



PIC16(L)F1574/5/8/9

带高精度 16 位 PWM 的 14/20 引脚 MCU

说明

PIC16(L)F1574/5/8/9 单片机将 16 位 PWM 与模拟功能相结合，可适应各种应用的需求。这些器件提供了 4 个带独立定时器的 16 位 PWM，适用于需要高分辨率的应用，例如 LED 照明、步进电机、电源和其他通用应用。独立于内核的外设（16 位 PWM 和互补波形发生器）、增强型通用同步 / 异步收发器（Enhanced Universal Synchronous Asynchronous Receiver Transceiver, EUSART）和模拟功能（ADC、比较器和 DAC）支持闭环反馈和通信，可用于多种细分市场。外设引脚选择（Peripheral Pin Select, PPS）功能可以对数字外设进行 I/O 引脚重映射，提供更高的灵活性。EUSART 外设支持通信，可用于诸如 LIN 之类的应用。

内核特性

- C 编译器优化的 RISC 架构
- 仅需学习 49 条指令
- 工作速度：
 - DC – 32 MHz 时钟输入
 - 最小指令周期为 125 ns
- 中断功能
- 16 级深硬件堆栈
- 2 个 8 位定时器
- 1 个 16 位定时器
- 使用 16 位 PWM 可提供 4 个额外的 16 位定时器
- 上电复位（Power-on Reset, POR）
- 上电延时定时器（Power-up Timer, PWRT）
- 低功耗欠压复位（Low-Power Brown-out Reset, LPBOR）
- 可编程看门狗定时器（Watchdog Timer, WDT），超时周期最长 256s
- 可编程代码保护

存储器

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

工作特性

- 工作电压范围：
 - 1.8V 至 3.6V (PIC16LF1574/5/8/9)
 - 2.3V 至 5.5V (PIC16F1574/5/8/9)
- 温度范围：
 - 工业级：-40°C 至 85°C
 - 扩展级：-40°C 至 125°C
- 内部参考电压模块
- 通过两个引脚进行在线串行编程（In-Circuit Serial Programming™, ICSPTM）

超低功耗（eXtreme Low-Power, XLP）特性：

- 休眠模式：20 nA (1.8V 时, 典型值)
- 看门狗定时器：260 nA (1.8V 时, 典型值)
- 工作电流：
 - 30 μA/MHz (1.8V 时, 典型值)

数字外设

- 16 位 PWM：
 - 4 个带有独立定时器的 16 位 PWM
 - 多种输出模式（标准、中心对齐、寄存器匹配时电平置 1 和翻转）
 - 用于相位、占空比、周期、偏移和极性的用户设置
 - 16 位定时器功能
 - 可基于定时器与偏移、占空比、周期和相位寄存器的匹配产生中断
- 互补波形发生器（Complementary Waveform Generator, CWG）：
 - 上升沿和下降沿死区控制
 - 多个信号源
- 增强型通用同步 / 异步收发器（EUSART）：
 - 支持 LIN 应用
- 外设引脚选择（PPS）：
 - 可对数字外设进行 I/O 引脚重映射

器件 I/O 端口特性

- 最多 18 个 I/O
- 可单独选择的弱上拉
- 边沿可选的电平变化中断引脚选项

PIC16(L)F1574/5/8/9

模拟外设

- 10 位模数转换器 (Analog-to-Digital Converter, ADC):
 - 最多 12 路外部通道
 - 可在休眠模式下进行转换
- 2 个比较器:
 - 低功耗 / 高速模式
 - (同相) 反相输入可使用固定参考电压
 - 比较器输出可供外部访问
 - 可与 Timer1 时钟源进行同步
 - 软件滞后使能
- 5 位数模转换器 (Digital-to-Analog Converter, DAC):
 - 5 位分辨率, 轨到轨
 - 正参考电压选择
 - 无缓冲的 I/O 引脚输出
 - 到 ADC 和比较器的内部连接
- 参考电压:
 - 具有 1.024V、2.048V 和 4.096V 输出的固定参考电压

时钟结构

- 高精度内部振荡器:
 - 出厂时精度校准为 $\pm 1\%$, 典型值
 - 可用软件选择时钟速度, 从 31 kHz 至 32 MHz
- 外部振荡器模块具有:
 - 2 种外部时钟模式, 频率最高为 32 MHz
- 有数字振荡器输入可供使用

表 1: PIC12(L)F1571/2 和 PIC16(L)F1574/5/8/9 系列类型

器件	数据手册索引	闪存程序存储器 (K字)	闪存程序存储器 (KB)	数据 SRAM (字节)	I/O 引脚	8 位 / 16 位定时器	比较器	16 位 PWM	10 位 ADC (通道)	5 位 DAC	CWG	EUSART	PPS	调试 ⁽¹⁾
PIC12(L)F1571	(A)	1	1.75	128	6	2/4 ⁽²⁾	1	3	4	1	1	0	无	—
PIC12(L)F1572	(A)	2	3.5	256	6	2/4 ⁽²⁾	1	3	4	1	1	1	无	—
PIC16(L)F1574	(B)	4	7	512	12	2/5 ⁽³⁾	2	4	8	1	1	1	有	—
PIC16(L)F1575	(B)	8	14	1024	12	2/5 ⁽³⁾	2	4	8	1	1	1	有	—
PIC16(L)F1578	(B)	4	7	512	18	2/5 ⁽³⁾	2	4	12	1	1	1	有	—
PIC16(L)F1579	(B)	8	14	1024	18	2/5 ⁽³⁾	2	4	12	1	1	1	有	—

注 1: — 片上集成的调试功能。

2: 不使用 16 位 PWM 输出时, 有 3 个额外的 16 位定时器可用。

3: 不使用 16 位 PWM 输出时, 有 4 个额外的 16 位定时器可用。

数据手册索引:

- A) DS-40001723 PIC12(L)F1571/2 数据手册, 带高精度 16 位 PWM 的 8 引脚闪存型 8 位 MCU
B) DS-40001782 PIC16(L)F1574/5/8/9 数据手册, 带高精度 16 位 PWM 的 8 引脚闪存型 8 位 MCU

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

表 2: 封装

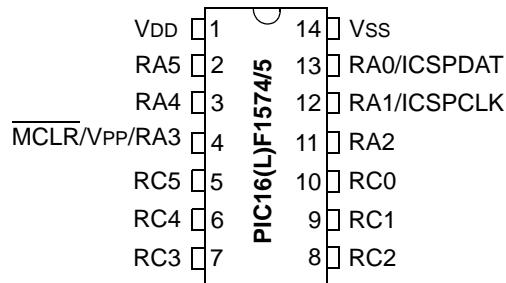
封装	PDIP	SOIC	TSSOP	SSOP	UQFN
PIC16(L)F1574	•	•	•		•
PIC16(L)F1575	•	•	•		•
PIC16(L)F1578	•	•		•	•
PIC16(L)F1579	•	•		•	•

注: 引脚详细信息可能有所变动。

PIC16(L)F1574/5/8/9

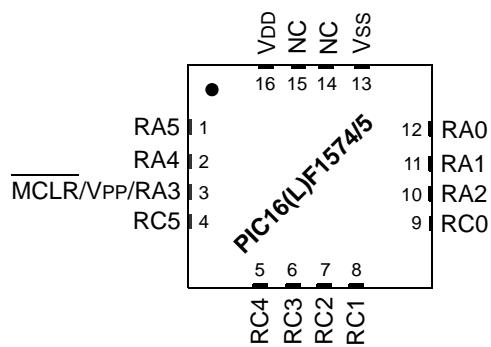
引脚图

图 1: 14 引脚 PDIP、SOIC 和 TSSOP



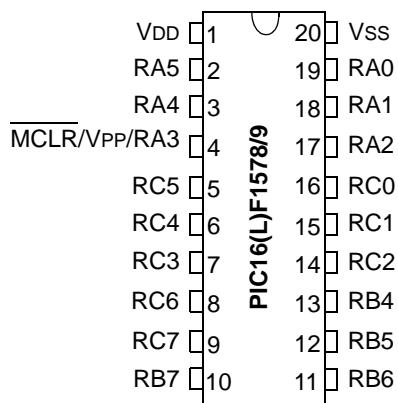
注: 关于引脚分配表, 请参见表 3。

图 2: 16 引脚 UQFN (4x4)



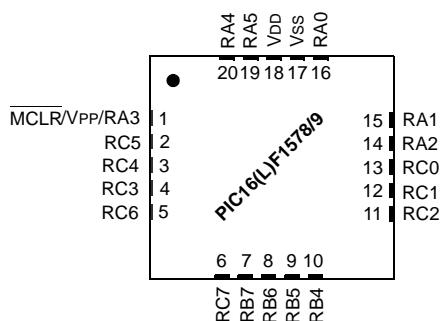
注: 关于引脚分配表, 请参见表 3。

图 3: 20 引脚 PDIP、SOIC 和 SSOP



注: 关于引脚分配表, 请参见表 4。

图 4: 20 引脚 UQFN (4x4)



注: 关于引脚分配表, 请参见表 4。

PIC16(L)F1574/5/8/9

引脚分配表

表 3: 14/16 引脚分配表 (PIC16(L)F1574/5)

I/O	14 引脚 PDIP/SOIC/TSSOP	16 引脚 UQFN	ADC	参考电压	比较器	定时器	PWM	EUSART	CWG	中断	上拉	基本功能
RA0	13	12	AN0	DAC1OUT1	C1IN+	—	—	—	—	IOC	有	ICSPDAT
RA1	12	11	AN1	VREF+	C1IN0-/C2IN0-	—	—	—	—	IOC	有	ICSPCLK
RA2	11	10	AN2	—	—	T0CKI ⁽¹⁾	—	—	CWG1IN ⁽¹⁾	INT ⁽¹⁾ /IOC	有	—
RA3	4	3	—	—	—	—	—	—	—	IOC	有	MCLR/VPP
RA4	3	2	AN3	—	—	T1G ⁽¹⁾	—	—	—	IOC	有	CLKOUT
RA5	2	1	—	—	—	T1CKI ⁽¹⁾	—	—	—	IOC	有	CLKIN
RC0	10	9	AN4	—	C2IN+	—	—	—	—	IOC	有	—
RC1	9	8	AN5	—	C1IN1-/C2IN1-	—	—	—	—	IOC	有	—
RC2	8	7	AN6	—	C1IN2-/C2IN2-	—	—	—	—	IOC	有	—
RC3	7	6	AN7	—	C1IN3-/C2IN3-	—	—	—	—	IOC	有	—
RC4	6	5	ADCACT ⁽¹⁾	—	—	—	—	CK ⁽¹⁾	—	IOC	有	—
RC5	5	4	—	—	—	—	—	RX ^(1,3)	—	IOC	有	—
VDD	1	16	—	—	—	—	—	—	—	—	—	VDD
Vss	14	13	—	—	—	—	—	—	—	—	—	VSS
OUT ⁽²⁾	—	—	—	—	C1OUT	—	PWM1OUT	DT ⁽³⁾	CWG1A	—	—	—
	—	—	—	—	C2OUT	—	PWM2OUT	CK	CWG1B	—	—	—
	—	—	—	—	—	—	PWM3OUT	TX	—	—	—	—
	—	—	—	—	—	—	PWM4OUT	—	—	—	—	—

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

2: 所有引脚输出均默认为端口锁存器数据。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。

3: 这些外设功能是双向的。输出引脚选择必须与输入引脚选择相同。

表 4: 20 引脚分配表 (PIC16(L)F1578/9)

I/O	20 引脚 PDIP/SOIC/SSOP			20 引脚 UQFN			ADC	参考电压	比较器	定时器	PWM	EUSART	CWG	中断	上拉	基本功能
RA0	19	16	AN0	DAC1OUT1	C1IN+	—	—	—	—	—	—	—	IOC	有	ICSPDAT	
RA1	18	15	AN1	VREF+	C1IN0-/C2IN0-	—	—	—	—	—	—	—	IOC	有	ICSPCLK	
RA2	17	14	AN2	—	—	T0CKI ⁽¹⁾	—	—	—	CWG1IN ⁽¹⁾	INT ⁽¹⁾ /IOC	有	—	—	—	—
RA3	4	1	—	—	—	—	—	—	—	—	—	—	IOC	有	MCLR/VPP	
RA4	3	20	AN3	—	—	T1G ⁽¹⁾	—	—	—	—	—	—	IOC	有	CLKOUT	
RA5	2	19	—	—	—	T1CKI ⁽¹⁾	—	—	—	—	—	—	IOC	有	CLKIN	
RB4	13	10	AN10	—	—	—	—	—	—	—	—	—	IOC	有	—	
RB5	12	9	AN11	—	—	—	—	—	—	RX ^(1,3)	—	—	IOC	有	—	
RB6	11	8	—	—	—	—	—	—	—	—	—	—	IOC	有	—	
RB7	10	7	—	—	—	—	—	—	CK ⁽¹⁾	—	—	—	IOC	有	—	
RC0	16	13	AN4	—	C2IN+	—	—	—	—	—	—	—	IOC	有	—	
RC1	15	12	AN5	—	C1IN1-/C2IN1-	—	—	—	—	—	—	—	IOC	有	—	
RC2	14	11	AN6	—	C1IN2-/C2IN2-	—	—	—	—	—	—	—	IOC	有	—	
RC3	7	4	AN7	—	C1IN3-/C2IN3-	—	—	—	—	—	—	—	IOC	有	—	
RC4	6	3	ADCACT ⁽¹⁾	—	—	—	—	—	—	—	—	—	IOC	有	—	
RC5	5	2	—	—	—	—	—	—	—	—	—	—	IOC	有	—	
RC6	8	5	AN8	—	—	—	—	—	—	—	—	—	IOC	有	—	
RC7	9	6	AN9	—	—	—	—	—	—	—	—	—	IOC	有	—	
VDD	1	18	—	—	—	—	—	—	—	—	—	—	—	—	VDD	
Vss	20	17	—	—	—	—	—	—	—	—	—	—	—	—	Vss	
OUT ⁽²⁾	—	—	—	—	C1OUT	—	PWM1OUT	DT ⁽³⁾	CWG1A	—	—	—	—	—	—	—
	—	—	—	—	C2OUT	—	PWM2OUT	CK	CWG1B	—	—	—	—	—	—	—
	—	—	—	—	—	—	PWM3OUT	TX	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	PWM4OUT	—	—	—	—	—	—	—	—	—

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

2: 所有引脚输出均默认为端口锁存器数据。通过 PPS 输出选择寄存器，可以选择任意引脚作为数字外设输出。

3: 这些外设功能是双向的。输出引脚选择必须与输入引脚选择相同。

目录

1.0 器件概述	10
2.0 增强型中档 CPU	17
3.0 存储器构成	19
4.0 器件配置	55
5.0 振荡器模块	61
6.0 复位	73
7.0 中断	81
8.0 掉电模式（休眠）	94
9.0 看门狗定时器（WDT）	97
10.0 闪存程序存储器控制	101
11.0 I/O 端口	117
12.0 外设引脚选择（PPS）模块	135
13.0 电平变化中断	141
14.0 固定参考电压（FVR）	147
15.0 温度指示器模块	150
16.0 模数转换器（ADC）模块	152
17.0 5 位数模转换器（DAC）模块	166
18.0 比较器模块	169
19.0 Timer0 模块	176
20.0 带门控的 Timer1 模块	179
21.0 Timer2 模块	189
22.0 增强型通用同步 / 异步收发器（EUSART）	192
23.0 16 位脉宽调制（PWM）模块	220
24.0 互补波形发生器（CWG）模块	246
25.0 在线串行编程（ICSP™）	258
26.0 指令集汇总	260
27.0 电气规范	274
28.0 直流和交流特性图表	298
29.0 开发支持	312
30.0 封装信息	316
附录 A: 数据手册版本历史	338
Microchip 网站	339
变更通知客户服务	339
客户支持	339
产品标识体系	340

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要求。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，请查询我公司的网站：

<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。文献编号中数字串后的字母是版本号，例如：DS30000000A 是 DS30000000 的 A 版本。

勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站：<http://www.microchip.com>

- 当地 Microchip 销售办事处（见最后一页）

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

客户通知系统

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

PIC16(L)F1574/5/8/9

1.0 器件概述

本数据手册介绍了 PIC16(L)F1574/5/8/9 器件。图 1-1 给出了这些器件的框图，表 1-1 列出了可用的外设，表 1-2 和表 1-3 列出了引脚说明。

表 1-1：器件外设汇总

外设	PIC16(L)F1574	PIC16(L)F1575	PIC16(L)F1578	PIC16(L)F1579
模数转换器 (ADC)	●	●	●	●
互补波形发生器 (CWG)	●	●	●	●
数模转换器 (DAC)	●	●	●	●
增强型通用 同步 / 异步收发器 (EUSART)	●	●	●	●
固定参考电压 (Fixed Voltage Reference, FVR)	●	●	●	●
温度指示器	●	●	●	●
比较器	C1	●	●	●
	C2	●	●	●
PWM 模块	PWM1	●	●	●
	PWM2	●	●	●
	PWM3	●	●	●
	PWM4	●	●	●
定时器	Timer0	●	●	●
	Timer1	●	●	●
	Timer2	●	●	●

1.1 寄存器和位的命名约定

1.1.1 寄存器名称

当器件中存在相同外设的多个实例时，外设控制寄存器将以外设标识符、外设实例和控制标识符相连接的形式来表示。控制寄存器部分将在外设实例编号的位置使用一个“x”，仅显示所有寄存器名称的一个实例。在器件中只有该外设的一个实例时，也可能对外设应用这种命名约定，以便与器件系列中包含多个外设的其他器件保持兼容。

1.1.2 位名称

位名称有两种形式：

- 短名称：位功能缩写
- 长名称：外设缩写 + 短名称

1.1.2.1 短位名称

短位名称是位功能的缩写。例如，一些外设使用 EN 位来使能。寄存器中显示的位名称为短名称形式。

短位名称在 C 程序中访问位时非常有用。通过短名称访问位的一般格式为 **寄存器名称bits. 短名称**。例如，COG1CON0 寄存器中的使能位 EN 可以在 C 程序中使用指令 COG1CON0bits.EN = 1 置 1。

短名称在汇编程序中通常没什么用，因为不同的外设可能在不同的位位置使用相同的名称。发生这种情况时，在生成包含文件期间，该短位名称的所有实例后都会被加上一个下划线以及位所在的寄存器的名称，以避免命名争用。

1.1.2.2 长位名称

长位名称通过在短名称前添加外设缩写前缀来构造。前缀对外设是唯一的，从而使每个长位名称也是唯一的。COG1 使能位的长位名称为 COG1 前缀 G1 后加上使能位短名称 EN，产生唯一的位名称 G1EN。

长位名称在 C 和汇编程序中都非常有用。例如，在 C 程序中，COG1CON0 使能位可以使用 G1EN = 1 指令置 1。在汇编程序中，该位可以使用 BSF COG1CON0,G1EN 指令置 1。

1.1.2.3 位域

位域是同一寄存器中的两个或更多个相邻位。位域仅遵从短位命名约定。例如，COG1CON0 寄存器的 3 个低位包含模式控制位。该位域的简称为 MD，没有长位名称形式。位域只能在 C 程序中进行访问。以下示例给出了将 COG1 设置为推挽模式的 C 程序指令：

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

位域中的各个位也可以使用长位名称和短位名称进行访问。每个位的名称为位域名称后加上位域内的位位置编号。例如，最高模式位的短位名称为 MD2，长位名称为 G1MD2。以下两个示例给出了将 COG1 设置为推挽模式的汇编程序序列：

例 1：

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

例 2：

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

1.1.3 寄存器和位的命名例外

1.1.3.1 状态、中断和镜像位

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

1.1.3.2 旧式外设

有一些外设并不严格遵从这些命名约定。已存在多年且几乎每个器件中都有的外设属于例外。这些例外是必要的，是为了限制新约定对于旧式代码的负面影响。遵从新约定的外设都在寄存器部分包含了一个表，指明每个外设实例的长名称前缀。属于例外类别的外设没有该表。这些外设包括但不限于以下外设：

- EUSART
- MSSP

PIC16(L)F1574/5/8/9

图 1-1： PIC16(L)F1574/5/8/9 框图

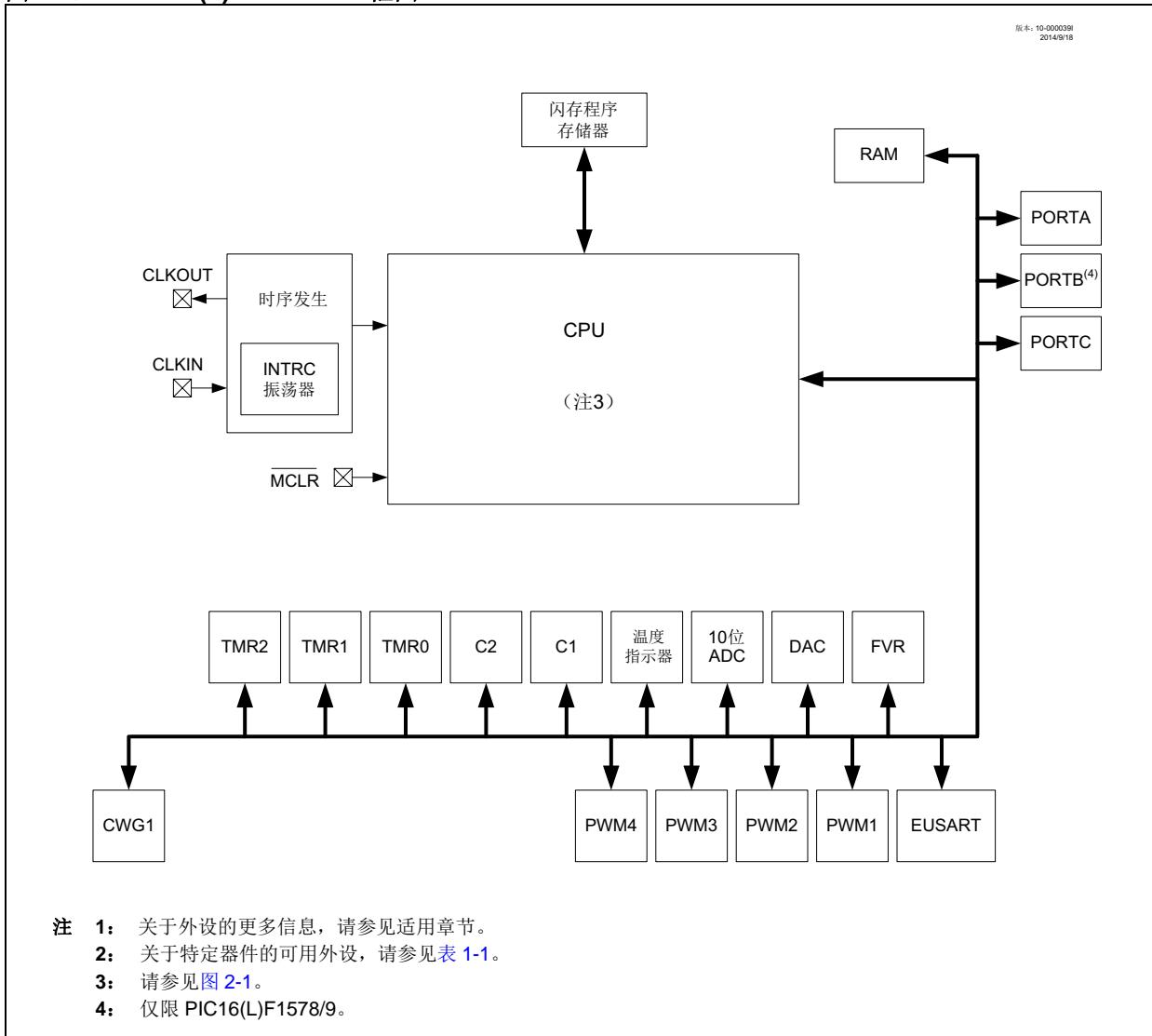


表 1-2: PIC16(L)F1574/5 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/DAC1OUT1/ ICSPDAT	RA0	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN0	AN	—	ADC 通道的输入。
	C1IN+	AN	—	比较器的同相输入。
	DAC1OUT1	—	AN	数模转换器输出。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RA1/AN1/VREF+/C1IN0-/C2IN0-/ ICSPCLK	RA1	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN1	AN	—	ADC 通道的输入。
	VREF+	AN	—	参考电压输入。
	C1IN0-	AN	—	比较器的反相输入。
	C2IN0-	AN	—	比较器的反相输入。
	ICSPCLK	ST	—	ICSP 编程时钟。
RA2/AN2/T0CKI ⁽¹⁾ /CWG1IN ⁽¹⁾ / INT ⁽¹⁾	RA2	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN2	AN	—	ADC 通道的输入。
	T0CKI	TTL/ST	—	Timer0 时钟输入。
	CWG1IN	TTL/ST	—	CWG 互补输入。
	INT	TTL/ST	—	外部中断。
RA3/VPP/MCLR	RA3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	VPP	HV	—	编程电压。
	MCLR	ST	—	带内部上拉的主复位。
RA4/AN3/T1G ⁽¹⁾ /CLKOUT	RA4	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN3	AN	—	ADC 通道的输入。
	T1G	TTL/ST	—	Timer1 门控输入。
	CLKOUT	CMOS/OD	CMOS	Fosc/4 输出。
RA5/CLKIN/T1CKI ⁽¹⁾	RA5	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	CLKIN	CMOS	—	外部时钟输入 (EC 模式)。
	T1CKI	TTL/ST	—	Timer1 时钟输入。
RC0/AN4/C2IN+	RC0	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN4	AN	—	ADC 通道的输入。
	C2IN+	AN	—	比较器的同相输入。
RC1/AN5/C1IN1-/C2IN1-	RC1	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN5	AN	—	ADC 通道的输入。
	C1IN1-	AN	—	比较器的反相输入。
	C2IN1-	AN	—	比较器的反相输入。
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN6	AN	—	ADC 通道的输入。
	C1IN2-	AN	—	比较器的反相输入。
	C2IN2-	AN	—	比较器的反相输入。
RC3/AN7/C1IN3-/C2IN3-	RC3	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN7	AN	—	ADC 通道的输入。
	C1IN3-	AN	—	比较器的反相输入。
	C2IN3-	AN	—	比较器的反相输入。

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

2: 所有引脚输出均默认为端口锁存器数据。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见 [寄存器 12-1](#)。

3: 这些 USART 功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F1574/5/8/9

表 1-2：PIC16(L)F1574/5 引脚说明（续）

名称	功能	输入类型	输出类型	说明
RC4/ADCACT ⁽¹⁾ /CK ⁽¹⁾	RC4	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	ADCACT	TTL/ST	—	ADC 自动转换触发输入。
	CK	ST	CMOS	USART 同步时钟。
RC5/RX ^(1,3)	RC5	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	RX	ST	—	USART 异步输入。
OUT ⁽²⁾	C1OUT	—	CMOS	比较器输出。
	C2OUT	—	CMOS	比较器输出。
	PWM1OUT	—	CMOS	PWM1 输出。
	PWM2OUT	—	CMOS	PWM2 输出。
	PWM3OUT	—	CMOS	PWM3 输出。
	PWM4OUT	—	CMOS	PWM4 输出。
	CWG1A	—	CMOS	互补输出发生器输出 A。
	CWG1B	—	CMOS	互补输出发生器输出 B。
	TX/CK	—	CMOS	USART 异步发送数据 / 同步时钟输出。
VDD	VDD	电源	—	正电源。
	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：默认外设输入。通过 PPS 输入选择寄存器，可以将输入移至任何其他引脚。
 2：所有引脚输出均默认为端口锁存器数据。通过 PPS 输出选择寄存器，可以选择任意引脚作为数字外设输出。请参见 [寄存器 12-1](#)。
 3：这些 USART 功能是双向的。输出引脚选择必须与输入引脚选择相同。

表 1-3: PIC16(L)F1578/9 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/DAC1OUT/ ICSPDAT	RA0	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN0	AN	—	ADC 通道的输入。
	C1IN+	AN	—	比较器的同相输入。
	DAC1OUT	—	AN	数模转换器输出。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RA1/AN1/VREF+/C1IN0-/C2IN0-/ ICSPCLK	RA1	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN1	AN	—	ADC 通道的输入。
	VREF+	AN	—	参考电压输入。
	C1IN0-	AN	—	比较器的反相输入。
	C2IN0-	AN	—	比较器的反相输入。
	ICSPCLK	ST	—	ICSP 编程时钟。
RA2/AN2/T0CKI ⁽¹⁾ /CWG1IN ⁽¹⁾ / INT ⁽¹⁾	RA2	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN2	AN	—	ADC 通道的输入。
	T0CKI	TTL/ST	—	Timer0 时钟输入。
	CWG1IN	TTL/ST	—	CWG 互补输入。
	INT	TTL/ST	—	外部中断。
RA3/VPP/MCLR	RA3	TTL/ST	—	带 IOC 和 WPU 的通用输入。
	VPP	HV	—	编程电压。
	MCLR	ST	—	带内部上拉的主复位。
RA4/AN3/T1G ⁽¹⁾ /CLKOUT	RA4	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN3	AN	—	ADC 通道的输入。
	T1G	TTL/ST	—	Timer1 门控输入。
	CLKOUT	—	CMOS	FOSC/4 输出。
RA5/CLKIN/T1CKI ⁽¹⁾	RA5	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	CLKIN	CMOS	—	外部时钟输入 (EC 模式)。
	T1CKI	TTL/ST	—	Timer1 时钟输入。
RB4/AN10	RB4	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN10	AN	—	ADC 通道的输入。
RB5/AN11/RX ⁽¹⁾	RB5	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN11	AN	—	ADC 通道的输入。
	RX	ST	—	USART 异步输入。
RB6	RB6	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
RB7/CK	RB7	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	CK	ST	CMOS	USART 同步时钟。
RC0/AN4/C2IN+	RC0	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN4	AN	—	ADC 通道的输入。
	C2IN+	AN	—	比较器的同相输入。

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

2: 所有引脚输出均默认为端口锁存器数据。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见 [寄存器 12-1](#)。

3: 这些 USART 功能是双向的。输出引脚选择必须与输入引脚选择相同。

PIC16(L)F1574/5/8/9

表 1-3: PIC16(L)F1578/9 引脚说明 (续)

名称	功能	输入类型	输出类型	说明
RC1/AN5/C1IN1-/C2IN1-	RC1	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN5	AN	—	ADC 通道的输入。
	C1IN1-	AN	—	比较器的反相输入。
	C2IN1-	AN	—	比较器的反相输入。
RC2/AN6/C1IN2-/C2IN2-	RC2	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN6	AN	—	ADC 通道的输入。
	C1IN2-	AN	—	比较器的反相输入。
	C2IN2-	AN	—	比较器的反相输入。
RC3/AN7/C1IN3-/C2IN3-	RC3	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	AN7	AN	—	ADC 通道的输入。
	C1IN3-	AN	—	比较器的反相输入。
	C2IN3-	AN	—	比较器的反相输入。
RC4/ADCACT ⁽¹⁾	RC4	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
	ADCACT	TTL/ST	—	ADC 自动转换触发输入。
RC5	RC5	TTL/ST	CMOS/OD	带 IOC 和 WPU 的通用输入。
OUT ⁽²⁾	C1OUT	—	CMOS	比较器输出。
	C2OUT	—	CMOS	比较器输出。
	PWM1OUT	—	CMOS	PWM1 输出。
	PWM2OUT	—	CMOS	PWM2 输出。
	PWM3OUT	—	CMOS	PWM3 输出。
	PWM4OUT	—	CMOS	PWM4 输出。
	CWG1A	—	CMOS	互补输出发生器输出 A。
	CWG1B	—	CMOS	互补输出发生器输出 B。
	TX/CK	—	CMOS	USART 异步发送数据 / 同步时钟输出。
DT ⁽³⁾	—	CMOS	—	USART 同步数据输出。
VDD	VDD	电源	—	正电源。
Vss	Vss	电源	—	参考地。

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

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

2: 所有引脚输出均默认为端口锁存器数据。通过 PPS 输出选择寄存器, 可以选择任意引脚作为数字外设输出。请参见 [寄存器 12-1](#)。

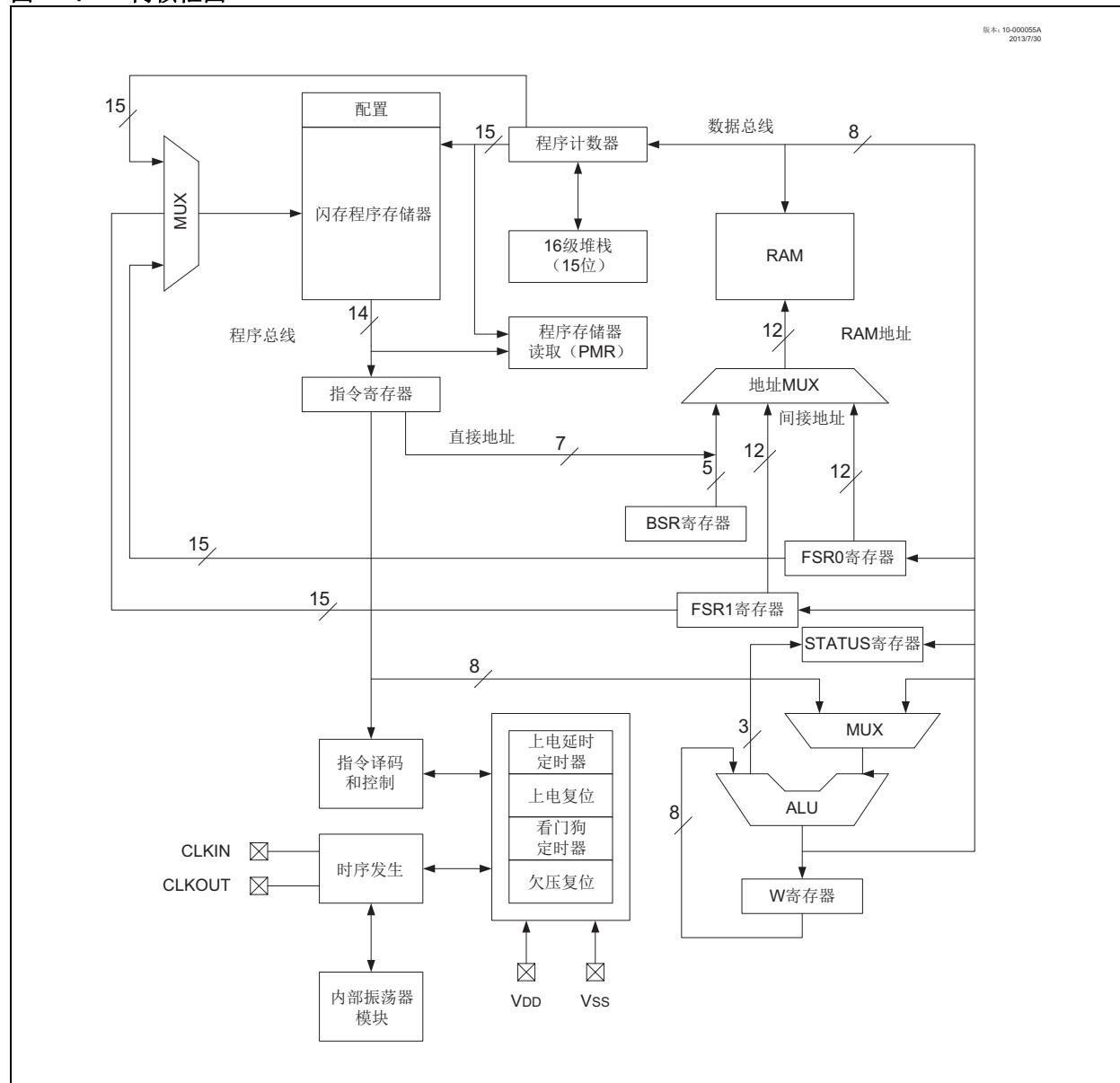
3: 这些 USART 功能是双向的。输出引脚选择必须与输入引脚选择相同。

2.0 增强型中档 CPU

本系列器件包含了增强型中档 8 位 CPU 内核。CPU 具有 49 条指令。中断功能包含了自动现场保护功能。硬件堆栈为 16 级深，具有上溢和下溢复位功能。器件提供了直接寻址、间接寻址和相对寻址模式。用户可以通过两个文件选择寄存器（File Select Register, FSR）来读取程序和数据存储器。

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

图 2-1：内核框图



2.1 自动中断现场保护

在中断期间，器件会自动将一些寄存器保存到影子寄存器中，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 3.5 节“堆栈”](#)。

2.3 文件选择寄存器

有两个 16 位文件选择寄存器（FSR）。FSR 可以访问所有文件寄存器和程序存储器，支持对所有存储器使用一个数据指针。当 FSR 指向程序存储器时，使用 INDF 的指令需要一个额外的指令周期来取数据。通用存储器现在可以进行线性寻址，支持访问大于 80 字节的连续数据。此外，还有一些支持 FSR 的新指令。更多详细信息，请参见[第 3.6 节“间接寻址”](#)。

2.4 指令集

增强型中档 CPU 具有 49 条指令，用于支持 CPU 的特性。更多详细信息，请参见[第 26.0 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

表 3-1：器件存储容量和地址

器件	程序存储空间（字）	程序存储器的最后一个地址	高耐用性闪存地址范围(1)
PIC16(L)F1574/8	4,096	0FFFh	0F80h-0FFFh
PIC16(L)F1575/9	8,192	1FFFh	1F80h-1FFFh

注 1：高耐用性闪存应用于范围内每个地址的低字节。

3.1 程序存储器构成

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

3.2 高耐用性闪存

该器件具有 128 字节的高耐用性闪存程序存储器（Program Flash Memory, PFM）部分，用于代替数据 EEPROM。该区域特别适合用于预期在最终产品使用寿命内经常更新的非易失性数据存储。关于向 PFM 写入数据的更多信息，请参见[第 10.2 节 “闪存程序存储器概述”](#)。关于使用 SFR 寄存器来读取存储在 PFM 中的字节数据的更多信息，请参见[第 3.3.2 节 “特殊功能寄存器”](#)。

PIC16(L)F1574/5/8/9

图 3-1： PIC16(L)F1574/8 的程序存储器映射和堆栈

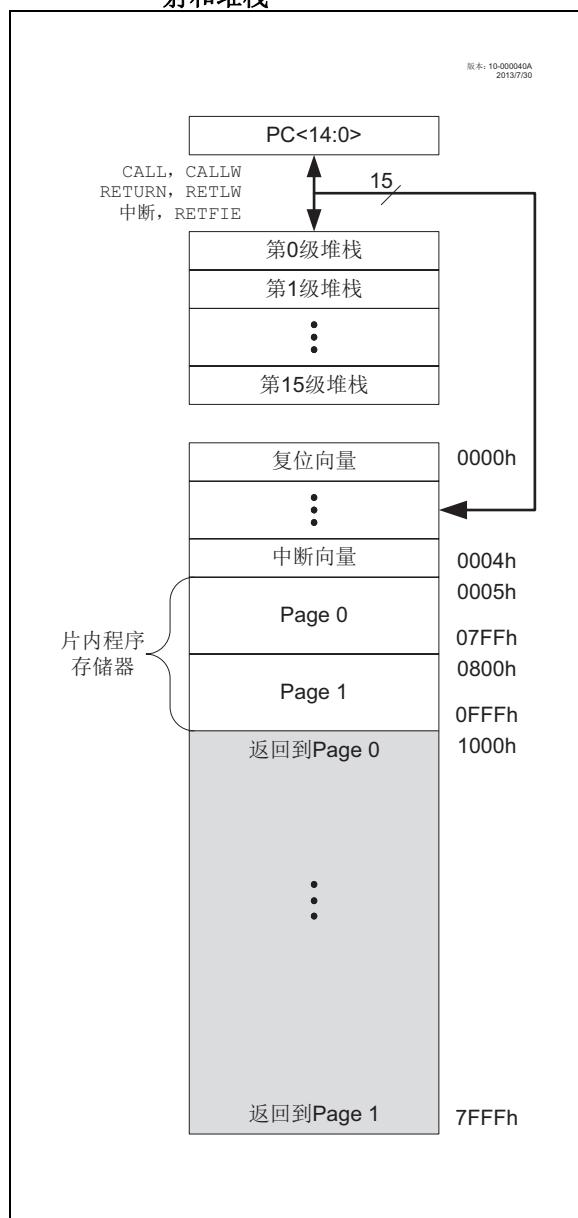
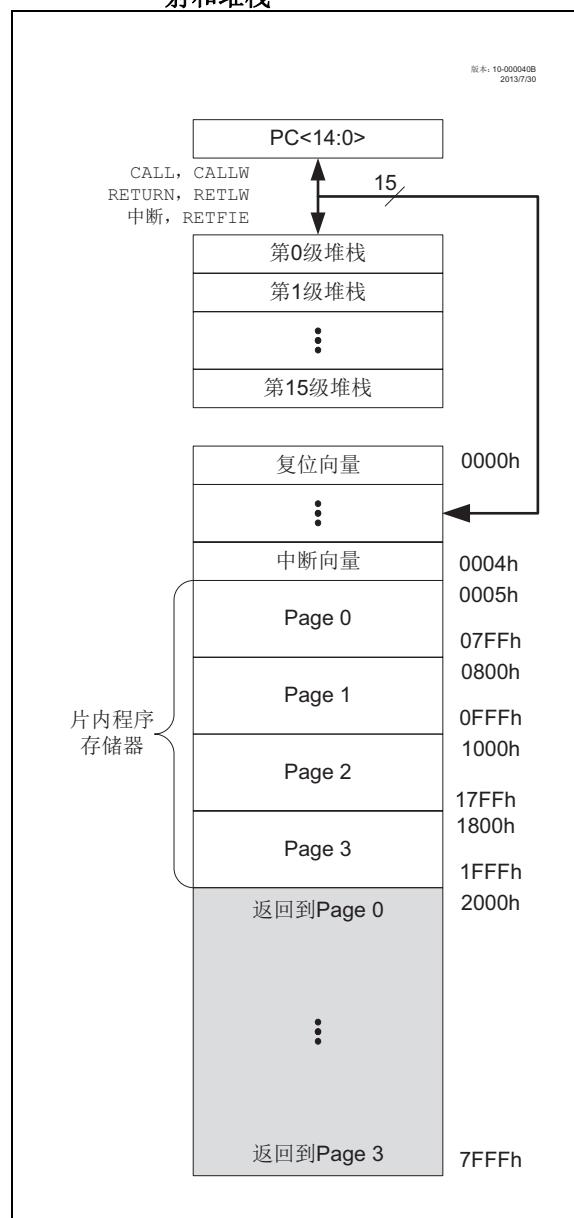


图 3-2： PIC16(L)F1575/9 的程序存储器映射和堆栈



3.2.1 将程序存储器作为数据进行读取

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

3.2.1.1 RETLW 指令

RETLW 指令可用于访问常量表。创建这种表的推荐方法如例 3-1 所示。

例 3-1： RETLW 指令

```
constants
    BRW           ;Add Index in W to
                  ;program counter to
                  ;select data
    RETLW DATA0   ;Index0 data
    RETLW DATA1   ;Index1 data
    RETLW DATA2
    RETLW DATA3

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

通过 BRW 指令，可以非常简单地实现这种表。如果代码必须保持对于前几代单片机的可移植性，则 BRW 指令不可用，此时代码必须使用先前的表读方法。

3.2.1.2 通过 FSR 间接读取

通过将 FSRxH 寄存器的 bit 7 置 1，并读取匹配的 INDFx 寄存器，可以将程序存储器作为数据进行访问。MOVIW 指令会将所寻址字的低 8 位放入 W 寄存器。对程序存储器的写操作不能通过 INDF 寄存器执行。对于通过 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 说明了如何通过 FSR 访问程序存储器。如果某个标号指向程序存储器中的存储单元，HIGH 操作符会将 bit<7> 置 1。

例 3-2： 通过 FSR 访问程序存储器

```
constants
    DW      DATA0          ;First constant
    DW      DATA1          ;Second constant
    DW      DATA2
    DW      DATA3

my_function
    ;... LOTS OF CODE...
    MOVLW   DATA_INDEX
    ADDLW   LOW constants
    MOVWF   FSR1L
    MOVLW   HIGH constants;MSb is set
                           automatically
    MOVWF   FSR1H
    BTFSC  STATUS,C       ;carry from ADDLW?
    INCF   FSR1H,f         ;yes
    MOVIW   0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

3.3 数据存储器构成

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

- 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。未实现的存储区将读为 0。所有数据存储器可以直接访问（通过使用文件寄存器的指令），也可以通过两个文件选择寄存器（FSR）间接访问。更多信息，请参见第 3.6 节“[间接寻址](#)”。

数据存储器使用一个 12 位地址。地址的高 5 位用于定义存储区地址，低 7 位用于选择该存储区中的寄存器/RAM。

3.3.1 内核寄存器

内核寄存器包含会直接影响基本操作的寄存器。内核寄存器占用每个数据存储区的前 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 3-14。

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

3.3.1.1 STATUS 寄存器

STATUS 寄存器如[寄存器 3-1](#) 所示，包括：

- ALU 的算术运算状态
- 复位状态

与任何其他寄存器一样，STATUS 寄存器可作为任何指令的目标寄存器。如果一条影响 Z、DC 或 C 位的指令将 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。此外，TO 和 PD 位均为不可写位。因此，当执行一条将 STATUS 寄存器作为目标寄存器的指令时，运行结果可能会与预想的不同。

寄存器 3-1： STATUS：状态寄存器

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	TO	PD	Z	DC ⁽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 TO：超时位

1 = 在上电或执行 CLRWDT 指令或 SLEEP 指令后
0 = 发生了 WDT 超时

bit 3 PD：掉电位

1 = 在上电或执行 CLRWDT 指令后
0 = 执行 SLEEP 指令

bit 2 Z：全零标志位

1 = 算术运算或逻辑运算的结果为零
0 = 算术运算或逻辑运算的结果不为零

bit 1 DC：半进位 / 半借位位（ADDWF、ADDLW、SUBLW 和 SUBWF 指令）⁽¹⁾

1 = 结果的第 4 个低位发生了进位
0 = 结果的第 4 个低位未发生进位

bit 0 C：进位 / 借位位⁽¹⁾（ADDWF、ADDLW、SUBLW 和 SUBWF 指令）⁽¹⁾

1 = 结果的最高有效位发生了进位
0 = 结果的最高有效位未发生进位

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

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 26.0 节 “指令集汇总”**。

注 1：在减法运算中，C 和 DC 位分别作为借位位和半借位位。

3.3.2 特殊功能寄存器

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

3.3.3 通用 RAM

每个数据存储区中有最大 80 字节的 GPR。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）。

3.3.3.1 线性访问 GPR

通用 RAM 可以通过 FSR 以非分区方式访问。这可以简化对大存储器结构的访问。更多信息，请参见[第 3.6.2 节“线性数据存储器”](#)。

3.3.4 公共 RAM

有 16 字节的公共 RAM 可以从所有存储区中进行访问。

3.3.5 器件存储器映射

[表 3-3 至表 3-13](#) 给出了存储器映射。

图 3-3：存储区构成

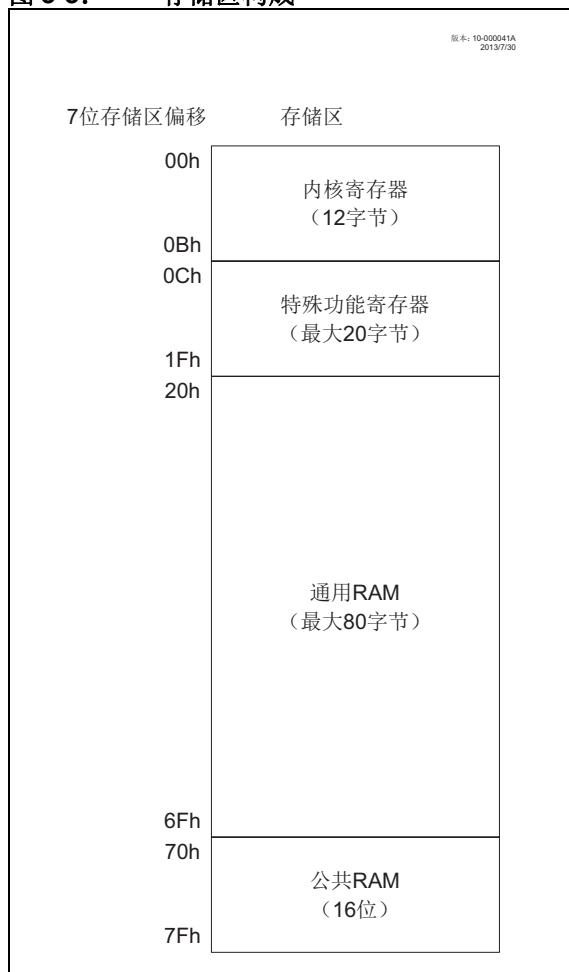


表 3-3: PIC16(L)F1574 存储器映射, BANK 0-7

	BANK0	BANK1	BANK2	BANK3	BANK4	BANK5	BANK6	BANK7
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)	280h
00Bh	PORTA	08Bh	TRISA	10Bh	LATA	18Bh	ANSELA	28Bh
00Ch	—	08Ch	—	10Dh	—	18Dh	—	28Ch
00Dh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSEL0	20Dh
00Eh	—	08Fh	—	10Fh	—	18Fh	WPUC	—
00Fh	—	—	—	110h	—	190h	—	20Eh
010h	—	—	—	—	—	—	—	20Fh
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON ⁽¹⁾	217h
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRGL	21Bh
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch
01Dh	—	09Dh	ADC0N0	11Dh	—	19Dh	RCSTA	21Dh
01Eh	—	09Eh	ADC0N1	11Eh	—	19Eh	TXSTA	21Eh
01Fh	—	09Fh	ADC0N2	11Fh	—	19Fh	BAUDCON	21Fh
020h	通用寄存器 80 字节	0A0h	通用寄存器 80 字节	120h	通用寄存器 80 字节	1A0h	通用寄存器 80 字节	2A0h
06Fh	公共 RAM	0EFh	快速操作存储区 70h – 7Fh	16Fh	快速操作存储区 70h – 7Fh	1EFh	快速操作存储区 70h – 7Fh	2EFh
070h		0F0h		170h		1F0h		2F0h
07Fh		0FFh		17Fh		1FFh		2FFh
								32Fh
								330h
								36Fh
								370h
								37Fh
								3FFh
								3A0h
								3E5h
								3F0h
								3FFh

图注: 灰色 = 未实现的数据存储单元, 读为 0。

注 1: PIC16LF1574 上未实现。

表 3-4: PIC16(L)F1575 存储器映射, BANK 0-7

BANK0	BANK1	BANK2	BANK3	BANK4	BANK5	BANK6	BANK7
000h 内核寄存器 (表 3-2)	080h 内核寄存器 (表 3-2)	100h 内核寄存器 (表 3-2)	180h 内核寄存器 (表 3-2)	200h 内核寄存器 (表 3-2)	280h 内核寄存器 (表 3-2)	300h 内核寄存器 (表 3-2)	380h 内核寄存器 (表 3-2)
00Bh PORTA	08Bh TRISA	10Bh LATA	18Bh ANSELA	20Bh WPUA	28Bh ODCONA	30Bh SLRCONA	38Bh INLVLA
00Ch —	08Ch —	10Dh —	18Dh —	20Dh —	28Dh —	30Dh —	38Dh —
00Dh PORTC	08Eh TRISC	10Eh LATC	18Eh ANSEL	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 —	311h —	391h IOCAP
012h PIR2	092h PIE2	112h CM1CON1	192h PMADRH	212h —	292h —	312h —	392h IOCAN
013h PIR3	093h PIE3	113h CM2CON0	193h PMDATL	213h —	293h —	313h —	393h IOCAF
014h —	094h —	114h CM2CON1	194h PMDATH	214h —	294h —	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 WDTCON	117h FVRCON	197h VREGCON ⁽¹⁾	217h —	297h —	317h —	397h IOCCP
018h T1CON	098h OSCTUNE	118h DACCON0	198h —	218h —	298h —	318h —	398h IOCCN
019h T1GCON	099h OSCCON	119h DACCON1	199h RCREG	219h —	299h —	319h —	399h IOCCF
01Ah TMR2	09Ah OSCSTAT	11Ah —	19Ah TXREG	21Ah —	29Ah —	31Ah —	39Ah —
01Bh PR2	09Bh ADRESL	11Bh —	19Bh SPBRGL	21Bh —	29Bh —	31Bh —	39Bh —
01Ch T2CON	09Ch ADRESH	11Ch —	19Ch SPBRGH	21Ch —	29Ch —	31Ch —	39Ch —
01Dh —	09Dh ADCON0	11Dh —	19Dh RCSTA	21Dh —	29Dh —	31Dh —	39Dh —
01Eh —	09Eh ADCON1	11Eh —	19Eh TXSTA	21Eh —	29Eh —	31Eh —	39Eh —
01Fh —	09Fh ADCON2	11Fh —	19Fh BAUDCON	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 通用寄存器 80 字节	220h 通用寄存器 80 字节	2A0h 通用寄存器 80 字节	320h 通用寄存器 80 字节	3A0h 通用寄存器 80 字节
06Fh 070h 公共 RAM	0EFh 0FOh 0FFh 快速操作存储区 70h – 7Fh	16Fh 170h 17Fh 快速操作存储区 70h – 7Fh	1EFh 1FOh 1FFh 快速操作存储区 70h – 7Fh	26Fh 270h 27Fh 快速操作存储区 70h – 7Fh	2EFh 2FOh 2FFh 快速操作存储区 70h – 7Fh	36Fh 370h 37Fh 快速操作存储区 70h – 7Fh	3EFh 3FOh 3FFh 快速操作存储区 70h – 7Fh

图注: █ = 未实现的数据存储单元, 读为 0。

注 1: PIC16LF1575 上未实现。

表 3-5: PIC16(L)F1578 存储器映射, BANK 0-7

	BANK0	BANK1	BANK2	BANK3	BANK4	BANK5	BANK6	BANK7
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
00Bh	PORTA	08Bh	TRISA	10Bh	LATA	18Bh	ANSELA	20Bh
00Ch	PORTB	08Ch	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh
00Dh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSEL0	20Eh
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh
010h	—	090h	—	110h	—	190h	—	210h
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	PMADRL	211h
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	PMADRH	212h
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	PMDATL	213h
014h	—	094h	—	114h	CM2CON1	194h	PMDATH	214h
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	VREGCON ⁽¹⁾	217h
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRGL	21Bh
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch
01Dh	—	09Dh	ADC0N0	11Dh	—	19Dh	RCSTA	21Dh
01Eh	—	09Eh	ADC0N1	11Eh	—	19Eh	TXSTA	21Eh
01Fh	—	09Fh	ADC0N2	11Fh	—	19Fh	BAUDCON	21Fh
020h	通用寄存器 80 字节	0A0h	通用寄存器 80 字节	120h	通用寄存器 80 字节	1A0h	通用寄存器 80 字节	2A0h
06Fh	公共 RAM	0EFh	快速操作存储区 70h – 7Fh	16Fh	快速操作存储区 70h – 7Fh	1EFh	快速操作存储区 70h – 7Fh	2EFh
070h		0F0h		170h		1F0h		2F0h
07Fh		0FFh		17Fh		1FFh		2FFh
								37Fh
							32Fh	未实现 读为 0
							330h	
							36Fh	未实现 读为 0
							370h	快速操作存储区 70h – 7Fh
							3FFh	快速操作存储区 70h – 7Fh
							3A0h	
							3EKh	未实现 读为 0
							3F0h	快速操作存储区 70h – 7Fh

图注: 灰色 = 未实现的数据存储单元, 读为 0。

注 1: PIC16LF1578 上未实现。

表 3-6: PIC16(L)F1579 存储器映射, BANK 0-7

BANK0	BANK1	BANK2	BANK3	BANK4	BANK5	BANK6	BANK7
000h 内核寄存器 (表 3-2)	080h 内核寄存器 (表 3-2)	100h 内核寄存器 (表 3-2)	180h 内核寄存器 (表 3-2)	200h 内核寄存器 (表 3-2)	280h 内核寄存器 (表 3-2)	300h 内核寄存器 (表 3-2)	380h 内核寄存器 (表 3-2)
00Bh PORTA	08Bh TRISA	10Bh LATA	18Bh ANSELA	20Bh WPUA	28Bh ODCONA	30Bh SLRCONA	38Bh INLVLA
00Ch PORTB	08Ch TRISB	10Dh LATB	18Dh ANSELB	20Dh WPUB	28Ch ODCONB	30Dh SLRCONB	38Ch INLVLB
00Dh PORTC	08Eh TRISC	10Eh LATC	18Eh ANSELC	20Eh WPUC	28Eh ODCONC	30Eh SLRCONC	38Dh INLVLC
00Fh —	08Fh —	10Fh —	18Fh —	20Fh —	28Fh —	30Fh —	38Eh —
010h —	090h PIR1	110h —	190h —	210h —	290h —	310h —	390h —
011h PIR2	091h PIE1	111h CM1CON0	191h PMADRL	211h —	291h —	311h —	391h IOCAP
012h PIR3	092h PIE2	112h CM1CON1	192h PMADRH	212h —	292h —	312h —	392h IOCAN
013h —	093h PIE3	113h CM2CON0	193h PMDATL	213h —	293h —	313h —	393h IOCAF
014h —	094h —	114h CM2CON1	194h PMDATH	214h —	294h —	314h —	394h IOCBP
015h TMR0	095h OPTION_REG	115h CMOUT	195h PMCON1	215h —	295h —	315h —	395h IOCBN
016h TMR1L	096h PCON	116h BORCON	196h PMCON2	216h —	296h —	316h —	396h IOCBF
017h TMR1H	097h WDTCON	117h FVRCON	197h VREGCON ⁽¹⁾	217h —	297h —	317h —	397h IOCCP
018h T1CON	098h OSCTUNE	118h DACCON0	198h —	218h —	298h —	318h —	398h IOCCN
019h T1GCON	099h OSCCON	119h DACCON1	199h RCREG	219h —	299h —	319h —	399h IOCCF
01Ah TMR2	09Ah OSCSTAT	11Ah —	19Ah TXREG	21Ah —	29Ah —	31Ah —	39Ah —
01Bh PR2	09Bh ADRESL	11Bh —	19Bh SPBRGL	21Bh —	29Bh —	31Bh —	39Bh —
01Ch T2CON	09Ch ADRESH	11Ch —	19Ch SPBRGH	21Ch —	29Ch —	31Ch —	39Ch —
01Dh —	09Dh ADCON0	11Dh —	19Dh RCSTA	21Dh —	29Dh —	31Dh —	39Dh —
01Eh —	09Eh ADCON1	11Eh —	19Eh TXSTA	21Eh —	29Eh —	31Eh —	39Eh —
01Fh —	09Fh ADCON2	11Fh —	19Fh BAUDCON	21Fh —	29Fh —	31Fh —	39Fh —
020h —	0A0h 通用寄存器 80 字节	120h 通用寄存器 80 字节	1A0h 通用寄存器 80 字节	220h 通用寄存器 80 字节	2A0h 通用寄存器 80 字节	320h 通用寄存器 80 字节	3A0h 通用寄存器 80 字节
06Fh 070h 公共 RAM	0EFh 0FOh 快速操作存储区 70h – 7Fh	16Fh 170h 快速操作存储区 70h – 7Fh	1EFh 1F0h 快速操作存储区 70h – 7Fh	26Fh 270h 快速操作存储区 70h – 7Fh	2EFh 2F0h 快速操作存储区 70h – 7Fh	36Fh 370h 快速操作存储区 70h – 7Fh	3EFh 3F0h 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。

注 1: PIC16LF1579 上未实现。

表 3-7: PIC16(L)F1574/8 存储器映射, BANK 8-15

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15			
400h	内核寄存器 (表 3-2)	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	—	791h	—		
412h	—	492h	—	512h	—	592h	—	612h	—	692h	CWG1DBF	712h	—	792h	—		
413h	—	493h	—	513h	—	593h	—	613h	—	693h	CWG1CON0	713h	—	793h	—		
414h	—	494h	—	514h	—	594h	—	614h	—	694h	CWG1CON1	714h	—	794h	—		
415h	—	495h	—	515h	—	595h	—	615h	—	695h	CWG1CON2	715h	—	795h	—		
416h	—	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—		
417h	—	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—		
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—		
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—		
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—		
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—		
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—		
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—		
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—		
41Fh	—	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		6EFh	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		6F0h	快速操作存储区 70h – 7Fh	
47Fh	快速操作存储区 70h – 7Fh		4FFh	快速操作存储区 70h – 7Fh		57Fh	快速操作存储区 70h – 7Fh		5FFh	快速操作存储区 70h – 7Fh		67Fh	快速操作存储区 70h – 7Fh		6FFh	快速操作存储区 70h – 7Fh	
图注: = 未实现的数据存储单元, 读为 0。																	

表 3-8: PIC16(L)F1575/9 存储器映射, BANK 8-15

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15			
400h	内核寄存器 (表 3-2)	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	—	791h	—		
412h	—	492h	—	512h	—	592h	—	612h	—	692h	CWG1DBF	712h	—	792h	—		
413h	—	493h	—	513h	—	593h	—	613h	—	693h	CWG1CON0	713h	—	793h	—		
414h	—	494h	—	514h	—	594h	—	614h	—	694h	CWG1CON1	714h	—	794h	—		
415h	—	495h	—	515h	—	595h	—	615h	—	695h	CWG1CON2	715h	—	795h	—		
416h	—	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—		
417h	—	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—		
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—		
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—		
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—		
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—		
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—		
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—		
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—		
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—		
420h	通用寄存器 80 字节	4A0h	通用寄存器 80 字节	520h	通用寄存器 80 字节	5A0h	通用寄存器 80 字节	620h	通用寄存器 32 字节	6A0h	未实现 读为 0	720h	未实现 读为 0	7A0h	未实现 读为 0		
46Fh	快速操作存储区 70h – 7Fh		4EFh	快速操作存储区 70h – 7Fh		56Fh	快速操作存储区 70h – 7Fh		5EFh	快速操作存储区 70h – 7Fh		66Fh	快速操作存储区 70h – 7Fh		6EFh	快速操作存储区 70h – 7Fh	
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		6F0h	快速操作存储区 70h – 7Fh	
47Fh	快速操作存储区 70h – 7Fh		4FFh	快速操作存储区 70h – 7Fh		57Fh	快速操作存储区 70h – 7Fh		5FFh	快速操作存储区 70h – 7Fh		67Fh	快速操作存储区 70h – 7Fh		6FFh	快速操作存储区 70h – 7Fh	

图注: = 未实现的数据存储单元, 读为 0。

表 3-9: PIC16(L)F1574/5/8/9 存储器映射, BANK 16-23

BANK16	BANK17	BANK18	BANK19	BANK20	BANK21	BANK22	BANK23					
800h 80Bh 80Ch 80Dh 80Eh 80Fh 810h 811h 812h 813h 814h 815h 816h 817h 818h 819h 81Ah 81Bh 81Ch 81Dh 81Eh 81Fh 820h 86Fh 870h 87Fh	内核寄存器 (表 3-2) — — — — — — — — — — — — — — — — — — — 未实现 读为 0 快速操作存储区 70h – 7Fh	880h 88Bh 88Ch 88Dh 88Eh 88Fh 890h 891h 892h 893h 894h 895h 896h 897h 898h 899h 89Ah 89Bh 89Ch 89Dh 89Eh 89Fh 8A0h 8EFh 8F0h 8FFh	内核寄存器 (表 3-2) — — — — — — — — — — — — — — — — — — — 未实现 读为 0 快速操作存储区 70h – 7Fh	900h 90Bh 90Ch 90Dh 90Eh 90Fh 910h 911h 912h 913h 914h 915h 916h 917h 918h 919h 91Ah 91Bh 91Ch 91Dh 91Eh 91Fh 920h 96Fh 970h 97Fh	内核寄存器 (表 3-2) — — — — — — — — — — — — — — — — — — — 未实现 读为 0 快速操作存储区 70h – 7Fh	980h 98Bh 98Ch 98Dh 98Eh 98Fh 990h 991h 992h 993h 994h 995h 996h 997h 998h 999h 99Ah 99Bh 99Ch 99Dh 99Eh 99Fh 9A0h 9EFh 9F0h 9FFh	A00h A0Bh A0Ch A0Dh A0Eh A0Fh A10h A11h A12h A13h A14h A15h A16h A17h A18h A19h A1Ah A1Bh A1Ch A1Dh A1Eh A1Fh A20h A6Fh A70h A7Fh AFFh	A80h A8Bh A8Ch A8Dh A8Eh A8Fh A90h A91h A92h A93h A94h A95h A96h A97h A98h A99h A9Ah A9Bh A9Ch A9Dh A9Eh A9Fh AA0h AEFh AF0h AFFh	B00h B0Bh B0Ch B0Dh B0Eh B0Fh B10h B11h B12h B13h B14h B15h B16h B17h B18h B19h B1Ah B1Bh B1Ch B1Dh B1Eh B1Fh B20h B6Fh B70h B7Fh	内核寄存器 (表 3-2) — — — — — — — — — — — — — — — — — — — 未实现 读为 0 快速操作存储区 70h – 7Fh	B80h B8Bh B8Ch B8Dh B8Eh B8Fh B90h B91h B92h B93h B94h B95h B96h B97h B98h B99h B9Ah B9Bh B9Ch B9Dh B9Eh B9Fh BA0h BEFh BF0h BFFh	内核寄存器 (表 3-2) — — — — — — — — — — — — — — — — — — — 未实现 读为 0 快速操作存储区 70h – 7Fh

图注: = 未实现的数据存储单元, 读为 0。

表 3-10: PIC16(L)F1574/5/8/9 存储器映射, BANK 24-31

BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31	
C00h	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	—	E0Bh	—	E8Bh	—	F0Bh	—	F8Bh	内核寄存器 (表 3-2)
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—	F8Ch	—
C0Dh	—	C8Dh	—	D0Dh	—	见表 3-11	—	见表 3-12	—	见表 3-13	—	F0Dh	—	—	—
C0Eh	—	C8Eh	—	D0Eh	—							F0Eh	—	—	—
C0Fh	—	C8Fh	—	D0Fh	—							F0Fh	—	—	—
C10h	—	C90h	—	D10h	—							F10h	—	—	—
C11h	—	C91h	—	D11h	—							F11h	—	—	—
C12h	—	C92h	—	D12h	—							F12h	—	—	—
C13h	—	C93h	—	D13h	—							F13h	—	—	—
C14h	—	C94h	—	D14h	—							F14h	—	—	—
C15h	—	C95h	—	D15h	—							F15h	—	—	—
C16h	—	C96h	—	D16h	—							F16h	—	—	—
C17h	—	C97h	—	D17h	—							F17h	—	—	—
C18h	—	C98h	—	D18h	—							F18h	—	—	—
C19h	—	C99h	—	D19h	—							F19h	—	—	—
C1Ah	—	C9Ah	—	D1Ah	—							F1Ah	—	—	—
C1Bh	—	C9Bh	—	D1Bh	—							F1Bh	—	—	—
C1Ch	—	C9Ch	—	D1Ch	—							F1Ch	—	—	—
C1Dh	—	C9Dh	—	D1Dh	—							F1Dh	—	—	—
C1Eh	—	C9Eh	—	D1Eh	—							F1Eh	—	—	—
C1Fh	—	C9Fh	—	D1Fh	—							F1Fh	—	—	—
C20h	—	CA0h	—	D20h	—							F20h	—	—	—
C6Fh	未实现 读为 0	CEFh	未实现 读为 0	D6Fh	未实现 读为 0	DEFh	—	E6Fh	快速操作存储区 70h – 7Fh	EEFh	快速操作存储区 70h – 7Fh	F6Fh	未实现 读为 0	FEFh	快速操作存储区 70h – 7Fh
C70h	快速操作存储区 70h – 7Fh	CF0h	快速操作存储区 70h – 7Fh	D70h	快速操作存储区 70h – 7Fh	DF0h	—	E70h	快速操作存储区 70h – 7Fh	EF0h	快速操作存储区 70h – 7Fh	F70h	快速操作存储区 70h – 7Fh	FF0h	快速操作存储区 70h – 7Fh
CFFh	—	CFFh	—	D7Fh	—	DFFh	—	E7Fh	—	EFFh	—	F7Fh	—	FFFh	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 3-11: PIC16(L)F1574/5/8/9 存储器映射,
BANK 27

Bank 27	
D8Ch	—
D8Dh	—
D8Eh	PWMEN
D8Fh	PWMLD
D90h	PWMOUT
D91h	PWM1PHL
D92h	PWM1PHH
D93h	PWM1DCL
D94h	PWM1DCH
D95h	PWM1PRL
D96h	PWM1PRH
D97h	PWM1OFL
D98h	PWM1OFH
D99h	PWM1TMRL
D9Ah	PWM1TMRH
D9Bh	PWM1CON
D9Ch	PWM1INTE
D9Dh	PWM1INTF
D9Eh	PWM1CLKCON
D9Fh	PWM1LDCON
DA0h	PWM1OFCON
DA1h	PWM2PHL
DA2h	PWM2PHH
DA3h	PWM2DCL
DA4h	PWM2DCH
DA5h	PWM2PRL
DA6h	PWM2PRH
DA7h	PWM2OFL
DA8h	PWM2OFH
DA9h	PWM2TMRL
DAAh	PWM2TMRH
DABh	PWM2CON
DACh	PWM2INTE
DADh	PWM2INTF
DAEh	PWM2CLKCON
DAFh	PWM2LDCON
DB0h	PWM2OFCON
DB1h	PWM3PHL
DB2h	PWM3PHH
DB3h	PWM3DCL
DB4h	PWM3DCH
DB5h	PWM3PRL
DB6h	PWM3PRH
DB7h	PWM3OFL
DB8h	PWM3OFH
DB9h	PWM3TMRL
DBAh	PWM3TMRH
DBBh	PWM3CON
DBCh	PWM3INTE
DBDh	PWM3INTF
DBEh	PWM3CLKCON
DBFh	PWM3LDCON
DC0h	PWM3OFCON
DC1h	PWM4PHL
DC2h	PWM4PHH
DC3h	PWM4DCL
DC4h	PWM4DCH
DC5h	PWM4PRL
DC6h	PWM4PRH
DC7h	PWM4OFL
DC8h	PWM4OFH
DC9h	PWM4TMRL
DCAh	PWM4TMRH
DCBh	PWM4CON
DCCh	PWM4INTE
DCDh	PWM4INTF
DCEh	PWM4CLKCON
DCFh	PWM4LDCON
DD0h	PWM4OFCON
DD1h	
DEFh	—

图注: = 未实现的数据存储单元, 读为 0。

表 3-12: PIC16(L)F1574/5/8/9 存储器映射,
BANK 28-29

Bank 28		Bank 29	
E0Ch	—	E8Ch	—
E0Dh	—	E8Dh	—
E0Eh	—	E8Eh	—
E0Fh	PPSLOCK	E8Fh	—
E10h	INTPPS	E90h	RA0PPS
E11h	T0CKIPPS	E91h	RA1PPS
E12h	T1CKIPPS	E92h	RA2PPS
E13h	T1GPPS	E93h	—
E14h	CWG1PPS	E94h	RA4PPS
E15h	RXPPS	E95h	RA5PPS
E16h	CKPPS	E96h	—
E17h	ADCACTPPS	E97h	—
E18h	—	E98h	—
E19h	—	E99h	—
E1Ah	—	E9Ah	—
E1Bh	—	E9Bh	—
E1Ch	—	E9Ch	RB4PPS ⁽¹⁾
E1Dh	—	E9Dh	RB5PPS ⁽¹⁾
E1Eh	—	E9Eh	RB6PPS ⁽¹⁾
E1Fh	—	E9Fh	RB7PPS ⁽¹⁾
E20h	—	EA0h	RC0PPS
E21h	—	EA1h	RC1PPS
E22h	—	EA2h	RC2PPS
E23h	—	EA3h	RC3PPS
E24h	—	EA4h	RC4PPS
E25h	—	EA5h	RC5PPS
E26h	—	EA6h	RC6PPS ⁽¹⁾
E27h	—	EA7h	RC7PPS ⁽¹⁾
E28h	—	EA8h	—
E29h	—	EA9h	—
E2Ah	—	EAAh	—
E2Bh	—	EABh	—
E2Ch	—	EACh	—
E2Dh	—	EADh	—
E2Eh	—	EAEh	—
E2Fh	—	EAFh	—
E30h	—	EB0h	—
E31h	—	EB1h	—
E32h	—	EB2h	—
E33h	—	EB3h	—
E34h	—	EB4h	—
E35h	—	EB5h	—
E36h	—	EB6h	—
E37h	—	EB7h	—
E38h	—	EB8h	—
E39h	—	EB9h	—
E3Ah	—	EBAh	—
E3Bh	—	EBBh	—
E3Ch	—	EBCh	—
E3Dh	—	EBDh	—
E3Eh	—	EBEh	—
E3Fh	—	EBFh	—
E40h	—	EC0h	—
E6Fh	—	EEFh	—

图注: = 未实现的数据存储单元, 读为 0。

注 1: PIC16(L)F1574/5 上未实现。

PIC16(L)F1574/5/8/9

表 3-13: PIC16(L)F1574/5/8/9 存储器映射,
BANK 31

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

图注: = 未实现的数据存储单元, 读为 0。

3.3.6 内核功能寄存器汇总

表 3-14 列出了内核功能寄存器，可从任何存储区寻址这些寄存器。

表 3-14： 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/BOR 时的值	所有其他复位时的值
Bank 0-31											
x00h 或 x80h	INDF0	使用 FSR0H/FSR0L 的内容寻址该存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx	uuuu uuuu
x01h 或 x81h	INDF1	使用 FSR1H/FSR1L 的内容寻址该存储单元来寻址数据存储器（不是物理寄存器）								xxxx xxxx	uuuu uuuu
x02h 或 x82h	PCL	程序计数器（Program Counter, PC）的低字节								0000 0000	0000 0000
x03h 或 x83h	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu
x04h 或 x84h	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu
x05h 或 x85h	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000
x06h 或 x86h	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu
x07h 或 x87h	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000
x08h 或 x88h	BSR	—	—	—	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	TMROIE	INTE	IOCIE	TMROIF	INTF	IOCIF	0000 0000	0000 0000

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

表 3-15：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/BOR 时的值	所有其他复 位时的值					
Bank 0																
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--xx xxxx					
00Dh	PORTB ⁽¹⁾	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ---	xxxx ---					
00Eh	PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	xxxx xxxx					
00Fh	—	未实现							—	—	—					
010h	—	未实现							—	—	—					
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	0000 --00	0000 --00					
012h	PIR2	—	C2IF	C1IF	—	—	—	—	—	-00- ----	-00- ----					
013h	PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	0000 ----	0000 ----					
014h	—	—	—	—	—	—	—	—	—	—	—					
015h	TMR0	8 位 Timer0 计数的保持寄存器							xxxx xxxx	uuuu uuuu	uuuu uuuu					
016h	TMR1L	16 位 TMR1 计数低字节的保持寄存器							xxxx xxxx	uuuu uuuu	uuuu uuuu					
017h	TMR1H	16 位 TMR1 计数高字节的保持寄存器							xxxx xxxx	uuuu uuuu	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	0000 0000					
01Bh	PR2	Timer2 周期寄存器							1111 1111	1111 1111	1111 1111					
01Ch	T2CON	—	T2OUTPS<3:0>			TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000	-000 0000					
01Dh	—	未实现							—	—	—					
01Eh	—	未实现							—	—	—					
01Fh	—	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 PIC16(L)F1578/9。

2：仅限 PIC16F1574/5/8/9。

3：未实现，读为 1。

表 3-15：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/BOR 时的值	所有其他复 位时的值
Bank 1											
08Ch	TRISA	—	—	TRISA5	TRISA4	— ⁽³⁾	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
08Dh	TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
08Eh	TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
08Fh	—	未实现								—	—
090h	—	未实现								—	—
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	0000 --00	0000 --00
092h	PIE2	—	C2IE	C1IE	—	—	—	—	—	-00- ----	-00- ----
093h	PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	0000 ----	0000 ----
094h	—									—	—
095h	OPTION_REG	WPUEEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111	1111 1111
096h	PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	00-1 11qq	qq-q qquu
097h	WDTCON	—	—	WDTPS<4:0>				SWDTEN	—	--01 0110	--01 0110
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	OSTS	HFIOPR	HFIOL	MFIOPR	LFIOPR	HFIOPS	-0q0 0q00	-qqq qqqq
09Bh	ADRESL	ADC 结果寄存器的低字节								xxxx xxxx	uuuu uuuu
09Ch	ADRESH	ADC 结果寄存器的高字节								xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>				GO/DONE	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 PIC16(L)F1578/9。

2：仅限 PIC16F1574/5/8/9。

3：未实现，读为1。

表 3-15：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/BOR 时的值	所有其他复 位时的值		
Bank 2													
10Ch	LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	--xx -xxx	--uu -uuu		
10Dh	LATB ⁽¹⁾	LATB7	LATB6	LATB5	LATB4	—	—	—	—	xxxxx ---	xxxxx ---		
10Eh	LATC	LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxxx xxxx	xxxxx xxxx		
10Fh	—	未实现							—	—	—		
110h	—	未实现							—	—	—		
111h	CM1CON0	C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC	00-0 -100	00-0 -100		
112h	CM1CON1	C1INTP	C1INTN	C1PCH<1:0>		—	C1NCH<2:0>			0000 -000	0000 -000		
113h	CM2CON0	C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC	00-0 -100	00-0 -100		
114h	CM2CON1	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	DACCON0	DACEN	—	DACOE	—	DACPSS<1:0>		—	—	0-0- 00--	0-0- 00--		
119h	DACCON1	—	—	—	DACR<4:0>					---0 0000	---0 0000		
11Ah 至 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 PIC16(L)F1578/9。

2：仅限 PIC16F1574/5/8/9。

3：未实现，读为1。

表 3-15：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/BOR 时的值	所有其他复 位时的值
Bank 3											
18Ch	ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	---1 -111	---1 -111
18Dh	ANSELB ⁽¹⁾	—	—	ANSB5	ANSB4	—	—	—	—	--11 ----	--11 ----
18Eh	ANSEL C	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	11-- 1111	11-- 1111
18Fh	—	未实现							—	—	—
190h	—	未实现							—	—	—
191h	PMADRL	闪存程序存储器地址寄存器的低字节							0000 0000	0000 0000	0000 0000
192h	PMADRH	— ⁽³⁾	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	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	— ⁽³⁾	CFG S	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	0000 0000
197h	VREGCON ⁽²⁾	—	—	—	—	—	—	VREGPM	保留	---- --01	---- --01
198h	—	未实现							—	—	—
199h	RCREG	USART 接收数据寄存器							0000 0000	0000 0000	0000 0000
19Ah	TXREG	USART 发送数据寄存器							0000 0000	0000 0000	0000 0000
19Bh	SPBRGL	波特率发生器数据寄存器的低字节							0000 0000	0000 0000	0000 0000
19Ch	SPBRGH	波特率发生器数据寄存器的高字节							0000 0000	0000 0000	0000 0000
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
19Fh	BAUDCON	ABDOVF	RIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00

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

注 1：仅限 PIC16(L)F1578/9。

2：仅限 PIC16F1574/5/8/9。

3：未实现，读为1。

表 3-15：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/BOR 时的值	所有其他复 位时的值
Bank 4											
20Ch	WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	--11 1111	--11 1111
20Dh	WPUB ⁽¹⁾	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----
20Eh	WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	1111 1111	1111 1111
20Fh 至 21Fh	—	未实现							—	—	—
Bank 5											
28Ch	ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	--00 -000	--00 -000
28Dh	ODCONB ⁽¹⁾	ODB7	ODB6	ODB5	ODB4	—	—	—	—	0000 ----	0000 ----
28Eh	ODCONC	ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	0000 0000	0000 0000
28Fh 至 29Fh	—	未实现							—	—	—
Bank 6											
30Ch	SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	--11 -111	--11 -111
30Dh	SLRCONB ⁽¹⁾	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	1111 ----	1111 ----
30Eh	SLRCONC	SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	1111 1111	1111 1111
30Fh 至 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 PIC16(L)F1578/9。

2：仅限 PIC16F1574/5/8/9。

3：未实现，读为1。

表 3-15：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/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	INLVLB ⁽¹⁾	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	1111 ----	1111 ----
38Eh	INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	1111 1111	1111 1111
38Fh 至 390h	—	未实现								—	—
391h	IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	--00 0000	--00 0000
392h	IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	--00 0000	--00 0000
393h	IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	--00 0000	--00 0000
394h	IOCBP ⁽¹⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	0000 ----	--00 ----
395h	IOCBN ⁽¹⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	0000 ----	--00 ----
396h	IOCBF ⁽¹⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	0000 ----	--00 ----
397h	IOCCP	IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	0000 0000	0000 0000
398h	IOCCN	IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	0000 0000	0000 0000
399h	IOCCF	IOCCF7 ⁽¹⁾	IOCCF6 ⁽¹⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	0000 0000	0000 0000
39Ah 至 39Fh	—	未实现								—	—
Bank 8											
40Ch 至 41Fh	—	未实现								—	—
Bank 9											
48Ch 至 49Fh	—	未实现								—	—

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

- 注 1：仅限 PIC16(L)F1578/9。
 2：仅限 PIC16F1574/5/8/9。
 3：未实现，读为1。

表 3-15：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/BOR 时的值	所有其他复 位时的值
Bank 10											
50Ch 至 51Fh	—	未实现								—	—
Bank 11											
58Ch 至 59Fh	—	未实现								—	—
Bank 12											
60Ch 至 61Fh	—	未实现								—	—
Bank 13											
68Ch 至 690h	—	未实现								—	—
691h	CWG1DBR	—	—	CWG1DBR<5:0>					--00 0000	--00 0000	
692h	CWG1DBF	—	—	CWG1DBF<5:0>					--xx xxxx	--xx xxxx	
693h	CWG1CON0	G1EN	—	—	G1POLB	G1POLA	—	—	G1CS0	0--0 0--0	0--0 0--0
694h	CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	G1IS<2:0>			0000 -000	0000 -000
695h	CWG1CON2	G1ASE	G1ARSEN	—	—	G1ASDSC2	G1ASDSC1	G1ASDSPPS	—	00-- 000-	00-- 000-
696h 至 69Fh	—	未实现								—	—
Bank 14-26											
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 PIC16(L)F1578/9。

2：仅限 PIC16F1574/5/8/9。

3：未实现，读为1。

表 3-15: 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/BOR 时的值	所有其他复 位时的值				
Bank 27															
D8Ch	—	未实现								—	—				
D8Dh	—	未实现								—	—				
D8Eh	PWMEN	—	—	—	—	PWM4EN_A	PWM3EN_A	PWM2EN_A	PWM1EN_A	---- 0000	---- 0000				
D8Fh	PWMLD	—	—	—	—	PWM4LDA_A	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A	---- 0000	---- 0000				
D90h	PWMOUT	—	—	—	—	PWM4OUT_A	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A	---- 0000	---- 0000				
D91h	PWM1PHL	PH<7:0>								xxxx xxxx	uuuu uuuu				
D92h	PWM1PHH	PH<15:8>								xxxx xxxx	uuuu uuuu				
D93h	PWM1DCL	DC<7:0>								xxxx xxxx	uuuu uuuu				
D94h	PWM1DCH	DC<15:8>								xxxx xxxx	uuuu uuuu				
D95h	PWM1PRL	PR<7:0>								xxxx xxxx	uuuu uuuu				
D96h	PWM1PRH	PR<15:8>								xxxx xxxx	uuuu uuuu				
D97h	PWM1OFL	OF<7:0>								xxxx xxxx	uuuu uuuu				
D98h	PWM1OFH	OF<15:8>								xxxx xxxx	uuuu uuuu				
D99h	PWM1TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu				
D9Ah	PWM1TMRH	TMR<15:8>								xxxx xxxx	uuuu uuuu				
D9Bh	PWM1CON	EN	—	OUT	POL	MODE<1:0>		—	—	0-00 00--	0-00 00--				
D9Ch	PWM1INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000				
D9Dh	PWM1INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000				
D9Eh	PWM1CLKCON	—	PS<2:0>		—	—	CS<1:0>		-000 -000	-000 --00					
D9Fh	PWM1LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		00-- -000	00-- --00				
DA0h	PWM1OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		-000 -000	-000 --00				
DA1h	PWM2PHL	PH<7:0>								xxxx xxxx	uuuu uuuu				
DA2h	PWM2PHH	PH<15:8>								xxxx xxxx	uuuu uuuu				
DA3h	PWM2DCL	DC<7:0>								xxxx xxxx	uuuu uuuu				
DA4h	PWM2DCH	DC<15:8>								xxxx xxxx	uuuu uuuu				
DA5h	PWM2PRL	PR<7:0>								xxxx xxxx	uuuu uuuu				
DA6h	PWM2PRH	PR<15:8>								xxxx xxxx	uuuu uuuu				
DA7h	PWM2OFL	OF<7:0>								xxxx xxxx	uuuu uuuu				
DA8h	PWM2OFH	OF<15:8>								xxxx xxxx	uuuu uuuu				
DA9h	PWM2TMRL	TMR<7:0>								xxxx xxxx	uuuu uuuu				
DAAh	PWM2TMRH	TMR<15:8>								xxxx xxxx	uuuu 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 PIC16(L)F1578/9。

2: 仅限 PIC16F1574/5/8/9。

3: 未实现, 读为 1。

表 3-15：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/BOR 时的值	所有其他复 位时的值		
Bank 27 (续)													
DABh	PWM2CON	EN	—	OUT	POL	MODE<1:0>	—	—	0-00 00--	0-00 00--			
DACh	PWM2INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000		
DADh	PWM2INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000		
DAEh	PWM2CLKCON	—	PS<2:0>			—	—	CS<1:0>	-000 -000	-000 --00			
DAFh	PWM2LDCON	LDA	LDT	—	—	—	—	LDS<1:0>	00-- -000	00-- --00			
DB0h	PWM2OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>	-000 -000	-000 --00			
DB1h	PWM3PHL	PH<7:0>								xxxxx xxxx	uuuuu uuuuu		
DB2h	PWM3PHH	PH<15:8>								xxxxx xxxx	uuuuu uuuuu		
DB3h	PWM3DCL	DC<7:0>								xxxxx xxxx	uuuuu uuuuu		
DB4h	PWM3DCH	DC<15:8>								xxxxx xxxx	uuuuu uuuuu		
DB5h	PWM3PRL	PR<7:0>								xxxxx xxxx	uuuuu uuuuu		
DB6h	PWM3PRH	PR<15:8>								xxxxx xxxx	uuuuu uuuuu		
DB7h	PWM3OFL	OF<7:0>								xxxxx xxxx	uuuuu uuuuu		
DB8h	PWM3OFH	OF<15:8>								xxxxx xxxx	uuuuu uuuuu		
DB9h	PWM3TMRL	TMR<7:0>								xxxxx xxxx	uuuuu uuuuu		
DBAh	PWM3TMRH	TMR<15:8>								xxxxx xxxx	uuuuu uuuuu		
DBBh	PWM3CON	EN	—	OUT	POL	MODE<1:0>	—	—	0-00 00--	0-00 00--			
DBCh	PWM3INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000		
DBDh	PWM3INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000		
DBEh	PWM3CLKCON	—	PS<2:0>			—	—	CS<1:0>	-000 -000	-000 --00			
DBFh	PWM3LDCON	LDA	LDT	—	—	—	—	LDS<1:0>	00-- -000	00-- --00			
DC0h	PWM3OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>	-000 -000	-000 --00			
DC1h	PWM4PHL	PH<7:0>								xxxxx xxxx	uuuuu uuuuu		
DC2h	PWM4PHH	PH<15:8>								xxxxx xxxx	uuuuu uuuuu		
DC3h	PWM4DCL	DC<7:0>								xxxxx xxxx	uuuuu uuuuu		
DC4h	PWM4DCH	DC<15:8>								xxxxx xxxx	uuuuu uuuuu		
DC5h	PWM4PRL	PR<7:0>								xxxxx xxxx	uuuuu uuuuu		
DC6h	PWM4PRH	PR<15:8>								xxxxx xxxx	uuuuu uuuuu		
DC7h	PWM4OFL	OF<7:0>								xxxxx xxxx	uuuuu uuuuu		
DC8h	PWM4OFH	OF<15:8>								xxxxx xxxx	uuuuu uuuuu		

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 PIC16(L)F1578/9。
 2: 仅限 PIC16F1574/5/8/9。
 3: 未实现, 读为 1。

表 3-15: 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/BOR 时的值	所有其他复 位时的值		
Bank 27 (续)													
DC9h	PWM4TMRL					TMR<7:0>				xxxx xxxx	uuuu uuuu		
DCAh	PWM4TMRH					TMR<15:8>				xxxx xxxx	uuuu uuuu		
DCBh	PWM4CON	EN	—	OUT	POL	MODE<1:0>	—	—	0000 00--	0000 00--			
DCCh	PWM4INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 000	---- 000		
DCDh	PWM4INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 000	---- 000		
DCEh	PWM4CLKCON	—	PS<2:0>			—	—	CS<1:0>		-000 -000	-000 --00		
DCFh	PWM4LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		00-- 000	00-- --00		
DD0h	PWM4OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		-000 -000	-000 --00		
DD1h 至 DEFh	—	未实现							—	—			
Bank 28													
E0Ch — E0Eh	—	未实现							—	—			
E0Fh	PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	---- ---0	---- ---0		
E10h	INTPPS	—	—	—	INTPPS<4:0>					--0 0010	--u uuuu		
E11h	T0CKIPPS	—	—	—	T0CKIPPS<4:0>					--0 0010	--u uuuu		
E12h	T1CKIPPS	—	—	—	T1CKIPPS<4:0>					--0 0101	--u uuuu		
E13h	T1GPPS	—	—	—	T1GPPS<4:0>					--0 0100	--u uuuu		
E14h	CWG1INPPS	—	—	—	CWGINPPS<4:0>					--0 0010	--u uuuu		
E15h	RXPPS	—	—	—	RXPPS<4:0>					--1 0101	--u uuuu		
E16h	CKPPS	—	—	—	CKPPS<4:0>					--1 0101	--u uuuu		
E17h	ADCACTPPS	—	—	—	ADCACTPPS<4:0>					--1 0101	--u uuuu		
E18h 至 E6Fh	—	未实现							—	—			

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

注 1: 仅限 PIC16(L)F1578/9。

2: 仅限 PIC16F1574/5/8/9。

3: 未实现, 读为 1。

表 3-15：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/BOR 时的值	所有其他复 位时的值
Bank 29											
E8Ch — E8Fh	—	未实现								—	—
E90h	RA0PPS	—	—	—		RA0PPS<4:0>			---0 0000	---u uuuu	
E91h	RA1PPS	—	—	—		RA1PPS<4:0>			---0 0000	---u uuuu	
E92h	RA2PPS	—	—	—		RA2PPS<4:0>			---0 0000	---u uuuu	
E93h	—	未实现								—	—
E94h	RA4PPS	—	—	—		RA4PPS<4:0>			---0 0000	---u uuuu	
E95h	RA5PPS	—	—	—		RA5PPS<4:0>			---0 0000	---u uuuu	
E96h — E9Bh	—	未实现								—	—
E9Ch	RB4PPS ⁽¹⁾	—	—	—		RB4PPS<4:0>			---0 0000	---u uuuu	
E9Dh	RB5PPS ⁽¹⁾	—	—	—		RB5PPS<4:0>			---0 0000	---u uuuu	
E9Eh	RB6PPS ⁽¹⁾	—	—	—		RB6PPS<4:0>			---0 0000	---u uuuu	
E9Fh	RB7PPS ⁽¹⁾	—	—	—		RB7PPS<4:0>			---0 0000	---u uuuu	
EA0h	RC0PPS	—	—	—		RC0PPS<4:0>			---0 0000	---u uuuu	
EA1h	RC1PPS	—	—	—		RC1PPS<4:0>			---0 0000	---u uuuu	
EA2h	RC2PPS	—	—	—		RC2PPS<4:0>			---0 0000	---u uuuu	
EA3h	RC3PPS	—	—	—		RC3PPS<4:0>			---0 0000	---u uuuu	
EA4h	RC4PPS	—	—	—		RC4PPS<4:0>			---0 0000	---u uuuu	
EA5h	RC5PPS	—	—	—		RC5PPS<4:0>			---0 0000	---u uuuu	
EA6h	RC6PPS ⁽¹⁾	—	—	—		RC6PPS<4:0>			---0 0000	---u uuuu	
EA7h	RC7PPS ⁽¹⁾	—	—	—		RC7PPS<4:0>			---0 0000	---u uuuu	
EA8h — EEFh	—	未实现								—	—
Bank 30											
F0Ch — F1Fh	—	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 PIC16(L)F1578/9。

2：仅限 PIC16F1574/5/8/9。

3：未实现，读为1。

表 3-15：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/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					
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 PIC16(L)F1578/9。

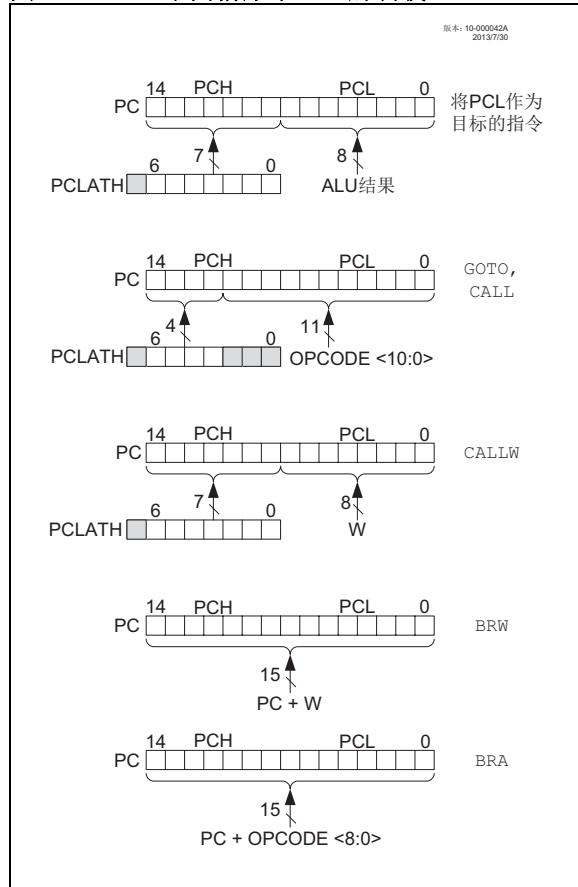
2：仅限 PIC16F1574/5/8/9。

3：未实现，读为 1。

3.4 PCL 和 PCLATH

程序计数器 (PC) 为 15 位宽。其低字节来自可读写的 PCL 寄存器, 高字节 (PC<14:8>) 来自 PCLATH, 不能直接读写。在发生任何复位时, PC 都会被清零。图 3-4 给出了 PC 装载的 5 种情形。

图 3-4: 不同情形下 PC 的装载



3.4.1 修改 PCL

在执行将 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 寄存器中和被写入 PCL 寄存器的值。

3.4.2 计算 GOTO

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

3.4.3 计算函数调用

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

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

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

3.4.4 转移

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

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

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

3.5 堆栈

所有器件都具有 16 级 x 15 位宽的硬件堆栈（见图 3-5 至 3-8）。堆栈既不占用程序存储空间，也不占用数据存储空间。当执行 CALL 或 CALLW 指令，或者中断导致程序跳转时，PC 值将被压入堆栈。而在执行 RETURN、RETLW 或 RETFIE 指令时，将从堆栈中弹出 PC 值。PCLATH 不受压栈或出栈操作的影响。

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

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

3.5.1 访问堆栈

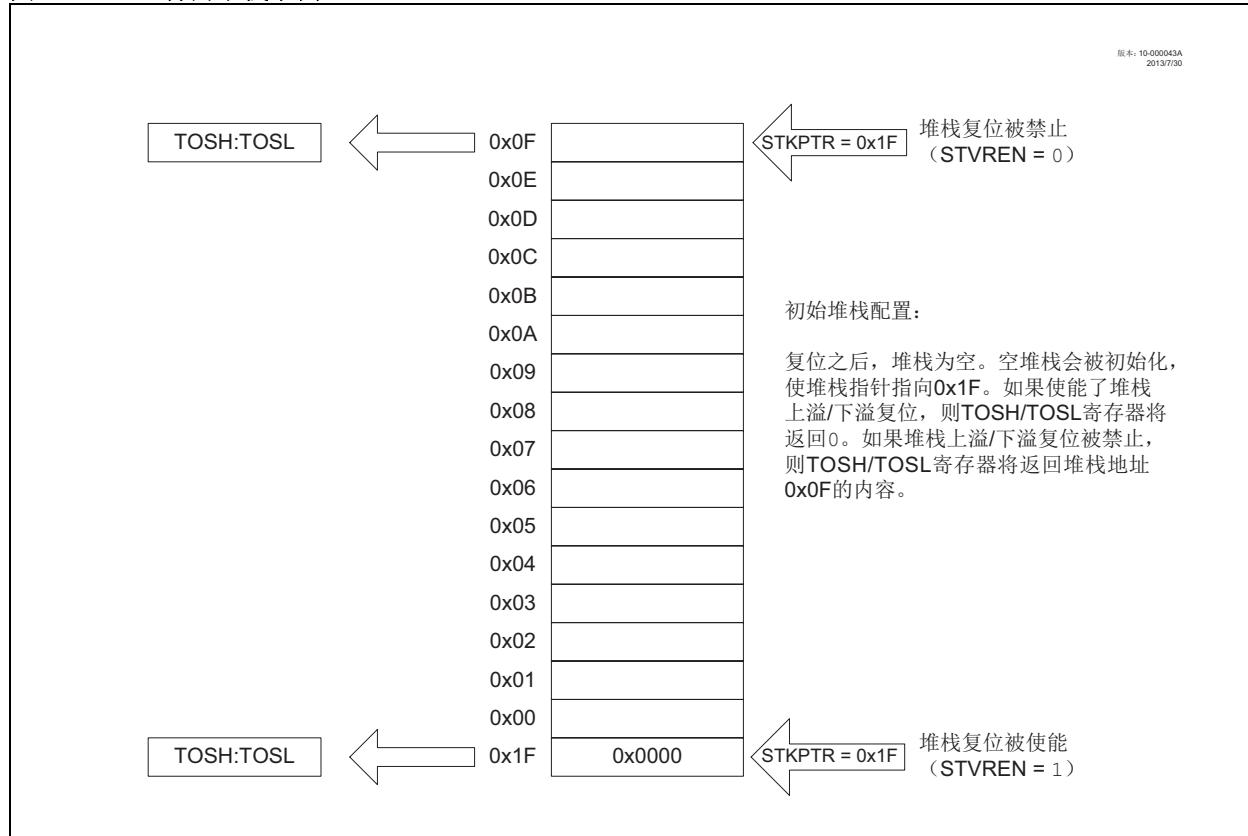
可通过 TOSH、TOSL 和 STKPTR 寄存器使用堆栈。STKPTR 是堆栈指针的当前值。TOSH:TOSL 寄存器对指向栈顶。两个寄存器都是可读写的。由于 PC 的大小为 15 位，所以 TOS 拆分为 TOSH 和 TOSL。要访问堆栈，可以调整 STKPTR 的值（它会决定 TOSH:TOSL 位置），然后读 / 写 TOSH:TOSL。STKPTR 的宽度为 5 位，以允许检测上溢和下溢。

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 3-5 至图 3-8。

图 3-5：访问堆栈示例 1



PIC16(L)F1574/5/8/9

图 3-6: 访问堆栈示例 2

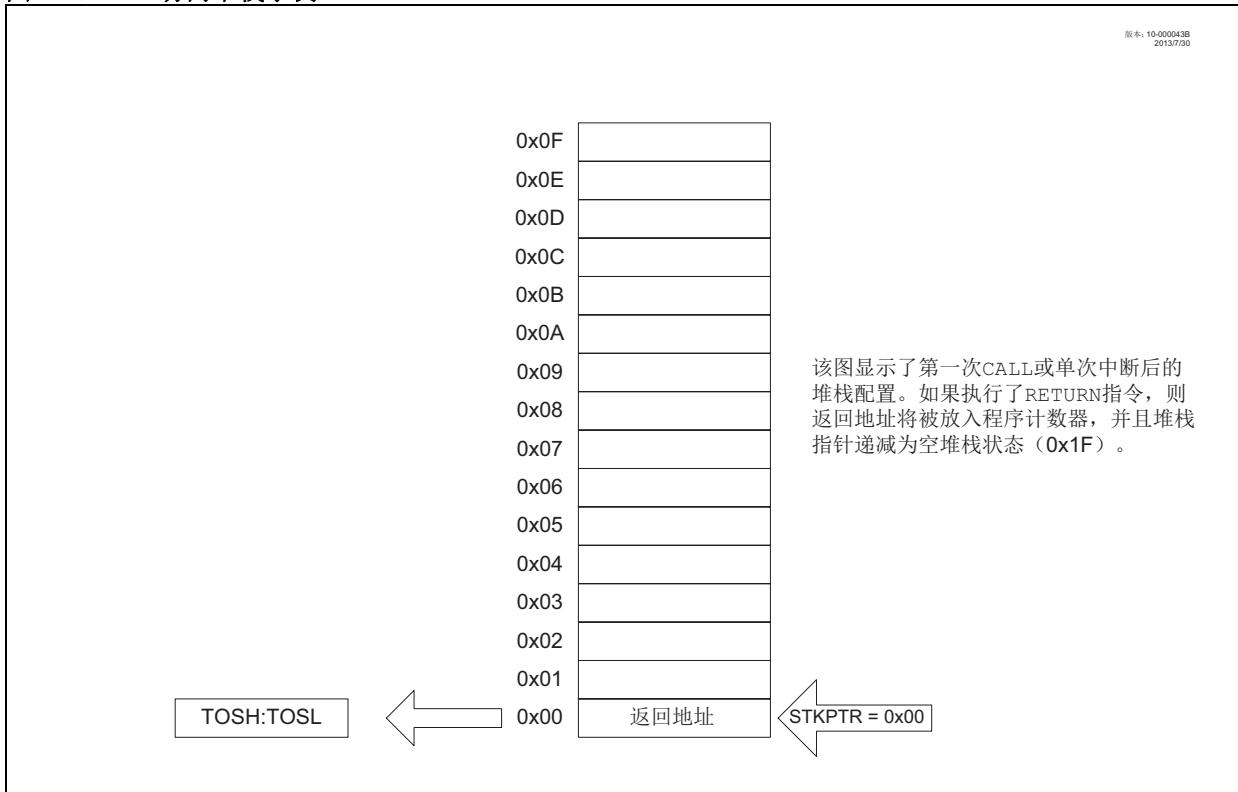


图 3-7: 访问堆栈示例 3

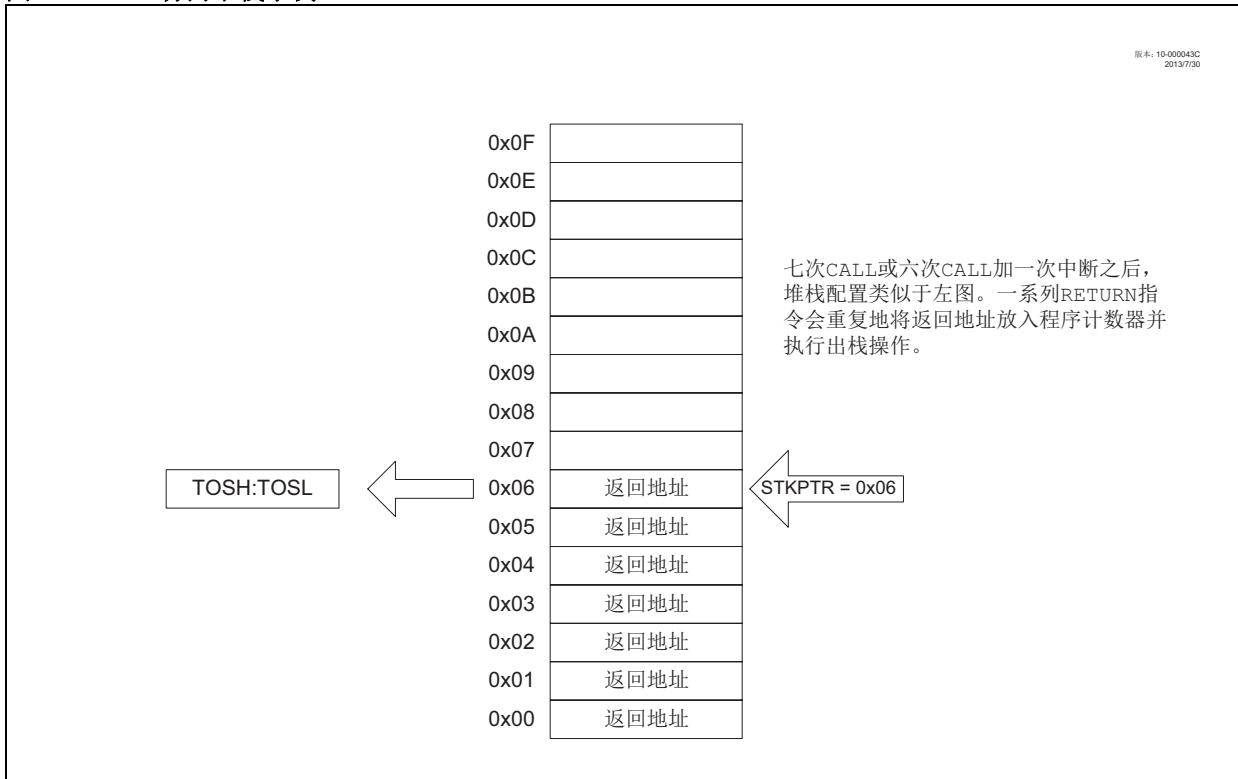
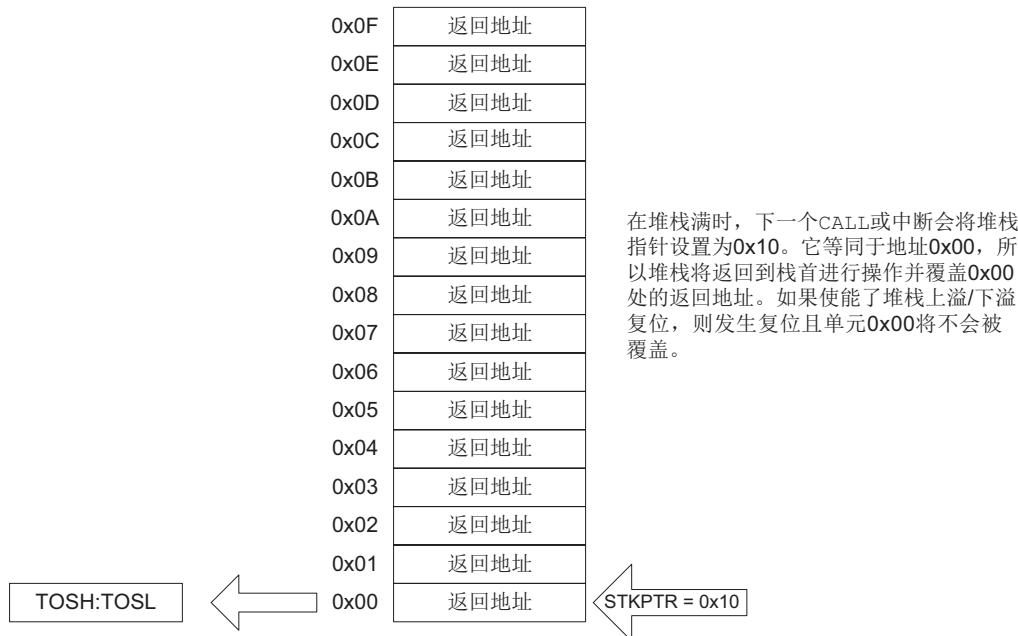


图 3-8：访问堆栈示例 4

版本: 10-000043D
2013/7/30

3.5.2 上溢 / 下溢复位

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

3.6 间接寻址

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

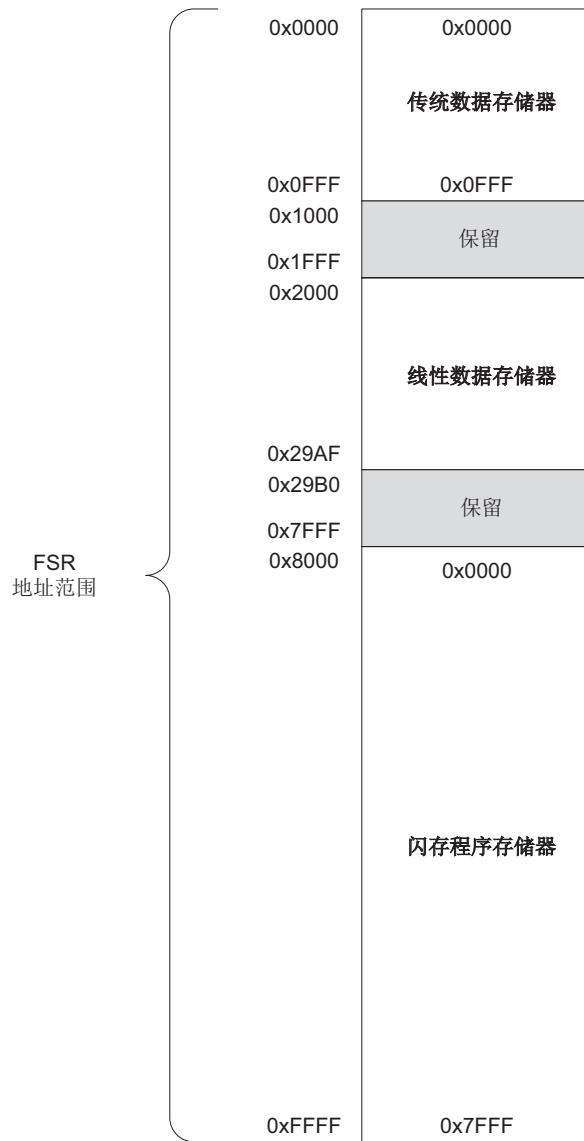
FSR 寄存器构成一个 16 位地址，支持 65536 个存储单元的寻址空间。这些存储单元分为 3 个存储区：

- 传统数据存储器
- 线性数据存储器
- 闪存程序存储器

PIC16(L)F1574/5/8/9

图 3-9：间接寻址

版本: 10-000044A
2013/7/30

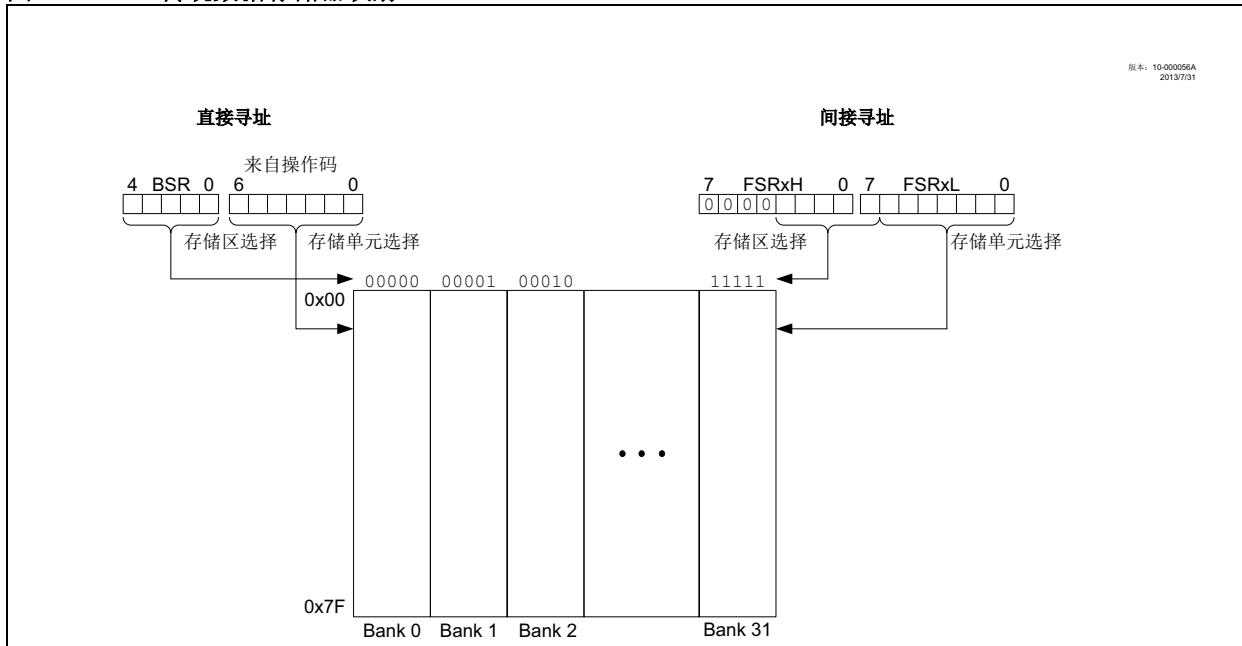


注：不是所有存储区都能完全实现。关于存储器限制，请参见器件存储器表。

3.6.1 传统数据存储器

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

图 3-10：传统数据存储器映射



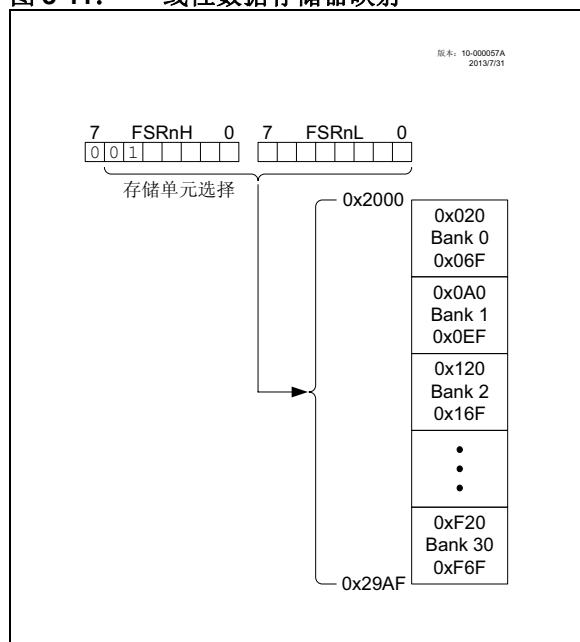
3.6.2 线性数据存储器

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

未实现的存储区将读为 0x00。通过使用线性数据存储区，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区中。

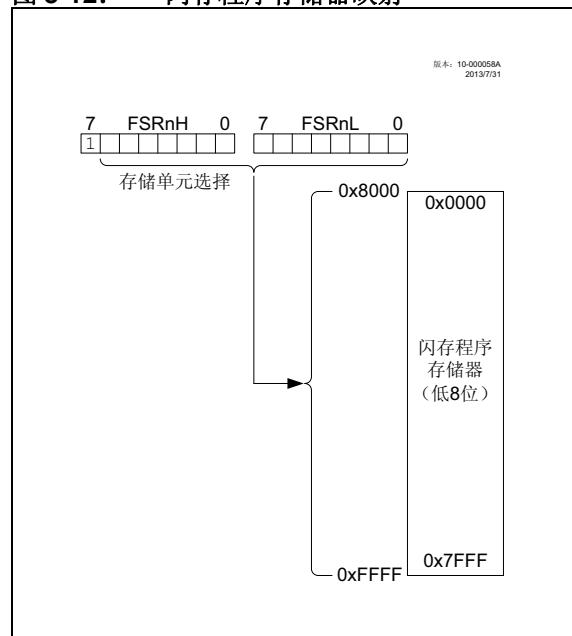
图 3-11： 线性数据存储器映射



3.6.3 闪存程序存储器

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

图 3-12： 闪存程序存储器映射



4.0 器件配置

器件配置功能由配置字、代码保护和器件 ID 组成。

4.1 配置字

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

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

4.2 寄存器定义：配置字

寄存器 4-1： 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	R/P-1	R/P-1	U-1	R/P-1	R/P-1
CP ⁽²⁾	MCLRE	PWRTE ⁽¹⁾	WDTE<1:0>		—	FOSC<1:0>	
bit 7							bit 0

图注：

R = 可读位
0 = 清零

P = 可编程位
1 = 置 1

U = 未实现位，读为 1
n = 空白时或批量擦除后的值

bit 13-12 未实现：读为 1

bit 11 CLKOUTEN：时钟输出使能位

1 = OFF——禁止 CLKOUT 功能。 CLKOUT 引脚为 I/O 或振荡器功能
0 = ON——在 CLKOUT 引脚上使能 CLKOUT 功能

bit 10-9 BOREN<1:0>：欠压复位使能位⁽¹⁾

11 = ON——使能欠压复位。 SBOREN 位被忽略。
10 = SLEEP——在运行时使能欠压复位，在休眠时禁止欠压复位。 SBOREN 位被忽略。
01 = SBODEN——欠压复位由 BORCON 寄存器中的 SBOREN 位控制
00 = OFF——禁止欠压复位。 SBOREN 位被忽略。

bit 8 未实现：读为 1

bit 7 CP：闪存程序存储器代码保护位⁽²⁾

1 = OFF——代码保护关闭。可以读写程序存储器。
0 = ON——代码保护开启。不能外部读写程序存储器。

bit 6 MCLRE：MCLR/VPP 引脚功能选择位

如果 LVP 位 = 1 (ON)：
该位被忽略。 MCLR/VPP 引脚功能为 MCLR；使能弱上拉。

如果 LVP 位 = 0 (OFF)：

1 = ON ——MCLR/VPP 引脚功能为 MCLR；使能弱上拉。
0 = OFF——MCLR/VPP 引脚功能为数字输入； MCLR 在内部被禁止；弱上拉由引脚的 WPU 控制位控制。

bit 5 PWRTE：上电延时定时器使能位⁽¹⁾

1 = OFF——禁止 PWRT
0 = ON——使能 PWRT

bit 4-3 WDTE<1:0>：看门狗定时器使能位

11 = ON——使能 WDT。 SWDTEN 被忽略。
10 = SLEEP——在运行时使能 WDT，在休眠时禁止 WDT。 SWDTEN 被忽略。
01 = SWDTEN——WDT 由 WDTCON 寄存器的 SWDTEN 位控制
00 = OFF——禁止 WDT。 SWDTEN 被忽略。

bit 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：CLKI 引脚为 CLKI 功能
10 = ECM——外部时钟，中等功耗模式：CLKI 引脚为 CLKI 功能
01 = ECL——外部时钟，低功耗模式：CLKI 引脚为 CLKI 功能
00 = INTOSC——CLKI 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。

寄存器 4-2: 配置字 2

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

U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1	R/P-1
—	—	—	—	—	PPS1WAY	WRT<1:0>	
bit 7	bit 0						

图注:

R = 可读位

P = 可编程位

U = 未实现位, 读为 1

0 = 清零

1 = 置 1

n = 空白时或批量擦除后的值

bit 13 **LVP**: 低电压编程使能位 ⁽¹⁾
 1 = ON —— 使能低电压编程。MCLR/VPP 引脚功能为 MCLR。MCLRE 配置位被忽略。
 0 = OFF —— 必须使用 MCLR/VPP 上的高电压进行编程

bit 12 **DEBUG**: 调试器模式位 ⁽²⁾
 1 = OFF —— 禁止在线调试器; ICSPCLK 和 ICSPDAT 是通用 I/O 引脚。
 0 = ON —— 使能在线调试器; ICSPCLK 和 ICSPDAT 专用于调试器。

bit 11 **LPBOREN**: 低功耗欠压复位使能位
 1 = OFF —— 禁止低功耗欠压复位
 0 = ON —— 使能低功耗欠压复位

bit 10 **BORV**: 欠压复位电压选择位 ⁽³⁾
 1 = LOW —— 欠压复位电压 (**VBOR**), 选择低跳变点
 0 = HIGH —— 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 = ON —— 堆栈上溢或下溢将导致复位
 0 = OFF —— 堆栈上溢或下溢不会导致复位

bit 8 **PLLEN**: PLL 使能位
 1 = ON —— 使能 4xPLL
 0 = OFF —— 禁止 4xPLL

bit 7-3 未实现: 读为 1

bit 2 **PPS1WAY**: PPSLOCK 位一次置 1 使能位
 1 = ON 执行解锁序列后 PPSLOCK 位只能置 1 一次; 当 PPSLOCK 置 1 后, 会阻止之后对 PPS 寄存器的所有更改
 0 = OFF PPSLOCK 位可以根据需要置 1 和清零 (前提是解锁序列已执行)

注 1: 当通过 LVP 进入编程模式时, 不能将该位编程为 0。

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

3: 关于具体跳变点电压, 请参见 **VBOR** 参数。

寄存器 4-2： 配置字 2（续）

bit 1-0 **WRT<1:0>**: 闪存自写保护位

4 kW 闪存: (PIC16(L)F1574/8):

11 = OFF	写保护关闭
10 = BOOT	0000h 至 01FFh 被写保护, 0200h 至 0FFFh 可由 PMCON 控制修改
01 = HALF	0000h 至 07FFh 被写保护, 0800h 至 0FFFh 可由 PMCON 控制修改
00 = ALL	0000h 至 0FFFh 被写保护, 没有地址可被 PMCON 控制修改

8 kW 闪存: (PIC16(L)F1575/9):

11 = OFF	写保护关闭
10 = BOOT	0000h 至 01FFh 被写保护, 0200h 至 1FFFh 可由 PMCON 控制修改
01 = HALF	0000h 至 0FFFh 被写保护, 1000h 至 1FFFh 可由 PMCON 控制修改
00 = ALL	0000h 至 1FFFh 被写保护, 没有地址可被 PMCON 控制修改

注 1: 当通过 LVP 进入编程模式时, 不能将该位编程为 0。

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

3: 关于具体跳变点电压, 请参见 **VBOR** 参数。

4.3 代码保护

通过代码保护，可以防止对器件的未授权访问。对程序存储器的内部访问不会受任何代码保护设置影响。

4.3.1 程序存储器保护

整个程序存储空间都通过配置字中的 \overline{CP} 位来防止外部读写操作。当 $CP = 0$ 时，将禁止对程序存储器的外部读写操作，读取时将返回全 0。无论保护位的设置如何，CPU 都可以继续读取程序存储器。对程序存储器的写操作则取决于写保护设置。更多信息，请参见第 4.4 节“写保护”。

4.4 写保护

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

配置字中的 $WRT<1:0>$ 位定义受保护的程序存储块的大小。

4.5 用户 ID

有 4 个存储单元（8000h-8003h）被指定为 ID 存储单元，供用户存储校验和其他代码标识号。在正常执行期间，这些存储单元是可读写的。关于访问这些存储单元的更多信息，请参见第 10.4 节“用户 ID、器件 ID 和配置字访问”。关于校验和计算的更多信息，请参见“*PIC16(L)F157x Memory Programming Specification*”（DS40001766）。

4.6 器件 ID 和版本 ID

14 位器件 ID 字位于 8006h，14 位版本 ID 位于 8005h。这些单元是只读的，不能擦除或修改。关于访问这些存储单元的更多信息，请参见第 10.4 节“用户 ID、器件 ID 和配置字访问”。

开发工具（如器件编程器和调试器）可用于读取器件 ID 和版本 ID。

PIC16(L)F1574/5/8/9

4.7 寄存器定义：器件 ID

寄存器 4-3: DEVICEID: 器件 ID 寄存器⁽¹⁾

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

0 = 清零

1 = 置 1

x = 未知

bit 13-0 **DEV<13:0>**: 器件 ID 位

要确定这些位在具体器件上的读取内容，请参见表 4-1。值为 3FFFh 是无效的。

注 1: 该存储单元无法写入。

寄存器 4-4: REVISIONID: 版本 ID 寄存器⁽¹⁾

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

0 = 清零

1 = 置 1

x = 未知

bit 13-0 **REV<13:0>**: 版本 ID 位

这些位用于标识器件版本。

注 1: 该存储单元无法写入。

表 4-1: 器件 ID 值

器件	器件 ID	版本 ID
PIC16F1574	3000h	2xxxh
PIC16F1575	3001h	2xxxh
PIC16F1578	3002h	2xxxh
PIC16F1579	3003h	2xxxh
PIC16LF1574	3004h	2xxxh
PIC16LF1575	3005h	2xxxh
PIC16LF1578	3006h	2xxxh
PIC16LF1579	3007h	2xxxh

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。

时钟源可以通过外部逻辑电平时钟提供。此外，系统时钟源可由两个内部振荡器之一和 PLL 电路提供，并通过软件来选择速度。其他时钟特性包括：

- 可通过软件选择外部或内部时钟源作为系统时钟源。

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）

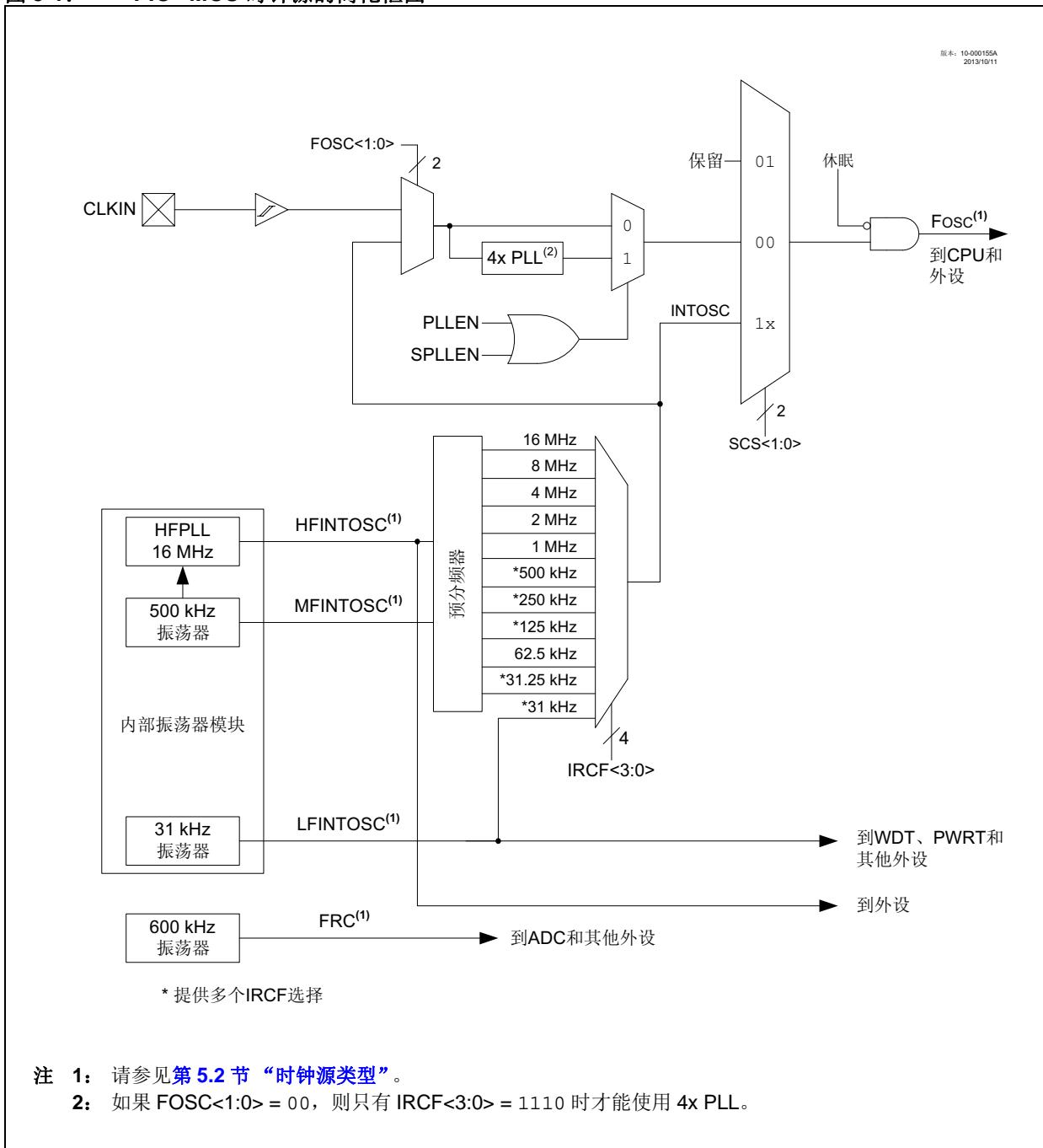
时钟源模式通过配置字中的 FOSC<1:0> 位进行选择。FOSC 位决定在器件刚上电时使用的振荡器类型。

ECH、ECM 和 ECL 时钟模式依靠外部逻辑电平信号作为器件时钟源。

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

PIC16(L)F1574/5/8/9

图 5-1： PIC® MCU 时钟源的简化框图



5.2 时钟源类型

时钟源可分为外部时钟源和内部时钟源。

外部时钟源依靠外部电路工作。

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)，用于生成 3 个内部系统时钟源：16 MHz 高频内部振荡器 (High-Frequency Internal Oscillator, HFINTOSC)、500 kHz 中频内部振荡器 (Medium-Frequency Internal Oscillator, MFINTOSC) 和 31 kHz 低频内部振荡器 (Low-Frequency Internal Oscillator, LFINTOSC)。

通过 OSCCON 寄存器中的系统时钟选择 (System Clock Select, 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：
 - Timer1 振荡器（在运行时），或者
 - 由 FOSC 位的值决定的外部时钟源。

更多信息，请参见[第 5.3 节“时钟切换”](#)。

5.2.1.1 EC 模式

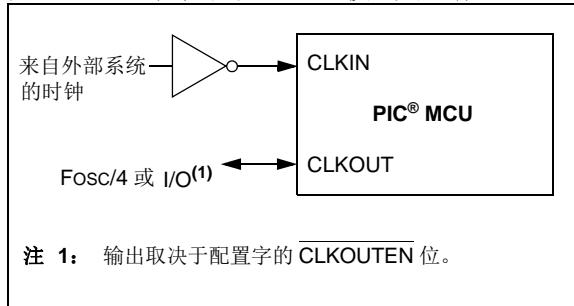
外部时钟 (External Clock, EC) 模式允许外部产生的逻辑电平信号作为系统时钟源。工作在该模式下时，外部时钟源连接到 CLKIN 输入。CLKOUT 可用作通用 I/O 或 CLKOUT。图 5-2 给出了 EC 模式的引脚连接图。

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 (Oscillator Start-up Timer, OST) 被禁止（如可用）。因此，上电复位 (POR) 后或者从休眠模式唤醒后的操作不存在延时。因为 PIC® MCU 的设计是完全静态的，停止外部时钟输入将使器件暂停工作并保持所有数据完整。当再次启动外部时钟时，器件恢复工作，就好像没有停止过一样。

图 5-2：外部时钟 (EC) 模式的工作原理



注 1： 输出取决于配置字的 CLKOUTEN 位。

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 5.2.2.8 节“内部振荡器时钟切换时序”。

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 **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 OSCCON 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 OSCCON 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”。

通过以下方式使能 HFINTOSC：

- 根据所需的 HF 频率配置 OSCCON 寄存器的 **IRCF<3:0>** 位，并且
- **FOSC<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性（HFIOFS）指示 HFINTOSC 何时在其最终值的 0.5% 范围内运行。

5.2.2.2 MFINTOSC

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

MFINTOSC 的输出连接到后分频器和多路开关（见图 5-1）。使用 OSCCON 寄存器的 **IRCF<3:0>** 位，可通过软件选择基于 MFINTOSC 产生的 9 个频率中的一个。更多信息，请参见第 5.2.2.8 节“内部振荡器时钟切换时序”。

发生以下情况时，MFINTOSC 被使能：

- 根据所需的 MF 频率配置 OSCCON 寄存器的 **IRCF<3:0>** 位，并且
- **FOSC<1:0> = 00**，或者
- 将 OSCCON 寄存器的系统时钟源（**SCS**）位设置为 **1x**

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

5.2.2.3 内部振荡器频率调整

500 kHz 内部振荡器出厂时已校准。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。其值是一个 6 位的二进制补码数字。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）和看门狗定时器（WDT）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 600 kHz（标称值）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 输出的后分频器输出连接到一个倍频器（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：

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

注：任何复位后，OSCCON 寄存器的 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，可以实现更快的时钟切换速度。

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相关的 4x PLL 配合使用，产生 32 MHz 内部系统时钟源。要使用 32 MHz 内部时钟源，需要以下设置：

- 配置字中的 FOSC 位必须设置为使用 INTOSC 源作为器件系统时钟（FOSC<1:0> = 00）。
- 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 位必须设置为使用 8 MHz HFINTOSC（IRCF<3:0> = 1110）。
- OSCCON 寄存器中的 SPLLEN 位必须置 1，以使能 4x PLL，或者配置字的 PLLEN 位必须编程为 1。

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

当 OSCCON 寄存器的 SCS 位设置为 1x 时，4x PLL 不可用于内部振荡器。要将 4x PLL 与内部振荡器配合使用，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，会经过一定的延时，新选择的频率才会生效。OSCSTAT 寄存器将反映 HFINTOSC、MFINTOSC 和 LFINTOSC 振荡器的当前工作状态。频率选择序列如下：

- 修改 OSCCON 寄存器的 IRCF<3:0> 位。
- 如果新时钟是关闭的，开始时钟起振延时。
- 时钟切换电路等待当前时钟下降沿出现。
- 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。
- 新时钟现在开始工作。
- OSCSTAT 寄存器按需进行更新。
- 时钟切换完成。

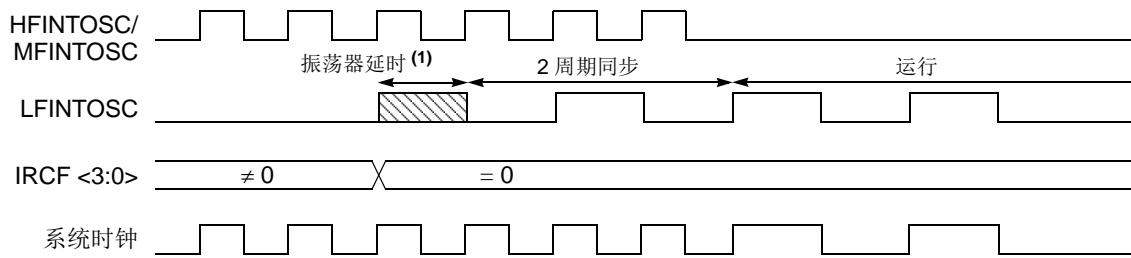
更多详细信息，请参见图 5-3。

如果在同一个内部时钟源的两个时钟之间进行切换（改变内部振荡器速度），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。

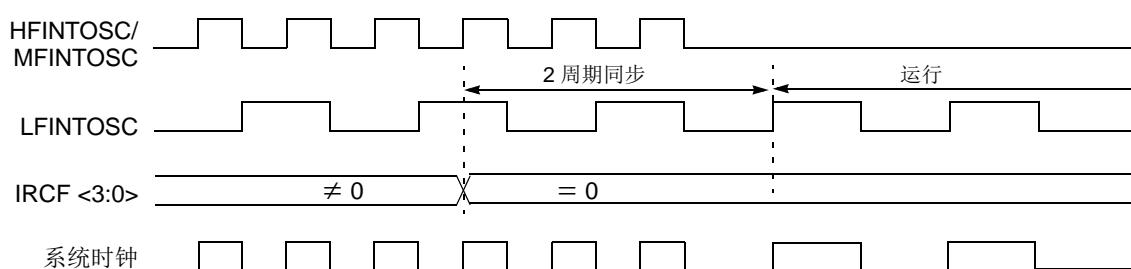
起振延时规范请参见第 27.0 节“电气规范”中的振荡器表。

图 5-3： 内部振荡器切换时序

**HFINTOSC/→ LFINTOSC (禁止 WDT)
MFINTOSC**

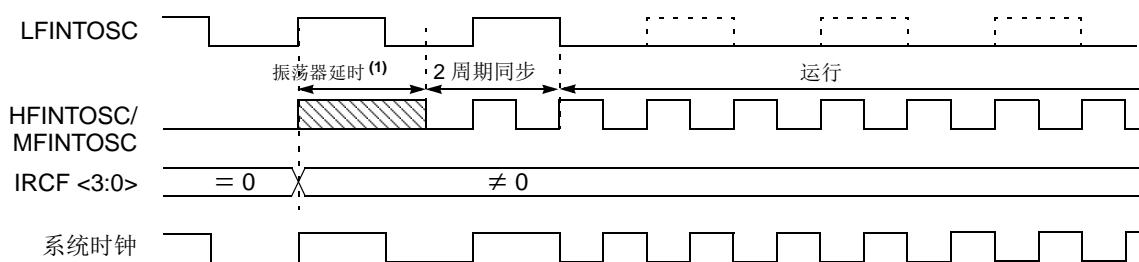


**HFINTOSC/→ LFINTOSC (使能 WDT)
MFINTOSC**



LFINTOSC → HFINTOSC/MFINTOSC

除非使能了 WDT, 否则 LFINTOSC 会关闭



注 1： 关于更多信息，请参见表 5-1 “振荡器切换延时”。

5.3 时钟切换

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

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

5.3.1 系统时钟选择 (SCS) 位

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

- 当 OSCCON 寄存器的 SCS 位 = 00 时，系统时钟源由配置字中的 FOSC<1:0> 位的值决定。
- 当 OSCCON 寄存器的 SCS 位 = 01 时，系统时钟源为 Timer1 振荡器。
- 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。

注：任何自动时钟切换都不会更新 OSCCON 寄存器的 SCS 位。用户可以监视 OSCSTAT 寄存器的 OSTS 位，以确定当前的系统时钟源。

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

5.4 休眠之前的时钟切换

如果在正要进入休眠模式之前请求从旧时钟切换为新时钟，则必须在执行 SLEEP 指令之前确认切换已完成。未这样做可能导致切换未完成，随之完全失去系统时钟。时钟切换通过监视 OSCSTAT 寄存器中的时钟状态位来确认。切换确认可以通过检测新时钟的就绪位置 1 或旧时钟的就绪位清零来完成。例如，当在使用 PLL 的内部振荡器和不使用 PLL 的内部振荡器之间切换时，监视 PLLR 位。当 PLLR 置 1 时，说明切换到 32 MHz 的操作已完成。反之，当 PLLR 清零时，说明从 32 MHz 切换到选定内部时钟的操作已完成。

图 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) ⁽²⁾
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。

2：请参见[第 27.0 节 “电气规范”](#)。

5.5 寄存器定义：振荡器控制

寄存器 5-1：OSCCON：振荡器控制寄存器

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN		IRCF<3:0>		—	—	SCS<1:0>	
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7

SPLLEN： 软件 PLL 使能位

如果配置字中的 PLLEN = 1:

SPLLEN 位被忽略。总是使能 4x PLL（受振荡器要求制约）

如果配置字中的 PLLEN = 0:

1 = 使能 4x PLL

0 = 禁止 4x PLL

bit 6-3

IRCF<3:0>： 内部振荡器频率选择位

1111 = 16 MHz HF

1110 = 8 MHz 或 32 MHz HF（见第 5.2.2.1 节“HFINTOSC”）

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

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。

寄存器 5-2： OSCSTAT： 振荡器状态寄存器

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-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

PLL R: 4x PLL 就绪位

1 = 4x PLL 就绪

0 = 4x PLL 未就绪

bit 5 **OSTS:** 振荡器起振定时器状态位

1 = 依靠由配置字中 FOSC<1:0> 位定义的时钟运行

0 = 依靠内部振荡器 (FOSC<1:0> = 00) 运行

bit 4 **HFIOFR:** 高频内部振荡器就绪位

1 = HFINTOSC 就绪

0 = HFINTOSC 未就绪

bit 3 **HFIOFL:** 高频内部振荡器锁定位

1 = HFINTOSC 的精度小于等于 2%

0 = HFINTOSC 的精度大于 2%

bit 2 **MFIOFR:** 中频内部振荡器就绪位

1 = MFINTOSC 就绪

0 = MFINTOSC 未就绪

bit 1 **LFIOFR:** 低频内部振荡器就绪位

1 = LFINTOSC 就绪

0 = LFINTOSC 未就绪

bit 0 **HFIOFS:** 高频内部振荡器稳定位

1 = HFINTOSC 的精度小于等于 0.5%

0 = HFINTOSC 的精度大于 0.5%

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

PIC16(L)F1574/5/8/9

图 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>		69
OSCSTAT	—	PLLRL	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	70
OSCTUNE	—	—	TUN<5:0>					—	71

图注: — = 未实现位, 读为 0。时钟源不使用阴影单元。

图 5-3: 与时钟源相关的配置字汇总

名称	Bit	Bit -7	Bit -6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	56
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		—	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

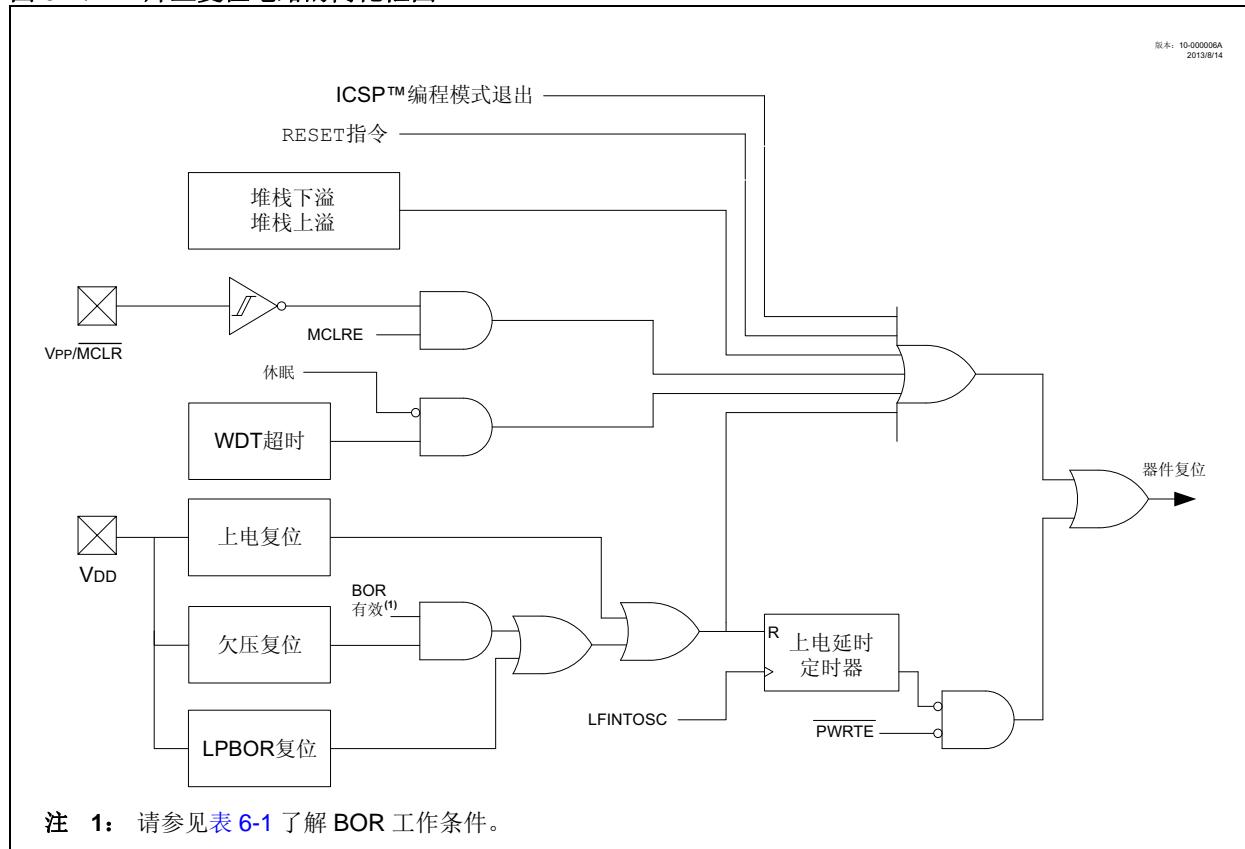
该器件的复位有多种方式:

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

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

图 6-1 给出了片上复位电路的简化框图。

图 6-1: 片上复位电路的简化框图



6.1 上电复位 (POR)

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

6.1.1 上电延时定时器 (PWRT)

上电延时定时器在 POR 或欠压复位时提供一个 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可接受的电压。可通过清零配置字中的 PWRTE 位使能上电延时定时器。

上电延时定时器会在 POR 和 BOR 释放之后启动。

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

6.2 欠压复位 (BOR)

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

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

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

更多信息，请参见表 6-1。

对配置字中的 BORV 位进行配置来选择欠压复位电压。

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

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

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

6.2.1 BOR 总是使能

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

BOR 保护在休眠期间有效。BOR 不会延迟从休眠模式唤醒。

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。

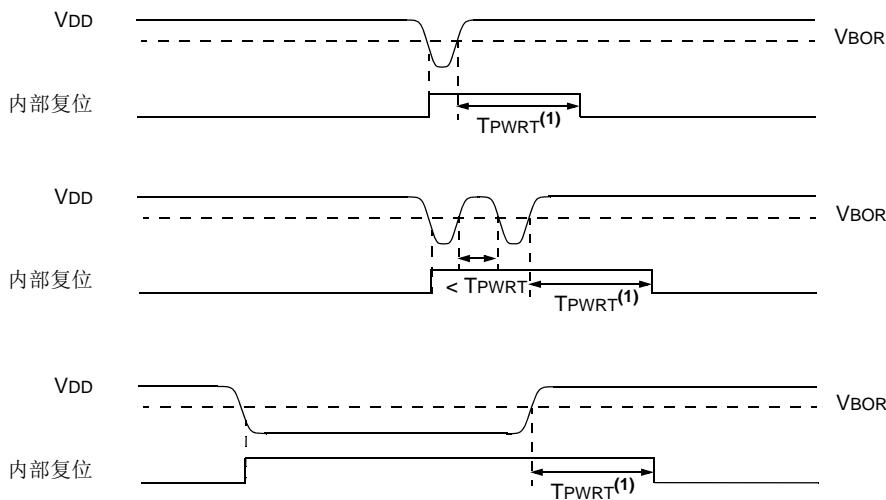
6.2.3 通过软件对 BOR 进行控制

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 电压条件影响而延迟。

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



注 1：仅在 PWRTE 位被编程为 0 时，才应用 TPWRT 延时。

6.3 寄存器定义：BOR 控制

寄存器 6-1：BORCON：欠压复位控制寄存器

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS	—	—	—	—	—	BORRDY
bit 7							
bit 0							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

q = 值取决于具体条件

bit 7 **SBOREN**: 软件欠压复位使能位

如果配置字中的 BOREN<1:0> = 01:

1 = 使能 BOR

0 = 禁止 BOR

如果配置字中的 BOREN<1:0> ≠ 01:

SBOREN 可读 / 写，但对 BOR 没有任何作用

bit 6 **BORFS**: 欠压复位快速启动位 (1)

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

1 = 总是强制开启带隙 (包括休眠 / 唤醒 / 工作等情形)

0 = 带隙正常工作，并且可以关闭

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

BORFS 可读 / 写，但不起任何作用。

bit 5-1 未实现：读为 0

bit 0 **BORRDY**: 欠压复位电路就绪状态位

1 = 欠压复位电路有效

0 = 欠压复位电路无效

注 1：BOREN<1:0> 位位于配置字中。

6.4 低功耗欠压复位 (LPBOR)

低功耗欠压复位 (LPBOR) 的工作方式类似于 BOR，以检测 VDD 引脚上的低电压条件。当检测到电压太低时，器件将保持在复位状态。发生这种情况时，有一个寄存器位 (BOR) 会发生改变，指示发生了 BOR 复位。PCON 中的 BOR 位同时用于 BOR 和 LPBOR。请参见 [寄存器 6-2](#)。

LPBOR 电压阈值 (VLPBOR) 的容许范围大于 BOR (VBOR)，但只需低得多的电流 (LPBOR 电流) 即可工作。LPBOR 在 BOR 配置为禁止 (BOREN = 00) 或在休眠模式下禁止 (BOREN = 10) 时使用。

关于 LPBOR 如何与其他模块进行交互的信息，请参见 [图 6-1](#)。

6.4.1 使能 LPBOR

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

6.5 MCLR

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

表 6-2: MCLR 配置

MCLRE	LVP	MCLR
0	0	禁止
1	0	使能
x	1	使能

6.5.1 MCLR 使能

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

器件在 MCLR 复位路径中有一个噪声滤波器。该滤波器检测并滤除小脉冲。

注: 复位不会将 MCLR 引脚驱动为低电平。

6.5.2 MCLR 禁止

当 MCLR 被禁止时，引脚将用作通用输入，内部弱上拉由软件控制。更多信息，请参见 [第 11.1 节 “PORTA 寄存器”](#)。

6.6 看门狗定时器 (WDT) 复位

如果固件未在超时周期内发出 CLRWDT 指令，看门狗定时器会产生复位。STATUS 寄存器中的 TO 和 PD 位会改变，指示发生了 WDT 复位。更多信息，请参见 [第 9.0 节 “看门狗定时器 \(WDT\)”](#)。

6.7 RESET 指令

RESET 指令会引起器件复位。PCON 寄存器中的 RI 位会被设置为 0。关于发生 RESET 指令之后的默认条件，请参见 [表 6-4](#)。

6.8 堆栈上溢 / 下溢复位

器件可以在堆栈上溢或下溢时复位。PCON 寄存器的 STKOVF 或 STKUNF 位用于指示复位条件。这些复位通过将配置字中的 STVREN 位置 1 来使能。更多信息，请参见 [第 3.5.2 节 “上溢 / 下溢复位”](#)。

6.9 编程模式退出

在退出编程模式时，器件的行为与刚刚发生 POR 时的情况相同。

6.10 上电延时定时器

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

上电延时定时器由配置字的 PWRTE 位控制。

6.11 启动序列

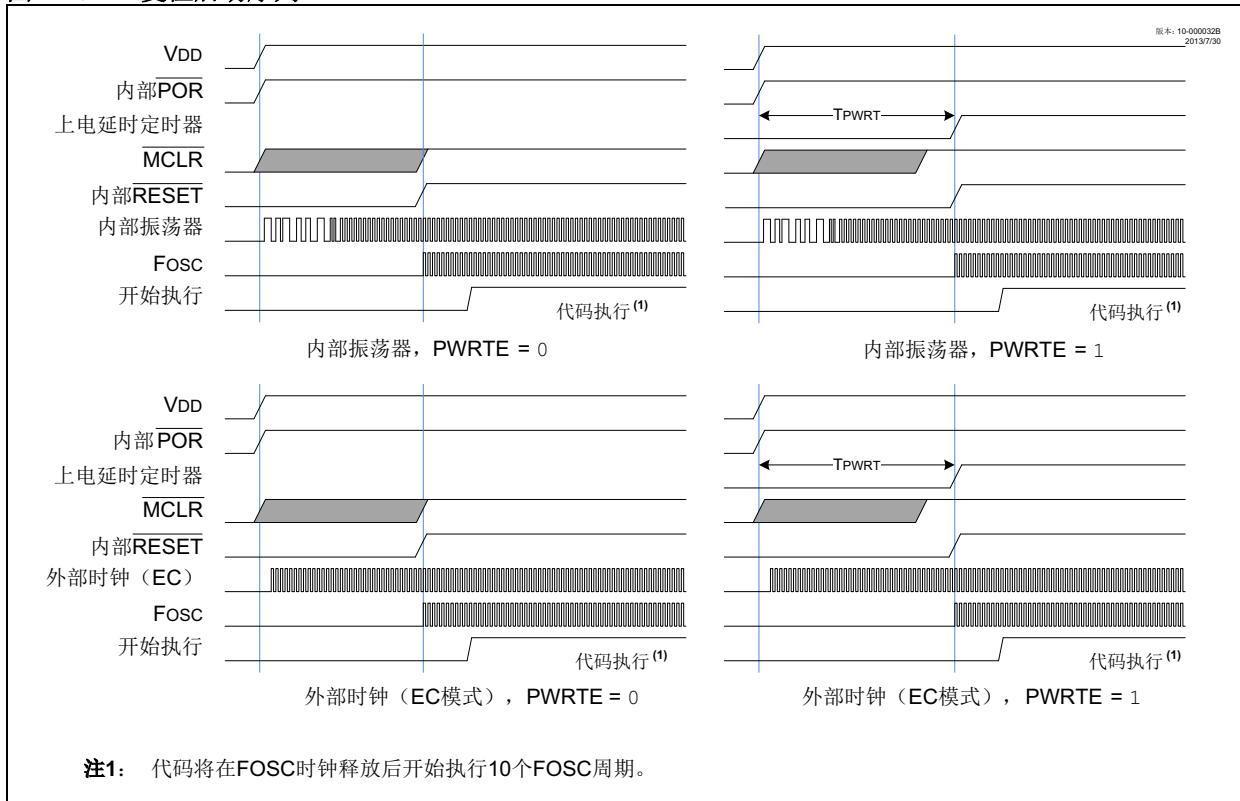
在 POR 或 BOR 释放时，只有先发生以下事件，器件才会开始执行：

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

总延时取决于振荡器配置和上电延时定时器配置。更多信息，请参见 [第 5.0 节 “振荡器模块”](#)。

上电延时定时器的运行与 MCLR 复位无关。如果 MCLR 保持低电平的时间足够长，上电延时定时器将会延时结束。当 MCLR 变为高电平时，器件将在 10 个 FOSC 周期之后开始执行代码（见 [图 6-3](#)）。这对于测试或同步多个并行工作的器件来说是非常有用的。

图 6-3：复位启动序列



6.12 确定复位原因

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

表 6-3：复位状态位及其含义

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

表 6-4：特殊寄存器的复位条件

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

图注：u = 不变，x = 未知，- = 未实现位，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（**POR**）
- 欠压复位（**BOR**）
- **RESET** 指令复位（**RI**）
- **MCLR** 复位（**RMCLR**）
- 看门狗定时器复位（**RWDT**）
- 堆栈下溢复位（**STKUNF**）
- 堆栈上溢复位（**STKOVF**）

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	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	RWDT	RMCLR	<u>RI</u>	<u>POR</u>	<u>BOR</u>
bit 7	bit 0						

图注：

HC = 硬件清零位

HS = 硬件置 1 位

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

q = 值取决于具体条件

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

PIC16(L)F1574/5/8/9

表 6-5: 与复位相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	75
PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	79
STATUS	—	—	—	TO	PD	Z	DC	C	23
WDTCON	—	—	—	—	WDTPS<4:0>	—	—	SWDTEN	99

图注: — = 未实现位, 读为 0。复位不使用阴影单元。

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

表 6-6: 与复位相关的配置字汇总

名称	Bit	Bit -7	Bit -6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	56
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOREN	BORV	STVREN	PLLEN	57
	7:0	—	—	—	—	—	PPS1WAY	WRT<1:0>		

图注: — = 未实现位, 读为 0。复位不使用阴影单元。

7.0 中断

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

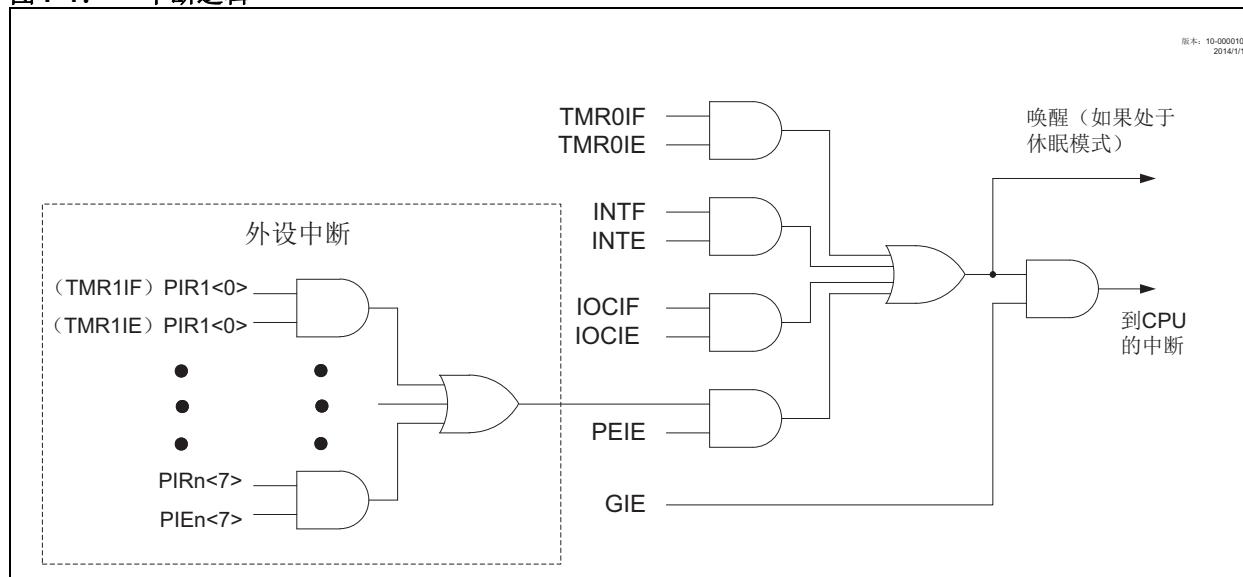
本章包含了关于中断的以下信息：

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

许多外设都会产生中断。详情请参见相应章节。

图 7-1 给出了中断逻辑的框图。

图 7-1： 中断逻辑



7.1 工作原理

任何器件复位时都会禁止中断。通过将以下位置 1 允许相应中断：

- INTCON 寄存器的 GIE 位
- 特定中断事件的中断允许位
- INTCON 寄存器的 PEIE 位（如果中断事件的中断允许位包含在 PIE1、PIE2 和 PIE3 寄存器中）

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

当中断事件发生时，若 GIE 位置 1，将发生以下事件：

- 清除当前预取的指令
- GIE 位清零
- 程序计数器（PC）的当前值压入堆栈
- 自动将关键寄存器保存到影子寄存器中
(见第 7.5 节“自动现场保护”。)
- 将中断向量 0004h 装入 PC

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

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

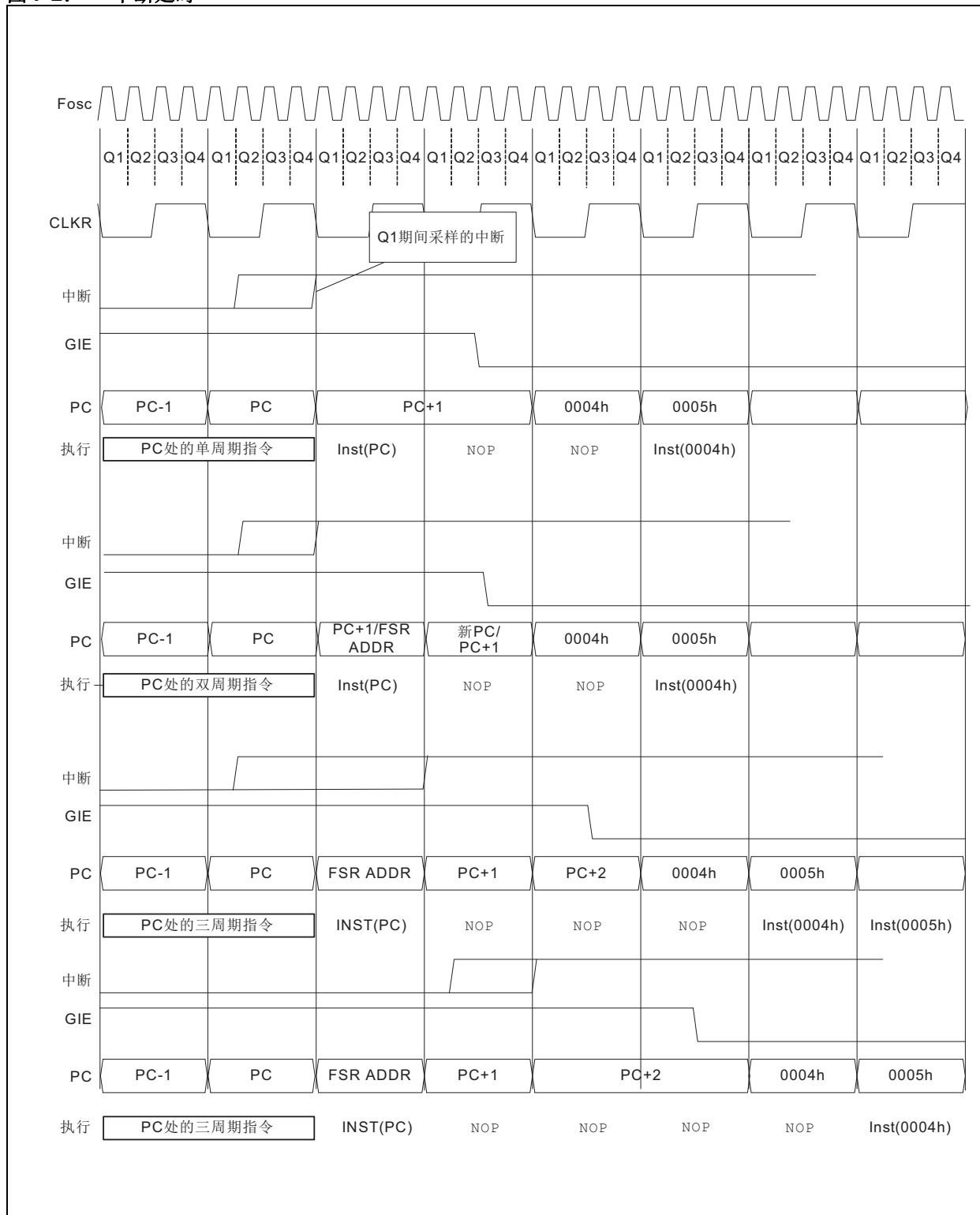
关于特定中断操作的更多信息，请参见其外设章节。

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

7.2 中断延时

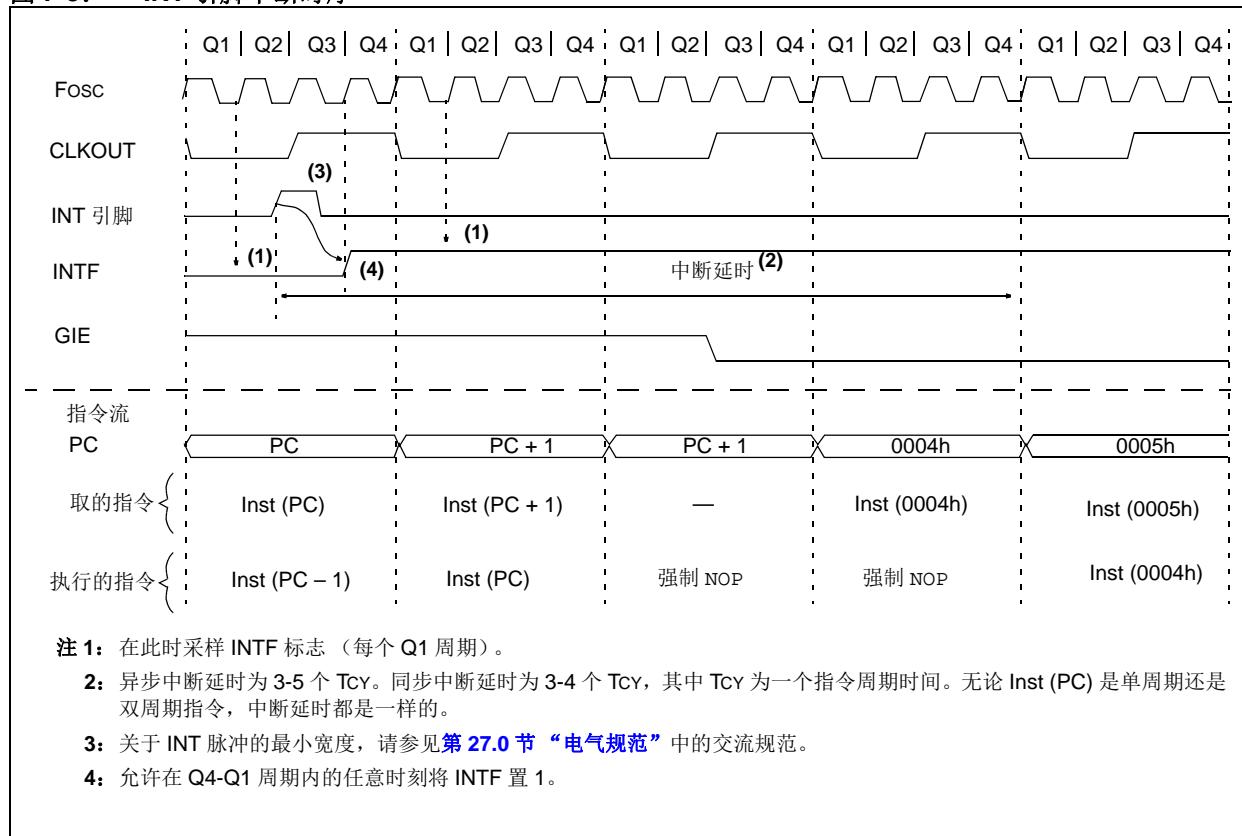
中断延时定义为从发生中断事件到开始执行中断向量处代码经过的时间。同步中断的延时为 3 或 4 个指令周期。对于异步中断，延时为 3 至 5 个指令周期，这取决于中断何时发生。更多详细信息，请参见图 7-2 和图 7-3。

图 7-2： 中断延时



PIC16(L)F1574/5/8/9

图 7-3： INT 引脚中断时序



7.3 休眠期间的中断

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

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

7.4 INT 引脚

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

OPTION_REG 寄存器的 INTEDG 位确定中断在哪个边沿发生。INTEDG 位置 1 时，上升沿将引起中断。INTEDG 位清零时，下降沿将引起中断。INTCON 寄存器的 INTF 位将在 INT 引脚上出现有效边沿时置 1。如果 GIE 和 INTE 位也置 1，则处理器会将程序执行重定向到中断向量。

7.5 自动现场保护

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

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

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

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⁾	TMROIE	INTE	IOCIE	TMROIF	INTF	IOCIF ⁽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 **TMROIE：** Timer0 溢出中断允许位

1 = 允许 Timer0 中断

0 = 禁止 Timer0 中断

bit 4 **INTE：** INT 外部中断允许位

1 = 允许 INT 外部中断

0 = 禁止 INT 外部中断

bit 3 **IOCIE：** 电平变化中断允许位

1 = 允许电平变化中断

0 = 禁止电平变化中断

bit 2 **TMROIF：** Timer0 溢出中断标志位

1 = TMRO 寄存器已溢出

0 = TMRO 寄存器未溢出

bit 1 **INTF：** INT 外部中断标志位

1 = 发生了 INT 外部中断

0 = 未发生 INT 外部中断

bit 0 **IOCIF：** 电平变化中断标志位⁽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： IOCIF 标志位是只读位，它在 IOCxR 寄存器中的所有电平变化中断标志都已用软件清零时清零。

寄存器 7-2: PIE1: 外设中断允许寄存器 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE
bit 7				—	—		bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7 **TMR1GIE:** Timer1 门控中断允许位

1 = 允许 Timer1 门控采集中断
0 = 禁止 Timer1 门控采集中断

bit 6 **ADIE:** 模数转换器 (ADC) 中断允许位

1 = 允许 ADC 中断
0 = 禁止 ADC 中断

bit 5 **RCIE:** USART 接收中断允许位

1 = 允许 USART 接收中断
0 = 禁止 USART 接收中断

bit 4 **TXIE:** USART 发送中断允许位

1 = 允许 USART 发送中断
0 = 禁止 USART 发送中断

bit 3-2 未实现: 读为 0

bit 1 **TMR2IE:** TMR2 与 PR2 匹配中断允许位

1 = 允许 Timer2 与 PR2 匹配中断
0 = 禁止 Timer2 与 PR2 匹配中断

bit 0 **TMR1IE:** Timer1 溢出中断允许位

1 = 允许 Timer1 溢出中断
0 = 禁止 Timer1 溢出中断

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

寄存器 7-3: PIE2: 外设中断允许寄存器 2

U-0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
—	C2IE	C1IE	—	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 = 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 4-0 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。

寄存器 7-4: PIE3: 外设中断允许寄存器 3

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7 **PWM4IE:** PWM4 中断允许位
1 = 允许 PWM4 中断
0 = 禁止 PWM4 中断
- bit 6 **PWM3IE:** PWM3 中断允许位
1 = 允许 PWM3 中断
0 = 禁止 PWM3 中断
- bit 5 **PWM2IE:** PWM2 中断允许位
1 = 允许 PWM2 中断
0 = 禁止 PWM2 中断
- bit 4 **PWM1IE:** PWM1 中断允许位
1 = 允许 PWM1 中断
0 = 禁止 PWM1 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。

PIC16(L)F1574/5/8/9

寄存器 7-5: PIR1: 外设中断请求寄存器 1

R/W-0/0	R/W-0/0	R-0/0	R-0/0	U-0	U-0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	—	—	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 **RCIF:** USART 接收中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 4 **TXIF:** USART 发送中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 3-2 未实现: 读为 0

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 溢出中断标志位

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

注: 当中断条件产生时, 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。USART RCIF 和 TXIF 位是只读位。

寄存器 7-6: 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	U-0	U-0	U-0
—	C2IF	C1IF	—	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 = 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 4-0 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。

寄存器 7-7：PIR3：外设中断请求寄存器 3

R-0/0	R-0/0	R-0/0	R-0/0	U-0	U-0	U-0	U-0
PWM4IF ⁽¹⁾	PWM3IF ⁽¹⁾	PWM2IF ⁽¹⁾	PWM1IF ⁽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 **PWM4IF:** PWM4 中断标志位⁽¹⁾

1 = 中断处于待处理状态

0 = 中断不处于待处理状态

bit 6 **PWM3IF:** PWM3 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 **PWM2IF:** PWM2 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 4 **PWM1IF:** PWM1 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 3-0 未实现：读为 0

注 1：这些位是只读位。必须通过对模块内的标志寄存器寻址来清零这些位。

2：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。

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	86
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			178
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIE2	—	C2IE	C1IE	—	—	—	—	—	88
PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
PIR2	—	C2IF	C1IF	—	—	—	—	—	91
PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	92

图注：— = 未实现位，读为 0。中断不使用阴影单元。

8.0 掉电模式（休眠）

可通过执行 SLEEP 指令进入掉电模式。

在进入休眠模式时，会存在以下情况：

1. 如果在休眠期间使能 WDT，则 WDT 会清零，但保持运行。
2. STATUS 寄存器的 PD 位被清零。
3. STATUS 寄存器的 TO 位被置 1。
4. CPU 时钟被禁止。
5. 31 kHz LFINTOSC 不受影响，使用它工作的外设可以在休眠模式下继续工作。
6. 当所选的 Timer1 时钟源为以下时钟源时，Timer1 和使用 Timer1 工作的外设可以在休眠模式下继续工作：
 - LFINTOSC
 - T1CKI
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
- 外部电路向 I/O 引脚灌电流
- 内部电路从 I/O 引脚拉电流
- 从带内部弱上拉的引脚汲取的电流
- 使用 31 kHz LFINTOSC 的模块
- 使用 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 引脚拉到 VDD 或 VSS。

可能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 从休眠模式唤醒

发生以下任一事件将器件从休眠模式唤醒：

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

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

当执行 SLEEP 指令时，下一条指令（PC + 1）被预先取出。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。

器件从休眠模式唤醒时，都将清零 WDT，而与唤醒原因无关。

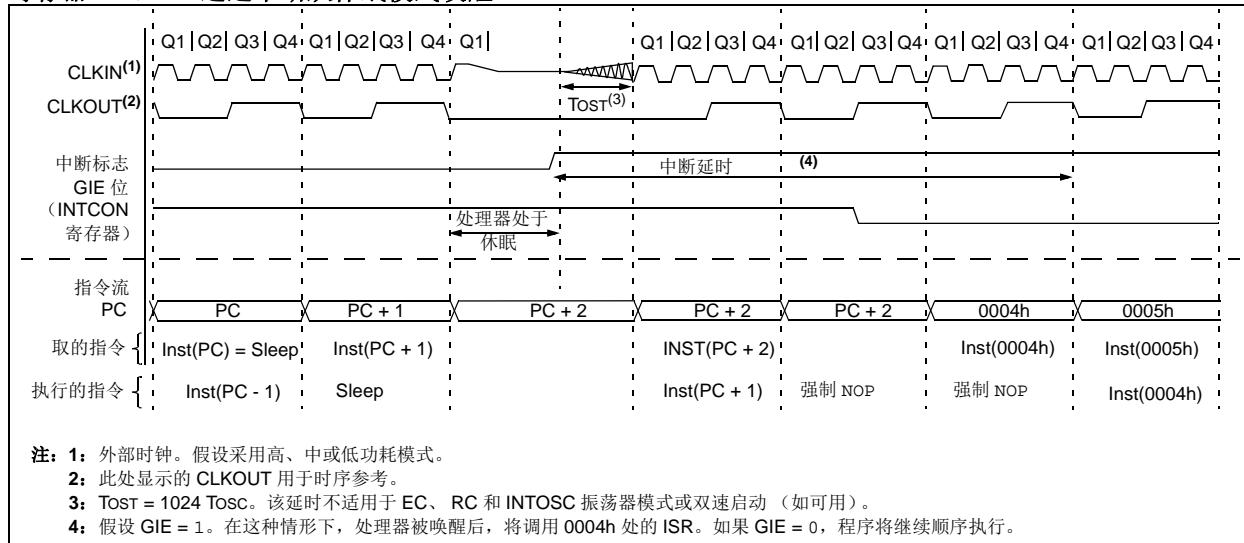
8.1.1 使用中断唤醒

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

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

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

寄存器 8-1：通过中断从休眠模式唤醒



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 及其相关的参考电压电路必须保持活动状态。

低功耗休眠模式允许用户优化休眠模式下的工作电流。通过将 VREGCON 寄存器的 VREGPM 位置 1，并在器件处于休眠模式时将 LDO 和参考电压电路置为低功耗状态，可以选择低功耗休眠模式。

8.2.1 休眠电流与唤醒时间

在默认工作模式下，处于休眠模式时，LDO 和参考电压电路会保持为正常配置。由于所有电路都保持活动状态，所以器件能够快速地退出休眠模式。在低功耗休眠模式下，从休眠模式唤醒时，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) 模块可以采用 HFINTOSC 振荡器作为时钟源或输入源。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 [24.10 “休眠期间的操作”](#) 一节。

注： PIC16LF1574/5/8/9 器件不具有可配置的低功耗休眠模式。PIC16LF1574/5/8/9 是非稳压器件，它们在休眠模式下总是处于最低功耗状态，并且没有唤醒时间延时。这些器件的最大 VDD 和 I/O 电压低于 PIC16F1574/5/8/9 器件。更多信息，请参见 [第 27.0 节 “电气规范”](#)。

8.3 寄存器定义：稳压器控制

寄存器 8-1： VREGCON： 稳压器控制寄存器⁽¹⁾

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-2 未实现：读为 0

bit 1 **VREGPM：** 稳压器功耗模式选择位

1 = 休眠时使能低功耗休眠模式⁽²⁾

休眠时消耗的电流最低，唤醒速度较慢

0 = 休眠时使能正常功耗模式⁽²⁾

休眠时消耗的电流较高，唤醒速度较快

bit 0 保留：读为 1。保持该位置 1。

注 1：仅限 PIC16F1574/5/8/9。

2：请参见第 27.0 节“电气规范”。

表 8-1：与掉电模式相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	143
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	143
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	143
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIE2	—	C2IE	C1IE	—	—	—	—	—	88
PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
PIR2	—	C2IF	C1IF	—	—	—	—	—	91
PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	92
STATUS	—	—	—	TO	PD	Z	DC	C	23
WDTCON	—	—	WDTPS<4:0>				SWDTEN	99	

注：— = 未实现，读为 0。掉电模式不使用阴影单元。

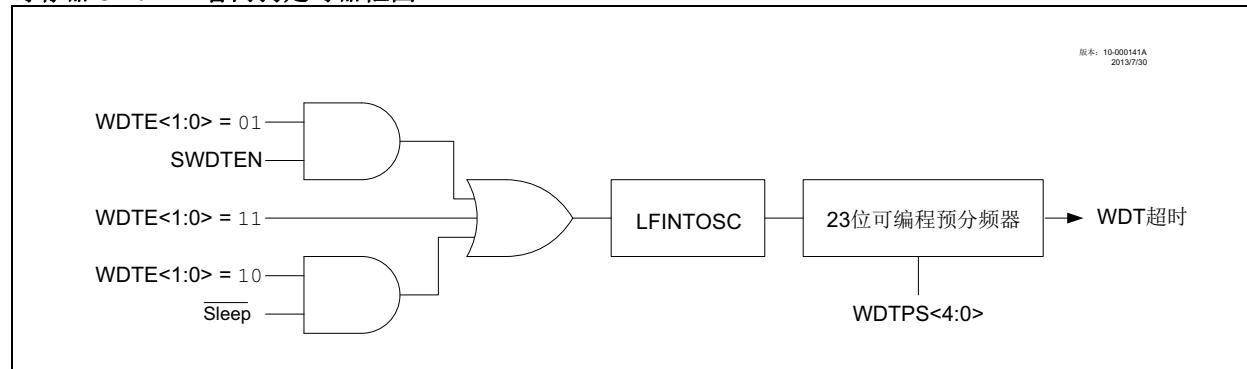
9.0 看门狗定时器 (WDT)

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

WDT 具有以下特性：

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

寄存器 9-1： 看门狗定时器框图



9.1 独立时钟源

WDT 的时基基于 31 kHz LFINTOSC 内部振荡器。本章中的时间间隔均基于 1 ms 的标称时间间隔。关于 LFINTOSC 容差，请参见第 27.0 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 WDTCON 寄存器的 SWDTEN 位进行控制。

WDT 保护在休眠期间不变。更多详细信息，请参见表 9-1。

表 9-1：WDT 工作模式

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

表 9-2：WDT 清零条件

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

9.6 寄存器定义：看门狗控制

寄存器 9-1： WDTCON： 看门狗定时器控制寄存器

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 未实现：读为 0

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

位值 = 预分频比

11111 = 保留。产生最小的时间间隔 (1:32)

•

•

•

10011 = 保留。产生最小的时间间隔 (1:32)

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

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

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

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

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

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

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

01011 = 1:65536 (时间间隔标称值为 2s) (复位值)

01010 = 1:32768 (时间间隔标称值为 1s)

01001 = 1:16384 (时间间隔标称值为 512 ms)

01000 = 1:8192 (时间间隔标称值为 256 ms)

00111 = 1:4096 (时间间隔标称值为 128 ms)

00110 = 1:2048 (时间间隔标称值为 64 ms)

00101 = 1:1024 (时间间隔标称值为 32 ms)

00100 = 1:512 (时间间隔标称值为 16 ms)

00011 = 1:256 (时间间隔标称值为 8 ms)

00010 = 1:128 (时间间隔标称值为 4 ms)

00001 = 1:64 (时间间隔标称值为 2 ms)

00000 = 1:32 (时间间隔标称值为 1 ms)

bit 0 **SWDTEN： 看门狗定时器软件使能 / 禁止位**

如果 WDTE<1:0> = 1x:

该位被忽略。

如果 WDTE<1:0> = 01:

1 = WDT 开启

0 = WDT 关闭

如果 WDTE<1:0> = 00:

该位被忽略。

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

PIC16(L)F1574/5/8/9

表 9-3: 与看门狗定时器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		69
PCON	STKOVF	STKUNF	—	RWDT	RMCLR	RI	POR	BOR	79
STATUS	—	—	—	TO	PD	Z	DC	C	23
WDTCON	—	—	WDTPS<4:0>				SWDTEN	—	99

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

表 9-4: 与看门狗定时器相关的配置字汇总

名称	Bit	Bit -7	Bit -6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	56
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		—	FOSC<1:0>		

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 / 擦除电压由片上电荷泵产生。

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和擦除操作。写保护不会影响器件编程器对器件进行读、写或擦除操作。

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

10.1 PMADRL 和 PMADRH 寄存器

PMADRH:PMADRL 寄存器对能寻址最大 16K 字的程序存储器。当选择程序地址值时，地址的 MSB 被写入 PMADRH 寄存器，而 LSB 被写入 PMADRL 寄存器。

10.1.1 PMCON1 和 PMCON2 寄存器

PMCON1 是访问闪存程序存储器的控制寄存器。

控制位 RD 和 WR 分别用于启动读和写操作。用软件只能将这两位置 1，无法清零。在读或写操作完成时，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。闪存程序存储器由多行构成。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 寄存器对来装入这些写锁存器。

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

关于闪存程序存储器的擦除行大小和写锁存器数量，请参见表 10-1。

表 10-1：闪存构成（按器件）

器件	行擦除（字）	写锁存器（字）
PIC16(L)F1574		
PIC16(L)F1575		
PIC16(L)F1578		
PIC16(L)F1579		
	32	32

10.2.1 读取闪存程序存储器

要读取程序存储单元，用户必须：

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

一旦读控制位置 1，闪存程序存储器控制器将使用第二个指令周期来读取数据。这会导致紧随“BSF PMCON1, RD”指令的第二条指令被忽略。在紧接着的下一个周期，PMDATH:PMDATL 寄存器对中即有数据；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或用户写入新值为止。

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

图 10-1：闪存程序存储器读操作流程图

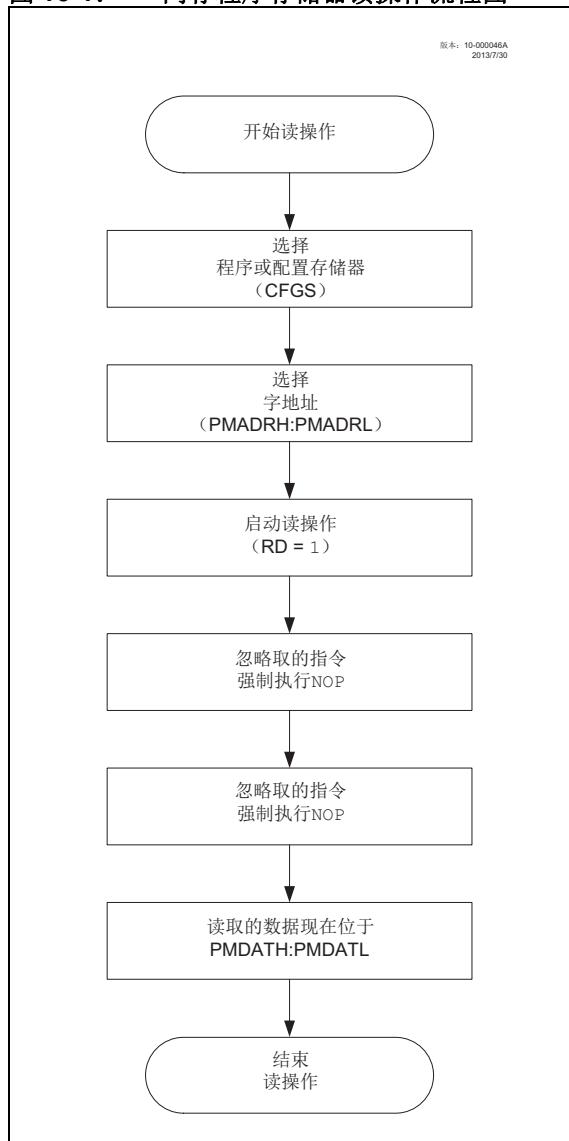
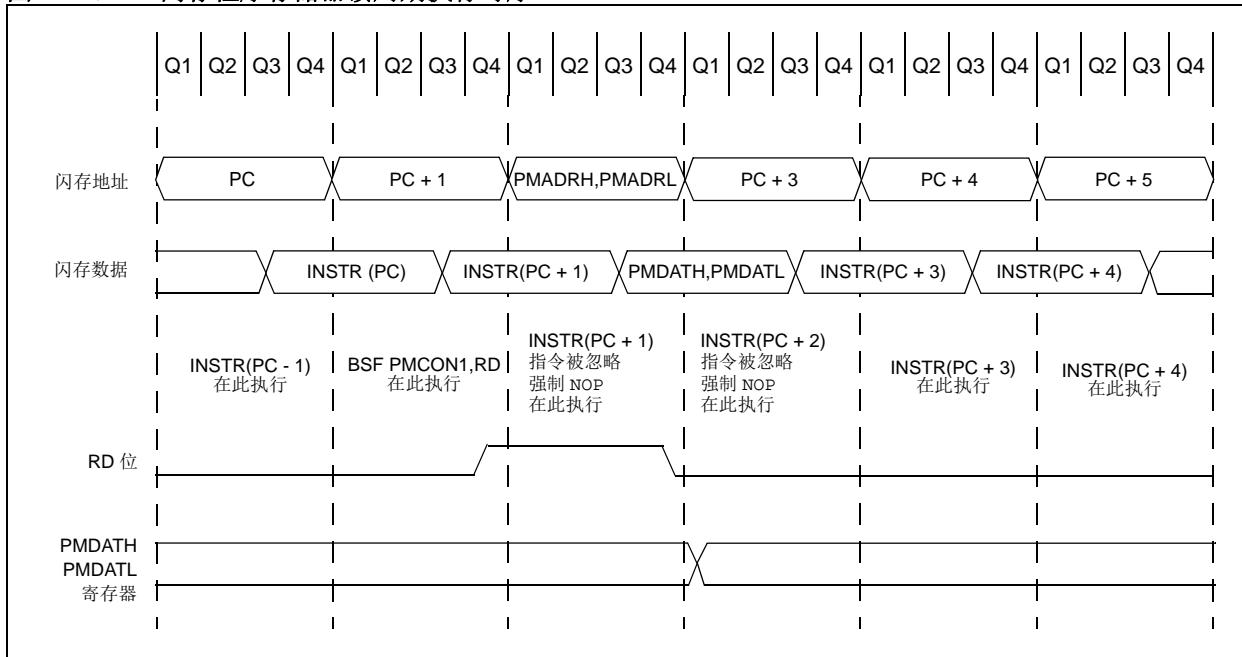


图 10-2：闪存程序存储器读周期执行时序



例 10-1：读取闪存程序存储器

```

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

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

BCF      PMCON1,CFG5      ; Do not select Configuration Space
BSF      PMCON1,RD        ; Initiate read
NOP      ; Ignored (Figure 10-2)
NOP      ; Ignored (Figure 10-2)

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

```

10.2.2 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
- 装入程序存储器写锁存器
- 将程序存储器写锁存器内容写入程序存储器
- 将程序存储器写锁存器内容写入用户 ID

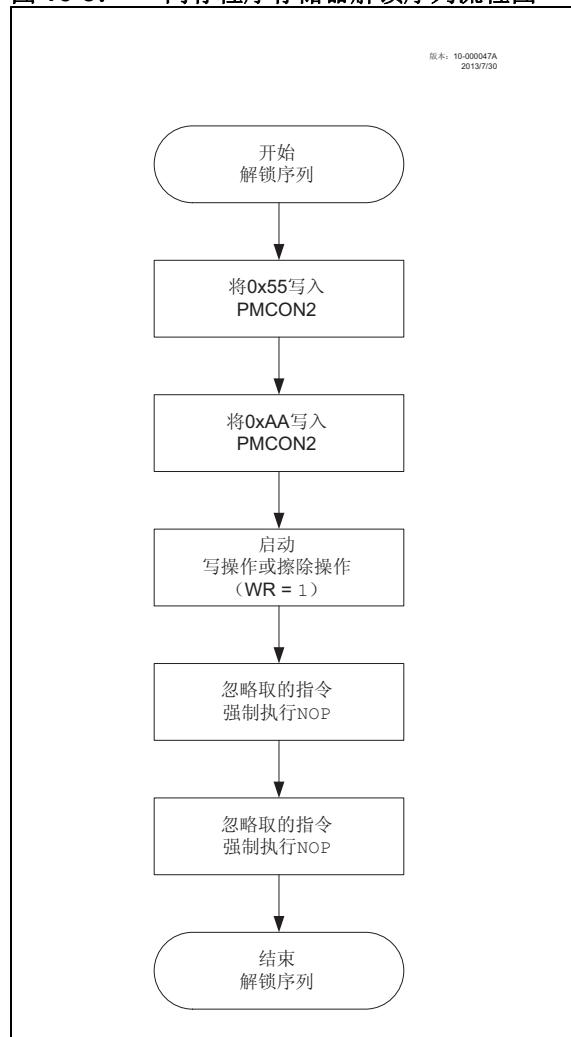
解锁序列包含以下步骤：

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

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

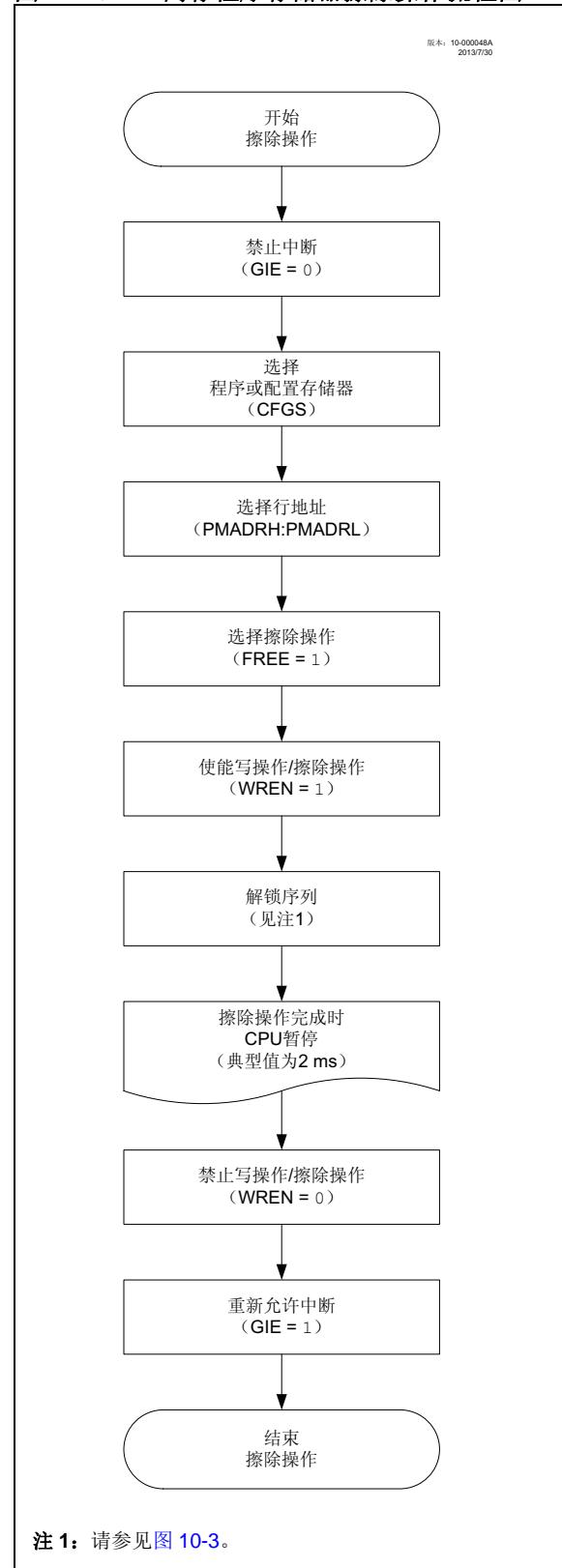
在执行代码时，程序存储器只能按行进行擦除。要擦除某行，请执行以下步骤：

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 WR 位置 1 指令之后放置两条 NOP 指令。处理器将在 2 ms（典型值）的擦除时间内暂停内部操作。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，处理器将继续执行 PMCON1 写指令之后的第三条指令。

图 10-4：闪存程序存储器擦除操作流程图



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    ADDR,W          ; Load lower 8 bits of erase address boundary  
MOVWF   PMADRL  
MOVF    ADDR,W          ; Load upper 6 bits of erase address boundary  
MOVWF   PMADRH  
BCF      PMCON1,CFG5    ; Not configuration space  
BSF      PMCON1,FREE    ; Specify an erase operation  
BSF      PMCON1,WREN    ; Enable writes  
  
MOVlw    55h             ; Start of required sequence to initiate erase  
MOVWF   PMCON2          ; Write 55h  
MOVlw    0AAh            ;  
MOVWF   PMCON2          ; Write AAh  
BSF      PMCON1,WR       ; Set WR bit to begin erase  
NOP                 ; NOP instructions are forced as processor starts  
NOP                 ; row erase of program memory.  
;  
; The processor stalls until the erase process is complete  
; after erase processor continues with 3rd instruction  
  
BCF      PMCON1,WREN    ; Disable writes  
BSF      INTCON,GIE     ; Enable interrupts
```

序列
必需的

10.2.4 写入闪存程序存储器

要编程程序存储器，请执行以下步骤：

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

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

程序存储器每次可以写入一个或多个字。每次可以写入的最多字数等于写锁存器的数量。更多详细信息，请参见图 10-5（使用 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，写序列将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 7，直到除最后一个写锁存器之外的所有写锁存器中都装入数据为止。
9. 将 PMCON1 寄存器的 LWLO 位清零。当 PMCON1 寄存器的 LWLO 位为 0 时，写序列会启动对闪存程序存储器的写操作。
10. 将要写入的程序存储器数据装入 PMDATH:PMDATL 寄存器对。
11. 执行解锁序列（[第 10.2.2 节“闪存解锁序列”](#)）。整个程序存储器锁存器的内容现在会被写入闪存程序存储器中。

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

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

图 10-5： 使用 32 个写锁存器对闪存程序存储器进行块写操作

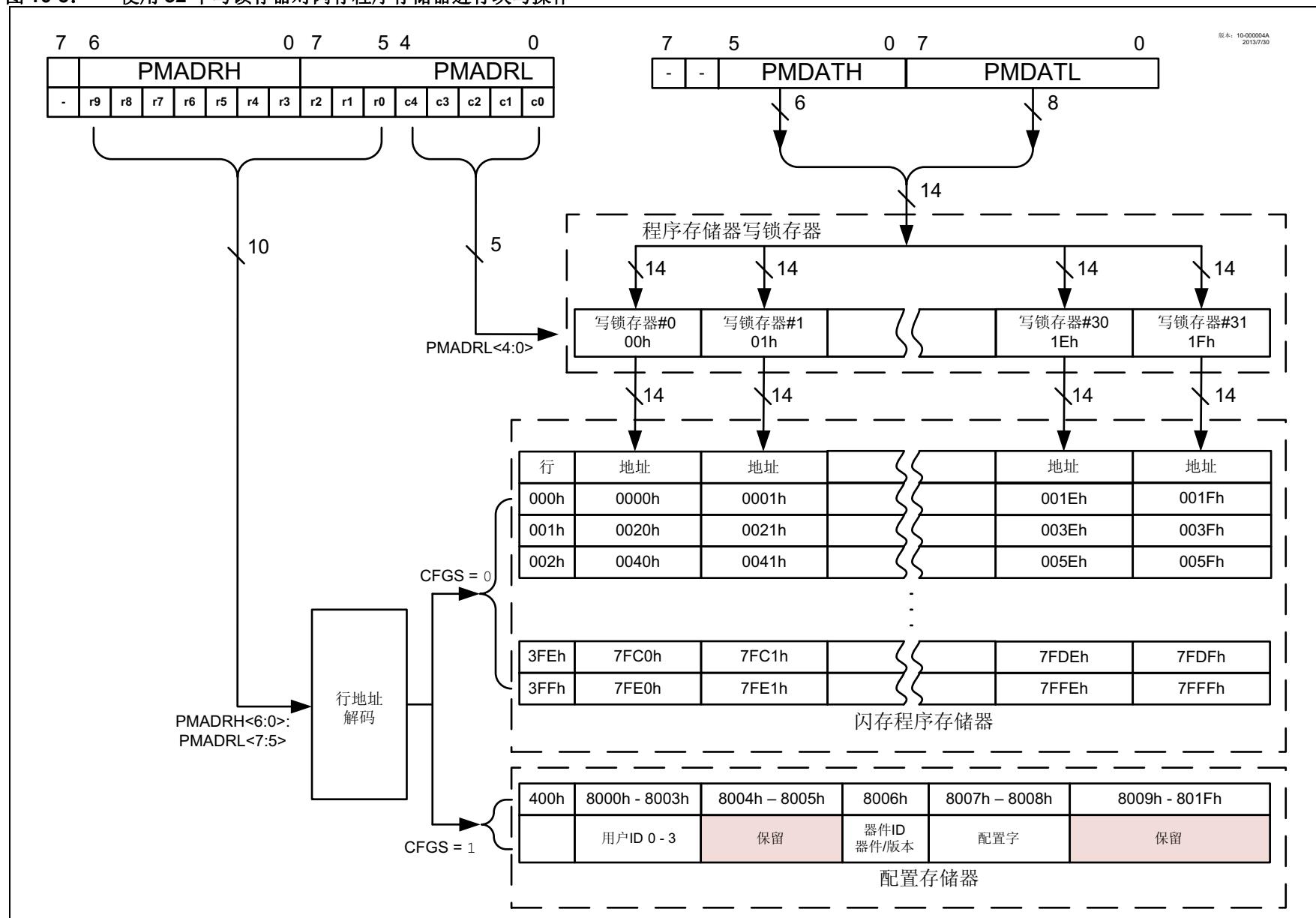
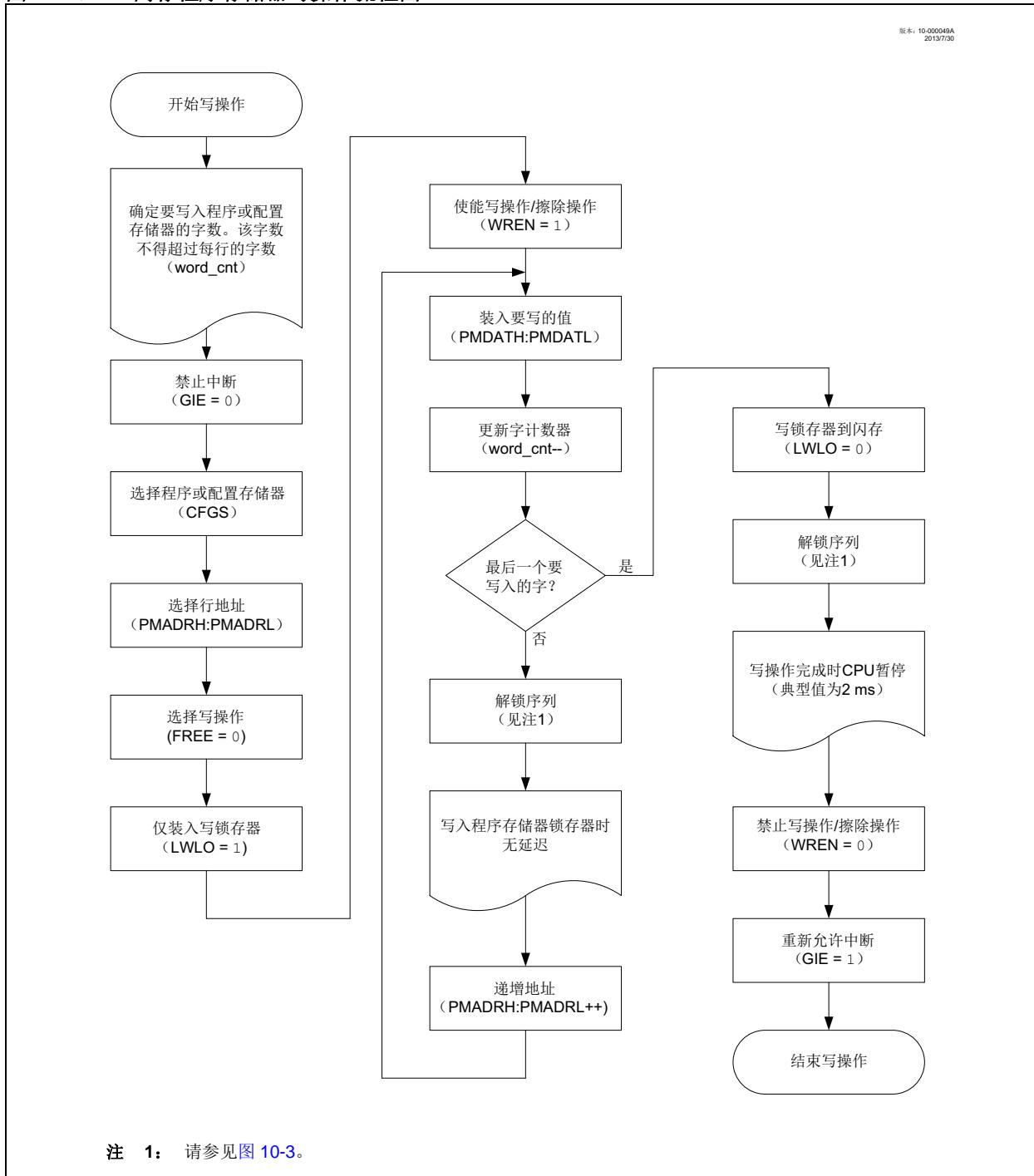


图 10-6：闪存程序存储器写操作流程图



PIC16(L)F1574/5/8/9

例 10-3：写入闪存程序存储器

```
; This write routine assumes the following:  
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR  
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,  
; stored in little endian format  
; 3. A valid starting address (the Least Significant bits = 00000) is loaded in ADDRH:ADDRL  
; 4. ADDRH and ADDRDL 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     ADDRDL,W       ;  
    MOVWF    PMADRL         ;  
    MOVLW    LOW DATA_ADDR  ; Load initial data address  
    MOVWF    FSR0L          ;  
    MOVLW    HIGH DATA_ADDR ; Load initial data address  
    MOVWF   FSR0H          ;  
    BCF      PMCON1,CFG5    ; Not configuration space  
    BSF      PMCON1,WREN    ; Enable writes  
    BSF      PMCON1,LWLO    ; Only Load Write Latches  
  
LOOP  
    MOVIW   FSR0++          ; Load first data byte into lower  
    MOVWF   PMDATL          ;  
    MOVIW   FSR0++          ; Load second data byte into upper  
    MOVWF   PMDATH          ;  
  
    MOVF     PMADRL,W       ; Check if lower bits of address are '00000'  
    XORLW   0x1F            ; Check if we're on the last of 32 addresses  
    ANDLW   0x1F            ;  
    BTFSC   STATUS,Z        ; Exit if last of 32 words,  
    GOTO    START_WRITE     ;  
  
    MOVLW    55h             ; Start of required write sequence:  
    MOVWF   PMCON2          ; Write 55h  
    MOVLW    0AAh            ;  
    MOVWF   PMCON2          ; Write AAh  
    BSF     PMCON1,WR        ; Set WR bit to begin write  
    NOP                 ; NOP instructions are forced as processor  
                        ; loads program memory write latches  
    NOP                 ;  
  
    INCF    PMADRL,F        ; Still loading latches Increment address  
    GOTO    LOOP             ; Write next latches  
  
START_WRITE  
    BCF      PMCON1,LWLO    ; No more loading latches - Actually start Flash program  
                        ; memory write  
  
    MOVLW    55h             ; Start of required write sequence:  
    MOVWF   PMCON2          ; Write 55h  
    MOVLW    0AAh            ;  
    MOVWF   PMCON2          ; Write AAh  
    BSF     PMCON1,WR        ; Set WR bit to begin write  
    NOP                 ; NOP instructions are forced as processor writes  
                        ; all the program memory write latches simultaneously  
    NOP                 ; to program memory.  
                        ; After NOPs, the processor  
                        ; stalls until the self-write process is complete  
                        ; after write processor continues with 3rd instruction  
    BCF      PMCON1,WREN    ; Disable writes  
    BSF      INTCON,GIE      ; Enable interrupts
```

必需的序列

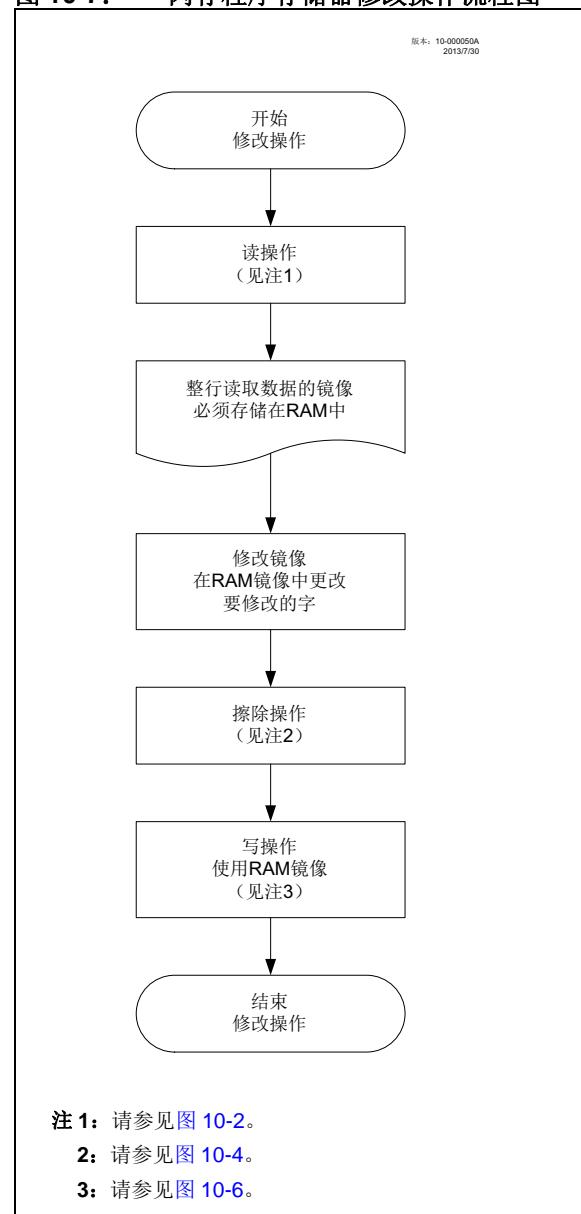
必需的序列

10.3 修改闪存程序存储器

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

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

图 10-7：闪存程序存储器修改操作流程图



10.4 用户 ID、器件 ID 和配置字访问

当 PMCON1 寄存器中的 CFGS = 1 时，可以访问用户 ID、器件 ID/ 版本 ID 和配置字，而不能访问程序存储器。这是在 PC<15> = 1 时指向的区域，但并不是所有地址都可以访问。可能存在不同的读写访问权限。请参见表 10-2。

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

表 10-2： 用户 ID、器件 ID 和配置字访问 (CFGs = 1)

地址	功能	读访问	写访问
8000h-8003h	用户 ID	是	是
8006h/8005h	器件 ID/ 版本 ID	是	否
8007h-8008h	配置字 1 和 2	是	否

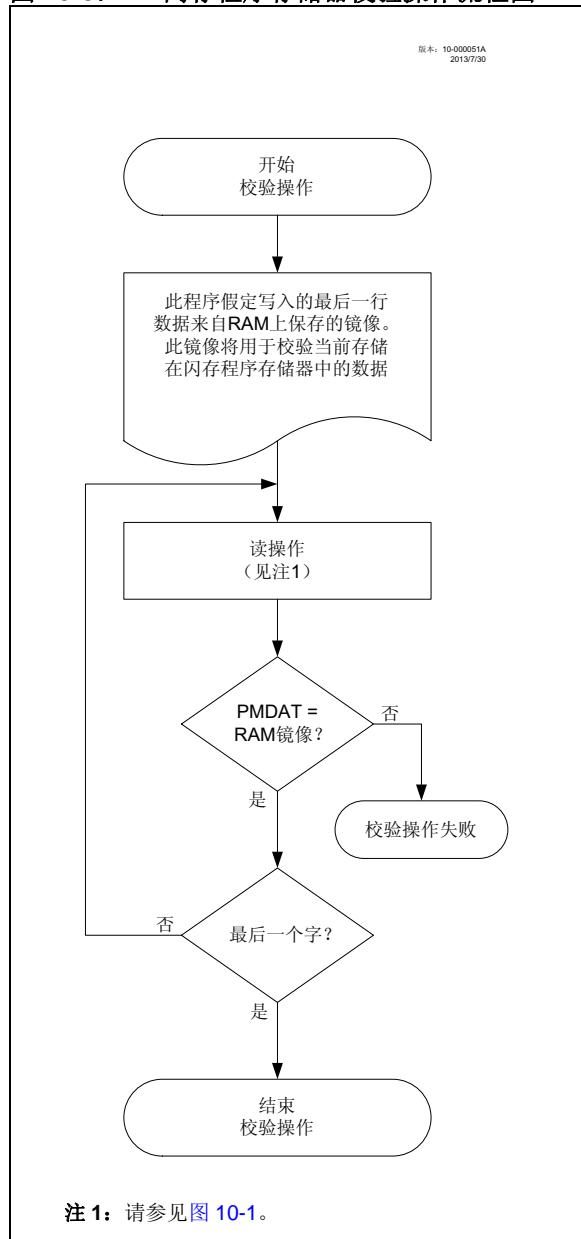
例 10-4： 配置字和器件 ID 访问

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



PIC16(L)F1574/5/8/9

10.6 寄存器定义：闪存程序存储器控制

寄存器 10-1： PMDATL：程序存储器数据低字节寄存器

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

PMDAT<7:0>: 程序存储器最低有效位的读 / 写值

寄存器 10-2： PMDATH：程序存储器数据高字节寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	PMDAT<13:8>					
bit 7							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6

未实现：读为 0

bit 5-0

PMDAT<13:8>: 程序存储器最高有效位的读 / 写值

寄存器 10-3： PMADRL：程序存储器地址低字节寄存器

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

PMADR<7:0>: 指定程序存储器地址的最低有效位

寄存器 10-4： PMADRH：程序存储器地址高字节寄存器

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—(1)	PMADR<14:8>						
bit 7							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/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。

寄存器 10-5: PMCON1: 程序存储器控制 1 寄存器

U-1	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-x/q ⁽²⁾	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
__(1)	CFGs	LWLO ⁽³⁾	FREE	WRERR	WREN	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	CFGs: 配置选择位 1 = 访问配置、用户 ID 和器件 ID 寄存器 0 = 访问闪存程序存储器
bit 5	LWLO: 仅装入写锁存器位 ⁽³⁾ 1 = 在下一条 WR 命令时仅装入 / 更新所寻址的程序存储器写锁存器 0 = 在下一条 WR 命令时装入 / 更新所寻址的程序存储器写锁存器，并启动对于所有程序存储器写锁存器的写操作
bit 4	FREE: 程序闪存擦除使能位 1 = 在下一条 WR 命令时执行擦除操作（完成时由硬件清零） 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 ⁽²⁾ 1 = 指示试图执行不合法的编程或擦除序列，或表示序列意外终止（试图将 WR 位置 1（写入 1）时自动将该位置 1）。 0 = 编程或擦除操作正常完成。
bit 2	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，一旦操作完成，该位即由硬件清零。 只能用软件将 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。

PIC16(L)F1574/5/8/9

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	TMROIF	INTF	IOCIF	86
PMCON1	— ⁽¹⁾	CFG5	LWLO	FREE	WRERR	WREN	WR	RD	115
PMCON2	程序存储器控制寄存器 2								116
PMADRL	PMADRL<7:0>								114
PMADRH	— ⁽¹⁾	PMADRH<6:0>							
PMDATL	PMDATL<7:0>								114
PMDATH	—	—	PMDATH<5:0>						114

图注: — = 未实现位, 读为 0。闪存程序存储器不使用阴影单元。

注 1: 未实现, 读为 1。

表 10-4: 与闪存程序存储器相关的配置字汇总

名称	Bit	Bit -7	Bit -6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	56
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		—	FOSC<1:0>		
CONFIG2	13:8	—	—	LVP	DEBUG	LPBOR	BORV	STVREN	PLLEN	57
	7:0	—	—	—	—	—	PPS1WAY	WRT<1:0>		

图注: — = 未实现位, 读为 0。闪存程序存储器不使用阴影单元。

11.0 I/O 端口

每个端口都有三个标准工作寄存器。这些寄存器是：

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

一些端口可能还具有以下一个或多个额外的寄存器。这些寄存器是：

- ANSEL_x（模拟选择）
- 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 11-1：每款器件可用的端口

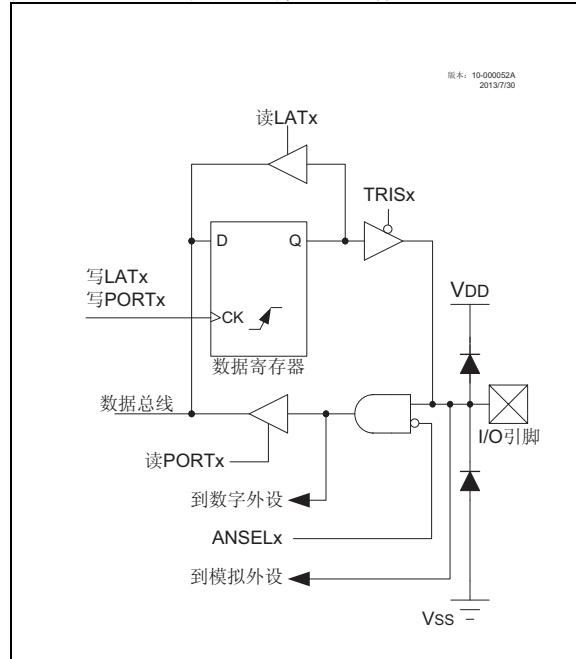
器件	PORTA	PORTB	PORTC
PIC16(L)F1574	●		●
PIC16(L)F1575	●		●
PIC16(L)F1578	●	●	●
PIC16(L)F1579	●	●	●

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

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

支持模拟输入的端口具有相关的 ANSEL_x 寄存器。当某个 ANSEL 位置 1 时，与该位相关的数字输入缓冲器会被禁止。禁止输入缓冲器可以防止该引脚上介于逻辑高电平和低电平之间的模拟信号电平在逻辑输入电路上产生过大的电流。图 11-1 给出了通用 I/O 端口的简化模型，没有给出与其他外设的接口。

图 11-1：通用 I/O 端口的工作原理



11.1 PORTA 寄存器

11.1.1 数据寄存器

PORTA 是一个 6 位宽的双向端口。对应的数据方向寄存器是 TRISA（[寄存器 11-2](#)）。将 TRISA 某位置 1 (= 1) 时，会将 PORTA 的相应引脚设为输入（即，禁止输出驱动器）。将 TRISA 某位清零 (= 0) 时，会将 PORTA 的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选定的引脚）。RA3 是个例外，仅可作为输入引脚，其 TRIS 位总是读为 1。[例 11-1](#) 显示了如何初始化 I/O 端口。

读 PORTA 寄存器（[寄存器 11-1](#)）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读 - 修改 - 写操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口数据锁存器（LATA）。

11.1.2 方向控制

TRISA 寄存器（[寄存器 11-2](#)）用于控制 PORTA 引脚输出驱动器，即使它们被用作模拟输入。当引脚用作模拟输入时，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。

11.1.3 漏极开路控制

ODCONA 寄存器（[寄存器 11-6](#)）用于控制端口的漏极开路功能。每个引脚的漏极开路操作可以独立进行选择。当 ODCONE 位置 1 时，相应的端口输出会变为只能灌入电流的漏极开路驱动器。当 ODCONA 位清零时，相应的端口输出引脚是能够拉出和灌入电流的标准推挽驱动器。

11.1.4 压摆率控制

SLRCONA 寄存器（[寄存器 11-7](#)）用于控制每个端口引脚的压摆率选项。每个端口引脚的压摆率控制可以独立进行选择。当 SLRCONA 位置 1 时，相应端口引脚驱动器的压摆率会受到限制。当 SLRCONA 位清零时，相应端口引脚驱动器的压摆率将为最大可能值。

11.1.5 输入阈值控制

INLVLA 寄存器（[寄存器 11-8](#)）用于控制每个可用 PORTA 输入引脚的输入电压阈值。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。输入阈值对于确定 PORTA 寄存器的读取值很重要，同时它也是发生电平变化中断的临界电压（如果使能该功能）。关于阈值电压的更多信息，请参见表 [27-4](#)。

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

11.1.6 模拟控制

ANSELA 寄存器（[寄存器 11-4](#)）用于将 I/O 引脚的输入模式配置为模拟。将相应的 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 11-1： 初始化 PORTA

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

BANKSEL  PORTA          ;
CLRF      PORTA          ; Init PORTA
BANKSEL  LATA           ; Data Latch
CLRF      LATA           ;
BANKSEL  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
```

11.1.7 PORTA 功能和输出优先级

在发生复位之后，每个引脚都默认为端口锁存器数据。其他功能都使用外设引脚选择逻辑进行选择。更多信息，请参见第 12.0 节“[外设引脚选择（PPS）模块](#)”。外设引脚选择列表中未列出模拟输入功能，例如ADC输入。这些输入在使用 ANSELA 寄存器将 I/O 引脚设置为模拟模式时有效。当引脚处于模拟模式时，数字输出功能可以继续控制该引脚。

PIC16(L)F1574/5/8/9

11.2 寄存器定义: PORTA

寄存器 11-1: PORTA: PORTA 寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 11-2: TRISA: PORTA 三态寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 引脚配置为输入 (三态)

0 = PORTA 引脚配置为输出

bit 3 未实现: 读为 1

bit 2-0 TRISA<2:0>: PORTA 三态控制位

1 = PORTA 引脚配置为输入 (三态)

0 = PORTA 引脚配置为输出

注 1: 未实现, 读为 1。

寄存器 11-3: LATA: PORTA 数据锁存器寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 未实现: 读为 0

bit 5-4 **LATA<5:4>**: RA<5:4> 输出锁存值位⁽¹⁾

bit 3 未实现: 读为 0

bit 2-0 **LATA<2:0>**: RA<2:0> 输出锁存值位⁽¹⁾

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

寄存器 11-4: ANSELA: PORTA 模拟选择寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-5 未实现: 读为 0

bit 4 **ANSA4**: 将 RA4 引脚选择为模拟或数字功能

1 = 模拟输入。引脚配置为模拟输入⁽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。

PIC16(L)F1574/5/8/9

寄存器 11-5: WPUA: PORTA 弱上拉寄存器

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 **WPUEN** 位, 从而使能各个上拉。

2: 如果引脚被配置为输出, 则自动禁止弱上拉器件。

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

寄存器 11-6: ODCONA: PORTA 漏极开路控制寄存器

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 未实现: 读为 0

bit 5-4 **ODA<5:4>**: PORTA 漏极开路使能位

对于各个 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)

寄存器 11-7: SLRCONA: PORTA 压摆率控制寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 未实现: 读为 0

bit 5-4 **SLRA<5:4>**: PORTA 压摆率使能位
对于各个 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 11-8: INLVLA: PORTA 输入电平控制寄存器

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	INLVLA5	INLVLA4	INLVLA3 ⁽¹⁾	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

注 1: 仅当未选择 MCLR 功能时, INLVLA3 位才选择此引脚上的输入类型。选择了 MCLR 功能时, 此引脚的输入类型将为 ST。

PIC16(L)F1574/5/8/9

表 11-2：与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSEL A	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	121
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	123
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	121
ODCONA	—	—	ODA5	ODA4	—	ODA2	ODA1	ODA0	122
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			178
POR TA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	120
SLRCONA	—	—	SLRA5	SLRA4	—	SLRA2	SLRA1	SLRA0	123
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	120
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	122

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

注 1：未实现，读为 1。

表 11-3：与 PORTA 相关的配置字汇总

名称	Bit	Bit -7	Bit -6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG1	13:8	—	—	—	—	CLKOUTEN	BOREN<1:0>		—	56
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			

图注：— = 未实现位，读为 0。PORTA 不使用阴影单元。

11.3 PORTB 寄存器 (仅限 PIC16(L)F1578/9)

PORTB 是一个 4 位宽的双向端口。对应的数据方向寄存器是 TRISB (寄存器 11-10)。将 TRISB 某位置 1 (=1) 时，会将 PORTB 的相应引脚设为输入 (即，使相应的输出驱动器呈高阻态模式)。将 TRISB 某位清零 (=0) 时，会将 PORTB 的相应引脚设为输出 (即，使能输出驱动器并将输出锁存器中的内容输出到选中引脚)。

例 11-1 显示了如何初始化 I/O 端口。

读 PORTB 寄存器 (寄存器 11-9) 将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读 - 修改 - 写操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口数据锁存器 (LATB)。

11.3.1 方向控制

TRISB 寄存器 (寄存器 11-10) 用于控制 PORTB 引脚输出驱动器，即使它们被用作模拟输入。当引脚用作模拟输入时，用户应确保 TRISB 寄存器中的各位保持置 1。配置为模拟输入的 I/O 引脚总是读为 0。

11.3.2 漏极开路控制

ODCONB 寄存器 (寄存器 11-14) 用于控制端口的漏极开路功能。每个引脚的漏极开路操作可以独立进行选择。当 ODCONB 位置 1 时，相应的端口输出会变为只能灌入电流的漏极开路驱动器。当 ODCONB 位清零时，相应的端口输出引脚是能够拉出和灌入电流的标准推挽驱动器。

11.3.3 压摆率控制

SLRCONB 寄存器 (寄存器 11-15) 用于控制每个端口引脚的压摆率选项。每个端口引脚的压摆率控制可以独立进行选择。当 SLRCONB 位置 1 时，相应端口引脚驱动器的压摆率会受到限制。当 SLRCONB 位清零时，相应端口引脚驱动器的压摆率将为最大可能值。

11.3.4 输入阈值控制

INLVLB 寄存器 (寄存器 11-16) 用于控制每个可用 PORTB 输入引脚的输入电压阈值。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。输入阈值对于确定 PORTB 寄存器的读取值很重要，同时它也是发生电平变化中断的临界电压 (如果使能该功能)。关于阈值电压的更多信息，请参见表 27-4。

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

11.3.5 模拟控制

ANSELB 寄存器 (寄存器 11-12) 用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSELB 位设置为高电平将使引脚上的所有数字读操作都读为 0，并允许引脚上的模拟功能正确工作。

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

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

11.3.6 PORTB 功能和输出优先级

在发生复位之后，每个引脚都默认认为端口锁存器数据。其他功能都使用外设引脚选择逻辑进行选择。更多信息，请参见第 12.0 节“外设引脚选择 (PPS) 模块”。外设引脚选择列表中未列出模拟输入功能，例如 ADC 和运放输入。这些输入在使用 ANSELB 寄存器将 I/O 引脚设置为模拟模式时有效。当引脚处于模拟模式时，数字输出功能可以继续控制该引脚。

PIC16(L)F1574/5/8/9

11.4 寄存器定义: PORTB

寄存器 11-9: PORTB: PORTB 寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 **RB<7:4>**: PORTB 通用 I/O 引脚位⁽¹⁾

1 = 端口引脚电平 $\geq V_{IH}$

0 = 端口引脚电平 $\leq V_{IL}$

bit 3-0 未实现: 读为 0

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

寄存器 11-10: TRISB: PORTB 三态寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 **TRISB<7:4>**: PORTB 三态控制位

1 = PORTB 引脚配置为输入 (三态)

0 = PORTB 引脚配置为输出

bit 3-0 未实现: 读为 0

寄存器 11-11: LATB: PORTB 数据锁存器寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 **LATB<7:4>**: PORTB 输出锁存值⁽¹⁾

bit 3-0 未实现: 读为 0

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

寄存器 11-12: ANSELB: PORTB 模拟选择寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6

未实现: 读为 0

bit 5-4

ANSB<5:4>: 将 RB<5:4> 引脚选择为模拟或数字功能

0 = 数字 I/O。引脚配置为端口或数字特殊功能。

1 = 模拟输入。引脚配置为模拟输入⁽¹⁾。数字输入缓冲器被禁止。

bit 3-0

未实现: 读为 0

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

寄存器 11-13: WPUB: PORTB 弱上拉寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4

WPUB<7:4>: 弱上拉寄存器位

1 = 使能上拉

0 = 禁止上拉

bit 3-0

未实现: 读为 0

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

2: 如果引脚被配置为输出, 则自动禁止弱上拉器件。

PIC16(L)F1574/5/8/9

寄存器 11-14: ODCONB: PORTB 漏极开路控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
ODB7	ODB6	ODB5	ODB4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 **ODB<7:4>: PORTB 漏极开路使能位**

对于各个 RB<7:4> 引脚

1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)

0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

bit 3-0 未实现: 读为 0

寄存器 11-15: SLRCONB: PORTB 压摆率控制寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 **SLRB<7:4>: PORTB 压摆率使能位**

对于各个 RB<7:4> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率将为最大值

bit 3-0 未实现: 读为 0

寄存器 11-16: INLVLB: PORTB 输入电平控制寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 **INLVLB<7:4>: PORTB 输入电平选择位**

对于各个 RB<7:4> 引脚

1 = 对于端口读操作和电平变化中断, 使用 ST 输入

0 = 对于端口读操作和电平变化中断, 使用 TTL 输入

bit 3-0 未实现: 读为 0

表 11-4：与 PORTB 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	127
INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	128
LATB	LATB7	LATB6	LATB5	LATB4	—	—	—	—	126
ODCONB	ODB7	ODB6	ODB5	ODB4	—	—	—	—	128
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	126
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—	128
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	128
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	127

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

11.5 PORTC 寄存器

11.5.1 数据寄存器

在 PIC16(L)F1574/5 器件中，PORTC 是 6 位宽的双向端口，在 PIC16(L)F1578/9 器件中，则为 8 位宽的双向端口。对应的数据方向寄存器是 TRISC（[寄存器 11-18](#)）。将 TRISC 某位置 1 (= 1) 时，会将 PORTC 的相应引脚设为输入（即，使相应的输出驱动器呈高阻态模式）。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中引脚）。[例 11-1](#) 显示了如何初始化 I/O 端口。

读 PORTC 寄存器（[寄存器 11-17](#)）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是读 - 修改 - 写操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口数据锁存器（LATC）。

11.5.2 方向控制

TRISC 寄存器（[寄存器 11-18](#)）用于控制 PORTC 引脚输出驱动器，即使它们被用作模拟输入。当引脚用作模拟输入时，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。

11.5.3 输入阈值控制

INLVLC 寄存器（[寄存器 11-24](#)）用于控制每个可用 PORTC 输入引脚的输入电压阈值。用户可以选择施密特触发器 CMOS 阈值或 TTL 兼容阈值。输入阈值对于确定 PORTC 寄存器的读取值很重要，同时它也是发生电平变化中断（如果使能该功能）的临界电压。关于阈值电压的更多信息，请参见表 27-4。

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

11.5.4 漏极开路控制

ODCONC 寄存器（[寄存器 11-22](#)）用于控制端口的漏极开路功能。每个引脚的漏极开路操作可以独立进行选择。当 ODCONC 位置 1 时，相应的端口输出会变为只能灌入电流的漏极开路驱动器。当 ODCONC 位清零时，相应的端口输出引脚是能够拉出和灌入电流的标准推挽驱动器。

11.5.5 压摆率控制

SLRCONC 寄存器（[寄存器 11-23](#)）用于控制每个端口引脚的压摆率选项。每个端口引脚的压摆率控制可以独立进行选择。当 SLRCONC 位置 1 时，相应端口引脚驱动器的压摆率会受到限制。当 SLRCONC 位清零时，相应端口引脚驱动器的压摆率将为最大可能值。

11.5.6 模拟控制

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

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

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

11.5.7 PORTC 功能和输出优先级

在发生复位之后，每个引脚都默认为端口锁存器数据。其他功能都使用外设引脚选择逻辑进行选择。更多信息，请参见[第 12.0 节“外设引脚选择 \(PPS\) 模块”](#)。

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

11.6 寄存器定义：PORTC

寄存器 11-17：PORTC：PORTC 寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RC7 ⁽²⁾	RC6 ⁽²⁾	RC5	RC4	RC3	RC2	RC1	RC0
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **RC<7:0>**: PORTC 通用 I/O 引脚位^(1,2)

1 = 端口引脚电平 $\geq V_{IH}$

0 = 端口引脚电平 $\leq V_{IL}$

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

2：RC<7:6> 仅在 PIC16(L)F1578/9 上可用。

寄存器 11-18：TRISC：PORTC 三态寄存器

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **TRISC<7:0>**: PORTC 三态控制位⁽¹⁾

1 = PORTC 引脚配置为输入（三态）

0 = PORTC 引脚配置为输出

注 1：TRISC<7:6> 仅在 PIC16(L)F1578/9 上可用。

寄存器 11-19：LATC：PORTC 数据锁存器寄存器

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

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

注 1：LATC<7:6> 仅在 PIC16(L)F1578/9 上可用。

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

PIC16(L)F1574/5/8/9

寄存器 11-20: ANSELC: PORTC 模拟选择寄存器

R/W-1/1	R/W-1/1	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSC7 ⁽²⁾	ANSC6 ⁽²⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 **ANSC<7:6>**: 将 RC<7:6> 引脚选择为模拟或数字功能 ^(1, 2)

0 = 数字 I/O。引脚配置为端口或数字特殊功能。

1 = 模拟输入。引脚配置为模拟输入 ⁽¹⁾。数字输入缓冲器被禁止。

bit 5-4 未实现: 读为 0

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

0 = 数字 I/O。引脚配置为端口或数字特殊功能。

1 = 模拟输入。引脚配置为模拟输入 ⁽¹⁾。数字输入缓冲器被禁止。

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

2: ANSC<7:6> 仅在 PIC16(L)F1578/9 上可用。

寄存器 11-21: WPUC: PORTC 弱上拉寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUC7 ⁽³⁾	WPUC6 ⁽³⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **WPUC<7:0>**: 弱上拉寄存器位 ⁽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 WPUEN 位, 从而使能各个上拉。

2: 如果引脚被配置为输出, 则自动禁止弱上拉器件。

3: WPUC<7:6> 仅在 PIC16(L)F1578/9 上可用。

寄存器 11-22: ODCONC: PORTC 漏极开路控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **ODC<7:0>: PORTC 漏极开路使能位⁽¹⁾**

对于各个 RC<7:0> 引脚

1 = 端口引脚作为漏极开路驱动器工作 (仅灌电流)

0 = 端口引脚作为标准推挽驱动器工作 (拉电流和灌电流)

注 1: ODC<7:6> 仅在 PIC16(L)F1578/9 上可用。

寄存器 11-23: SLRCONC: PORTC 压摆率控制寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **SLRC<7:0>: PORTC 压摆率使能位⁽¹⁾**

对于各个 RC<7:0> 引脚

1 = 端口引脚的压摆率受到限制

0 = 端口引脚的压摆率将为最大值

注 1: SLRC<7:6> 仅在 PIC16(L)F1578/9 上可用。

寄存器 11-24: INLVLC: PORTC 输入电平控制寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **INLVLC<7:0>: PORTC 输入电平选择位⁽¹⁾**

对于各个 RC<7:0> 引脚

1 = 对于端口读操作和电平变化中断, 使用 ST 输入

0 = 对于端口读操作和电平变化中断, 使用 TTL 输入

注 1: INLVLC<7:6> 仅在 PIC16(L)F1578/9 上可用。

PIC16(L)F1574/5/8/9

表 11-5：与 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	132
INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	133
LATC	LATC7 ⁽¹⁾	LATC6 ⁽¹⁾	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	131
ODCCONC	ODC7 ⁽¹⁾	ODC6 ⁽¹⁾	ODC5	ODC4	ODC3	ODC2	ODC1	ODC0	133
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			178
PORTC	RC7 ⁽¹⁾	RC6 ⁽¹⁾	RC5	RC4	RC3	RC2	RC1	RC0	131
SLRCONC	SLRC7 ⁽¹⁾	SLRC6 ⁽¹⁾	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	133
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	131
WPUC	WPUC7 ⁽¹⁾	WPUC6 ⁽¹⁾	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	132

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

注 1： 仅限 PIC16(L)F1578/9。

12.0 外设引脚选择 (PPS) 模块

外设引脚选择 (PPS) 模块可将外设输入和输出连接到器件 I/O 引脚。选择范围只包含数字信号。所有模拟输入和输出都保持固定连接到其指定引脚。输入和输出选择是独立的，如图 12-1 所示。

12.1 PPS 输入

每个外设均具有一个用于选择外设输入的 PPS 寄存器。输入包括器件引脚。

多个外设可以同时使用同一个源工作。端口读操作总是返回引脚电平，无论外设 PPS 选择如何。如果某个引脚也具有相关的模拟功能，则必须清零该引脚的 ANSEL 位才会使能数字输入缓冲器。

虽然每个外设都具有自己的 PPS 输入选择寄存器，但每个外设的选择范围是相同的，如 [寄存器 12-1](#) 中所示。

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

12.2 PPS 输出

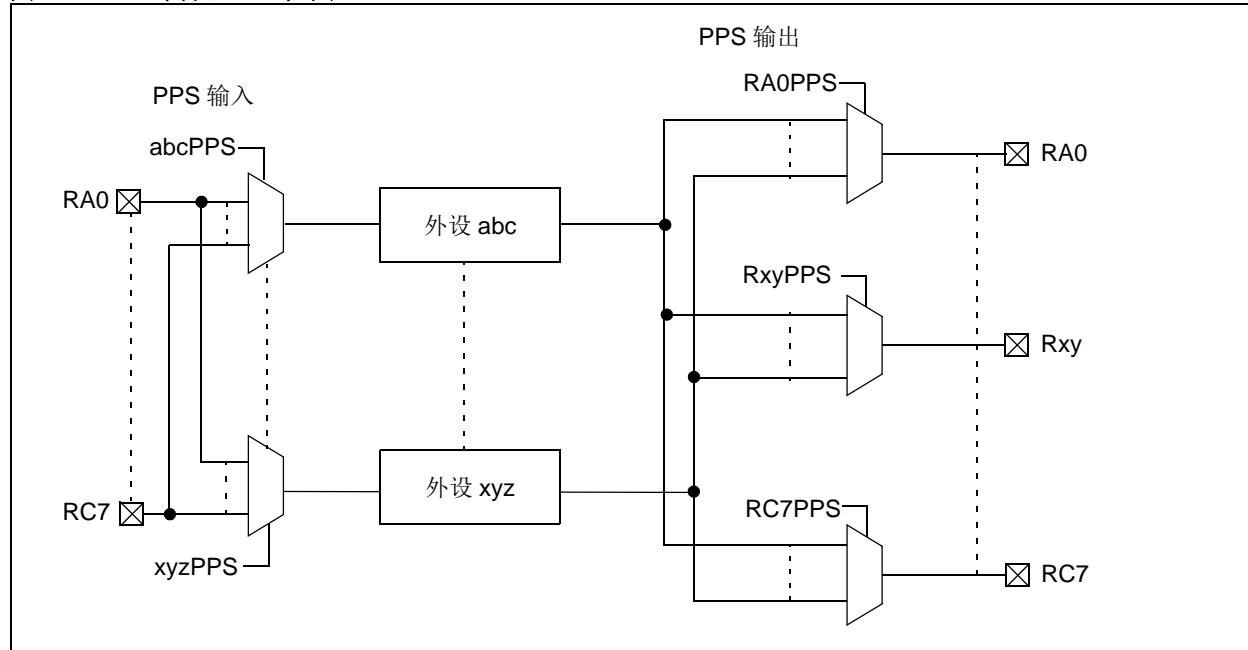
每个 I/O 引脚均具有一个用于选择引脚输出源的 PPS 寄存器。除了少数例外，与该引脚相关的端口 TRIS 控制都会保持对引脚输出驱动器的控制权。作为外设操作的一部分而控制引脚输出驱动器的外设会根据需要改写 TRIS 控制。这些外设包括：

- EUSART (同步操作)
- MSSP (I²C)
- CWG (自动关断)

虽然每个引脚都具有自己的 PPS 外设选择寄存器，但每个引脚的选择范围是相同的，如 [寄存器 12-2](#) 中所示。

注： “Rxy” 符号是引脚标识符的占位符。例如，RA0PPS。

图 12-1：简化 PPS 框图



12.3 双向引脚

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

- USART（同步操作）
- MSSP（I²C）

注： I²C 默认输入引脚与 I²C 和 SMBus 兼容，并且是器件上具有这种兼容性的仅有引脚。

12.4 PPS 锁定

PPS 包含了一种锁定模式，在该模式下可以锁定所有输入和输出选择，以防止意外更改。PPS 选择通过将 PPSLOCK 寄存器的 PPSLOCKED 位置 1 来锁定。置 1 和清零该位需要一个特殊的序列作为额外的预防措施，以防止意外更改。[例 12-1](#) 给出了置 1 和清零 PPSLOCKED 位的示例。

例 12-1：PPS 锁定 / 解锁序列

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

12.5 PPS 永久锁定

可以通过将 PPS1WAY 配置位置 1 来永久锁定 PPS。当该位置 1 时，PPSLOCKED 位只能在器件复位后清零和置 1 一次。这使用户可以清零 PPSLOCKED 位，以便可以在初始化期间进行输入和输出选择。当 PPSLOCKED 位在进行所有选择之后置 1 时，它会保持置 1，直到下一个器件复位事件之后才能清零。

12.6 休眠期间的操作

休眠不会影响 PPS 输入和输出选择。

12.7 复位的影响

器件上电复位（POR）会将所有 PPS 输入和输出选择清除为其默认值。所有其他复位会将这些选择保留不变。默认输入选择如[表 12-1](#) 所示。

12.8 寄存器定义：PPS 输入选择

寄存器 12-1: xxxPPS: 外设 xxx 输入选择

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

q = 值取决于外设

bit 7-5 未实现：读为 0

bit 4-3 xxxPPS<4:3>：外设 xxx 输入端口选择位

11 = 保留。不要使用。

10 = 外设输入为 PORTC

01 = 外设输入为 PORTB⁽²⁾

00 = 外设输入为 PORTA

bit 2-0 xxxPPS<2:0>：外设 xxx 输入位选择位⁽¹⁾

111 = 外设输入来自 PORTx 的 Bit 7 (Rx7)

110 = 外设输入来自 PORTx 的 Bit 6 (Rx6)

101 = 外设输入来自 PORTx 的 Bit 5 (Rx5)

100 = 外设输入来自 PORTx 的 Bit 4 (Rx4)

011 = 外设输入来自 PORTx 的 Bit 3 (Rx3)

010 = 外设输入来自 PORTx 的 Bit 2 (Rx2)

001 = 外设输入来自 PORTx 的 Bit 1 (Rx1)

000 = 外设输入来自 PORTx 的 Bit 0 (Rx0)

注 1：关于 xxxPPS 寄存器列表和复位值，请参见表 12-1。

2：仅限 PIC16(L)F1578/9。

寄存器 12-2: RxyPPS: 引脚 Rxy 输出源选择寄存器

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-5 未实现：读为 0

bit 4-0 RxyPPS<4:0>：引脚 Rxy 输出源选择位

选择编码决定端口引脚上的输出信号。

关于选择编码，请参见表 12-2

PIC16(L)F1574/5/8/9

寄存器 12-3: PPSLOCK: PPS 锁定寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-1 未实现: 读为 0

bit 0 **PPSLOCKED:** PPS 锁定位

1 = PPS 已锁定。无法更改 PPS 选择。

0 = PPS 未锁定。可以更改 PPS 选择。

表 12-1: PPS 输入寄存器复位值

外设	xxxPPS 寄存器	默认引脚选择		复位值 (xxxPPS<4:0>)	
		PIC16(L)F1578/9	PIC16(L)F1574/5	PIC16(L)F1578/9	PIC16(L)F1574/5
电平变化中断	INTPPS	RA2	RA2	00010	00010
Timer0 时钟	T0CKIPPS	RA2	RA2	00010	00010
Timer1 时钟	T1CKIPPS	RA5	RA5	00101	00101
Timer1 门控	T1GPPS	RA4	RA4	00100	00100
CWG1	CWG1INPPS	RA2	RA2	00010	00010
EUSART RX	RXPPS	RB5	RC5	01101	10101
EUSART CK	CKPPS	RB7	RC4	01111	10100
ADC 自动转换触发器	ADCACTPPS	RC4	RC4	10100	10100

示例: ADCACTPPS = 0x14 会选择 RC4 作为 ADC 自动转换触发器输入。

表 12-2: 可供外设用于输出的端口 (2)

RxyPPS<3:0>	输出信号	PIC16(L)F1578/9			PIC16(L)F1574/5	
		PORTA	PORTB	PORTC	PORTA	PORTC
1111	保留	—	—	—	—	—
1110	保留	—	—	—	—	—
1101	保留	—	—	—	—	—
1100	保留	—	—	—	—	—
1011	保留	—	—	—	—	—
1010	DT ⁽¹⁾	•	•	•	•	•
1001	TX/CK ⁽¹⁾	•	•	•	•	•
1000	CWG1OUTB ⁽¹⁾	•	•	•	•	•
0111	CWG1OUTA ⁽¹⁾	•	•	•	•	•
0110	PWM4_out	•	•	•	•	•
0101	PWM3_out	•	•	•	•	•
0100	PWM2_out	•	•	•	•	•
0011	PWM1_out	•	•	•	•	•
0010	sync_C2OUT	•	•	•	•	•
0001	sync_C1OUT	•	•	•	•	•
0000	LATxy	•	•	•	•	•

注 1: TRIS 控制由外设根据需要改写。

2: 不受支持的外设将输出 0。

表 12-3：与 PPS 模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	138
INTPPS	—	—	—		INTPPS<4:0>				137
T0CKIPPS	—	—	—		T0CKIPPS<4:0>				137
T1CKIPPS	—	—	—		T1CKIPPS<4:0>				137
T1GPPS	—	—	—		T1GPPS<4:0>				137
CWG1INPPS	—	—	—		CWG1INPPS<4:0>				137
RXPPS	—	—	—		RXPPS<4:0>				137
CKPPS	—	—	—		CKPPS<4:0>				137
ADCACTPPS	—	—	—		ADCACTPPS<4:0>				137
RA0PPS	—	—	—	—	RA0PPS<3:0>				137
RA1PPS	—	—	—	—	RA1PPS<3:0>				137
RA2PPS	—	—	—	—	RA2PPS<3:0>				137
RA4PPS	—	—	—	—	RA4PPS<3:0>				137
RA5PPS	—	—	—	—	RA5PPS<3:0>				137
RB4PPS ⁽¹⁾	—	—	—	—	RB4PPS<3:0>				137
RB5PPS ⁽¹⁾	—	—	—	—	RB5PPS<3:0>				137
RB6PPS ⁽¹⁾	—	—	—	—	RB6PPS<3:0>				137
RB7PPS ⁽¹⁾	—	—	—	—	RB7PPS<3:0>				137
RC0PPS	—	—	—	—	RC0PPS<3:0>				137
RC1PPS	—	—	—	—	RC1PPS<3:0>				137
RC2PPS	—	—	—	—	RC2PPS<3:0>				137
RC3PPS	—	—	—	—	RC3PPS<3:0>				137
RC4PPS	—	—	—	—	RC4PPS<3:0>				137
RC5PPS	—	—	—	—	RC5PPS<3:0>				137
RC6PPS ⁽¹⁾	—	—	—	—	RC6PPS<3:0>				137
RC7PPS ⁽¹⁾	—	—	—	—	RC7PPS<3:0>				137

注 1：仅限 PIC16(L)F1578/9。

13.0 电平变化中断

PORTA、PORTB⁽¹⁾ 和 PORTC 引脚可以配置为作为电平变化中断（Interrupt-On-Change, IOC）引脚工作。中断可以通过检测具有上升沿或下降沿的信号而产生。任意一个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。

注 1: PORTB 仅在 PIC16(L)F1578/9 上可用。

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 IOCxF 寄存器的相关位置 1。要允许引脚检测下降沿，需要将 IOCxF 寄存器的相关位置 1。

通过将 IOCxF 和 IOCxF 寄存器的相关位均置 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、IOCBF 和 IOCCF 寄存器中的 IOCAF_x、IOCBF_x 和 IOCCF_x 位是对应于相关端口的电平变化中断引脚的状态标志。如果在正确使能的引脚上检测到期望的边沿，则对应于该引脚的状态标志会置 1，并且如果 IOCIE 位也置 1，则还会产生中断。INTCON 寄存器的 IOCIF 位会反映所有 IOCAF_x、IOCBF_x 和 IOCCF_x 位的状态。

13.4 清零中断标志

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

为了确保在清零标志时不会丢失任何已检测到的边沿，应当仅执行可屏蔽已知更改位的与操作。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
Andw IOCAF, f
```

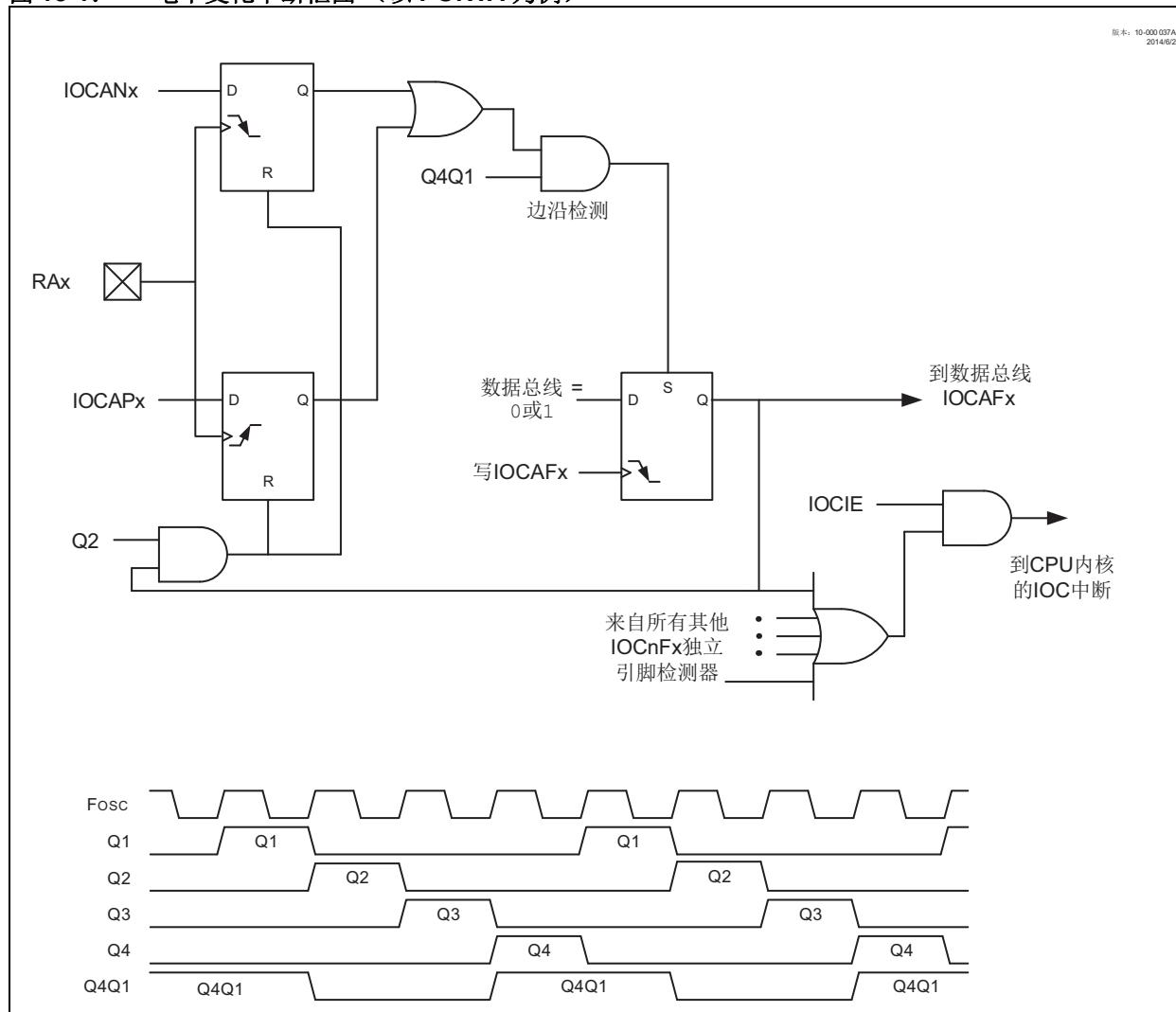
13.5 休眠模式下的操作

如果 IOCIE 位置 1，电平变化中断序列会将器件从休眠模式唤醒。

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

PIC16(L)F1574/5/8/9

图 13-1：电平变化中断框图（以 PORTA 为例）



13.6 寄存器定义：电平变化中断控制

寄存器 13-1： IOCAP: PORTA 电平变化中断正边沿寄存器

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>:** PORTA 电平变化中断正边沿使能位

1 = 使能由引脚电平的正边沿触发电平变化中断。 IOCAF_x 位和 IOCIF 标志将在检测到相应边沿时置 1。
0 = 禁止相关引脚的电平变化中断。

寄存器 13-2： IOCAN: PORTA 电平变化中断负边沿寄存器

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>:** PORTA 电平变化中断负边沿使能位

1 = 使能由引脚电平的负边沿触发电平变化中断。 IOCAF_x 位和 IOCIF 标志将在检测到相应边沿时置 1。
0 = 禁止相关引脚的电平变化中断。

寄存器 13-3： IOCAF: PORTA 电平变化中断标志寄存器

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>:** PORTA 电平变化中断标志位

1 = 在相关引脚上检测到有效的电平变化。

在 IOCAF_x = 1，并在 RAx 上检测到上升沿时置 1，或者在 IOCAN_x = 1，并在 RAx 上检测到下降沿时置 1。
0 = 未检测到电平变化，或者用户清除了检测到的电平变化。

PIC16(L)F1574/5/8/9

寄存器 13-4: IOCBP: PORTB 电平变化中断正边沿寄存器⁽¹⁾

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4

IOCBP<7:4>: PORTB 电平变化中断正边沿使能位

1 = 使能由引脚电平的正边沿触发电平变化中断。 IOCBFx 位和 IOCIF 标志将在检测到相应边沿时置 1。

0 = 禁止相关引脚的电平变化中断。

bit 3-0

未实现: 读为 0

注 1: PORTB 功能仅在 PIC16(L)F1578/9 器件上可用。

寄存器 13-5: IOCBN: PORTB 电平变化中断负边沿寄存器⁽¹⁾

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4

IOCBN<7:4>: PORTB 电平变化中断负边沿使能位

1 = 使能由引脚电平的负边沿触发电平变化中断。 IOCBFx 位和 IOCIF 标志将在检测到相应边沿时置 1。

0 = 禁止相关引脚的电平变化中断。

bit 3-0

未实现: 读为 0

注 1: PORTB 功能仅在 PIC16(L)F1578/9 器件上可用。

寄存器 13-6: IOCBI: PORTB 电平变化中断标志寄存器⁽¹⁾

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

HS = 硬件置 1 位

bit 7-4

IOCBF<7:4>: PORTB 电平变化中断标志位

1 = 在相关引脚上检测到有效的电平变化。

在 IOCBPx = 1, 并在 RBx 上检测到上升沿时置 1, 或者在 IOCBNx = 1, 并在 RBx 上检测到下降沿时置 1。

0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

bit 3-0

未实现: 读为 0

注 1: PORTB 功能仅在 PIC16(L)F1578/9 器件上可用。

寄存器 13-7: IOCCP: PORTC 电平变化中断正边沿寄存器⁽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
IOCCP7 ⁽¹⁾	IOCCP6 ⁽¹⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

IOCCP<7:0>: PORTC 电平变化中断正边沿使能位⁽¹⁾

1 = 使能由引脚电平的正边沿触发电平变化中断。IOCCFx 位和 IOCIF 标志将在检测到相应边沿时置 1。

0 = 禁止相关引脚的电平变化中断。

注 1: IOCCP<7:6> 仅在 PIC16(L)F1578/9 器件上可用。

寄存器 13-8: IOCCN: PORTC 电平变化中断负边沿寄存器⁽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
IOCCN7 ⁽¹⁾	IOCCN6 ⁽¹⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

IOCCN<7:0>: PORTC 电平变化中断负边沿使能位⁽¹⁾

1 = 使能由引脚电平的负边沿触发电平变化中断。IOCCFx 位和 IOCIF 标志将在检测到相应边沿时置 1。

0 = 禁止相关引脚的电平变化中断。

注 1: IOCCN<7:6> 仅在 PIC16(L)F1578/9 器件上可用。

寄存器 13-9: IOCCF: PORTC 电平变化中断标志寄存器⁽¹⁾

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

HS = 硬件置 1 位

bit 7-0

IOCCF<7:0>: PORTC 电平变化中断标志位⁽¹⁾

1 = 在相关引脚上检测到有效的电平变化。

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

0 = 未检测到电平变化, 或者用户清除了检测到的电平变化。

注 1: IOCCF<7:6> 仅在 PIC16(L)F1578/9 器件上可用。

PIC16(L)F1574/5/8/9

表 13-1：与电平变化中断相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	121
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	143
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	143
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	143
IOCBP ⁽²⁾	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	144
IOCBN ⁽²⁾	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	144
IOCBF ⁽²⁾	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	144
IOCCP	IOCCP7 ⁽²⁾	IOCCP6 ⁽²⁾	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	145
IOCCN	IOCCN7 ⁽²⁾	IOCCN6 ⁽²⁾	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	145
IOCCF	IOCCF7 ⁽²⁾	IOCCF6 ⁽²⁾	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	145
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	120
TRISC	TRISC7 ⁽²⁾	TRISC7 ⁽²⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	131

图注：— = 未实现位，读为 0。电平变化中断不使用阴影单元。

注 1：未实现，读为 1。

2：仅限 PIC16(L)F1578/9。

14.0 固定参考电压 (FVR)

固定参考电压 (FVR) 是稳定的参考电压，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：

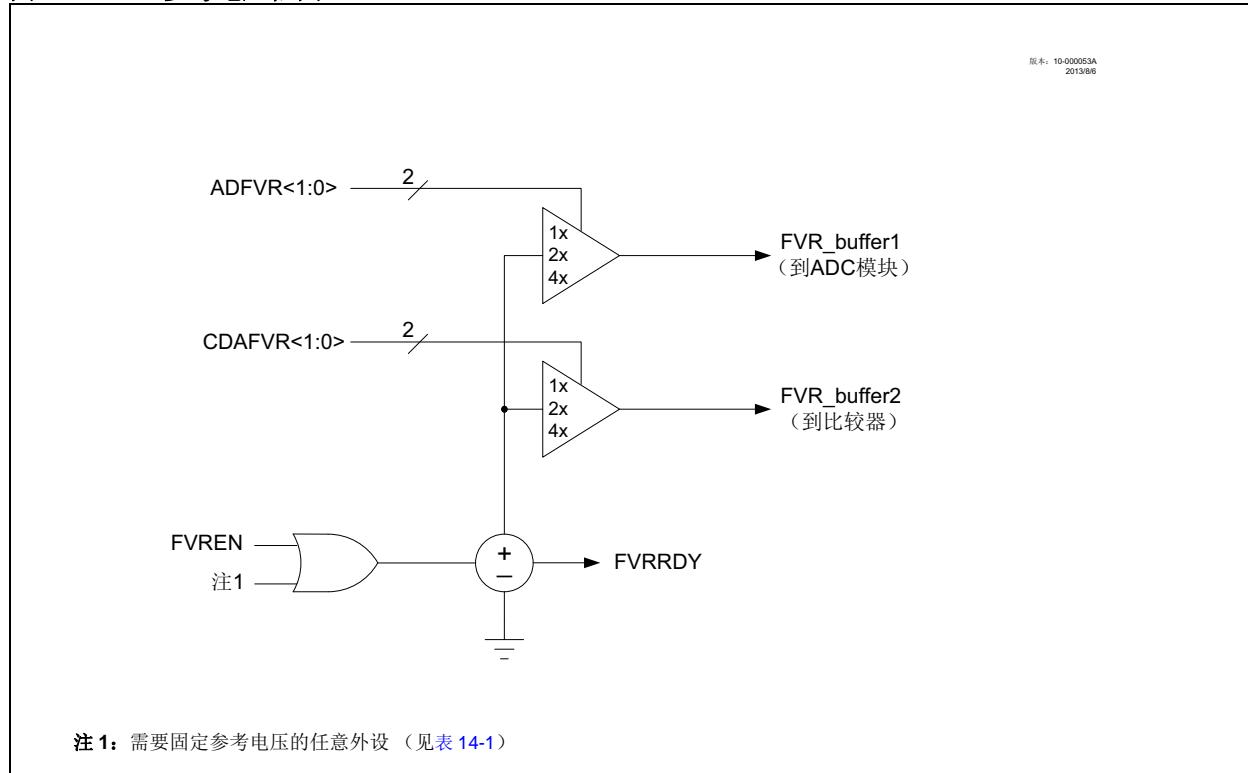
- ADC 输入通道
- 比较器的同相输入
- 比较器的反向输入

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。

图 14-1：参考电压框图



注 1：需要固定参考电压的任意外设（见表 14-1）

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，应通过清零缓冲器增益选择位来关闭 FVR 缓冲器。

14.2 FVR 稳定周期

FVR 可以通过将 FVRCON 寄存器的 FVREN 位置 1 来使能。

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。

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模式。
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 PIC16F1574/5/8/9 器件	处于休眠模式时，器件依靠低功耗稳压器运行。

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 = 固定参考电压输出已就绪 0 = 固定参考电压输出未就绪或未使能
bit 5	TSEN: 温度指示器使能位 ⁽³⁾ 1 = 使能温度指示器 0 = 禁止温度指示器
bit 4	TSRNG: 温度指示器范围选择位 ⁽³⁾ 1 = $V_{OUT} = V_{DD} - 4V_T$ (高电压范围) 0 = $V_{OUT} = V_{DD} - 2V_T$ (低电压范围)
bit 3-2	CDAFVR<1:0>: 比较器 FVR 缓冲器增益选择位 ⁽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 PIC16F1574/5/8/9 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>			149

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管芯工作温度的温度电路。电路的工作温度范围介于 -40°C 和 $+85^{\circ}\text{C}$ 之间。其输出是与器件温度成正比的电压。温度指示器的输出在内部与器件 ADC 连接。

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

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

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

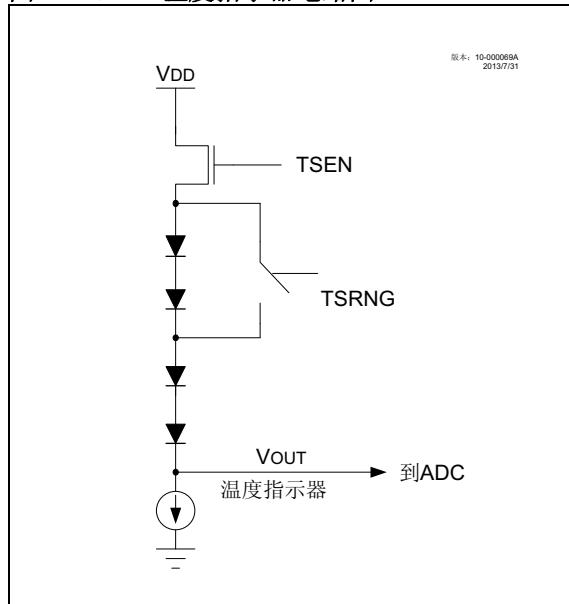
温度检测电路与固定参考电压 (FVR) 模块集成在一起。更多信息，请参见第 14.0 节“固定参考电压 (FVR)”。

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

电路可以在高电压范围或低电压范围内工作。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，但器件个体之间的一致性较低。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}

当温度电路在低电压范围内工作时，器件可以在规范范围内的任意工作电压下工作。

当温度电路在高电压范围内工作时，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 $200 \mu\text{s}$ 。

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>			118

图注： 温度指示器模块不使用阴影单元。

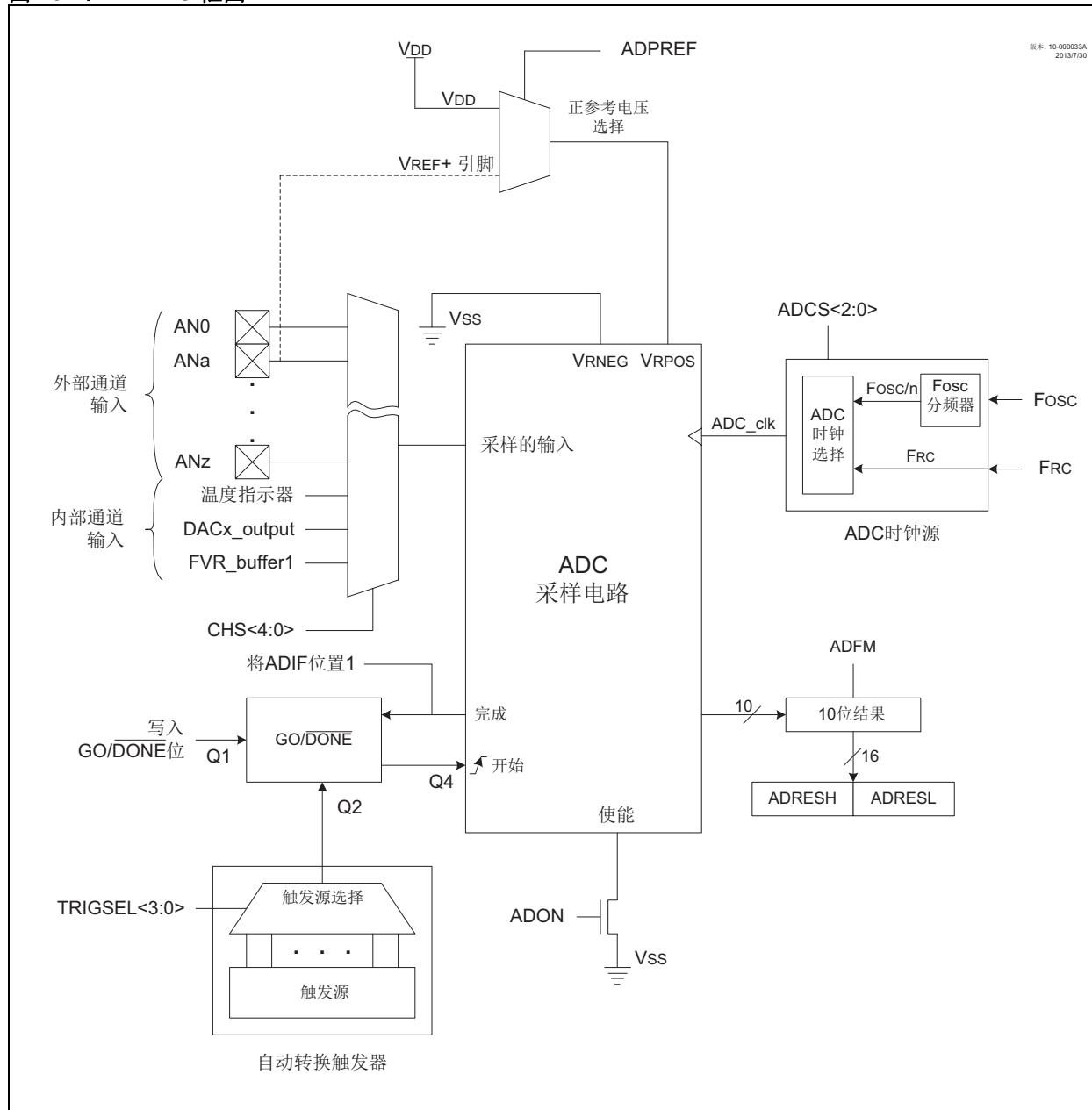
16.0 模数转换器 (ADC) 模块

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，这些模拟输入通过多路开关连接到同一个采样保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制结果，并将转换结果存储在 ADC 结果寄存器 (ADRESH:ADRESL 寄存器对) 中。图 16-1 给出了 ADC 的框图。

可通过软件方式选择内部产生的电压或外部提供的电压作为 ADC 参考电压。

ADC 可在转换完成时产生中断。该中断可用于将器件从休眠模式唤醒。

图 16-1：ADC 框图



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 11.0 节 “I/O 端口”](#)。

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

16.1.2 通道选择

有最多 15 个通道可供选择：

- AN<7:0> 引脚（仅限 PIC16(L)F1574/5）
- AN<11:0> 引脚（仅限 PIC16(L)F1578/9）
- 温度指示器
- DAC1_output
- FVR_buffer1

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 位来选择。正参考电压源可以是：

- VREF+ 引脚
- VDD
- FVR_buffer1

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)

完成一个位的转换所需的时间定义为 TAD。一次完整的 10 位转换需要 11.5 个 TAD 周期，如图 16-2 所示。

为正确转换，必须满足合适的 TAD 规范。关于更多信息，请参见[第 27.0 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 ADC 结果。

PIC16(L)F1574/5/8/9

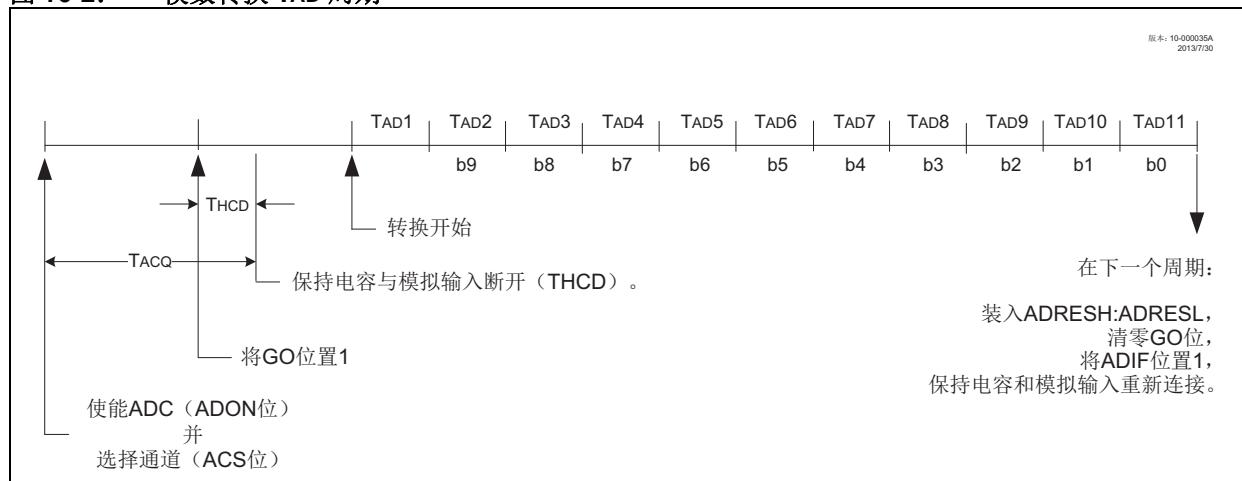
表 16-1: ADC 时钟周期 (TAD) 与器件工作频率关系表

ADC 时钟周期 (TAD)		器件频率 (Fosc)				
ADC 时钟源	ADCS<2:0>	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	100 ns	125 ns	250 ns	500 ns	2.0 μ s
Fosc/4	100	200 ns	250 ns	500 ns	1.0 μ s	4.0 μ s
Fosc/8	001	400 ns	500 ns	1.0 μ s	2.0 μ s	8.0 μ s
Fosc/16	101	800 ns	1.0 μ s	2.0 μ s	4.0 μ s	16.0 μ s
Fosc/32	010	1.6 μ s	2.0 μ s	4.0 μ s	8.0 μ s	32.0 μ s
Fosc/64	110	3.2 μ s	4.0 μ s	8.0 μ s	16.0 μ s	64.0 μ s
FRC	x11	1.0-6.0 μ s				

图注: 阴影单元超出了建议范围。

注: 使用 FRC 时钟源时的 TAD 周期可以处于规定范围内 (见 TAD 参数)。使用基于 FOSC 的时钟源时的 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



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。

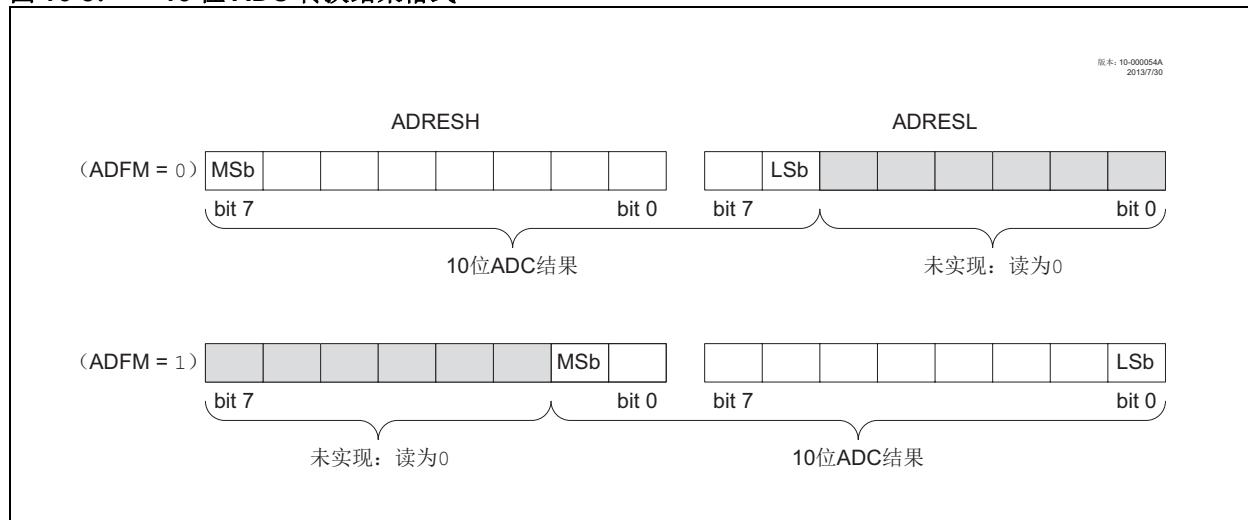
器件工作或休眠时都可产生该中断。如果器件处于休眠模式，该中断会唤醒器件。从休眠模式唤醒时，总是执行紧跟 SLEEP 指令后的下一条指令。如果用户尝试从休眠模式唤醒，并继续主代码执行，则 PIE1 寄存器的 ADIE 位和 INTCON 寄存器的 PEIE 位都必须置 1，INTCON 寄存器的 GIE 位必须清零。如果这 3 个位全都置 1，执行将切换到中断服务程序。

16.1.6 结果格式

10 位 ADC 转换结果可通过两种格式提供：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 位必须置 1。将 ADCON0 寄存器的 GO/DONE 位置 1 会启动模数转换。

注: 不应在启动ADC的同一条指令中将GO/DONE 位置1。请参见[第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为复位状态。因此, ADC 模块被关闭, 任何待处理的转换被终止。

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模式唤醒。如果禁止了 ADC 中断, 尽管 ADON 位仍保持置 1, 但转换完成后 ADC 模块将关闭。

ADC 时钟源不是 FRC 时, 尽管 ADON 位仍保持置 1, 但 SLEEP 指令会导致当前转换中止, ADC 模块被关闭。

16.2.5 自动转换触发器

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

自动转换触发源使用ADCON2寄存器的TRIGSEL<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。

PWM 模块可以使用两种方式来触发 ADC, 即直接通过 PWMx_OF_match 或通过全部 4 个匹配信号产生的中断。请参见[第 23.0 节“16 位脉宽调制 \(PWM\) 模块”](#)。如果选择中断, PWMxINTE 中的每个已允许中断都会触发转换。更多信息, 请参见图 16-4。

关于自动转换源, 请参见[表 16-2](#)。

图 16-4: 16 位 PWM 中断框图

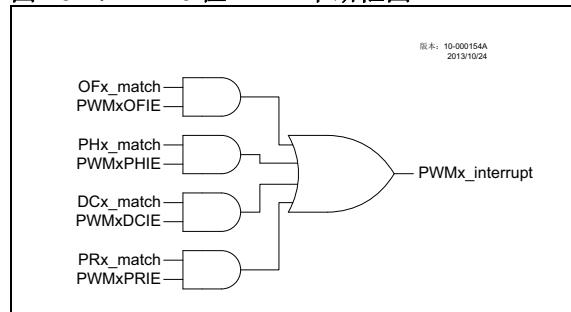


表 16-2: 自动转换源

源外设	信号名称
Timer0	T0_overflow
Timer1	T1_overflow
Timer2	T2_match
比较器 C1	C1OUT_sync
比较器 C2	C2OUT_sync
PWM1	PWM1_OF_match
PWM1	PWM1_interrupt
PWM2	PWM2_OF_match
PWM2	PWM2_interrupt
PWM3	PWM3_OF_match
PWM3	PWM3_interrupt
PWM4	PWM4_OF_match
PWM4	PWM4_interrupt
ADC 触发信号	ADCACT
CWG 输入引脚	CWGIN

16.2.6 ADC 转换步骤

以下是使用 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）
 - 全局禁止弱上拉（见 OPTION_REG 寄存器）或单独禁止各个弱上拉（见相应的 WPUX 寄存器）
2. 配置 ADC 模块：
 - 选择 ADC 转换时钟
 - 配置参考电压
 - 选择 ADC 输入通道
 - 开启 ADC 模块
3. 配置 ADC 中断（可选）：
 - 清零 ADC 中断标志
 - 允许 ADC 中断
 - 允许外设中断
 - 允许全局中断 **(1)**
4. 等待所需采集时间 **(2)**。
5. 通过将 GO/DONE 位置 1 启动转换。
6. 通过以下方式之一等待 ADC 转换完成：
 - 查询 GO/DONE 位
 - 等待 ADC 中断（已允许中断）
7. 读取 ADC 结果。
8. 清零 ADC 中断标志（如果已允许中断则需要此操作）。

例 16-1：ADC 转换

```
;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
;are included.
;
BANKSEL    ADCON1           ;
MOVLW     B'11110000'        ;Right justify, FRC
                           ;oscillator
MOVWF     ADCON1           ;Vdd and Vss Vref+
BANKSEL    TRISA            ;
BSF       TRISA,0          ;Set RA0 to input
BANKSEL    ANSEL            ;
BSF       ANSEL,0          ;Set RA0 to analog
BANKSEL    WPUA             ;
BCF       WPUA,0           ;Disable weak
                           ;pull-up on RA0
BANKSEL    ADCON0           ;
MOVLW     B'00000001'        ;Select channel AN0
MOVWF     ADCON0           ;Turn ADC On
CALL      SampleTime        ;Acquisition delay
BSF       ADCON0,ADGO        ;Start conversion
BTFSR   ADCON0,ADGO        ;Is conversion done?
GOTO     $-1                ;No, test again
BANKSEL    ADRESH           ;
MOVF     ADRESH,W          ;Read upper 2 bits
MOVWF    RESULTHI          ;Store in GPR space
BANKSEL    ADRESL           ;
MOVF     ADRESL,W          ;Read lower 8 bits
MOVWF    RESULTLO          ;Store in GPR space
```

注 1：如果用户试图从休眠模式唤醒器件并恢复主代码执行，必须禁止全局中断。

2：请参见第 16.4 节“ADC 采集要求”。

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

00000 = AN0

00001 = AN1

00010 = AN2

00011 = AN3

00100 = AN4

00101 = AN5

00110 = AN6

00111 = AN7

01000 = AN8⁽⁴⁾

01001 = AN9⁽⁴⁾

01010 = AN10⁽⁴⁾

01011 = AN11⁽⁴⁾

01100 = 保留。不连接任何通道

•

•

•

11100 = 保留。不连接任何通道

11101 = 温度指示器 ⁽¹⁾

11110 = DAC（数模转换器）⁽²⁾

11111 = FVR（固定参考电压）缓冲器 1 输出 ⁽³⁾

bit 1 **GO/DONE**：ADC 转换状态位

1 = ADC 转换正在进行。将该位置 1 可启动 ADC 转换周期。

ADC 转换完成时，该位由硬件自动清零。

0 = ADC 转换已完成 / 未进行

bit 0 **ADON**：ADC 使能位

1 = 使能 ADC

0 = 禁止 ADC，不消耗工作电流

注 1：更多信息，请参见第 15.0 节“温度指示器模块”。

2：更多信息，请参见第 17.0 节“5 位数模转换器（DAC）模块”。

3：更多信息，请参见第 14.0 节“固定参考电压（FVR）”。

4：仅在 PIC16(L)F1578/9 器件上可用。

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

000 = Fosc/2

001 = Fosc/8

010 = Fosc/32

011 = FRC (由内部 RC 振荡器提供的时钟)

100 = Fosc/4

101 = Fosc/16

110 = Fosc/64

111 = FRC (由内部 RC 振荡器提供的时钟)

bit 3-2 未实现: 读为 0

bit 1-0 **ADPREF<1:0>:** ADC 正参考电压配置位

00 = VRPOS 连接到 VDD

01 = 保留

10 = VRPOS 连接到外部 VREF+ 引脚 ⁽¹⁾

11 = VRPOS 连接到内部固定参考电压 (FVR)

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 27.0 节 “电气规范”。

PIC16(L)F1574/5/8/9

寄存器 16-3: ADCON2: ADC 控制寄存器 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	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⁾

- 0000 = 未选择任何自动转换触发源
- 0001 = PWM1 —— PWM1_interrupt
- 0010 = PWM2 —— PWM2_interrupt
- 0011 = Timer0 —— T0_overflow⁽²⁾
- 0100 = Timer1 —— T1_overflow⁽²⁾
- 0101 = Timer2 —— T2_match
- 0110 = 比较器 C1 —— C1OUT_sync
- 0111 = 比较器 C2 —— C2OUT_sync
- 1000 = PWM1 —— PWM1_OF_match
- 1001 = PWM2 —— PWM2_OF_match
- 1010 = PWM3 —— PWM3_OF_match
- 1011 = PWM3 —— PWM3_interrupt
- 1100 = PWM4 —— PWM4_OF_match
- 1101 = PWM4 —— PWM4_interrupt
- 1110 = ADC 自动转换触发信号输入引脚
- 1111 = CWG 输入引脚

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。

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

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

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

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

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 **ADRES<1:0>: ADC 结果寄存器位**
10 位转换结果的低 2 位
bit 5-0 保留: 不要使用。

PIC16(L)F1574/5/8/9

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

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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

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（**C_{HOLD}**）完全充电至输入通道的电压。模拟输入模型如图 16-5 所示。模拟信号源阻抗（**R_S**）和内部采样开关阻抗（**R_{SS}**）直接影响电容 **C_{HOLD}** 的充电时间。采样开关阻抗（**R_{SS}**）随器件电压（**V_{DD}**）的变化而变化，参见图 16-5。模拟信号源的最大阻抗推荐值为 10 kΩ。

采集时间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 1,024 个步阶)。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 且外部阻抗为 10kW 5.0V VDD

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

Tc 值可以用以下公式近似计算：

$$\begin{aligned} V_{APPLIED} \left(1 - \frac{1}{(2^{n+1}) - 1} \right) &= V_{CHOLD} && ; [1] \text{ 充电到 } V_{CHOLD} \text{ (1/2 LSb 误差范围)} \\ V_{APPLIED} \left(1 - e^{-\frac{TC}{RC}} \right) &= V_{CHOLD} && ; [2] \text{ 响应 } V_{CHOLD} \text{ 充电到 } V_{APPLIED} \\ V_{APPLIED} \left(1 - e^{-\frac{TC}{RC}} \right) &= V_{APPLIED} \left(1 - \frac{1}{(2^{n+1}) - 1} \right) && ; \text{合并[1] 和[2]} \end{aligned}$$

注：其中，N = ADC 的位数。

求解 *TC*：

$$\begin{aligned} TC &= -C_{HOLD}(R_{IC} + R_{SS} + R_S) \ln(1/2047) \\ &= -12.5pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885) \\ &= 1.715\mu s \end{aligned}$$

因此：

$$\begin{aligned} TACQ &= 2\mu s + 1.715\mu s + [(50^{\circ}C - 25^{\circ}C)(0.05\mu s/^{\circ}C)] \\ &= 4.96\mu s \end{aligned}$$

注 1：因为参考电压（VRPOS）自行抵消，因此它对该公式没有影响。

2：充电保持电容（**C_{HOLD}**）在每次转换后不会放电。

3：建议模拟信号源的最大阻抗为 10 kΩ。此要求是为了符合引脚泄漏电流规范。

PIC16(L)F1574/5/8/9

图 16-5： 模拟输入模型

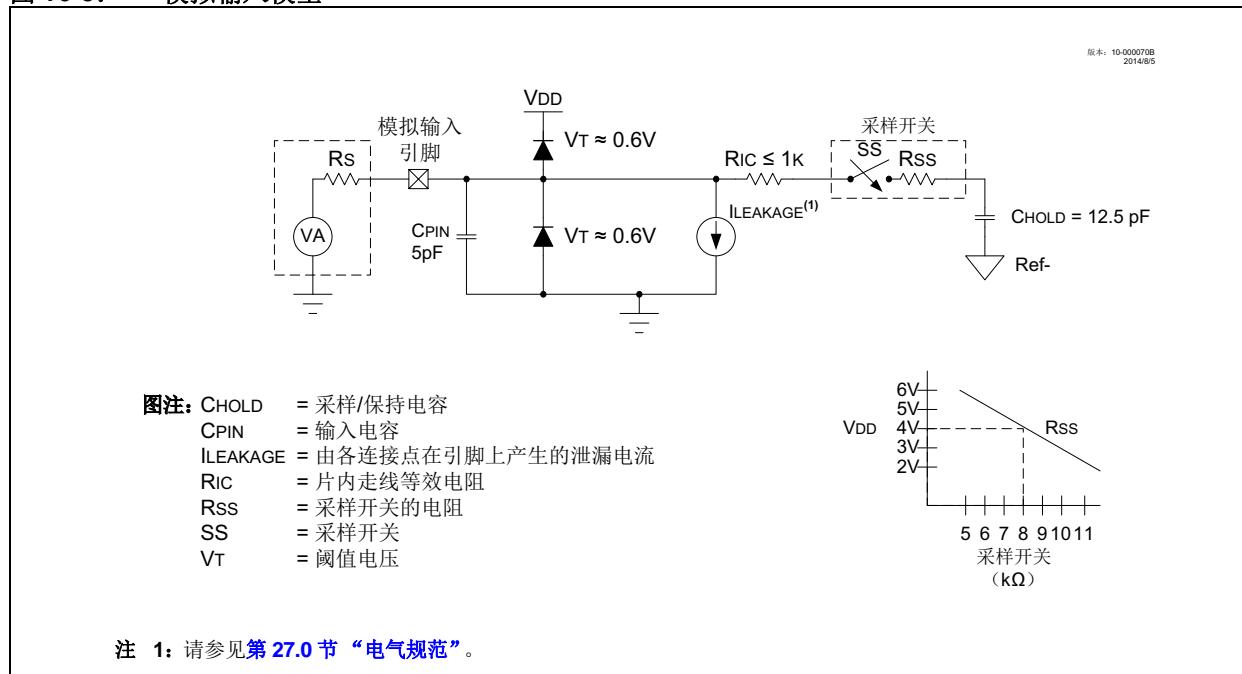


图 16-6： ADC 传递函数

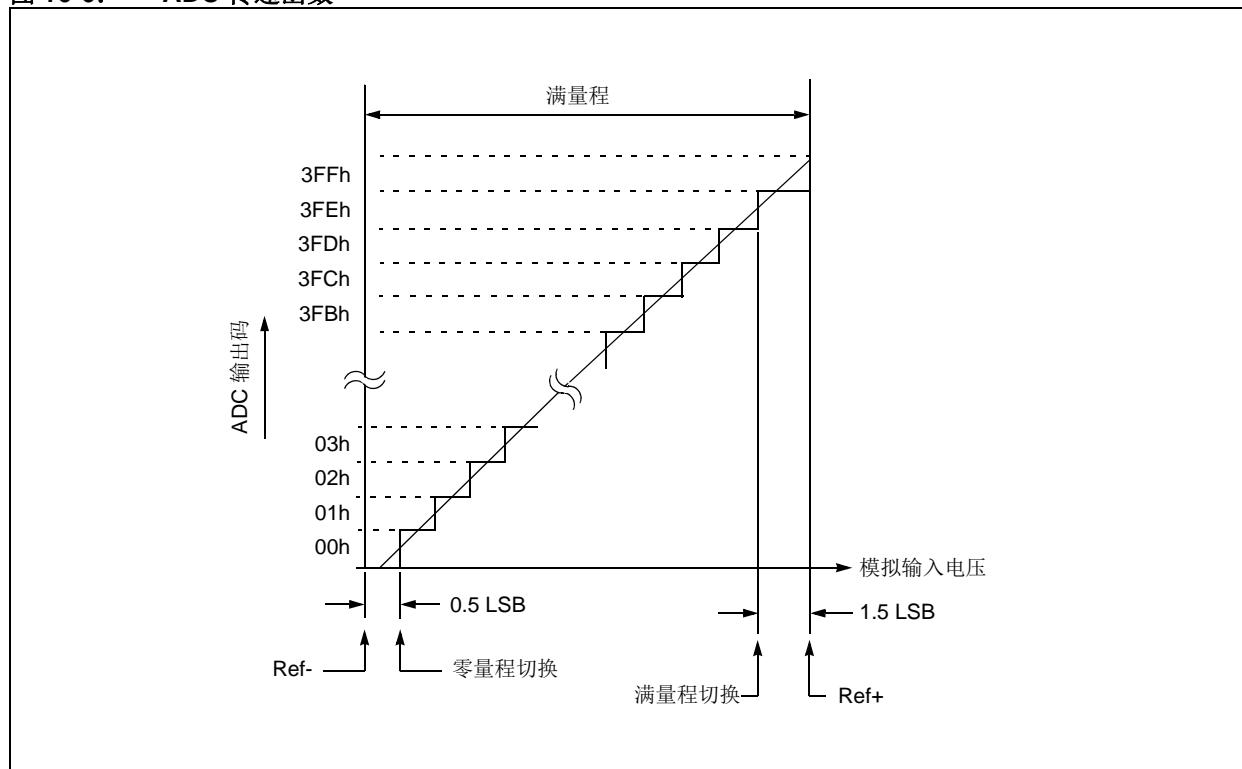


表 16-3：与 ADC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ADCON0	—	CHS<4:0>				GO/DONE	ADON	158	
ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		159
ADCON2	TRIGSEL<3:0>			—	—	—	—	160	
ADRESH	ADC 结果寄存器的高字节								161, 162
ADRESL	ADC 结果寄存器的低字节								161, 162
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	121
INTCON	GIE	PEIE	TMROIE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	120
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		149

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

注 1：未实现，读为 1。

17.0 5 位数模转换器（DAC）模块

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

DAC 的正输入源（VSOURCE+）可以连接到：

- 外部 VREF+ 引脚
- VDD 供电电压
- FVR_buffer1

DAC 的负输入源（VSOURCE-）可以连接到：

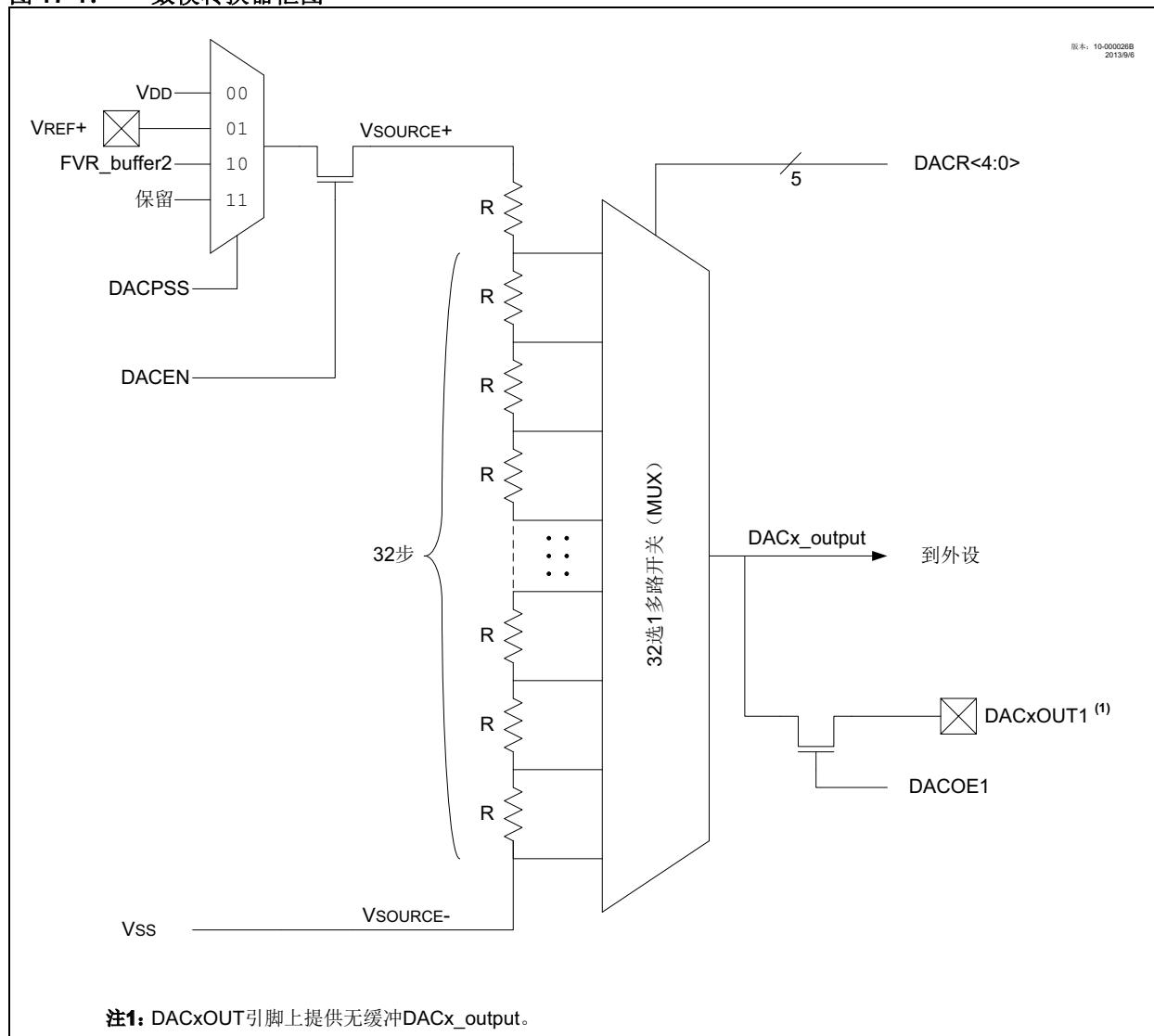
- Vss

可以选择 DAC 的输出（DACx_output）作为以下对象的参考电压：

- 比较器的同相输入
- ADC 输入通道
- DACxOUT1 引脚

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

图 17-1：数模转换器框图



17.1 输出电压选择

DAC 具有 32 个电压范围。32 个电平通过 DACxCON1 寄存器的 **DACR<4:0>** 位进行设置。

DAC 输出电压可以使用[公式 17-1](#) 确定。

当 **DACxOUTn** 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。

注： 无缓冲的 DAC 输出 (**DACxOUTn**) 不用于驱动外部负载。

17.2 比例输出电压

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

[表 27-16](#) 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

可以通过将 **DACxCON0** 寄存器的相应 **DACOE_n** 位置 1，将无缓冲的 DAC 电压输出到 **DACxOUTn** 引脚。选择将 DAC 参考电压输出到 **DACxOUTn** 引脚会自动改写数字输出缓冲器和该引脚的弱上拉和数字输入阈值检测器功能。

公式 17-1： DAC 输出电压

如果 $DACEN = 1$

$$DACx_output = \left((VSOURCE+ - VSOURCE-) \times \frac{DACR[4:0]}{2^5} \right) + VSOURCE-$$

注： 有关可用的 **VSOURCE+** 和 **VSOURCE-** 选择，请参见 **DACxCON0** 寄存器。

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

17.5 复位的影响

器件复位会产生以下影响：

- 禁止 **DACx**。
- **DACx** 输出电压从 **DACxOUTn** 引脚上被移除。
- **DACR<4:0>** 范围选择位被清零。

PIC16(L)F1574/5/8/9

17.6 寄存器定义: DAC 控制

寄存器 17-1: DACCON0: 参考电压控制寄存器 0

R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0
DACEN	—	DACOE	—	DACPSS<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 **DACEN:** DAC 使能位
1 = 使能 DAC
0 = 禁止 DAC

bit 6 未实现: 读为 0

bit 5 **DACOE:** DAC 电压输出使能位
1 = DAC 电压从 DACOUT1 引脚输出
0 = DAC 电压从 DACOUT1 引脚断开

bit 4 未实现: 读为 0

bit 3-2 **DACPSS<1:0>:** DAC 正参考电压源选择位
11 = 保留
10 = FVR_buffer2
01 = VREF+ 引脚
00 = VDD

bit 1-0 未实现: 读为 0

寄存器 17-2: DACCON1: 参考电压控制寄存器 1

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-5 未实现: 读为 0

bit 4-0 **DACR<4:0>:** DAC 电压输出选择位

表 17-1: 与 DAC 模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
DACCON0	DACEN	—	DACOE	—	DACPSS<1:0>	—	—	—	168
DACCON1	—	—	—	—	DACR<4:0>	—	—	—	168

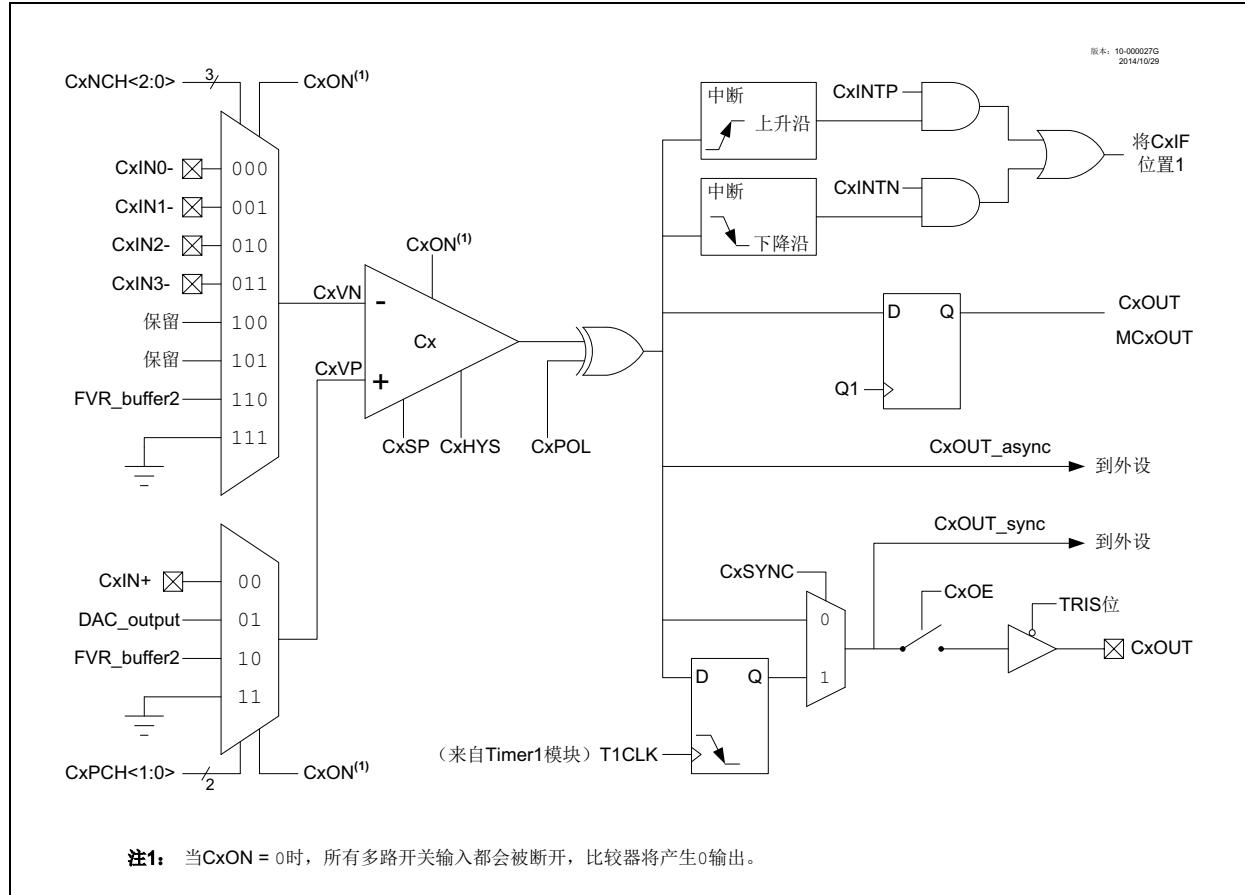
图注: — = 未实现位, 读为 0。 DAC 模块不使用阴影单元。

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
- 从休眠模式唤醒
- 可编程的速度 / 功耗优化
- PWM 关闭
- 可编程和固定参考电压

图 18-1：比较器模块的简化框图



18.1 比较器概述

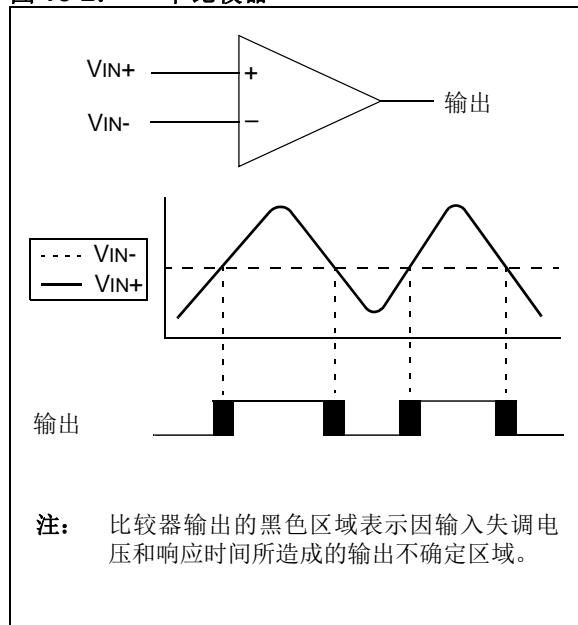
图 18-2 所示为单比较器以及模拟输入电压与数字输出之间的关系。当 V_{IN+} 上的模拟电压小于 V_{IN-} 上的模拟电压时，比较器输出为数字低电平。当 V_{IN+} 上的模拟电压大于 V_{IN-} 上的模拟电压时，比较器输出为数字高电平。

表 18-1 列出了该器件可用的比较器。

表 18-1: 可用的比较器

器件	C1	C2
PIC16(L)F1574	●	●
PIC16(L)F1575	●	●
PIC16(L)F1578	●	●
PIC16(L)F1579	●	●

图 18-2：单比较器



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
- 延后使能
- 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 比较器同相输入选择

通过配置 CMxCON1 寄存器的 CxPCH<1:0> 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
- DAC1_output
- FVR_buffer2
- Vss

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

关于 DAC 输入信号的更多信息，请参见 [第 17.0 节“5 位数模转换器 \(DAC\) 模块”](#)。

每当禁止比较器 (CxON = 0) 时，所有比较器输入都会被禁止。

18.2.3 比较器反相输入选择

CMxCON0 寄存器的 CxNCH<2:0> 位指示输入源之一连接到比较器的反相输入。

注：要将 CxIN+ 和 CxINx- 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。

18.2.4 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：

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

同步比较器输出信号 (CxOUT_sync) 可提供给以下外设：

- 模数转换器 (ADC)
- Timer1

异步比较器输出信号 (CxOUT_async) 可提供给以下外设：

- 互补波形发生器 (CWG)

注： 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。

18.2.5 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）。

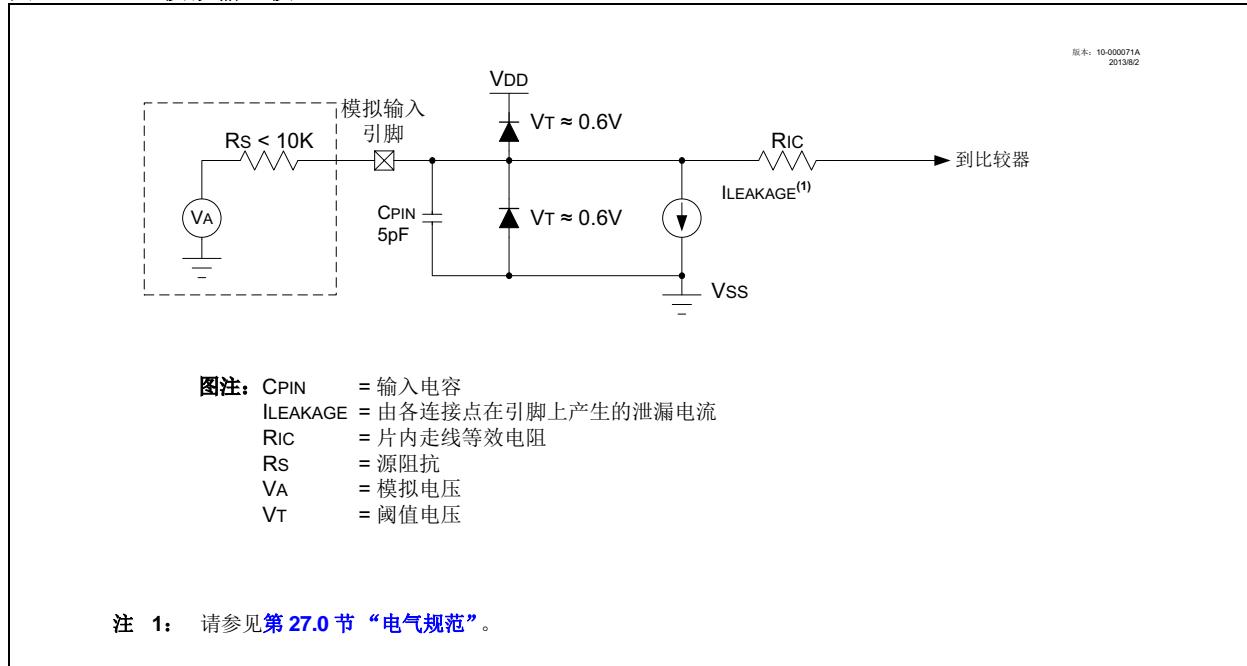
表 18-2: 比较器输出状态与输入条件

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

18.2.6 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，即选择正常速度模式。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。

图 18-3: 模拟输入模型



18.3 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 V_{DD} 和 V_{SS} 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 V_{SS} 和 V_{DD} 之间。如果输入电压与这一范围偏离的绝对值超过 0.6V，就可能发生一个二极管正向导通，从而可能导致锁死发生。

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

注 1: 读端口寄存器时，所有配置为模拟输入的引脚将读为 0。配置为数字输入的引脚将按照输入规范转换为模拟输入。

2: 定义为数字输入引脚上的模拟电压可能会使输入缓冲器的电流消耗超过规定值。

18.4 比较器滞后

通过在每个比较器的输入引脚上施加一个可选大小的分压量，可以为整体操作提供滞后功能。滞后功能通过将 CMxCON0 寄存器的 CxHYS 位置 1 来使能。

更多信息，请参见[第 27.0 节 “电气规范”](#)。

18.5 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 20.5 节 “Timer1 门控”](#)。该功能可用于对模拟事件的持续时间或间隔时间进行计时。

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

18.5.1 比较器输出同步

通过将 CMxCON0 寄存器的 CxSYNC 位置 1，可以使 Cx 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 20-1](#)）。

18.6 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，如果其相关的允许位已置 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 和 CxPOL 位
- PIE2 寄存器的 CxIE 位
- CMxCON1 寄存器的 CxINTP 位（对于上升沿检测）
- CMxCON1 寄存器的 CxINTN 位（对于下降沿检测）
- INTCON 寄存器的 PEIE 和 GIE 位

相关的中断标志位（PIR2 寄存器的 CxIF 位）必须用软件清零。如果在清零该标志时检测到另一个边沿，则标志仍然会在序列结束时置 1。

注： 即使比较器被禁止，还是可以通过使用 CMxCON0 寄存器的 CxPOL 位更改输出极性来产生中断，或者通过使用 CMxCON0 寄存器的 CxON 位开启或关闭比较器来产生中断。

18.7 比较器响应时间

在改变输入源或选择新的参考电压后的一段时间内，比较器的输出状态都是不确定的。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 27.0 节 “电气规范”](#)中的比较器和参考电压规范。

18.8 寄存器定义：比较器控制

寄存器 18-1： CMxCON0：比较器 Cx 控制寄存器 0

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

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7	CxON: 比较器使能位 1 = 使能比较器 0 = 禁止比较器，不消耗有功功率
bit 6	CxOUT: 比较器输出位 <u>如果 CxPOL = 1 (极性反相):</u> 1 = CxVP < CxVN 0 = CxVP > CxVN <u>如果 CxPOL = 0 (极性不反相):</u> 1 = CxVP > CxVN 0 = CxVP < CxVN
bit 5	未实现：读为 0
bit 4	CxPOL: 比较器输出极性选择位 1 = 比较器输出反相 0 = 比较器输出不反相
bit 3	未实现：读为 0
bit 2	CxSP: 比较器速度 / 功耗选择位 1 = 比较器工作在正常功耗、高速模式下 0 = 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 时钟源的变化同步。输出在 Timer1 时钟源的下降沿更新。 0 = 送到 Timer1 和 I/O 引脚的比较器输出是异步的

PIC16(L)F1574/5/8/9

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7

CxINTP: 比较器正向边沿中断允许位

1 = 在 CxOUT 位的正向边沿, CxIF 中断标志将置 1
0 = 在 CxOUT 位的正向边沿, CxIF 中断标志不会置 1

bit 6

CxINTN: 比较器负向边沿中断允许位

1 = 在 CxOUT 位的负向边沿, CxIF 中断标志将置 1
0 = 在 CxOUT 位的负向边沿, CxIF 中断标志不会置 1

bit 5-4

CxPCH<1:0>: 比较器同相输入通道选择位

11 = CxVP 连接到 VSS
10 = CxVP 连接到 FVR 参考电压
01 = CxVP 连接到 DAC 参考电压
00 = CxVP 连接到 CxIN+ 引脚

bit 3

未实现: 读为 0

bit 2-0

CxNCH<1:0>: 比较器反相输入通道选择位

111 = CxVN 连接到 GND
110 = CxVN 连接到 FVR 参考电压
101 = 保留
100 = 保留
011 = CxVN 连接到 CxIN3- 引脚
010 = CxVN 连接到 CxIN2- 引脚
001 = CxVN 连接到 CxIN1- 引脚
000 = CxVN 连接到 CxIN0- 引脚

寄存器 18-3: CMOUT: 比较器输出寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-2

未实现: 读为 0

bit 1

MC2OUT: C2OUT 的镜像副本位

bit 0

MC1OUT: C1OUT 的镜像副本位

表 18-3：与比较器模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	121
CM1CON0	C1ON	C1OUT	—	C1POL	—	C1SP	C1HYS	C1SYNC	173
CM1CON1	C1NTP	C1INTN	C1PCH<1:0>	—	—	C1SP	C1HYS	C1SYNC	174
CM2CON0	C2ON	C2OUT	—	C2POL	—	C2SP	C2HYS	C2SYNC	173
CM2CON1	C2NTP	C2INTN	C2PCH<1:0>	—	—	C2SP	C2HYS	C2SYNC	174
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	174
DACCON0	DACEN	—	DACOE	—	DACPSS<1:0>	—	—	—	168
DACCON1	—	—	—	—	DACR<4:0>	—	—	—	168
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>	—	ADFVR<1:0>	—	149
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE2	—	C2IE	C1IE	—	—	—	—	—	88
PIR2	—	C2IF	C1IF	—	—	—	—	—	91
PORATA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	120
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	121
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	120

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。

19.0 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
- 溢出时产生中断
- TMR0 可用于门控 Timer1

图 19-1 给出了 Timer0 模块的框图。

19.1 Timer0 工作原理

Timer0 模块可用作 8 位定时器或 8 位计数器。

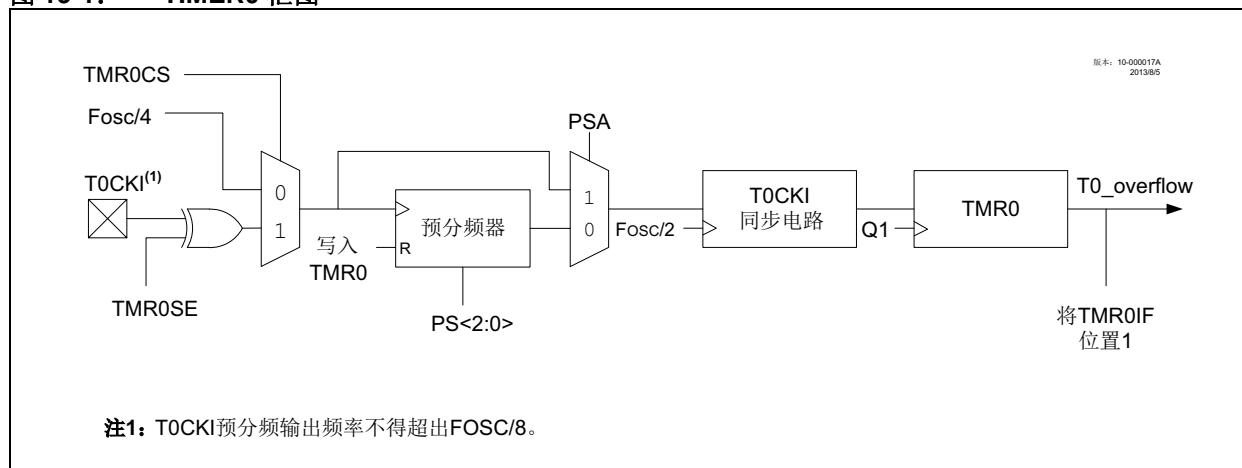
19.1.1 8 位定时器模式

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

当写 TMR0 时，紧跟写操作之后的两个指令周期内禁止 TMR0 递增。

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

图 19-1：TIMER0 框图



注1： T0CKI预分频输出频率不得超出FOSC/8。

19.1.3 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。写入 TMRO 寄存器的所有指令都会清零预分频器。

19.1.4 TIMER0 中断

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

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

19.1.5 8 位计数器模式同步

在 8 位计数器模式下，T0CKI 引脚的递增边沿必须与指令时钟保持同步。同步可通过在指令时钟的 Q2 和 Q4 周期对预分频器的输出进行采样实现。外部时钟源的高低电平周期必须满足第 27.0 节“电气规范”中所示的时序要求。

19.1.6 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，TMRO 寄存器的内容将保持不变。

PIC16(L)F1574/5/8/9

19.2 寄存器定义：选项寄存器

寄存器 19-1：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 = TOCKI 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 TOCKI 引脚电平从高至低跳变时，递增计数
0 = 在 TOCKI 引脚电平从低至高跳变时，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 19-1：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>				—	—	—	—	160
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			178
TMRO	8 位 Timer0 计数的保持寄存器							—	176*
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	120

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。

20.0 带门控的 TIMER1 模块

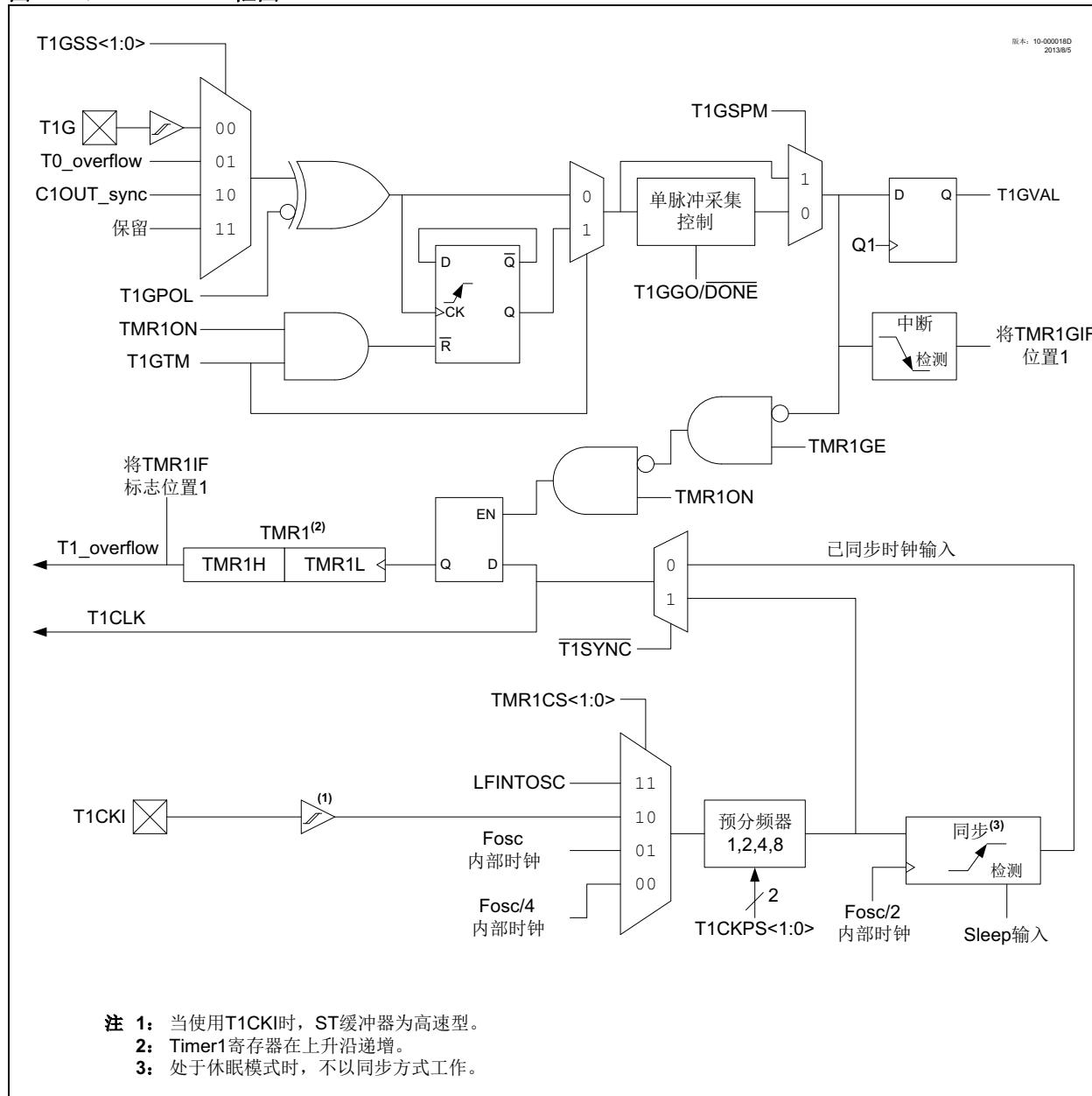
Timer1 模块是 16 位定时器 / 计数器，具有以下特性：

- 16 位定时器 / 计数器寄存器对 (TMR1H:TMR1L)
- 可编程内部或外部时钟源
- 2 位预分频器
- 可选的同步比较器输出
- 多个 Timer1 门控 (计数使能) 源
- 溢出时产生中断

- 溢出触发唤醒 (仅限外部时钟, 异步模式)
- ADC 自动转换触发器
- 可选择的门控源极性
- 门控翻转模式
- 门控单脉冲模式
- 门控值状态
- 门控事件中断

图 20-1 给出了 Timer1 模块的框图。

图 20-1：TIMER1 框图



20.1 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 通过分别配置 T1CON 和 T1GCON 寄存器中的 TMR1ON 和 TMR1GE 位使能。[表 20-1](#) 显示了 Timer1 使能选择。

表 20-1: TIMER1 使能选择

TMR1ON	TMR1GE	Timer1 工作
0	0	关闭
0	1	关闭
1	0	总是开启
1	1	计数使能

20.2 时钟源选择

T1CON 寄存器的 TMR1CS<1:0> 位用于选择 Timer1 的时钟源。[表 20-2](#) 显示了时钟源选择。

20.2.1 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 LSB 误差。为了利用 Timer1 的全部分辨率，必须使用异步输入信号来对 Timer1 时钟输入进行门控。

可以使用以下异步源：

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

20.2.2 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。外部时钟源既可以与单片机系统时钟同步，也可以异步运行。

注： 在计数器模式下，发生以下任何一个或多个情况后，计数器在首个递增上升沿前，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 20-2: 时钟源选择

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

注 1： T1OSC 并非在所有器件上都可用。

20.3 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。

20.4 异步计数器模式下的 Timer1 操作

如果 T1CON 寄存器的控制位 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，并在溢出时产生中断从而唤醒处理器。但是，用软件对定时器进行读 / 写操作时要特别当心（见第 20.4.1 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。

20.4.1 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溢出。

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

20.5 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。

20.5.1 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 20-3。

表 20-3：TIMER1 门控使能选择

T1CLK	T1GPOL	T1G	Timer1 工作原理
↑	0	0	计数
↑	0	1	保持计数
↑	1	0	保持计数
↑	1	1	计数

20.5.2 TIMER1 门控源选择

表 20-4 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 20-4：TIMER1 门控源

T1GSS	Timer1 门控源
00	Timer1 门控引脚 (T1G)
01	Timer0 溢出 (T0_overflow) (TMR0 从 FFh 递增到 00h)
10	比较器 C1 输出 (C1OUT_sync) (1)
11	比较器 C2 输出 (C2OUT_sync) (1)

注 1：可选的同步比较器输出

20.5.2.1 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。

20.5.2.2 Timer0 溢出门控操作

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

20.5.3 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 20-4。

Timer1 门控翻转模式通过将 T1GCON 寄存器的 T1GTM 位置 1 使能。T1GTM 位清零时，将清零触发器并保持清零。这对于控制测量哪个边沿是必需的。

注： 在使能翻转模式的同时改变门控极性，可能会导致不确定的操作。

20.5.4 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。首先通过将 T1GCON 寄存器中的 T1GSPM 位置 1 使能 Timer1 门控单脉冲模式。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 位再次用软件置 1。时序详细信息请参见图 20-5。

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 位也应清零。

同时使能翻转模式和单脉冲模式将允许两部分协同工作。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 20-6。

20.5.5 TIMER1 门控值状态

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

20.5.6 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，则会识别到一个中断。

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

20.6 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。

20.7 休眠期间的 Timer1 操作

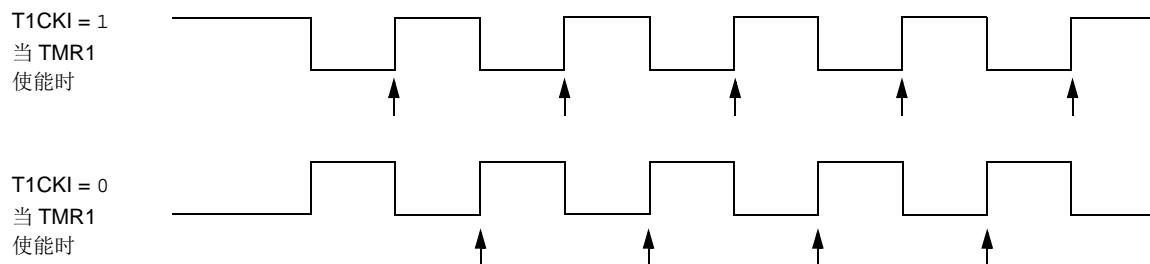
只有在设置为异步计数器模式时，Timer1 才能在休眠模式下工作。在该模式下，可使用外部晶振或时钟源使计数器递增计数。要设置定时器以唤醒器件：

- 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

器件将在溢出时被唤醒并执行下一条指令。如果将 INTCON 寄存器的 GIE 位置 1，器件将调用中断服务程序。

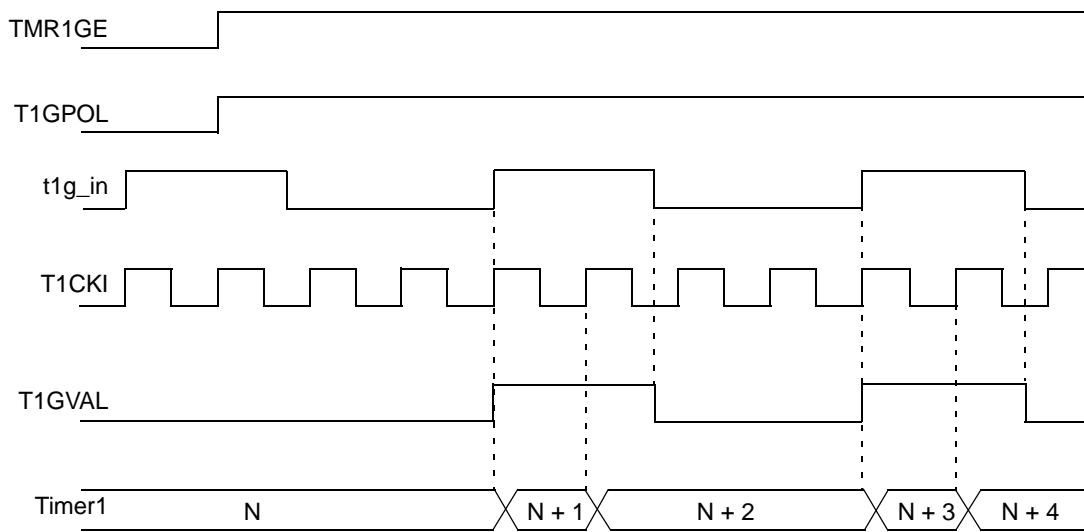
无论 T1SYNC 位的设置如何，Timer1 振荡器都会在休眠模式下继续工作。

图 20-2： TIMER1 递增边沿



注 1： 箭头表明计数器递增。
2： 在计数器模式下，计数器在时钟的首个递增上升沿之前，必须先经过一个下降沿。

图 20-3： TIMER1 门控使能模式



PIC16(L)F1574/5/8/9

图 20-4： TIMER1 门控翻转模式

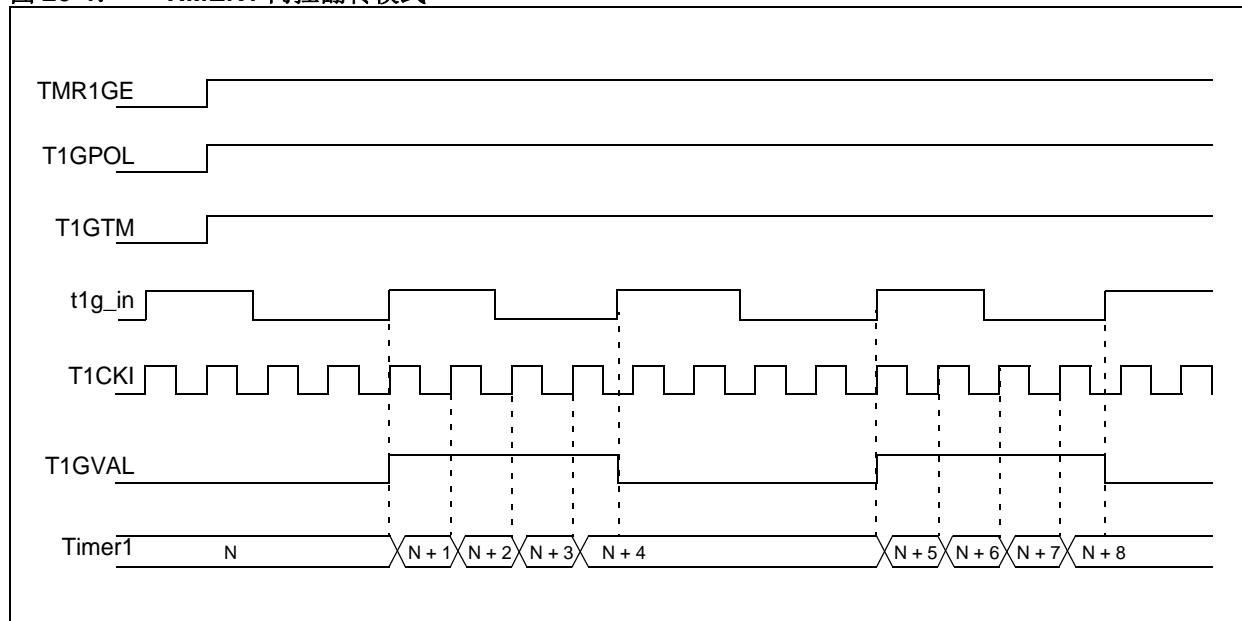


图 20-5： TIMER1 门控单脉冲模式

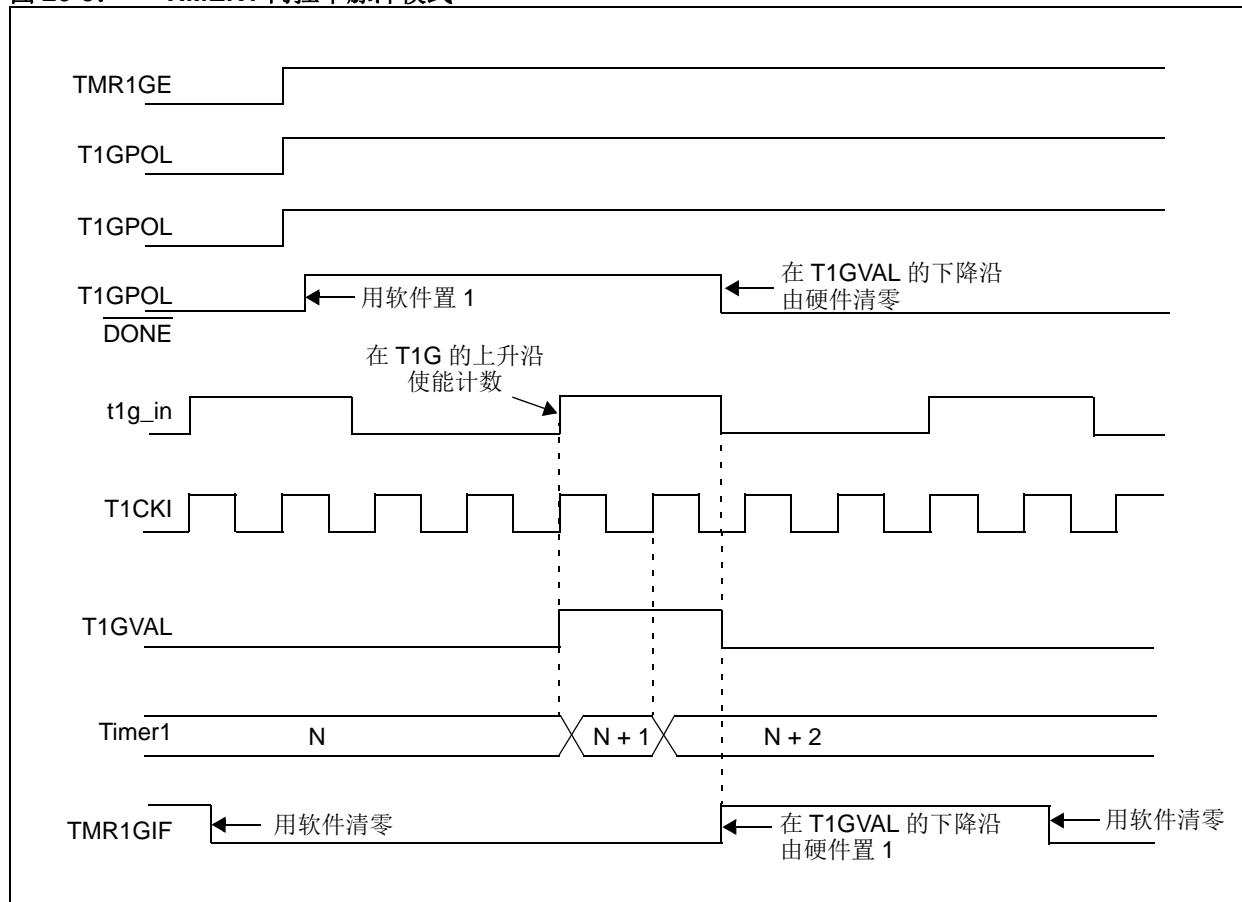
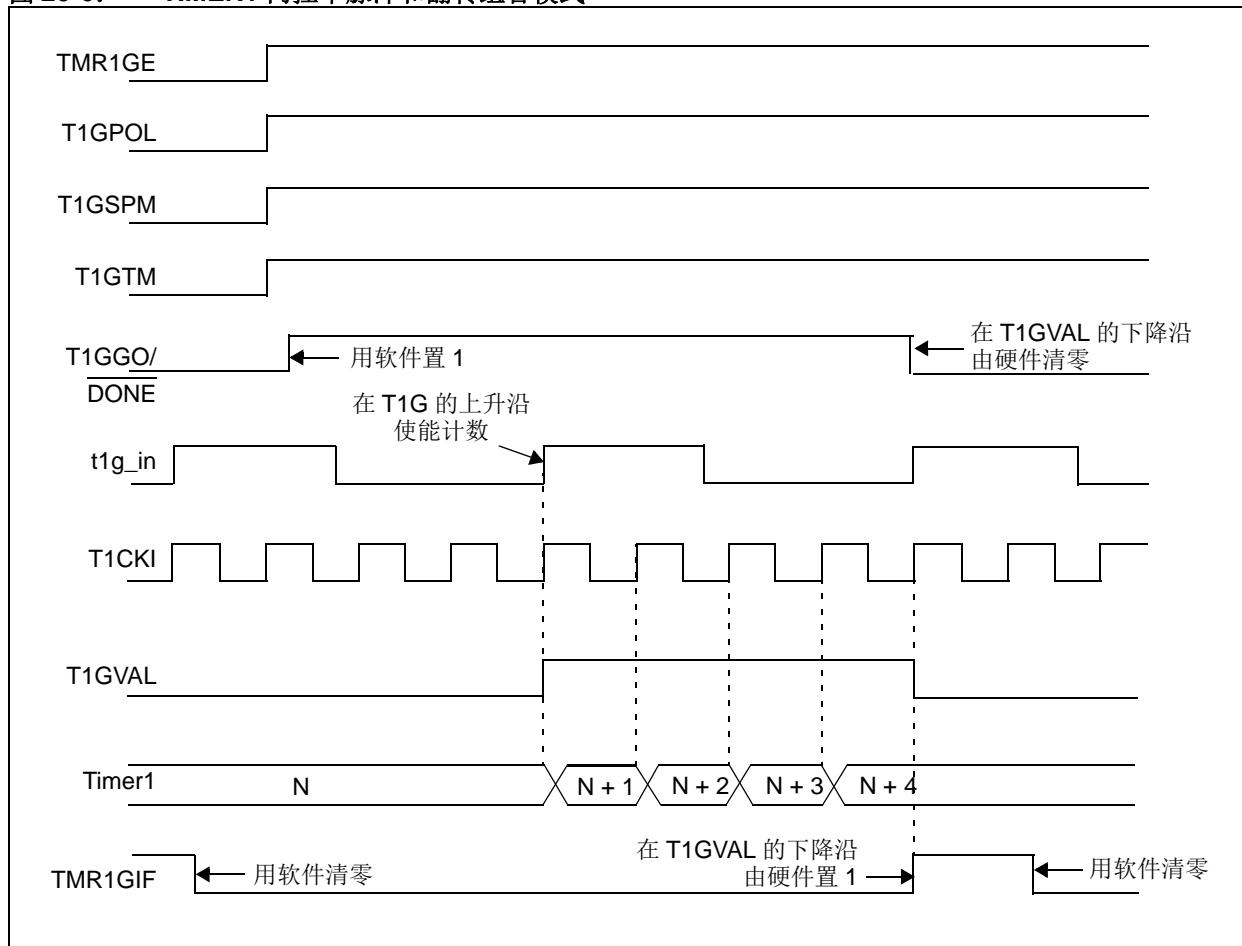


图 20-6： TIMER1 门控单脉冲和翻转组合模式



20.8 寄存器定义: Timer1 控制

寄存器 20-1: 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>		—	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 = Timer1 时钟源为 LFINTOSC

10 = Timer1 时钟源为 T1CKI 引脚 (上升沿触发计数)

01 = Timer1 时钟源为系统时钟 (Fosc)

00 = Timer1 时钟源为指令时钟 (Fosc/4)

bit 5-4 **T1CKPS<1:0>**: Timer1 输入时钟预分频比选择位

11 = 1:8 预分频值

10 = 1:4 预分频值

01 = 1:2 预分频值

00 = 1:1 预分频值

bit 3 未实现: 读为 0

bit 2 **T1SYNC**: Timer1 同步控制位

1 = 不同步异步时钟输入

0 = 将异步时钟输入与系统时钟 (Fosc) 同步

bit 1 未实现: 读为 0

bit 0 **TMR1ON**: Timer1 使能位

1 = 使能 Timer1

0 = 停止 Timer1 并清零 Timer1 门控触发器

寄存器 20-2: T1GCON: TIMER1 门控寄存器

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
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 门控翻转模式并清零门控触发器的输出
 Timer1 门控触发器的输出在每个上升沿翻转。
- 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 门控值状态位
 指示可提供给 TMR1H:TMR1L 的 Timer1 门控信号的当前状态。
 不受 Timer1 门控使能 (TMR1GE) 的影响。
- bit 1-0 **T1GSS<1:0>:** Timer1 门控源选择位
 11 = 比较器 C2 的可选同步输出 (C2OUT_sync)
 10 = 比较器 C1 的可选同步输出 (C1OUT_sync)
 01 = Timer0 溢出输出 (T0_overflow)
 00 = Timer1 门控引脚 (T1G)

PIC16(L)F1574/5/8/9

表 20-5：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	121
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
OSCSTAT	—	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	70
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	91
TMR1H	16 位 TMR1 计数高字节的保持寄存器								183*
TMR1L	16 位 TMR1 计数低字节的保持寄存器								183*
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	120
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	T1SYNC	—	TMR1ON	186
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		187

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 PIC16(L)F1575。

21.0 TIMER2 模块

Timer2 模块具有以下特性：

- 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、1:4、1:16 和 1:64）
- 可软件编程的后分频器（分频比为 1:1 至 1:16）
- TMR2 与 PR2 匹配时产生中断

Timer2 框图请参见图 21-1。

图 21-1：TIMER2 框图

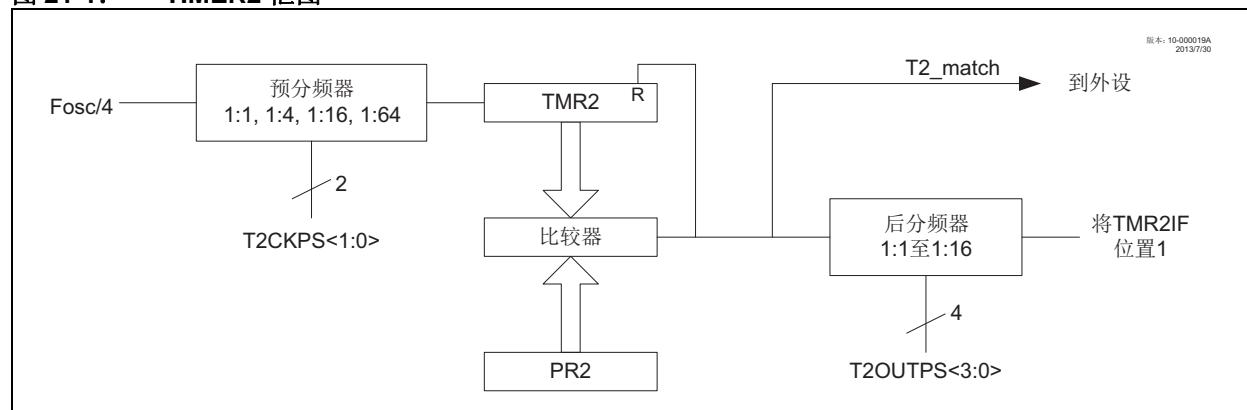
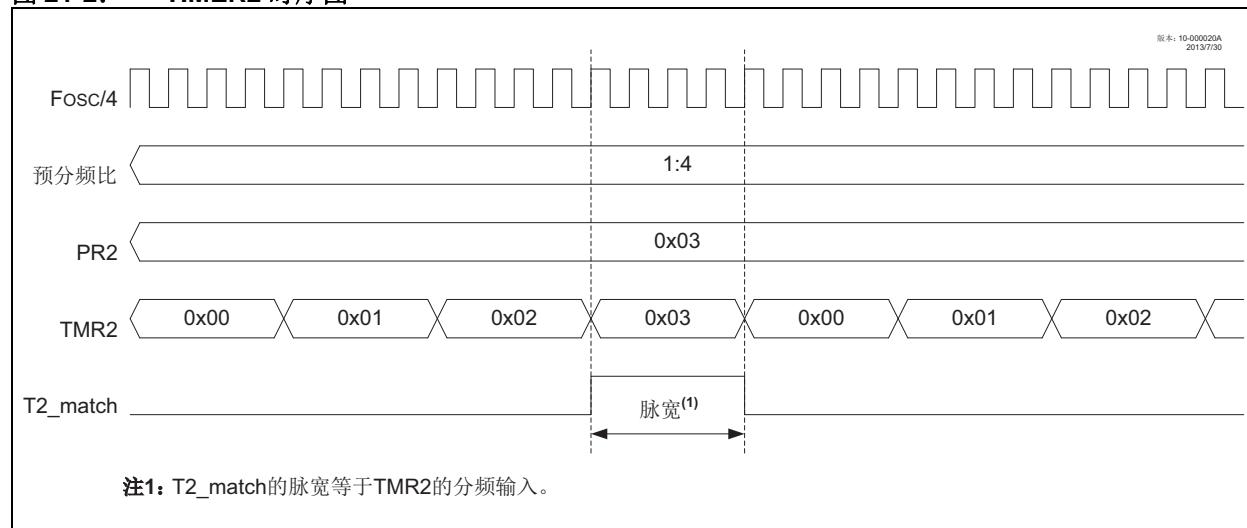


图 21-2：TIMER2 时序图



21.1 Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟 (Fosc/4)。

TMR2 会从 00h 开始在每个时钟边沿递增。

4 位计数器 / 预分频器提供了对时钟输入不分频、4 分频和 16 分频三个预分频选项。这些选项通过 T2CON 寄存器的预分频比控制位 T2CKPS<1:0> 进行选择。在每个时钟周期，TMR2 的值都会与周期寄存器 PR2 中的值进行比较。当两个值匹配时，由比较器产生匹配信号作为定时器的输出。该信号也会将 TMR2 的值在下一个周期复位为 00h，并驱动输出计数器 / 后分频器（见第 21.2 节“Timer2 中断”）。

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

- 对 TMR2 寄存器进行写操作
- 对 T2CON 寄存器进行写操作
- 上电复位 (POR)
- 欠压复位 (BOR)
- MCLR 复位
- 看门狗定时器 (WDT) 复位
- 堆栈上溢复位
- 堆栈下溢复位
- RESET 指令

注： 写 T2CON 时 TMR2 不会清零。

21.2 Timer2 中断

Timer2 也可以产生可选的器件中断。Timer2 输出信号 (T2_match) 为 4 位计数器 / 后分频器提供输入。该计数器产生 TMR2 匹配中断，对应的中断标志位为 PIR1 寄存器的 TMR2IF 位。可以通过将 PIE1 寄存器的 TMR2 匹配中断允许位 TMR2IE 置 1 来允许该中断。

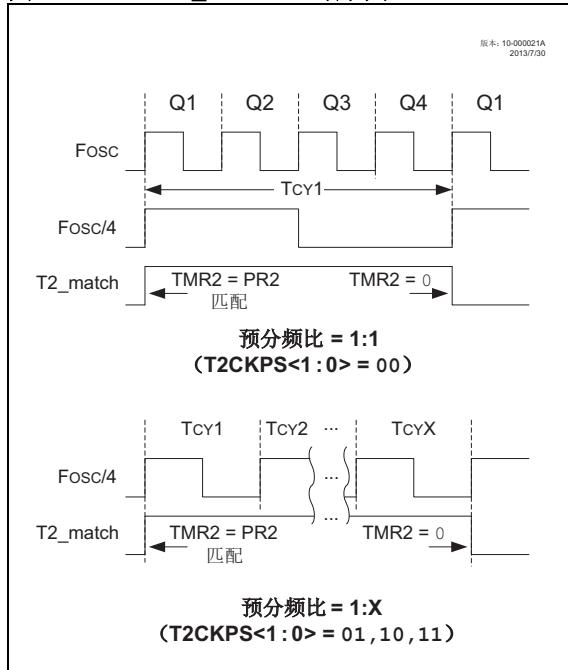
可以通过 T2CON 寄存器的后分频比控制位 T2OUTPS<3:0> 在 16 个后分频比选项（从 1:1 至 1:16）中选择其一。

21.3 Timer2 输出

TMR2 的输出为 T2_match。

T2_match 信号与系统时钟同步。图 21-3 给出了 T2_match 信号相对于 Fosc 和预分频值 T2CKPS<1:0> 的时序的两个示例。上面的图显示了预分频比为 1:1 的时序，下面的图显示了预分频比为 1:X 的时序。

图 21-3：T2_MATCH 时序图



21.4 休眠期间的 Timer2 操作

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。

21.5 寄存器定义：Timer2 控制

寄存器 21-1：T2CON：TIMER2 控制寄存器

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7 未实现：读为 0

bit 6-3 T2OUTPS<3:0>：Timer2 输出后分频比选择位

0000 = 1:1 后分频比

0001 = 1:2 后分频比

0010 = 1:3 后分频比

0011 = 1:4 后分频比

0100 = 1:5 后分频比

0101 = 1:6 后分频比

0110 = 1:7 后分频比

0111 = 1:8 后分频比

1000 = 1:9 后分频比

1001 = 1:10 后分频比

1010 = 1:11 后分频比

1011 = 1:12 后分频比

1100 = 1:13 后分频比

1101 = 1:14 后分频比

1110 = 1:15 后分频比

1111 = 1:16 后分频比

bit 2 TMR2ON：Timer2 使能位

1 = 使能 Timer2

0 = 关闭 Timer2

bit 1-0 T2CKPS<1:0>：Timer2 时钟预分频比选择位

00 = 预分频比为 1

01 = 预分频比为 4

10 = 预分频比为 16

11 = 预分频比为 64

表 21-1：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
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
PR2	Timer2 模块周期寄存器								189*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		191
TMR2	8 位 TMR2 计数的保持寄存器								189*

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。

注 1：仅限 PIC16(L)F1575。

22.0 增强型通用同步 / 异步收发器 (EUSART)

增强型通用同步 / 异步收发器 (EUSART) 模块是一种串行 I/O 通信外设。它包含用来完成与器件程序执行无关的输入或输出串行数据传输所需的所有时钟发生器、移位寄存器和数据缓冲区等。EUSART 也可称为串行通信接口 (Serial Communication Interface, SCI)，可配置为全双工异步系统或半双工同步系统。全双工模式可用来与外设系统通信，如 CRT 终端和个人计算机。半双工同步模式用于与外设通信，如 A/D 或 D/A 集成电路、串行 EEPROM 或其他单片机。这些器件通常不具备用以产生波特率的内部时钟，并需要由主同步器件提供外部时钟信号。

EUSART 模块具备以下功能：

- 全双工异步收发
- 双字符输入缓冲区
- 单字符输出缓冲区
- 可编程 8 位或 9 位字符长度
- 9 位模式下的地址检测
- 输入缓冲区溢出错误检测
- 接收字符帧错误检测
- 半双工同步主模式
- 半双工同步从模式
- 同步模式下的可编程时钟极性
- 休眠模式下的操作

EUSART 模块还具备以下特性，使其成为局域互联网 (Local Interconnect Network, LIN) 总线系统的理想选择：

- 波特率的自动检测和校准
- 接收到间隔字符时唤醒
- 13 位间隔字符发送

EUSART 发送器和接收器的框图如图 22-1 和图 22-2 所示。

图 22-1：EUSART 发送框图

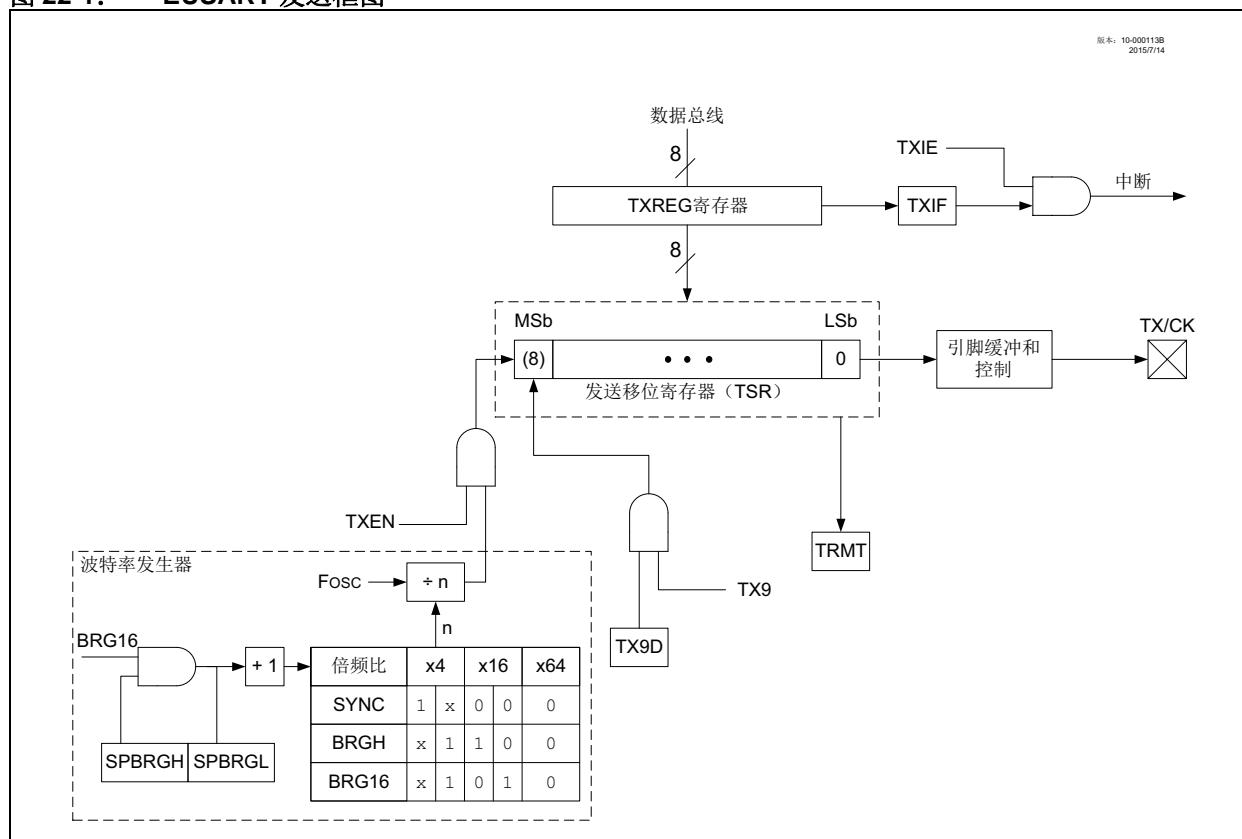
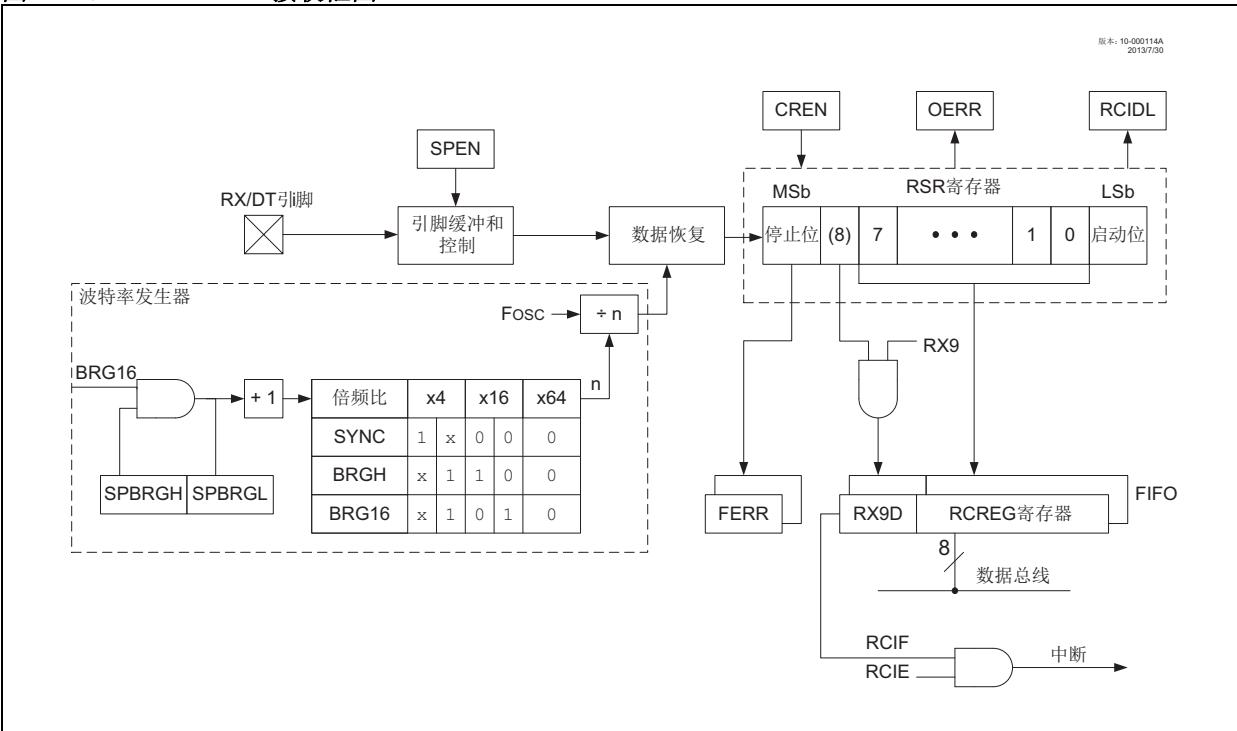


图 22-2：EUSART 接收框图



EUSART 模块的操作由以下 3 个寄存器控制：

- 发送状态和控制寄存器 (TXSTA)
- 接收状态和控制寄存器 (RCSTA)
- 波特率控制寄存器 (BAUDCON)

这些寄存器将在 [寄存器 22-1](#)、[寄存器 22-2](#) 和 [寄存器 22-3](#) 中分别详细介绍。

当未使能接收器或发送器部分时，对应的 RX 或 TX 引脚可用于通用输入和输出。

22.1 EUSART 异步模式

EUSART 采用标准不归零 (non-return-to-zero, NRZ) 格式发送和接收数据。NRZ 使用两种电平实现: VOH 标记状态 (mark state) 代表 “1” 数据位, 而 VOL 空格状态 (space state) 代表 “0” 数据位。NRZ 指的是对于连续发送的具有相同值的数据位, 它们保持在该位的输出电压, 而不会在发送完每个位之后回到中间电压。NRZ 发送端口在标记状态空闲。每个字符发送包含 1 个启动位及随后的 8 个或 9 个数据位, 并始终由 1 个或多个停止位终止。启动位始终是一个空格, 停止位始终是标记。最常用的数据格式为 8 位。每个发送位保持时间为 1/(波特率)。使用片上专用 8 位 /16 位波特率发生器从系统振荡器产生标准波特率频率。波特率配置示例请参见表 22-5。

EUSART 先发送和接收 Lsb。EUSART 的发送器和接收器在功能上是相互独立的, 但它们的数据格式和波特率相同。硬件不支持奇偶校验, 但可通过软件实现奇偶校验, 并将奇偶校验位作为第 9 个数据位存储。

22.1.1 EUSART 异步发送器

图 22-1 给出了 EUSART 发送器框图。发送器的核心是串行发送移位寄存器 (Transmit Shift Register, TSR), 该寄存器不可用软件直接访问。TSR 从发送缓冲区 (即 TXREG 寄存器) 取得数据。

22.1.1.1 使能发送器

EUSART 发送器可通过配置以下 3 个控制位使能为异步操作:

- TXEN = 1
- SYNC = 0
- SPEN = 1

假定所有其他 EUSART 控制位均处于其默认状态。

将 TXSTA 寄存器的 TXEN 位置 1, 可以使能 EUSART 的发送器电路。清零 TXSTA 寄存器的 SYNC 位将 EUSART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1, 可以使能 EUSART 并自动将 TX/CK I/O 引脚配置为输出。如果 TX/CK 引脚与模拟外设共用, 则必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

注: TXEN 中断允许位置 1 时, TXIF 发送器中断标志位置 1。

22.1.1.2 发送数据

向 TXREG 寄存器写入一个字符时启动发送。如果这是首字符, 或前一个字符被完全从 TSR 中送出, TXREG 中的数据就立即被传送到 TSR 寄存器。如果 TSR 中仍保存前一个字符的全部或部分, 则新字符数据被保存在 TXREG 中, 直到前一个字符的停止位被发送。之后, 在 TXREG 中等待的字符在停止位发送后 1 个 TCY 内被传送到 TSR 中。TXREG 中的数据被传送到 TSR 后, 启动位、数据位和停止位序列的发送立即开始。

22.1.1.3 发送数据极性

可通过 BAUDCON 寄存器的 SCKP 位来控制发送数据的极性。该位的默认状态为 0, 选择高电平有效发送空闲和数据位。将 SCKP 位设置为 1 将发送数据的极性取反, 从而选择低电平有效空闲和数据位。SCKP 位仅在异步模式下控制发送数据的极性。在同步模式下, SCKP 位有不同的功能。请参见第 22.5.1.2 节 “时钟极性”。

22.1.1.4 发送中断标志

只要 EUSART 发送器被使能且 TXREG 中没有等待发送的字符, PIR1 寄存器的 TXIF 中断标志位就被置 1。换句话说, 只有在 TSR 正在处理字符且 TXREG 中还有一个排队等待发送的新字符时, TXIF 位才被清零。写入 TXREG 后并不立即清零 TXIF 标志位, 而是在之后的第二个指令周期将其清零。写入 TXREG 后立即查询 TXIF 位将返回无效结果。TXIF 位是只读位, 不能用软件置 1 或清零。

将 PIE1 寄存器的 TXIE 中断允许位置 1 可允许 TXIF 中断。但是, 只要 TXREG 为空, 无论 TXIE 中断允许位的状态如何, TXIF 标志位就会被置 1。

要在发送数据时使用中断, 应只在仍有数据要发送时才将 TXIE 位置 1。在将发送的最后一个字符写入 TXREG 后应清零 TXIE 中断允许位。

22.1.1.5 TSR 状态

TXSTA 寄存器的 TRMT 位指示 TSR 寄存器的状态。该位是只读位。TSR 寄存器为空时，TRMT 位置 1，而当一个字符从 TXREG 传送到 TSR 寄存器中时，该位清零。TRMT 位将保持清零，直到所有位移出 TSR 寄存器。该位不与任何中断逻辑关联，因此用户必须查询该位以确定 TSR 的状态。

注： TSR 寄存器不映射到数据存储器中，因此用户无法使用。

22.1.1.6 发送 9 位字符

EUSART 支持 9 位字符发送。当 TXSTA 寄存器的 TX9 位置 1 时，EUSART 将在发送每个字符时移出 9 位。TXSTA 寄存器的 TX9D 位是第 9 个数据位，也是最高有效位。发送 9 位数据时，TX9D 数据位必须先于低 8 位写入 TXREG。写入 TXREG 后，所有 9 位将被立即传送到 TSR 移位寄存器中。

有多个接收器时，可使用一种特殊的 9 位地址模式。

关于地址模式的更多信息，请参见第 22.1.2.7 节“地址检测”。

22.1.1.7 异步发送设置

1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位，获得目标波特率（见第 22.4 节“EUSART 波特率发生器”）。
2. 通过清零 SYNC 位并将 SPEN 位置 1，使能异步串口。
3. 如果需要 9 位发送，将 TX9 控制位置 1。接收器置于地址检测模式时，第 9 个数据位置 1 表示 8 个低数据位为地址。
4. 如果需要将发送数据的极性取反，将 SCKP 位置 1。
5. 将 TXEN 控制位置 1 使能发送。这将导致 TXIF 中断标志位置 1。
6. 如果需要中断，将 PIE1 寄存器的 TXIE 中断允许位置 1。如果 INTCON 寄存器的 GIE 和 PEIE 位也置 1，则立即产生中断。
7. 如果选择了 9 位发送，应将第 9 位装入 TX9D 数据位。
8. 将 8 位数据装入 TXREG 寄存器。这将启动发送。

图 22-3：异步发送

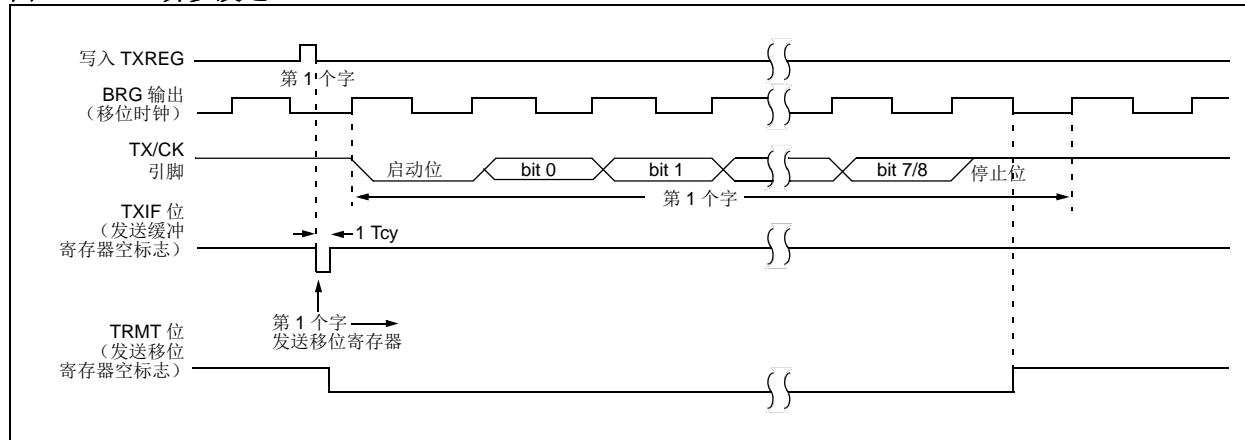
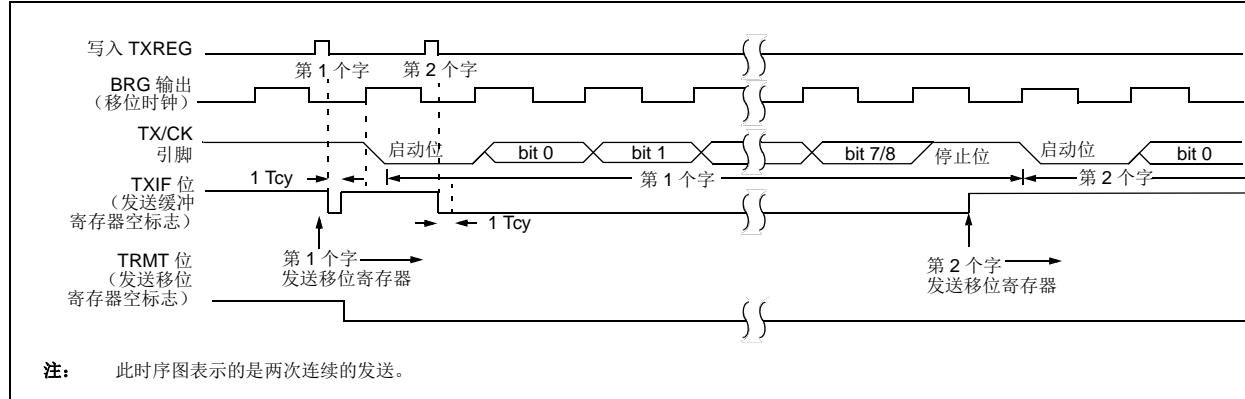


图 22-4：异步发送（连续）



PIC16(L)F1574/5/8/9

表 22-1：与异步发送相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	204
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	203*
SPBRGL	BRG<7:0>								205*
SPBRGH	BRG<15:8>								205*
TXREG	EUSART 发送数据寄存器								194
TXSTA	CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D	202

图注：— = 未实现位，读为 0。异步发送不使用阴影单元。

* 提供寄存器信息的页。

22.1.2 EUSART 异步接收器

异步模式通常用于 RS-232 系统中。图 22-2 给出了接收器框图。数据在 RX/DT 引脚上接收并驱动数据恢复模块。数据恢复模块实际上是一个高速移位器，工作频率为 16 倍波特率，而串行接收移位寄存器（Receive Shift Register, RSR）工作频率为比特率。所有 8 位或 9 位字符移入后被立即传送到双字符的先进先出（First-In-First-Out, FIFO）存储区中。FIFO 缓冲区允许先接收两个完整字符和第三个字符的开始部分后，再开始用软件处理 EUSART 接收器。FIFO 和 RSR 寄存器不能直接用软件访问。通过 RCREG 寄存器访问接收数据。

22.1.2.1 使能接收器

EUSART 接收器可通过配置以下 3 个控制位使能为异步操作：

- CREN = 1
- SYNC = 0
- SPEN = 1

假定所有其他 EUSART 控制位均处于其默认状态。

将 RCSTA 寄存器的 CREN 位置 1 使能 EUSART 的接收器电路。清零 TXSTA 寄存器的 SYNC 位将 EUSART 配置为异步操作。通过将 RCSTA 寄存器的 SPEN 位置 1，可使能 EUSART。编程人员必须将相应的 TRIS 位置 1，将 RX/DT I/O 引脚配置为输入。

注：如果 RX/DT 功能位于模拟引脚上，则必须清零相应的 ANSEL 位使接收器工作。

22.1.2.2 接收数据

接收器的数据恢复电路在第一个位的下降沿启动字符接收。第一个位也称启动（Start）位，始终为零。数据恢复电路计数传输半个位的时间至启动位的中点并验证该位是否仍为零。如果该位非零，则数据恢复电路中止字符接收，不产生错误，并恢复寻找启动位的下降沿。如果启动位被验证为零，则数据恢复电路计数一个位时间至下个位的中点。该位被一个择多检测电路采样，其结果（0 或 1）被移入 RSR。重复此过程直到所有数据位均被采样并移入 RSR。最后一个位时间被测量且其电平被采样。此为停止（Stop）位，始终为 1。如果数据恢复电路在停止位位置采样到 0，则置 1 此字符的帧错误标志位，否则清零此字符的帧错误标志位。关于帧错误的更多信息，请参见第 22.1.2.4 节“接收帧错误”。

所有数据位和停止位被接收后，RSR 中的字符就被立即传送到 EUSART 接收 FIFO，且 PIR1 寄存器的 RCIF 中断标志位被置 1。读取 RCREG 寄存器时，FIFO 中顶部的字符被送出 FIFO。

注：如果接收 FIFO 溢出，在溢出条件被清除前不会接收更多字符。关于溢出错误的更多信息，请参见第 22.1.2.5 节“接收溢出错误”。

22.1.2.3 接收中断

只要 EUSART 接收器被使能且接收 FIFO 中存在未被读取的字符，PIR1 寄存器的 RCIF 中断标志位就会被置 1。RCIF 中断标志位是只读位，不能用软件置 1 或清零。

将以下位置 1 可允许 RCIF 中断：

- PIE1 寄存器的中断允许位 RCIE
- INTCON 寄存器的外设中断允许位 PEIE
- INTCON 寄存器的全局中断允许位 GIE

当 FIFO 中存在未被读取的字符时，无论中断允许位的状态如何，RCIF 中断标志位均会被置 1。

22.1.2.4 接收帧错误

接收 FIFO 缓冲区中的每个字符都有相应的帧错误状态位。帧错误表明在预期时间内未见到停止位。通过 RCSTA 寄存器的 FERR 位可访问帧错误状态。FERR 位表示接收 FIFO 中顶部的未读字符的状态。因此，在读取 RCREG 前必须先读 FERR 位。

FERR 位是只读位，只用于接收 FIFO 中顶部的未读字符。帧错误（FERR = 1）并不会禁止接收更多字符。此时不必将 FERR 位清零。从 FIFO 缓冲区读出下一个字符将使 FIFO 进入下一个字符和下一个相应的帧错误。

将 RCSTA 寄存器的 SPEN 位清零可复位 EUSART，这样就可将 FERR 位强制清零。将 RCSTA 寄存器的 CREN 位清零不影响 FERR 位。自身产生的帧错误不会产生中断。

注：如果接收 FIFO 中的所有接收字符均有帧错误，反复读取 RCREG 不会将 FERR 位清零。

22.1.2.5 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。在访问 FIFO 前接收到完整的第三个字符时会产生溢出错误。此时，RCSTA 寄存器的 OERR 位置 1。FIFO 缓冲区中已有的字符可被读出，但溢出错误被清除前不能再接收其他字符。将 RCSTA 寄存器的 CREN 位清零或通过将 RCSTA 寄存器的 SPEN 位清零复位 EUSART，可清除该错误。

22.1.2.6 接收 9 位字符

EUSART 支持 9 位字符接收。当 RCSTA 寄存器的 RX9 位置 1 时，EUSART 将在接收每个字符时将 9 个位移入 RSR。RCSTA 寄存器的 RX9D 位是第 9 位，也是接收 FIFO 顶部未读字符的最高有效数据位。从接收 FIFO 缓冲区读取 9 位数据时，在读取 RCREG 的低 8 位前必须先读取 RX9D 数据位。

22.1.2.7 地址检测

当多个接收器共用同一条传输线时（如在 RS-485 系统中），有一个特殊的地址检测模式可供使用。将 RCSTA 寄存器的 ADDEN 位置 1 可使能地址检测。

地址检测要求接收 9 位字符。使能地址检测时，只有第 9 个数据位置 1 的字符会被传送到接收 FIFO 缓冲区，并将 RCIF 中断标志位置 1。所有其他字符均被忽略。

接收到地址字符后，用户软件可判断地址是否与自身匹配。地址匹配时，发生下一个停止位前，用户软件必须通过清零 ADDEN 位禁止地址检测。当用户软件根据所使用的报文协议检测到报文的末尾时，软件将 ADDEN 位置 1，将接收器重新置于地址检测模式。

22.1.2.8 异步接收设置

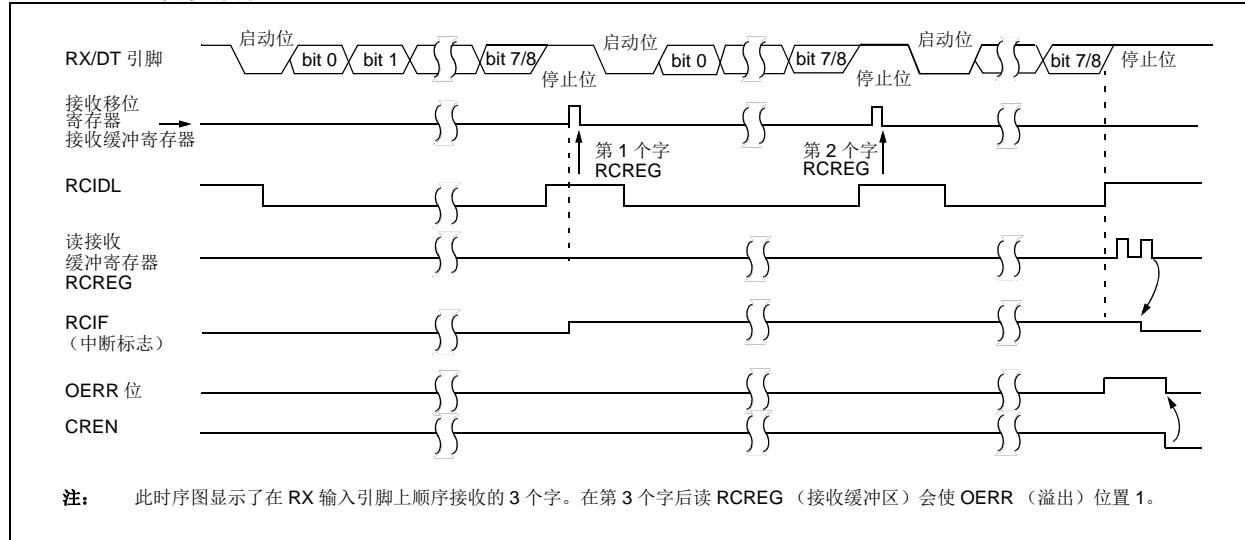
1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位，获得目标波特率（见 [第 22.4 节 “EUSART 波特率发生器”](#)）。
2. 清零 RX 引脚的 ANSEL 位（如适用）。
3. 将 SPEN 位置 1 使能串口。SYNC 位必须清零才能进行异步操作。
4. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 如果需要接收 9 位数据，将 RX9 位置 1。
6. 将 CREN 位置 1 使能接收。
7. 当字符从 RSR 被移入接收缓冲区时，RCIF 中断标志位将被置 1。如果 RCIE 中断允许位也置 1，则产生中断。
8. 读取 RCSTA 寄存器取得错误标志和第 9 个数据位（9 位数据接收使能时）。
9. 读取 RCREG 寄存器从接收缓冲区取得接收的 8 个低数据位。
10. 发生溢出时，通过清零 CREN 接收器使能位清零 OERR 标志位。

22.1.2.9 9 位地址检测模式设置

此模式通常用于 RS-485 系统中。设置使能地址检测的异步接收的步骤如下：

1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位，获得目标波特率（见 [第 22.4 节 “EUSART 波特率发生器”](#)）。
2. 清零 RX 引脚的 ANSEL 位（如适用）。
3. 将 SPEN 位置 1 使能串口。SYNC 位必须清零才能进行异步操作。
4. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 将 RX9 位置 1 使能 9 位接收。
6. 将 ADDEN 位置 1 使能地址检测。
7. 将 CREN 位置 1 使能接收。
8. 当第 9 位置 1 的字符从 RSR 被移入接收缓冲区时，RCIF 中断标志位将被置 1。如果 RCIE 中断允许位也置 1，则产生中断。
9. 读取 RCSTA 寄存器取得错误标志。第 9 个数据位将始终置 1。
10. 读取 RCREG 寄存器从接收缓冲区取得接收的 8 个低数据位。软件将判断此地址是否是器件地址。
11. 发生溢出时，通过清零 CREN 接收器使能位清零 OERR 标志位。
12. 如果器件被寻址，将 ADDEN 位清零以允许所有接收到的数据被送入接收缓冲区并产生中断。

图 22-5：异步接收



PIC16(L)F1574/5/8/9

表 22-2：与异步接收相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	204
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
RCREG	EUSART 接收数据寄存器								197*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	203*
SPBRGL	BRG<7:0>								205*
SPBRGH	BRG<15:8>								205*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D	202

图注：— = 未实现位，读为 0。异步接收不使用阴影单元。

* 提供寄存器信息的页。

22.2 异步操作的时钟精度

内部振荡器模块输出（INTOSC）在出厂时做了校准。但是，VDD 或温度变化时，INTOSC 频率有可能漂移，这将直接影响异步波特率。

自动波特率检测功能（见第 22.4.1 节“[自动波特率检测](#)”）可以用于补偿 INTOSC 频率的变化。

通过调整波特率发生器来补偿外设时钟频率的逐渐变化时，可能无法足够细微地调节分辨率。

PIC16(L)F1574/5/8/9

22.3 寄存器定义：EUSART 控制

寄存器 22-1：TXSTA：发送状态和控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDDB	BRGH	TRMT	TX9D
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7 **CSRC**: 时钟源选择位

异步模式:

无关

同步模式:

1 = 主模式（时钟由内部 BRG 产生）

0 = 从模式（时钟来自外部时钟源）

bit 6 **TX9**: 9 位发送使能位

1 = 选择 9 位发送

0 = 选择 8 位发送

bit 5 **TXEN**: 发送使能位⁽¹⁾

1 = 使能发送

0 = 禁止发送

bit 4 **SYNC**: EUSART 模式选择位

1 = 同步模式

0 = 异步模式

bit 3 **SENDDB**: 发送间隔字符位

异步模式:

1 = 在下一次发送时发送同步间隔字符（完成后由硬件清零）

0 = 同步间隔字符发送完成

同步模式:

无关

bit 2 **BRGH**: 高波特率选择位

异步模式:

1 = 高速

0 = 低速

同步模式:

在此模式下未使用

bit 1 **TRMT**: 发送移位寄存器状态位

1 = TSR 空

0 = TSR 满

bit 0 **TX9D**: 发送数据的第 9 位

可以是地址 / 数据位或奇偶校验位。

注 1: 在同步模式下，SREN/CREN 可改写 TXEN。

寄存器 22-2: RCSTA: 接收状态和控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7

SPEN: 串口使能位

1 = 使能串口 (配置 RX/DT 和 TX/CK 引脚作为串口引脚)

0 = 禁止串口 (保持在复位状态)

bit 6

RX9: 9 位接收使能位

1 = 选择 9 位接收

0 = 选择 8 位接收

bit 5

SREN: 单字符接收使能位

异步模式:

无关

同步模式 —— 主模式:

1 = 使能单字符接收

0 = 禁止单字符接收

此位在接收完成后清零。

同步模式 —— 从模式

无关

bit 4

CREN: 连续接收使能位

异步模式:

1 = 使能接收器

0 = 禁止接收器

同步模式:

1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 设置改写 SREN)

0 = 禁止连续接收

bit 3

ADDEN: 地址检测使能位

9 位异步模式 (RX9 = 1):

1 = 当 RSR<8> 置 1 时, 使能地址检测, 允许中断并装入接收缓冲区

0 = 禁止地址检测, 接收所有字节并且第 9 位可作为奇偶校验位

8 位异步模式 (RX9 = 0):

无关

bit 2

FERR: 帧错误位

1 = 帧错误 (可以通过读 RCREG 寄存器更新该位并接收下一个有效字节)

0 = 无帧错误

bit 1

OERR: 溢出错误位

1 = 溢出错误 (可以通过清零 CREN 位来清零该位)

0 = 无溢出错误

bit 0

RX9D: 接收数据的第 9 位

该位可以是地址 / 数据位或奇偶校验位, 并且必须由用户固件计算得到。

PIC16(L)F1574/5/8/9

寄存器 22-3： BAUDCON： 波特率控制寄存器

R/W-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7 **ABDOVF**: 自动波特率检测溢出位

异步模式:

1 = 自动波特率定时器溢出

0 = 自动波特率定时器未溢出

同步模式:

无关

bit 6 **RCIDL**: 接收空闲标志位

异步模式:

1 = 接收器空闲

0 = 已接收到启动位且接收器正在接收

同步模式:

无关

bit 5 **未实现**: 读为 0

SCKP: 同步时钟极性选择位

异步模式:

1 = 将反相数据发送到 TX/CK 引脚

0 = 将未反相数据发送到 TX/CK 引脚

同步模式:

1 = 数据在时钟上升沿同步

0 = 数据在时钟下降沿同步

bit 3 **BRG16**: 16 位波特率发生器位

1 = 使用 16 位波特率发生器

0 = 使用 8 位波特率发生器

未实现: 读为 0

bit 1 **WUE**: 唤醒使能位

异步模式:

1 = 接收器正在等待下降沿。不会接收字符。 RCIF 位将被置 1。 RCIF 置 1 后 WUE 将被自动清零。

0 = 接收器正常工作

同步模式:

无关

bit 0 **ABDEN**: 自动波特率检测使能位

异步模式:

1 = 使能自动波特率检测模式（完成自动波特率检测后清零）

0 = 禁止自动波特率检测模式

同步模式:

无关

22.4 EUSART 波特率发生器

波特率发生器（Baud Rate Generator, BRG）是 8 位或 16 位定时器，专用于支持异步和同步 EUSART 操作。默认情况下，BRG 工作在 8 位模式下。将 BAUDCON 寄存器的 BRG16 位置 1 可选择 16 位模式。

SPBRGH 和 SPBRGL 寄存器对决定自由运行波特率定时器的周期。在异步模式下，波特率周期的倍频值由 TXSTA 寄存器的 BRGH 位和 BAUDCON 寄存器的 BRG16 位决定。在同步模式下，BRGH 位被忽略。

表 22-3 包含了用于确定波特率的公式。例 22-1 给出了确定波特率和波特率误差的计算示例。

为便于您使用，各种异步模式的典型波特率和误差值已经计算出来，如表 22-3 所示。使用高波特率（BRGH = 1）或 16 位 BRG（BRG16 = 1）有助于降低波特率误差。16 位 BRG 模式用于在高速振荡器频率下实现低波特率。

将新值写入 SPBRGH 和 SPBRGL 寄存器对将导致 BRG 定时器复位（或清零）。这可以确保 BRG 无需等待定时器溢出就可以输出新的波特率。

如果系统时钟在有效的接收操作过程中被更改，可能会导致接收错误或数据丢失。为避免此问题，应检查 RCIDL 位的状态，以确保在改变系统时钟前接收操作处于空闲状态。

例 22-1：计算波特率误差

针对工作在异步模式下、 $F_{OSC} = 16\text{ MHz}$ 、目标波特率 = 9600 且采用 8 位 BRG 的器件：

$$\text{目标波特率} = \frac{F_{OSC}}{64(\lceil SPBRGH:SPBRGL \rceil + 1)}$$

求解 SPBRGH:SPBRGL：

$$X = \frac{\frac{F_{OSC}}{\text{目标波特率}} - 1}{64}$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{计算得出波特率} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{错误} = \frac{\text{计算得出波特率} - \text{目标波特率}}{\text{目标波特率}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

PIC16(L)F1574/5/8/9

表 22-3： 波特率公式

配置位			BRG/EUSART 模式	波特率公式
SYNC	BRG16	BRGH		
0	0	0	8 位 / 异步	Fosc/[64 (n+1)]
0	0	1	8 位 / 异步	Fosc/[16 (n+1)]
0	1	0	16 位 / 异步	
0	1	1	16 位 / 异步	Fosc/[4 (n+1)]
1	0	x	8 位 / 同步	
1	1	x	16 位 / 同步	

图注： x = 无关位， n = SPBRGH:SPBRGL 寄存器对的值。

表 22-4： 与波特率发生器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	204
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	203
SPBRGL	BRG<7:0>								205*
SPBRGH	BRG<15:8>								205*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	202

图注： — = 未实现位，读为 0。波特率发生器不使用阴影单元。

* 提供寄存器信息的页。

表 22-5： 异步模式下的波特率

波特率	SYNC = 0, BRGH = 0, BRG16 = 0															
	Fosc = 20.000 MHz				Fosc = 18.432 MHz				Fosc = 16.000 MHz				Fosc = 11.0592 MHz			
	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值				
300	—	—	—	—	—	—	—	—	—	—	—	—				
1200	1221	1.73	255	1200	0.00	239	1202	0.16	207	1200	0.00	143				
2400	2404	0.16	129	2400	0.00	119	2404	0.16	103	2400	0.00	71				
9600	9470	-1.36	32	9600	0.00	29	9615	0.16	25	9600	0.00	17				
10417	10417	0.00	29	10286	-1.26	27	10417	0.00	23	10165	-2.42	16				
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.23k	0.16	12	19.20k	0.00	8				
57.6k	—	—	—	57.60k	0.00	7	—	—	—	57.60k	0.00	2				
115.2k	—	—	—	—	—	—	—	—	—	—	—	—				

波特率	SYNC = 0, BRGH = 0, BRG16 = 0															
	Fosc = 8.000 MHz				Fosc = 4.000 MHz				Fosc = 3.6864 MHz				Fosc = 1.000 MHz			
	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值				
300	—	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51			
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12				
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—				
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—				
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—				
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—				
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—				
115.2k	—	—	—	—	—	—	—	—	—	—	—	—				

波特率	SYNC = 0, BRGH = 1, BRG16 = 0															
	Fosc = 20.000 MHz				Fosc = 18.432 MHz				Fosc = 16.000 MHz				Fosc = 11.0592 MHz			
	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值	实际 波特率	SPBRG 值				
300	—	—	—	—	—	—	—	—	—	—	—	—				
1200	—	—	—	—	—	—	—	—	—	—	—	—				
2400	—	—	—	—	—	—	—	—	—	—	—	—				
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71				
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65				
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35				
57.6k	56.82k	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11				
115.2k	113.64k	-1.36	10	115.2k	0.00	9	111.1k	-3.55	8	115.2k	0.00	5				

PIC16(L)F1574/5/8/9

表 22-5： 异步模式下的波特率（续）

波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)
300	—	—	—	—	—	—	—	—	300	0.16	207	
1200	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51	
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 16.000 MHz			Fosc = 11.0592 MHz		
	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)
300	300.0	-0.01	4166	300.0	0.00	3839	300.03	0.01	3332	300.0	0.00	2303
1200	1200	-0.03	1041	1200	0.00	959	1200.5	0.04	832	1200	0.00	575
2400	2399	-0.03	520	2400	0.00	479	2398	-0.08	416	2400	0.00	287
9600	9615	0.16	129	9600	0.00	119	9615	0.16	103	9600	0.00	71
10417	10417	0.00	119	10378	-0.37	110	10417	0.00	95	10473	0.53	65
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.23k	0.16	51	19.20k	0.00	35
57.6k	56.818	-1.36	21	57.60k	0.00	19	58.82k	2.12	16	57.60k	0.00	11
115.2k	113.636	-1.36	10	115.2k	0.00	9	111.11k	-3.55	8	115.2k	0.00	5

波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)	实际 波特率	SPBRG 值 误差 % (十进制)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

表 22-5： 异步模式下的波特率（续）

波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 20.000 MHz				Fosc = 18.432 MHz				Fosc = 16.000 MHz			
	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)
300	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	13332	300.0	0.00	9215
1200	1200	-0.01	4166	1200	0.00	3839	1200.1	0.01	3332	1200	0.00	2303
2400	2400	0.02	2082	2400	0.00	1919	2399.5	-0.02	1666	2400	0.00	1151
9600	9597	-0.03	520	9600	0.00	479	9592	-0.08	416	9600	0.00	287
10417	10417	0.00	479	10425	0.08	441	10417	0.00	383	10433	0.16	264
19.2k	19.23k	0.16	259	19.20k	0.00	239	19.23k	0.16	207	19.20k	0.00	143
57.6k	57.47k	-0.22	86	57.60k	0.00	79	57.97k	0.64	68	57.60k	0.00	47
115.2k	116.3k	0.94	42	115.2k	0.00	39	114.29k	-0.79	34	115.2k	0.00	23

波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 8.000 MHz				Fosc = 4.000 MHz				Fosc = 3.6864 MHz			
	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)	实际 波特率	SPBRG 值 (十进制)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

22.4.1 自动波特率检测

EUSART 模块支持波特率自动检测和校准。

在自动波特率检测（Auto-Baud Detect, ABD）模式下，提供给 BRG 的时钟信号是反向的。BRG 并不为传入的 RX 信号提供时钟信号，而是由 RX 信号为 BRG 定时。波特率发生器用于为接收的 55h (ASCII “U”) 的周期定时，55h 是 LIN 总线的同步字符。此字符的特殊之处在于它具有包括停止位边沿在内的 5 个上升沿。

通过将 BAUDCON 寄存器的 ABDEN 位置 1，可以启动自动波特率校准序列（图 22-6）。当发生 ABD 序列时，EUSART 状态机保持在空闲状态。在接收线的第一个上升沿（启动位之后），SPBRG 使用 BRG 计数器时钟递增计数，如表 22-6 所示。在第 8 位周期的末尾将在 RX 引脚上出现第 5 个上升沿。此时，累计值（即正确的 BRG 周期总数）被保存在 SPBRGH 和 SPBRGL 寄存器对中，ABDEN 位被自动清零而 RCIF 中断标志被置 1。要清除 RCIF 中断，需要读取 RCREG 中的值。

RCREG 的内容应该被丢弃。在不使用 SPBRGH 寄存器的模式下进行校准时，用户可通过查询 SPBRGH 寄存器的值是否为 00h 来验证 SPBRGL 寄存器是否溢出。

BRG 自动波特率时钟由 BRG16 和 BRGH 位决定，如表 22-6 所示。在 ABD 期间，SPBRGH 和 SPBRGL 寄存器都被用作 16 位计数器，与 BRG16 位的设置无关。

在校准波特率周期时，SPBRGH 和 SPBRGL 寄存器的时钟频率为 BRG 基本时钟频率的 1/8。得到的字节测量结果为全速时的平均位时间。

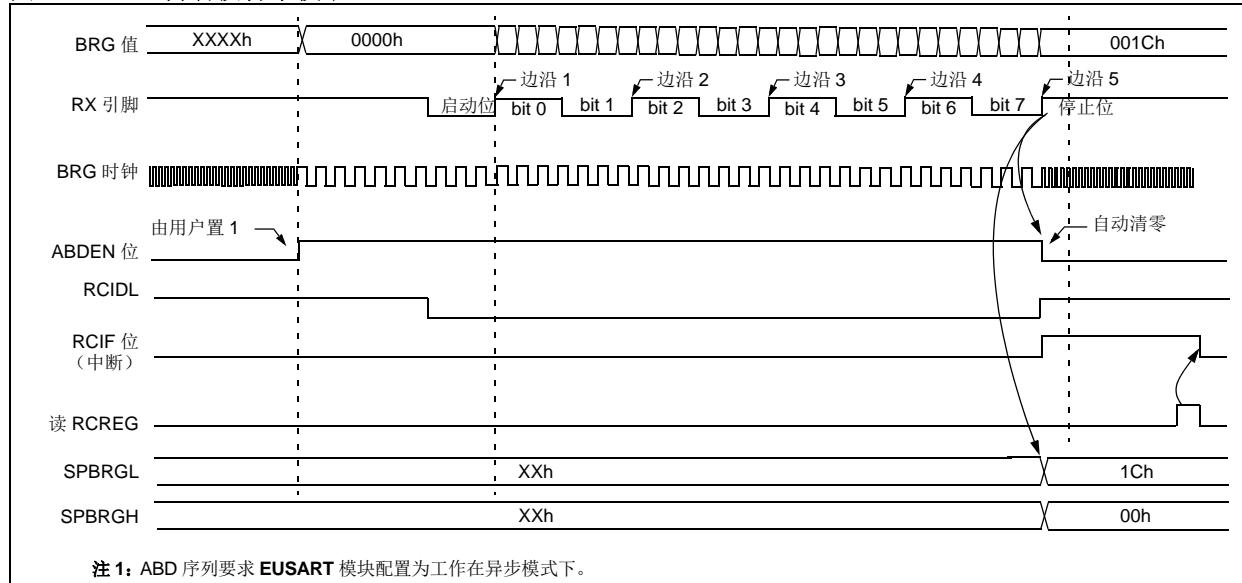
- 注 1:** 如果 WUE 位和 ABDEN 位都置 1，自动波特率检测将从间隔字符之后的字节开始（见第 22.4.3 节“接收到间隔字符时自动唤醒”）。
- 2:** 需要由用户来判断输入字符的波特率是否处于所选 BRG 时钟源范围内。可能无法实现某些振荡器频率和 EUSART 波特率组合。
- 3:** 在自动波特率过程中，自动波特率计数器从 1 开始计数。自动波特率序列完成后，为了得到最准确的结果，应从 SPBRGH:SPBRGL 寄存器对的值中减去 1。

表 22-6：BRG 计数器时钟速率

BRG16	BRGH	BRG 基本时钟	BRG ABD 时钟
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

注: 在 ABD 序列期间，SPBRGL 和 SPBRGH 寄存器都被用作 16 位计数器，与 BRG16 的设置无关。

图 22-6：自动波特率校准



22.4.2 自动波特率溢出

在自动波特率检测过程中，如果在 RX 引脚上检测到第 5 个上升沿之前波特率计数器溢出，则 BAUDCON 寄存器的 ABDOVF 位将被置 1。ABDOVF 位指示计数器已超出 SPBRGH:SPBRL 寄存器对的 16 位所能允许的最大计数值。溢出条件会将 RCIF 标志置 1。计数器将继续计数，直到在 RX 引脚上检测到第 5 个上升沿为止。RCIDL 位会一直保持为假 (0)，直到出现第 5 个上升沿，此时 RDICL 位会置 1。如果在发生溢出之后但在第 5 个上升沿之前读取 RCREG，第 5 个上升沿会再次将 RCIF 置 1。

如果通过提前终止自动波特率检测过程来清除溢出条件，将无法正确检测同步字符的第 5 个上升沿。如果在 ABDEN 位清零时尚未出现同步字符的任何下降沿，则它们会被错误地检测为启动位。要清除溢出条件，建议使用以下步骤：

1. 通过读取 RCREG 来清零 RCIF。
2. 如果 RCIDL 为 0，则等待 RCIF 并重复步骤 1。
3. 清零 ABDOVF 位。

22.4.3 接收到间隔字符时自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于不工作状态，不能正常进行字符接收。自动唤醒功能使控制器可被 RX/DT 线上的活动唤醒。该功能只在异步模式下可用。

自动唤醒功能可通过将 BAUDCON 寄存器的 WUE 位置 1 来使能。一旦置 1，RX/DT 上的正常接收序列就被禁止，EUSART 保持在空闲状态，监视与 CPU 模式无关的唤醒事件。唤醒事件包含 RX/DT 线上电平由高至低的跳变。(这与 LIN 协议的同步间隔字符或唤醒信号字符的启动条件一致。)

EUSART 模块产生的 RCIF 中断与唤醒事件同步。在正常 CPU 工作模式下，与 Q 时钟同步产生中断（图 22-7）；在休眠模式下，与 Q 时钟异步产生中断（图 22-8）。通过读 RCREG 寄存器可清除中断条件。

RX 线在间隔字符末尾由低至高的跳变将自动清零 WUE 位。这向用户表明间隔事件结束。此时，EUSART 模块处于空闲模式，等待接收下一个字符。

22.4.3.1 特殊注意事项

间隔字符

在发生唤醒事件期间为了避免字符错误或字符碎片，唤醒字符必须为全零。

唤醒被使能时，其工作状况与数据流的低电平时间无关。如果 WUE 位置 1 并接收到有效的非零字符，则从启动位至第一个上升沿的低电平时间将被解读为唤醒事件。字符的其余位将作为碎片字符接收，后续字符有可能产生帧错误或溢出错误。

因此，发送的首字符必须为全 0。这必须持续 10 个或更长的位时间，对于 LIN 总线，建议持续 13 个位时间，而对于标准 RS-232 器件，可为任意个位时间。

振荡器起振时间

必须考虑振荡器起振时间，特别在使用起振时间较长的振荡器（即，LP、XT 或 HS/PLL 模式）的应用中。同步间隔（或唤醒信号）字符必须足够长，并随后有一个足够长的间隔时间，以使所选的振荡器有足够的起振时间并在这段时间对 EUSART 进行正确初始化。

WUE 位

唤醒事件会通过将 RCIF 位置 1 产生一个接收中断。WUE 位在 RX/DT 的上升沿由硬件清零。然后，通过读取 RCREG 寄存器并丢弃其内容，可以由软件清除中断条件。

要确保不丢失实际数据，应在将 WUE 位置 1 前检查 RCIDL 位，验证没有接收操作在进行。如果未发生接收操作，可在进入休眠模式前将 WUE 位置 1。

PIC16(L)F1574/5/8/9

图 22-7：正常工作时的自动唤醒位（WUE）时序

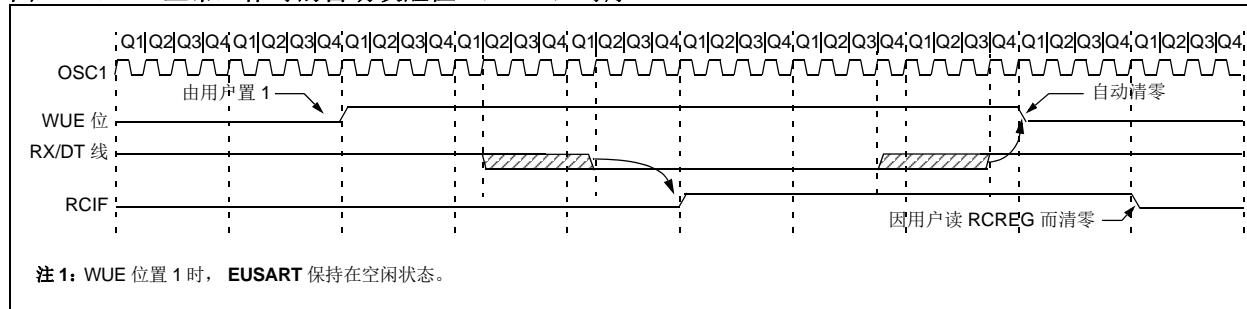
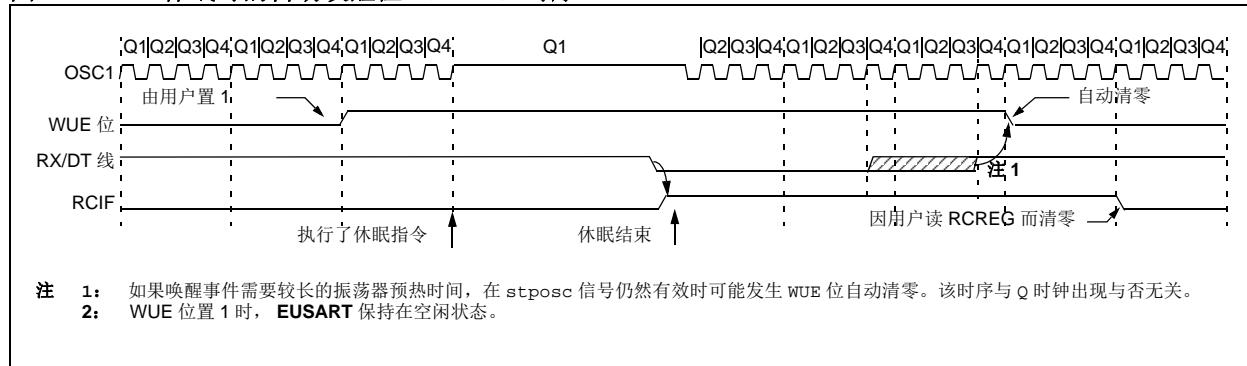


图 22-8：休眠时的自动唤醒位（WUE）时序



22.4.4 间隔字符序列

EUSART 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。间隔字符包含 1 个启动位以及随后的 12 个 0 位和 1 个停止位。

要发送间隔字符，应将 TXSTA 寄存器的 SENDB 和 TXEN 位置 1。随后对 TXREG 执行写操作可启动间隔字符发送。写入 TXREG 的数据值会被忽略并发送全 0。

在发送了相应的停止位后，硬件会自动将 SENDB 位复位。这样用户可以在间隔字符（在 LIN 规范中通常是同步字符）后预先将下一个要发送字节装入发送 FIFO。

TXSTA 寄存器的 TRMT 位表明发送操作何时处于有效或空闲状态，这与正常发送时相同。图 22-9 给出了发送间隔字符的时序。

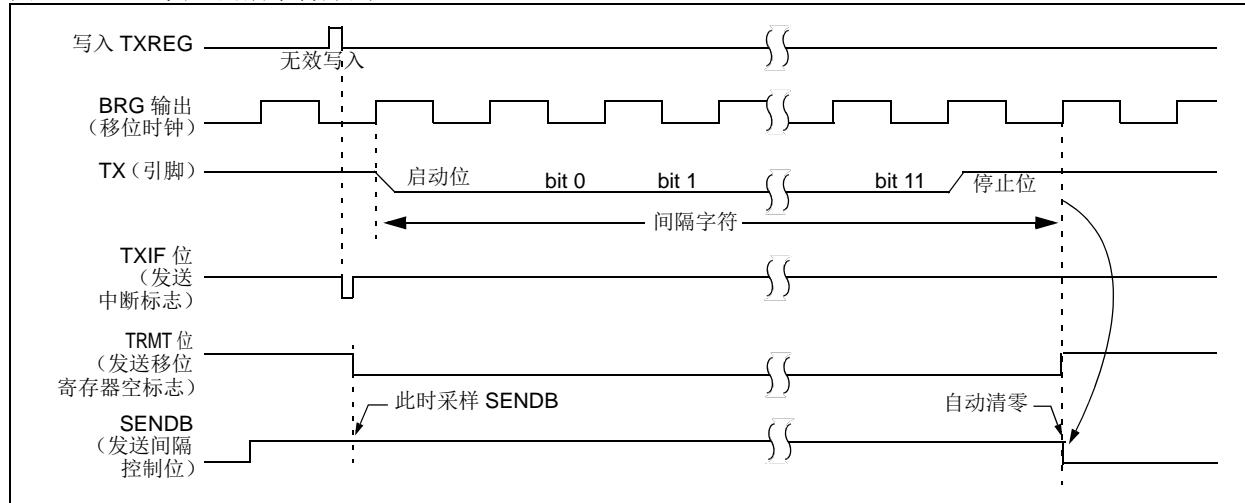
22.4.4.1 间隔和同步发送序列

以下序列将启动报文帧头，它由间隔字符和其后的自动波特率同步字节组成。此序列适用于典型的 LIN 总线主器件。

1. 将 EUSART 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置 1 使能间隔序列。
3. 将无效字符装入 TXREG，启动发送（该值会被忽略）。
4. 将 “55h” 写入 TXREG，以便将同步字符装入发送 FIFO 缓冲区。
5. 发送间隔字符后，SENDB 位被硬件复位，同步字符随后被发送。

当 TXREG 为空时（由 TXIF 指示），下一个数据字节会写入 TXREG。

图 22-9：发送间隔字符序列



22.4.5 接收间隔字符

增强型 EUSART 模块接收间隔字符有两种方法。

第一种检测间隔字符的方法采用 RCSTA 寄存器的 FERR 位和如 RCREG 所指示的接收数据。假定波特率发生器已初始化为所需的波特率。

发生以下情况时，表明接收到间隔字符：

- RCIF 位被置 1
- FERR 位被置 1
- RCREG = 00h

第二种方法采用第 22.4.3 节“接收到间隔字符时自动唤醒”中所述的自动唤醒功能。通过使能此功能，EUSART 将采样 RX/DT 上的下两次跳变，产生 RCIF 中断，并接收下一个数据字节并再产生一次中断。

请注意，在间隔字符后，用户通常希望使能自动波特率检测功能。采用这两种方法时，用户均可在 EUSART 进入休眠模式前将 BAUDCON 寄存器的 ABDEN 位置 1。

22.5 EUSART 同步模式

同步串行通信通常用于具有一个主器件和一个或多个从器件的系统中。主器件包含生成波特率所需的电路，可将时钟提供给系统中的所有器件。从器件使用主时钟，可不再需要内部时钟生成电路。

同步模式下具有两条信号线：双向数据线和时钟线。从器件使用主器件提供的外部时钟将串行数据移入或移出相应的接收和发送移位寄存器。由于数据线是双向的，同步操作只能是半双工的。半双工指从器件能够接收和发送数据，但不能同时进行。EUSART 可作为主器件，也可作为从器件。

同步发送时不使用启动位和停止位。

22.5.1 同步主模式

使用以下位将 EUSART 配置为同步主操作：

- SYNC = 1
- CSRC = 1
- SREN = 0 (用于发送) ; SREN = 1 (用于接收)
- CREN = 0 (用于发送) ; CREN = 1 (用于接收)
- SPEN = 1

通过将 TXSTA 寄存器的 SYNC 位置 1，可将器件配置为同步操作。通过将 TXSTA 寄存器的 CSRC 位置 1，可将器件配置为主器件。通过将 RCSTA 寄存器的 SREN 和 CREN 位清零，可确保器件处于发送模式，否则器件将被配置为接收。通过将 RCSTA 寄存器的 SPEN 位置 1，可使能 EUSART。

22.5.1.1 主时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为主器件的器件将时钟信号发送到 TX/CK 线上。

EUSART 配置为同步发送或接收操作时，自动使能 TX/CK 引脚输出驱动器。串行数据位在时钟前沿改变，以确保其在每个时钟的后沿有效。为每个数据位产生一个时钟周期。数据位有多少，就产生多少个时钟周期。

22.5.1.2 时钟极性

为了与 Microwire 兼容，提供了时钟极性选项。时钟极性通过 BAUDCON 寄存器的 SCKP 位进行选择。将 SCKP 位置 1 时，可将时钟空闲状态设置为高电平。当

SCKP 位置 1 时，数据在每个时钟的下降沿改变。将 SCKP 位清零将时钟空闲状态设置为低电平。当 SCKP 位清零时，数据在每个时钟的上升沿改变。

22.5.1.3 同步主发送

从器件的 RX/DT 引脚传输输出数据。EUSART 配置为同步主发送操作时，RX/DT 和 TX/CK 引脚的输出驱动器被自动使能。

向 TXREG 寄存器写入一个字符时启动发送。如果 TSR 中仍保存前一个字符的全部或部分，则新字符数据被保存在 TXREG 中，直到前一个字符的最后一一位被发送。如果这是首字符，或前一个字符被完全从 TSR 中送出，TXREG 中的数据就立即被传送到 TSR。字符发送在数据从 TXREG 送入 TSR 后立即开始。

每个数据位在主时钟的时钟前沿改变，并在下一个时钟前沿到来前保持有效。

注： TSR 寄存器不映射到数据存储器中，因此用户无法使用。

22.5.1.4 同步主发送设置

1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位，获得目标波特率（见第 22.4 节“EUSART 波特率发生器”）。
2. 将 SYNC、SPEN 和 CSRC 位置 1，以使能同步主串口。
3. 将 SREN 和 CREN 位清零，以禁止接收模式。
4. 将 TXEN 位置 1，以使能发送模式。
5. 如果需要 9 位发送，将 TX9 位置 1。
6. 如果需要中断，将 PIE1 寄存器的 TXIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
7. 如果选择了 9 位发送，应将第 9 位装入 TX9D 位。
8. 将数据装入 TXREG 寄存器，启动发送。

图 22-10：同步发送

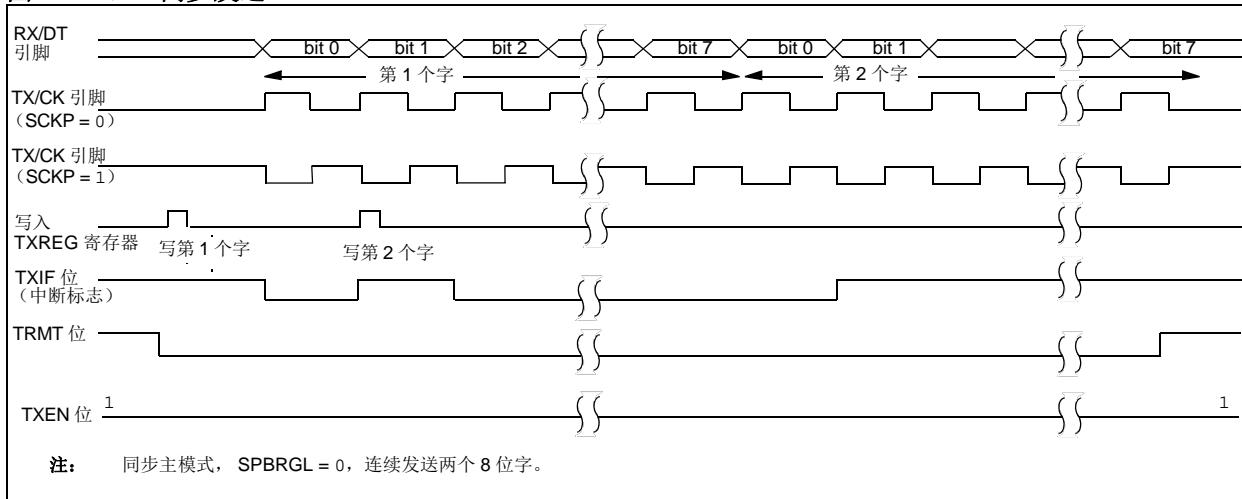


图 22-11：同步发送（由 TXEN 位控制）

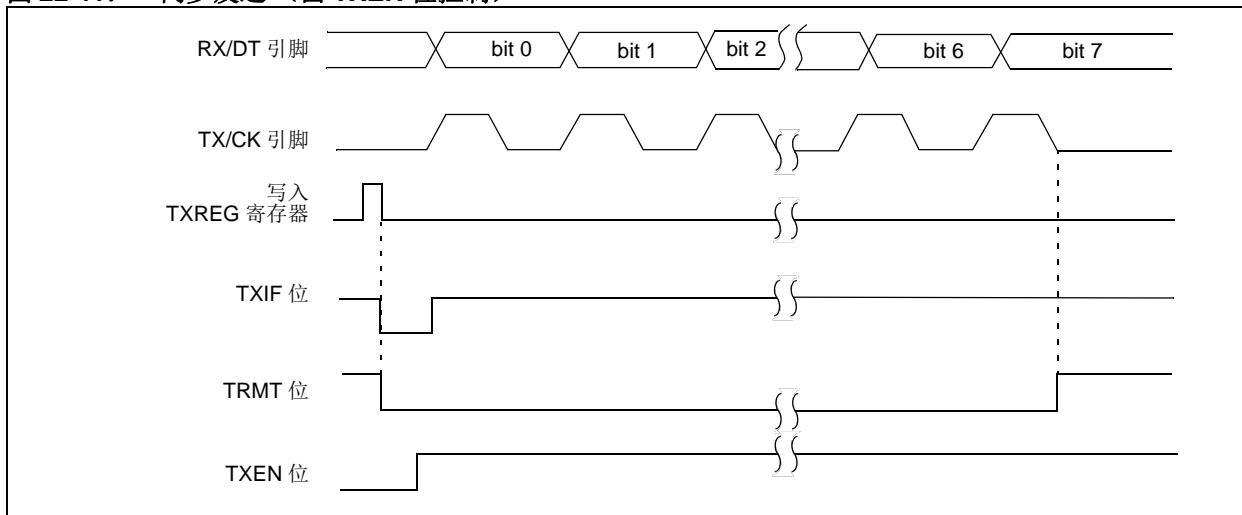


表 22-7：与同步主发送相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	204
INTCON	GIE	PEIE	TMROIE	INTE	IOCIE	TMROIF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	203
SPBRGL	BRG<7:0>								205*
SPBRGH	BRG<15:8>								205*
TXREG	EUSART 发送数据寄存器								194*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDDB	BRGH	TRMT	TX9D	202

图注：— = 未实现位，读为 0。同步主发送不使用阴影单元。

* 提供寄存器信息的页。

22.5.1.5 同步主接收

数据在 RX/DT 引脚上接收。将 EUSART 配置为同步主接收操作时，自动禁止 RX/DT 引脚输出驱动器。

在同步模式下，可通过将单字符接收使能位（RCSTA 寄存器的 SREN）或连续接收使能位（RCSTA 寄存器的 CREN）置 1 使能接收。

SREN 置 1 且 CREN 清零时，一个字符中有多少数据位就产生多少个时钟周期。一个字符接收完成后 SREN 位被自动清零。CREN 置 1 时，将连续产生时钟直到 CREN 被清零。如果 CREN 在字符接收过程中被清零，则 CK 时钟立即停止，接收到的部分字符被丢弃。如果 SREN 和 CREN 同时置 1，则首字符接收完成时 SREN 被清零，CREN 优先。

要启动接收，将 SREN 或 CREN 置 1。在 TX/CK 时钟引脚的后沿对 RX/DT 引脚上的数据进行采样，并移入接收移位寄存器（RSR）。在完整的字符被接收进 RSR 后，RCIF 位置 1 且该字符被自动送入两个字符的接收 FIFO。接收 FIFO 中顶部字符的低 8 位在 RCREG 中。只要接收 FIFO 中有未读字符，RCIF 位就保持置 1。

注： 如果 RX/DT 功能位于模拟引脚上，则必须清零相应的 ANSEL 位使接收器工作。

22.5.1.6 从时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为从器件的器件在 TX/CK 线上接收时钟信号。将器件配置为同步从发送或接收操作时，自动禁止 TX/CK 引脚输出驱动器。串行数据位在时钟前沿改变，以确保其在每个时钟的后沿有效。每个时钟周期传送一个数据位。数据位有多少，就产生多少个接收时钟周期。

注： 如果将器件配置为从器件并且 TX/CK 功能位于模拟引脚上，则必须清零相应的 ANSEL 位。

22.5.1.7 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。在 RCREG 被读取以访问 FIFO 前，接收到完整的第三个字符时会产生溢出错误。此时，RCSTA 寄存器的 OERR 位置 1。FIFO 中的前一个数据不会被覆盖。FIFO 缓冲区中的两个字符可被读出，但错误被清除前不能再接收其他字符。只

有清除了溢出条件，才可将 OERR 位清零。如果 SREN 位置 1 且 CREN 清零时发生溢出错误，则读取 RCREG 可清除错误。如果 CREN 位置 1 时发生溢出错误，则通过清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位（该位将 EUSART 复位），可以清除错误条件。

22.5.1.8 接收 9 位字符

EUSART 支持 9 位字符接收。当 RCSTA 寄存器的 RX9 位置 1 时，EUSART 将在接收每个字符时将 9 个位移入 RSR。RCSTA 寄存器的 RX9D 位是第 9 位，也是接收 FIFO 顶部未读字符的最高数据位。从接收 FIFO 缓冲区读取 9 位数据时，在读取 RCREG 的低 8 位前必须先读取 RX9D 数据位。

22.5.1.9 同步主接收设置

1. 初始化 SPBRGH:SPBRGL 寄存器对，获得所需的波特率。按需将 BRGH 和 BRG16 位置 1 或清零，获得目标波特率。
2. 清零 RX 引脚的 ANSEL 位（如适用）。
3. 将 SYNC、SPEN 和 CSRC 位置 1，以使能同步主串口。
4. 确保将 CREN 和 SREN 位清零。
5. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
6. 如果需要接收 9 位数据，将 RX9 位置 1。
7. 将 SREN 位置 1 启动接收，或将 CREN 位置 1 使能连续接收。
8. 字符接收完成时中断标志位 RCIF 将被置 1。如果中断允许位 RCIE 已置 1，则产生中断。
9. 读取 RCSTA 寄存器取得第 9 位（如果已使能），并确定接收时是否发生了错误。
10. 通过读取 RCREG 寄存器来读取接收到的 8 位数据。
11. 如果发生了溢出错误，通过清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位（该位将 EUSART 复位），可以清除错误。

图 22-12: 同步接收（主模式， SREN）

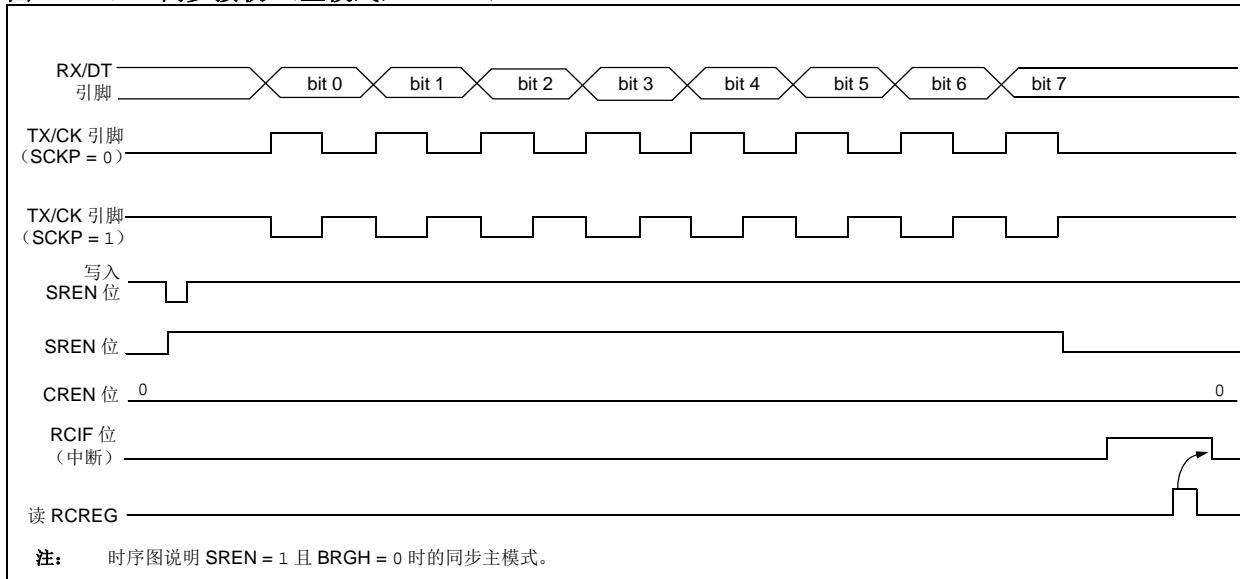


表 22-8: 与同步主接收相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BAUDCON	ABDOVF	RIDL	—	SCKP	BRG16	—	WUE	ABDEN	204
INTCON	GIE	PEIE	TMROIE	INTE	IOCIE	TMROIF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
RCREG	EUSART 接收数据寄存器								197*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	203
SPBRGL	BRG<7:0>								205*
SPBRGH	BRG<15:8>								205*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	202

图注：— = 未实现位，读为 0。同步主接收不使用阴影单元。

* 提供寄存器信息的页。

22.5.2 同步从模式

使用以下位将 EUSART 配置为同步从操作：

- SYNC = 1
- CSRC = 0
- SREN = 0 (用于发送); SREN = 1 (用于接收)
- CREN = 0 (用于发送); CREN = 1 (用于接收)
- SPEN = 1

通过将 TXSTA 寄存器的 SYNC 位置 1，可将器件配置为同步操作。通过将 TXSTA 寄存器的 CSRC 位清零，可将器件配置为从器件。通过将 RCSTA 寄存器的 SREN 和 CREN 位清零，可确保器件处于发送模式，否则器件将被配置为接收。通过将 RCSTA 寄存器的 SPEN 位置 1，可使能 EUSART。

22.5.2.1 EUSART 同步从发送

除了休眠模式以外，同步主模式和从模式的工作原理是相同的（见第 22.5.1.3 节“同步主发送”）。

如果向 TXREG 写入两个字，然后执行 SLEEP 指令，则会发生以下事件：

1. 第一个字符将立即传送到 TSR 寄存器并发送。
2. 第二个字将保留在 TXREG 寄存器中。
3. TXIF 位不会被置 1。
4. 第一个字符移出 TSR 后，TXREG 寄存器会将第二个字符传送到 TSR，此时 TXIF 位将置 1。
5. 如果 PEIE 和 TXIE 位均置 1，则发生中断，将器件从休眠唤醒，并执行下一条指令。如果 GIE 位也置 1，程序将调用中断服务程序。

22.5.2.2 同步从发送设置

1. 将 SYNC 和 SPEN 位置 1 并清零 CSRC 位。
2. 清零 CK 引脚的 ANSEL 位（如适用）。
3. 清零 CREN 和 SREN 位。
4. 如果需要中断，将 PIE1 寄存器的 TXIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 如果需要 9 位发送，将 TX9 位置 1。
6. 将 TXEN 位置 1，以使能发送。
7. 如果选择了 9 位发送，将最高有效位插入 TX9D 位。
8. 将低 8 位写入 TXREG 寄存器，启动发送。

表 22-9：与同步从发送相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	204
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	203
TXREG	EUSART 发送数据寄存器								194*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	202

图注：— = 未实现位，读为 0。同步从发送不使用阴影单元。

* 提供寄存器信息的页。

22.5.2.3 EUSART 同步从接收

除下列各项外，同步主模式和从模式的工作原理是相同的（[第 22.5.1.5 节“同步主接收”](#)）：

- 休眠
- CREN 位始终置 1，因此接收器从不空闲
- SREN 位在从模式下为“无关位”

进入休眠前将 CREN 置位 1，可在休眠模式下接收一个字符。接收到该字后，RSR 寄存器会将数据传输到 RCREG 寄存器。如果 RCIE 中断允许位置 1，产生的中断会将器件从休眠唤醒并执行下一条指令。如果 GIE 位也置 1，程序将跳转到中断向量。

22.5.2.4 同步从接收设置

1. 将 SYNC 和 SPEN 位置 1 并清零 CSRC 位。
2. 清零 CK 和 DT 引脚的 ANSEL 位（如适用）。
3. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
4. 如果需要接收 9 位数据，将 RX9 位置 1。
5. 将 CREN 置位 1，以便能接收。
6. 接收完成时 RCIF 位将被置 1。如果 RCIE 位已置 1，则产生中断。
7. 如果使能了 9 位模式，从 RCSTA 寄存器的 RX9D 位取出最高有效位。
8. 读取 RCREG 寄存器，从接收 FIFO 取出低 8 位。
9. 如果发生了溢出错误，通过清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位（该位将 EUSART 复位），可以清除错误。

表 22-10：与同步从接收相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	204
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
RCREG	EUSART 接收数据寄存器								197*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	203
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	202

图注：— = 未实现位，读为 0。同步从接收不使用阴影单元。

* 提供寄存器信息的页。

23.0 16 位脉宽调制 (PWM) 模块

脉宽调制 (Pulse-Width Modulation, PWM) 模块可产生由相位、占空比、周期和偏移事件计数决定的脉宽调制信号，相位、占空比、周期和偏移事件计数则包含在以下寄存器中：

- PWMxPH 寄存器
- PWMxDC 寄存器
- PWMxPR 寄存器
- PWMxOF 寄存器

图 23-1 给出了 PWM 操作的简化框图。

每个 PWM 模块均具有 4 种工作模式：

- 标准
- 匹配时电平置 1
- 匹配时电平翻转
- 中心对齐

关于每种 PWM 模式的更详细说明，请参见第 23.2 节“**PWM 模式**”。

图 23-1：16 位 PWM 框图

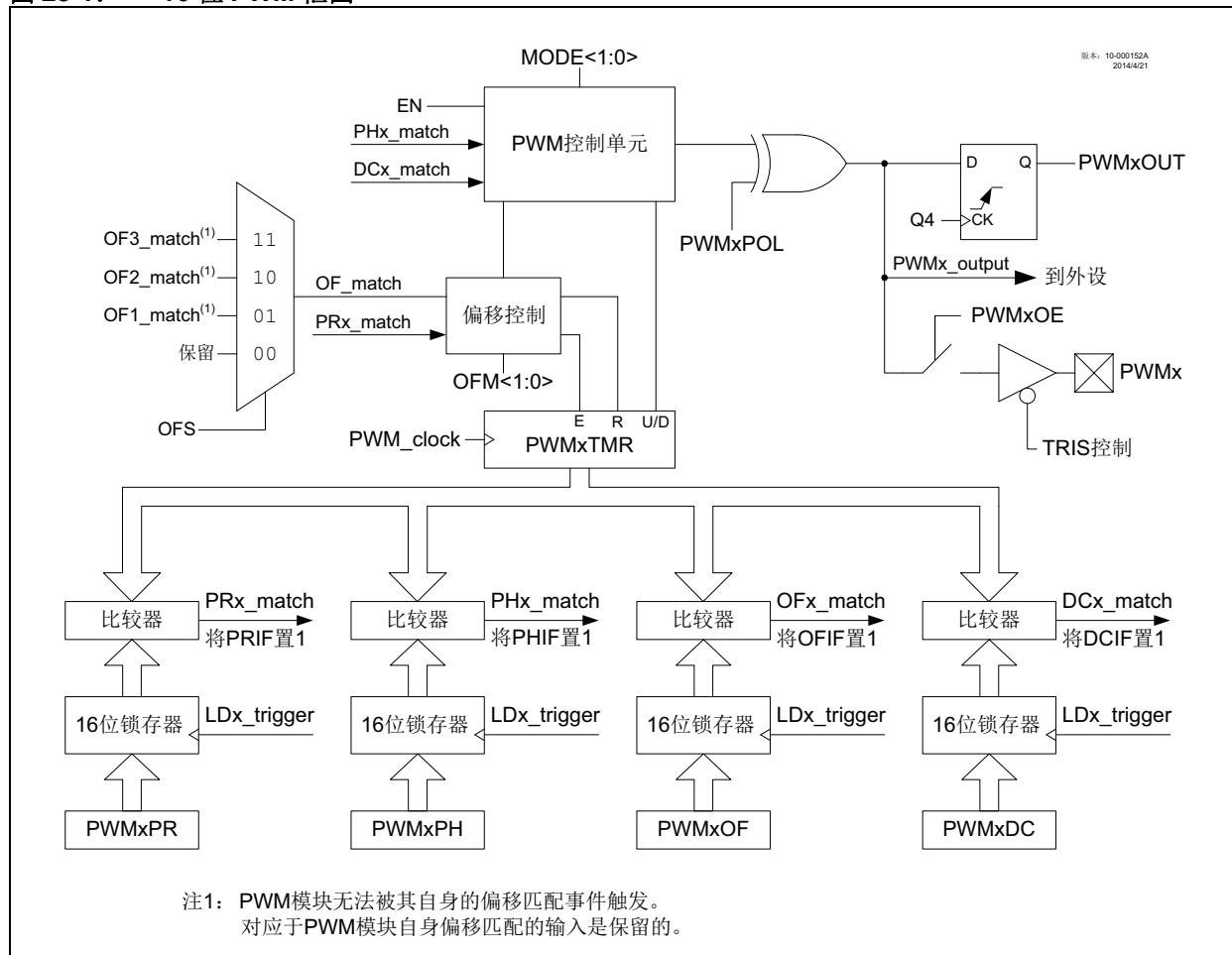
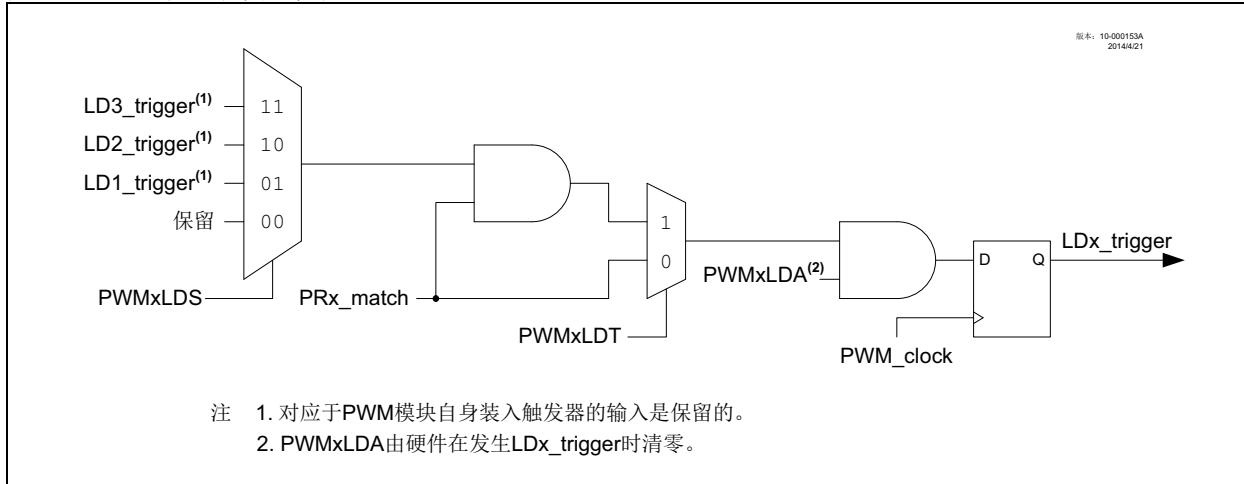


图 23-2：装入触发器框图



23.1 基本操作

PWM 模块可产生一个 16 位分辨率的脉宽调制输出。

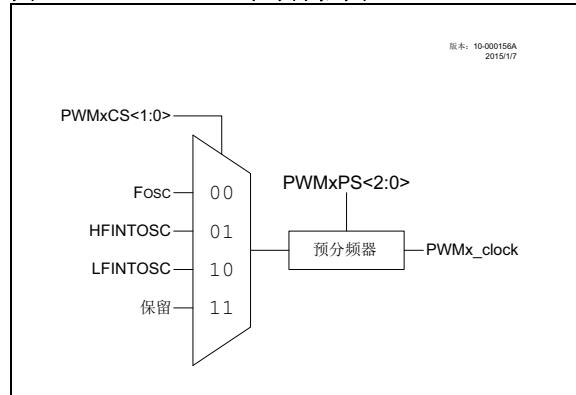
每个 PWM 模块均具有一个独立的定时器，该定时器通过由 PWMxCLKCON 寄存器（[寄存器 23-4](#)）决定的时钟源驱动。模块会将定时器值与事件计数寄存器进行比较，以生成 PWM 波形的各种事件，例如周期和占空比。关于描述时钟源的框图，请参见 [图 23-3](#)。

可以使用 PWMxCON 寄存器的 EN 位来单独使能每个 PWM 模块，也可以使用 PWMEN 寄存器的镜像位来同时使能几个 PWM 模块。

可以使用 PWMxCON 寄存器的 OUT 位来读取 PWM 输出的当前状态。在一些模式下，可以用软件置 1 和清零该位，从而可以对 PWM 波形提供额外的软件控制。该位与 Fosc/4 同步，因此不会相对于 PWM_clock 而实时变化。

注：如果 PWM_clock > Fosc/4，则 OUT 位可能并不准确表示 PWM 的输出状态。

图 23-3：PWM 时钟源框图



23.1.1 PWMx 引脚配置

所有 PWM 输出都与端口数据锁存器进行复用，所以还必须通过清零相关的端口 TRIS 位，将引脚配置为输出。

可以通过配置压摆率特性来优化要与 PWM 输出配合使用的压摆率。高速输出开关通过清零相关的端口 SLRCON 位来实现。

可以通过置 1 相关的端口 ODCON 位，将 PWM 输出配置为漏极开路输出。

23.1.2 PWMx 输出极性

输出极性通过将 PWMxCON 寄存器的 POL 位置 1 来进行反相。极性控制会影响 PWM 输出，即使并未使能该模块。

23.2 PWM 模式

PWM 模式使用 PWM_xCON 寄存器的 MODE<1:0> 位来进行选择（[寄存器 23-1](#)）。

在所有 PWM 模式下，还可以使用偏移匹配事件来在 3 种偏移模式下同步 PWM_xTMR。更多信息，请参见 [第 23.3 节“偏移模式”](#)。

23.2.1 标准模式

标准模式（MODE = 00）会选择单相 PWM 输出。该模式下的 PWM 输出由周期、占空比和相位计数与 PWM_xTMR 值匹配的时刻决定。占空比在相位匹配时开始，在占空比匹配时结束。周期匹配会复位定时器。此外，还可以使用偏移匹配来在偏移模式下同步 PWM_xTMR。更多信息，请参见 [第 23.3 节“偏移模式”](#)。

[公式 23-1](#) 用于计算标准模式下的 PWM 周期。

[公式 23-2](#) 用于计算标准模式下的 PWM 占空比。

公式 23-1：标准模式下的 PWM 周期

$$\text{周期} = \frac{(PWM_{xPR} + I) \cdot \text{预分频比}}{PWM_{xCLK}}$$

公式 23-2：标准模式下的 PWM 占空比

$$\text{占空比} = \frac{(PWM_{xDC} - PWM_{xPH})}{PWM_{xPR} + 1}$$

[图 23-4](#) 给出了标准模式的详细时序图。

23.2.2 匹配时输出置 1 模式

匹配时输出置 1 模式（MODE = 01）会在相位计数与 PWM_xTMR 值匹配时生成有效输出。输出会一直保持有效，直到 PWM_xCON 寄存器的 OUT 位清零或 PWM 模块被禁止为止。在该模式下，占空比计数没有任何作用。周期计数仅决定最大 PWM_xTMR 值，高于该值时不会发生任何相位匹配。

在该模式下，可以使用 PWM_xOUT 位来置 1 或清零 PWM 输出。在写入该位之后，对该位的写操作将在 PWM_clock 的下一个上升沿发生。

[图 23-5](#) 给出了匹配时输出置 1 模式的详细时序图。

23.2.3 匹配时输出翻转模式

匹配时输出翻转模式（MODE = 10）会生成占空比为 50% 的 PWM 输出，且周期为对于标准 PWM 模式计算得到的周期的两倍。在该模式下，占空比计数没有任何作用。相位计数决定输出在周期事件之后的多少个 PWM_xTMR 周期处发生翻转。

在该模式下，对 PWM_xCON 寄存器的 OUT 位的写操作没有任何作用。

[图 23-6](#) 给出了匹配时输出翻转模式的详细时序图。

23.2.4 中心对齐模式

中心对齐模式（MODE = 11）会生成以周期中点为中心的 PWM 波形。在该模式下，周期是 PWM_xPR 计数的两倍。PWM_xTMR 值会递增至周期值，然后递减回 0。占空比计数决定有效 PWM 输出的开始和结束。占空比在 PWM_xTMR 进行递增的情况下发生匹配事件时开始，在 PWM_xTMR 进行递减的情况下发生匹配事件时结束。递增匹配值等于周期计数减去占空比计数。递减匹配值等于递增匹配值加 1。

[公式 23-3](#) 用于计算中心对齐模式下的 PWM 周期。

公式 23-3：中心对齐模式下的 PWM 周期

$$\text{周期} = \frac{(PWM_{xPR} + I) \cdot \text{预分频比} \cdot 2}{PWM_{xCLK}}$$

[公式 23-4](#) 用于计算中心对齐模式下的 PWM 占空比。

公式 23-4：中心对齐模式下的 PWM 占空比

$$\text{占空比} = \frac{PWM_{xDC} \cdot 2}{(PWM_{xPR} + I) \cdot 2}$$

在该模式下，对 PWM_xOUT 的写操作没有任何作用。

[图 23-7](#) 给出了中心对齐模式的详细时序图。

图 23-4：标准 PWM 模式时序图

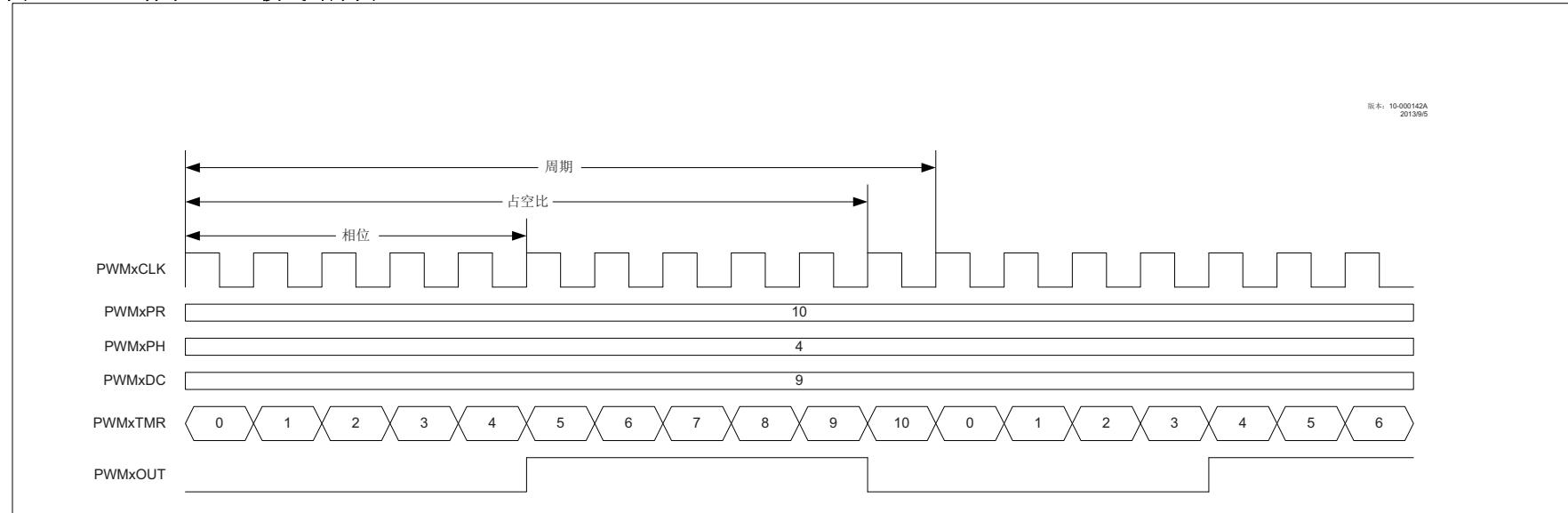


图 23-5：匹配时输出置 1 PWM 模式时序图

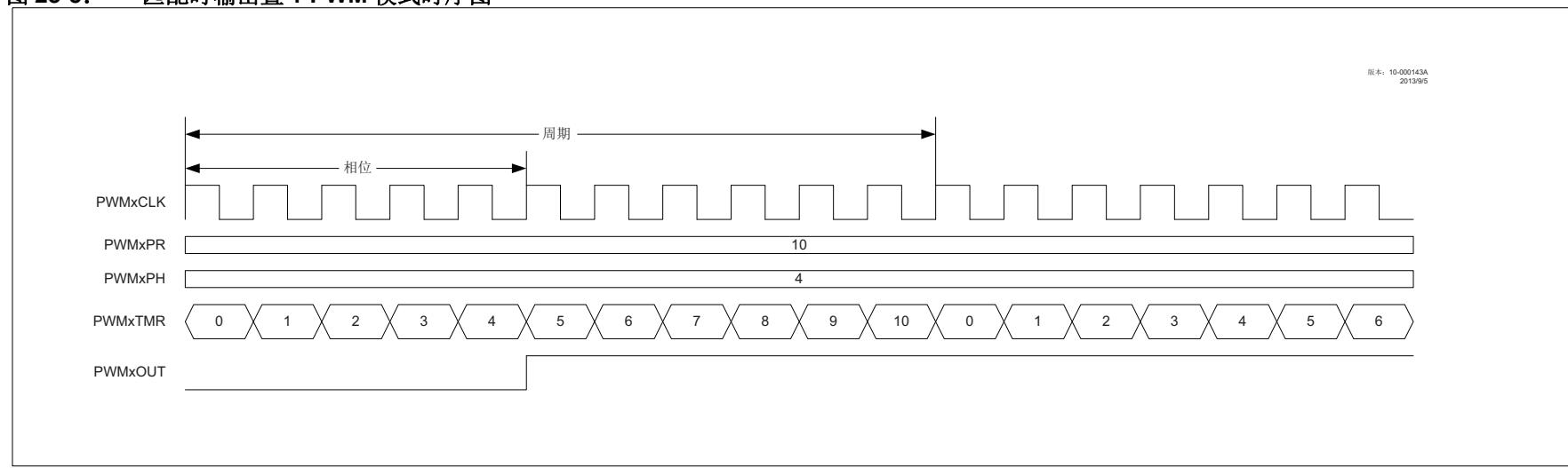


图 23-6：匹配时输出翻转 PWM 模式时序图

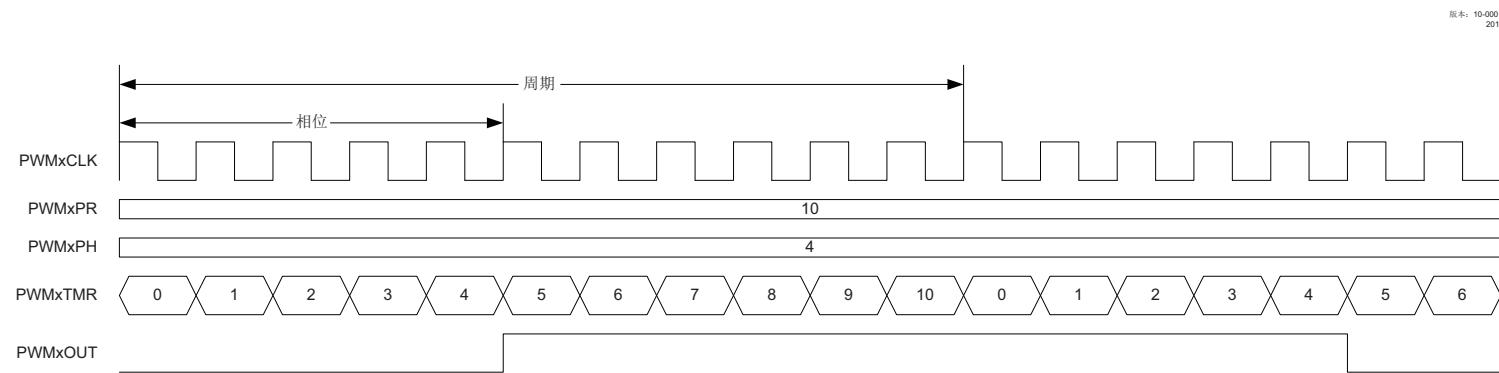
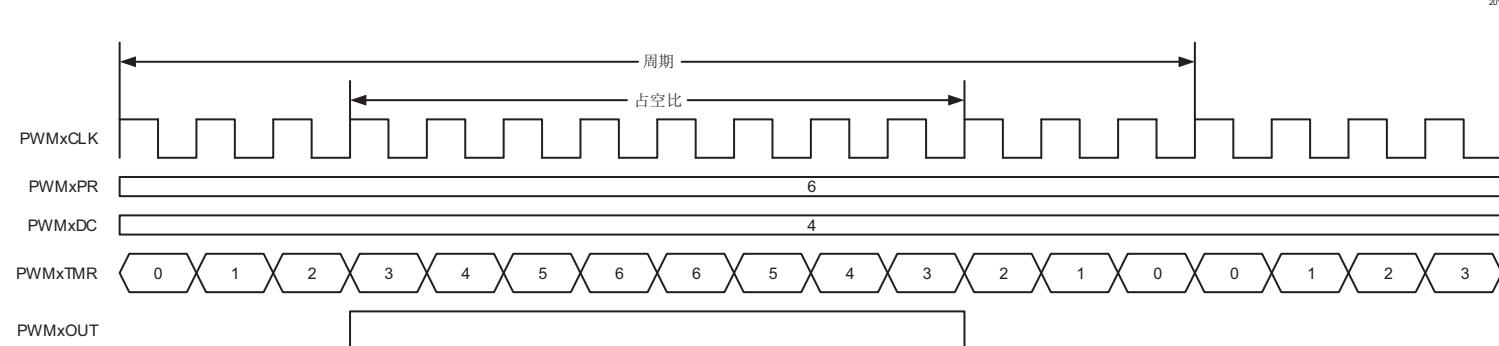


图 23-7：中心对齐 PWM 模式时序图



23.3 偏移模式

偏移模式用于相对于同一器件中的主 PWM 模块的波形来调整从 PWM 模块的波形。

23.3.1 独立运行模式

在独立运行模式 ($OFM = 00$) 下, PWM 模块不会受器件中的其他 PWM 模块影响。在该模式下, 与 PWM 模块相关的 PWMxTMR 在与该 PWM 模块相关的 EN 位置 1 时立即开始计数, 并一直保持计数, 直到 EN 位被清零为止。周期事件会将 PWMxTMR 复位为零, 此后定时器会继续计数。

[图 23-8](#) 给出了该模式与标准 PWM 模式配合使用的详细时序图。

23.3.2 带同步启动的从运行模式

在带同步启动的从运行模式 ($OFM = 01$) 下, 从模块的 PWMxTMR 会等待主模块的 OF_match 事件。发生该事件时, 如果 EN 位置 1, PWMxTMR 会开始计数并一直保持计数, 直到软件清零 EN 位为止。从模块的周期事件会将 PWMxTMR 复位为零, 此后定时器会继续计数。

[图 23-9](#) 给出了该模式与标准 PWM 模式配合使用的详细时序图。

23.3.3 带同步启动的单事件从模式

在带同步启动的单事件从模式 ($OFM = 10$) 下, 从模块的 PWMxTMR 会等待主模块的 OF_match 事件。然后定时器会开始计数, 从定时器中已有的值开始并继续计数, 直到发生周期匹配事件为止。当发生周期事件时, 定时器会复位为 0 并停止计数。然后, 定时器会等待发生下一个主模块 OF_match 事件, 之后它会重新开始计数, 重复此循环。如果在从 PWM 模块完成之前发生 OF_match 事件, 则先前触发的周期会被忽略。如果从模块周期大于主模块周期, 但小于主模块周期的两倍, 则会导致每隔一个主模块周期产生一个从模块输出。

注: 在从模块的定时器被复位为 0 期间, 如果接收到另一个偏移匹配事件, 从 PWM 模块有可能不会识别到该匹配事件, 从模块定时器不会再次开始计数。这会导致从 PWM 模块的输出中丢失占空比。为了防止发生这种情况, 请避免对主 PWM 模块和从 PWM 模块使用相同的周期。

[图 23-10](#) 给出了该模式与标准 PWM 模式配合使用的详细时序图。

23.3.4 带同步启动和定时器复位的连续运行从模式

在带同步启动和定时器复位的连续运行从模式 ($OFM = 11$) 下, 会禁止从模块的 PWMxTMR 在置 1 从 PWM 模块使能位之后开始计数。第一个主模块 OF_match 事件会启动从模块的 PWMxTMR。后续的主模块 OF_match 事件会将从模块的 PWMxTMR 定时器值复位为 1, 之后从模块的 PWMxTMR 会继续计数。下一个主模块 OF_match 事件会将从模块的 PWMxTMR 复位回 1, 重复此循环。在主模块的 OF_match 事件之前发生的从模块周期事件会将从模块的 PWMxTMR 复位为 0, 之后定时器会继续计数。在该模式下工作的从模块的 PWMxPH 寄存器对值必须大于等于 1, 否则就不会发生相位匹配事件, 从而不会启动 PWM 输出占空比。

如果主模块和从模块的 PWMxPR 值相等, 并且在 PWM 模块正在工作时从偏移从模式更改为独立运行模式, 则偏移时序会持续下去。

[图 23-11](#) 给出了在标准 PWM 模式下使用该模式的详细时序图。

注: 如果从模块 PWM_clock 的频率高于主模块 PWM_clock, 则会发生意外的结果。

23.3.5 中心对齐模式下的偏移匹配

当主模块在中心对齐模式下工作时, 偏移匹配事件将取决于 PWMxTMR 的计数方向。将 PWMxOFCON 寄存器的 OFO 位清零会导致在定时器递增时发生 OF_match 事件。将 PWMxOFCON 寄存器的 OFO 位置 1 会导致在定时器递减时发生 OF_match 事件。在非中心对齐模式下, OFO 位会被忽略。

OFO 位是双重缓冲的, 在 PWM 模块正在工作时, 需要将 LDA 位置 1 才会生效。

[图 23-12](#) 和 [图 23-13](#) 给出了在带同步启动的独立从模式下使用偏移匹配控制时, 中心对齐模式的详细时序图。

图 23-8：独立运行模式时序图

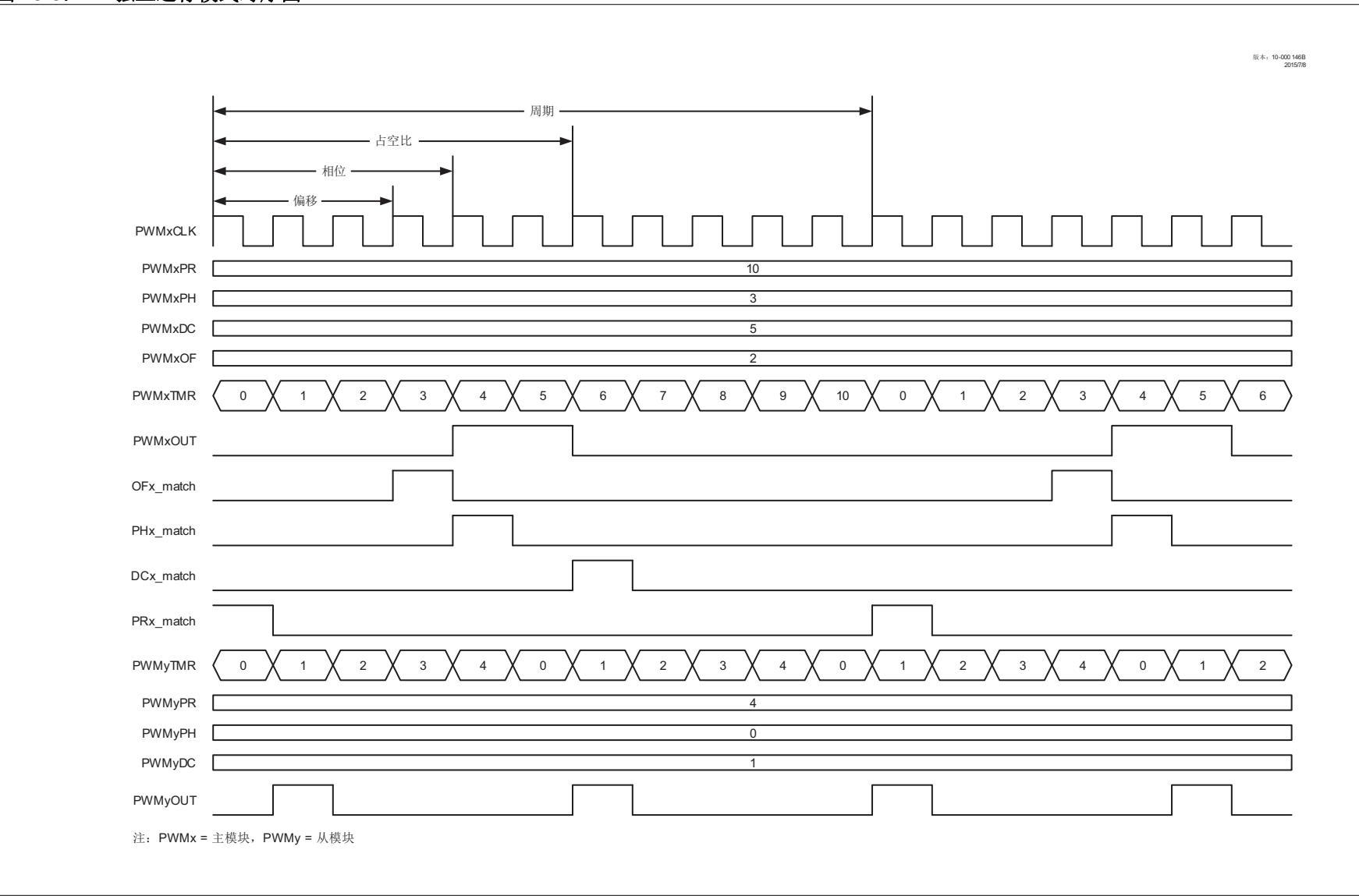


图 23-9：带同步启动的从运行模式的时序图

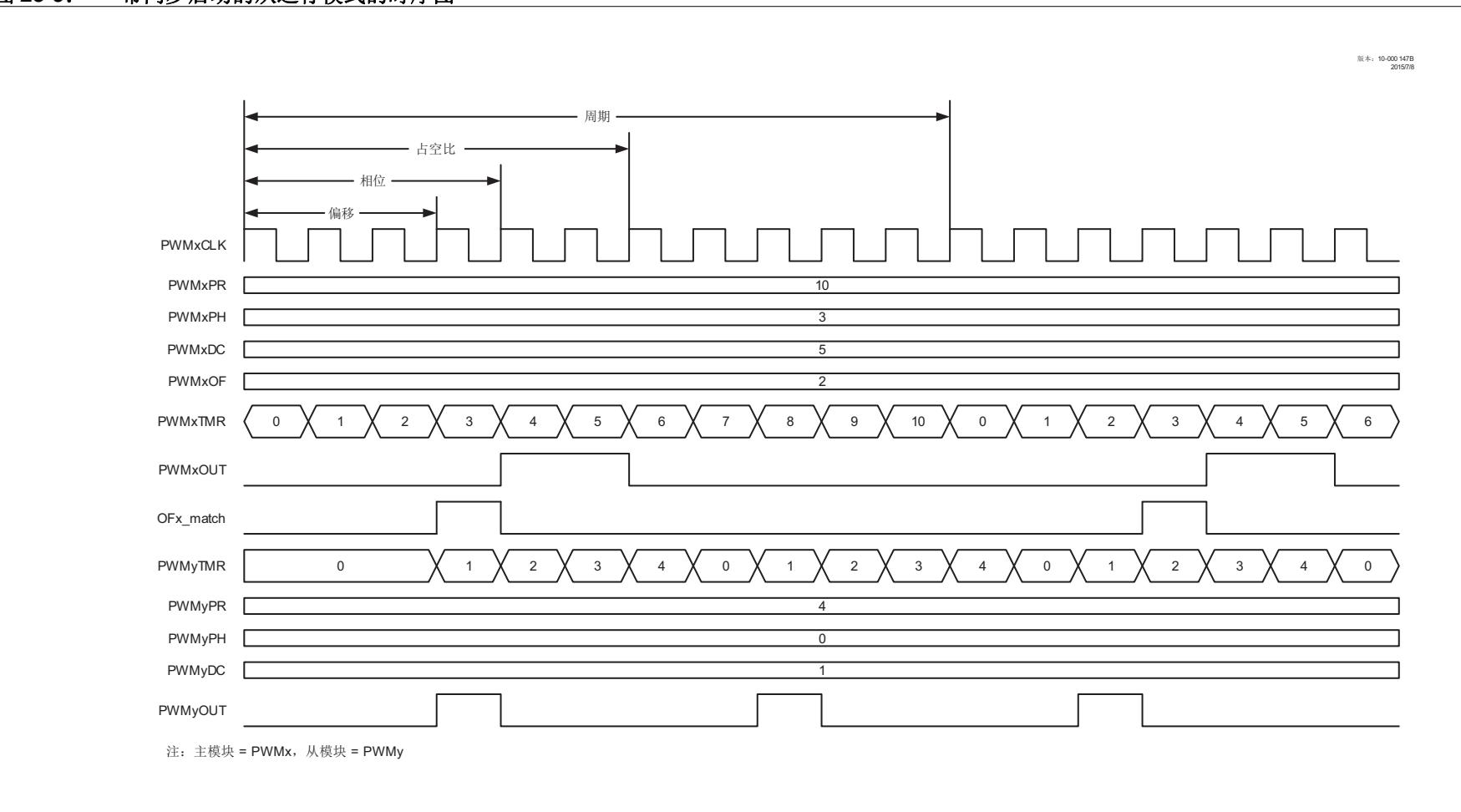


图 23-10：带同步启动的单事件从运行模式的时序图

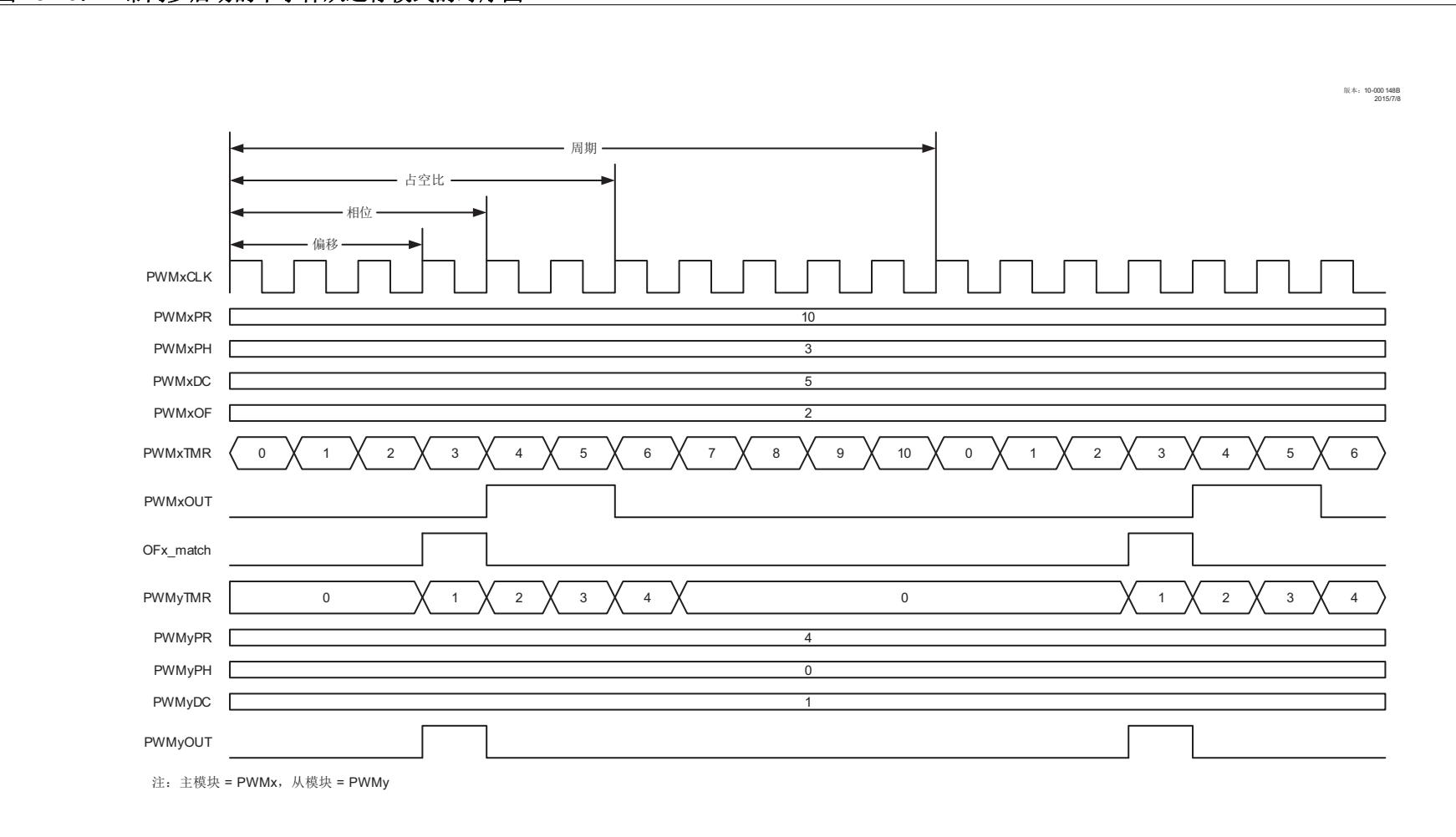


图 23-11：带立即复位和同步启动的连续从运行模式的时序图

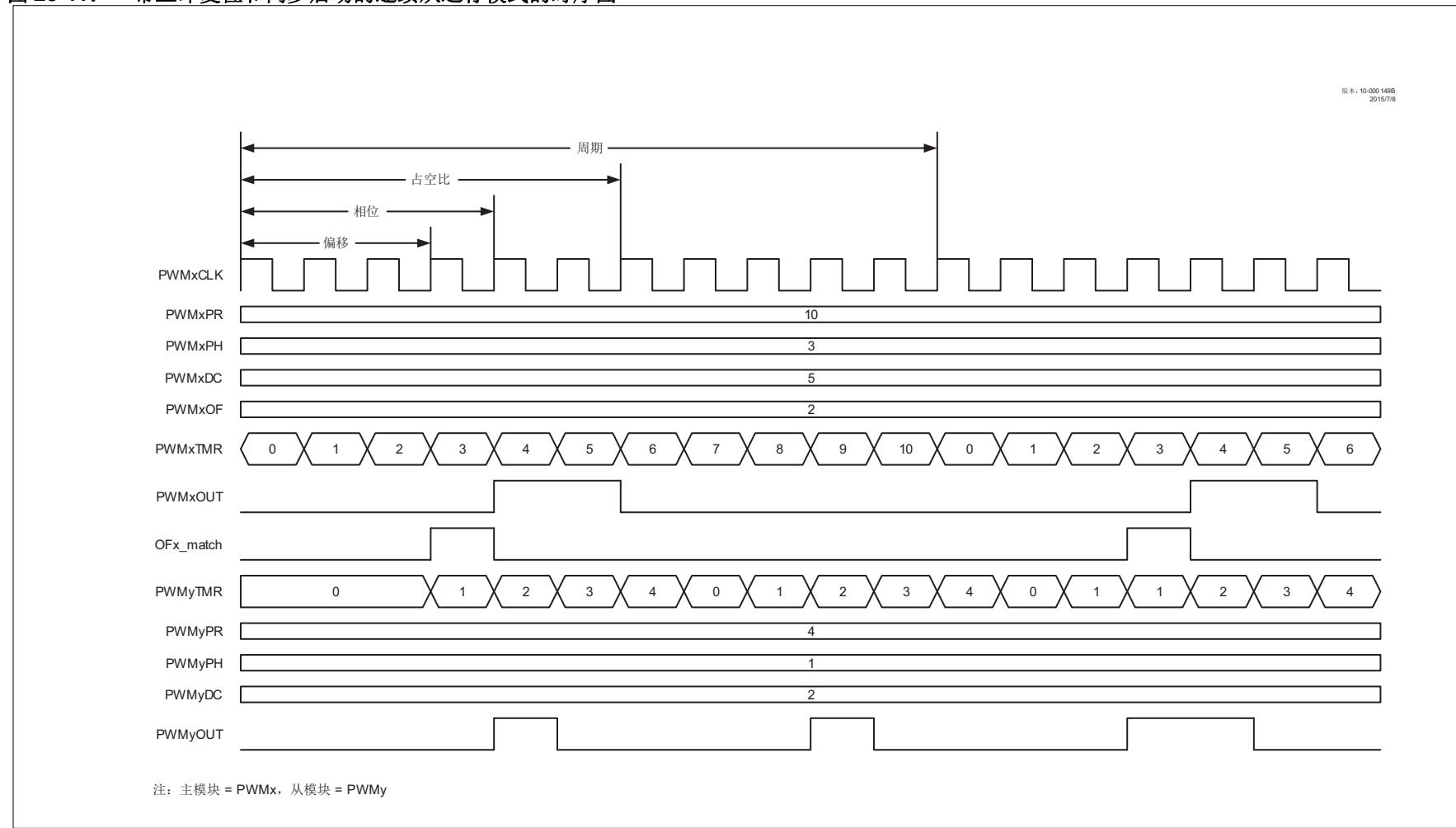


图 23-12：递增定时器时的偏移匹配的时序图

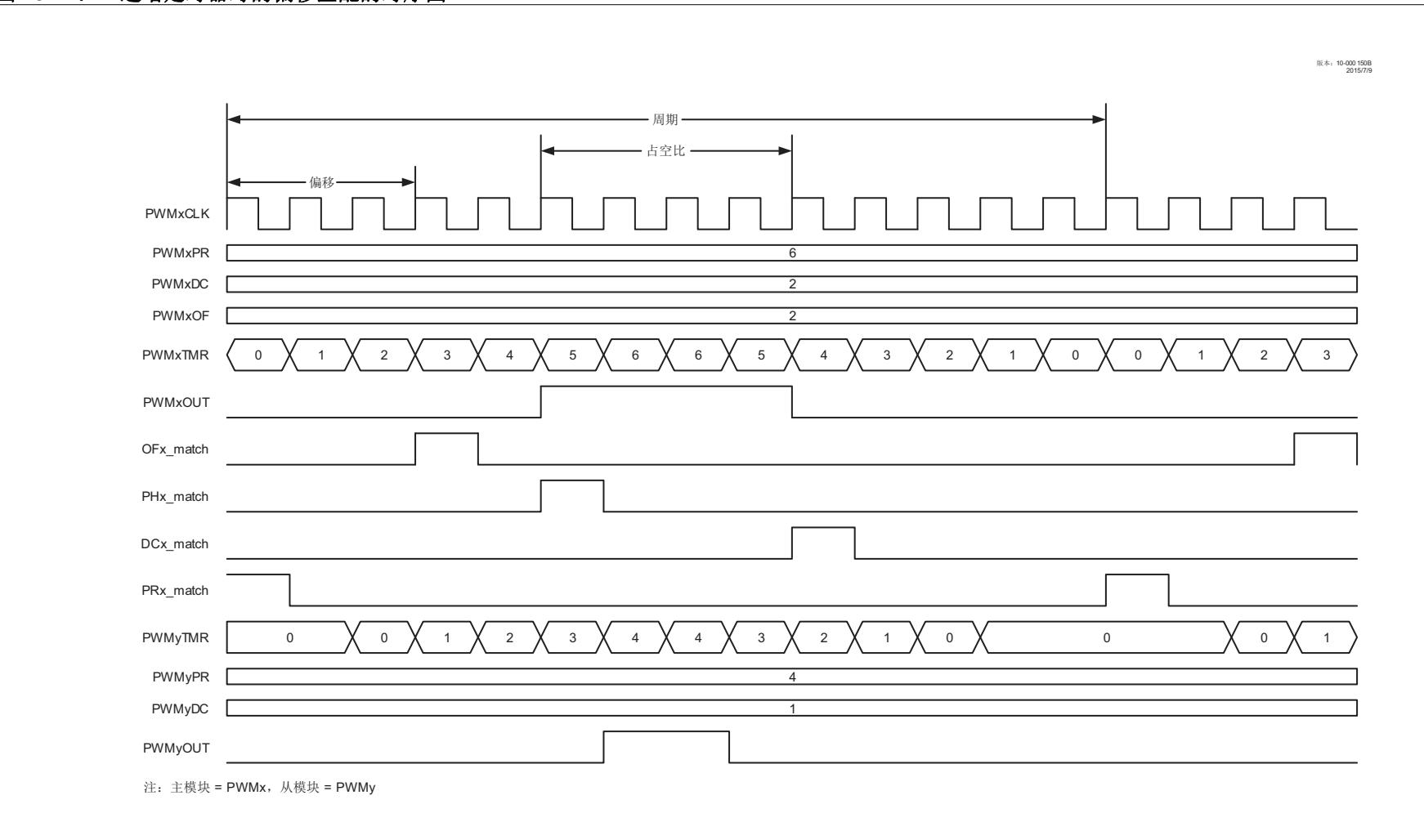
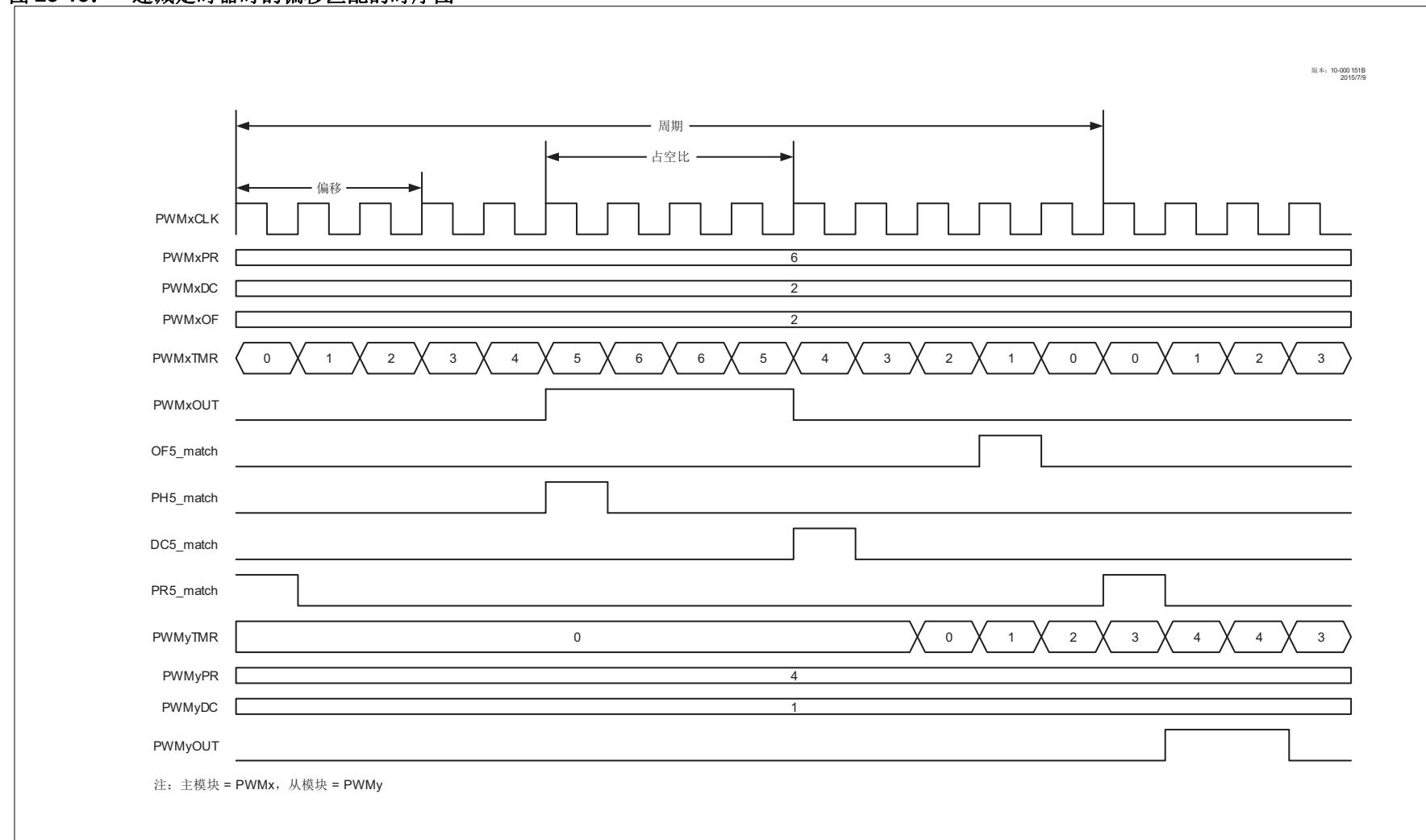


图 23-13：递减定时器时的偏移匹配的时序图



23.4 重载操作

4 个 PWM 模块控制寄存器对和 1 个控制位是双重缓冲的，从而可以同时更新它们。包括：

- PWM_xPHH:PWM_xPHL 寄存器对
- PWM_xDCH:PWM_xDCL 寄存器对
- PWM_xPRH:PWM_xPRL 寄存器对
- PWM_xOFL:PWM_xOFL 寄存器对
- OFO 控制位

在写入时，这些寄存器不会立即影响 PWM 的操作。默认情况下，只有在满足激活条件之后，写入这些寄存器的数据才会被装入 PWM 工作缓冲寄存器。激活控制有两种操作方法：

- 立即
- 触发

PWM_xLDCON 寄存器的 LDT 位控制激活方法。这两种方法都要求 LDA 位被置 1。全部 4 个缓冲寄存器对会在发生装入事件时同时装入。

23.4.1 立即重载

当 LDT 位清零时，会选择立即模式，缓冲寄存器将在 LDA 位置 1 之后的第一个周期事件时装入。立即重载在 PWM 模块独立工作时或 PWM 模块作为其他从 PWM 模块的主模块工作时使用。

23.4.2 触发重载

当 LDT 位置 1 时，会选择触发模式，只有发生触发事件时 LDA 位才会生效。触发源是器件中其他 PWM 模块之一的缓冲寄存器装入事件。通过 PWM_xLDCON 寄存器的 LDS<1:0> 位选择触发源。缓冲寄存器将在触发事件之后发生第一个周期事件时装入。触发重载在 PWM 模块作为另一个 PWM 的从模块工作，并在需要同步这两个模块中的缓冲寄存器重载时使用。

注 1: 缓冲寄存器装入操作会清零 LDA 位。

2: 如果 LDA 位在 PWM_xTMR = PWM_xPR 的同一时刻置 1，则 LDA 位会被忽略，直到发生下一个周期事件为止。当选择触发重载，且触发事件与目标的周期事件同时发生时，就属于这种情况。

23.5 休眠模式下的操作

通过 PWM_xCLKCON<1:0> 选择 HFINTOSC 或 LFINTOSC 作为时钟源时，每个 PWM 模块会在休眠模式下继续工作。

23.6 中断

每个 PWM 模块都具有 4 个独立中断，分别基于相位、占空比、周期和偏移匹配事件。中断标志在其中每个信号的上升沿置 1。关于匹配信号的详细时序图，请参见图 23-8 和 23-12。

23.7 寄存器定义：PWM 控制

表 23-1 列出了 16 位 PWM 外设的长位名称前缀。更多信息，请参见第 1.1 节“寄存器和位的命名约定”。

表 23-1：

外设	位名称前缀
PWM1	PWM1
PWM2	PWM2
PWM3	PWM3
PWM4	PWM4

寄存器 23-1： PWMxCON： PWM 控制寄存器

R/W-0/0	U-0	R/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EN	—	OUT	POL	MODE<1:0>	—	—	bit 0
bit 7							

图注：

HC = 硬件清零位

HS = 硬件置 1 位

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- | | |
|---------|--|
| bit 7 | EN: PWM 模块使能位
1 = 使能模块
0 = 禁止模块 |
| bit 6 | 未实现: 读为 0 |
| bit 5 | OUT: PWM 模块的输出状态 |
| bit 4 | POL: PWM 输出极性控制位
1 = PWM 输出有效状态为低电平
0 = PWM 输出有效状态为高电平 |
| bit 3-2 | MODE<1:0>: PWM 模式控制位
11 = 中心对齐模式
10 = 匹配时输出翻转模式
01 = 匹配时输出置 1 模式
00 = 标准 PWM 模式 |
| bit 1-0 | 未实现: 读为 0 |

PIC16(L)F1574/5/8/9

寄存器 23-2: PWMxINTE: PWM 中断允许寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	OFIE	PHIE	DCIE	PRIE
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 未实现: 读为 0

bit 3 **OFIE:** 偏移中断允许位

1 = 在发生偏移匹配时中断 CPU
0 = 在发生偏移匹配时不中断 CPU

bit 2 **PHIE:** 相位中断允许位

1 = 在发生相位匹配时中断 CPU
0 = 在发生相位匹配时不中断 CPU

bit 1 **DCIE:** 占空比中断允许位

1 = 在发生占空比匹配时中断 CPU
0 = 在发生占空比匹配时不中断 CPU

bit 0 **PRIE:** 周期中断允许位

1 = 在发生周期匹配时中断 CPU
0 = 在发生周期匹配时不中断 CPU

寄存器 23-3: PWMxINTF: PWM 中断请求寄存器

U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	OFIF	PHIF	DCIF	PRIF
bit 7	bit 0						

图注:

HC = 硬件清零位

HS = 硬件置 1 位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 未实现: 读为 0

bit 3 **OFIF:** 偏移中断标志位⁽¹⁾

1 = 发生偏移匹配事件
0 = 未发生偏移匹配事件

bit 2 **PHIF:** 相位中断标志位⁽¹⁾

1 = 发生相位匹配事件
0 = 未发生相位匹配事件

bit 1 **DCIF:** 占空比中断标志位⁽¹⁾

1 = 发生占空比匹配事件
0 = 未发生占空比匹配事件

bit 0 **PRIF:** 周期中断标志位⁽¹⁾

1 = 发生周期匹配事件
0 = 未发生周期匹配事件

注 1: 在模块被禁止时 (EN = 0) 时, 硬件会强制清零该位。

寄存器 23-4: PWMxCLKCON: PWM 时钟控制寄存器

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—		PS<2:0>		—	—	CS<1:0>	
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7 未实现: 读为 0

bit 6-4 **PS<2:0>**: 时钟源预分频比选择位

- 111 = 对时钟源进行 128 分频
- 110 = 对时钟源进行 64 分频
- 101 = 对时钟源进行 32 分频
- 100 = 对时钟源进行 16 分频
- 011 = 对时钟源进行 8 分频
- 010 = 对时钟源进行 4 分频
- 001 = 对时钟源进行 2 分频
- 000 = 不预分频

bit 3-2 未实现: 读为 0

bit 1-0 **CS<1:0>**: 时钟源选择位

- 11 = 保留
- 10 = LFINTOSC (在休眠期间继续工作)
- 01 = HFINTOSC (在休眠期间继续工作)
- 00 = FOSC

PIC16(L)F1574/5/8/9

寄存器 23-5： PWMxLDCON： PWM 重载触发源选择寄存器

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
LDA ⁽¹⁾	LDT	—	—	—	—	LDS<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 **LDA:** 装入缓冲寄存器激活位⁽¹⁾

如果 LDT = 1:

1 = 在出现选定的触发信号时，在周期结束时装入 OFx、PHx、DCx 和 PRx 缓冲寄存器。

0 = 不装入缓冲寄存器 / 装入已完成

如果 LDT = 0:

1 = 在当前周期结束时装入 OF、PH、DC 和 PR 缓冲寄存器

0 = 不装入缓冲寄存器或装入已完成

bit 6 **LDT:** 触发时装入缓冲寄存器位

1 = 使能在触发时装入缓冲寄存器

0 = 禁止在触发时装入

在出现选定的触发信号之后，在每个周期结束时装入 OFx、PHx、DCx 和 PRx 缓冲寄存器。

在当前周期结束时重载内部双重缓冲寄存器。PWMxLDS 位会被忽略。

bit 5-2 未实现：读为 0

bit 1-0 **LDS<1:0>:** 装入触发源选择位

11 =LD4_trigger⁽²⁾

10 =LD3_trigger⁽²⁾

01 =LD2_trigger⁽²⁾

00 =LD1_trigger⁽²⁾

注 1: 该位由模块在重载操作之后清零。它可以用软件清零，以清除现有的激活事件。

2: 对应于所用 PWM 的 LD_trigger 会变为保留。

寄存器 23-6: PWMxOFCON: PWM 偏移触发源选择寄存器

U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	OFM<1:0>	OFO(1)	—	—	—	OFS<1:0>	bit 0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7

未实现: 读为 0

bit 6-5

OFM<1:0>: 偏移模式选择位

11 = 带立即复位和同步启动的连续从运行模式, 在选定的偏移触发信号出现时。

10 = 带同步启动的单事件从运行模式, 在选定的偏移触发信号出现时。

01 = 带同步启动的独立从运行模式, 在选定的偏移触发信号出现时。

00 = 独立运行模式

bit 4

OFO: 偏移匹配输出控制位

如果 MODE<1:0> = 11 (PWM 中心对齐模式):

1 = 在计数器递减情况下, 在发生计数器匹配时产生 OFx_match (第二次匹配)

0 = 在计数器递增情况下, 在发生计数器匹配时产生 OFx_match (第一次匹配)

如果 MODE<1:0> = 00、01 或 10 (所有其他模式):

该位被忽略

bit 3-2

未实现: 读为 0

bit 1-0

OFS<1:0>: 偏移触发源选择位

11 = OF4_match(1)

10 = OF3_match(1)

01 = OF2_match(1)

00 = OF1_match(1)

注 1: 对应于所用 PWM 的 OF_match 会变为保留。

PIC16(L)F1574/5/8/9

寄存器 23-7: PWMxPHH: PWMx 相位计数高字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PH<15:8>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **PH<15:8>: PWM 相位高字节位**
 PWM 相位计数的高 8 位

寄存器 23-8: PWMxPHL: PWMx 相位计数低字节寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **PH<7:0>: PWM 相位低字节位**
 PWM 相位计数的低 8 位

寄存器 23-9: PWMxDCH: PWMx 占空比计数高字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
DC<15:8>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

DC<15:8>: PWM 占空比高字节位
PWM 占空比计数的高 8 位

寄存器 23-10: PWMxDCL: PWMx 占空比计数低字节寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

DC<7:0>: PWM 占空比低字节位
PWM 占空比计数的低 8 位

PIC16(L)F1574/5/8/9

寄存器 23-11: PWMxPRH: PWMx 周期计数高字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PR<15:8>							
bit 7							bit 0

图注:

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

bit 7-0 **PR<15:8>: PWM 周期高字节位**
 PWM 周期计数的高 8 位

寄存器 23-12: PWMxPRL: PWMx 周期计数低字节寄存器

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

图注:

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

bit 7-0 **PR<7:0>: PWM 周期低字节位**
 PWM 周期计数的低 8 位

寄存器 23-13: PWMxOFH: PWMx 偏移计数高字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
OF<15:8>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **OF<15:8>: PWM 偏移高字节位**
 PWM 偏移计数的高 8 位

寄存器 23-14: PWMxOFL: PWMx 偏移计数低字节寄存器

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **OF<7:0>: PWM 偏移低字节位**
 PWM 偏移计数的低 8 位

PIC16(L)F1574/5/8/9

寄存器 23-15: PWMxTMRH: PWMx 定时器高字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
TMR<15:8>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **TMR<15:8>: PWM 定时器高字节位**
 PWM 定时器计数器的高 8 位

寄存器 23-16: PWMxTMRL: PWMx 定时器低字节寄存器

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
TMR<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **TMR<7:0>: PWM 定时器低字节位**
 PWM 定时器计数器的低 8 位

注: 以下 3 个镜像寄存器没有长位名称和短位名称形式

寄存器 23-17: PWMEN: PWMEN 位访问寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	PWM4EN_A	PWM3EN_A	PWM2EN_A	PWM1EN_A
bit 7					bit 0		

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 **PWMxEN:** PWM4/PWM3/PWM2/PWM1 使能位

PWMxCON<7> 中的 EN 位的镜像副本

寄存器 23-18: PWMLD: LD 位访问寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	PWM4LDA_A	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A
bit 7					bit 0		

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 **PWMxLDA:** PWM4/PWM3/PWM2/PWM1 LD 位

PWMxLDCON<7> 中的 LD 位的镜像副本

寄存器 23-19: PWMOUT: PWMOUT 位访问寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	PWM4OUT_A	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A
bit 7					bit 0		

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/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 **PWMxOUT:** PWM4/PWM3/PWM2/PWM1 输出位

PWMxCON<5> 中的 OUT 位的镜像副本

PIC16(L)F1574/5/8/9

表 23-2：与 PWM 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页				
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		69				
PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	89				
PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	92				
PWMEN	—	—	—	—	PWM4EN_A	PWM3EN_A	PWM2EN_A	PWM1EN_A	243				
PWMLD	—	—	—	—	PWM4LDA_A	PWM3LDA_A	PWM2LDA_A	PWM1LDA_A	243				
PWMOUT	—	—	—	—	PWM4OUT_A	PWM3OUT_A	PWM2OUT_A	PWM1OUT_A	243				
PWM1PHL	PH<7:0>								238				
PWM1PHH	PH<15:8>								238				
PWM1DCL	DC<7:0>								239				
PWM1DCH	DC<15:8>								239				
PWM1PRL	PR<7:0>								240				
PWM1PRH	PR<15:8>								240				
PWM1OFL	OF<7:0>								241				
PWM1OFH	OF<15:8>								241				
PWM1TMRL	TMR<7:0>								242				
PWM1TMRH	TMR<15:8>								242				
PWM1CON	EN	—	OUT	POL	MODE<1:0>		—	—	233				
PWM1INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	234				
PWM1INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	234				
PWM1CLKCON	—	PS<2:0>			—	—	CS<1:0>		235				
PWM1LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		236				
PWM1OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		237				
PWM2PHL	PH<7:0>								238				
PWM2PHH	PH<15:8>								238				
PWM2DCL	DC<7:0>								239				
PWM2DCH	DC<15:8>								239				
PWM2PRL	PR<7:0>								240				
PWM2PRH	PR<15:8>								240				
PWM2OFL	OF<7:0>								241				
PWM2OFH	OF<15:8>								241				
PWM2TMRL	TMR<7:0>								242				
PWM2TMRH	TMR<15:8>								242				
PWM2CON	EN	—	OUT	POL	MODE<1:0>		—	—	233				
PWM2INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	234				
PWM2INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	234				
PWM2CLKCON	—	PS<2:0>			—	—	CS<1:0>		235				
PWM2LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		236				
PWM2OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		237				
PWM3PHL	PH<7:0>								238				
PWM3PHH	PH<15:8>								238				
PWM3DCL	DC<7:0>								239				
PWM3DCH	DC<15:8>								239				
PWM3PRL	PR<7:0>								240				
PWM3PRH	PR<15:8>								240				
PWM3OFL	OF<7:0>								241				
PWM3OFH	OF<15:8>								241				
PWM3TMRL	TMR<7:0>								242				
PWM3TMRH	TMR<15:8>								242				
PWM3CON	EN	—	OUT	POL	MODE<1:0>		—	—	233				
PWM3INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	234				
PWM3INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	234				

图注：— = 未实现位，读为 0。PWM 不使用阴影单元。

表 23-2：与 PWM 相关的寄存器汇总（续）

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页				
PWM3CLKCON	—	PS<2:0>				—	—	CS<1:0>		235			
PWM3LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		236				
PWM3OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		237				
PWM4PHL	PH<7:0>								238				
PWM4PHH	PH<15:8>								238				
PWM4DCL	DC<7:0>								239				
PWM4DCH	DC<15:8>								239				
PWM4PRL	PR<7:0>								240				
PWM4PRH	PR<15:8>								240				
PWM4OFL	OF<7:0>								241				
PWM4OFH	OF<15:8>								241				
PWM4TMRL	TMR<7:0>								242				
PWM4TMRH	TMR<15:8>								242				
PWM4CON	EN	—	OUT	POL	MODE<1:0>		—	—	233				
PWM4INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	234				
PWM4INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	234				
PWM4CLKCON	—	PS<2:0>				—	—	CS<1:0>		235			
PWM4LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		236				
PWM4OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		237				

图注：— = 未实现位，读为 0。PWM 不使用阴影单元。

表 23-3：与时钟源相关的配置字汇总

名称	位数	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>		—	56
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		—	FOSC<1:0>		

图注：— = 未实现位，读为 0。时钟源不使用阴影单元。

24.0 互补波形发生器（CWG）模块

互补波形发生器（CWG）可从选择的输入源产生带死区延时的互补波形。

CWG 模块具有以下特性：

- 可选死区时钟源控制
- 可选输入源
- 输出使能控制
- 输出极性控制
- 使用独立的 6 位上升沿和下降沿死区计数器进行死区控制
- 可使用以下功能进行自动关断控制：
 - 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 基于所选的输入源产生两个输出波形。

每路输出由关到开的变化可能会因其他输出由开到关的变化而受到延时，因而立即产生延时，在这段延时期间不对两个输出进行驱动。这称为死区，[第 24.5 节“死区控制”](#)对它进行了介绍。[图 24-2](#)给出了基于单输入信号产生的带死区的典型工作波形。

可能需要防止电路发生故障、反馈事件太晚送达或根本不送达的可能性。在这种情况下，必须在故障条件造成损坏之前终止有效驱动。这称为自动关断，[第 24.9 节“自动关断控制”](#)对它进行了介绍。

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）

使用 CWGxCON0 寄存器（[寄存器 24-1](#)）的 G1CS0 位选择时钟源。

24.3 可选输入源

CWG 基于[表 24-1](#) 中的输入源产生两个输出波形。

表 24-1：可选输入源

源外设	信号名称
CWG 输入引脚	CWGxIN 引脚
比较器 C1	C1OUT_sync
比较器 C2	C2OUT_sync
PWM1	PWM1_output
PWM2	PWM2_output
PWM3	PWM3_output
PWM4	PWM4_output

使用 CWGxCON1 寄存器（[寄存器 24-2](#)）中的 GxIS<2:0>位选择输入源。

24.4 输出控制

使能 CWG 模块之后，立即配置互补驱动，并清零 CWGxA 和 CWGxB 驱动。

24.4.1 极性控制

可以单独选择每个 CWG 输出的极性。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。使用 CWGxCON0 寄存器的 GxPOLA 和 GxPOLB 位选择输出极性。

图 24-1：简化的 CWG 框图

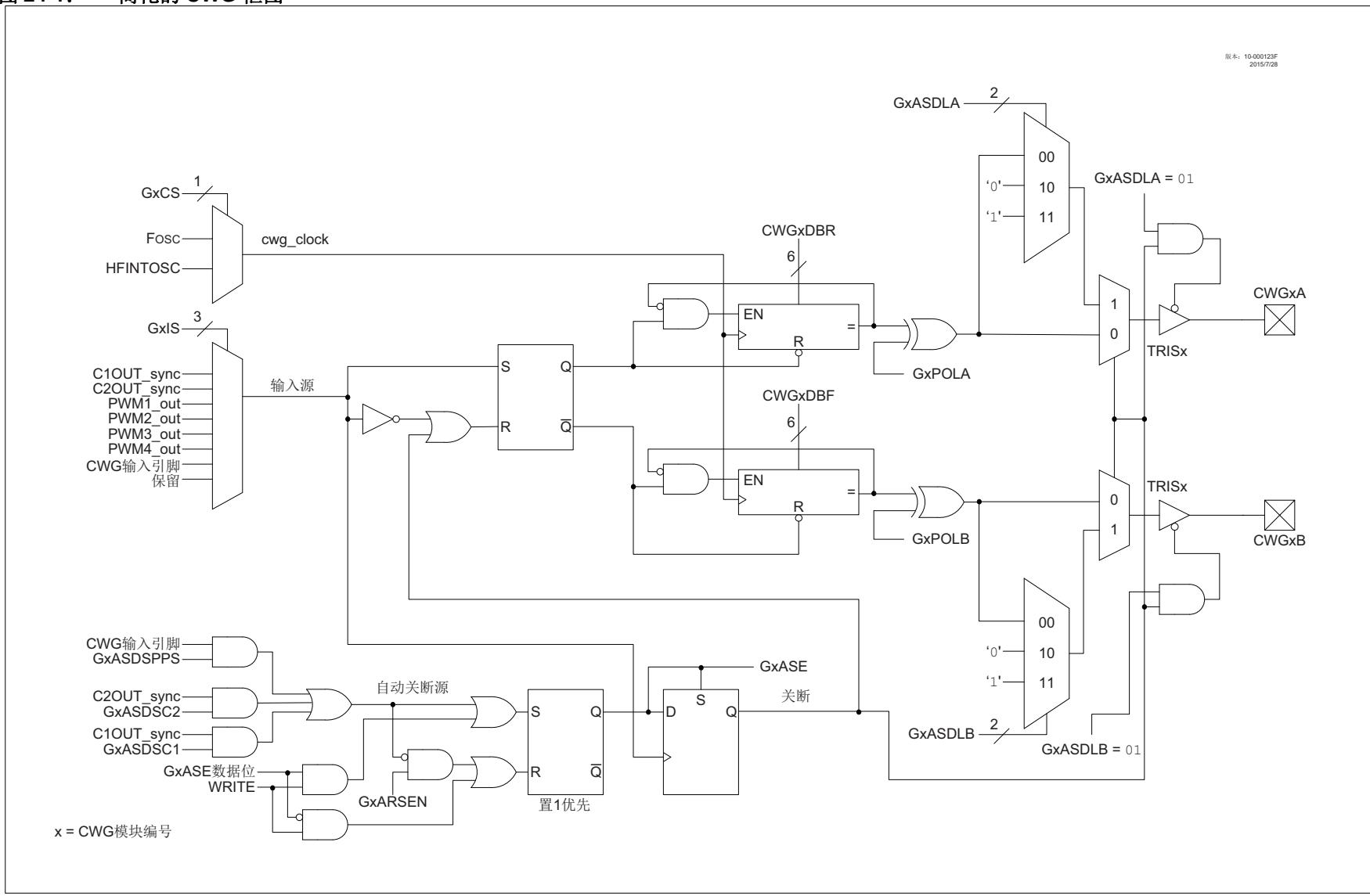
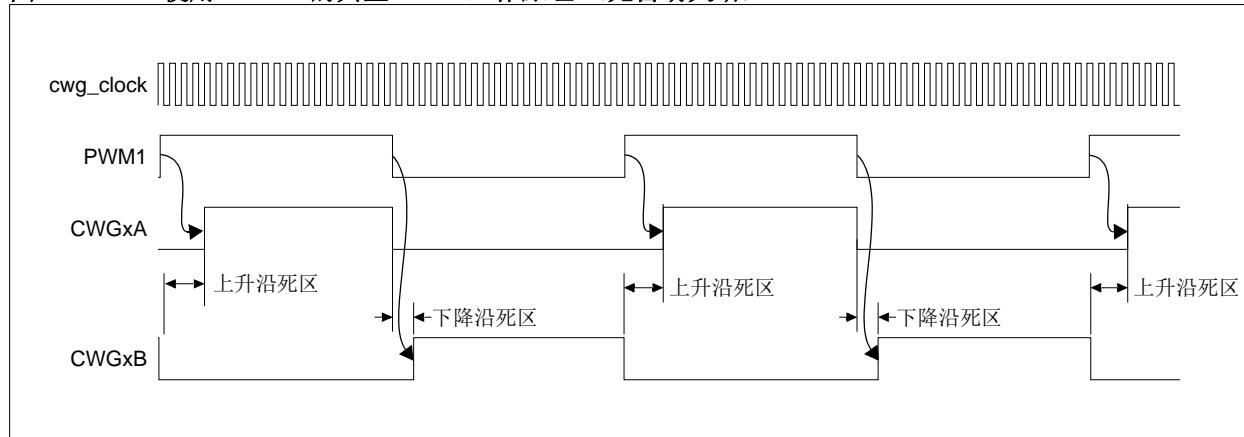


图 24-2： 使用 PWM1 的典型 CWG 工作原理（无自动关断）



24.5 死区控制

死区控制用于提供不重叠的输出信号，以防止功率开关中产生直通电流。CWG 包含两个 6 位死区计数器。一个死区计数器用于输入源控制的上升沿。另一个用于输入源控制的下降沿。

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。请参见 CWGxDBR 和 CWGxDBF 寄存器（分别为 [寄存器 24-4](#) 和 [寄存器 24-5](#)）。

24.6 上升沿死区

上升沿死区在禁止 CWGxB 输出时延迟使能 CWGxA 输出。当输入源信号出现上升沿时，上升沿死区计时开始。发生这种情况时，CWGxB 输出会立即禁止，上升沿死区延时开始计时。当达到上升沿死区延时后，使能 CWGxA 输出。

CWGxDBR 寄存器用于设置输入源信号上升沿死区时间间隔的持续时间。该持续时间为 0 至 64 个死区计数。

死区总是在输入源信号的下降沿停止计数。计数为 0 表示不存在死区。

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 下降沿死区

下降沿死区在禁止 CWGxA 输出时延迟使能 CWGxB 输出。当输入源出现下降沿时，下降沿死区计时开始。发生这种情况时，CWGxA 输出会立即禁止，下降沿死区延时开始计时。当达到下降沿死区延时时，使能 CWGxB 输出。

CWGxDBF 寄存器用于设置输入源信号下降沿死区时间间隔的持续时间。该持续时间为 0 至 64 个死区计数。

死区总是在输入源信号的下降沿停止计数。计数为 0 表示不存在死区。

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 24-3](#) 和 [图 24-4](#)。

图 24-3：死区工作原理（CWGxDBR = 01H, CWGxDBF = 02H）

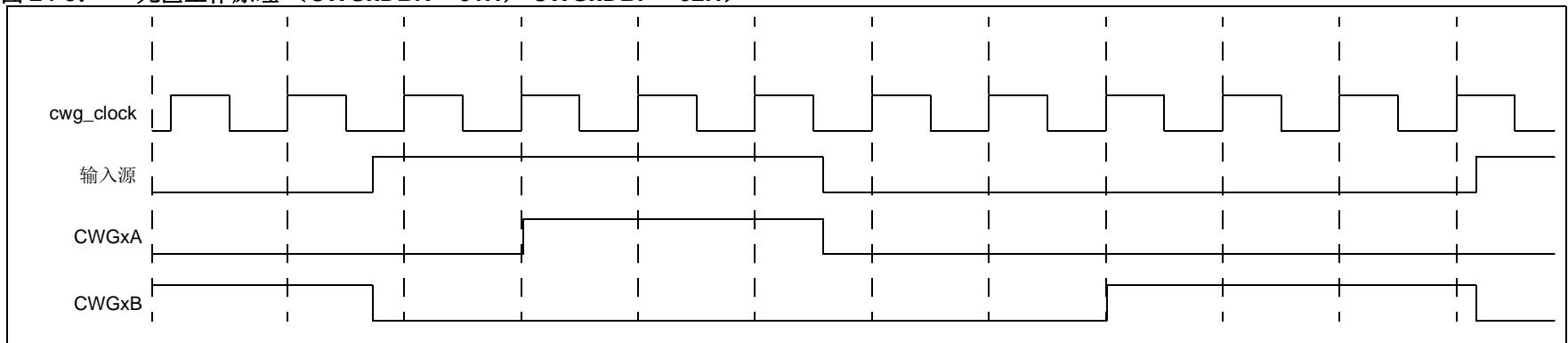
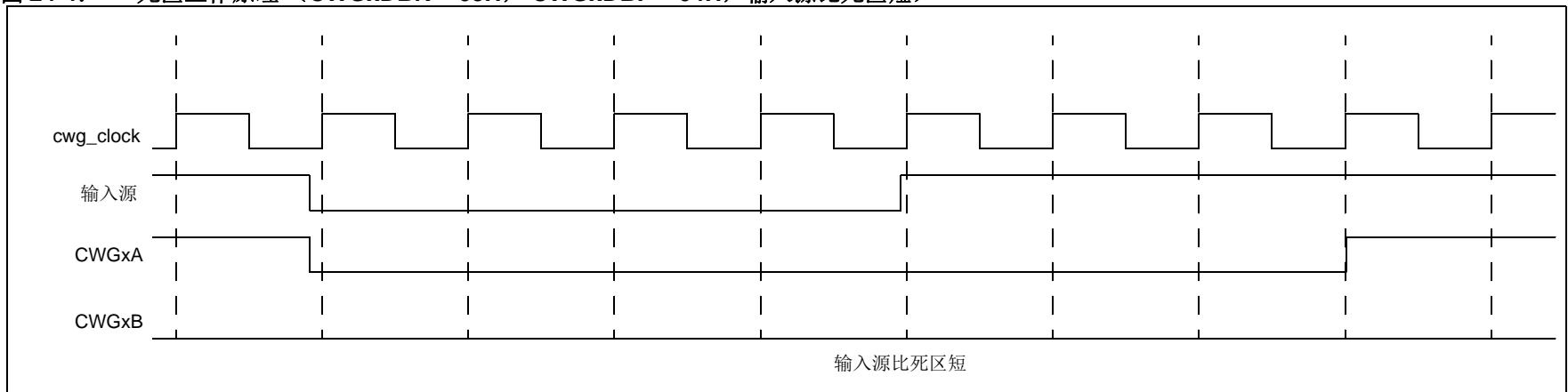


图 24-4：死区工作原理（CWGxDBR = 03H, CWGxDBF = 04H, 输入源比死区短）



24.8 死区偏差

在输入源的上升沿和下降沿触发死区计数器时，输入可能是异步的。这会在死区延时中产生一定的偏差。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

$$TDEADBAND_UNCERTAINTY = \frac{1}{Fcwg_clock}$$

示例：

$$Fcwg_clock = 16 \text{ MHz}$$

因此：

$$\begin{aligned} TDEADBAND_UNCERTAINTY &= \frac{1}{Fcwg_clock} \\ &= \frac{1}{16 \text{ MHz}} \\ &= 62.5 \text{ ns} \end{aligned}$$

24.9 自动关断控制

自动关断是一种使用特定改写信号立即改写 CWG 输出电平，从而安全关断电路的方法。关断状态可以自动清除，也可以一直保持，直到用软件清除。

24.9.1 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

24.9.1.1 由软件产生的关断

将 CWGxCON2 寄存器的 GxAE 位置 1 可以强制 CWG 进入关断状态。

在禁止自动重启时，只要 GxAE 位置 1，就会一直保持关断状态。

在使能自动重启时，GxAE 位会自动清零，并在发生下一个上升沿事件时继续工作。请参见图 24-6。

24.9.1.2 外部输入源

外部关断输入提供了在出现故障条件时安全暂停 CWG 工作的最快方式。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。可以选择两个输入源的任意组合来产生关断条件。这些输入源是：

- 比较器 C1——C1OUT_sync
- 比较器 C2——C2OUT_sync
- $\overline{\text{CWG1FLT}}$

关断输入在 CWGxCON2 寄存器中进行选择。（[寄存器 24-3](#)。）

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。

24.10 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。

只要使能了 CWG 模块，输入源保持活动状态，并且选择 HFINTOSC 作为时钟源，则无论所选择的系统时钟源如何，HFINTOSC 都会在休眠期间保持活动状态。

即，如果在 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 CWG 会继续工作，并且 HFINTOSC 将保持活动状态。

这会直接影响休眠模式的电流。

24.11 配置 CWG

以下步骤说明了如何正确配置 CWG 来确保同步启动：

1. 确保对应于 CWGxA 和 CWGxB 的 TRIS 控制位置 1，从而将它们都配置为输入。
2. 清零 GxEN 位（如果尚未清零）。
3. 使用 CWGxDBR 和 CWGxDBF 寄存器设置所需的死区时间。
4. 在 CWGxCON2 自动关断寄存器中设置以下控制：
 - 选择所需的关断源。
 - 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）。
 - 将 GxAE 位置 1，将 GxARSEN 位清零。
5. 使用 CWGxCON1 寄存器选择所需的输入源。
6. 在 CWGxCON0 寄存器中配置以下控制：
 - 选择所需的时钟源。
 - 选择所需的输出极性。
7. 将 GxEN 位置 1。
8. 将对应于要使用的 CWGxA 和 CWGxB 的 TRIS 控制位清零，从而将这些引脚配置为输出。
9. 如果要使用自动重启，则将 GxARSEN 位置 1，GxAE 位将会自动清零。否则，通过清零 GxAE 位来启动 CWG。

24.11.1 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通过 CWGxCON1 寄存器（[寄存器 24-3](#)）的 GxASDLA 和 GxASDLB 位进行控制。GxASDLA 控制 CWG1A 改写电平，GxASDLB 控制 CWG1B 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。极性控制不应用于改写电平。

24.11.2 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 GxARSEN 位进行选择。[图 24-5](#) 和 [图 24-6](#) 给出了软件控制重启和自动重启的波形。

24.11.2.1 软件控制重启

当 CWGxCON2 寄存器的 GxARSEN 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 GxAE 位将保持置 1。改写电平将一直保持有效，直到 GxAE 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。

24.11.2.2 自动重启

当 CWGxCON2 寄存器的 GxARSEN 位置 1 时，CWG 将从自动关断状态自动重启。

当所有关断源变为低电平时，GxAE 位将自动清零。改写电平将一直保持有效，直到 GxAE 位清零之后发生第一个上升沿事件为止。然后，CWG 将继续工作。

图 24-5：禁止自动重启时的关断功能 ($GxARSEN = 0$, $GxASDLA = 01$, $GxASDLB = 01$)

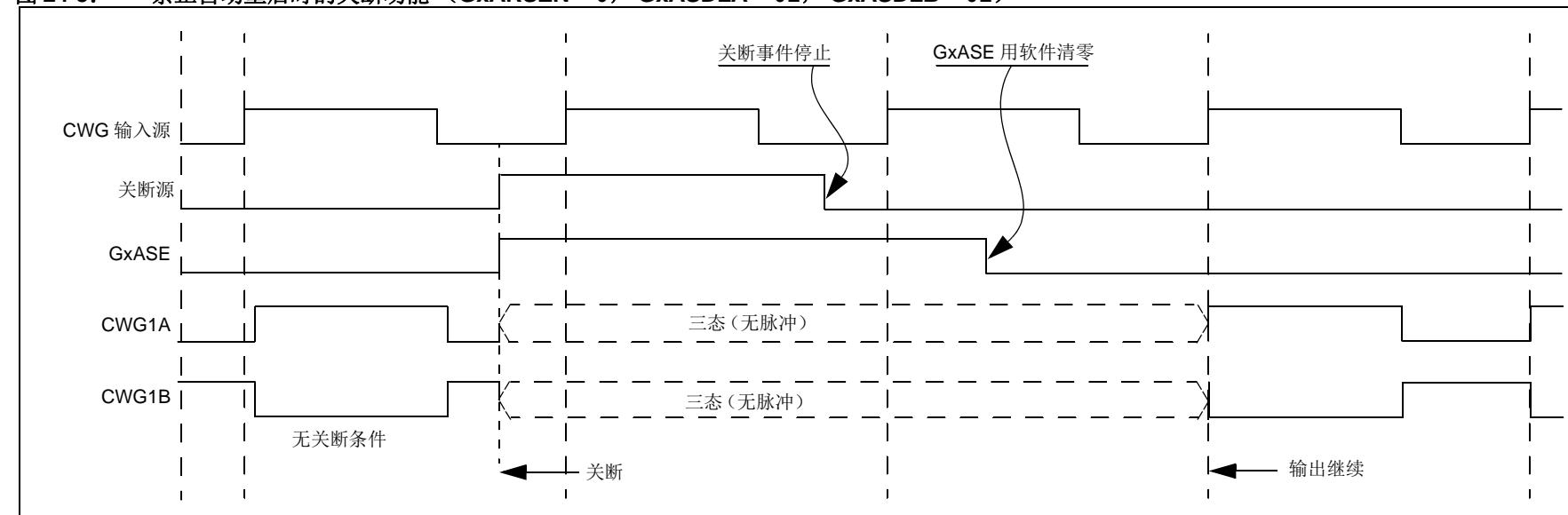
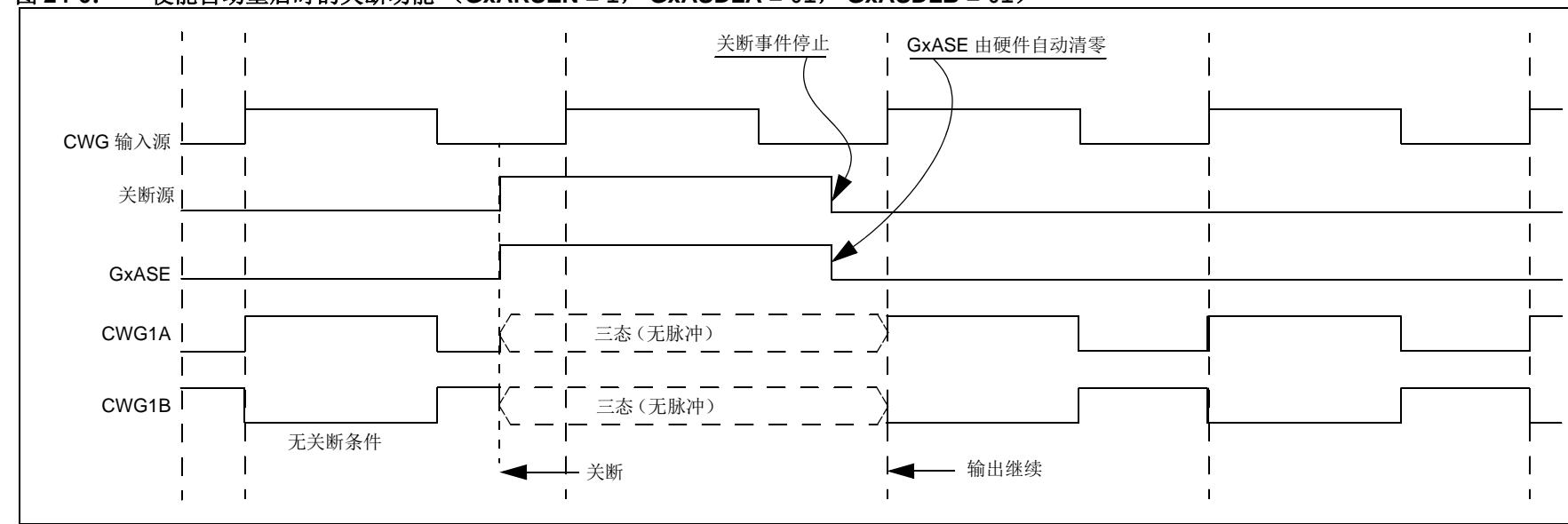


图 24-6：使能自动重启时的关断功能 ($GxARSEN = 1$, $GxASDLA = 01$, $GxASDLB = 01$)



24.12 寄存器定义：CWG 控制

寄存器 24-1：CWGxCON0：CWG 控制寄存器 0

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
GxEN	—	—	GxPOLB	GxPOLA	—	—	GxCs0
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-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 **GxEN:** CWGx 使能位

1 = 使能模块

0 = 禁止模块

bit 6-5 未实现：读为 0

bit 4 **GxPOLB:** CWGxB 输出极性位

1 = 输出极性翻转

0 = 输出极性不翻转

bit 3 **GxPOLA:** CWGxA 输出极性位

1 = 输出极性翻转

0 = 输出极性不翻转

bit 2-1 未实现：读为 0

bit 0 **GxCs0:** CWGx 时钟源选择位

1 = HFINTOSC

0 = Fosc

PIC16(L)F1574/5/8/9

寄存器 24-2： CWGxCON1： CWG 控制寄存器 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-0/0	R/W-0/0	R/W-0/0
GxASDLB<1:0>		GxASDLA<1:0>		—		GxIS<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 7-6 **GxASDLB<1:0>:** CWGxB 的 CWGx 关断状态
当发生自动关断事件（GxASe = 1）时：
11 = CWGxB 引脚被驱动为 1，与 GxPOLB 位的设置无关。
10 = CWGxB 引脚被驱动为 0，与 GxPOLB 位的设置无关。
01 = CWGxB 引脚为三态
00 = 在选定的死区时间间隔之后，CWGxB 引脚被驱动为其无效状态。GxPOLB 仍将控制输出的极性。
- bit 5-4 **GxASDLA<1:0>:** CWGxA 的 CWGx 关断状态
当发生自动关断事件（GxASe = 1）时：
11 = CWGxA 引脚被驱动为 1，与 GxPOLA 位的设置无关。
10 = CWGxA 引脚被驱动为 0，与 GxPOLA 位的设置无关。
01 = CWGxA 引脚为三态
00 = 在选定的死区时间间隔之后，CWGxA 引脚被驱动为其无效状态。GxPOLA 仍将控制输出的极性。
- bit 3 未实现：读为 0
- bit 2-0 **GxIS<2:0>:** CWGx 输入源选择位
111 = 保留
110 = CWG 输入引脚
101 = PWM4——PWM4_out
100 = PWM3——PWM3_out
011 = PWM2——PWM2_out
010 = PWM1——PWM1_out
001 = 比较器 C2——C2OUT_sync
000 = 比较器 C1——C1OUT_sync

寄存器 24-3: CWGxCON2: CWG 控制寄存器 2

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0
GxAE	GxARSEN	—	—	GxASDSC2	GxASDSC1	GxASDSPPS	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-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 | GxAE: 自动关断事件状态位
1 = 发生了自动关断事件
0 = 未发生自动关断事件 |
| bit 6 | GxARSEN: 自动重启使能位
1 = 使能自动重启
0 = 禁止自动重启 |
| bit 5-4 | 未实现: 读为 0 |
| bit 3 | GxASDSC2: 基于比较器 C2 的 CWG 自动关断使能位
1 = 在比较器 C2 的输出 (C2OUT_sync) 为高电平时关断
0 = 比较器 C2 的输出对关断没有任何作用 |
| bit 2 | GxASDSC1: 基于比较器 C1 的 CWG 自动关断使能位
1 = 在比较器 C1 的输出 (C1OUT_sync) 为高电平时关断
0 = 比较器 C1 的输出对关断没有任何作用 |
| bit 1 | GxASDSPPS: CWG 输入引脚使能位
1 = 在 CWG 输入引脚 (CWGXIN) 为高电平时关断
0 = CWG 输入引脚 (CWGXIN) 信号对关断没有任何作用 |
| bit 0 | 未实现: 读为 0 |

PIC16(L)F1574/5/8/9

寄存器 24-4: CWGxDBR: 互补波形发生器 (CWGx) 上升沿死区计数寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	CWGxDBR<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 CWGxDBR<5:0>: 互补波形发生器 (CWGx) 上升沿计数

11 1111 = 死区为 63-64 个计数

11 1110 = 死区为 62-63 个计数

•

•

•

00 0010 = 死区为 2-3 个计数

00 0001 = 死区为 1-2 个计数

00 0000 = 死区为 0 个计数

寄存器 24-5: CWGxDBF: 互补波形发生器 (CWGx) 下降沿死区计数寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	CWGxDBF<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 CWGxDBF<5:0>: 互补波形发生器 (CWGx) 下降沿计数

11 1111 = 死区为 63-64 个计数

11 1110 = 死区为 62-63 个计数

•

•

•

00 0010 = 死区为 2-3 个计数

00 0001 = 死区为 1-2 个计数

00 0000 = 死区为 0 个计数。死区生成被旁路。

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
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	121
CWG1CON0	G1EN	—	—	G1POLB	G1POLA	—	—	G1CS0	253
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	G1IS<2:0>			254
CWG1CON2	G1ASE	G1ARSEN	—	—	G1ASDSC2	G1ASDSC1	G1ASDSPPS	—	255
CWG1DBF	—	—	CWG1DBF<5:0>						256
CWG1DBR	—	—	CWG1DBR<5:0>						256
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	120

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: 未实现, 读为 1。

25.0 在线串行编程 (ICSP™)

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

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- VSS

在编程 / 校验模式下，通过串行通信对程序存储器、用户 ID 和配置字进行编程。ICSPDAT 引脚是用于传输串行数据的双向 I/O，ICSPCLK 引脚是时钟输入引脚。关于 ICSP 的更多信息，请参见 “*PIC16(L)F157x Memory Programming Specification*” (DS40001766)。

25.1 高电压编程模式

通过将 ICSPCLK 和 ICSPDAT 引脚保持为低电平，然后将 MCLR/VPP 上的电压升至 VIHH，将器件置于高电压编程模式。

25.2 低电压编程模式

通过低电压编程模式，只需使用 VDD 就可以对 PIC® 闪存 MCU 进行编程，而无需使用高电压。当配置字的 LVP 位设置为 1 时，将会使能 ICSP 低电压编程模式。要禁止低电压 ICSP 模式，LVP 位必须设定为 0。

进入低电压编程模式需要执行以下步骤：

1. MCLR 电压设置为 VIL。
2. 在提供 ICSPCLK 时钟的同时，在 ICSPDAT 上送出 32 位密钥序列。

完成密钥序列后，在需要维持编程/校验模式的时间内，必须将 MCLR 保持为 VIL。

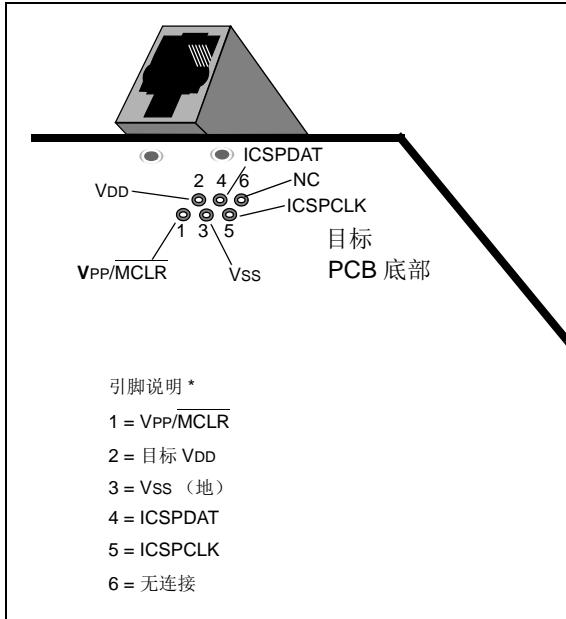
如果使能了低电压编程 (LVP = 1)，则 MCLR 复位功能会被自动使能，无法禁止。更多信息，请参见第 6.5 节“**MCLR**”。

LVP 位只能通过使用高电压编程模式重新设定为 0。

25.3 常用编程接口

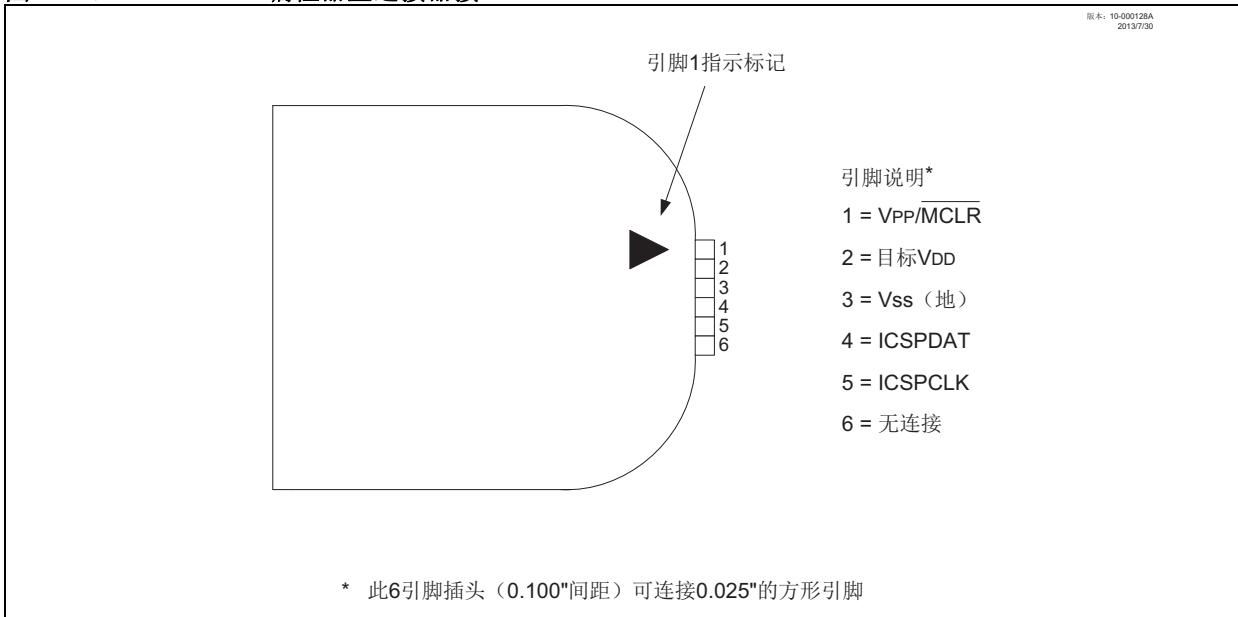
与目标器件的连接通常通过 ICSP 连接器来实现。开发工具中常见的连接器是采用 6P6C (6 引脚，6 连接器) 配置的 RJ-11。请参见图 25-1。

图 25-1：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 25-2。

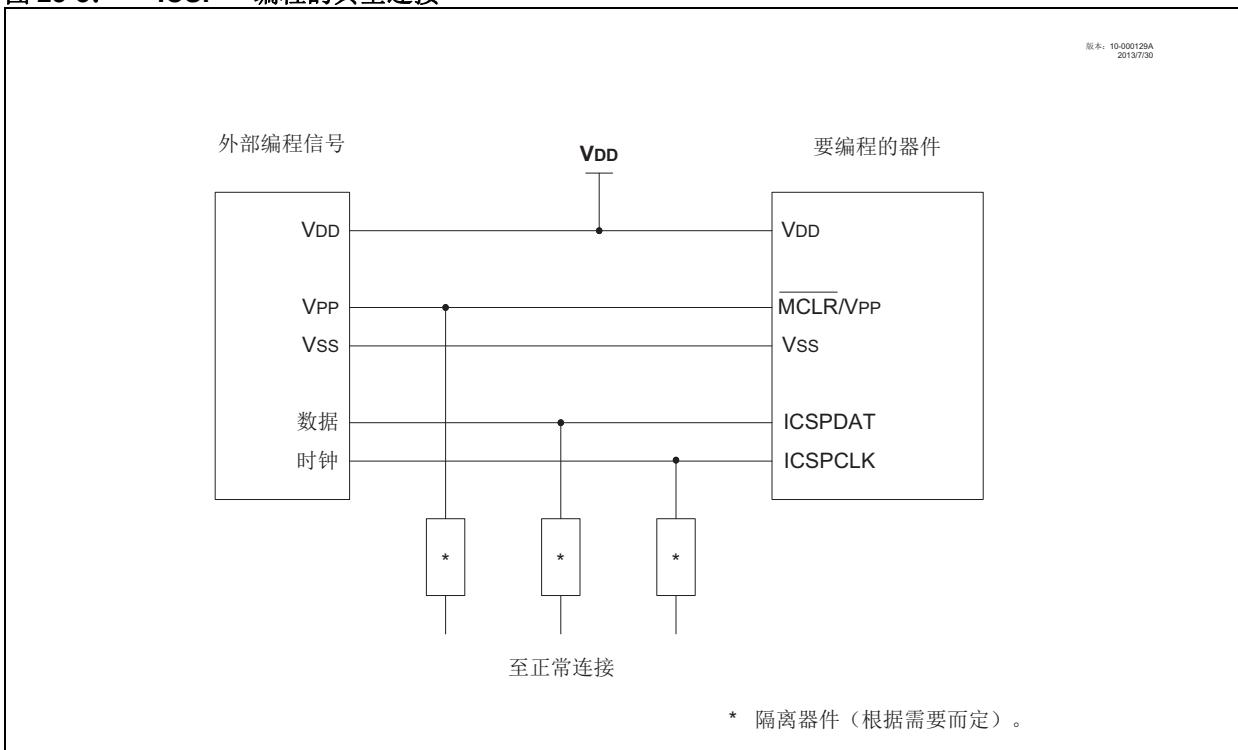
图 25-2: PICkit™ 编程器型连接器接口



关于其他接口建议，请在进行 PCB 设计之前参考具体的器件编程手册。

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 25-3。

图 25-3: ICSP™ 编程的典型连接



26.0 指令集汇总

每条指令都是一个包含操作码和所有必需操作数的14位字。操作码可以分为三大类。

- 针对字节的操作类指令
- 针对位的操作类指令
- 立即数和控制操作类指令

立即数和控制类指令字格式最为丰富。

表 26-3 列出了 MPASM™ 汇编器可识别的指令。

除了以下指令（可能需要 2 或 3 个周期），所有指令都在单个指令周期内执行：

- 子程序指令需要两个周期（CALL 和 CALLW）
- 中断或子程序返回指令需要两个周期（RETURN、RETLW 和 RETFIE）
- 程序跳转指令需要两个周期（GOTO、BRA、BRW、BTFSZ、BTFSZ、DECFSZ 和 INCFSZ）
- 当任意指令引用某个间接文件寄存器，并且文件选择寄存器指向程序存储器时，将需要使用一个额外的指令周期。

一个指令周期包含 4 个振荡器周期；振荡器频率为 4 MHz 时，得到的标称指令执行速率为 1 MHz。

所有指令示例均使用格式 0xhh 来表示一个十六进制数，其中 h 表示一个十六进制位。

26.1 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执行读 - 修改 - 写（Read-Modify-Write, R-M-W）操作。读寄存器、修改数据并根据指令或目标标识符 d 存储结果。即使是写寄存器的指令也将先对该寄存器进行读操作。

表 26-1： 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	立即数段、常数或标号
x	无关位（= 0 或 1）。汇编器将产生 x = 0 的代码。建议使用这种形式，以便与所有 Microchip 软件工具兼容。
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 26-2： 缩写说明

字段	说明
PC	程序计数器
TO	超时位
C	进位位
DC	半进位位
Z	全零标志位
PD	掉电位

图 26-1： 指令的通用格式

针对字节的文件寄存器操作类指令				
13 8 7 6 0				
<table border="1"> <tr> <td>操作码</td> <td>d</td> <td>f(寄存器地址)</td> </tr> </table>		操作码	d	f(寄存器地址)
操作码	d	f(寄存器地址)		
d = 0, 结果存入 W d = 1, 结果存入 f f = 7 位文件寄存器地址				
针对位的文件寄存器操作类指令				
13 10 9 7 6 0				
<table border="1"> <tr> <td>操作码</td> <td>b(位编号)</td> <td>f(寄存器地址)</td> </tr> </table>		操作码	b(位编号)	f(寄存器地址)
操作码	b(位编号)	f(寄存器地址)		
b = 3 位位地址 f = 7 位文件寄存器地址				
立即数和控制操作类指令				
常规				
13 8 7 0				
<table border="1"> <tr> <td>操作码</td> <td>k(立即数)</td> </tr> </table>		操作码	k(立即数)	
操作码	k(立即数)			
k = 8 位立即数值				
仅限 CALL 和 GOTO 指令				
13 11 10 0				
<table border="1"> <tr> <td>操作码</td> <td>k(立即数)</td> </tr> </table>		操作码	k(立即数)	
操作码	k(立即数)			
k = 11 位立即数值				
仅限 MOVLW 指令				
13 7 6 0				
<table border="1"> <tr> <td>操作码</td> <td>k(立即数)</td> </tr> </table>		操作码	k(立即数)	
操作码	k(立即数)			
k = 7 位立即数值				
仅限 MOVLB 指令				
13 5 4 0				
<table border="1"> <tr> <td>操作码</td> <td>k(立即数)</td> </tr> </table>		操作码	k(立即数)	
操作码	k(立即数)			
k = 5 位立即数值				
仅限 BRA 指令				
13 9 8 0				
<table border="1"> <tr> <td>操作码</td> <td>k(立即数)</td> </tr> </table>		操作码	k(立即数)	
操作码	k(立即数)			
k = 9 位立即数值				
FSR 偏移指令				
13 7 6 5 0				
<table border="1"> <tr> <td>操作码</td> <td>n</td> <td>k(立即数)</td> </tr> </table>		操作码	n	k(立即数)
操作码	n	k(立即数)		
n = 相应的 FSR k = 6 位立即数值				
FSR 递增指令				
13 3 2 1 0				
<table border="1"> <tr> <td>操作码</td> <td>n</td> <td>m(模式)</td> </tr> </table>		操作码	n	m(模式)
操作码	n	m(模式)		
n = 相应的 FSR m = 2 位模式值				
仅限操作码				
13 0				
<table border="1"> <tr> <td>操作码</td> </tr> </table>		操作码		
操作码				

表 26-3：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				
针对字节的文件寄存器操作类指令								
ADDWF f, d	W 与 f 相加	1	00	0111	dfff	ffff	C、DC 和 Z	2
ADDWFC f, d	W 与 f 相加 (带进位)	1	11	1101	dfff	ffff	C、DC 和 Z	2
ANDWF f, d	W 和 f 作逻辑与运算	1	00	0101	dfff	ffff	Z	2
ASRF f, d	算术右移	1	11	0111	dfff	ffff	C 和 Z	2
LSLF f, d	逻辑左移	1	11	0101	dfff	ffff	C 和 Z	2
LSRF f, d	逻辑右移	1	11	0110	dfff	ffff	C 和 Z	2
CLRF f	将 f 清零	1	00	0001	1fff	ffff	Z	2
CLRW -	将 W 清零	1	00	0001	0000	00xx	Z	
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 的内容送入 f	1	00	0000	1fff	ffff		2
RLF f, d	f 带进位循环左移	1	00	1101	dfff	ffff	C	2
RRF f, d	f 带进位循环右移	1	00	1100	dfff	ffff	C	2
SUBWF f, d	f 减去 W	1	00	0010	dfff	ffff	C、DC 和 Z	2
SUBWFB f, d	f 减去 W (带借位)	1	11	1011	dfff	ffff	C、DC 和 Z	2
SWAPF f, d	将 f 中的两个半字节进行交换	1	00	1110	dfff	ffff		2
XORWF f, d	W 和 f 作逻辑异或运算	1	00	0110	dfff	ffff	Z	2
针对字节的跳过操作类指令								
DECFSZ f, d	f 递减 1, 为 0 则跳过	1 (2)	00	1011	dfff	ffff		1 和 2
INCFSZ f, d	f 递增 1, 为 0 则跳过	1 (2)	00	1111	dfff	ffff		1 和 2
针对位的文件寄存器操作类指令								
BCF f, b	将 f 中的某位清零	1	01	00bb	bfff	ffff		2
BSF f, b	将 f 中的某位置 1	1	01	01bb	bfff	ffff		2
针对位的跳过操作类指令								
BTFS f, b	测试 f 中的某位, 为 0 则跳过	1 (2)	01	10bb	bfff	ffff		1 和 2
BTFF f, b	测试 f 中的某位, 为 1 则跳过	1 (2)	01	11bb	bfff	ffff		1 和 2
立即数操作类指令								
ADDLW k	立即数与 W 相加	1	11	1110	kkkk	kkkk	C、DC 和 Z	
ANDLW k	立即数和 W 作逻辑与运算	1	11	1001	kkkk	kkkk	Z	
IORLW k	立即数和 W 作逻辑或运算	1	11	1000	kkkk	kkkk	Z	
MOVLB k	将立即数送入 BSR	1	00	0000	001k	kkkk		
MOVLP k	将立即数送入 PCLATH	1	11	0001	1kkk	kkkk		
MOVLW k	将立即数传送到 W	1	11	0000	kkkk	kkkk		
SUBLW k	立即数减去 W	1	11	1100	kkkk	kkkk	C、DC 和 Z	
XORLW k	立即数和 W 作逻辑异或运算	1	11	1010	kkkk	kkkk	Z	

注 1：如果程序计数器 (PC) 被修改或条件测试结果为真，则该指令需要两个周期。第二个周期执行一条 NOP 指令。

2：如果该指令寻址的是 INDF 寄存器，并且相应 FSR 的 MSb 置 1，则该指令将需要一个额外的指令周期。

表 26-3：增强型中档指令设置（续）

助记符, 操作数	说明	周期数	14 位操作码				受影响的状态位	注	
			MSb	LSb					
控制操作类指令									
BRA k	相对跳转	2	11	001k	kkkk	kkkk			
BRW -	使用 W 进行相对跳转	2	00	0000	0000	1011			
CALL k	调用子程序	2	10	0kkk	kkkk	kkkk			
CALLW -	使用 W 调用子程序	2	00	0000	0000	1010			
GOTO k	跳转到地址	2	10	1kkk	kkkk	kkkk			
RETFIE k	从中断返回	2	00	0000	0000	1001			
RETLW k	返回并将立即数送入 W	2	11	0100	kkkk	kkkk			
RETURN -	从子程序返回	2	00	0000	0000	1000			
固有操作类指令									
CLRWDT -	将看门狗定时器清零	1	00	0000	0110	0100	TO 和 PD		
NOP -	空操作	1	00	0000	0000	0000			
OPTION -	将 W 的内容装入 OPTION_REG 寄存器	1	00	0000	0110	0010			
RESET -	软件器件复位	1	00	0000	0000	0001			
SLEEP -	进入待机模式	1	00	0000	0110	0011			
TRIS f	将 W 的内容装入 TRIS 寄存器	1	00	0000	0110	0fff			
C 编译器优化的指令									
ADDFSR n, k	将立即数 k 加到 FSRn	1	11	0001	0nk	kkkk	Z	2 和 3	
MOVIW n mm	将间接寄存器 FSRn 传送到 W, 带有预 / 后递增 / 递减修改量 mm	1	00	0000	0001	0nm mm kkkk			
MOVWI n mm	k[n]	将 INDFn 送入 W, 间接变址寻址	1	11	1111	0nk			
		将 W 传送到间接寄存器 FSRn, 带有预 / 后递增 / 递减修改量 mm	1	00	0000	0001	kkkk		
	k[n]	将 W 送入 INDFn, 间接变址寻址	1	11	1111	1nk		2	

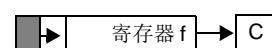
注 1: 如果程序计数器（Program Counter, PC）被修改或条件测试结果为真，则该指令需要两个周期。第二个周期执行一条 NOP 指令。

2: 如果该指令寻址的是 INDF 寄存器，并且相应 FSR 的 MSb 置 1，则该指令将需要一个额外的指令周期。

3: 请参见 MOVIW 和 MOVWI 指令说明。

26.2 指令说明

ADDSR	将立即数加到 FSRn	ANDLW	立即数和 W 作逻辑与运算
语法:	[标号] ADDSR FSRn, k	语法:	[标号] ANDLW k
操作数:	-32 ≤ k ≤ 31	操作数:	0 ≤ k ≤ 255
	n ∈ [0, 1]		
操作:	FSR(n) + k → FSR(n)	操作:	(W) .AND.(k) → (W)
受影响的状态位:	无	受影响的状态位:	Z
说明:	将有符号 6 位立即数 k 加到 FSRnH:FSRnL 寄存器对的内容。	说明:	将 W 寄存器的内容与 8 位立即数 k 进行逻辑与运算。结果存入 W 寄存器。
	FSRn 地址范围限制为 0000h-FFFFh。传送地址超出边界时, FSR 会发生折回。		
ADDLW	立即数与 W 相加	ANDWF	W 和 f 作逻辑与运算
语法:	[标号] ADDLW k	语法:	[标号] ANDWF f,d
操作数:	0 ≤ k ≤ 255	操作数:	0 ≤ f ≤ 127
			d ∈ [0,1]
操作:	(W) + k → (W)	操作:	(W) .AND.(f) → (目标寄存器)
受影响的状态位:	C、DC 和 Z	受影响的状态位:	Z
说明:	将 W 寄存器的内容与 8 位立即数 k 相加, 结果存入 W 寄存器。	说明:	将 W 寄存器的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。
ADDWF	W 与 f 相加	ASRF	算术右移
语法:	[标号] ADDWF f,d	语法:	[标号] ASRF f {,d}
操作数:	0 ≤ f ≤ 127	操作数:	0 ≤ f ≤ 127
	d ∈ [0,1]		d ∈ [0,1]
操作:	(W) + (f) → (目标寄存器)	操作:	(f<7>) → dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
受影响的状态位:	C、DC 和 Z	受影响的状态位:	C 和 Z
说明:	将 W 寄存器的内容与寄存器 f 的内容相加。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。	说明:	将寄存器 f 的内容连同进位标志位一起右移 1 位。MSb 保持不变。如果 d 为 0, 结果存入 W 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 内容、进位标志位与数据存储单元 f 的内容相加。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存入数据存储单元 f。		



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

BTFSC	测试 f 中的某位, 为 0 则跳过
语法:	[标号] BTFSC f,b
操作数:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
操作:	如果 $(f) = 0$, 则跳过
受影响的状态位:	无
说明:	如果寄存器 f 的 bit b 为 1, 则执行下一条指令。 如果寄存器 f 的 bit b 为 0, 则丢弃下一条指令, 改为执行一条 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:	$-256 \leq label - PC + 1 \leq 255$ $-256 \leq k \leq 255$
操作:	$(PC) + 1 + k \rightarrow PC$
受影响的状态位:	无
说明:	将有符号 9 位立即数 k 加到 PC。由于 PC 将递增以便取出下一条指令, 所以新地址将为 $PC + 1 + k$ 。该指令为一条双周期指令。该跳转的地址范围存在限制。

BTFSS	测试 f 中的某位, 为 1 则跳过
语法:	[标号] BTFSS f,b
操作数:	$0 \leq f \leq 127$ $0 \leq b < 7$
操作:	如果 $(f) = 1$, 则跳过
受影响的状态位:	无
说明:	如果寄存器 f 的 bit b 为 0, 则执行下一条指令。 如果 bit b 为 1, 则丢弃下一条指令, 改为执行一条 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:	$(PC) + (W) \rightarrow PC$
受影响的状态位:	无
说明:	将 W 的内容 (无符号) 加到 PC。由于 PC 将递增以便取出下一条指令, 所以新地址将为 $PC + 1 + (W)$ 。该指令为一条双周期指令。

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

PIC16(L)F1574/5/8/9

CALL	调用子程序	CLRWDT	将看门狗定时器清零
语法:	[标号] CALL k	语法:	[标号] CLRWDT
操作数:	$0 \leq k \leq 2047$	操作数:	无
操作:	$(PC)+1 \rightarrow TOS,$ $k \rightarrow PC<10:0>,$ $(PCLATH<6:3>) \rightarrow PC<14:11>$	操作:	$00h \rightarrow WDT$ $0 \rightarrow \underline{WDT}$ 预分频器, $1 \rightarrow \underline{TO}$ $1 \rightarrow \underline{PD}$
受影响的状态位:	无	受影响的状态位:	\overline{TO} 和 \overline{PD}
说明:	调用子程序。首先, 将返回地址 (PC + 1) 压入堆栈。11 位直接地址被装入 PC 的 bit<10:0>。PC 的高 4 位值从 PCLATH 装入。CALL 是一条双周期指令。	说明:	CLRWDT 指令复位看门狗定时器及其预分频器。状态位 \overline{TO} 和 \overline{PD} 置 1。
CALLW	使用 W 调用子程序	COMF	对 f 取反
语法:	[标号] CALLW	语法:	[标号] COMF f,d
操作数:	无	操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(PC)+1 \rightarrow TOS,$ $(W) \rightarrow PC<7:0>,$ $(PCLATH<6:0>) \rightarrow PC<14:8>$	操作:	$(\bar{f}) \rightarrow (\text{目标寄存器})$
受影响的状态位:	无	受影响的状态位:	Z
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。	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。
CLRF	将 f 清零	DEC F	f 递减 1
语法:	[标号] CLRF f	语法:	[标号] DEC F f,d
操作数:	$0 \leq f \leq 127$	操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$00h \rightarrow (f)$ $1 \rightarrow Z$	操作:	$(f) - 1 \rightarrow (\text{目标寄存器})$
受影响的状态位:	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。	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。
CLRW	将 W 清零		
语法:	[标号] CLRW		
操作数:	无		
操作:	$00h \rightarrow (W)$ $1 \rightarrow Z$		
受影响的状态位:	Z		
说明:	W 寄存器被清零。全零标志位 (Z) 被置 1。		

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

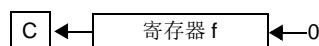
GOTO	无条件跳转
语法:	[标号] GOTO k
操作数:	$0 \leq k \leq 2047$
操作:	$k \rightarrow \text{PC} <10:0>$ $\text{PCLATH} <6:3> \rightarrow \text{PC} <14:11>$
受影响的状态位:	无
说明:	GOTO 是一条无条件跳转指令。11 位立即数值被装入 PC 的 <10:0> 位。PC 的高 4 位从 PCLATH<4:3> 装入。 GOTO 是一条双周期指令。

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。

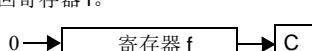
INCF	f 递增 1
语法:	[标号] INCF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) + 1 \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将寄存器 f 的内容递增 1。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回 f 寄存器。

IORWF	W 和 f 作逻辑或运算
语法:	[标号] IORWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) .OR.(f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将 W 寄存器的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回 f 寄存器。

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

MOVIW	将 INDFn 的内容传送到 W	MOVLP	将立即数送入 PCLATH
语法:	[标号] MOVIW ++FSRn [标号] MOVIW --FSRn [标号] MOVIW FSRn++ [标号] MOVIW FSRn-- [标号] MOVIW k[FSRn]	语法:	[标号] MOVLP k
操作数:	n ∈ [0,1] mm ∈ [00,01, 10, 11] -32 ≤ k ≤ 31	操作数:	0 ≤ k ≤ 127
操作:	INDFn → 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 (全递增) • FSR - 1 (全递减) • 不变	操作:	k → PCLATH
受影响的状态位:	Z	受影响的状态位:	无
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 / 递减到超出边界时, 将导致地址发生折回。	说明:	将 7 位立即数 k 装入 PCLATH 寄存器。
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	
示例:	MOVWI 0x5A	执行指令后	W = 0x5A

MOVLW	将立即数传送到 W
语法:	[标号] MOVLW k
操作数:	0 ≤ k ≤ 255
操作:	k → (W)
受影响的状态位:	无
说明:	将 8 位立即数 k 装入 W 寄存器。无关位均汇编为 0。
指令字数:	1
指令周期数:	1
示例:	MOVLW 0x5A
	执行指令后
	W = 0x5A

MOVWF	将 W 的内容送入 f
语法:	[标号] MOVWF f
操作数:	0 ≤ f ≤ 127
操作:	(W) → (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

MOVLB	将立即数送入 BSR
语法:	[标号] MOVLB k
操作数:	0 ≤ k ≤ 31
操作:	k → BSR
受影响的状态位:	无
说明:	将 5 位立即数 k 装入存储区选择寄存器 (BSR)。

MOVWI 将 W 的内容送入 INDFn

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

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

操作:
 $W \rightarrow INDFn$
 有效地址通过以下方式确定
 • 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 (全递增)
 • FSR - 1 (全递减)
 • 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 / 递减到超出边界时, 将导致地址发生折回。

对于 FSRn 的递增 / 递减操作不会影响任何状态位。

NOP 空操作

语法: [标号] NOP
 操作数: 无
 操作: 空操作
 受影响的状态位: 无
 说明: 不执行任何操作。
 指令字数: 1
 指令周期数: 1
 示例: NOP

OPTION 将 W 的内容装入 OPTION_REG 寄存器

语法: [标号] OPTION
 操作数: 无
 操作: (W) → OPTION_REG
 受影响的状态位: 无
 说明: 将 W 寄存器的数据传送到 OPTION_REG 寄存器。

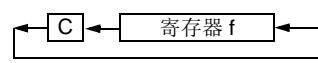
RESET 软件复位

语法: [标号] RESET
 操作数: 无
 操作: 执行器件复位。复位 PCON 寄存器的 nRI 标志。
 受影响的状态位: 无
 说明: 此指令可实现用软件执行硬件复位。

RETFIE	从中断返回
语法:	[标号] RETFIE
操作数:	无
操作:	TOS → PC, 1 → GIE
受影响的状态位:	无
说明:	从中断返回。执行出栈操作，将栈顶 (Top-of-Stack, TOS) 的内容装入 PC。通过将全局中断允许位 GIE (INTCON<7>) 置 1 来允许中断。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	RETFIE 中断后 PC = TOS GIE = 1

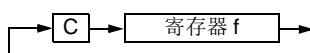
RETURN	从子程序返回
语法:	[标号] RETURN
操作数:	无
操作:	TOS → PC
受影响的状态位:	无
说明:	从子程序返回。执行出栈操作，将栈顶 (TOS) 内容装入程序计数器。这是一条双周期指令。

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

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

RRF f 带进位循环右移

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



SLEEP 进入休眠模式

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

SUBWF f 减去 W

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

C = 0	W > f
C = 1	W ≤ f
DC = 0	$W<3:0> > f<3:0>$
DC = 1	$W<3:0> \leq f<3:0>$

SUBWFB f 减去 W (带借位)

语法:	SUBWFB f {,d}
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) - (W) - (\bar{B}) \rightarrow \text{dest}$
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。

SWAPF	将 f 中的两个半字节进行交换
语法:	[标号] SWAPF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f<3:0>) \rightarrow (\text{目标寄存器 } <7:4>)$, $(f<7:4>) \rightarrow (\text{目标寄存器 } <3:0>)$
受影响的状态位:	无
说明:	寄存器 f 的高半字节和低半字节相互交换。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存入寄存器 f。

XORLW	立即数和 W 作逻辑异或运算
语法:	[标号] XORLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) .XOR. k \rightarrow (W)$
受影响的状态位:	Z
说明:	将 W 寄存器的内容与 8 位立即数 k 进行逻辑异或运算。结果存入 W 寄存器。

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

XORWF	W 和 f 作逻辑异或运算
语法:	[标号] XORWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) .XOR.(f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将 W 寄存器的内容与寄存器 f 的内容进行逻辑异或运算。如果 d 为 0, 结果存入 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

27.0 电气规范

27.1 绝对最大值^(†)

偏置时的环境温度	-40°C 至 +125°C
储存温度	-65°C 至 +150°C
引脚相对于 VSS 的电压	
VDD 引脚	
PIC16F1574/5/8/9	-0.3V 至 +6.5V
PIC16LF1574/5/8/9	-0.3V 至 +4.0V
MCLR 引脚	-0.3V 至 +9.0V
所有其他引脚	-0.3V 至 (VDD + 0.3V)
最大电流	
VSS 引脚 ⁽¹⁾	
-40°C ≤ TA ≤ +85°C	250 mA
+85°C ≤ TA ≤ +125°C	85 mA
VDD 引脚 ⁽¹⁾	
-40°C ≤ TA ≤ +85°C	250 mA
+85°C ≤ TA ≤ +125°C	85 mA
任一标准 I/O 引脚的灌电流	50 mA
任一标准 I/O 引脚的拉电流	50 mA
钳位电流, IK (VPIN < 0 或 VPIN > VDD)	±20 mA
总功耗 ⁽²⁾	800 mW

注 1: 最大电流值要求 I/O 引脚上具有均匀的负载分布。最大电流值可以通过器件封装功率耗散特性进行限制, 请参见表 27-6: “温度特性” 来计算器件规范值。
2: 功耗按如下公式计算: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ 。

† 注: 如果器件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值, 我们建议不要使器件在或超过该规范规定的条件下运行。器件长时间工作在最大值条件下, 其可靠性会受到影响。

27.2 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⁾

PIC16LF1574/5/8/9

V_{DDMIN} ($F_{osc} \leq 16$ MHz)	+1.8V
V_{DDMIN} ($F_{osc} \leq 32$ MHz)	+2.5V
V_{DDMAX}	+3.6V

PIC16F1574/5/8/9

V_{DDMIN} ($F_{osc} \leq 16$ MHz)	+2.3V
V_{DDMIN} ($F_{osc} \leq 32$ 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

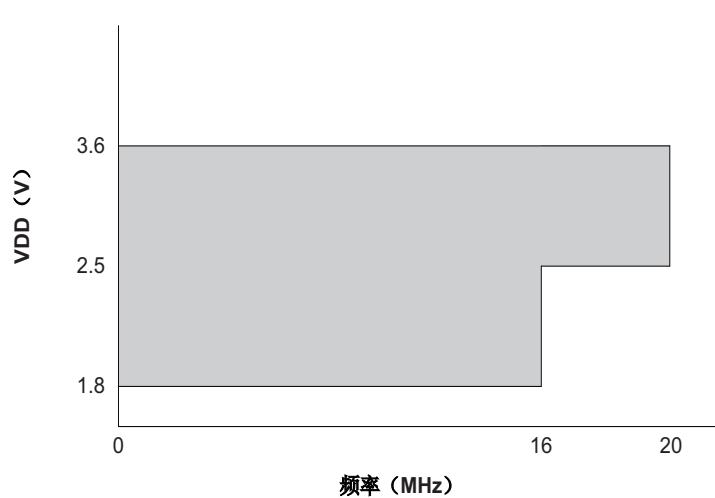
扩展级温度

T_{A_MIN}	-40°C
T_{A_MAX}	+125°C

注 1: 请参见直流特性：供电电压中的参数 D001。

PIC16(L)F1574/5/8/9

图 27-1：电压 — 频率关系图， $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$ ，仅限 PIC16F1574/5/8/9

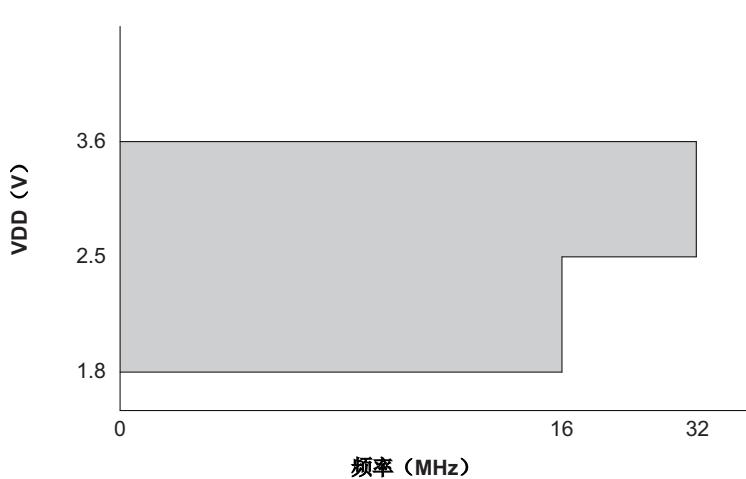


版本: 10-000131A
2013/8/5

注 1: 阴影区域表示允许的电压频率组合。

2: 请参见表 27-7 了解每种振荡器模式所支持的频率。

图 27-2：电压 — 频率关系图， $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$ ，仅限 PIC16LF1574/5/8/9



版本: 10-000131B
2013/9/19

注 1: 阴影区域表示允许的电压频率组合。

2: 请参见表 27-7 了解每种振荡器模式所支持的频率。

27.3 直流特性

表 27-1：供电电压

PIC16LF1574/5/8/9		标准工作条件（除非另外声明）					
PIC16F1574/5/8/9							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D001	VDD	供电电压					
			VDDMIN 1.8 2.5	— —	VDDMAX 3.6 3.6	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz (注 3)
D001			2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz FOSC ≤ 32 MHz (注 3)
D002*	VDR	RAM 数据保持电压 ⁽¹⁾					
			1.5	—	—	V	器件处于休眠模式
D002*			1.7	—	—	V	器件处于休眠模式
D002A*	VPOR	上电复位释放电压 ⁽²⁾					
			—	1.6	—	V	
D002A*			—	1.6	—	V	
D002B*	VPORR*	上电复位重新激活电压 ⁽²⁾					
			—	0.8	—	V	
D002B*			—	1.5	—	V	
D003	VFVR	固定参考电压	—	1.024	—	V	-40°C ≤ TA ≤ +85°C
D003A	VADFVR	ADC 的 FVR 增益电压精度	-4 -4 -5	—	4 4 5	%	1x VFVR, ADFVR = 01, VDD ≥ 2.5V 2x VFVR, ADFVR = 10, VDD ≥ 2.5V 4x VFVR, ADFVR = 11, VDD ≥ 4.75V
D003B	VCDAFVR	比较器的 FVR 增益电压精度	-4 -4 -5	—	4 4 5	%	1x VFVR, CDAFVR = 01, VDD ≥ 2.5V 2x VFVR, CDAFVR = 10, VDD ≥ 2.5V 4x VFVR, CDAFVR = 11, 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。

* 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。

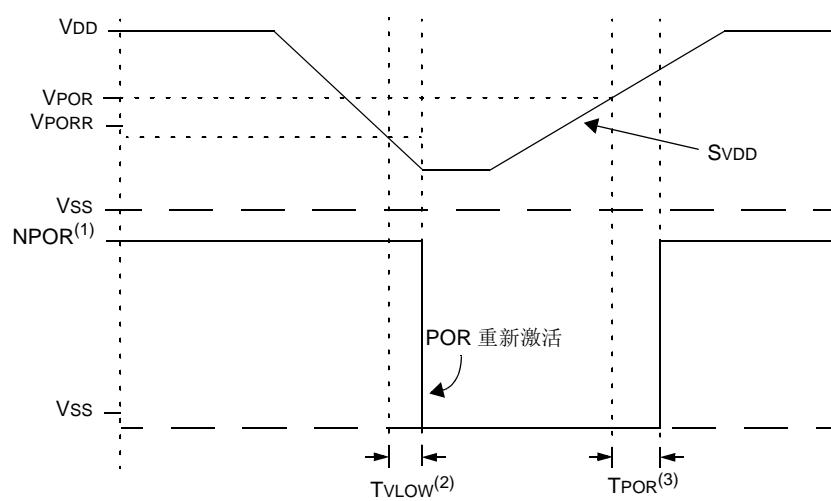
注 1：这是在不丢失 RAM 数据的前提下，休眠模式下 VDD 所能降到的最小电压值。

2：请参见图 27-3 “VDD 缓慢上升时，POR 和 POR 重新激活”。

3：32 MHz 工作需要使用 PLL。

PIC16(L)F1574/5/8/9

图 27-3: VDD 缓慢上升时, POR 和 POR 重新激活



注 1: 当 NPOR 为低电平时, 器件保持在复位状态。

2: TPOR 典型值为 1 μ s。

3: TVLOW 典型值为 2.7 μ s。

表 27-2: 供电电流 (IDD) (1,2)

PIC16LF1574/5/8/9		标准工作条件 (除非另外声明)					
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						VDD	注
D013		—	41	51	µA	1.8	FOSC = 1 MHz, 外部时钟 (ECM), 中等功耗模式
		—	69	80	µA	3.0	
D013		—	79	107	µA	2.3	FOSC = 1 MHz, 外部时钟 (ECM), 中等功耗模式
		—	105	138	µA	3.0	
		—	151	184	µA	5.0	
D014		—	134	152	µA	1.8	FOSC = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	234	268	µA	3.0	
D014		—	201	255	µA	2.3	FOSC = 4 MHz, 外部时钟 (ECM), 中等功耗模式
		—	270	329	µA	3.0	
		—	344	431	µA	5.0	
D015		—	7	19	µA	1.8	FOSC = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	9	20	µA	3.0	
D015		—	15	25	µA	2.3	FOSC = 31 kHz, LFINTOSC, -40°C ≤ TA ≤ +85°C
		—	18	28	µA	3.0	
		—	20	29	µA	5.0	
D016		—	128	174	µA	1.8	FOSC = 500 kHz, MFINTOSC
		—	153	203	µA	3.0	
D016		—	166	241	µA	2.3	FOSC = 500 kHz, MFINTOSC
		—	187	273	µA	3.0	
		—	249	332	µA	5.0	
D017*		—	0.6	0.7	mA	1.8	FOSC = 8 MHz, HFINTOSC
		—	0.9	1.1	mA	3.0	
D017*		—	0.7	1.0	mA	2.3	FOSC = 8 MHz, HFINTOSC
		—	1.0	1.1	mA	3.0	
		—	1.1	1.2	mA	5.0	
D018		—	0.9	1.0	mA	1.8	FOSC = 16 MHz, HFINTOSC
		—	1.3	1.4	mA	3.0	
D018		—	1.1	1.3	mA	2.3	FOSC = 16 MHz, HFINTOSC
		—	1.3	1.5	mA	3.0	
		—	1.5	1.8	mA	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：CLKIN = 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；禁止 WDT。

2: 供电电流主要受工作电压和频率的影响。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等其他因素也会对电流消耗产生影响。

3: 32 MHz 工作需要使用 PLL。

PIC16(L)F1574/5/8/9

表 27-2： 供电电流 (ID_D) (1,2) (续)

PIC16LF1574/5/8/9		标准工作条件 (除非另外声明)					
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						V _{DD}	注
D018A*		—	2.3	2.8	mA	3.0	Fosc = 32 MHz, HFINTOSC (注 3)
D018A*		—	2.5	2.9	mA	3.0	Fosc = 32 MHz, HFINTOSC (注 3)
		—	2.6	3.0	mA	5.0	
D019A		—	2.0	2.2	mA	3.0	Fosc = 32 MHz, 外部时钟 (ECH), 高功耗模式 (注 3)
D019A		—	2.1	2.3	mA	3.0	Fosc = 32 MHz, 外部时钟 (ECH), 高功耗模式 (注 3)
		—	2.2	2.7	mA	5.0	
D019B		—	2.6	16	μA	1.8	Fosc = 32 kHz, 外部时钟 (ECL), 低功耗模式
		—	5.0	22	μA	3.0	
D019B		—	14	23	μA	2.3	Fosc = 32 kHz, 外部时钟 (ECL), 低功耗模式
		—	18	29	μA	3.0	
		—	20	30	μA	5.0	
D019C		—	21	29	μA	1.8	Fosc = 500 kHz, 外部时钟 (ECL), 低功耗模式
		—	35	44	μA	3.0	
D019C		—	34	46	μA	2.3	Fosc = 500 kHz, 外部时钟 (ECL), 低功耗模式
		—	43	59	μA	3.0	
		—	49	61	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 ID_D 测量值的测试条件为：CLKIN = 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}；禁止 WDT。
- 2：供电电流主要受工作电压和频率的影响。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等其他因素也会对电流消耗产生影响。
- 3：32 MHz 工作需要使用 PLL。

表 27-3: 掉电电流 (IPD) (1,2)

PIC16LF1574/5/8/9		工作条件: (除非另外声明) 低功耗休眠模式						
PIC16F1574/5/8/9		低功耗休眠模式, VREGPM = 1						
参数编号	器件特性	最小值	典型值 †	最大值 +85°C	最大值 +125°C	单位	条件	
							VDD	注
D022	IPD 基本电流	—	0.10	1	8	μA	1.8	禁止 WDT、BOR 和 FVR, 所有外设不工作
		—	0.10	2	9	μA	3.0	
D022	IPD 基本电流	—	0.3	3	10	μA	2.3	禁止 WDT、BOR 和 FVR, 所有外设不工作, 低功耗休眠模式, VREGPM = 1。
		—	0.4	4	12	μA	3.0	
		—	0.5	6	15	μA	5.0	
D022A	IPD 基本电流	—	10.4	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
		—	12.7	18	20	μA	3.0	
		—	13.8	21	26	μA	5.0	
D023		—	0.4	2	9	μA	1.8	WDT 电流
		—	0.6	3	10	μA	3.0	
D023		—	0.6	6	15	μA	2.3	WDT 电流
		—	0.7	7	20	μA	3.0	
		—	0.9	8	22	μA	5.0	
D023A		—	15	28	30	μA	1.8	FVR 电流
		—	26	33	34	μA	3.0	
D023A		—	19	28	30	μA	2.3	FVR 电流
		—	22	35	36	μA	3.0	
		—	23	38	41	μA	5.0	
D024		—	7.5	17	20	μA	3.0	BOR 电流
D024		—	8.1	17	30	μA	3.0	BOR 电流
		—	9.2	20	40	μA	5.0	
D24A		—	0.3	4	10	μA	3.0	LPBOR 电流
D24A		—	0.5	5	14	μA	3.0	LPBOR 电流
		—	0.6	8	17	μA	5.0	
D026		—	0.1	1.5	9	μA	1.8	ADC 电流 (注 3), 不在进行转换
		—	0.1	2.7	10	μA	3.0	
D026		—	0.3	4	11	μA	2.3	ADC 电流 (注 3), 不在进行转换
		—	0.4	5	13	μA	3.0	
		—	0.5	8	16	μA	5.0	
D026A*		—	288	—	—	μA	1.8	ADC 电流 (注 3), 正在进行转换
		—	288	—	—	μA	3.0	
D026A*		—	322	—	—	μA	2.3	ADC 电流 (注 3), 正在进行转换
		—	322	—	—	μA	3.0	
		—	322	—	—	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: 可通过从该限定值中减去 IPD 基本电流来确定外设 Δ 电流。在计算总电流消耗时应使用最大值。**2:** 在休眠模式下, 掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 Vss 时测得的。**3:** ADC 时钟源是 FRC。

PIC16(L)F1574/5/8/9

表 27-3: 掉电电流 (IPD) (1,2) (续)

PIC16LF1574/5/8/9		工作条件: (除非另外声明) 低功耗休眠模式						
PIC16F1574/5/8/9		低功耗休眠模式, VREGPM = 1						
参数编号	器件特性	最小值	典型值 †	最大值 +85°C	最大值 +125°C	单位	条件	
							VDD	注
D027		—	5	22	25	μA	1.8	比较器, CxSP = 0
		—	5	23	27	μA	3.0	
D027		—	15	23	25	μA	2.3	比较器, CxSP = 0
		—	17	27	29	μA	3.0	
		—	19	28	30	μA	5.0	
D028A		—	23	41	42	μA	1.8	比较器, 正常功耗, CxSP = 1 (注 1)
		—	25	42	44	μA	3.0	
D028A		—	33	55	56	μA	2.3	比较器, 正常功耗, CxSP = 1 VREGPM = 1 (注 1)
		—	34	59	60	μA	3.0	
		—	36	60	61	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: 可通过从该限定值中减去 IPD 基本电流来确定外设 Δ 电流。在计算总电流消耗时应使用最大值。
 2: 在休眠模式下, 掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 Vss 时测得的。
 3: ADC 时钟源是 FRC。

表 27-4: 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	VIL VIH	输入低电压					
		I/O 端口:					
		带 TTL 缓冲器	—	0.8	V		$4.5V \leq VDD \leq 5.5V$
			—	0.15 VDD	V		$1.8V \leq VDD \leq 4.5V$
		带施密特触发器缓冲器	—	0.2 VDD	V		$2.0V \leq VDD \leq 5.5V$
		带 I ² C 电平	—	0.3 VDD	V		
		带 SMBus 电平	—	0.8	V		$2.7V \leq VDD \leq 5.5V$
		MCLR	—	0.2 VDD	V		
		输入高电压					
		I/O 端口:					
D040 D040A D041 D042	VIL VIH	带 TTL 缓冲器	2.0	—	—	V	$4.5V \leq VDD \leq 5.5V$
			0.25 VDD + 0.8	—	—	V	$1.8V \leq VDD \leq 4.5V$
		带施密特触发器缓冲器	0.8 VDD	—	—	V	$2.0V \leq VDD \leq 5.5V$
		带 I ² C 电平	0.7 VDD	—	—	V	
		带 SMBus 电平	2.1	—	—	V	$2.7V \leq VDD \leq 5.5V$
		MCLR	0.8 VDD	—	—	V	
D060 D061	IIL	输入泄漏电流(1)					
		I/O 端口	—	±5	±125	nA	$VSS \leq VPIN \leq VDD$, 引脚处于高阻态, 85°C
			—	±5	±1000	nA	$VSS \leq VPIN \leq VDD$, 引脚处于高阻态, 125°C
		MCLR ⁽²⁾	—	±50	±200	nA	$VSS \leq VPIN \leq VDD$, 引脚处于高阻态, 85°C
D070*	IPUR	弱上拉电流					
			25	100	200	μA	$VDD = 3.3V$, $VPIN = VSS$
			25	140	300	μA	$VDD = 5.0V$, $VPIN = VSS$
D080	VOL	输出低电压					
		I/O 端口	—	—	0.6	V	$IOL = 8 mA$, $VDD = 5V$ $IOL = 6 mA$, $VDD = 3.3V$ $IOL = 1.8 mA$, $VDD = 1.8V$
D090	VOH	输出高电压					
		I/O 端口	VDD - 0.7	—	—	V	$IOH = 3.5 mA$, $VDD = 5V$ $IOH = 3 mA$, $VDD = 3.3V$ $IOH = 1 mA$, $VDD = 1.8V$
D101A*	CIO	输出引脚上的容性负载规范					
		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。

表 27-5： 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	V _{IHH}	程序存储器编程规范 MCLR/V _{PP} 引脚上的电压	8.0	—	9.0	V	(注 2)
D111	I _{DDP}	编程时的供电电流	—	—	10	mA	
D112	V _{BE}	批量擦除时的 V _{DD}	2.7	—	V _{DDMAX}	V	
D113	V _{PEW}	写或行擦除时的 V _{DD}	V _{DDMIN}	—	V _{DDMAX}	V	
D114	I _{PPPGM}	擦除 / 写操作时 MCLR/V _{PP} 上的电流	—	1.0	—	mA	
D115	I _{DDPGM}	擦除 / 写操作时 V _{DD} 上的电流	—	5.0	—	mA	
D121	E _P	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	V _{PRW}	读 / 写操作时的 V _{DD}	V _{DDMIN}	—	V _{DDMAX}	V	
D123	T _{IW}	自定时写周期时间	—	2	2.5	ms	
D124	T _{RETD}	特性保持时间	—	40	—	年	假设没有违反其他规范
D125	E _{HEFC}	高耐用性闪存单元	100k	—	—	E/W	0°C ≤ TA ≤ +60°C，最后 128 个地址的低字节

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

注 1：自写和块擦除。

2：仅当禁止单电源编程时才需要。

表 27-6：温度特性

标准工作条件（除非另外声明）

参数编号	符号	特性	典型值	单位	条件
TH01	θJA	热阻（结到环境）	70	°C/W	14 引脚 PDIP 封装
			95.3	°C/W	14 引脚 SOIC 封装
			100	°C/W	14 引脚 TSSOP 封装
			31.8	°C/W	16 引脚 UQFN 4x4 mm 封装
			62.2	°C/W	20 引脚 PDIP 封装
			77.7	°C/W	20 引脚 SOIC 封装
			87.3	°C/W	20 引脚 SSOP 封装
			32.8	°C/W	20 引脚 UQFN 4x4 mm 封装
TH02	θJC	热阻（结到管壳）	32.75	°C/W	14 引脚 PDIP 封装
			31	°C/W	14 引脚 SOIC 封装
			24.4	°C/W	14 引脚 TSSOP 封装
			24.4	°C/W	16 引脚 UQFN 4x4 mm 封装
			27.5	°C/W	20 引脚 PDIP 封装
			23.1	°C/W	20 引脚 SOIC 封装
			31.1	°C/W	20 引脚 SSOP 封装
			27.4	°C/W	20 引脚 UQFN 4x4 mm 封装
TH03	TJMAX	最高结温	150	°C	
TH04	PD	功耗	—	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	内部功耗	—	W	PINTERNAL = IDD × VDD ⁽¹⁾
TH06	PI/O	I/O 功耗	—	W	PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))
TH07	PDER	降额功耗	—	W	PDER = PDMAX (TJ - TA)/θJA ⁽²⁾

注 1: IDD 为输出引脚上不驱动任何负载时使芯片独立运行的电流。

2: TA = 环境温度, TJ = 结温。

PIC16(L)F1574/5/8/9

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

大写字母及其含义:

S		P	
F	下降	R	上升
H	高电平	V	有效
I	无效 (高阻)	Z	高阻
L	低电平		

图 27-4: 负载条件

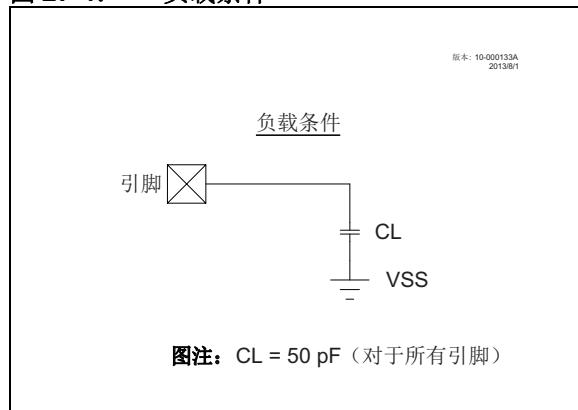


图 27-5: 时钟时序

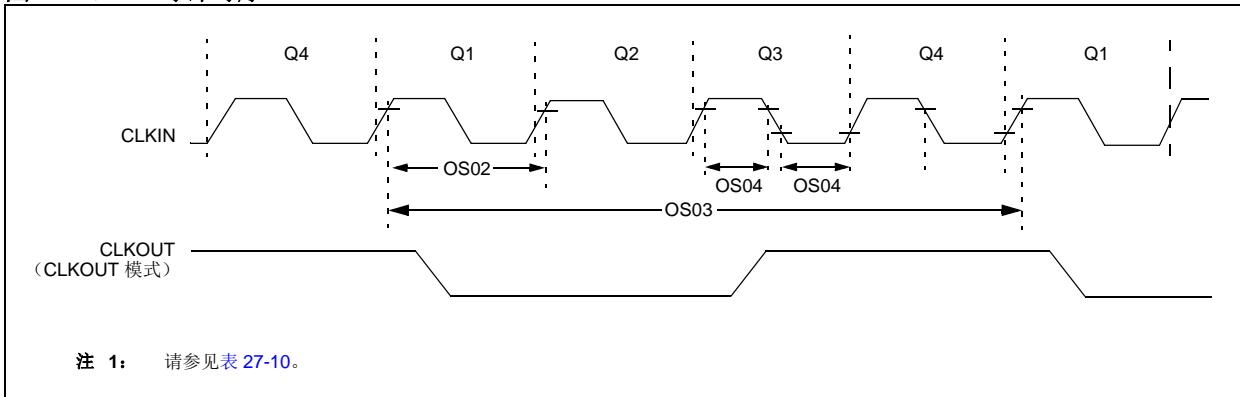


表 27-7: 时钟振荡器时序要求

标准工作条件（除非另外声明）

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS01	FOSC	外部 CLKIN 频率 (1)	DC	—	0.5	MHz	外部时钟 (ECL)
			DC	—	4	MHz	外部时钟 (ECM)
			DC	—	20	MHz	外部时钟 (ECH)
OS02	Tosc	外部 CLKIN 周期 (1)	50	—	∞	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）。

PIC16(L)F1574/5/8/9

表 27-8：振荡器参数

标准工作条件（除非另外声明）								
参数编号	符号	特性	频率容差	最小值	典型值 †	最大值	单位	条件
OS08	HFOSC	内部已校准的 HFINTOSC 频率 (1)	±2%	—	16.0	—	MHz	VDD = 3.0V, TA = 25°C, (注 2)
OS09	LFOSC	内部 LFINTOSC 频率	—	—	31	—	kHz	
OS10*	TWARM	HFINTOSC 从休眠模式唤醒的启动时间 LFINTOSC 从休眠模式唤醒的启动时间	—	—	5	15	μs	
			—	—	0.5	—	ms	

* 这些参数为特性值，但未经测试。

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

注 1：为了确保振荡器频率容差，必须尽可能靠近器件，在 VDD 和 VSS 之间接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

2：请参见图 27-6：“器件不同 VDD 和温度下的 HFINTOSC 频率精度”。

图 27-6：器件不同 VDD 和温度下的 HFINTOSC 频率精度

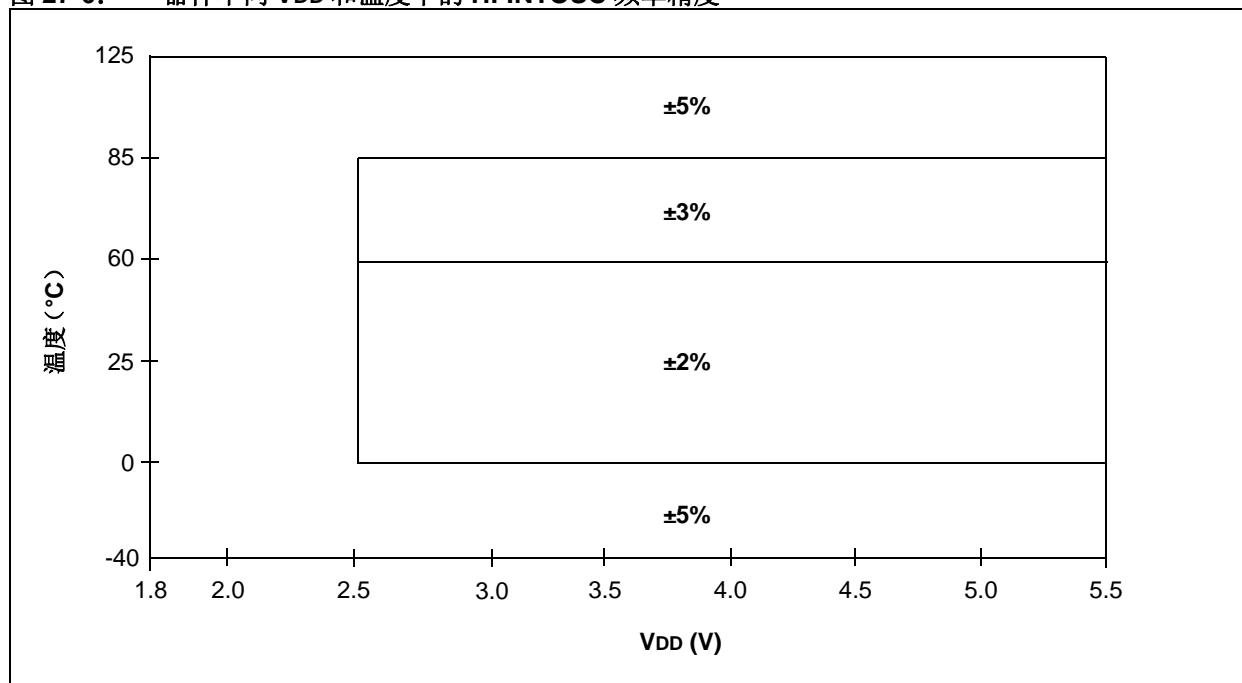


表 27-9：PLL 时钟时序规范 (VDD = 2.7V 至 5.5V)

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 启动时间（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 3V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

图 27-7: CLKOUT 和 I/O 时序

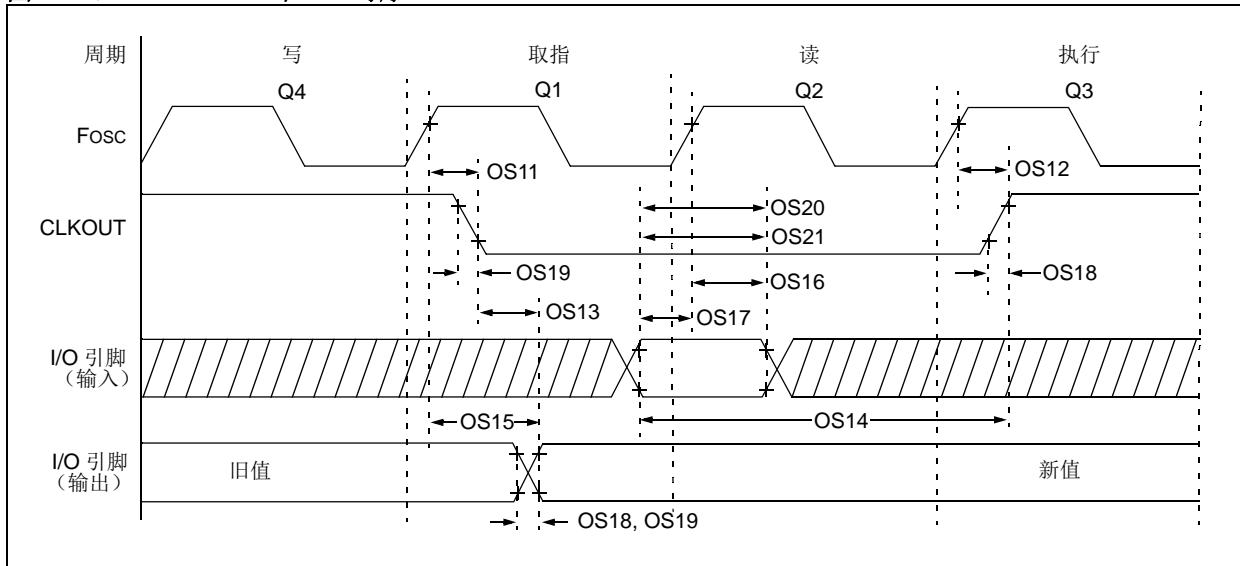


表 27-10: CLKOUT 和 I/O 时序参数

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 \uparrow 到 CLKOUT \downarrow 的时间 (1)	—	—	70	ns	3.3V \leq VDD \leq 5.0V
OS12	TosH2ckH	Fosc \uparrow 到 CLKOUT \uparrow 的时间 (1)	—	—	72	ns	3.3V \leq VDD \leq 5.0V
OS13	TckL2ioV	CLKOUT \downarrow 到端口输出有效的时间 (1)	—	—	20	ns	
OS14	TioV2ckH	CLKOUT \uparrow 之前端口输入有效的时间 (1)	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc \uparrow (Q1 周期) 到端口输出有效的时间	—	50	70*	ns	3.3V \leq VDD \leq 5.0V
OS16	TosH2iol	Fosc \uparrow (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 \leq VDD \leq 5.0V
OS17	TioV2osH	端口输入有效到 Fosc \uparrow (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 \leq VDD \leq 5.0V
S19*	TioF	端口输出下降时间	—	28 15	55 30	ns	VDD = 1.8V 3.3V \leq VDD \leq 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。

PIC16(L)F1574/5/8/9

图 27-8：复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序

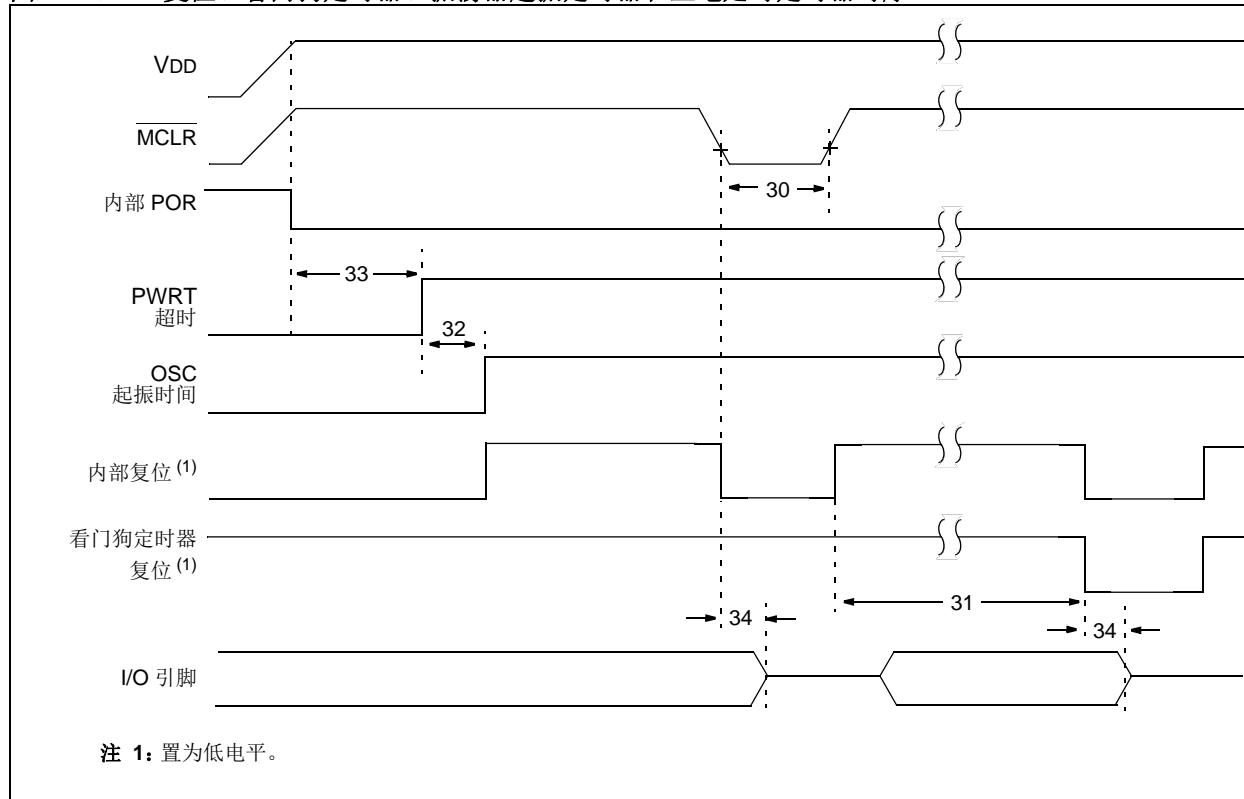


表 27-11：复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位参数

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 脉宽（低电平）	2	—	—	μs	
31	TWDTLP	低功耗看门狗定时器超时周期	10	16	27	ms	VDD = 3.3V-5V, 使用的预分频比为 1:512
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 2.35 1.80	2.70 2.45 1.90	2.85 2.58 2.05	V V V	BORV = 0 BORV = 1 (PIC16F1574/5/8/9) BORV = 1 (PIC16LF1574/5/8/9)
36*	VHYST	欠压复位滞后电压	0	25	60	mV	-40°C ≤ TA ≤ +85°C
37*	TBORDC	欠压复位直流响应时间	1	16	35	μs	VDD ≤ VBOR
38	VLPBOR	低功耗欠压复位电压	1.8	2.1	2.5	V	LPBOR = 1

* 这些参数为特性值，但未经测试。

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

注 1：根据设计，振荡器起振定时器（OST）计数前 1024 个周期，与频率无关。

2：为了确保这些电压容差，必须尽可能靠近器件，在 VDD 和 VSS 之间接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

图 27-9：欠压复位时序和特性

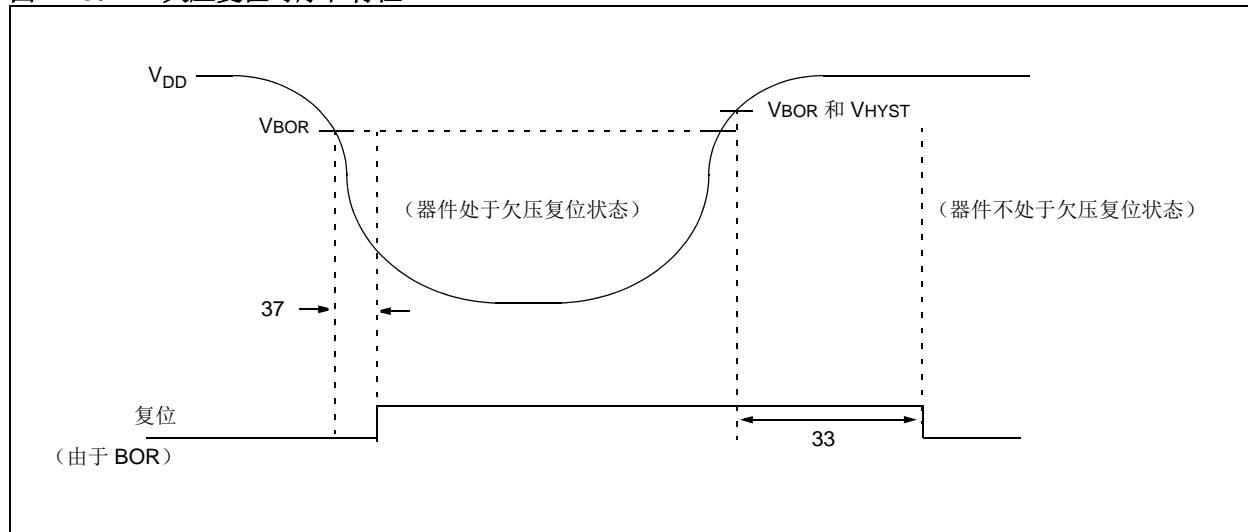


图 27-10: TIMER0 和 TIMER1 外部时钟时序

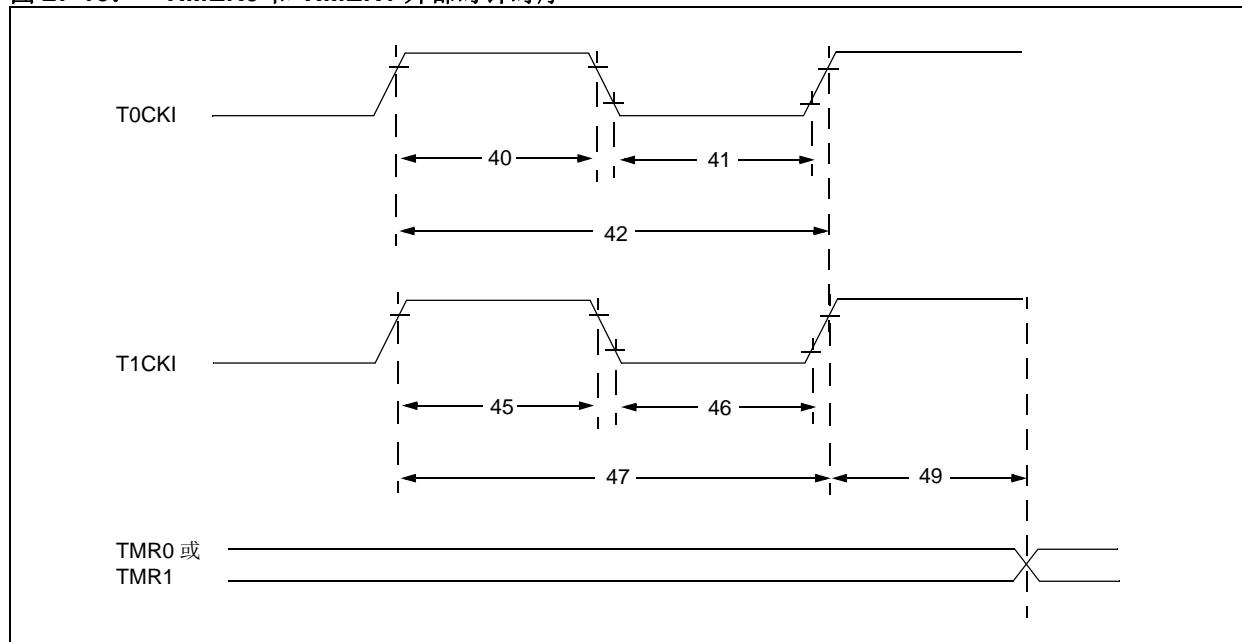


表 27-12: TIMER0 和 TIMER1 外部时钟要求

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 高电平脉宽	无预分频器	0.5 TCY + 20	—	—	ns	
			带预分频器	10	—	—	ns	
41*	TT0L	T0CKI 低电平脉宽	无预分频器	0.5 TCY + 20	—	—	ns	
			带预分频器	10	—	—	ns	
42*	TT0P	T0CKI 周期		取如下二者中 较大值: 20 或 $\frac{TCY + 40}{N}$	—	—	ns	N = 预分频值
45*	TT1H	T1CKI 高电 平时间	同步, 无预分频器	0.5 TCY + 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 TCY + 20	—	—	ns	
			同步, 带预分频器	15	—	—	ns	
			异步	30	—	—	ns	
47*	TT1P	T1CKI 输入 周期	同步		—	—	ns	N = 预分频值
			异步		60	—	—	
49*	TCKEZTMR1	从外部时钟边沿到定时器递增的延时		2 TOSC	—	7 TOSC	—	定时器工作在同步 模式下

* 这些参数为特性值，但未经测试。

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

表 27-13：模数转换器（ADC）特性^(1,2,3)

工作条件（除非另外声明）

VDD = 3.0V, TA = 25°C

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
AD01	NR	分辨率	—	—	10	位	
AD02	EIL	积分误差	—	±1	±1.7	LSb	VREF = 3.0V
AD03	EDL	微分误差	—	±1	±1	LSb	无丢失编码 VREF = 3.0V
AD04	E0FF	失调误差	—	±1	±2.5	LSb	VREF = 3.0V
AD05	EGN	增益误差	—	±1	±2.0	LSb	VREF = 3.0V
AD06	VREF	参考电压	1.8	—	VDD	V	VREF = (VRPOS - VRNEG) (注 4)
AD07	VAIN	满量程	VSS	—	VREF	V	
AD08	ZAIN	模拟信号源的推荐阻抗	—	—	10	kΩ	如果输入引脚上接有 0.01 μF 的外部电容，则该值可能更高。

* 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 28.0 节 “直流和交流特性图表”**。**4：**ADC VREF 通过 ADPREF<0> 位进行选择。

PIC16(L)F1574/5/8/9

图 27-11: ADC 转换时序 (ADC 时钟基于 Fosc)

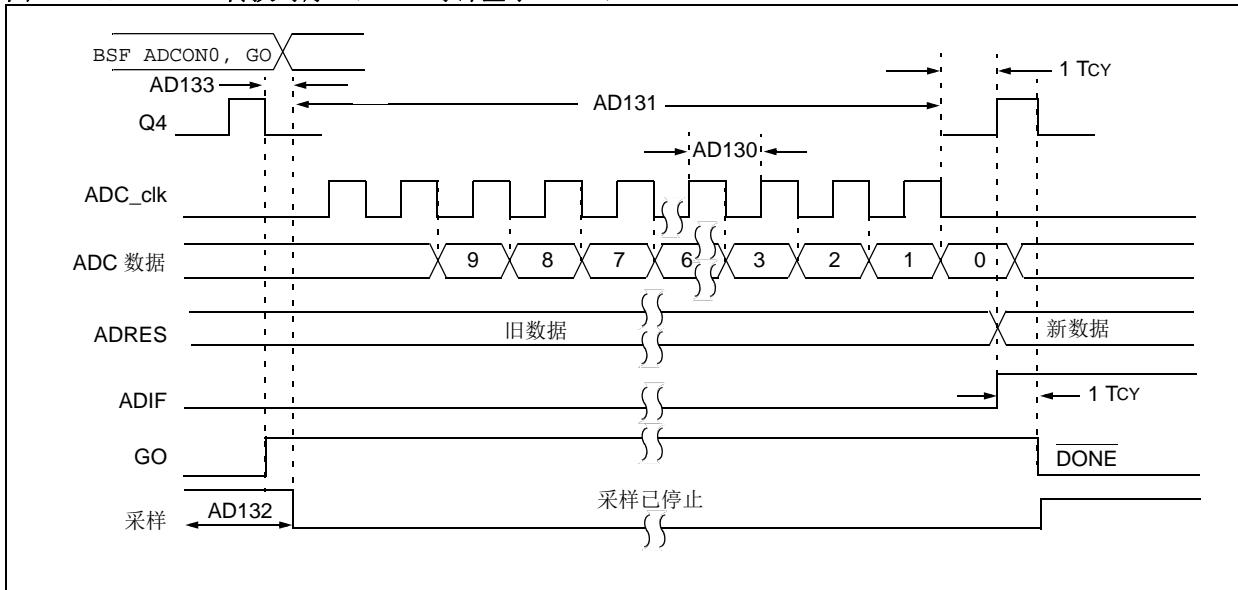
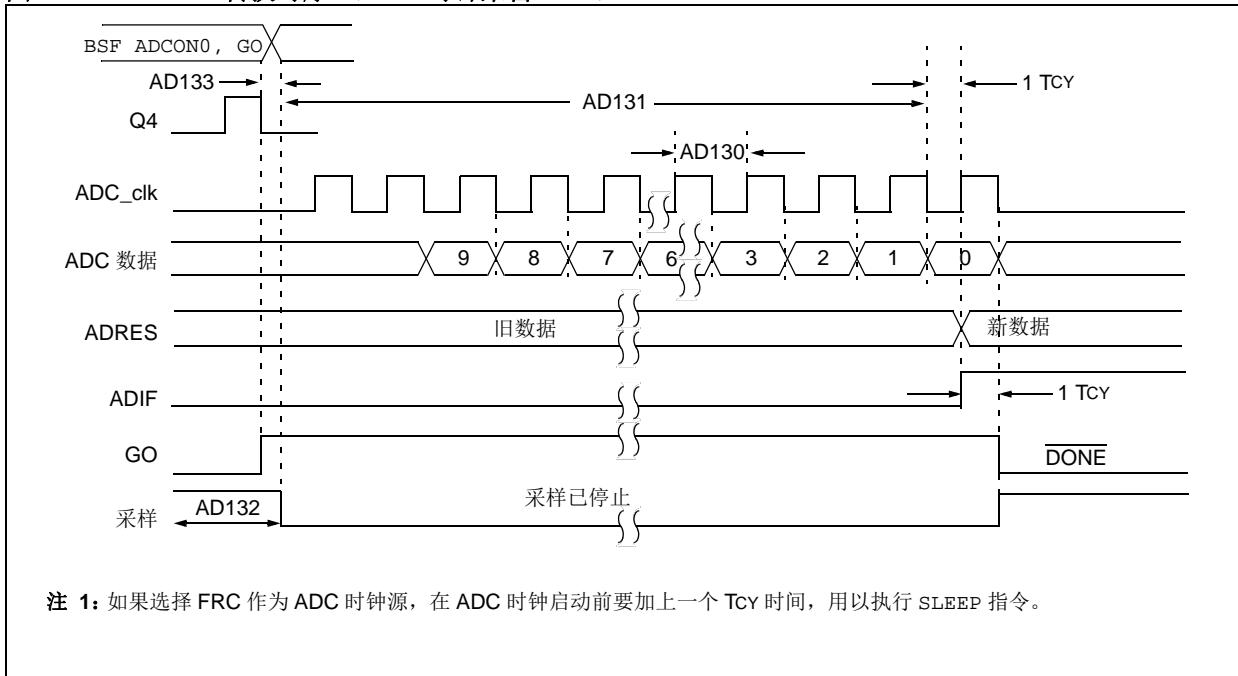


图 27-12: ADC 转换时序 (ADC 时钟来自 FRC)



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。

表 27-14: 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 ADCS<2:0> = x11 (ADC FRC 模式)
		ADC 内部 FRC 振荡器周期 (TFRC)	1.0	2.0	6.0	μs	
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 1/2 TAD + 1 TCY	—		基于 FOSC 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。

PIC16(L)F1574/5/8/9

表 27-15: 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 28.0 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。

表 27-16: 数模转换器 (DAC) 规范⁽¹⁾

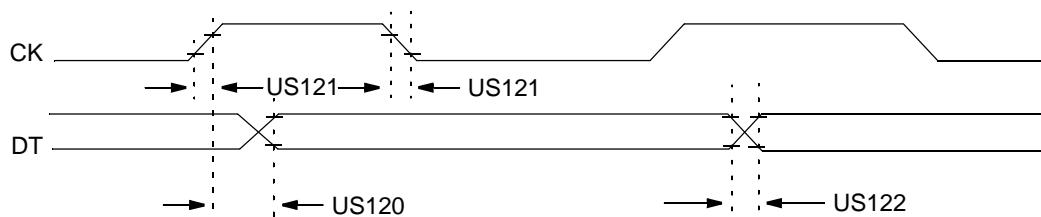
工作条件 (除非另外声明) VDD = 3.0V, TA = 25°C							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
DAC01*	CLSB	步长	—	VDD/32	—	V	
DAC02*	CACC	绝对精度	—	—	± 1/2	LSb	
DAC03*	CR	单位电阻值 (R)	—	5k	—	Ω	
DAC04*	Cst	稳定时间 ⁽²⁾	—	—	10	μs	

* 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 28.0 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。

图 27-13: USART 同步发送 (主 / 从) 时序



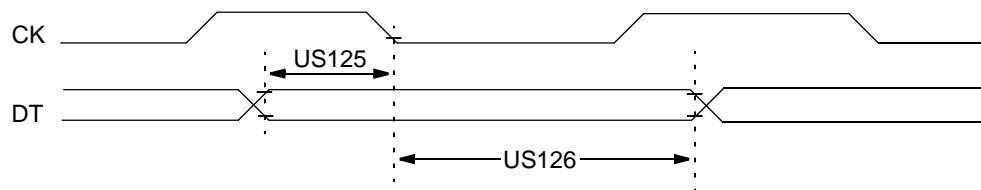
注: 负载条件请参见图 27-4。

表 27-17: USART 同步发送要求

标准工作条件 (除非另外声明)

参数编号	符号	特性	最小值	最大值	单位	条件
US120	TCKH2DTV	同步发送 (主模式和从模式) 时钟高电平到数据输出有效的时间	—	80	ns	3.0V ≤ VDD ≤ 5.5V
			—	100	ns	1.8V ≤ VDD ≤ 5.5V
US121	TCKRF	时钟输出上升时间和下降时间 (主模式)	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V
US122	TDTRF	数据输出上升时间和下降时间	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V

图 27-14: USART 同步接收 (主 / 从模式) 时序



注: 负载条件请参见图 27-4。

表 27-18: USART 同步接收要求

标准工作条件 (除非另外声明)

参数编号	符号	特性	最小值	最大值	单位	条件
US125	TDTV2CKL	同步接收 (主模式和从模式) CK ↓ 之前的数据保持时间 (DT 保持时间)	10	—	ns	
US126	TCKL2DTL	CK ↓ 之后的数据保持时间 (DT 保持时间)	15	—	ns	

28.0 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 V_{DD} 范围）。这些信息仅供参考，保证器件只在规定的范围内正常工作。

除非另外声明，否则所有图表均同时适用于 L 和 LF 器件。

注：以下图表为基于有限数量样片的统计结果，仅供参考。此处列出的特性未经测试，不做任何担保。一些图表中列出的数据可能超出规定的工作范围（例如，超出了规定的电源范围），因此不在担保范围内。

“典型值”代表 25°C 时的分布的平均值。“最大值”或“最小值”分别代表（平均值 +3σ）或（平均值 -3σ）；其中， σ 是每个温度范围内的标准差。

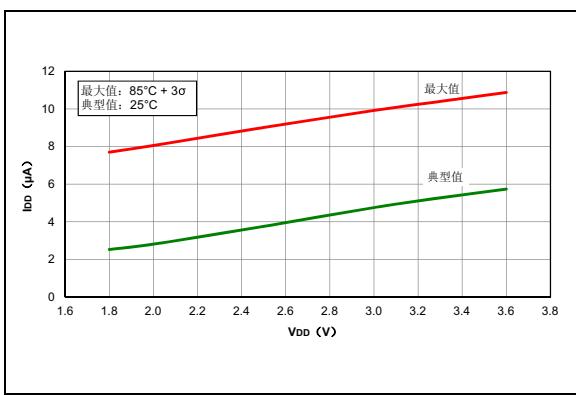


图28-1: IDD , EC 振荡器, 低功耗模式,
 $F_{OSC} = 32 \text{ kHz}$, 仅限 PIC16LF1574/5/8/9

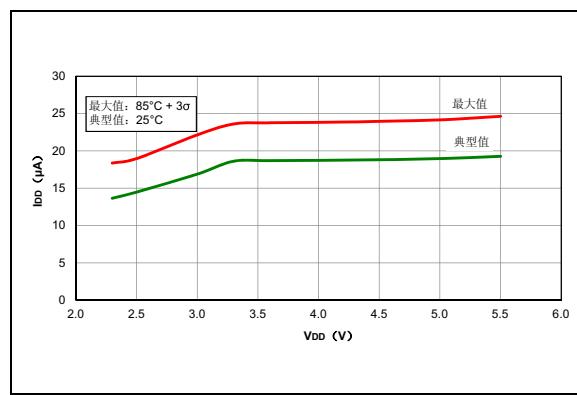


图28-2: IDD , EC 振荡器, 低功耗模式,
 $F_{OSC} = 32 \text{ kHz}$, 仅限 PIC16F1574/5/8/9

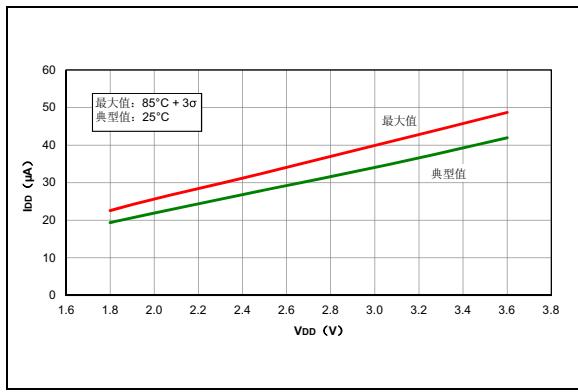


图28-3: IDD , EC 振荡器, 低功耗模式,
 $F_{OSC} = 500 \text{ kHz}$, 仅限 PIC16LF1574/5/8/9

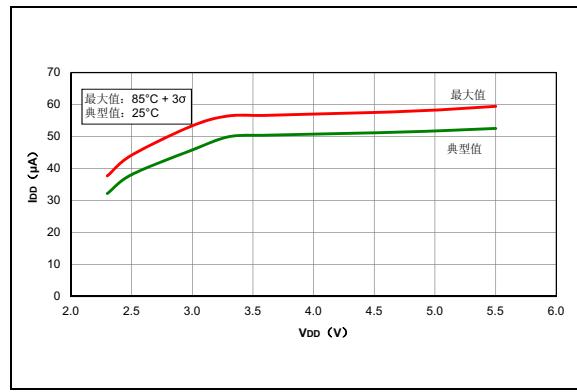


图28-4: IDD , EC 振荡器, 低功耗模式,
 $F_{OSC} = 500 \text{ kHz}$, 仅限 PIC16F1574/5/8/9

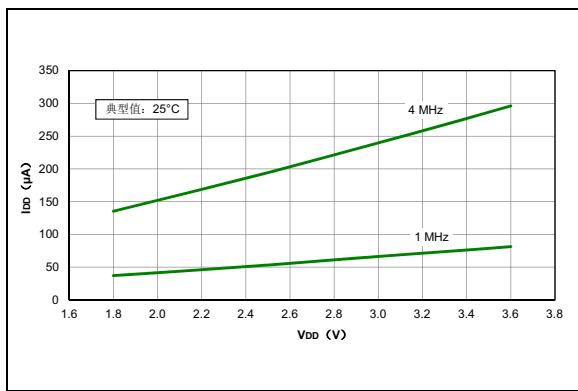


图28-5: IDD 典型值, EC 振荡器, 中等功
耗模式, 仅限 PIC16LF1574/5/8/9

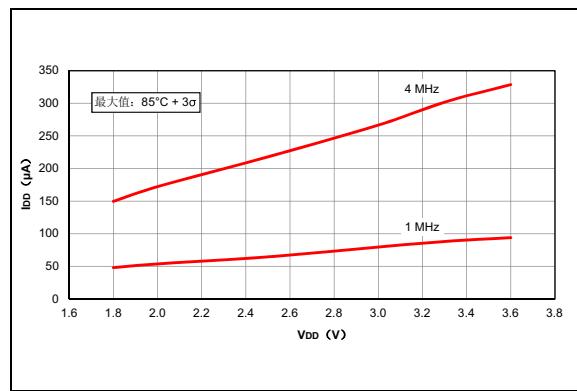


图28-6: IDD 最大值, EC 振荡器, 中等功
耗模式, 仅限 PIC16LF1574/5/8/9

PIC16(L)F1574/5/8/9

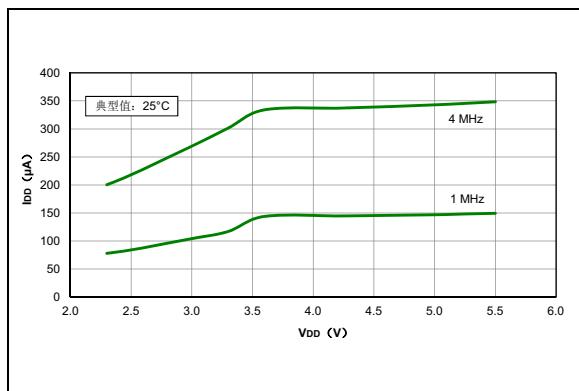


图28-7: IDD 典型值, EC 振荡器, 中等功耗模式, 仅限 PIC16F1574/5/8/9

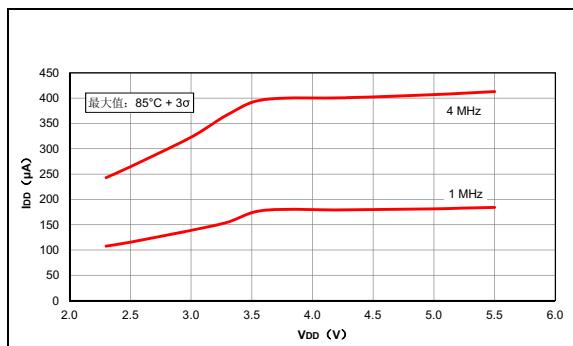


图28-8: IDD 最大值, EC 振荡器, 中等功耗模式, 仅限 PIC16F1574/5/8/9

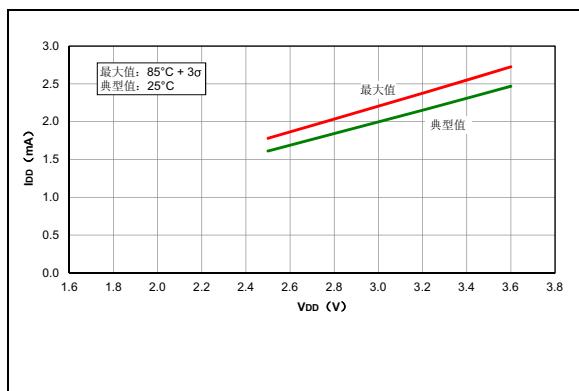


图28-9: IDD 典型值, EC 振荡器, 高功耗模式, FOSC = 32 kHz, 仅限 PIC16LF1574/5/8/9

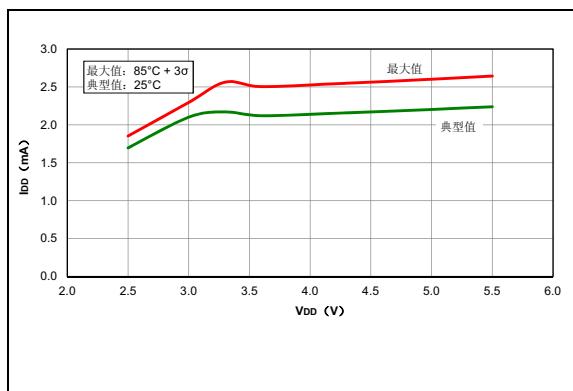


图28-10: IDD 典型值, EC 振荡器, 高功耗模式, FOSC = 32 kHz, 仅限 PIC16F1574/5/8/9

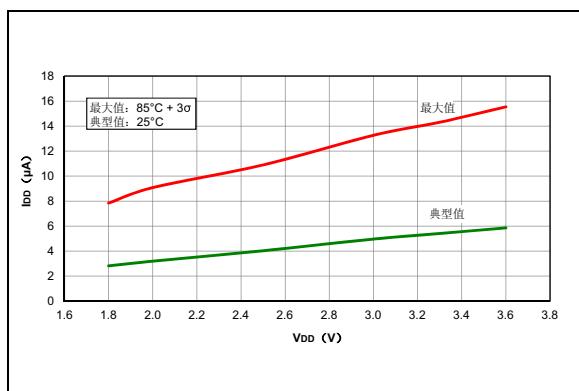


图28-11: IDD, LFINTOSC 模式, FOSC = 31 kHz, 仅限 PIC16LF1574/5/8/9

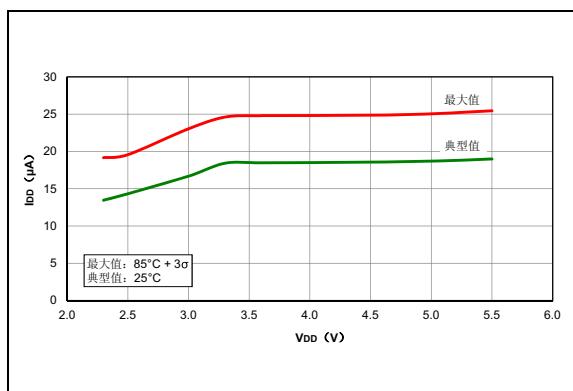


图28-12: IDD, LFINTOSC 模式, FOSC = 31 kHz, 仅限 PIC16F1574/5/8/9

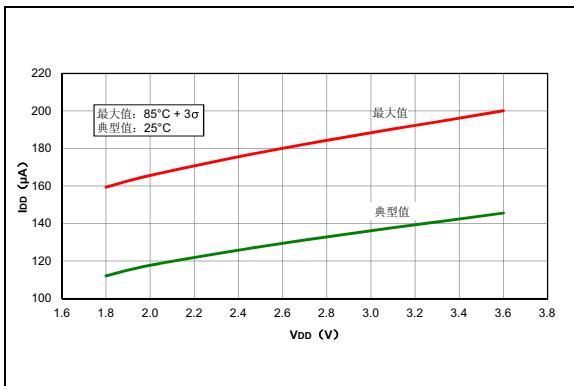


图28-13: IDD , MFINTOSC 模式,
 $FOSC = 500 \text{ kHz}$, 仅限 PIC16LF1574/5/8/9

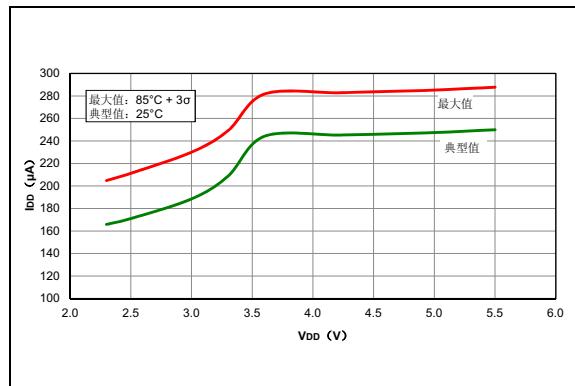


图28-14: IDD , MFINTOSC 模式,
 $FOSC = 500 \text{ kHz}$, 仅限 PIC16F1574/5/8/9

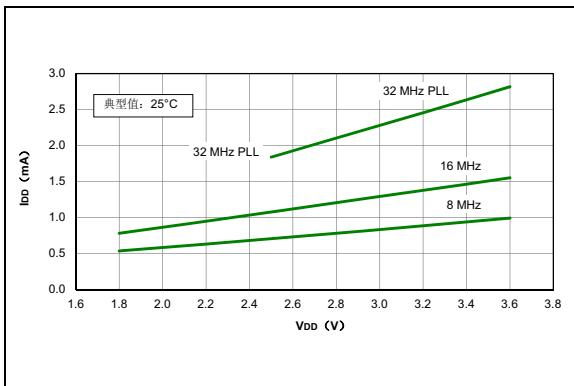


图28-15: IDD 典型值, HFINTOSC 模式,
仅限 PIC16LF1574/5/8/9

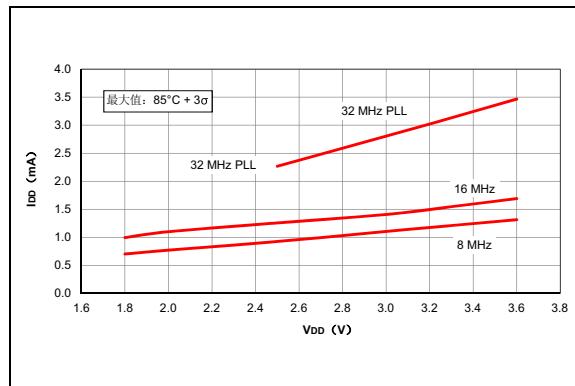


图28-16: IDD 最大值, HFINTOSC 模式,
仅限 PIC16LF1574/5/8/9

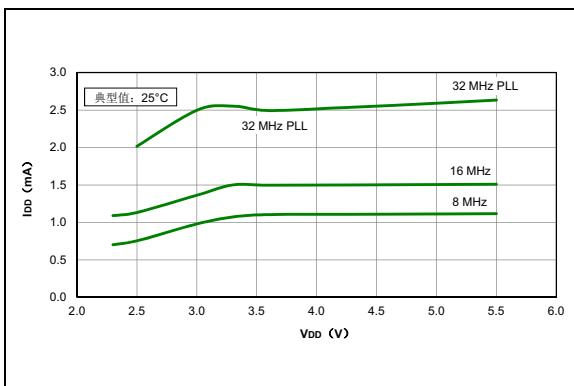


图28-17: IDD 典型值, HFINTOSC 模式,
仅限 PIC16F1574/5/8/9

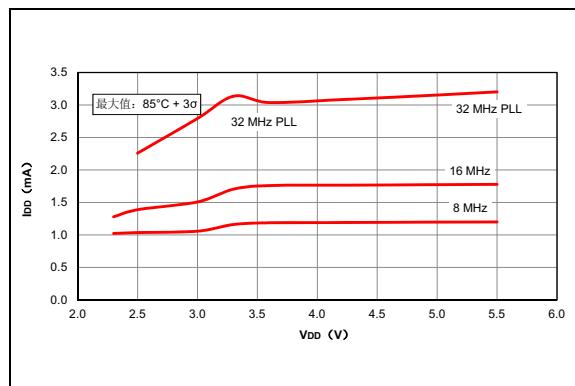


图28-18: IDD 最大值, HFINTOSC 模式,
仅限 PIC16F1574/5/8/9

PIC16(L)F1574/5/8/9

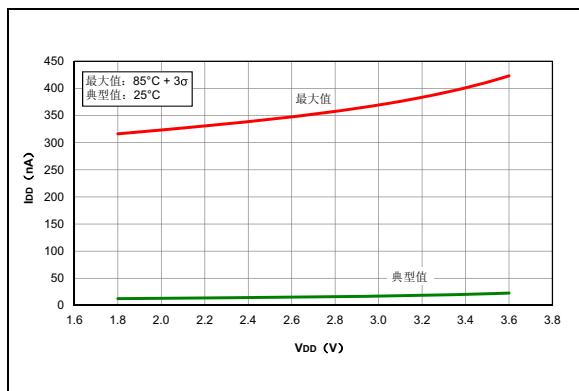


图28-19: IPD 基本电流, 低功耗休眠模式,
仅限 PIC16LF1574/5/8/9

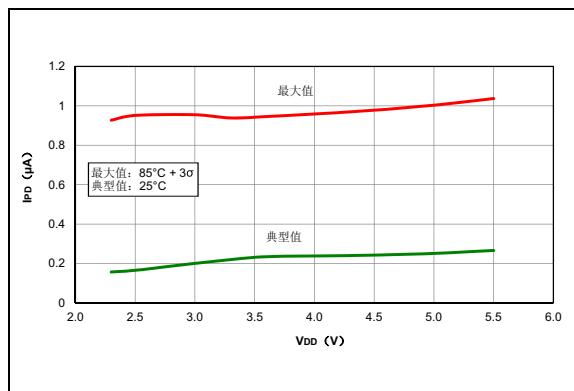


图28-20: IPD 基本电流, 低功耗休眠模式
($V_{REGPM} = 1$), 仅限 PIC16F1574/5/8/9

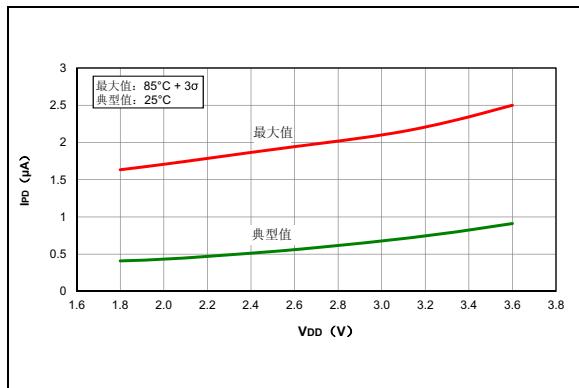


图28-21: IPD, 看门狗定时器 (WDT),
仅限 PIC16LF1574/5/8/9

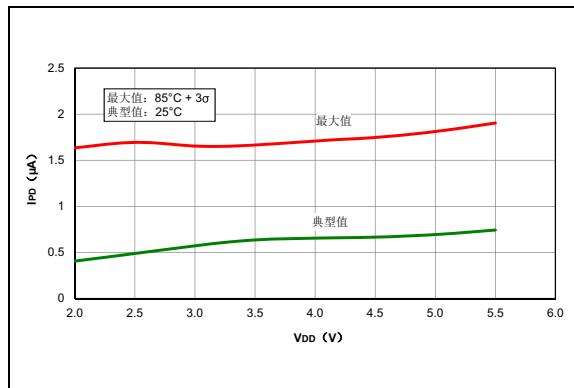


图28-22: IPD, 看门狗定时器 (WDT),
仅限 PIC16F1574/5/8/9

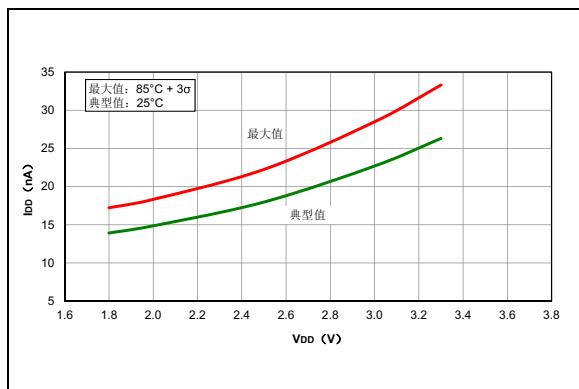


图28-23: IPD, 固定参考电压 (FVR),
仅限 PIC16LF1574/5/8/9

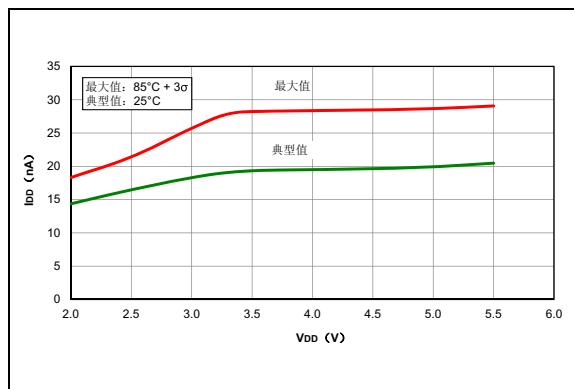


图28-24: IPD, 固定参考电压 (FVR),
仅限 PIC16F1574/5/8/9

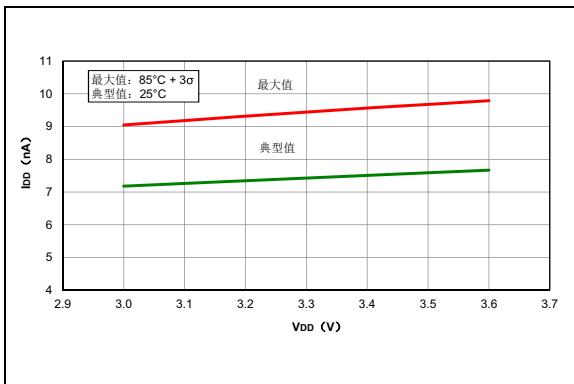


图28-25: IPD, 欠压复位 (BOR),
BORV = 1, 仅限 PIC16LF1574/5/8/9

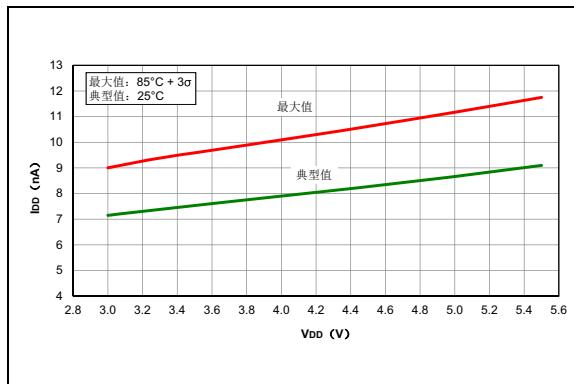


图28-26: IPD, 欠压复位 (BOR),
BORV = 1, 仅限 PIC16F1574/5/8/9

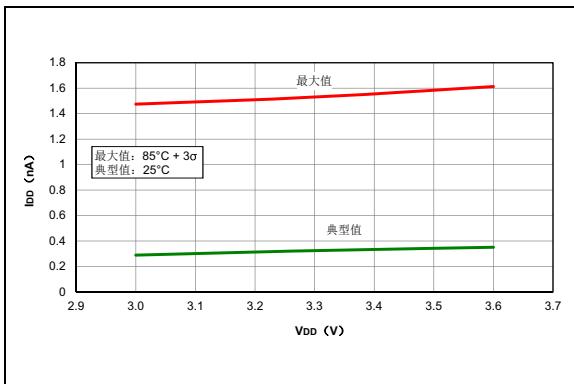


图28-27: IPD, 低功耗欠压复位
(LPBOR = 0), 仅限 PIC16LF1574/5/8/9

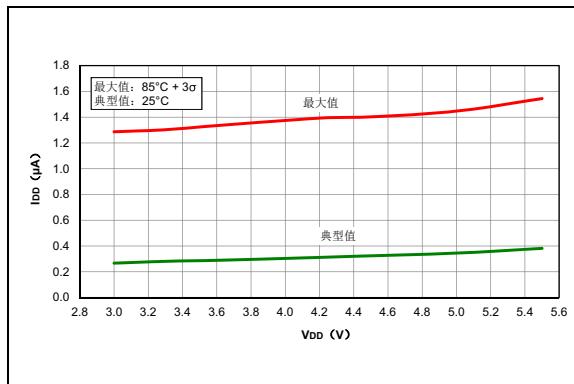


图28-28: IPD, 低功耗欠压复位
(LPBOR = 0), 仅限 PIC16F1574/5/8/9

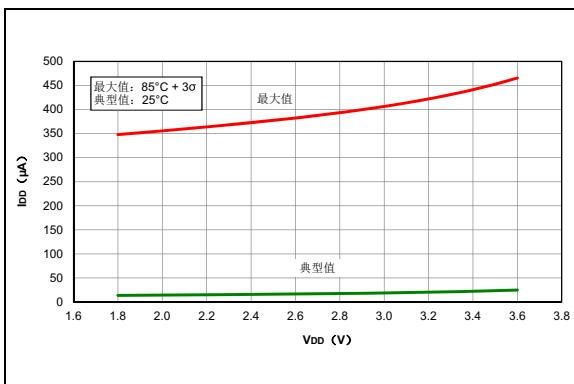


图28-29: IPD, ADC 同相, 仅限
PIC16LF1574/5/8/9

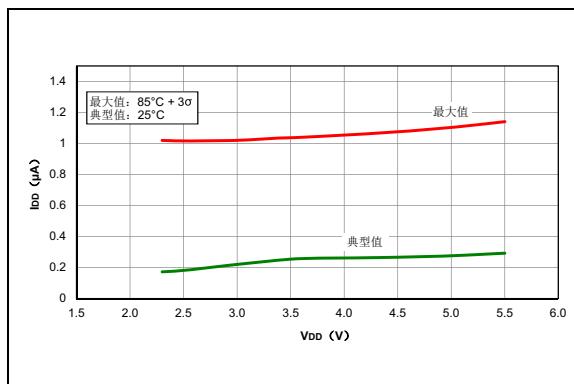


图28-30: IPD, ADC 同相, 仅限
PIC16F1574/5/8/9

PIC16(L)F1574/5/8/9

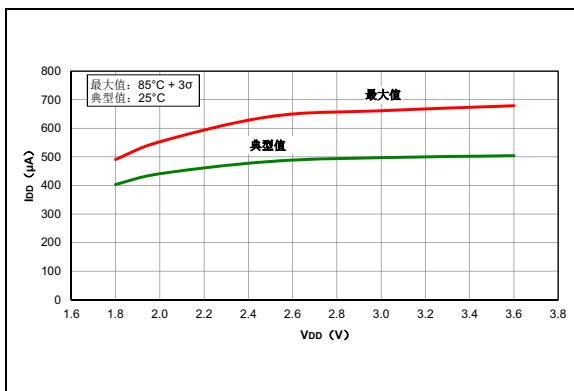


图 28-31: IPD, 比较器, 低功耗模式
(CxSP = 0), 仅限 PIC16LF1574/5/8/9

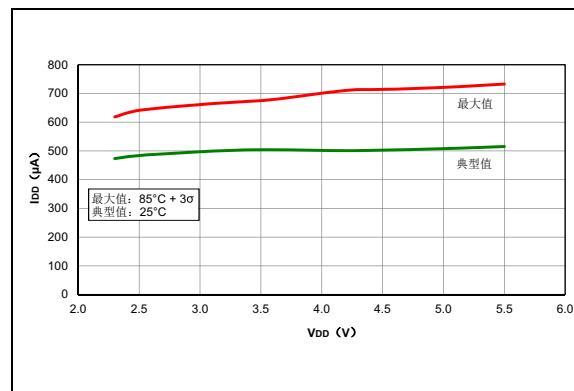


图 28-32: IPD, 比较器, 低功耗模式
(CxSP = 0), 仅限 PIC16F1574/5/8/9

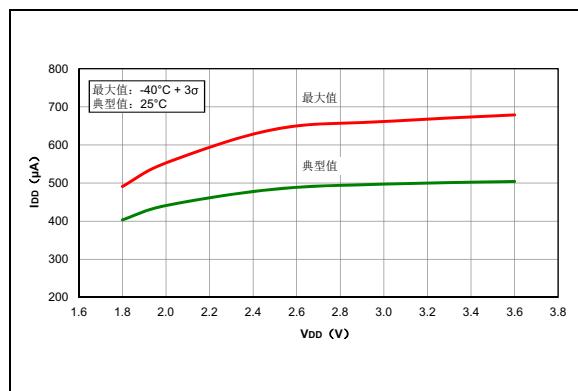


图 28-33: IPD, 比较器, 正常功耗模式
(CxSP = 1), 仅限 PIC16LF1574/5/8/9

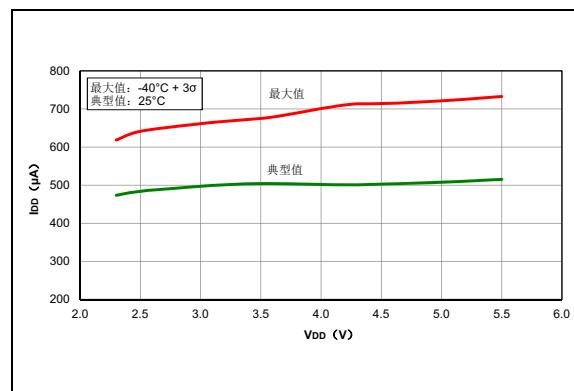


图 28-34: IPD, 比较器, 正常功耗模式
(CxSP = 1), 仅限 PIC16F1574/5/8/9

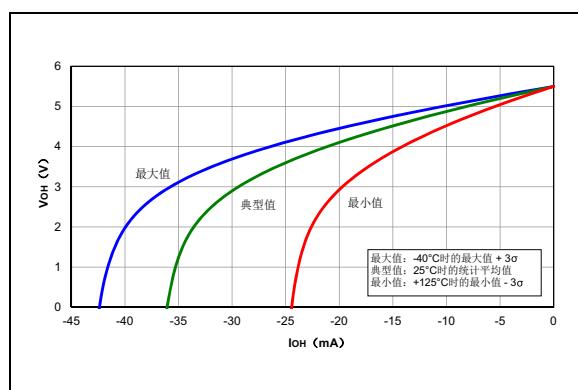


图 28-35: 整个温度范围内的 VOH—IOH 曲线, VDD = 5.5V, 仅限 PIC16F1574/5/8/9

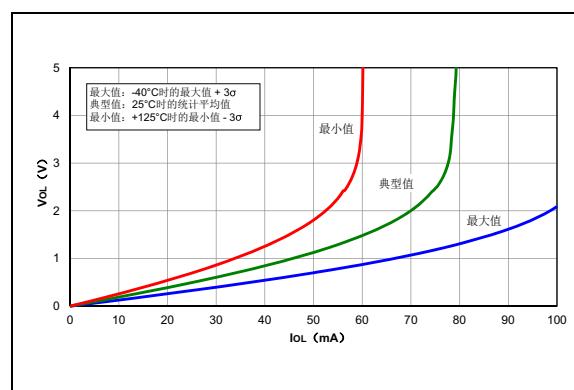


图 28-36: 整个温度范围内的 VOL—IOL 曲线, VDD = 5.5V, 仅限 PIC16F1574/5/8/9

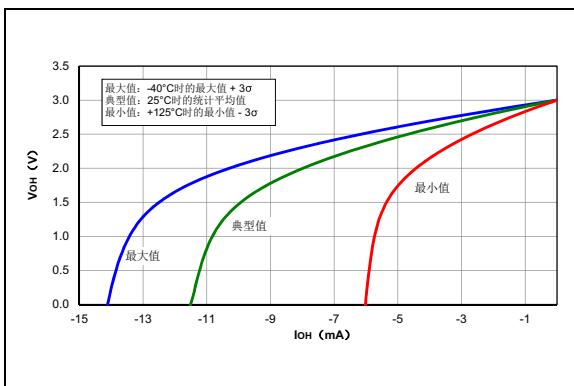


图28-37：整个温度范围内的 V_{OH} — I_{OH} 曲线， $V_{DD} = 3.0V$

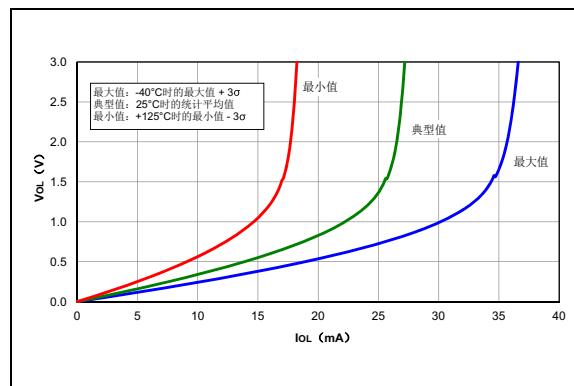


图28-38：整个温度范围内的 V_{OL} — I_{OL} 曲线， $V_{DD} = 3.0V$

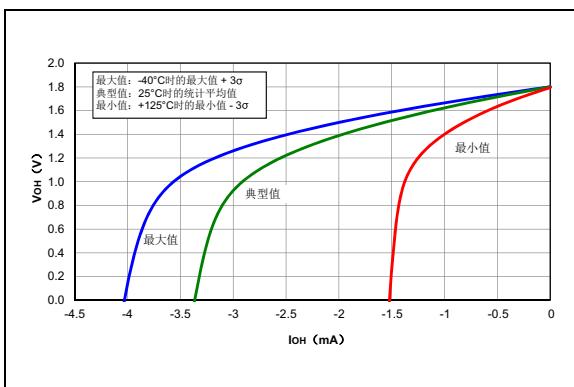


图28-39：整个温度范围内的 V_{OH} — I_{OH} 曲线， $V_{DD} = 1.8V$ ，仅限PIC16LF1574/5/8/9

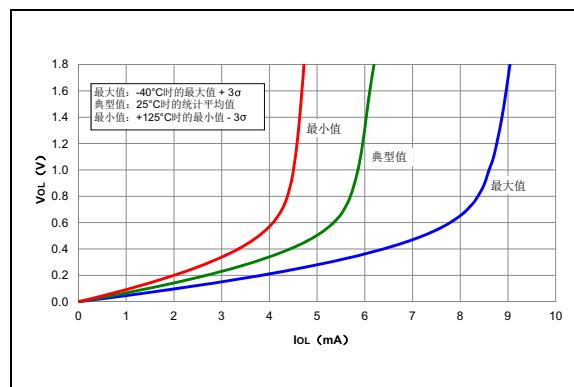


图28-40：整个温度范围内的 V_{OL} — I_{OL} 曲线， $V_{DD} = 1.8V$ ，仅限PIC16LF1574/5/8/9

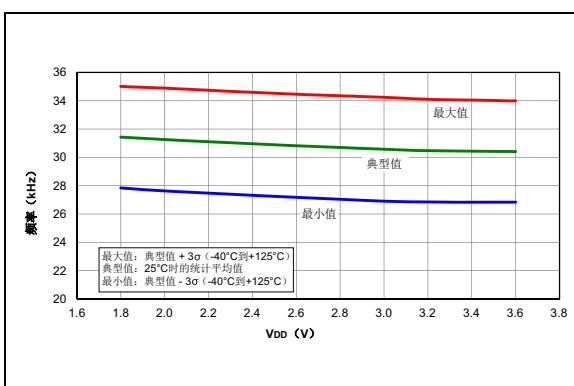


图28-41：整个 V_{DD} 和温度范围内的LFINTOSC频率，仅限PIC16LF1574/5/8/9

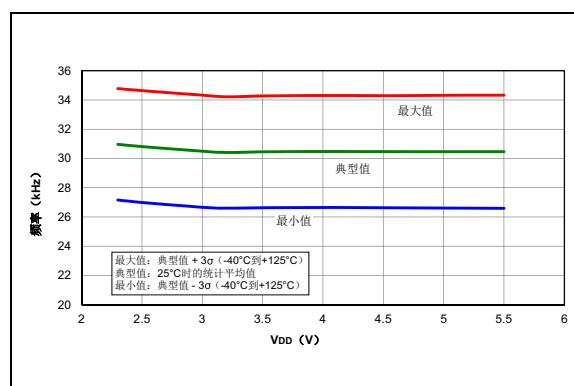


图28-42：整个 V_{DD} 和温度范围内的LFINTOSC频率，仅限PIC16F1574/5/8/9

PIC16(L)F1574/5/8/9

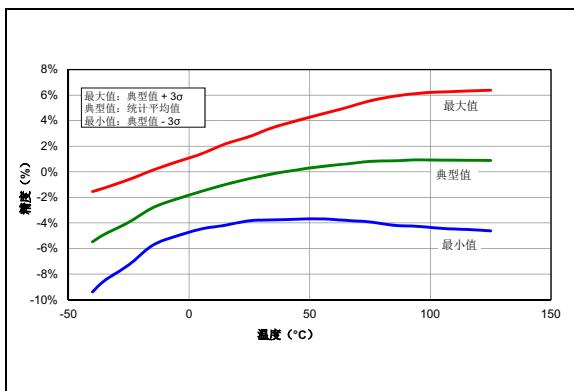


图 28-43: 整个温度范围内的 HFINTOSC 精度, $VDD = 1.8V$, 仅限 LF 器件

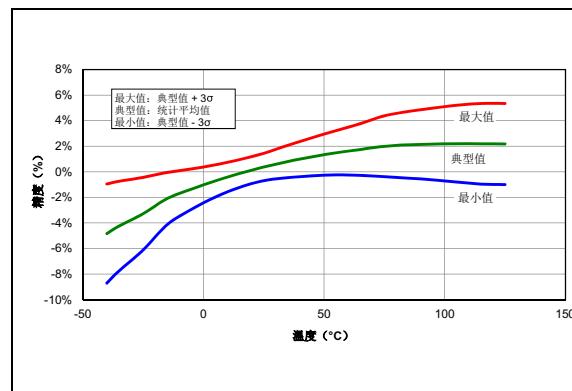


图 28-44: 整个温度范围内的 HFINTOSC 精度, $2.3V \leq VDD \leq 5.5V$

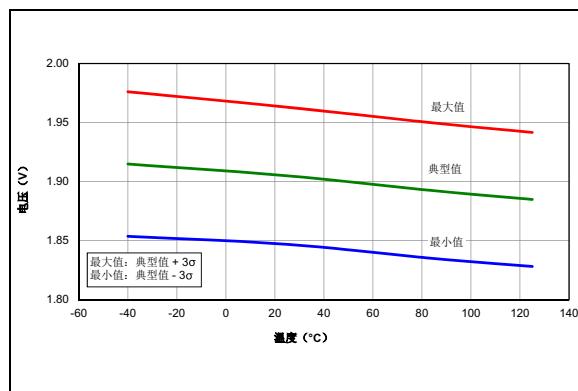


图 28-45: 欠压复位电压, $BORV = 1$, 仅限 PIC16LF1574/5/8/9

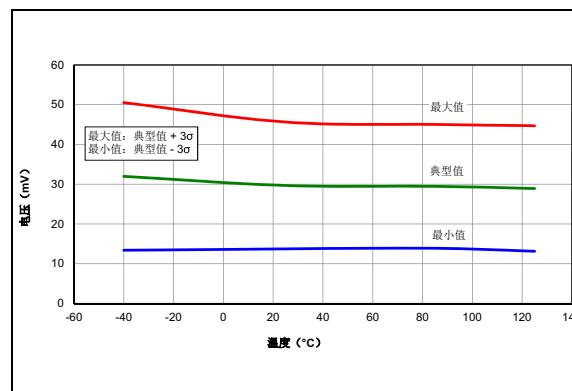


图 28-46: 欠压复位滞后, $BORV = 1$, 仅限 PIC16LF1574/5/8/9

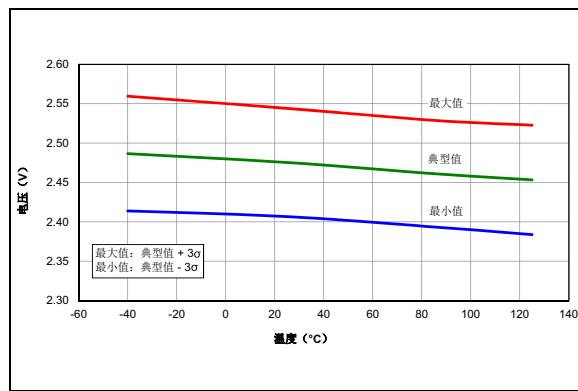


图 28-47: 欠压复位电压, $BORV = 1$, 仅限 PIC16F1574/5/8/9

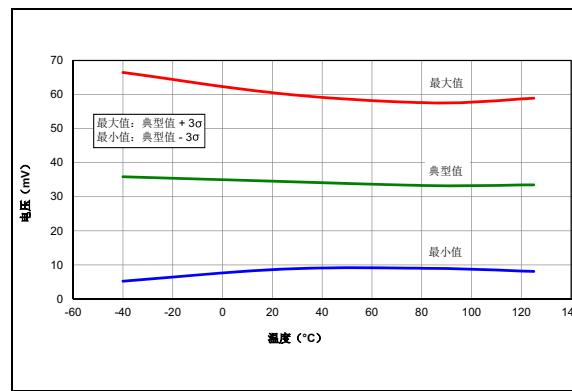


图 28-48: 欠压复位滞后, $BORV = 1$, 仅限 PIC16F1574/5/8/9

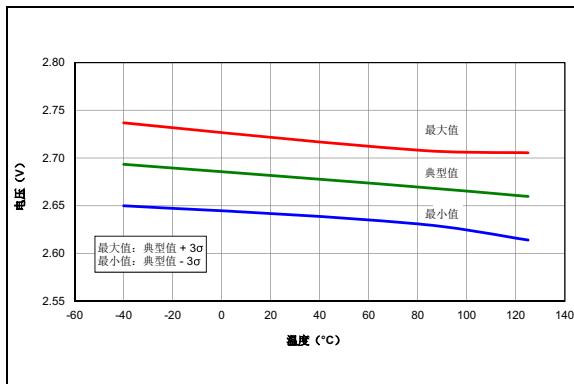


图28-49: 欠压复位电压, $BORV = 0$

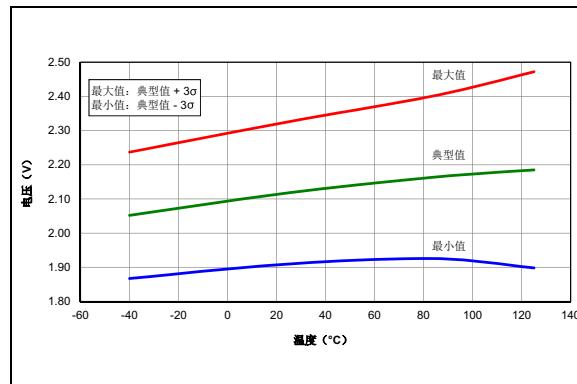


图28-50: 低功耗欠压复位电压, $LPBOR = 0$

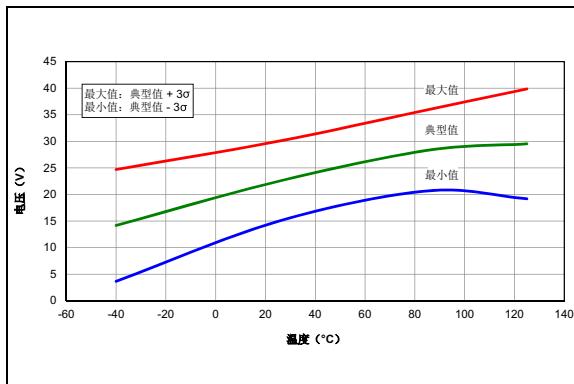


图28-51: 低功耗欠压复位滞后, $LPBOR = 0$

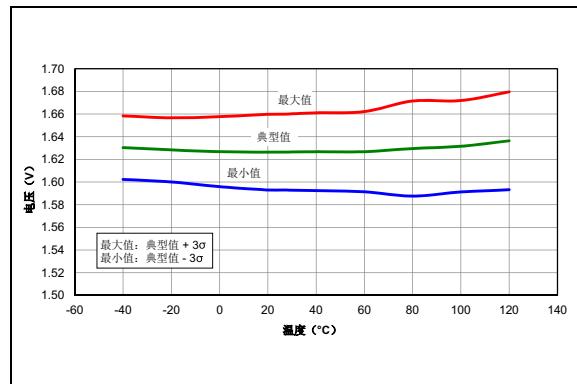


图28-52: POR 释放电压

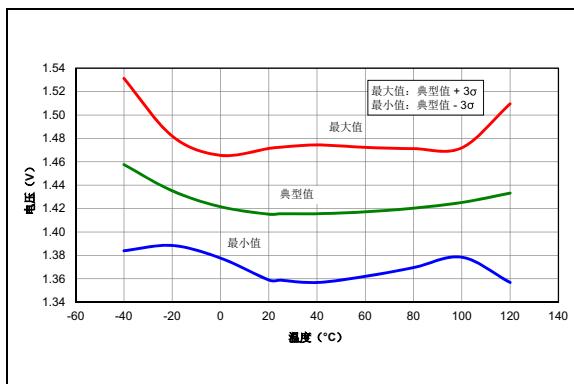


图28-53: POR 重新激活电压, 仅限
PIC16F1574/5/8/9

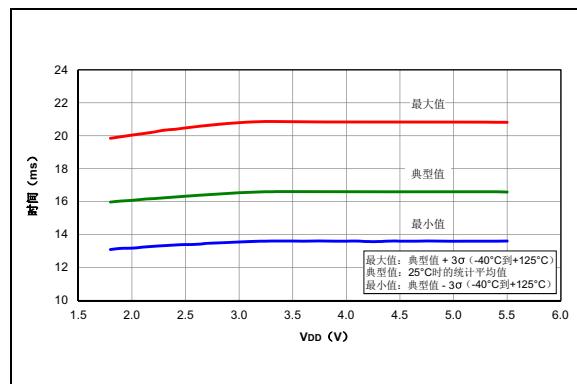


图28-54: WDT 超时周期

PIC16(L)F1574/5/8/9

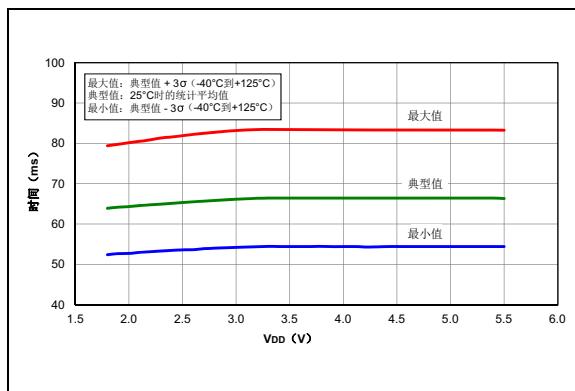


图 28-55: PWRT 周期

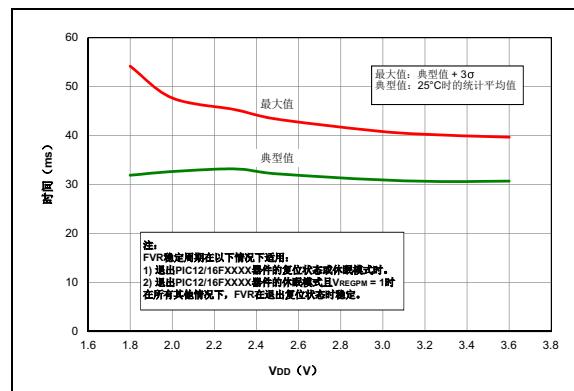


图 28-56: FVR 稳定周期

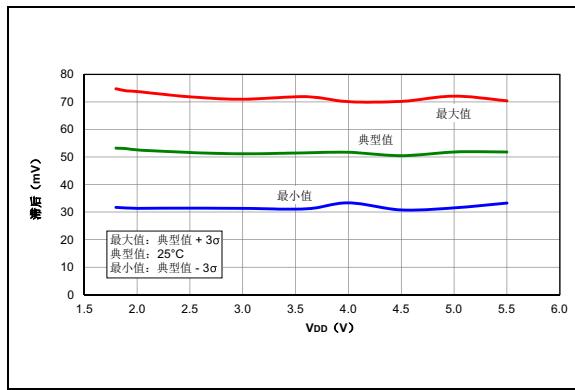


图 28-57: 比较器滞后, 正常功耗模式
(CxSP = 1, CxHYS = 1)

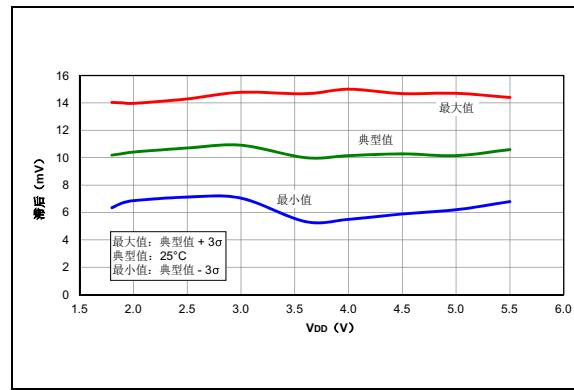


图 28-58: 比较器滞后, 低功耗模式
(CxSP = 0, CxHYS = 1)

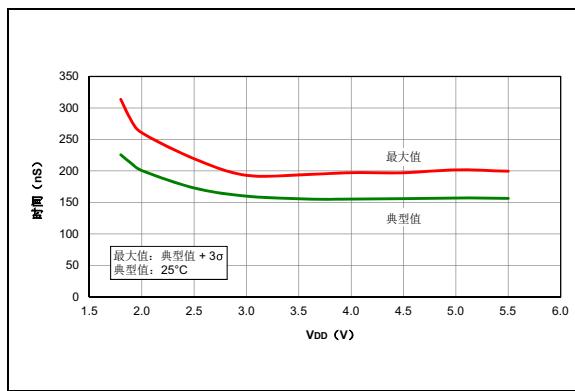


图 28-59: 比较器响应时间, 正常功耗模式
(CxSP = 1)

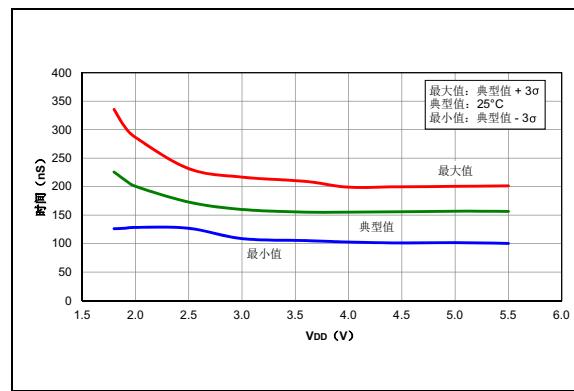


图 28-60: 整个温度范围内的比较器响应时间, 正常功耗模式 (CxSP = 1)

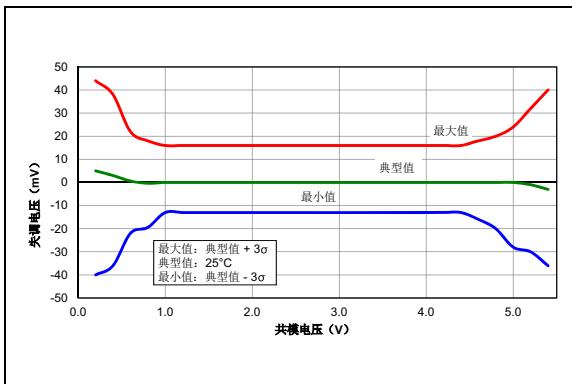


图28-61: 25°C 下的比较器输入, 正常功耗模式 ($\text{CxSP} = 1$)

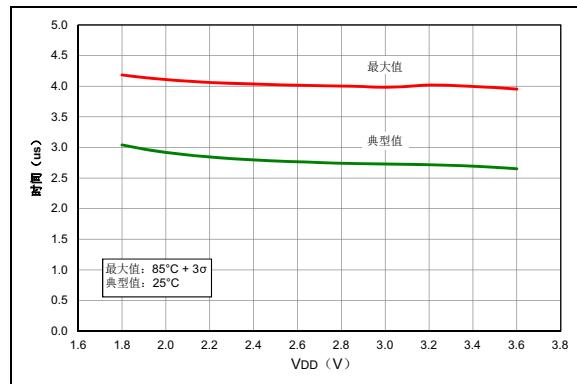


图28-62: 休眠模式, 唤醒周期与 HFINTOSC 源, 仅限 LF 器件

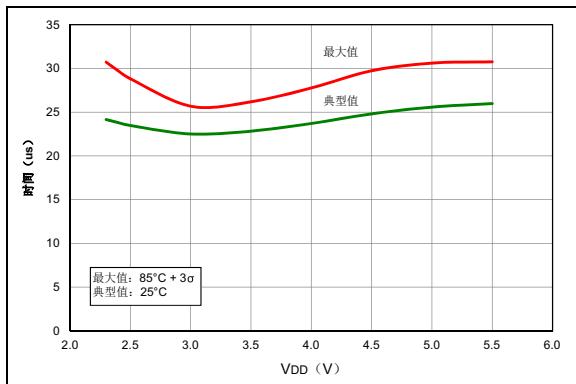


图28-63: 低功耗休眠模式, 唤醒周期与 HFINTOSC 源, $\text{VREGPM} = 1$, 仅限 F 器件

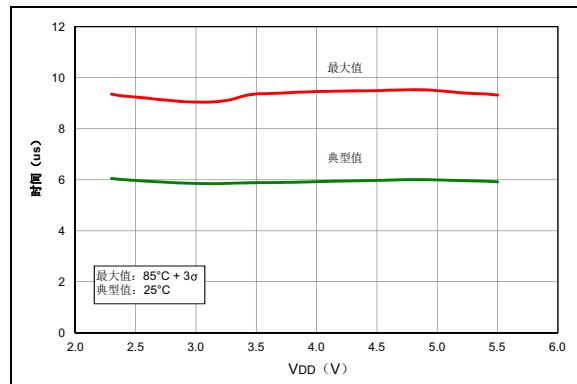


图28-64: 休眠模式, 唤醒周期与 HFINTOSC 源, $\text{VREGPM} = 0$, 仅限 F 器件

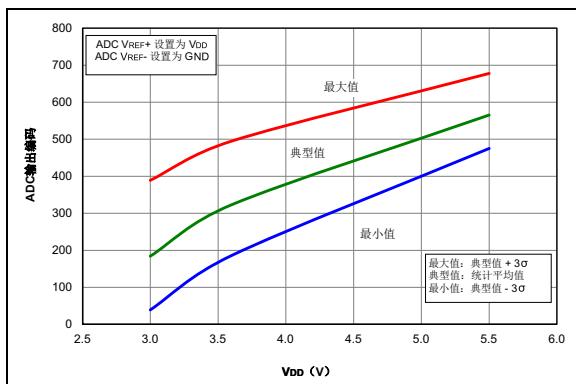


图28-65: 温度指示器初始偏移, 高电压范围, 温度 = 20°C , 仅限 F 器件

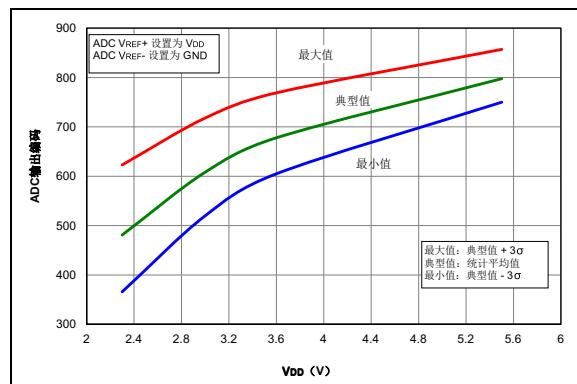


图28-66: 温度指示器初始偏移, 低电压范围, 温度 = 20°C , 仅限 F 器件

PIC16(L)F1574/5/8/9

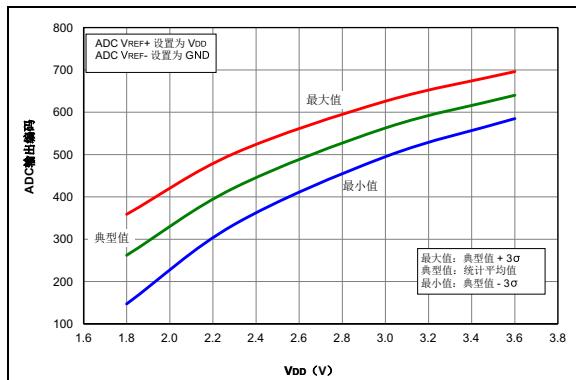


图 28-67: 温度指示器初始偏移, 低电压范围, 温度 = 20°C, 仅限 LF 器件

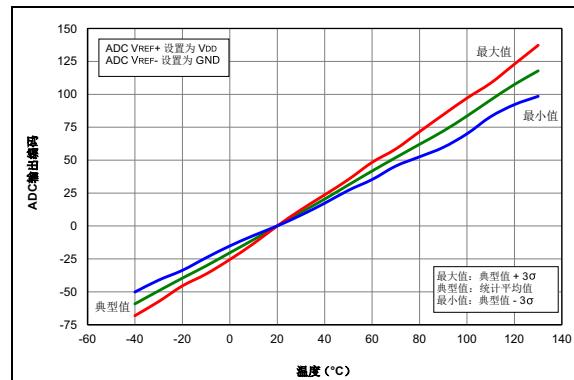


图 28-68: 规范化到 20°C 的温度指示器斜率, 高电压范围, VDD = 5.5V, 仅限 F 器件

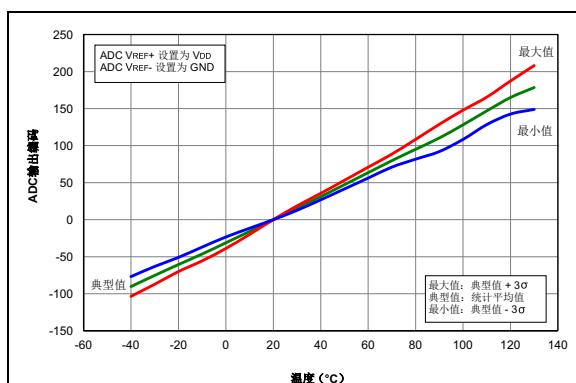


图 28-69: 规范化到 20°C 的温度指示器斜率, 高电压范围, VDD = 3.6V, 仅限 F 器件

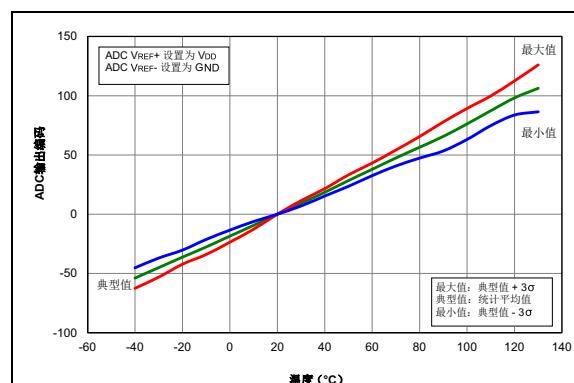


图 28-70: 规范化到 20°C 的温度指示器斜率, 低电压范围, VDD = 3.0V, 仅限 F 器件

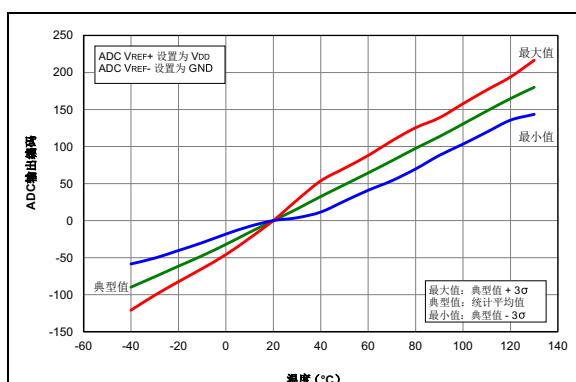


图 28-71: 规范化到 20°C 的温度指示器斜率, 低电压范围, VDD = 1.8V, 仅限 LF 器件

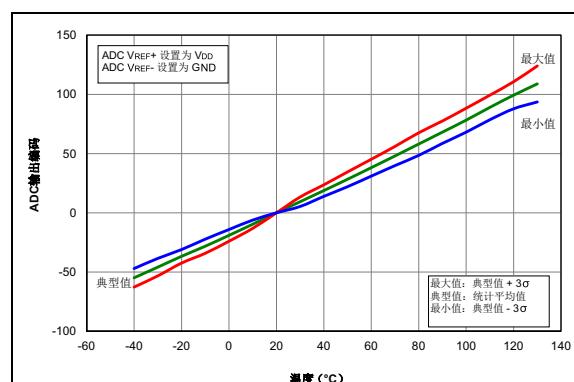


图 28-72: 规范化到 20°C 的温度指示器斜率, 低电压范围, VDD = 3.0V, 仅限 LF 器件

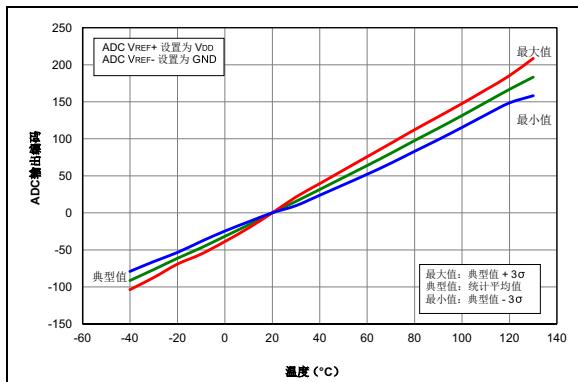


图28-73: 规范化到20°C 的温度指示器斜率, 高电压范围, V_{DD} = 3.6V, 仅限LF器件

29.0 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（Digital Signal Controller, 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

29.1 MPLAB X 集成开发环境软件

MPLAB X IDE 是用于 Microchip 及第三方软件和硬件开发工具的独立而统一的图形用户界面，在 Windows®、Linux 和 Mac OS® X 上运行。MPLAB X IDE 基于 NetBeans IDE，是一款全新的 IDE，带有大量免费软件组件和插件，用于高性能应用程序开发和调试。通过无缝的用户界面，您可以在工具之间切换，从软件模拟器升级到硬件调试和编程工具，这一切都变得如此简单。

MPLAB X IDE 提供全面的项目管理功能、可视调用图、可配置观察窗口和功能丰富的编辑器（包含代码补全功能和上下文菜单），对于新用户来说足够灵活和友好。MPLAB X IDE 支持对多个项目使用多种工具并实现同时调试，也适合有经验用户的需要。

功能丰富的编辑器：

- 彩色语法高亮显示
- 智能代码补全可在您输入时提出建议和提供提示
- 基于用户定义规则的自动代码格式化
- 即时解析

用户友好的可定制界面：

- 可完全定制的界面：工具栏、工具栏按钮、窗口和窗口位置等
- 调用图窗口

基于项目的工作区：

- 多个项目
- 多种工具
- 多个配置
- 同时调试会话

文件历史记录和问题跟踪：

- 本地文件历史记录功能
- 内置支持 Bugzilla 问题跟踪器

29.2 MPLAB XC 编译器

MPLAB XC 编译器是完全的 ANSI C 编译器，适用于 Microchip 的所有 8 位、16 位和 32 位 MCU 和 DSC 器件。这些编译器可提供强大的集成功能和出众的代码优化能力，且使用方便。MPLAB XC 编译器在 Windows、Linux 或 Mac OS X 上运行。

为便于源代码调试，编译器提供了针对 MPLAB X IDE 优化的调试信息。

免费的 MPLAB XC 编译器版本支持所有器件和命令，没有时间或存储器限制，并对大多数应用程序提供足够的代码优化。

MPLAB XC 编译器包括汇编器、链接器和实用程序。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。MPLAB XC 编译器使用该汇编器来生成目标文件。此汇编器的突出特性包括：

- 支持整个器件指令集
- 支持定点数据和浮点数据
- 命令行接口
- 丰富的指令集
- 灵活的宏语言
- MPLAB X IDE 兼容性

29.3 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引用的 MAP 文件、包含源代码行和生成的机器码的绝对 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
- 通过用户定义的宏来简化汇编代码
- 可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

29.4 MPLINK 目标链接器 /MPLIB 目标库管理器

MPLINK 目标链接器可以组合由 MPASM 汇编器创建的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

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

29.5 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、PIC32 和 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来生成目标文件。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。此汇编器的突出特性包括：

- 支持整个器件指令集
- 支持定点数据和浮点数据
- 命令行接口
- 丰富的指令集
- 灵活的宏语言
- MPLAB X IDE 兼容性

29.6 MPLAB X SIM 软件模拟器

MPLAB X SIM 软件模拟器可通过在指令级对 PIC MCU 和 dsPIC DSC 进行模拟，从而在 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。

29.7 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 器件推出的下一代高速仿真器。结合 MPLAB X IDE 所具有的功能强大但易于使用的图形用户界面，它可对所有 8 位、16 位和 32 位 MCU 和 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 连接，并利用与在线调试器系统兼容的连接器（RJ-11）或新型抗噪声、高速低压差分信号（Low-Voltage Differential Signal, LVDS）互连电缆（CAT5）与目标板连接。

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、运行时变量监视、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。

29.8 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 闪存 DSC 和 MCU 器件。结合 MPLAB IDE 所具有的功能强大但易于使用的图形用户界面，它可对 PIC 闪存单片机和 dsPIC DSC 进行调试和编程。

MPLAB ICD 3 在线调试器的探针通过高速 USB 2.0 接口与设计工程师的 PC 连接，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器（RJ-11）与目标板连接。MPLAB ICD 3 支持所有 MPLAB ICD 2 转接头。

29.9 PICkit 3 在线调试器 / 编程器

通过 MPLAB IDE 功能强大的图形用户界面，可使用 MPLAB PICkit 3 对 PIC 和 dsPIC 闪存单片机进行调试和编程，且价位较低。MPLAB PICkit 3 通过全速 USB 接口与设计工程师的 PC 连接，并利用 Microchip 调试（RJ-11）连接器（与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容）与目标板连接。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（ICSP™）。

29.10 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保证可靠性最高。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 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。

29.11 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，该工具包包含体验指定器件所需的一切。通常包括单个应用程序及调试功能，都在一块板上。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页（www.microchip.com）。

29.12 第三方开发工具

Microchip 还提供了许多来自第三方供应商的工具。这些工具经过精心挑选，可以提供良好的价值和独特的功能。

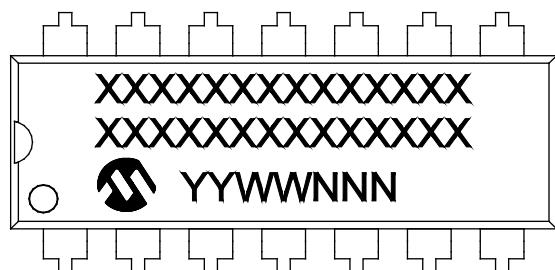
- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 IPLogika® 等公司提供的嵌入式以太网解决方案

PIC16(L)F1574/5/8/9

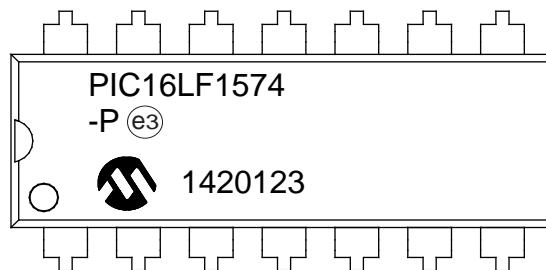
30.0 封装信息

30.1 封装标识信息

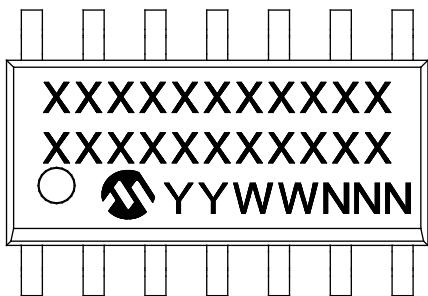
14 引脚 PDIP (300 mil)



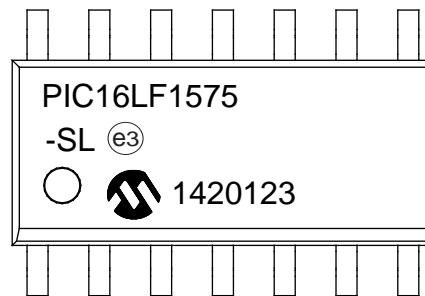
示例



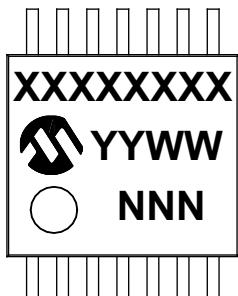
14 引脚 SOIC (3.90 mm)



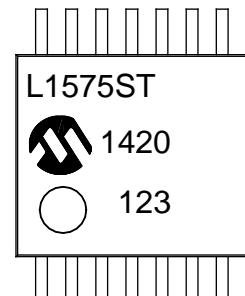
示例



14 引脚 TSSOP (4.4 mm)



示例



图注:

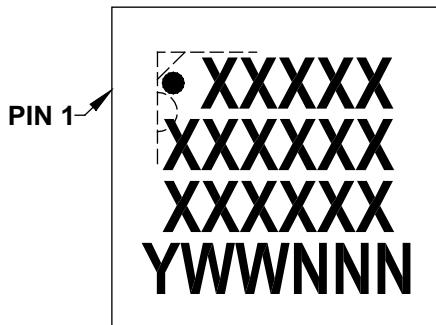
- | | |
|--------|--------------------------------------|
| XX...X | 客户指定信息 |
| Y | 年份代码（日历年的最后一位数字） |
| YY | 年份代码（日历年的最后两位数字） |
| WW | 星期代码（一月一日的星期代码为“01”） |
| NNN | 以字母数字排序的追踪代码 |
| (e3) | 雾锡（Matte Tin, Sn）的 JEDEC® 无铅标志 |
| * | 表示无铅封装。JEDEC 无铅标志 (e3) 标示于此种封装的外包装上。 |

注:

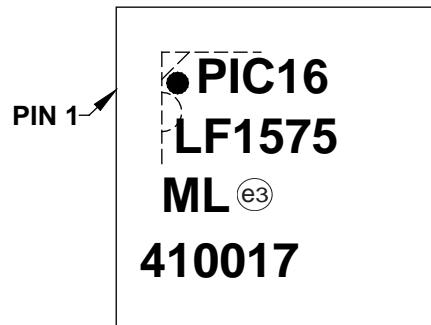
Microchip 部件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户指定信息的字符数。

封装标识信息（续）

16 引脚 UQFN (4x4x0.5 mm)



示例



图注:

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

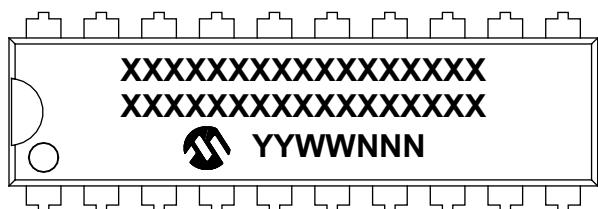
注:

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

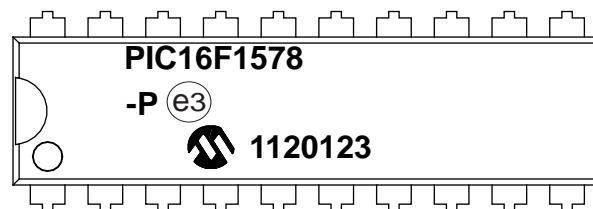
PIC16(L)F1574/5/8/9

封装标识信息（续）

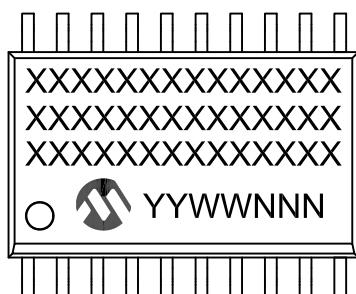
20 引脚 PDIP (300 mil)



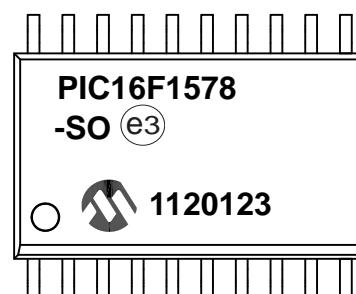
示例



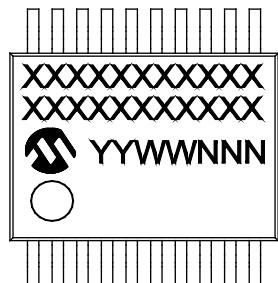
20 引脚 SOIC (7.50 mm)



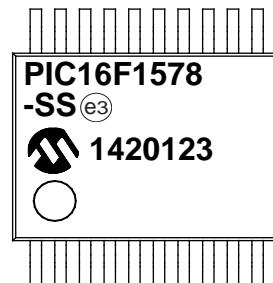
示例



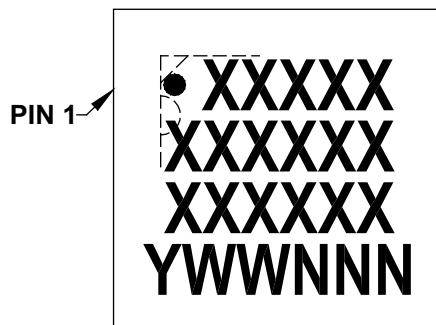
20 引脚 SSOP (5.30 mm)



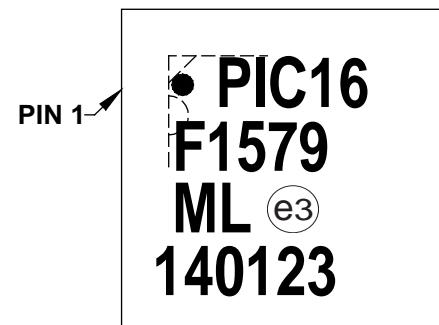
示例



20 引脚 UQFN (4x4x0.5 mm)



示例

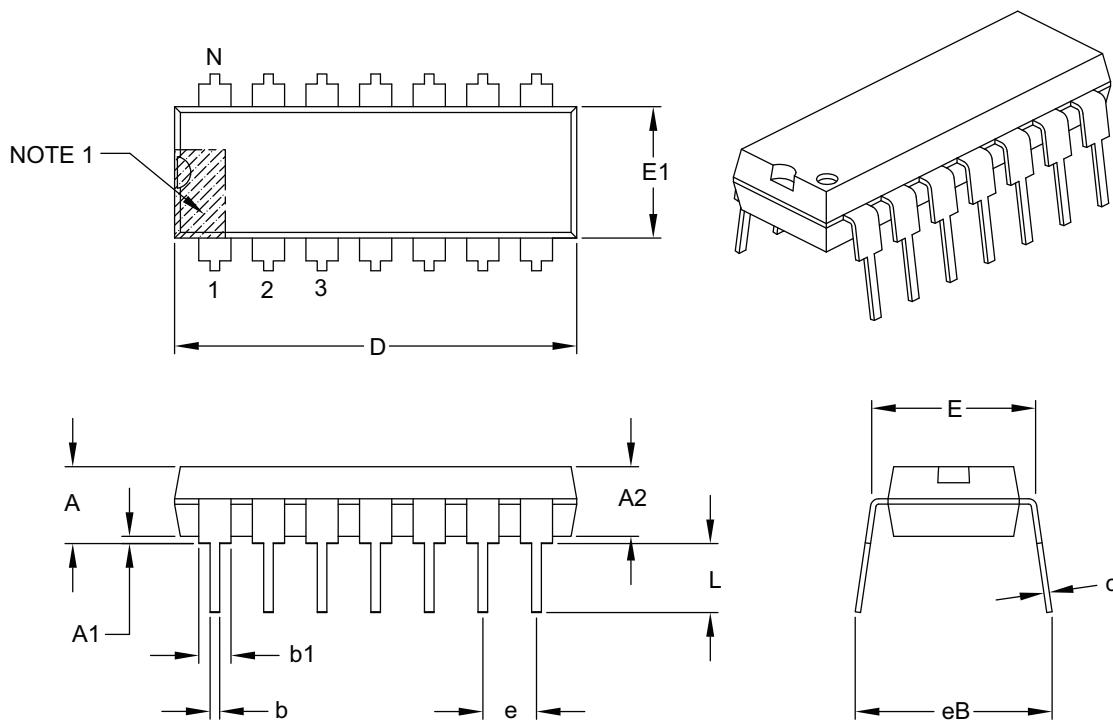


30.2 封装详细信息

以下部分将介绍各种封装的技术细节。

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

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



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

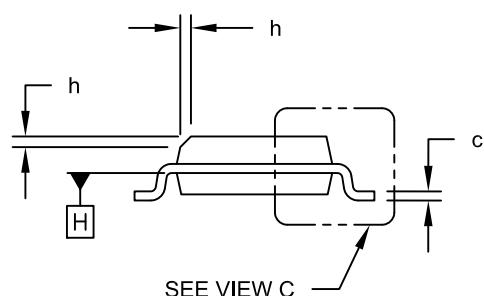
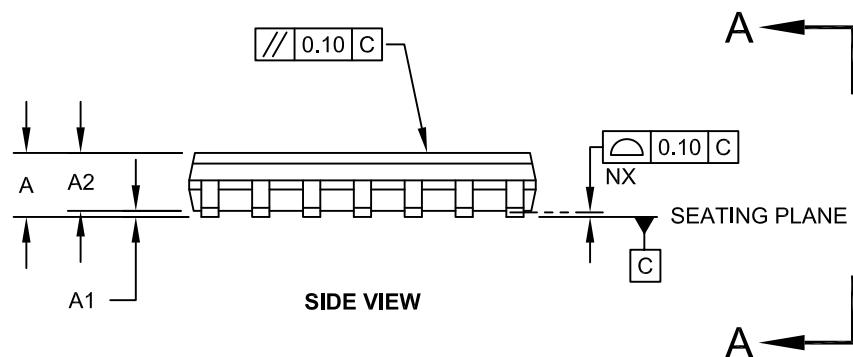
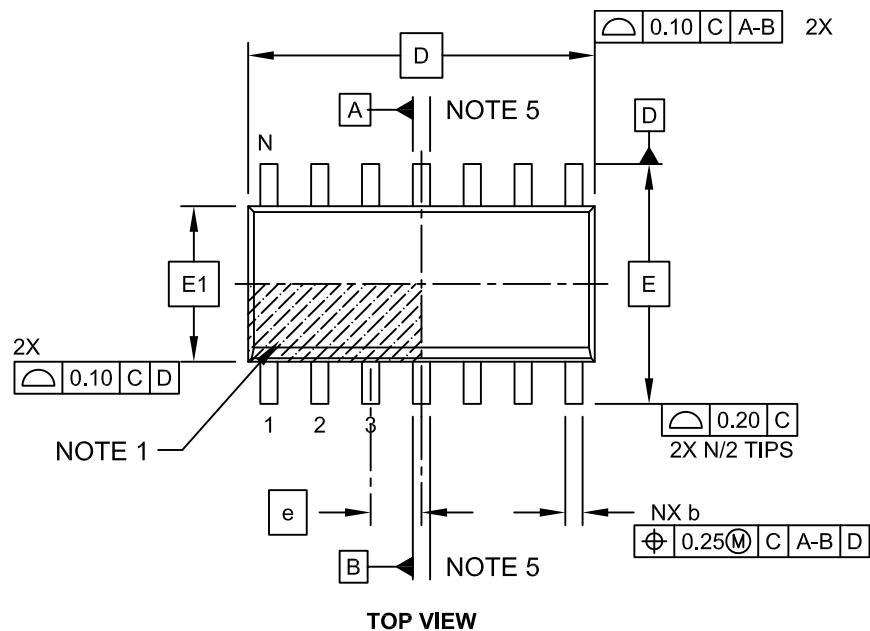
Notes:

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

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

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

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

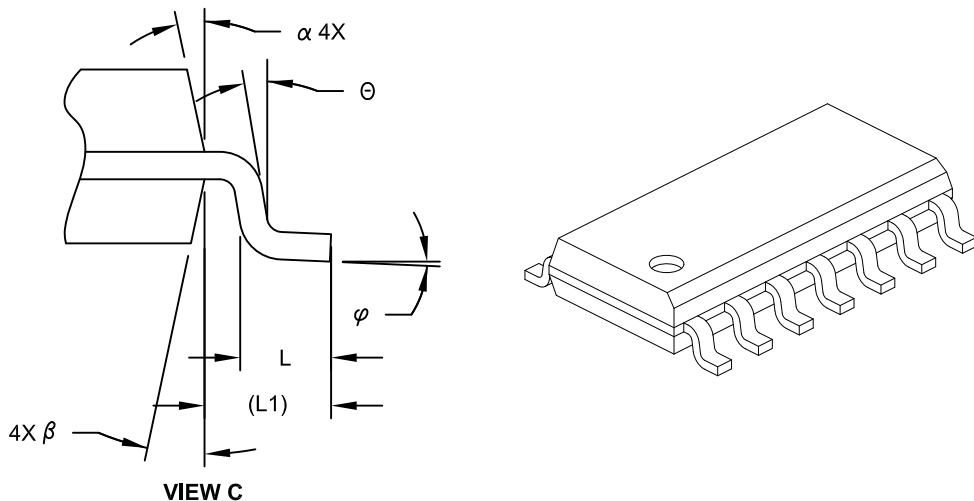


VIEW A-A

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

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

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



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

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. § Significant Characteristic

3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.

4. Dimensioning and tolerancing per ASME Y14.5M

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

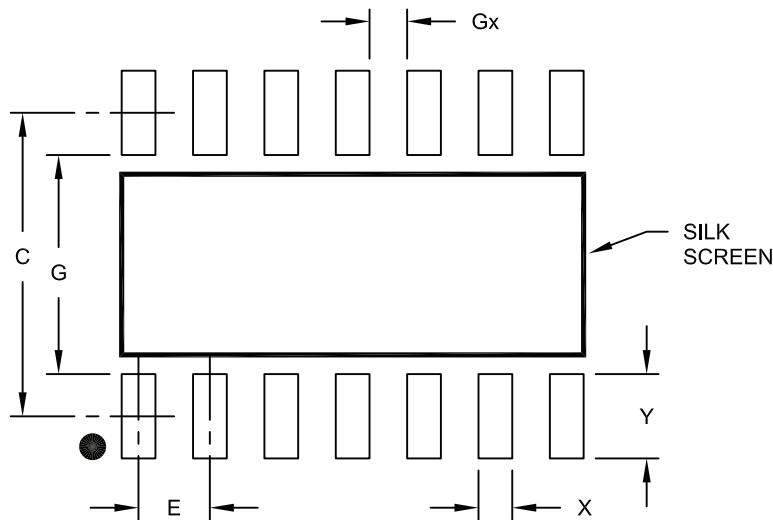
REF: Reference Dimension, usually without tolerance, for information purposes only.

5. Datums A & B to be determined at Datum H.

PIC16(L)F1574/5/8/9

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

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		5.40	
Contact Pad Width	X			0.60
Contact Pad Length	Y			1.50
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	3.90		

Notes:

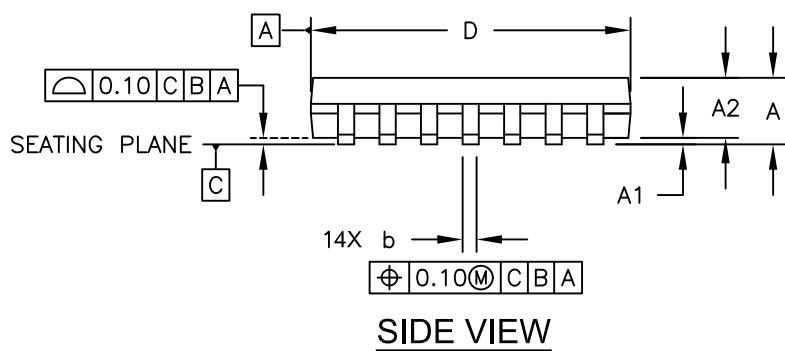
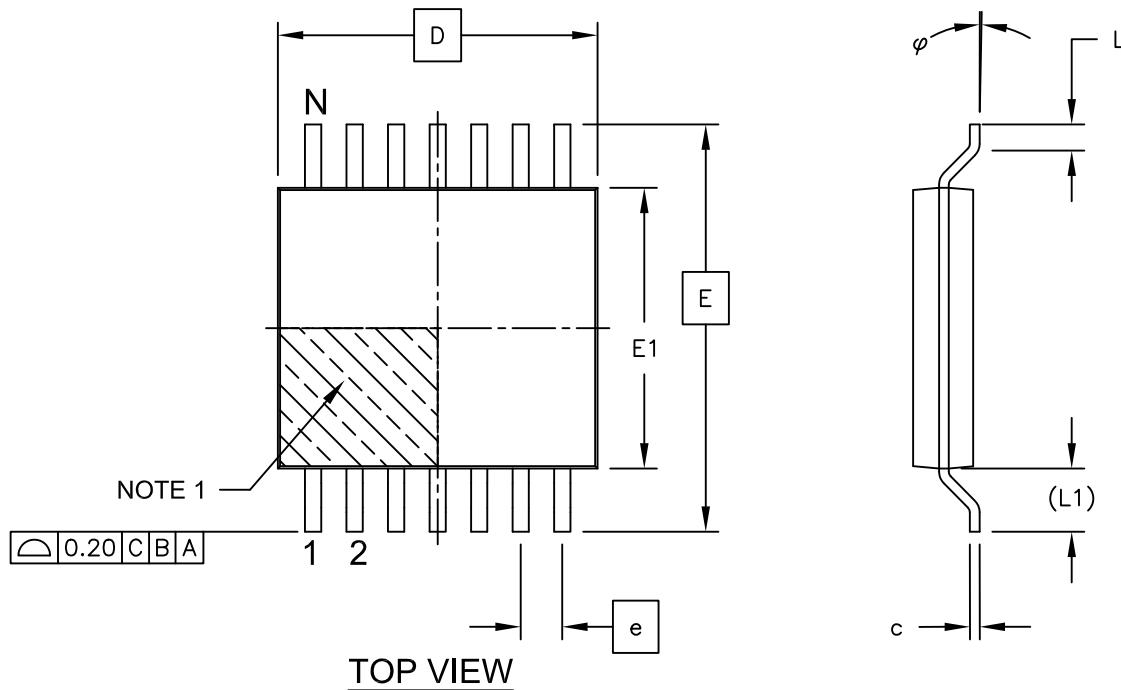
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2065A

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

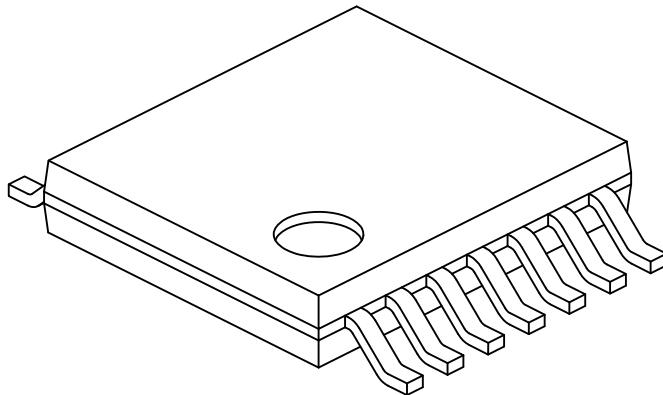
注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



PIC16(L)F1574/5/8/9

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

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



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

Notes:

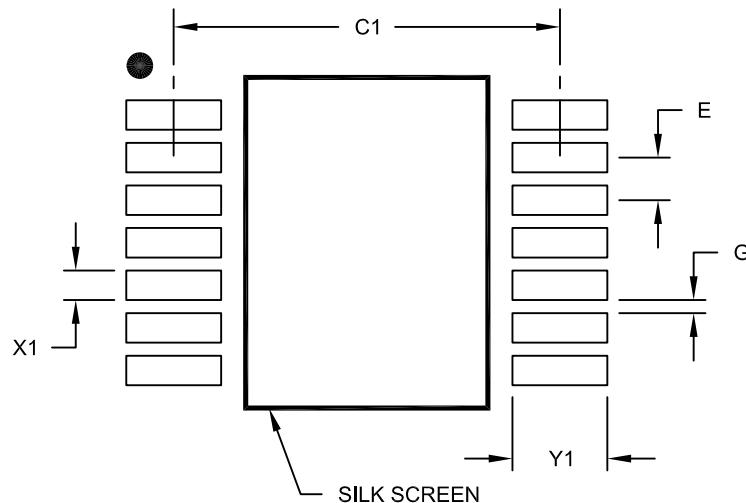
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

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

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

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

Units		MILLIMETERS		
Dimension Limits		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:

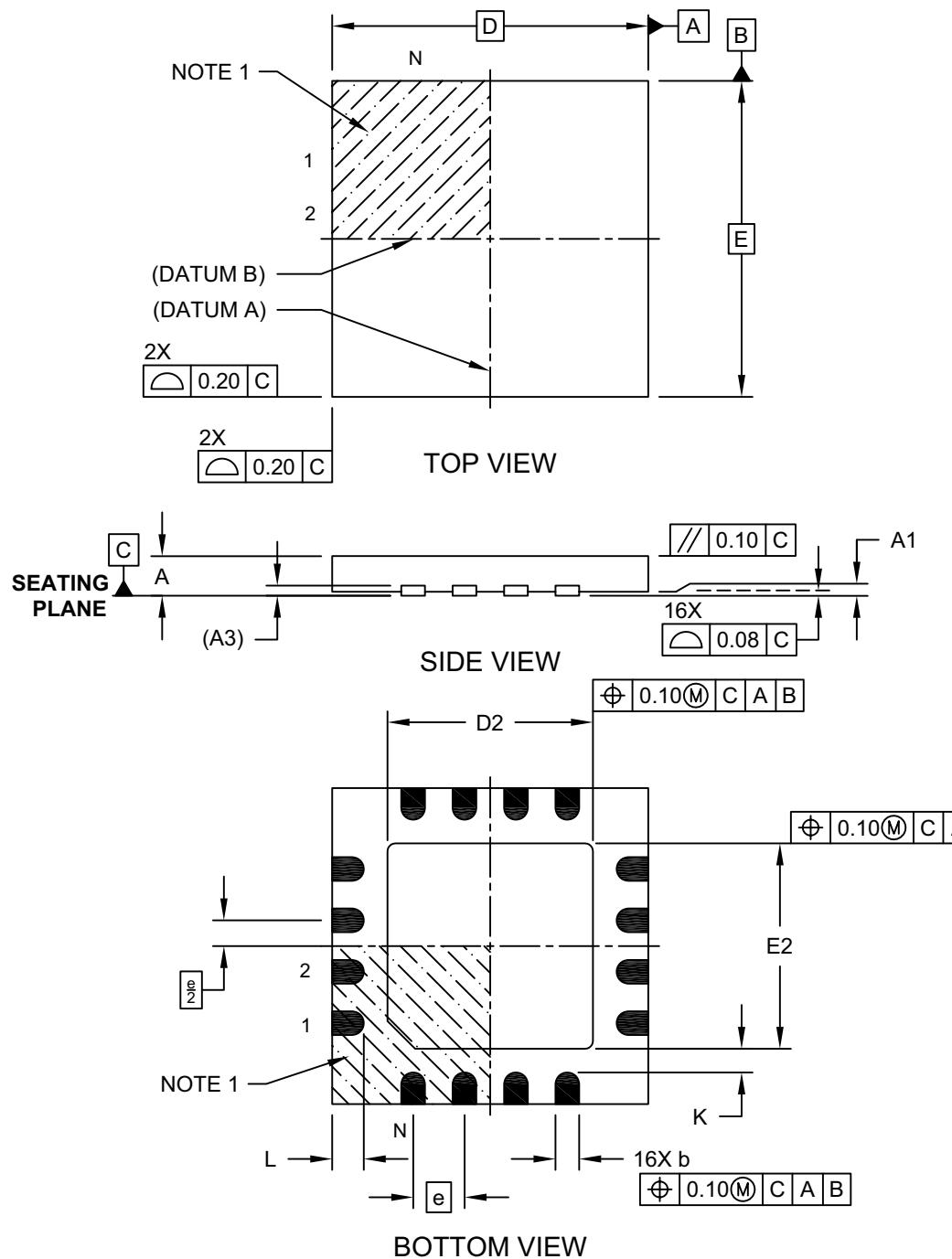
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2087A

16 引脚超薄塑封正方扁平无脚封装 (JQ) —— 主体 4x4x0.5 mm [UQFN]

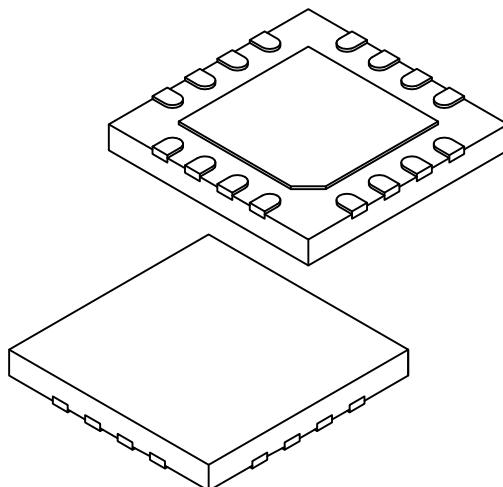
注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Microchip Technology Drawing C04-257A Sheet 1 of 2

16 引脚超薄塑封正方扁平无脚封装 (JQ) —— 主体 4x4x0.5 mm [UQFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins		N		16
Pitch		e		0.65 BSC
Overall Height		A		0.45 0.50 0.55
Standoff		A1		0.00 0.02 0.05
Terminal Thickness		A3		0.127 REF
Overall Width		E		4.00 BSC
Exposed Pad Width		E2		2.50 2.60 2.70
Overall Length		D		4.00 BSC
Exposed Pad Length		D2		2.50 2.60 2.70
Terminal Width		b		0.25 0.30 0.35
Terminal Length		L		0.30 0.40 0.50
Terminal-to-Exposed-Pad		K		- -

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Package is saw singulated

3. Dimensioning and tolerancing per ASME Y14.5M

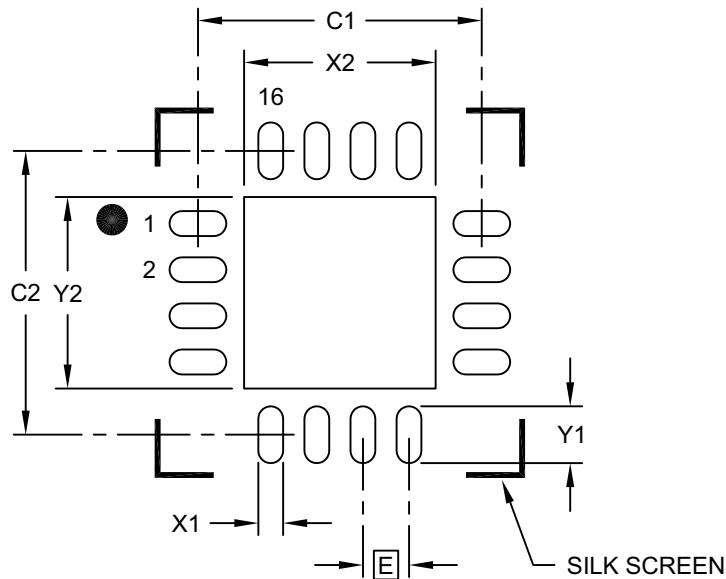
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

PIC16(L)F1574/5/8/9

16 引脚超薄塑封正方扁平无脚封装 (JQ) —— 主体 4x4x0.5 mm [UQFN]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch		0.65 BSC		
Optional Center Pad Width	X2			2.70
Optional Center Pad Length	Y2			2.70
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X16)	X1			0.35
Contact Pad Length (X16)	Y1			0.80

Notes:

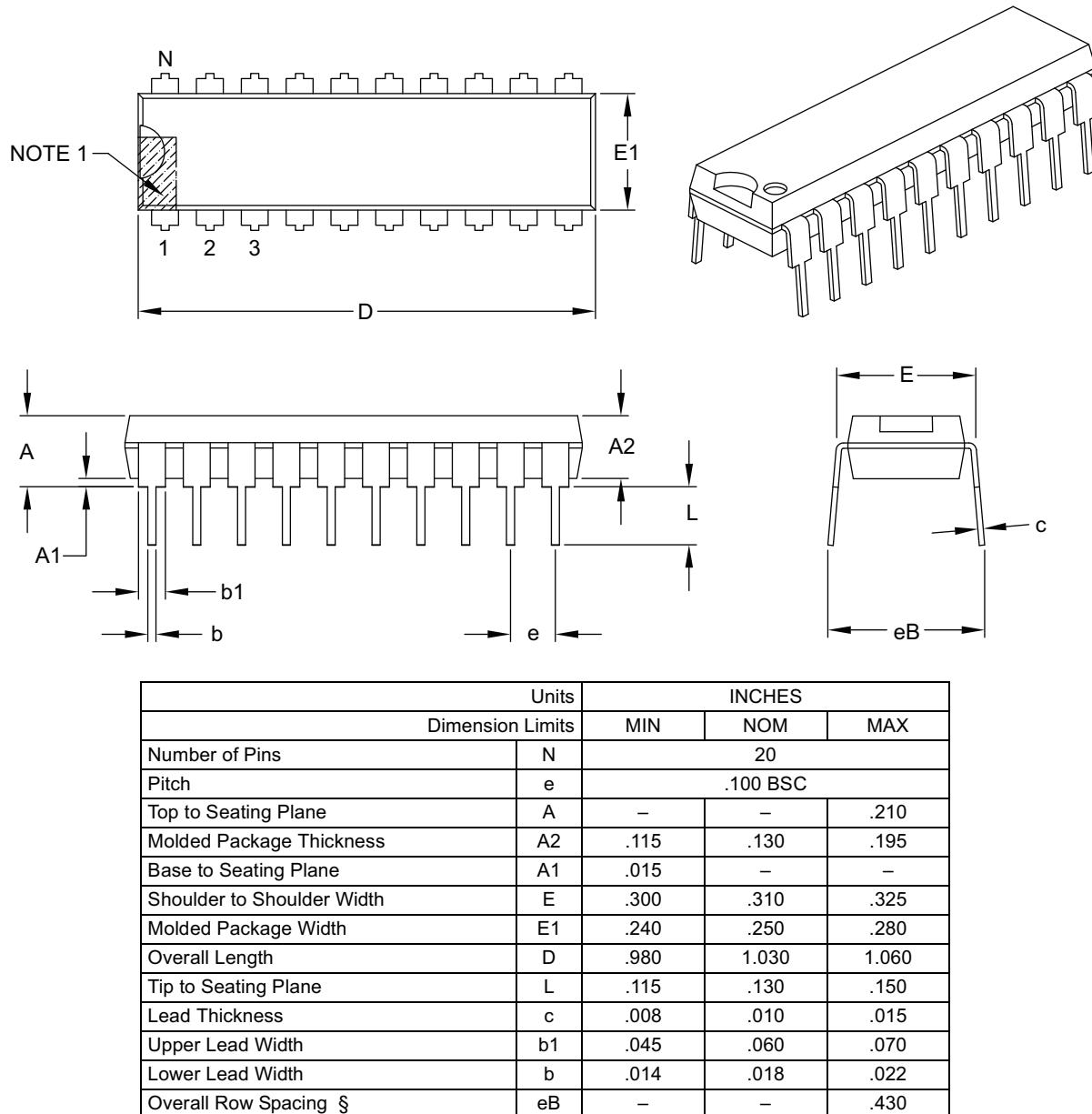
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2257A

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

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



Notes:

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

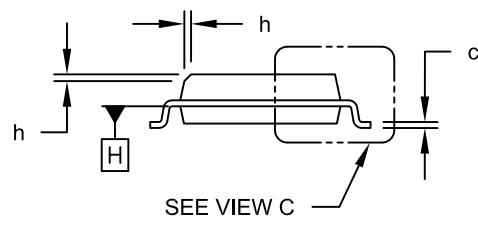
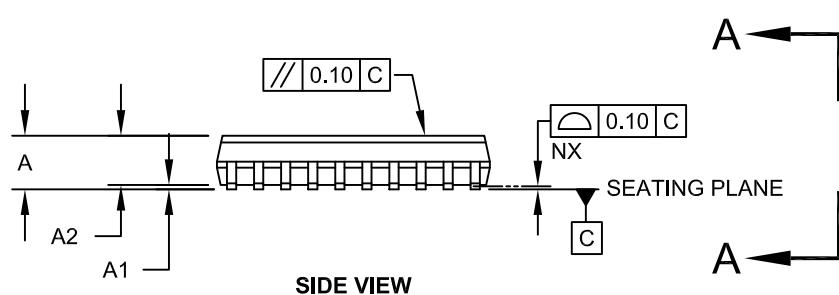
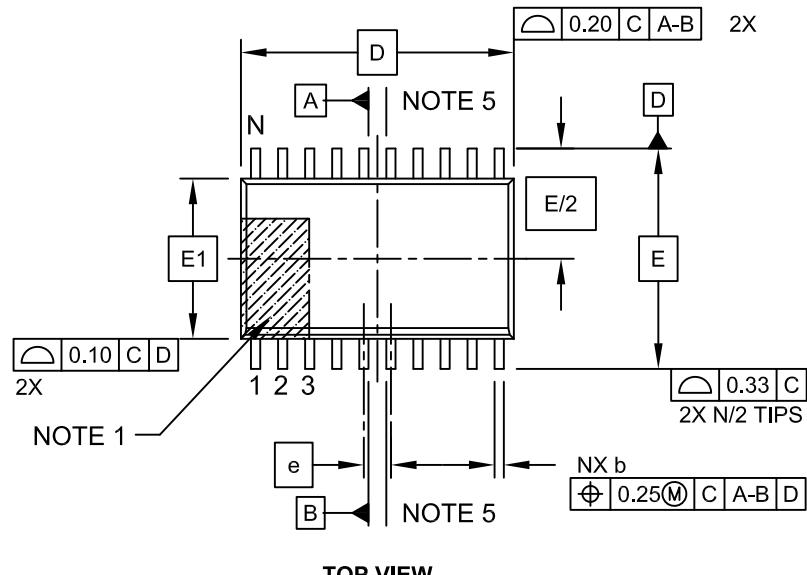
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B

PIC16(L)F1574/5/8/9

20 引脚塑封小外形封装 (SO) — 宽条，主体 7.50 mm [SOIC]

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

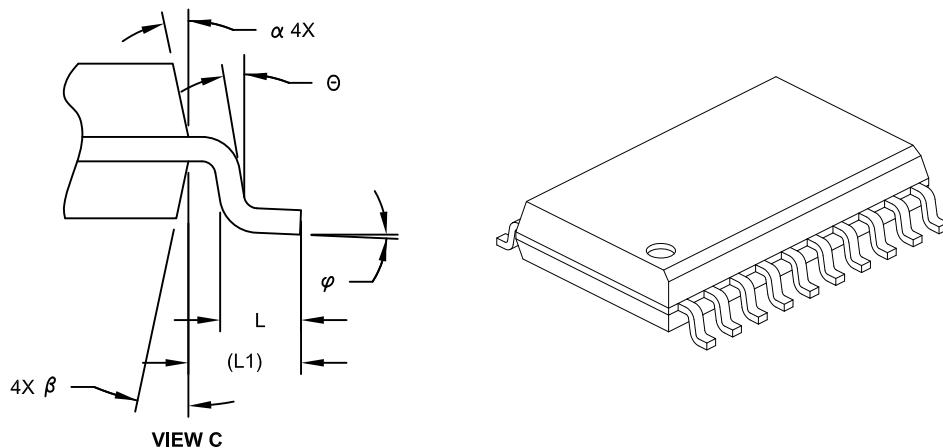


VIEW A-A

Microchip Technology Drawing C04-094C Sheet 1 of 2

20 引脚塑封小外形封装 (SO) —— 宽条, 主体 7.50 mm [SOIC]

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



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Pins	N		20		
Pitch	e		1.27	BSC	
Overall Height	A		-	-	2.65
Molded Package Thickness	A2		2.05	-	-
Standoff	§	A1	0.10	-	0.30
Overall Width	E		10.30	BSC	
Molded Package Width	E1		7.50	BSC	
Overall Length	D		12.80	BSC	
Chamfer (Optional)	h	0.25	-	0.75	
Foot Length	L	0.40	-	1.27	
Footprint	L1		1.40	REF	
Lead Angle	θ	0°	-	-	
Foot Angle	φ	0°	-	8°	
Lead Thickness	c	0.20	-	0.33	
Lead Width	b	0.31	-	0.51	
Mold Draft Angle Top	α	5°	-	15°	
Mold Draft Angle Bottom	β	5°	-	15°	

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M

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

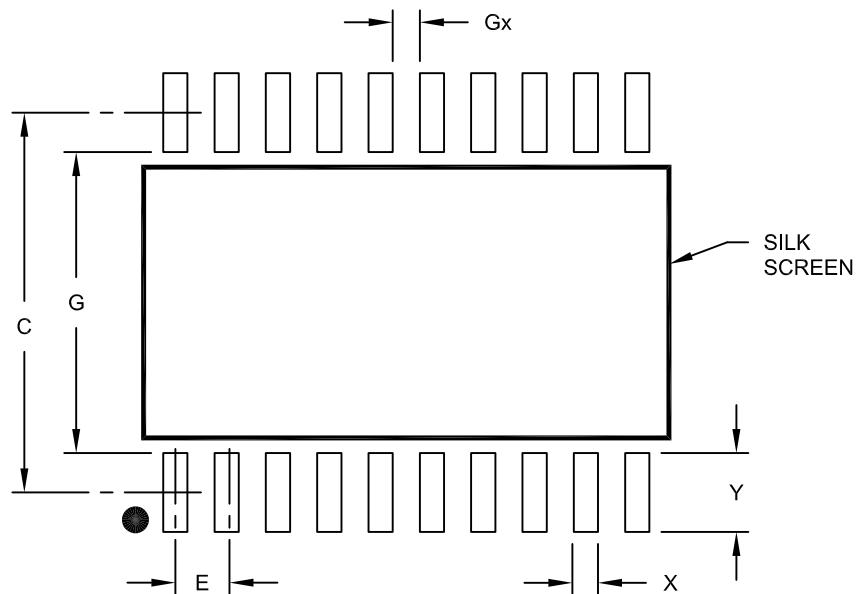
REF: Reference Dimension, usually without tolerance, for information purposes only.

- Datums A & B to be determined at Datum H.

PIC16(L)F1574/5/8/9

20 引脚塑封小外形封装 (SO) — 宽条, 主体 7.50 mm [SOIC]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27	BSC
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

Notes:

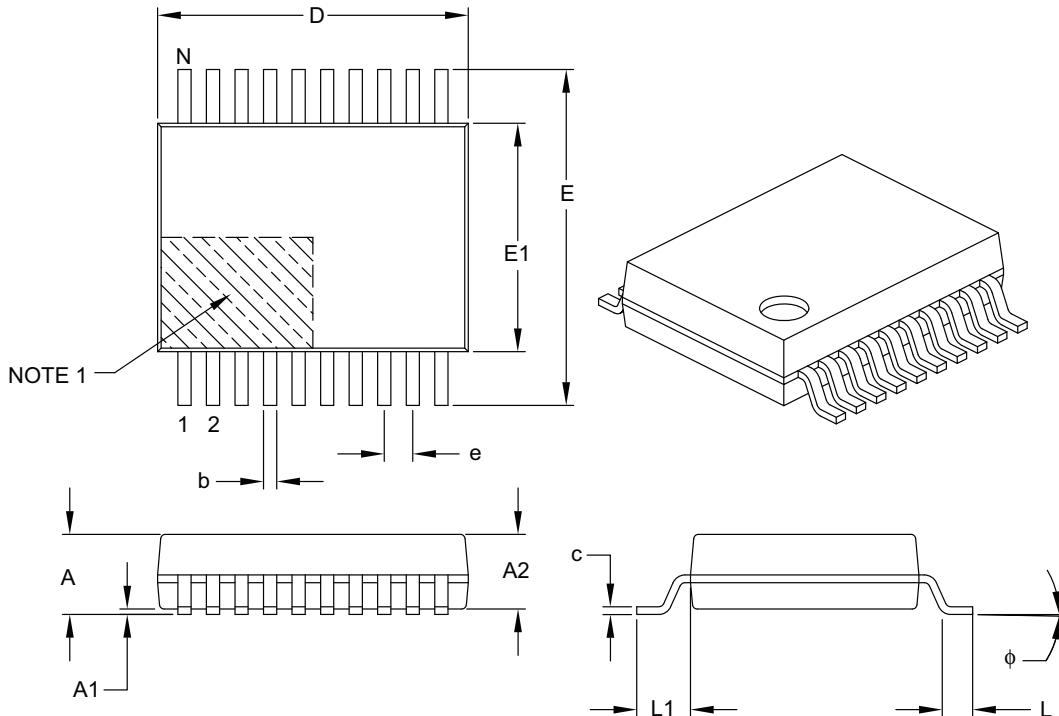
- ## 1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2094A

20 引脚塑封紧缩小外形封装 (SS) —— 主体 5.30 mm [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	-	-	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	-	-
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	-	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	-	0.38

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

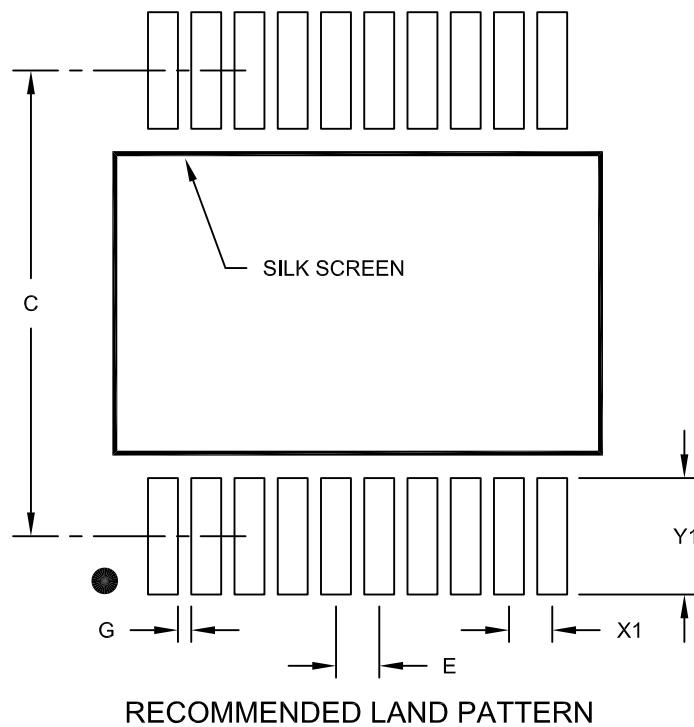
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

PIC16(L)F1574/5/8/9

20 引脚塑封紧缩小外形封装 (SS) —— 主体 5.30 mm [SSOP]

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



Units		MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX
Contact Pitch	E	0.65	BSC	
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

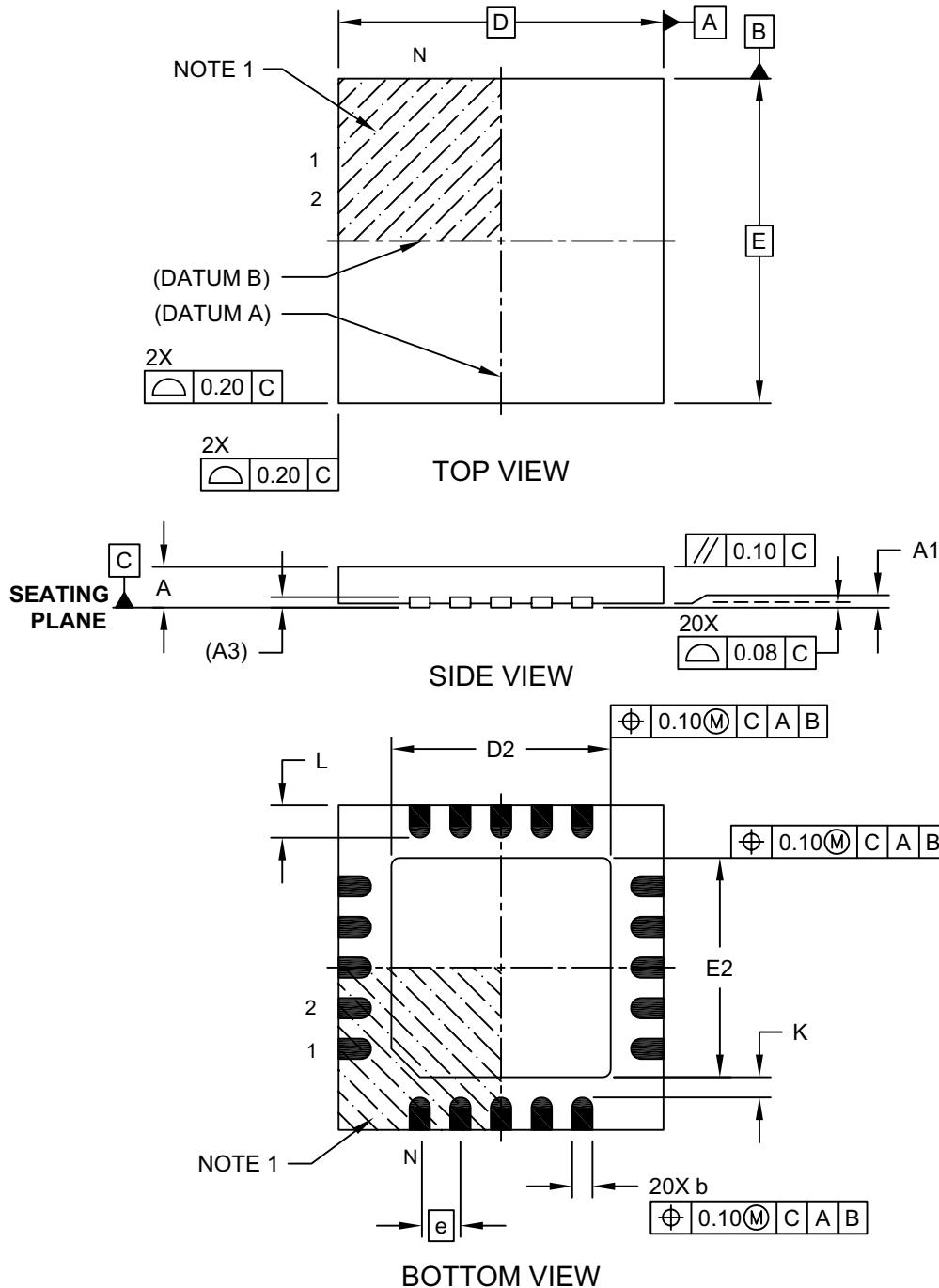
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2072A

20 引脚超薄塑封正方扁平无脚封装 (GZ) —— 主体 4x4x0.5 mm [UQFN]

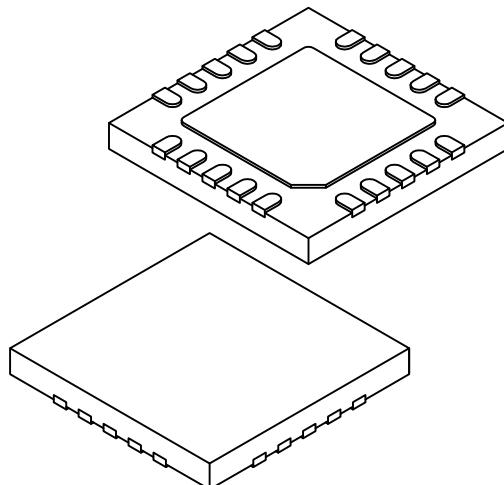
注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



PIC16(L)F1574/5/8/9

20 引脚超薄塑封正方扁平无脚封装 (GZ) —— 主体 4x4x0.5 mm [UQFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals		N		20
Pitch		e		0.50 BSC
Overall Height		A		0.45 0.50 0.55
Standoff		A1		0.00 0.02 0.05
Terminal Thickness		A3		0.127 REF
Overall Width		E		4.00 BSC
Exposed Pad Width		E2		2.60 2.70 2.80
Overall Length		D		4.00 BSC
Exposed Pad Length		D2		2.60 2.70 2.80
Terminal Width		b		0.20 0.25 0.30
Terminal Length		L		0.30 0.40 0.50
Terminal-to-Exposed-Pad		K		- -

Notes:

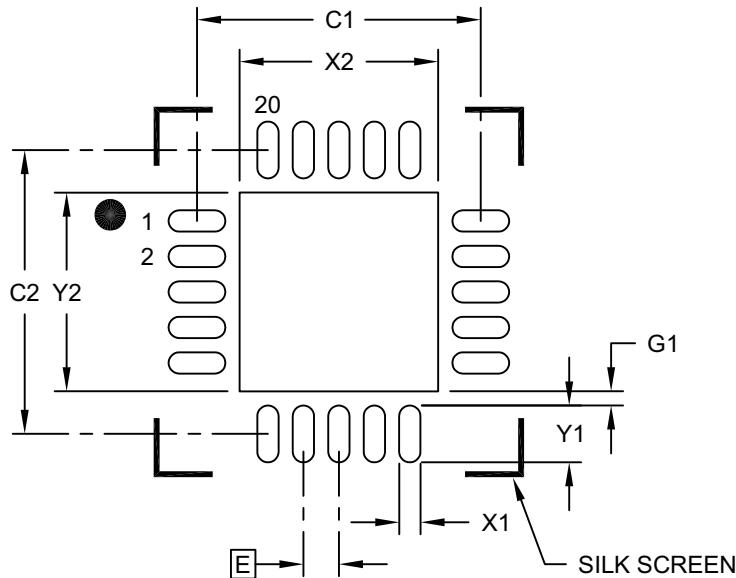
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

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

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

20 引脚超薄塑封正方扁平无脚封装 (GZ) —— 主体 4x4x0.5 mm [UQFN]

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



RECOMMENDED LAND PATTERN

		UNITS			MILLIMETERS		
		DIMENSION LIMITS			MIN	NOM	MAX
Contact Pitch	E				0.50	BSC	
Optional Center Pad Width	X2					2.80	
Optional Center Pad Length	Y2					2.80	
Contact Pad Spacing	C1				4.00		
Contact Pad Spacing	C2				4.00		
Contact Pad Width (X20)	X1					0.30	
Contact Pad Length (X20)	Y1					0.80	
Contact Pad to Center Pad (X20)	G1	0.20					

Notes:

- Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2255A

附录 A: 版本历史

版本 A (2015 年 2 月)

本文档的初始版本。

版本 B (2015 年 9 月)

增加了第 5.4 节：“休眠之前的时钟切换”。

更新了封面页上的“低功耗特性和存储器”部分。

更新了例 3-2 和例 16-1；图 8-1、图 22-1 和图 23-8 至图 23-13；寄存器 8-1、寄存器 23-6、寄存器 24-2 和寄存器 24-3；第 8.2.2 节、第 16.2.6 节、第 22.0 节、第 23.3.3 节、第 24.9.1.2 节、第 24.11.1 节和第 27.1 节；以及表 27-1、表 27-2、表 27-3、表 27-8 和表 27-11。

版本 C (2016 年 1 月)

在“直流和交流特性图表”章节中增加了一些图表。其他少量修正。

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。只要使用常用的因特网浏览器即可访问。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存档软件
- **一般技术支持** —— 常见问题 (Frequently Asked Questions, 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 的客户通知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://www.microchip.com/support> 获得网上技术支持

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

部件编号 器件	[X] ⁽¹⁾ 卷带式选项	-	X 温度范围	/XX 封装	XXX 定制信息	示例:
器件: PIC16LF1574、PIC16F1574、PIC16LF1575、PIC16F1575 PIC16LF1578、PIC16F1578、PIC16LF1579 和 PIC16F1579						<ul style="list-style-type: none">a) PIC16LF1578T - I/SO 卷带式， 工业级温度， SOIC 封装b) PIC16F1575 - I/P 工业级温度 PDIP 封装c) PIC16LF1574-E/JQ 扩展级温度 UQFN 封装
卷带式选项: 空白 = 标准封装（料管或托盘封装） T = 卷带式 ⁽¹⁾						
温度范围: I = -40°C 至 +85°C (工业级) E = -40°C 至 +125°C (扩展级)						
封装: (2) GZ = 20 引脚 UQFN (4x4x0.5 mm) JQ = 16 引脚 UQFN (4x4x0.5 mm) P = 塑封 DIP SL = 14 引脚 SOIC SO = 20 引脚 SOIC SS = 20 引脚 SSOP ST = 14 引脚 TSSOP						<p>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。</p> <p>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。</p>
定制编号: QTP、SQTP、代码或特殊要求（否则为空白）						

请注意以下有关 **Microchip** 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 **Microchip** 产品性能和使用情况的有用信息。**Microchip Technology Inc.** 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 **Microchip Technology Inc.** 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。**Microchip** 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe，与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV = ISO/TS 16949 =

商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BeaconThings、BitCloud、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、RightTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MLIB、MLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM.net、PICDEM、PICtail、PureSilicon、QMatrix、RightTouch 徽标、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQL、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国内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 在除美国内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。

© 2017, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-1464-3



全球销售及服务网点

美洲

公司总部 Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持：
<http://www.microchip.com/support>
网址：www.microchip.com

亚特兰大 Atlanta

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

奥斯汀 Austin, TX

Tel: 1-512-257-3370

波士顿 Boston

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

芝加哥 Chicago

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

达拉斯 Dallas

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

底特律 Detroit

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

休斯敦 Houston, TX

Tel: 1-281-894-5983

印第安纳波利斯

Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles

Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC

Tel: 1-919-844-7510

纽约 New York, NY

Tel: 1-631-435-6000

圣何塞 San Jose, CA

Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto

Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

亚太总部 Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong

Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京

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

中国 - 成都

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

中国 - 重庆

Tel: 86-23-8980-9588

Fax: 86-23-8980-9500

中国 - 东莞

Tel: 86-769-8702-9880

中国 - 广州

Tel: 86-20-8755-8029

中国 - 杭州

Tel: 86-571-8792-8115

Fax: 86-571-8792-8116

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: 86-21-3326-8000

Fax: 86-21-3326-8021

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 - 香港特别行政区

Tel: 852-2943-5100

Fax: 852-2401-3431

亚太地区

中国 - 珠海

Tel: 86-756-321-0040
Fax: 86-756-321-0049

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: 886-3-5778-366

澳大利亚 Australia - Sydney

Tel: 61-2-9868-6733

印度 India - Bangalore

Tel: 91-80-3090-4444

印度 India - New Delhi

Tel: 91-11-4160-8631

印度 India - Pune

Tel: 91-20-3019-1500

日本 Japan - Osaka

Tel: 81-6-6152-7160

日本 Japan - Tokyo

Tel: 81-3-6880-3770

韩国 Korea - Daegu

Tel: 82-53-744-4301

韩国 Korea - Seoul

Tel: 82-2-554-7200

马来西亚 Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857

马来西亚 Malaysia - Penang

Tel: 60-4-227-8870

菲律宾 Philippines - Manila

Tel: 63-2-634-9065

新加坡 Singapore

Tel: 65-6334-8870

泰国 Thailand - Bangkok

Tel: 66-2-694-1351

欧洲

奥地利 Austria - Wels

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark - Copenhagen

Tel: 45-4450-2828
Fax: 45-4485-2829

芬兰 Finland - Espoo

Tel: 358-9-4520-820

法国 France - Paris

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

法国 France - Saint Cloud

Tel: 33-1-30-60-70-00

德国 Germany - Garching

Tel: 49-8931-9700

德国 Germany - Haan

Tel: 49-2129-3766400

德国 Germany - Heilbronn

Tel: 49-7131-67-3636

德国 Germany - Karlsruhe

Tel: 49-721-625370

德国 Germany - Munich

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

德国 Germany - Rosenheim

Tel: 49-8031-354-560

以色列 Israel - Ra'anana

Tel: 972-9-744-7705

意大利 Italy - Milan

Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Padova

Tel: 39-049-7625286

荷兰 Netherlands - Drunen

Tel: 31-416-690399

Fax: 31-416-690340

挪威 Norway - Trondheim

Tel: 47-7289-7561

波兰 Poland - Warsaw

Tel: 48-22-3325737

罗马尼亚

Romania - Bucharest

Tel: 40-21-407-87-50

西班牙 Spain - Madrid

Tel: 34-91-708-08-90

Fax: 34-91-708-08-91

瑞典 Sweden - Gothenberg

Tel: 46-31-704-60-40

瑞典 Sweden - Stockholm

Tel: 46-8-5090-4654

英国 UK - Wokingham

Tel: 44-118-921-5800

Fax: 44-118-921-5820