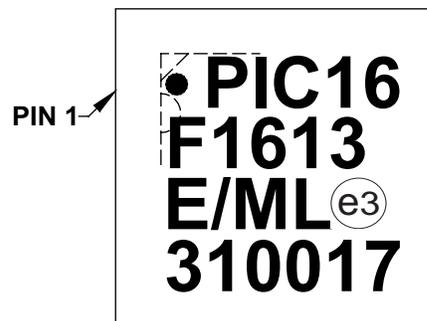
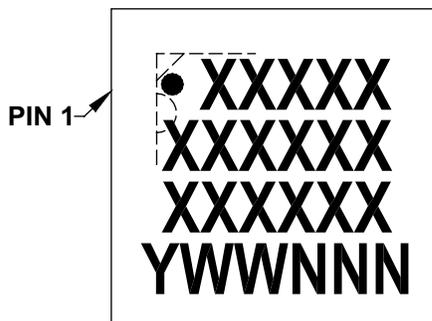
14 引脚 TSSOP

示例



16 引脚 QFN (4x4x0.9 mm)

示例



PIC12(L)F1612/16(L)F1613

表 31-1: 8 引脚 3x3 DFN (MF) 顶部标识

部件编号	标识
PIC12F1612-E/MF	MGU0
PIC12LF1612-E/MF	MGW0
PIC12F1612-I/MF	MGV0
PIC12LF1612-I/MF	MGX0
PIC12F1612T-I/MF	MGV0
PIC12LF1612T-I/MF	MGX0

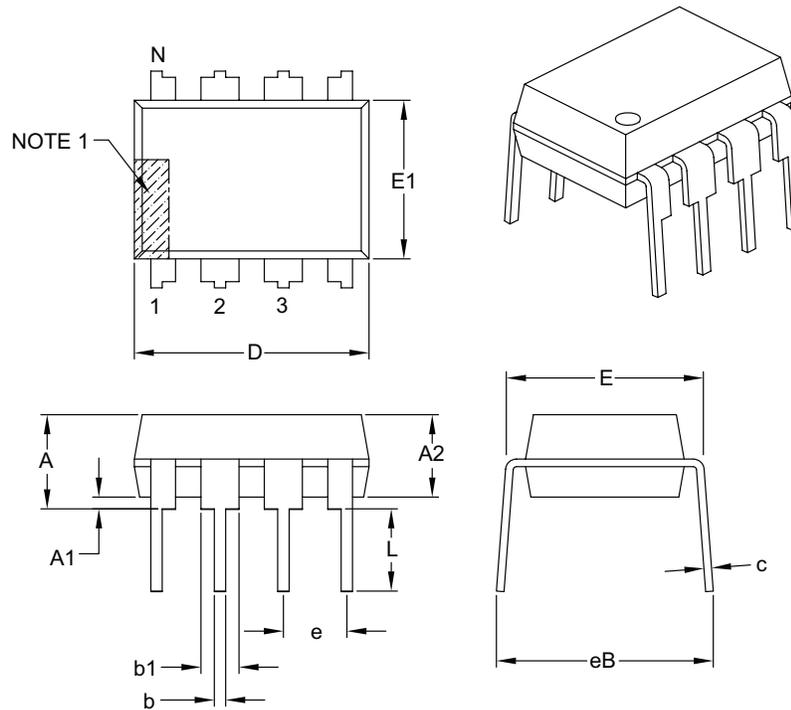
PIC12(L)F1612/16(L)F1613

31.2 封装详细信息

以下部分将介绍各种封装的技术细节。

8 引脚塑封双列直插式封装 (P) —— 主体 300 mil [PDIP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	8		
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	.348	.365	.400
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

- Pin 1 visual index feature may vary, but must be located with the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

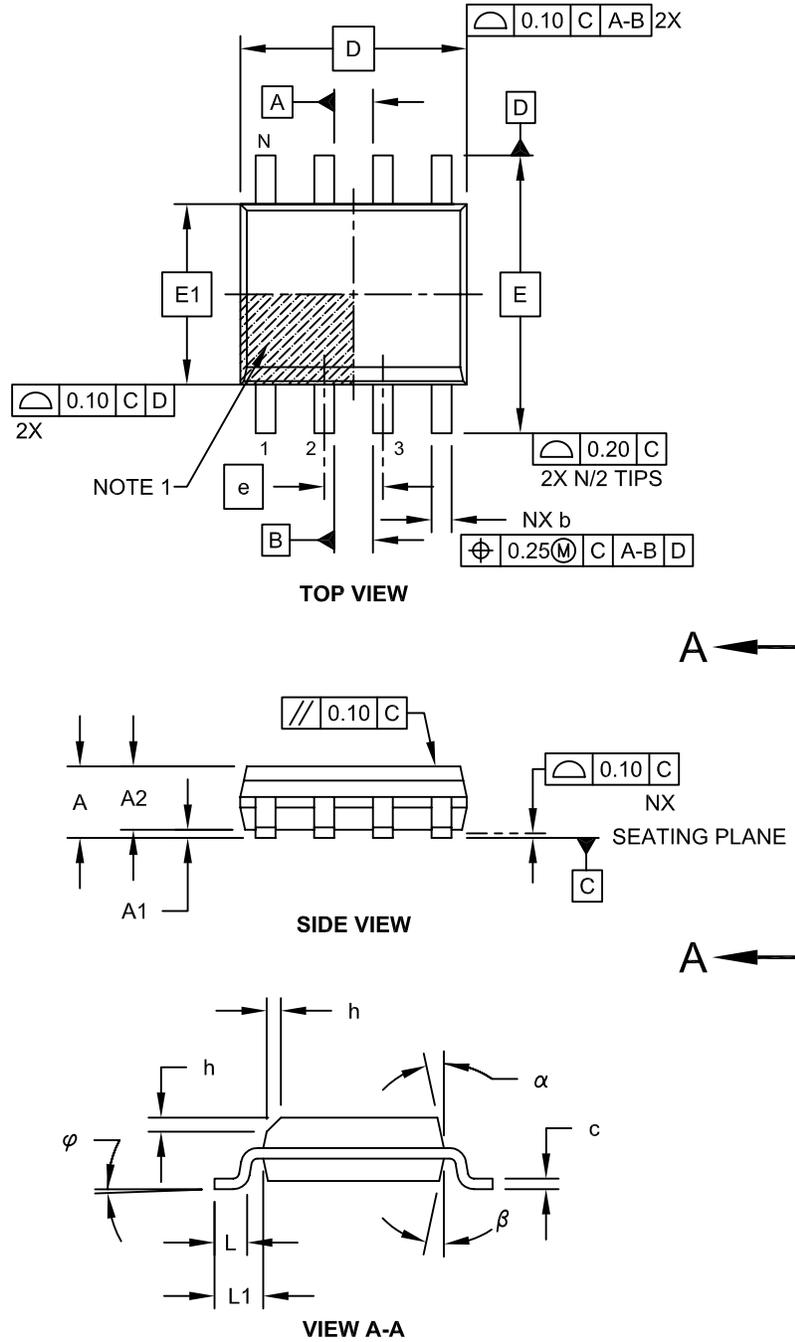
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-018B

PIC12(L)F1612/16(L)F1613

8 引脚塑封窄条小外形封装 (SN) —— 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。

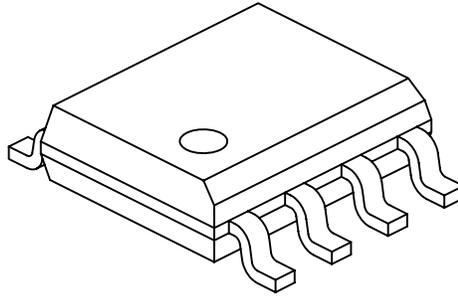


Microchip Technology Drawing No. C04-057C Sheet 1 of 2

PIC12(L)F1612/16(L)F1613

8 引脚塑封窄条小外形封装 (SN) —— 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		8		
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		4.90 BSC		
Chamfer (Optional)	h		0.25	-	0.50
Foot Length	L		0.40	-	1.27
Footprint	L1		1.04 REF		
Foot Angle	φ		0°	-	8°
Lead Thickness	c		0.17	-	0.25
Lead Width	b		0.31	-	0.51
Mold Draft Angle Top	α		5°	-	15°
Mold Draft Angle Bottom	β		5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

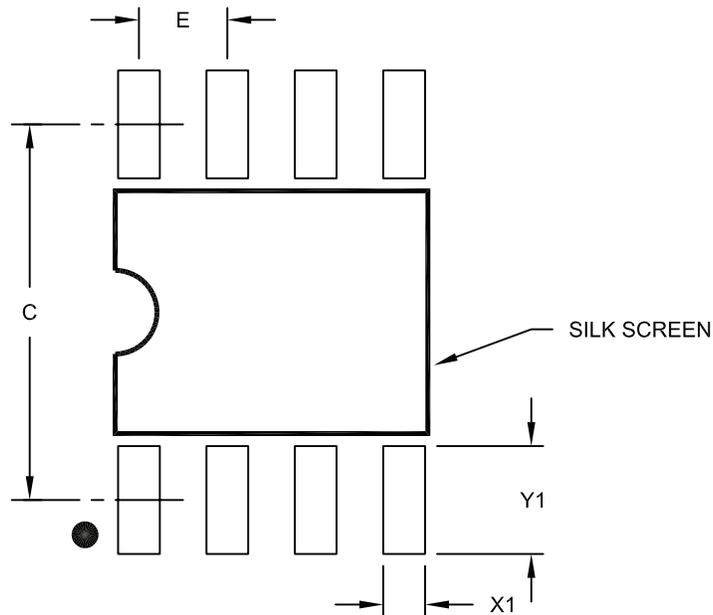
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-057C Sheet 2 of 2

PIC12(L)F1612/16(L)F1613

8 引脚塑封窄条小外形封装 (SN) —— 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 (X8)	X1			0.60
Contact Pad Length (X8)	Y1			1.55

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

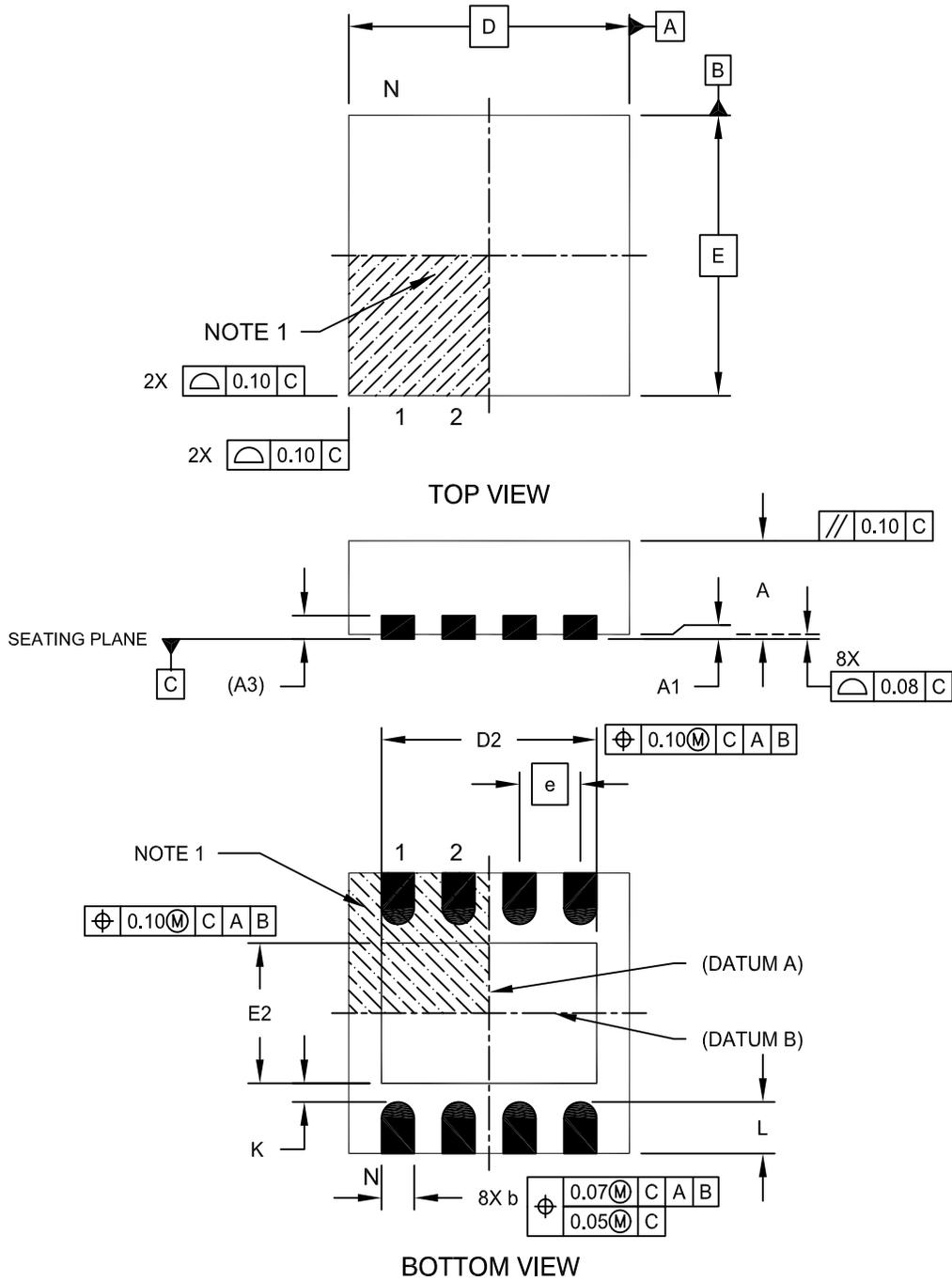
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2057A

PIC12(L)F1612/16(L)F1613

8 引脚塑封双列扁平无脚封装 (MF) —— 主体 3x3x0.9 mm [DFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

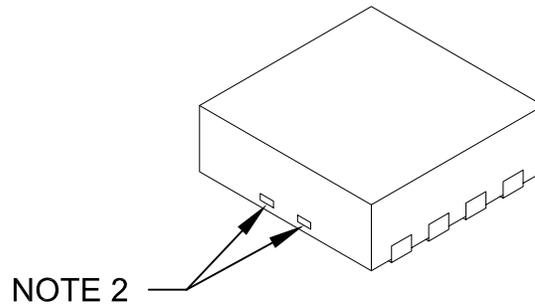


Microchip Technology Drawing No. C04-062C Sheet 1 of 2

PIC12(L)F1612/16(L)F1613

8 引脚塑封双列扁平无脚封装 (MF) —— 主体 3x3x0.9 mm [DFN]

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	8		
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 Length	D	3.00 BSC		
Exposed Pad Width	E2	1.34	-	1.60
Overall Width	E	3.00 BSC		
Exposed Pad Length	D2	1.60	-	2.40
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.20	0.30	0.55
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

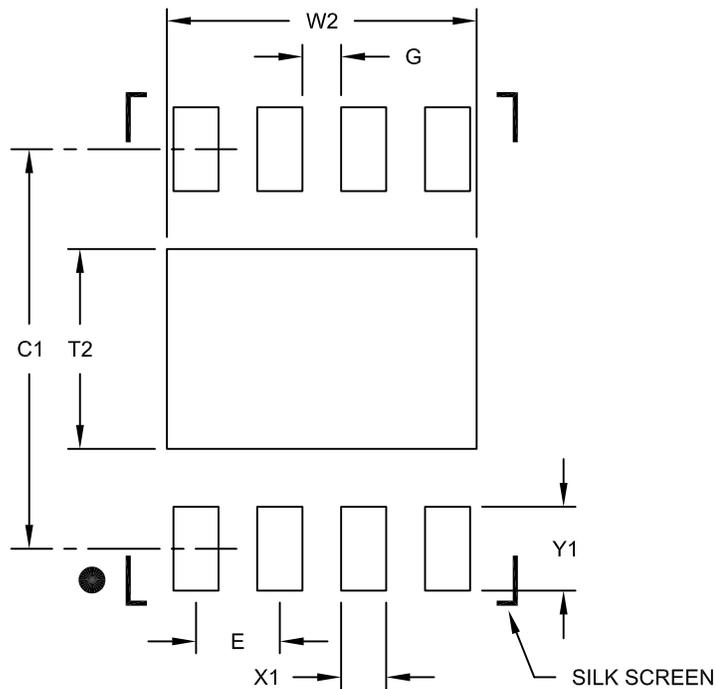
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package may have one or more exposed tie bars at ends.
3. Package is saw singulated
4. Dimensioning and tolerancing per ASME Y14.5M
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-062C Sheet 2 of 2

PIC12(L)F1612/16(L)F1613

8 引脚塑封双列扁平无脚封装 (MF) —— 主体 3x3x0.9 mm [DFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			2.40
Optional Center Pad Length	T2			1.55
Contact Pad Spacing	C1		3.10	
Contact Pad Width (X8)	X1			0.35
Contact Pad Length (X8)	Y1			0.65
Distance Between Pads	G	0.30		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

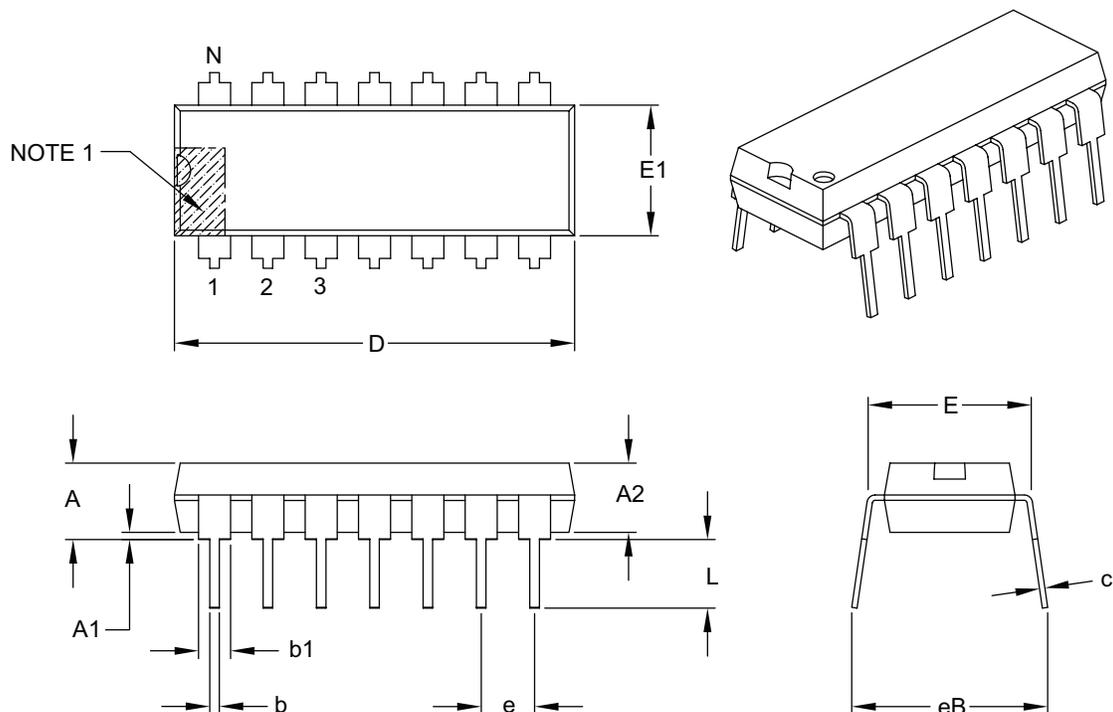
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2062B

PIC12(L)F1612/16(L)F1613

14 引脚塑封双列直插式封装 (P) —— 主体 300 mil [PDIP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

- Pin 1 visual index feature may vary, but must be located with the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

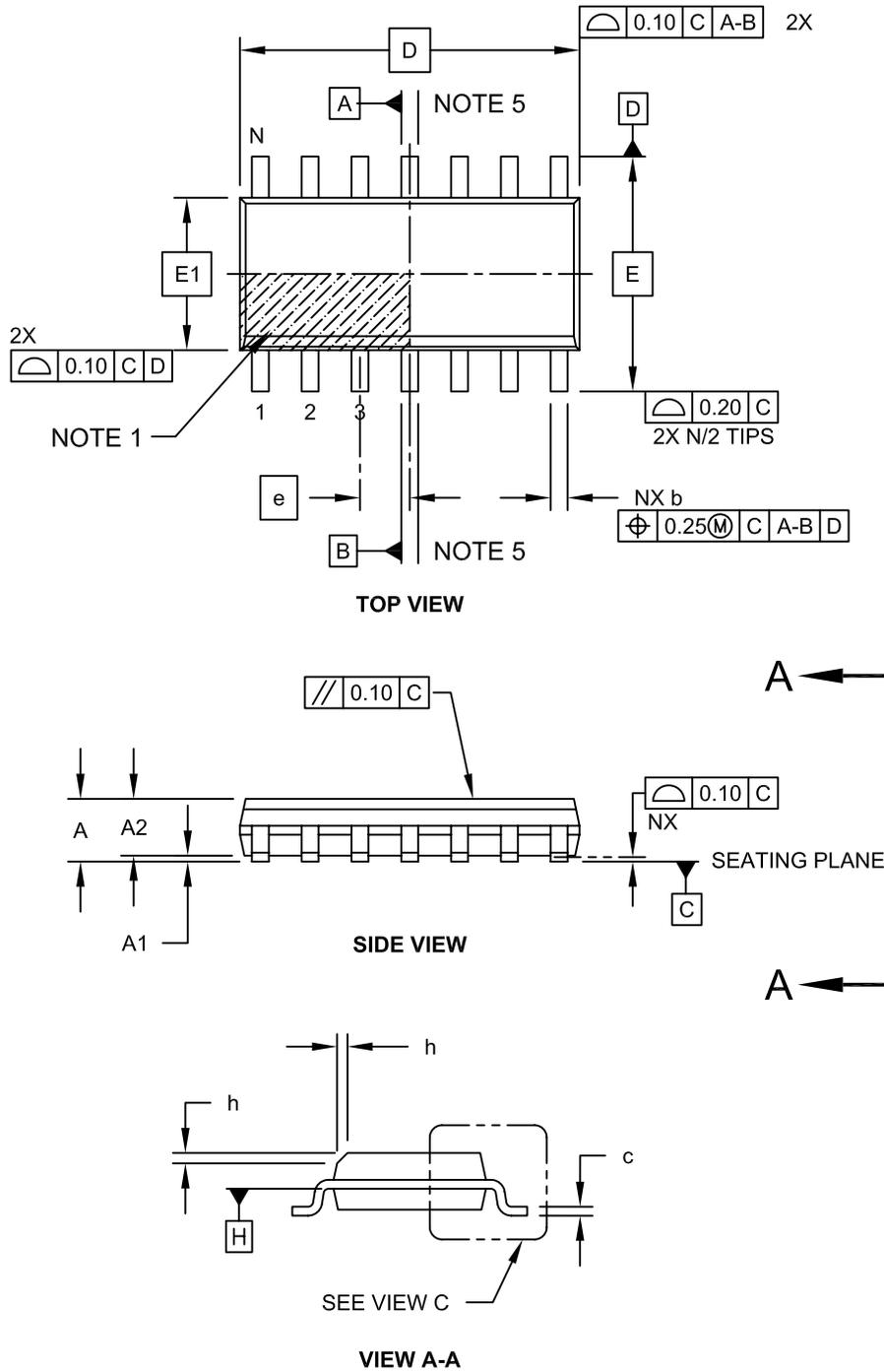
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

PIC12(L)F1612/16(L)F1613

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。

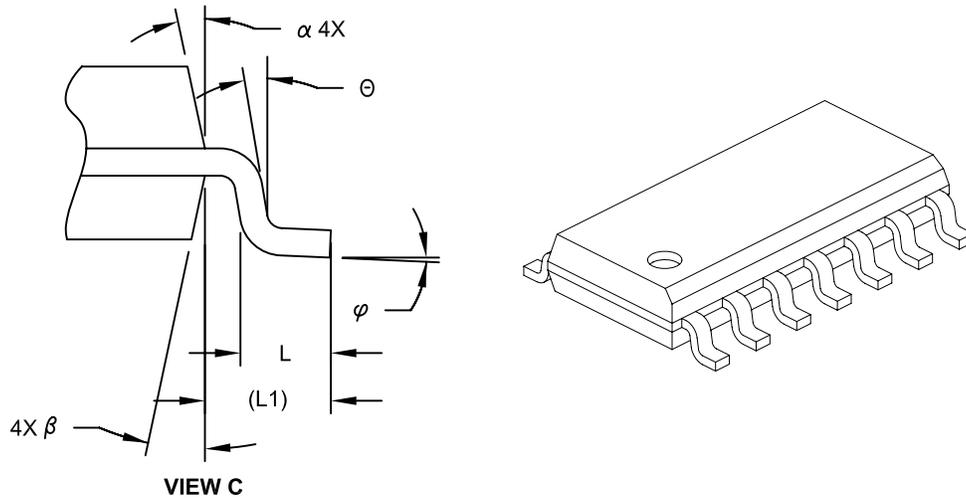


Microchip Technology Drawing No. C04-065C Sheet 1 of 2

PIC12(L)F1612/16(L)F1613

14 引脚塑封窄条小外形封装 (SL) —— 主体 3.90 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	8.65 BSC		
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1	1.04 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.10	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

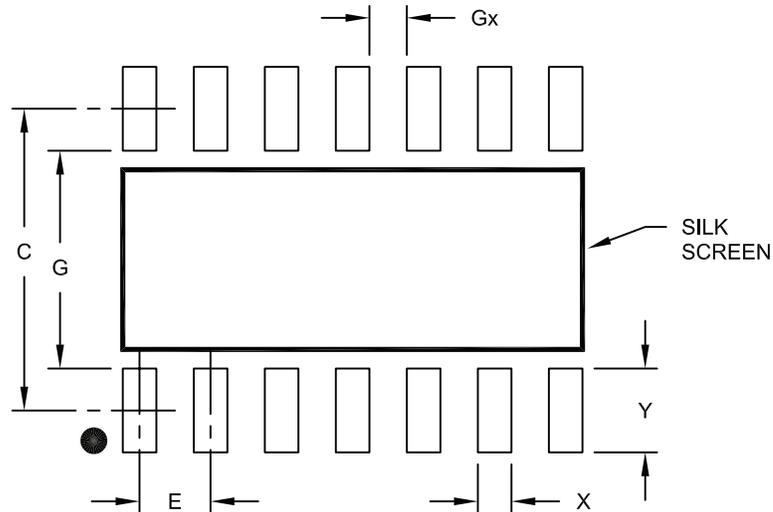
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2

PIC12(L)F1612/16(L)F1613

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

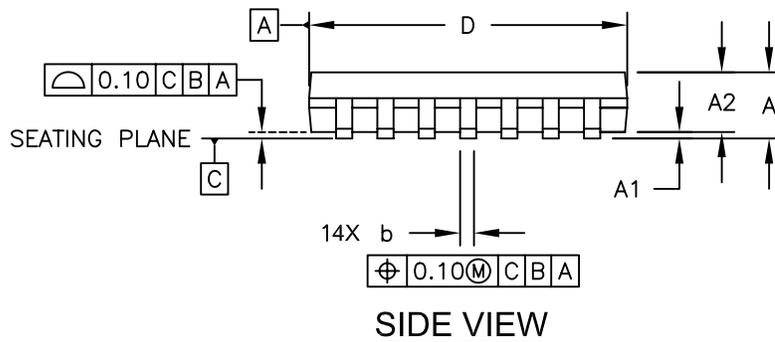
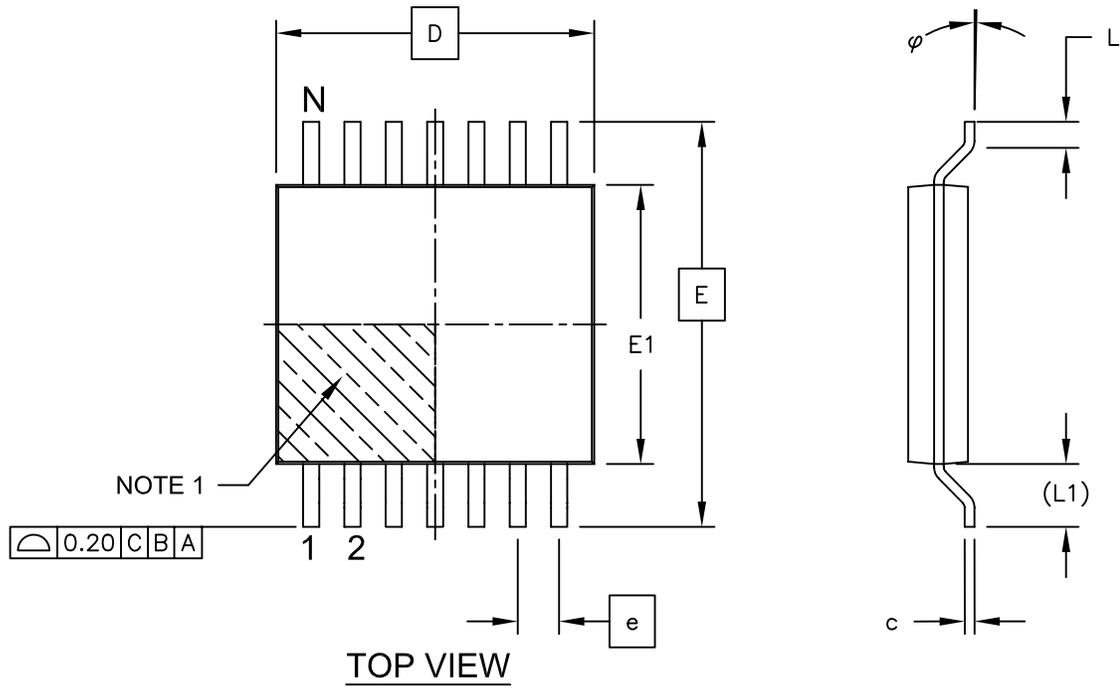
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2065A

PIC12(L)F1612/16(L)F1613

14 引脚塑封薄型缩小外形封装 (ST) —— 主体 4.4 mm [TSSOP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

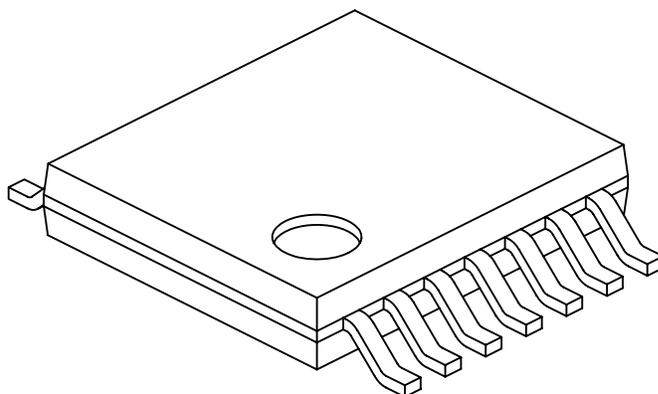


Microchip Technology Drawing C04-087C Sheet 1 of 2

PIC12(L)F1612/16(L)F1613

14 引脚塑封薄型缩小外形封装 (ST) ——主体 4.4 mm [TSSOP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	0.65 BSC		
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	-	0.15
Overall Width	E	6.40 BSC		
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	(L1)	1.00 REF		
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.19	-	0.30

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

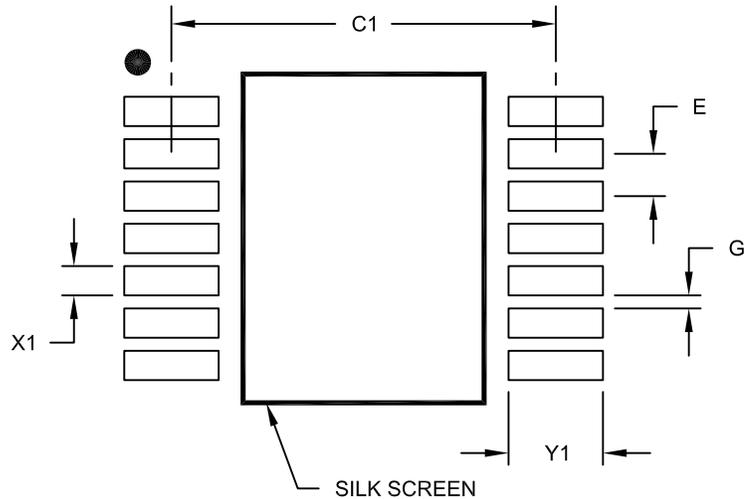
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

PIC12(L)F1612/16(L)F1613

14 引脚塑封薄型缩小外形封装 (ST) ——主体 4.4 mm [TSSOP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



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

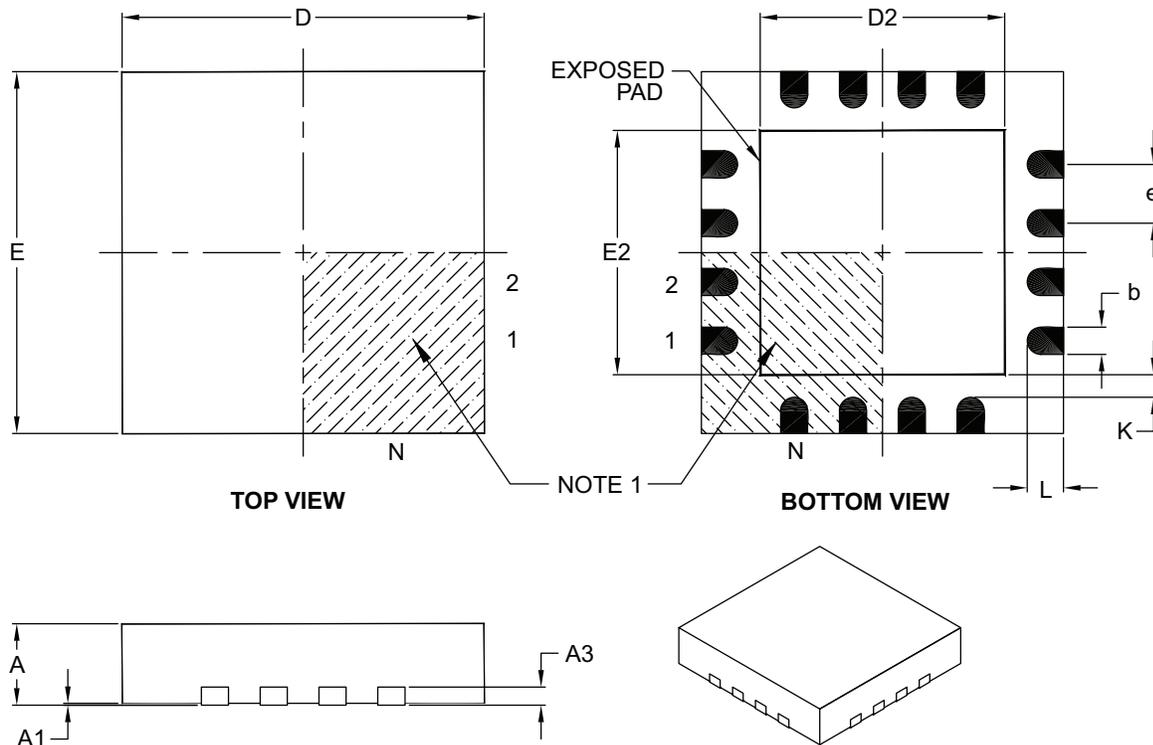
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

PIC12(L)F1612/16(L)F1613

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 Limits	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.

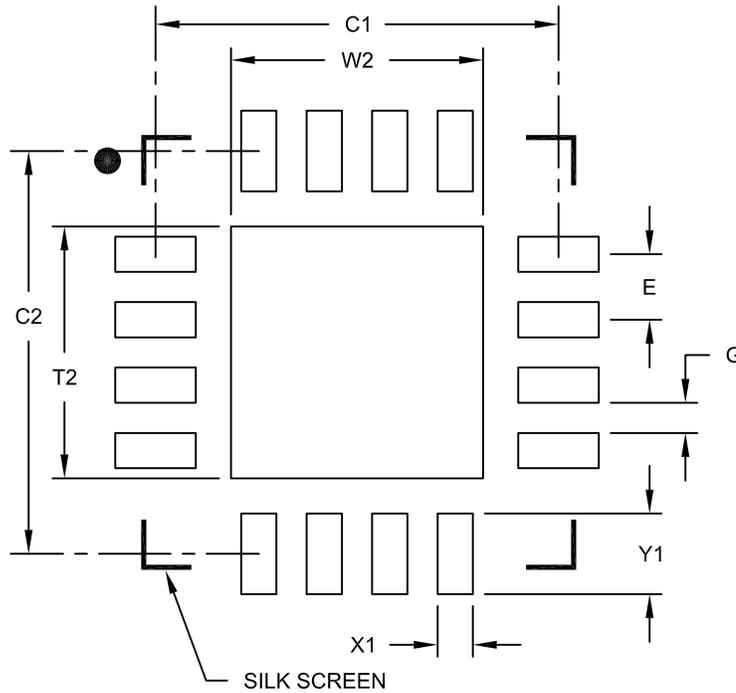
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-127B

PIC12(L)F1612/16(L)F1613

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。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		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

PIC12(L)F1612/16(L)F1613

附录 A: 版本历史

版本 A (2014 年 1 月)

初始版本。

MICROCHIP 网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 www.microchip.com。在“支持” (Support) 下，点击“变更通知客户 (Customer Change Notification)” 服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

PIC12(L)F1612/16(L)F1613

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

部件编号	[X] ⁽¹⁾	-	X	XX	XXX
器件	卷带式选项		温度范围	封装	定制信息
器件:	PIC12LF1612、PIC12F1612、 PIC16LF1613 和 PIC16F1613				
卷带式选项:	空白 = 标准封装 (料管或托盘封装) T = 卷带式 ⁽¹⁾				
温度范围:	I = -40°C 至 +85°C (工业级) E = -40°C 至 +125°C (扩展级)				
封装:⁽²⁾	MF = DFN (8 引脚) ML = QFN (16 引脚) P = 塑封 DIP SL = SOIC (14 引脚) SN = SOIC (8 引脚) ST = TSSOP				
定制编号:	QTP、SQTP、代码或特殊要求 (其他情况空白)				

示例:

- a) PIC12LF1612T - I/SN
卷带式,
工业级温度,
SOIC 封装
- b) PIC16F1613 - I/P
工业级温度,
PDIP 封装
- c) PIC16F1613 - 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部件编号描述中。该标识符用于订货目的，不会印刷在器件封装上。关于封装是否提供卷带式选项的信息，请咨询当地的 Microchip 销售办事处。

注 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，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。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、PIC³² 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、PICKtail、RightTouch 徽标、REAL ICE、SQL、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。

© 2014, Microchip Technology Inc. 版权所有。

ISBN: 978-1-63276-601-4

**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 and 模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

全球销售及及服务网点

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:

<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta

Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX

Tel: 1-512-257-3370

波士顿 Boston

Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago

Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland

Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas

Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit

Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX

Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis

Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles

Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

纽约 New York, NY

Tel: 1-631-435-6000

圣何塞 San Jose, CA

Tel: 1-408-735-9110

加拿大多伦多 Toronto

Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京

Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆

Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 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: 886-7-213-7830

台湾地区 - 台北

Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹

Tel: 886-3-5778-3666
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 - Munich

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Pforzheim

Tel: 49-7231-424750

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 - Druenen

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