

PIC18F2450/4450
数据手册

采用纳瓦技术的
28/40/44 引脚高性能
12 MIPS 增强型闪存
USB 单片机

© 2007 Microchip Technology Inc.

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下， Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是 “牢不可破”的。

代码保护功能处于持续发展中。 Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 （Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。**Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。** Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、 Microchip 徽标、 Accuron、 dsPIC、 KEELOQ、 microID、 MPLAB、 PIC、 PICmicro、 PICSTART、 PRO MATE、 PowerSmart、 rfPIC 和 SmartShunt 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

AmpLab、 FilterLab、 Migratable Memory、 MXDEV、 MXLAB、SEEVAL、SmartSensor 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、 Application Maestro、 CodeGuard、 dsPICDEM、 dsPICDEM.net、 dsPICworks、 ECAN、 ECONOMONITOR、 FanSense、 FlexROM、 fuzzyLAB、In-Circuit Serial Programming、ICSP、ICEPIC、 Linear Active Thermistor、 Mindi、 MiWi、 MPASM、 MPLIB、 MPLINK、 PICkit、 PICDEM、 PICDEM.net、 PICLAB、 PICtail、 PowerCal、 PowerInfo、 PowerMate、 PowerTool、REAL ICE、rfLAB、rfPICDEM、Select Mode、 Smart Serial、 SmartTel、 Total Endurance、 UNI/O、 WiperLock和ZENA均为Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2007, Microchip Technology Inc. 版权所有。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip 位于美国亚利桑那州 Chandler 和 Tempe、位于俄勒冈州 Gresham 及位于加利福尼亚州 Mountain View 的全球总部、设计中心和晶圆生产厂均通过了 ISO/TS-16949:2002 认证。公司在 PIC® 8 位单片机、 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方面的质量体系流程均符合 ISO/TS-16949:2002。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。

PIC18F2450/4450

采用纳瓦技术的 28/40/44 引脚高性能 12 MIPS 增强型闪存 USB 单片机

通用串行总线特性：

- 符合 USB V2.0 规范
- 低速 （1.5 Mb/s）和全速 （12 Mb/s）
- 支持控制、中断、同步和批量传输
- 支持最多 32 个端点 （双向为 16 个）
- 256 字节双存取 RAM （用于 USB）
- 带有片上稳压器的片上 USB 收发器
- 与外部 USB 收发器相连的接口

功耗管理模式：

- 运行：CPU 工作，外设打开
- 空闲：CPU 不工作，外设打开
- 休眠：CPU 不工作，外设关闭
- 空闲模式时电流可降至 5.8 μA （典型值）
- 休眠模式时电流可降至 0.1 μA （典型值）
- Timer1 振荡器：1.8 μA， 32 kHz， 2V （典型值）
- 看门狗定时器：2.1 μA （典型值）
- 双速振荡器启动

灵活的振荡器结构：

- 四种晶振模式，包括用于 USB 的高精度 PLL
- 两种外部时钟模式，频率最高为 48 MHz
- 内部 31 kHz 振荡器
- 辅助振荡器使用 Timer1 （工作频率为 32 kHz）
- 双振荡器选项允许单片机和 USB 模块以不同的时钟速率运行
- 故障保护时钟监视器：
  - 当任何时钟停止时可使器件安全断电

外设特性：

- 高灌 / 拉电流：25 mA/25 mA
- 3 个外部中断
- 3 个定时器模块 （Timer0 到 Timer2）
- 捕捉 / 比较 /PWM （CCP）模块：
  - 16 位捕捉，最大分辨率 5.2 ns
  - 16 位捕捉，最大分辨率 83.3 ns
  - PWM 输出：PWM 分辨率是 1 至 10 位
- 增强型 USART 模块：
  - 支持 LIN 总线
- 最多 13 路通道的 10 位模数转换器模块 （A/D）：
  - 采样率最高为 100 ksps
  - 采集时间可编程

单片机特性：

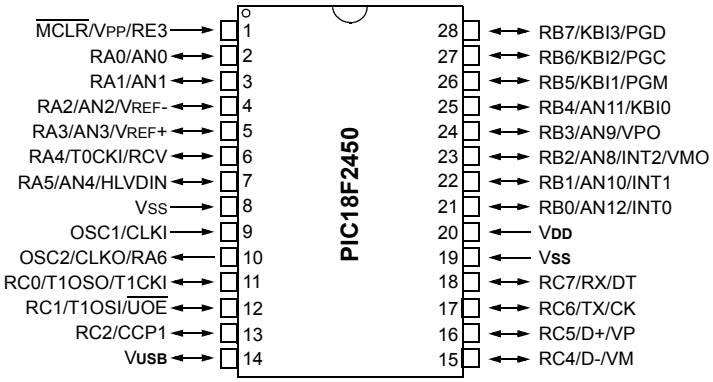
- 优化的 C 编译器架构，带有可选的扩展指令集
- 闪存存储器保存时间：> 40 年
- 可在软件控制下自行编程
- 中断优先级
- 8 x 8 单周期硬件乘法器
- 扩展型看门狗定时器 （WDT）：
  - 可编程周期从 4 ms 到 131s
- 可编程代码保护
- 通过两个引脚进行单电源在线串行编程 （In-Circuit Serial Programming™， ICSP™）
- 通过两个引脚进行在线调试 （In-Circuit Debug，ICD）
- 可选的专用 ICD/ICSP 端口 （仅 44 引脚 TQFP 器件）
- 宽工作电压范围 （2.0V 到 5.5V）

器件	程序存储器		数据存储器 SRAM （字节）	I/O	10 位 A/D （通道）	CCP	EUSART	8/16 位定时器
	闪存 （字节）	单字指令数						
PIC18F2450	16K	8192	768*	23	10	1	1	1/2
PIC18F4450	16K	8192	768*	34	13	1	1	1/2

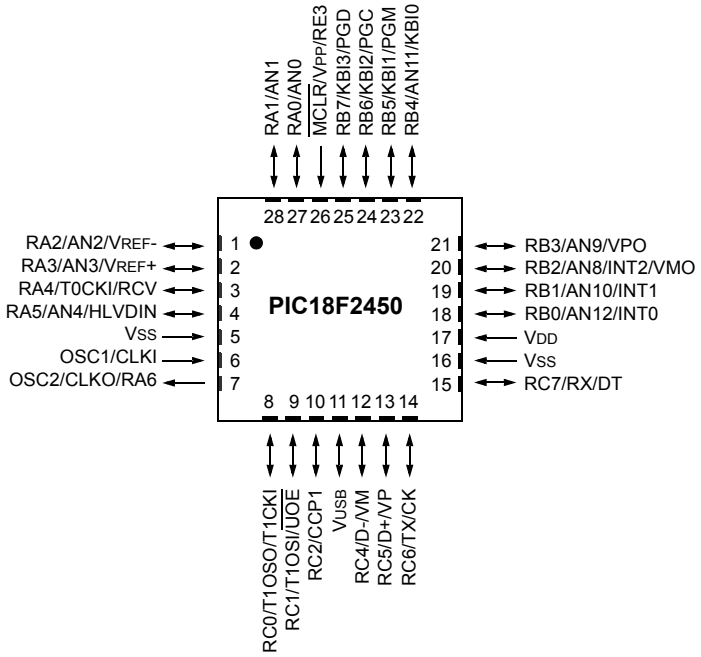
* 包括用于 USB 模块的 256 字节双存取 RAM，与数据存储器共享。

引脚示意图

28 引脚 SDIP 和 SOIC



28 引脚 QFN

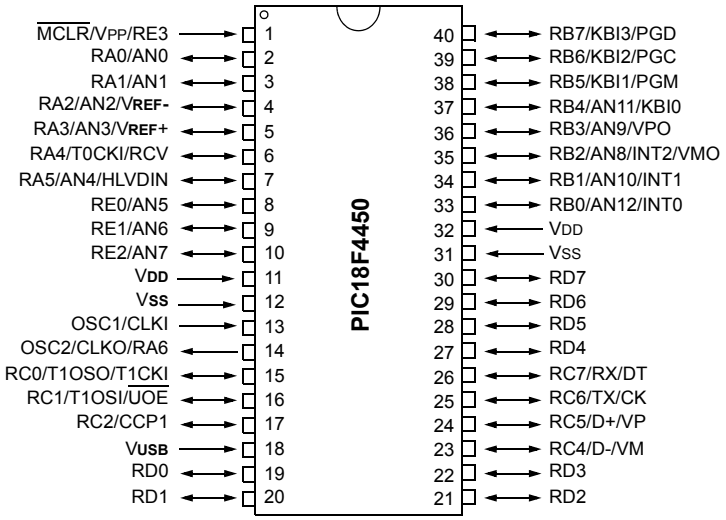


注： 引脚排列可能有变化。

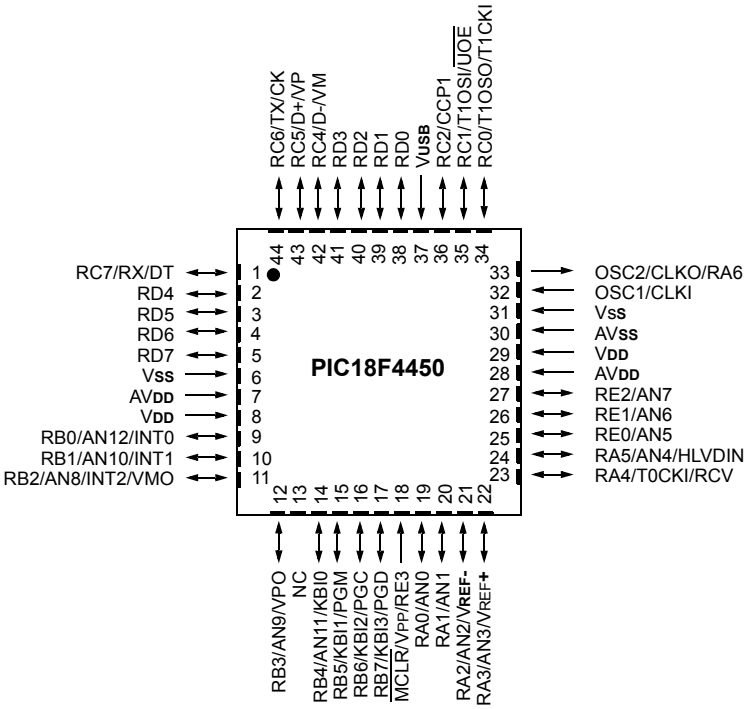
 © 2007 Microchip Technology Inc.

引脚示意图 （续）

40 引脚 PDIP



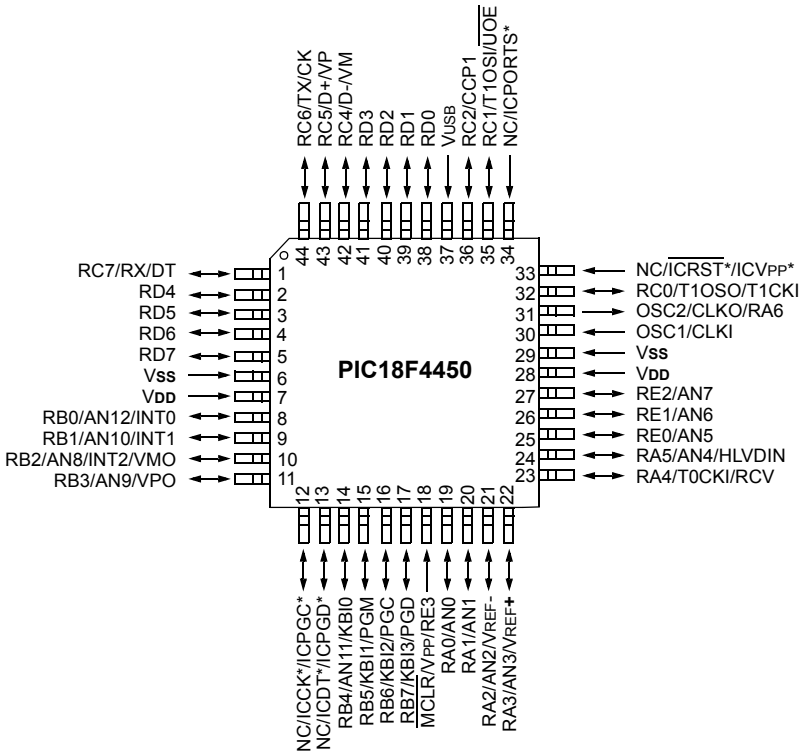
44 引脚 QFN



注： 引脚排列可能有变化。

引脚示意图 （续）

44 引脚 TQFP



注： 引脚排列可能有变化。

* 引脚的分配取决于器件配置。

 © 2007 Microchip Technology Inc.

目录

1.0 器件概述 7
2.0 振荡器配置 23
3.0 功耗管理模式 33
4.0 复位 41
5.0 存储器构成 53
6.0 闪存程序存储器 73
7.0 8 x 8 硬件乘法器 83
8.0 中断 85
9.0 I/O 端口 99
10.0 Timer0 模块 111
11.0 Timer1 模块 115
12.0 Timer2 模块 121
13.0 捕捉 / 比较 /PWM （CCP）模块 123
14.0 通用串行总线 （USB） 129
15.0 增强型通用同步 / 异步收发器 （EUSART） 153
16.0 10 位模数转换器 （A/D）模块 173
17.0 高 / 低压检测 （HLVD） 183
18.0 CPU 的特性 189
19.0 指令集汇总 211
20.0 开发支持 261
21.0 电气特性 265
22.0 直流和交流特性图表 293
23.0 封装信息 295
附录 A： 版本历史 303
附录 B： 器件差异 303
附录 C： 转换注意事项 304
附录 D： 从基本器件移植到增强型器件 304
附录 E： 从中档器件移植到增强型器件 305
附录 F： 从高档器件移植到增强型器件 305
索引 307
Microchip 网站 315
变更通知客户服务 315
客户支持 315
读者反馈表 316
PIC18F2450/4450 产品标识体系 317

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**，或将本数据手册后附的 **《读者反馈表》**传真到 86-21-5407 5066。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：DS30000A是DS30000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。

 © 2007 Microchip Technology Inc.

1.0 器件概述

本文档涉及以下器件的具体信息：

- PIC18F2450
- PIC18F4450

该器件系列具备所有 PIC18 单片机固有的优点，即以实惠的价格提供出色的计算性能，以及高耐久性的增强型闪存程序存储器。除了这些优点之外，PIC18F2450/4450 系列还增强了器件设计，使得该系列单片机成为许多高性能、低功耗应用的明智选择。

1.1 新的内核功能

1.1.1 纳瓦技术

PIC18F2450/4450 系列的所有器件具有一系列能在工作时显著降低功耗的功能。主要包含以下几项：

- **备用运行模式**：通过将 Timer1 或内部 RC 振荡器作为单片机时钟源，可使代码执行时的功耗降低大约 90%。
- **多种空闲模式**：单片机还可工作在其 CPU 内核禁止而外设仍然工作的情况下。处于这些状态时，功耗能降得更低，只有正常工作时的 4%。
- **动态模式切换**：在器件工作期间可由用户代码调用该功耗管理模式，允许用户将节能的理念融入到他们的应用软件设计中。
- **关键模块低功耗**：Timer1 和看门狗定时器模块的功耗需求可降至最小。具体数值请参见**第 21.0 节“电气特性”**。

1.1.2 通用串行总线 （USB）

PIC18F2450/4450 系列中的器件包括功能齐全的通用串行总线通信模块，该模块符合 USB 规范 2.0 版。该模块可实现所有支持的数据传输类型的低速和全速通信。它还包含片上收发器和 3.3V 稳压器，同时支持使用外部收发器和稳压器。

1.1.3 多个振荡器选项和特性

PIC18F2450/4450 系列的所有器件可提供 12 个不同的振荡器选项，使用户在开发应用硬件时有很大的选择范围。这些选项包括：

- 4 种晶振模式，使用晶振或陶瓷谐振器。
- 4 种外部时钟模式，提供使用两个引脚 （振荡器输入引脚和四分频时钟输出引脚）或一个引脚 （振荡器输入引脚，四分频时钟输出引脚重新分配为通用 I/O 引脚）的选项。
- 1 个 INTRC 时钟源（大约 31 kHz，在温度和 V_{DD} 的工作范围内保持稳定）。此选项可以空出一个振荡器引脚作为额外的通用 I/O 引脚。
- 1 个锁相环 （PLL）倍频器，可在高速晶振和外部振荡器模式下使用，使得时钟频率的范围更宽，从 4 MHz 到 48 MHz。
- 异步双时钟操作，允许 USB 模块使用高频振荡器运行，而单片机其余部分使用内部低功耗振荡器作为时钟。

内部振荡器还提供了一个稳定的参考源，增加了以下功能以使器件更安全地工作：

- **故障保护时钟监视器**：该部件持续监视主时钟源，将其与内部振荡器提供的参考信号作比较。如果时钟发生了故障，单片机会将时钟源切换到内部振荡器，使器件可继续低速工作或安全地关闭应用。
- **双速启动**：该功能允许在上电复位或从休眠模式唤醒时将内部振荡器用作时钟源，直到主时钟源可用为止。

1.2 其他特殊功能

- **存储器耐久性：**程序存储器的增强型闪存单元经评测，可以耐受数万次擦写，高达 100,000 次。
- **自编程能力：**这些器件能在内部软件控制下写入各自的程序存储器空间。通过使用受保护的引导块（位于程序存储器的顶端）中的自举程序，可创建能在现场进行自我更新的应用。
- **扩展指令集：** PIC18F2450/4450 系列在 PIC18 指令集的基础上进行了可选择的扩展，添加了 8 条新指令和变址寻址模式。此扩展可以使用一个器件配置选项使能，它是为优化重入应用程序代码而特别设计的，这些代码原来是使用高级语言（如 C 语言）开发的。
- **增强型可寻址 USART：**该串行通信模块可进行标准的 RS-232 通信并支持 LIN 总线协议。其他增强功能包括自动波特率检测和分辨率更高的 16 位波特率发生器。
- **10 位 A/D 转换器：**该模块具备可编程采集时间，从而不必在选择通道和启动转换之间等待一个采样周期，因而减少了代码开销。
- **专用 ICD/ICSP 端口：**这些器件采用了专用的调试以及编程引脚，这些引脚不与其它单片机功能复用。该功能作为选择封装时的选项提供，允许用户开发 I/O 密集型应用，同时保留在线编程和调试功能。

1.3 系列中各产品的详细说明

PIC18F2450/4450 系列器件具有 28 引脚和 40/44 引脚两种封装形式。图 1-1 和图 1-2 分别为这两类器件的框图。

这两类器件在以下两个方面存在差异：

1. A/D 通道数（28 引脚器件有 10 路， 40/44 引脚器件有 13 路）。
2. I/O 端口数（28 引脚器件上有 3 个双向端口和 1 个仅输入端口， 40/44 引脚器件上有 5 个双向端口）。

本系列器件的所有其他功能都是相同的。表 1-1 总结了这些功能。

表 1-2 和表 1-3 给出了本系列中所有器件的引脚说明。

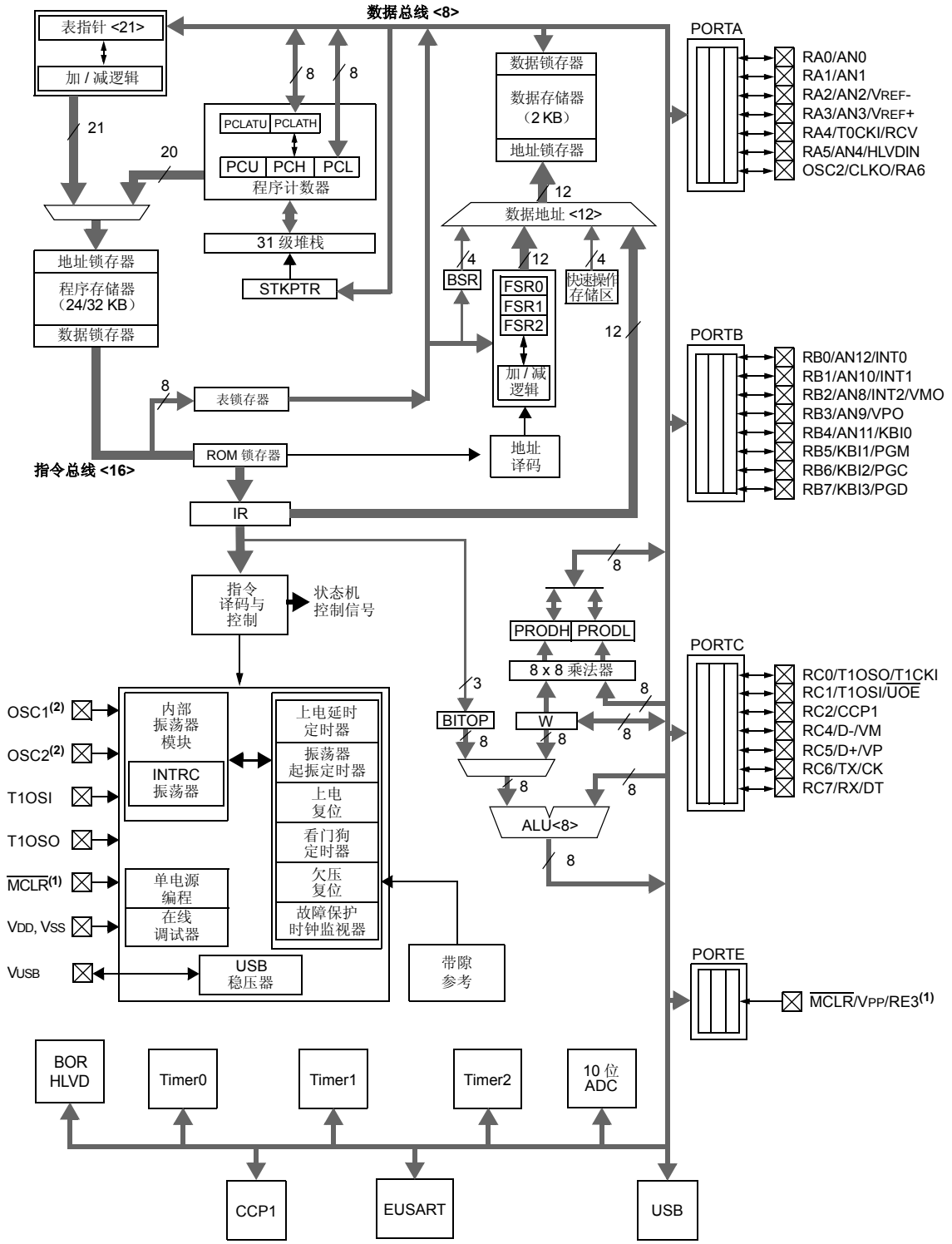
和所有的 Microchip PIC18 器件一样，PIC18F2450/4450 系列中也分标准器件和低压器件。器件编号中标有字母“F”的是带有增强型闪存存储器的标准器件（如 PIC18**F**2450），其工作电压 V_{DD} 范围为 4.2V 到 5.5V。编号中标有“LF”的低压器件（如 PIC18**LF**2450）可工作在扩展的 V_{DD} 范围（2.0V 到 5.5V）中。

 © 2007 Microchip Technology Inc.

表 1-1： 器件特性

特性	PIC18F2450	PIC18F4450
工作频率	DC – 48 MHz	DC – 48 MHz
程序存储器 （字节）	16384	16384
程序存储器 （指令）	8192	8192
数据存储器 （字节）	768	768
中断源	13	13
I/O 端口	端口 A, B, C, (E)	端口 A, B, C, D, E
定时器	3	3
捕捉 / 比较 /PWM 模块	1	1
增强型 USART	1	1
通用串行总线 （USB）模块	1	1
10 位模数转换模块	10 路输入通道	13 路输入通道
复位 （和延时）	POR、 BOR、 RESET 指令、 堆栈满、 堆栈下溢 （PWRT 和 OST）、 MCLR （可选）和 WDT	POR、 BOR、 RESET 指令、 堆栈满、 堆栈下溢 （PWRT 和 OST）、 MCLR （可选）和 WDT
可编程低压检测	有	有
可编程欠压复位	有	有
指令集	75 条指令； 使能了扩展指令集后总共为 83 条指令	75 条指令； 使能了扩展指令集后总共为 83 条指令
封装	28 引脚 PDIP 28 引脚 SOIC	40 引脚 PDIP 44 引脚 QFN 44 引脚 TQFP

图 1-1： PIC18F2450 （28 引脚）框图

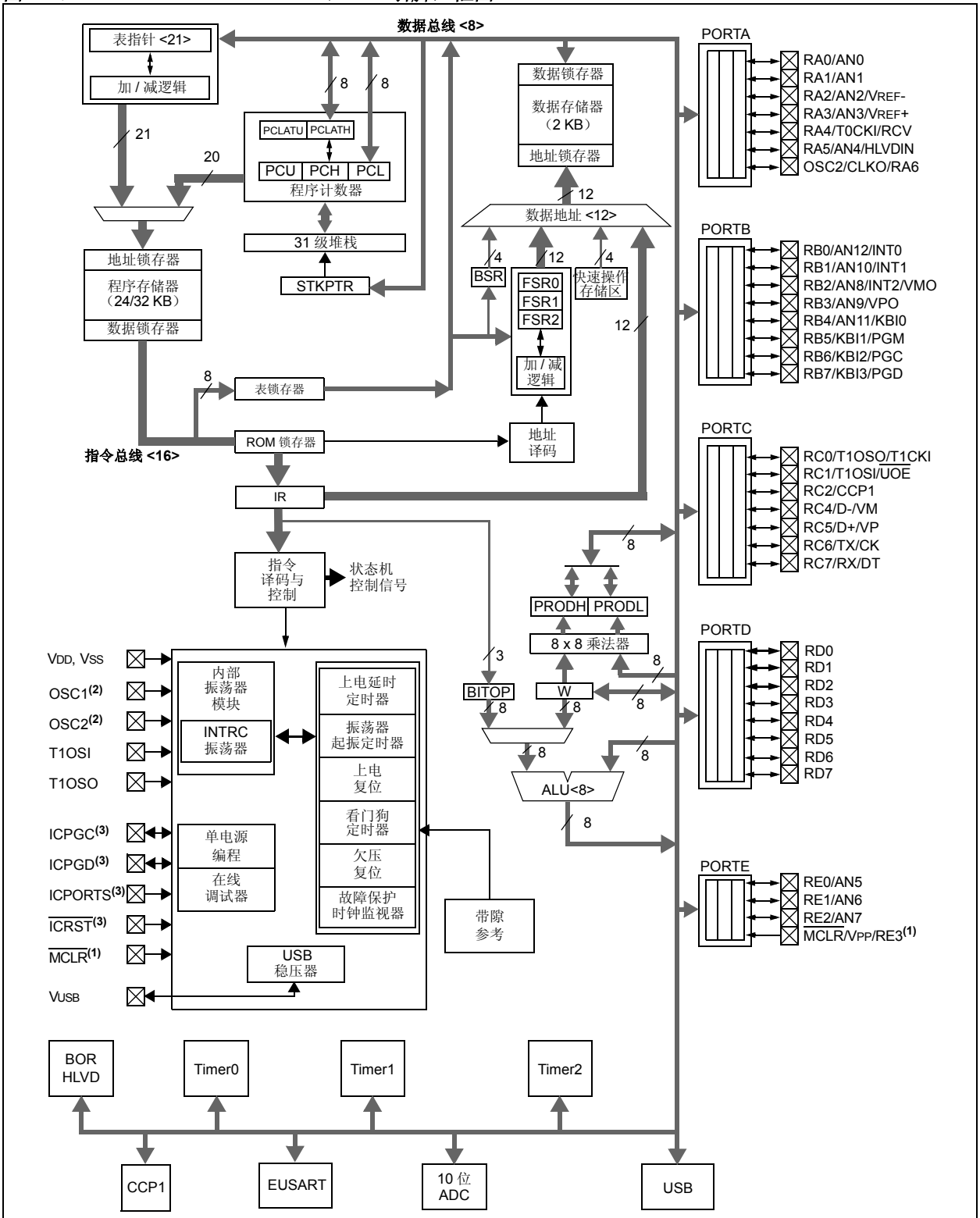


注 1： RE3 与 MCLR 复用，仅在 MCLR 复位禁止时可用。

2： OSC1/CLKI 和 OSC2/CLKO 只有在选择振荡器模式时才可用，此时这些引脚不能被用作数字 I/O 引脚。更多信息，请参见**第 2.0 节“振荡器配置”**。

© 2007 Microchip Technology Inc.

图 1-2： PIC18F4450 （40/44 引脚）框图



注 1： RE3 与 MCLR 复用，仅在 MCLR 复位禁止时可用。

2： OSC1/CLKI 和 OSC2/CLKO 只有在选择振荡器模式时才可用，此时这些引脚不能被用作数字 I/O 引脚。更多信息，请参见**第 2.0 节“振荡器配置”**。

3： 这些引脚仅在某些条件下在 44 引脚 TQFP 上可用。更多信息，请参见**第 18.9 节 “串行 ICPORT 功能 （仅对指定封装）”**。

表 1-2： PIC18F2450 I/O 引脚说明

引脚名称	引脚号		引脚类型	缓冲器类型	说明
	PDIP, SOIC	QFN			
$\overline{MCLR}$/Vpp/RE3	1	26			主复位 （输入）或编程电压 （输入）。
$\overline{MCLR}$			I	ST	主清零 （复位）输入。此引脚为低电平时，器件复位。
VPP			P		编程电压输入。
RE3			I	ST	数字输入。
OSC1/CLKI	9	6			振荡器晶振或外部时钟输入。
OSC1			I	模拟	振荡器晶振或外部时钟源输入。
CLKI			I	模拟	外部时钟源输入。总是与OSC1 引脚功能复用。（见 OSC2/CLKO 引脚信息。）
OSC2/CLKO/RA6	10	7			振荡器晶振或时钟输出。
OSC2			O	—	振荡器晶振输出。在晶振振荡器模式下，该引脚与晶振或谐振器相连。
CLKO			O	—	在选择模式下， OSC2 引脚输出 CLKO 振荡信号，该信号是OSC1 引脚上振荡信号的 4 分频，该频率等于指令周期的倒数。
RA6			I/O	TTL	通用 I/O 引脚。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

© 2007 Microchip Technology Inc.

表 1-2： PIC18F2450 I/O 引脚说明 （续）

引脚名称	引脚号		引脚类型	缓冲器类型	说明
	PDIP, SOIC	QFN			
					PORTA 是双向 I/O 端口。
RA0/AN0	2	27			
RA0			I/O	TTL	数字 I/O。
AN0			I	模拟	模拟输入 0。
RA1/AN1	3	28			
RA1			I/O	TTL	数字 I/O。
AN1			I	模拟	模拟输入 1。
RA2/AN2/VREF-	4	1			
RA2			I/O	TTL	数字 I/O。
AN2			I	模拟	模拟输入 2。
VREF-			I	模拟	A/D 参考电压 （低电平）输入。
RA3/AN3/VREF+	5	2			
RA3			I/O	TTL	数字 I/O。
AN3			I	模拟	模拟输入 3。
VREF+			I	模拟	A/D 参考电压 （高电平）输入。
RA4/T0CKI/RCV	6	3			
RA4			I/O	ST	数字 I/O。
T0CKI			I	ST	Timer0 外部时钟输入。
RCV			I	TTL	外部 USB 收发器 RCV 输入。
RA5/AN4/HLVDIN	7	4			
RA5			I/O	TTL	数字 I/O。
AN4			I	模拟	模拟输入 4。
HLVDIN			I	模拟	高 / 低压检测输入。
RA6	—	—	—	—	请参见 OSC2/CLKO/RA6 引脚信息。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

表 1-2： PIC18F2450 I/O 引脚说明 （续）

引脚名称	引脚号 PDIP, SOIC	引脚号 QFN	引脚类型	缓冲器类型	说明
					PORTB 是双向 I/O 端口。 PORTB 在所有输入端都可软件编程为内部弱上拉。
RB0/AN12/INT0	21	18			
RB0			I/O	TTL	数字 I/O。
AN12			I	模拟	模拟输入 12。
INT0			I	ST	外部中断 0。
RB1/AN10/INT1	22	19			
RB1			I/O	TTL	数字 I/O。
AN10			I	模拟	模拟输入 10。
INT1			I	ST	外部中断 1。
RB2/AN8/INT2/VMO	23	20			
RB2			I/O	TTL	数字 I/O。
AN8			I	模拟	模拟输入 8。
INT2			I	ST	外部中断 2。
VMO			O	—	外部 USB 收发器 VMO 输出。
RB3/AN9/VPO	24	21			
RB3			I/O	TTL	数字 I/O。
AN9			I	模拟	模拟输入 9。
VPO			O	—	外部 USB 收发器 VPO 输出。
RB4/AN11/KBI0	25	22			
RB4			I/O	TTL	数字 I/O。
AN11			I	模拟	模拟输入 11。
KBI0			I	TTL	电平变化中断引脚。
RB5/KBI1/PGM	26	23			
RB5			I/O	TTL	数字 I/O。
KBI1			I	TTL	电平变化中断引脚。
PGM			I/O	ST	低压 ICSP™ 编程使能引脚。
RB6/KBI2/PGC	27	24			
RB6			I/O	TTL	数字 I/O。
KBI2			I	TTL	电平变化中断引脚。
PGC			I/O	ST	在线调试器和 ICSP 编程时钟引脚。
RB7/KBI3/PGD	28	25			
RB7			I/O	TTL	数字 I/O。
KBI3			I	TTL	电平变化中断引脚。
PGD			I/O	ST	在线调试器和 ICSP 编程数据引脚。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

© 2007 Microchip Technology Inc.

表 1-2： PIC18F2450 I/O 引脚说明 （续）

引脚名称	引脚号 PDIP, SOIC	引脚号 QFN	引脚类型	缓冲器类型	说明
					PORTC 是双向 I/O 端口。
RC0/T1OSO/T1CKI	11	8			
RC0			I/O	ST	数字 I/O。
T1OSO			O	—	Timer1 振荡器输出。
T1CKI			I	ST	Timer1 外部时钟输入。
RC1/T1OSI/$\overline{UOE}$	12	9			
RC1			I/O	ST	数字 I/O。
T1OSI			I	CMOS	Timer1 振荡器输入。
$\overline{UOE}$			—	—	外部 USB 收发器 $\overline{OE}$ 输出。
RC2/CCP1	13	10			
RC2			I/O	ST	数字 I/O。
CCP1			I/O	ST	捕捉 1 输入 / 比较 1 输出 /PWM 1 输出。
RC4/D-/VM	15	12			
RC4			I	TTL	数字输入。
D-			I/O	—	USB 差分负信号线 （输入 / 输出）。
VM			I	TTL	外部 USB 收发器 VM 输入。
RC5/D+/VP	16	13			
RC5			I	TTL	数字输入。
D+			I/O	—	USB 差分正信号线 （输入 / 输出）。
VP			O	TTL	外部 USB 收发器 VP 输入。
RC6/TX/CK	17	14			
RC6			I/O	ST	数字 I/O。
TX			O	—	EUSART 异步发送。
CK			I/O	ST	EUSART 同步时钟 （见 RX/DT 引脚信息）。
RC7/RX/DT	18	15			
RC7			I/O	ST	数字 I/O。
RX			I	ST	EUSART 异步接收。
DT			I/O	ST	EUSART 同步数据 （见 TX/CK 引脚信息）。
RE3	—	—	—	—	请参见 $\overline{MCLR}$/VPP/RE3 引脚信息。
VUSB	14	11	O	—	内部 USB 3.3V 稳压器。
VSS	8, 19	5, 16	P	—	逻辑和 I/O 引脚的参考地。
VDD	20	17	P	—	逻辑和 I/O 引脚的正电源。

图注： TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出
CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

表 1-3： PIC18F4450 I/O 引脚说明

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
MCLR/Vpp/RE3	1	18	18			主复位 （输入）或编程电压 （输入）。
MCLR				I	ST	主清零 （复位）输入。此引脚为低电平时，器件复位。
VPP				P		编程电压输入。
RE3				I	ST	数字输入。
OSC1/CLKI	13	32	30			振荡器晶振或外部时钟输入。
OSC1				I	模拟	振荡器晶振或外部时钟源输入。
CLKI				I	模拟	外部时钟源输入。总是与OSC1 引脚功能复用。（见 OSC2/CLKO 引脚信息。）
OSC2/CLKO/RA6	14	33	31			振荡器晶振或时钟输出。
OSC2				O	—	振荡器晶振输出。在晶振振荡器模式下，该引脚与晶振或谐振器相连。
CLKO				O	—	在选择模式下， OSC2 引脚输出 CLKO 振荡信号，该信号是OSC1 引脚上振荡信号的 4 分频，该频率等于指令周期的倒数。
RA6				I/O	TTL	通用 I/O 引脚。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

注 1： 除非 ICPRT 配置位被置 1，否则这些引脚是 “无连接”的。对于 NC/ICPORTS，该引脚是 “无连接”的，除非 ICPRT 被置 1 并且 DEBUG 配置位被清零。

© 2007 Microchip Technology Inc.

表 1-3： PIC18F4450 I/O 引脚说明 （续）

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
						PORTA 是双向 I/O 端口。
RA0/AN0	2	19	19			
RA0				I/O	TTL	数字 I/O。
AN0				I	模拟	模拟输入 0。
RA1/AN1	3	20	20			
RA1				I/O	TTL	数字 I/O。
AN1				I	模拟	模拟输入 1。
RA2/AN2/VREF-	4	21	21			
RA2				I/O	TTL	数字 I/O。
AN2				I	模拟	模拟输入 2。
VREF-				I	模拟	A/D 参考电压 （低电平）输入。
RA3/AN3/VREF+	5	22	22			
RA3				I/O	TTL	数字 I/O。
AN3				I	模拟	模拟输入 3。
VREF+				I	模拟	A/D 参考电压 （高电平）输入。
RA4/T0CKI/RCV	6	23	23			
RA4				I/O	ST	数字 I/O。
T0CKI				I	ST	Timer0 外部时钟输入。
RCV				I	TTL	外部 USB 收发器 RCV 输入。
RA5/AN4/HLVDIN	7	24	24			
RA5				I/O	TTL	数字 I/O。
AN4				I	模拟	模拟输入 4。
HLVDIN				I	模拟	高 / 低压检测输入。
RA6	—	—	—	—	—	请参见 OSC2/CLKO/RA6 引脚信息。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

注 1： 除非 ICPRT 配置位被置 1，否则这些引脚是 “无连接”的。对于 NC/ICPORTS，该引脚是 “无连接”的，除非 ICPRT 被置 1 并且 DEBUG 配置位被清零。

表 1-3： PIC18F4450 I/O 引脚说明 （续）

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
						PORTB 是双向 I/O 端口。 PORTB 在所有输入端都可软件编程为内部弱上拉。
RB0/AN12/INT0	33	9	8			
RB0				I/O	TTL	数字 I/O。
AN12				I	模拟	模拟输入 12。
INT0				I	ST	外部中断 0。
RB1/AN10/INT1	34	10	9			
RB1				I/O	TTL	数字 I/O。
AN10				I	模拟	模拟输入 10。
INT1				I	ST	外部中断 1。
RB2/AN8/INT2/VMO	35	11	10			
RB2				I/O	TTL	数字 I/O。
AN8				I	模拟	模拟输入 8。
INT2				I	ST	外部中断 2。
VMO				O	—	外部 USB 收发器 VMO 输出。
RB3/AN9/VPO	36	12	11			
RB3				I/O	TTL	数字 I/O。
AN9				I	模拟	模拟输入 9。
VPO				O	—	外部 USB 收发器 VPO 输出。
RB4/AN11/KBI0	37	14	14			
RB4				I/O	TTL	数字 I/O。
AN11				I	模拟	模拟输入 11。
KBI0				I	TTL	电平变化中断引脚。
RB5/KBI1/PGM	38	15	15			
RB5				I/O	TTL	数字 I/O。
KBI1				I	TTL	电平变化中断引脚。
PGM				I/O	ST	低压 ICSP™ 编程使能引脚。
RB6/KBI2/PGC	39	16	16			
RB6				I/O	TTL	数字 I/O。
KBI2				I	TTL	电平变化中断引脚。
PGC				I/O	ST	在线调试器和 ICSP 编程时钟引脚。
RB7/KBI3/PGD	40	17	17			
RB7				I/O	TTL	数字 I/O。
KBI3				I	TTL	电平变化中断引脚。
PGD				I/O	ST	在线调试器和 ICSP 编程数据引脚。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

注 1： 除非 ICPRT 配置位被置 1，否则这些引脚是 “无连接”的。对于 NC/ICPORTS，该引脚是 “无连接”的，除非 ICPRT 被置 1 并且 DEBUG 配置位被清零。

© 2007 Microchip Technology Inc.

表 1-3： PIC18F4450 I/O 引脚说明 （续）

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
						PORTC 是双向 I/O 端口。
RC0/T1OSO/T1CKI	15	34	32			
RC0				I/O	ST	数字 I/O。
T1OSO				O	—	Timer1 振荡器输出。
T1CKI				I	ST	Timer1 外部时钟输入。
RC1/T1OSI/$\overline{UOE}$	16	35	35			
RC1				I/O	ST	数字 I/O。
T1OSI				I	CMOS	Timer1 振荡器输入。
$\overline{UOE}$				O	—	外部 USB 收发器 $\overline{OE}$ 输出。
RC2/CCP1	17	36	36			
RC2				I/O	ST	数字 I/O。
CCP1				I/O	ST	捕捉 1 输入 / 比较 1 输出 /PWM 1 输出。
RC4/D-/VM	23	42	42			
RC4				I	TTL	数字输入。
D-				I/O	—	USB 差分负信号线 （输入 / 输出）。
VM				I	TTL	外部 USB 收发器 VM 输入。
RC5/D+/VP	24	43	43			
RC5				I	TTL	数字输入。
D+				I/O	—	USB 差分正信号线 （输入 / 输出）。
VP				I	TTL	外部 USB 收发器 VP 输入。
RC6/TX/CK	25	44	44			
RC6				I/O	ST	数字 I/O。
TX				O	—	EUSART 异步发送。
CK				I/O	ST	EUSART 同步时钟 （见 RX/DT 引脚信息）。
RC7/RX/DT	26	1	1			
RC7				I/O	ST	数字 I/O。
RX				I	ST	EUSART 异步接收。
DT				I/O	ST	EUSART 同步数据 （见 TX/CK 引脚信息）。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

注 1： 除非 ICPRT 配置位被置 1，否则这些引脚是 “无连接” 的。对于 NC/ICPORTS，该引脚是 “无连接” 的，除非 ICPRT 被置 1 并且 $\overline{DEBUG}$ 配置位被清零。

表 1-3： PIC18F4450 I/O 引脚说明 （续）

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
						PORTD 是双向 I/O 端口。
RD0	19	38	38	I/O	ST	数字 I/O。
RD1	20	39	39	I/O	ST	数字 I/O。
RD2	21	40	40	I/O	ST	数字 I/O。
RD3	22	41	41	I/O	ST	数字 I/O。
RD4	27	2	2	I/O	ST	数字 I/O。
RD5	28	3	3	I/O	ST	数字 I/O。
RD6	29	4	4	I/O	ST	数字 I/O。
RD7	30	5	5	I/O	ST	数字 I/O。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

注 1： 除非 ICPRT 配置位被置 1，否则这些引脚是 “无连接”的。对于 NC/ICPORTS，该引脚是 “无连接”的，除非 ICPRT 被置 1 并且 DEBUG 配置位被清零。

© 2007 Microchip Technology Inc.

表 1-3： PIC18F4450 I/O 引脚说明 （续）

引脚名称	引脚号			引脚类型	缓冲器类型	说明
	PDIP	QFN	TQFP			
RE0/AN5	8	25	25			PORTE 是双向 I/O 端口。
RE0				I/O	ST	数字 I/O。
AN5				I	模拟	模拟输入 5。
RE1/AN6	9	26	26			
RE1				I/O	ST	数字 I/O。
AN6				I	模拟	模拟输入 6。
RE2/AN7	10	27	27			
RE2				I/O	ST	数字 I/O。
AN7				I	模拟	模拟输入 7。
RE3	—	—	—	—	—	请参见 $\overline{MCLR}$/VPP/RE3 引脚信息。
VSS	12, 31	6, 30, 31	6, 29	P	—	逻辑和 I/O 引脚的参考地。
VDD	11, 32	7, 8, 28, 29	7, 28	P	—	逻辑和 I/O 引脚的正电源。
VUSB	18	37	37	O	—	内部 USB 3.3V 稳压器输出。
NC/ICCK/ICPGC(1)	—	—	12			无连接或专用 ICD/ICSP™ 端口时钟。
ICCK				I/O	ST	在线调试器时钟。
ICPGC				I/O	ST	ICSP 编程时钟。
NC/ICDT/ICPGD(1)	—	—	13			无连接或专用 ICD/ICSP 端口数据。
ICDT				I/O	ST	在线调试器数据。
ICPGD				I/O	ST	ICSP 编程数据。
NC/$\overline{ICRST}$/ICVPP(1)	—	—	33			无连接或专用 ICD/ICSP 端口复位。
$\overline{ICRST}$				I	—	主清零 （复位）输入。
ICVPP				P	—	编程电压输入。
NC/ICPORTS(1)	—	—	34	P	—	无连接或 28 引脚器件仿真。
ICPORTS						连接到 VSS 时使能 28 引脚器件仿真。
NC	—	13	—	—	—	无连接。

图注： TTL = TTL 兼容输入　　CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入　　I = 输入
O = 输出　　P = 电源

注 1： 除非 ICPRT 配置位被置 1，否则这些引脚是 “无连接”的。对于 NC/ICPORTS，该引脚是 “无连接”的，除非 ICPRT 被置 1 并且 $\overline{DEBUG}$ 配置位被清零。

注：

 © 2007 Microchip Technology Inc.

2.0 振荡器配置

2.1 概述

PIC18F2450/4450 系列器件的振荡器和单片机时钟系统不同于非 USB PIC18F 器件的。增加的 USB 模块特别需要一个稳定时钟源，因此必须为其提供同时符合 USB 低速和全速规范的独立时钟源。

为满足这些要求， PIC18F2450/4450 器件包含了新的时钟分支，为全速 USB 操作提供 48 MHz 时钟。由于它是从主时钟源取得的，所以添加了额外的预分频器和后分频器，以便产生范围更广的振荡器频率。图 2-1 中简要说明了振荡器的结构。

PIC18 增强型单片机中使用的其他振荡器功能（例如内部 RC 振荡器和时钟切换）仍保持不变。它们将在本章中稍后讨论。

2.1.1 振荡器控制

PIC18F2450/4450 器件中振荡器的运行通过两个配置寄存器和两个控制寄存器控制。配置寄存器 CONFIG1L 和 CONFIG1H 选择振荡器模式和 USB 预分频器 / 后分频器选项。作为配置位，它们在器件编程时被设置，并保留该配置直到对器件重新编程。

OSCCON 寄存器 （寄存器 2-1）选择 “有效时钟”模式；它主要用于控制功耗管理模式中的时钟切换。它的使用在**第 2.4.1 节 “振荡器控制寄存器”**中讨论。

2.2 振荡器类型

PIC18F2450/4450 器件可以在 12 种不同的振荡器模式下工作。与非 USB PIC18 增强型单片机形成对比的是，这些模式中的 4 种牵涉到同时使用两种振荡器类型。用户可以设定 FOSC3:FOSC0 配置位以选择以下模式之一：

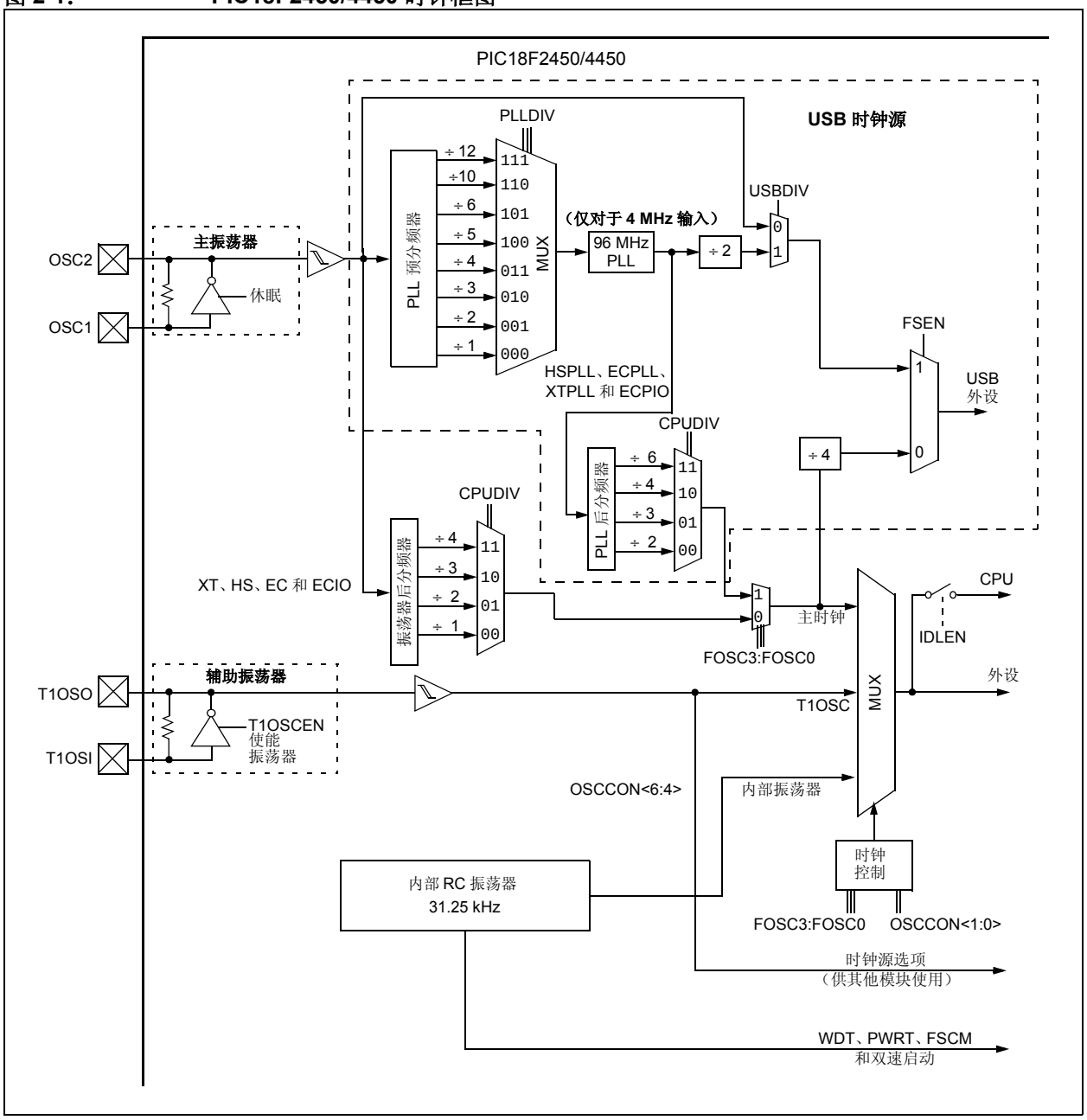
1. XT 晶振 / 谐振器模式
2. XTPLL 使能 PLL 的晶振 / 谐振器模式
3. HS 高速晶振 / 谐振器模式
4. HSPLL 使能 PLL 的高速晶振 / 谐振器模式
5. EC 带 FOSC/4 输出的外部时钟模式
6. ECIO RA6 用作 I/O 引脚的外部时钟模式
7. ECPLL 使能 PLL 的外部时钟模式，通过 RA6 引脚输出 FOSC/4 信号
8. ECPIO 使能 PLL 的外部时钟模式，RA6 作为 I/O 引脚
9. INTHS 内部振荡器用作单片机时钟源， HS 振荡器用作 USB 时钟源
10. INTXT 内部振荡器用作单片机时钟源， XT 振荡器用作 USB 时钟源
11. INTIO 内部振荡器用作单片机时钟源， EC 振荡器用作 USB 时钟源，RA6 引脚用作数字 I/O
12. INTCKO 内部振荡器用作单片机时钟源， EC 振荡器用作 USB 时钟源，通过 RA6 引脚输出 FOSC/4 信号

2.2.1 振荡器模式和 USB 操作

由于 USB 模块的独特要求，需要不同的时钟工作方式。在早期的 PIC® 器件中，所有内核和外设时钟都是由同一个振荡器源驱动的；通常的来源是主振荡器、辅助振荡器或内部振荡器。对 PIC18F2450/4450 器件，主振荡器成为 USB 模块的一部分，无法关联到任何其他时钟源。因此，USB 模块必须从主时钟源获得时钟；但是，单片机内核与其他外设可以独立地从辅助或内部振荡器获得时钟，就像以前一样。

由于 USB 特定的时序要求，使能 USB 模块时，必须使用 6 MHz 或 48 MHz 内部时钟。幸运的是，使用主振荡器时，单片机和其他外设无需以该时钟频率运行。有许多选项可满足 USB 模块的时钟要求，同时仍由主振荡器源为器件的其他部分提供灵活的时钟。**第2.3节“USB 的振荡器设置”**详细描述了这些功能。

图 2-1： PIC18F2450/4450 时钟框图



 © 2007 Microchip Technology Inc.

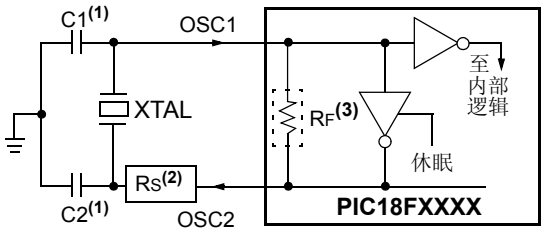
2.2.2 晶振 / 陶瓷谐振器

在 HS、HSPLL、XT 或 XTPLL 振荡器模式下，晶振或陶瓷谐振器与 OSC1 和 OSC2 引脚相连来产生振荡信号。图 2-2 显示了引脚连接方式。

振荡器的设计要求使用平行切割的晶体。

注： 使用顺序切割的晶体，可能会使振荡器产生的频率超出晶体制造厂商所给的参数范围。

图 2-2： 晶振 / 陶瓷谐振器工作原理（XT、HS 或 HSPLL 配置）



注 1： 关于 C1 和 C2 的初始值，请参见表 2-1 和表 2-2。

2： 对于 AT 条形切割的晶体可能会需要一个串联电阻（RS）。

3： RF 的值随选定的振荡器模式变化。

表 2-1： 陶瓷谐振器的电容选择

使用的典型电容值：			
模式	**频率**	**OSC1**	**OSC2**
XT	4.0 MHz	33 pF	33 pF
HS	8.0 MHz 16.0 MHz	27 pF 22 pF	27 pF 22 pF

上述电容值仅供设计参考。

使用下面列出的谐振器在基本启动和运行过程中对这些电容值作了测试。**这些值未经优化。**

要得到理想的振荡器工作状况，可能需要不同的电容值。用户应当在设计的 VDD 和温度条件下测试振荡器的性能。

更多信息，请参见表 2-2 下方的“注”。

使用的谐振器：
4.0 MHz
8.0 MHz
16.0 MHz

表 2-2： 晶振的电容选择

振荡类型	**晶振频率**	**已测试的典型电容值：**	
		C1	**C2**
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

上述电容值仅供设计参考。

已使用下面列出的晶振在基本的启动和运行过程中对这些电容值作了测试。**这些值未经优化。**

要得到理想的振荡器工作状况，可能需要不同的电容值。用户应当在设计的 VDD 和温度条件下测试振荡器的性能。

更多信息，请参见本表下方的“注”。

使用的晶振：
4 MHz
8 MHz
20 MHz

注 1： 电容值越大，振荡器的稳定性越高，但同时起振时间也越长。

2： 当工作电压 VDD 低于 3V，或在任何电压下使用某些陶瓷谐振器时，可能需要使用 HS 振荡器模式或切换到晶振模式。

3： 因为每种谐振器 / 晶振都有其自身特性，用户应当向谐振器 / 晶振制造厂商询问外部元件的适当值。

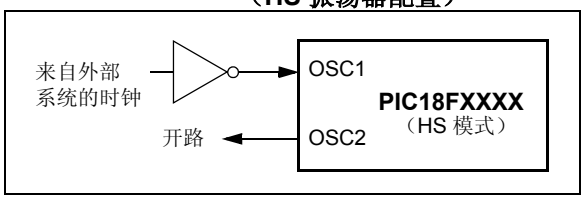
4： 可能需要使用电阻 Rs 以避免对低驱动规格的晶体造成过驱动。

5： 请始终在设计的 VDD 和温度范围下验证振荡器性能。

内部后分频器允许用户选择晶振或谐振器之外的时钟频率。分频是由 CPUDIV 配置位决定的。用户选择的时钟频率可以是振荡器频率，或该频率的 1/2、 1/3 或 1/4。

单片机处于 HS 振荡器模式时，也可使用外部时钟。在这种情况下， OSC2/CLKO 引脚保持开路状态（图 2-3）。

图 2-3：　　外部时钟输入工作原理（HS 振荡器配置）

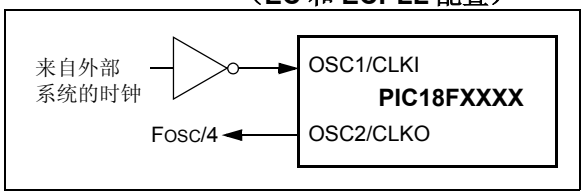


2.2.3　外部时钟输入

EC、ECIO、ECPLL 和 ECPIO 振荡器模式要求 OSC1 引脚与一个外部时钟源相连。在上电复位后或从休眠模式退出后，不需要振荡器起振时间。

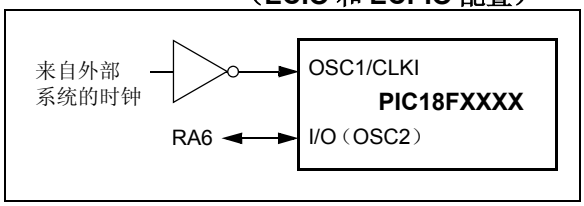
在 EC 和 ECPLL 振荡器模式下，由 OSC2 引脚输出振荡器频率的 4 分频信号。此信号可用于测试或同步其他逻辑。图 2-4 显示了 EC 振荡器模式的引脚连接方式。

图 2-4：　　外部时钟输入工作原理（EC 和 ECPLL 配置）



ECIO 和 ECPIO 振荡器模式的工作方式类似于 EC 和 ECPLL 模式，不同之处在于 OSC2 引脚变成了一个额外的通用 I/O 引脚。该 I/O 引脚成为 PORTA 的 bit 6（RA6）。图 2-5 显示了 ECIO 振荡器模式下的引脚连接方式。

图 2-5：　　外部时钟输入工作原理（ECIO 和 ECPIO 配置）



用于降低 XT 和 HS 模式中的时钟频率的内部后分频器也可在 EC 和 ECIO 模式中使用。

2.2.4　PLL 倍频器

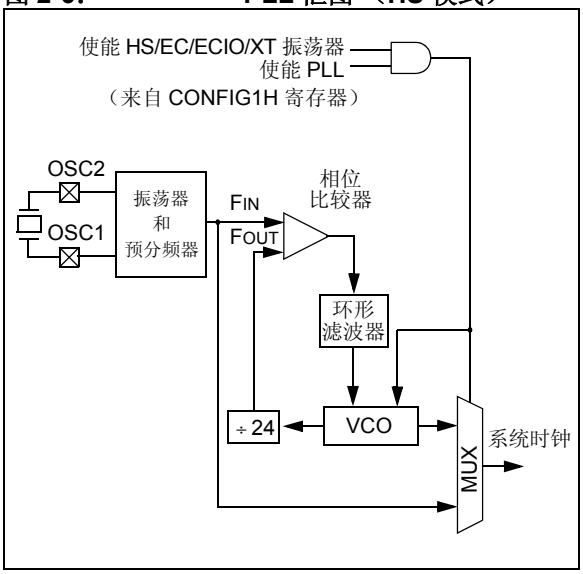
PIC18F2450/4450 器件包含锁相环（PLL）电路。这是专为低速振荡器的 USB 应用提供的，也可用作单片机时钟源。

PLL 在 HSPLL、XTPLL、ECPLL 和 ECPIO 振荡器模式中使能。它设计为从固定的 4 MHz 输入信号产生固定的 96 MHz 参考时钟。输出随后可进行分频，用于 USB 和单片机内核时钟。由于 PLL 具有固定频率的输入和输出，有 8 个预分频选项用来匹配振荡器到 PLL 的输入频率。

还有独立的后分频器选项，用于从 PLL 得到单片机时钟。这就允许 USB 外设和单片机使用同一个振荡器输入，却在不同的时钟速率下工作。与 XT、HS 和 EC 模式的后分频器形成对比的是，可用选项是 PLL 输出的 1/2、1/3、1/4 和 1/6。

HSPLL、ECPLL 和 ECPIO 模式使用 HS 模式振荡器产生最高 48 MHz 的频率。预分频器将振荡器输入最多进行 12 分频，产生 4 MHz 信号驱动 PLL。XTPLL 模式只能使用直接驱动 PLL 的 4 MHz 输入频率。

图 2-6：　　PLL 框图（HS 模式）



© 2007 Microchip Technology Inc.

2.2.5 内部振荡器

PIC18F2450/4450 器件包含一个内部 RC 振荡器（INTRC），它提供了标称值为 31 kHz 的输出。如果选择 INTRC 作为器件的时钟源，它就会被使能；当使能以下任一功能时，也将自动使能 INTRC：

- 上电延时定时器
- 故障保护时钟监视器
- 看门狗定时器
- 双速启动

第 18.0 节 “CPU 的特殊功能”将详细讨论以上功能。

2.2.5.1 内部振荡器模式

内部振荡器用作单片机时钟源时，必须使用其他振荡器（外部时钟或外部晶振/谐振器）作为USB时钟源。USB时钟源的选择由特定内部振荡器模式决定。

有 4 种不同模式可用：

1. INTHS 模式：由 HS 模式下的振荡器提供 USB 时钟。
2. INTXT 模式：由 XT 模式下的振荡器提供 USB 时钟。
3. INTCKO 模式：OSC1/CLKI 上的外部时钟输入提供 USB 时钟； OSC2/CLKO 引脚输出 FOSC/4。
4. INTIO 模式：OSC1/CLKI 上的外部时钟输入提供 USB 时钟； OSC2/CLKO 引脚用作数字 I/O（RA6）。

在这 4 种模式中，只有 INTIO 模式释放引脚 （OSC2/CLKO/RA6）用作端口 I/O。

2.3 USB 的振荡器设置

当使用 PIC18F2450/4450 USB 连接时，必须有 6 MHz 或 48 MHz 时钟用于 USB 操作，具体取决于使用的是低速还是全速模式。这需要在选择振荡器频率和对器件编程时预先计划好。

表 2-3 中给出了与 USB 操作兼容的所有可能的振荡器配置。

2.3.1 低速操作

低速模式的 USB 时钟来自主振荡器链，而不是直接来自 PLL。它被四分频，产生实际的 6 MHz 时钟。因此，USB 模块激活时，单片机只能使用 24 MHz 时钟频率，而控制器时钟源是一种主振荡器模式 （带有或不带有 PLL 的 XT、 HS 或 EC）。

如果单片机时钟源是辅助振荡器或内部振荡器，这一限制不适用。

2.3.2 运行不同的 USB 和单片机时钟

USB 模块无论在哪种模式下都可以与单片机内核及其他外设异步运行。这意味着可使用主振荡器作为 USB 时钟，而单片机运行在低速的独立时钟源下。如果整个应用必须使用一个时钟源运行，全速操作模式为单片机时钟频率提供了更多选择。

表 2-3： USB 操作的振荡器配置选项

输入振荡器频率	PLL 分频（PLLDIV2:PLLDIV0）	时钟模式（FOSC3:FOSC0）	MCU 时钟分频（CPUDIV1:CPUDIV0）	单片机时钟频率
48 MHz	N/A(1)	EC 和 ECIO	无（00）	48 MHz
			÷2（01）	**24 MHz**
			÷3（10）	16 MHz
			÷4（11）	12 MHz
48 MHz	÷12（111）	EC 和 ECIO	无（00）	48 MHz
			÷2（01）	**24 MHz**
			÷3（10）	16 MHz
			÷4（11）	12 MHz
		ECPLL 和 ECPIO	÷2（00）	48 MHz
			÷3（01）	32 MHz
			÷4（10）	**24 MHz**
			÷6（11）	16 MHz
40 MHz	÷10（110）	EC 和 ECIO	无（00）	40 MHz
			÷2（01）	20 MHz
			÷3（10）	13.33 MHz
			÷4（11）	10 MHz
		ECPLL 和 ECPIO	÷2（00）	48 MHz
			÷3（01）	32 MHz
			÷4（10）	**24 MHz**
			÷6（11）	16 MHz
24 MHz	÷6（101）	HS、EC 和 ECIO	无（00）	**24 MHz**
			÷2（01）	12 MHz
			÷3（10）	8 MHz
			÷4（11）	6 MHz
		HSPLL、ECPLL 和 ECPIO	÷2（00）	48 MHz
			÷3（01）	32 MHz
			÷4（10）	**24 MHz**
			÷6（11）	16 MHz
20 MHz	÷5（100）	HS、EC 和 ECIO	无（00）	20 MHz
			÷2（01）	10 MHz
			÷3（10）	6.67 MHz
			÷4（11）	5 MHz
		HSPLL、ECPLL 和 ECPIO	÷2（00）	48 MHz
			÷3（01）	32 MHz
			÷4（10）	**24 MHz**
			÷6（11）	16 MHz
16 MHz	÷4（011）	HS、EC 和 ECIO	无（00）	16 MHz
			÷2（01）	8 MHz
			÷3（10）	5.33 MHz
			÷4（11）	4 MHz
		HSPLL、ECPLL 和 ECPIO	÷2（00）	48 MHz
			÷3（01）	32 MHz
			÷4（10）	**24 MHz**
			÷6（11）	16 MHz

图注： 所有时钟频率（除了 24 MHz）都只与全速 USB 操作（48 MHz 的 USB 时钟）模式相关。**粗体**用于强调与低速 USB 操作（系统时钟 24 MHz，USB 时钟 6 MHz）兼容的时钟选择。

注 1： 仅在 USBDIV 配置位清零时有效。

© 2007 Microchip Technology Inc.

表 2-3： USB 操作的振荡器配置选项 （续）

输入振荡器频率	PLL 分频 （PLLDIV2:PLLDIV0）	时钟模式 （FOSC3:FOSC0）	MCU 时钟分频 （CPUDIV1:CPUDIV0）	单片机时钟频率
12 MHz	÷3 （010）	HS、 EC 和 ECIO	无 （00）	12 MHz
			÷2 （01）	6 MHz
			÷3 （10）	4 MHz
			÷4 （11）	3 MHz
		HSPLL、 ECPLL 和 ECPIO	÷2 （00）	48 MHz
			÷3 （01）	32 MHz
			÷4 （10）	**24 MHz**
			÷6 （11）	16 MHz
8 MHz	÷2 （001）	HS、 EC 和 ECIO	无 （00）	8 MHz
			÷2 （01）	4 MHz
			÷3 （10）	2.67 MHz
			÷4 （11）	2 MHz
		HSPLL、 ECPLL 和 ECPIO	÷2 （00）	48 MHz
			÷3 （01）	32 MHz
			÷4 （10）	**24 MHz**
			÷6 （11）	16 MHz
4 MHz	÷1 （000）	XT、 HS、 EC 和 ECIO	无 （00）	4 MHz
			÷2 （01）	2 MHz
			÷3 （10）	1.33 MHz
			÷4 （11）	1 MHz
		HSPLL、 ECPLL、 XTPLL 和 ECPIO	÷2 （00）	48 MHz
			÷3 （01）	32 MHz
			÷4 （10）	**24 MHz**
			÷6 （11）	16 MHz

图注： 所有时钟频率 （除了 24 MHz）都只与全速 USB 操作 （48 MHz 的 USB 时钟）模式相关。**粗体**用于强调与低速 USB 操作 （系统时钟 24 MHz， USB 时钟 6 MHz）兼容的时钟选择。

注 1： 仅在 USBDIV 配置位清零时有效。

2.4 时钟源与振荡器切换

与早期的 PIC18 增强型器件一样， PIC18F2450/4450 系列包含允许将器件时钟源从主振荡器切换到备用低频时钟源的功能。PIC18F2450/4450 器件提供了两个备用时钟源。当使能备用时钟源时，可以使用多种功耗管理工作模式。

基本上，这些器件都有 3 种时钟源：

- 主振荡器
- 辅助振荡器
- 内部振荡器

主振荡器包括外部晶振和谐振器模式、外部时钟模式和内部振荡器。特定的模式由 FOSC3:FOSC0 配置位定义。这些模式的详细信息已在本章前面的内容中作过介绍。

辅助振荡器是指那些不与 OSC1 或 OSC2 引脚连接的外部时钟源。即使在控制器处于功耗管理模式时这些时钟源仍可继续工作。

PIC18F2450/4450 器件将 Timer1 振荡器作为辅助振荡器。此振荡器（在所有功耗管理模式中）通常是实时时钟等功能的时基。大部分情况下，在 RC0/T1OSO/T1CKI 和 RC1/T1OSI/UOE 引脚之间接有一个 32.768 kHz 的时钟晶振。与 XT 和 HS 振荡器模式电路类似，在每个引脚与地之间均接有负载电容。将在**第 11.3 节“Timer1 振荡器”**中详细讨论 Timer1 振荡器。

除了作为主时钟源之外，**内部振荡器**还可以作为功耗管理模式的时钟源。INTRC 源也可作为几种特殊功能部件（例如 WDT 和故障保护时钟监视器）的时钟源。

2.4.1 振荡器控制寄存器

OSCCON 寄存器 （寄存器 2-1）控制全功耗模式和功耗管理模式下器件时钟工作的多个方面。

系统时钟选择位 SCS1:SCS0 用于选择时钟源。可用的时钟源包括主时钟 （由 FOSC3:FOSC0 配置位定义）、辅助时钟（Timer1 振荡器）和内部振荡器。当写入一个或多个位之后，接着是一段很短的时钟转换间隔，然后时钟源会立即改变。在所有形式的复位中 SCS 位都会被清零。

INTRC 总是作为看门狗定时器和故障保护时钟监视器之类部件的时钟源。

OSTS 和 T1RUN 位指出当前提供器件时钟的是哪一个时钟源。 OSTS 位置 1 表明振荡器起振定时器已超时且主时钟在主时钟模式下作为器件时钟。 T1RUN 位（T1CON<6>）置 1 表明 Timer1 振荡器正在辅助时钟模式下提供器件时钟。在功耗管理模式下，任何时候这 3 个位中只有一个会置 1。如果这些位都没有置 1，则表示当前时钟源是 INTRC，或内部振荡器刚刚起振且尚未稳定。

IDLEN 位决定当执行 SLEEP 指令时器件是进入休眠模式还是某个空闲模式。

第 3.0 节“功耗管理模式”更详细地讨论了 OSCCON 寄存器中标志位和控制位的使用。

注	**1：** 要选择辅助时钟源，必须使能 Timer1 振荡器。通过将 Timer1 控制寄存器中的 T1OSCEN 位 （T1CON<3>）置 1，可以使能 Timer1 振荡器。如果未使能 Timer1 振荡器，则选择辅助时钟源的任何尝试都会被忽略。
	2： 建议在Timer1振荡器稳定工作之后再选择它为时钟源，否则当Timer1 振荡器起振时可能会发生很长的延时。

2.4.2 振荡器转换

PIC18F2450/4450 器件包含了防止在切换时钟源时发生时钟 “毛刺”的电路。在切换时钟时，系统时钟会有短暂的停顿。该停顿的时间长度是旧时钟源的两个周期加上新时钟源的三到四个周期的和。此公式假设新时钟源是稳定的。

第 3.1.2 节“进入功耗管理模式”详细讨论了时钟转换。

© 2007 Microchip Technology Inc.

寄存器 2-1：　　OSCCON：振荡器控制寄存器

R/W-0	U-0	U-0	U-0	R(1)	U-0	R/W-0	R/W-0
IDLEN	—	—	—	OSTS	—	SCS1	SCS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　　**IDLEN**：空闲使能位
1 = 执行 SLEEP 指令后器件进入空闲模式
0 = 执行 SLEEP 指令后器件进入休眠模式

bit 6-4　　**未用**：读为 0

bit 3　　**OSTS**：振荡器起振延时状态位 (1)
1 = 振荡器起振定时器延时已结束；主振荡器正在运行
0 = 振荡器起振定时器延时正在进行；主振荡器尚未就绪

bit 2　　**未用**：读为 0

bit 1-0　　**SCS1:SCS0**：系统时钟选择位
1x = 内部振荡器
01 = Timer1 振荡器
00 = 主振荡器

注　1：取决于 IESO 配置位的状态。

2.5 功耗管理模式对各种时钟源的影响

当选定了 PRI_IDLE 模式后，指定的主振荡器会继续运行而不中断。对于所有其他功耗管理模式，使用 OSC1 引脚的振荡器会被禁止。除非使能了 USB 模块，否则 OSC1 引脚（以及由振荡器使用的 OSC2 引脚）将会停止振荡。

在辅助时钟模式下（SEC_RUN和SEC_IDLE），Timer1 振荡器作为器件时钟源工作。如果需要，Timer1 振荡器也可以运行在所有功耗管理模式下为Timer1提供时钟。

在内部振荡器下 （RC_RUN 和 RC_IDLE），由内部振荡器模块提供器件时钟。无论是哪种功耗管理模式，31 kHz 的 INTRC 输出均可被直接用来提供时钟并且可被使能来支持多种特殊的功能部件 （关于 WDT、故障保护时钟监视器和双速启动的更多信息，请参见**第 18.2 节“看门狗定时器（WDT）”**、**第 18.3 节“双速启动”**和**第 18.4 节 “故障保护时钟监视器”**）。

无论选择运行还是空闲模式， USB 时钟源都将继续运行。如果器件使用的是基于晶振或谐振器的振荡器，振荡器将继续为 USB 模块提供时钟。内核和所有其他模块都将切换到新的时钟源。

如果选择了休眠模式，所有的时钟源都会被停止。因为休眠模式切断了所有晶体管的开启电流，休眠模式能实现最小的器件电流消耗 （仅泄漏电流）。

USB 模块在运行并连接时，不能进入休眠模式。唯一的例外是器件已通过 USB 发出了 “暂停”命令。一旦模块暂停运行并切换到低功耗状态，单片机就可以安全地进入休眠模式。

在休眠期间使能任何片上功能都将增加休眠时的电流消耗。要支持 WDT 工作，需要使能 INTRC。Timer1 振荡器可以用来为实时时钟提供时钟源。不需要器件时钟源的其他功能部件也可以工作（即，PSP、INTn引脚等）。在**第 21.2 节 “直流特性：掉电和供电电流”**中列出了可能显著增加电流消耗的外设。

2.6 上电延时

由两个定时器控制上电延时，这样大多数应用都无需外接复位电路。上电延时可以确保在器件电源稳定 （常规环境下）和主时钟稳定工作之前器件保持在复位状态。关于上电延时的更多信息，请参见**第 4.5 节 “器件复位定时器”**。

第一个定时器是上电延时定时器 （PWRT），在上电时它提供了固定的延迟时间（表 21-10 中的参数 33）。通过清零 （= 0） PWRTEN 配置位可使能它。

第二个定时器是振荡器起振定时器 （OST），用于在晶振稳定前使芯片保持在复位状态 （XT 和 HS 模式）。OST 在计数 1024 个振荡周期后允许振荡器为器件提供时钟。

当选定 HSPLL 振荡器模式时，器件将在 HS 模式下的 OST 延时之后另外保持 2 ms 的复位状态，这样可使 PLL 锁定输入时钟频率。

POR之后有一个TCSD间隔的延时（表21-10中的参数38），在延时期间控制器为执行指令做准备。此延时与其他延时并行发生。当将 EC 或内部振荡器模式之一用作主时钟源时，这可能是唯一的延时。

表 2-4： 休眠模式下 OSC1 和 OSC2 引脚的状态

振荡器模式	OSC1 引脚	OSC2 引脚
INTCKO	悬空，由外部时钟驱动	处于逻辑低电平 （时钟 /4 输出）
INTIO	悬空，由外部时钟驱动	配置为 PORTA 的 bit 6
ECIO 和 ECPIO	悬空，由外部时钟驱动	配置为 PORTA 的 bit 6
EC	悬空，由外部时钟驱动	处于逻辑低电平 （时钟 /4 输出）
XT 和 HS	反馈反相器被禁止，处于静止电平	反馈反相器被禁止，处于静止电平

注： 关于由休眠和 MCLR 复位引起的延时，请参见**第 4.0 节 “复位”**中的表 4-2。

 © 2007 Microchip Technology Inc.

3.0 功耗管理模式

PIC18F2450/4450 器件总共提供 7 种工作模式，可以更有效地进行功耗管理。这些工作模式提供了多种选择，可在资源受限的应用（即，电池供电的设备）中节省功耗。

功耗管理模式有三种类别：

- 运行模式
- 空闲模式
- 休眠模式

这些类别定义了需要为器件的哪些部分提供时钟，有时还需要定义时钟的速度。运行和空闲模式可以使用三种时钟源（主时钟源、辅助时钟源或内部振荡器）中的任意一种；而休眠模式则不使用时钟源。

功耗管理模式包括几个由早期的 PIC® 器件提供的节省功耗的功能。其中之一就是其他 PIC18 器件也提供的时钟切换功能，该功能允许使用 Timer1 振荡器代替主振荡器。节省功耗的功能还包括所有 PIC 器件都提供的休眠模式，在该模式下，器件所有的时钟都停止。

3.1 选择功耗管理模式

选择功耗管理模式之前需要先做出两个决定：是否为 CPU 提供时钟源以及选择何种时钟源。IDLEN 位（OSCCON<7>）控制是否为 CPU 提供时钟源，而 SCS1:SCS0 位（OSCCON<1:0>）选择时钟源。表 3-1 总结了各个模式下的位设置、时钟源和受影响的模块。

3.1.1 时钟源

SCS1:SCS0 位允许为功耗管理模式在三个时钟源中任选其一。它们是：

- 主时钟，由 FOSC3:FOSC0 配置位定义
- 辅助时钟 （Timer1 振荡器）
- 内部振荡器 （用于 RC 模式）

3.1.2 进入功耗管理模式

可以通过装载 OSCCON 寄存器从一种功耗管理模式切换到另一种功耗管理模式。 SCS1:SCS0 位选择时钟源并确定使用运行模式还是空闲模式。更改这些位会导致立即切换到一个新的时钟源 （假定新时钟源正在运行）。此切换可能会引起时钟转换延时。**第 3.1.3 节“时钟转换和状态指示”**及其后续章节将会讨论这些问题。

执行SLEEP指令可以触发进入功耗管理空闲模式或休眠模式。最后实际进入哪个模式由 IDLEN 状态位决定。

更改功耗管理模式并不总是要求设置所有的位，而是取决于当前的模式和将要切换到的模式。通过在发出SLEEP指令之前更改振荡器选择位或更改IDLEN位可完成多种模式转换。如果已经正确配置了 IDLEN 位，可能只需执行 SLEEP 指令就可实现模式切换。

表 3-1： 功耗管理模式

模式	OSCCON 位		模块时钟		可用时钟和振荡器源
	IDLEN(1)	SCS1:SCS0	CPU	外设	
休眠	0	N/A	关闭	关闭	无——所有时钟被禁止
PRI_RUN	N/A	00	提供时钟	提供时钟	主时钟—— 所有振荡器模式。这是正常的全功耗执行模式。
SEC_RUN	N/A	01	提供时钟	提供时钟	辅助时钟——Timer1 振荡器
RC_RUN	N/A	1x	提供时钟	提供时钟	内部振荡器 (2)
PRI_IDLE	1	00	关闭	提供时钟	主时钟——所有振荡器模式
SEC_IDLE	1	01	关闭	提供时钟	辅助时钟——Timer1 振荡器
RC_IDLE	1	1x	关闭	提供时钟	内部振荡器 (2)

注 1： IDLEN 反映它在执行 SLEEP 指令时的值。

2： 时钟是 INTRC 源。

3.1.3 时钟转换和状态指示

在两个时钟源之间进行转换所需的时间长度是旧时钟源的两个周期与新时钟源的三到四个周期的和。此公式假设新时钟源是稳定的。

以下两位用于指明当前的时钟源及其状态。它们是：

• OSTS （OSCCON<3>）

• T1RUN （T1CON<6>）

一般来说，在一个给定的功耗管理模式中，这两个位中只有一个位会置 1。当 OSTS 位置 1 时，表明由主时钟提供器件时钟。当 T1RUN 位置 1 时，表明由 Timer1 振荡器提供时钟源。

注：	执行 SLEEP 指令并不一定会将器件置于休眠模式。它只是作为触发条件，让器件进入休眠模式或一种空闲模式，具体何种模式由 IDLEN 位的设置决定。

3.1.4 多条 SLEEP 命令

使用 SLEEP 指令调用功耗管理模式时，具体进入何种模式在该指令执行那一刻由 IDLEN 位的设置决定。如果执行了另一条 SLEEP 指令，器件将进入由此时 IDLEN 位指定的功耗管理模式。如果 IDLEN 位已更改，器件将进入由新的设置指定的新的功耗管理模式。

3.2 运行模式

在运行模式中，内核和外设的时钟都是激活的。这些运行模式之间的区别就在于时钟源的不同。

3.2.1 PRI_RUN 模式

PRI_RUN 模式是单片机的正常全功耗执行模式。除非使能了双速启动 （详细信息，请参见**第 18.3 节 “双速启动”**），该模式也是器件复位后的默认模式。在此模式下， OSTS 位置 1。

3.2.2 SEC_RUN 模式

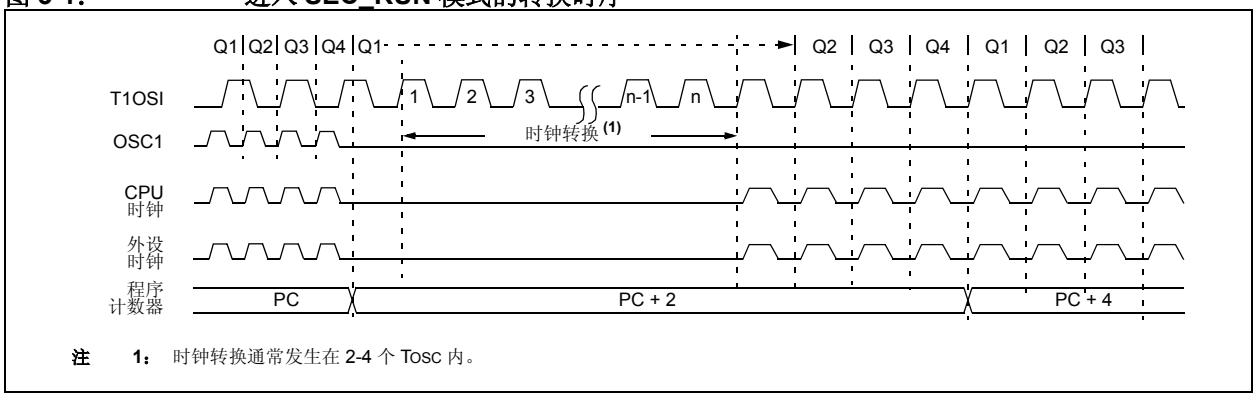
SEC_RUN 模式与其他 PIC18 器件提供的“时钟切换”功能兼容。在此模式下， CPU 和外设将 Timer1 振荡器作为时钟源。这允许用户在使用高精度时钟源的情况下仍可获得较低的功耗。

通过将 SCS1:SCS0 位置为 01 进入 SEC_RUN 模式。器件时钟源被切换到 Timer1 振荡器（见图 3-1），主振荡器被关闭， T1RUN 位 （T1CON<6>）被置 1 并且 OSTS 位被清零。

注：	Timer1 振荡器应该在进入 SEC_RUN 模式之前就已经运行了。如果在SCS1:SCS0位被置为 01 时 T1OSCEN 位没有置 1，就不会进入 SEC_RUN 模式。如果 Timer1 振荡器已经被使能，但没有开始运行，器件时钟将会延时直到该振荡器起振。在这种情况下，最初的振荡器运行很不稳定，可能会导致无法预料的结果。

在从 SEC_RUN 模式转换到 PRI_RUN 模式期间，外设和 CPU 继续使用 Timer1 振荡器作为时钟源，直到主时钟启动。当主时钟准备好以后，时钟切换回主时钟（见图 3-2）。当时钟切换完成后，T1RUN 位被清零，OSTS 位被置 1 并且由主时钟提供器件时钟。这种唤醒不会影响 IDLEN 和 SCS 位。 Timer1 振荡器继续运行。

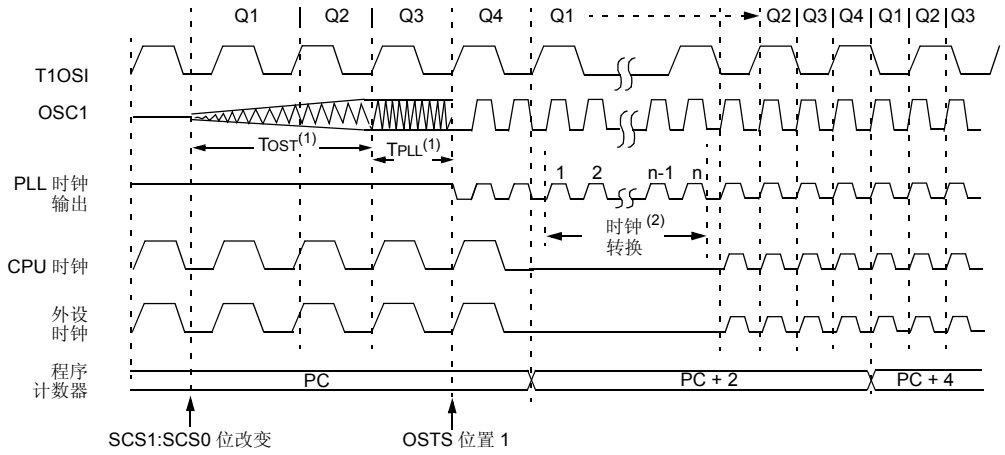
图 3-1： 进入 SEC_RUN 模式的转换时序



注 1： 时钟转换通常发生在 2-4 个 TOSC 内。

 © 2007 Microchip Technology Inc.

图 3-2：　　从 SEC_RUN 模式切换到 PRI_RUN 模式的转换时序 （HSPLL）



注　1： TOST = 1024 TOSC； TPLL = 2 ms （近似值）。这些时间间隔未按比例绘制。

2： 时钟转换通常发生在 2-4 个 TOSC 内。

3.2.3　RC_RUN 模式

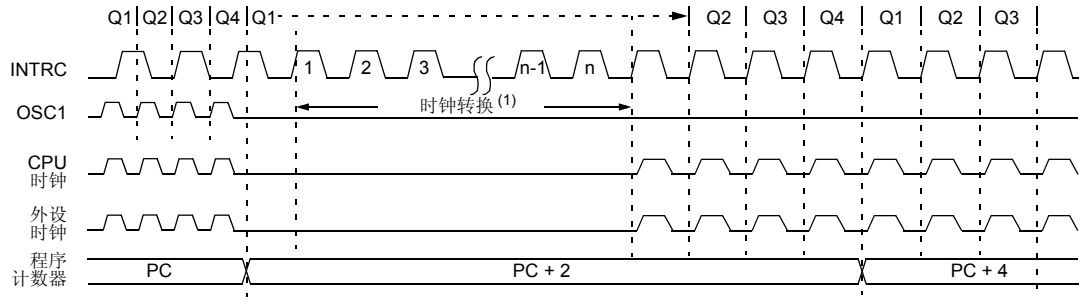
在 RC_RUN 模式下，内部振荡器作为 CPU 和外设的时钟源；主时钟关闭。在使用 INTRC 时钟源时，此模式是在代码执行期间所有运行模式中最节省功耗的运行模式。它非常适用于对定时精度要求不高或者不是一直需要高速时钟的应用。

如果主时钟源为内部振荡器 （INTRC），在代码执行期间，PRI_RUN 和 RC_RUN 这两种模式区别不大。但是在进入和退出 RC_RUN 模式时会发生时钟切换时间延迟。因此，如果主时钟源为内部振荡器，建议不要使用 RC_RUN 模式。

通过将 SCS1 位置 1 可以进入此模式。虽然常常被忽略，但还是建议将 SCS0 位清零，从而保证与未来器件的软件兼容性。当将时钟源切换到 INTRC （见图 3-3）时，主振荡器将被关闭并且 OSTS 位被清零。

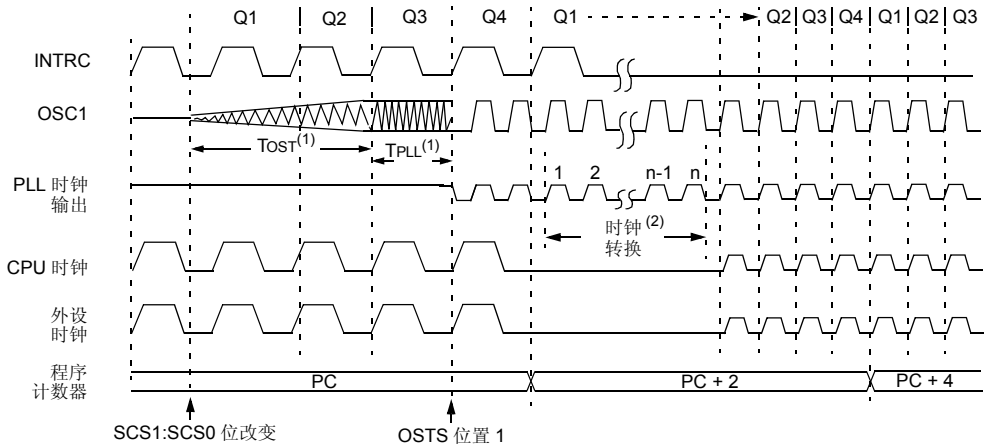
在从 RC_RUN 模式转换到 PRI_RUN 模式期间，在主时钟处于启动状态时，器件将继续使用 INTRC 作为时钟源。当主时钟准备好以后，时钟切换回主时钟 （见图 3-4）。当时钟切换完成后，OSTS 位被置 1 并且由主时钟提供器件时钟。这种切换不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或故障保护时钟监视器，INTRC 源将继续运行。

图 3-3：　　到 RC_RUN 模式的转换时序



注　1： 时钟转换通常发生在 2-4 个 TOSC 内。

图 3-4：　　从 RC_RUN 模式切换到 PRI_RUN 模式的转换时序



注　1： TOST = 1024 TOSC； TPLL = 2 ms （近似值）。这些时间间隔未按比例绘制。

2： 时钟转换通常发生在 2-4 个 TOSC 内。

© 2007 Microchip Technology Inc.

3.3 休眠模式

PIC18F2450/4450 器件的功耗管理休眠模式和所有其他 PIC 器件提供的传统休眠模式相同。通过清零 IDLEN 位 （器件复位时的默认状态）并执行 SLEEP 指令即可进入此模式。这将关闭所选择的振荡器（见图 3-5），并将所有的时钟源状态位清零。

从其他模式进入休眠模式不需要切换时钟。这是因为单片机一旦进入休眠模式就不需要时钟了。如果选择了 WDT， INTRC 时钟源将继续工作。如果使能了 Timer1 振荡器，它也将继续运行。

当在休眠模式中发生唤醒事件时 （通过中断、复位或 WDT 超时），在时钟源（通过 SCS1:SCS0 位选择）准备好之前器件将没有时钟源 （见图 3-6），或者如果使能了双速启动或故障保护时钟监视器，它将使用内部振荡器作为时钟源（见**第 18.0 节 “CPU 的特殊功能”**）。在这两种情况下，当由主时钟提供器件时钟时， OSTS 位将置 1。这种唤醒不会影响 IDLEN 和 SCS 位。

3.4 空闲模式

空闲模式允许在外设继续工作的同时有选择地关闭单片机的 CPU。选择特定的空闲模式允许用户进一步管理功耗。

如果在执行 SLEEP 指令时， IDLEN 位置 1，外设将使用由 SCS1:SCS0 位选择的时钟源，而 CPU 没有时钟源。时钟源状态位不受影响。将 IDLEN 置 1 并执行 SLEEP 指令可以从给定的运行模式快速切换到相应的空闲模式。

如果选择了 WDT， INTRC 时钟源将继续工作。如果使能了 Timer1 振荡器，它也将继续运行。

由于 CPU 没有执行指令，器件只能通过中断、WDT 超时或复位从空闲模式退出。当发生唤醒事件时，CPU 会在其准备好执行代码前延时一个 TCSD 间隔 （表 21-10 中的参数 38）。当 CPU 开始执行代码时，它将沿用当前空闲模式所使用的时钟源。例如，当从 RC_IDLE 模式唤醒时，将使用内部振荡器为 CPU 和外设提供时钟（即 RC_RUN 模式）。这种唤醒不会影响 IDLEN 和 SCS 位。

当处于任何空闲模式或休眠模式中时， WDT 超时会导致 WDT 唤醒并进入当前由 SCS1:SCS0 位指定的运行模式。

图 3-5： 进入休眠模式的转换时序

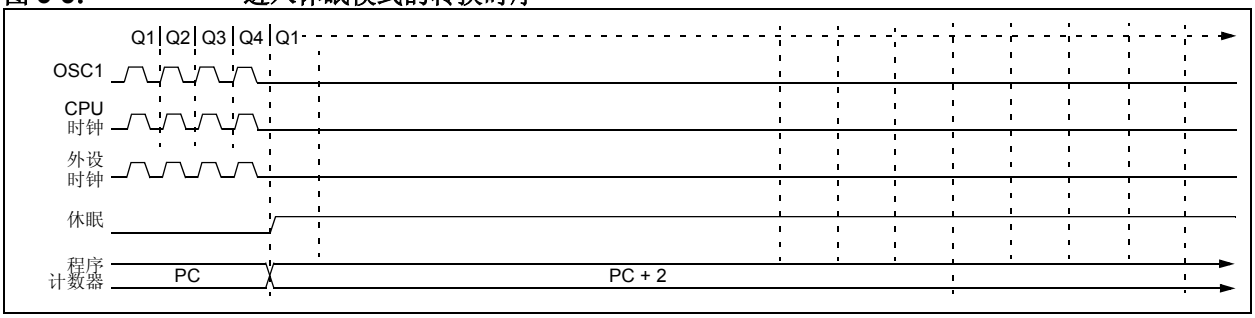
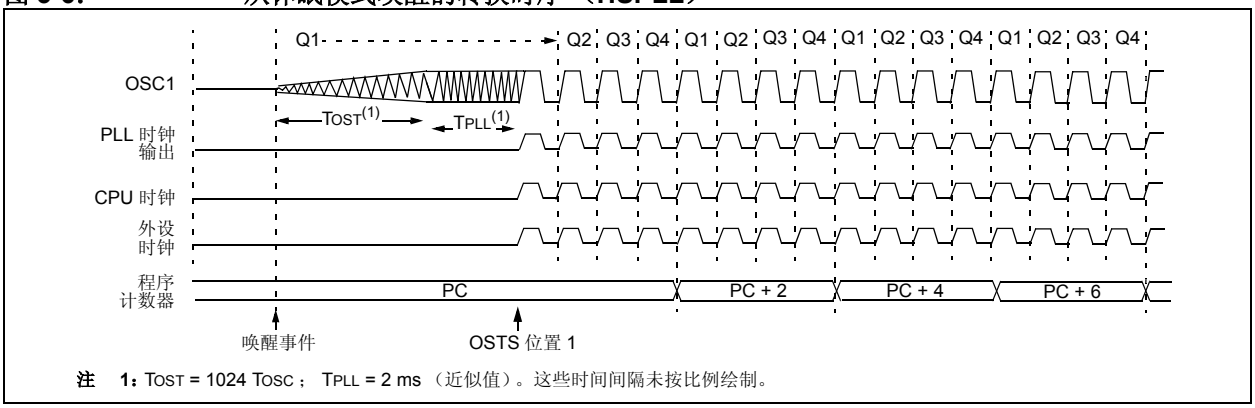


图 3-6： 从休眠模式唤醒的转换时序 （HSPLL）



注 1： TOST = 1024 TOSC； TPLL = 2 ms （近似值）。这些时间间隔未按比例绘制。

3.4.1 PRI_IDLE 模式

在三种低功耗空闲模式中，只有该模式不会禁止主器件时钟。由于时钟源不需要“热身”或是从其他振荡器转换过来，选用此模式可以使对时间要求较高的应用以最快的速度恢复器件运行并使用较精确的主时钟源。

可以通过将 IDLEN 位置 1 并执行 SLEEP 指令以实现从 PRI_RUN 模式进入 PRI_IDLE 模式。如果器件在另一种运行模式，可以先将 IDLEN 位置 1，然后清零 SCS 位并执行 SLEEP。虽然 CPU 已被禁止，但外设仍可继续使用由FOSC3:FOSC0配置位指定的主时钟源为其提供时钟信号。 OSTS 位保持置 1 （见图 3-7）。

当发生唤醒事件时，由主时钟源为 CPU 提供时钟。在唤醒事件和代码执行开始之间需要一个TCSD间隔的延时。该延时用来让 CPU 做好执行指令的准备。在唤醒之后，OSTS 位保持置 1 状态。这种唤醒不会影响 IDLEN 和 SCS 位 （见图 3-8）。

3.4.2 SEC_IDLE 模式

在 SEC_IDLE 模式下， CPU 被禁止，但外设继续将 Timer1 振荡器作为时钟源。可以通过将 IDLEN 位置 1 并执行 SLEEP 指令从 SEC_RUN 模式进入此模式。如果器件处于另一种运行模式，首先将 IDLEN 位置 1，然后将 SCS1:SCS0 置为 01 并执行 SLEEP。当时钟源切换到 Timer1 振荡器时，主振荡器关闭， OSTS 位被清零并且 T1RUN 位置 1。

当唤醒事件发生时，外设继续将 Timer1 振荡器作为时钟源。唤醒事件发生后经过一个 TCSD 时间间隔， CPU 开始执行代码并使用 Timer1 振荡器作为其时钟源。这种唤醒不会影响 IDLEN 和 SCS 位。Timer1 振荡器继续运行（见图 3-8）。

注：	Timer1 振荡器应该在进入 SEC_IDLE 模式之前就已经运行了。如果执行 SLEEP 指令时 T1OSCEN 位没有置 1，就会忽略 SLEEP 指令并不会进入 SEC_IDLE 模式。如果使能了 Timer1 振荡器，但它尚未运行，外设时钟将会延时直到该振荡器起振。在这种情况下，最初的振荡器运行很不稳定，可能会导致无法预料的结果。

图 3-7： 进入空闲模式的转换时序

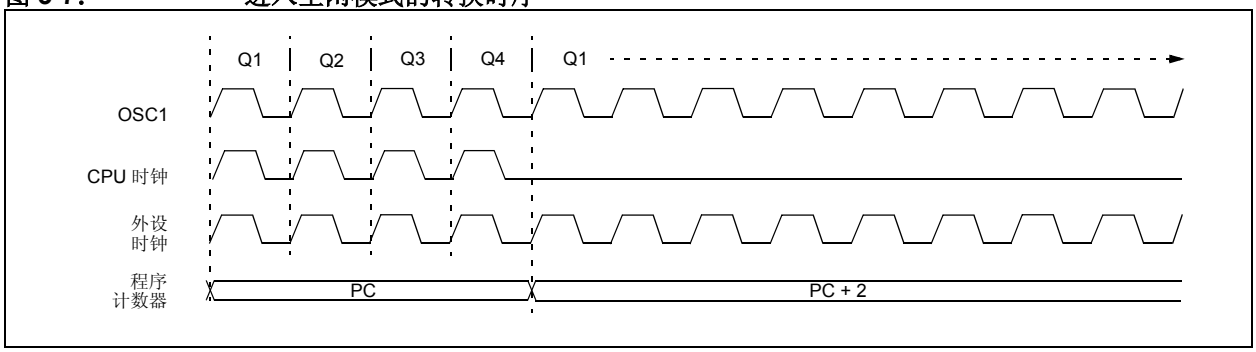
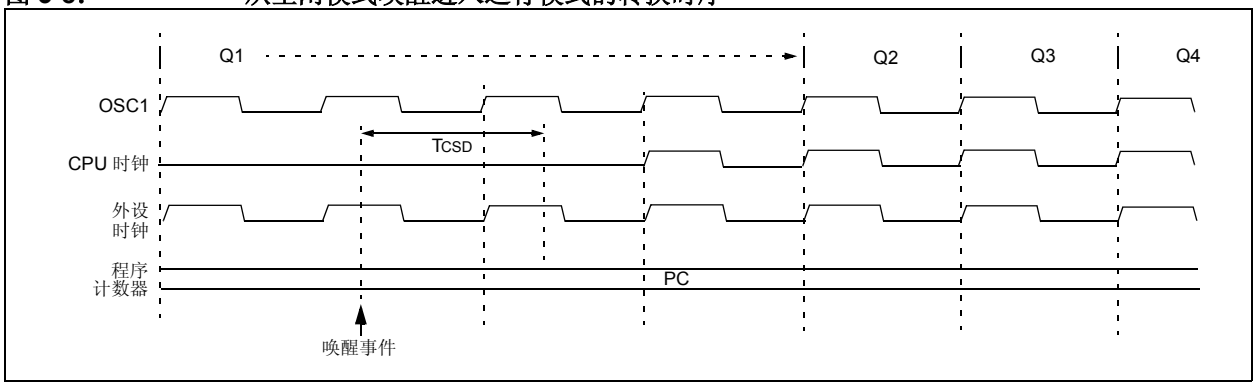


图 3-8： 从空闲模式唤醒进入运行模式的转换时序



 © 2007 Microchip Technology Inc.

3.4.3 RC_IDLE 模式

RC_IDLE 模式禁止 CPU，但仍继续由内部振荡器 INTRC 为外设提供时钟。该模式允许在空闲期间对功耗进行控制。

可以通过将IDLEN位置1并执行SLEEP指令从RC_RUN模式进入此模式。如果器件处于另一种运行模式，可以先将 IDLEN 位置 1，然后再将 SCS1 位置 1 并执行 SLEEP。虽然 SCS0 的值常常被忽略，但仍建议将其清零，这将保证与未来器件的软件兼容性。当时钟源切换到 INTRC 时，主振荡器被关闭， OSTS 位被清零。

当唤醒事件发生时，外设继续将 INTRC 作为时钟源。在唤醒事件后的 TCSD 间隔之后，CPU 开始执行代码并使用 INTRC 作为时钟源。这种唤醒不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或故障保护时钟监视器，INTRC 源将继续运行。

3.5 退出空闲和休眠模式

由中断、复位或 WDT 超时触发从休眠模式或任何空闲模式的退出。本节将讨论从功耗管理模式退出的触发方式。在每种功耗管理模式中我们还将讨论其时钟源子系统的作用（见**第 3.2 节“运行模式”**、**第 3.3 节“休眠模式”**和**第 3.4 节“空闲模式”**）。

3.5.1 通过中断退出

任何可用的中断源都可导致器件从空闲模式或休眠模式退出到运行模式。要使能此功能，必须将对应 INTCON 或 PIE 寄存器中的中断源允许位置 1 来允许中断源。当相应的中断标志位置 1 时，触发退出操作。

当使用中断从空闲或休眠模式退出时，如果GIE/GIEH位（INTCON<7>）置 1，程序就会跳转到中断向量处执行代码。否则代码就会顺序执行（见**第 8.0 节“中断”**）。

唤醒事件之后需要一个固定的 TCSD 间隔的延时，器件才会退出休眠和空闲模式。CPU需要此延时来准备执行代码。在延时后的第一个时钟周期重新开始执行指令。

3.5.2 通过 WDT 超时退出

根据 WDT 超时发生时器件所处的不同功耗管理模式会引发不同的操作。

如果器件不在执行代码 （所有空闲模式和休眠模式），超时将导致从功耗管理模式退出（见**第 3.2 节“运行模式”**和**第 3.3 节 “休眠模式”**）。如果器件正在执行代码 （所有运行模式），超时将导致 WDT 复位 （见**第 18.2 节 “看门狗定时器 （WDT）”**）。

3.5.3 通过复位退出

通常，器件通过振荡器起振定时器（OST）保持在复位状态，直到主时钟就绪。主时钟就绪后，OSTS 位置 1，器件开始执行代码。

从复位状态退出到开始执行代码期间的延迟时间由唤醒前后的时钟源以及主时钟振荡器的类型（如果新的时钟源为主时钟）决定。表 3-2 总结了退出延时。

可以在主时钟就绪之前开始执行代码。如果使能了双速启动 （见**第 18.3 节 “双速启动”**）或故障保护时钟监视器（见**第 18.4 节“故障保护时钟监视器”**），器件可以在复位源被清除之后马上开始执行代码。由内部振荡器驱动的 INTRC 作为代码执行的时钟源。执行代码时，由内部振荡器提供时钟源直到主时钟就绪，或者在主时钟就绪前进入功耗管理模式，随后将关闭主时钟。

3.5.4 在没有振荡器起振延时的情况下退出

从某些功耗管理模式退出完全不需要 OST 延时。有以下两种情形：

- 主时钟源不停止的 PRI_IDLE 模式
- 主时钟源不是 XT 或 HS

在这些情况下，主时钟源不需要振荡器起振延时，因为它已经在运行（PRI_IDLE），或者它本来就不需要振荡器起振延时 （EC 和任何内部振荡器模式）。但是，当器件退出休眠和空闲模式时，在唤醒事件之后仍然需要一个固定的 TCSD 间隔的延时，以便让 CPU 准备好执行代码。在延时后的第一个时钟周期重新开始执行指令。

表 3-2： 通过复位从休眠模式或空闲模式唤醒的退出延时 （按时钟源分类）

单片机时钟源		退出延时	时钟就绪状态位（OSCCON）
唤醒之前的时钟源	唤醒之后的时钟源		
主器件时钟（PRI_IDLE 模式）	XT 或 HS	无	OSTS
	XTPLL 或 HSPLL		
	EC		
	INTRC(1)		
T1OSC 或 INTRC(1)	XT 或 HS	Tost(3)	OSTS
	XTPLL 或 HSPLL	Tost + t_{rc}(3)	
	EC	Tcsd(2)	
	INTRC(1)	Tiobst(4)	
INTRC(1)	XT 或 HS	Tost(3)	OSTS
	XTPLL 或 HSPLL	Tost + t_{rc}(3)	
	EC	Tcsd(2)	
	INTRC(1)	无	
无（休眠模式）	XT 或 HS	Tost(3)	OSTS
	XTPLL 或 HSPLL	Tost + t_{rc}(3)	
	EC	Tcsd(2)	
	INTRC(1)	Tiobst(4)	

注 1： 在此示例中，特指 31 kHz INTRC 时钟源。

2： 当从休眠模式和空闲模式唤醒时都需要 TCSD （表 21-10 的参数 38）延时，该延时与所需的其他延时并行（见**第 3.4 节 “空闲模式”**）。

3： TOST 是振荡器起振定时器周期 （表 21-10 的参数 32）。t_{rc} 是 PLL 锁定延时 （表 21-7 的参数 F12）。后者也被称为 TPLL。

4： 在 INTRC 稳定周期——TIOBST （表 21-10 的参数 39）延时期间，代码继续执行。

© 2007 Microchip Technology Inc.

4.0 复位

PIC18F2450/4450 器件有以下几种不同的复位方式：

a） 上电复位 （POR）
b） 正常工作状态下的 $\overline{\text{MCLR}}$ 复位
c） 功耗管理模式下的 $\overline{\text{MCLR}}$ 复位
d） 看门狗定时器 （WDT）复位 （执行程序期间）
e） 可编程欠压复位 （BOR）
f） RESET 指令复位
g） 堆栈满复位
h） 堆栈下溢复位

本节将讨论由 $\overline{\text{MCLR}}$、POR 和 BOR 产生的复位，并涉及各种起振定时器的工作方式。堆栈复位事件将在**第 5.1.2.4 节 “堆栈满和下溢复位”**中讨论。WDT 复位将在**第 18.2 节 “看门狗定时器 （WDT）”**中讨论。

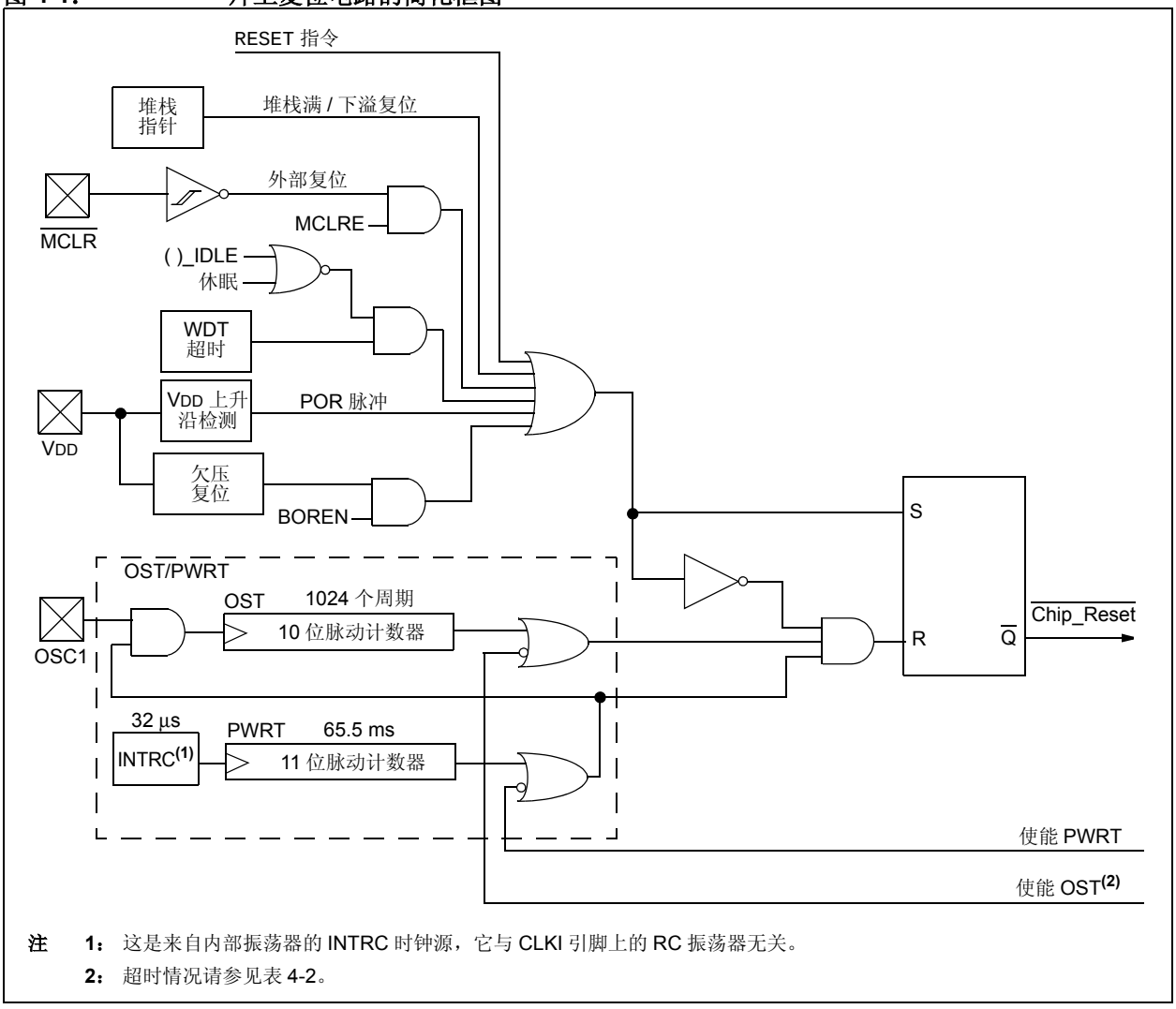
图 4-1 给出了片上复位电路的简化框图。

4.1 RCON 寄存器

通过 RCON 寄存器 （寄存器 4-1）跟踪器件复位事件。该寄存器的低 5 位表明是否已经发生了特定的复位事件。在大多数情况下，只能通过事件将这些位清零，而且必须在事件发生后由应用程序将它们置 1。需要读取所有这些标志位来确定刚发生的复位的类型。在**第 4.6 节 “寄存器的复位状态”**中对此进行了更详细的说明。

RCON 寄存器中还有设置中断优先级的控制位 （IPEN）和对 BOR 进行软件控制的控制位 （SBOREN）。在**第 8.0 节 “中断”**中讨论了中断优先级。在**第 4.4 节 “欠压复位 （BOR）”**中讨论了 BOR。

图 4-1： 片上复位电路的简化框图



注 1： 这是来自内部振荡器的 INTRC 时钟源，它与 CLKI 引脚上的 RC 振荡器无关。
2： 超时情况请参见表 4-2。

寄存器 4-1：　　RCON：复位控制寄存器

R/W-0	R/W-1(1)	U-0	R/W-1	R-1	R-1	R/W-0(2)	R/W-0
IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　**IPEN**：中断优先级使能位
1 = 使能中断优先级
0 = 禁止中断优先级 （PIC16CXXX 兼容模式）

bit 6　**SBOREN**：BOR 软件使能位 (1)
如果 BOREN1:BOREN0 = 01：
1 = 使能 BOR
0 = 禁止 BOR
如果 BOREN1:BOREN0 = 00、 10 或 11：
该位被禁止并读为 0。

bit 5　**未用**：读为 0

bit 4　**$\overline{RI}$**：RESET 指令标志位
1 = 未执行 RESET 指令 （只能由固件置 1）
0 = 已执行了 RESET 指令，导致器件复位 （必须在发生欠压复位后由软件置 1）

bit 3　**$\overline{TO}$**：看门狗超时标志位
1 = 通过上电、 CLRWDT 指令或 SLEEP 指令置 1
0 = 发生了 WDT 超时溢出

bit 2　**$\overline{PD}$**：掉电检测标志位
1 = 通过上电或 CLRWDT 指令置 1
0 = 通过执行 SLEEP 指令置 1

bit 1　**$\overline{POR}$**：上电复位状态位 (2)
1 = 未发生上电复位 （只能由固件置 1）
0 = 已发生上电复位 （必须在发生上电复位后由软件置 1）

bit 0　**$\overline{BOR}$**：欠压复位状态位
1 = 未发生欠压复位 （只能由固件置 1）
0 = 已发生欠压复位 （必须在发生欠压复位后由软件置 1）

注　1： 如果使能了 SBOREN 位，其复位状态为 1；否则为 0。
2： $\overline{POR}$ 的实际复位值由器件复位的类型决定。更多信息，请参见本寄存器下方的 “注”和**第 4.6 节 “寄存器的复位状态”**。

注　1： 建议在检测到上电复位后，将 $\overline{POR}$ 位置 1，以便继续检测后续的上电复位。
2： 当 $\overline{BOR}$ 为 0 并且 $\overline{POR}$ 为 1 时（假定在 POR 之后立即用软件将 $\overline{POR}$ 置 1），可以说已发生了欠压复位。

© 2007 Microchip Technology Inc.

4.2 主清零复位（MCLR）

MCLR 引脚提供了用外部硬件触发器件复位的方法。将该引脚拉低可以产生复位信号。这些器件在 MCLR 复位路径上有一个噪声滤波器，该滤波器可以检测并滤除小的干扰脉冲。

任何内部复位，包括 WDT 复位，均不能将 MCLR 引脚驱动为低电平。

在 PIC18F2450/4450 器件中，可以用 MCLRE 配置位禁止 MCLR 输入。当禁止 MCLR 时，该引脚将成为一个数字输入引脚。更多信息，请参见**第9.5节“PORTE、TRISE 和 LATE 寄存器”**。

4.3 上电复位（POR）

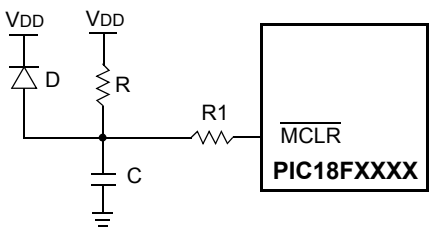
只要当 VDD 上升超过某个门限时，就会在片上产生上电复位脉冲。这使得 VDD 达到满足器件正常工作的数值时，器件会以初始化状态启动。

为了利用 POR 电路，将 MCLR 引脚通过一个电阻（阻值范围为 1 kΩ 到 10 kΩ）连接到 VDD。这样可以省去产生上电复位延时通常所需的外部 RC 元件。VDD 的最小上升速率已指定（**第 21.1 节 “直流特性”**中的参数 D004）。上升速率缓慢的情况，请参见图 4-2。

当器件开始正常工作（即，退出复位状态）时，器件的工作参数（电压、频率和温度等）必须得到满足，以确保其正常工作。如果不满足这些条件，那么器件必须保持在复位状态，直到满足工作条件为止。

POR 事件由 POR 位（RCON<1>）捕获。每当发生 POR 时，该位的状态就会被置为 0；任何其他复位事件均不能改变它。任何硬件事件均不能将 POR 复位为 1。要捕获多个事件，用户必须在 POR 之后用软件手动将该位复位为 1。

图 4-2：　外部上电复位电路（VDD 缓慢上电的情况）



注　1： 仅当 VDD 上电速率过慢时才需要外部上电复位电路。二极管 D 有助于在 VDD 掉电时使电容迅速放电。

2： 建议 R < 40 kΩ，确保电阻 R 两端压降符合器件的电气规范。

3： R1 ≥ 1 kΩ 将限制任何电流从外部电容 C 流入 MCLR，以避免由于静电放电（Electrostatic Discharge，ESD）或电过载（Electrical Overstress，EOS）导致 MCLR/VPP 引脚损坏。

4.4 欠压复位 （BOR）

PIC18F2450/4450 器件带有一个 BOR 电路，它将为用户提供一系列配置和节能选项。 BOR 由 BORV1:BORV0 和 BOREN1:BOREN0 配置位控制。总共有 4 种 BOR 配置，归纳在表 4-1 中。

BOR 门限值由 BORV1:BORV0 位设置。如果使能了 BOR（BOREN1:BOREN0 为除“00”以外的任何值），当 VDD 跌落到低于 VBOR （**第 21.1 节 “直流特性：供电电压”**中的参数 D005）值的时间大于 TBOR（表 21-10 中的参数 35）就会复位器件。如果 VDD 跌落到 VBOR 以下的时间小于 TBOR 可能发生也可能不会发生复位。发生欠压复位以后，芯片将保持这种状态，直至 VDD 上升到 VBOR 以上。

如果使能了上电延时定时器，则它将在 VDD 上升到超过 VBOR 之后开始工作，并使芯片在延时 TPWRT （表 21-10 中的参数 33）期间保持复位。如果在上电延时定时器运行过程中， VDD 电压降到了 VBOR 以下，芯片将重新回到欠压复位状态并且上电延时定时器会恢复为初始状态。一旦 VDD 电压上升到 VBOR 以上，上电延时定时器将重新执行延时。

BOR 和上电延时定时器 （PWRT）是分别配置的。使能 BOR 复位并不会自动使能 PWRT。

4.4.1 用软件使能 BOR

当 BOREN1:BOREN0 = 01 时，用户可以用软件使能或禁止 BOR。这可通过控制位 SBOREN （RCON<6>）完成。如前所述，将 SBOREN 置 1 可使能 BOR。清零 SBOREN 将完全禁止 BOR。SBOREN 位只工作在这个模式，其他情况下它将读为 0。

用软件控制BOR位可使用户能更灵活地定制应用程序以使其适应环境，而无需通过对器件重新编程来更改 BOR 配置。它还允许用户通过减少 BOR 消耗的电流，用软件调节器件的功耗。虽然 BOR 的电流通常很小，但是它可能对低功耗应用有一些影响。

注：	即使当 BOR 受软件控制时， BOR 复位电平仍将由 BORV1:BORV0 配置位设置。该值不能用软件更改。

4.4.2 检测 BOR

使能 BOR 后，当发生 BOR 或 POR 事件时，$\overline{\text{BOR}}$ 位总是复位为 0。因此只通过读 $\overline{\text{BOR}}$ 位的状态很难确定是否发生过 BOR 事件。更可靠的方法是同时检查 $\overline{\text{POR}}$ 和 $\overline{\text{BOR}}$ 的状态。假定在发生任何 $\overline{\text{POR}}$ 事件后，$\overline{\text{POR}}$ 位被立即用软件复位为 1。如果 $\overline{\text{BOR}}$ 为 0 同时 $\overline{\text{POR}}$ 为 1，那么就可以断定已经发生了 BOR 事件。

4.4.3 在休眠模式下禁止 BOR

当 BOREN1:BOREN0 = 10 时， BOR 受硬件控制并且像前面描述的那样工作。每当器件进入休眠模式时，就会自动禁止 BOR。当器件返回到任何其他工作模式时，又将自动重新使能 BOR。

此模式使应用程序能在有效执行代码的同时从欠压状态恢复，这也是器件最需要 BOR 保护的状况。同时，通过消除增加的 BOR 电流，可以省去休眠模式下的额外功耗。

表 4-1： BOR 配置

BOR 配置		SBOREN （RCON<6>）的状态	BOR 操作
BOREN1	BOREN0		
0	0	不可用	禁止 BOR ； 必须对配置位重新编程才能使能 BOR。
0	1	可用	用软件使能 BOR ； 工作模式由 SBOREN 控制。
1	0	不可用	用硬件在运行和空闲模式下使能 BOR，在休眠模式下禁止。
1	1	不可用	用硬件使能 BOR ； 必须对配置位重新编程才能禁止 BOR。

 © 2007 Microchip Technology Inc.

4.5 器件复位定时器

PIC18F2450/4450 器件包含了三个独立的片上定时器，有助于调节上电复位过程。它们的主要功能是确保在代码执行之前器件时钟稳定。这些定时器是：

- 上电延时定时器 （PWRT）
- 振荡器起振定时器 （OST）
- PLL 锁定延时定时器

4.5.1 上电延时定时器 （PWRT）

PIC18F2450/4450 器件的上电延时定时器（PWRT）是一个 11 位计数器，它使用 INTRC 时钟源作为时钟输入。该定时器可产生大约 2048 x 32 µs = 65.6 ms 的时间间隔。 PWRT 计数期间，器件保持在复位状态。

上电延时时间取决于 INTRC 时钟，并且由于温度和工艺的不同，不同器件的延迟时间也将各不相同。详情请参见直流参数 33 （表 21-10）。

通过清零 $\overline{\text{PWRTEN}}$ 配置位可使能 PWRT。

4.5.2 振荡器起振定时器 （OST）

在 PWRT 延时 （表 21-10 中的参数 33）结束以后，由振荡器起振定时器 （OST）提供一个 1024 振荡器周期（来自 OSC1 输入）的延时，从而确保晶振或谐振器的起振和稳定工作。

只有在 XT、 HS 和 HSPLL 模式下，并且仅当发生上电复位或从大多数功耗管理模式退出时，才启动 OST 延时。

4.5.3 PLL 锁定延时定时器

当在 PLL 模式下使能 PLL 时，上电复位后的延时时序与其他振荡器模式略有不同。在 PLL 模式下需要使用一个单独的定时器来提供一段足够让 PLL 锁定主振荡器频率的固定延时。PLL 锁定延时（TPLL）通常为 2ms，且在振荡器起振延时后发生。

4.5.4 延时时序

上电延时时序如下：

1. POR 条件清除后，启动 PWRT 延时 （如果使能）。
2. 然后， OST 被激活。

总延迟时间将取决于振荡器的配置和 PWRT 的状态。图 4-3、图 4-4、图 4-5、图 4-6 和图 4-7 各自描述了不同的上电延时时序，其中上电延时定时器被使能，并且器件工作在 HS 振荡器模式下。图 4-3 到图 4-6 也适用于在 XT 模式下工作的器件。对于工作在 RC 模式下的器件以及禁止了 PWRT 的器件，将根本没有延时。

由于延时是由 POR 脉冲触发的，因此如果 $\overline{\text{MCLR}}$ 保持足够长时间的低电平，所有延时都将结束。将 MCLR 电平拉高后器件将立即开始执行代码 （图 4-5）。这对于测试或同步多个并行工作的 PIC18FXXXX 器件是非常有用的。

表 4-2： 不同情形下的延时

振荡器配置	上电复位 (2) 和欠压复位		从功耗管理模式退出
	$\overline{\text{PWRTEN}}$ = 0	$\overline{\text{PWRTEN}}$ = 1	
HS 和 XT	66 ms(1) + 1024 TOSC	1024 TOSC	1024 TOSC
HSPLL 和 XTPLL	66 ms(1) + 1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)
EC 和 ECIO	66 ms(1)	—	—
ECPLL 和 ECPIO	66 ms(1) + 2 ms(2)	2 ms(2)	2 ms(2)
INTIO 和 INTCKO	66 ms(1)	—	—
INTHS 和 INTXT	66 ms(1) + 1024 TOSC	1024 TOSC	1024 TOSC

注 1： 66ms （65.5ms）是上电延时定时器 （PWRT）延迟时间的标称值。

2： 2ms 是 PLL 锁定所需的标称时间。

图 4-3：　　上电延时时序（$\overline{MCLR}$ 连接到 VDD，VDD 电压上升时间 < TPWRT）

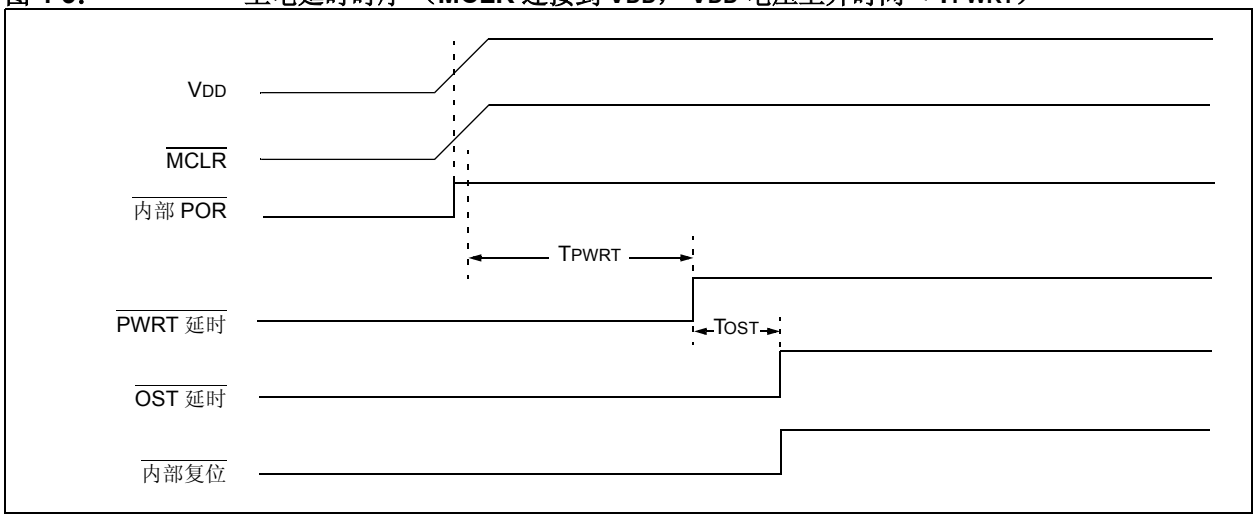


图 4-4：　　上电延时时序（$\overline{MCLR}$ 未连接到 VDD）：情形 1

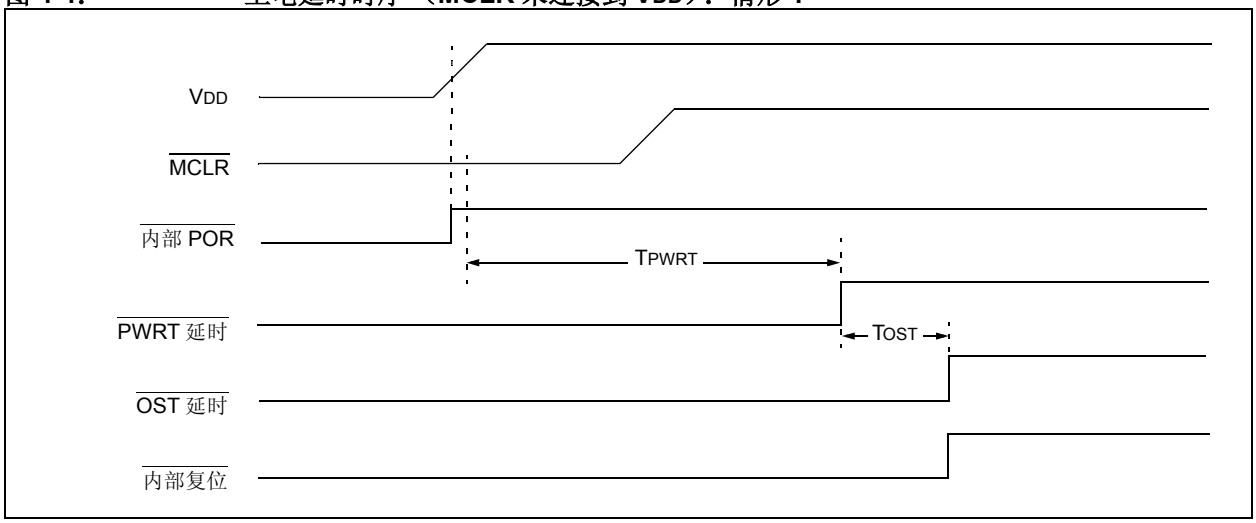
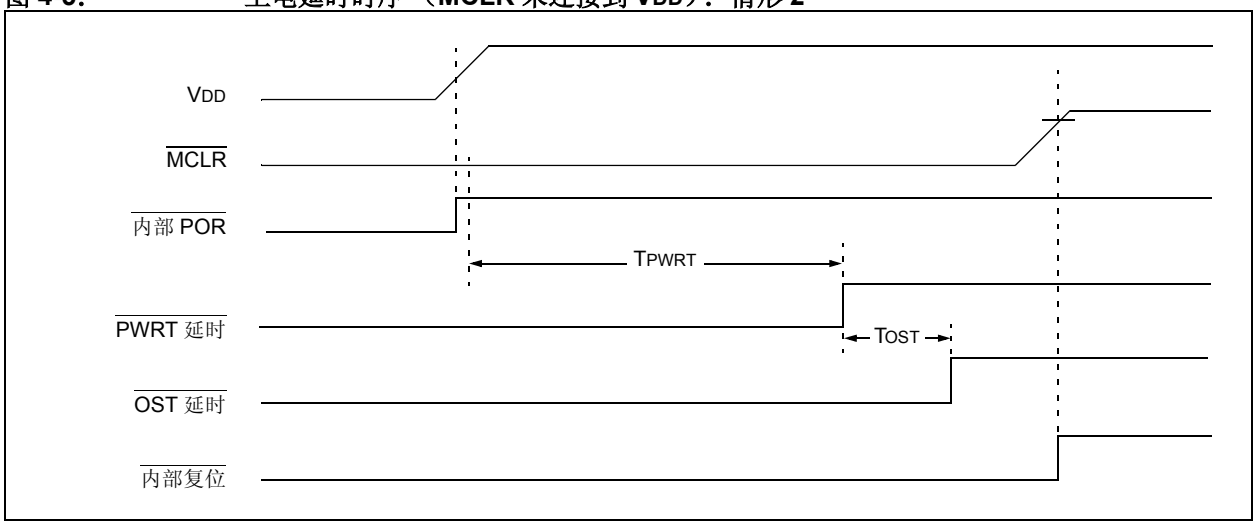


图 4-5：　　上电延时时序（$\overline{MCLR}$ 未连接到 VDD）：情形 2



© 2007 Microchip Technology Inc.

图 4-6：　　缓慢上升时间 （$\overline{\text{MCLR}}$ 连接到 VDD， VDD 电压上升时间 > TPWRT）

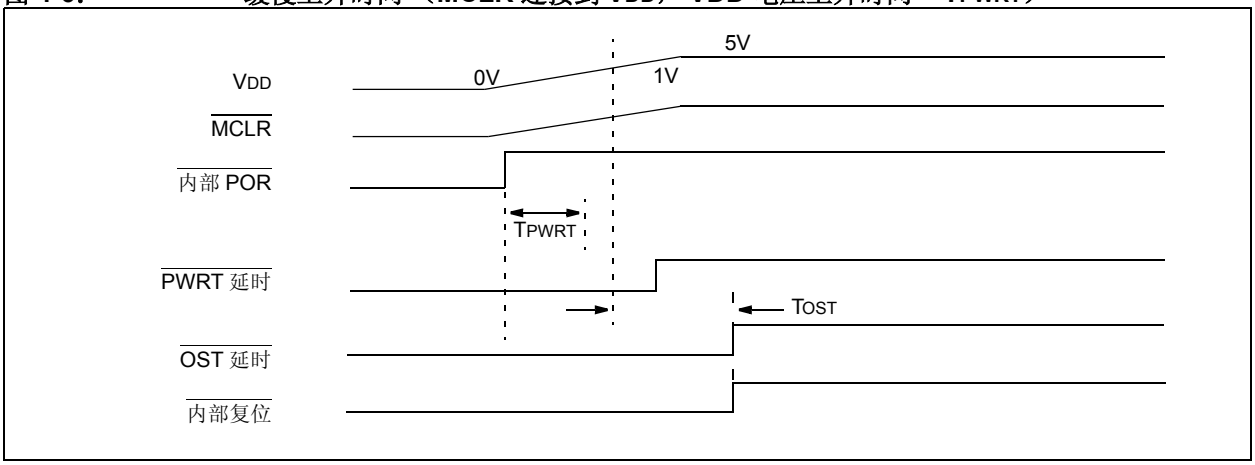
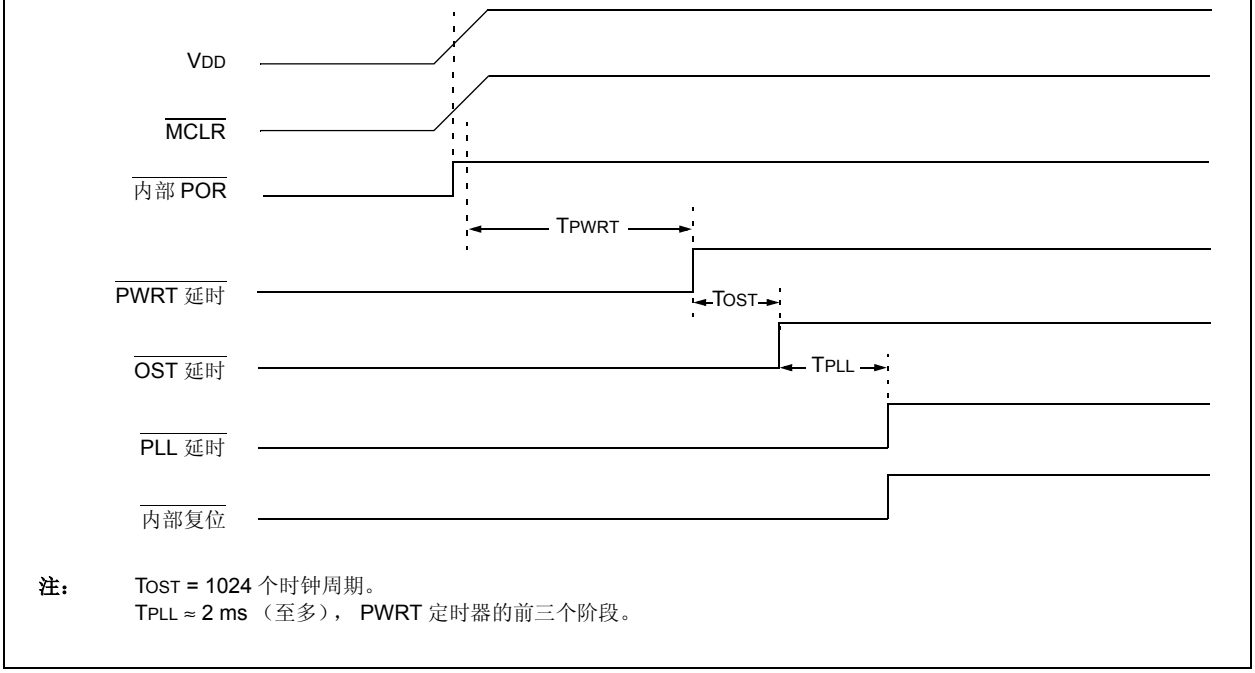


图 4-7：　　在 PLL 使能时 POR 的延时时序 （$\overline{\text{MCLR}}$ 连接到 VDD）



注：　TOST = 1024 个时钟周期。
TPLL ≈ 2 ms （至多）， PWRT 定时器的前三个阶段。

4.6 寄存器的复位状态

大多数寄存器不受复位的影响。在 POR 时这些寄存器的状态不确定，而在其他复位时它们的状态不变。而剩余寄存器则根据不同的复位类型被强制为 “复位状态”。

大多数寄存器不受 WDT 唤醒的影响，这是因为 WDT 唤醒被视为对正常工作的恢复。如表 4-3 所示，RCON 寄存器中的状态位：$\overline{RI}$、$\overline{TO}$、$\overline{PD}$、$\overline{POR}$ 和 $\overline{BOR}$，在不同的复位情形中会分别被置 1 或清零。可在软件中使用这些状态位判断复位的性质。

表 4-4 描述了所有特殊功能寄存器的复位状态。可以将这些复位状态分类为上电和欠压复位、主清零、WDT 复位以及 WDT 唤醒。

表 4-3： RCON 寄存器的状态位、含义以及初始化状态

条件	程序计数器	RCON 寄存器						STKPTR 寄存器	
		SBOREN	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	STKFUL	STKUNF
上电复位	0000h	1	1	1	1	0	0	0	0
RESET 指令	0000h	u(2)	0	u	u	u	u	u	u
欠压复位	0000h	u(2)	1	1	1	u	0	u	u
功耗管理运行模式下的 $\overline{MCLR}$ 复位	0000h	u(2)	u	1	u	u	u	u	u
功耗管理空闲和休眠模式下的 $\overline{MCLR}$ 复位	0000h	u(2)	u	1	0	u	u	u	u
全功耗或功耗管理运行模式期间的 WDT 超时	0000h	u(2)	u	0	u	u	u	u	u
全功耗执行期间的 $\overline{MCLR}$ 复位	0000h	u(2)	u	u	u	u	u	u	u
堆栈满复位 （STVREN = 1）	0000h	u(2)	u	u	u	u	u	1	u
堆栈下溢复位 （STVREN = 1）	0000h	u(2)	u	u	u	u	u	u	1
堆栈下溢错误 （不是真正的复位，STVREN = 0）	0000h	u(2)	u	u	u	u	u	u	1
功耗管理空闲或休眠模式下的 WDT 超时	PC + 2	u(2)	u	0	0	u	u	u	u
通过中断从功耗管理模式退出	PC + 2(1)	u(2)	u	u	0	u	u	u	u

图注： u = 不变

注 1： 当器件被中断唤醒且 GIEH 或 GIEL 置 1 时， PC 装入中断向量 （0008h 或 0018h）。

2： 当软件使能 BOR 时（BOREN1:BOREN0 配置位 = 01 且 SBOREN = 1），POR 时的复位状态是 1 且其他复位不能改变该状态。否则，其复位状态是 0。

© 2007 Microchip Technology Inc.

表 4-4： 所有寄存器的初始化状态

寄存器	适用器件		上电复位，欠压复位	MCLR 复位，WDT 复位，RESET 指令，堆栈复位	通过 WDT 或中断唤醒器件
TOSU	2450	4450	---0 0000	---0 0000	---0 uuuu(1)
TOSH	2450	4450	0000 0000	0000 0000	uuuu uuuu(1)
TOSL	2450	4450	0000 0000	0000 0000	uuuu uuuu(1)
STKPTR	2450	4450	00-0 0000	uu-0 0000	uu-u uuuu(1)
PCLATU	2450	4450	---0 0000	---0 0000	---u uuuu
PCLATH	2450	4450	0000 0000	0000 0000	uuuu uuuu
PCL	2450	4450	0000 0000	0000 0000	PC + 2(3)
TBLPTRU	2450	4450	--00 0000	--00 0000	--uu uuuu
TBLPTRH	2450	4450	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	2450	4450	0000 0000	0000 0000	uuuu uuuu
TABLAT	2450	4450	0000 0000	0000 0000	uuuu uuuu
PRODH	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	2450	4450	0000 000x	0000 000u	uuuu uuuu(2)
INTCON2	2450	4450	1111 -1-1	1111 -1-1	uuuu -u-u(2)
INTCON3	2450	4450	11-0 0-00	11-0 0-00	uu-u u-uu(2)
INDF0	2450	4450	N/A	N/A	N/A
POSTINC0	2450	4450	N/A	N/A	N/A
POSTDEC0	2450	4450	N/A	N/A	N/A
PREINC0	2450	4450	N/A	N/A	N/A
PLUSW0	2450	4450	N/A	N/A	N/A
FSR0H	2450	4450	---- 0000	---- 0000	---- uuuu
FSR0L	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	2450	4450	N/A	N/A	N/A
POSTINC1	2450	4450	N/A	N/A	N/A
POSTDEC1	2450	4450	N/A	N/A	N/A
PREINC1	2450	4450	N/A	N/A	N/A
PLUSW1	2450	4450	N/A	N/A	N/A
FSR1H	2450	4450	---- 0000	---- 0000	---- uuuu
FSR1L	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	2450	4450	---- 0000	---- 0000	---- uuuu

图注： u = 不变， x = 未知， - = 未用位 （读为 0）， q = 值取决于具体条件。
阴影单元表示不适用于指定器件。

注 1： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时，用 PC 的当前值更新 TOSU、 TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。

2： INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 （引起唤醒）。

3： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断向量 （0008h 或 0018h）。

4： 具体条件下的复位值，请参见表 4-3。

5： 根据所选择的振荡器模式使能 PORTA<6>、 LATA<6> 和 TRISA<6>。如果未被配置为 PORTA 引脚，则它们将被禁止并读为 0。

表 4-4： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位，欠压复位	MCLR 复位，WDT 复位，RESET 指令，堆栈复位	通过 WDT 或中断唤醒器件
INDF2	2450	4450	N/A	N/A	N/A
POSTINC2	2450	4450	N/A	N/A	N/A
POSTDEC2	2450	4450	N/A	N/A	N/A
PREINC2	2450	4450	N/A	N/A	N/A
PLUSW2	2450	4450	N/A	N/A	N/A
FSR2H	2450	4450	---- 0000	---- 0000	---- uuuu
FSR2L	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	2450	4450	---x xxxx	---u uuuu	---u uuuu
TMR0H	2450	4450	0000 0000	0000 0000	uuuu uuuu
TMR0L	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	2450	4450	1111 1111	1111 1111	uuuu uuuu
OSCCON	2450	4450	0--- q-00	0--- 0-q0	u--- u-qu
HLVDCON	2450	4450	0-00 0101	0-00 0101	u-uu uuuu
WDTCON	2450	4450	---- ---0	---- ---0	---- ---u
RCON(4)	2450	4450	0q-1 11q0	0q-q qquu	uq-u qquu
TMR1H	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	2450	4450	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	2450	4450	0000 0000	0000 0000	uuuu uuuu
PR2	2450	4450	1111 1111	1111 1111	1111 1111
T2CON	2450	4450	-000 0000	-000 0000	-uuu uuuu
ADRESH	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	2450	4450	--00 0000	--00 0000	--uu uuuu
ADCON1	2450	4450	--00 qqqq	--00 qqqq	--uu uuuu
ADCON2	2450	4450	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	2450	4450	--00 0000	--00 0000	--uu uuuu
SPBRG	2450	4450	0000 0000	0000 0000	uuuu uuuu
RCREG	2450	4450	0000 0000	0000 0000	uuuu uuuu
TXREG	2450	4450	0000 0000	0000 0000	uuuu uuuu

图注： u = 不变， x = 未知， - = 未用位 （读为 0）， q = 值取决于具体条件。
阴影单元表示不适用于指定器件。

注 1： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时，用 PC 的当前值更新 TOSU、 TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。

2： INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 （引起唤醒）。

3： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断向量 （0008h 或 0018h）。

4： 具体条件下的复位值，请参见表 4-3。

5： 根据所选择的振荡器模式使能 PORTA<6>、 LATA<6> 和 TRISA<6>。如果未被配置为 PORTA 引脚，则它们将被禁止并读为 0。

© 2007 Microchip Technology Inc.

表 4-4： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位，欠压复位	MCLR 复位，WDT 复位，RESET 指令，堆栈复位	通过 WDT 或中断唤醒器件
TXSTA	2450	4450	0000 0010	0000 0010	uuuu uuuu
RCSTA	2450	4450	0000 000x	0000 000x	uuuu uuuu
EECON2	2450	4450	0000 0000	0000 0000	0000 0000
EECON1	2450	4450	-x-0 x00-	-u-0 u00-	-u-0 u00-
IPIR2	2450	4450	1-1- -1--	1-1- -1--	u-u- -u--
PIR2	2450	4450	0-0- -0--	0-0- -0--	u-u- -u--(2)
PIE2	2450	4450	0-0- -0--	0-0- -0--	u-u- -u--
IPR1	2450	4450	-111 -111	-111 -111	-uuu -uuu
PIR1	2450	4450	-000 -000	-000 -000	-uuu -uuu(2)
PIE1	2450	4450	-000 -000	-000 -000	-uuu -uuu
TRISE	2450	4450	---- -111	---- -111	uuuu -uuu
TRISD	2450	4450	1111 1111	1111 1111	uuuu uuuu
TRISC	2450	4450	11-- -111	11-- -111	uu-- -uuu
TRISB	2450	4450	1111 1111	1111 1111	uuuu uuuu
TRISA(5)	2450	4450	-111 1111(5)	-111 1111(5)	-uuu uuuu(5)
LATE	2450	4450	---- -xxx	---- -uuu	---- -uuu
LATD	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	2450	4450	xx-- -xxx	uu-- -uuu	uu-- -uuu
LATB	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA(5)	2450	4450	-xxx xxxx(5)	-uuu uuuu(5)	-uuu uuuu(5)
PORTE	2450	4450	---- x000	---- x000	---- uuuu
PORTD	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	2450	4450	xxxx -xxx	uuuu -uuu	uuuu -uuu
PORTB	2450	4450	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA(5)	2450	4450	-x0x 0000(5)	-u0u 0000(5)	-uuu uuuu(5)
UEP15	2450	4450	---0 0000	---0 0000	---u uuuu
UEP14	2450	4450	---0 0000	---0 0000	---u uuuu
UEP13	2450	4450	---0 0000	---0 0000	---u uuuu
UEP12	2450	4450	---0 0000	---0 0000	---u uuuu
UEP11	2450	4450	---0 0000	---0 0000	---u uuuu
UEP10	2450	4450	---0 0000	---0 0000	---u uuuu
UEP9	2450	4450	---0 0000	---0 0000	---u uuuu

图注： u = 不变， x = 未知， - = 未用位 （读为 0）， q = 值取决于具体条件。
阴影单元表示不适用于指定器件。

注 1： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时，用 PC 的当前值更新 TOSU、 TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。

2： INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 （引起唤醒）。

3： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断向量 （0008h 或 0018h）。

4： 具体条件下的复位值，请参见表 4-3。

5： 根据所选择的振荡器模式使能 PORTA<6>、 LATA<6> 和 TRISA<6>。如果未被配置为 PORTA 引脚，则它们将被禁止并读为 0。

表 4-4： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位， 欠压复位	MCLR 复位， WDT 复位， RESET 指令， 堆栈复位	通过 WDT 或 中断唤醒器件
UEP8	2450	4450	---0 0000	---0 0000	---u uuuu
UEP7	2450	4450	---0 0000	---0 0000	---u uuuu
UEP6	2450	4450	---0 0000	---0 0000	---u uuuu
UEP5	2450	4450	---0 0000	---0 0000	---u uuuu
UEP4	2450	4450	---0 0000	---0 0000	---u uuuu
UEP3	2450	4450	---0 0000	---0 0000	---u uuuu
UEP2	2450	4450	---0 0000	---0 0000	---u uuuu
UEP1	2450	4450	---0 0000	---0 0000	---u uuuu
UEP0	2450	4450	---0 0000	---0 0000	---u uuuu
UCFG	2450	4450	00-0 0000	00-0 0000	uu-u uuuu
UADDR	2450	4450	-000 0000	-000 0000	-uuu uuuu
UCON	2450	4450	-0x0 000-	-0x0 000-	-uuu uuu-
USTAT	2450	4450	-xxx xxx-	-xxx xxx-	-uuu uuu-
UEIE	2450	4450	0--0 0000	0--0 0000	u--u uuuu
UEIR	2450	4450	0--0 0000	0--0 0000	u--u uuuu
UIE	2450	4450	-000 0000	-000 0000	-uuu uuuu
UIR	2450	4450	-000 0000	-000 0000	-uuu uuuu
UFRMH	2450	4450	---- -xxx	---- -xxx	---- -uuu
UFRML	2450	4450	xxxx xxxx	xxxx xxxx	uuuu uuuu

图注： u = 不变， x = 未知， - = 未用位 （读为 0）， q = 值取决于具体条件。
阴影单元表示不适用于指定器件。

注 1： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时，用 PC 的当前值更新 TOSU、 TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。

2： INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 （引起唤醒）。

3： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断向量 （0008h 或 0018h）。

4： 具体条件下的复位值，请参见表 4-3。

5： 根据所选择的振荡器模式使能 PORTA<6>、 LATA<6> 和 TRISA<6>。如果未被配置为 PORTA 引脚，则它们将被禁止并读为 0。

 © 2007 Microchip Technology Inc.

5.0 存储器构成

PIC18F2450/4450 单片机器件有两种类型的存储器：

- 程序存储器
- 数据 RAM

由于是哈佛架构的器件，数据和程序存储器使用不同的总线，因而可同时访问这两种存储器空间。

第 6.0 节 “闪存程序存储器”提供了关于闪存程序存储器操作的更多详细信息。

5.1 程序存储器构成

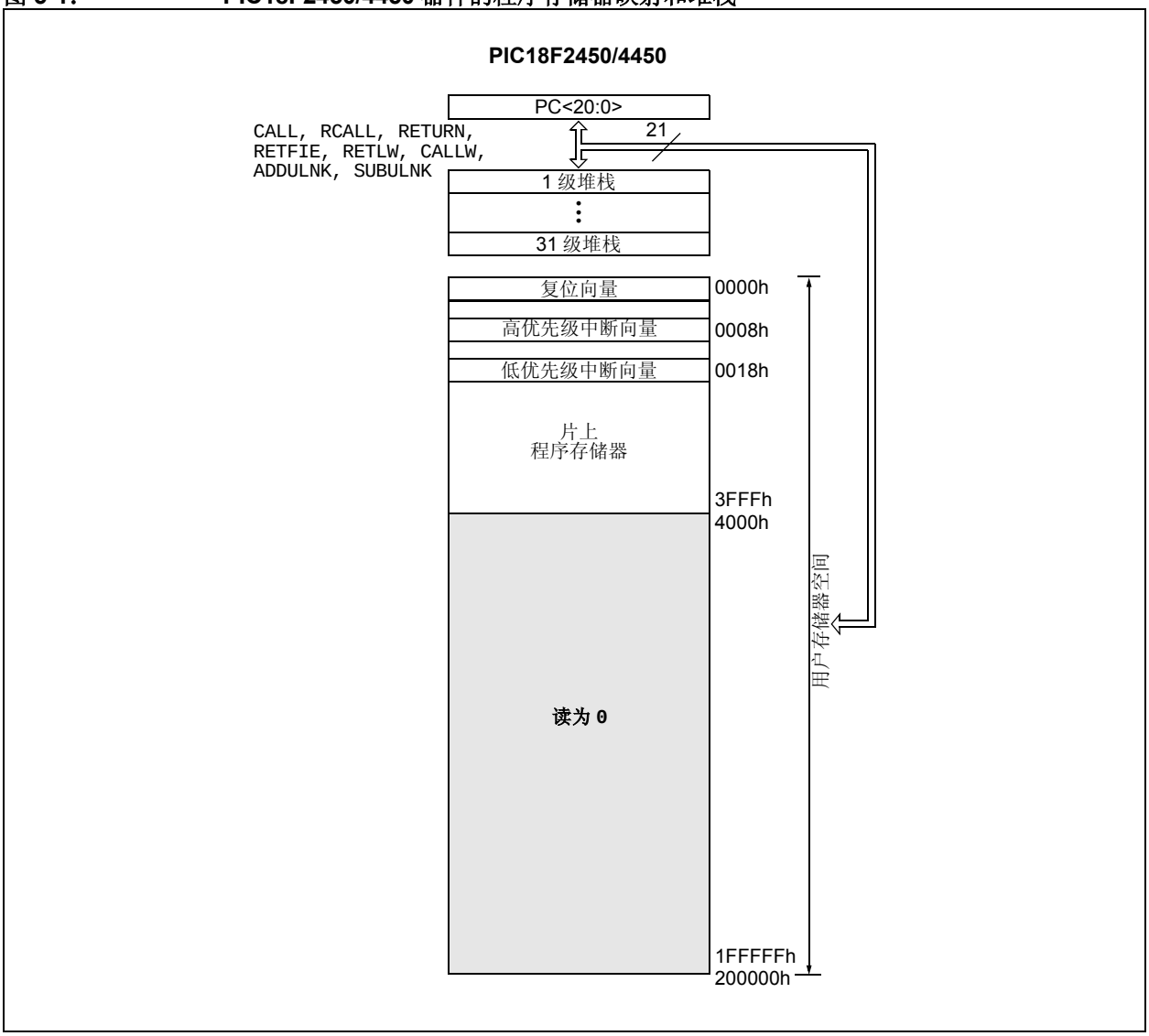
PIC18 单片机具有 21 位程序计数器，可以对 2 MB 的程序存储器空间进行寻址。访问存储器物理地址上边界和这个2 MB地址之间的存储单元会返回全0（NOP指令）。

PIC18F2450 和 PIC18F4450 各有 16 KB 的闪存存储器，可以存储最多 8192 条单字指令。

PIC18 器件有两个中断向量。复位向量地址为 0000h，中断向量地址为 0008h 和 0018h。

图 5-1 给出了 PIC18F2450 和 PIC18F4450 器件的程序存储器映射。

图 5-1： PIC18F2450/4450 器件的程序存储器映射和堆栈



5.1.1 程序计数器

程序计数器（Program Counter， PC）指定欲取出执行的指令的地址。 PC 为 21 位宽，且保存在三个不同的 8 位寄存器中。存储低字节的寄存器称为 PCL 寄存器，该寄存器可读写。存储高字节的寄存器，即 PCH 寄存器，存储 PC<15:8> 位；该寄存器不可直接读写。更新 PCH 寄存器的操作是通过 PCLATH 寄存器实现的。存储最高字节的寄存器称为 PCU。该寄存器存储 PC<20:16> 位；它也不能直接读写。更新 PCU寄存器的操作是通过 PCLATU 寄存器实现的。

PCLATH 和 PCLATU 的内容通过执行写 PCL 的操作被传送到程序计数器。同样，程序计数器的两个高字节通过读 PCL 的操作被传送到 PCLATH 和 PCLATU。这对于计算 PC 的偏移量很有用处（见**第 5.1.4.1 节 “计算 GOTO”**）。

PC 是按字节寻址程序存储器的。为了防止 PC 不能正确获取字指令，需要将 PCL 的最低有效位固定取值为 0。 PC 每次加 2 来寻址程序存储器中的顺序指令。

CALL、RCALL 和 GOTO 程序转移指令直接写入程序计数器。对于这些指令，PCLATH 和 PCLATU 的内容不会传送到程序计数器。

5.1.2 返回地址堆栈

返回地址堆栈允许保存最多 31 个程序调用地址和中断向量。当执行 CALL、 RCALL 指令或响应中断时， PC 值会被压入该堆栈。而在执行 RETURN、 RETLW 或 RETFIE 指令时， PC 值会从堆栈弹出。 PCLATU 和 PCLATH 不受 RETURN 或 CALL 指令的影响。

通过 21 位的 RAM 和一个 5 位的堆栈指针 STKPTR 来实现 31 字的堆栈操作。堆栈既不占用程序存储器空间，也不占用数据存储器空间。堆栈指针是可读写的，并且通过栈顶的特殊功能寄存器可以读写栈顶地址。也可以使用这些寄存器将数据压入堆栈或者从堆栈弹出。

执行 CALL 类型的指令时，执行进栈操作：首先堆栈指针加 1，并且将 PC 的内容写入堆栈指针所指向的地址单元（PC 已经指向 CALL 下一条指令）。执行 RETURN 类型的指令时，执行出栈操作：STKPTR 寄存器所指向的地址单元的内容会被传送给 PC，然后堆栈指针减 1。

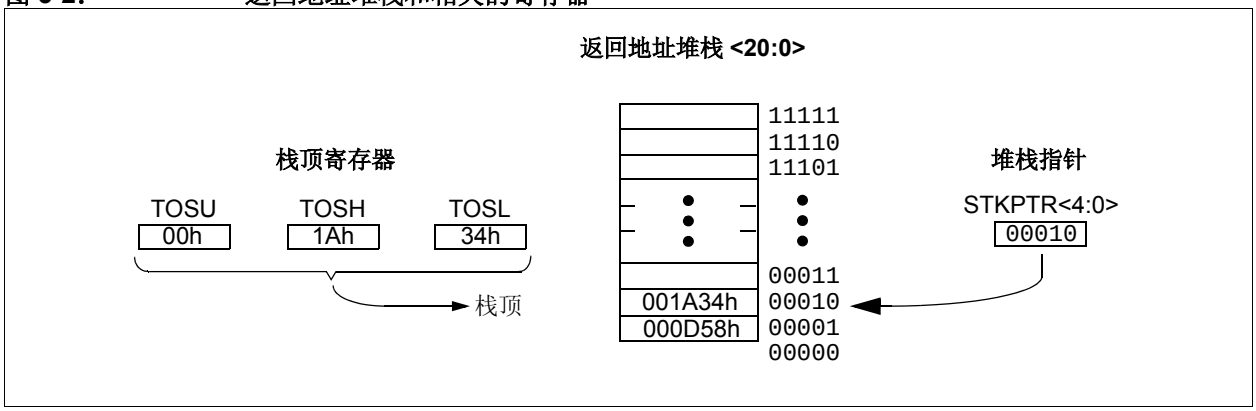
所有复位后，堆栈指针均会初始化为 00000。堆栈指针 00000不指向任何RAM单元；它仅仅是一个复位值。状态位表明堆栈是已满、上溢还是下溢。

5.1.2.1 访问栈顶

只可读写返回地址堆栈的栈顶（Top-of-Stack，TOS）。有三个寄存器 TOSU:TOSH:TOSL 用于保存由 STKPTR 寄存器所指向的堆栈单元的内容（图 5-2）。这可以让用户在必要时实现软件堆栈。在 CALL、 RCALL 或中断后，软件可以通过读取 TOSU:TOSH:TOSL 寄存器来读取压入堆栈的值。这些值可以被置入由用户定义的软件堆栈。返回时，软件将这些值存回 TOSU:TOSH:TOSL 并执行返回。

为防止意外的堆栈操作，访问堆栈时用户必须禁止全局中断允许位。

图 5-2： 返回地址堆栈和相关的寄存器



 © 2007 Microchip Technology Inc.

5.1.2.2 返回堆栈指针 （STKPTR）

STKPTR 寄存器（寄存器 5-1）包含堆栈指针值、STKFUL（堆栈满）状态位和 STKUNF（堆栈下溢）状态位。堆栈指针值可为 0 到 31 范围内的值。向堆栈压入值前，堆栈指针加 1；而从堆栈弹出值后，堆栈指针减 1。复位时，堆栈指针值为零。用户可以读写堆栈指针的值。实时操作系统（Real-Time Operating System，RTOS）可以利用此特性对返回堆栈进行维护。

向堆栈压入 PC 值 31 次（且没有值从堆栈弹出）后，STKFUL 位置 1。通过软件或 POR 使 STKFUL 位清零。

由 STVREN （堆栈溢出复位使能）配置位的状态决定堆栈满时将执行的操作。（关于器件配置位的说明，请参见**第 18.1 节 “配置位”**。）如果 STVREN 位已经置 1（默认），第 31 次进栈将把（PC + 2）值压入堆栈，从而将 STKFUL 位置 1 并复位器件。STKFUL 位将保持置 1，而堆栈指针将被清零。

如果将 STVREN 清零，第 31 次进栈时 STKFUL 位将会置 1，堆栈指针递增到 31。后续的进栈操作都不会覆盖第 31 次进栈的值，并且 STKPTR 将保持为 31。

当出栈次数足够卸空堆栈时，下一次出栈操作会向 PC 返回一个零值，并将 STKUNF 位置 1，而堆栈指针则保持为零。STKUNF 位将保持置 1，直到软件清零或发生 POR 为止。

注：	下溢时，将零值返回给 PC，会使程序指向复位向量，此时可以验证堆栈状态并采取相应的操作。这与复位不同，因为下溢时 SFR 的内容不受影响。

5.1.2.3 PUSH 和 POP 指令

由于栈顶是可以读写的，因此将值压入堆栈或从堆栈弹出而不影响程序的正常执行是非常理想的。PIC18 指令集包含两个指令 PUSH 和 POP，使用这两个指令可在软件控制下对 TOS 执行操作。然后就可以修改 TOSU、TOSH 和 TOSL，将数据或返回地址压入堆栈。

PUSH 指令将当前的 PC 值压入堆栈。执行该指令会使堆栈指针加 1 并将当前的 PC 值装入堆栈。

POP 指令通过将堆栈指针减 1 来放弃当前的 TOS 值。然后前一个入栈的值就成为了 TOS 值。

寄存器 5-1： STKPTR：堆栈指针寄存器

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL(1)	STKUNF(1)	—	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **STKFUL：**堆栈满标志位 (1)
1 = 堆栈满或溢出
0 = 堆栈未满或未溢出

bit 6 **STKUNF：**堆栈下溢标志位 (1)
1 = 发生堆栈下溢
0 = 未发生堆栈下溢

bit 5 **未用：**读为 0

bit 4-0 **SP4:SP0：**堆栈指针地址位

注 1： 通过用户软件或 POR 清零 bit 7 和 bit 6。

5.1.2.4 堆栈满和下溢复位

通过将配置寄存器 4L 中的 STVREN 位置 1，来使能在堆栈溢出或下溢时的器件复位。当 STVREN 置 1 时，堆栈满或堆栈下溢状态会将相应的 STKFUL 或 STKUNF 位置 1，然后使器件复位。当 STVREN 清零时，堆栈满或堆栈下溢状态会将相应的 STKFUL 或 STKUNF 位置 1，但不会使器件复位。只能通过用户软件或上电复位使 STKFUL 或 STKUNF 位清零。

5.1.3 快速寄存器堆栈

为 STATUS、WREG 和 BSR 寄存器提供的快速寄存器堆栈具有从中断“快速返回”的功能。每个堆栈只有一级且不可读写。当处理器转入中断向量处执行时，它装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用 RETFIE, FAST 指令从中断返回，这些寄存器中的值就会重新装回相关的寄存器。

如果同时允许低优先级中断和高优先级中断，从低优先级中断返回时，无法可靠地使用堆栈寄存器。如果在为低优先级中断提供服务时，发生了高优先级中断，则低优先级中断存储在堆栈寄存器中的值将被覆盖。在为低优先级中断提供服务时，用户必须用软件保存关键寄存器的值。

如果未使用中断优先级，所有中断都可以使用快速寄存器堆栈从中断返回。如果没有使用中断，快速寄存器堆栈可以用于在子程序调用结束后恢复 STATUS、WREG 和 BSR 寄存器。要在子程序调用中使用快速寄存器堆栈，必须执行 CALL label, FAST 指令，将 STATUS、WREG 和 BSR 寄存器的内容存入快速寄存器堆栈。然后执行 RETURN, FAST 指令，从快速寄存器堆栈恢复这些寄存器。

例 5-1 给出了一个在子程序调用和返回期间使用快速寄存器堆栈的源代码示例。

例 5-1： 快速寄存器堆栈代码示例

```
CALL  SUB1, FAST     ;STATUS, WREG, BSR
                     ;SAVED IN FAST REGISTER
                     ;STACK
      •
      •

SUB1  •
      •
      RETURN, FAST   ;RESTORE VALUES SAVED
                     ;IN FAST REGISTER STACK
```

5.1.4 程序存储器中的查找表

有的编程需要在程序存储器中创建数据结构或查找表。对于 PIC18 器件，可以用两种方式实现查找表：

- 计算 GOTO
- 表读

5.1.4.1 计算 GOTO

计算 GOTO 是通过向程序计数器加一个偏移量来实现的。例 5-2 给出了一个示例。

使用 ADDWF PCL 指令和一组 RETLW nn 指令可以创建一个查找表。在调用该表前，会先将查找表中的偏移量装入 W 寄存器。被调用子程序的第一条指令应该是 ADDWF PCL 指令。接下去执行的一条是 RETLW nn 指令，它将值 nn 返回给调用函数。

偏移量（WREG 中）指定程序计数器应该增加的字节数，其值应该为 2 的倍数（LSb = 0）。

在这种方式中，每个指令单元只能存储一个数据字节，并且要求返回地址堆栈还有空闲单元。

例 5-2： 使用偏移量计算 GOTO

```
        MOVF   OFFSET, W
        CALL   TABLE
ORG     nn00h
TABLE   ADDWF  PCL
        RETLW  nnh
        RETLW  nnh
        RETLW  nnh
        .
        .
        .
```

5.1.4.2 表读与表写

有一种更好的方法可以将数据存储在程序存储器中，这种方法允许在每个指令单元存储 2 个字节的数据。

使用表读和表写，每个程序字可以存储 2 个字节的查找表数据。表指针（TBLPTR）寄存器指定字节地址，而表锁存（TABLAT）寄存器存储从程序存储器中读取或写入的数据。一次只能写入 / 读出一个字节。

第 6.1 节“表读与表写”将进一步讨论表读和表写操作。

© 2007 Microchip Technology Inc.

5.2 PIC18 指令周期

5.2.1 时钟分配

来自内部或外部时钟源的单片机时钟输入都将在内部被四分频以产生四个互不重叠的正交时钟信号（Q1、Q2、Q3 和 Q4）。程序计数器在每个 Q1 递增；在 Q4 期间，从程序存储器取指令并将指令锁存到指令寄存器 （IR）中。指令的译码和执行在下一个 Q1 到 Q4 周期完成。图 5-3 所示为时钟和指令执行的流程图。

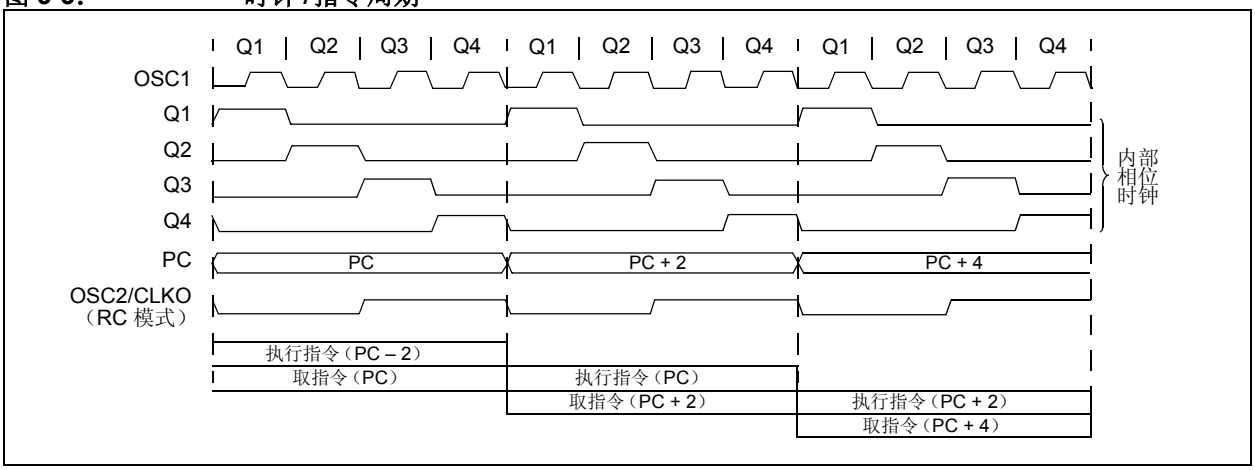
5.2.2 指令流 / 流水线

一个 “指令周期”由 4 个 Q 周期组成：Q1 到 Q4。取指令和执行指令是以流水线方式进行的，用一个指令周期来取指令，而用另一个指令周期译码和执行指令。但由于是流水线操作，所以每条指令的等效执行时间都是一个指令周期。如果某条指令改变了程序计数器（如 GOTO），则需要两个指令周期才能完成该指令 （例 5-3）。

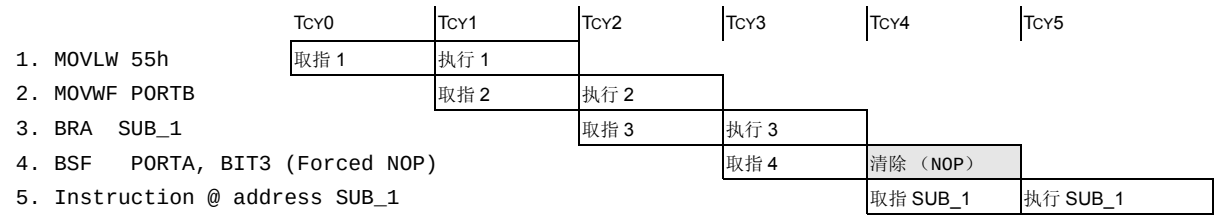
取指周期中：程序计数器 （PC）在 Q1 周期加 1，开始取指令。

指令执行周期中：在 Q1 周期，将所取指令锁存到指令寄存器 （IR）。在随后的 Q2、Q3 和 Q4 周期中译码并执行该指令。其中读数据存储器 （读操作数）发生在 Q2 周期，写操作发生在 Q4 周期 （写目标寄存器）。

图 5-3： 时钟 / 指令周期



例 5-3： 指令流水线流程



注： 除程序转移指令外，所有的指令都是单周期指令。由于程序转移指令将导致一条已取指令从流水线清除，需要重新取指令，然后执行指令，所以程序转移指令需要两个周期。

5.2.3 程序存储器中的指令

程序存储器按字节寻址。指令以 2 字节或 4 字节的形式存储在程序存储器中。指令字的最低有效字节始终存储在地址为偶数的程序存储器单元中（LSb = 0）。要保证正确指向指令单元， PC 必须以 2 为单位递增，并且 LSb 总是读为 0（见**第 5.1.1 节 “程序计数器”**）。

图 5-4 给出了指令字存储在程序存储器中的一个示例。

CALL 和 GOTO 指令在指令中嵌入了程序存储器的绝对地址。由于指令总是存储为一个字长，因而指令所包含的数据为一个字地址。字地址会写入 PC<20:1>，用于访问程序存储器中的目标字节。图 5-4 中的指令 #2 给出了指令 GOTO 0006h 在程序存储器中的译码过程。程序转移指令也采取同样的方式对相对地址偏移量进行译码。存储在转移指令中的偏移量代表单字指令数，PC 将以此作为偏移量跳转到指定的地址单元。**第 19.0 节 “指令集汇总”** 提供了指令集的更多详细信息。

图 5-4： 程序存储器中的指令

			LSB = 1	LSB = 0	字地址 ↓
	程序存储器 字节单元 →				
					000000h
					000002h
					000004h
					000006h
指令 1：	MOVLW	055h	0Fh	55h	000008h
指令 2：	GOTO	0006h	EFh	03h	00000Ah
			F0h	00h	00000Ch
指令 3：	MOVFF	123h, 456h	C1h	23h	00000Eh
			F4h	56h	000010h
					000012h
					000014h

5.2.4 双字指令

标准的 PIC18 指令集有 4 条双字指令：CALL、MOVFF、GOTO 和 LSFR。这些指令第二个字的高 4 位均为 1111；其他 12 位是立即数数据，通常为一个数据存储器地址。

指令的高 4 位为 1111，用于代表一条特殊的 NOP 指令。指令顺序执行的正确顺序为：执行完第一个字之后立即按顺序访问并使用第二个字中的数据。如果由于某些原因跳过了第一个字而自动执行指令的第二个字，那么将作为一条 NOP 指令执行。如果双字指令跟在修改 PC 的条件指令后，就有必要执行此操作。例 5-4 给出了它的执行过程。

注：	关于扩展指令集中的双字指令信息，请参见**第 5.5 节 “程序存储器和扩展指令集”**。

例 5-4： 双字指令

情形 1：

目标代码	源代码		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; No, skip this word
1111 0100 0101 0110			; Execute this word as a NOP
0010 0100 0000 0000	ADDWF	REG3	; continue code

情形 2：

目标代码	源代码		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; Yes, execute this word
1111 0100 0101 0110			; 2nd word of instruction
0010 0100 0000 0000	ADDWF	REG3	; continue code

© 2007 Microchip Technology Inc.

5.3 数据存储器构成

注：	当使能了 PIC18 扩展指令集时，数据存储器某些方面的操作会有所改变。更多信息，请参见**第 5.6 节 “数据存储器和扩展指令集”**。

PIC18 器件中的数据存储器是用静态 RAM 实现的。每个寄存器有 12 位地址，可存储数据达 4096 个字节。存储器空间被分为 16 个存储区，每个存储区包含 256 个字节。PIC18F2450/4450 器件用到 3 个完整的存储区，总共768个字节。图5-5给出了器件的数据存储器构成。

数据存储器由特殊功能寄存器（Special Function Register，SFR）和通用寄存器（General Purpose Register，GPR）组成。SFR 用于单片机和外设功能模块的控制和状态显示，GPR则用在用户应用程序中用于存储数据和临时存储操作的中间结果。任何未用单元均读为 0。

这样的指令集和架构支持跨存储区的操作。可以通过直接、间接或变址寻址模式访问整个数据存储器。本章后面的部分将讨论寻址模式。

为确保能在一个周期存取常用寄存器（SFR 和所选的 GPR），PIC18 器件设置了一个快速操作存储区。该存取区是一个 256 字节的存储器空间，它可实现对 SFR 和 GPR Bank 0 的低地址单元的快速存取，而无需使用 BSR。**第 5.3.3 节 “快速操作存储区”**提供了对于快速操作 RAM 的详细说明。

5.3.1 USB RAM

数据存储器的 Bank 4 实际上被映射为特殊的双端口 RAM。USB 模块禁止时，这些存储区中的 GPR 就像任何其他 GPR 一样使用。

USB 模块使能时，该存储区中的单元被分配为 USB 操作的缓冲 RAM。该区域是单片机内核与 USB 串行接口引擎（SIE）共享的，用于直接在两者间传输数据。

理论上讲可以使用未分配为 USB 缓冲的 USB RAM 存储临时或其他变量。实践中，由于缓冲分配的动态特性，这种做法在最好的情况下也是有风险的。使能该模块时，Bank 4 也用作 USB 缓冲管理，此时不应用作任何其他用途。

USB RAM 和缓冲操作的更多信息请参见**第 14.0 节“通用串行总线（USB）”**。

5.3.2 存储区选择寄存器（BSR）

容量较大的数据存储器需要有效的寻址机制，以便对所有地址进行快速存取。理想状况下，这意味着不必为每次读写操作提供完整地址。PIC18 器件是使用 RAM 存储区机制实现快速存取的。这种机制将存储器空间分成连续的 16 个 256 字节的存储区。根据不同的指令，可以通过完整的 12 位地址直接寻址每个单元，或通过 8 位的低字节地址和 4 位存储区指针间接寻址每个单元。

PIC18 指令集中的大部分指令都使用存储区指针，也就是存储区选择寄存器（Bank Select Register，BSR）。SFR 保存单元地址的高 4 位，而指令本身则包括单元地址的低 8 位。只使用 BSR 的低 4 位（BSR3:BSR0），不使用高 4 位；它们始终读为 0 且不能被写入。可以通过使用 MOVLB 指令直接装载 BSR。

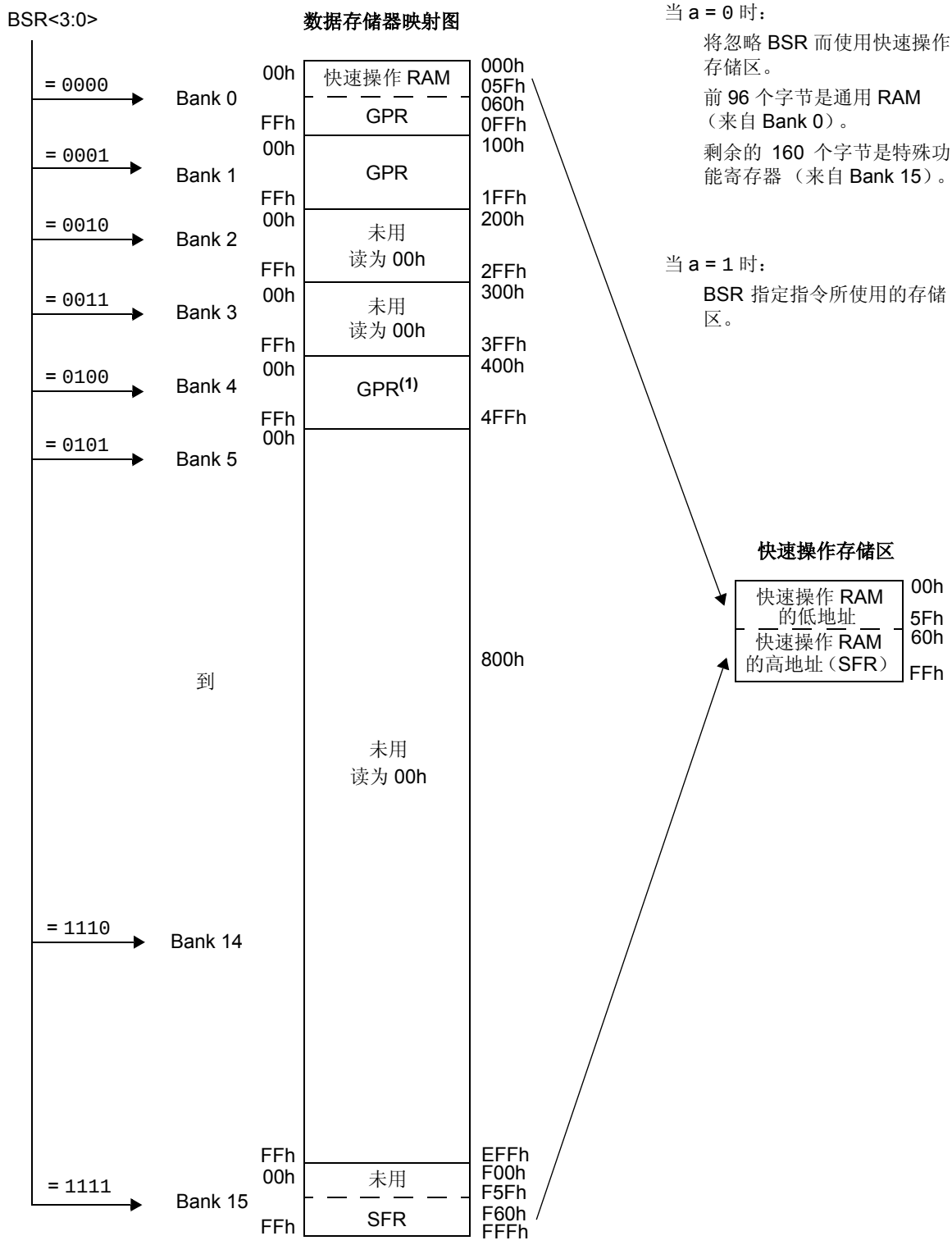
BSR 的值代表数据存储器中的存储区，指令中的 8 位指向存储区中的存储单元，可以将它看作距离存储区下边界的偏移量。图 5-6 显示了 BSR 的值与数据存储器中的存储区之间的关系。

由于最多可有 16 个寄存器共享同一个低位地址，用户必须非常小心以确保在执行数据读或写之前选择了正确的存储区。例如，当 BSR 为 0Fh 时将程序数据写入地址为 F9h 的 8 位地址单元，将导致程序计数器的复位。

当选择存储区时，只有实际可使用的存储区才可以读写。对不可使用的存储区进行的写操作将被忽略，而读这些存储区会返回 0。虽然是这样，状态寄存器仍然会受到影响，好像操作是成功的。图 5-6 中的数据存储器映射图指出了可使用的存储区。

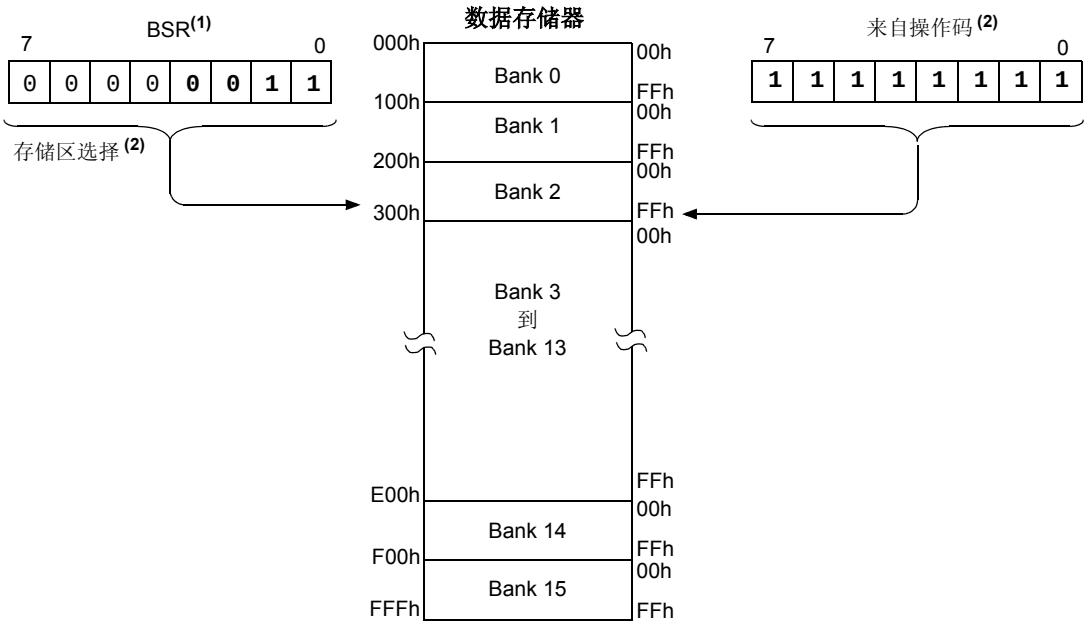
在 PIC18 的内核指令集中，只有 MOVFF 指令指定源寄存器和目标寄存器的完整 12 位地址。该指令在执行时完全忽略 BSR。所有其他指令仅包含作为操作数的低位地址，而且必须使用 BSR 或快速操作存储区来寻址目标寄存器。

图 5-5：　　PIC18F2450/4450 器件的数据存储器映射图



注　　1： 这个存储区也用作 USB 操作的 RAM 缓冲。更多信息，请参见**第 5.3.1 节 “USB RAM”**。

图 5-6：　　使用存储区选择寄存器 （直接寻址）



注　　1： 指令的快速操作 RAM 位强制跳过选定的存储区 （BSR<3:0>）直接寻址快速操作存储区的寄存器。

2： MOVFF 指令嵌入了完整的 12 位地址。

5.3.3　　快速操作存储区

使用 BSR 和嵌入的 8 位地址，用户可以寻址数据存储器的整个空间，但这同时也意味着用户必须始终确保选择了正确的存储区。否则，可能会从错误的单元读取数据或将数据写入错误的单元。如果本来是向 GPR 进行写操作，却将结果写入了 SFR，后果是非常严重的。但是在每次向数据存储器进行读或写操作时确认和 / 或更改 BSR 会严重影响工作效率。

为了连续访问大多数常用数据存储器单元，现为数据存储器配置了快速操作存储区，这样可以允许用户访问被映射的存储区而无需指定 BSR。快速操作存储区由 Bank 0 的前 96 个字节 （00h-5Fh）和 Bank 15 的后 160 个字节 （60h-FFh）组成。地址较低的部分被称为“快速操作 RAM”，由 GPR 组成。地址较高的部分则被映射为器件的 SFR。这两个区域被连续地映射到快速操作存储区并且可以用一个 8 位地址进行线性寻址（图 5-5）。

包括快速操作 RAM 位（指令中的“a”参数）的 PIC18 内核指令在执行时使用快速操作存储区。当 “a”等于 1时，指令使用BSR和包含在操作码中的8位地址对数据存储器寻址。当 “a”为 0 时，强制指令使用快速操作存储区地址映射，此时忽略 BSR 的当前值。

此“强制”寻址方式可使指令在一个周期内对数据地址进行操作，而不需要首先更新 BSR。这意味着用户可以更有效地对 8 位地址为 60h 或以上的 SFR 进行取值和操作。地址为 60h 以下的快速操作 RAM 非常适合于存储那些用户可能需要快速访问的数据值，如直接计算结果或常用程序变量。快速操作 RAM 也可实现更加快速和有效的现场保护和变量切换。

使能扩展的指令集 （XINST 配置位 = 1）时的快速操作存储区的映射略有不同。在**第 5.6.3 节 “在立即数变址模式中映射快速操作存储区”**中对此进行了更详细的讨论。

5.3.4　　通用寄存器文件

PIC18 器件可能在 GPR 区中划分了一部分存储区。这部分存储区为数据 RAM，所有指令均可访问它。 GPR 区从 Bank 0 的底部 （地址 000h）开始向上直到 SFR 区的底部。上电复位不会初始化 GPR，并且其他复位也不会改变其内容。

5.3.5 特殊功能寄存器

特殊功能寄存器（SFR）是 CPU 和外设模块用来控制器件操作的寄存器。这些寄存器用作数据存储器空间的静态 RAM。SFR 从数据存储器的顶部（FFFh）开始向下，它占用了 Bank 15 上半部分的空间（F60h 到 FFFh）。表 5-1 和表 5-2 列出了这些寄存器。

可以将 SFR 归类为两组：与“内核”器件功能（ALU、复位和中断）相关的寄存器和与外设功能相关的寄存器。复位和中断寄存器在相关的章节中进行讨论，本章后面的部分将对 ALU 状态寄存器进行说明。与外设操作相关的寄存器将在该外设的章节中进行说明。

SFR 通常分布在受其控制的外设中。未使用的 SFR 单元是未实现的，读为 0。

表 5-1： PIC18F2450/4450 器件的特殊功能寄存器映射图

地址	名称	地址	名称	地址	名称	地址	名称	地址	名称
FFFh	TOSU	FDFh	INDF2(1)	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEh	TOSH	FDEh	POSTINC2(1)	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2(1)	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2(1)	FBCh	—(2)	F9Ch	—(2)	F7Ch	UEP12
FFBh	PCLATU	FDBh	PLUSW2(1)	FBBh	—(2)	F9Bh	—(2)	F7Bh	UEP11
FFAh	PCLATH	FDAh	FSR2H	FBAh	—(2)	F9Ah	—(2)	F7Ah	UEP10
FF9h	PCL	FD9h	FSR2L	FB9h	—(2)	F99h	—(2)	F79h	UEP9
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	—(2)	F78h	UEP8
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	—(2)	F97h	—(2)	F77h	UEP7
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	—(2)	F96h	TRISE(3)	F76h	UEP6
FF5h	TABLAT	FD5h	T0CON	FB5h	—(2)	F95h	TRISD(3)	F75h	UEP5
FF4h	PRODH	FD4h	—(2)	FB4h	—(2)	F94h	TRISC	F74h	UEP4
FF3h	PRODL	FD3h	OSCCON	FB3h	—(2)	F93h	TRISB	F73h	UEP3
FF2h	INTCON	FD2h	HLVDCON	FB2h	—(2)	F92h	TRISA	F72h	UEP2
FF1h	INTCON2	FD1h	WDTCON	FB1h	—(2)	F91h	—(2)	F71h	UEP1
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	—(2)	F70h	UEP0
FEFh	INDF0(1)	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—(2)	F6Fh	UCFG
FEEh	POSTINC0(1)	FCEh	TMR1L	FAEh	RCREG	F8Eh	—(2)	F6Eh	UADDR
FEDh	POSTDEC0(1)	FCDh	T1CON	FADh	TXREG	F8Dh	LATE(3)	F6Dh	UCON
FECh	PREINC0(1)	FCCh	TMR2	FACh	TXSTA	F8Ch	LATD(3)	F6Ch	USTAT
FEBh	PLUSW0(1)	FCBh	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FEAh	FSR0H	FCAh	T2CON	FAAh	—(2)	F8Ah	LATB	F6Ah	UEIR
FE9h	FSR0L	FC9h	—(2)	FA9h	—(2)	F89h	LATA	F69h	UIE
FE8h	WREG	FC8h	—(2)	FA8h	—(2)	F88h	—(2)	F68h	UIR
FE7h	INDF1(1)	FC7h	—(2)	FA7h	EECON2(1)	F87h	—(2)	F67h	UFRMH
FE6h	POSTINC1(1)	FC6h	—(2)	FA6h	EECON1	F86h	—(2)	F66h	UFRML
FE5h	POSTDEC1(1)	FC5h	—(2)	FA5h	—(2)	F85h	—(2)	F65h	—(2)
FE4h	PREINC1(1)	FC4h	ADRESH	FA4h	—(2)	F84h	PORTE	F64h	—(2)
FE3h	PLUSW1(1)	FC3h	ADRESL	FA3h	—(2)	F83h	PORTD(3)	F63h	—(2)
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	—(2)
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	—(2)
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	—(2)

注 **1：** 不是实际存在的寄存器。
2： 不可用的寄存器，读为 0。
3： 这些寄存器只在 40/44 引脚器件上存在。

© 2007 Microchip Technology Inc.

表 5-2： 寄存器文件汇总 （PIC18F2450/4450）

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	详情请见（页）
TOSU	—	—	—	栈顶最高字节 （TOS<20:16>）					---0 0000	49, 54
TOSH	栈顶高字节 （TOS<15:8>）								0000 0000	49, 54
TOSL	栈顶低字节 （TOS<7:0>）								0000 0000	49, 54
STKPTR	STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0	00-0 0000	49, 55
PCLATU	—	—	—	PC<20:16> 的保持寄存器					---0 0000	49, 54
PCLATH	PC<15:8> 的保持寄存器								0000 0000	49, 54
PCL	PC 低字节 （PC<7:0>）								0000 0000	49, 54
TBLPTRU	—	—	bit 21(1)	程序存储器表指针最高字节 （TBLPTR<20:16>）					--00 0000	49, 76
TBLPTRH	程序存储器表指针高字节 （TBLPTR<15:8>）								0000 0000	49, 76
TBLPTRL	程序存储器表指针低字节 （TBLPTR<7:0>）								0000 0000	49, 76
TABLAT	程序存储器表锁存器								0000 0000	49, 76
PRODH	乘积寄存器高字节								xxxx xxxx	49, 83
PRODL	乘积寄存器低字节								xxxx xxxx	49, 83
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	49, 87
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1	49, 88
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00	49, 89
INDF0	使用 FSR0 的内容寻址数据存储器——FSR0 的值不变 （不是实际存在的寄存器）								N/A	49, 68
POSTINC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后增 （不是实际存在的寄存器）								N/A	49, 69
POSTDEC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后减 （不是实际存在的寄存器）								N/A	49, 69
PREINC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值预增 （不是实际存在的寄存器）								N/A	49, 69
PLUSW0	使用 FSR0 的内容寻址数据存储器——FSR0 的值预增 （不是实际存在的寄存器）， FSR0 的偏移量由 W 寄存器提供								N/A	49, 69
FSR0H	—	—	—	—	间接数据存储器地址指针 0 的高字节				---- 0000	49, 68
FSR0L	间接数据存储器地址指针 0 的低字节								xxxx xxxx	49, 68
WREG	工作寄存器								xxxx xxxx	49,
INDF1	使用 FSR1 的内容寻址数据存储器——FSR1 的值不变 （不是实际存在的寄存器）								N/A	49, 68
POSTINC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值后增 （不是实际存在的寄存器）								N/A	49, 69
POSTDEC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值后减 （不是实际存在的寄存器）								N/A	49, 69
PREINC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值预增 （不是实际存在的寄存器）								N/A	49, 69
PLUSW1	使用 FSR1 的内容寻址数据存储器——FSR1 的值预增 （不是实际存在的寄存器）， FSR1 的偏移量由 W 寄存器提供								N/A	49, 69
FSR1H	—	—	—	—	间接数据存储器地址指针 1 的高字节				---- 0000	49, 68
FSR1L	间接数据存储器地址指针 1 的低字节								xxxx xxxx	49, 68
BSR	—	—	—	—	存储区选择寄存器				---- 0000	49, 59
INDF2	使用 FSR2 的内容寻址数据存储器——FSR2 的值不变 （不是实际存在的寄存器）								N/A	50, 68
POSTINC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值后增 （不是实际存在的寄存器）								N/A	50, 69
POSTDEC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值后减 （不是实际存在的寄存器）								N/A	50, 69
PREINC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值预增 （不是实际存在的寄存器）								N/A	50, 69
PLUSW2	使用 FSR2 的内容寻址数据存储器——FSR2 的值预增 （不是实际存在的寄存器）， FSR2 的偏移量由 W 寄存器提供								N/A	50, 69
FSR2H	—	—	—	—	间接数据存储器地址指针 2 的高字节				---- 0000	50, 68
FSR2L	间接数据存储器地址指针 2 的低字节								xxxx xxxx	50, 68
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	50, 66
TMR0H	Timer0 寄存器的高字节								0000 0000	50, 113
TMR0L	Timer0 寄存器的低字节								xxxx xxxx	50, 113
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	50, 111

图注： x = 未知， u = 不变， - = 未用， q = 值取决于具体条件。未使用阴影单元，读为 0。

注 1： TBLPTRU 的 bit 21 允许访问器件配置位。

2： SBOREN 位仅在 BOREN<1:0> = 01 时可用，否则该位读为 0。

3： 这些寄存器和 / 或位在 28 引脚的器件上不存在，读为 0。显示的是 40/44 引脚器件的复位值；单独的未用位表示为 -。

4： 根据不同的主振荡器模式， RA6 可配置为端口引脚。禁止端口引脚时，所有相关位均读为 0。

5： 在 MCLRE 配置位清零时， RE3 仅可用作端口引脚，否则该位读为 0。

6： 当 USB 模块被禁止时 （UCON<3> = 0）， RC5 和 RC4 只能用作端口引脚。

表 5-2： 寄存器文件汇总 （PIC18F2450/4450）（续）

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	详情请见（页）
OSCCON	IDLEN	—	—	—	OSTS	—	SCS1	SCS0	0--- q-00	50, 31
HLVDCON	VDIRMAG	—	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0-00 0101	50, 183
WDTCON	—	—	—	—	—	—	—	SWDTEN	--- ---0	50, 202
RCON	IPEN	SBOREN(2)	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0q-1 11q0	50, 42
TMR1H	Timer1 寄存器的高字节								xxxx xxxx	50, 119
TMR1L	Timer1 寄存器的低字节								xxxx xxxx	50, 119
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	50, 115
TMR2	Timer2 寄存器								0000 0000	50, 122
PR2	Timer2 周期寄存器								1111 1111	50, 122
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	50, 121
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	50, 182
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	50, 182
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	--00 0000	50, 173
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 qqqq	50, 174
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	50, 175
CCPR1H	捕捉 / 比较 /PWM 寄存器 1 的高字节								xxxx xxxx	50, 124
CCPR1L	捕捉 / 比较 /PWM 寄存器 1 的低字节								xxxx xxxx	50, 124
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	50, 123,
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	51, 156,
SPBRGH	EUSART 波特率发生器寄存器的高字节								0000 0000	50, 157
SPBRG	EUSART 波特率发生器寄存器的低字节								0000 0000	50, 157
RCREG	EUSART 接收寄存器								0000 0000	50, 164
TXREG	EUSART 发送寄存器								0000 0000	50, 162
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	51, 154
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	51, 155
EECON2	数据存储器控制寄存器 2（不是实际存在的寄存器）								0000 0000	51, 74
EECON1	—	CFGS	—	FREE	WRERR	WREN	WR	—	x--- x000	51, 75
IPR2	OSCFIP	—	USBIP	—	—	HLVDIP	—	—	1-1- -1--	51, 95
PIR2	OSCFIF	—	USBIF	—	—	HLVDIF	—	—	0-0- -0--	51, 91
PIE2	OSCFIE	—	USBIE	—	—	HLVDIE	—	—	0-0- -0--	51, 93
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	51, 94
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	51, 90
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	51, 92
TRISE(3)	—	—	—	—	—	TRISE2	TRISE1	TRISE0	---- -111	51, 110
TRISD(3)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	51, 108
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	11-- -111	51, 106
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	51, 103
TRISA	—	TRISA6(4)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	-111 1111	51, 100
LATE(3)	—	—	—	—	—	LATE2	LATE1	LATE0	---- -xxx	51, 110
LATD(3)	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx	51, 108
LATC	LATC7	LATC6	—	—	—	LATC2	LATC1	LATC0	xx-- -xxx	51, 106
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	51, 103
LATA	—	LATA6(4)	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	-xxx xxxx	51, 100
PORTE	—	—	—	—	RE3(5)	RE2(3)	RE1(3)	RE0(3)	---- x000	51, 109
PORTD(3)	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	51, 108
PORTC	RC7	RC6	RC5(6)	RC4(6)	—	RC2	RC1	RC0	xxxx -xxx	51, 106

图注： x = 未知， u = 不变， - = 未用， q = 值取决于具体条件。未使用阴影单元，读为 0。

注 1： TBLPTRU 的 bit 21 允许访问器件配置位。

2： SBOREN 位仅在 BOREN<1:0> = 01 时可用，否则该位读为 0。

3： 这些寄存器和 / 或位在 28 引脚的器件上不存在，读为 0。显示的是 40/44 引脚器件的复位值；单独的未用位表示为 -。

4： 根据不同的主振荡器模式， RA6 可配置为端口引脚。禁止端口引脚时，所有相关位均读为 0。

5： 在 MCLRE 配置位清零时， RE3 仅可用作端口引脚，否则该位读为 0。

6： 当 USB 模块被禁止时 （UCON<3> = 0）， RC5 和 RC4 只能用作端口引脚。

© 2007 Microchip Technology Inc.

表 5-2：　　寄存器文件汇总 （PIC18F2450/4450）（续）

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	详情请见（页）
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	51, 100
PORTA	—	RA6(4)	RA5	RA4	RA3	RA2	RA1	RA0	-x0x 0000	51, 100
UEP15	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP14	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP13	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP12	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP11	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP10	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP9	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	51, 135
UEP8	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP7	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP6	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP5	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP4	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP3	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP2	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP1	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UEP0	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	---0 0000	52, 135
UCFG	UTEYE	UOEMON	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0	00-0 0000	52, 132
UADDR	—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	-000 0000	52, 136
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	-0x0 000-	52, 130
USTAT	—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	—	-xxx xxx-	52, 134
UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0--0 0000	52, 147
UEIR	BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0--0 0000	52, 146
UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	-000 0000	52, 145
UIR	—	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	-000 0000	52, 144
UFRMH	—	—	—	—	—	FRM10	FRM9	FRM8	---- -xxx	52, 136
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	xxxx xxxx	52, 136

图注： x = 未知， u = 不变， - = 未用， q = 值取决于具体条件。未使用阴影单元，读为 0。

注 1： TBLPTRU 的 bit 21 允许访问器件配置位。

2： SBOREN 位仅在 BOREN<1:0> = 01 时可用，否则该位读为 0。

3： 这些寄存器和 / 或位在 28 引脚的器件上不存在，读为 0。显示的是 40/44 引脚器件的复位值；单独的未用位表示为 -。

4： 根据不同的主振荡器模式， RA6 可配置为端口引脚。禁止端口引脚时，所有相关位均读为 0。

5： 在 MCLRE 配置位清零时， RE3 仅可用作端口引脚，否则该位读为 0。

6： 当 USB 模块被禁止时 （UCON<3> = 0）， RC5 和 RC4 只能用作端口引脚。

5.3.6 状态寄存器

如寄存器 5-2 所示，状态（Status）寄存器包含 ALU 的算术运算状态。和任何其他 SFR 一样，它也可以作为任何指令的操作数。

如果一条影响 Z、DC、C、OV 或 N 位的指令以状态寄存器作为目标寄存器，将不会直接写入结果，而是根据指令的执行结果更新寄存器状态位。因此，当执行一条把状态寄存器作为目标寄存器的指令后，状态寄存器的结果可能和预想的不一样。例如，`CLRF STATUS` 将 Z 位置 1 并保持其余状态位不变（`000u u1uu`）。

因此，建议仅使用 BCF、BSF、SWAPF、MOVFF 和 MOVWF 指令来改变状态寄存器，因为这些指令不会影响状态寄存器中的 Z、C、DC、OV 或 N 位。

关于其他不会影响状态位的指令，请参见表 19-2 和表 19-3 中的指令集汇总。

注：	在减法运算中，C 和 DC 位分别作为借位位和辅助借位位。

寄存器 5-2： 状态寄存器

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC(1)	C(2)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 **未用：** 读为 0

bit 4 **N：** 负数位
此位用于有符号的算术运算（2 进制补码）。它表明结果是否为负（ALU MSB = 1）。
1 = 结果为负
0 = 结果为正

bit 3 **OV：** 溢出位
此位用于有符号的算术运算（2 进制补码）。表明运算结果溢出了 7 位二进制数的范围，溢出导致符号位（bit 7）发生改变。
1 = 有符号算术运算中发生溢出（本次运算）
0 = 未发生溢出

bit 2 **Z：** 零位
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：** 辅助进位 / 借位位 (1)
用于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令：
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 (2)
用于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令：
1 = 结果中最高位发生了进位
0 = 结果中最高位未发生进位

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 2 进制补码来执行的。对于移位指令（RRF 和 RLF），此位来自源寄存器的 bit 4 或 bit 3。

2： 对于借位，极性是相反的。减法是通过加上第二个操作数的 2 进制补码来执行的。对于移位指令（RRF 和 RLF），此位来自源寄存器的最高位或最低位。

© 2007 Microchip Technology Inc.

5.4 数据寻址模式

注： 当使能 PIC18 扩展指令集时，PIC18 内核指令集中某些指令的执行方式会发生改变。更多信息，请参见**第 5.6 节 “数据存储器和扩展指令集”**。

程序存储器只能用一种方式寻址（通过程序计数器），而数据存储器空间的信息可用多种方式寻址。大部分指令的寻址模式都是固定的。其他指令可能使用最多三种模式，根据它们所使用的操作数和是否使能了扩展指令集而定。

这些寻址模式为：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

当使能了扩展指令集时（XINST 配置位 = 1），还可使用其他寻址模式，即立即数变址寻址模式。**第 5.6.1 节 “使用立即数偏移量进行变址寻址”**将更详细讨论它的操作。

5.4.1 固有和立即数寻址

很多 PIC18 控制指令根本不需要任何参数；执行这些指令要么对整个器件造成影响，要么仅针对一个寄存器进行操作。此寻址模式就是固有寻址。例如指令 SLEEP、RESET 和 DAW。

其他指令的工作方式与此类似，但需要操作码中有其他的参数。由于需要一些立即数作为参数，这种寻址模式被称为立即数寻址。例如 ADDLW 和 MOVLW，它们分别向 W 寄存器添加或从中移除立即数值。其他立即数寻址指令，例如 CALL 和 GOTO，它们包括 20 位的程序存储器地址。

5.4.2 直接寻址

直接寻址方式在操作码中指定操作的全部或部分源地址和 / 或目标地址。此选项由指令附带的参数指定。

在 PIC18 内核指令集中，针对位和针对字节的指令默认情况下使用直接寻址。所有这些指令都包含某个 8 位的直接地址作为它们的最低有效字节。此地址指定数据 RAM 的某个存储区中寄存器的地址（**第 5.3.4 节 “通用寄存器文件”**）或快速操作存储区（**第 5.3.3 节 “快速操作存储区”**）中作为指令数据源的单元地址。

快速操作 RAM 位 “a” 决定地址的解析方式。当 “a” 为 1 时，BSR（**第 5.3.2 节 “存储区选择寄存器（BSR）”**）的内容将和指令中的直接地址一起用于确定寄存器的完整 12 位地址。当 “a” 为 0 时，此直接地址将被解析为快速操作存储区中的一个寄存器。使用快速操作 RAM 的寻址模式有时也被称为直接强制寻址模式。

有几个指令，例如 MOVFF，在操作码中包含完整的 12 位地址（源地址或目标地址）。在这些情况下，BSR 被完全忽略。

操作目标地址由目标位 “d” 确定。当 “d” 为 1 时，结果被存回源寄存器并覆盖原来的内容。当 “d” 为 0 时，结果被存储在 W 寄存器中。没有 “d” 参数的指令的目标地址是隐含的，它们是正在操作的目标寄存器或 W 寄存器。

5.4.3 间接寻址

间接寻址允许用户访问数据存储器中的单元而无需在指令中给出一个固定的地址。这种寻址方式是通过使用文件选择寄存器（File Select Register，FSR）作为指向被读写单元的指针实现的。由于 FSR 本身作为特殊功能寄存器位于 RAM 中，因此也可在程序中直接控制它们。这使得 FSR 对于在数据存储器中实现诸如表和数组等数据结构非常有用。

也可以使用间接文件操作数（Indirect File Operand，INDF）进行间接寻址。这种操作允许自动递增、递减或偏移指针，从而自动控制指针的值。它通过使用循环提高代码执行效率，如例 5-5 所示的清零整个 RAM 存储区的操作。

例 5-5： 使用间接寻址清零 RAM（BANK 1）

```
        LFSR    FSR0, 100h  ;
NEXT    CLRF    POSTINC0    ; Clear INDF
                            ; register then
                            ; inc pointer
        BTFSS   FSR0H, 1    ; All done with
                            ; Bank1?
        BRA     NEXT        ; NO, clear next
CONTINUE                    ; YES, continue
```

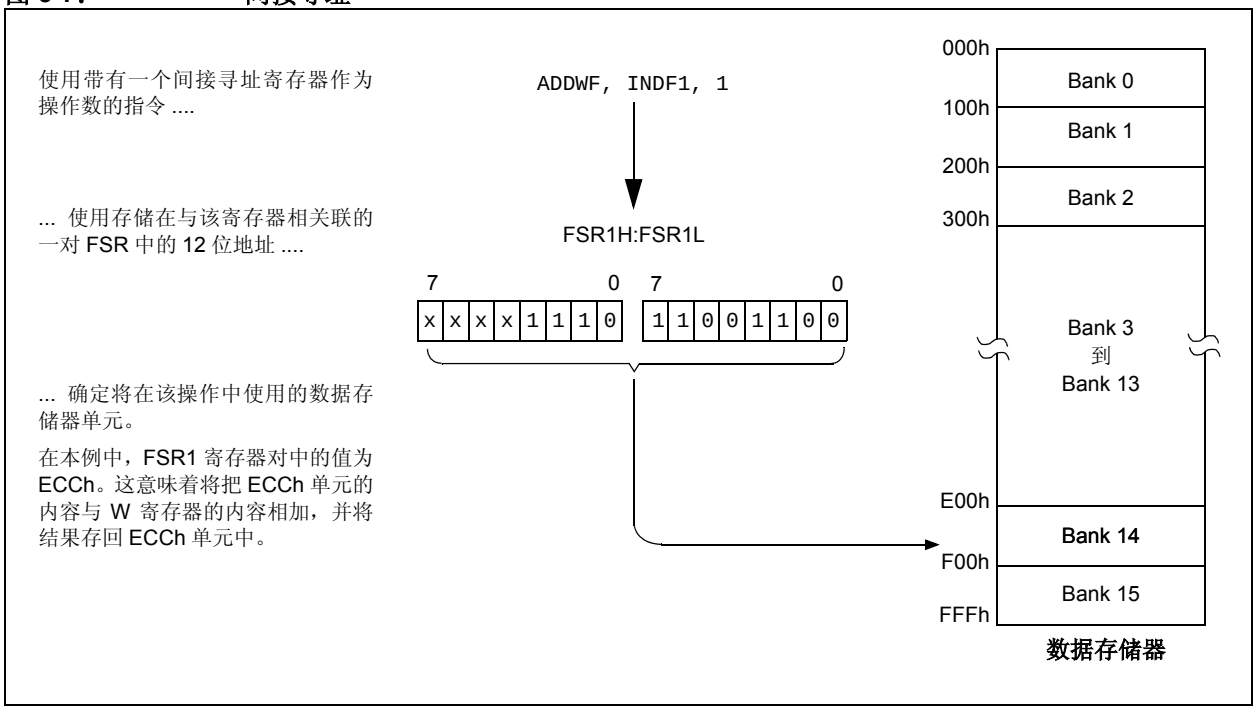
5.4.3.1 FSR 寄存器和 INDF 操作数

间接寻址的核心是三组寄存器：FSR0、FSR1 和 FSR2。每组寄存器都含有一对 8 位寄存器：FSRnH 和 FSRnL。FSRnH 寄存器的高 4 位未使用，因此每对 FSR 只保存一个 12 位值，从而可以线性寻址数据存储器的整个空间。因此， FSR 寄存器对被用作数据存储器的地址指针。

间接寻址是通过一组间接文件操作数（INDF0 到 INDF2）完成的。这些操作数可被看作“虚拟”寄存器：它们被映射到 SFR 空间而不是通过物理方式实现的。对特定的 INDF 寄存器执行读或写操作实际上访问的是与之对应的一对 FSR 寄存器。例如，读 INDF1 就是读 FSR1H:FSR1L 指向单元中的数据。使用 INDF 寄存器作为操作数的指令实际上使用的是相应的FSR的内容，该内容为指向目标地址的指针。INDF 操作数只是使用该指针的一种简便方法。

由于间接寻址使用完整的 12 位地址，因此没有必要进行数据 RAM 分区。所以 BSR 的当前内容和快速操作 RAM 位对于确定目标地址没有影响。

图 5-7： 间接寻址



 © 2007 Microchip Technology Inc.

5.4.3.2 FSR 寄存器和 POSTINC、POSTDEC、 PREINC 以及 PLUSW

除了 INDF 操作数之外，每对 FSR 寄存器还有 4 个额外的间接操作数。和 INDF 一样，它们也都是不能直接读写的“虚拟”寄存器。访问这些寄存器实际上访问的是与之相关的一对FSR寄存器，并对其存储的内容进行特定的操作。这些寄存器是：

- POSTDEC：访问 FSR 的值，然后将它自动减 1
- POSTINC：访问 FSR 的值，然后将它自动加 1
- PREINC：将 FSR 的值加 1，然后在操作中使用该值
- PLUSW：将 W 寄存器中有符号的值 （-127 到 128）与 FSR 寄存器中的值相加，并在操作中使用得到的新值。

在本文中使用 FSR 寄存器中的值 （不会更改此值）访问 INDF 寄存器。同样，访问 PLUSW 寄存器是将 W 寄存器中的值作为FSR的偏移量。该操作不会改变这两个寄存器中的值，而访问其他虚拟寄存器均会更改FSR寄存器的值。

使用 POSTDEC、 POSTINC 和 PREINC 对 FSR 进行操作会影响整对寄存器 FSRnL 寄存器从 FFh 到 00h 溢出时会向 FSRnH 寄存器进位。但这些操作的结果不会更改状态寄存器中的标志位 （如 Z、 N 和 OV 等）。

PLUSW 寄存器可用于在数据存储器空间实现变址寻址。通过控制 W 寄存器中的值，用户可以访问相对当前指针地址有固定偏移量的地址单元。在某些应用中，该功能可被用于在数据存储器内部实现某些强大的程序控制结构，如软件堆栈。

5.4.3.3 通过 FSR 对其他 FSR 进行操作

在某些特殊情况下，间接寻址操作以其他FSR或虚拟寄存器作为目标。例如，使用 FSR 指向一个虚拟寄存器会导致操作不成功。假设如下特殊情况：FSR0H:FSR0L 保存的是 INDF1 的地址 FE7h。尝试使用 INDF0 作为操作数读取 INDF1 的值，将返回 00h。尝试使用 INDF0 作为操作数写入 INDF1，将会导致执行一条 NOP 指令。

另一方面，使用虚拟寄存器对一对FSR寄存器进行写操作可能会产生与预期不同的结果。在这些情况下，会将值写入一对 FSR 寄存器，但 FSR 不会递增或递减。因此，写入 INDF2 或 POSTDEC2 时会把同样的值写入 FSR2H:FSR2L。

由于 FSR 是映射到 SFR 空间中的物理寄存器，所以可以通过直接寻址对它们进行操作。用户在使用这些寄存器时应该特别小心，尤其是在代码使用间接寻址的方式时。

同样，通常允许通过间接寻址对所有其他 SFR 进行操作。用户在进行此类操作时应该特别小心，以免不小心更改设置从而影响器件操作。

5.5 程序存储器和扩展指令集

程序存储器的操作不受扩展指令集的影响。

使能扩展指令集将 8 条额外的双字命令添加到现有的 PIC18 指令集中：ADDFSR、ADDULNK、CALLW、MOVSF、MOVSS、PUSHL、SUBFSR 和 SUBULNK。这些指令如**第 5.2.4 节 “双字指令”**中所述执行。

5.6 数据存储器和扩展指令集

使能 PIC18 扩展指令集（XINST 配置位 =1）显著改变了数据存储器及其寻址的某些方面。特别是许多 PIC18 内核指令使用快速操作存储区的方式有所不同。这是由于扩展指令集引入了对数据存储器空间的新的寻址模式。该模式还会更改使用 FSR2 及其相关操作数进行间接寻址的方式。

同样需要了解哪些部分保持不变。数据存储器空间的大小及其线性寻址方式都不会改变。 SFR 映射也保持不变。PIC18 内核指令也仍然以直接和间接寻址模式进行操作；固有和立即数指令操作照旧。 FSR0 和 FSR1 的间接寻址方式也保持不变。

5.6.1 使用立即数偏移量进行变址寻址

使能 PIC18 扩展指令集将更改使用 FSR2 寄存器对及其相关文件操作数进行间接寻址的方式。在适当的条件下，使用快速操作存储区的指令（即绝大多数针对位和针对字节的指令）可以利用指令中的偏移量来执行变址寻址。这种特定的寻址模式被称为使用立即数偏移量的变址寻址或立即数变址寻址模式。

使用扩展指令集时，这种寻址模式有如下要求：

- 强制使用快速操作存储区 （a = 0）；且
- 文件地址参数要小于或等于 5Fh。

在这些条件下，指令的文件地址不会被解析为地址的低字节 （在直接寻址中和 BSR 一起使用），或快速操作存储区中的 8 位地址，而是被解析为由 FSR2 指定的地址指针的偏移量。将该偏移量与 FSR2 的内容相加以获取操作的目标地址。

5.6.2 受立即数变址寻址模式影响的指令

任何采用直接寻址模式的 PIC18 内核指令均会受到立即数变址寻址模式的潜在影响，包括所有针对字节和针对位的指令，或标准 PIC18 指令集中几乎一半的指令。只有使用固有或立即数寻址模式的指令不受影响。

此外，如果针对字节和针对位的指令使用快速操作存储区 （快速操作 RAM 位为 1）或包含 60h 以上的文件地址，它们也不受影响。符合这些条件的指令会像以前一样执行。图 5-8 显示了当使能扩展指令集时，各种寻址模式之间的对比。

那些想要在立即数变址寻址模式中使用针对位或针对字节的指令的用户，应该注意此模式下汇编语法的改变。在**第 19.2.1 节 “扩展指令的语法”**中对此进行了更详细的说明。

 © 2007 Microchip Technology Inc.

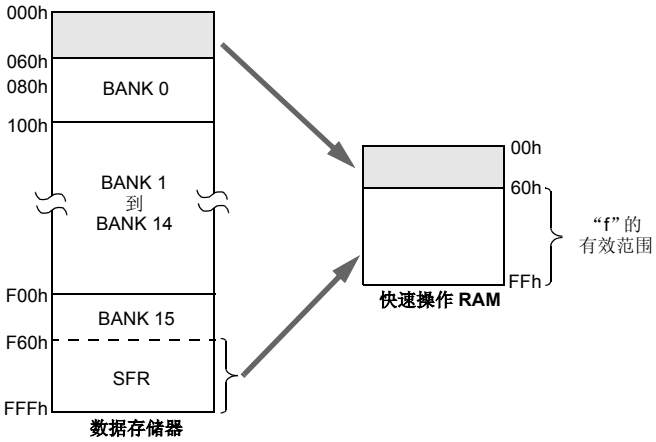
图 5-8：　　针对位和针对字节的指令的寻址方式对比 （使能了扩展指令集）

示例指令： ADDWF, f, d, a（操作码：0010 01da ffff ffff）

当 a = 0 且 f ≥ 60h 时：

此指令以直接强制模式执行。“f”被解析为快速操作RAM中060h 到 0FFh 之间的单元地址，该地址也是 SFR 或数据存储器的 F60h 到 0FFh （Bank 15）。

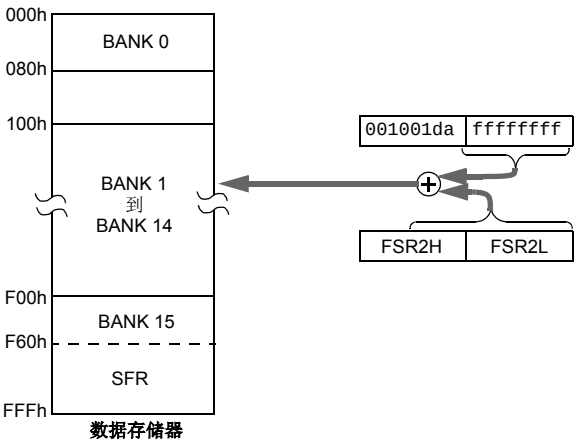
不可用此模式寻址地址低于60h 的单元。



当 a = 0 且 f ≤ 5Fh 时：

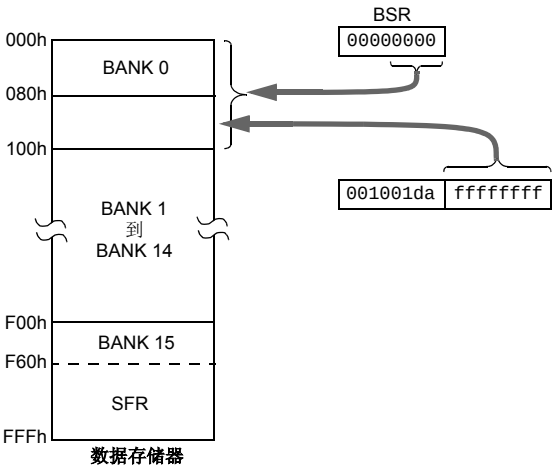
此指令以立即数变址寻址模式执行。“f”被解析为 FSR2 中地址值的偏移量。将这两个值相加可以得到指令的目标寄存器的地址。此地址可以在数据存储器空间的任何地方。

注意在此模式中，正确的语法如下：
ADDWF [k], d
其中 “k”就是 “f”。



当 a = 1（f 可为任何值）时：

指令以直接寻址模式 （也称为直接长地址寻址模式）执行。“f”被解析为数据存储器空间的16个存储区中的一个单元地址。存储区由存储区选择寄存器 （BSR）指定。此地址可以在数据存储器空间的任何地方。



5.6.3 在立即数变址模式中映射快速操作存储区

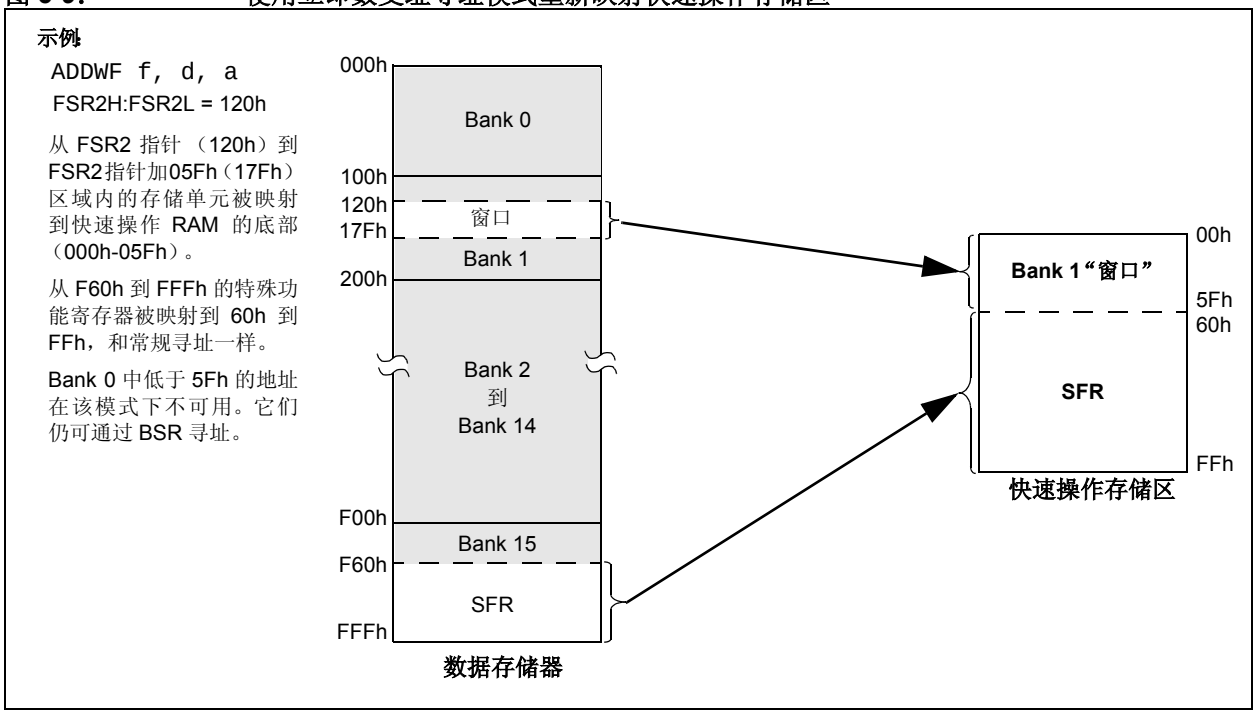
使用立即数变址寻址模式能有效改变快速操作 RAM 低地址单元（00h 到 5Fh）的映射方式。此模式映射 Bank 0 的内容和由用户定义的、可位于数据存储器空间中任何地方的“窗口”内容，而不仅仅映射 Bank 0 底部的内容。 FSR2 的值定义映射到窗口的地址的下边界，而上边界则由 FSR2 加 95 （5Fh）决定。地址为 5Fh 以上的快速操作 RAM 的映射方法如前所述 （见**第 5.3.3 节 “快速操作存储区”**）。图 5-9 显示了在此寻址模式下重新映射的快速操作存储区示例。

快速操作存储区的重新映射*仅*适用于立即数变址寻址模式。使用 BSR （快速操作 RAM 位为 1）的操作和以前一样继续使用直接寻址模式。任何明确使用间接文件操作数 （包括 FSR2）的间接或变址操作都将像标准间接寻址一样操作。任何使用快速操作存储区的指令（除非包括大于 05Fh 的寄存器地址）都将使用直接寻址和常规的快速操作存储区映射。

5.6.4 立即数变址模式中的 BSR

尽管使能扩展指令集时会重新映射快速操作存储区，但 BSR 的操作不变。使用 BSR 选择数据存储区的直接寻址操作方式和以前描述的相同。

图 5-9： 使用立即数变址寻址模式重新映射快速操作存储区



 © 2007 Microchip Technology Inc.

6.0 闪存程序存储器

正常工作状态下，闪存程序存储器在整个 VDD 范围内都是可读写可擦除的。

读程序存储器时，每次读取一个字节。写程序存储器时，每次写入一个 16 字节的块。擦除程序存储器时，每次擦除一个 64 字节的块。用户代码不能执行批量擦除操作。

在擦写程序存储器时，系统会停止取指令直到操作完成。擦写期间不能访问程序存储器，因此也就无法执行代码。由内部编程定时器来终止程序存储器的擦写操作。

写入程序存储器的值不一定非要是有效指令。执行存储无效指令的程序存储器单元会导致执行 NOP。

6.1 表读与表写

为了读写程序存储器，有两个操作指令可供处理器在程序存储器空间和数据 RAM 之间移动字节：

- 表读 （TBLRD）
- 表写 （TBLWT）

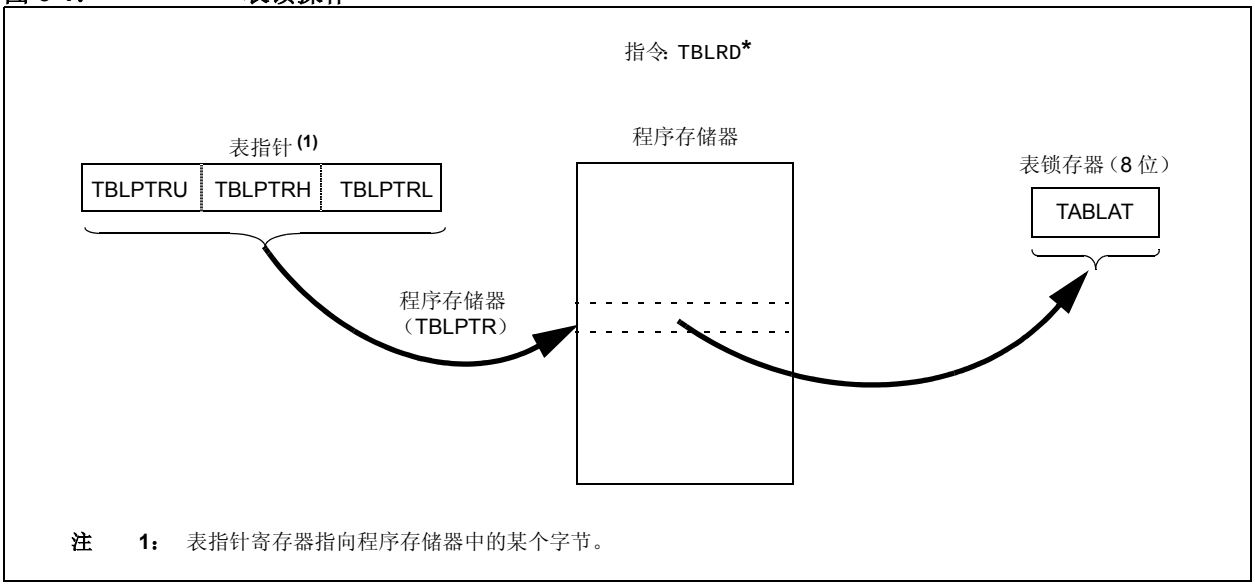
程序存储器空间为 16 位宽，而数据 RAM 空间为 8 位宽。表读和表写操作通过一个 8 位寄存器 （TABLAT）在这两个存储器空间之间传送数据。

表读操作从程序存储器获取数据并将其放入数据 RAM 空间。图 6-1 显示了程序存储器和数据 RAM 之间的一次表读操作。

表写操作将数据存储器中的数据存入程序存储器的保持寄存器。**第 6.5 节 “写闪存程序存储器”**详细介绍了将保持寄存器内容写入程序存储器的过程。图 6-2 显示了程序存储器和数据 RAM 之间的一次表写操作。

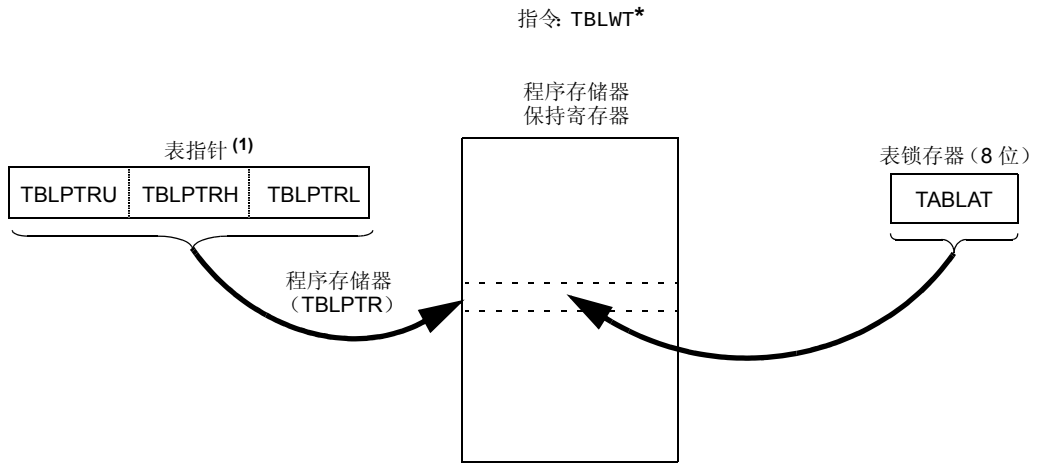
表操作以字节为单位。包含数据而非程序指令的表块不需要按字对齐。因此，表块可以在任何字节地址开始和结束。如果使用表写操作向程序存储器写入可执行代码，程序指令必须按字对齐。

图 6-1： 表读操作



注 1： 表指针寄存器指向程序存储器中的某个字节。

图 6-2：　　表写操作



注　1： 表指针实际指向 16 个保持寄存器中的某一个，其地址由 TBLPTRL<3:0> 决定。**第 6.5 节 “写闪存程序存储器”** 中将讨论向程序存储器阵列写入数据的物理过程。

6.2　控制寄存器

TBLRD 和 TBLWT 指令要用到几个控制寄存器。包括：

- EECON1 寄存器
- EECON2 寄存器
- TABLAT 寄存器
- TBLPTR 寄存器

6.2.1　EECON1 和 EECON2 寄存器

EECON1 寄存器（寄存器 6-1）是存储器访问的控制寄存器。EECON2 寄存器不是实际存在的寄存器，专用于存储器的擦写序列。读 EECON2 将得到全 0。

控制位 CFGS 决定访问的是配置 / 校准寄存器还是程序存储器。

当 FREE 位置 1 时，允许对程序存储器进行擦除操作。当 FREE 置 1 时，擦除操作由下一条 WR 命令启动。当 FREE 清零时，则仅使能写操作。

当 WREN 位置 1 时，允许进行写操作。上电时，WREN 位被清零。WRERR 位在 WREN 位置 1 时由硬件置 1，在内部编程定时器超时、写操作结束时被清零。

注：	在正常工作期间，WRERR 读为 1。这表明写操作被复位提早终止，或进行了不合法的写操作。

WR 控制位用于启动写操作。用软件只能将该位置 1 而无法清零。在写操作完成后，由硬件将其清零。

© 2007 Microchip Technology Inc.

寄存器 6-1：　　EECON1：存储器控制寄存器 1

U-0	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	U-0
—	CFGS	—	FREE	WRERR(1)	WREN	WR	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　　**未用：**读为 0

bit 6　　**CFGS：**闪存程序存储器或配置寄存器选择位
1 = 访问配置寄存器
0 = 访问闪存程序存储器

bit 5　　**未用：**读为 0

bit 4　　**FREE：**闪存行擦除使能位
1 = 在下一条 WR 命令时擦除 TBLPTR 指定的程序存储器行 （擦除操作完成后清零）
0 = 仅执行写操作

bit 3　　**WRERR：**闪存程序存储器错误标志位 (1)
1 = 写操作提早终止 （正常操作中自定时编程期间的任何复位，或不合法的写操作）
0 = 写操作完成

bit 2　　**WREN：**闪存程序存储器写使能位
1 = 允许对闪存程序存储器的写周期
0 = 禁止对闪存程序存储器的写周期

bit 1　　**WR：**写控制位
1 = 启动程序存储器的擦写周期 （操作是自定时的，一旦写操作完成，该位即由硬件清零。用软件只能将 WR 位置 1，但不能清零。）
0 = 写周期完成

bit 0　　**未用：**读为 0

注　1： 当发生 WRERR 时， CFGS 位不会被清零。这样可以跟踪错误状况。

6.2.2 表锁存寄存器 （TABLAT）

表锁存器（TABLAT）是映射到 SFR 空间的一个 8 位寄存器。表锁存器用于在程序存储器和数据 RAM 之间传输数据时保存 8 位数据。

6.2.3 表指针寄存器 （TBLPTR）

表指针 （TBLPTR）寄存器在程序存储器中以字节为单位进行寻址。TBLPTR 由 3 个 SFR 寄存器组成：表指针最高字节、表指针高字节和表指针低字节（TBLPTRU:TBLPTRH:TBLPTRL）。这 3 个寄存器合起来组成一个 22 位宽的指针。其中低 21 位允许器件寻址高达 2MB 程序存储器空间。第 22 位则允许访问器件 ID、用户 ID 和配置位。

TBLRD 和 TBLWT 指令要使用表指针 TBLPTR。这些指令可以基于表操作以 4 种方法更新 TBLPTR。表 6-1 列出了这些操作。这些操作只会影响 TBLPTR 的低 21 位。

6.2.4 表指针范围

TBLPTR 用于读、写和擦除闪存程序存储器。

当执行 TBLRD 时，TBLPTR 的所有 22 位决定将程序存储器的哪个字节读入 TABLAT。

当执行 TBLWT 时，表指针寄存器的低 4 位（TBLPTR<3:0>）决定要写入程序存储器的哪个保持寄存器 （共有 16 个）。当程序存储器的定时写入（通过 WR 位）开始时，TBLPTR 的高 16 位 （TBLPTR<21:4>）将决定要写入哪个程序存储器块 （每块 16 字节）。更多详细信息，请参见**第 6.5 节 “写闪存程序存储器”**。

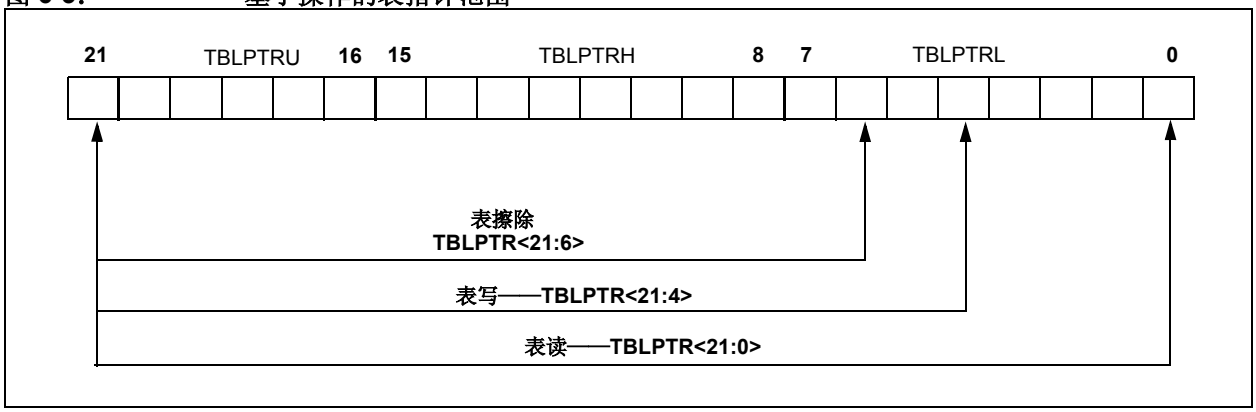
当执行擦除程序存储器时，表指针寄存器的高 16 位 （TBLPTR<21:6>）指向将要擦除的 64 字节块。低有效位 （TBLPTR<5:0>）被忽略。

图 6-3 说明了基于闪存程序存储器操作的 TBLPTR 相关范围。

表 6-1： 执行 TBLRD 和 TBLWT 指令的表指针操作

示例	表指针操作
TBLRD* TBLWT*	不修改 TBLPTR
TBLRD*+ TBLWT*+	TBLPTR 在读 / 写后递增
TBLRD*- TBLWT*-	TBLPTR 在读 / 写后递减
TBLRD+* TBLWT+*	TBLPTR 在读 / 写前递增

图 6-3： 基于操作的表指针范围



 © 2007 Microchip Technology Inc.

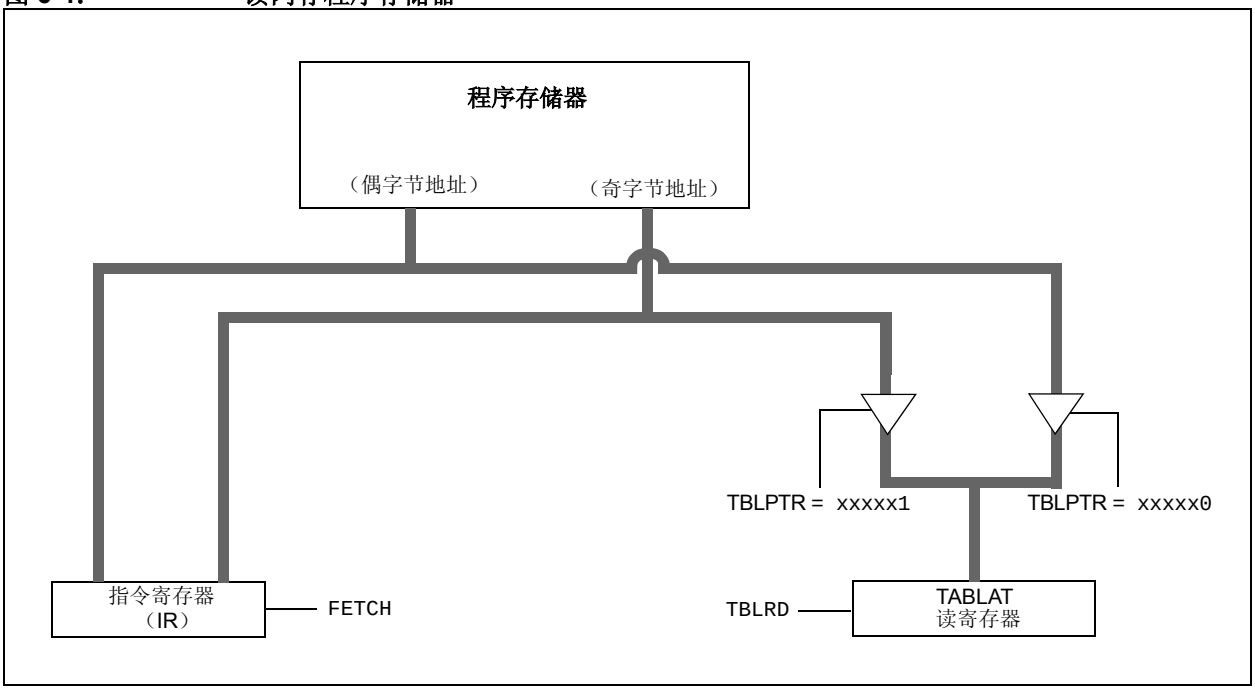
6.3 读闪存程序存储器

TBLRD 指令用于从程序存储器获取数据并放入数据 RAM。表读操作每次从程序存储器读取一个字节。

TBLPTR 指向程序存储器空间的某个字节地址。执行 TBLRD 指令将把指向的字节装入 TABLAT。此外，还可以自动修改 TBLPTR 以进行下一次表读操作。

内部程序存储器通常以字为单位进行组织。由地址的最低有效位来选择字的高字节或者低字节。图 6-4 显示了内部程序存储器和 TABLAT 之间的接口。

图 6-4：　　读闪存程序存储器



例 6-1：　　读闪存程序存储器的一个字

```
            MOVLW     CODE_ADDR_UPPER       ; Load TBLPTR with the base
            MOVWF     TBLPTRU               ; address of the word
            MOVLW     CODE_ADDR_HIGH
            MOVWF     TBLPTRH
            MOVLW     CODE_ADDR_LOW
            MOVWF     TBLPTRL
READ_WORD
            TBLRD*+                         ; read into TABLAT and increment
            MOVF      TABLAT, W             ; get data
            MOVWF     WORD_EVEN
            TBLRD*+                         ; read into TABLAT and increment
            MOVF      TABLAT, W             ; get data
            MOVF      WORD_ODD
```

6.4 擦除闪存程序存储器

最小擦除块大小为 32 个字或 64 字节。只有通过使用外部编程器，或通过 ICSP 控制，才能够批量擦除更大的程序存储器块。闪存阵列不支持字擦除。

当单片机本身开始一个擦除过程时，会擦除一个 64 字节的程序存储器块。 16 个最高有效位 TBLPTR<21:6> 指向要擦除的块。 TBLPTR<5:0> 被忽略。

擦除操作由 EECON1 寄存器控制。 WREN 位必须被置 1 以使能写操作。 FREE 位被置 1 以选择擦除操作。

为了安全起见，必须使用 EECON2 的写启动序列。

擦除内部闪存必须执行长写操作。在长写周期中，指令停止执行。由内部编程定时器终止长写操作。

6.4.1 闪存程序存储器擦除序列

擦除内部程序存储器块的步骤如下：

1. 将要擦除的行地址装入表指针寄存器。
2. 设置 EECON1 寄存器来执行擦除操作：
   - 将 CFGS 位清零以访问程序存储器；
   - 将 WREN 位置 1 以使能写操作；
   - 将 FREE 位置 1 以使能擦除操作。
3. 禁止中断。
4. 向 EECON2 写入 55h。
5. 向 EECON2 写入 0AAh。
6. 将 WR 位置 1。这将开始行擦除周期。
7. CPU 在擦除期间将会停止工作 （使用内部定时器约为 2 ms）。
8. 重新允许中断。

例 6-2： 擦除闪存程序存储器的一行

```
                MOVLW   CODE_ADDR_UPPER    ; load TBLPTR with the base
                MOVWF   TBLPTRU            ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
ERASE_ROW
                BCF     EECON1, CFGS       ; access Flash program memory
                BSF     EECON1, WREN       ; enable write to memory
                BSF     EECON1, FREE       ; enable Row Erase operation
                BCF     INTCON, GIE        ; disable interrupts
                MOVLW   55h
                MOVWF   EECON2             ; write 55h
必需的序列      MOVLW   0AAh
                MOVWF   EECON2             ; write 0AAh
                BSF     EECON1, WR         ; start erase (CPU stall)
                BSF     INTCON, GIE        ; re-enable interrupts
```


© 2007 Microchip Technology Inc.

6.5 写闪存程序存储器

最小编程块大小为 8 字或 16 字节。不支持字或字节编程。

在内部使用表写命令将需要写入闪存存储器的内容装入保持寄存器中。表写操作使用 16 个保持寄存器进行编程。

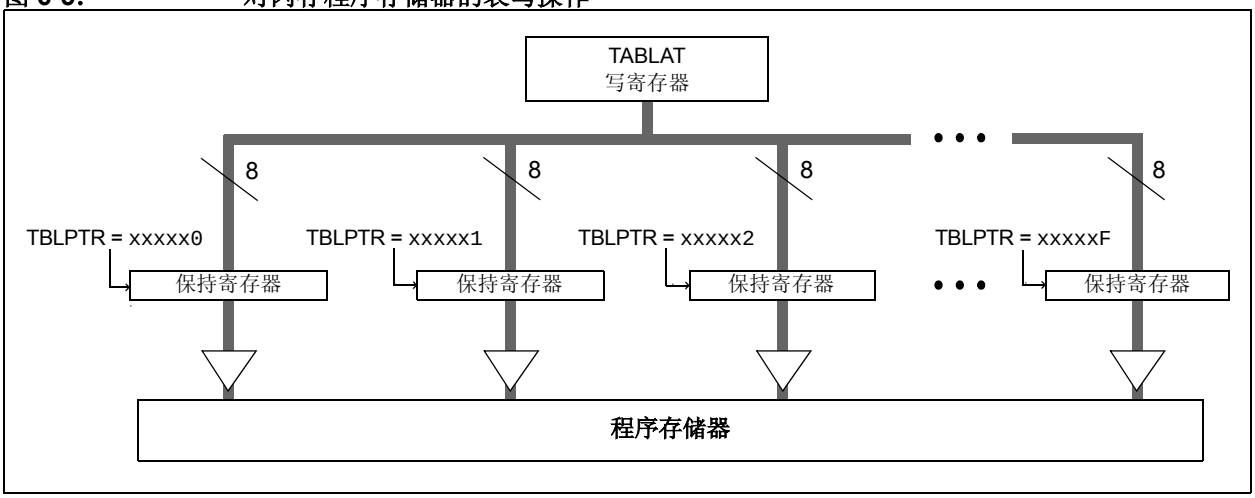
由于表锁存器 （TABLAT）只是单字节寄存器，所以每次编程操作， TBLWT 指令都必须执行 16 次。因为只写保持寄存器，所以所有的表写操作实际上都是短写。更新 16 个保持寄存器后，必须写 EECON1 寄存器，以便启动长写周期开始编程操作。

对内部闪存编程要求使用长写操作。在长写周期中，指令停止执行。由内部编程定时器终止长写操作。

写入 / 擦除电压由片上的电荷泵产生，该电荷泵可以工作在器件的电压范围内。

注：	器件复位和写操作完成后保持寄存器的默认值为 FFh。将 FFh 写入保持寄存器不会修改其中的数值。这意味着可以修改程序存储器的单独字节 （假如不是想把任何位从 0 更改为 1）。当修改单独字节时，无需在执行写操作前装入所有 16 个保持寄存器。

图 6-5： 对闪存程序存储器的表写操作



6.5.1 写入闪存程序存储器操作顺序

内部程序存储器单元的编程事件顺序应为：

1. 将 64 字节读入 RAM。
2. 必要时更新 RAM 中的数据值。
3. 把要擦除的目标地址装入表指针寄存器。
4. 执行行擦除。
5. 将要写入的第一个字节的地址装入表指针寄存器。
6. 将 16 个字节写入保持寄存器 （自动递增）。
7. 设置 EECON1 寄存器来执行写操作：
   - 将 CFGS 位清零以访问程序存储器；
   - 将 WREN 位置 1 以使能字节写操作。
8. 禁止中断。
9. 向 EECON2 写入 55h。
10. 向 EECON2 写入 0AAh。
11. 将 WR 位置 1。这将开始写周期。
12. CPU 在写入期间将会停止工作 （使用内部定时器约为 2 ms）。
13. 重新允许中断。
14. 再一次重复步骤 6 至 14 以写入全部 64 字节。
15. 验证存储器 （表读）。

此过程需要大约 8 ms 以更新存储器的一行 64 字节。例 6-3 给出了所需代码的示例。

注：	在将 WR 位置 1 前，表指针必须指向保持寄存器中的 16 个字节的地址范围内。

例 6-3：　写闪存程序存储器

```
                MOVLW   D'64'                   ; number of bytes in erase block
                MOVWF   COUNTER
                MOVLW   BUFFER_ADDR_HIGH        ; point to buffer
                MOVWF   FSR0H
                MOVLW   BUFFER_ADDR_LOW
                MOVWF   FSR0L
                MOVLW   CODE_ADDR_UPPER         ; Load TBLPTR with the base
                MOVWF   TBLPTRU                 ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
READ_BLOCK
                TBLRD*+                         ; read into TABLAT, and inc
                MOVF    TABLAT, W               ; get data
                MOVWF   POSTINC0                ; store data
                DECFSZ  COUNTER                 ; done?
                BRA     READ_BLOCK              ; repeat
MODIFY_WORD
                MOVLW   DATA_ADDR_HIGH          ; point to buffer
                MOVWF   FSR0H
                MOVLW   DATA_ADDR_LOW
                MOVWF   FSR0L
                MOVLW   NEW_DATA_LOW            ; update buffer word
                MOVWF   POSTINC0
                MOVLW   NEW_DATA_HIGH
                MOVWF   INDF0
ERASE_BLOCK
                MOVLW   CODE_ADDR_UPPER         ; load TBLPTR with the base
                MOVWF   TBLPTRU                 ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
                BCF     EECON1, CFGS            ; access Flash program memory
                BSF     EECON1, WREN            ; enable write to memory
                BSF     EECON1, FREE            ; enable Row Erase operation
                BCF     INTCON, GIE             ; disable interrupts

                MOVLW   55h
                MOVWF   EECON2                  ; write 55h
必需的序列      MOVLW   0AAh
                MOVWF   EECON2                  ; write 0AAh
                BSF     EECON1, WR              ; start erase (CPU stall)
                BSF     INTCON, GIE             ; re-enable interrupts
                TBLRD*-                         ; dummy read decrement
                MOVLW   BUFFER_ADDR_HIGH        ; point to buffer
                MOVWF   FSR0H
                MOVLW   BUFFER_ADDR_LOW
                MOVWF   FSR0L
                MOVLW   D'4'
                MOVWF   COUNTER1
WRITE_BUFFER_BACK
                MOVLW   D'16'                   ; number of bytes in holding register
                MOVWF   COUNTER
WRITE_BYTE_TO_HREGS
                MOVF    POSTINC0, W             ; get low byte of buffer data
                MOVWF   TABLAT                  ; present data to table latch
                TBLWT+*                         ; write data, perform a short write
                                                ; to internal TBLWT holding register.
                DECFSZ  COUNTER                 ; loop until buffers are full
                BRA     WRITE_WORD_TO_HREGS
```


© 2007 Microchip Technology Inc.

例 6-3：　　写闪存程序存储器 （续）

```
PROGRAM_MEMORY
                BCF     EECON1, CFGS            ; access Flash program memory
                BSF     EECON1, WREN            ; enable write to memory
                BCF     INTCON, GIE             ; disable interrupts

                MOVLW   55h
                MOVWF   EECON2                  ; write 55h
    必需的序列  MOVLW   0AAh
                MOVWF   EECON2                  ; write 0AAh
                BSF     EECON1, WR              ; start program (CPU stall)
                DECFSZ  COUNTER1
                BRA     WRITE_BUFFER_BACK
                BSF     INTCON, GIE             ; re-enable interrupts
                BCF     EECON1, WREN            ; disable write to memory
```

6.5.2　写校验

根据具体应用情况，一个比较好的编程习惯可能要求将写入存储器的值与原始值作比较，进行校验。过多的写操作可能会导致某些位接近规范的极限，在这样的应用场合就应该采用写校验。

6.5.3　意外终止写操作

如果由于意外事件（如掉电或意外复位）终止了写操作，应该对刚刚编程的存储器单元进行验证，如有必要，还要重新进行编程。当写操作在正常操作过程中因 MCLR 复位或 WDT 超时复位而中断时，用户可以检查 WRERR 位，并根据需要重写该单元。

6.5.4　防止误写操作的保护措施

为防止对闪存程序存储器的误写操作，必须遵循写操作的启动顺序。更多详细信息，请参见**第 18.0 节 “CPU 的特殊功能”**。

6.6　代码保护期间闪存程序存储器的操作

关于闪存程序存储器代码保护的详细信息，请参见**第 18.5 节 “程序校验和代码保护”**。

表 6-2：　　与闪存程序存储器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
TBLPTRU	—	—	bit 21	程序存储器表指针最高字节 （TBLPTR<20:16>）					49
TBPLTRH	程序存储器表指针高字节 （TBLPTR<15:8>）								49
TBLPTRL	程序存储器表指针低字节 （TBLPTR<7:0>）								49
TABLAT	程序存储器表锁存器								49
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
EECON2	控制寄存器 2 （不是实际存在的寄存器）								51
EECON1	—	CFGS	—	FREE	WRERR	WREN	WR	—	51
IPR2	OSCFIP	—	USBIP	—	—	HLVDIP	—	—	51
PIR2	OSCFIF	—	USBIF	—	—	HLVDIF	—	—	51
PIE2	OSCFIE	—	USBIE	—	—	HLVDIE	—	—	51

图注：　– = 未用，读为 0。闪存访问期间不使用阴影单元。

注：

 © 2007 Microchip Technology Inc.

7.0 8 x 8 硬件乘法器

7.1 简介

所有 PIC18 器件均包含一个 8 x 8 硬件乘法器（是 ALU 的一部分）。该乘法器可执行无符号运算并产生一个 16 位运算结果，该结果存储在一对乘积寄存器 PRODH:PRODL 中。该乘法器执行的运算不会影响状态寄存器中的任何标志。

通过硬件执行乘法运算只需要一个指令周期。硬件乘法器具有更高的计算吞吐量并减少了乘法算法的代码长度，从而可在许多先前仅能使用数字信号处理器的应用中使用 PIC18 器件。表 7-1 给出了硬件和软件乘法运算的比较，包括所需存储器空间和执行时间。

7.2 工作原理

例 7-1 给出了一个 8 x 8 无符号乘法运算的指令序列。当已在 WREG 寄存器中装入了一个乘数时，实现该运算仅需一条指令。

例 7-2 给出了一个 8 x 8 有符号乘法运算的指令序列。要弄清乘数的符号位，必须检查每个乘数的最高有效位（MSb），并做相应的减法。

例 7-1： 8 x 8 无符号乘法程序

```
MOVF    ARG1, W     ;
MULWF   ARG2        ; ARG1 * ARG2 ->
                    ; PRODH:PRODL
```

例 7-2： 8 x 8 有符号乘法程序

```
MOVF    ARG1, W
MULWF   ARG2        ; ARG1 * ARG2 ->
                    ; PRODH:PRODL
BTFSC   ARG2, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                    ;         - ARG1
MOVF    ARG2, W
BTFSC   ARG1, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                    ;         - ARG2
```

表 7-1： 各种乘法运算的性能比较

程序	乘法实现方法	程序存储器（字）	周期数（最多）	时间		
				40 MHz 时	10 MHz 时	4 MHz 时
8 x 8 无符号	无硬件乘法	13	69	6.9 µs	27.6 µs	69 µs
	硬件乘法	1	1	100 ns	400 ns	1 µs
8 x 8 有符号	无硬件乘法	33	91	9.1 µs	36.4 µs	91 µs
	硬件乘法	6	6	600 ns	2.4 µs	6 µs
16 x 16 无符号	无硬件乘法	21	242	24.2 µs	96.8 µs	242 µs
	硬件乘法	28	28	2.8 µs	11.2 µs	28 µs
16 x 16 有符号	无硬件乘法	52	254	25.4 µs	102.6 µs	254 µs
	硬件乘法	35	40	4.0 µs	16.0 µs	40 µs

例 7-3 给出了一个 16 x 16 无符号乘法运算的指令序列。公式 7-1 为所使用的算法。32 位结果存储在 4 个寄存器（RES3:RES0）中。

公式 7-1： 16 x 16 无符号乘法算法

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L)
\end{aligned}
$$

例 7-3： 16 x 16 无符号乘法程序

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

例 7-4 给出了 16 x 16 有符号乘法运算的指令序列。公式 7-2 为所使用的算法。 32 位结果存储在 4 个寄存器（RES3:RES0）中。要弄清乘数的符号位，必须检查每个乘数的最高有效位 （MSb），并做相应的减法。

公式 7-2： 16 x 16 有符号乘法算法

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L) + \\
&\quad (-1 \bullet ARG2H\langle 7\rangle \bullet ARG1H:ARG1L \bullet 2^{16}) + \\
&\quad (-1 \bullet ARG1H\langle 7\rangle \bullet ARG2H:ARG2L \bullet 2^{16})
\end{aligned}
$$

例 7-4： 16 x 16 有符号乘法程序

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L,W
    MULWF   ARG2H           ; ARG1L * ARG2H ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    BTFSS   ARG2H, 7        ; ARG2H:ARG2L neg?
    BRA     SIGN_ARG1       ; no, check ARG1
    MOVF    ARG1L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG1H, W        ;
    SUBWFB  RES3
;
SIGN_ARG1
    BTFSS   ARG1H, 7        ; ARG1H:ARG1L neg?
    BRA     CONT_CODE       ; no, done
    MOVF    ARG2L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG2H, W        ;
    SUBWFB  RES3
;
CONT_CODE
   :
```


© 2007 Microchip Technology Inc.

8.0 中断

PIC18F2450/4450 器件具有多个中断源及一个中断优先级功能，该功能可以给每个中断源分配高优先级或者低优先级。高优先级中断向量位于 000008h，低优先级中断向量位于 000018h。高优先级中断事件可以中断正在处理的低优先级中断。

有 10 个寄存器用于控制中断操作。这些寄存器是：

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1 和 PIR2
- PIE1 和 PIE2
- IPR1 和 IPR2

建议使用 MPLAB® IDE 提供的 Microchip 头文件命名这些寄存器中的位。这使得汇编器 / 编译器能够自动识别指定寄存器内的这些位。

每个中断源有 3 个位用于控制其操作。这些位的功能分别是：

- 标志位表明发生了中断事件
- 允许位允许程序跳转到中断向量地址处执行 （当标志位置 1 时）
- 优先级位用于选择高优先级还是低优先级

通过将 IPEN 位（RCON<7>）置 1，可使能中断优先级功能。当使能中断优先级时，有 2 个全局中断允许位。将 GIEH 位（INTCON<7>）置 1，可允许所有优先级位已置 1（高优先级）的中断。将 GIEL 位 （INTCON<6>）置 1，可允许所有优先级位已清零（低优先级）的中断。当中断标志位、允许位及相应的全局中断允许位均被置 1 时，中断将根据设置的中断优先级立即跳转到地址 000008h 或 000018h。也可以通过设置相应的允许位来禁止单个中断。

当 IPEN 位清零 （默认状态）时，便会禁止中断优先级功能，此时中断是与 PIC® 中档系列器件兼容的。在兼容模式下，各个中断源的中断优先级位不起作用。INTCON<6> 是 PEIE 位，用于允许 / 禁止所有的外设中断源。 INTCON<7> 是 GIE 位，用于允许 / 禁止所有中断源。在兼容模式下，所有中断均跳转到 000008h。

当响应中断时，全局中断允许位被清零以禁止其他中断。清零后的 IPEN 位就是 GIE 位。如果使用了中断优先级，这个位就是 GIEH 位或者 GIEL 位。高优先级中断源会中断低优先级中断。在处理高优先级中断时，低优先级中断将不被响应。

返回地址被压入堆栈，中断向量地址 （000008h 或 000018h）被装入 PC。只要在中断服务程序中，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应中断。

执行“从中断返回”指令 RETFIE 将退出中断程序，同时将 GIE 位（若使用中断优先级则为 GIEH 或 GIEL 位）置 1，从而重新允许中断。

对于外部中断事件，例如 INT 引脚中断或者 PORTB 输入电平变化中断，中断响应延时将会是 3 到 4 个指令周期。对于单周期或双周期指令，中断响应延时完全相同。各中断标志位的置 1 不受对应的中断允许位和 GIE 位状态的影响。

注：	当允许**任何**中断时，不要使用 MOVFF 指令修改中断控制寄存器。否则可能导致单片机操作出错。

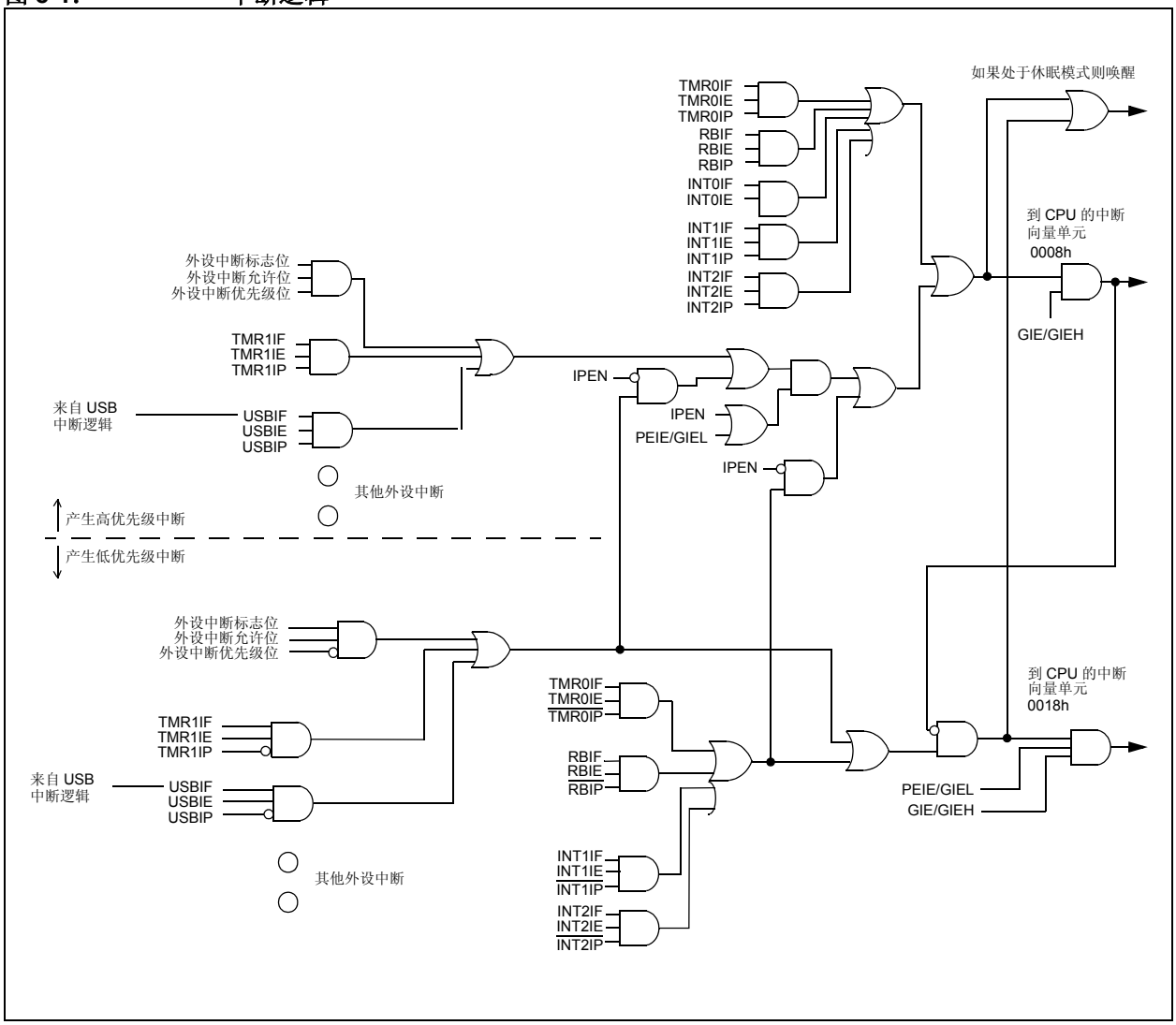
8.1 USB 中断

USB 模块不同于其他外设，它能够为多种类型的事件产生范围较宽的中断。这些包括几种类型的正常通信和状态事件，以及几种模块级出错事件。

要处理这些事件，USB 模块配备了自己的中断逻辑。逻辑功能类似于单片机级中断传送的方式，每个中断源均有独立的标志位和允许位。所有事件均被传送到单个器件级中断 USBIF （PIR2<5>）。与器件级中断逻辑不同的是，独立的 USB 中断事件无法单独分配其自己的优先级。对于所有 USB 事件统一由 USBIP 位决定。

有关 USB 中断逻辑的其他详细信息，请参见**第 14.5 节“USB 中断”**。

图 8-1：　　中断逻辑



© 2007 Microchip Technology Inc.

8.2 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，包含多个允许位、优先级位和标志位。

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。中断标志位可由软件查询。

寄存器 8-1： INTCON：中断控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **GIE/GIEH**：全局中断允许位
当 IPEN = 0 时：
1 = 允许所有未被屏蔽的中断
0 = 禁止所有中断
当 IPEN = 1 时：
1 = 允许所有高优先级中断
0 = 禁止所有中断

bit 6 **PEIE/GIEL**：外设中断允许位
当 IPEN = 0 时：
1 = 允许所有未被屏蔽的外设中断
0 = 禁止所有外设中断
当 IPEN = 1 时：
1 = 允许所有低优先级的外设中断
0 = 禁止所有低优先级的外设中断

bit 5 **TMR0IE**：TMR0 溢出中断允许位
1 = 允许 TMR0 溢出中断
0 = 禁止 TMR0 溢出中断

bit 4 **INT0IE**：INT0 外部中断允许位
1 = 允许 INT0 外部中断
0 = 禁止 INT0 外部中断

bit 3 **RBIE**：RB 端口电平变化中断允许位
1 = 允许 RB 端口电平变化中断
0 = 禁止 RB 端口电平变化中断

bit 2 **TMR0IF**：TMR0 溢出中断标志位
1 = TMR0 寄存器已溢出 （必须用软件清零）
0 = TMR0 寄存器未溢出

bit 1 **INT0IF**：INT0 外部中断标志位
1 = 发生了 INT0 外部中断 （必须用软件清零）
0 = 未发生 INT0 外部中断

bit 0 **RBIF**：RB 端口电平变化中断标志位 (1)
1 = RB7:RB4 引脚中至少有一个引脚的电平状态发生了改变 （必须用软件清零）
0 = RB7:RB4 引脚电平状态没有改变

注 1： 电平的不匹配会不断地将 RBIF 位置 1。读取 PORTB 可以结束不匹配情况，并将 RBIF 位清零。

寄存器 8-2：　　INTCON2：中断控制寄存器 2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　**$\overline{\text{RBPU}}$**：PORTB 上拉使能位
1 = 禁止所有 PORTB 上拉
0 = 根据各端口锁存值使能 PORTB 上拉

bit 6　**INTEDG0**：外部中断 0 边沿选择位
1 = 上升沿触发中断
0 = 下降沿触发中断

bit 5　**INTEDG1**：外部中断 1 边沿选择位
1 = 上升沿触发中断
0 = 下降沿触发中断

bit 4　**INTEDG2**：外部中断 2 边沿选择位
1 = 上升沿触发中断
0 = 下降沿触发中断

bit 3　**未用**：读为 0

bit 2　**TMR0IP**：TMR0 溢出中断优先级位
1 = 高优先级
0 = 低优先级

bit 1　**未用**：读为 0

bit 0　**RBIP**：RB 端口电平变化中断优先级位
1 = 高优先级
0 = 低优先级

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。中断标志位可由软件查询。

© 2007 Microchip Technology Inc.

寄存器 8-3：　　INTCON3：中断控制寄存器 3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　　**INT2IP**：INT2 外部中断优先级位
1 = 高优先级
0 = 低优先级

bit 6　　**INT1IP**：INT1 外部中断优先级位
1 = 高优先级
0 = 低优先级

bit 5　　**未用**：读为 0

bit 4　　**INT2IE**：INT2 外部中断允许位
1 = 允许 INT2 外部中断
0 = 禁止 INT2 外部中断

bit 3　　**INT1IE**：INT1 外部中断允许位
1 = 允许 INT1 外部中断
0 = 禁止 INT1 外部中断

bit 2　　**未用**：读为 0

bit 1　　**INT2IF**：INT2 外部中断标志位
1 = 发生了 INT2 外部中断 （必须用软件清零）
0 = 未发生 INT2 外部中断

bit 0　　**INT1IF**：INT1 外部中断标志位
1 = 发生了 INT1 外部中断 （必须用软件清零）
0 = 未发生 INT1 外部中断

注：　当中断条件产生时，不管相应的中断允许位或全局中断允许位的状态如何，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。中断标志位可由软件查询。

8.3 PIR 寄存器

PIR 寄存器包含各外设中断的标志位。根据外设中断源的数量，有两个外设中断请求 （标志）寄存器 （PIR1 和 PIR2）。

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（INTCON<7>）的状态如何，中断标志位都将置 1。

2: 用户软件应在允许一个中断前和处理完一次中断后，将相应的中断标志位清零。

寄存器 8-4： PIR1：外设中断请求 （标志）寄存器 1

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

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用：** 读为 0

bit 6 **ADIF：** A/D 转换器中断标志位
1 = 一次 A/D 转换已完成 （必须用软件清零）
0 = A/D 转换未完成

bit 5 **RCIF：** EUSART 接收中断标志位
1 = EUSART 接收缓冲器 RCREG 已满 （读取 RCREG 时清零）
0 = EUSART 接收缓冲器为空

bit 4 **TXIF：** EUSART 发送中断标志位
1 = EUSART 发送缓冲器 TXREG 已空 （写入 TXREG 时清零）
0 = EUSART 发送缓冲器满

bit 3 **未用：** 读为 0

bit 2 **CCP1IF：** CCP1 中断标志位
捕捉模式：
1 = 发生了 TMR1 寄存器捕捉 （必须用软件清零）
0 = 未发生 TMR1 寄存器捕捉
比较模式：
1 = 发生了 TMR1 寄存器的比较匹配 （必须用软件清零）
0 = 未发生 TMR1 寄存器的比较匹配
PWM 模式：
在此模式下未使用。

bit 1 **TMR2IF：** TMR2 与 PR2 匹配中断标志位
1 = TMR2 与 PR2 发生匹配 （必须用软件清零）
0 = TMR2 与 PR2 未发生匹配

bit 0 **TMR1IF：** TMR1 溢出中断标志位
1 = TMR1 寄存器已溢出 （必须用软件清零）
0 = TMR1 寄存器未溢出

© 2007 Microchip Technology Inc.

寄存器 8-5：　　PIR2：外设中断请求 （标志）寄存器 2

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
OSCFIF	—	USBIF	—	—	HLVDIF	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　　**OSCFIF**：振荡器失效中断标志位
1 = 系统振荡器失效，改成由 INTRC 作为时钟输入 （必须用软件清零）
0 = 系统时钟正常运行

bit 6　　**未用**：读为 0

bit 5　　**USBIF**：USB 中断标志位
1 = USB 已请求中断 （必须用软件清零）
0 = 无 USB 中断请求

bit 4-3　　**未用**：读为 0

bit 2　　**HLVDIF**：高 / 低压检测中断标志位
1 = 发生了高 / 低压条件
0 = 未发生高 / 低压条件

bit 1-0　　**未用**：读为 0

8.4 PIE 寄存器

PIE 寄存器包含各外设中断的允许位。根据外设中断源的数量，有两个外设中断允许寄存器（PIE1 和 PIE2）。当 IPEN = 0 时，要允许任一外设中断，必须将 PEIE 位置 1。

寄存器 8-6： PIE1：外设中断允许寄存器 1

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用**：读为 0

bit 6 **ADIE**：A/D 转换器中断允许位
1 = 允许 A/D 中断
0 = 禁止 A/D 中断

bit 5 **RCIE**：EUSART 接收中断允许位
1 = 允许 EUSART 接收中断
0 = 禁止 EUSART 接收中断

bit 4 **TXIE**：EUSART 发送中断允许位
1 = 允许 EUSART 发送中断
0 = 禁止 EUSART 发送中断

bit 3 **未用**：读为 0

bit 2 **CCP1IE**：CCP1 中断允许位
1 = 允许 CCP1 中断
0 = 禁止 CCP1 中断

bit 1 **TMR2IE**：TMR2 与 PR2 匹配中断允许位
1 = 允许 TMR2 与 PR2 匹配中断
0 = 禁止 TMR2 与 PR2 匹配中断

bit 0 **TMR1IE**：TMR1 溢出中断允许位
1 = 允许 TMR1 溢出中断
0 = 禁止 TMR1 溢出中断

 © 2007 Microchip Technology Inc.

寄存器 8-7：　　PIE2：外设中断允许寄存器 2

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
OSCFIE	—	USBIE	—	—	HLVDIE	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　**OSCFIE**：振荡器失效中断允许位
1 = 允许
0 = 禁止

bit 6　**未用**：读为 0

bit 5　**USBIE**：USB 中断允许位
1 = 允许
0 = 禁止

bit 4-3　**未用**：读为 0

bit 2　**HLVDIE**：高 / 低压检测中断允许位
1 = 允许
0 = 禁止

bit 1-0　**未用**：读为 0

8.5 IPR 寄存器

IPR 寄存器包含各外设中断的优先级位。根据外设中断源的数量，有两个外设中断优先级寄存器（IPR1 和 IPR2）。使用优先级位时，要求将中断优先级允许（IPEN）位置 1。

寄存器 8-8： IPR1：外设中断优先级寄存器 1

U-0	R/W-1	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用**：读为 0

bit 6 **ADIP**：A/D 转换器中断优先级位
1 = 高优先级
0 = 低优先级

bit 5 **RCIP**：EUSART 接收中断优先级位
1 = 高优先级
0 = 低优先级

bit 4 **TXIP**：EUSART 发送中断优先级位
1 = 高优先级
0 = 低优先级

bit 3 **未用**：读为 0

bit 2 **CCP1IP**：CCP1 中断优先级位
1 = 高优先级
0 = 低优先级

bit 1 **TMR2IP**：TMR2 与 PR2 匹配中断优先级位
1 = 高优先级
0 = 低优先级

bit 0 **TMR1IP**：TMR1 溢出中断优先级位
1 = 高优先级
0 = 低优先级

© 2007 Microchip Technology Inc.

寄存器 8-9：　　IPR2：外设中断优先级寄存器 2

R/W-1	U-0	R/W-1	U-0	U-0	R/W-1	U-0	U-0
OSCFIP	—	USBIP	—	—	HLVDIP	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　　**OSCFIP**：振荡器失效中断优先级位
1 = 高优先级
0 = 低优先级

bit 6　　**未用**：读为 0

bit 5　　**USBIP**：USB 中断优先级位
1 = 高优先级
0 = 低优先级

bit 4-3　　**未用**：读为 0

bit 2　　**HLVDIP**：高 / 低压检测中断优先级位
1 = 高优先级
0 = 低优先级

bit 1-0　　**未用**：读为 0

8.6 RCON 寄存器

RCON寄存器中包含的位可用来确定器件上次复位或从空闲或休眠模式唤醒的原因。 RCON 还包含一个可允许中断优先级的 IPEN 位。

寄存器 8-10： RCON：复位控制寄存器

R/W-0	R/W-1(1)	U-0	R/W-1	R-1	R-1	R/W-0(2)	R/W-0
IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **IPEN**：中断优先级允许位
1 = 允许中断优先级
0 = 禁止中断优先级 （PIC16CXXX 兼容模式）

bit 6 **SBOREN**：BOR 软件使能位 (1)
位操作的详细信息，请参见寄存器 4-1。

bit 5 **未用**：读为 0

bit 4 **$\overline{RI}$**：RESET 指令标志位
位操作的详细信息，请参见寄存器 4-1。

bit 3 **$\overline{TO}$**：看门狗超时标志位
位操作的详细信息，请参见寄存器 4-1。

bit 2 **$\overline{PD}$**：掉电检测标志位
位操作的详细信息，请参见寄存器 4-1。

bit 1 **$\overline{POR}$**：上电复位状态位 (2)
位操作的详细信息，请参见寄存器 4-1。

bit 0 **$\overline{BOR}$**：欠压复位状态位
位操作的详细信息，请参见寄存器 4-1。

注 1： 如果使能了 SBOREN 位，其复位状态为 1；否则为 0。更多信息，请参见寄存器 4-1。
2： $\overline{POR}$ 的实际复位值由器件复位的类型决定。更多信息，请参见寄存器 4-1。

© 2007 Microchip Technology Inc.

8.7 INTn 引脚中断

RB0/AN12/INT0、RB1/AN10/INT1 和 RB2/AN8/INT2/VMO 引脚上的外部中断都是边沿触发的。如果 INTCON2 寄存器中相应的 INTEDGx 位被置 1（= 1），则为上升沿触发；如果该位被清零，则为下降沿触发。当 RBx/INTx 引脚上出现一个有效边沿时，相应的标志位 INTxIF 被置 1。通过清零相应的允许位 INTxIE，可禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将中断标志位 INTxIF 清零。

如果 INTxIE 位在进入功耗管理模式前被置 1，则所有的外部中断（INT0、INT1 和 INT2）均能将处理器从功耗管理模式唤醒。如果全局中断允许位 GIE 被置 1，则处理器将在被唤醒之后转移到中断向量处执行程序。

INT1 和 INT2 的中断优先级由中断优先级位 INT1IP（INTCON3<6>）和 INT2IP（INTCON3<7>）中的值决定。没有与 INT0 相关的优先级位。INT0 始终是一个高优先级的中断源。

8.8 TMR0 中断

在 8 位模式（默认模式）下，TMR0 寄存器的溢出（FFh → 00h）会使 TMR0IF 标志位置 1。在 16 位模式下，TMR0H:TMR0L 寄存器对的溢出（FFFFh → 0000h）会使 TMR0IF 标志位置 1。通过将允许位 TMR0IE（INTCON<5>）置 1 或清零，可以允许或禁止该中断。Timer0 的中断优先级由中断优先级位 TMR0IP（INTCON2<2>）中的值决定。欲进一步了解 Timer0 模块的详细信息，请参见**第 10.0 节“Timer0 模块”**。

8.9 PORTB 电平变化中断

PORTB<7:4> 上的输入电平变化会将标志位 RBIF（INTCON<0>）置 1。通过将允许位 RBIE（INTCON<3>）置 1 或清零，可以允许或禁止该中断。PORTB 电平变化中断的优先级由中断优先级位 RBIP（INTCON2<0>）中的值决定。

8.10 中断的现场保护

在中断期间，PC 的返回地址被保存在堆栈中。另外，WREG、STATUS 和 BSR 寄存器的值被压入快速返回堆栈。如果未使用从中断快速返回功能（见**第 5.3 节“数据存储器构成”**），那么用户可能需要在进入中断服务程序前，保存 WREG、STATUS 和 BSR 寄存器的值。根据用户的具体应用，还可能需要保存其他寄存器的值。例 8-1 在执行中断服务程序期间，保存并恢复 WREG、STATUS 和 BSR 寄存器的值。

例 8-1： 将 STATUS、WREG 和 BSR 寄存器的值保存在 RAM 中

```
MOVWF   W_TEMP                  ; W_TEMP is in virtual bank
MOVFF   STATUS, STATUS_TEMP     ; STATUS_TEMP located anywhere
MOVFF   BSR, BSR_TEMP           ; BSR_TMEP located anywhere
;
; USER ISR CODE
;
MOVFF   BSR_TEMP, BSR           ; Restore BSR
MOVF    W_TEMP, W               ; Restore WREG
MOVFF   STATUS_TEMP, STATUS     ; Restore STATUS
```

注：

 © 2007 Microchip Technology Inc.

9.0 I/O 端口

根据选定的器件和使能的功能，最多有 5 个端口可供使用。I/O 端口的一些引脚与器件上外设功能复用。一般来说，当外设被使能时，其对应的引脚就不能被用作通用 I/O 引脚。

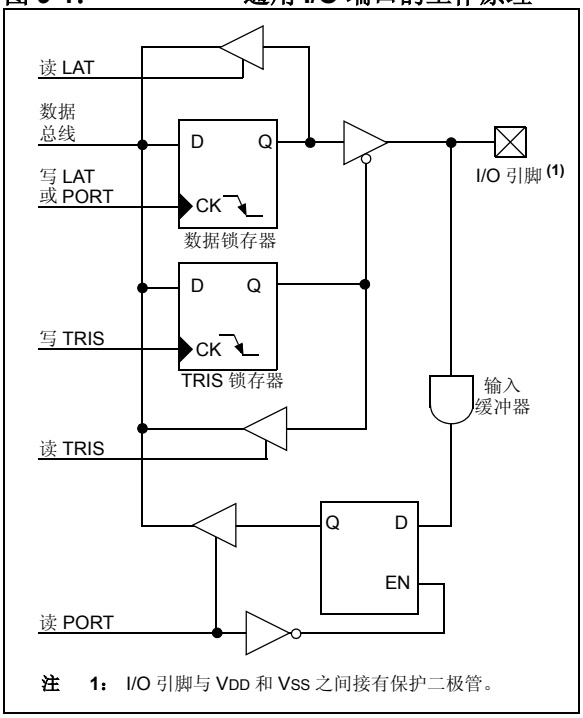
每个端口都有三个工作寄存器。这些寄存器是：

- TRIS 寄存器 （数据方向寄存器）
- PORT 寄存器 （读取器件引脚的电平）
- LAT 寄存器 （输出锁存器）

在对 I/O 引脚驱动值进行读－修改－写时会用到数据锁存寄存器 （LATA）。

图 9-1 给出了通用 I/O 端口的简化模型，没有给出到其他外设的接口。

图 9-1： 通用 I/O 端口的工作原理



注 1： I/O 引脚与 VDD 和 VSS 之间接有保护二极管。

9.1 PORTA、TRISA 和 LATA 寄存器

PORTA 是一个 8 位宽的双向端口，对应的数据方向寄存器是 TRISA。将 TRISA 某位置 1 （= 1）时，会将 PORTA 的相应引脚设为输入 （即，使相应的输出驱动器呈高阻状态）。将 TRISA 某位清零 （= 0）时，会将 PORTA 的相应引脚设为输出 （即，将输出锁存器中的内容置于选中引脚）。

读 PORTA 寄存器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。

数据锁存器（LATA）也是存储器映射的。对 LATA 寄存器执行读－修改－写操作将读写 PORTA 的输出锁存值。

RA4 引脚与 Timer0 模块的时钟输入复用，成为 RA4/T0CKI/C1OUT 引脚。 RA6 引脚与主振荡器引脚复用，通过在配置寄存器 1H （详细信息，请参见**第 18.1 节 “配置位”** ）中对主振荡器进行配置可将这两个引脚使能为振荡器或 I/O 引脚。当没被用作端口引脚时， RA6 及其相关的 TRIS 和 LAT 位均读为 0。

RA4 也和 USB 模块复用；它用作外部 USB 收发器的接收器输入。关于 USB 模块配置的详细信息，请参见**第 14.2 节 “USB 状态和控制”**。

几个 PORTA 端口和模拟输入复用。通过将 ADCON1 寄存器 （A/D 控制寄存器 1）中的控制位清零或置 1，可将 RA3:RA0 和 RA5 引脚选作 A/D 转换器输入引脚。

> **注：** 在上电复位时，RA5 和 RA3:RA0 被配置为模拟输入并读为 0。 RA4 则被配置为数字输入。

所有其他 PORTA 引脚都是 TTL 电平输入和全 CMOS 驱动输出的。

TRISA 寄存器控制着 RA 引脚的方向，即使它们被用作模拟输入。当引脚用于模拟输入时，用户必须确保 TRISA 寄存器中相应的位保持置 1。

例 9-1： 初始化 PORTA

```
CLRF    PORTA   ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF    LATA    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0Fh     ; Configure A/D
MOVWF   ADCON1  ; for digital inputs
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISA   ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
```

表 9-1： PORTA I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RA0/AN0	RA0	0	输出	DIG	LATA<0> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTA<0> 数据输入；当使能模拟输入时被禁止。
	AN0	1	输入	ANA	A/D 输入通道 0。 POR 时的默认配置；不影响数字输出。
RA1/AN1	RA1	0	输出	DIG	LATA<1> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTA<1> 数据输入； POR 时读为 0。
	AN1	1	输入	ANA	A/D 输入通道 1。 POR 时的默认配置；不影响数字输出。
RA2/AN2/ VREF-	RA2	0	输出	DIG	LATA<2> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTA<2> 数据输入。当使能模拟功能时被禁止。
	AN2	1	输入	ANA	A/D 输入通道 2。 POR 时的默认配置；不受模拟输出影响。
	VREF-	1	输入	ANA	A/D 参考电压低电平输入。
RA3/AN3/ VREF+	RA3	0	输出	DIG	LATA<3> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTA<3> 数据输入；当使能模拟输入时被禁止。
	AN3	1	输入	ANA	A/D 输入通道 3。 POR 时的默认配置。
	VREF+	1	输入	ANA	A/D 参考电压高电平输入。
RA4/T0CKI/ RCV	RA4	0	输出	DIG	LATA<4> 数据输出；不受模拟输入影响。
		1	输入	ST	PORTA<4> 数据输入；当使能模拟输入时被禁止。
	T0CKI	1	输入	ST	Timer0 的时钟输入。
	RCV	x	输入	TTL	外部 USB 收发器 RCV 输入。
RA5/AN4/ HLVDIN	RA5	0	输出	DIG	LATA<5> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTA<5> 数据输入；当使能模拟输入时被禁止。
	AN4	1	输入	ANA	A/D 输入通道 4。 POR 时的默认配置。
	HLVDIN	1	输入	ANA	高 / 低压检测外部跳变点输入。
OSC2/CLKO/ RA6	OSC2	x	输出	ANA	主振荡器反馈输出连接 （XT 和 HS 模式）。
	CLKO	x	输出	DIG	EC、 ECPLL 和 INTCKO 模式下的系统周期时钟输出 （FOSC/4）。
	RA6	0	输出	DIG	LATA<6> 数据输出。仅在 ECIO、 ECPIO 和 INTIO 模式下可用；否则读为 0。
		1	输入	TTL	PORTA<6> 数据输入。仅在 ECIO、 ECPIO 和 INTIO 模式下可用；否则读为 0。

图注： OUT = 输出， IN = 输入， ANA = 模拟信号， DIG = 数字输出， ST = 施密特缓冲器输入，
TTL = TTL 缓冲器输入， x = 无关位 （TRIS 位不影响端口方向或在此可忽略）

表 9-2： 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	复位值所在页
PORTA	—	RA6(1)	RA5	RA4	RA3	RA2	RA1	RA0	51
LATA	—	LATA6(1)	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	51
TRISA	—	TRISA6(1)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	51
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	52

图注： – = 未用，读为 0。 PORTA 不使用阴影单元。

注 1： RA6 及其相关的锁存器和数据方向位根据振荡器配置使能为 I/O 引脚；否则，它们将被读为 0。

© 2007 Microchip Technology Inc.

9.2 PORTB、 TRISB 和 LATB 寄存器

PORTB 是一个 8 位宽的双向端口，对应的数据方向寄存器是 TRISB。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状态）。将 TRISB 某位清零 （= 0）时，会将 PORTB 的相应引脚设为输出 （即，将输出锁存器中的内容置于选中引脚）。

数据锁存器 （LATB）也是存储器映射的。对 LATB 寄存器执行读－修改－写操作将读写 PORTB 的输出锁存值。

每个 PORTB 引脚都具有内部弱上拉电路。一个控制位即可接通所有上拉电路。这是通过清零 RBPU 位（INTCON2<7>）实现的。当端口引脚被配置为输出时，其弱上拉电路会自动切断。上电复位会禁止弱上拉电路。

注：	上电复位时，默认情况下 RB4:RB0 被配置为模拟输入且读为 0； RB7:RB5 则被配置为数字输入。 通过对配置位 PBADEN（CONFIG3H<1>）进行编程， RB4:RB0 可在 POR 时被配置为数字输入。

PORTB 的四个引脚 （RB7:RB4）具有电平变化中断功能。仅当将这些引脚配置为输入时，才可使用此中断功能。当 RB7:RB4 中的任何一个引脚被配置为输出时，该引脚将不再具有电平变化中断功能。将引脚上的输入电平与 PORTB 上次读入锁存器的旧值进行比较。对 RB7:RB4 上的“不匹配”输出进行或运算，产生 RB 端口电平变化中断，并将标志位 RBIF （INTCON<0>）置 1。

可使用电平变化中断将器件从休眠中唤醒。用户可用以下方式在中断服务程序中清除该中断：

a） 读或写 PORTB （MOVFF (ANY), PORTB 指令除外）。这将结束不匹配条件。

b） 将标志位 RBIF 清零。

不匹配条件将继续把标志位 RBIF 置 1。读 PORTB 将结束不匹配条件并允许将标志位 RBIF 清零。

建议使用电平变化中断功能实现按键唤醒操作，以及那些仅用到 PORTB 的电平变化中断功能的操作。在使用电平变化中断功能时，建议不要查询 PORTB 的状态。

引脚 RB2 和 RB3 与 USB 外设复用，用作外部 USB 收发器（TRIS 配置）的差分信号输出。关于配置 USB 模块与外部收发器协同工作的更多信息，请参见**第 14.2.2.2 节 “外部收发器”**。

例 9-2： 初始化 PORTB

```
CLRF    PORTB   ; Initialize PORTB by
                ; clearing output
                ; data latches
CLRF    LATB    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0Eh     ; Set RB<4:0> as
MOVWF   ADCON1  ; digital I/O pins
                ; (required if config bit
                ; PBADEN is set)
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISB   ; Set RB<3:0> as inputs
                ; RB<5:4> as outputs
                ; RB<7:6> as inputs
```

表 9-3： PORTB I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RB0/AN12/ INT0	RB0	0	输出	DIG	LATB<0> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTB<0> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	AN12	1	输入	ANA	A/D 输入通道 12。(1)
	INT0	1	输入	ST	外部中断 0 输入。
RB1/AN10/ INT1	RB1	0	输出	DIG	LATB<1> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTB<1> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	AN10	1	输入	ANA	A/D 输入通道 10。(1)
	INT1	1	输入	ST	外部中断 1 输入。
RB2/AN8/ INT2/VMO	RB2	0	输出	DIG	LATB<2> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTB<2> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	AN8	1	输入	ANA	A/D 输入通道 8。(1)
	INT2	1	输入	ST	外部中断 2 输入。
	VMO	0	输出	DIG	外部 USB 收发器 VMO 数据输出。
RB3/AN9/VPO	RB3	0	输出	DIG	LATB<3> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTB<3> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	AN9	1	输入	ANA	A/D 输入通道 9。(1)
	VPO	0	输出	DIG	外部 USB 收发器 VPO 数据输出。
RB4/AN11/ KBI0	RB4	0	输出	DIG	LATB<4> 数据输出；不受模拟输入影响。
		1	输入	TTL	PORTB<4> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。当使能模拟输入时被禁止。(1)
	AN11	1	输入	ANA	A/D 输入通道 11。(1)
	KBI0	1	输入	TTL	引脚电平变化中断。
RB5/KBI1/ PGM	RB5	0	输出	DIG	LATB<5> 数据输出。
		1	输入	TTL	PORTB<5> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。
	KBI1	1	输入	TTL	引脚电平变化中断。
	PGM	x	输入	ST	单电源供电编程模式选择 （ICSP™）。由 LVP 配置位使能；所有其他引脚功能被禁止。
RB6/KBI2/ PGC	RB6	0	输出	DIG	LATB<6> 数据输出。
		1	输入	TTL	PORTB<6> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。
	KBI2	1	输入	TTL	引脚电平变化中断。
	PGC	x	输入	ST	供 ICSP 和 ICD 工作使用的串行执行 （ICSP）时钟输入。(2)
RB7/KBI3/ PGD	RB7	0	输出	DIG	LATB<7> 数据输出。
		1	输入	TTL	PORTB<7> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时启用弱上拉。
	KBI3	1	输入	TTL	引脚电平变化中断。
	PGD	x	输出	DIG	供 ICSP 和 ICD 工作使用的串行执行数据输出。(2)
		x	输入	ST	供 ICSP 和 ICD 工作使用的串行执行数据输入。(2)

图注： OUT = 输出， IN = 输入， ANA = 模拟信号， DIG = 数字输出， ST = 施密特缓冲器输入，
TTL = TTL 缓冲器输入， x = 无关位 （TRIS 位不影响端口方向或在此可忽略）

注 1： POR 时的配置由 PBADEN 配置位决定。当 PBADEN 置 1 时，引脚被配置为模拟输入；当 PBADEN 清零时，则被配置为数字输入。

2： 当使能 ICSP™ 或 ICD 操作时，禁止所有其他引脚的功能。

© 2007 Microchip Technology Inc.

表 9-4：　　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	复位值所在页
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	51
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	51
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	51
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	49
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	49
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	52

图注：　　– = 未用，读为 0。 PORTB 不使用阴影单元。

9.3 PORTC、 TRISC 和 LATC 寄存器

PORTC 是一个 7 位宽的双向端口，对应的数据方向寄存器是 TRISC。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状态）。将 TRISC 某位清零 （= 0）时，会将 PORTC 的相应引脚设为输出 （即，将输出锁存器中的内容置于选中引脚）。

在 PIC18F2450/4450 器件中， RC3 引脚未实现。

数据锁存器 （LATC）也是存储器映射的。对 LATC 寄存器执行读－修改－写操作将读写 PORTC 的输出锁存值。

PORTC 主要和串行通信模块复用，包括 EUSART 和 USB 模块（表 9-5）。除了 RC4 和 RC5，PORTC 使用施密特触发输入缓冲器。

引脚 RC4 和 RC5 与 USB 模块复用。根据模块的配置，它们可用作片上 USB 收发器的差分数据线或来自外部 USB 收发器的数据输入线。 RC4 和 RC5 引脚上都有 TTL 输入缓冲器，而不是施密特触发缓冲器。

和其他 PORTC 引脚不同， RC4 和 RC5 没有和它们关联的 TRISC 位。作为数字端口，它们只能用作数字输入。为 USB 操作配置时，数据方向是由 USB 模块的配置和状态决定的。如果使用了外部收发器，RC4 和 RC5 总是用作来自收发器的输入。如果使用了片上收发器，数据方向由那时模块执行的操作决定。

外部收发器使能时， RC2 也用作收发器的输出使能控制。关于配置 USB 选项的更多信息，请参见**第 14.2.2.2 节 “外部收发器”**。

使能除 RC4 和 RC5 以外的 PORTC 引脚上的外设功能时，应小心定义 TRIS 位。有些外设会无视 TRIS 位的设置，将引脚定义为输出引脚或输入引脚。用户应该查阅相应的外设章节来正确设置 TRIS 位。

注：	上电复位时，这些引脚（除了RC4和RC5）都被配置为数字输入。要将引脚 RC4 和 RC5 用作数字输入，必须禁止 USB 模块（UCON<3> = 0）并且禁止片上 USB收发器 （UCFG<3> = 1）。

外设对引脚的改写会影响TRISC寄存器的内容。尽管如此，读 TRISC 总是会返回其当前的内容。

例 9-3： 初始化 PORTC

```
CLRF    PORTC   ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   07h     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; RC<5:0> as outputs
                ; RC<7:6> as inputs
```

 © 2007 Microchip Technology Inc.

表 9-5： PORTC I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RC0/T1OSO/ T1CKI	RC0	0	输出	DIG	LATC<0> 数据输出。
		1	输入	ST	PORTC<0> 数据输入。
	T1OSO	x	输出	ANA	Timer1 振荡器输出；当使能 Timer1 振荡器时被使能。 禁止数字 I/O。
	T1CKI	1	输入	ST	Timer1 计数器输入。
RC1/T1OSI/ $\overline{UOE}$	RC1	0	输出	DIG	LATC<1> 数据输出。
		1	输入	ST	PORTC<1> 数据输入。
	T1OSI	x	输入	ANA	Timer1 振荡器输入；当使能 Timer1 振荡器时被使能。 禁止数字 I/O。
	$\overline{UOE}$	0	输出	DIG	外部 USB 收发器 $\overline{OE}$ 输出。
RC2/CCP1	RC2	0	输出	DIG	LATC<2> 数据输出。
		1	输入	ST	PORTC<2> 数据输入。
	CCP1	0	输出	DIG	CCP1 比较输出和 PWM 输出；优先于端口数据。
		1	输入	ST	CCP1 捕捉输入。
RC4/D-/VM	RC4	—(1)	输入	TTL	PORTC<4> 数据输入；当使能 USB 模块或片上收发器时被禁止。
	D-	—(1)	输出	XCVR	USB 总线差分负信号线输出 （内部收发器）。
		—(1)	输入	XCVR	USB 总线差分负信号线输入 （内部收发器）。
	VM	—(1)	输入	TTL	外部 USB 收发器 VM 输入。
RC5/D+/VP	RC5	—(1)	输入	TTL	PORTC<5> 数据输入；当使能 USB 模块或片上收发器时被禁止。
	D+	—(1)	输出	XCVR	USB 总线差分正信号线输出 （内部收发器）。
		—(1)	输入	XCVR	USB 总线差分正信号线输入 （内部收发器）。
	VP	—(1)	输入	TTL	外部 USB 收发器 VP 输入。
RC6/TX/CK	RC6	0	输出	DIG	LATC<6> 数据输出。
		1	输入	ST	PORTC<6> 数据输入。
	TX	0	输出	DIG	异步串行发送数据输出 （EUSART 模块）；优先于端口数据。用户必须将其配置为输出。
	CK	0	输出	DIG	同步串行时钟输出 （EUSART 模块）；优先于端口数据。
		1	输入	ST	同步串行时钟输入 （EUSART 模块）。
RC7/RX/DT	RC7	0	输出	DIG	LATC<7> 数据输出。
		1	输入	ST	PORTC<7> 数据输入。
	RX	1	输入	ST	异步串行接收数据输入 （EUSART 模块）。
	DT	1	输出	DIG	同步串行数据输出 （EUSART 模块）。
		1	输入	ST	同步串行数据输入 （EUSART 模块）。用户必须将其配置为输入。

图注： OUT = 输出， IN = 输入， ANA = 模拟信号， DIG = 数字输出， ST = 施密特缓冲器输入，
TTL = TTL 缓冲器输入， XCVR = USB 收发器， x = 无关位 （TRIS 位不影响端口方向或在此可忽略）

注 1： RC4 和 RC5 没有对应的 TRISC 位。在端口模式下，这些引脚只作为输入。 USB 数据方向是由 USB 配置决定的。

表 9-6：　　与 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
PORTC	RC7	RC6	RC5(1)	RC4(1)	—	RC2	RC1	RC0	51
LATC	LATC7	LATC6	—	—	—	LATC2	LATC1	LATC0	51
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	51
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	52

图注：　　– = 未用，读为 0。 PORTC 不使用阴影单元。

注　1： RC5 和 RC4 在 USB 模块被禁止时 （UCON<3> = 0），只能用作端口引脚。

© 2007 Microchip Technology Inc.

9.4 PORTD、 TRISD 和 LATD 寄存器

注： PORTD 仅在 40/44-pin 的器件上存在。

PORTD 是一个 8 位宽的双向端口，对应的数据方向寄存器是 TRISD。将 TRISD 某位置 1 （= 1）时，会将 PORTD 的相应引脚设为输入 （即，使相应的输出驱动器呈高阻状态）。将 TRISD 某位清零 （= 0）时，会将 PORTD 的相应引脚设为输出 （即，将输出锁存器中的内容置于选中引脚）。

数据锁存器 （LATD）也是存储器映射的。对 LATD 寄存器执行读－修改－写操作将读写 PORTD 的输出锁存值。

PORTD 上的所有引脚都配有施密特触发输入缓冲器。每个引脚都可被单独配置为输入或输出。

注： 上电复位时，这些引脚被配置为数字输入。

例 9-4： 初始化 PORTD

```
CLRF    PORTD   ; Initialize PORTD by
                ; clearing output
                ; data latches
CLRF    LATD    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISD   ; Set RD<3:0> as inputs
                ; RD<5:4> as outputs
                ; RD<7:6> as inputs
```

表 9-7： PORTD I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RD0	RD0	0	输出	DIG	LATD<0> 数据输出。
		1	输入	ST	PORTD<0> 数据输入。
RD1	RD1	0	输出	DIG	LATD<1> 数据输出。
		1	输入	ST	PORTD<1> 数据输入。
RD2	RD2	0	输出	DIG	LATD<2> 数据输出。
		1	输入	ST	PORTD<2> 数据输入。
RD3	RD3	0	输出	DIG	LATD<3> 数据输出。
		1	输入	ST	PORTD<3> 数据输入。
RD4	RD4	0	输出	DIG	LATD<4> 数据输出。
		1	输入	ST	PORTD<4> 数据输入。
RD5	RD5	0	输出	DIG	LATD<5> 数据输出。
		1	输入	ST	PORTD<5> 数据输入
RD6	RD6	0	输出	DIG	LATD<6> 数据输出。
		1	输入	ST	PORTD<6> 数据输入。
RD7	RD7	0	输出	DIG	LATD<7> 数据输出。
		1	输入	ST	PORTD<7> 数据输入。

图注： OUT = 输出， IN = 输入， DIG = 数字输出， ST = 施密特缓冲器输入

表 9-8： 与 PORTD 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
PORTD(1)	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	51
LATD(1)	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	51
TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	51

注 1： 这些寄存器和 / 或位在 28-pin 器件上未实现。

© 2007 Microchip Technology Inc.

9.5 PORTE、 TRISE 和 LATE 寄存器

取决于特定的 PIC18F2450/4450 器件，PORTE 是通过两种不同的方式实现的。

对于 40/44-pin 器件，PORTE 是 4 位宽的端口。三个引脚（RE0/AN5、RE1/AN6 和 RE2/AN7）可被单独配置为输入或输出。这些引脚配有施密特触发输入缓冲器。当被选为模拟输入时，这些引脚将读为 0。

对应的数据方向寄存器是 TRISE。将 TRISE 某位置 1（= 1）时，会将 PORTE 的相应引脚设为输入（即，使相应的输出驱动器呈高阻状态）。将 TRISE 某位清零（= 0）时，会将 PORTE 的相应引脚设为输出（即，将输出锁存器中的内容置于选中引脚）。

TRISE 控制着 RE 引脚的方向，即使它们被用作模拟输入。用户在将这些引脚用作模拟输入时，必须确保将它们配置为输入。

注：	上电复位时， RE2:RE0 被配置为模拟输入。

数据锁存器 （LATE）也是存储器映射的。对 LATE 寄存器执行读－修改－写操作将读写 PORTE 的输出锁存值。

PORTE 的第 4 个引脚 （$\overline{\text{MCLR}}$/VPP/RE3）只能作为输入引脚。其操作由 MCLRE 配置位控制。当被选为端口引脚时（MCLRE = 0），它仅用作数字输入引脚；这样，它就没有与操作相关的 TRIS 或 LAT 位。否则，它用作器件的主清零输入。在任何一种配置中， RE3 都用作编程期间的编程电压输入。

注：	上电复位时，仅当主清零功能禁止时，才能将 RE3 使能为数字输入。

例 9-5： 初始化 PORTE

```
CLRF    PORTE   ; Initialize PORTE by
                ; clearing output
                ; data latches
CLRF    LATE    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0Ah     ; Configure A/D
MOVWF   ADCON1  ; for digital inputs
MOVLW   03h     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; Set RE<0> as inputs
                ; RE<1> as inputs
                ; RE<2> as outputs
```

9.5.1 28 引脚器件上的 PORTE

对于 28-pin 器件，仅当主清零功能禁止时（MCLRE = 0），才能使用 PORTE。在这些情况下， PORTE 是只有一位、仅由 RE3 组成的输入端口。引脚操作如前所述。

寄存器 9-1： PORTE 寄存器

U-0	U-0	U-0	U-0	R/W-x	R/W-0	R/W-0	R/W-0
—	—	—	—	RE3(1,2)	RE2(3)	RE1(3)	RE0(3)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **未用：**读为 0

bit 3-0 **RE3:RE0：**PORTE 数据输入位 (1,2,3)

注 1： 仅当主清零功能被禁止时 （MCLRE 配置位 = 0）可用，否则读为 0。

2： RE3 是在 28-pin 和 40/44-pin 器件上均可使用的唯一 PORTE 位。所有其他位只能在 PORTE 存在时使用（即 40/44-pin 器件）。

3： 在 28 引脚器件上未实现，读为 0。

表 9-9： PORTE I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RE0/AN5	RE0	0	输出	DIG	LATE<0> 数据输出；不受模拟输入影响。
		1	输入	ST	PORTE<0> 数据输入；当使能模拟输入时被禁止。
	AN5	1	输入	ANA	A/D 输入通道 5； POR 时的默认配置。
RE1/AN6	RE1	0	输出	DIG	LATE<1> 数据输出；不受模拟输入影响。
		1	输入	ST	PORTE<1> 数据输入；当使能模拟输入时被禁止。
	AN6	1	输入	ANA	A/D 输入通道 6； POR 时的默认配置。
RE2/AN7	RE2	0	输出	DIG	LATE<2> 数据输出；不受模拟输入影响。
		1	输入	ST	PORTE<2> 数据输入；当使能模拟输入时被禁止。
	AN7	1	输入	ANA	A/D 输入通道 7； POR 时的默认配置。
$\overline{MCLR}$/V_{PP}/RE3	RE3	—(1)	输入	ST	PORTE<3> 数据输入；当 MCLRE 配置位清零时被使能。
	$\overline{MCLR}$	—(1)	输入	ST	外部主清零输入；当 MCLRE 配置位置 1 时被使能。
	V_{PP}	—(1)	输入	ANA	高压检测，用于 ICSP™ 模式输入检测。始终可用，与引脚模式无关。

图注： OUT = 输出， IN = 输入， ANA = 模拟信号， DIG = 数字输出， ST = 施密特缓冲器输入。

注 1： RE3 没有相应的 TRISE<3> 位。无论模式如何，该引脚总是输入。

表 9-10： 与 PORTE 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
PORTE	—	—	—	—	RE3(1,2)	RE2(3)	RE1(3)	RE0(3)	51
LATE(3)	—	—	—	—	—	LATE2	LATE1	LATE0	51
TRISE(3)	—	—	—	—	—	TRISE2	TRISE1	TRISE0	51
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50

图注： – = 未用，读为 0

注 1： 仅当主清零功能被禁止时 （MCLRE 配置位 = 0）可用，否则读为 0。

2： RE3 是在 28-pin 和 40/44-pin 器件上均可使用的唯一 PORTE 位。所有其他位只能在 PORTE 存在时使用（即 40/44-pin 器件）。

3： 这些寄存器和 / 或位在 28-pin 器件上未实现。

© 2007 Microchip Technology Inc.

10.0 TIMER0 模块

Timer0 模块具有以下特性：

- 可由软件选择作为 8 位或 16 位定时器 / 计数器
- 可读写寄存器
- 专用的 8 位软件可编程预分频器
- 可选的时钟源 （内部或外部）
- 外部时钟的边沿选择
- 溢出时中断

T0CON 寄存器 （寄存器 10-1）控制该模块操作的所有方面，包括预分频比的选择。它是可读写的。

图 10-1 给出了 8 位模式下 Timer0 模块的简化框图。
图 10-2 给出了 16 位模式下 Timer0 模块的简化框图。

寄存器 10-1： T0CON：TIMER0 控制寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **TMR0ON**：Timer0 开 / 关控制位
1 = 使能 Timer0
0 = 停止 Timer0

bit 6 **T08BIT**：Timer0 8 位 /16 位控制位
1 = Timer0 被配置为 8 位定时器 / 计数器
0 = Timer0 被配置为 16 位定时器 / 计数器

bit 5 **T0CS**：Timer0 时钟源选择位
1 = T0CKI 引脚上的传输信号
0 = 内部指令周期时钟 （CLKO）

bit 4 **T0SE**：Timer0 时钟源边沿选择位
1 = 在 T0CKI 引脚上电平的下降沿递增
0 = 在 T0CKI 引脚上电平的上升沿递增

bit 3 **PSA**：Timer0 预分频器分配位
1 = 未分配 Timer0 预分频器。 Timer0 时钟输入不经过预分频器。
0 = 已分配 Timer0 预分频器。 Timer0 时钟输入来自预分频器的输出。

bit 2-0 **T0PS2:T0PS0**：Timer0 预分频值选择位
111 = 1:256 预分频值
110 = 1:128 预分频值
101 = 1:64 预分频值
100 = 1:32 预分频值
011 = 1:16 预分频值
010 = 1:8 预分频值
001 = 1:4 预分频值
000 = 1:2 预分频值

10.1 Timer0 工作原理

Timer0 既可用作定时器也可用作计数器。可通过清零 T0CS 位（T0CON<5>）来选择模式。在定时器模式下，该模块在每个时钟周期计时都会递增（默认情况下），除非选择了其他预分频值（见**第 10.3 节 “预分频器”**）。如果写入 TMR0 寄存器，那么在随后的两个指令周期内，计时都不再递增。用户可通过将调整值写入 TMR0 寄存器来避开这一问题。

通过将 T0CS 位置 1（= 1）选择计数器模式。在该模式下，Timer0 可在 RA4/T0CKI 引脚上信号的每个上升沿或下降沿递增。递增边沿由 Timer0 时钟源边沿选择位 T0SE（T0CON<4>）决定。清零该位即选择上升沿。下面讨论外部时钟输入的限制条件。

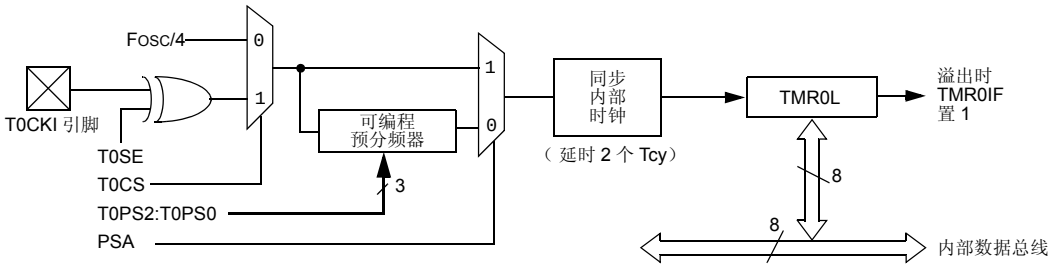
可以使用外部时钟源来驱动 Timer0。但是，必须满足一定要求，以确保外部时钟和内部相位时钟（Tosc）保持同步。在同步之后，定时器 / 计数器仍需要一定的延时才会引发递增操作。

10.2 Timer0 的 16 位读写模式

TMR0H 并不是 16 位模式下 Timer0 的高字节，而是被缓存的 Timer0 高字节，不可以被直接读写（见图 10-2）。在读 TMR0L 时使用 Timer0 高字节的内容更新 TMR0H。这样可以一次读取 Timer0 的全部 16 位，而无需验证读到的高字节和低字节的有效性（在连续读取高字节和低字节时，由于可能存在进位，因此需要验证读到的高字节和低字节的有效性）。

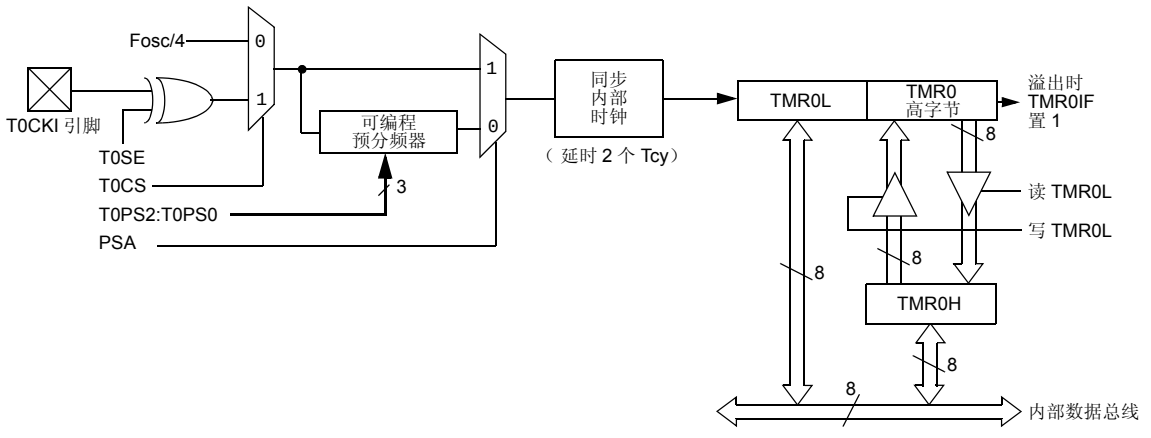
同样，写入 Timer0 的高字节也是通过 TMR0H 缓冲寄存器来操作的。在写入 TMR0L 的同时，使用 TMR0H 的内容更新 Timer0 的高字节。这样一次就可以完成 Timer0 全部 16 位的更新。

图 10-1： TIMER0 框图（8 位模式）



注： 复位时，Timer0 被使能为在 8 位模式下工作，其时钟输入来自 T0CKI 引脚的最大预分频信号。

图 10-2： TIMER0 框图（16 位模式）



注： 复位时，Timer0 被使能为在 8 位模式下工作，其时钟输入来自 T0CKI 引脚的最大预分频信号。

 © 2007 Microchip Technology Inc.

10.3 预分频器

Timer0 模块的预分频器为一个 8 位计数器。该预分频器不可直接读写。通过 PSA 和 T0PS2:T0PS0 位（T0CON<3:0>）进行预分频器的分配和设定预分频比值。

将 PSA 位清零可将预分频器分配给 Timer0 模块。预分频值可以在 1:2 到 1:256 之间进行选择，以 2 的整数次幂递增。

如果将预分频器分配给 Timer0 模块，所有写入 TMR0 寄存器的指令（例如，`CLRF TMR0`、`MOVWF TMR0` 和 `BSF TMR0` 等），都会将预分频器的计数值清零。

注： 如果将预分频器分配给 Timer0，写入 TMR0 会将预分频器的计数值清零，但不会改变预分频器的分配。

10.3.1 切换预分频器的分配

预分频器的分配完全由软件控制，并且在程序执行期间可以随时更改。

10.4 Timer0 中断

8 位模式下的 TMR0 寄存器从 FFh 到 00h 发生溢出，或 16 位模式下的 TMR0 从 FFFFh 到 0000h 发生溢出时，将产生 TMR0 中断。这种溢出会使 TMR0IF 标志位置 1。可以通过清零 TMR0IE 位（INTCON<5>）来屏蔽该中断。在重新允许该中断前，必须在中断服务程序中用软件清零 TMR0IF 位。

由于 Timer0 在休眠模式下是关闭的，所以 TMR0 中断无法将处理器从休眠状态唤醒。

表 10-1： 与 TIMER0 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
TMR0L	Timer0 寄存器的低字节								50
TMR0H	Timer0 寄存器的高字节								50
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	50
TRISA	—	TRISA6(1)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	51

图注： – = 未用单元，读为 0。 Timer0 不使用阴影单元。

注 1： RA6 根据主振荡器模式可被配置为端口引脚。当端口引脚被禁止时，所有相关位均读为 0。

注：

 © 2007 Microchip Technology Inc.

11.0 TIMER1 模块

Timer1 定时器 / 计数器模块具有以下特性：

- 可由软件选择作为 16 位定时器或计数器
- 可读写的 8 位寄存器 （TMR1H 和 TMR1L）
- 可选择器件时钟或 Timer1 内部振荡器作为时钟源（内部或外部）
- 溢出时中断
- CCP 特殊事件触发模块复位
- 器件时钟状态标志位 （T1RUN）

图 11-1 给出了 Timer1 模块的简化框图。图 11-2 给出了此模块在读写模式下的工作原理框图。

此模块自身带有低功耗振荡器可提供额外的时钟选项。Timer1 振荡器也可作为单片机处于节能状态时的低功耗时钟源。

在对外部元件数量和代码开销要求苛刻的应用中，Timer1 可以用于提供实时时钟 （RTC）。

Timer1 由 T1CON 控制寄存器 （寄存器 11-1）控制。该寄存器还包含 Timer1 振荡器使能位 （T1OSCEN）。可以通过将控制位 TMR1ON （T1CON<0>）置 1 或清零来使能或禁止 Timer1。

寄存器 11-1： T1CON：TIMER1 控制寄存器

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **RD16**：16 位读 / 写模式使能位
1 = 使能 Timer1 通过一次 16 位操作进行寄存器读 / 写
0 = 使能 Timer1 通过两次 8 位操作进行寄存器读 / 写

bit 6 **T1RUN**：Timer1 系统时钟状态位
1 = 器件时钟由 Timer1 振荡器产生
0 = 器件时钟由另一个时钟源产生

bit 5-4 **T1CKPS1:T1CKPS0**：Timer1 输入时钟预分频值选择位
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 **T1OSCEN**：Timer1 振荡器使能位
1 = 使能 Timer1 振荡器
0 = 关闭 Timer1 振荡器
关闭振荡器的反相器和反馈电阻以降低功耗。

bit 2 **$\overline{\text{T1SYNC}}$**：Timer1 外部时钟输入同步选择位
当 TMR1CS = 1 时：
1 = 不同步外部时钟输入
0 = 同步外部时钟输入
当 TMR1CS = 0 时：
该位为无关位。当 TMR1CS = 0 时， Timer1 使用内部时钟。

bit 1 **TMR1CS**：Timer1 时钟源选择位
1 = 使用 RC0/T1OSO/T1CKI 引脚上的外部时钟 （上升沿计数）
0 = 内部时钟 （FOSC/4）

bit 0 **TMR1ON**：Timer1 使能位
1 = 使能 Timer1
0 = 停止 Timer1

11.1 Timer1 工作原理

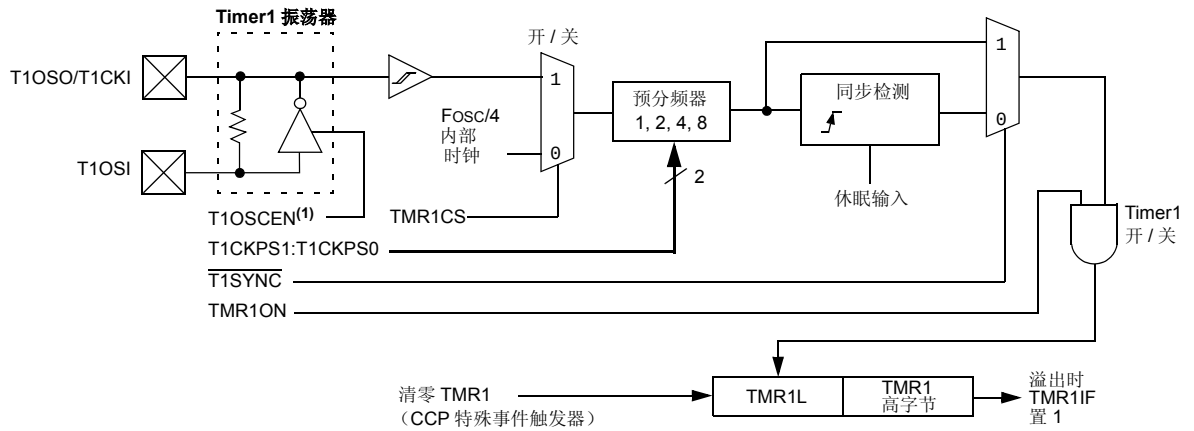
Timer1 可在以下模式工作：

- 定时器
- 同步计数器
- 异步计数器

工作模式由时钟选择位 TMR1CS（T1CON<1>）决定。当 TMR1CS 清零 （= 0）时， Timer1 在每个内部指令周期（FOSC/4）递增。当该位置 1 时，Timer1 在 Timer1 外部时钟输入信号或 Timer1 振荡器输出信号 （如果使能）的每个上升沿递增。

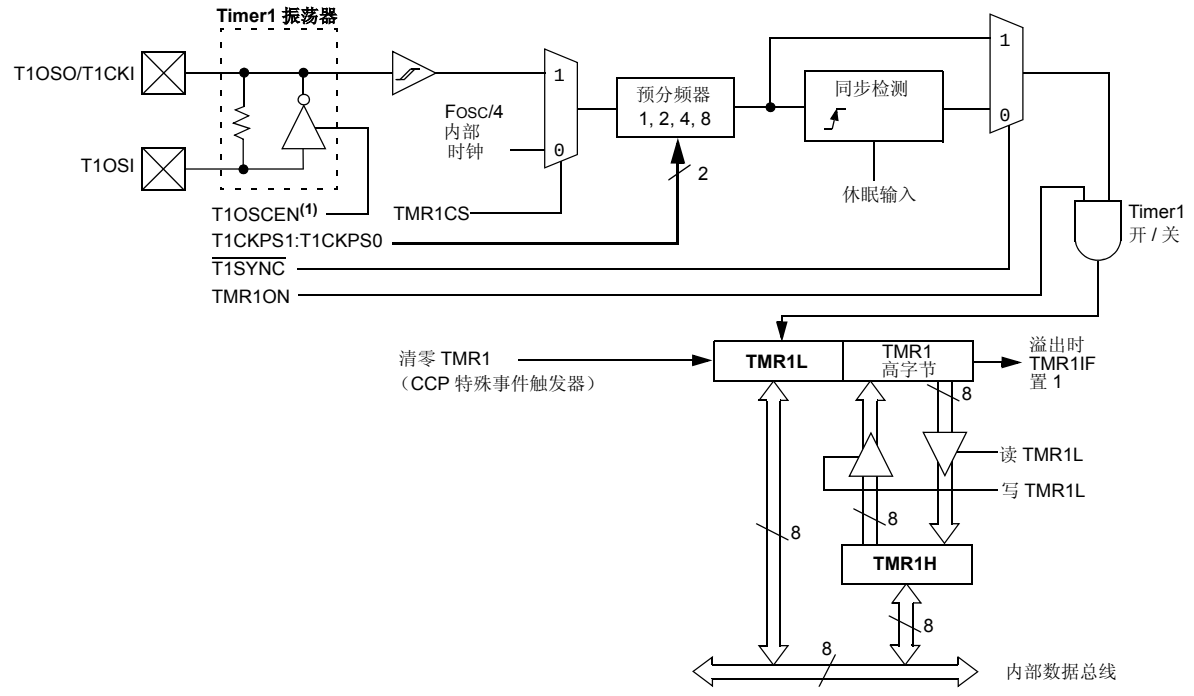
当使能 Timer1 时， RC1/T1OSI/$\overline{UOE}$ 和 RC0/T1OSO/T1CKI 引脚变为输入引脚。这意味着 TRISC<1:0> 的值被忽略并且这些引脚将读为 0。

图 11-1： TIMER1 框图



注 1： 当使能位 T1OSCEN 清零时，将关断振荡器的反相器和反馈电阻以减少功耗。

图 11-2： TIMER1 框图 （16 位读 / 写模式）



注 1： 当使能位 T1OSCEN 清零时，将关断振荡器的反相器和反馈电阻以减少功耗。

© 2007 Microchip Technology Inc.

11.2 Timer1 的 16 位读 / 写模式

可将 Timer1 配置为 16 位读写模式（见图 11-2）。当 RD16 控制位（T1CON<7>）置 1 时，TMR1H 的地址被映射到 Timer1 的高字节缓冲寄存器。读 TMR1L 将把 Timer1 的高字节的内容装入 Timer1 高字节缓冲器。这种方式使用户可以精确地读取 Timer1 的全部 16 位，而不需要像先读高字节再读低字节那样，由于两次读取之间可能存在进位，而不得不验证读取的有效性。

写 Timer1 的高字节也必须通过 TMR1H 缓冲寄存器进行。在写入 TMR1L 的同时，使用 TMR1H 的内容更新 Timer1 的高字节。这样允许用户将 16 位值一次写入 Timer1 的高字节和低字节。

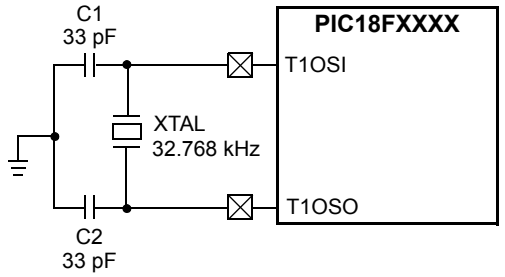
在该模式下不能直接读写 Timer1 的高字节。所有读写都必须通过 Timer1 高字节缓冲寄存器来进行。写入 TMR1H 不会清零 Timer1 预分频器。只有在写 TMR1L 时才会清零该预分频器。

11.3 Timer1 振荡器

片上晶体振荡器电路连接在 T1OSI（输入）引脚和 T1OSO（放大器输出）引脚之间。可以通过将 Timer1 振荡器使能位 T1OSCEN（T1CON<3>）置 1 来使能该振荡电路。该振荡电路是一种低功耗电路，它采用了额定振荡频率为 32 kHz 的晶振。在所有功耗管理模式下都可继续运行。图 11-3 所示是典型的 LP 振荡器电路。表 11-1 给出了供 Timer1 振荡器选择的电容值。

用户必须提供软件延时来确保 Timer1 振荡器的正常起振。

图 11-3： TIMER1 LP振荡器的外部元件



注： 关于电容选择的更多信息，请参见表11-1下方的“注”。

表 11-1： TIMER1振荡器的电容选择(2,3,4)

振荡器类型	频率	C1	C2
LP	32 kHz	27 pF(1)	27 pF(1)

注

1： Microchip 建议将该值作为验证振荡电路的起始点。

2： 电容越大，振荡器越稳定，但起振时间越长。

3： 因为每种谐振器 / 晶振都有其自身特性，用户应当向谐振器 / 晶振制造厂商咨询外部元件的适当值。

4： 上述电容值仅供设计参考。

11.3.1 使用 TIMER1 作为时钟源

在功耗管理模式下也可以将Timer1振荡器用作时钟源。通过将时钟选择位 SCS1:SCS0（OSCCON<1:0>）设置为 01，器件可以切换到 SEC_RUN 模式，CPU 和外设都可以用 Timer1 振荡器作为时钟源。如果 IDLEN 位（OSCCON<7>）被清零并且执行了 SLEEP 指令，器件将进入 SEC_IDLE 模式。更多详细信息，请参见**第 3.0 节 “功耗管理模式”**。

无论何时将 Timer1 振荡器用作时钟源，Timer1 系统时钟状态标志位 T1RUN（T1CON<6>）均会置 1。这可用于确定控制器的当前时钟模式。该位也可指示故障保护时钟监视器当前正使用的时钟源。如果使能了故障保护时钟监视器并且 Timer1 振荡器在提供时钟信号时发生了故障，查询 T1RUN 位可以确定时钟源是 Timer1 振荡器还是其他时钟源。

11.3.2 低功耗 TIMER1 选项

根据器件配置，Timer1 振荡器可以在两种不同的功耗级别下工作。当 LPT1OSC 配置位置 1 时，Timer1 振荡器在低功耗模式下工作。当 LPT1OSC 清零时，Timer1 在高功耗模式下工作。不管器件工作在什么模式下，特定模式的功耗都是相对固定的。默认将 Timer1 配置为工作在功耗较高的模式下。

由于低功耗 Timer1 模式对干扰更加敏感，噪声环境可能会导致振荡器工作不稳定。因此低功耗选项最适合那些需要重点考虑节省功耗的低噪声应用。

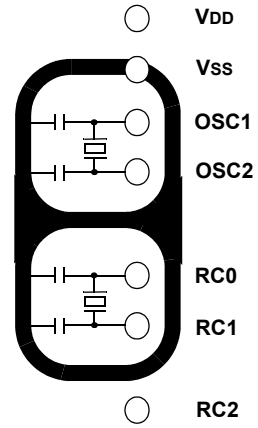
11.3.3 TIMER1 振荡器布线注意事项

Timer1 振荡器电路在工作期间仅消耗极少的电流。鉴于此振荡器的低功耗特性，它对附近变化较快的信号比较敏感。

如图 11-3 所示，振荡电路应该尽可能靠近单片机。除了 VSS 或 VDD 外，在该振荡电路区域内不应有其他电路。

如果必须要在该振荡器附近布置高速电路（如输出比较模式或 PWM 模式的 CCP1 引脚，或使用 OSC2 引脚的主振荡器），那么在该振荡电路周围布置接地保护环（如图 11-4 所示），对于单面 PCB 板或外加接地层的电路板来讲可能会有帮助。

图 11-4： 带有接地保护环的振荡电路



注: 未按比例绘制。

11.4 Timer1 中断

TMR1 寄存器对（TMR1H:TMR1L）从 0000h 开始，增加到 FFFFh，然后溢出返回到 0000h 重新开始计数。如果允许了 Timer1 中断，则溢出时会产生 Timer1 中断，并由中断标志位 TMR1IF （PIR1<0>）捕捉。可以通过对 Timer1 中断允许位 TMR1IE （PIE1<0>）置 1 或清零来允许或禁止该中断。

11.5 使用 CCP 特殊事件触发信号来复位 Timer1

如果CCP模块配置为在比较模式下产生特殊事件触发信号 （CCP1M3:CCP1M0 = 1011），该信号将复位 Timer1。如果使能了 A/D 模块，来自 CCP1 的触发信号还将启动 A/D 转换 （更多信息，请参见**第 13.3.4 节“特殊事件触发器”**）。

要使用这一功能，必须将模块配置为定时器或同步计数器。在这种情况下，CCPRH:CCPRL 这对寄存器实际上变成了 Timer1 的周期寄存器。

如果 Timer1 在异步计数器模式下运行，复位操作可能不起作用。

如果 Timer1 的写操作和特殊事件触发同时发生，则写操作优先。

注：	CCP1 模块产生的特殊事件触发信号不会将 TMR1IF 中断标志位 （PIR1<0>）置 1。

11.6 使用 Timer1 作为实时时钟

为 Timer1 外接一个 LP 振荡器（如**第 11.3 节“Timer1 振荡器”**中所述），可以允许用户在他们的应用中包括 RTC 功能。只需通过一个提供精确时基的廉价时钟晶振以及几行计算时间的应用程序代码就可以实现这一功能。当器件在休眠模式下工作并使用电池或超大容量电容作为电源时，可省去另外的 RTC 器件和备用电池。

应用代码程序 RTCisr （如例 11-1 所示），演示了使用中断服务程序以 1 秒的间隔递增计数器的简单方法。将 TMR1 寄存器对的值递增至溢出将触发中断并调用中断服务程序，该程序会使秒计数器加 1，其他的分钟和小时计数器则会在前面的计数器溢出时加 1。

由于这对寄存器为 16 位宽，因此使用 32.768 kHz 时钟，将其计数到溢出需要 2 秒。要使溢出按所需的 1 秒间隔进行，必须预先装载这对寄存器。最简单的方法是使用 BSF 指令将 TMR1H 的最高有效位置 1。请注意决不要预先加载或改变 TMR1L 寄存器，这样做可能会引起多个周期的累积误差。

要使此方法精确，Timer1 必须工作于异步模式且必须允许 Timer1 溢出中断 （PIE1<0> = 1），如程序 RTCinit 所示。同时 Timer1 振荡器也必须被使能并始终运行。

 © 2007 Microchip Technology Inc.

例 11-1：　　使用 TIMER1 中断服务实现实时时钟

```
RTCinit
            MOVLW    80h           ; Preload TMR1 register pair
            MOVWF    TMR1H         ; for 1 second overflow
            CLRF     TMR1L
            MOVLW    b'00001111'   ; Configure for external clock,
            MOVWF    T1OSC         ; Asynchronous operation, external oscillator
            CLRF     secs          ; Initialize timekeeping registers
            CLRF     mins          ;
            MOVLW    d'12'
            MOVWF    hours
            BSF      PIE1, TMR1IE  ; Enable Timer1 interrupt
            RETURN
RTCisr
            BSF      TMR1H, 7      ; Preload for 1 sec overflow
            BCF      PIR1, TMR1IF  ; Clear interrupt flag
            INCF     secs, F       ; Increment seconds
            MOVLW    d'59'         ; 60 seconds elapsed?
            CPFSGT   secs
            RETURN                 ; No, done
            CLRF     secs          ; Clear seconds
            INCF     mins, F       ; Increment minutes
            MOVLW    d'59'         ; 60 minutes elapsed?
            CPFSGT   mins
            RETURN                 ; No, done
            CLRF     mins          ; clear minutes
            INCF     hours, F      ; Increment hours
            MOVLW    d'23'         ; 24 hours elapsed?
            CPFSGT   hours
            RETURN                 ; No, done
            MOVLW    d'01'         ; Reset hours to 1
            MOVWF    hours
            RETURN                 ; Done
```

表 11-2：　　与 TIMER1 作为定时器 / 计数器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
TMR1L	Timer1 寄存器的低字节								50
TMR1H	Timer1 寄存器的高字节								50
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	50

图注：　– = 未用，读为 0。 Timer1 模块不使用阴影单元。

注：

 © 2007 Microchip Technology Inc.

12.0 TIMER2 模块

Timer2 模块定时器具有以下特性：

- 8 位定时器和周期寄存器（分别为 TMR2 和 PR2）
- 可读写 （以上两个寄存器）
- 可软件编程的预分频器 （分频比为 1:1、 1:4 和 1:16）
- 可软件编程的后分频器 （分频比为 1:1 到 1:16）
- TMR2 与 PR2 匹配时产生中断

此模块由 T2CON 寄存器（寄存器 12-1）控制，此寄存器使能或禁止定时器并配置预分频器和后分频器。可以通过清零控制位 TMR2ON（T2CON<2>）关闭 Timer2，以实现功耗最小。

图 12-1 给出了此模块的简化框图。

12.1 Timer2 工作原理

在正常工作模式下，TMR2 从 00h 开始，每个时钟周期（FOSC/4）加 1。2 位计数器 / 预分频器提供了对时钟输入不分频、4 分频和 16 分频三种选项，并可通过预分频控制位 T2CKPS1:T2CKPS0 （T2CON<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。此信号也会使 TMR2 的值在下一个周期复位到 00h，并驱动计数器 / 后分频器（见 **第 12.2 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。预分频和后分频计数器均会在发生以下事件时清零：

- 对 TMR2 寄存器进行写操作
- 对 T2CON 寄存器进行写操作
- 任何器件复位 （上电复位、 $\overline{\text{MCLR}}$ 复位、看门狗定时器复位或欠压复位）

写 T2CON 时 TMR2 不会清零。

寄存器 12-1： T2CON：TIMER2 控制寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用：**读为 0

bit 6-3 **T2OUTPS3:T2OUTPS0：**Timer2 输出后分频比选择位
0000 = 1:1 后分频比
0001 = 1:2 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 **T2CKPS1:T2CKPS0：**Timer2 时钟预分频值选择位
00 = 预分频值为 1
01 = 预分频值为 4
1x = 预分频值为 16

12.2 Timer2 中断

Timer2 也可以产生可选的器件中断。 Timer2 输出信号（TMR2 与 PR2 匹配时）为 4 位输出计数器 / 后分频器提供输入。此计数器产生的 TMR2 匹配中断标志位为 TMR2IF （PIR1<1>）。可以通过将 TMR2 匹配中断允许位 TMR2IE （PIE1<1>）置 1 来允许此中断。

可以通过后分频控制位T2OUTPS3:T2OUTPS0（T2CON<6:3>）在 16 个后分频比值选项（从 1:1 到 1:16）中选择其一。

12.3 TMR2 输出

TMR2 的不经分频的输出主要用于 CCP 模块，它用作 CCP 模块在 PWM 模式下工作时的时基。

图 12-1： TIMER2 框图

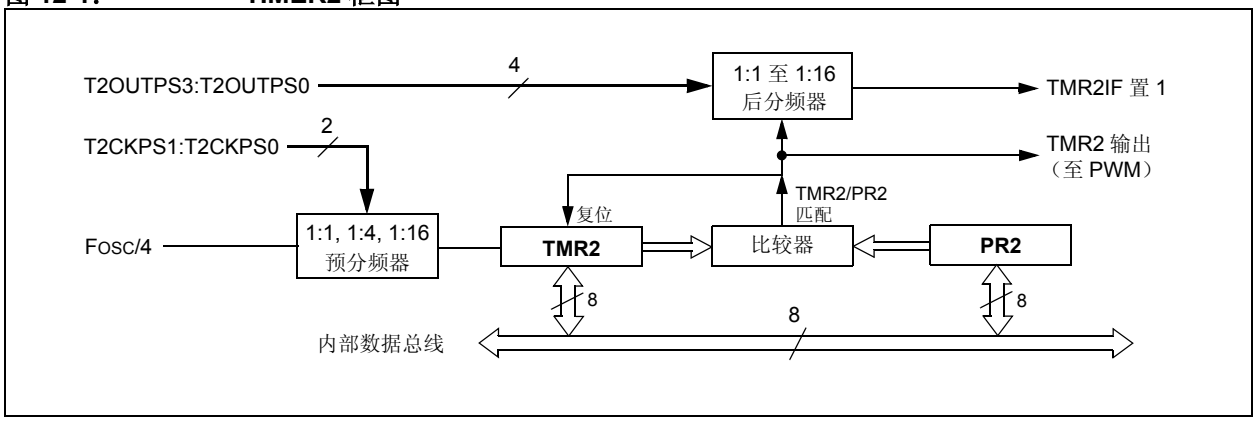


表 12-1： 与 TIMER2 作为定时器 / 计数器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
TMR2	Timer2 寄存器								50
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	50
PR2	Timer2 周期寄存器								50

图注： – = 未用，读为 0。 Timer2 模块不使用阴影单元。

© 2007 Microchip Technology Inc.

13.0 捕捉 / 比较 /PWM （CCP）模块

PIC18F2450/4450器件有一个CCP（捕捉/比较/PWM）模块。该模块包含一个 16 位寄存器，可用作 16 位捕捉寄存器、16位比较寄存器或PWM主/从占空比寄存器。

寄存器 13-1： CCP1CON：捕捉 / 比较 /PWM 控制寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未用：**读为 0

bit 5-4 **DC1B1:DC1B0**：CCP 模块的 PWM 占空比位
捕捉模式：
未使用。
比较模式：
未使用。
PWM 模式：
这两位是 10 位 PWM 占空比的两个最低位 （bit 1 和 bit 0）。占空比的高 8 位在 CCPR1L 中。

bit 3-0 **CCP1M3:CCP1M0**：CCP 模块模式选择位
0000 = 禁止捕捉 / 比较 /PWM （复位 CCP 模块）
0001 = 保留
0010 = 比较模式：匹配时输出电平翻转 （CCP1IF 位置 1）
0011 = 保留
0100 = 捕捉模式：每个下降沿
0101 = 捕捉模式：每个上升沿
0110 = 捕捉模式：每 4 个上升沿
0111 = 捕捉模式：每 16 个上升沿
1000 = 比较模式：初始化CCP1引脚为低电平；比较匹配时强制CCP1引脚为高电平（CCP1IF位置1）
1001 = 比较模式：初始化CCP1引脚为高电平；比较匹配时强制CCP1引脚为低电平（CCP1IF位置1）
1010 = 比较模式：比较匹配时产生软件中断 （CCP1IF 位置 1， CCP1 引脚反映 I/O 状态）
1011 = 比较模式：当CCP1发生匹配时触发特殊事件信号、复位定时器并启动A/D转换（CCP1IF位置1）
11xx = PWM 模式

13.1 CCP 模块配置

该捕捉 / 比较 /PWM 模块与一个控制寄存器 （通常为 CCP1CON）和一个数据寄存器（CCPR1）相对应。数据寄存器由两个 8 位寄存器组成：CCPR1L （低字节）和 CCPR1H （高字节）。所有寄存器都是可读写的。

13.1.1 CCP 模块和定时器资源

CCP 模块根据选定的模式使用 Timer1 或 Timer2。该模块在捕捉或比较模式下使用 Timer1，而在 PWM 模式下使用 Timer2。

表 13-1： CCP 模式——定时器资源

CCP 模式	定时器资源
捕捉	Timer1
比较	Timer1
PWM	Timer2

在异步计数器模式下的 Timer1 中，捕捉操作将无法进行。

13.2 捕捉模式

在捕捉模式下，当在相应的 CCP1 引脚上发生事件时，CCPR1H:CCPR1L 寄存器对捕捉 TMR1 寄存器的 16 位值。事件定义为下列情况之一：

- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

事件由模式选择位CCP1M3:CCP1M0（CCP1CON<3:0>）选择。当完成一次捕捉时，中断请求标志位 CCP1IF 置 1 ；它必须用软件清零。如果在读取寄存器 CCPR1 值之前发生了另一次捕捉，那么原来的捕捉值会被新的捕捉值覆盖。

13.2.1 CCP1 引脚配置

在捕捉模式下，应通过将相应的 TRIS 方向位置 1 把 CCP1 引脚配置为输入。

注：	如果将 RC2/CCP1 引脚配置为输出，则对该端口的写操作将会产生捕捉条件。

13.2.2 软件中断

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该保持 CCP1IE 中断允许位清零以避免错误中断。应在工作模式改变后清零中断标志位 CCP1IF。

13.2.3 CCP 预分频器

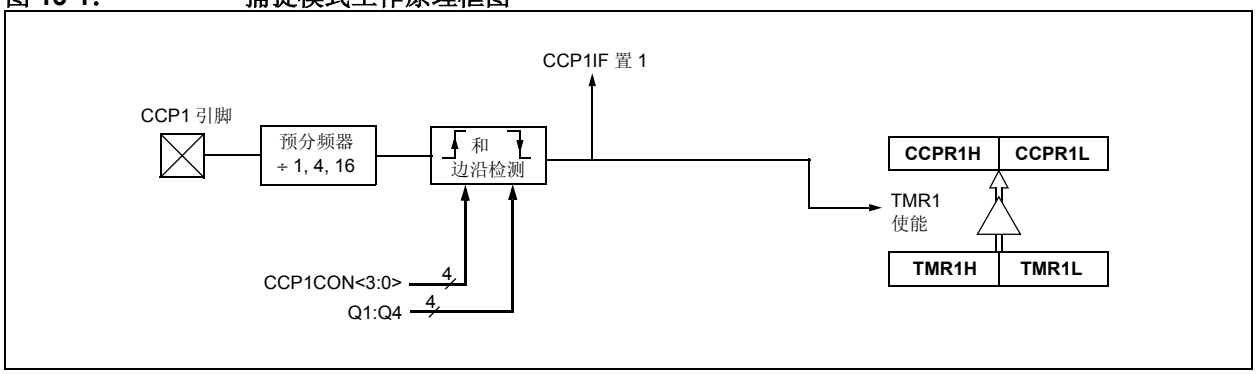
在捕捉模式下有 4 种预分频比设置。它们可作为工作模式的一部分由模式选择位（CCP1M3:CCP1M0）选择。只要关闭 CCP 模块或禁止捕捉模式，预分频计数器就会被清零。这意味着任何复位都会将预分频计数器清零。

在两个预分频比之间切换会产生中断。而且，预分频计数器不会被清零；因此第一次捕捉可能来自于一个非零的预分频器。例 13-1 给出了切换捕捉预分频比时建议采用的方法。这个示例使预分频计数器清零且不会产生错误中断。

例 13-1： 改变捕捉预分频比（以 CCP1 为例）

```
CLRF   CCP1CON          ; Turn CCP module off
MOVLW  NEW_CAPT_PS      ; Load WREG with the
                        ; new prescaler mode
                        ; value and CCP ON
MOVWF  CCP1CON          ; Load CCP1CON with
                        ; this value
```

图 13-1： 捕捉模式工作原理框图



 © 2007 Microchip Technology Inc.

13.3 比较模式

在比较模式下，16 位 CCPR1 寄存器的值不断与 TMR1 寄存器对的值作比较。当两者匹配时，CCP1 引脚将会：

- 驱动为高电平
- 驱动为低电平
- 电平翻转 （高电平变为低电平或低电平变为高电平）
- 保持不变 （即反映 I/O 锁存器的状态）

引脚动作取决于模式选择位 （CCP1M3:CCP1M0）的值。同时，中断标志位 CCP1IF 置 1。

13.3.1 CCP1 引脚配置

用户必须通过将相应的 TRIS 位清零把 CCP1 引脚配置为输出。

注：	清零 CCP1CON 寄存器将把 RC2 比较输出锁存器强制设为默认的低电平状态。

13.3.2 TIMER1 模式选择

如果 CCP 模块使用比较功能，则 Timer1 必须运行在定时器模式或同步计数器模式下。在异步计数器模式下，比较操作可能无法进行。

13.3.3 软件中断模式

当选择了 “生成软件中断模式” 时（CCP1M3:CCP1M0 = 1010），CCP1 引脚不受影响。如果已经使能，将仅产生 CCP 中断并将 CCP1IE 位置 1。

13.3.4 特殊事件触发器

CCP 模块配备了一个 “特殊事件触发器”。在比较模式下可产生内部硬件信号以触发其他模块动作。通过选择 “比较特殊事件触发” 模式（CCP1M3:CCP1M0 = 1011），使能 “特殊事件触发器”。

对于 CCP 模块，“特殊事件触发器” 将把 Timer1 寄存器对复位。这样 CCPR1 寄存器可用作 Timer1 的可编程周期寄存器。

CCP1 的 “特殊事件触发器” 还能启动 A/D 转换。要实现此功能，必须首先使能 A/D 转换器。

图 13-2： 比较模式工作原理框图

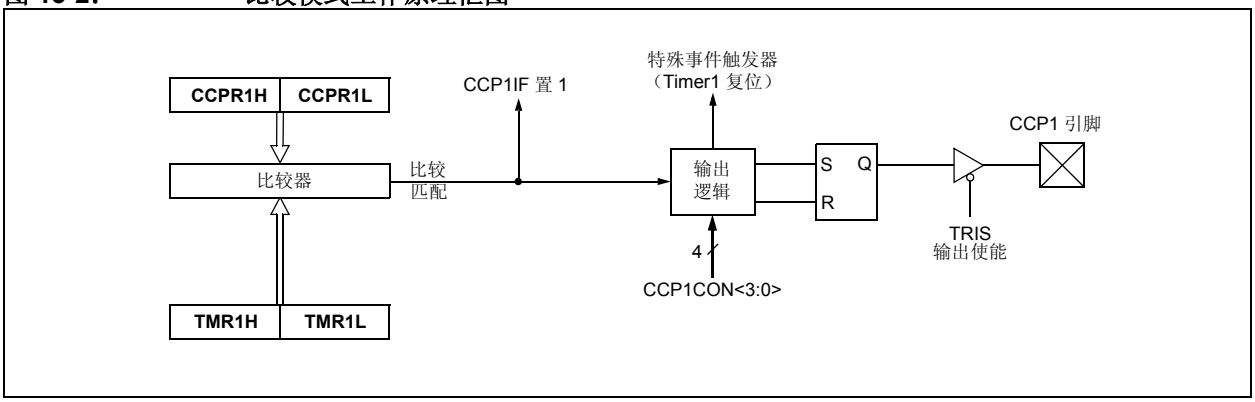


表 13-2： 与捕捉、比较和 TIMER1 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
RCON	IPEN	SBOREN(1)	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	50
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	51
TMR1L	Timer1 寄存器的低字节								50
TMR1H	Timer1 寄存器的高字节								50
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	50
CCPR1L	捕捉 / 比较 /PWM 寄存器 1 的低字节								50
CCPR1H	捕捉 / 比较 /PWM 寄存器 1 的高字节								50
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	50

图注： – = 未用，读为 0。捕捉 / 比较和 Timer1 不使用阴影单元。

注 1： SBOREN 位仅在 BOREN<1:0> = 01 时可用；否则，它被禁止且读为 0。

© 2007 Microchip Technology Inc.

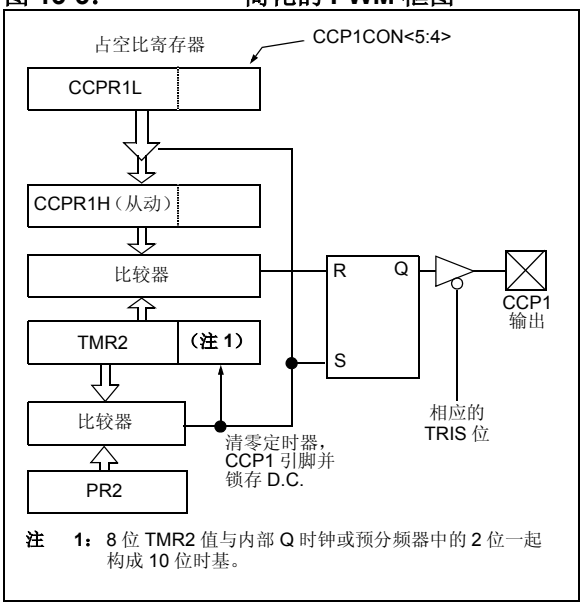
13.4 PWM 模式

在脉宽调制（Pulse-Width Modulation，PWM）模式下，CCP1引脚会产生高达10位分辨率的PWM输出信号。

图 13-3 给出了 PWM 模式下 CCP 模块的简化框图。

关于如何设置 CCP 模块使之工作于 PWM 模式的步骤，请参见**第 13.4.3 节 “设置 PWM 工作模式”**。

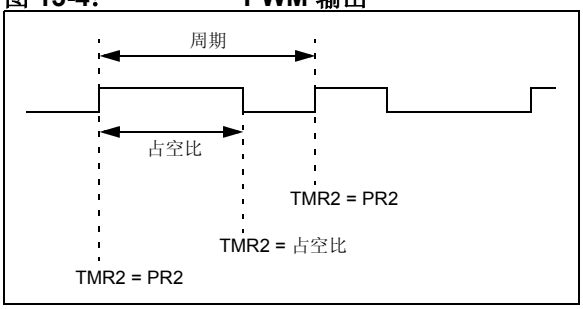
图 13-3： 简化的 PWM 框图



注 1： 8 位 TMR2 值与内部 Q 时钟或预分频器中的 2 位一起构成 10 位时基。

PWM 输出 （图 13-4）有一个时基 （周期）和一段输出保持为高电平的时间（占空比）。PWM 的频率是周期的倒数 （1/ 周期）。

图 13-4： PWM 输出



13.4.1 PWM 周期

可通过写 PR2 寄存器指定 PWM 周期。PWM 周期可由以下公式计算：

公式 13-1：

$$\text{PWM 周期} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 预分频值})$$

PWM 频率定义为 1/[PWM 周期]。

当 TMR2 中的值与 PR2 中的值相等时，在下一个递增周期将发生以下 3 个事件：

- TMR2 被清零
- CCP1 引脚置 1（例外：如果 PWM 占空比 = 0%，CCP1 引脚将不会置 1）
- PWM 占空比从 CCPR1L 锁存到 CCPR1H

注： 确定 PWM 频率时不会用到 Timer2 后分频器（见**第 12.0 节 “Timer2 模块”**）。后分频器可用不同于PWM输出频率的频率进行数据更新。

13.4.2 PWM 占空比

通过写 CCPR1L 寄存器和 CCP1CON<5:4> 位来指定 PWM 占空比。分辨率最高可达 10 位。CCPR1L 包含占空比的高 8 位而 CCP1CON<5:4> 包含低 2 位。这 10 位值由 CCPR1L:CCP1CON<5:4> 表示。以下公式用于计算 PWM 的占空比时间：

公式 13-2：

$$\text{PWM 占空比} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 预分频值})$$

可以在任何时候写入 CCPR1L 和 CCP1CON<5:4>，但是在 PR2 和 TMR2 发生匹配 （即周期结束）前占空比值不会被锁存到 CCPR1H 中。在 PWM 模式下，CCPR1H 是只读寄存器。

CCPR1H 寄存器和一个 2 位的内部锁存器用于给 PWM 占空比提供双重缓冲。这种双重缓冲结构非常重要，它可以避免在 PWM 操作中产生毛刺。

当 CCPR1H 和 2 位锁存值与 TMR2 （以及内部 2 位 Q 时钟或 TMR2 预分频值的 2 位）匹配时，CCP1 引脚被清零。

在给定 PWM 频率的情况下，最大的 PWM 分辨率（位）由以下公式给出：

公式 13-3：

$$\text{PWM 分辨率 （最大）} = \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)} \text{位}$$

注：	如果 PWM 占空比的值大于 PWM 周期，则 CCP1 引脚将不会被清零。

13.4.3 设置 PWM 工作模式

当配置 CCP 模块使之工作于 PWM 模式时，应遵循以下步骤：

1. 通过写 PR2 寄存器设置 PWM 周期。
2. 通过写 CCPR1L 寄存器和 CCP1CON<5:4> 位设置 PWM 占空比。
3. 通过清零相应的 TRIS 位将 CCP1 引脚设为输出引脚。
4. 设置 TMR2 预分频值并随后通过写 T2CON 使能 Timer2。
5. 配置 CCP 模块使之工作于 PWM 模式。

表 13-3： 40 MHz 时 PWM 频率和分辨率示例

PWM 频率	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
定时器预分频值 （1、4 和 16）	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最大分辨率 （位）	10	10	10	8	7	6.58

表 13-4： 与 PWM 和 TIMER2 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
RCON	IPEN	SBOREN(1)	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	50
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	51
TMR2	Timer2 寄存器								50
PR2	Timer2 周期寄存器								50
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	50
CCPR1L	捕捉 / 比较 /PWM 寄存器 1 的低字节								50
CCPR1H	捕捉 / 比较 /PWM 寄存器 1 的高字节								50
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	50

图注： – = 未用，读为 0。 PWM 或 Timer2 不使用阴影单元。

注 1： SBOREN 位仅在 BOREN<1:0> = 01 时可用；否则，它被禁止且读为 0。

© 2007 Microchip Technology Inc.

14.0 通用串行总线 （USB）

本节将详细介绍 USB外设。由于本模块的特殊性质，要求读者具备 USB 方面的基础知识。在**第 14.9 节 “USB 概述”**中提供了一些 USB 的高级信息，仅供应用设计参考。我们鼓励设计人员参考 USB Implementers Forum（USB-IF）发布的官方规范，以获得最新的信息。本文档发布时，USB 规范 2.0 版（USB Specification Revision 2.0）是最新的规范。

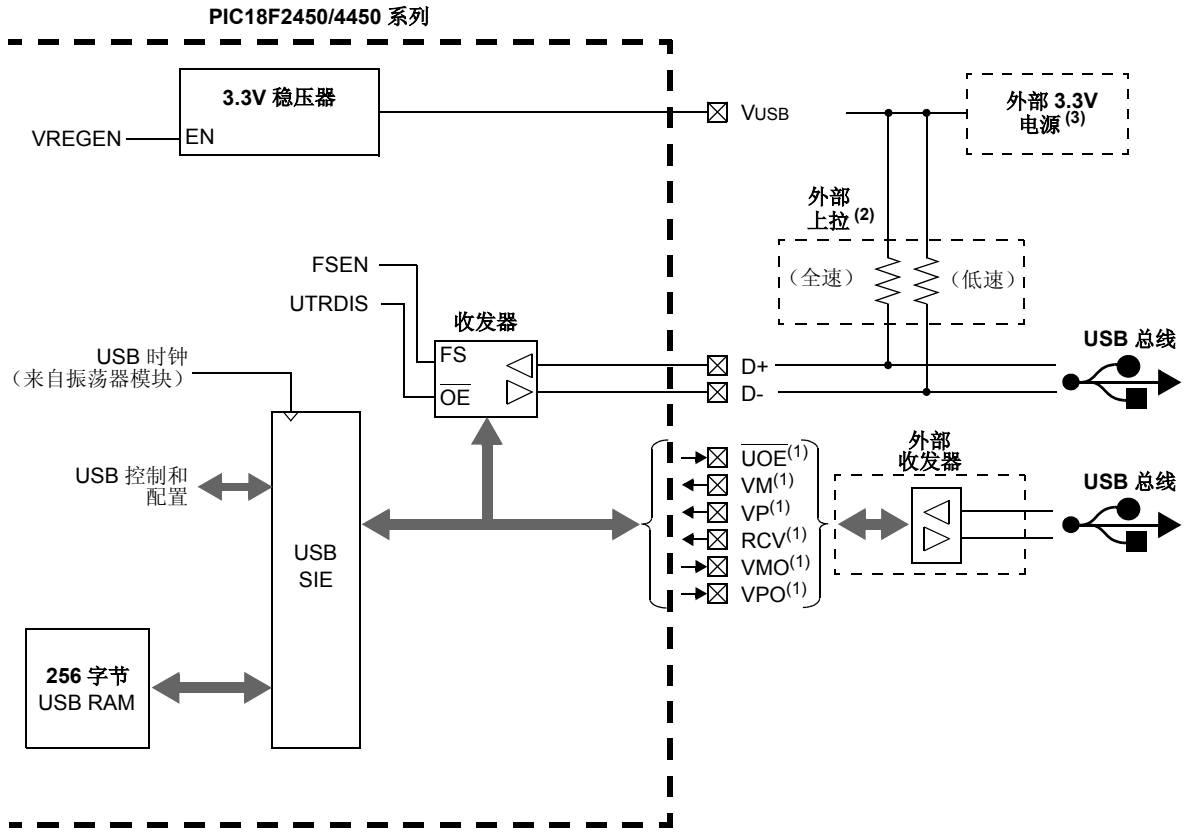
14.1 USB 外设概述

PIC18F2450/4450器件系列包含全速和低速兼容的USB串行接口引擎（SIE），它允许在 USB 主机和 PIC® 单片机之间实现快速通信。 SIE 可使用内部收发器直接与 USB 接口，也可以与外部收发器相连。还有一个内部 3.3V 稳压器可用来对 5V应用中的内部收发器供电。

该模块包括了一些特殊硬件功能以便改善性能，如数据存储器空间 （USB RAM）中的双端口存储器，用于单片机内核与 SIE 之间的直接存储器访问；还有缓冲器描述符，允许用户自由地对 USB RAM 空间内的端点存储器编程。

图 14-1 提供了 USB 外设及其功能的总体概述。

图 14-1： USB 外设和选项



注 1： 该信号仅当内部收发器被禁止时 （UTRDIS = 1）可用。

2： 可由 VUSB 引脚或外部 3.3V 电源线提供上拉电阻。

3： 使用外部 3.3V 电源时，不要使能内部稳压器。

14.2 USB 状态和控制

USB 模块的运行是通过 3 个控制寄存器配置和管理的。此外，总共使用了 22 个寄存器管理实际 USB 事务。这些寄存器是：

- USB 控制寄存器 （UCON）
- USB 配置寄存器 （UCFG）
- USB 传输状态寄存器 （USTAT）
- USB 器件地址寄存器 （UADDR）
- 帧编号寄存器 （UFRMH:UFRML）
- 端点使能寄存器 0 到 15 （UEPn）

14.2.1 USB 控制寄存器 （UCON）

USB 控制寄存器 （寄存器 14-1）包含控制传输期间的模块行为所需的控制位。寄存器包含控制以下内容的位：

- 主 USB 外设使能
- 乒乓缓冲器指针复位
- 挂起模式的控制
- 数据包传输禁止

此外，USB控制寄存器还包含状态位SE0（UCON<5>），用于指示总线上出现了单端 0。USB 模块使能时，应监视该位，以确定差分数据线是否出自单端 0 条件。这有助于区分初始上电状态和 USB 复位信号。

USB 模块的总体运行是 USBEN 位 （UCON<3>）控制的。将该位置 1 将激活模块，并将缓冲器描述符表中的所有 PPBI 位复位为 0。该位还将激活片上稳压器 （如果已使能）。因此，该位可用作 USB 的软连接 / 断开控制。尽管清零该位时，所有状态和控制位都将被忽略，但是还是需要在设置该位前预先配置好 USB 模块。

寄存器 14-1： UCON：USB 控制寄存器

U-0	R/W-0	R-x	R/C-0	R/W-0	R/W-0	R/W-0	U-0
—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用**：读为 0

bit 6 **PPBRST**：乒乓缓冲器复位位
1 = 将所有乒乓缓冲器指针复位到偶数缓冲器描述符 （BD）存储区
0 = 乒乓缓冲器指针不复位

bit 5 **SE0**：有效单端 0 标志位
1 = 在 USB 总线上存在有效的单端 0
0 = 没有检测到单端 0

bit 4 **PKTDIS**：数据包传输禁止位
1 = 禁止处理 SIE 令牌和数据包，收到 SETUP 令牌时自动置 1
0 = 使能 SIE 令牌和数据包处理

bit 3 **USBEN**：USB 模块使能位
1 = 使能 USB 模块和支持电路 （器件已连接）
0 = 禁止 USB 模块和支持电路 （器件已断开连接）

bit 2 **RESUME**：Resume 信号使能位
1 = 激活 Resume 信号
0 = 禁止 Resume 信号

bit 1 **SUSPND**：挂起 USB 位
1 = USB 模块和支持电路处于节能模式， SIE 时钟无效
0 = USB 模块和支持电路处于正常运行模式， SIE 时钟按配置的速率运行

bit 0 **未用**：读为 0

© 2007 Microchip Technology Inc.

PPBRST 位（UCON<6>）控制使用双重缓冲模式（乒乓缓冲器）时的复位状态。PPBRST 位置 1 时，所有乒乓缓冲器指针都指向偶数缓冲器。PPBRST 必须由固件清零。该位在不使用乒乓缓冲模式时被忽略。

PKTDIS 位 （UCON<4>）是指示 SIE 已禁止数据包发送和接收的标志。该位在收到允许设置处理的SETUP令牌时由 SIE 置 1。单片机无法将该位置 1，只能清零；将它清零可使 SIE 继续发送和 / 或接收。缓冲器描述符表中的任何挂起事件仍将可用，由 USTAT 寄存器的 FIFO 缓冲器指示。

RESUME位（UCON<2>）允许外设通过执行Resume信号执行远程唤醒。为了产生有效的远程唤醒，固件必须将 RESUME 置 1 并保持 10 ms，然后将该位清零。关于 Resume 信号的更多信息，请参见 USB 2.0规范中第 7.1.7.5 节、第 11.4.4 节 和第 11.9 节。

SUSPND 位 （UCON<1>）将模块和支持电路 （即稳压器）置于低功耗模式。 SIE 的输入时钟也将被禁止。在响应 IDLEIF 中断时，应用软件将其置 1 ；在发生 ACTVIF 中断后由单片机固件复位。该位有效时，器件保持与总线的连接，但是收发器输出保持空闲。VUSB 引脚上的电压可能随该位值的不同而变化。在 IDLEIF 请求之前将该位置 1 将导致不可预测的总线行为。

注： 在挂起模式下，典型总线供电的器件的电流上限是 500 µA。这是 PIC 器件及其支持电路消耗的全部电流。器件进入挂起模式时，要小心确保消耗电流最低。

14.2.2 USB 配置寄存器 （UCFG）

通过 USB 进行通信前，必须配置该模块的相关内部和 / 或外部硬件。多数配置都是通过 UCFG 寄存器 （寄存器 14-2）执行的。独立的 USB 稳压器 （见**第 14.2.2.7 节 “内部稳压器”**）通过配置寄存器控制。

UFCG 寄存器包含大多数控制 USB 模块系统级行为的控制位。这些操作包括：

- 总线速度 （全速或低速）
- 片上收发器使能
- 乒乓缓冲器使用

UCFG 寄存器还包含辅助模块测试、调试和 USB 认证的两个位。这些位控制输出使能状态监视和眼图生成。

注： 只应在模块设置阶段配置 USB 速度、收发器和上拉电路。建议不要在使能模块时改变这些设置。

14.2.2.1 内部收发器

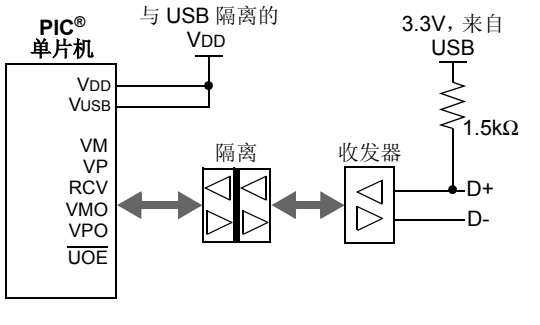
USB 外设具有内置的、USB 2.0 全速和低速模式兼容的收发器，从内部连接到 SIE。该功能部件对于低成本单芯片应用很有用。 UTRDIS 位 （UCFG<3>）控制收发器，默认情况下它是被使能的（UTRDIS = 0）。FSEN 位 （UCFG<2>）控制收发器速度，将该位置 1 将使能全速运行。

USB 规范要求使用 3.3V 电压进行通信；但是，芯片的其余部分可能需要在更高的电压下运行。因此，收发器由独立电源 VUSB 供电。

14.2.2.2 外部收发器

该模块支持外部收发器。外部收发器用于实际条件决定了收发器必须远离 SIE 的应用中。例如，要求与 USB 隔离的应用可通过使用某些隔离元件将外部收发器与单片机的 SIE 相连 （图 14-2）。通过将 UTRDIS 位置 1 使能外部收发器操作。

图 14-2： 带隔离的典型外部收发器



注： 以上是全速配置的简化示意图，使用的是带隔离的外部收发器。

寄存器 14-2： UCFG：USB 配置寄存器

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UTEYE	UOEMON(1)	—	UPUEN (2,3)	UTRDIS(2)	FSEN(2)	PPB1	PPB0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **UTEYE**：USB 眼图测试使能位
1 = 使能眼图测试
0 = 禁止眼图测试

bit 6 **UOEMON**：USB $\overline{OE}$ 监视器使能位 (1)
1 = $\overline{UOE}$ 信号有效；它指示驱动 D+/D- 线的间隔
0 = $\overline{UOE}$ 信号无效

bit 5 **未用**：读为 0

bit 4 **UPUEN**：USB 片上上拉使能位 (2,3)
1 = 片上上拉使能 （FSEN = 1 时， D+ 上拉或 FSEN = 0 时， D- 上拉）
0 = 片上上拉禁止

bit 3 **UTRDIS**：片上收发器禁止位 (2)
1 = 片上收发器禁止；数字收发器接口使能
0 = 片上收发器有效

bit 2 **FSEN**：全速使能位 (2)
1 = 全速器件：控制收发器边沿速率；要求输入时钟运行在 48 MHz
0 = 低速器件：控制收发器边沿速率；要求输入时钟运行在 6 MHz

bit 1-0 **PPB1:PPB0**：乒乓缓冲器配置位
11 = 使能除端点 0 以外的所有端点
10 = 使能所有端点的奇 / 偶乒乓缓冲器
01 = 使能所有 OUT 端点 0 的奇 / 偶乒乓缓冲器
00 = 禁止奇 / 偶乒乓缓冲器

注 1： 如果 UTRDIS 被置 1，无论 UOEMON 位设置如何， $\overline{UOE}$ 信号都将有效。
2： UPUEN、 UTRDIS 和 FSEN 位在 USB 模块使能期间不要更改。必须在使能模块前预先配置这些值。
3： 该位仅在片上收发器有效 （UTRDIS = 0）时有效；否则将被忽略。

模块有 6 条信号线用于与外部收发器通信，并控制其操作：

- VM：单端 D- 线输入
- VP：单端 D+ 线输入
- RCV：差分收发器输入
- VMO：差分线路驱动器输出
- VPO：差分线路驱动器输出
- $\overline{UOE}$：输出使能

VPO 和 VMO 信号是从 SIE 到外部收发器的输出。RCV 信号是从外部收发器到 SIE 的输出；它代表来自串行总线并转变为单脉冲列的差分信号。 VM 和 VP 信号用于将无法用 RCV 信号捕捉的串行总线情况报告给 SIE。这些信号的状态组合及其解释列在表 14-1 和表 14-2 中。

© 2007 Microchip Technology Inc.

表 14-1：　　到收发器的差分输出

VPO	VMO	总线状态
0	0	单端 0
0	1	差分 0
1	0	差分 1
1	1	非法条件

表 14-2：　　来自收发器的单端输入

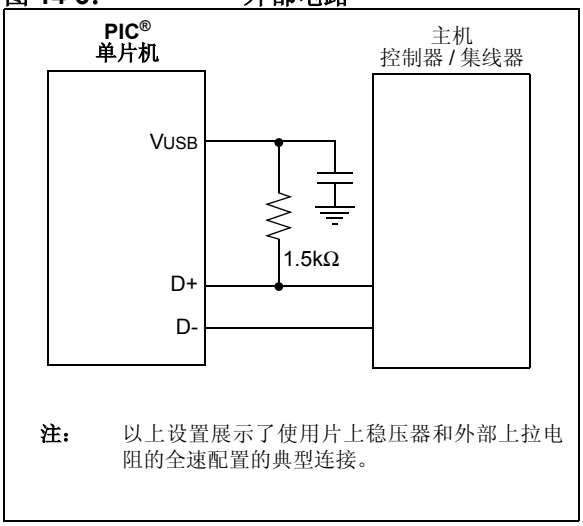
VP	VM	总线状态
0	0	单端 0
0	1	低速
1	0	高速
1	1	出错

$\overline{UOE}$ 信号可使外部收发器的状态发生翻转。该线被器件拉低，以便使能从 SIE 到外部器件的数据传输。

14.2.2.3　上拉电阻

PIC18F2450/4450 器件要求使用外部上拉电阻，以便满足低速和全速 USB 的要求。可用外部 3.3V 电源或 VUSB 引脚上拉 D+ 或 D-。按照 USB 规范的要求，上拉电阻必须是 1.5 kΩ（±5%）。图 14-3 给出了使用 VUSB 引脚的示例。

图 14-3：　　外部电路



注：　以上设置展示了使用片上稳压器和外部上拉电阻的全速配置的典型连接。

14.2.2.4　乒乓缓冲器配置

乒乓缓冲器的使用是通过 PPB1:PPB0 位配置的。请参见**第 14.4.4 节 “乒乓缓冲器”**中关于乒乓缓冲器的完整解释。

14.2.2.5　USB 输出使能监视器

USB $\overline{OE}$ 监视器可以显示 SIE 是在监听总线还是在主动驱动主线。在使用外部收发器或 UCFG<6> = 1 时，默认是使能的。

USB $\overline{OE}$ 监控对于初始系统调试和眼图生成测试期间的示波器触发很有用。

14.2.2.6　使能眼图测试

UCFG<7> 位置 1 时，可通过该模块生成自动的眼图测试。眼图输出的可观察性取决于模块设置，这意味着用户首先要负责配置 SIE 时钟设置、上拉电阻和收发器模式。此外，必须使能该模块。

一旦 UTEYE 被置 1，该模块将仿真从接收切换到发送状态，并开始发送 J-K-J-K 位序列（对于全速是 K-J-K-J）。当使能了眼图测试模式时，该序列将无限重复。

注意，该模块连接到实际 USB 系统时，不应将该位置 1。该测试模式用于帮助进行 USB 认证时的电路板检验。它供系统开发者了解 USB 信号的噪声完整性，USB 信号噪声受电路板布线、阻抗不匹配和邻近其他系统组件的影响。它不能正确测试从接收到发送状态的转换过程。尽管眼图不能代替更复杂的 USB 认证测试，但它在首轮系统调试中会有帮助。

14.2.2.7　内部稳压器

PIC18F2450/4450 器件有内置的 3.3V 稳压器，用来为内部收发器和外部上拉电路供电。为了稳定工作，需要外接一个 220 nF（±20%）电容。

注：　VUSB 除了驱动内部收发器，仅能再驱动一个外部上拉电路。

默认情况下稳压器是使能的，可通过 VREGEN 配置位禁止。使能时，引脚 VUSB 上可观察到电压。禁止稳压器时，必须通过 VUSB 引脚向内部收发器提供 3.3V 电源。如果禁止内部收发器，则不使用 VUSB。

注　1：　如果外部稳压器与 VUSB 相连，则不要使能内部稳压器。

2：　VDD 必须始终大于 VUSB，即使禁止了稳压器。

14.2.3 USB 状态寄存器 （USTAT）

USB 状态寄存器报告 SIE 内的事务状态。SIE 发出 USB 传输完成中断时，应读取 USTAT 以确定传输的状态。USTAT 包含传输端点号、方向和乒乓缓冲器指针值（如果使用了）。

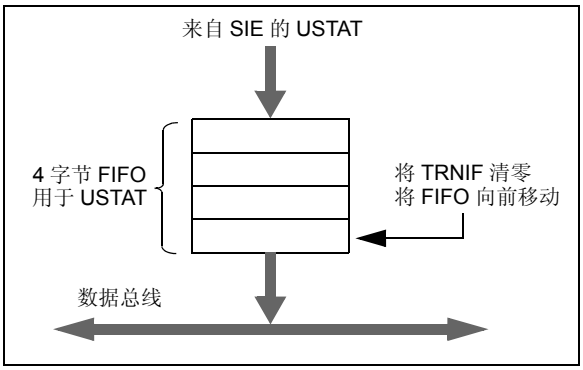
注：	仅当 TRNIF 中断标志被置位时，USB 状态寄存器中的数据才有效。

USTAT 寄存器实际上是由 SIE 维护的四字节状态 FIFO 的读取窗口。它允许单片机在 SIE 处理其他端点时实施一次传输（图 14-4）。当 SIE 使用缓冲器完成读写数据时，它将更新 USTAT 寄存器。如果在一次传输完成中断前进行了另一次 USB 传输，SIE 将把下次传输的状态存储到状态 FIFO 中。

将传输完成标志位 TRNIF 清零将使 SIE 把 FIFO 向前移动。如果 FIFO 保持寄存器中的下个数据有效， SIE 将立即重新将中断标志置位。如果没有其他数据， TRNIF 将保持清零， USTAT 数据将不再可靠。

注：	如果 USTAT FIFO 装满时收到端点请求，SIE 将自动将 NAK 发回主机。

图 14-4： USTAT FIFO



寄存器 14-3： USTAT：USB 状态寄存器

U-0	R-x	R-x	R-x	R-x	R-x	R-x	U-0
—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI(1)	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用：**读为 0

bit 6-3 **ENDP3:ENDP0：**上个端点活动位的编码号 （代表上个 USB 传输更新的 BDT 号）
1111 = 端点 15
1110 = 端点 14
....
0001 = 端点 1
0000 = 端点 0

bit 2 **DIR：**上次 BD 方向指示位
1 = 上次事务是 IN 令牌
0 = 上次事务是 OUT 或 SETUP 令牌

bit 1 **PPBI：**乒乓 BD 指针指示位 (1)
1 = 上次事务在奇数 BD 存储区
0 = 上次事务在偶数 BD 存储区

bit 0 **未用：**读为 0

注 1： 该位仅对具有可用的偶数和奇数 BD 寄存器的端点有效。

© 2007 Microchip Technology Inc.

14.2.4 USB 端点控制

16 个可能的双向端点中的每一个都有各自的独立控制寄存器 UEPn （其中 “n”代表端点号）。每个寄存器内控制位的设置都是相同的，如寄存器 14-4 所示。

EPHSHK 位 （UEPn<4>）控制端点的握手；将该位置 1 将使能 USB 握手。通常该位总是置 1 的，除非使用同步端点。

EPCONDIS 位 （UEPn<3>）用于使能或禁止通过端点进行 USB 控制操作 （SETUP） 。将该位清零将使能 SETUP 事务。注意，对应的 EPINEN 和 EPOUTEN 位必须置 1，才能使能 IN 和 OUT 事务。对于端点 0，该位应总是清零的，因为 USB 规范将端点 0 视为默认的控制端点。

EPOUTEN 位（UEPn<2>）用于从主机使能或禁止 USB OUT 事务。将该位置 1 将使能 OUT 事务。类似地，EPINEN 位 （UEPn<1>）使能或禁止来自主机的 USB IN 事务。

EPSTALL 位 （UEPn<0>）用于指示端点的 STALL 状态。如果在特定端点上发出了 STALL，SIE 将把那个端点对的 EPSTALL 位置 1。EPSTALL 位将保持置 1 状态直到固件将其清零，或 SIE 复位。

寄存器 14-4： UEPn：USB 端点 n 控制寄存器 （UEP0 到 UEP15）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 **未用：**读为 0

bit 4 **EPHSHK：**端点握手使能位
1 = 使能端点握手
0 = 禁止端点握手 （通常用于同步端点）

bit 3 **EPCONDIS：**双向端点控制位
如果 EPOUTEN = 1 并且 EPINEN = 1：
1 = 禁止端点 n 的控制传输；只允许 IN 和 OUT 传输
0 = 使能端点 n 的控制 （SETUP）传输；同时允许 IN 和 OUT 传输

bit 2 **EPOUTEN：**端点输出使能位
1 = 使能端点 n 输出
0 = 禁止端点 n 输出

bit 1 **EPINEN：**端点输入使能位
1 = 使能端点 n 输入
0 = 禁止端点 n 输入

bit 0 **EPSTALL：**端点停止使能位 (1)
1 = 端点 n 已停止
0 = 端点 n 未停止

注 1： 仅当使能端点 n 时有效；否则将忽略该位。

14.2.5 USB 地址寄存器 （UADDR）

USB 地址寄存器包含唯一的 USB 地址，当此地址有效时，可被外设识别。在收到 URSTIF 指示的 “USB 复位”或从单片机收到 “复位”时， UADDR 将复位为 00h。 USB 地址必须在 USB 启动阶段 （枚举）由单片机写入，该操作受 Microchip USB 固件支持。

14.2.6 USB 帧编号寄存器 （UFRMH:UFRML）

帧编号寄存器包含 11 位的帧编号。低位字节保存在 UFRML 中，而高 3 位保存在 UFRMH 中。每当收到 SOF 令牌时，就用当前帧编号更新寄存器对。对于单片机，这些寄存器是只读的。帧编号寄存器主要用于同步传输。

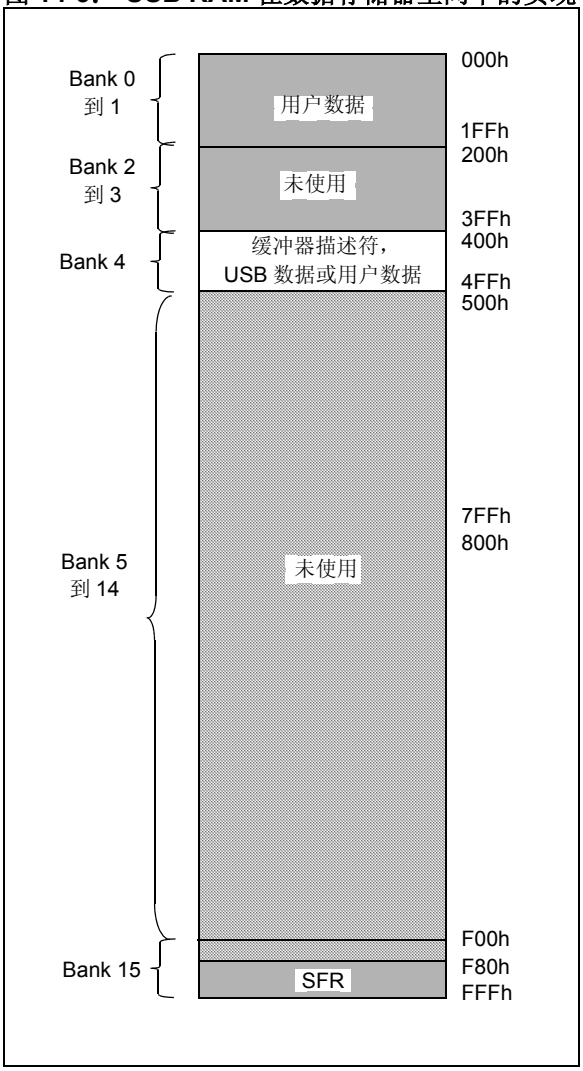
14.3 USB RAM

USB 数据通过称为 USB RAM 的存储器空间在单片机内核与 SIE 之间传送。这是特殊的双端口存储器，被映射到 Bank 4 （400h 到 4FFh）中的普通数据存储空间，总共 256 字节 （图 14-5）。

Bank 4（400h 到 4FFh）的某些部分专门用于端点缓冲控制，而剩余部分可用于 USB 数据。根据使用的缓冲类型，除 8 个字节外， Bank4 的所有其他单元都可用作 USB 缓冲空间。

尽管单片机可使用 USB RAM 作为数据存储器，但是正由 SIE 访问的部分不能被单片机访问。任意给定时间访问哪个特定缓冲器由旗语机制决定。这将在**第 14.4.1.1 节 “缓冲器所有权”**中讨论。

图 14-5： USB RAM 在数据存储器空间中的实现



 © 2007 Microchip Technology Inc.

14.4 缓冲器描述符和缓冲器描述符表

Bank 4 中的寄存器组成称为缓冲器描述符表（BDT）的结构，专门用于端点缓冲控制。它为用户提供了灵活的方法，用于构造和控制不同长度及配置的端点缓冲器。

BDT 由缓冲器描述符 （BD）组成，后者用于定义和控制 USB RAM 空间中的实际缓冲器。每个 BD 依次由四个寄存器组成，其中 n 代表 64 个可能的 BD 中的一个（范围是 0 到 63）：

- BDnSTAT：BD 状态寄存器
- BDnCNT：BD 字节计数寄存器
- BDnADRL：BD 低位地址寄存器
- BDnADRH：BD 高位地址寄存器

BD 在序列中总是以一个四字节的数据块 （BDnSTAT:BDnCNT:BDnADRL:BDnADRH）出现。BDnSTAT 的地址总是一个相对于 400h 的偏移量 4n － 1 （十六进制），其中 n 是缓冲器描述符编号。

根据使用的缓冲配置 （**第 14.4.4 节 “乒乓缓冲器”**），最多可以有 32、 33 或 64 组缓冲器描述符。 BDT 至少要有 8 个字节长。这是因为 USB 规范要求每个器件都必须有输入和输出端点 0，用于初始启动。根据端点和缓冲配置， BDT 可以长达 256 字节。

尽管缓冲器描述符状态和地址寄存器可被视为特殊功能寄存器，但它们和 Bank 15 中的传统单片机 SFR 不同，并不是硬件映射的。如果对应于特定 BD 的端点未被使能，其寄存器将不会被用到。它们将显示为可用的 RAM，而非未实现地址。仅当通过将 UEPn<1> 位置 1 使能了端点时，那些存储器才能用作 BD 寄存器。正如数据存储器空间中的任何单元一样，BD 寄存器在任何器件复位时，包含的都是一个不确定的值。

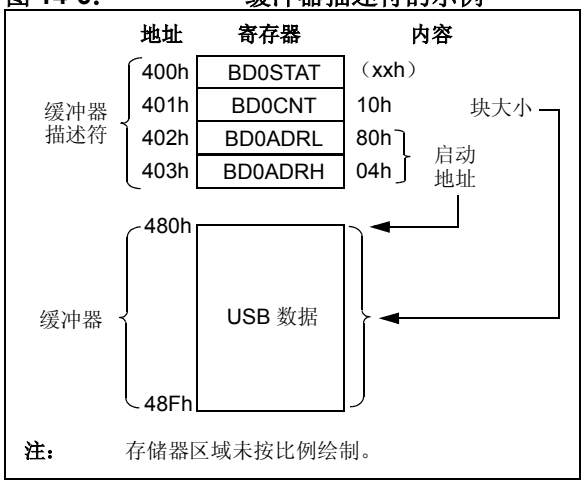
Bank 4 中总共有 256 个字节的地址空间可用于 BDT 和 USB 数据 RAM。在乒乓缓冲器模式中，不可能实现所有 16 个双向端点，因为 BDT 自身就要占用 256 字节。通常在多数 USB 应用中，仅有少数几个端点需要实现。因此，256 个字节的一小部分用于 BDT，而剩余部分用于 USB 数据。

图 14-6 所示为用于 16 字节缓冲的 BD（从 480h 开始）的示例。特定的BD寄存器组仅当相应端点已使用UEPn寄存器使能后才有效。所有 BD 寄存器都在 USB RAM 中可用。每个端点的 BD 都应在使能该端点前设置。

14.4.1 BD 状态和配置

缓冲器描述符不仅定义端点缓冲器的大小，还确定其配置和控制。多数配置是通过 BD 状态寄存器 BDnSTAT 完成的。每个 BD 都有其唯一的、对应编号的 BDnSTAT 寄存器。

图 14-6： 缓冲器描述符的示例



注： 存储器区域未按比例绘制。

和其他控制寄存器不同， BDnSTAT 寄存器的位的设置与具体应用息息相关。有两种不同的配置，取决于在某个特定时间修改 BD 和缓冲器的是单片机还是 USB 模块。只有 3 个位的定义是两者共享的。

14.4.1.1 缓冲器所有权

由于缓冲器及其BD是在CPU和USB模块之间共享的，因此采用了简单的旗语机制来区分允许哪个模块更新存储器中的 BD 和相关缓冲器。

这是通过使用 UOWN 位 （BDnSTAT<7>）作为旗语来实现的。UOWN 是这两个配置在 BDnSTAT 中唯一共享的位。

UOWN 清零时，BD 归单片机内核“所有”。UOWN 位置 1 时，BD 和缓冲存储器归 USB 外设“所有”，此时内核不应修改 BD 或其对应的数据缓冲器。注意， SIE 拥有缓冲器时，单片机内核仍可以读取 BDnSTAT，反之亦然。

根据寄存器的更新源的不同，缓冲器描述符具有不同的意义。在分配给 USB 外设所有权之前，用户可以通过 BDnSTAT 位配置外设的基本操作。此时也可设置字节计数和缓冲位置寄存器。

UOWN 置 1 时，用户不能再信赖写入 BD 的值。从此时起，SIE 根据需要更新 BD，覆盖原始 BD 值。BDnSTAT 寄存器由 SIE 用令牌 PID 更新，传输计数 BDnCNT 也被更新。

在准备设置端点时，最后更新的字节应该总是 BDT 的 BDnSTAT 字节。SIE 在事务完成时将把 UOWN 位清零。唯一的例外是在使能了 KEN 和 / 或 BSTALL 时。

UOWN 位置 1 时，没有任何硬件机制来阻止单片机对存储器的访问。因此，如果单片机试图在 SIE 拥有存储器时修改该存储器，可能会出现无法预料的结果。同样，在 USB 外设将所有权归还单片机之前，读取该存储器可能会导致数据出错。

14.4.1.2 BDnSTAT 寄存器 （CPU 模式）

当 UOWN = 0 时，单片机内核拥有 BD。此时，寄存器的其他 7 位担负控制功能。

数据 Toggle 同步使能位 DTSEN（BDnSTAT<3>）控制数据传输的奇偶检验。将 DTSEN 置 1 将使能 SIE 数据 Toggle 同步。使能时，会将数据包的奇偶校验位与 DTS 的值（BDnSTAT<6>）进行比对。如果数据包到达时同步不正确，数据基本上将被忽略。它不会被写入 USB RAM， USB 传输完成中断标志也不会被置 1。但是，SIE 将把 ACK 令牌发送回主机，对接收作出应答。表 14-3 总结了 DTSEN 位对 SIE 的影响

缓冲器停止位 BSTALL（BDnSTAT<2>）支持控制传输，通常在端点 0 停止一次。它还支持 USB 规范第九章中指定的 SET_FEATURE/CLEAR_FEATURE 命令；通常在除了默认控制端点外的其他端点上连续停止。

BSTALL 位使能缓冲器停止。将 BSTALL 置 1， SIE 会把 STALL 令牌归还主机 （如果收到的令牌使用该地址的 BD）。将 STALL 发送到主机时，对应的 UEPn 控制寄存器中的 EPSTALL 位置 1 并产生 STALL 中断。收到 SETUP 令牌前，UOWN 位将保持置 1 且 BD 不会更改。在这种情况下，STALL 状态将被清零，BD 的所有权将归还给单片机内核。

BD9:BD8 位 （BDnSTAT<1:0>）存储 SIE 字节计数的高 2 位；低 8 位将存储在对应的 BDnCNT 寄存器中。更多信息，请参见**第 14.4.2 节 “BD 字节计数”**。

表 14-3： DTSEN 位对奇 / 偶 （DATA0/DATA1）数据包接收的影响

来自主机的 OUT 数据包	BDnSTAT 设置		收到数据包后的器件响应			
	DTSEN	DTS	握手	UOWN	TRNIF	BDnSTAT 和 USTAT 状态
DATA0	1	0	ACK	0	1	已更新
DATA1	1	0	ACK	1	0	未更新
DATA0	1	1	ACK	0	1	已更新
DATA1	1	1	ACK	1	0	未更新
两者中任一	0	x	ACK	0	1	已更新
任一，有错误	x	x	NAK	1	0	未更新

图注： x = 无关位

© 2007 Microchip Technology Inc.

寄存器 14-5：　BDnSTAT：缓冲器描述符 n 状态寄存器 （BD0STAT 到 BD63STAT）， CPU 模式（数据写入 CPU 控制的存储区）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
UOWN(1)	DTS(2)	—(3)	—(3)	DTSEN	BSTALL	BC9	BC8
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　**UOWN**：USB 所有权位 (1)
0 = 单片机内核拥有 BD 及其对应缓冲器

bit 6　**DTS**：数据 Toggle 同步位 (2)
1 = 数据 1 数据包
0 = 数据 0 数据包

bit 5-4　**保留**：这些位应总是设定为 0(3)

bit 3　**DTSEN**：数据 Toggle 同步使能位
1 = 数据 Toggle 同步已使能；同步值不正确的数据包将被忽略
0 = 不会执行数据 Toggle 同步

bit 2　**BSTALL**：缓冲器停止使能位
1 = 使能缓冲器停止；如果收到使用指定地址 BD 的令牌，将发出停止握手（UOWN 位保持置 1，BD 值不变）
0 = 禁止缓冲器停止

bit 1-0　**BC9:BC8**：字节计数的 bit9 和 bit8
该字节计数位代表将为某 IN 令牌发送或 OUT 令牌期间接收的字节数。和 BC<7:0> 一起，有效字节数是 0-1023。

注　1：必须在使能 USB 模块之前，由用户将该位初始化为所需值。
2：除非 DTSEN = 1，否则该位将被忽略。
3：如果这些位被置 1， USB 可能无法进行通信。因此，这些位应该始终保持为 0。

14.4.1.3 BDnSTAT 寄存器 （SIE 模式）

BD 及其缓冲器由 SIE 所有时，BDnSTAT 中的大多数位都有了不同的含义。配置如寄存器 14-6 所示。一旦 UOWN 被置 1，先前用户写入的任何数据或控制设置都将被来自 SIE 的数据覆盖。

SIE 使用令牌包标识符（PID）更新 BDnSTAT 寄存器，该 PID 存储在 BDnSTAT<5:3> 中 。对应 BDnCNT 寄存器中的传输计数值也会更新。该 8 位寄存器的值溢出得到计数值的高两位，存储在 BDnSTAT<1:0> 中。

14.4.2 BD 字节计数

字节计数代表一次 IN 传输期间将发送的字节总数。 IN 传输完成后， SIE 将返回发送到主机的字节数。

对于 OUT 传输，字节计数代表可接收并存储在 USB RAM 中的最大字节数。OUT 传输完成后， SIE 将返回接收到的实际字节数。如果接收到的字节数超过了设定的字节计数，数据包将被拒绝，并生成一个 NAK 握手信号。在这种情况下，不会更新字节计数。

10 位字节计数保存在两个寄存器中。计数的低 8 位驻留在 BDnCNT 寄存器中，而高 2 位驻留在 BDnSTAT<1:0> 中。这代表字节数的有效范围是 0 到 1023。

14.4.3 BD 地址验证

BD 地址寄存器对包含对应端点缓冲器的起始 RAM 地址。为使端点起始有效，它必须落在 USB RAM 400h 到 7FFh 的范围内。没有可以验证 BD 地址有效性的硬件机制。

如果 BD 地址的值没有指向 USB RAM，或指向其他端点缓冲器中的地址，数据很可能丢失或被覆盖。类似地，如果接收缓冲器（OUT 端点）和使用中的 BD 地址重叠，也会导致无法预料的后果。开发 USB 应用程序时，用户可能需要考虑在代码中加入基于软件的地址验证。

寄存器 14-6： BDnSTAT：缓冲器描述符 n 状态寄存器 （BD0STAT 到 BD63STAT），SIE 模式 （从该侧返回到单片机的数据）

R/W-x	U-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
UOWN	—	PID3	PID2	PID1	PID0	BC9	BC8
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **UOWN**：USB 所有权位
1 = SIE 拥有 BD 及其对应缓冲器

bit 6 **保留**：不是由 SIE 写入的

bit 5-2 **PID3:PID0**：数据包标识符位
上次传输收到的令牌 PID 值 （仅适用于 IN、 OUT 或 SETUP 事务）。

bit 1-0 **BC9:BC8**：字节计数的 bit9 和 bit8
这些位由 SIE 更新，以便反映 OUT 传输中收到的和 IN 传输中发送的实际字节数。

© 2007 Microchip Technology Inc.

14.4.4 乒乓缓冲器

当端点有两组 BD 项时（一组用于偶数数据传输，一组用于奇数数据传输），则该端点被定义为具有乒乓缓冲器。这样就允许在 CPU 处理一组 BD 的同时，SIE 处理另一组 BD。以这种方式实现的双重缓冲 BD 能获得最高的 USB 输入 / 输出流量。

USB 模块支持三种操作模式：

- 不支持乒乓缓冲
- 仅对 OUT 端点 0 提供乒乓缓冲器支持
- 对所有端点提供乒乓缓冲器支持

乒乓缓冲器设置是用 UCFG 寄存器中的 PPB1:PPB0 位配置的。

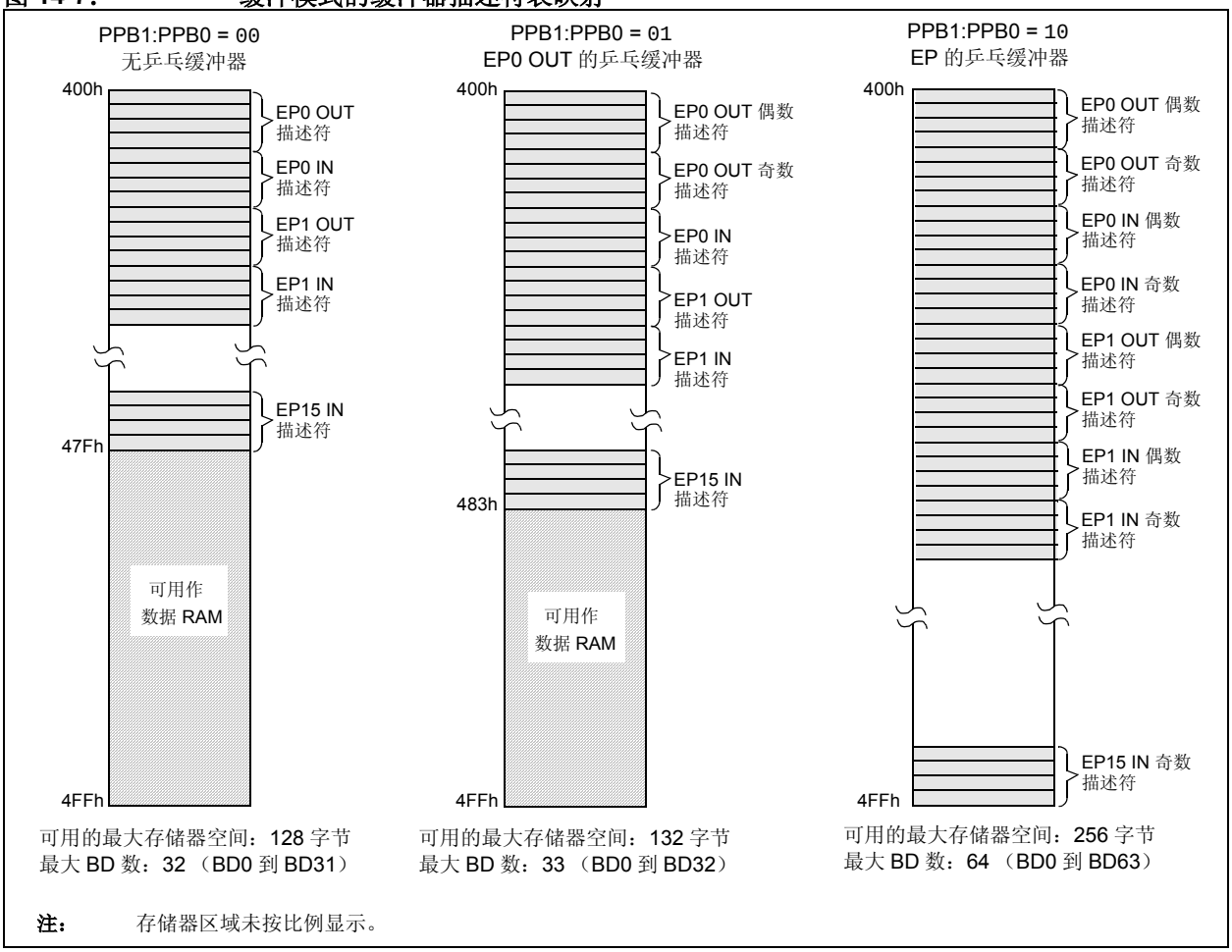
USB 模块跟踪每个端点的乒乓指针。该模块使能时，所有指针初始时都复位到偶数 BD。事务完成后（UOWN 由 SIE 清零），指针转向奇数 BD。下个事务完成后，指针指回偶数 BD，以此类推。

上个事务的奇 / 偶状态存储在 USTAT 寄存器的 PPBI 位中。用户可以用 PPBRST 位将所有乒乓指针复位到偶数 BD。

图 14-7 显示了操作的三种不同模式，以及 BD 是如何填充 USB RAM 的。

根据缓冲配置， BD 和特定端点有着固定的对应关系。BD 到端点的映射在表 14-4 中有详细描述。这一关系也意味着如果端点不是连续使能的，BDT 中可能出现空白区域。已禁止端点的 BD 理论上可用作缓冲空间，但在实际应用中，用户应避免使用 BDT 中的这类空间，除非采取了验证 BD 地址的措施。

图 14-7： 缓冲模式的缓冲器描述符表映射



注： 存储器区域未按比例显示。

表 14-4： 不同缓冲模式的缓冲器描述符分配

端点	分配给端点的 BD					
	模式 0（无乒乓缓冲器）		模式 1（EP0 OUT 的乒乓缓冲器）		模式 2（EP 的乒乓缓冲器）	
	输出	输入	输出	输入	输出	输入
0	0	1	0 (E), 1 (O)	2	0 (E), 1 (O)	2 (E), 3 (O)
1	2	3	3	4	4 (E), 5 (O)	6 (E), 7 (O)
2	4	5	5	6	8 (E), 9 (O)	10 (E), 11 (O)
3	6	7	7	8	12 (E), 13 (O)	14 (E), 15 (O)
4	8	9	9	10	16 (E), 17 (O)	18 (E), 19 (O)
5	10	11	11	12	20 (E), 21 (O)	22 (E), 23 (O)
6	12	13	13	14	24 (E), 25 (O)	26 (E), 27 (O)
7	14	15	15	16	28 (E), 29 (O)	30 (E), 31 (O)
8	16	17	17	18	32 (E), 33 (O)	34 (E), 35 (O)
9	18	19	19	20	36 (E), 37 (O)	38 (E), 39 (O)
10	20	21	21	22	40 (E), 41 (O)	42 (E), 43 (O)
11	22	23	23	24	44 (E), 45 (O)	46 (E), 47 (O)
12	24	25	25	26	48 (E), 49 (O)	50 (E), 51 (O)
13	26	27	27	28	52 (E), 53 (O)	54 (E), 55 (O)
14	28	29	29	30	56 (E), 57 (O)	58 (E), 59 (O)
15	30	31	31	32	60 (E), 61 (O)	62 (E), 63 (O)

图注： (E) = 偶数事务缓冲， (O) = 奇数事务缓冲

表 14-5： USB 缓冲器描述符表寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BDnSTAT(1)	UOWN	DTS(4)	PID3(2)	PID2(2)	PID1(2) DTSEN(3)	PID0(2) BSTALL(3)	BC9	BC8
BDnCNT(1)	字节计数							
BDnADRL(1)	缓冲器地址低字节							
BDnADRH(1)	缓冲器地址高字节							

注 1： 对于缓冲器描述符寄存器， n 是 0 到 63 之间的某个值。为了简明起见，所有 64 个寄存器都显示为一个通用名称代号。所有寄存器都有不确定的复位值 （xxxx xxxx）。

2： BDnSTAT 寄存器的 bit 5 到 bit 2 是在该寄存器受控于 SIE （UOWN 位置 1 时）后，由 SIE 用来返回 PID3:PID0 值的。一旦寄存器处于 SIE 控制下， DTSEN 和 BSTALL 写入的值就不再有效。

3： 将缓冲器描述符所有权移交给 SIE 前 （UOWN 位清零）， BDnSTAT 寄存器的 bit 3 和 bit 2 用于配置 DTSEN 和 BSTALL。

4： 除非 DTSEN = 1，否则该位将被忽略。

© 2007 Microchip Technology Inc.

14.5 USB 中断

USB 模块可生成多个中断条件。为了识别所有这些中断源，该模块配备有自身的中断逻辑结构（和单片机的结构相似）。USB 中断是用一组控制寄存器允许的，并用一组独立的标志寄存器来捕获中断事件。所有中断源在单片机的中断逻辑中都归为一个 USB 中断请求 USBIF（PIR2<5>）。

图 14-8 给出了 USB 模块的中断逻辑。USB 模块的中断寄存器分为两层。顶层全部由 USB 状态中断组成；这些中断分别是在 UIE 及 UIR 寄存器中允许和标记的。第二层由USB错误条件中断组成，分别是在UEIR及UEIE寄存器中允许和标记的。任意一个中断条件都将触发顶层的 USB 错误中断标志 （UERRIF）。

可使用中断来处理 USB 传输中的常规事务。图 14-9 展示了USB帧传输过程中的某些常见事件及其对应中断。

图 14-8： USB 中断逻辑

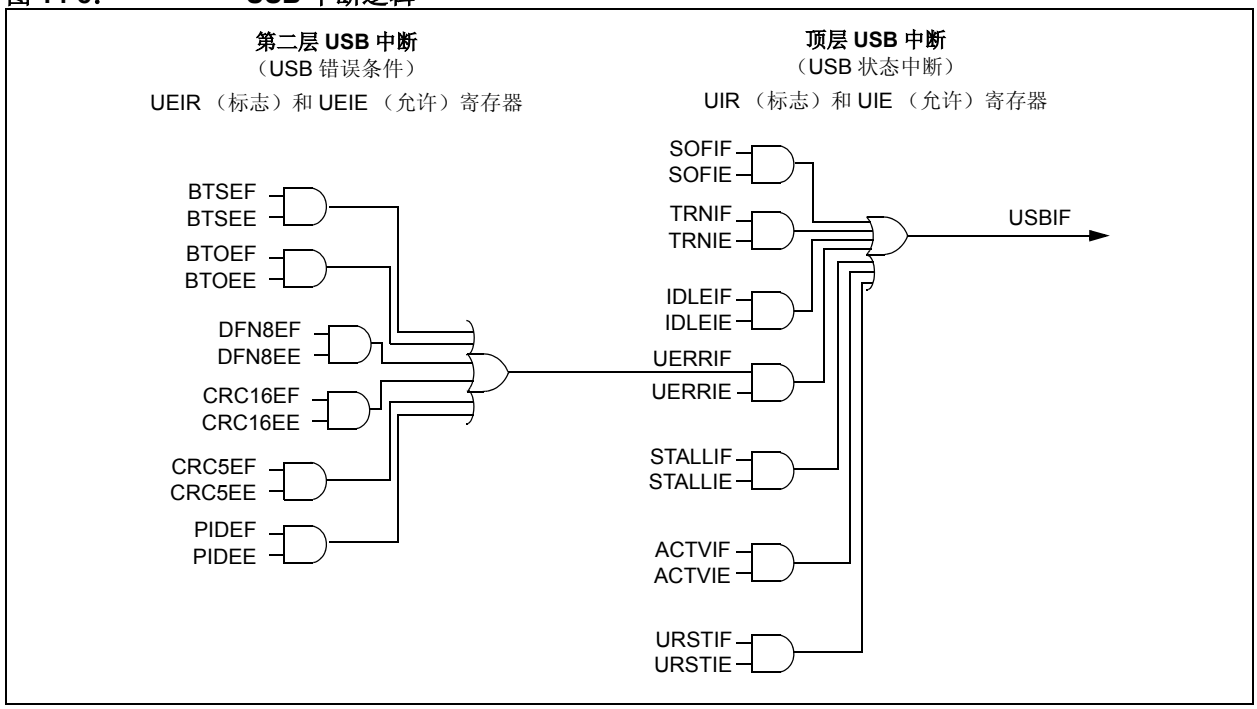
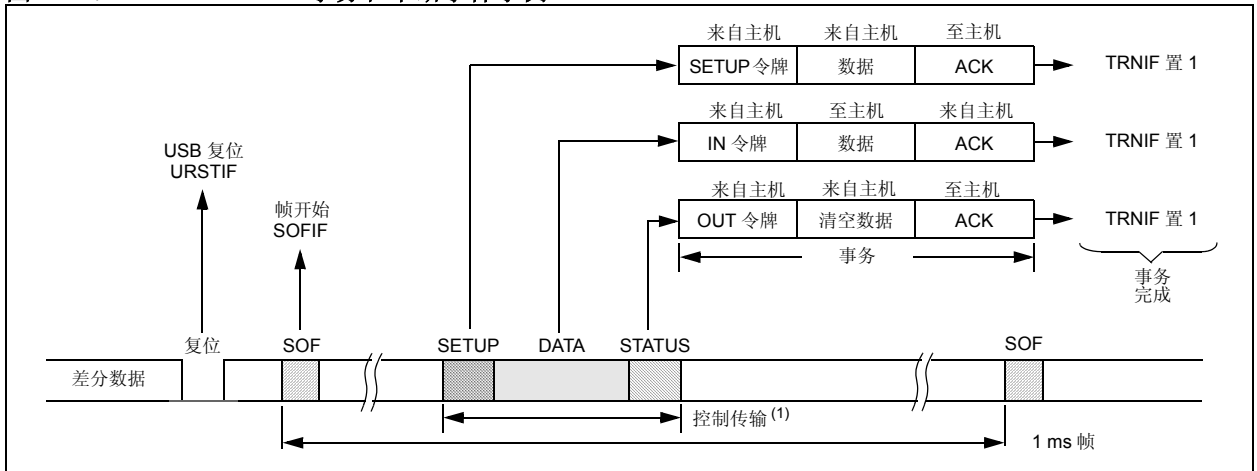


图 14-9： USB 事务和中断事件示例



注 1： 这里展示的控制传输只是每个事务可能发生的事件的一个示例。典型的控制传输会用到多个帧。

14.5.1 USB 中断状态寄存器 （UIR）

USB 中断状态寄存器 （寄存器 14-7）包含每个 USB 状态中断源的标志位。这些中断源中的每一个在 UIE 寄存器中都有相应的中断允许位。所有 USB 状态标志一起作或运算，生成单片机的 USBIF 中断标志。

一旦由 SIE 将中断位置 1 后，必须由软件通过写入 0 来清零。也可以用软件置 1 标志位，这有助于固件调试。

寄存器 14-7： UIR：USB 中断状态寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
—	SOFIF	STALLIF	IDLEIF(1)	TRNIF(2)	ACTVIF(3)	UERRIF(4)	URSTIF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用：**读为 0

bit 6 **SOFIF：**START-OF-FRAME 令牌中断位
1 = SIE 收到 START-OF-FRAME 令牌
0 = SIE 没有收到 START-OF-FRAME 令牌

bit 5 **STALLIF：**STALL 握手中断位
1 = SIE 发出了一个 STALL 握手
0 = 未发出 STALL 握手

bit 4 **IDLEIF：**空闲检测中断位 (1)
1 = 检测到空闲状态 （3 ms 或更长的连续空闲状态）
0 = 未检测到空闲状态

bit 3 **TRNIF：**事务完成中断位 (2)
1 = 待处理事务处理完成；从 USTAT 寄存器读取端点信息
0 = 未处理完待处理事务或没有待处理事务

bit 2 **ACTVIF：**总线活动检测中断位 (3)
1 = D+/D- 线上检测到活动
0 = D+/D- 线上没有检测到活动

bit 1 **UERRIF：**USB 错误条件中断位 (4)
1 = 发生了未屏蔽的错误条件
0 = 未发生未屏蔽的错误条件

bit 0 **URSTIF：**USB 复位中断位
1 = 发生有效的 USB 复位； 00h 载入 UADDR 寄存器
0 = 没有发生 USB 复位

注 1： 一旦检测到空闲状态，用户可能希望将 USB 模块置于挂起模式中。
2： 将该位清零可导致 USTAT FIFO 前移 （仅对 IN、 OUT 和 SETUP 令牌有效）。
3： 该位通常只在检测到 UIDLE 中断事件后是未屏蔽的。
4： 只有通过 UEIE 寄存器使能的错误条件才能将该位置 1。该位只是状态位，用户不能将它置 1 或清零。

© 2007 Microchip Technology Inc.

14.5.2 USB 中断允许寄存器 （UIE）

USB 中断允许寄存器（寄存器 14-8）包含 USB 状态中断源的允许位。将这些位中的任何一个置 1 将允许 UIR 寄存器中的对应中断源。

该寄存器中的值只影响是否将该中断条件传递给单片机中断逻辑。标志位仍是由其中断条件置 1 的，允许对它们进行查询和处理而无需实际生成中断。

寄存器 14-8： UIE：USB 中断允许寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未用：**读为 0

bit 6 **SOFIE：**START-OF-FRAME 令牌中断允许位
1 = 允许 START-OF-FRAME 令牌中断
0 = 禁止 START-OF-FRAME 令牌中断

bit 5 **STALLIE：**STALL 握手中断允许位
1 = 允许 STALL 中断
0 = 禁止 STALL 中断

bit 4 **IDLEIE：**空闲检测中断允许位
1 = 允许空闲检测中断
0 = 禁止空闲检测中断

bit 3 **TRNIE：**事务完成中断允许位
1 = 允许事务中断
0 = 禁止事务中断

bit 2 **ACTVIE：**总线活动检测中断允许位
1 = 允许总线活动检测中断
0 = 禁止总线活动检测中断

bit 1 **UERRIE：**USB 错误中断允许位
1 = 允许 USB 错误中断
0 = 禁止 USB 错误中断

bit 0 **URSTIE：**USB 复位中断允许位
1 = 允许 USB 复位中断
0 = 禁止 USB 复位中断

14.5.3 USB 错误中断状态寄存器 （UEIR）

USB 错误中断状态寄存器（寄存器 14-9）包含 USB 外设中每个错误源的标志位。这些源中的每一个都由 UEIE 寄存器中相应的中断允许位控制。所有 USB 错误标志一起进行或运算，生成中断逻辑顶层的 USB 错误中断标志 （UERRIF）。

一旦检测到错误条件，就会立即将对应的错误标志位置 1。因此，中断通常不会在令牌处理结束时产生。

一旦 SIE 将某中断位置 1，就必须通过软件写入 0 进行清零。

寄存器 14-9： UEIR：USB 错误中断状态寄存器

R/C-0	U-0	U-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
bit 7							bit 0

图注：			
R = 可读位	C = 可清零位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **BTSEF**：位填充错误标志位
1 = 检测到位填充错误
0 = 无位填充错误

bit 6-5 **未用**：读为 0

bit 4 **BTOEF**：总线周转 （Turn-round）超时标志位
1 = 发生总线周转超时 （在上个 EOP 之后，已经过了 16 比特以上的空闲时间）
0 = 未发生总线周转超时

bit 3 **DFN8EF**：数据字段大小错误标志位
1 = 数据字段的字节数不是整数
0 = 数据字段的字节数是整数

bit 2 **CRC16EF**：CRC16 失败标志位
1 = CRC16 未通过
0 = CRC16 已通过

bit 1 **CRC5EF**：CRC5 主机错误标志位
1 = 令牌包由于 CRC5 错误而被拒绝
0 = 令牌包被接受

bit 0 **PIDEF**：PID 检查失败标志位
1 = PID 检查未通过
0 = PID 检查已通过

© 2007 Microchip Technology Inc.

14.5.4 USB 错误中断允许寄存器 （UEIE）

USB错误中断允许寄存器（寄存器14-10）包含每个USB错误中断源的允许位。将这些位中的任何一个置 1，都将允许把UEIR寄存器中的对应错误中断源传递到中断逻辑顶层的 UERR 位中。

同 UIE 寄存器一样，该允许位只影响是否将中断条件传递给单片机中断逻辑。标志位仍是由其中断条件置 1 的，允许对它们进行查询和处理而无需实际生成中断。

寄存器 14-10： UEIE：USB 错误中断允许寄存器

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **BTSEE**：位填充错误中断允许位
1 = 允许位填充错误中断
0 = 禁止位填充错误中断

bit 6-5 **未用**：读为 0

bit 4 **BTOEE**：总线周转超时错误中断允许位
1 = 允许总线周转超时错误中断
0 = 禁止总线周转超时错误中断

bit 3 **DFN8EE**：数据字段大小错误中断允许位
1 = 允许数据字段大小错误中断
0 = 禁止数据字段大小错误中断

bit 2 **CRC16EE**：CRC16 失败中断允许位
1 = 允许 CRC16 失败中断
0 = 禁止 CRC16 失败中断

bit 1 **CRC5EE**：CRC5 主机错误中断允许位
1 = 允许 CRC5 主机错误中断
0 = 禁止 CRC5 主机错误中断

bit 0 **PIDEE**：PID 检查失败中断允许位
1 = 允许 PID 检查失败中断
0 = 禁止 PID 检查失败中断

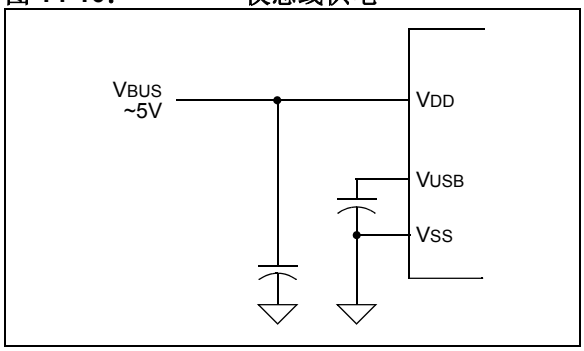
14.6 USB 电源模式

许多USB应用都可能有多组不同的电源要求和配置。最常见的电源模式是仅总线供电（Bus Power Only）、仅自供电（Self-Power Only）和以自供电为主的双电源供电 （Dual Power）模式。这里列出了最常见的情况。

14.6.1 仅总线供电

在仅总线供电模式中，应用的所有电源都来自 USB（图 14-10）。这是对设备有效供电的最简单方式。

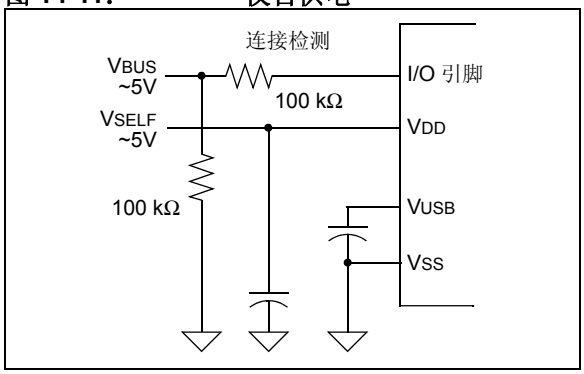
图 14-10： 仅总线供电



14.6.2 仅自供电

在仅自供电模式中，USB 应用为其自身提供电源，仅有很少一部分电源来自 USB，如图 14-11 所示。注意，该示例加上了一个连接指示器，用于显示何时连接上 USB。

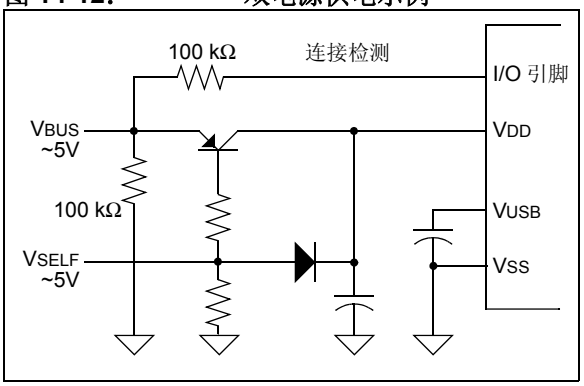
图 14-11： 仅自供电



14.6.3 以自供电为主的双电源供电

某些应用可能需要两个供电选项。在该供电模式下，应用主要使用内部电源，但无内部电源可用时，可切换到 USB 供电。图 14-12 所示为一个该供电模式的示例，它可在仅自供电模式和仅 USB 总线供电模式之间自动切换。

图 14-12： 双电源供电示例



注：	用户应牢记由 USB 总线供电的电流限制。根据 USB 规范 2.0，对单个低能耗设备，不能超过100 mA；对单个高能耗设备，不能超过 500 mA。

14.7 振荡器

USB 模块有特定的时钟要求。对于全速工作模式，时钟源必须是 48 MHz。虽然如此，单片机内核和其他外设不一定要以该时钟频率运行，甚至不一定要使用同一个时钟源。可用的时钟选项在**第 2.3 节“USB 的振荡器设置”**中有详细描述。

14.8 USB 固件和驱动程序

Microchip 提供一系列针对特定应用的资源，例如 USB 固件和驱动程序。请访问 www.microchip.com 获得最新的固件和驱动程序。

© 2007 Microchip Technology Inc.

表 14-6：　　与 USB 模块运行相关的寄存器 (1)

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	详情请见（页）
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
IPR2	OSCFIP	—	USBIP	—	—	HLVDIP	—	—	51
PIR2	OSCFIF	—	USBIF	—	—	HLVDIF	—	—	51
PIE2	OSCFIE	—	USBIE	—	—	HLVDIE	—	—	51
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	52
UCFG	UTEYE	UOEMON	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0	52
USTAT	—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	—	52
UADDR	—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	52
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	52
UFRMH	—	—	—	—	—	FRM10	FRM9	FRM8	52
UIR	—	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	52
UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	52
UEIR	BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	52
UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	52
UEP0	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP1	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP2	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP3	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP4	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP5	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP6	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP7	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP8	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	52
UEP9	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	51
UEP10	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	51
UEP11	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	51
UEP12	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	51
UEP13	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	51
UEP14	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	51
UEP15	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	51

图注：　– = 未用，读为 0。 USB 模块不使用阴影单元。

注　1：　该表只包含在数据存储器空间 Bank 15 中硬件映射的 SFR。缓冲器描述符寄存器映射到 Bank 4，它们不是真正的 SFR，另外列在表 14-5 中。

14.9 USB 概述

本节介绍 USB 的一些基本概念和设计 USB 设备所需的有用信息。尽管本节涵盖了很多内容，但 USB 规范和分类规范中提供了更为详尽的信息。因此，建议读者参考 USB 规范了解更多信息（www.usb.org）。如果您对 USB 非常熟悉，阅读本节可以帮助您快速重温 USB 的相关知识。

14.9.1 分层框架

USB 设备的功能是以分层架构的形式实现的，如图 14-13 所示。每一层都和设备的某个功能层次关联。最高层是配置层。一个设备可有多个配置，例如，在仅自供电和仅总线供电模式的基础上，某个特定设备可能有多种供电要求。

对每种配置，可能有多个接口。每个接口支持该配置的某个特定模式。

接口之下是端点。数据直接在该层传送。最多可有 16 个双向端点。端点 0 总是控制端点，并且当设备在总线上时，默认情况下端点 0 必须可用于配置设备。

14.9.2 帧

在总线上通信的信息通常被划分为多个 1 ms 长的时隙，称为“帧”。每个帧可包含不同设备和端点的许多事务。图 14-9 展示了帧内一个事务的示例。

14.9.3 传输

USB 规范中定义了四种传输类型。

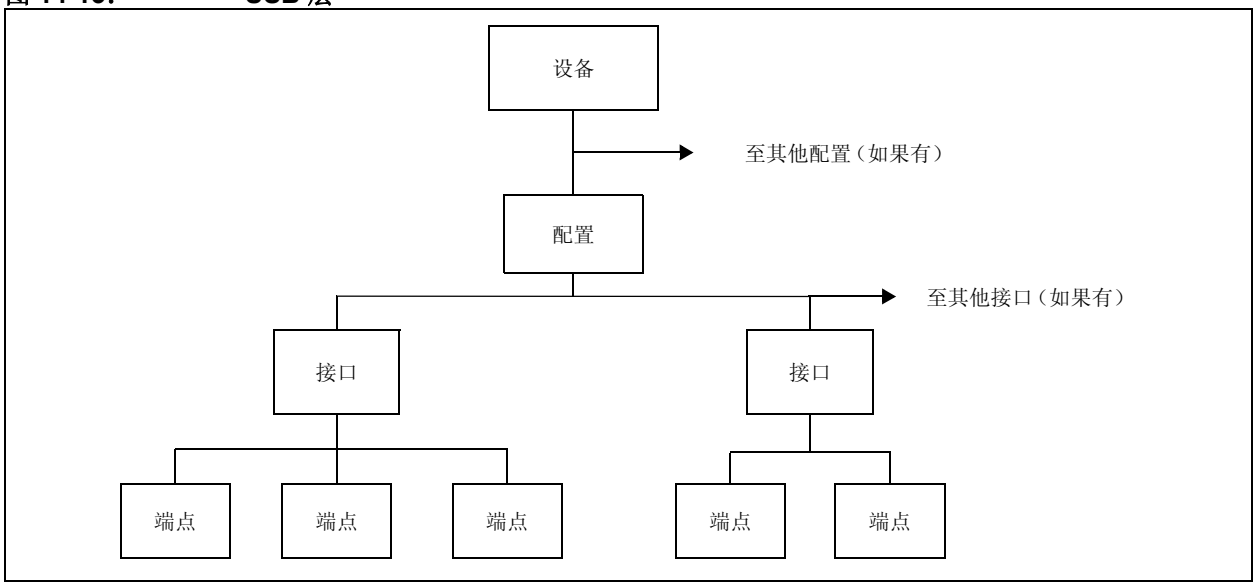
- **同步：**该类型可确保及时传递大量数据 （最多可达 1023 字节），但不能保证数据的完整性。这适用于少量数据丢失影响不大的数据流应用，例如音频。
- **批量：**这类传输方式允许传输大量数据并确保数据完整性；但不能保证传输时限。
- **中断：**这类传输可确保小块数据的及时传递；并能保证数据的完整性。
- **控制：**这类传输是为设备启动控制提供的。

全速设备支持所有传输类型，而低速设备只限于中断和控制传输。

14.9.4 电源

电源可从 USB 获得。USB 规范定义了总线供电的要求。设备既可以使用自身电源，也可以通过总线供电。自供电设备从外部电源获得电能，而总线供电设备使用总线提供的电源。

图 14-13： USB 层



 © 2007 Microchip Technology Inc.

USB 规范对取自总线的电能做出了限制。在电压大约为 5V 的条件下，每个设备确保能获得 100 mA 的电流（一个单位负载）。若需要更多的电量，最多可达 500 mA。注意，可以请求供给超过一个单位负载的电源功率，主机或集线器不一定会提供额外电流。因此，耗电量可能超过单位负载的设备必要时必须能够置于低功耗配置（单位负载以下）中。

USB 规范还定义了挂起模式。在这种情况下，1 秒内的平均电流不能超过 500 μA。总线上没有任何活动 3 ms（即 3 ms 内没有 SOF 令牌）后，器件必须进入挂起状态。进入挂起模式的设备必须在挂起后 10 ms 内降低电流消耗。类似地，在唤醒时，设备必须在 10 ms 内拉高电流超出挂起门限值以示唤醒。

14.9.5 枚举

设备第一次连接到总线时，主机进入枚举进程，试图识别设备。基本上，主机会询问设备，收集类似功耗、数据传输速率和大小、协议及其他描述信息；描述符将包含这些信息。下面是一个典型的枚举进程：

1. USB 复位：将设备复位。因此，设备未配置，也没有地址（地址 0）。
2. 获取设备描述符：主机请求设备描述符的一小部分。
3. USB 复位：再次将设备复位。
4. 设置地址：主机将地址分配给设备。
5. 获取设备描述符：主机检索设备描述符，收集类似制造商、设备类型、最大控制包大小等信息。
6. 获取配置描述符。
7. 获取任何其他描述符。
8. 设置配置。

具体的枚举进程取决于主机。

14.9.6 描述符

有 8 种不同的标准描述符类型，其中 5 种对设备来讲非常重要。

14.9.6.1 设备描述符

设备描述符提供常规信息，例如制造商、产品编号、序列号、设备类别和配置数。只有一个设备描述符。

14.9.6.2 配置描述符

配置描述符提供设备电源要求以及在该配置下支持多少不同接口的信息。一个设备可能有多个配置（即低功耗和高功耗配置）。

14.9.6.3 接口描述符

接口描述符详细描述该接口中使用的端点数，以及接口类型。一个配置可能有多个接口。

14.9.6.4 端点描述符

端点描述符确定传输类型（**第 14.9.3 节 “传输”**）和方向，以及端点的一些其他定义。一个设备中可以有许多端点，端点可在不同配置间共享。

14.9.6.5 字符串描述符

许多前面的描述符都引用了一个或多个字符串描述符。字符串描述符提供关于它们描述的层（**第 14.9.1 节 “分层框架”**）的可读信息。通常，这些字符串出现在主机中，帮助用户识别设备。为了节省存储器空间，字符串描述符通常是可选的，采用 unicode 格式编码。

14.9.7 总线速度

每个 USB 设备都必须告知主机其总线状况和速度。在连接时这是通过连接到总线的 1.5 kΩ 电阻实现的。

取决于设备速度，电阻可将 D+ 或 D- 线上拉到 3.3V。对于低速设备，上拉电阻连接到 D- 线。对于全速设备，上拉电阻连接到 D+ 线。

14.9.8 分类规范和驱动程序

USB 规范包含操作系统供应商可选择支持的分类规范。类别包括诸如音频、大容量存储、通信和人机界面（Human Interface， HID）。多数情况下，主机端需要驱动程序来与 USB 设备 “对话”。在定制应用中，可能需要开发驱动程序。幸运的是，对于绝大多数常见类型的设备来说，常见主机系统中均配有驱动程序。因此，这些驱动程序可以重复使用。

注：

 © 2007 Microchip Technology Inc.

15.0 增强型通用同步 / 异步收发器（EUSART）

通用同步 / 异步收发器（Universal Synchronous Asynchronous Receiver Transmitter，USART）模块是两个串行 I/O 模块之一。（USART 也称为“串行通信接口”或 SCI。）可以将 USART 配置为能与 CRT 终端和个人计算机等外设通信的全双工异步系统，也可以将它配置成能够与 A/D 或 D/A 集成电路、串行 EEPROM 等外设通信的半双工同步系统。

增强型通用同步 / 异步收发器（Enhanced Universal Synchronous Asynchronous Receiver Transmitter，EUSART）模块还实现了其他功能，包括自动波特率检测（Automatic Baud Rate Detection，ABD）和校准、接收到同步间隔字符时的自动唤醒和 12 位间隔字符发送。因为具有这些功能，所以用在局域互连网络（Local Interconnect Network，LIN）总线系统非常理想。

EUSART 可配置为以下几种工作模式：

- 异步模式（全双工）：
  - 接收到字符时自动唤醒
  - 自动波特率校准
  - 12 位间隔字符发送
- 同步——主控（半双工）模式，时钟极性可选
- 同步——从动（半双工）模式，时钟极性可选

增强型USART的引脚与PORTC的功能复用。要把RC6/TX/CK 和 RC7/RX/DT 引脚配置为 EUSART：

- SPEN（RCSTA<7>）位必须置 1（= 1）
- TRISC<7> 位必须置 1（= 1）
- TRISC<6> 位必须清零（= 0）（异步和同步主控模式）或置 1（= 1）（同步从动模式）

注： EUSART 控制在需要时会自动将引脚从输入重新配置为输出。

增强型 USART 模块的操作是由以下 3 个寄存器控制的：

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

这些寄存器将在寄存器 15-1、寄存器 15-2 和寄存器 15-3 中分别详细介绍。

寄存器 15-1： TXSTA：发送状态和控制寄存器

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

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **CSRC**：时钟源选择位
异步模式：
无关位。
同步模式：
1 = 主控模式 （时钟来自内部 BRG）
0 = 从动模式 （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 **SENDB**：发送间隔字符位
异步模式：
1 = 在下一次发送时发送 “同步间隔”字符 （在完成时由硬件清零）
0 = “同步间隔”字符发送完成
同步模式：
无关位。

bit 2 **BRGH**：高波特率选择位
异步模式：
1 = 高速
0 = 低速
同步模式：
在此模式下未使用。

bit 1 **TRMT**：发送移位寄存器状态位
1 = TSR 空
0 = TSR 满

bit 0 **TX9D**：发送数据的第 9 位
可以是地址 / 数据位或奇偶校验位。

注 1： 在同步模式下， SREN/CREN 的优先级高于 TXEN，例外情况是在同步从动模式下 SREN 不起作用。

© 2007 Microchip Technology Inc.

寄存器 15-2： RCSTA：接收状态和控制寄存器

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

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

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**：单字节接收使能位
异步模式：
无关位。
同步主控模式：
1 = 使能单字节接收
0 = 禁止单字节接收
此位在接收完成后清零。
同步从动模式：
无关位。

bit 4 **CREN**：连续接收使能位
异步模式：
1 = 使能接收器
0 = 禁止接收器
同步模式：
1 = 使能连续接收，直到使能位 CREN 清零 （CREN 比 SREN 优先级高）
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、允许中断和装入接收缓冲器
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
9 位异步模式 （RX9 = 0）：
无关位。

bit 2 **FERR**：帧错误位
1 = 帧错误 （可以通过读 RCREG 寄存器刷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 （可以通过清除 CREN 位清零）
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。

寄存器 15-3： BAUDCON：波特率控制寄存器

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

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **ABDOVF**：自动波特率采样进位状态位
1 = 在自动波特率检测模式下出现了 BRG 进位 （必须用软件清零）
0 = 没有发生 BRG 进位

bit 6 **RCIDL**：接收操作空闲状态位
1 = 接收操作处于空闲状态
0 = 接收操作处于活动状态

bit 5 **未用**：读为 0

bit 4 **SCKP**：同步时钟极性选择位
异步模式：
在此模式下未使用。
同步模式：
1 = 空闲状态时钟 （CK）为高电平
0 = 空闲状态时钟 （CK）为低电平

bit 3 **BRG16**：16 位波特率寄存器使能位
1 = 16 位波特率发生器——SPBRGH 和 SPBRG
0 = 8 位波特率发生器——仅 SPBRG （兼容模式），忽略 SPBRGH 的值

bit 2 **未用**：读为 0

bit 1 **WUE**：唤醒使能位
异步模式：
1 = EUSART 将继续采样 RX 引脚——中断在下降沿产生，在下一个上升沿由硬件清零该位
0 = 未监测 RX 引脚或检测到了上升沿
同步模式：
在此模式下未使用。

bit 0 **ABDEN**：自动波特率检测使能位
异步模式：
1 = 在下一个字符使能波特率检测。需要收到 “同步”字段 （55h），完成时由硬件清零
0 = 禁止波特率检测或检测已完成
同步模式：
在此模式下未使用。

© 2007 Microchip Technology Inc.

15.1 波特率发生器 （BRG）

BRG 是一个专用的 8 位或 16 位发生器，支持 EUSART 的异步和同步模式。默认情况下，BRG 工作在 8 位模式下，将 BRG16 位（BAUDCON<3>）置 1 可选择 16 位模式。

SPBRGH:SPBRG 寄存器对控制自由运行的定时器周期。在异步模式下， BRGH （TXSTA<2>）和 BRG16（BAUDCON<3>）也用于控制波特率。在同步模式下，BRGH 位会被忽略。表 15-1 所示为不同 EUSART 模式的波特率计算公式，但仅适用于主控模式（由内部产生时钟信号）。

给出期望的波特率和 Fosc 值，就可以使用表 15-1 中的公式计算 SPBRGH:SPBRG 寄存器的最近似整数值。这样就可以判断波特率误差。例 15-1 给出了计算示例。表 15-2 中给出了不同异步模式下典型的波特率和误差值。

使用高波特率 （BRGH = 1）或 16 位 BRG 有利于减小波特率误差，或者在快速振荡频率条件下实现低波特率。

向 SPBRGH:SPBRG 寄存器写入新值会使 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。

15.1.1 在功耗管理模式下的操作

器件时钟用于产生所需的波特率。当进入一种功耗管理模式时，新时钟源可能会工作在一个不同的频率下。这可能需要调整 SPBRG 寄存器对中的值。

15.1.2 采样

检测电路对 RX 引脚采样三次，以判定 RX 引脚上出现的是高电平还是低电平。

表 15-1： 波特率公式

配置位			BRG/EUSART 模式	波特率计算公式
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:SPBRG 寄存器对的值

例 15-1： 计算波特率误差

器件工作在 Fosc = 16 MHz，目标波特率 = 9600，异步模式， 8 位 BRG：

目标波特率 = Fosc/(64 ([SPBRGH:SPBRG] + 1)

求解 SPBRGH:SPBRG：

X = ((Fosc/ 目标波特率)/64) - 1
= ((16000000/9600)/64) – 1
= [25.042] = 25

波特率计算结果 = 16000000/(64 (25 + 1))
= 9615

误差 = (波特率计算结果 – 目标波特率)/ 目标波特率
= (9615 – 9600)/9600 = 0.16%

表 15-2： 与波特率发生器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART 波特率发生器寄存器的高字节								50
SPBRG	EUSART 波特率发生器寄存器的低字节								50

图注： – = 未用，读为 0。 BRG 不使用阴影单元。

表 15-3： 异步模式下的波特率

波特率 (K)	SYNC = 0， BRGH = 0， BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

波特率 (K)	SYNC = 0， BRGH = 0， BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

波特率 (K)	SYNC = 0， BRGH = 1， BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

波特率 (K)	SYNC = 0， BRGH = 1， BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

© 2007 Microchip Technology Inc.

表 15-3： 异步模式下的波特率 （续）

波特率 (K)	SYNC = 0， BRGH = 0， BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	0.300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1.201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2.403	-0.16	207
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

波特率 (K)	SYNC = 0， BRGH = 0， BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.04	832	0.300	-0.16	415	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

波特率 (K)	SYNC = 0， BRGH = 1， BRG16 = 1 或 SYNC = 1， BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	0.300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1.200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2.400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9.615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19.230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57.142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117.647	-2.12	16

波特率 (K)	SYNC = 0， BRGH = 1， BRG16 = 1 或 SYNC = 1， BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)	实际波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.01	3332	0.300	-0.04	1665	0.300	-0.04	832
1.2	1.200	0.04	832	1.201	-0.16	415	1.201	-0.16	207
2.4	2.404	0.16	415	2.403	-0.16	207	2.403	-0.16	103
9.6	9.615	0.16	103	9.615	-0.16	51	9.615	-0.16	25
19.2	19.231	0.16	51	19.230	-0.16	25	19.230	-0.16	12
57.6	58.824	2.12	16	55.555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

15.1.3 自动波特率检测

增强型 USART 模块支持波特率自动检测和校准。此功能仅在异步模式下当 WUE 位清零时有效。

只要接收到起始位并且 ABDEN 位已置 1，就会开始自动波特率检测（图 15-1）。波特率计算采用自平均的方式。

在自动波特率检测（Auto-Baud Rate Detect，ABD）模式下，BRG 的时钟是反向的。不是由 BRG 为进入的 RX 信号提供时钟信号，而是由 RX 信号为 BRG 定时。在 ABD 模式下，内部波特率发生器被用作计数器来计算输入的串行字节流的位间隔时间。

一旦 ABDEN 位置 1，状态机就会将 BRG 清零并寻找起始位。为了正确计算比特率，自动波特率检测必须接收到一个值为 55h（ASCII 字符 U，也是 LIN 总线的同步字符）的字节。为了尽量减少输入信号不对称造成的影响，测量时段内要包含一个高位和一个低位时间。在起始位后， SPBRG 使用预先选择的时钟源在 RX 的第一个上升沿开始计数。在 RX 引脚传输了 8 个位，或在检测到第 5 个上升沿后，会将相应 BRG 周期内的累加值保存在 SPBRGH:SPBRG 寄存器对中。当第 5 个时钟周期出现时 （应与停止位对应）， ABDEN 位会自动清零。

如果发生了 BRG 计满返回 （从 FFFFh 到 0000h 的溢出），会在 ABDOVF 状态位（BAUDCON<7>）有所反映。该位可在 BRG 溢出时由硬件置 1，也可以由用户通过软件置 1 或清零。在发生进位事件后，ABD 模式继续有效， ABDEN 位保持置 1 （图 15-2）。

在校准波特率周期时，BRG 寄存器时钟频率为预配置时钟频率的 1/8。请注意 BRG 时钟将由 BRG16 和 BRGH 位配置。不管 BRG16 如何设置， SPBRG 和 SPBRGH 都将被用作 16 位计数器。用户通过检查 SPBRGH 寄存器的值是否为 00h，可以验证 8 位模式下是否发生了进位。表 15-4 所示为 BRG 计数器的时钟速率。

当产生 ABD 时序时，EUSART 状态机保持在空闲状态。一旦在 RX 上检测到第 5 个上升沿，中断标志位 RCIF 就会置 1。需要读取 RCREG 中的值，来清除中断标志位 RCIF。应丢弃 RCREG 的值。

注	**1：** 如果 WUE 位与 ABDEN 位同时置 1，自动波特率检测会在间隔字符*之后*的字节开始。 **2：** 需要由用户来判断进入字符波特率是否处于所选 BRG 时钟源范围内。由于位错误率的原因，某些振荡频率和 EUSART 波特率的组合是无法实现的。使用自动波特率检测功能时，必须综合考虑系统总的时序和通迅波特率。

表 15-4： BRG 计数器时钟速率

BRG16	BRGH	BRG 计数器时钟
0	0	FOSC/512
0	1	FOSC/128
1	0	FOSC/128
1	1	FOSC/32

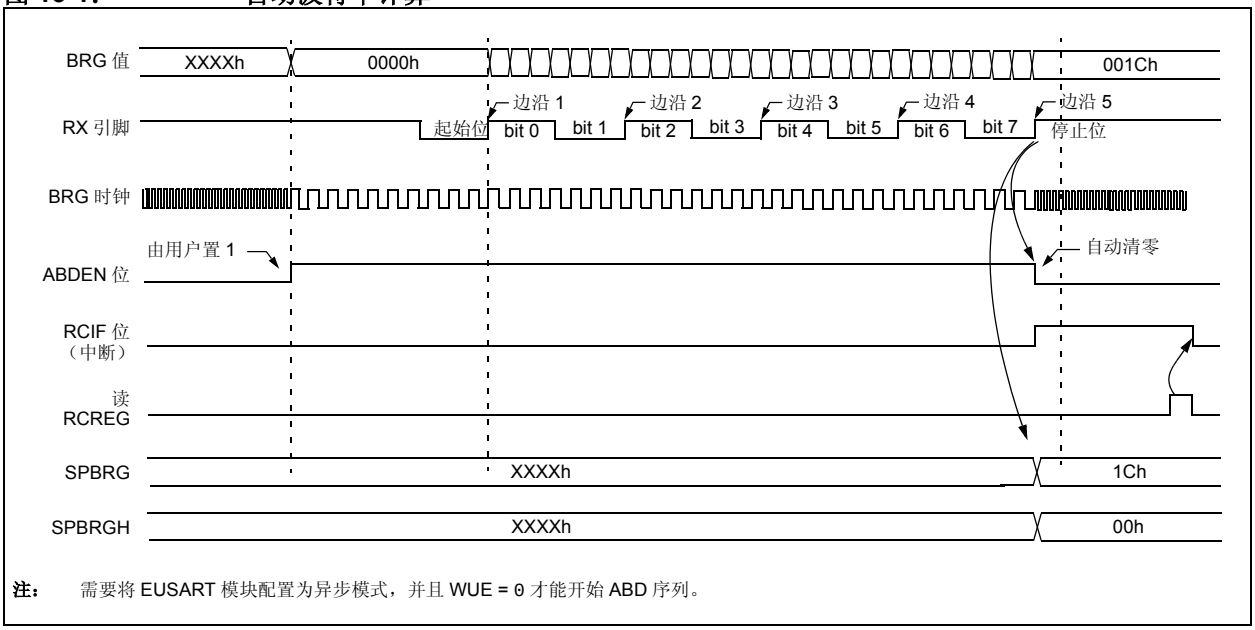
注： 在产生 ABD 时序时，不管 BRG16 如何设置，SPBRG 和 SPBRGH 都被用作 16 位计数器。

15.1.3.1 ABD 和 EUSART 发送

由于 ABD 采样期间 BRG 时钟是反向的，因此在 ABD 期间不能使用 EUSART 发送器。这意味着只要 ABDEN 位置 1，就不能写入 TXREG。用户还应确保在发送期间 ABDEN 不能为置 1 状态，否则可能会导致无法预料的 EUSART 操作。

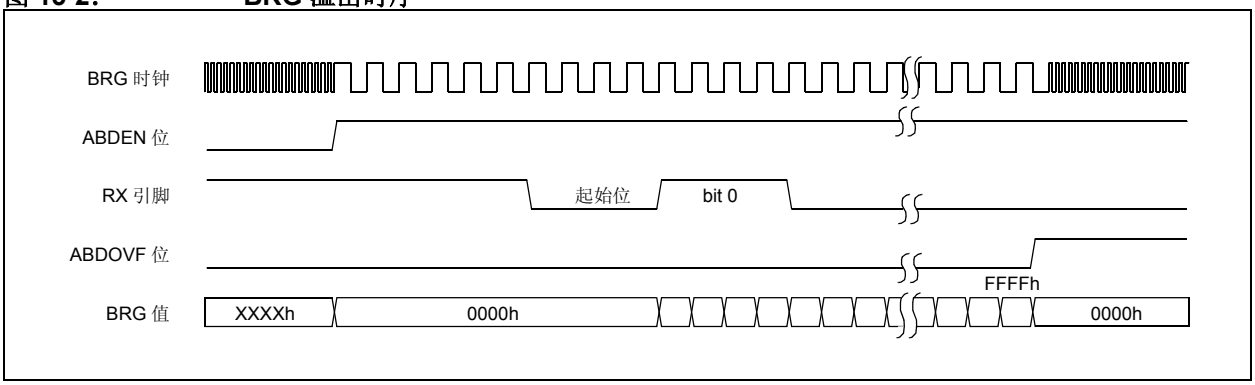
© 2007 Microchip Technology Inc.

图 15-1：　　自动波特率计算



注：　需要将 EUSART 模块配置为异步模式，并且 WUE = 0 才能开始 ABD 序列。

图 15-2：　　BRG 溢出时序



15.2 EUSART 异步模式

通过将 SYNC 位（TXSTA<4>）清零可选择异步工作模式。在此模式下， EUSART 使用标准的不归零 （Non-Return-to-Zero，NRZ）格式（1 个起始位、8 个或 9 个数据位和 1 个停止位）。最常用的数据格式为 8 位。片上专用 8 位 /16 位波特率发生器可借助于振荡器产生标准波特率频率。

EUSART 首先发送和接收 LSb。EUSART 的发送器和接收器在功能上是独立的，但采用相同的数据格式和波特率。波特率发生器可以根据 BRGH 位和 BRG16 位（TXSTA<2> 和BAUDCON<3>）的设置值产生两种不同的波特率时钟，频率分别为移位速率的 16 倍或 64 倍。硬件不支持奇偶校验，但可以用软件实现，校验值保存在第 9 个数据位中。

当工作在异步模式下时，EUSART 模块包括以下重要组成部分：

- 波特率发生器
- 采样电路
- 异步发送器
- 异步接收器
- 同步间隔字符自动唤醒
- 12 位间隔字符发送
- 自动波特率检测

15.2.1 EUSART 异步发送器

图 15-3 给出了 EUSART 发送器框图。发送器的核心是发送 （串行）移位寄存器 （Transmit Shift Register，TSR）。移位寄存器从读 / 写发送缓冲寄存器 TXREG 中获取数据。 TXREG 寄存器中的数据由软件写入。在前一次装入数据的停止位发送前，不会向 TSR 寄存器装入数据。一旦停止位发送完毕， TXREG 寄存器中的新数据 （如果有）就会被装入 TSR。

一旦 TXREG 寄存器向 TSR 寄存器传输了数据 （在 1 个 TCY 内发生）， TXREG 寄存器就为空，同时标志位 TXIF（PIR1<4>）置 1。可以通过将中断允许位 TXIE（PIE1<4>）置 1 或清零来允许或禁止该中断。不管 TXIE 的状态如何，只要中断发生，TXIF 就会置 1 并且不能用软件清零。TXIF 不会在 TXREG 装入新数据时立即被清零，而是在装入指令后的第二个指令周期被清零。因此在 TXREG 装入新数据后立即查询 TXIF，会得到无效结果。

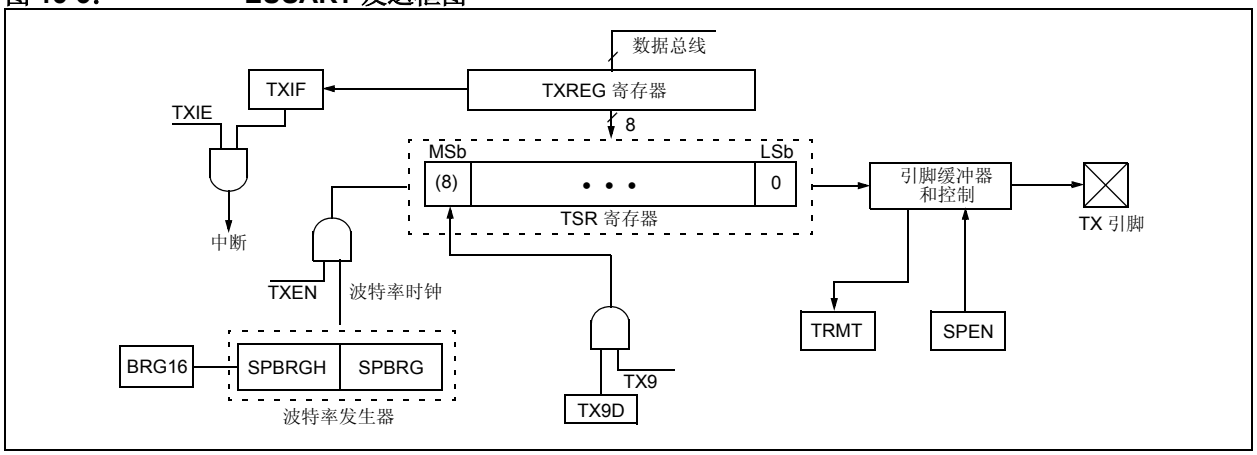
TXIF 指示的是 TXREG 寄存器的状态，而另一个位 TRMT（TXSTA<1>）则指示 TSR 寄存器的状态。TRMT 是只读位，它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断均无关联，因此要确定 TSR 寄存器是否为空，用户只能对此位进行查询。

注	**1：** TSR 寄存器并未映射到数据存储器中，因此用户不能直接访问它。
	2： 当使能位 TXEN 置 1 时，标志位 TXIF 置 1。

设置异步发送操作的步骤如下：

1. 对 SPBRGH:SPBRG 寄存器进行初始化，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1，使能异步串行口。
3. 如果需要中断，将允许位 TXIE 置 1。
4. 如果需要 9 位发送，将发送位 TX9 置 1。可以作为地址 / 数据位使用。
5. 通过将 TXEN 位置 1 使能发送，此操作同时也会将 TXIF 位置 1。
6. 如果选择了 9 位发送，将第 9 位装入 TX9D 位。
7. 将数据装入 TXREG 寄存器 （开始发送）。
8. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）已置 1。

图 15-3： EUSART 发送框图



 © 2007 Microchip Technology Inc.

图 15-4：　　异步发送

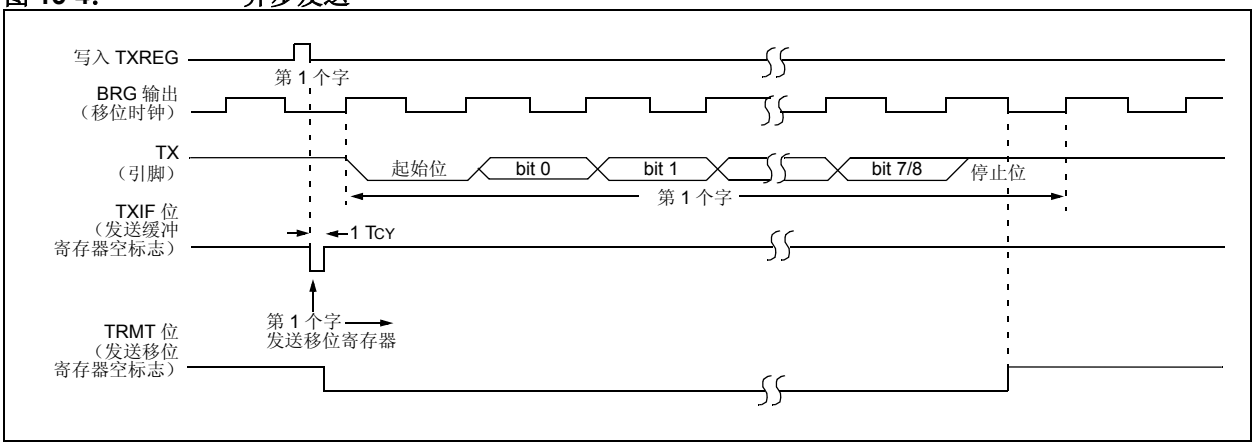
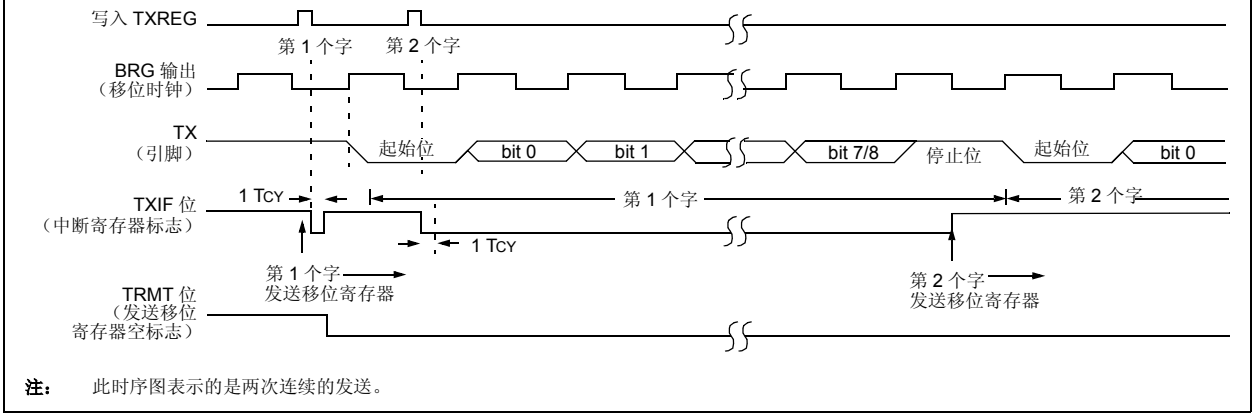


图 15-5：　　异步发送时序 （背对背）



注：　此时序图表示的是两次连续的发送。

表 15-5：　　与异步发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
TXREG	EUSART 发送寄存器								50
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART 波特率发生器寄存器的高字节								50
SPBRG	EUSART 波特率发生器寄存器的低字节								50

图注：　– = 未用的存储单元，读为 0。异步发送不使用阴影单元。

15.2.2 EUSART 异步接收器

图 15-6 给出了接收器框图。在 RX 引脚上接收数据，并驱动数据恢复电路。数据恢复电路实际上是一个工作频率为 16 倍波特率的高速移位器，而主接收串行移位器的工作频率等于比特率或 Fosc。此模式通常用于 RS-232 系统。

设置异步接收操作的步骤如下：

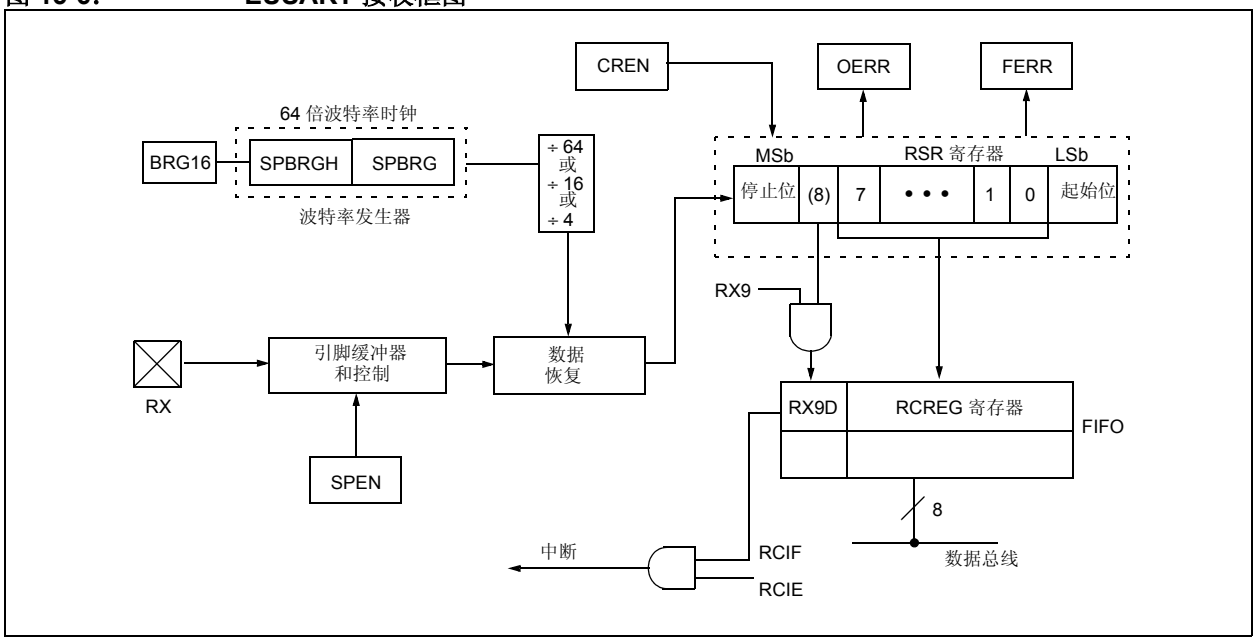
1. 对 SPBRGH:SPBRG 寄存器进行初始化，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1，使能异步串行口。
3. 如果需要中断，将允许位 RCIE 置 1。
4. 如果需要接收 9 位数据，将 RX9 位置 1。
5. 通过将 CREN 位置 1 使能接收。
6. 当接收完成时标志位 RCIF 将被置 1，并且如果中断允许位 RCIE 已置 1，还将产生一个中断。
7. 读 RCSTA 寄存器获取第 9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
8. 读 RCREG 寄存器来读取接收到的 8 位数据。
9. 如果发生错误，通过将使能位 CREN 清零来清除错误。
10. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）已置 1。

15.2.3 设置带有地址检测功能的 9 位模式

此模式通常用于 RS-485 系统。按如下步骤设置带有地址检测功能的异步接收操作：

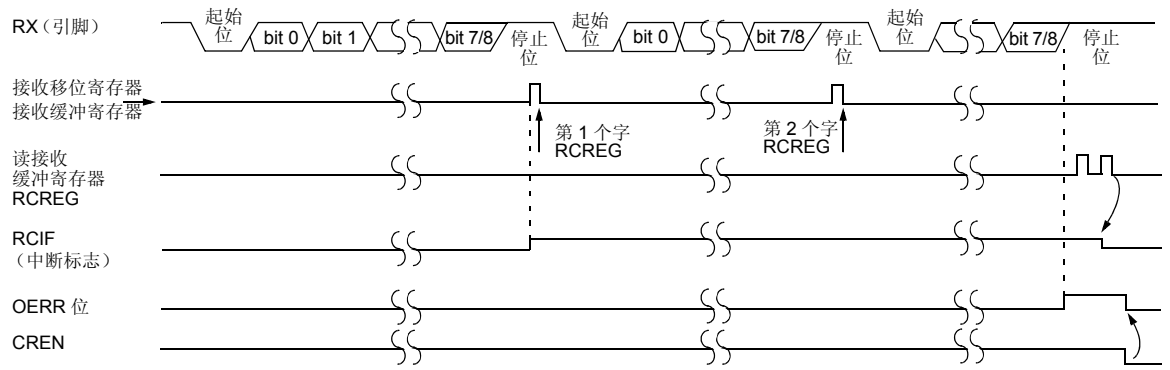
1. 对 SPBRGH:SPBRG 寄存器进行初始化，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1，使能异步串行口。
3. 如果需要中断，将 RCEN 位置 1 并用 RCIP 位选择所需的优先级别。
4. 将 RX9 位置 1，使能 9 位接收。
5. 将 ADDEN 位置 1，使能地址检测。
6. 将 CREN 位置 1，使能接收。
7. 当接收完成时 RCIF 位将被置 1。此时如果 RCIE 和 GIE 位已置 1，还将响应中断。
8. 读 RCSTA 寄存器判断在接收时是否发生了错误，同时读取第 9 位数据（如果适用）。
9. 读 RCREG 判断是否正在对器件进行寻址。
10. 如果发生错误，将 CREN 位清零。
11. 如果已找到器件，将 ADDEN 位清零，允许所有的接收数据进入接收缓冲器并中断 CPU。

图 15-6： EUSART 接收框图



 © 2007 Microchip Technology Inc.

图 15-7：　　异步接收时序



注：　此时序图显示了在 RX 输入引脚上顺序接收的 3 个字。在第 3 个字之后读 RCREG （接收缓冲寄存器）会使 OERR （溢出）位置 1。

表 15-6：　　与异步接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
RCREG	EUSART 接收寄存器								50
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART 波特率发生器寄存器的高字节								50
SPBRG	EUSART 波特率发生器寄存器的低字节								50

图注：　– = 未用的存储单元，读为 0。异步接收不使用阴影单元。

15.2.4 同步间隔字符自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于非激活状态，且无法进行正确的数据接收。自动唤醒功能允许当RX/DT线上有事件发生时唤醒控制器，该功能需要 EUSART 工作在异步模式下。

通过将 WUE 位（BAUDCON<1>）置 1，使能自动唤醒功能。该功能启用后，将禁止 RX/DT 上的典型接收操作，且 EUSART 保持在空闲状态并监视唤醒事件（不管 CPU 运行模式如何）。唤醒事件是指 RX/DT 线上发生高电平到低电平的转换。（这刚好与“同步间隔”字符或 LIN 协议唤醒信号字符的启动条件一致。）

唤醒事件后，模块产生一个 RCIF 中断。在正常工作模式下，中断会与 Q 时钟同步产生（图 15-8）；如果器件处于休眠模式，则两者是不同步的（图 15-9）。通过读 RCREG 寄存器可清除中断条件。

唤醒事件后，当 RX 线上出现由低向高的电平转换时，WUE 位自动清零。此时，EUSART 模块将从空闲状态返回正常工作模式，由此用户可知“同步间隔”事件已经结束。

15.2.4.1 使用自动唤醒功能的特别注意事项

因为自动唤醒功能是通过检测RX/DT上的上升沿跳变实现的，所以在停止位前该引脚上任何的状态改变都可能会产生错误的结束信号并导致数据或帧错误。因此，为了确保正常的传输，必须首先发送全 0 字符。对于标准的 RS-232 器件，该字符是 00h（8 字节），而对于 LIN 总线器件则是 000h（12 位）。

另外还必须考虑振荡器起振时间，尤其在采用起振时间较长的振荡器（即，XT 或 HS 模式）应用中更要注意这一点。“同步间隔”（或唤醒信号）字符必须足够长，并且跟有足够长的时间间隔，以便使选定振荡器有充足的时间起振并保证 EUSART 正确初始化。

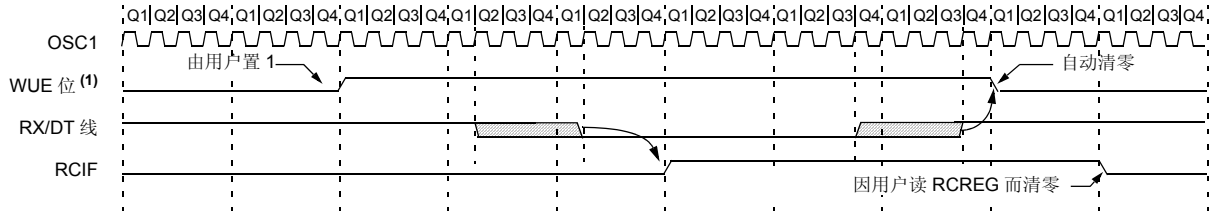
15.2.4.2 使用 WUE 位的特别注意事项

使用 WUE 和 RCIF 事件的时序来判断接收数据的有效性时，有可能会引起一些混淆。如前所述，将 WUE 位置 1 会使 EUSART 进入空闲状态。唤醒事件会通过将 RCIF 位置 1 产生一个接收中断。此后当 RX/DT 出现上升沿时，WUE 位被清零。然后通过读 RCREG 寄存器清除中断条件。一般情况下，RCREG 中的数据是无效数据，应该丢弃。

WUE 位清零（或仍然置 1）同时 RCIF 标志位置 1 并不能表明 RCREG 中数据接收是完整的。用户应该考虑在固件中同时验证是否完整地接收了数据。

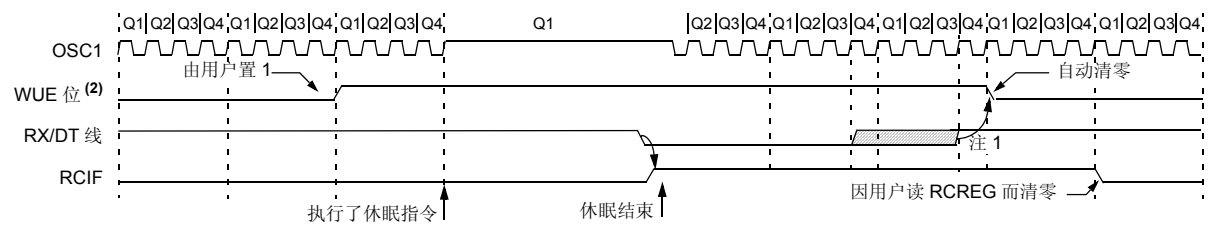
要确保没有丢失有效数据，应检查RCIDL位来验证是否还在接收数据。如果不在接收数据，则可将 WUE 位置 1，使器件立即进入休眠模式。

图 15-8： 正常工作模式下的自动唤醒位（WUE）时序



注 1： WUE 位置 1 时，EUSART 保持在空闲状态。

图 15-9： 休眠模式下的自动唤醒位（WUE）时序



注 1： 如果唤醒事件需要较长的振荡器预热时间，在 *stposc* 信号仍然有效时会发生 WUE 位自动清零。该时序与 Q 时钟出现与否无关。

2： WUE 位置 1 时，EUSART 保持在空闲状态。

 © 2007 Microchip Technology Inc.

15.2.5 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。发送的间隔字符包括 1 个起始位，后面跟有 12 个 0 位和一个停止位。当发送移位寄存器装有数据时，只要 SENDB 和 TXEN 位（TXSTA<3> 和 TXSTA<5>）置 1，就会发送帧间隔字符。请注意写入 TXREG 的数据值会被忽略，并会发送全 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 队列。

请注意在发送间隔字符时写入 TXREG 的数据值会被忽略。写入仅仅是为了启动正确的序列。

正如其在正常发送操作中一样， TRMT 位表明发送正在进行还是处于空闲状态。关于发送间隔字符的时序，请参见图 15-10。

15.2.5.1 间隔和同步发送序列

下述序列会发送一个报文帧头，包括一个间隔字符和其后的自动波特率同步字节。此序列适用于典型的 LIN 总线主控器件。

1. 将 EUSART 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置 1，以设置间隔字符。
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，以便把同步字符装入 FIFO 缓冲器。
5. 间隔字符发送后，硬件会将 SENDB 位复位。此时，同步字符会以预先配置的模式发送。

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

15.2.6 接收间隔字符

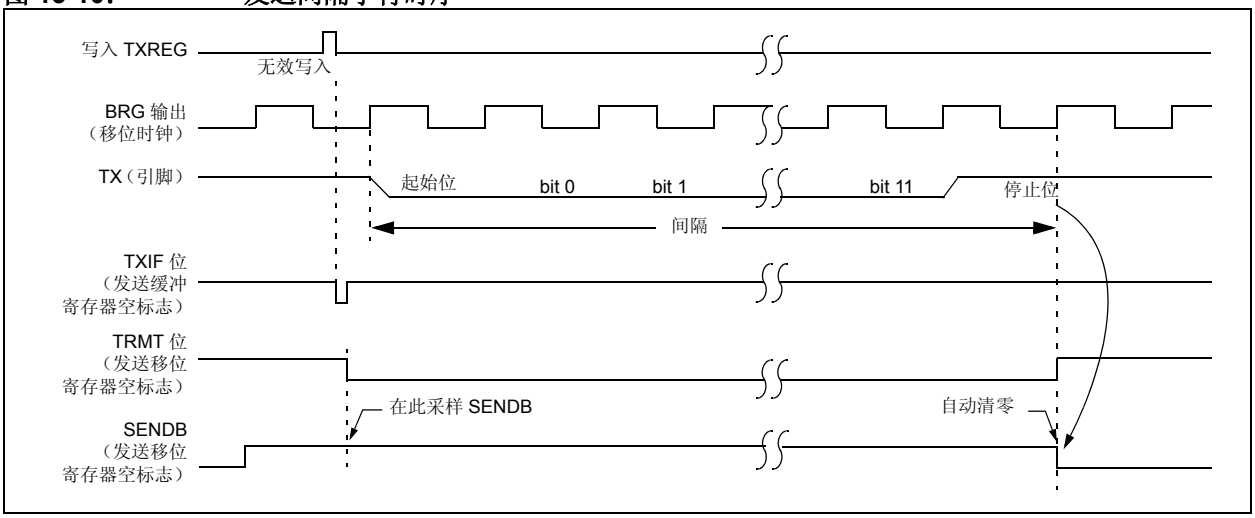
增强型 USART 模块接收间隔字符有两种方法。

第一种方法是强制将波特率配置为典型速率的 9/13。这可以使停止位在正确的采样点 （对于间隔字符为起始位之后的 13 位，对于典型数据则是 8 个数据位）产生。

第二种方法是使用**第 15.2.4 节 “同步间隔字符自动唤醒”**中描述的自动唤醒功能。通过使能此功能，EUSART 将采样 RX/DT 上电平的下两次跳变，产生一个 RCIF 中断，接收下一个数据字节，并在随后产生另一个中断。

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。无论使用哪种方法，用户都可以在检测到 TXIF 中断时马上将 ABD 位置 1。

图 15-10： 发送间隔字符时序



15.3 EUSART 同步主控模式

将 CSRC 位 （TXSTA<7>）置 1 可以进入同步主控模式。在此模式中，数据以半双工方式发送（即发送和接收不能同时进行） 。发送数据时禁止接收，反之亦然。将 SYNC 位（TXSTA<4>）置 1 可以进入同步模式。此外，应将使能位 SPEN（RCSTA<7>）置 1，分别把 TX 和 RX 引脚配置为 CK （时钟）和 DT （数据）线。

主控模式意味着处理器在 CK 时钟线上发送主控时钟信号。时钟极性是通过 SCKP 位（BAUDCON<4>）选择的。将 SCKP 置 1 是将空闲状态时的 CK 设为高电平，将该位清零则将空闲状态时的 CK 设为低电平。此选项支持将本模块与 Microwire 器件配合使用。

15.3.1 EUSART 同步主控发送

图 15-3 给出了 EUSART 发送器框图。发送器的核心是发送 （串行）移位寄存器 （TSR）。移位寄存器从读 / 写发送缓冲寄存器 TXREG 中获取数据。 TXREG 寄存器中的数据由软件写入。在前一次装入数据的最后一位发送完成后，才向 TSR 寄存器装入新数据。一旦最后一位发送完成，就会将 TXREG 寄存器的新数据 （如果有）装入 TSR。

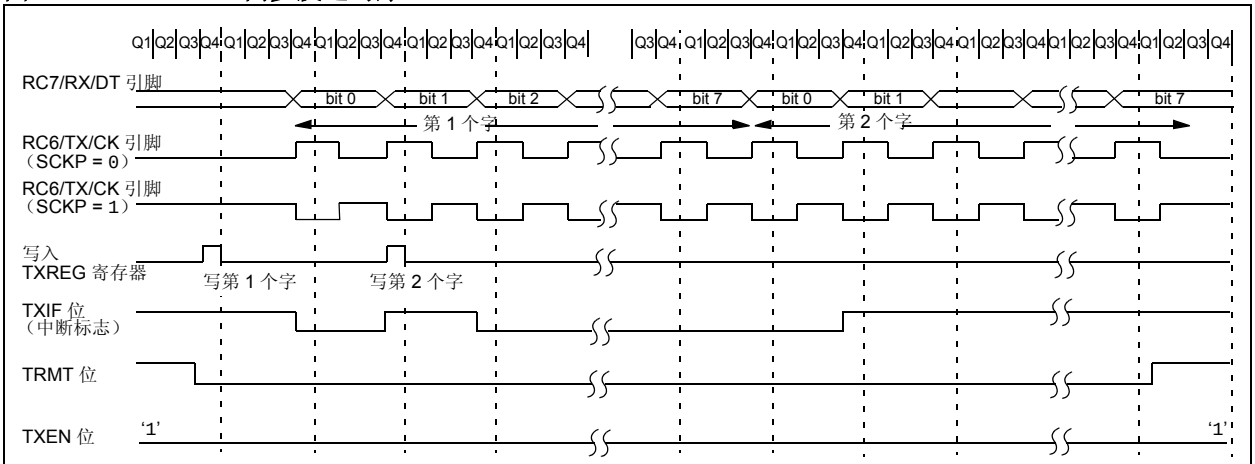
一旦 TXREG 寄存器向 TSR 寄存器传输了数据 （在 1 个 TCY 内发生）， TXREG 寄存器就为空，同时标志位 TXIF（PIR1<4>）被置 1。可以通过将中断允许位 TXIE（PIE1<4>）置 1 或清零来允许或禁止该中断。TXIF 的设置不受 TXIE 状态的影响，且不能用软件清零。只有在新数据写入 TXREG 寄存器时， TXIF 才会复位。

标志位 TXIF 表示的是 TXREG 寄存器的状态，而另一个标志位 TRMT （TXSTA<1>）则表示 TSR 寄存器的状态。TRMT 位是一个只读位，当 TSR 为空时，TRMT 被置 1。 TRMT 位与任何中断均无关联，因此要确定 TSR 寄存器是否为空，用户只能对此位进行查询。TSR 并未映射到数据存储器中，所以用户不能直接访问它。

设置同步主控发送操作的步骤如下：

1. 对 SPBRGH:SPBRG 寄存器进行初始化，设置合适的波特率。按需要将 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC、SPEN 和 CSRC 位置 1，使能同步主控串行口。
3. 如果需要中断，将允许位 TXIE 置 1。
4. 如果需要 9 位发送，将 TX9 位置 1。
5. 将 TXEN 位置 1，使能发送。
6. 如果选择了 9 位发送，将第 9 位装入 TX9D 位。
7. 将数据装入 TXREG 寄存器，启动发送。
8. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）已置 1。

图 15-11： 同步发送时序



注： 同步主控模式 （SPBRG = 0），连续发送两个 8 位字。

© 2007 Microchip Technology Inc.

图 15-12：　　同步发送时序 （由 TXEN 位控制）

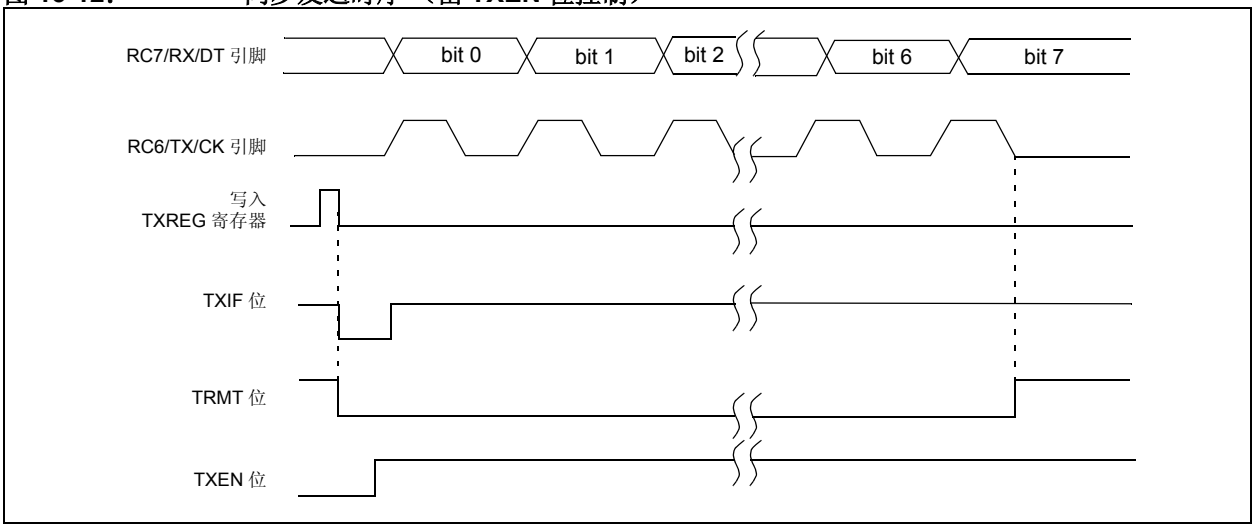


表 15-7：　　与同步主控发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
TXREG	EUSART 发送寄存器								50
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART 波特率发生器寄存器的高字节								50
SPBRG	EUSART 波特率发生器寄存器的低字节								50

图注：　– = 未用，读为 0。同步主控发送不使用阴影单元。

15.3.2 EUSART 同步主控接收

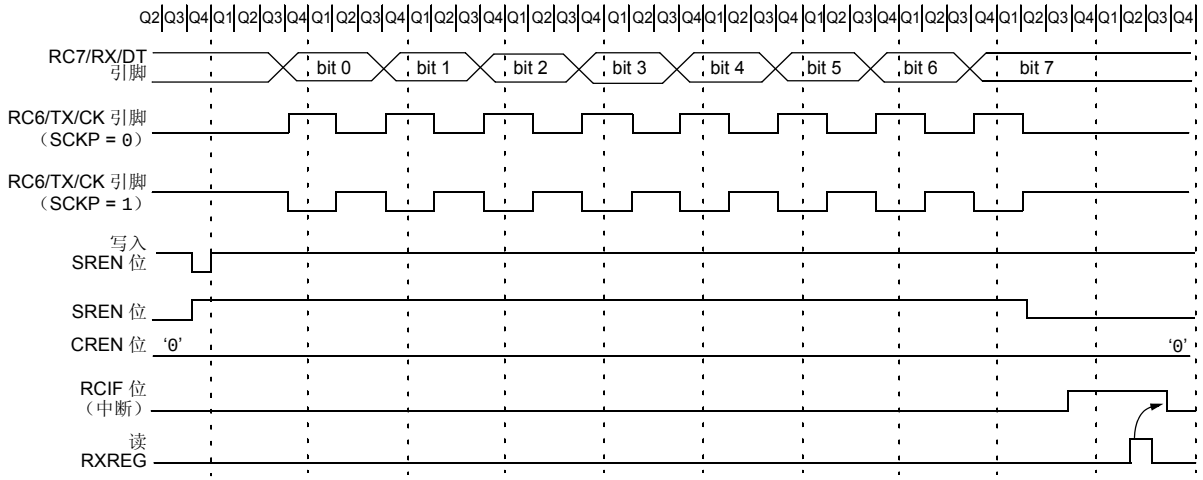
一旦选择了同步模式，只要将单字节接收使能位 SREN（RCSTA<5>）或连续接收使能位CREN（RCSTA<4>）置 1，即可使能接收。在时钟的下降沿采样 RX 引脚上的数据。

如果使能位 SREN 置 1，则只接收单个字。如果将使能位 CREN 置 1，则会连续接收数据，直到将 CREN 位清零。如果两个位均被置 1，则 CREN 优先。

设置同步主控接收操作的步骤如下：

1. 对 SPBRGH:SPBRG 寄存器进行初始化，设置合适的波特率。按需要将 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC、SPEN 和 CSRC 位置 1，使能同步主控串行口。
3. 确保将 CREN 和 SREN 位清零。
4. 如果需要中断，将允许位 RCIE 置 1。
5. 如果需要接收 9 位数据，将 RX9 位置 1。
6. 如果需要单字节接收，将 SREN 位置 1。如果需要连续接收，将 CREN 位置 1。
7. 当接收完成时中断标志位 RCIF 将置 1，此时如果中断允许位 RCIE 已置 1，则还将产生一个中断。
8. 读 RCSTA 寄存器获取第 9 位数据（如果已使能），并判断在接收过程中是否发生了错误。
9. 读 RCREG 寄存器来读取接收到的 8 位数据。
10. 如果发生错误，将 CREN 位清零以清除错误。
11. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）已置 1。

图 15-13： 同步接收（主控模式， SREN）



注： 时序图说明 SREN = 1 且 BRGH = 0 时的同步主控模式。

表 15-8： 与同步主控接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
RCREG	EUSART 接收寄存器								50
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART 波特率发生器寄存器的高字节								51
SPBRG	EUSART 波特率发生器寄存器的低字节								50

图注： — = 未用，读为 0。同步主控接收不使用阴影单元。

© 2007 Microchip Technology Inc.

15.4 EUSART 同步从动模式

将 CSRC （TXSTA<7>）清零可进入同步从动模式。此模式与同步主控模式的区别在于移位时钟由CK引脚上的外部时钟提供 （主控模式中由内部时钟提供）。这使得器件能在任何低功耗模式下发送或接收数据。

15.4.1 EUSART 同步从动发送

除了休眠模式以外，同步主控模式和从动模式的工作方式是相同的。

如果向 TXREG 寄存器写 2 个字，然后执行 SLEEP 指令，则会发生以下事件：

a） 第一个字立即传送到 TSR 寄存器进行发送。
b） 第二个字仍保留在 TXREG 寄存器中。
c） 不会将标志位 TXIF 置 1。
d） 当第一个字移出 TSR 后， TXREG 寄存器将把第二个字传送给 TSR，然后将标志位 TXIF 置 1。
e） 如果中断允许位 TXIE 置 1，中断将把器件从休眠状态唤醒。如果允许了全局中断，程序则会跳转到中断向量处执行。

设置同步从动发送的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1、CSRC 位清零，使能同步从动串行口。
2. 将 CREN 和 SREN 位清零。
3. 如果需要中断，将允许位 TXIE 置 1。
4. 如果需要 9 位发送，将 TX9 位置 1。
5. 将使能位 TXEN 置 1 以使能发送。
6. 如果选择了 9 位发送，将第 9 位装入 TX9D 位。
7. 将数据装入 TXREG 寄存器，启动发送。
8. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）已置 1。

表 15-9： 与同步从动发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
TXREG	EUSART 发送寄存器								50
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART 波特率发生器寄存器的高字节								50
SPBRG	EUSART 波特率发生器寄存器的低字节								50

图注： – = 未用，读为 0。同步从动发送不使用阴影单元。

15.4.2 EUSART 同步从动接收

除了休眠模式、空闲模式以及在从动模式下忽略 SREN 位以外，同步主控和从动模式的工作方式完全相同。

如果在进入休眠或空闲模式前将 CREN 位置 1 使能接收，那么在低功耗模式下可以接收到一个数据字。接收到该字后， RSR 寄存器将把数据发送到 RCREG 寄存器，如果中断允许位 RCIE 已置 1，产生的中断将把芯片从低功耗模式唤醒。如果允许了全局中断，程序则会跳转到中断向量处执行。

设置同步从动接收操作的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零，使能同步从动串行口。
2. 如果需要中断，将允许位 RCIE 置 1。
3. 如果需要接收 9 位数据，将 RX9 位置 1。
4. 将使能位 CREN 置 1 以使能接收。
5. 接收完成时标志位 RCIF 被置 1。如果中断允许位 RCIE 已置 1，还将产生一个中断。
6. 读 RCSTA 寄存器获取第 9 位数据 （如果已使能），并判断在接收过程中是否发生了错误。
7. 读 RCREG 寄存器来读取接收到的 8 位数据。
8. 如果发生错误，将 CREN 位清零以清除错误。
9. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）已置 1。

表 15-10： 与同步从动接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	51
RCREG	EUSART 接收寄存器								50
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	51
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	51
SPBRGH	EUSART 波特率发生器寄存器的高字节								50
SPBRG	EUSART 波特率发生器寄存器的低字节								50

图注： – = 未用，读为 0。同步从动接收不使用阴影单元。

© 2007 Microchip Technology Inc.

16.0 10 位模数转换器 （A/D）模块

28-pin 器件的模数 （Analog-to-Digital， A/D）转换器模块有 10 路输入，40/44-pin 器件有 13 路输入。此模块能将一个模拟输入信号转换成相应的 10 位数字信号。

此模块有五个寄存器：

- A/D 转换结果高位寄存器 （ADRESH）
- A/D 转换结果低位寄存器 （ADRESL）
- A/D 转换控制寄存器 0 （ADCON0）
- A/D 转换控制寄存器 1 （ADCON1）
- A/D 转换控制寄存器 2 （ADCON2）

ADCON0 寄存器（如寄存器 16-1 所示）控制 A/D 模块的工作。ADCON1 寄存器（如寄存器 16-2 所示）配置端口引脚功能。 ADCON2 寄存器 （如寄存器 16-3 所示）配置 A/D 时钟源、可编程采样时间和输出结果的对齐方式。

寄存器 16-1： ADCON0：A/D 控制寄存器 0

U0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未用：**读为 0

bit 5-2 **CHS3:CHS0：**模拟通道选择位
0000 = 通道 0 （AN0）
0001 = 通道 1 （AN1）
0010 = 通道 2 （AN2）
0011 = 通道 3 （AN3）
0100 = 通道 4 （AN4）
0101 = 通道 5 （AN5）**(1,2)**
0110 = 通道 6 （AN6）**(1,2)**
0111 = 通道 7 （AN7）**(1,2)**
1000 = 通道 8 （AN8）
1001 = 通道 9 （AN9）
1010 = 通道 10 （AN10）
1011 = 通道 11 （AN11）
1100 = 通道 12 （AN12）
1101 = 未实现通道 **(2)**
1110 = 未实现通道 **(2)**
1111 = 未实现通道 **(2)**

bit 1 **GO/$\overline{\text{DONE}}$：**A/D 转换状态位
当 ADON = 1 时：
1 = A/D 转换正在进行
0 = A/D 空闲

bit 0 **ADON：**A/D 模块使能位
1 = 使能 A/D 转换器模块
0 = 禁止 A/D 转换器模块

注 1： 这些通道在 28-pin 器件上未实现。
2： 在未实现通道上执行转换会返回不确定的输入值。

寄存器 16-2： ADCON1：A/D 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0(1)	R/W(1)	R/W(1)	R/W(1)
—	—	VCFG0	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未用：**读为 0

bit 5 **VCFG0：**参考电压配置位 （VREF- 电压源）
1 = VREF- (AN2)
0 = VSS

bit 4 **VCFG0：**参考电压配置位 （VREF+ 电压源）
1 = VREF+ (AN3)
0 = VDD

bit 3-0 **PCFG3:PCFG0：**A/D 端口配置控制位：

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7(2)	AN6(2)	AN5(2)	AN4	AN3	AN2	AN1	AN0
0000(1)	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111(1)	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = 模拟输入 D = 数字 I/O

注 1： PCFG 位的 POR 值取决于 PBADEN 的值。当 PBADEN = 1 时， PCFG<3:0> = 0000 ；当 PBADEN = 0 时， PCFG<3:0> = 0111。

2： AN5 到 AN7 仅在 40/44-pin 器件上可用。

© 2007 Microchip Technology Inc.

寄存器 16-3：　　ADCON2：A/D 控制寄存器 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7　　**ADFM**：A/D 结果格式选择位
1 = 右对齐
0 = 左对齐

bit 6　　**未用**：读为 0

bit 5-3　　**ACQT2:ACQT0**：A/D 采样时间选择位
111 = 20 个 TAD
110 = 16 个 TAD
101 = 12 个 TAD
100 = 8 个 TAD
011 = 6 个 TAD
010 = 4 个 TAD
001 = 2 个 TAD
000 = 0 个 TAD(1)

bit 2-0　　**ADCS2:ADCS0**：A/D 转换时钟选择位
111 = FRC （时钟来自 A/D 模块 RC 振荡器）(1)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC （时钟来自 A/D 模块 RC 振荡器）(1)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

注　1： 如果选择了 FRC 时钟源，在 A/D 时钟启动之前会加上一个 TCY （指令周期）的延时。这可以保证在开始转换之前执行 SLEEP 指令。

可通过软件选择将器件的正负电源电压（VDD 和 VSS）或 RA3/AN3/VREF+ 和 RA2/AN2/VREF- 引脚上的电压作为 A/D 转换的模拟参考电压。

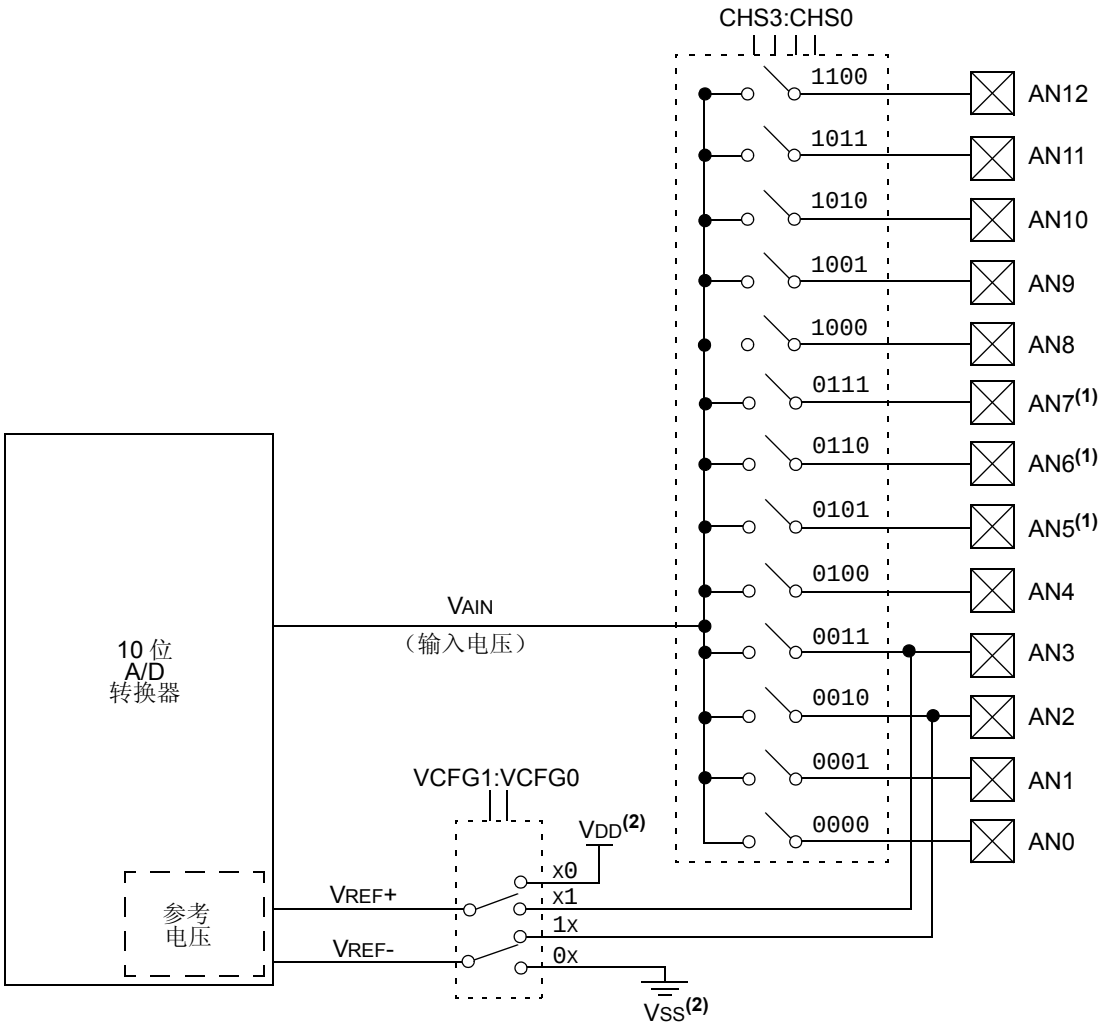
A/D 转换器具有可在休眠状态下工作的特性。要使 A/D 转换器在休眠状态下工作，其时钟必须来自于 A/D 模块内部的 RC 振荡器。

采样保持电路的输出是转换器的输入，A/D 转换器采用逐次逼近法得到转换结果。

器件复位将强制所有寄存器进入复位状态。这将迫使 A/D 模块关闭并中止正在进行的转换。

与 A/D 转换器相关的每个端口引脚都可以被配置为模拟输入或数字 I/O。ADRESH 和 ADRESL 寄存器保存 A/D 转换的结果。当 A/D 转换完成时，结果被装入 ADRESH:ADRESL 寄存器，GO/DONE 位（在 ADCON0 寄存器中）被清零且 A/D 中断标志位 ADIF 被置 1。图 16-1 给出了 A/D 模块的框图。

图 16-1： A/D 框图



注 1： 通道 AN5 到 AN7 在 28-pin 器件上不可用。

2： 在 I/O 引脚与 VDD 和 VSS 之间连有保护二极管。

© 2007 Microchip Technology Inc.

上电复位时，ADRESH:ADRESL 寄存器的值保持不变。上电复位后， ADRESH:ADRESL 寄存器的值不确定。

按要求配置好 A/D 模块后，在开始转换之前必须采样选定的通道。模拟输入通道的相应 TRIS 位必须设置为输入。采集时间的确定，请参见**第 16.1 节 “A/D 采集要求”**。在采样完成之后，即可启动 A/D 转换。采集时间可以被编程置于 GO/DONE 位置 1 和启动转换之间。

在执行 A/D 转换时应该遵循以下步骤：

1. 配置 A/D 模块：
   - 配置模拟引脚、参考电压和数字 I/O （通过 ADCON1 寄存器）
   - 选择 A/D 输入通道 （通过 ADCON0 寄存器）
   - 选择 A/D 采集时间 （通过 ADCON2 寄存器）
   - 选择 A/D 转换时钟 （通过 ADCON2 寄存器）
   - 使能 A/D 模块 （通过 ADCON0 寄存器）
2. 需要时，配置 A/D 中断：
   - ADIF 位清零
   - ADIE 位置 1
   - GIE 位置 1
3. 需要时，等待所需的采样时间。
4. 启动转换：
   - 将 GO/DONE 位置 1（通过 ADCON0 寄存器）
5. 等待 A/D 转换完成，可通过以下两种方法之一来判断转换是否完成：
   - 查询 GO/DONE 位是否被清零

   或
   - 等待 A/D 中断
6. 读取 A/D 结果寄存器（ADRESH:ADRESL），需要时将 ADIF 位清零。
7. 如需再次进行 A/D 转换，返回步骤 1 或步骤 2。每位的 A/D 转换时间定义为 TAD。在下一次采样开始前需要等待至少 3TAD 的时间。

图 16-2： A/D 转换方式

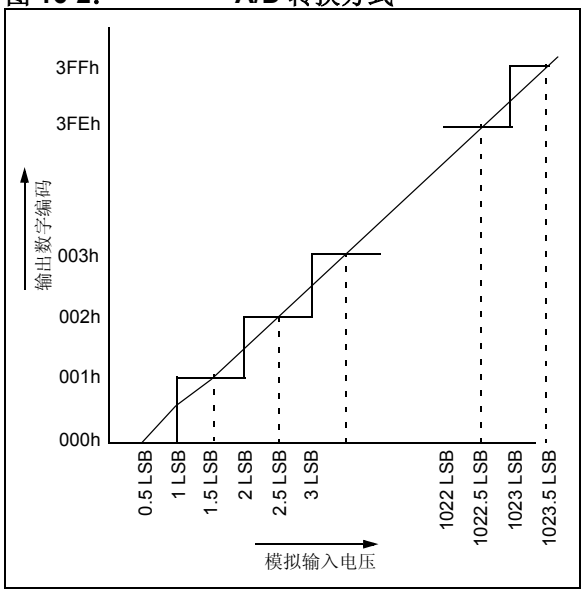
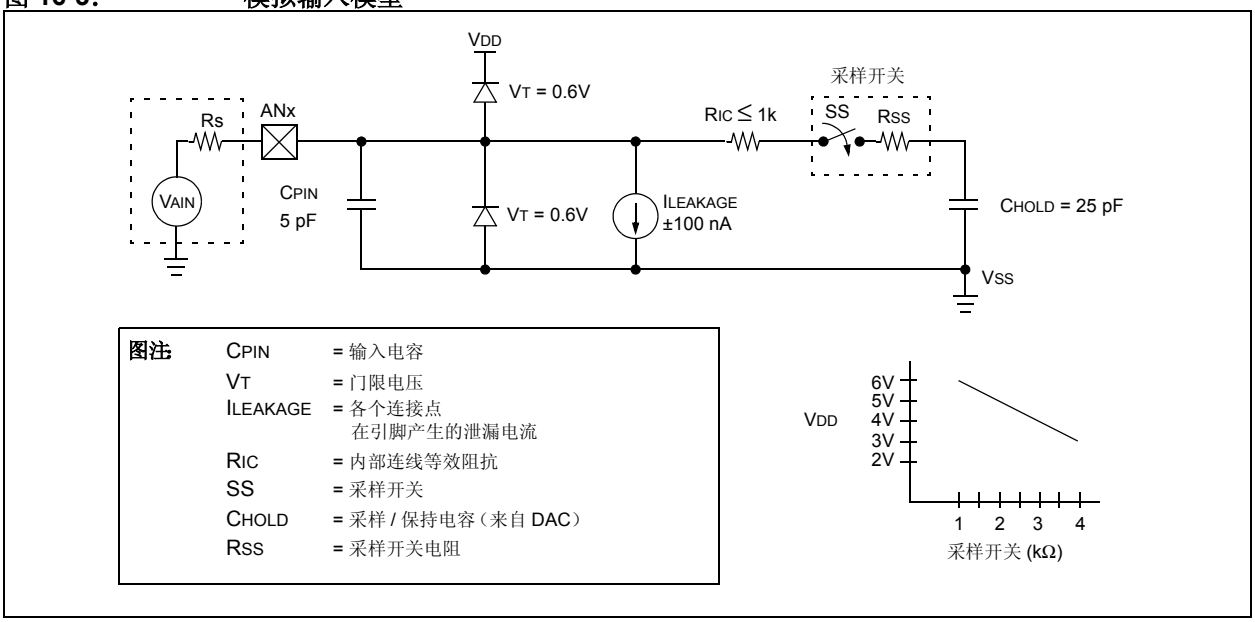


图 16-3： 模拟输入模型



16.1 A/D 采集要求

为了使 A/D 转换器达到规定的精度，必须使充电保持电容（CHOLD）充满至输入通道的电压电平。模拟输入模型见图 16-3。电源阻抗（RS）和内部采样开关阻抗（RSS）直接影响给电容 CHOLD 充电所需要的时间。采样开关阻抗（RSS）值随器件电压（VDD）变化而改变。电源阻抗影响模拟输入的失调电压（由于引脚泄漏电流的原因）。**模拟信号源的最大阻抗推荐值为 2.5 kΩ**。选择（改变）模拟输入通道后，必须对通道进行采样才能启动转换，采集时间必须大于最小采集时间。

注：	当开始转换时，将保持电容与输入引脚断开。

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（A/D 转换需要 1024 步）。1/2 LSb 误差是 A/D 达到规定分辨率所允许的最大误差。

例 16-3 显示了所需的最小采集时间 TACQ 的计算过程。计算结果基于以下假设：

CHOLD	=	25 pF
Rs	=	2.5 kΩ
转换误差	≤	1/2 LSb
VDD	=	5V → RSS = 2 kΩ
温度	=	85°C （系统最大值）

公式 16-1： 采集时间

$$T_{ACQ} = \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数}$$
$$= T_{AMP} + T_C + T_{COFF}$$

公式 16-2： A/D 最小充电时间

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$

或

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)$$

公式 16-3： 计算所需要的最小采集时间

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$
$$T_{AMP} = 0.2\ \mu s$$
$$T_{COFF} = (Temp - 25°C)(0.02\ \mu s/°C)$$
$$= (85°C - 25°C)(0.02\ \mu s/°C)$$
$$= 1.2\ \mu s$$

只有在温度 > 25°C 时才需要温度系数。当温度低于 25°C 时， $T_{COFF} = 0$ ms。

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2047)\ \mu s$$
$$= -(25\ pF)(1\ k\Omega + 2\ k\Omega + 2.5\ k\Omega)\ln(0.0004883)\ \mu s$$
$$= 1.05\ \mu s$$
$$T_{ACQ} = 0.2\ \mu s + 1\ \mu s + 1.2\ \mu s$$
$$= 2.4\ \mu s$$

© 2007 Microchip Technology Inc.

16.2 选择和配置采集时间

用户可以利用 ADCON2 寄存器选择采集时间，该采集时间发生在每次 GO/DONE 位置 1 之后。用户也可以使用自动的采集时间。

采集时间可由 ACQT2:ACQT0 位（ADCON2<5:3>）设置，它提供了 2 到 20 个 TAD 范围。当 GO/DONE 位被置 1 时，A/D 模块会继续在选定采集时间内采样输入通道，然后自动开始一次转换。由于采集时间是编程的，因此没有必要在选择通道和将 GO/DONE 位置 1 之间等待一个采集时间。

当 ACQT2:ACQT0 = 000 时选择手动采集。当 GO/DONE 位被置 1 时，采样停止并开始转换。用户必须确保在选择输入通道和将 GO/DONE 位置 1 之间已插入了所需的采集时间。此选项也是 ACQT2:ACQT0 位的默认复位状态，并且与不提供可编程采集时间的器件相兼容。

在这两种情况下，当转换完成时，GO/DONE 位均被清零，ADIF 标志位均被置 1 并且 A/D 开始再次对当前选择的通道进行采样。如果采集时间已经被编程，那么将不会有任何指示显示采集时间是否结束，或转换是否开始。

16.3 选择 A/D 转换时钟

每位的 A/D 转换时间定义为 TAD。每完成一次 10 位 A/D 转换需要 11 个 TAD。可用软件选择 A/D 转换的时钟源。TAD 可有以下 7 种选择：

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- 内部 RC 振荡器

为了实现正确的 A/D 转换，A/D 转换时钟（TAD）必须尽可能得小，但它必须大于最小 TAD（更多信息，请参见表 21-18 中的参数 130）。

表 16-1 显示了器件在不同的工作频率下和选择不同的 A/D 时钟源时得到的 TAD。

表 16-1： 不同器件工作频率下的 TAD

A/D 时钟源 （TAD）		最高器件频率	
工作状态	ADCS2:ADCS0	PIC18FXX31	PIC18LFXXXX(4)
2 TOSC	000	2.86 MHz	1.43 kHz
4 TOSC	100	5.71 MHz	2.86 MHz
8 TOSC	001	11.43 MHz	5.72 MHz
16 TOSC	101	22.86 MHz	11.43 MHz
32 TOSC	010	40.0 MHz	22.86 MHz
64 TOSC	110	40.0 MHz	22.86 MHz
RC(3)	x11	1.00 MHz(1)	1.00 MHz(2)

注 1： RC 源的典型 TAD 时间为 1.2 μs。

2： RC 源的典型 TAD 时间为 2.5 μs。

3： 当器件工作频率高于 1 MHz 时，整个转换过程必须在休眠模式下进行，否则 A/D 转换精度可能超出规范允许的范围。

4： 仅适用于低功耗器件。

16.4 在功耗管理模式下的操作

在功耗管理模式中，自动采集时间和 A/D 转换时钟的选择一定程度上可由时钟源和频率决定。

如果希望器件处于功耗管理模式时进行 A/D 采集转换，就应该根据该模式下使用的时钟对 ADCON2 中的 ACQT2:ACQT0 和 ADCS2:ADCS0 位进行更新。在进入功耗管理模式之后，就可以开始 A/D 采集或转换。采集或转换开始以后，器件应继续使用相同的时钟源直到转换完成。

如果需要，在转换期间也可以将器件置于相应的空闲模式。如果器件的时钟频率小于 1 MHz，就应该选择 A/D 模块的 RC 时钟源。

在休眠模式下工作需要选择 A/D 模块的 FRC 时钟。如果将 ACQT2:ACQT0 设置为 000 并启动 A/D 转换，转换将延时一个指令周期以允许执行SLEEP指令并进入休眠模式。 IDLEN 位 （OSCCON<7>）必须在转换开始之前被清零。

16.5 配置模拟端口引脚

ADCON1、TRISA、TRISB 和 TRISE 寄存器均可用于配置 A/D 端口引脚。如果希望端口引脚为模拟输入，则必须将相应的 TRIS 位置 1（输入）。如果将 TRIS 位清零（输出），则数字输出电平（VOH 或 VOL）将被转换。

A/D 转换与 CHS3:CHS0 位及 TRIS 位的状态无关。

注 1： 读取端口寄存器时，所有配置为模拟输入通道的引脚均读为 0（低电平）。配置为数字输入的引脚将按模拟输入进行转换。配置为数字输入的引脚将模拟输入电平精确转换为数字引脚电平。

2： 定义为数字输入引脚上的模拟电平可能会导致数字输入缓冲器消耗的电流超出器件规范。

3： 通过控制 ADCON1 中的 PCFG0 位的复位方式，配置寄存器 3H 中的 PBADEN 位可将 PORTB 引脚配置成复位时为模拟或数字引脚。

© 2007 Microchip Technology Inc.

16.6 A/D 转换

图 16-4 显示了在 GO/$\overline{\text{DONE}}$ 位置 1 且 ACQT2:ACQT0 位被清零后 A/D 转换器的工作状态。转换在下一条指令执行之后开始，以允许器件在转换开始之前进入休眠模式。

图 16-5 显示了在 GO/$\overline{\text{DONE}}$ 位置 1， ACQT2:ACQT0 位被设置为 010，且在转换开始之前选择 4 TAD 采集时间后 A/D 转换器的工作状态。

在转换期间将 GO/$\overline{\text{DONE}}$ 位清零将中止当前的 A/D 转换。不会用尚未完成的 A/D 转换结果更新 A/D 结果寄存器对。这意味着 ADRESH:ADRESL 寄存器仍将保持上一次转换的结果（或上一次写入 ADRESH:ADRESL 寄存器的值）。

在 A/D 转换完成或停止以后，需要等待 2 个 TAD 才能开始下一次采集。等待时间一到，将自动开始对所选通道进行采集。

注：	不应在启动A/D模块的指令中将GO/$\overline{\text{DONE}}$位置 1。

16.7 放电

放电过程用于对电容阵列的值进行初始化。在每次采样之前都会对此阵列放电。这一特性有助于优化单位增益放大器，因为每次需要重新为电容阵列充电，而不是根据以前测量的值进行充放电。

图 16-4： A/D 转换 TAD 周期 （ACQT<2:0> = 000， TACQ = 0）

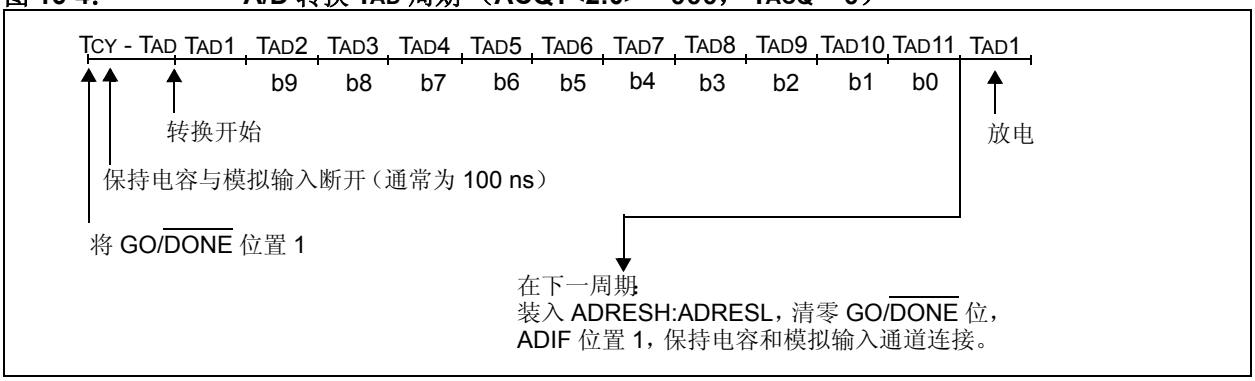
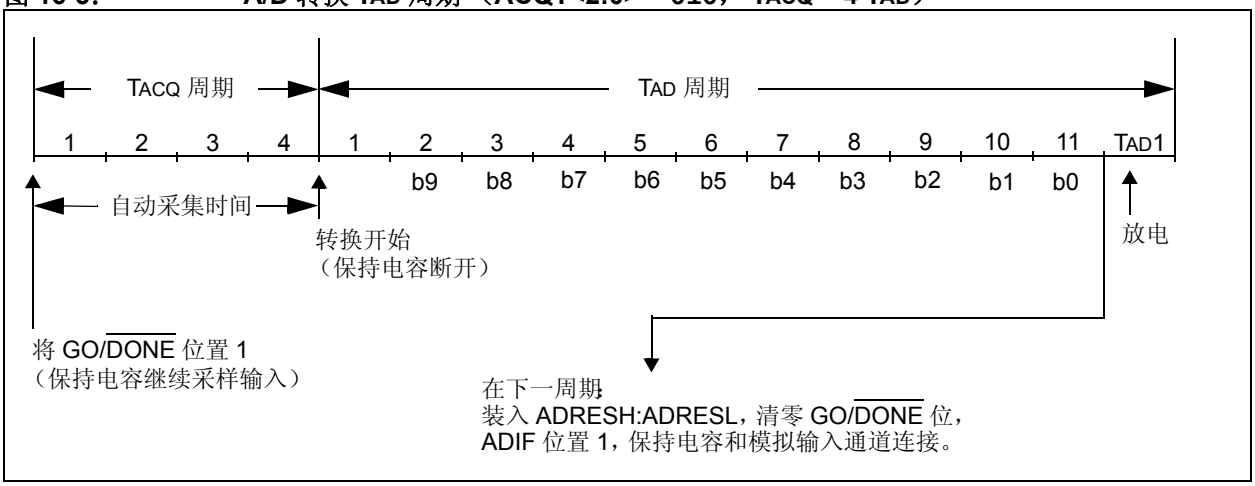


图 16-5： A/D 转换 TAD 周期 （ACQT<2:0> = 010， TACQ = 4 TAD）



16.8 CCP1 触发器的使用

CCP1 模块的特殊事件触发信号可以启动 A/D 转换。这需要将 CCP1M3:CCP1M0 位 （CCP1CON<3:0>）设置为 1011，且使能 A/D 模块 （ADON 位置 1）。发生触发事件时， GO/$\overline{\text{DONE}}$ 位被置 1，启动 A/D 采集和转换并将 Timer1 计数器复位为 0。复位 Timer1 可自动重复 A/D 采集周期，最大限度地降低了软件开销 （将 ADRESH:ADRESL 内容移到指定单元）。特殊事件触发信号将 GO/$\overline{\text{DONE}}$ 位置 1（启动转换）之前，用户必须选择正确的模拟输入通道和最小采集时间，或选择合适的 TACQ 时间。

如果未使能 A/D 模块（ADON 清零），则特殊事件触发信号将被 A/D 模块忽略，但它仍会将 Timer1 计数器复位。

表 16-2： 与 A/D 操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	51
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	51
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	51
PIR2	OSCFIF	—	USBIF	—	—	HLVDIF	—	—	51
PIE2	OSCFIE	—	USBIE	—	—	HLVDIE	—	—	51
IPR2	OSCFIP	—	USBIP	—	—	HLVDIP	—	—	51
ADRESH	A/D 结果寄存器的高字节								50
ADRESL	A/D 结果寄存器的低字节								50
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	50
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	50
PORTA	—	RA6(2)	RA5	RA4	RA3	RA2	RA1	RA0	51
TRISA	—	TRISA6(2)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	51
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	51
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	51
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	51
PORTE	—	—	—	—	RE3(1,3)	RE2(4)	RE1(4)	RE0(4)	51
TRISE(4)	—	—	—	—	—	TRISE2(4)	TRISE1(4)	TRISE0(4)	51
LATE(4)	—	—	—	—	—	LATE2(4)	LATE1(4)	LATE0(4)	51

图注： – = 未用，读为 0。 A/D 转换不使用阴影单元。

注 1： 仅当主复位功能被禁止时 （MCLRE 配置位 = 0）可用。

2： RA6 及其相关的锁存器和数据方向位根据振荡器配置使能为 I/O 引脚；否则，它们将被读为 0。

3： RE3 端口位在 MCLRE 配置位为 0 时仅可用作输入引脚。

4： 这些寄存器和 / 或位在 28-pin 器件上未使用。

© 2007 Microchip Technology Inc.

17.0 高 / 低压检测 （HLVD）

PIC18F2450/4450 器件配有一个高 / 低压检测模块（HLVD）。该模块是一个可编程的电路，它允许用户指定器件的电压跳变点和变化方向。如果器件电压按照指定的方向相对于该跳变点发生了偏离，就会将中断标志位置 1。如果允许了中断，程序将跳转到中断向量地址处执行，由软件响应该中断。

高 / 低压检测控制寄存器（寄存器 17-1）完全控制 HLVD 模块的工作。用户可通过软件控制该寄存器将电路“关闭”，从而使器件的电流消耗降至最低。

图 17-1 给出了 HLVD 模块的框图。

寄存器 17-1： HLVDCON：高 / 低压检测控制寄存器

R/W-0	U-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1
VDIRMAG	—	IRVST	HLVDEN	HLVDL3(1)	HLVDL2(1)	HLVDL1(1)	HLVDL0(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7 **VDIRMAG**：电压方向大小选择位
1 = 当电压等于或超过跳变点 （HLVDL3:HLVDL0）时，事件发生
0 = 当电压等于或低于跳变点 （HLVDL3:HLVDL0）时，事件发生

bit 6 **未用**：读为 0

bit 5 **IRVST**：内部参考电压稳定标志位
1 = 表示电压检测逻辑在检测到指定的电压跳变点时，产生中断标志
0 = 表示电压检测逻辑在检测到指定的电压跳变点时，不会产生中断标志，并且 HLVD 中断不被使能

bit 4 **HLVDEN**：高 / 低压检测模块使能位
1 = 使能 HLVD
0 = 禁止 HLVD

bit 3-0 **HLVDL3:HLVDL0**：电压检测门限位 (1)
1111 = 保留
1110 = 最大设置
.
.
.
0000 = 最小设置

注 1： 具体规范，请参见**第 21.0 节 “电气特性”**的表 21-4。

通过将 HLVDEN 位置 1 使能该模块。每次使能 HLVD 模块时，电路需要一定时间才能稳定下来。IRVST 位是一个只读位，用来表明电路是否稳定。仅当电路稳定且 IRVST 位置 1 后，该模块才能产生中断。

VDIRMAG 位决定该模块的整体工作状态。当 VDIRMAG 清零时，模块监视 VDD 看它是否跌落到预先确定的设置点以下。当该位置 1 时，模块监视 VDD 看它是否上升到设置点以上。

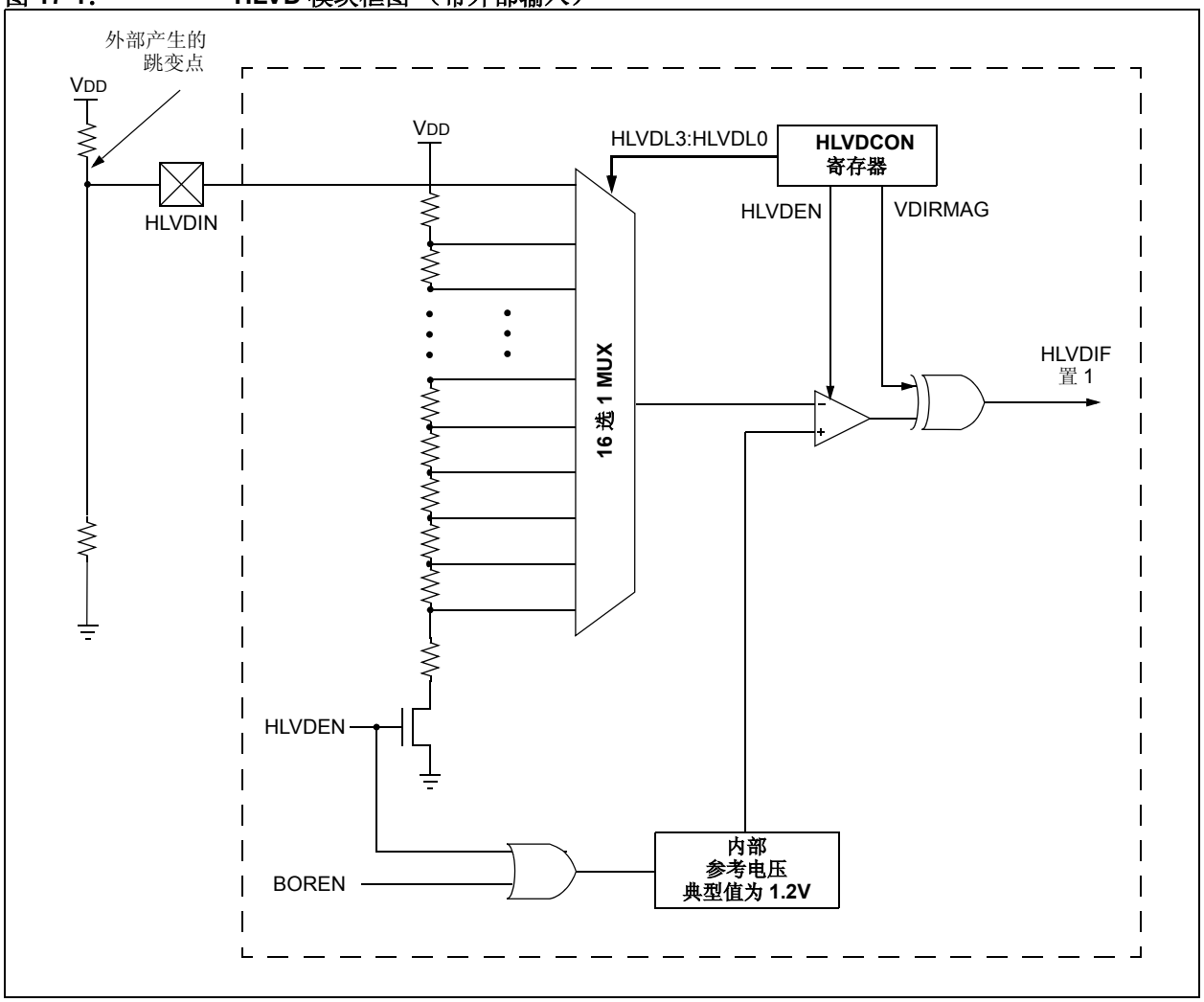
17.1 工作原理

当使能了 HLVD 模块时，比较器使用内部产生的参考电压作为设置点。将设置点的电压与跳变点电压作比较，其中电阻分压器中的每个节点均代表一个电压跳变点。“跳变点”电压是被检测到的高压或低压事件的电平，它取决于该模块的配置。当供电电压等于跳变点电压时，电阻阵列的节点电压输出值等于由参考电压源模块产生的内部参考电压。然后比较器通过将 HLVDIF 位置 1 产生一个中断信号。

可用软件编程指定跳变点电压为 16 个值中的任何一个。通过对 HLVDL3:HLVDL0 位（HLVDCON<3:0>）进行编程可以选择跳变点。

HLVD 模块允许用户通过外部电源向模块提供跳变电压。当 HLVDL3:HLVDL0 位被设置为 1111 时，使能此模式。在此状态下，比较器输入与外部输入引脚 HLVDIN 复用。因此用户可以灵活地配置高 / 低压检测中断，使之可以在有效工作范围内的任何电压点上产生。

图 17-1： HLVD 模块框图 （带外部输入）



 © 2007 Microchip Technology Inc.

17.2 HLVD 设置

要设置 HLVD 模块，需要遵循以下步骤：

1. 通过清零 HLVDEN 位（HLVDCON<4>）禁止该模块。
2. 将值写入 HLVDL3:HLVDL0，选择所需的 HLVD 跳变点。
3. 将 VDIRMAG 位设置为检测高压（VDIRMAG = 1）或低压 （VDIRMAG = 0）。
4. 通过将 HLVDEN 位置 1，使能 HLVD 模块。
5. 清零 HLVD 中断标志位 HLVDIF （PIR2<2>），该位可能被上次中断置 1。
6. 如果需要中断，将 HLVDIE 和 GIE/GIEH 位（PIE2<2> 和 INTCON<7>）置 1 允许 HLVD 中断。直到 IRVST 位也置 1 时才会发生中断。

17.3 电流消耗

使能了该模块就使能了 HLVD 比较器和分压器，并将消耗静态电流。电气规范中的参数 D022（**第 21.2 节“DC 特性”**）给出了使能该模块时的电流总消耗。

HLVD 模块无需一直工作，工作与否取决于具体的应用。要降低电流消耗，只需要在检测电压时，短时间地使能HLVD电路，在检测完成之后马上禁止HLVD模块。

17.4 HLVD 启动时间

电气规范中的参数 D420（见**第 21.0 节“电气特性”**中的表 21-4）规定了 HLVD 模块的内部参考电压，该参考电压也可供其他内部电路 （如可编程欠压复位电路）使用。如果禁止了 HLVD 或其他使用参考电压的电路以降低器件的电流消耗，则参考电压电路将需要一段时间稳定下来以后才能可靠地检测低压或高压条件。 HLVD 启动时间 TIRVST 与器件时钟速度无关。它由电气规范中的参数 36 （表 21-10）规定。

直到 TIRVST 结束并且参考电压达到稳定后才会允许 HLVD 中断标志。基于此原因，在此时间间隔期间，超出设置点的偏离可能不会被检测到。请参见图 17-2 或图 17-3。

图 17-2： 低压检测工作原理 （VDIRMAG = 0）

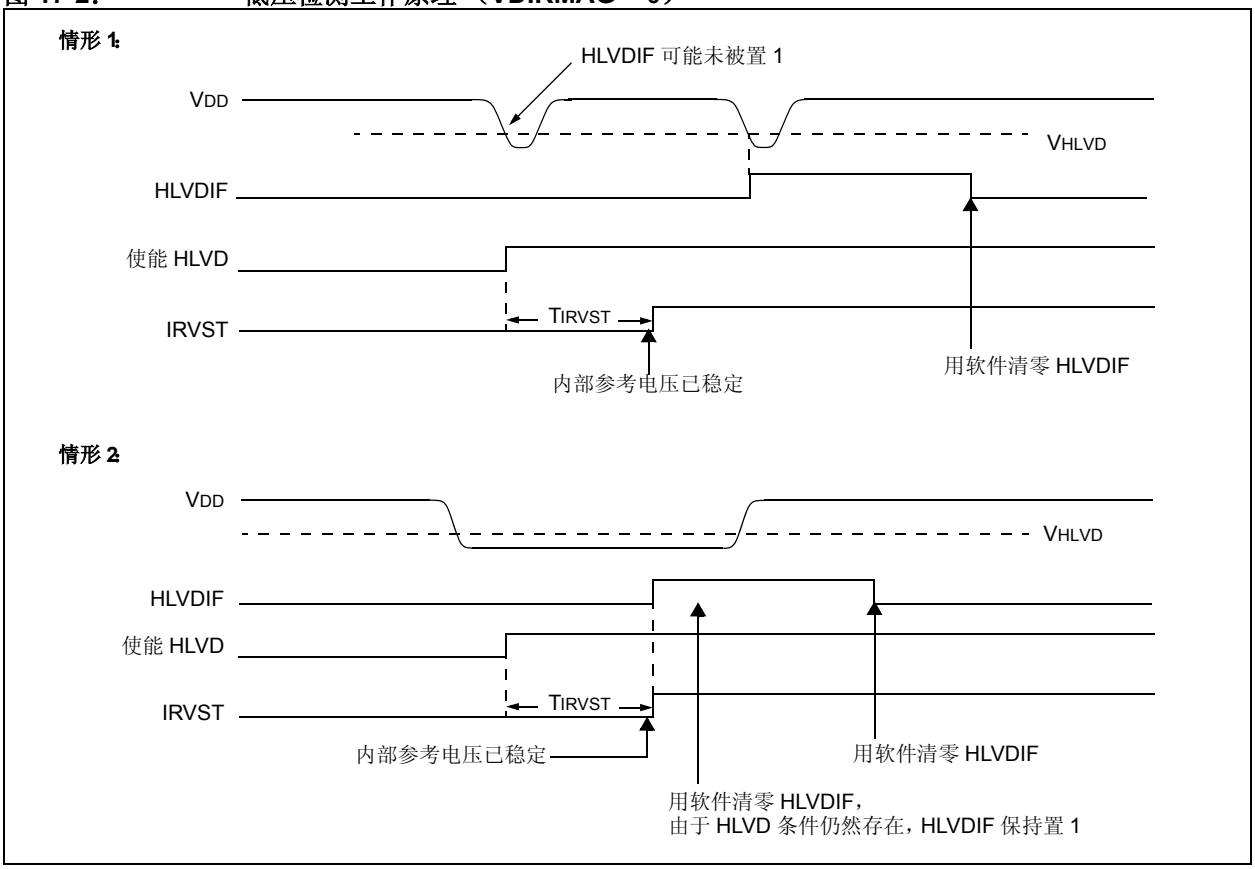
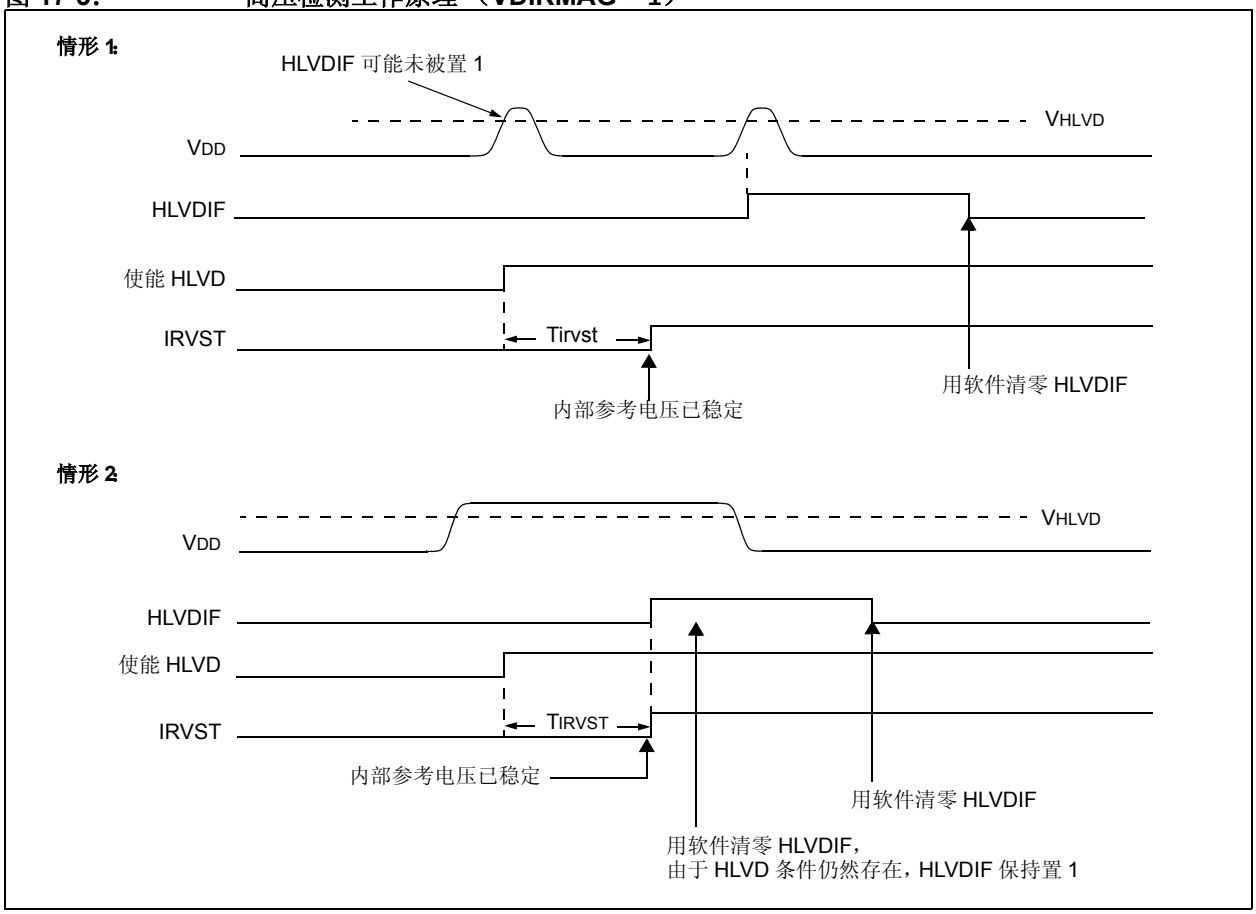


图 17-3：　　高压检测工作原理 （VDIRMAG = 1）

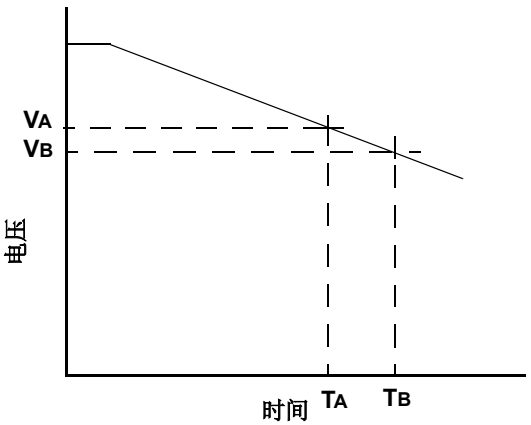


17.5　应用

在许多应用中，当电压低于或高于某个门限值时，系统希望可以检测到该事件。例如，可以定期使能 HLVD 模块来检测是否连接通用串行总线（Universal Serial Bus， USB）。这里假设断开连接时器件的供电电压低于 USB 电压。若连接了 USB，将检测到 3.3V 到 5V 的高压 （USB 上的电压）；如果断开连接，情况正好相反。此功能可以省去一些额外的元件和连接信号（输入引脚）。

对于一般的电池应用，图 17-4 给出了一个近似的电压曲线。器件电压会随时间逐渐下降。当器件电压达到电压 VA 时， HLVD 逻辑电路会在时间 TA 产生中断。该中断将导致执行中断服务子程序，从而使应用程序能在器件电压退出有效工作范围 （对应的时间为 TB）之前执行“日常任务”，并安全关闭。因此，HLVD 将会提供一个时间窗（表示为 TA 和 TB 的时间差）使应用程序能安全地退出。

图 17-4：　　典型高 / 低压检测应用



图注： VA = HLVD 跳变点
VB = 器件的最低有效工作电压

© 2007 Microchip Technology Inc.

17.6 休眠期间的操作

如果被使能时，HLVD 电路在休眠期间将继续工作。如果器件电压越过了跳变点，HLVDIF 位将会被置 1 并且器件将从休眠状态中被唤醒。如果已经允许了全局中断，程序将跳转到中断向量地址处继续执行。

17.7 复位的影响

器件复位将强制所有寄存器进入复位状态。这会强制关闭 HLVD 模块。

表 17-1： 与高 / 低压检测模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
HLVDCON	VDIRMAG	—	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	50
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR2	OSCFIF	—	USBIF	—	—	HLVDIF	—	—	51
PIE2	OSCFIE	—	USBIE	—	—	HLVDIE	—	—	51
IPR2	OSCFIP	—	USBIP	—	—	HLVDIP	—	—	51

图注： – = 未用，读为 0。HLVD 模块不使用阴影单元。

注：

 © 2007 Microchip Technology Inc.

18.0 CPU 的特殊功能

PIC18F2450/4450 器件具有几项特殊的功能旨在最大限度地提高系统可靠性，并通过减少外部元件将成本降至最低。这些功能包括：

- 振荡器选择
- 复位：
  - 上电复位 （POR）
  - 上电延时定时器 （PWRT）
  - 振荡器起振定时器 （OST）
  - 欠压复位 （BOR）
- 中断
- 看门狗定时器 （WDT）
- 故障保护时钟监视器
- 双速启动
- 代码保护
- ID 单元
- 在线串行编程

要根据具体应用对频率、功耗、精度和成本的要求来选择振荡器。在**第 2.0 节 “振荡器配置”**中详细讨论了所有的选项。

在本数据手册的前面几章中已完整地讨论了器件的复位和中断。

除了为复位提供了上电延时定时器和振荡器起振定时器之外，PIC18F2450/4450 器件还提供了一个看门狗定时器，该定时器可通过配置位永久使能或用软件控制 （如果配置为禁止）。

器件自带的内部 RC 振荡器还提供了故障保护时钟监视器（FSCM）和双速启动这两个额外的功能。FSCM 对外设时钟进行后台监视，并在外设时钟发生故障时自动切换时钟源。双速启动使得几乎可在启动发生那一刻立即执行代码，同时主时钟源继续其起振延时。

通过设置相应的配置寄存器位可以使能和配置所有这些功能。

18.1 配置位

可以通过对配置位编程（读为 0）或不编程（读为 1）来选择不同的器件配置。这些配置位被映射到程序存储器以 300000h 开始的单元中。

用户会注意到地址 300000h 超出了用户程序存储器空间范围。事实上，它属于配置存储器空间（300000h – 3FFFFFh），这一空间仅能通过表读和表写进行访问。

对配置寄存器进行编程的方式与对闪存存储器进行编程的方式类似。 EECON1 寄存器中的 WR 位启动自定时写入配置寄存器。在正常操作模式下，带有指向配置寄存器的 TBLPTR 的 TBLWT 指令会建立写配置寄存器要用到的地址和数据。将 WR 位置 1 会启动对配置寄存器的长写操作。配置寄存器一次被写入一个字节。TBLWT 指令可以将 1 或 0 写入单元来改写配置单元的内容。关于闪存编程的更多详细信息，请参见**第 6.5 节“写闪存程序存储器”**。

表 18-1： 配置位和器件 ID

寄存器名称		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	默认 / 未编程值
300000h	CONFIG1L	—	—	USBDIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0	--00 0000
300001h	CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	00-- 0101
300002h	CONFIG2L	—	—	VREGEN	BORV1	BORV0	BOREN1	BOREN0	$\overline{\text{PWRTEN}}$	--01 1111
300003h	CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	---1 1111
300005h	CONFIG3H	MCLRE	—	—	—	—	LPT1OSC	PBADEN	—	1--- -01-
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	XINST	ICPRT(2)	—	BBSIZ	LVP	—	STVREN	100- 01-1
300008h	CONFIG5L	—	—	—	—	—	—	CP1	CP0	---- --11
300009h	CONFIG5H	—	CPB	—	—	—	—	—	—	-1-- ----
30000Ah	CONFIG6L	—	—	—	—	—	—	WRT1	WRT0	---- --11
30000Bh	CONFIG6H	—	WRTB	WRTC	—	—	—	—	—	-11- ----
30000Ch	CONFIG7L	—	—	—	—	—	—	EBTR1	EBTR0	---- --11
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx(1)
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0001 0010(1)

图注： x = 未知， u = 不变， - = 未用。阴影单元未用，读为 0。

注 1： 关于器件 ID 值，请参见寄存器 18-13 和寄存器 18-14。 DEVID 寄存器为只读寄存器，用户不能对其进行编程。

2： 仅在 44 引脚 TQFP 封装的 PIC18F4450 器件上可用。在所有其他器件中始终保持该位清零。

© 2007 Microchip Technology Inc.

寄存器 18-1： CONFIG1L：配置寄存器 1 的低字节 （字节地址为 300000h）

U-0	U-0	R/P-0	R/P-0	R/P-0	R/P-1	R/P-1	R/P-1
—	—	USBDIV	CPUDIV1	CPUDIV0	PLLDIV2	PLLDIV1	PLLDIV0
bit 7							bit 0

图注：		
R = 可读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-6 **未用：**读为 0

bit 5 **USBDIV：**USB 时钟选择位 （仅用在全速 USB 模式； UCFG:FSEN = 1）
1 = USB 时钟源来自 96 MHz PLL 的 2 分频信号
0 = USB 时钟源直接来自主振荡器时钟，无后分频

bit 4-3 **CPUDIV1:CPUDIV0：**系统时钟后分频器选择位
对于 XT、 HS、 EC 和 ECIO 振荡器模式：
11 = 主振荡器 4 分频产生系统时钟
10 = 主振荡器 3 分频产生系统时钟
01 = 主振荡器 2 分频产生系统时钟
00 = 直接用于系统时钟的主振荡器 （无后分频器）
对于 XTPLL、 HSPLL、 ECPLL 和 ECPIO 振荡器模式：
11 = 96 MHz PLL 6 分频产生系统时钟
10 = 96 MHz PLL 4 分频产生系统时钟
01 = 96 MHz PLL 3 分频产生系统时钟
00 = 96 MHz PLL 2 分频产生系统时钟

bit 2-0 **PLLDIV2:PLLDIV0：**PLL 预分频器选择位
111 = 12 分频 （48 MHz 振荡器输入）
110 = 10 分频 （40 MHz 振荡器输入）
101 = 6 分频 （24 MHz 振荡器输入）
100 = 5 分频 （20 MHz 振荡器输入）
011 = 4 分频 （16 MHz 振荡器输入）
010 = 3 分频 （12 MHz 振荡器输入）
001 = 2 分频 （8 MHz 振荡器输入）
000 = 无预分频 （4 MHz 振荡器输入直接驱动 PLL）

寄存器 18-2： CONFIG1H：配置寄存器 1 的高字节 （字节地址为 300001h）

R/P-0	R/P-0	U-0	U-0	R/P-0	R/P-1	R/P-1	R/P-1
IESO	FCMEN	—	—	FOSC3(1)	FOSC2(1)	FOSC1(1)	FOSC0(1)
bit 7							bit 0

图注：		
R = 可读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7 **IESO**：内部 / 外部振荡器切换位
1 = 使能振荡器切换模式
0 = 禁止振荡器切换模式

bit 6 **FCMEN**：故障保护时钟监视器使能位
1 = 使能故障保护时钟监视器
0 = 禁止故障保护时钟监视器

bit 5-4 **未用**：读为 0

bit 3-0 **FOSC3:FOSC0**：振荡器选择位 (1)
111x = HS 振荡器， PLL 使能 （HSPLL）
110x = HS 振荡器 （HS）
1011 = 内部振荡器， USB 使用 HS 振荡器 （INTHS）
1010 = 内部振荡器， USB 使用 XT （INTXT）
1001 = 内部振荡器， RA6 用作 CLKO 引脚， USB 使用 EC （INTCKO）
1000 = 内部振荡器， RA6 用作端口引脚， USB 使用 EC （INTIO）
0111 = EC 振荡器， PLL 使能， RA6 用作 CLKO 引脚 （ECPLL）
0110 = EC 振荡器， PLL 使能， RA6 用作端口引脚 （ECPIO）
0101 = EC 振荡器， RA6 用作 CLKO 引脚 （EC）
0100 = EC 振荡器， RA6 用作端口引脚 （ECIO）
001x = XT 振荡器， PLL 使能 （XTPLL）
000x = XT 振荡器 （XT）

注 1： 单片机和 USB 模块都使用所选振荡器作为它们在 XT、 HS 和 EC 模式下的时钟源。当单片机使用内部振荡器时， USB 模块使用所示的 XT、 HS 或 EC 振荡器作为其时钟源。

© 2007 Microchip Technology Inc.

寄存器 18-3： CONFIG2L：配置寄存器 2 的低字节 （字节地址为 300002h）

U-0	U-0	R/P-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	VREGEN	BORV1(1)	BORV0(1)	BOREN1(2)	BOREN0(2)	$\overline{\text{PWRTEN}}$(2)
bit 7							bit 0

图注：		
R = 可读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-6 **未用**：读为 0

bit 5 **VREGEN**：USB 内部稳压器使能位
1 = 使能 USB 稳压器
0 = 禁止 USB 稳压器

bit 4-3 **BORV1:BORV0**：欠压复位电压位 (1)
11 = 最小设置
.
.
.
00 = 最大设置

bit 2-1 **BOREN1:BOREN0**：欠压复位使能位 (2)
11 = 由硬件使能欠压复位 （禁止 SBOREN）
10 = 由硬件使能欠压复位，休眠模式下被禁止 （禁止 SBOREN）
01 = 由软件使能和控制欠压复位 （使能 SBOREN）
00 = 禁止使用硬件或软件使能欠压复位

bit 0 **$\overline{\text{PWRTEN}}$**：上电延时定时器 （PWRT）使能位 (2)
1 = 禁止 PWRT
0 = 使能 PWRT

注 1： 请参见**第 21.0 节 “电气特性”**获取规范信息。
2： 上电延时定时器与欠压复位是相互独立的，可以分别控制两者的操作。

寄存器 18-4： CONFIG2H：配置寄存器 2 的高字节 （字节地址为 300003h）

U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN
bit 7							bit 0

图注：		
R = 可读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-5 **未用：**读为 0

bit 4-1 **WDTPS3:WDTPS0：**看门狗定时器后分频比选择位
1111 = 1:32,768
1110 = 1:16,384
1101 = 1:8,192
1100 = 1:4,096
1011 = 1:2,048
1010 = 1:1,024
1001 = 1:512
1000 = 1:256
0111 = 1:128
0110 = 1:64
0101 = 1:32
0100 = 1:16
0011 = 1:8
0010 = 1:4
0001 = 1:2
0000 = 1:1

bit 0 **WDTEN：**看门狗定时器使能位
1 = 使能 WDT
0 = 禁止 WDT （由 SWDTEN 位控制）

 © 2007 Microchip Technology Inc.

寄存器 18-5： CONFIG3H：配置寄存器 3 的高字节 （字节地址为 300005h）

R/P-1	U-0	U-0	U-0	U-0	R/P-0	R/P-1	U-0
MCLRE	—	—	—	—	LPT1OSC	PBADEN	—
bit 7							bit 0

图注：		
R = 可读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7 **MCLRE**：$\overline{\text{MCLR}}$ 引脚使能位
1 = 使能$\overline{\text{MCLR}}$ 引脚，禁止 RA5 输入引脚
0 = 使能RA5 输入引脚，禁止 $\overline{\text{MCLR}}$ 引脚

bit 6-3 **未用**：读为 0

bit 2 **LPT1OSC**：低功耗 Timer1 振荡器使能位
1 = Timer1 配置为低功耗运行
0 = Timer1 配置为高功耗运行

bit 1 **PBADEN**：PORTB A/D 使能位
（影响 ADCON1 复位状态。 ADCON1 控制 PORTB<4:0> 引脚配置。）
1 = PORTB<4:0> 引脚在复位时被配置为模拟输入通道
0 = PORTB<4:0> 引脚在复位时被配置为数字 I/O

bit 0 **未用**：读为 0

寄存器 18-6： CONFIG4L：配置寄存器 4 的低字节 （字节地址为 300006h）

R/P-1	R/P-0	R/P-0	U-0	R/P-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$	XINST	ICPRT(1)	—	BBSIZ	LVP	—	STVREN
bit 7							bit 0

图注：		
R = 可读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7 **$\overline{\text{DEBUG}}$：**后台调试器使能位
1 = 禁止后台调试器， RB6 和 RB7 被配置为通用 I/O 引脚
0 = 使能后台调试器， RB6 和 RB7 专用于在线调试

bit 6 **XINST：**扩展指令集使能位
1 = 使能指令集扩展和变址寻址模式
0 = 禁止指令集扩展和变址寻址模式 （传统模式）

bit 5 **ICPRT：**专用在线调试 / 编程端口 （ICPORT）使能位 (1)
1 = 使能 ICPORT
0 = 禁止 ICPORT

bit 4 **未用：**读为 0

bit 3 **BBSIZ：**引导区大小选择位
1 = 2 kW 引导区大小
0 = 1 kW 引导区大小

bit 2 **LVP：**单电源 ICSP™ 使能位
1 = 使能单电源 ICSP
0 = 禁止单电源 ICSP

bit 1 **未用：**读为 0

bit 0 **STVREN：**堆栈满 / 下溢复位使能位
1 = 堆栈满 / 下溢导致复位
0 = 堆栈满 / 下溢不会导致复位

注 1： 仅在 44 引脚 TQFP 封装的 PIC18F4450 器件上可用。在所有其他器件中始终保持该位清零。

© 2007 Microchip Technology Inc.

寄存器 18-7： CONFIG5L：配置寄存器 5 的低字节 （字节地址为 300008h）

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1	R/C-1
—	—	—	—	—	—	CP1	CP0
bit 7							bit 0

图注：		
R = 可读位	C = 可清零位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-2 **未用：**读为 0

bit 1 **CP1：**代码保护位
1 = Block 1 （002000-003FFFh）不受代码保护
0 = Block 1 （002000-003FFFh）受代码保护

bit 0 **CP0：**代码保护位
1 = Block 0 （000800-001FFFh）或 （001000-001FFFh）不受代码保护
0 = Block 0 （000800-001FFFh）或 （001000-001FFFh）受代码保护

寄存器 18-8： CONFIG5H：配置寄存器 5 的高字节 （字节地址为 300009h）

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	CPB	—	—	—	—	—	—
bit 7							bit 0

图注：		
R = 可读位	C = 可清零位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7 **未用：**读为 0

bit 6 **CPB：**引导区代码保护位
1 = 引导区 （000000-0007FFh）或 （000000-000FFFh）不受代码保护
0 = 引导区 （000000-0007FFh）或 （000000-000FFFh）受代码保护

bit 5-0 **未用：**读为 0

寄存器 18-9： CONFIG6L：配置寄存器 6 的低字节 （字节地址为 30000Ah）

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

图注：		
R = 可读位	C = 可清零位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-2 **未用：**读为 0

bit 1 **WRT1：**写保护位
1 = Block 1 （002000-003FFFh）不受写保护
0 = Block 1 （002000-003FFFh）受写保护

bit 0 **WRT0：**写保护位
1 = Block 0 （000800-001FFFh）或 （001000-001FFFh）不受写保护
0 = Block 0 （000800-001FFFh）或 （001000-001FFFh）受写保护

寄存器 18-10： CONFIG6H：配置寄存器 6 的高字节 （字节地址为 30000Bh）

U-0	R/C-1	R-1	U-0	U-0	U-0	U-0	U-0
—	WRTB	WRTC(1)	—	—	—	—	—
bit 7							bit 0

图注：		
R = 可读位	C = 可清零位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7 **未用：**读为 0

bit 6 **WRTB：**引导区写保护位
1 = 引导区 （000000-0007FFh）或 （000000-000FFFh）不受写保护
0 = 引导区 （000000-0007FFh）或 （000000-000FFFh）受写保护

bit 5 **WRTC：**配置寄存器写保护位 (1)
1 = 配置寄存器 （300000-3000FFh）不受写保护
0 = 配置寄存器 （300000-3000FFh）受写保护

bit 4-0 **未用：**读为 0

注 1： 在正常执行模式下，该位是只读的；该位仅在编程模式下可写入。

© 2007 Microchip Technology Inc.

寄存器 18-11： CONFIG7L：配置寄存器 7 的低字节 （字节地址为 30000Ch）

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1	R/C-1
—	—	—	—	—	—	EBTR1	EBTR0
bit 7							bit 0

图注：		
R = 可读位	C = 可清零位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-2 **未用：**读为 0

bit 1 **EBTR1：**表读保护位
1 = 其他块可对 Block 1 （002000-003FFFh）执行表读操作
0 = 禁止其他块对 Block 1 （002000-003FFFh）执行表读操作

bit 0 **EBTR0：**表读保护位
1 = 其他块可对 Block 0 （000800-001FFFh）或 （001000-001FFFh）执行表读操作
0 = 禁止其他块对 Block 0 （000800-001FFFh）或 （001000-001FFFh）执行表读操作

寄存器 18-12： CONFIG7H：配置寄存器 7 的高字节 （字节地址为 30000Dh）

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—
bit 7							bit 0

图注：		
R = 可读位	C = 可清零位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7 **未用：**读为 0

bit 6 **EBTRB：**引导区表读保护位
1 = 其他块可对引导区 （000000-0007FFh）或 （000000-000FFFh）执行表读操作
0 = 禁止其他块对引导区 （000000-0007FFh）或 （000000-000FFFh）执行表读操作

bit 5-0 **未用：**读为 0

寄存器 18-13： DEVID1：PIC18F2450/4450 器件的器件 ID 寄存器 1

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

图注：		
R = 只读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-5 **DEV2:DEV0**：器件 ID 位
001 = PIC18F2450
000 = PIC18F4450

bit 4-0 **REV3:REV0**：版本 ID 位
这些位用于表明器件版本。

寄存器 18-14： DEVID2：PIC18F2450/4450 器件的器件 ID 寄存器 2

R	R	R	R	R	R	R	R
DEV10(1)	DEV9(1)	DEV8(1)	DEV7(1)	DEV6(1)	DEV5(1)	DEV4(1)	DEV3(1)
bit 7							bit 0

图注：		
R = 只读位	P = 可编程位	U = 未用位，读为 0
-n = 未对器件编程时的值		u = 编程后状态不变

bit 7-0 **DEV10:DEV3**：器件 ID 位 (1)
这些位与 DEVID1 寄存器 1 中的 DEV2:DEV0 位一起用于标识器件号。
0010 0100 = PIC18F2450/4450 器件

注 1： DEV10:DEV3 的值可能会用于其他器件。特定器件是通过使用整个 DEV10:DEV0 位序列来标识的。

© 2007 Microchip Technology Inc.

18.2 看门狗定时器 （WDT）

PIC18F2450/4450 器件的 WDT 是由 INTRC 时钟源驱动的。当使能 WDT 时，时钟源也将同时使能。WDT 定时周期的标称值为 4 ms，其稳定性与 INTRC 振荡器相同。

4 ms 的 WDT 周期将与 16 位后分频器的值相乘来得到更长的时间周期。通过配置寄存器 2H 来控制一个多路开关以对 WDT 后分频器的输出进行选择。因此可获得的定时周期范围为 4 ms 至 131.072 秒 （2.18 分钟）。当发生以下任一事件时，WDT 和后分频器将被清零，这些事件包括：执行了 SLEEP 或 CLRWDT 指令，或者发生了时钟故障。

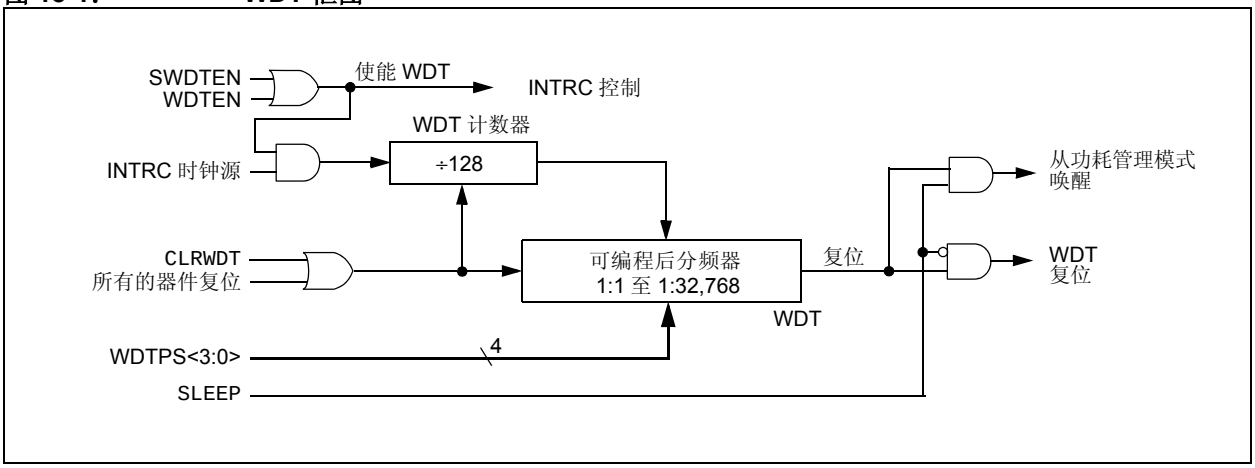
.

注	1：当执行 CLRWDT 和 SLEEP 指令时， WDT 和后分频器的计数值将被清零。 2：当执行 CLRWDT 指令时，后分频器的计数值将被清零。

18.2.1 控制寄存器

寄存器 18-15 表示 WDTCON 寄存器。它是一个可读写的寄存器并包含一个控制位，当WDT被配置为禁止时，该控制位允许使用软件来控制 WDT。

图 18-1： WDT 框图



寄存器 18-15：　　WDTCON：看门狗定时器控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未用位，读为 0	
- n = POR 值	1 = 置 1	0 = 清零	x = 未知

bit 7-1　　**未用：**读为 0

bit 0　　**SWDTEN：**由软件控制的看门狗定时器使能位 (1)
1 = 打开看门狗定时器
0 = 关闭看门狗定时器

注　1：当使能 WDTEN 配置位时该位不起作用。

表 18-2：　　看门狗定时器寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
RCON	IPEN	SBOREN(1)	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	50
WDTCON	—	—	—	—	—	—	—	SWDTEN	50

图注：　– = 未用，读为 0。看门狗定时器不使用阴影单元。
注　1：SBOREN 位仅在 BOREN<1:0> = 01 时可用，否则该位读为 0。

© 2007 Microchip Technology Inc.

18.3 双速启动

双速启动功能允许单片机在主时钟源稳定之前使用INTRC振荡器作为时钟源，从而帮助器件最大限度地缩短从振荡器起振到代码执行之间的延时。通过将 IESO配置位置 1 可使能该功能。

仅当主振荡器模式为 XT、HS、XTPLL 或 HSPLL（基于晶振的模式）时才可使用双速启动。其他时钟源不需要 OST 起振延时；对于这些时钟源，应禁止双速启动。

当使能双速启动时，在上电延时定时器发生超时（使能上电复位）后，器件复位或从休眠模式中被唤醒，此时器件将被配置成使用内部振荡器作为时钟源。这使得在主振荡器起振、OST 运行的同时，代码开始执行。一旦 OST 超时，器件就自动切换到 PRI_RUN 模式。

由于 OSCCON 寄存器在复位事件时被清零，INTRC 时钟直接使用其基础频率。

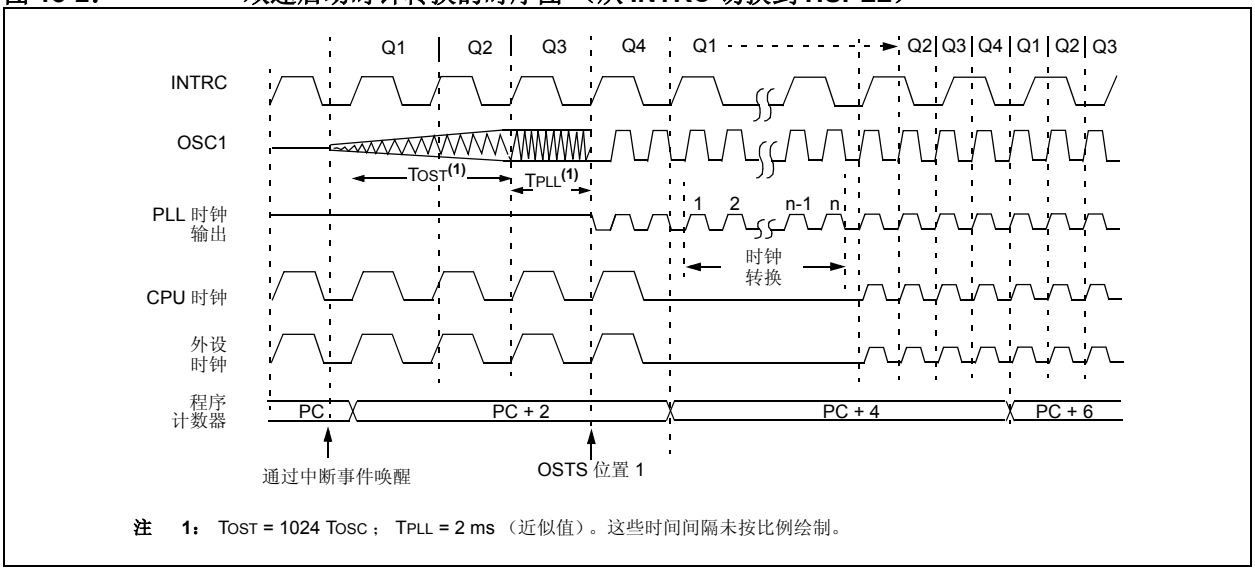
在其他功耗管理模式下，不使用双速启动。器件将使用当前选定的时钟源直到主时钟源可用为止。IESO位的设置被忽略。

18.3.1 使用双速启动时的注意事项

当在双速启动模式中使用 INTRC 振荡器时，器件仍将遵守进入功耗管理模式的正常指令顺序，包括执行多条 SLEEP 指令（见**第 3.1.4 节“多条 SLEEP 命令”**）。实际上，这意味着在 OST 超时前用户代码可以改变 SCS1:SCS0 位的设置或执行 SLEEP 指令。这就使应用程序能短暂地唤醒器件，执行“日常事务”，并在器件开始使用主时钟源前返回休眠状态。

用户代码还能通过检查 OSTS 位（OSCCON<3>）的状态来确定主时钟源是否正在为系统提供时钟。如果该位置 1，则表示主振荡器正在为系统提供时钟。否则，表示当器件复位或从休眠模式被唤醒期间由内部振荡器为系统提供时钟。

图 18-2： 双速启动时钟转换的时序图 （从 INTRC 切换到 HSPLL）



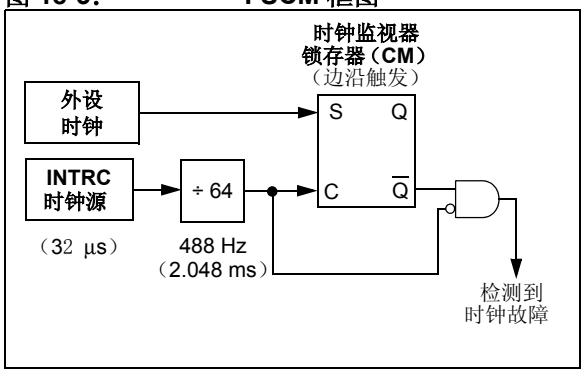
注 1： TOST = 1024 TOSC； TPLL = 2 ms （近似值）。这些时间间隔未按比例绘制。

18.4 故障保护时钟监视器

故障保护时钟监视器 （FSCM）可使单片机在外部时钟发生故障时，自动将系统时钟切换到内部振荡器以保证器件能继续运行。将 FCMEN 配置位置 1 可使能 FSCM 功能。

当使能 FSCM 时，INTRC 振荡器将一直保持运行以监视外设时钟，并且在外设时钟发生故障时作为备用时钟。时钟监视（如图 18-3 所示）通过创建一个采样时钟信号实现，该信号为 INTRC 输出的 64 分频。这样就使得 FSCM 采样时钟脉冲之间有充足的时间间隔，从而保证在此期间至少有一个外设时钟沿出现。外设时钟和采样时钟作为时钟监视锁存器（CM）的输入。CM 在器件时钟源的下降沿被置 1，在采样时钟的上升沿被清零。

图 18-3： FSCM 框图



在采样时钟的下降沿检测外部时钟故障。如果在出现采样时钟的下降沿时， CM 仍置 1，就表示检测到外部时钟故障 （图 18-4）。这将引发以下事件：

- 通过将 OSCFIF （PIR2<7>）置 1，由 FSCM 产生振荡器故障中断；
- 器件时钟源切换为内部振荡器 （OSCCON 不会被更新，因此无法显示当前时钟源——这就是故障保护状态）；并且
- WDT 复位。

FSCM 只能检测出主时钟源或辅助时钟源的故障。如果内部振荡器发生故障，将不会被检测到，当然也不可能采取任何措施。

18.4.1 FSCM 和看门狗定时器

FSCM 和 WDT 均以 INTRC 振荡器作为时钟源。由于 WDT 使用独立的分频器和计数器，使能 FSCM 时，禁止 WDT 对 INTRC 振荡器的运行没有任何影响。

如果使能 WDT 的时候使用的是小预分频值，时钟速度的下降将引起 WDT 超时，随后使器件复位。由于这个原因，故障保护时钟监视器事件也会使 WDT 和后分频器复位，使 WDT 从执行速度发生变化那一刻起开始重新计数，从而避免发生错误超时。

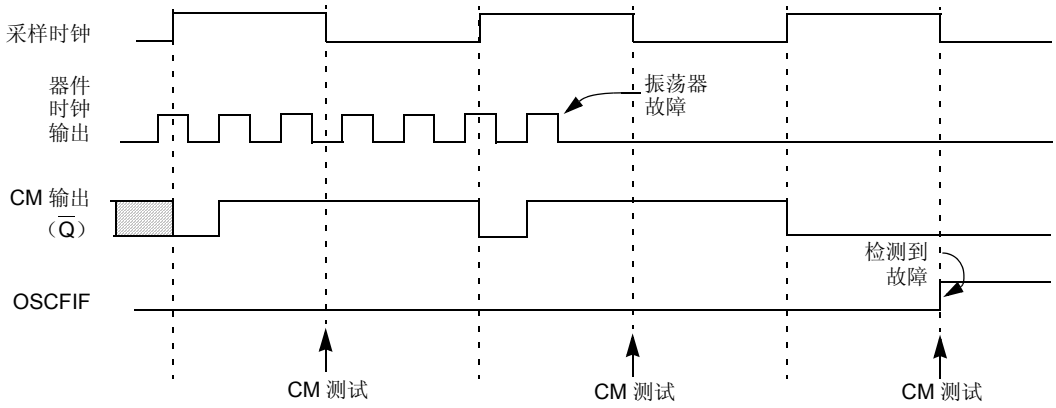
18.4.2 退出故障保护运行模式

器件复位或进入功耗管理模式均可结束故障保护状态。发生复位时，控制器启动在配置寄存器 1H 中指定的主时钟源 （伴有如 OST 或 PLL 定时器等所需的起振延时）。INTRC 将在主时钟源就绪之前提供系统时钟（类似于双速启动）。当主时钟源可用时，系统时钟源将切换为主时钟（OSCCON 寄存器中的 OSTS 位将置 1，表明当前使用的是主时钟源）。然后，故障保护时钟监视器恢复对外设时钟的监视。

在启动期间，主时钟源可能永远不能就绪。在这种情况下，器件运行将以 INTRC 作为时钟源。OSCCON 寄存器将保持复位状态直到进入功耗管理模式为止。

© 2007 Microchip Technology Inc.

图 18-4：　　FSCM 时序图



注：　通常，器件时钟的频率比采样时钟频率高很多。本例中选择了相对频率以便说明。

18.4.3　功耗管理模式下的 FSCM 中断

进入功耗管理模式时，时钟多路开关选择由 OSCCON 寄存器选定的时钟源。在该模式下将恢复对功耗管理时钟源的故障保护时钟监视。

如果在功耗管理运行期间发生了振荡器故障，接下来的操作取决于是否允许了振荡器故障中断。如果允许了（OSCFIF = 1），代码执行将以 INTRC 作为时钟源，并且不会自动转换回发生故障的时钟源。

如果禁止了该中断，处于空闲模式之后产生的中断将使 CPU 开始执行指令，同时以 INTRC 源作为系统时钟源。

18.4.4　POR 或从休眠中唤醒

FSCM 在器件退出上电复位（POR）或低功耗休眠模式后开始检测振荡器故障。当器件主时钟为 EC 或 INTRC 时，监视会在这些事件发生后立即开始。

当振荡器模式使用了晶振或谐振器时（如 HS、HSPLL 或 XT），情况会有些不同。由于这类振荡器需要的起振时间可能比 FCSM 采样时钟的周期长很多，因此可能会检测到假的时钟故障。为了避免这种情况，在此类模式中，内部振荡器会被自动配置为器件时钟并一直工作到主时钟稳定下来为止（OST 和 PLL 定时器发生超时）。这与双速启动模式相同。一旦主时钟稳定下来，INTRC 就将重新作为 FSCM 时钟源。

注：	用于防止在 POR 或从休眠状态唤醒时发生错误中断的逻辑电路，同样也将阻止随后对振荡器故障的检测。通过监视 OSTS 位，并使用定时程序来确定振荡器起振时间是否过长，可避免这个问题。即使如此，在检测到振荡器故障时也不会将振荡器故障中断标志位置 1。

正如**第 18.3.1 节 “使用双速启动时的注意事项”**中所述，在等待系统主时钟稳定的过程中，可以选择另一种时钟配置和另一种功耗管理模式。当选择了新的功耗管理模式时，主时钟将被禁止。

18.5 程序校验和代码保护

PIC18 闪存器件的整个代码保护结构与以往的 PIC® 器件截然不同。

用户程序存储器被分成了三个区域。其中一个存储区是 1 或 2 KB 的引导区。存储器的剩余部分以二进制为单位被分成两个存储区。

这三个块均有与其关联的三个代码保护位。它们是：

- 代码保护位 （CPn）
- 写保护位 （WRTn）
- 外部块表读位 （EBTRn）

图 18-5 给出了 24KB 和 32KB 器件的程序存储器构成以及与每个块关联的特定代码保护位。表 18-3 中总结了这些位的实际地址。

图 18-5： PIC18F2450/4450 的受代码保护的程序存储器

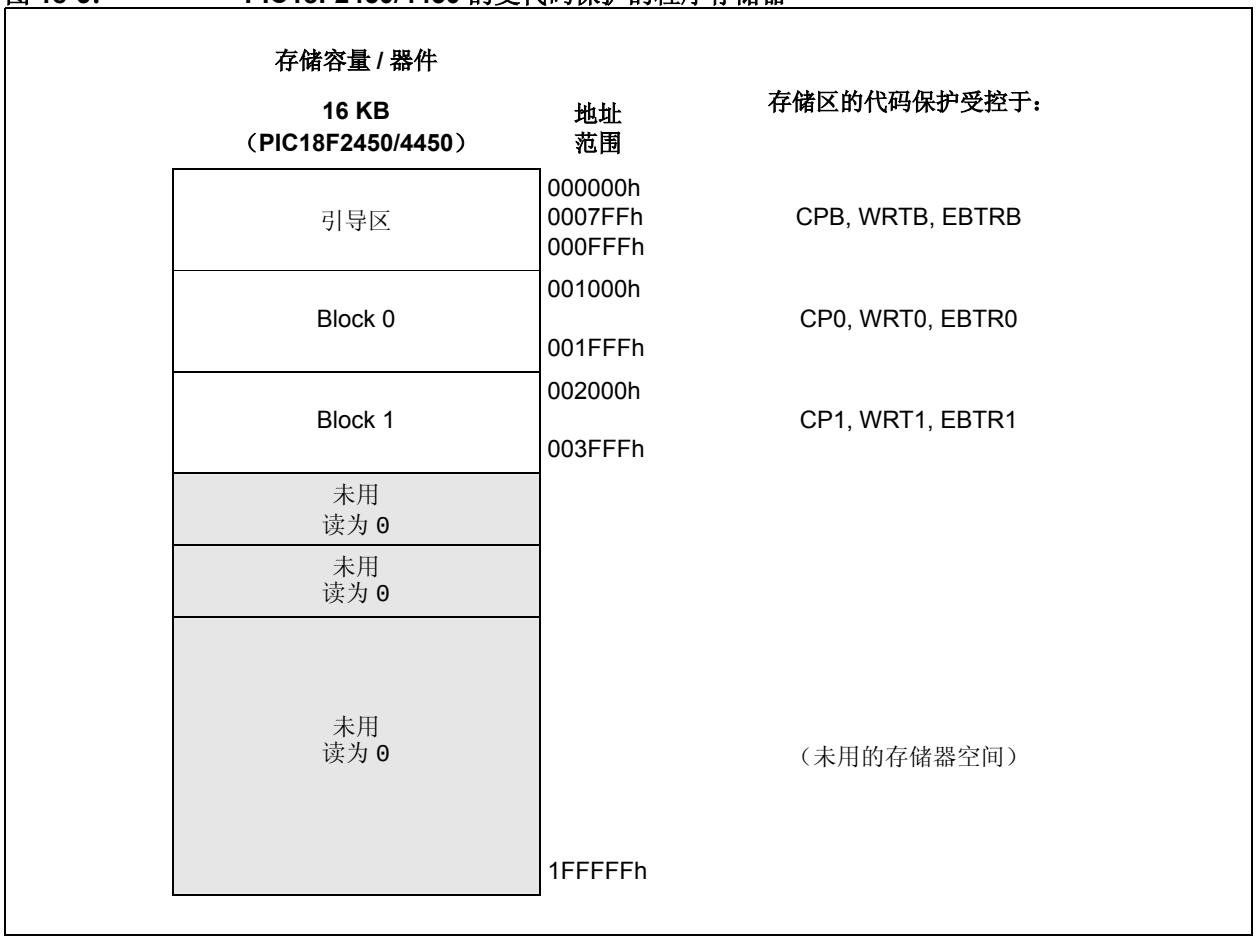


表 18-3： 代码保护寄存器汇总

寄存器名称		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
300008h	CONFIG5L	—	—	—	—	—	—	CP1	CP0
300009h	CONFIG5H	—	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	—	—	—	—	—	—	WRT1	WRT0
30000Bh	CONFIG6H	—	WRTB	WRTC	—	—	—	—	—
30000Ch	CONFIG7L	—	—	—	—	—	—	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

图注： 未使用阴影单元。

© 2007 Microchip Technology Inc.

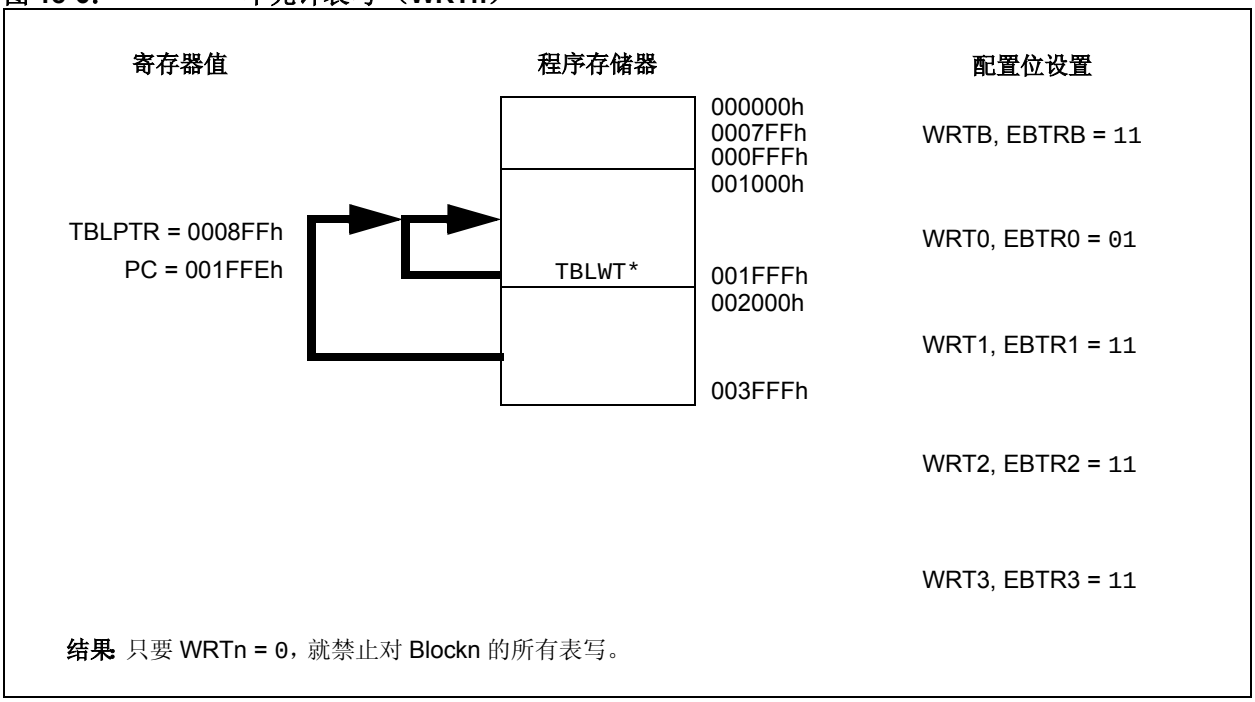
18.5.1 程序存储器代码保护

可使用表读和表写指令从任何存储单元读写程序存储器。器件 ID 可以使用表读指令进行读取。配置寄存器可以通过表读和表写指令进行读写操作。

在正常执行模式下，CPn 位不起任何作用。CPn 位禁止外部读写。如果 WRTn 配置位为 0，则用户存储器模块可被保护不受表写指令的影响。 EBTRn 位控制表读操作。对于 EBTRn 位设置为 0 的用户存储器模块，允许在该模块内执行表读操作。模块以外的存储单元执行的表读指令则不被允许，将导致读为 0。图 207 到 208 例举了表写和表读保护。

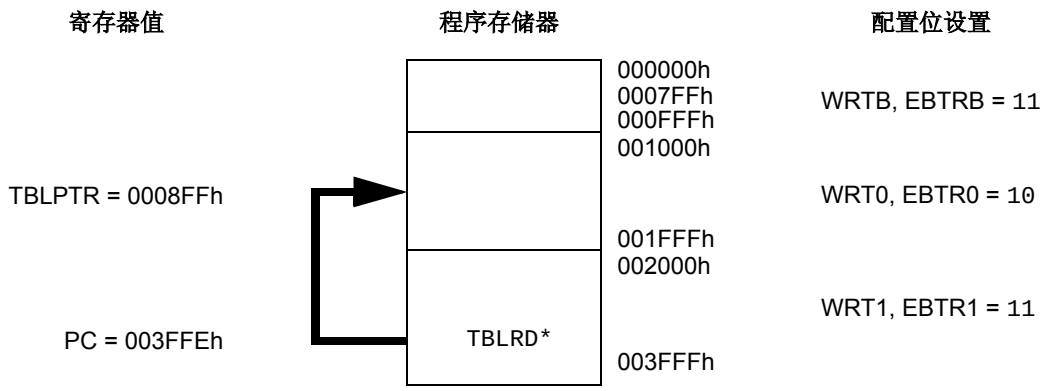
注：	代码保护位只能从 1 改写到 0 状态。不可能将处于 0 状态的位改写为 1。只有通过整片擦除或块擦除功能才能将代码保护位设置为 1。整片擦除和块擦除功能只能通过 ICSP 操作或外部编程器启动。

图 18-6： 不允许表写 （WRTn）



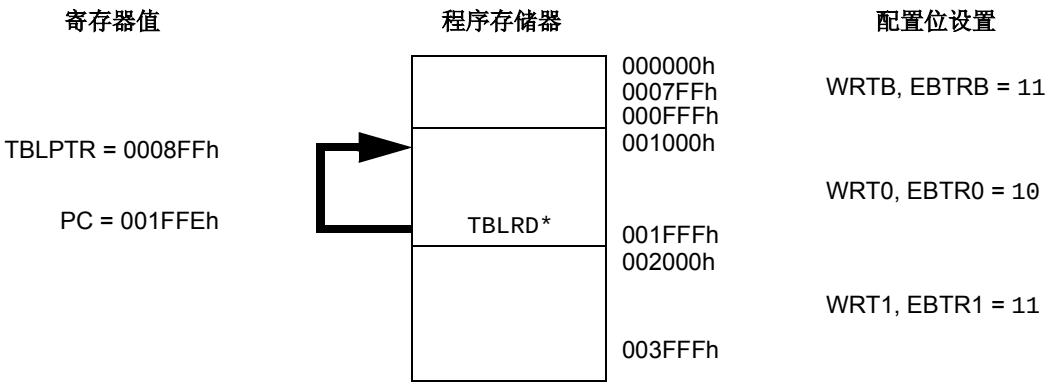
结果: 只要 WRTn = 0，就禁止对 Blockn 的所有表写。

图 18-7：　　不允许外部模块的表读 （EBTRn）



结果: 只要 EBTRn = 0，就禁止来自外部模块对 Blockn 的所有表读。TABLAT 寄存器返回 0。

图 18-8：　　允许外部模块的表读 （EBTRn）



结果: 即使当 EBTRBn = 0 时，也允许在 Blockn 内执行表读。
TABLAT 寄存器返回 TBLPTR 指向的存储单元的数据值。

© 2007 Microchip Technology Inc.

18.5.2 配置寄存器保护

配置寄存器可以是写保护的。WRTC 位控制配置寄存器的保护。在正常执行模式下， WRTC 位是只读的。WRTC 只能通过 ICSP 操作或外部编程器写入。

18.6 ID 地址单元

有 8 个存储器单元 （200000h-200007h）被指定为 ID 单元，供用户存储校验和或其他代码标识。在执行程序时可通过 TBLRD 和 TBLWT 指令读写这些单元；在编程 / 验证时，也可读写这些地址单元。当器件有代码保护时，也可读取 ID 单元。

18.7 在线串行编程

PIC18F2450/4450 单片机可以在最终的应用电路中进行串行编程。只需要 5 根线即可实现这一操作，其中时钟线、数据线各一根，其余 3 根分别是电源线、接地线和编程电压线。这允许用户在制造电路板时使用未编程器件，仅在产品交付之前才对单片机进行编程，从而可以使用最新版本的固件或者定制固件。

18.8 在线调试器

将 $\overline{\text{DEBUG}}$ 配置位编程为 0，可使能在线调试功能。该功能允许使用 MPLAB® IDE 进行一些简单的调试。当使能了单片机的这项功能时，有些资源就不再是通用的了。表 18-4 给出了后台调试器所需的资源。

表 18-4： 调试器资源

I/O 引脚：	RB6 和 RB7
堆栈：	2 级
程序存储器：	512 字节
数据存储器：	10 字节

要使用单片机的在线调试器功能，在设计时就必须实现在线编程模块与 MCLR/VPP/RE3、VDD、VSS、RB7 和 RB6 的正确连接，从而为 Microchip 或第三方开发工具公司提供的在线调试器模块提供交互的接口。

18.9 串行 ICPORT 功能 （仅对指定封装）

在特定情况下， 44 引脚 TQFP 封装的 PIC18F4450 器件的无连接 （NC）引脚可提供额外的功能。这些功能由器件配置位控制，只在该封装类型和引脚数器件上可用。

18.9.1 专用 ICD/ICSP 端口

44 引脚 TQFP 器件可使用 NC 引脚作为在线调试（ICD）和在线串行编程 （ICSP）的备用端口。这些引脚统称为专用 ICSP/ICD 端口，因为它们不与器件的任何其他功能共享。

使用时，专用端口激活三个 NC 引脚，作为备用器件复位、数据和时钟端口。这些端口中没有一个和标准 I/O 引脚重叠，从而可以节省出 I/O 引脚用于用户应用中。

专用的 ICSP/ICD 端口通过将 ICPRT 配置位置 1 来使能。端口功能与 RB6/RB7 上的传统 ICSP/ICD 端口相同。表 18-5 列出了用于 ICSP 和 ICD 的功能相当的引脚。

表 18-5： 传统和专用 ICD/ICSP™ 端口功能引脚

引脚名称		引脚类型	引脚功能
传统端口	专用端口		
$\overline{\text{MCLR}}$/VPP/RE3	NC/$\overline{\text{ICRST}}$/ICVPP	P	器件复位和编程使能
RB6/KBI2/PGC	NC/ICCK/ICPGC	I	串行时钟
RB7/KBI3/PGD	NC/ICDT/ICPGD	I/O	串行数据

图注： I = 输入， O = 输出， P = 电源

即使专用端口使能时， ICSP 和 ICD 功能仍可通过传统端口使用。在 MCLR/VPP/RE3 引脚上观察到 VIH 时，将忽略 ICRST/ICVPP 引脚的状态。

注 1： ICPRT 配置位只能通过默认的 ICSP 端口进行设置。

2： 对所有 28 引脚和 40 引脚器件，ICPRT 配置位必须保持清零；否则可能出现意外的操作。

18.9.2 28 引脚仿真

44 引脚 TQFP 封装的 PIC18F4450 器件在外部控制下也可更改配置，用于调试。这使得器件可以仿真 PIC18F2455/2550 28 引脚器件。

该 28 引脚配置模式通过 NC/ICPORTS 控制。将该引脚连接到 VSS 将迫使器件仿真 28 引脚器件。与 40/44 引脚器件相关的功能连同对应的控制寄存器和位将被禁止。另一方面，将引脚连接到 VDD 将迫使器件工作在其默认配置下。

配置选项仅在后台调试和专用 ICD/ICSP 端口同时使能时可用 （DEBUG 配置位清零， ICPRT 配置位置 1）。禁止时， NC/ICPORTS 是无连接引脚。

18.10 单电源 ICSP 编程

LVP 配置位使能单电源 ICSP 编程（原来称为 *低压 ICSP 编程*或 *LVP*）。当使能单电源编程时，单片机可以在无需对 MCLR/VPP/RE3 引脚施加高压的情况下进行编程，但 RB5/KBI1/PGM 引脚专用于控制编程模式选择，不能再用作通用 I/O 引脚。

使用单电源编程时， VDD 被施加到 MCLR/VPP/RE3 引脚，如同在正常执行模式下一样。要进入编程模式，需要把 VDD 施加到 PGM 引脚。

注 1： 不管 LVP 位的状态如何，只要将 VIHH 施加到 MCLR 引脚，就可以进行高压编程。

2： 在低压 ICSP 编程模式下， RB5 引脚不能再用作通用 I/O 引脚，并且应在正常运行中保持低电平。

3： 使用低压 ICSP 编程（LVP）并在 PORTB 上使能上拉时，TRISB 寄存器中的 bit 5 必须清零，以便禁止 RB5 上拉，从而确保器件正常运行。

4： 如果禁止了器件主复位，应验证执行了以下操作之一，确保能正确进入 ICSP 模式：

a） 禁止低压编程（CONFIG4L<2> = 0）；或者

b） 确保进入 ICSP 时 RB5/KBI1/PGM 保持低电平。

如果不再使用单电源 ICSP 编程模式，则 LVP 位可以被清零。RB5/KBI1/PGM 随后可作为数字 I/O 引脚 RB5。LVP 位仅在使用标准高压编程时可被置 1 或清零（VIHH 被施加到 MCLR/VPP/RE3 引脚）。一旦 LVP 被禁止，只能使用标准高压编程来对器件进行编程。

不受代码保护的存储器可以使用块擦除或逐行擦除进行擦除，然后在任何指定的 VDD 下进行写入。如果要擦除受代码保护的存储器，需要进行块擦除。如果在使用低压编程时执行块擦除，则器件的 VDD 必须为 4.5V 到 5.5V。

 © 2007 Microchip Technology Inc.

19.0 指令集汇总

PIC18F2450/4450 器件具有一个含有 75 条 PIC18 内核指令的标准指令集，和一个含有优化递归或软件堆栈代码的 8 条新指令的扩展指令集。本章后面的部分将讨论扩展指令集。

19.1 标准指令集

标准的 PIC18 指令集与以前的 PIC 指令集相比，添加了很多增强功能，并保持了易于从其他 PIC 指令集移植的特点。大部分指令为单字指令（16 位），只有 4 条指令是双字指令。

每个单字指令都是一个 16 位字，由操作码 （指明指令类型）和一个或多个操作数 （指定指令操作）组成。

整个指令集具有高度的正交性，可以分为以下 4 种基本类型：

- **字节**操作类指令
- **位**操作类指令
- **立即数**操作类指令
- **控制**操作类指令

表 19-2 为 PIC18 指令集汇总，列出了上述四类指令。表 19-1 给出了操作码字段的说明。

大部分**字节**操作类的指令都含有三种操作数：

1. 文件寄存器 （由 “f” 指定）
2. 保存结果的目标寄存器 （由 “d” 指定）
3. 被访问存储器 （由 “a” 指定）

文件寄存器标识符 “f” 指定了指令将会使用哪一个文件寄存器。目标寄存器标识符 “d” 指定了操作结果的存放位置。如果 “d” 为 0，操作结果存入 WREG 寄存器中。如果 “d” 为 1，操作结果存入指令指定的文件寄存器中。

所有**位**操作类指令都含有三种操作数：

1. 文件寄存器 （由 “f” 指定）
2. 文件寄存器中的位 （由 “b” 指定）
3. 被访问存储器 （由 “a” 指定）

位域标识符 “b” 选择操作所影响的位的编号，而文件寄存器标识符 “f” 则代表这些位所在的寄存器编号。

立即数操作指令使用以下操作数：

- 要装入文件寄存器中的立即数 （由 “k” 指定）
- 要装入立即数的 FSR 寄存器 （由 “f” 指定）
- 不需要操作数 （由 “—” 指定）

控制类指令可以使用以下操作数：

- 程序存储器地址 （由 “n” 指定）
- CALL 或 RETURN 指令的模式 （由 “s” 指定）
- 表读和表写指令的模式 （由 “m” 指定）
- 不需要操作数 （由 “—” 指定）

除了 4 条双字指令外，所有的指令都是单字指令。双字指令将所需的信息保存在 32 位中。第二个字的高 4 位都是1。如果第二个字作为一条指令执行，它会执行NOP 指令。

除非条件测试结果为 true 或者指令执行改变了程序计数器的值，否则执行所有的单字指令都只需要一个指令周期。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条 NOP 指令。

执行双字指令需要两个指令周期。

每个指令周期由 4 个振荡器周期组成。因此，如果振荡器频率为 4 MHz，正常的指令执行时间为 1 µs。如果条件测试结果为 true 或指令执行改变了程序计数器的值，则该指令的执行时间为 2 µs。双字跳转指令 （如果为 true）的执行则需要 3 µs。

图 19-1 给出了指令的几种通用格式。所有示例均使用 “nnh” 来表示十六进制数。

指令集汇总（见表 19-2）列出了可被Microchip MPASM™ 汇编器识别的标准指令。

第 19.1.1 节 “标准指令集” 中对每个指令进行了介绍。

表 19-1： 操作码字段说明

字段	说明
a	快速操作 RAM 位 a = 0：快速操作 RAM 内的 RAM 单元 （BSR 寄存器被忽略） a = 1：由 BSR 寄存器指定 RAM 存储区
bbb	8 位文件寄存器内的位地址 （0 到 7）。
BSR	存储区选择寄存器。用于选择当前的 RAM 存储区。
C、 DC、 Z、 OV 和 N	ALU 状态位：进位、辅助进位、全零、溢出和负标志位。
d	目标寄存器选择位 d = 0：结果保存至 WREG 寄存器 d = 1：结果保存至文件寄存器 f
dest	目标寄存器：可以是 WREG 寄存器或指定的文件寄存器地址。
f	8 位文件寄存器地址 （00h 到 FFh），或 2 位 FSR 标识符 （0h 到 3h）。
f_s	12 位文件寄存器地址 （000h 到 FFFh）。这是源地址。
f_d	12 位文件寄存器地址 （000h 到 FFFh）。这是目标地址。
GIE	全局中断允许位。
k	立即数、常数或者标号 （可能是 8 位、 12 位或 20 位的值）
label	标号名称。
mm * *+ *- +*	表读和表写指令的 TBLPTR 寄存器模式。只与表读和表写指令一起使用： 不改变寄存器 （如用于表读和表写的 TBLPTR） 后增寄存器 （如用于表读和表写的 TBLPTR） 后减寄存器 （如用于表读和表写的 TBLPTR） 预增寄存器 （如用于表读和表写的 TBLPTR）
n	相对跳转指令的相对地址 （二进制补码），或 Call/Branch 和 Return 指令的直接地址。
PC	程序计数器。
PCL	程序计数器低字节。
PCH	程序计数器高字节。
PCLATH	程序计数器高字节锁存器。
PCLATU	程序计数器最高字节锁存器。
$\overline{PD}$	掉电位。
PRODH	乘积的高字节。
PRODL	乘积的低字节。
s	快速调用 / 返回模式选择位 s = 0：不对影子寄存器进行更新，也不用影子寄存器的内容更新其他寄存器 s = 1：将寄存器的值存入影子寄存器或把影子寄存器中的值载入寄存器 （快速模式）
TBLPTR	21 位表指针 （指向程序存储器地址）。
TABLAT	8 位表锁存器。
$\overline{TO}$	超时溢出位。
TOS	栈顶。
u	未使用或未改变。
WDT	看门狗定时器。
WREG	工作寄存器 （累加器）。
x	无关位 （0 或 1）。汇编器将产生 x =0 的代码。为了与所有的 Microchip 软件工具兼容，建议使用这种格式。
z_s	对寄存器 （源）进行间接寻址的 7 位偏移量。
z_d	对寄存器 （目标）进行间接寻址的 7 位偏移量。
{ }	可选参数。
[text]	表示变址地址。
(text)	text 的内容。
[expr]<n>	表示由指针 expr 指定的寄存器中的位 n。
→	赋值。
< >	寄存器位域。
∈	表示属于某个集合。
斜体文字	用户定义项 （字体为 Courier）。

 © 2007 Microchip Technology Inc.

图 19-1：　　指令的通用格式

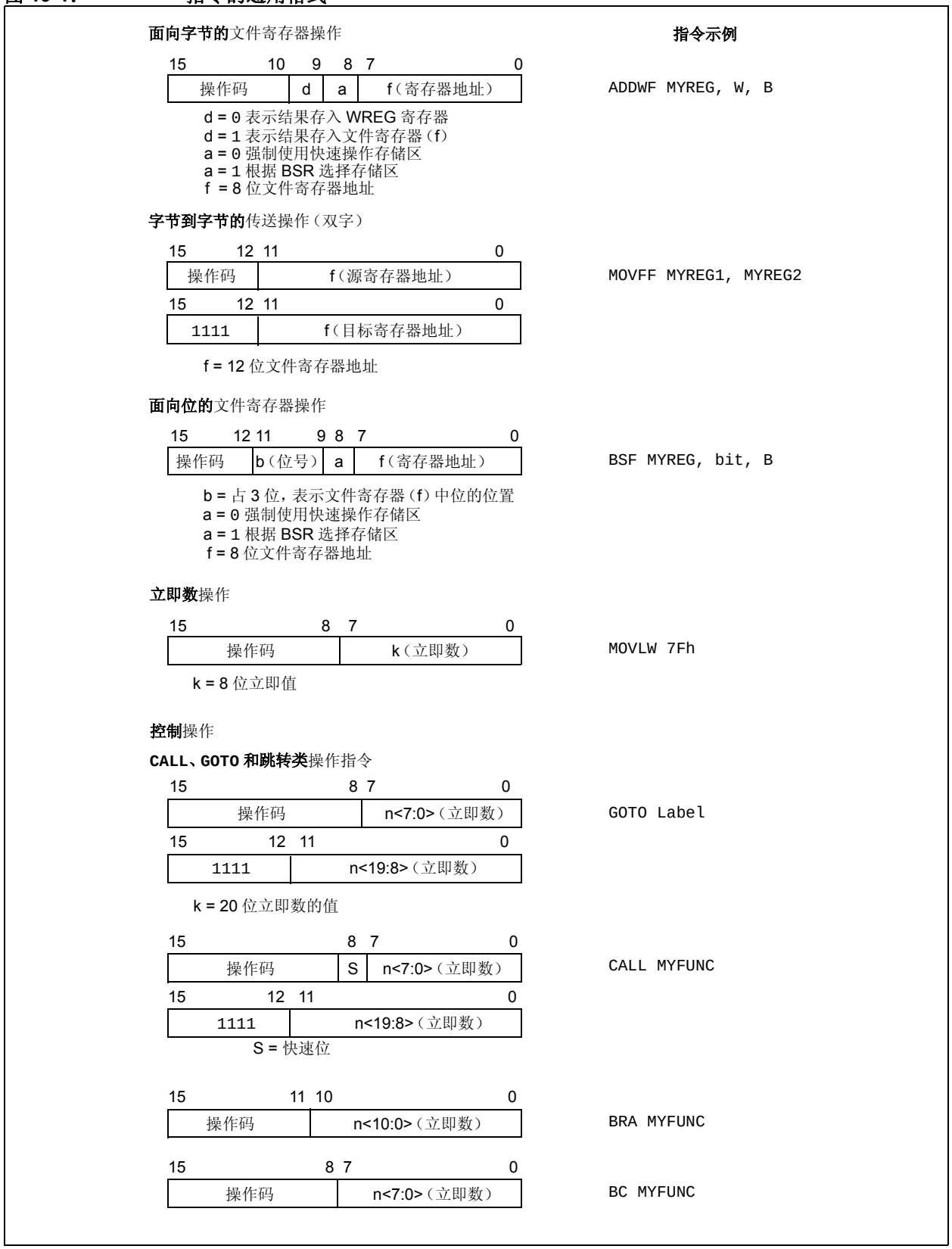


表 19-2： PIC18FXX31 指令集

助记符，操作数		说明	周期	16 位指令字 MSb			LSb	受影响的状态位	注
面向字节的操作类指令									
ADDWF	f, d, a	WREG 与 f 相加	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	1, 2
ADDWFC	f, d, a	WREG 与 f 带进位相加	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	WREG 和 f 作与运算	1	0001	01da	ffff	ffff	Z, N	1,2
CLRF	f, a	将 f 清零	1	0110	101a	ffff	ffff	Z	2
COMF	f, d, a	对 f 取反	1	0001	11da	ffff	ffff	Z, N	1, 2
CPFSEQ	f, a	将 f 与 WREG 作比较，相等则跳过	1 (2或3)	0110	001a	ffff	ffff	无	4
CPFSGT	f, a	将 f 与 WREG 作比较，大于则跳过	1 (2或3)	0110	010a	ffff	ffff	无	4
CPFSLT	f, a	将 f 与 WREG 作比较，小于则跳过	1 (2或3)	0110	000a	ffff	ffff	无	1, 2
DECF	f, d, a	f 减 1	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	f 减 1，为 0 则跳过	1 (2或3)	0010	11da	ffff	ffff	无	1, 2, 3, 4
DCFSNZ	f, d, a	f 减 1，非 0 则跳过	1 (2或3)	0100	11da	ffff	ffff	无	1, 2
INCF	f, d, a	f 加 1	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	f 加 1，为 0 则跳过	1 (2或3)	0011	11da	ffff	ffff	无	4
INFSNZ	f, d, a	f 加 1，非 0 则跳过	1 (2或3)	0100	10da	ffff	ffff	无	1, 2
IORWF	f, d, a	WREG 和 f 作或运算	1	0001	00da	ffff	ffff	Z, N	1, 2
MOVF	f, d, a	移动 f	1	0101	00da	ffff	ffff	Z, N	1
MOVFF	f_s, f_d	f_s （源）地址装入第一个字	2	1100	ffff	ffff	ffff	无	
		f_d （目标）地址装入第二个字		1111	ffff	ffff	ffff		
MOVWF	f, a	将 WREG 移入 f	1	0110	111a	ffff	ffff	无	
MULWF	f, a	WREG 乘以 f	1	0000	001a	ffff	ffff	无	1, 2
NEGF	f, a	对 f 取补	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	
RLCF	f, d, a	f 带进位循环左移	1	0011	01da	ffff	ffff	C, Z, N	1, 2
RLNCF	f, d, a	f 循环左移 （不带进位）	1	0100	01da	ffff	ffff	Z, N	
RRCF	f, d, a	f 带进位循环右移	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF	f, d, a	f 循环右移 （不带进位）	1	0100	00da	ffff	ffff	Z, N	
SETF	f, a	将 f 的内容置为全 1	1	0110	100a	ffff	ffff	无	1, 2
SUBFWB	f, d, a	WREG 减去 f （带借位）	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	
SUBWF	f, d, a	f 减去 WREG	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	1, 2
SUBWFB	f, d, a	f 减去 WREG （带借位）	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	
SWAPF	f, d, a	将 f 中的两个半字节进行交换	1	0011	10da	ffff	ffff	无	4
TSTFSZ	f, a	测试 f，为 0 则跳过	1 (2或3)	0110	011a	ffff	ffff	无	1, 2
XORWF	f, d, a	WREG 和 f 作异或运算	1	0001	10da	ffff	ffff	Z, N	

注 1： PORT 寄存器的值随端口状态的变化而不断修改 （例如， `MOVF PORTB, 1, 0`），修改时使用的值是引脚上的当前值。例如，如果将一引脚配置为输入，其对应数据锁存器中的值为 1，但此时若有外部器件将该引脚驱动为低电平，则被写回数据锁存器的数据值将是 0。

2： 当对 TMR0 寄存器执行该指令 （并且 d = 1）时，如果已为其分配了预分频器，则将该预分频器清零。

3： 如果程序计数器 （PC）被修改或者条件检测为 true，则该指令需要两个周期。第二个周期执行一条 NOP 指令。

4： 某些指令是双字指令。除非指令的第一个字获取这 16 位中包含的信息，否则第二个字将作为 NOP 指令执行。这将确保所有程序存储器单元内存储的都是合法的指令。

© 2007 Microchip Technology Inc.

表 19-2： PIC18FXX31 指令集 （续）

助记符，操作数		说明	周期	16 位指令字 MSb LSb	受影响的状态位	注
面向位的操作类指令						
BCF	f, b, a	将 f 中的某位清零	1	1001 bbba ffff ffff	无	1, 2
BSF	f, b, a	将 f 中的某位置 1	1	1000 bbba ffff ffff	无	1, 2
BTFSC	f, b, a	测试 f 中的某位，为 0 则跳过	1 （2或3）	1011 bbba ffff ffff	无	3, 4
BTFSS	f, b, a	测试 f 中的某位，为 1 则跳过	1 （2或3）	1010 bbba ffff ffff	无	3, 4
BTG	f, d, a	将 f 中的某位取反	1	0111 bbba ffff ffff	无	1, 2
控制类指令						
BC	n	进位则跳转	1 (2)	1110 0010 nnnn nnnn	无	
BN	n	为负则跳转	1 (2)	1110 0110 nnnn nnnn	无	
BNC	n	无进位则跳转	1 (2)	1110 0011 nnnn nnnn	无	
BNN	n	不为负则跳转	1 (2)	1110 0111 nnnn nnnn	无	
BNOV	n	不溢出则跳转	1 (2)	1110 0101 nnnn nnnn	无	
BNZ	n	不为零则跳转	1 (2)	1110 0001 nnnn nnnn	无	
BOV	n	溢出则跳转	1 (2)	1110 0100 nnnn nnnn	无	
BRA	n	无条件跳转	2	1101 0nnn nnnn nnnn	无	
BZ	n	为零则跳转	1 (2)	1110 0000 nnnn nnnn	无	
CALL	n, s	调用子程序 第一个字	2	1110 110s kkkk kkkk	无	
		第二个字		1111 kkkk kkkk kkkk		
CLRWDT	—	将看门狗定时器清零	1	0000 0000 0000 0100	$\overline{TO}$, $\overline{PD}$	
DAW	—	对 WREG 进行十进制调整	1	0000 0000 0000 0111	C	
GOTO	n	跳转到地址 第一个字	2	1110 1111 kkkk kkkk	无	
		第二个字		1111 kkkk kkkk kkkk		
NOP	—	空操作	1	0000 0000 0000 0000	无	
NOP	—	空操作	1	1111 xxxx xxxx xxxx	无	4
POP	—	返回堆栈栈顶 （TOS）出栈	1	0000 0000 0000 0110	无	
PUSH	—	返回堆栈栈顶 （TOS）进栈	1	0000 0000 0000 0101	无	
RCALL	n	相对调用	2	1101 1nnn nnnn nnnn	无	
RESET		用软件使器件复位	1	0000 0000 1111 1111	全部	
RETFIE	s	中断返回允许	2	0000 0000 0001 000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	返回时将立即数送入 WREG	2	0000 1100 kkkk kkkk	无	
RETURN	s	从子程序返回	2	0000 0000 0001 001s	无	
SLEEP	—	进入待机模式	1	0000 0000 0000 0011	$\overline{TO}$, $\overline{PD}$	

注 1： PORT 寄存器的值随端口状态的变化而不断修改 （例如， MOVF PORTB, 1, 0），修改时使用的值是引脚上的当前值。例如，如果将一引脚配置为输入，其对应数据锁存器中的值为 1，但此时若有外部器件将该引脚驱动为低电平，则被写回数据锁存器的数据值将是 0。

2： 当对 TMR0 寄存器执行该指令 （并且 d = 1）时，如果已为其分配了预分频器，则将该预分频器清零。

3： 如果程序计数器 （PC）被修改或者条件检测为 true，则该指令需要两个周期。第二个周期执行一条 NOP 指令。

4： 某些指令是双字指令。除非指令的第一个字获取这 16 位中包含的信息，否则第二个字将作为 NOP 指令执行。这将确保所有程序存储器单元内存储的都是合法的指令。

表 19-2： PIC18FXX31 指令集 （续）

助记符，操作数		说明	周期	16 位指令字 MSb LSb	受影响的状态位	注
立即数操作类指令						
ADDLW	k	WREG 与立即数相加	1	0000 1111 kkkk kkkk	C, DC, Z, OV, N	
ANDLW	k	WREG 和立即数进行与运算	1	0000 1011 kkkk kkkk	Z, N	
IORLW	k	WREG 和立即数进行或运算	1	0000 1001 kkkk kkkk	Z, N	
LFSR	f, k	移动立即数 （12 位）第二个字到 FSR （f） 第一个字	2	1110 1110 00ff kkkk 1111 0000 kkkk kkkk	无	
MOVLB	k	将立即数移入 BSR<3:0>	1	0000 0001 0000 kkkk	无	
MOVLW	k	将立即数移入 WREG	1	0000 1110 kkkk kkkk	无	
MULLW	k	WREG 和立即数相乘	1	0000 1101 kkkk kkkk	无	
RETLW	k	返回时将立即数送入 WREG	2	0000 1100 kkkk kkkk	无	
SUBLW	k	立即数减去 WREG	1	0000 1000 kkkk kkkk	C, DC, Z, OV, N	
XORLW	k	WREG 和立即数进行异或运算	1	0000 1010 kkkk kkkk	Z, N	
数据存储器 ↔ 程序存储器操作						
TBLRD*		表读	2	0000 0000 0000 1000	无	
TBLRD*+		后增表读		0000 0000 0000 1001	无	
TBLRD*-		后减表读		0000 0000 0000 1010	无	
TBLRD+*		预增表读		0000 0000 0000 1011	无	
TBLWT*		表写	2	0000 0000 0000 1100	无	
TBLWT*+		后增表写		0000 0000 0000 1101	无	
TBLWT*-		后减表写		0000 0000 0000 1110	无	
TBLWT+*		预增表写		0000 0000 0000 1111	无	

注 1： PORT 寄存器的值随端口状态的变化而不断修改 （例如， MOVF PORTB, 1, 0），修改时使用的值是引脚上的当前值。例如，如果将一引脚配置为输入，其对应数据锁存器中的值为 1，但此时若有外部器件将该引脚驱动为低电平，则被写回数据锁存器的数据值将是 0。

2： 当对 TMR0 寄存器执行该指令 （并且 d = 1）时，如果已为其分配了预分频器，则将该预分频器清零。

3： 如果程序计数器 （PC）被修改或者条件检测为 true，则该指令需要两个周期。第二个周期执行一条 NOP 指令。

4： 某些指令是双字指令。除非指令的第一个字获取这 16 位中包含的信息，否则第二个字将作为 NOP 指令执行。这将确保所有程序存储器单元内存储的都是合法的指令。

© 2007 Microchip Technology Inc.

19.1.1 标准指令集

ADDLW **W 与立即数相加**

语法： ADDLW k

操作数： 0 ≤ k ≤ 255

操作： (W) + k → W

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0000	1111	kkkk	kkkk

说明： 将 W 寄存器的内容与 8 位立即数 k 相加，结果保存在 W 寄存器。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例： ADDLW 15h

执行指令前
W = 10h
执行指令后
W = 25h

ADDWF **W 与 f 寄存器相加**

语法： ADDWF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (W) + (f) → dest

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0010	01da	ffff	ffff

说明： 将 W 的内容与 f 寄存器的内容相加。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： ADDWF REG, 0, 0

执行指令前
W = 17h
REG = 0C2h
执行指令后
W = 0D9h
REG = 0C2h

注： 所有的 PIC18 指令都可能在其指令助记符之前使用可选的标号参数，用于符号寻址。如果使用了标号，那么指令语法将变为：{label} 指令参数。

ADDWFC **W 与 f 带进位相加**

语法： ADDWFC f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (W) + (f) + (C) → dest

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0010	00da	ffff	ffff

说明： 将 W 的内容、进位标志位与寄存器 f 的内容相加。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存储在寄存器 f 中。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： ADDWFC REG, 0, 1

执行指令前
进位标志位= 1
REG = 02h
W = 4Dh
执行指令后
进位标志位= 0
REG = 02h
W = 50h

ANDLW **立即数和 W 寄存器作逻辑与运算**

语法： ANDLW k

操作数： 0 ≤ k ≤ 255

操作： (W) .AND. k → W

受影响的状态位： N 和 Z

机器码：

0000	1011	kkkk	kkkk

说明： 将 W 的内容与 8 位立即数 k 进行逻辑与运算。结果存储在 W 寄存器中。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例： ANDLW 05Fh

执行指令前
W = A3h
执行指令后
W = 03h

© 2007 Microchip Technology Inc.

ANDWF **将 W 和 f 作逻辑与运算**

语法： ANDWF f {,d {,a}}

操作数： $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

操作： (W) .AND.(f) → dest

受影响的状态位： N 和 Z

机器码：

0001	01da	ffff	ffff

说明： 将 W 的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 $f \le 95$（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例：
```
ANDWF   REG, 0, 0
```
执行指令前
W = 17h
REG = C2h
执行指令后
W = 02h
REG = C2h

BC **进位则跳转**

语法： BC n

操作数： $-128 \le n \le 127$

操作： 如果进位标志位为 1
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0010	nnnn	nnnn

说明： 如果进位标志位为 1，程序将跳转。
二进制补码 “2n”与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例：
```
HERE    BC   5
```
执行指令前
PC = 地址 （HERE）
执行指令后
如果进位标志位= 1；
PC = 地址 （HERE + 12）
如果进位标志位= 0；
PC = 地址 （HERE + 2）

BCF **将 f 寄存器中的某位清零**

语法： BCF f, b {,a}

操作数： 0 ≤ f ≤ 255
0 ≤ b ≤ 7
a ∈ [0,1]

操作： 0 → f<b>

受影响的状态位： 无

机器码：

1001	bbba	ffff	ffff

说明： 将寄存器 f 中的位 b 清零。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例： BCF FLAG_REG, 7, 0

执行指令前
FLAG_REG = C7h
执行指令后
FLAG_REG = 47h

BN **为负则跳转**

语法： BN n

操作数： -128 ≤ n ≤ 127

操作： 如果负标志位为 1
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0110	nnnn	nnnn

说明： 如果负标志位为 1，程序将跳转。
二进制补码 “2n” 与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例： HERE BN Jump

执行指令前
PC = 地址 （HERE）
执行指令后
如果负标志位 = 1；
PC = 地址 （Jump）
如果负标志位 = 0；
PC = 地址 （HERE + 2）

© 2007 Microchip Technology Inc.

BNC **无进位则跳转**

语法： BNC n

操作数： -128 ≤ n ≤ 127

操作： 如果进位标志位为 0
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0011	nnnn	nnnn

说明： 如果进位标志位为 0，程序将跳转。
二进制补码 “2n”与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例： HERE BNC Jump

执行指令前
PC = 地址 （HERE）
执行指令后
如果进位标志位= 0；
PC = 地址 （Jump）
如果进位标志位= 1；
PC = 地址 （HERE +2）

BNN **不为负则跳转**

语法： BNN n

操作数： -128 ≤ n ≤ 127

操作： 如果负标志位为 0
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0111	nnnn	nnnn

说明： 如果负标志位为 0，程序将跳转。
二进制补码 “2n”与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例： HERE BNN Jump

执行指令前
PC = 地址 （HERE）
执行指令后
如果负标志位 = 0；
PC = 地址 （Jump）
如果负标志位 = 1；
PC = 地址 （HERE +2）

BNOV **不溢出则跳转**

语法： BNOV n

操作数： -128 ≤ n ≤ 127

操作： 如果溢出标志位为 0
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0101	nnnn	nnnn

说明： 如果溢出标志位为 0，程序将跳转。
二进制补码 “2n”与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例： HERE BNOV Jump

执行指令前
PC = 地址 （HERE）
执行指令后
如果溢出标志位= 0；
PC = 地址 （Jump）
如果溢出标志位= 1；
PC = 地址 （HERE + 2）

BNZ **不为零则跳转**

语法： BNZ n

操作数： -128 ≤ n ≤ 127

操作： 如果全零标志位为 0
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0001	nnnn	nnnn

说明： 如果全零标志位为 0，程序将跳转。
二进制补码 “2n”与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例： HERE BNZ Jump

执行指令前
PC = 地址 （HERE）
执行指令后
如果全零标志位= 0；
PC = 地址 （Jump）
如果全零标志位= 1；
PC = 地址 （HERE + 2）

© 2007 Microchip Technology Inc.

BRA **无条件跳转**

语法： BRA n

操作数： -1024 ≤ n ≤ 1023

操作： (PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1101	0nnn	nnnn	nnnn

说明： 二进制补码 “2n”与 PC 相加。因为 PC 要先递增才能取下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

示例： HERE BRA Jump

执行指令前
PC = 地址 （HERE）
执行指令后
PC = 地址 （Jump）

BSF **将 f 寄存器中的某位置 1**

语法： BSF f, b {,a}

操作数： 0 ≤ f ≤ 255
0 ≤ b ≤ 7
a ∈ [0,1]

操作： 1 → f<b>

受影响的状态位： 无

机器码：

1000	bbba	ffff	ffff

说明： 将寄存器 f 的位 b 置 1。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例： BSF FLAG_REG, 7, 1

执行指令前
FLAG_REG = 0Ah
执行指令后
FLAG_REG = 8Ah

BTFSC **测试寄存器中的位，为 0 则跳过**

语法： BTFSC f, b {,a}

操作数： 0 ≤ f ≤ 255
0 ≤ b ≤ 7
a ∈ [0,1]

操作： 如果 (f<b>) = 0，则跳过

受影响的状态位： 无

机器码：

1011	bbba	ffff	ffff

说明： 如果寄存器 f 的位 b 为 0，则跳过下一条指令。即在 b 位为 0 时，丢弃下一条指令（执行当前指令期间取的指令）而执行一条 NOP 指令，使该指令变成双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 BTFSC 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

```
HERE    BTFSC   FLAG, 1, 0
FALSE   :
TRUE    :
```

执行指令前
PC = 地址（HERE）
执行指令后
如果 FLAG<1> = 0；
PC = 地址（TRUE）
如果 FLAG<1> = 1；
PC = 地址（FALSE）

BTFSS **测试寄存器中的位，为 1 则跳过**

语法： BTFSS f, b {,a}

操作数： 0 ≤ f ≤ 255
0 ≤ b < 7
a ∈ [0,1]

操作： 如果 (f<b>) = 1，则跳过

受影响的状态位： 无

机器码：

1010	bbba	ffff	ffff

说明： 如果寄存器 f 的位 b 为 1，则跳过下一条指令。即在 b 位为 1 时，丢弃下一条指令（执行当前指令期间取的指令）而执行一条 NOP 指令，使该指令变成双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 BTFSS 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

```
HERE    BTFSS   FLAG, 1, 0
FALSE   :
TRUE    :
```

执行指令前
PC = 地址（HERE）
执行指令后
如果 FLAG<1> = 0；
PC = 地址（FALSE）
如果 FLAG<1> = 1；
PC = 地址（TRUE）

© 2007 Microchip Technology Inc.

BTG **将 f 中的某位取反**

语法： BTG f, b {,a}

操作数： 0 ≤ f ≤ 255
0 ≤ b < 7
a ∈ [0,1]

操作： $(\overline{f<b>}) \rightarrow f<b>$

受影响的状态位： 无

机器码：

0111	bbba	ffff	ffff

说明： 将数据存储单元 f 中的位 b 取反。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例：
```
BTG     PORTC,  4, 0
```
执行指令前：
```
PORTC  =   0111 0101 [75h]
```
执行指令后：
```
PORTC  =   0110 0101 [65h]
```

BOV **溢出则跳转**

语法： BOV n

操作数： -128 ≤ n ≤ 127

操作： 如果溢出标志位为 1
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0100	nnnn	nnnn

说明： 如果溢出标志位为 1，程序将跳转。
二进制补码 “2n”与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例：
```
HERE    BOV   Jump
```
执行指令前
```
PC          =   地址 （HERE）
```
执行指令后
```
如果溢出标志位=   1；
PC          =   地址 （Jump）
如果溢出标志位=   0；
PC          =   地址 （HERE + 2）
```

BZ **为零则跳转**

语法： BZ n

操作数： -128 ≤ n ≤ 127

操作： 如果全零标志位为 1
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1110	0000	nnnn	nnnn

说明： 如果全零标志位为 1，程序将跳转。
二进制补码 “2n”与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 1(2)

Q 周期操作：
如果跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转：

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例： HERE BZ Jump

执行指令前
PC = 地址 （HERE）
执行指令后
如果全零标志位= 1；
PC = 地址 （Jump）
如果全零标志位= 0；
PC = 地址 （HERE + 2）

CALL **调用子程序**

语法： CALL k {,s}

操作数： 0 ≤ k ≤ 1048575
s ∈ [0,1]

操作： (PC) + 4 → TOS，
k → PC<20:1>，
如果 s = 1
(W) → WS，
(STATUS) → STATUSS，
(BSR) → BSRS

受影响的状态位： 无

机器码：

第一个字 （k<7:0>）	1110	110s	$k_7$kkk	kkkk$_0$
第二个字 （k<19:8>）	1111	k_{19}kkk	kkkk	kkkk$_8$

说明： 可在整个 2MB 的存储器范围内进行子程序调用。首先，将返回地址 （PC + 4）压入返回堆栈。如果 s = 1，还会将 W、STATUS 和 BSR 寄存器的内容存入它们各自的影子寄存器 WS、 STATUSS 和 BSRS。如果 s = 0，将不会进行任何更新 （默认）。然后，将 20 位的值 k 装入 PC<20:1>。 CALL 是一条双周期指令。

指令字数： 2

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	将 PC 压入堆栈	读立即数 k<19:8>，写入 PC
空操作	空操作	空操作	空操作

示例： HERE CALL THERE,1

执行指令前
PC = 地址 （HERE）
执行指令后
PC = 地址 （THERE）
TOS = 地址 （HERE + 4）
WS = W
BSRS = BSR
STATUSS = STATUS

© 2007 Microchip Technology Inc.

CLRF **将 f 清零**

语法： CLRF f {,a}

操作数： $0 \le f \le 255$
$a \in [0,1]$

操作： 000h → f，
1 → Z

受影响的状态位： Z

机器码：

0110	101a	ffff	ffff

说明： 清零指定寄存器的内容。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例： CLRF FLAG_REG,1

执行指令前
FLAG_REG = 5Ah
执行指令后
FLAG_REG = 00h

CLRWDT **将看门狗定时器清零**

语法： CLRWDT

操作数： 无

操作： 000h → WDT，
000h → WDT 后分频器，
1 → $\overline{TO}$，
1 → $\overline{PD}$

受影响的状态位： $\overline{TO}$ 和 $\overline{PD}$

机器码：

0000	0000	0000	0100

说明： CLRWDT指令复位看门狗定时器及其后分频器。状态位 $\overline{TO}$ 和 $\overline{PD}$ 置 1。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	处理数据	空操作

示例： CLRWDT

执行指令前
WDT 计数器 = ?
执行指令后
WDT 计数器 = 00h
WDT 后分频器 = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **对 f 取反**

语法： COMF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： $(\bar{f}) \rightarrow$ dest

受影响的状态位： N 和 Z

机器码：

0001	11da	ffff	ffff

说明： 将寄存器 f 的内容取反。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： COMF REG, 0, 0

执行指令前
REG = 13h
执行指令后
REG = 13h
W = ECh

CPFSEQ **比较 f 和 W，如果 f = W 则跳过**

语法： CPFSEQ f {,a}

操作数： 0 ≤ f ≤ 255
a ∈ [0,1]

操作： (f) – (W)，
如果 (f) = (W)，则跳过 （无符号比较）

受影响的状态位： 无

机器码：

0110	001a	ffff	ffff

说明： 通过执行无符号的减法，将数据存储单元 f 的内容与 W 的内容作比较。
如果 f = W，则所取的指令被丢弃并执行一条 NOP 指令，使这个指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 CPFSEQ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：
```
HERE    CPFSEQ REG, 0
NEQUAL  :
EQUAL   :
```

执行指令前
PC 地址 = HERE
W = ?
REG = ?
执行指令后
如果 REG = W；
PC = 地址 （EQUAL）
如果 REG ≠ W；
PC = 地址 （NEQUAL）

© 2007 Microchip Technology Inc.

CPFSGT **比较 f 和 W，如果 f > W 则跳过**

语法： CPFSGT f {,a}

操作数： 0 ≤ f ≤ 255
a ∈ [0,1]

操作： (f) – (W)，
如果 (f) > (W)，则跳过 （无符号比较）

受影响的状态位： 无

机器码：

0110	010a	ffff	ffff

说明： 通过执行无符号的减法，将数据存储单元 f 的内容与 W 的内容作比较。
如果 f > W ，则所取的指令被丢弃并执行一条 NOP 指令，使这个指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 CPFSGT 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：
```
HERE      CPFSGT REG, 0
NGREATER  :
GREATER   :
```

执行指令前
PC = 地址 （HERE）
W = ?
执行指令后
如果 REG > W；
PC = 地址 （GREATER）
如果 REG ≤ W；
PC = 地址 （NGREATER）

CPFSLT **比较 f 和 W，如果 f < W 则跳过**

语法： CPFSLT f {,a}

操作数： 0 ≤ f ≤ 255
a ∈ [0,1]

操作： (f) – (W)，
如果 (f) < (W)，则跳过 （无符号比较）

受影响的状态位： 无

机器码：

0110	000a	ffff	ffff

说明： 通过执行无符号的减法，将数据存储单元 f 的内容与 W 的内容作比较。
如果 f < W ，则所取的指令被丢弃并执行一条 NOP 指令，使这个指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 CPFSLT 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：
```
HERE    CPFSLT REG, 1
NLESS   :
LESS    :
```

执行指令前
PC = 地址 （HERE）
W = ?
执行指令后
如果 REG < W；
PC = 地址 （LESS）
如果 REG ≥ W；
PC = 地址 （NLESS）

DAW **对 W 寄存器进行十进制调整**

语法： DAW

操作数： 无

操作： 如果 [W<3:0> > 9] 或 [DC = 1]，则
(W<3:0>) + 6 → W<3:0>；
否则 (W<3:0>) → W<3:0>

如果 [W<7:4> + DC > 9] 或 [C = 1]，则
(W<7:4>) + 6 + DC → W<7:4>；
否则 (W<7:4>) + DC → W<7:4>

受影响的状态位： C

机器码：

0000	0000	0000	0111

说明： DAW指令调整W寄存器内的8位值，即前两个压缩 BCD 格式的变量之和，并产生一个正确的压缩 BCD 格式结果。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 W	处理数据	写 W

例 1： DAW

执行指令前
W = A5h
C = 0
DC = 0
执行指令后
W = 05h
C = 1
DC = 0

例 2：

执行指令前
W = CEh
C = 0
DC = 0
执行指令后
W = 34h
C = 1
DC = 0

DECF **f 减 1**

语法： DECF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) – 1 → dest

受影响的状态位： C、 DC、 N、 OV 和 Z

机器码：

0000	01da	ffff	ffff

说明： 将寄存器 f 的内容减 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： DECF CNT, 1, 0

执行指令前
CNT = 01h
Z = 0
执行指令后
CNT = 00h
Z = 1

 © 2007 Microchip Technology Inc.

DECFSZ **f 减 1，为 0 则跳过**

语法： DECFSZ f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) – 1 → dest，
如果结果 = 0 则跳过

受影响的状态位： 无

机器码：

0010	11da	ffff	ffff

说明： 将寄存器 f 的内容减 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果结果为 0，则丢弃已取的指令而执行一条 NOP 指令，使该指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 DECFSZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

```
HERE    DECFSZ  CNT, 1, 1
        GOTO    LOOP
CONTINUE
```

执行指令前
PC = 地址 （HERE）
执行指令后
CNT = CNT – 1
如果 CNT = 0；
PC = 地址 （CONTINUE）
如果 CNT ≠ 0；
PC = 地址 （HERE + 2）

DCFSNZ **f 减 1，非 0 则跳过**

语法： DCFSNZ f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) – 1 → dest，
如果结果 ≠ 0 则跳过

受影响的状态位： 无

机器码：

0100	11da	ffff	ffff

说明： 将寄存器 f 的内容减 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果结果不为 0，则丢弃已取的指令而执行一条 NOP 指令，使该指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 DCFSNZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

```
HERE    DCFSNZ  TEMP, 1, 0
ZERO    :
NZERO   :
```

执行指令前
TEMP = ?
执行指令后
TEMP = TEMP – 1，
如果 TEMP = 0；
PC = 地址 （ZERO）
如果 TEMP ≠ 0；
PC = 地址 （NZERO）

GOTO **无条件跳转**

语法： GOTO k

操作数： 0 ≤ k ≤ 1048575

操作： k → PC<20:1>

受影响的状态位： 无

机器码：

第一个字 （k<7:0>）	1110	1111	$k_7$kkk	kkkk$_0$
第二个字 （k<19:8>）	1111	k_{19}kkk	kkkk	kkkk$_8$

说明： GOTO 指令允许无条件跳转到整个 2 MB 存储器范围中的任何位置。将 20 位值 k 装入 PC<20:1>。 GOTO 始终为双周期指令。

指令字数： 2

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	空操作	读立即数 k<19:8>，写入 PC
空操作	空操作	空操作	空操作

示例： GOTO THERE

执行指令后
PC = 地址 （THERE）

INCF **f 加 1**

语法： INCF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) + 1 → dest

受影响的状态位： C、 DC、 N、 OV 和 Z

机器码：

0010	10da	ffff	ffff

说明： 将寄存器 f 的内容加 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f （默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： INCF CNT, 1, 0

执行指令前
CNT = FFh
Z = 0
C = ?
DC = ?
执行指令后
CNT = 00h
Z = 1
C = 1
DC = 1

© 2007 Microchip Technology Inc.

INCFSZ **f 加 1，为 0 则跳过**

语法： INCFSZ f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) + 1 → dest，
如果结果 = 0 则跳过

受影响的状态位： 无

机器码：

0011	11da	ffff	ffff

说明： 将寄存器 f 的内容加 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果结果为 0，则丢弃已取的指令而执行一条 NOP 指令，使该指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 INCFSZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

```
HERE    INCFSZ   CNT, 1, 0
NZERO   :
ZERO    :
```

执行指令前
PC = 地址 （HERE）
执行指令后
CNT = CNT + 1
如果 CNT = 0；
PC = 地址 （ZERO）
如果 CNT ≠ 0；
PC = 地址 （NZERO）

INFSNZ **f 加 1，非 0 则跳过**

语法： INFSNZ f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) + 1 → dest，
如果结果 ≠ 0 则跳过

受影响的状态位： 无

机器码：

0100	10da	ffff	ffff

说明： 将寄存器 f 的内容加 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果结果不为 0，则丢弃已取的指令而执行一条 NOP 指令，使该指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 INFSNZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

```
HERE    INFSNZ  REG, 1, 0
ZERO
NZERO
```

执行指令前
PC = 地址 （HERE）
执行指令后
REG = REG + 1
如果 REG ≠ 0；
PC = 地址 （NZERO）
如果 REG = 0；
PC = 地址 （ZERO）

IORLW **将立即数与 W 作逻辑或运算**

语法： IORLW k

操作数： 0 ≤ k ≤ 255

操作： (W) .OR. k → W

受影响的状态位： N 和 Z

机器码：

0000	1001	kkkk	kkkk

说明： 将 W 的内容与 8 位立即数 k 进行逻辑或运算。结果存储在 W 寄存器中。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例： IORLW 35h

执行指令前
W = 9Ah
执行指令后
W = BFh

IORWF **将 W 与 f 作逻辑或运算**

语法： IORWF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (W) .OR.(f) → dest

受影响的状态位： N 和 Z

机器码：

0001	00da	ffff	ffff

说明： 将 W 的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： IORWF RESULT, 0, 1

执行指令前
RESULT = 13h
W = 91h
执行指令后
RESULT = 13h
W = 93h

© 2007 Microchip Technology Inc.

LFSR **载入 FSR**

语法： LFSR f, k

操作数： $0 \le f \le 2$
$0 \le k \le 4095$

操作： $k \rightarrow FSRf$

受影响的状态位： 无

机器码：

1110 1111	1110 0000	00ff $k_7$kkk	k_{11}kkk kkkk

说明： 将 12 位立即数 k 载入 f 所指向的文件选择寄存器。

指令字数： 2

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k 的 MSB	处理数据	将立即数 k 的 MSB 写入 FSRfH
译码	读立即数 k 的 LSB	处理数据	将立即数 k 的 LSB 写入 FSRfL

示例： LFSR 2, 3ABh

执行指令后
FSR2H = 03h
FSR2L = ABh

MOVF **移动 f**

语法： MOVF f {,d {,a}}

操作数： $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

操作： $f \rightarrow dest$

受影响的状态位： N 和 Z

机器码：

0101	00da	ffff	ffff

说明： 根据 d 的状态，将寄存器 f 的内容移入目标单元。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
f 可以为 256 字节存储区中的任何单元。如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 $f \le 95$（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写 W

示例： MOVF REG, 0, 0

执行指令前
REG = 22h
W = FFh
执行指令后
REG = 22h
W = 22h

MOVFF **将源寄存器的内容移到目标寄存器**

语法： MOVFF f_s,f_d

操作数： $0 \le f_s \le 4095$
$0 \le f_d \le 4095$

操作： $(f_s) \rightarrow f_d$

受影响的状态位： 无

机器码：

第一个字（源） 1100	ffff	ffff	$ffff_s$
第二个字（目标） 1111	ffff	ffff	$ffff_d$

说明： 将源寄存器 f_s 的内容移入目标寄存器 f_d。
源寄存器 f_s 可以是 4096 字节数据空间（000h 到 FFFh）中的任何单元，目标寄存器 f_d 也可以是 000h 到 FFFh 中的任何单元。
源或目标寄存器都可以是 W（这是个有用的特例）。
MOVFF 指令对于将数据存储单元中的内容移入外设寄存器（如发送缓冲器或 I/O 端口）的场合非常有用。
MOVFF 指令不能使用 PCL、TOSU、TOSH 或 TOSL 作为目标寄存器。

指令字数： 2

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f（源寄存器）	处理数据	空操作
译码	空操作 无效读取	空操作	写寄存器 f（目标寄存器）

示例： `MOVFF REG1, REG2`

执行指令前
REG1 = 33h
REG2 = 11h
执行指令后
REG1 = 33h
REG2 = 33h

MOVLB **将立即数移入 BSR 的低半字节**

语法： MOVLB k

操作数： $0 \le k \le 255$

操作： $k \rightarrow BSR$

受影响的状态位： 无

机器码：

0000	0001	kkkk	kkkk

说明： 将 8 位立即数 k 装入存储区选择寄存器（BSR）。不管 k_7:k_4 的值如何，BSR<7:4> 的值将始终保持为 0。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	将立即数 k 写入 BSR

示例： `MOVLB 5`

执行指令前
BSR 寄存器 = 02h
执行指令后
BSR 寄存器 = 05h

© 2007 Microchip Technology Inc.

MOVLW **将立即数移入 W**

语法： MOVLW k

操作数： 0 ≤ k ≤ 255

操作： k → W

受影响的状态位： 无

机器码：

0000	1110	kkkk	kkkk

说明： 将 8 位立即数 k 装入 W。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例： MOVLW 5Ah

执行指令后
W = 5Ah

MOVWF **将 W 的内容移入 f**

语法： MOVWF f {,a}

操作数： 0 ≤ f ≤ 255
a ∈ [0,1]

操作： (W) → f

受影响的状态位： 无

机器码：

0110	111a	ffff	ffff

说明： 将 W 寄存器中的数据移入寄存器 f。
f 可以是 256 字节存储区中的任何地址单元。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例： MOVWF REG, 0

执行指令前
W = 4Fh
REG = FFh
执行指令后
W = 4Fh
REG = 4Fh

MULLW **将立即数与 W 中的内容相乘**

语法： MULLW k

操作数： 0 ≤ k ≤ 255

操作： (W) x k → PRODH:PRODL

受影响的状态位： 无

机器码：

0000	1101	kkkk	kkkk

说明： 将 W 的内容与 8 位立即数 k 进行无符号的乘法运算。 16 位的结果存储在 PRODH:PRODL 寄存器对中，其中 PRODH 用于存储高字节。
W 的内容不改变。
所有状态标志位都不受影响。
请注意此操作不可能发生溢出或进位。结果有可能为零，但不会被检测到。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写寄存器 PRODH: PRODL

示例： MULLW 0C4h

执行指令前
W = E2h
PRODH = ?
PRODL = ?
执行指令后
W = E2h
PRODH = ADh
PRODL = 08h

MULWF **将 W 与 f 的内容相乘**

语法： MULWF f {,a}

操作数： 0 ≤ f ≤ 255
a ∈ [0,1]

操作： (W) x (f) → PRODH:PRODL

受影响的状态位： 无

机器码：

0000	001a	ffff	ffff

说明： 将 W 的内容与寄存器单元 f 的内容执行无符号的乘法运算。运算的 16 位结果保存在 PRODH:PRODL 寄存器对中，其中 PRODH 用于存储高字节。
W 和 f 的内容都不改变。
所有状态标志位都不受影响。
请注意此操作不可能发生溢出或进位。结果有可能为零，但不会被检测到。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区 （默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 PRODH: PRODL

示例： MULWF REG, 1

执行指令前
W = C4h
REG = B5h
PRODH = ?
PRODL = ?
执行指令后
W = C4h
REG = B5h
PRODH = 8Ah
PRODL = 94h

© 2007 Microchip Technology Inc.

NEGF **对 f 取补**

语法： NEGF f {,a}

操作数： $0 \le f \le 255$
$a \in [0,1]$

操作： $(\bar{f}) + 1 \rightarrow f$

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0110	110a	ffff	ffff

说明： 用二进制补码对单元 f 取补，结果存储在数据存储单元 f 中。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例：
```
NEGF    REG, 1
```
执行指令前
```
REG     =   0011 1010 [3Ah]
```
执行指令后
```
REG     =   1100 0110 [C6h]
```

NOP **空操作**

语法： NOP

操作数： 无

操作： 空操作

受影响的状态位： 无

机器码：

0000 1111	0000 xxxx	0000 xxxx	0000 xxxx

说明： 不执行任何操作。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作

示例：

无。

POP **弹出返回堆栈栈顶的内容**

语法： POP

操作数： 无

操作： (TOS) → 丢弃

受影响的状态位： 无

机器码：

0000	0000	0000	0110

说明： 从返回堆栈弹出 TOS 值并丢弃。然后，前一个压入返回堆栈的值成为 TOS 值。此指令可以让用户正确管理返回堆栈，从而实现软件堆栈。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	弹出 TOS 值	空操作

示例：

```
POP
GOTO    NEW
```

执行指令前
TOS = 0031A2h
堆栈 （下一级） = 014332h

执行指令后
TOS = 014332h
PC = NEW

PUSH **将数据压入返回堆栈栈顶**

语法： PUSH

操作数： 无

操作： (PC + 2) → TOS

受影响的状态位： 无

机器码：

0000	0000	0000	0101

说明： PC + 2 的值被压入返回堆栈的栈顶。原先的 TOS 值被压入堆栈的下一级。此指令允许通过修改 TOS 并将其压入返回堆栈来实现软件堆栈。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	将 PC + 2 压入返回堆栈	空操作	空操作

示例：

```
PUSH
```

执行指令前
TOS = 345Ah
PC = 0124h

执行指令后
PC = 0126h
TOS = 0126h
堆栈 （下一级） = 345Ah

© 2007 Microchip Technology Inc.

RCALL **相对调用**

语法： RCALL n

操作数： -1024 ≤ n ≤ 1023

操作： (PC) + 2 → TOS，
(PC) + 2 + 2n → PC

受影响的状态位： 无

机器码：

1101	1nnn	nnnn	nnnn

说明： 从当前地址跳转（最多 1K）来调用子程序。首先，将返回地址（PC + 2）压入返回堆栈。然后，将二进制补码“2n”与 PC 相加。因为 PC 要先递增才能取下一条指令，因此新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 n 将 PC 压入堆栈	处理数据	写入 PC
空操作	空操作	空操作	空操作

示例： HERE RCALL Jump

执行指令前
PC = 地址 （HERE）
执行指令后
PC = 地址 （Jump）
TOS = 地址 （HERE + 2）

RESET **复位**

语法： RESET

操作数： 无

操作： 将所有受 $\overline{\text{MCLR}}$ 复位影响的寄存器和标志位复位。

受影响的状态位： 全部

机器码：

0000	0000	1111	1111

说明： 此指令可实现用软件执行 $\overline{\text{MCLR}}$ 复位。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	开始复位	空操作	空操作

示例： RESET

执行指令后
寄存器 = 复位值
标志位 * = 复位值

RETFIE **从中断返回**

语法： RETFIE {s}

操作数： s ∈ [0,1]

操作： (TOS) → PC，
1 → GIE/GIEH 或 PEIE/GIEL，
如果 s = 1
(WS) → W，
(STATUSS) → STATUS，
(BSRS) → BSR，
PCLATU 和 PCLATH 保持不变。

受影响的状态位： GIE/GIEH 和 PEIE/GIEL。

机器码：

0000	0000	0001	000s

说明： 从中断返回。执行出栈操作，将栈顶（TOS）的内容装入 PC。通过将高或低优先级全局中断允许位置 1，来允许中断。如果 s = 1，则影子寄存器 WS、STATUSS 和 BSRS 的内容将被装入对应的寄存器 W、STATUS 和 BSR。如果 s = 0，则不更新这些寄存器 （默认）。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	空操作	从堆栈弹出 PC 值 将 GIEH 或 GIEL 置 1
空操作	空操作	空操作	空操作

示例： RETFIE 1

```
中断后
    PC                  =   TOS
    W                   =   WS
    BSR                 =   BSRS
    STATUS              =   STATUSS
    GIE/GIEH，PEIE/GIEL =   1
```

RETLW **将立即数返回给 W**

语法： RETLW k

操作数： 0 ≤ k ≤ 255

操作： k → W，
(TOS) → PC，
PCLATU 和 PCLATH 保持不变。

受影响的状态位： 无

机器码：

0000	1100	kkkk	kkkk

说明： 将 8 位立即数 k 装入 W。将栈顶内容（返回地址）装入程序计数器。高位地址锁存器 （PCLATH）内容保持不变。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	从堆栈弹出 PC 值，写入 W
空操作	空操作	空操作	空操作

示例：

```
    CALL TABLE ; W contains table
               ; offset value
               ; W now has
               ; table value
    :
TABLE
    ADDWF PCL  ; W = offset
    RETLW k0   ; Begin table
    RETLW k1   ;
    :
    :
    RETLW kn   ; End of table

    执行指令前
        W      =   07h
    执行指令后
        W      =   kn 的值
```


© 2007 Microchip Technology Inc.

RETURN **从子程序返回**

语法： RETURN {s}

操作数： s ∈ [0,1]

操作： (TOS) → PC，
如果 s = 1
(WS) → W，
(STATUSS) → STATUS，
(BSRS) → BSR，
PCLATU 和 PCLATH 保持不变。

受影响的状态位： 无

机器码：

0000	0000	0001	001s

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。如果 s = 1，将影子寄存器 WS、 STATUSS 和 BSRS 的内容装入相应的 W、 STATUS 和 BSR 寄存器。如果 s = 0，则不更新这些寄存器 （默认）。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	处理数据	从堆栈弹出 PC 值
空操作	空操作	空操作	空操作

示例： RETURN

执行指令后：
PC = TOS

RLCF **f 带进位循环左移**

语法： RLCF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

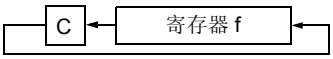
操作： (f<n>) → dest<n + 1>，
(f<7>) → C，
(C) → dest<0>

受影响的状态位： C、 N 和 Z

机器码：

0011	01da	ffff	ffff

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，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f （默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区 （默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。



指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例：
```
RLCF    REG, 0, 0
```
执行指令前
```
REG  =  1110 0110
C    =  0
```
执行指令后
```
REG  =  1110 0110
W    =  1100 1100
C    =  1
```

RLNCF **f 循环左移 （不带进位）**

语法： RLNCF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

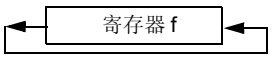
操作： (f<n>) → dest<n + 1>,
(f<7>) → dest<0>

受影响的状态位： N 和 Z

机器码：

0100	01da	ffff	ffff

说明： 将寄存器 f 的内容循环左移 1 位。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。



指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例：
```
RLNCF   REG, 1, 0
```
执行指令前
```
REG  =  1010 1011
```
执行指令后
```
REG  =  0101 0111
```

RRCF **f 带进位循环右移**

语法： RRCF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

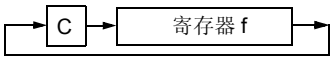
操作： (f<n>) → dest<n – 1>,
(f<0>) → C,
(C) → dest<7>

受影响的状态位： C、 N 和 Z

机器码：

0011	00da	ffff	ffff

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，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。



指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例：
```
RRCF    REG, 0, 0
```
执行指令前
```
REG  =  1110 0110
C    =  0
```
执行指令后
```
REG  =  1110 0110
W    =  0111 0011
C    =  0
```

 © 2007 Microchip Technology Inc.

RRNCF **f 循环右移 （不带进位）**

语法： RRNCF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

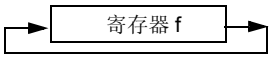
操作： (f<n>) → dest<n – 1>,
(f<0>) → dest<7>

受影响的状态位： N 和 Z

机器码：

0100	00da	ffff	ffff

说明： 将寄存器 f 的内容循环右移 1 位。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f （默认）。
如果 a 为 0，选择快速操作存储区，忽略 BSR 的值。如果 a 为 1，使用 BSR 选择 GPR 存储区 （默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。



指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

例 1：
```
RRNCF   REG, 1, 0
执行指令前
    REG    =   1101 0111
执行指令后
    REG    =   1110 1011
```

例 2：
```
RRNCF   REG, 0, 0
执行指令前
    W      =   ?
    REG    =   1101 0111
执行指令后
    W      =   1110 1011
    REG    =   1101 0111
```

SETF **将 f 的内容置为全 1**

语法： SETF f {,a}

操作数： 0 ≤ f ≤ 255
a ∈ [0,1]

操作： FFh → f

受影响的状态位： 无

机器码：

0110	100a	ffff	ffff

说明： 将指定寄存器的内容置为 FFh。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例：
```
SETF    REG,1
执行指令前
    REG    =   5Ah
执行指令后
    REG    =   FFh
```

SLEEP **进入休眠模式**

语法： SLEEP

操作数： 无

操作： 00h → WDT，
0 →WDT 后分频器，
1 → $\overline{TO}$，
0 → $\overline{PD}$

受影响的状态位： $\overline{TO}$ 和 $\overline{PD}$

机器码：

0000	0000	0000	0011

说明： 掉电状态位（$\overline{PD}$）清零。超时状态位（$\overline{TO}$）置 1。看门狗定时器及其后分频器清零。
振荡器停振，处理器进入休眠模式。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	处理数据	进入休眠模式

示例： SLEEP

执行指令前
$\overline{TO}$ = ?
$\overline{PD}$ = ?

执行指令后
$\overline{TO}$ = 1 †
$\overline{PD}$ = 0

† 如果由 WDT 引起唤醒，则此位将被清零。

SUBFWB **W 减去 f（带借位）**

语法： SUBFWB f {,d {,a}}

操作数： 0 £ f £ 255
d Œ [0,1]
a Œ [0,1]

操作： (W) – (f) – ($\overline{C}$) → dest

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0101	01da	ffff	ffff

说明： 将 W 的内容减去 f 寄存器的内容和进位（借位）（通过二进制补码方式进行运算）。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。
如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

例 1： SUBFWB REG, 1, 0

执行指令前
REG = 3
W = 2
C = 1

执行指令后
REG = FF
W = 2
C = 0
Z = 0
N = 1 ; 结果为负

例 2： SUBFWB REG, 0, 0

执行指令前
REG = 2
W = 5
C = 1

执行指令后
REG = 2
W = 3
C = 1
Z = 0
N = 0 ; 结果为正

例 3： SUBFWB REG, 1, 0

执行指令前
REG = 1
W = 2
C = 0

执行指令后
REG = 0
W = 2
C = 1
Z = 1 ; 结果为零
N = 0

© 2007 Microchip Technology Inc.

SUBLW **立即数减去 W 的内容**

语法： SUBLW k

操作数： 0 ≤ k ≤ 255

操作： k – (W) → W

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0000	1000	kkkk	kkkk

说明 用 8 位立即数 k 减去 W。结果存储在 W 寄存器中。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

例 1： SUBLW 02h

执行指令前
W = 01h
C = ?
执行指令后
W = 01h
C = 1 ；结果为正
Z = 0
N = 0

例 2： SUBLW 02h

执行指令前
W = 02h
C = ?
执行指令后
W = 00h
C = 1 ；结果为零
Z = 1
N = 0

例 3： SUBLW 02h

执行指令前
W = 03h
C = ?
执行指令后
W = FFh ；（2 进制补码）
C = 0 ；结果为负
Z = 0
N = 1

SUBWF **f 减去 W**

语法： SUBWF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) – (W) → dest

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0101	11da	ffff	ffff

说明： 用寄存器 f 中的内容减去 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，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f （默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区 （默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

例 1： SUBWF REG, 1, 0

执行指令前
REG = 3
W = 2
C = ?
执行指令后
REG = 1
W = 2
C = 1 ；结果为正
Z = 0
N = 0

例 2： SUBWF REG, 0, 0

执行指令前
REG = 2
W = 2
C = ?
执行指令后
REG = 2
W = 0
C = 1 ；结果为零
Z = 1
N = 0

例 3： SUBWF REG, 1, 0

执行指令前
REG = 1
W = 2
C = ?
执行指令后
REG = FFh ；（2 进制补码）
W = 2
C = 0 ；结果为负
Z = 0
N = 1

SUBWFB **f 减去 W （带借位）**

语法： SUBWFB f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f) – (W) – ($\overline{C}$) → dest

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0101	10da	ffff	ffff

说明： 用 f 寄存器的内容减去 W 的内容和进位（借位）（通过二进制补码方式进行运算）。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f （默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

例 1：
```
SUBWFB  REG, 1, 0
执行指令前
    REG  =  19h     (0001 1001)
    W    =  0Dh     (0000 1101)
    C    =  1
执行指令后
    REG  =  0Ch     (0000 1011)
    W    =  0Dh     (0000 1101)
    C    =  1
    Z    =  0
    N    =  0       ; 结果为正
```

例 2：
```
SUBWFB  REG, 0, 0
执行指令前
    REG  =  1Bh     (0001 1011)
    W    =  1Ah     (0001 1010)
    C    =  0
执行指令后
    REG  =  1Bh     (0001 1011)
    W    =  00h
    C    =  1
    Z    =  1       ; 结果为零
    N    =  0
```

例 3：
```
SUBWFB  REG, 1, 0
执行指令前
    REG  =  03h     (0000 0011)
    W    =  0Eh     (0000 1101)
    C    =  1
执行指令后
    REG  =  F5h     (1111 0100)
                    ; [2 进制补码]
    W    =  0Eh     (0000 1101)
    C    =  0
    Z    =  0
    N    =  1       ; 结果为负
```

SWAPF **将 f 的高半字节和低半字节交换**

语法： SWAPF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (f<3:0>) → dest<7:4>,
(f<7:4>) → dest<3:0>

受影响的状态位： 无

机器码：

0011	10da	ffff	ffff

说明： f 寄存器的高半字节和低半字节相互交换。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f （默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见 **第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例：
```
SWAPF   REG, 1, 0
执行指令前
    REG  =  53h
执行指令后
    REG  =  35h
```


© 2007 Microchip Technology Inc.

TBLRD **表读**

语法： TBLRD (*; *+; *-; +*)

操作数： 无

操作： 如果执行 TBLRD *
(程序存储器 (TBLPTR))→ TABLAT ;
TBLPTR 不改变;
如果执行 TBLRD *+
(程序存储器 (TBLPTR))→ TABLAT ;
(TBLPTR) + 1→ TBLPTR ;
如果执行 TBLRD *-
(程序存储器 (TBLPTR))→ TABLAT ;
(TBLPTR) – 1 → TBLPTR ;
如果执行 TBLRD +*
(TBLPTR) + 1 → TBLPTR ;
(程序存储器 (TBLPTR)) → TABLAT

受影响的状态位： 无

机器码：

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*

说明： 该指令用于读取程序存储器 （P.M.）的内容。使用表指针 （TBLPTR）对程序存储器进行寻址。
TBLPTR （一个 21 位指针）指向程序存储器的每个字节。 TBLPTR 的寻址范围为 2MB。

TBLPTR[0] = 0：程序存储器字的低有效字节

TBLPTR[0] = 1：程序存储器字的高有效字节

TBLRD 指令可用如下方法修改 TBLPTR 的值：

- 不变
- 后加
- 后减
- 预加

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作
空操作	空操作 （读程序存储器）	空操作	空操作 （写 TABLAT）

TBLRD **表读 （续）**

例 1： `TBLRD *+ ;`

执行指令前
TABLAT = 55h
TBLPTR = 00A356h
存储单元 （00A356h） = 34h
执行指令后
TABLAT = 34h
TBLPTR = 00A357h

例 2： `TBLRD +* ;`

执行指令前
TABLAT = AAh
TBLPTR = 01A357h
存储单元 （01A357h） = 12h
存储单元 （01A358h） = 34h
执行指令后
TABLAT = 34h
TBLPTR = 01A358h

TBLWT **表写**

语法： TBLWT (*; *+; *-; +*)

操作数： 无

操作： 如果执行 TBLWT*
(TABLAT) → 保持寄存器；
TBLPTR 不改变；
如果执行 TBLWT*+
(TABLAT) → 保持寄存器；
(TBLPTR) + 1 → TBLPTR ；
如果执行 TBLWT*-
(TABLAT) → 保持寄存器；
(TBLPTR) – 1 → TBLPTR ；
如果执行 TBLWT+*
(TBLPTR) + 1 → TBLPTR ；
(TABLAT) → 保持寄存器

受影响的状态位：无

机器码：

0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*

说明 此指令使用 TBLPTR 的低 3 位来确定要将 TABLAT 中的内容写入 8 个保持寄存器中的哪一个。该保持寄存器用于对程序存储器（P.M.）的内容编程。（关于对闪存程序存储器编程的更多详细信息，请参见**第 6.0 节“闪存程序存储器”**。）
TBLPTR（一个 21 位指针）指向程序存储器的每个字节。 TBLPTR 的寻址范围为 2MB。 TBLPTR 的 LSb 选择要访问的程序存储器单元。
TBLPTR[0] = 0：程序存储器字的低有效字节
TBLPTR[0] = 1：程序存储器字的高有效字节
TBLWT 指令可用如下方法修改 TBLPTR 的值：
- 不变
- 后加
- 后减
- 预加

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作
空操作	空操作（读 TABLAT）	空操作	空操作（写保持寄存器）

TBLWT **表写 （续）**

例 1： TBLWT *+;

执行指令前
TABLAT = 55h
TBLPTR = 00A356h
保持寄存器 (00A356h) = FFh
执行指令后 （表写操作完成）
TABLAT = 55h
TBLPTR = 00A357h
保持寄存器 (00A356h) = 55h

例 2： TBLWT +*;

执行指令前
TABLAT = 34h
TBLPTR = 01389Ah
保持寄存器 (01389Ah) = FFh
保持寄存器 (01389Bh) = FFh
执行指令后 （表写操作完成）
TABLAT = 34h
TBLPTR = 01389Bh
保持寄存器 (01389Ah) = FFh
保持寄存器 (01389Bh) = 34h

© 2007 Microchip Technology Inc.

TSTFSZ **测试 f，为 0 则跳过**

语法： TSTFSZ f {,a}

操作数： 0 ≤ f ≤ 255
a ∈ [0,1]

操作： f 为 0 则跳过

受影响的状态位： 无

机器码：

0110	011a	ffff	ffff

说明： 如果 f = 0，丢弃已取的指令并执行一条 NOP 指令，使这条指令成为双周期指令。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有 2 字指令，则执行 TSTFSZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

```
HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

执行指令前
PC = 地址 （HERE）
执行指令后
如果 CNT = 00h，
PC = 地址 （ZERO）
如果 CNT ≠ 00h，
PC = 地址 （NZERO）

XORLW **将立即数与 W 作逻辑异或运算**

语法： XORLW k

操作数： 0 ≤ k ≤ 255

操作： (W) .XOR. k → W

受影响的状态位： N 和 Z

机器码：

0000	1010	kkkk	kkkk

说明： 将 W 的内容与 8 位立即数 k 进行逻辑异或运算。结果存储在 W 寄存器中。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例：

```
XORLW   0AFh
```

执行指令前
W = B5h
执行指令后
W = 1Ah

XORWF **将 W 与 f 作逻辑异或运算**

语法： XORWF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (W) .XOR.(f) → dest

受影响的状态位： N 和 Z

机器码：

0001	10da	ffff	ffff

说明： 将 W 的内容与寄存器 f 的内容进行逻辑异或运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详细信息，请参见**第 19.2.3 节 “立即数变址寻址模式中面向字节和面向位的指令”**。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： XORWF REG, 1, 0

执行指令前
REG = AFh
W = B5h
执行指令后
REG = 1Ah
W = B5h

 © 2007 Microchip Technology Inc.

19.2 扩展指令集

除了PIC18指令集的75条标准指令之外，PIC18F2450/4450 器件还提供了针对内核 CPU 功能的可选扩展指令。这些新增的功能包括 8 条额外的指令，它们可以实现间接和变址寻址操作，并使得许多标准 PIC18 指令可以实现立即数变址寻址。

扩展指令集的额外功能在默认情况下是禁止的。用户必须通过将 XINST 配置位置 1，才能使能它们。

扩展指令集中的指令可以全部被归为立即数操作类指令，它们既可以控制文件选择寄存器，也可以使用这些寄存器进行变址寻址。其中的两个指令 ADDFSR 和SUBFSR，可以直接对 FSR2 进行操作。而 ADDULNK 和SUBULNK 指令允许在执行后自动返回。

这些扩展的指令专门用于优化用高级语言特别是C语言编写的重入程序代码（也就是递归调用或使用软件堆栈的代码）。此外，它们使用户能更有效地用高级语言对数据结构执行特定的操作。这些操作包括：

- 在进入和退出子程序时对软件堆栈空间进行动态分配和释放
- 功能指针调用
- 对软件堆栈指针进行控制
- 对软件堆栈中的变量进行控制

表 19-3 提供了扩展指令集中的指令汇总。**第 19.2.2 节“扩展指令集”**对这些指令进行了详细说明。第 212 页的表 19-1 提供了标准和扩展的 PIC18 指令集的操作码字段说明。

注：	扩展指令集和立即数变址寻址模式是专为优化用 C 语言编写的应用程序而设计的，用户可能不会在汇编器中直接使用这些指令。对于那些查看编译器生成代码的用户，这些命令的语法可作为参考。

19.2.1 扩展指令的语法

大部分扩展指令都使用变址参数，同时使用一个文件选择寄存器和某一偏移量来指定源寄存器或目标寄存器。当指令的参数作为变址寻址的一部分时，会用方括号（“[ ]”）把它括起来。这时表示此参数用作变址地址或偏移量。如果 MPASM™ 汇编器发现一个变址地址或偏移量没有被括起来，它就会给出出错信息。

当使能扩展指令集时，括号也用于表示面向字节和面向位的指令中的变址参数。这是对指令语法的额外更改。更多详细信息，请参见**第 19.2.3.1 节 “标准 PIC18 命令的扩展指令语法”**。

注：	以前，在 PIC18 和早期的指令集中使用方括号来表示可选参数。在此文本和以后的文本中，可选参数将用大扩号（“{}”）表示。

表 19-3： PIC18 指令集的扩展

助记符，操作数		说明	周期	16 位指令字				受影响的状态位
				MSb			LSb	
ADDFSR	f, k	将立即数与 FSR 相加	1	1110	1000	ffkk	kkkk	无
ADDULNK	k	将立即数与 FSR2 相加并返回	2	1110	1000	11kk	kkkk	无
CALLW		使用 WREG 调用子程序	2	0000	0000	0001	0100	无
MOVSF	z_s, f_d	将 z_s（源）移入（第一个字） f_d（目标）移入（第二个字）	2	1110 1111	1011 ffff	0zzz ffff	zzzz ffff	无
MOVSS	z_s, z_d	将 z_s（源）移入（第一个字） z_d（目标）移入（第二个字）	2	1110 1111	1011 xxxx	1zzz xzzz	zzzz zzzz	无
PUSHL	k	将立即数保存在 FSR2，FSR2 减 1	1	1110	1010	kkkk	kkkk	无
SUBFSR	f, k	FSR 减去立即数	1	1110	1001	ffkk	kkkk	无
SUBULNK	k	FSR2 减去立即数并返回	2	1110	1001	11kk	kkkk	无

19.2.2 扩展指令集

ADDFSR **FSR 的内容与立即数相加**

语法： ADDFSR f, k

操作数： 0 ≤ k ≤ 63
f ∈ [0, 1, 2]

操作： FSR(f) + k → FSR(f)

受影响的状态位： 无

机器码：

1110	1000	ffkk	kkkk

说明： 将由 f 指定的 FSR 的内容加上一个 6 位的立即数 k。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 FSR

示例： ADDFSR 2, 23h

执行指令前
FSR2 = 03FFh
执行指令后
FSR2 = 0422h

ADDULNK **FSR2 的内容与立即数相加并返回**

语法： ADDULNK k

操作数： 0 ≤ k ≤ 63

操作： FSR2 + k → FSR2，
(TOS) → PC

受影响的状态位： 无

机器码：

1110	1000	11kk	kkkk

说明： 将 FSR2 的内容加上一个 6 位的立即数 k。然后通过将 TOS 的值装入 PC，执行一条 RETURN 指令。
执行该指令需要两个周期；在第 2 个周期执行一条 NOP 指令。
该指令可以被认为是 ADDFSR 指令的特例，其中 f = 3 （二进制 “11”），它仅针对 FSR2 进行操作。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 FSR
空操作	空操作	空操作	空操作

示例： ADDULNK 23h

执行指令前
FSR2 = 03FFh
PC = 0100h
执行指令后
FSR2 = 0422h
PC = (TOS)

注： 所有的 PIC18 指令都可能在其指令助记符之前使用可选的标号参数，用于符号寻址。如果使用了标号，那么指令语法将变为：{label} 指令参数。

© 2007 Microchip Technology Inc.

CALLW **使用 WREG 调用子程序**

语法： CALLW

操作数： 无

操作： (PC + 2) → TOS，
(W) → PCL，
(PCLATH) → PCH，
(PCLATU) → PCU

受影响的状态位： 无

机器码：

0000	0000	0001	0100

说明： 首先，返回地址 （PC + 2）被压入返回堆栈。接下来，将 W 寄存器的内容写入 PCL， PCL 现有的值被丢弃。然后，PCLATH 和 PCLATU 的内容被分别锁存到 PCH 和 PCU。第二个周期执行一条 NOP 指令，并同时取下一条指令。
和 CALL 不一样，该指令没有更新 W、STATUS 或 BSR 寄存器的选项。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读 WREG	将 PC 压入堆栈	空操作
空操作	空操作	空操作	空操作

示例： HERE CALLW

执行指令前
PC = 地址 （HERE）
PCLATH = 10h
PCLATU = 00h
W = 06h

执行指令后
PC = 001006h
TOS = 地址 （HERE + 2）
PCLATH = 10h
PCLATU = 00h
W = 06h

MOVSF **将变址寻址单元内容移入 f**

语法： MOVSF [z_s], f_d

操作数： $0 \le z_s \le 127$
$0 \le f_d \le 4095$

操作： ((FSR2) + z_s) → f_d

受影响的状态位： 无

机器码：

第一个字 （源） 1110	1011	0zzz	$zzzz_s$
第二个字 （目标） 1111	ffff	ffff	$ffff_d$

说明： 将源寄存器的内容移入目标寄存器 f_d。通过将第一个字中的 7 位立即数偏移量 z_s 与 FSR2 的值相加来确定源寄存器的实际地址。第二个字中的 12 位立即数 f_d 指向目标寄存器的地址。两个地址均可以是 4096 字节的数据空间 （000h 到 FFFh）中的任何单元。
MOVSF 指令中的目标寄存器不能是 PCL、 TOSU、 TOSH 或 TOSL。
如果计算得到的源地址指向间接寻址寄存器，将返回 00h。

指令字数： 2

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	读源寄存器
译码	空操作 无效读取	空操作	写寄存器 f （目标寄存器）

示例： MOVSF [05h], REG2

执行指令前
FSR2 = 80h
85h 单元的内容= 33h
REG2 = 11h

执行指令后
FSR2 = 80h
85h 单元的内容= 33h
REG2 = 33h

MOVSS **变址寻址移动数据**

语法： MOVSS [z_s], [z_d]

操作数： $0 \le z_s \le 127$
$0 \le z_d \le 127$

操作： ((FSR2) + z_s) → ((FSR2) + z_d)

受影响的状态位： 无

机器码：

第一个字 （源） 1110	1011	1zzz	$zzzz_s$
第二个字 （目标） 1111	xxxx	xzzz	$zzzz_d$

说明： 将源寄存器的内容移到目标寄存器。通过将 FSR2 中的值分别加上 7 位立即数偏移量 z_s 和 z_d 来确定源寄存器和目标寄存器的地址。两个寄存器都可以是 4096 字节数据存储器空间 （000h 到 FFFh）中的任意单元。
MOVSS 指令不能使用 PCL、 TOSU、 TOSH 或 TOSL 作为目标寄存器。
如果计算得到的源地址指向间接寻址寄存器，将返回 00h。如果计算得到的目标地址指向间接寻址寄存器，将执行一条 NOP 指令。

指令字数： 2

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	读源寄存器
译码	确定目标地址	确定目标地址	写目标寄存器

示例： MOVSS [05h], [06h]

执行指令前
FSR2 = 80h
85h 单元的内容= 33h
86h 单元的内容= 11h
执行指令后
FSR2 = 80h
85h 单元的内容= 33h
86h 单元的内容= 33h

PUSHL **将立即数装入 FSR2 所指单元， FSR2 减 1**

语法： PUSHL k

操作数： $0 \le k \le 255$

操作： k → (FSR2),
FSR2 – 1 → FSR2

受影响的状态位： 无

机器码：

1111	1010	kkkk	kkkk

说明： 8 位立即数 k 被写入由 FSR2 指定的数据存储器单元。操作完后 FSR2 减 1。
此指令允许用户将值压入软件堆栈。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写入目标寄存器

示例： PUSHL 08h

执行指令前
FSR2H:FSR2L = 01ECh
存储单元 （01ECh） = 00h

执行指令后
FSR2H:FSR2L = 01EBh
存储单元 （01ECh） = 08h

 © 2007 Microchip Technology Inc.

SUBFSR **FSR 减去立即数**

语法： SUBFSR f, k

操作数： $0 \le k \le 63$

$f \in [0, 1, 2]$

操作： FSRf – k → FSRf

受影响的状态位： 无

机器码：

1110	1001	ffkk	kkkk

说明： 用 f 指定的 FSR 的内容减去 6 位立即数 k。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： SUBFSR 2, 23h

执行指令前
FSR2= 03FFh
执行指令后
FSR2= 03DCh

SUBULNK **FSR2 减去立即数并返回**

语法： SUBULNK k

操作数： $0 \le k \le 63$

操作： FSR2 – k → FSR2

(TOS) → PC

受影响的状态位： 无

机器码：

1110	1001	11kk	kkkk

说明： 用 FSR2 的内容减去 6 位立即数 k，然后通过将 TOS 的值装入 PC，执行一条 RETURN 指令。
执行该指令需要两个指令周期，第二个指令周期执行一条 NOP 指令。
该指令可以被看作是 SUBFSR 指令的特殊情况，其中 f = 3（二进制数 “11”）；它只针对 FSR2 进行操作。

指令字数： 1

指令周期数： 2

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器
空操作	空操作	空操作	空操作

示例： SUBULNK 23h

执行指令前
FSR2 = 03FFh
PC = 0100h
执行指令后
FSR2 = 03DCh
PC = (TOS)

19.2.3 立即数变址寻址模式中面向字节和面向位的指令

注：	使能 PIC18 扩展指令集可能导致常规应用程序运行不正常或完全失败。

一旦使能扩展指令集，除了可以使用 8 条新命令之外，还可以使用立即数变址寻址模式（**第 5.6.1 节“使用立即数偏移量进行变址寻址”**）。这将导致标准 PIC18 指令的地址解析方法有很大变化。

当禁止扩展指令集时，嵌入在操作码中的地址被视作立即数存储单元：可以是快速操作存储区中的单元（a = 0），或由 BSR 指定的 GPR 存储区中的单元（a = 1）。当使能扩展指令集且 a = 0 时，地址为 5Fh 或以下的文件寄存器参数被解析为 FSR2 中的指针值的偏移量，而不是一个立即数地址。对于实际应用来说，这意味着所有使用快速操作 RAM 位作为参数的指令，即所有面向字节或面向位的指令，或者几乎半数的 PIC18 内核指令，在使能了扩展指令集时操作都会有所不同。

当 FSR2 的内容为 00h 时，快速操作 RAM 的边界会被重新映射到它们的原始值。这对于编写向下兼容的代码很有用处。如果使用此技术，有必要在 “C” 程序调用汇编子程序时保存 FSR2 的值并在返回时将它恢复，这样做的目的是保护堆栈指针。用户还必须记住扩展指令集的语法要求（见**第 19.2.3.1 节 “标准 PIC18 命令的扩展指令语法”**）。

虽然立即数变址寻址模式对于动态堆栈和指针控制很有用处，但是如果不小心误用了寄存器也会非常麻烦。已经习惯使用 PIC18 编程的用户必须记住，在使能了扩展指令集后，地址小于或等于 5Fh 的寄存器用于立即数变址寻址。

下面是在立即数变址寻址模式中，一些面向字节和位的指令的示例，通过示例可以看出指令执行如何受到影响。示例中的操作数条件适用于所有这一类的指令。

19.2.3.1 标准 PIC18 命令的扩展指令语法

当使能了扩展指令集时，立即数偏移量 “k” 被用来替换标准的面向字节和位的命令中的文件寄存器参数 “f”。如前所述，只有在 “f” 小于或等于 5Fh 时才会发生这种情况。当使用偏移量时，该偏移量必须用方括号 “[]” 标出。因为在扩展指令集中，编译器将括号中的数值解析为变址地址或偏移量。省略括号，或在括号内使用大于 5Fh 的值会在 MPASM 汇编器中产生错误。

如果变址参数已被加上了括号，那么就不再需要指定快速操作 RAM 参数；此参数被假定为 0。这与标准操作（禁止扩展指令集时）刚好相反。在变址寻址模式中，定义快速操作 RAM 位也将在 MPASM 汇编器中产生错误。

目标参数 “d” 的操作和以前一样。

在 MPASM 汇编器的最新版本中，必须明确调用对扩展指令集的语言支持。可以通过命令行选项 /y 或在源代码中加入 PE 伪指令进行调用。

19.2.4 使能扩展指令集时的注意事项

需要注意的是并非所有用户都有必要使用扩展指令集，尤其是那些不使用软件堆栈的用户。

此外，立即数变址寻址模式可能会给写入 PIC18 汇编器的常规应用程序带来问题。这是因为常规的指令会尝试寻址快速操作存储区中地址低于 5Fh 的寄存器。当使能了扩展指令集时，这些地址被解析为相对于 FSR2 的立即数偏移量，所以应用程序会读或写错误的地址。

将应用程序移植到 PIC18F2450/4450 器件时，代码的类型是非常重要的。在使用扩展指令集时，用 C 语言编写的代码较长的重入应用程序会运行地很好，而大量使用快速操作存储区的常规应用程序不会获得任何益处。

© 2007 Microchip Technology Inc.

ADDWF **将 W 与变址寻址单元的内容相加（立即数变址寻址模式）**

语法： ADDWF [k] {,d}

操作数： $0 \le k \le 95$
$d \in [0,1]$

操作： (W) + ((FSR2) + k) → dest

受影响的状态位： N、 OV、 C、 DC 和 Z

机器码：

0010	01d0	kkkk	kkkk

说明： 将 W 的内容与由 FSR2 加上偏移量 k 指定的寄存器的内容相加。
如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写入目标寄存器

示例： ADDWF [OFST],0

执行指令前
W = 17h
OFST = 2Ch
FSR2 = 0A00h
0A2Ch 单元的内容 = 20h
执行指令后
W = 37h
0A2Ch 单元的内容 = 20h

BSF **将变址寻址单元相应位置 1（立即数变址寻址模式）**

语法： BSF [k], b

操作数： $0 \le f \le 95$
$0 \le b \le 7$

操作： 1 → ((FSR2) + k)<b>

受影响的状态位： 无

机器码：

1000	bbb0	kkkk	kkkk

说明： 将由 FSR2 加上偏移量 k 指定的寄存器中的位 b 置 1。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例： BSF [FLAG_OFST], 7

执行指令前
FLAG_OFST = 0Ah
FSR2 = 0A00h
0A0Ah 单元的内容 = 55h
执行指令后
0A0Ah 单元的内容 = D5h

SETF **将变址寻址单元置全 1（立即数变址寻址模式）**

语法： SETF [k]

操作数： $0 \le k \le 95$

操作： FFh → ((FSR2) + k)

受影响的状态位： 无

机器码：

0110	1000	kkkk	kkkk

说明： 将由 FSR2 加上偏移量 k 指定的寄存器的内容置为 FFh。

指令字数： 1

指令周期数： 1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写寄存器

示例： SETF [OFST]

执行指令前
OFST = 2Ch
FSR2 = 0A00h
0A2Ch 单元的内容= 00h
执行指令后
0A2Ch 单元的内容= FFh

19.2.5 使用 MICROCHIP MPLAB® IDE 工具的注意事项

最新版本的 Microchip 软件工具完全支持 PIC18F2450/4450 系列器件的扩展指令集。包括 MPLAB C18 C 语言编译器、 MPASM 汇编语言和 MPLAB 集成开发环境（IDE）。

在选择了使用软件开发的目标器件后， MPLAB IDE 将自动按默认模式设置该器件的配置位。XINST 配置位的默认设置是 0，禁用扩展指令集和立即数变址寻址模式。在编程过程中必须将 XINST 位置 1 才能确保使用扩展指令集开发的应用程序能够正确执行。

要使用扩展指令集开发软件，用户必须设置他们的语言工具以实现对扩展指令和变址寻址模式的支持。根据所使用的环境，可以通过以下几种方法：

- 开发环境中的菜单选项或对话框，允许用户配置项目的语言工具及其设置
- 命令行选项
- 源代码中的伪指令

这些选项在不同的编译器、汇编器和开发环境中将有所不同。建议用户在其开发系统所附带的文档中查询相应的信息。

© 2007 Microchip Technology Inc.

20.0 开发支持

一系列硬件及软件开发工具对 PIC® 单片机提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 汇编器 / 编译器 / 链接器
  - MPASM™ 汇编器
  - MPLAB C18 和 MPLAB C30 C 编译器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB ICE 2000 在线仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 2
- 器件编程器
  - PICSTART® Plus 开发编程器
  - MPLAB PM3 器件编程器
  - PICkit™ 2 开发编程器
- 低成本演示和开发板及评估工具包

20.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
  - 模拟器
  - 编程器 （单独销售）
  - 仿真器 （单独销售）
  - 在线调试器 （单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件 （汇编语言或 C 语言）
- 点击一次即可完成汇编 （或编译）并将代码下载到 PIC MCU 仿真器和模拟器工具中 （自动更新所有项目信息）
- 可使用如下各项进行调试：
  - 源文件 （汇编语言或 C 语言）
  - 混合汇编语言和 C 语言
  - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能更强大的工具时的学习时间。

20.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PIC MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特征：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

20.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完全的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 系列单片机及 dsPIC30F、dsPIC33 和 PIC24 系列数字信号控制器。这些编译器可提供其他编译器并不具备的强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

20.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。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中。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征：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起来增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

20.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。 MPLAB C30 C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特征：

- 支持整个 dsPIC30F 指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

20.6 MPLAB SIM 软件模拟器

MPLAB SIM软件模拟器在指令级对PIC MCU和dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器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的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及MPASM和MAPLAB ASM30 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。

 © 2007 Microchip Technology Inc.

20.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PIC 单片机的设计工具。 MPLAB ICE 2000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PIC 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

20.8 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 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境（IDE）所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC® 和 dsPIC® 闪存单片机进行调试和编程。IDE 是随每个工具包一起提供的。

MPLAB REAL ICE 探针通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与常用 MPLAB ICD 2 系统兼容的连接器 （RJ11）或新型抗噪声、高速低压差分信号（LVDS）互连电缆 （CAT5）与目标板相连。

可通过 MPLAB IDE 下载将来版本的固件，对 MPLAB REAL ICE 进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性，如软件断点和汇编代码跟踪等。在同类仿真器中， MPLAB REAL ICE 的优势十分明显：低成本、高速仿真、实时变量监视、跟踪分析、复杂断点、耐用的探针接口及较长（长达 3 米）的互连电缆。

20.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PIC MCU，可用于开发本系列及其他 PIC MCU 和 dsPIC DSC。MPLAB ICD 2 使用了闪存器件中内建的在线调试功能。该功能结合 Microchip 的在线串行编程（In-Circuit Serial Programming™， ICSP™）协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU 状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。 MPLAB ICD 2 还可用作某些 PIC 器件的开发编程器。

20.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 VDDMIN 和 VDDMAX 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 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、验证和编程。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，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

20.11 PICSTART Plus 开发编程器

PICSTART Plus开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM（RS-232）端口与 PC 相连。 MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PIC 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

20.12 PICkit 2 开发编程器

PICkit™ 2 开发编程器是一个低成本编程器；对于某些选定闪存器件，它也是一个调试器，通过其易于使用的接口可对众多 Microchip 的低档、中档和 PIC18F 系列闪存单片机进行编程。 PICkit 2 入门工具包中包含一个有实验布线区的开发板、十二堂系列课程、软件和 HI-TECH 的 PICC™ Lite C 编译器，有助于用户快速掌握 PIC® 单片机的使用。这一工具包为使用 Microchip 功能强大的中档闪存系列单片机进行编程、评估和应用开发，提供了所需的一切。

20.13 演示、开发和评估板

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，用于测试和修改。

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。

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）以及最新的 *“Product Selector Guide（产品选型指南）”*（DS00148）。

 © 2007 Microchip Technology Inc.

21.0 电气特性

绝对极限参数值 (†)

偏置电压下的环境温度 ..... -40°C 至 +85°C
储存温度 ..... -65°C 至 +150°C
任一引脚（除 VDD、 $\overline{\text{MCLR}}$ 和 RA4 外）相对于 VSS 的电压 ..... -0.3V 至（VDD + 0.3V）
VDD 引脚相对于 VSS 的电压 ..... -0.3V 至 +7.5V
$\overline{\text{MCLR}}$ 引脚相对于 VSS 的电压 **（注 2）** ..... 0V 至 +13.25V
总功耗 **（注 1）** 1.0W
流出 VSS 引脚的最大电流 ..... 300 mA
流入 VDD 引脚的最大电流 ..... 250 mA
输入箝位电流， IIK （VI < 0 或 VI > VDD） ..... ±20 mA
输出箝位电流， IOK （VO < 0 或 VO > VDD） ..... ±20 mA
任一 I/O 引脚的最大输出灌电流 ..... 25 mA
任一 I/O 引脚的最大输出拉电流 ..... 25 mA
所有端口的最大灌电流 ..... 200 mA
所有端口的最大拉电流 ..... 200 mA

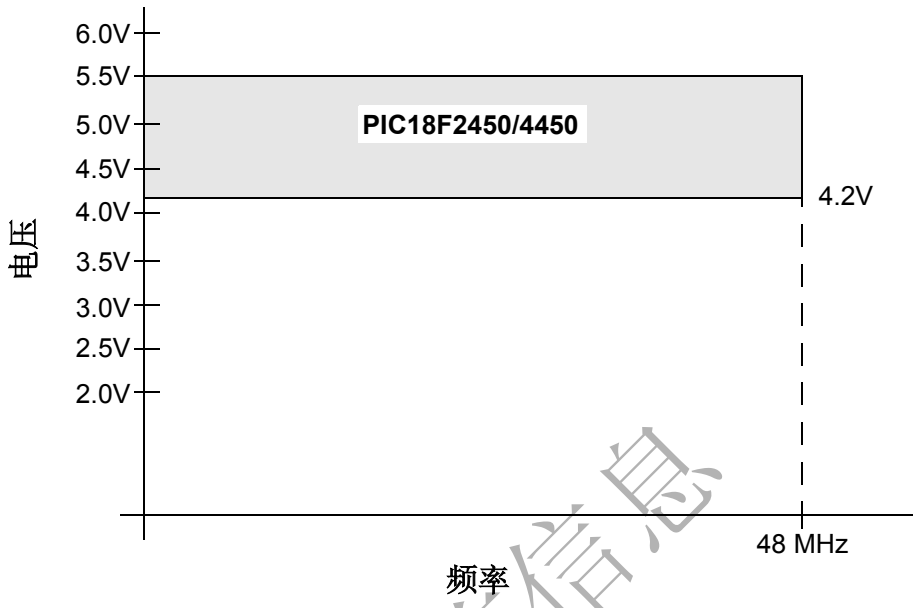
注 1： 功耗计算公式为：
$$Pdis = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

2： $\overline{\text{MCLR}}$/VPP/RE3 引脚上若出现低于 VSS 的尖峰电压，感应电流超过 80 mA，可能导致闭锁。因此，在 $\overline{\text{MCLR}}$/VPP/RE3 引脚上施加低电平时，应使用一个 50-100Ω 的串联电阻，而不是将该引脚直接与 VSS 连接。

† 注： 如果运行条件超过了上述“绝对极限参数值”，即可能对器件造成永久性损坏。上述值仅为运行条件的极大值，我们不建议器件运行在该规范范围以外。器件长时间工作在绝对极限参数条件下，其稳定性可能受到影响。

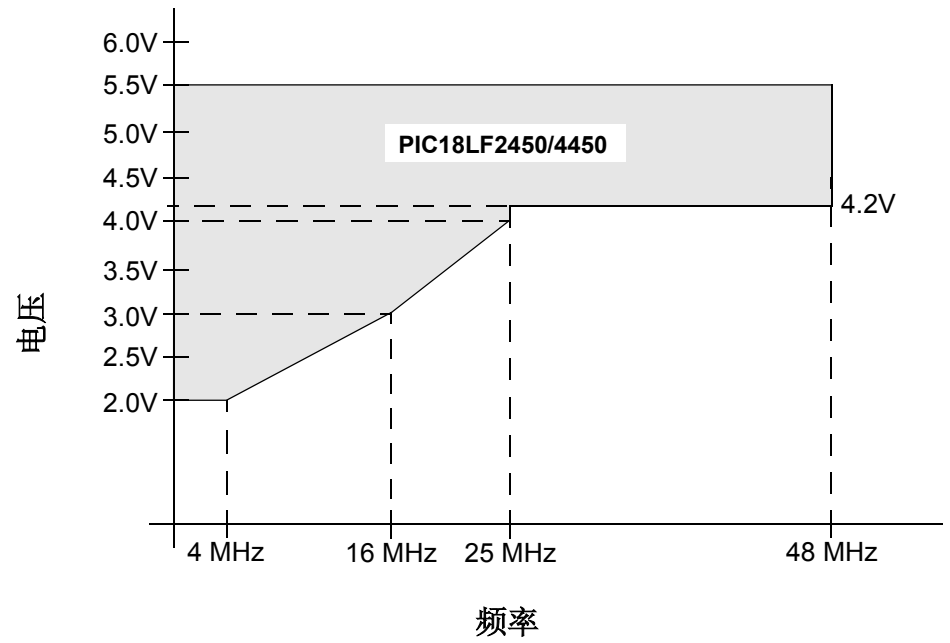
超前信息

图 21-1： PIC18F2450/4450 电压－频率关系图 （工业级）



超前信息

图 21-2： PIC18LF2450/4450 电压－频率关系图 （工业级）



注：VDDAPPMIN 是 PIC® 器件在应用中的最小电压。

 © 2007 Microchip Technology Inc.

21.1 直流特性： 供电电压
PIC18F2450/4450 （工业级）
PIC18LF2450/4450 （工业级）

PIC18LF2450/4450 （工业级）			**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C				
PIC18F2331/2431/4331/4431 （工业级）			**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C				
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**	2.0	—	5.5	V	EC、 HS、 XT 和内部振荡器模式
			3.0	—	5.5	V	HSPLL、 XTPLL、 ECPIO 和 ECPLL 振荡器模式
D002	VDR	**RAM 数据保持电压 (1)**	1.5	—	—	V	
D003	VPOR	**VDD 启动电压** （确保内部上电复位信号）	—	—	0.7	V	详见**第 4.3 节 “上电复位 （POR）”**。
D004	SVDD	**VDD 上升率** （确保内部上电复位信号）	0.05	—	—	V/ms	详见**第 4.3 节 “上电复位 （POR）”**。
D005	VBOR	**欠压复位电压**					
		BORV1:BORV0 = 11	2.00	2.05	2.16	V	
		BORV1:BORV0 = 10	2.65	2.79	2.93	V	
		BORV1:BORV0 = 01	4.11	4.33	4.55	V	
		BORV1:BORV0 = 00	4.36	4.59	4.82	V	

图注： 阴影行是为了增强表的可读性。

注 1： 这是在不丢失 RAM 数据的前提下，休眠模式或器件复位期间 VDD 所能降到的最小电压值。

超前信息

21.2 直流特性： 掉电和供电电流 PIC18F2450/4450 （工业级） PIC18LF2450/4450 （工业级）

PIC18LF2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C
PIC18F2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C

参数编号	器件	典型值	最大值	单位	条件	
	掉电电流 （IPD） (1)					
	PIC18F2450/4450	0.1	TBD	µA	-40°C	VDD = 2.0V （**休眠**模式）
		0.1	TBD	µA	+25°C	
		0.2	TBD	µA	+85°C	
	PIC18LF2450/4450	0.1	TBD	µA	-40°C	VDD = 3.0V （**休眠**模式）
		0.1	TBD	µA	+25°C	
		0.3	TBD	µA	+85°C	
	所有器件	0.1	TBD	µA	-40°C	VDD = 5.0V （**休眠**模式）
		0.1	TBD	µA	+25°C	
		0.4	TBD	µA	+85°C	

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。
上电工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS；
MCLR = VDD；根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

超前信息

 © 2007 Microchip Technology Inc.

21.2 直流特性： 掉电和供电电流 PIC18F2450/4450 （工业级） PIC18LF2450/4450 （工业级） （续）

PIC18LF2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C
PIC18F2450/4450 （工业级）	**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C

参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD）(2)						
	PIC18LF2450/4450	15	TBD	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz （**RC_RUN** 模式， INTRC 时钟源）
		15	TBD	μA	+25°C		
		15	TBD	μA	+85°C		
	PIC18LF2450/4450	40	TBD	μA	-40°C	VDD = 3.0V	
		35	TBD	μA	+25°C		
		30	TBD	μA	+85°C		
	所有器件	105	TBD	μA	-40°C	VDD = 5.0V	
		90	TBD	μA	+25°C		
		80	TBD	μA	+85°C		

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件 （如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。
上电工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS；
MCLR = VDD； 根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

超前信息

21.2 直流特性： 掉电和供电电流 PIC18F2450/4450 （工业级） PIC18LF2450/4450 （工业级） （续）

PIC18LF2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C
PIC18F2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C

参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD） (2)						
	PIC18LF2450/4450	2.9	TBD	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz （**RC_IDLE** 模式，INTRC 时钟源）
		3.1	TBD	μA	+25°C		
		3.6	TBD	μA	+85°C		
	PIC18LF2450/4450	4.5	TBD	μA	-40°C	VDD = 3.0V	
		4.8	TBD	μA	+25°C		
		5.8	TBD	μA	+85°C		
	所有器件	9.2	TBD	μA	-40°C	VDD = 5.0V	
		9.8	TBD	μA	+25°C		
		11.4	TBD	μA	+85°C		

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件 （如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。
上电工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS ；
MCLR = VDD ； 根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

超前信息

 © 2007 Microchip Technology Inc.

21.2 直流特性： 掉电和供电电流 PIC18F2450/4450 （工业级） PIC18LF2450/4450 （工业级） （续）

PIC18LF2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C
PIC18F2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C

参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD） (2)						
	PIC18LF2450/4450	250	TBD	μA	-40°C	VDD = 2.0V	FOSC = 1 MHz （PRI_RUN，EC 振荡器）
		250	TBD	μA	+25°C		
		250	TBD	μA	+85°C		
	PIC18LF2450/4450	550	TBD	μA	-40°C	VDD = 3.0V	
		480	TBD	μA	+25°C		
		460	TBD	μA	+85°C		
	所有器件	1.2	TBD	mA	-40°C	VDD = 5.0V	
		1.1	TBD	mA	+25°C		
		1.0	TBD	mA	+85°C		
	PIC18LF2450/4450	0.74	TBD	mA	-40°C	VDD = 2.0V	FOSC = 4 MHz （PRI_RUN，EC 振荡器）
		0.74	TBD	mA	+25°C		
		0.74	TBD	mA	+85°C		
	PIC18LF2450/4450	1.3	TBD	mA	-40°C	VDD = 3.0V	
		1.3	TBD	mA	+25°C		
		1.3	TBD	mA	+85°C		
	所有器件	2.7	TBD	mA	-40°C	VDD = 5.0V	
		2.6	TBD	mA	+25°C		
		2.5	TBD	mA	+85°C		
	所有器件	15	TBD	mA	-40°C	VDD = 4.2V	FOSC = 40 MHz （PRI_RUN，EC 振荡器）
		16	TBD	mA	+25°C		
		16	TBD	mA	+85°C		
	所有器件	21	TBD	mA	-40°C	VDD = 5.0V	
		21	TBD	mA	+25°C		
		21	TBD	mA	+85°C		
	所有器件	20	TBD	mA	-40°C	VDD = 4.2V	FOSC = 48 MHz （PRI_RUN，EC 振荡器）
		20	TBD	mA	+25°C		
		20	TBD	mA	+85°C		
	所有器件	25	TBD	mA	-40°C	VDD = 5.0V	
		25	TBD	mA	+25°C		
		25	TBD	mA	+85°C		

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件 （如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。

上电工作模式下，所有 IDD 测量值的测试条件为：

OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS；

MCLR = VDD；根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

超前信息

21.2 直流特性： 掉电和供电电流
PIC18F2450/4450 （工业级）
PIC18LF2450/4450 （工业级）（续）

PIC18LF2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C
PIC18F2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C

参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD） (2)						
	PIC18LF2450/4450	65	TBD	µA	-40°C	VDD = 2.0V	FOSC = 1 MHz （**PRI_IDLE** 模式，EC 振荡器）
		65	TBD	µA	+25°C		
		70	TBD	µA	+85°C		
	PIC18LF2450/4450	120	TBD	µA	-40°C	VDD = 3.0V	
		120	TBD	µA	+25°C		
		130	TBD	µA	+85°C		
	所有器件	230	TBD	µA	-40°C	VDD = 5.0V	
		240	TBD	µA	+25°C		
		250	TBD	µA	+85°C		
	PIC18LF2450/4450	255	TBD	µA	-40°C	VDD = 2.0V	FOSC = 4 MHz （**PRI_IDLE** 模式，EC 振荡器）
		260	TBD	µA	+25°C		
		270	TBD	µA	+85°C		
	PIC18LF2450/4450	420	TBD	µA	-40°C	VDD = 3.0V	
		430	TBD	µA	+25°C		
		450	TBD	µA	+85°C		
	所有器件	0.9	TBD	mA	-40°C	VDD = 5.0V	
		0.9	TBD	mA	+25°C		
		0.9	TBD	mA	+85°C		
	所有器件	6.0	TBD	mA	-40°C	VDD = 4.2V	FOSC = 40 MHz （**PRI_IDLE** 模式，EC 振荡器）
		6.2	TBD	mA	+25°C		
		6.6	TBD	mA	+85°C		
	所有器件	8.1	TBD	mA	-40°C	VDD = 5.0V	
		8.3	TBD	mA	+25°C		
		9.0	TBD	mA	+85°C		
	所有器件	8.0	TBD	mA	-40°C	VDD = 4.2V	FOSC = 48 MHz （**PRI_IDLE** 模式，EC 振荡器）
		8.1	TBD	mA	+25°C		
		8.2	TBD	mA	+85°C		
	所有器件	9.8	TBD	mA	-40°C	VDD = 5.0V	
		10.0	TBD	mA	+25°C		
		10.5	TBD	mA	+85°C		

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件 （如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。
上电工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS；
MCLR = VDD； 根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

超前信息

 © 2007 Microchip Technology Inc.

21.2 直流特性： 掉电和供电电流
PIC18F2450/4450 （工业级）
PIC18LF2450/4450 （工业级） （续）

PIC18LF2450/4450 （工业级）	**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C
PIC18F2450/4450 （工业级）	**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C

参数编号	**器件**	**典型值**	**最大值**	**单位**	**条件**		
	供电电流 （IDD）(2)						
	PIC18LF2450/4450	14	TBD	µA	-40°C	VDD = 2.0V	FOSC = 32 kHz(3)（**SEC_RUN** 模式，Timer1 作为时钟源）
		15	TBD	µA	+25°C		
		16	TBD	µA	+85°C		
	PIC18LF2450/4450	40	TBD	µA	-40°C	VDD = 3.0V	
		35	TBD	µA	+25°C		
		31	TBD	µA	+85°C		
	所有器件	99	TBD	µA	-40°C	VDD = 5.0V	
		81	TBD	µA	+25°C		
		75	TBD	µA	+85°C		
	PIC18LF2450/4450	2.5	TBD	µA	-40°C	VDD = 2.0V	FOSC = 32 kHz(3)（**SEC_IDLE** 模式，Timer1 作为时钟源）
		3.7	TBD	µA	+25°C		
		4.5	TBD	µA	+85°C		
	PIC18LF2450/4450	5.0	TBD	µA	-40°C	VDD = 3.0V	
		5.4	TBD	µA	+25°C		
		6.3	TBD	µA	+85°C		
	所有器件	8.5	TBD	µA	-40°C	VDD = 5.0V	
		9.0	TBD	µA	+25°C		
		10.5	TBD	µA	+85°C		

超前信息

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件 （如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。
上电工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS ；
MCLR = VDD ； 根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

21.2 直流特性： 掉电和供电电流 PIC18F2450/4450 （工业级） PIC18LF2450/4450 （工业级）（续）

PIC18LF2450/4450 （工业级）	标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C
PIC18F2450/4450 （工业级）	**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C

参数编号	器件	典型值	最大值	单位	条件		
D022 （ΔIWDT）	**模块差分电流 （ΔIWDT、 ΔIBOR、 ΔILVD、 ΔIOSCB 和 ΔIAD）**						
	看门狗定时器	1.3	TBD	µA	-40°C	VDD = 2.0V	
		1.4	TBD	µA	+25°C		
		2.0	TBD	µA	+85°C		
		1.9	TBD	µA	-40°C	VDD = 3.0V	
		2.0	TBD	µA	+25°C		
		2.8	TBD	µA	+85°C		
		4.0	TBD	µA	-40°C	VDD = 5.0V	
		5.5	TBD	µA	+25°C		
		5.6	TBD	µA	+85°C		
D022A （ΔIBOR）	**欠压复位 (4)**	35	TBD	µA	-40°C至+85°C	VDD = 3.0V	
		40	TBD	µA	-40°C至+85°C	VDD = 5.0V	
		0	TBD	µA	-40°C至+85°C		休眠模式，BOREN1:BOREN0 = 10
D022B （ΔILVD）	**高 / 低压检测 (4)**	22	TBD	µA	-40°C至+85°C	VDD = 2.0V	
		25	TBD	µA	-40°C至+85°C	VDD = 3.0V	
		29	TBD	µA	-40°C至+85°C	VDD = 5.0V	
D025 （ΔIOSCB）	**Timer1 振荡器**	2.1	TBD	µA	-40°C	VDD = 2.0V	Timer1(3) 为 32 kHz
		1.8	TBD	µA	+25°C		
		2.1	TBD	µA	+85°C		
		2.2	TBD	µA	-40°C	VDD = 3.0V	Timer1(3) 为 32 kHz
		2.6	TBD	µA	+25°C		
		2.9	TBD	µA	+85°C		
		3.0	TBD	µA	-40°C	VDD = 5.0V	Timer1(3) 为 32 kHz
		3.2	TBD	µA	+25°C		
		3.4	TBD	µA	+85°C		
D026 （ΔIAD）	**A/D 转换器**	1.0	TBD	µA	-40°C至+85°C	VDD = 2.0V	A/D 启动，但不进行转换
		1.0	TBD	µA	-40°C至+85°C	VDD = 3.0V	
		1.0	TBD	µA	-40°C至+85°C	VDD = 5.0V	

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件 （如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。
上电工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS；
MCLR = VDD；根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

超前信息

 © 2007 Microchip Technology Inc.

21.2 直流特性： 掉电和供电电流
PIC18F2450/4450 （工业级）
PIC18LF2450/4450 （工业级） （续）

PIC18LF2450/4450 （工业级）		**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C					
PIC18F2450/4450 （工业级）		**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C					
参数编号	**器件**	**典型值**	**最大值**	**单位**	**条件**		
	USB 和相关模块差分电流 （ΔIUSBx、 ΔIPLL 和 ΔIUREG）						
ΔIUSBx	**带有片上收发器的 USB 模块**	TBD	TBD	mA	+25°C	VDD = 3.3V	
		TBD	TBD	mA	+25°C	VDD = 5.0V	
ΔIPLL	**96 MHz PLL** **（振荡器模块）**	TBD	TBD	TBD	+25°C	VDD = 3.3V	
		TBD	TBD	TBD	+25°C	VDD = 5.0V	
ΔIUREG	**USB 内部稳压器**	TBD	TBD	TBD	+25°C	VDD = 5.0V	

图注： TBD = 待定。阴影行是为了增强表的可读性。

注 1： 在休眠模式下，掉电电流并不取决于振荡器的类型。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻状态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、 Timer1 振荡器、 BOR 等）时测得的。

2： 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。
上电工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨摆幅；所有 I/O 引脚均为三态，上拉至 VDD 或 VSS；
MCLR = VDD；根据具体应用使能 / 禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围为 -10°C 至 +70°C。扩展温度晶振成本则高很多。

4： BOR 和 HLVD 使能内部带隙参考源。当同时使能两个模块时，电流消耗将少于两个规范值的和。

超前信息

21.3 直流特性： PIC18F2450/4450 （工业级）
PIC18LF2450/4450 （工业级）

直流特性			标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C			
参数编号	符号	特性	最小值	最大值	单位	条件
	VIL	**输入低电压**				
		I/O 端口 （USB 模式下的 RC4/RC5 除外）：				
D030		带 TTL 缓冲器	VSS	0.15 VDD	V	VDD < 4.5V
D030A			—	0.8	V	4.5V ≤ Vdd ≤ 5.5V
D031		带施密特触发缓冲器	VSS	0.2 VDD	V	
		RC3 和 RC4	VSS	0.3 VDD	V	
D032		$\overline{MCLR}$	VSS	0.2 VDD	V	
D032A		OSC1 和 T1OSI	VSS	0.3 VDD	V	XT、HS 和 HSPLL 模式 (1)
D033		OSC1	VSS	0.2 VDD	V	EC 模式 (1)
	VILU	D+/D- 输入	—	0.8	V	VDD = 4.35V， USB 挂起 (5)
	VIH	**输入高电压**				
		I/O 端口 （USB 模式下的 RC4/RC5 除外）：				
D040		带 TTL 缓冲器	0.25 VDD + 0.8V	VDD	V	VDD < 4.5V
D040A			2.0	VDD	V	4.5V ≤ Vdd ≤ 5.5V
D041		带施密特触发缓冲器	0.8 VDD	VDD	V	
		RC3 和 RC4	0.7 VDD	VDD	V	
D042		$\overline{MCLR}$	0.8 VDD	VDD	V	
D042A		OSC1 和 T1OSI	0.7 VDD	VDD	V	XT、HS 和 HSPLL 模式 (1)
D043		OSC1	0.8 VDD	VDD	V	EC 模式 (1)
	VIHU	D+/D- 输入	2.4	—	V	VDD = 4.35V， USB 挂起 (5)
	IIL	**输入泄漏电流 (2,3)**				
D060		I/O 端口	—	±1	µA	VSS ≤ VPIN ≤ VDD， 引脚处于高阻状态
D061		$\overline{MCLR}$	—	±5	µA	Vss ≤ VPIN ≤ VDD
D063		OSC1	—	±5	µA	Vss ≤ VPIN ≤ VDD
	IPU	**弱上拉电流**				
D070	IPURB	PORTB 弱上拉电流	50	400	µA	VDD = 5V， VPIN = VSS

注 1： 在 RC 振荡器配置中， OSC1/CLKI 引脚被配置为施密特触发器输入。在 RC 模式下，建议不要使用外部时钟驱动 PIC® 器件。

2： $\overline{MCLR}$ 引脚上的泄漏电流主要取决于所施加电平。规定电平为正常工作条件下的电平。在不同的输入电压下可测得更高的泄漏电流。

3： 负电流定义引脚的拉电流。

4： 参数为特性值，但未经测试。

5： USB 规范 2.0 的 D+ 参数。

超前信息

 © 2007 Microchip Technology Inc.

21.3 直流特性： PIC18F2450/4450 （工业级） PIC18LF2450/4450 （工业级） （续）

直流特性			标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C			
参数编号	符号	特性	最小值	最大值	单位	条件
	VOL	**输出低电压**				
D080		I/O 端口 （USB 模式下的 RC4/RC5 除外）	—	0.6	V	IOL = 8.5 mA， VDD = 4.5V，-40°C 至 +85°C
D083		OSC2/CLKO （EC 和 ECIO 模式）	—	0.6	V	IOL = 1.6 mA， VDD = 4.5V，-40°C 至 +85°C
	VOLU	D+/D- 输出	—	0.3		VDD = 4.35V，USB 挂起 (5)
	VOH	**输出高电压 (3)**				
D090		I/O 端口 （USB 模式下的 RC4/RC5 除外）	VDD – 0.7	—	V	IOH = -3.0 mA，VDD = 4.5V，-40°C 至 +85°C
D092		OSC2/CLKO （EC、 ECIO 和 ECPIO 模式）	VDD – 0.7	—	V	IOH = -1.3 mA，VDD = 4.5V，-40°C 至 +85°C
	VOHU	D+/D- 输出	2.8	3.6	V	VDD = 4.35V，USB 挂起 (5)
		输出引脚上的容性负载规范				
D100(4)	COSC2	OSC2 引脚	—	15	pF	当外部时钟用于驱动 OSC1 时处于 XT 和 HS 模式下
D101	CIO	所有 I/O 引脚和 OSC2 （在 RC 模式下）	—	50	pF	满足交流时序规范

超前信息

注 1： 在 RC 振荡器配置中， OSC1/CLKI 引脚被配置为施密特触发器输入。在 RC 模式下，建议不要使用外部时钟驱动 PIC® 器件。

2： $\overline{\text{MCLR}}$ 引脚上的泄漏电流主要取决于所施加电平。规定电平为正常工作条件下的电平。在不同的输入电压下可测得更高的泄漏电流。

3： 负电流定义引脚的拉电流。

4： 参数为特性值，但未经测试。

5： USB 规范 2.0 的 D+ 参数。

表 21-1： 存储器编程要求

直流特性			标准运行条件 （除非另外说明） 运行温度 工业级为 -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
		内部程序存储器编程规范 (1)					
D110	VPP	$\overline{MCLR}$/VPP/RE3 引脚上的电压	9.00	—	13.25	V	（注 **2**）
D113	IDDP	编程期间的供电电流	—	—	10	mA	
		程序闪存					
D130	EP	单元擦写次数	10K	100K	—	E/W	-40°C 至 +85°C
D131	VPR	用于读取的 VDD	VMIN	—	5.5	V	VMIN = 最小工作电压
D132	VIE	用于块擦除的 VDD	4.5	—	5.5	V	使用 ICSP™ 端口
D132A	VIW	用于外部定时擦除或写的 VDD	3.0	—	5.5	V	使用 ICSP 端口
D132B	VPEW	自定时写的 VDD	VMIN	—	5.5	V	VMIN = 最小工作电压
D133	TIE	ICSP 块擦除周期时间	—	4	—	ms	VDD > 4.5V
D133A	TIW	ICSP 擦除或写周期时间 （外部定时）	1	—	—	ms	VDD > 4.5V
D133A	TIW	自定时写周期时间	—	2	—	ms	
D134	TRETD	特性保存期	40	100	—	年	假设没有违反其他规范

† 除非另外说明，否则“典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 这些规范用于通过使用表写指令对片上程序存储器进行编程。

2： 仅在禁止单电源编程时才需要。

超前信息

 © 2007 Microchip Technology Inc.

表 21-2： USB 模块规范

运行条件：-40°C < TA < +85°C （除非另外说明）。							
参数编号	**符号**	**特性**	**最小值**	**典型值**	**最大值**	**单位**	**注释**
D313	VUSB	USB 电压	3.0	—	3.6	V	总线上的电压必须在此范围内以确保 USB 正确运行
D314	IIL	引脚上的输入泄漏电流	—	—	±1	µA	VSS ≤ VPAD ≤ VDD；引脚处于高阻状态
D315	VILUSB	USB 缓冲器的输入低电压	—	—	0.8	V	用于 VUSB 范围
D316	VIHUSB	USB 缓冲器的输入高电压	2.0	—	—	V	用于 VUSB 范围
D317	VCRS	交叉电压	1.3		2.0	V	pad_dp 和 pad_dm 交叉电压发生的电压范围
D318	VDIFS	差分输入灵敏度	—	—	0.2	V	当达到 VCM 时， D+ 和 D- 之间的差值必须超出此值。
D319	VCM	差分共模范围	0.8	—	2.5	V	
D320	ZOUT	驱动器输出阻抗	28	—	44	Ω	
D321	VOL	输出低电压	0.0	—	0.3	V	1.5 kΩ 负载连接到 3.6V
D322	VOH	输出高电压	2.8	—	3.6	V	15 kΩ 负载接地

表 21-3： USB 内部稳压器规范

运行条件：-40°C < TA < +85°C （除非另外说明）。							
参数编号	**符号**	**特性**	**最小值**	**典型值**	**最大值**	**单位**	**注释**
D323	VUSBANA	稳压器输出电压 *	3.0	—	3.6	V	
D324	CUSB	外部滤波器电容值 *	220	—	—	nF	必须为最高负载 （最小压降）保持足够的电量

* 这些参数为特性值，但未经测试。尚未为这些规范分配参数编号。

超前信息

图 21-3：　　高 / 低压检测特性

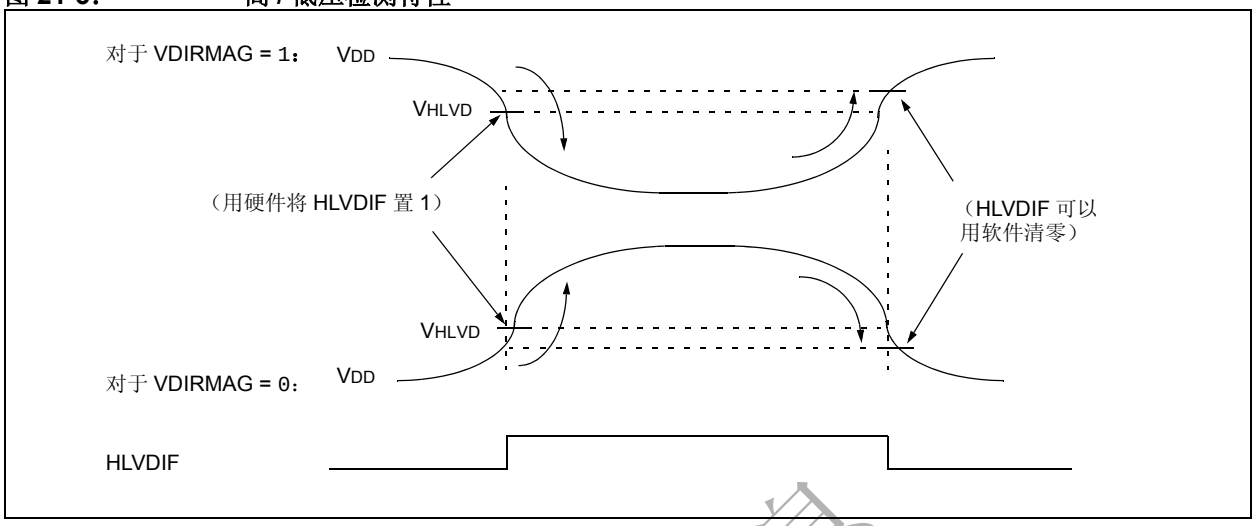


表 21-4：　　高 / 低压检测特性

标准运行条件（除非另外说明） 运行温度　　工业级为 -40°C ≤ TA ≤ +85°C								
参数编号	**符号**	**特性**		**最小值**	**典型值**	**最大值**	**单位**	**条件**
D420		VDD 由高转变为低时的 HLVD 电压	HLVDL<3:0> = 0000	2.06	2.17	2.28	V	
			HLVDL<3:0> = 0001	2.12	2.23	2.34	V	
			HLVDL<3:0> = 0010	2.24	2.36	2.48	V	
			HLVDL<3:0> = 0011	2.32	2.44	2.56	V	
			HLVDL<3:0> = 0100	2.47	2.60	2.73	V	
			HLVDL<3:0> = 0101	2.65	2.79	2.93	V	
			HLVDL<3:0> = 0110	2.74	2.89	3.04	V	
			HLVDL<3:0> = 0111	2.96	3.12	3.28	V	
			HLVDL<3:0> = 1000	3.22	3.39	3.56	V	
			HLVDL<3:0> = 1001	3.37	3.55	3.73	V	
			HLVDL<3:0> = 1010	3.52	3.71	3.90	V	
			HLVDL<3:0> = 1011	3.70	3.90	4.10	V	
			HLVDL<3:0> = 1100	3.90	4.11	4.32	V	
			HLVDL<3:0> = 1101	4.11	4.33	4.55	V	
			HLVDL<3:0> = 1110	4.36	4.59	4.82	V	

超前信息

© 2007 Microchip Technology Inc.

21.4 交流 （时序）特性

21.4.1 时序参数符号体系

时序参数符号采用以下格式之一进行创建：

1. TppS2ppS
2. TppS

T			
	F	频率	T	时间

小写字母 （pp）及其含义：

pp			
		mc	$\overline{\text{MCLR}}$
cc	CCP1	osc	OSC1
ck	CLKO	wr	$\overline{\text{WR}}$
dt	数据输入	t0	T0CKI
io	I/O 端口	t1	T1CKI

大写字母及其含义：

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

超前信息

21.4.2 时序条件

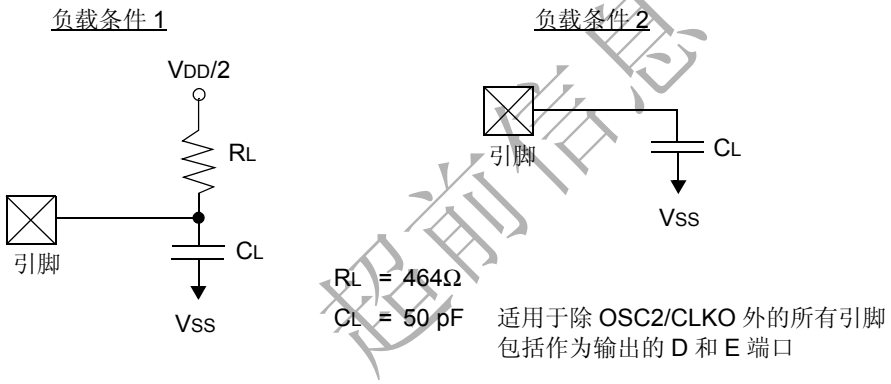
表21-5中指定的温度和电压适用于所有的时序规范（除非另外说明）。图 21-4 规定了时序规范的负载条件。

注： 由于篇幅所限，本节中通称的“PIC18FXXXX”和“PIC18LFXXXX”特指（而且仅指代）PIC18F2450/4450 和 PIC18LF2450/4450 系列器件。

表 21-5： 温度和电压规范——交流

交流特性	**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C 直流规范**第 21.1 节**和**第 21.3 节**描述了工作电压 VDD 的范围。 LF 器件仅在工业级温度下工作。

图 21-4： 器件时序规范的负载条件



超前信息

 © 2007 Microchip Technology Inc.

21.4.3 时序图和规范

图 21-5： 外部时钟时序 （除 PLL 外的所有模式）

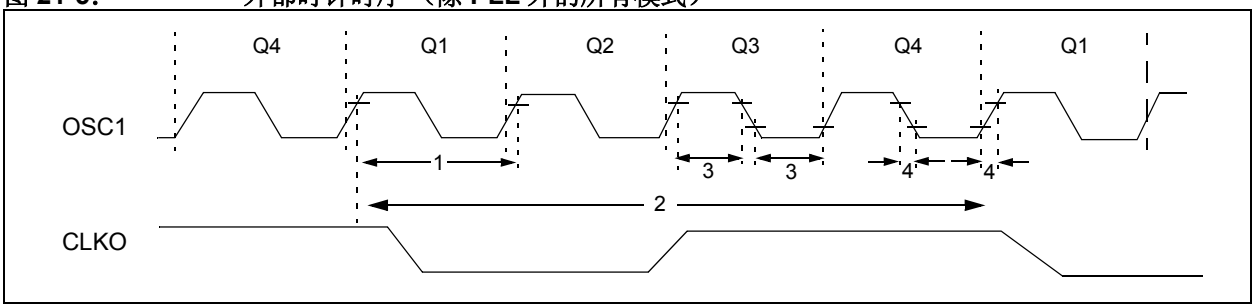


表 21-6： 外部时钟时序要求

参数编号	符号	特性	最小值	最大值	单位	条件
1A	FOSC	外部 CLKI 频率 (1) 振荡器频率 (1)	DC 0.1 4 4	48 4 25 48	MHz MHz MHz MHz	EC 和 ECIO 振荡器模式 XT 和 XTPLL 振荡器模式 HS 振荡器模式 HSPLL 振荡器模式
1	TOSC	外部 CLKI 周期 (1) 振荡器周期 (1)	20.8 250 25 20.8	— 10,000 250 250	ns ns ns ns	EC 和 ECIO 振荡器模式 XT 振荡器模式 HS 振荡器模式 HSPLL 振荡器模式
2	TCY	指令周期时间 (1)	83.3	—	ns	TCY = 4/FOSC
3	TosL， TosH	外部时钟输入 （OSC1）的高电平或低电平时间	30 10	— —	ns ns	XT 振荡器模式 HS 振荡器模式
4	TosR， TosF	外部时钟输入 （OSC1）的上升或下降时间	— —	20 7.5	ns ns	XT 振荡器模式 HS 振荡器模式

注 1： 对于除 PLL 外的所有配置，指令周期时间 （TCY）等于输入振荡器时基周期的四倍。所有规定值均为基于器件在标准运行条件下执行代码所对应的特定振荡器类型的特性数据。超出规定值可能导致振荡器运行不稳定和 / 或电流消耗超出预期值。所有器件在测试 “最小值” 时，都在 OSC1/CLKI 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）。

超前信息

表 21-7： PLL 时钟时序规范 （VDD = 4.2V 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	—	48	MHz	
F11	FSYS	片上 VCO 系统频率	—	96	—	MHz	
F12	t_{rc}	PLL 起振时间 （锁定时间）	—	—	2	ms	
F13	ΔCLK	CLKO 稳定性 （抗抖动）	-0.25	—	+0.25	%	

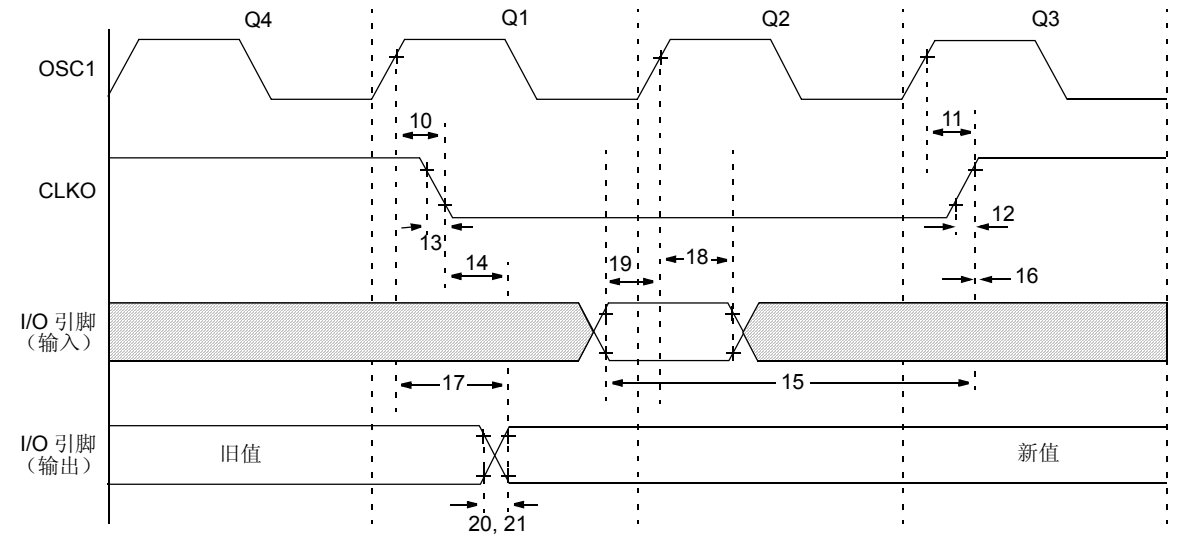
† 除非另外说明，否则 “典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

表 21-8： 交流特性： 内部 RC 精度
PIC18F2450/4450 （工业级）
PIC18LF2450/4450 （工业级）

PIC18LF2450/4450（工业级）	**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C					
PIC18F2450/4450（工业级）	**标准运行条件 （除非另外说明）** 运行温度 工业级为 -40°C ≤ TA ≤ +85°C					
参数编号	**器件**	**最小值**	**典型值**	**最大值**	**单位**	**条件**
	在频率为 31 kHz(1) 时的 INTRC 精度					
	PIC18LF2450/4450	26.562	—	35.938	kHz	-40°C 至 +85°C, VDD = 2.7-3.3V
	PIC18F2450/4450	26.562	—	35.938	kHz	-40°C 至 +85°C, VDD = 4.5-5.5V

图注： 阴影行是为了增强表的可读性。
注 1： 校准后的 INTRC 频率。
2： VDD 改变时 INTRC 频率的变化。

图 21-6： CLKO 和 I/O 时序



注： 负载条件请参见图 21-4。

© 2007 Microchip Technology Inc.

表 21-9： CLKO 和 I/O 时序要求

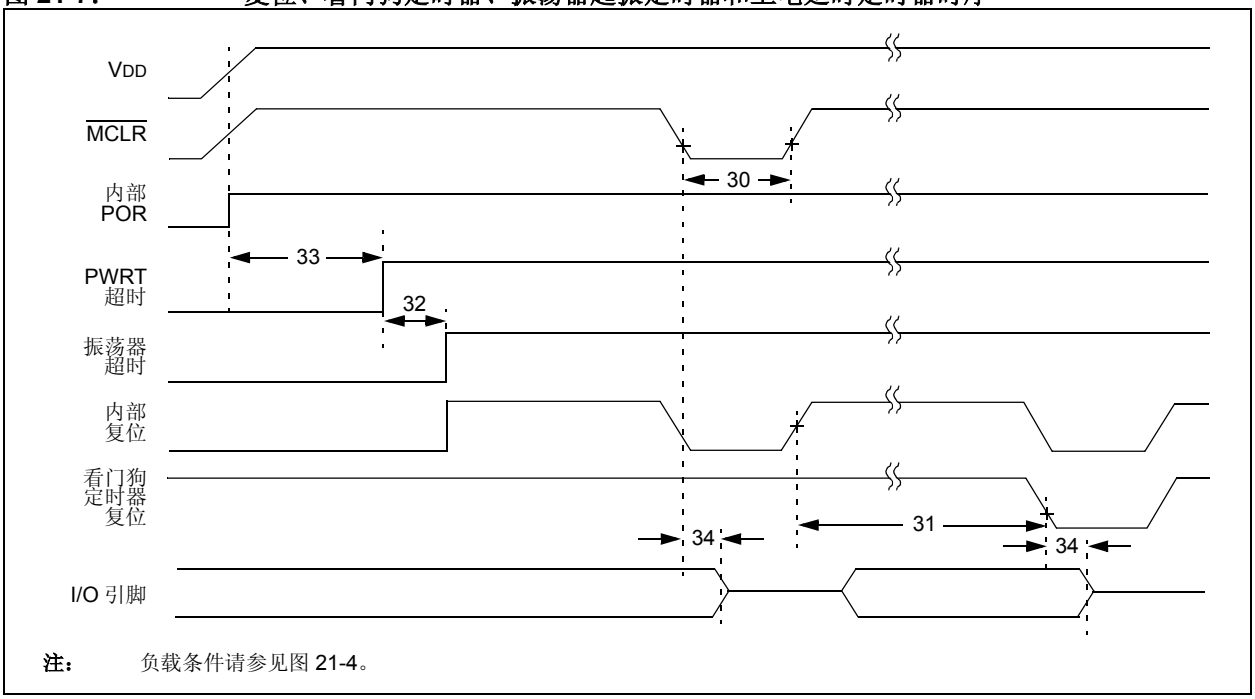
参数编号	符号	特性		最小值	典型值	最大值	单位	条件
10	TosH2ckL	OSC1↑到 CLKO↓		—	75	200	ns	（注 1）
11	TosH2ckH	OSC1↑到 CLKO↑		—	75	200	ns	（注 1）
12	TckR	CLKO 上升时间		—	35	100	ns	（注 1）
13	TckF	CLKO 下降时间		—	35	100	ns	（注 1）
14	TckL2ioV	CLKO↓到端口输出有效		—	—	0.5 TCY + 20	ns	（注 1）
15	TioV2ckH	CLKO↑之前端口输入有效		0.25 TCY +25	—	—	ns	（注 1）
16	TckH2ioI	CLKO↑之后端口输入保持		0	—	—	ns	（注 1）
17	TosH2ioV	OSC1↑（Q1 周期）到端口输出有效		—	50	150	ns	
18	TosH2ioI	OSC1↑（Q2 周期）到端口输入无效（I/O 输入保持时间）	PIC18**F**XXXX	100	—	—	ns	
18A			PIC18**LF**XXXX	200	—	—	ns	VDD = 2.0V
19	TioV2osH	端口输入有效到 OSC1↑（I/O 输入建立时间）		0	—	—	ns	
20	TioR	端口输出上升时间	PIC18**F**XXXX	—	10	25	ns	
20A			PIC18**LF**XXXX	—	—	60	ns	VDD = 2.0V
21	TioF	端口输出下降时间	PIC18**F**XXXX	—	10	25	ns	
21A			PIC18**LF**XXXX	—	—	60	ns	VDD = 2.0V
22†	TINP	INT 引脚高电平或低电平时间		TCY	—	—	ns	
23†	TRBP	RB7:RB4 电平变化中断高电平或低电平时间		TCY	—	—	ns	

† 这些参数是与任何内部时钟边沿无关的异步事件。

注 1： 测量是在 RC 模式下进行的，其中 CLKO 输出为 4 x TOSC。

超前信息

图 21-7： 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



注： 负载条件请参见图 21-4。

图 21-8：　　欠压复位时序

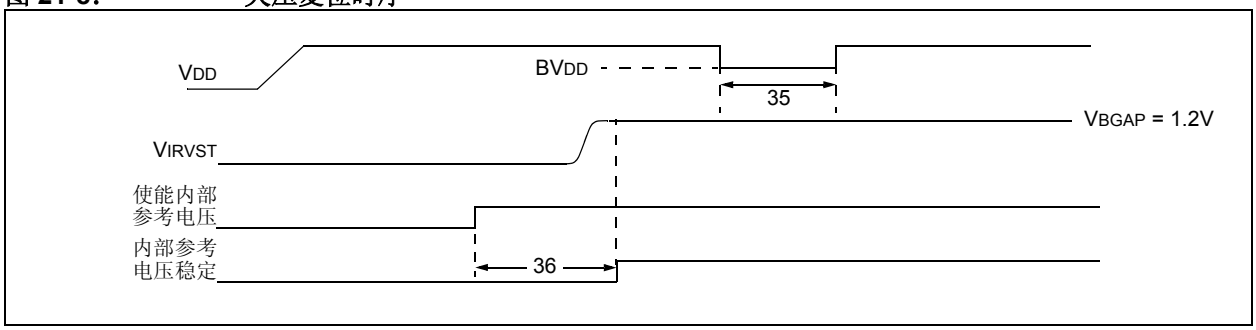


表 21-10：　　复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位要求

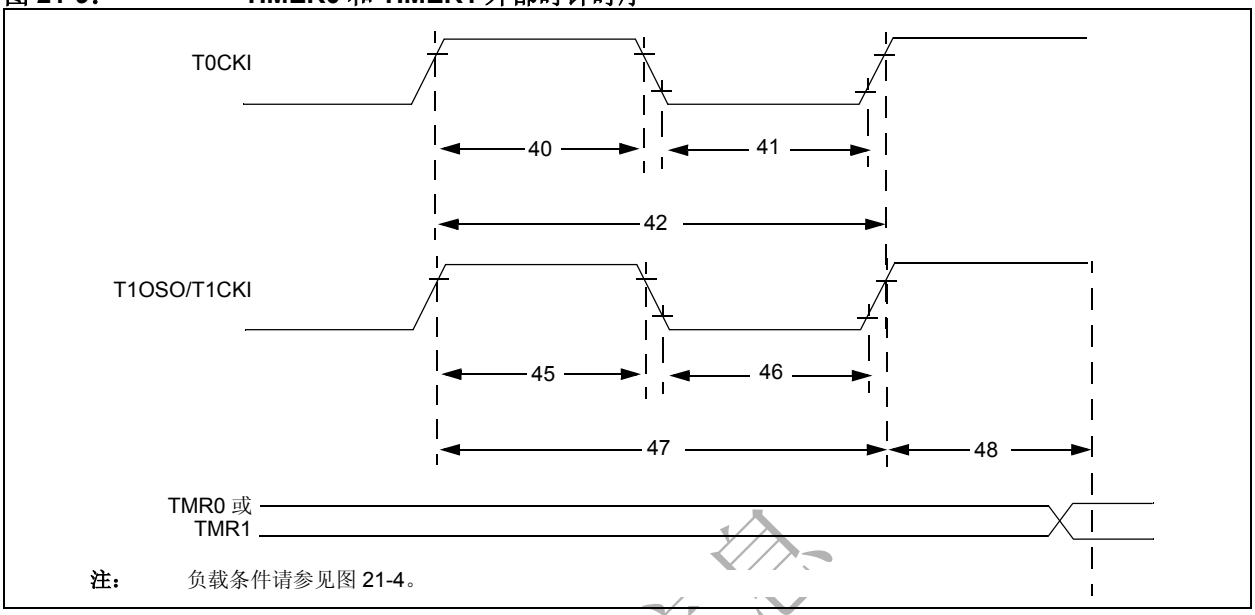
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
30	TmcL	$\overline{\text{MCLR}}$ 脉冲宽度 （低电平）	2	—	—	µs	
31	TWDT	看门狗定时器超时周期 （无后分频器）	—	4.00	TBD	ms	
32	TOST	振荡器起振定时器周期	1024 TOSC	—	1024 TOSC	—	TOSC = OSC1 周期
33	TPWRT	上电延时定时器周期	—	65.5	TBD	ms	
34	TIOZ	自 $\overline{\text{MCLR}}$ 低电平或看门狗定时器复位起 I/O 处于高阻状态的时间	—	2	—	µs	
35	TBOR	欠压复位脉冲宽度	200	—	—	µs	VDD ≤ BVDD （见 D005）
36	TIRVST	内部参考电压稳定时间	—	20	50	µs	
37	TLVD	低压检测脉冲宽度	200	—	—	µs	VDD ≤ VLVD
38	TCSD	CPU 启动时间	5	—	10	µs	
39	TIOBST	INTRC 稳定时间	—	1	—	ms	

图注：　TBD = 待定

超前信息

© 2007 Microchip Technology Inc.

图 21-9：　　TIMER0 和 TIMER1 外部时钟时序



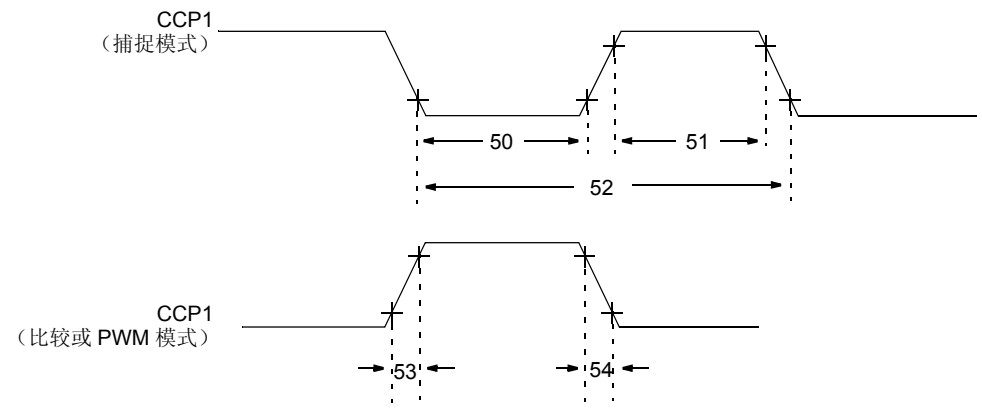
注：　负载条件请参见图 21-4。

表 21-11：　　TIMER0 和 TIMER1 外部时钟要求

参数编号	符号	特性			最小值	最大值	单位	条件
40	Tt0H	T0CKI 高电平脉冲宽度		无预分频器	0.5 TCY + 20	—	ns	
				有预分频器	10	—	ns	
41	Tt0L	T0CKI 低电平脉冲宽度		无预分频器	0.5 TCY + 20	—	ns	
				有预分频器	10	—	ns	
42	Tt0P	T0CKI 周期		无预分频器	TCY + 10	—	ns	
				有预分频器	取较大值：20 ns 或（TCY + 40)/N	—	ns	N = 预分频值（1, 2, 4,..., 256）
45	Tt1H	T1CKI 高电平时间	同步，无预分频器		0.5 TCY + 20	—	ns	
			同步，有预分频器	PIC18**F**XXXX	10	—	ns	
				PIC18**LF**XXXX	25	—	ns	VDD = 2.0V
			异步	PIC18**F**XXXX	30	—	ns	
				PIC18**LF**XXXX	50	—	ns	VDD = 2.0V
46	Tt1L	T1CKI 低电平时间	同步，无预分频器		0.5 TCY + 5	—	ns	
			同步，有预分频器	PIC18**F**XXXX	10	—	ns	
				PIC18**LF**XXXX	25	—	ns	VDD = 2.0V
			异步	PIC18**F**XXXX	30	—	ns	
				PIC18**LF**XXXX	50	—	ns	VDD = 2.0V
47	Tt1P	T1CKI 输入周期	同步		取较大值：20 ns 或（TCY + 40)/N	—	ns	N = 预分频值（1、2、4 和 8）
			异步		60	—	ns	
	Ft1	T1CKI 振荡器输入频率范围			DC	50	kHz	
48	Tcke2tmrl	从外部 T1CKI 时钟边沿到定时器递增的延时			2 TOSC	7 TOSC	—	

超前信息

图 21-10：　　捕捉 / 比较 /PWM 时序 （CCP 模块）



注：　负载条件请参见图 21-4。

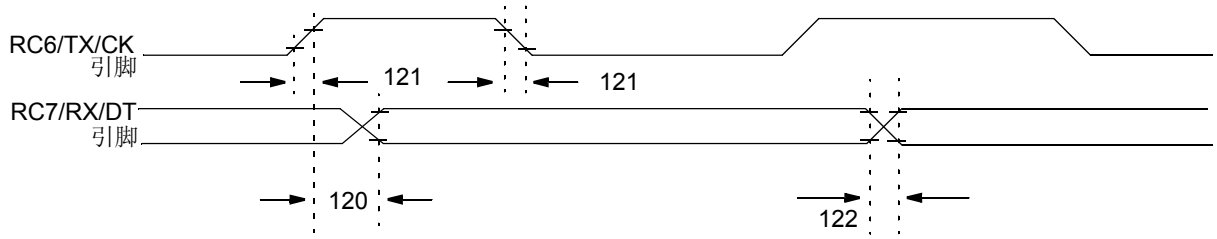
表 21-12：　　捕捉 / 比较 /PWM 要求

参数编号	符号	特性			最小值	最大值	单位	条件
50	TccL	CCP1 输入低电平时间	无预分频器		0.5 TCY + 20	—	ns	
			有预分频器	PIC18**F**XXXX	10	—	ns	
				PIC18**LF**XXXX	20	—	ns	VDD = 2.0V
51	TccH	CCP1 输入高电平时间	无预分频器		0.5 TCY + 20	—	ns	
			有预分频器	PIC18**F**XXXX	10	—	ns	
				PIC18**LF**XXXX	20	—	ns	VDD = 2.0V
52	TccP	CCP1 输入周期			$\frac{3\ T_{CY} + 40}{N}$	—	ns	N = 预分频值（1、 4 或 16）
53	TccR	CCP1 输出上升时间		PIC18**F**XXXX	—	25	ns	
				PIC18**LF**XXXX	—	45	ns	VDD = 2.0V
54	TccF	CCP1 输出下降时间		PIC18**F**XXXX	—	25	ns	
				PIC18**LF**XXXX	—	45	ns	VDD = 2.0V

超前信息

© 2007 Microchip Technology Inc.

图 21-11：　　EUSART 同步发送 （主控 / 从动）时序



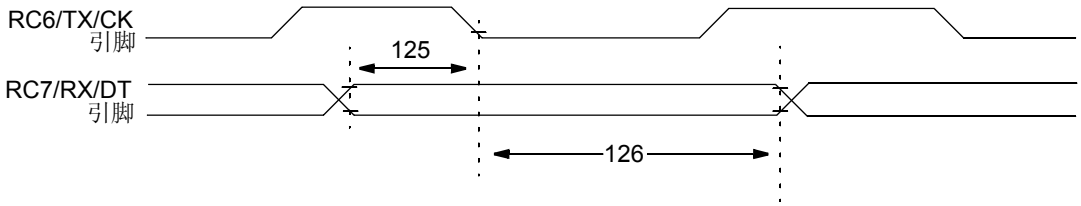
注：　负载条件请参见图 21-4。

表 21-13：　　EUSART 同步发送要求

参数编号	符号	特性		最小值	最大值	单位	条件
120	TckH2dtV	SYNC XMIT （主控和从动） 从时钟高电平到数据输出有效	PIC18**F**XXXX	—	40	ns	
			PIC18**LF**XXXX	—	100	ns	VDD = 2.0V
121	Tckrf	时钟输出上升时间和下降时间 （主控模式）	PIC18**F**XXXX	—	20	ns	
			PIC18**LF**XXXX	—	50	ns	VDD = 2.0V
122	Tdtrf	数据输出上升时间和下降时间	PIC18**F**XXXX	—	20	ns	
			PIC18**LF**XXXX	—	50	ns	VDD = 2.0V

超前信息

图 21-12：　　EUSART 同步接收 （主控 / 从动）时序



注：　负载条件请参见图 21-4。

表 21-14：　　EUSART 同步接收要求

参数编号	符号	特性	最小值	最大值	单位	条件
125	TDTV2CKL	SYNC RCV （主控和从动） 在 CK↓之前数据的保持时间 （DT 保持时间）	10	—	ns	
126	TCKL2DTL	在 CK↓之后数据的保持时间 （DT 保持时间）	15	—	ns	

图 21-13：　　USB 信号时序

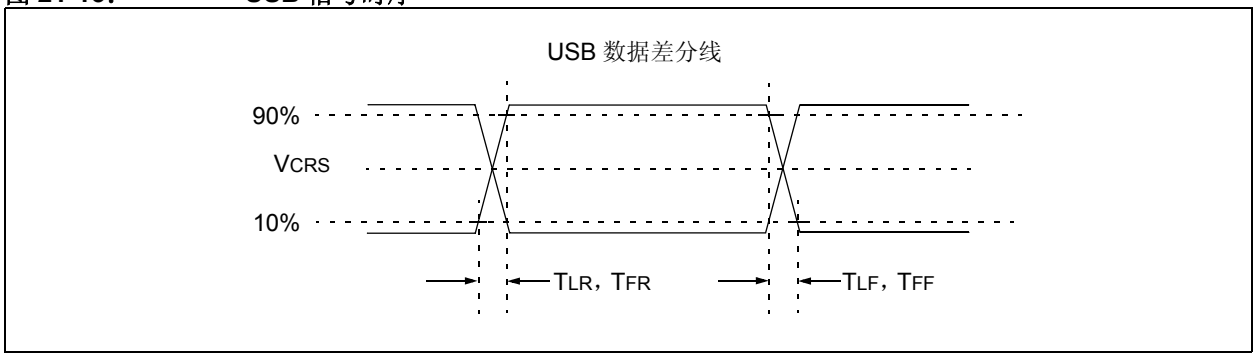


表 21-15：　　USB 低速时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
	TLR	转换上升时间	75	—	300	ns	CL = 200 至 600 pF
	TLF	转换下降时间	75	—	300	ns	CL = 200 至 600 pF
	TLRFM	上升 / 下降时间匹配	80	—	125	%	

表 21-16：　　USB 全速要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
	TFR	转换上升时间	4	—	20	ns	CL = 50 pF
	TFF	转换下降时间	4	—	20	ns	CL = 50 pF
	TFRFM	上升 / 下降时间匹配	90	—	111.1	%	

超前信息

© 2007 Microchip Technology Inc.

表 21-17： A/D 转换器特性： PIC18F2450/4450 （工业级）
PIC18LF2450/4450（工业级）

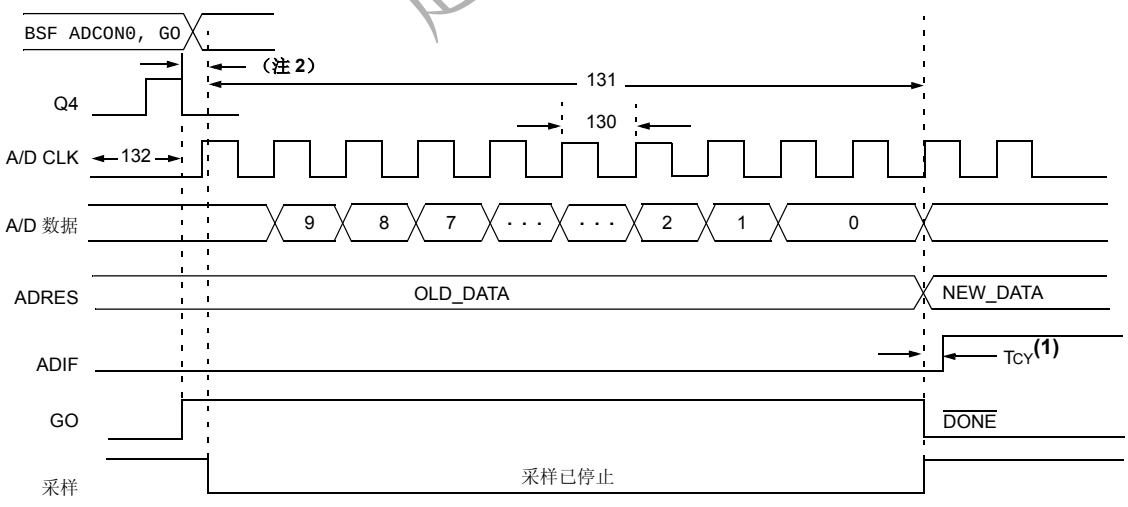
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
A01	NR	分辨率	—	—	10	位	ΔVREF ≥ 3.0V
A03	EIL	积分线性误差	—	—	<±1	LSb	ΔVREF ≥ 3.0V
A04	EDL	微分线性误差	—	—	<±1	LSb	ΔVREF ≥ 3.0V
A06	EOFF	失调误差	—	—	<±1.5	LSb	ΔVREF ≥ 3.0V
A07	EGN	增益误差	—	—	<±1	LSb	ΔVREF ≥ 3.0V
A10	—	单调性	保证 (1)			—	VSS ≤ VAIN ≤ VREF
A20	ΔVREF	参考电压范围（VREFH – VREFL）	1.8 3	— —	— —	V V	VDD < 3.0V VDD ≥ 3.0V
A21	VREFH	参考电压高电平	VSS	—	VREFH	V	
A22	VREFL	参考电压低电平	VSS – 0.3V	—	VDD – 3.0V	V	
A25	VAIN	模拟输入电压	VREFL	—	VREFH	V	
A30	ZAIN	模拟电源阻抗的推荐值	—	—	2.5	kΩ	
A50	IREF	VREF 输入电流 (2)	— —	— —	5 150	μA μA	在采集 VAIN 期间。 在 A/D 转换周期期间。

注 1： A/D 转换结果不会因输入电压的增加而减小，并且不会丢失代码。

2： VREFH 电流来自作为 VREFH 源的 RA3/AN3/VREF+ 引脚或 VDD。
VREFL 电流来自作为 VREFL 源的 RA2/AN2/VREF- 引脚或 VSS。

超前信息

图 21-14： A/D 转换时序



注 1： 如果选择 A/D 时钟源作为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

2： 这是最小的 RC 延时 （典型值为 100 ns），在这段时间保持电容与模拟输入端是断开的。

表 21-18： A/D 转换要求

参数编号	符号	特性		最小值	最大值	单位	条件
130	TAD	A/D 时钟周期	PIC18**F**XXXX	0.7	25.0(1)	µs	基于 TOSC， VREF ≥ 3.0V
			PIC18**LF**XXXX	1.4	25.0(1)	µs	VDD = 2.0V， 基于 TOSC， VREF 满量程
			PIC18**F**XXXX	TBD	1	µs	A/D RC 模式
			PIC18**LF**XXXX	TBD	3	µs	VDD = 2.0V， A/D RC 模式
131	TCNV	转换时间 （不包括采集时间） (2)		11	12	TAD	
132	TACQ	采集时间 (3)		1.4 TBD	— —	µs µs	-40°C 至 +85°C 0°C ≤ 至 ≤ +85°C
135	TSWC	转换 → 采样的切换时间		—	（注 4）		
137	TDIS	电容放电时间		0.2	—	µs	

图注： TBD = 待定

注 1： A/D 时钟周期取决于器件频率和 TAD 时钟分频比。

2： 可在后续 TCY 周期内读 ADRES 寄存器。

3： 转换完成后当电压满量程变化时 （VDD 至 VSS 或 VSS 至 VDD），保持电容采样一个 “新”输入电压所需的时间。在输入通道上的电源阻抗 （*R*S）为 50Ω。

4： 在器件时钟的下一个周期。

超前信息

 © 2007 Microchip Technology Inc.

22.0 直流和交流特性图表

当前没有可用图表。

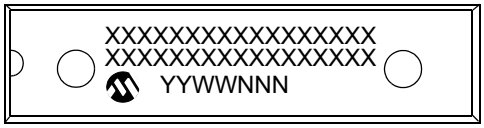
注：

 © 2007 Microchip Technology Inc.

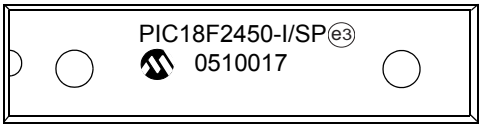
23.0 封装信息

封装标识信息

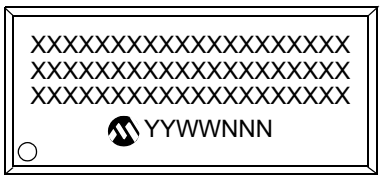
28 引脚 PDIP（窄型 DIP）



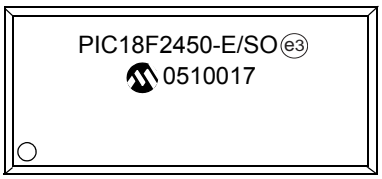
示例



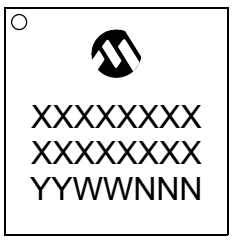
28 引脚 SOIC



示例



28 引脚 QFN



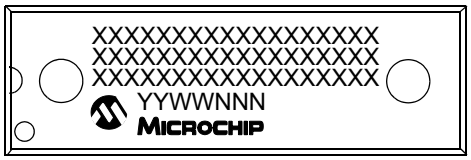
示例



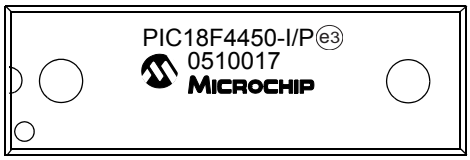
图注：	XX...X	客户指定信息
	Y	年代码 （公历年份的最后一位）
	YY	年代码 （公历年份的最后两位）
	WW	星期代码 （1 月 1 日的星期代码为 “01”）
	NNN	字母数字的追踪代码
	(e3)	雾锡 （Sn）的无铅 JEDEC 标志
	*	本封装为无铅封装。在封装的外部包装上可以找到无铅 JEDEC 标志 ((e3))。
注：	如果 Microchip 器件编号不能在一行中完全标出，它将换行继续标出。因此限制了用户指定信息的可用字符数。	

封装标识信息 （续）

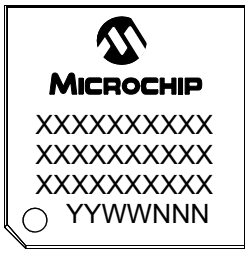
40 引脚 PDIP



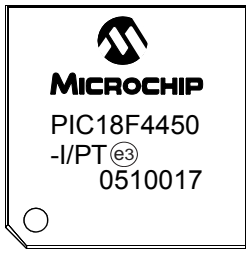
示例



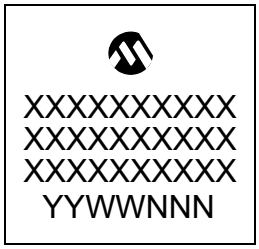
44 引脚 TQFP



示例



44 引脚 QFN



示例



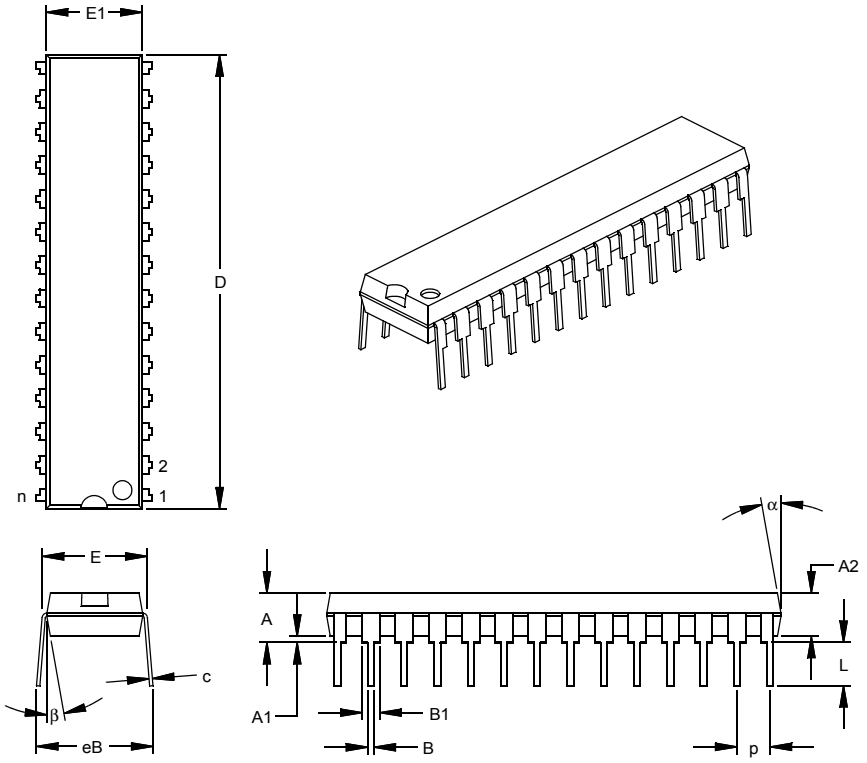
 © 2007 Microchip Technology Inc.

23.1 封装详细信息

以下部分将介绍各种封装的技术细节。

28 引脚窄型塑封双列直插式封装 （SP）——300 mil 主体 （PDIP）

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

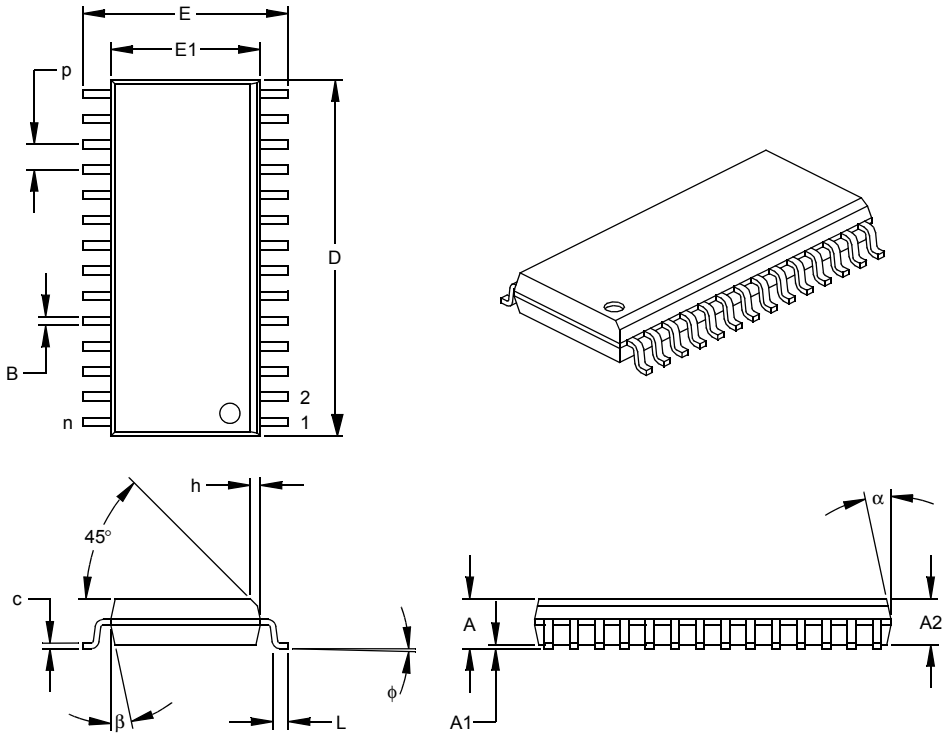


	单位	英寸 *			毫米		
	尺寸范围	最小	正常	最大	最小	正常	最大
引脚数	n		28			28	
引脚间距	p		.100			2.54	
顶端到固定面距离	A	.140	.150	.160	3.56	3.81	4.06
塑模封装厚度	A2	.125	.130	.135	3.18	3.30	3.43
塑模底面到固定面高度	A1	.015			0.38		
肩与肩宽度	E	.300	.310	.325	7.62	7.87	8.26
塑模封装宽度	E1	.275	.285	.295	6.99	7.24	7.49
总长度	D	1.345	1.365	1.385	34.16	34.67	35.18
引脚尖到固定面高度	L	.125	.130	.135	3.18	3.30	3.43
引脚厚度	c	.008	.012	.015	0.20	0.29	0.38
引脚上部宽度	B1	.040	.053	.065	1.02	1.33	1.65
引脚下部宽度	B	.016	.019	.022	0.41	0.48	0.56
总排列间距 §	eB	.320	.350	.430	8.13	8.89	10.92
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

***** 控制参数
§ 重要特性
注：
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸（0.254 毫米）。
等同于 JEDEC 号: MO-095
图号 C04-070

28 引脚塑封小外形封装 （SO）——宽条， 300 mil 主体 （SOIC）

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



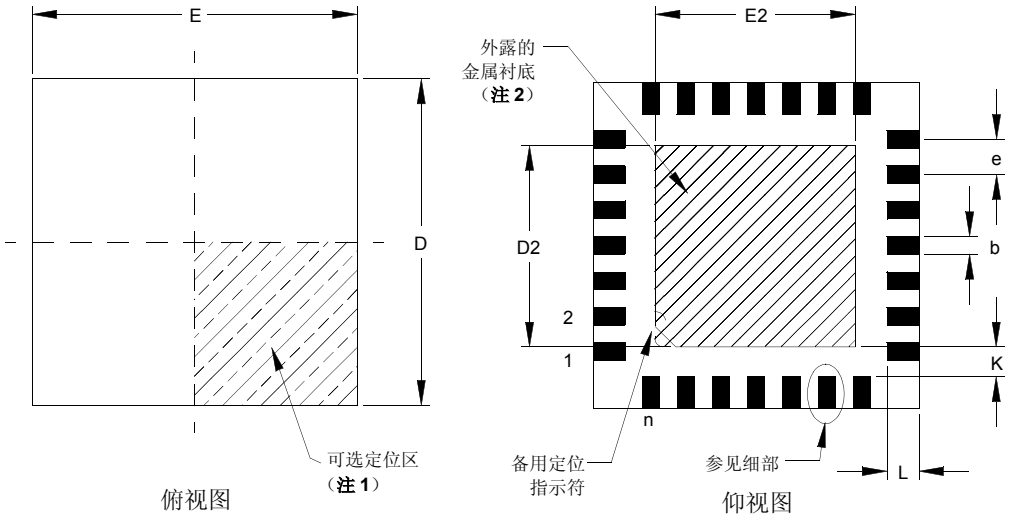
单位		英寸 *			毫米		
尺寸范围		最小	正常	最大	最小	正常	最大
引脚数	n		28			28	
引脚间距	p		.050			1.27	
总高度	A	.093	.099	.104	2.36	2.50	2.64
塑模封装厚度	A2	.088	.091	.094	2.24	2.31	2.39
悬空间隙 §	A1	.004	.008	.012	0.10	0.20	0.30
总宽度	E	.394	.407	.420	10.01	10.34	10.67
塑模封装宽度	E1	.288	.295	.299	7.32	7.49	7.59
总长度	D	.695	.704	.712	17.65	17.87	18.08
斜面距离	h	.010	.020	.029	0.25	0.50	0.74
底脚长度	L	.016	.033	.050	0.41	0.84	1.27
底脚倾斜角	φ	0	4	8	0	4	8
引脚厚度	c	.009	.011	.013	0.23	0.28	0.33
引脚宽度	B	.014	.017	.020	0.36	0.42	0.51
塑模顶部锥度	α	0	12	15	0	12	15
塑模底部锥度	β	0	12	15	0	12	15

***** 控制参数
§ 重要特性
注：
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸（0.254 毫米）。
等同于 JEDEC 号: MS-013
图号 C04-052

© 2007 Microchip Technology Inc.

28 引脚塑封正方扁平无脚封装 （ML） 6x6 mm 主体 （QFN）——具有 0.55 mm 触点长度（Saw Singulated）

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



单位		英寸			毫米 *		
尺寸范围		最小	正常	最大	最小	正常	最大
引脚数	n		28			28	
引脚间距	e	.026 BSC			0.65 BSC		
总高度	A	.031	.035	.039	0.80	0.90	1.00
悬空间隙	A1	.000	.001	.002	0.00	0.02	0.05
触点厚度	A3	.008 REF			0.20 REF		
总宽度	E	.232	.236	.240	5.90	6.00	6.10
外露金属衬底宽度	E2	.153	.167	.169	3.89	4.24	4.29
总长度	D	.232	.236	.240	5.90	6.00	6.10
外露金属衬底长度	D2	.153	.167	.169	3.89	4.24	4.29
触点宽度	β	.009	.011	.013	0.23	0.28	0.33
触点长度 §	L	.018	.022	.024	0.45	0.55	0.65
连接到外露金属衬底的触点 §	K	.008	–	–	0.20	–	–

***** 控制参数

§ 重要特性

注：

1. 引脚 1 的可见定位功能可能不同，但必须在阴影区域内。

2. 外露金属衬底尺寸随管芯叶片大小而变化。

BSC: 基本尺寸。理论上显示的是没有公差的精确值。
参见 ASME Y14.5M

REF: 参考尺寸，通常无公差，仅供参考。
参见 ASME Y14.5M

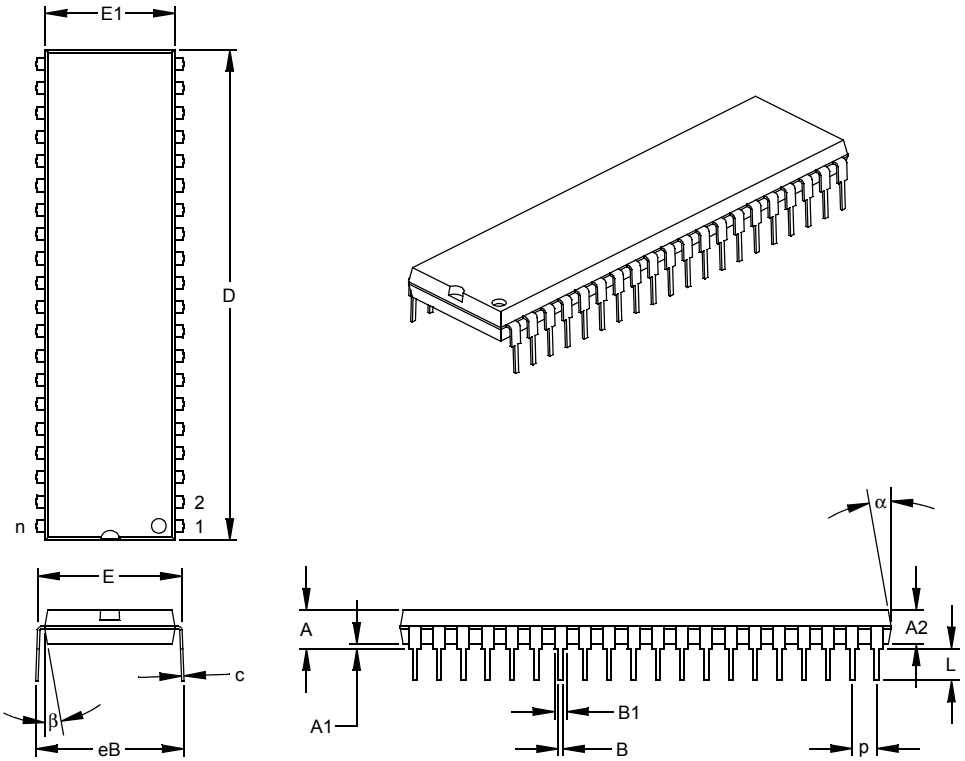
等同于 JEDEC 号: MO-220

图号 C04-105

修订于 09-12-05

40 引脚塑封双列直插式封装 （P）——600 mil 主体 （PDIP）

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



单位		英寸 *			毫米		
尺寸范围		最小	正常	最大	最小	正常	最大
引脚数	n		40			40	
引脚间距	p		.100			2.54	
顶端到固定面距离	A	.160	.175	.190	4.06	4.45	4.83
塑模封装厚度	A2	.140	.150	.160	3.56	3.81	4.06
塑模底面到固定面高度	A1	.015			0.38		
肩与肩宽度	E	.595	.600	.625	15.11	15.24	15.88
塑模封装宽度	E1	.530	.545	.560	13.46	13.84	14.22
总长度	D	2.045	2.058	2.065	51.94	52.26	52.45
引脚尖到固定面高度	L	.120	.130	.135	3.05	3.30	3.43
引脚厚度	c	.008	.012	.015	0.20	0.29	0.38
引脚上部宽度	B1	.030	.050	.070	0.76	1.27	1.78
引脚下部宽度	B	.014	.018	.022	0.36	0.46	0.56
总排列间距 §	eB	.620	.650	.680	15.75	16.51	17.27
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

***** 控制参数

§ 重要特性

注：

尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸（0.254 毫米）。

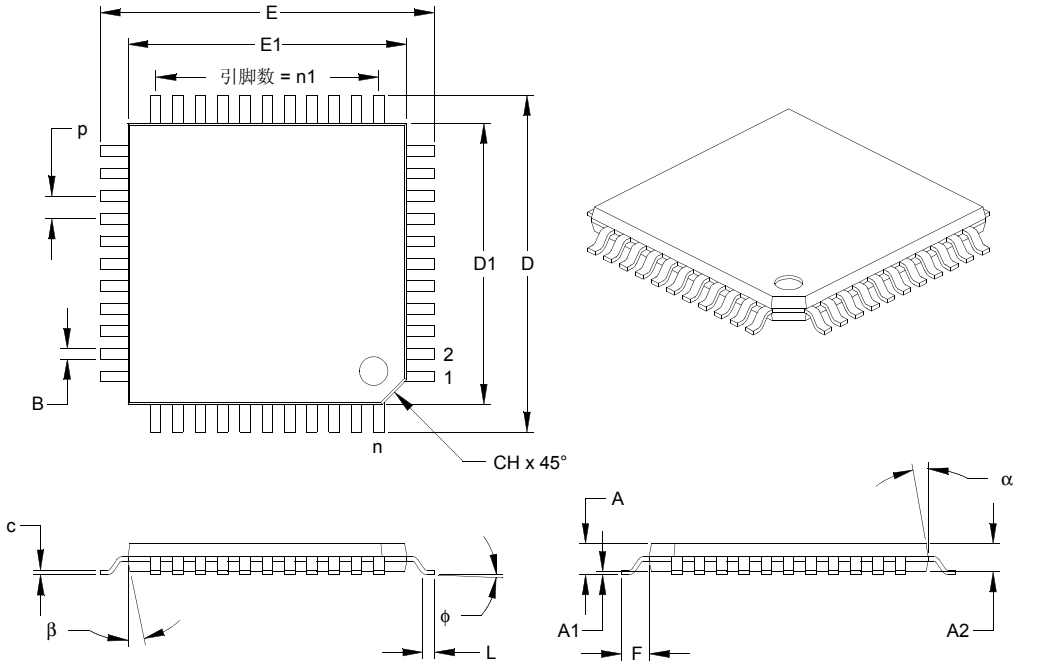
等同于 JEDEC 号: MO-011

图号 C04-016

 © 2007 Microchip Technology Inc.

44 引脚塑封薄型正方扁平封装 （PT） 10x10x1 mm 主体， 1.0/0.10 mm 引脚形式 （TQFP）

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



	单位	英寸			毫米 *		
	尺寸范围	最小	正常	最大	最小	正常	最大
引脚数	n		44			44	
引脚间距	p		.031			0.80	
每侧引脚数	n1		11			11	
总高度	A	.039	.043	.047	1.00	1.10	1.20
塑模封装厚度	A2	.037	.039	.041	0.95	1.00	1.05
悬空间隙	A1	.002	.004	.006	0.05	0.10	0.15
底脚长度	L	.018	.024	.030	0.45	0.60	0.75
引脚投影长度（参考）	F		.039 REF.			1.00 REF.	
底脚倾斜角	ϕ	0	3.5	7	0	3.5	7
总宽度	E	.463	.472	.482	11.75	12.00	12.25
总长度	D	.463	.472	.482	11.75	12.00	12.25
塑模封装宽度	E1	.390	.394	.398	9.90	10.00	10.10
塑模封装长度	D1	.390	.394	.398	9.90	10.00	10.10
引脚厚度	c	.004	.006	.008	0.09	0.15	0.20
引脚宽度	B	.012	.015	.017	0.30	0.38	0.44
引脚 1 切角斜面	CH	.025	.035	.045	0.64	0.89	1.14
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

***** 控制参数

注：

尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸（0.254 毫米）。

REF: 参考尺寸，通常无公差，仅供参考。

参见 ASME Y14.5M

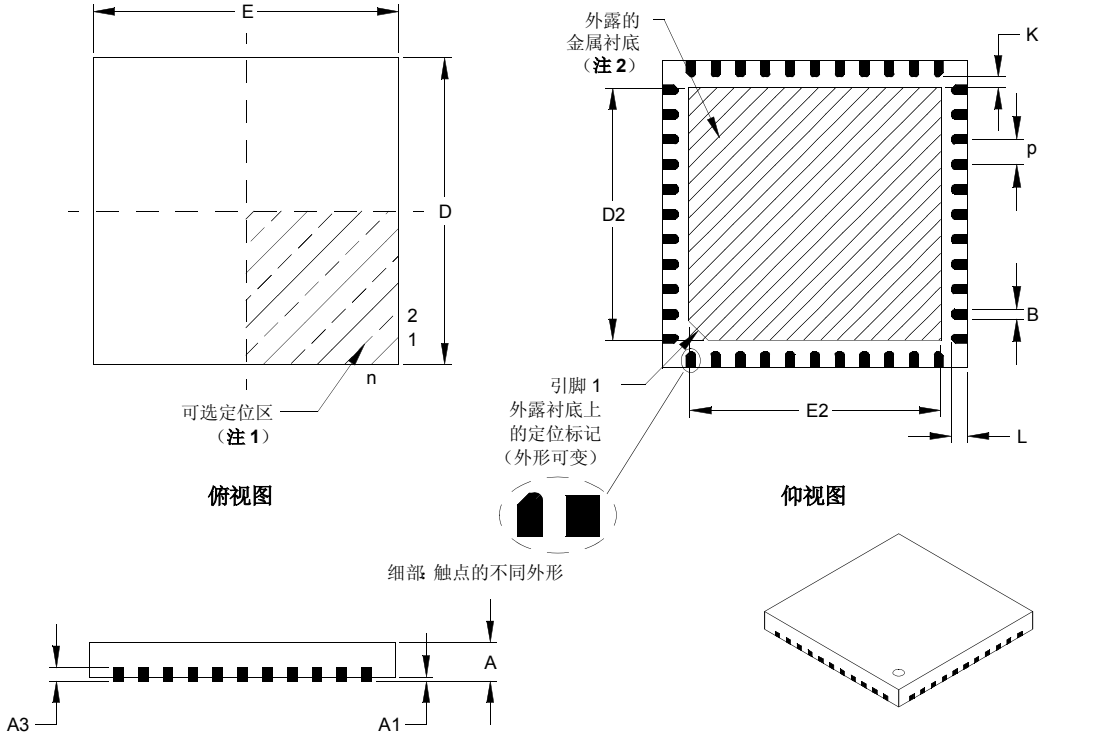
等同于 JEDEC 号: MS-026

图号 C04-076

修订于 07-22-05

44 引脚塑封正方扁平无脚封装 （ML） 8x8 mm 主体 （QFN）

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



单位		英寸			毫米 *		
尺寸范围		最小	正常	最大	最小	正常	最大
触点数	n		44			44	
触点间距	p	.026 BSC			0.65 BSC		
总高度	A	.031	.035	.039	0.80	0.90	1.00
悬空间隙	A1	.000	.001	.002	0	0.02	0.05
引脚顶部到塑模底部距离	A3	.010 REF			0.25 REF		
总宽度	E	.309	.315	.321	7.85	8.00	8.15
外露金属衬底宽度	E2	.236	.258	.260	5.99	6.55	6.60
总长度	D	.309	.315	.321	7.85	8.00	8.15
外露金属衬底长度	D2	.236	.258	.260	5.99	6.55	6.60
触点宽度	B	.008	.013	.013	0.20	0.33	0.35
触点长度 §	L	.014	.016	.019	0.35	0.40	0.48
连接到外露金属衬底的触点 §	K	.014	-	-	0.20	-	-

***** 控制参数

§ 重要特性

注：

1. 引脚 1 的可见定位功能可能不同，但必须在阴影区域内。

2. 外露金属衬底尺寸随管芯叶片大小而变化。

BSC: 基本尺寸。理论上显示的是没有公差的精确值。
参见 ASME Y14.5M

REF: 参考尺寸，通常无公差，仅供参考。
参见 ASME Y14.5M

等同于 JEDEC 号: M0-220

图号 C04-103

修订于 09-12-05

© 2007 Microchip Technology Inc.

附录 A： 版本历史

版本 A （2006 年 1 月）

PIC18F2450/4450 器件的原始数据手册。

附录 B： 器件差异

表 B-1 为本数据手册中所列器件之间的差异。

表 B-1： 器件差异

功能部件	PIC18F2450	PIC18F4450
程序存储器 （字节）	16384	16384
程序存储器 （指令）	8192	8192
中断源	13	13
I/O 端口	端口 A, B, C, (E)	端口 A, B, C, D, E
捕捉 / 比较 /PWM 模块	1	1
10 位模数转换模块	10 路输入通道	13 路输入通道
封装	28 引脚 SDIP 28 引脚 SOIC 28 引脚 QFN	40 引脚 PDIP 44 引脚 TQFP 44 引脚 QFN

附录 C： 转换注意事项

本附录讨论了器件从老版本升级到本数据手册中所列版本时的注意事项。这些变化通常是由于所采用的工艺技术之间的差异引起的，正如从PIC16C74A到PIC16C74B的升级。

不适用

附录 D： 从基本器件移植到增强型器件

本节讨论如何从基本器件（如 PIC16C5X）移植到增强型 MCU 器件 （如 PIC18FXXX）。

下表列出了本系列器件与 PIC16C5X 单片机系列相比所作的修改：

当前未提供

© 2007 Microchip Technology Inc.

附录 E： 从中档器件移植到增强型器件

在 *AN716，“Migrating Designs from PIC16C74A/74B to PIC18C442”* 中详细讨论了中档 MCU 器件（即 PIC16CXXX）与增强型器件（即 PIC18FXXX）之间的差异。虽然所讨论的变化都是针对特定器件的，但是通常适用于从中档器件移植到增强型器件的所有情况。

上述应用笔记的文献编号为 DS00716。

附录 F： 从高档器件移植到增强型器件

在 *AN726，“PIC17CXXX to PIC18CXXX Migration”* 中详细讨论了从高档 MCU 器件（即 PIC17CXXX）移植到增强型器件（即 PIC18FXXX）的步骤以及这两类器件之间的差异。上述应用笔记的文献编号为 DS00726。

注：

 © 2007 Microchip Technology Inc.

索引

A

A/D 173
ADCON0 寄存器 173
ADCON1 寄存器 173
ADCON2 寄存器 173
ADRESH 寄存器 173, 176
ADRESL 寄存器 173
CCP1 触发器的使用 182
采集要求 178
放电 181
模拟端口引脚，配置 180
配置模块 177
特殊事件触发信号 （CCP1） 182
相关的寄存器 182
选择和配置采集时间 179
在功耗管理模式下的操作 180
转换 181
转换器特性 291
转换器中断，配置 177
转换时钟 （TAD） 179
转换要求 292
转换状态 （GO/DONE 位） 176
ADCON0 寄存器 173
GO/DONE 位 176
ADCON1 寄存器 173
ADCON2 寄存器 173
ADDFSR 254
ADDLW 217
ADDWF 217
ADDWFC 218
ADDULNK 254
ADRESH 寄存器 173
ADRESL 寄存器 173, 176
ANDLW 218
ANDWF 219

B

BC 219
BCF 220
BN 220
BNC 221
BNN 221
BNOV 222
BNZ 222
BOR。*请参见*欠压复位。
BOV 225
BRA 223
BSF 223
BTFSC 224
BTFSS 224
BTG 225
BZ 226
版本历史 303
比较 （CCP 模块） 125
CCP1 引脚配置 125
CCPR1 寄存器 125
Timer1 模式选择 125
软件中断 125
特殊事件触发器 125
相关的寄存器 126
编程，器件指令 211
变更通知客户服务 315
表读 / 表写 56
表指针操作 （表） 76
不同情形下的延时 （表） 45
捕捉 / 比较 /PWM （CCP） 123
CCP 模式和定时器资源 124
CCPR1H 寄存器 124
CCPR1L 寄存器 124
比较模式。*请参见*比较。
捕捉模式。*请参见*捕捉。
模块配置 124
捕捉 （CCP 模块） 124
CCP1 引脚配置 124
CCPR1H:CCPR1L 寄存器 124
软件中断 124
相关的寄存器 126
预分频器 124

C

CALL 226
CALLW 255
C 编译器
MPLAB C18 262
MPLAB C30 262
CLRF 227
CLRWDT 227
COMF 228
CPFSEQ 228
CPFSGT 229
CPFSLT 229
CPU 的特殊功能 189
程序存储器
查找表 56
代码保护 207
复位向量 53
和扩展指令集 70
映射和堆栈 （图） 53
指令 58
双字 58
中断向量 53
程序计数器 54
PCLATH 和 PCLATU 寄存器 54
PCL、 PCH 和 PCU 寄存器 54
程序校验和代码保护 206
相关的寄存器 206
从高档器件移植到增强型器件 305
从基本器件移植到增强型器件 304
从中档器件移植到增强型器件 305
存储器编程要求 278
存储器构成 53
程序存储器 53
数据存储器 59

D

DAW 230
DCFSNZ 231
DECF 230
DECFSZ 231
代码保护 189
代码示例
16 x 16 无符号乘法程序 84
16 x 16 有符号乘法程序 84
8 x 8 无符号乘法程序 83
8 x 8 有符号乘法程序 83
擦除闪存程序存储器的一行 78
初始化 PORTA 99

初始化 PORTB 101
初始化 PORTC 104
初始化 PORTD 107
初始化 PORTE 109
读闪存程序存储器的一个字 77
改变捕捉预分频比 124
将 STATUS、 WREG 和 BSR 寄存器的值
保存在 RAM 中 97
快速寄存器堆栈 56
使用间接寻址清零 RAM （Bank 1） 67
使用偏移量计算 GOTO 56
使用 Timer1 中断服务实现实时时钟 119
写闪存程序存储器 80–81
单电源 ICSP 编程 210
低压 ICSP 编程。*请参见*单电源 ICSP 编程。
电气特性 265
对标准 PIC MCU 的影响 258
堆栈满 / 下溢复位 56

E

EUSART
波特率发生器 （BRG） 157
波特率误差，计算 157
波特率，异步模式 158
采样 157
高波特率选择 （BRGH 位） 157
相关的寄存器 157
在功耗管理模式下的操作 157
自动波特率检测 160
同步从动模式 171
发送 171
接收 172
相关的寄存器，发送 171
相关的寄存器，接收 172
同步主控模式 168
发送 168
接收 170
相关的寄存器，发送 169
相关的寄存器，接收 170
异步模式 162
发送器 162
间隔字符序列 167
接收器 164
设置带有地址检测功能的 9 位模式 164
同步间隔自动唤醒 166
相关的寄存器，发送 163
相关的寄存器，接收 165

F

FSCM。*请参见*故障保护时钟监视器。
返回地址堆栈 54
和相关的寄存器 54
返回堆栈指针 （STKPTR） 55
访问栈顶 54
封装信息 295
标识 295
详细信息 297
复位 41, 189
欠压复位 （BOR） 189
上电复位 （POR） 189
上电延时定时器 （PWRT） 189
振荡器起振定时器 （OST） 189
复位定时器 45
PLL 锁定延时定时器 45
上电延时定时器 （PWRT） 45
振荡器起振定时器 （OST） 45

G

GOTO 232
高 / 低压检测 183
电流消耗 185
典型应用 186
复位的影响 187
工作原理 184
休眠期间 187
启动时间 185
设置 185
特性 280
相关的寄存器 187
应用 186
功耗管理模式 33
对各种时钟源的影响 32
多条 SLEEP 命令 34
和 A/D 操作 180
汇总 （表） 33
进入 33
空闲 37
空闲模式
PRI_IDLE 38
RC_IDLE 39
SEC_IDLE 38
时钟源 33
时钟转换和状态指示 34
退出空闲和休眠模式 39
通过复位 39
通过 WDT 超时 39
通过中断 39
在没有振荡器起振延时的情况下 40
休眠 37
选择 33
运行模式 34
PRI_RUN 34
RC_RUN 35
SEC_RUN 34
公式
A/D 采集时间 178
A/D 最小充电时间 178
计算所需要的最小 A/D 采集时间 178
固件指令 211
故障保护时钟监视器 189, 204
功耗管理模式下的中断 205
POR 或从休眠中唤醒 205
振荡器故障期间的 WDT 204

H

HLVD。*请参见*高 / 低压检测。 183
汇编器
MPASM 汇编器 262

I

I/O 端口 99
I/O 引脚说明
PIC18F2450 12
PIC18F4450 16
ID 单元 189
ID 地址单元 209
INCF 232
INCFSZ 233
INFSNZ 233
INTCON 寄存器 87
RBIF 位 101
INTOSC， INTRC。*请参见*内部振荡器模块。
IORLW 234

 © 2007 Microchip Technology Inc.

IORWF 234
IPR 寄存器 94

J

寄存器
ADCON0 （A/D 控制 0） 173
ADCON1 （A/D 控制 1） 174
ADCON2 （A/D 控制 2） 175
BAUDCON （波特率控制） 156
BDnSTAT （缓冲描述器 n 状态， CPU 模式） 139
BDnSTAT （缓冲描述器 n 状态， SIE 模式） 140
CCP1CON （捕捉 / 比较 /PWM 控制） 123
CONFIG1H （配置 1 高字节） 192
CONFIG1L （配置 1 低字节） 191
CONFIG2H （配置 2 高字节） 194
CONFIG2L （配置 2 低字节） 193
CONFIG3H （配置 3 高字节） 195
CONFIG4L （配置 4 低字节） 196
CONFIG5H （配置 5 高字节） 197
CONFIG5L （配置 5 低字节） 197
CONFIG6H （配置 6 高字节） 198
CONFIG6L （配置 6 低字节） 198
CONFIG7 （配置 7 低字节） 199
CONFIG7 （配置 7 高字节） 199
DEVID1 （器件 ID 1） 200
DEVID2 （器件 ID 2） 200
EECON1 （存储器控制 1） 75
HLVDCON （高 / 低压检测控制） 183
INTCON2 （中断控制 2） 88
INTCON3 （中断控制 3） 89
INTCON （中断控制） 87
IPR1 （外设中断优先级 1） 94
IPR2 （外设中断优先级 2） 95
OSCCON （振荡器控制） 31
PIE1 （外设中断允许 1） 92
PIE2 （外设中断允许 2） 93
PIR1 （外设中断请求 （标志） 1） 90
PIR2 （外设中断请求 （标志） 2） 91
PORTE 109
RCON （复位控制） 42, 96
RCSTA （接收状态和控制寄存器） 155
STATUS 66
STKPTR （堆栈指针） 55
T0CON （Timer0 控制） 111
T1CON （Timer1 控制） 115
T2CON （Timer2 控制） 121
TXSTA （发送状态和控制寄存器） 154
UCFG （USB 配置） 132
UCON （USB 控制） 130
WDTCON （看门狗定时器控制） 202
UEIR （USB 错误中断状态） 146
UEPn （USB 端点 n 控制） 135
UIE （USB 错误中断允许） 147
UIE （USB 中断允许） 145
UIR （USB 中断状态） 144
USTAT （USB 状态） 134
寄存器的复位状态 48
寄存器文件汇总 63–65
间接寻址 68
交流特性
内部 RC 精度 284
交流 （时序）特性 281
参数符号体系 281
器件时序规范的负载条件 282
时序条件 282
温度和电压规范 282
晶振 / 陶瓷谐振器 25
绝对极限参数值 265

K

开发支持 261
看门狗定时器 （WDT） 189, 201
编程注意事项 201
控制寄存器 201
相关的寄存器 202
振荡器故障期间 204
勘误表 6
客户支持 315
空闲模式 37
快速寄存器堆栈 56
框图
A/D 176
EUSART 发送 162
EUSART 接收 164
PIC18F2450 9
PIC18F4450 10
PLL （HS 模式） 26
PWM 工作原理 （简化） 127
Timer0 （16 位模式） 112
Timer0 （8 位模式） 112
Timer2 122
Timer1 116
Timer1 （16 位读 / 写模式） 116
USB 外设和选项 129
USB 中断逻辑归结 143
比较模式工作原理 125
表读操作 73
表写操作 74
捕捉模式工作原理 124
带外部输入的高 / 低压检测 184
读闪存程序存储器 77
对闪存程序存储器的表写操作 79
故障保护时钟监视器 204
看门狗定时器 201
模拟输入模型 177
片上复位电路 41
器件时钟 24
通用 I/O 端口 99
外部上电复位电路 （VDD 缓慢上电的情况） 43
中断逻辑 86
扩展指令集 253
ADDFSR 254
ADDULNK 254
CALLW 255
MOVSF 255
MOVSS 256
PUSHL 256
SUBFSR 257
SUBULNK 257
和使用 MPLAB IDE 工具 260
使用注意事项 258
语法 253

L

LFSR 235
立即数变址寻址模式 258
立即数变址寻址
和标准 PIC18 指令 258

M

MOVF 235
MOVFF 236

MOVLB 236
MOVLW 237
MOVSF 255
MOVSS 256
MOVWF 237
MPLAB ASM30 汇编器、链接器和库管理器 262
MPLAB ICD 2 在线调试器 263
MPLAB ICE 2000 高性能通用在线仿真器 263
MPLAB ICE 4000 高性能通用在线仿真器 263
MPLAB PM3 器件编程器 263
MPLAB 集成开发环境软件 261
MPLINK 目标链接器 /MPLIB 目标库管理器 262
MULLW 238
MULWF 238
脉宽调制。*请参见* PWM （CCP 模块）。
模数转换器。*请参见* A/D。

N

NEGF 239
NOP 239
内部 RC 振荡器
    与 WDT 一起使用 201
内部振荡器模块
    INTHS、 INTXT、 INTCKO 和 INTIO 模式 27

P

PICSTART 2 开发编程器 264
PICSTART Plus 开发编程器 264
PIE 寄存器 92
PIR 寄存器 90
PLL 倍频器 26
    HSPLL、 XTPLL、 ECPLL 和 ECPIO 振荡器模式 26
PLL 锁定延时定时器 45
POP 240
PORTA
    I/O 汇总 100
    LATA 寄存器 99
    PORTA 寄存器 99
    TRISA 寄存器 99
    相关的寄存器 100
PORTB
    I/O 汇总 102
    LATB 寄存器 101
    PORTB 寄存器 101
    RB7:RB4 电平变化中断标志 （RBIF 位） 101
    TRISB 寄存器 101
    相关的寄存器 103
PORTC
    I/O 汇总 105
    LATC 寄存器 104
    PORTC 寄存器 104
    TRISC 寄存器 104
    相关的寄存器 106
PORTD
    I/O 汇总 108
    LATD 寄存器 107
    PORTD 寄存器 107
    TRISD 寄存器 107
    相关的寄存器 108
PORTE
    I/O 汇总 110
    LATE 寄存器 109
    PORTE 寄存器 109
    TRISE 寄存器 109
    相关的寄存器 110
POR。*请参见*上电复位。
PRI_IDLE 模式 38
PRI_RUN 模式 34
PWM （CCP 模块）
    TMR2 到 PR2 匹配 127
    频率 / 分辨率示例 128
    设置 PWM 工作模式 128
    相关的寄存器 128
    占空比 127
    周期 127
PUSH 240
PUSH 和 POP 指令 55
PUSHL 256
配置寄存器保护 209
配置位 190

Q

Q 时钟 128
器件差异 303
器件概述 7
    其他特殊功能 8
    特性 （表） 9
    新的内核功能 7
欠压复位 （BOR） 44
    检测 44
    软件使能 44
    在休眠模式下禁止 44

R

RAM。*请参见*数据存储器。
RC_IDLE 模式 39
RC_RUN 模式 35
RCALL 241
RCON 寄存器
    初始化时位的状态 48
RESET 241
RETFIE 242
RETLW 242
RETURN 243
RLCF 243
RLNCF 244
RRCF 244
RRNCF 245
软件模拟器 （MPLAB SIM） 262

S

SEC_IDLE 模式 38
SEC_RUN 模式 34
SETF 245
SLEEP 246
SWAPF 248
SUBFSR 257
SUBFWB 246
SUBLW 247
SUBWF 247
SUBWFB 248
SUBULNK 257
闪存程序存储器 73
    表读与表写 73
    表指针
        基于操作的范围 76
    表指针范围 76
    擦除 78
    擦除序列 78
    代码保护期间的操作 81
    读取 77
    防止误写操作的保护措施 81

 © 2007 Microchip Technology Inc.

控制寄存器 74
EECON1 和 EECON2 74
TABLAT （表锁存）寄存器 76
TBLPTR （表指针）寄存器 76
相关的寄存器 81
写入 79
写顺序 79
写校验 81
意外终止写操作 81
上电复位 （POR） 43
上电延时 32
上电延时定时器 （PWRT） 32, 45
时序图
A/D 转换 291
BRG 溢出时序 161
CLKO 和 I/O 284
EUSART 同步发送 （主控 / 从动） 289
EUSART 同步接收 （主控 / 从动） 289
POR 在 PLL 使能时的延时时序
（MCLR 连接到 VDD） 47
PWM 输出 127
Timer0 和 Timer1 外部时钟 287
USB 信号 290
捕捉 / 比较 /PWM （CCP） 288
从空闲模式唤醒进入运行模式的转换 38
从 RC_RUN 模式切换到 PRI_RUN 模式的转换 36
从 SEC_RUN 模式切换到 PRI_RUN 模式的
转换 （HSPLL） 35
从休眠模式唤醒的转换 （HSPLL） 37
到 RC_RUN 模式的转换 36
低压检测 （VDIRMAG = 0） 185
发送间隔字符序列 167
复位、看门狗定时器 （WDT）、振荡器起振定时器
（OST）和上电延时定时器 （PWRT） 285
高 / 低压检测特性 280
高压检测 （VDIRMAG = 1） 186
故障保护时钟监视器 205
缓慢上升时间 （MCLR 连接到 VDD，
VDD 电压上升时间 > TPWRT） 47
进入空闲模式的转换 38
进入 SEC_RUN 模式的转换 34
进入休眠模式的转换 37
欠压复位 （BOR） 286
上电延时时序 （MCLR 连接到 VDD，
VDD 电压上升时间 TPWRT） 46
上电延时时序 （MCLR 未连接到 VDD），情形 1 46
上电延时时序 （MCLR 未连接到 VDD），情形 2 46
时钟 / 指令周期 57
双速启动转换的时序图
（从 INTOSC 切换到 HSPLL） 203
同步发送 168
同步发送 （由 TXEN 位控制） 169
同步接收 （主控模式， SREN） 170
外部时钟 （除 PLL 外的所有模式） 283
休眠模式下的自动唤醒位 （WUE） 166
异步发送 163
异步发送 （背靠背） 163
异步接收 165
正常工作模式下的自动唤醒位 （WUE） 166
自动波特率计算 161
时序图和规范 283
CLKO 和 I/O 要求 285
EUSART 同步发送要求 289
EUSART 同步接收要求 289
PLL 时钟 284
Timer0 和 Timer1 外部时钟要求 287
USB 低速要求 290
USB 全速要求 290
捕捉 / 比较 /PWM 要求 （CCP） 288
复位、看门狗定时器、振荡器起振定时器、上电延时定时
器和欠压复位要求 286
外部时钟要求 283
时钟源 30
使用 OSCCON 寄存器选择 30
数据存储器 59
PIC18F2450/4450 器件的映射 60
USB RAM 59
存储区选择寄存器 （BSR） 59
和扩展指令集 70
快速操作存储区 61
特殊功能寄存器 62
映射图 62
通用寄存器 61
数据寻址模式 67
固有和立即数寻址 67
间接寻址 67
立即数变址寻址 70
BSR 操作 72
受影响的指令 70
映射快速操作存储区 72
使能了扩展指令集的对比寻址模式 71
直接寻址 67
双速启动 189, 203
双字指令
示例情形 58
所有寄存器的初始化状态 49

T

T0CON 寄存器
PSA 位 113
T0CS 位 112
T0PS2:T0PS0 位 113
T0SE 位 112
TBLRD 249
TBLWT 250
Timer0 111
定时器的 16 位读写模式 112
工作原理 112
时钟源边沿选择 （T0SE 位） 112
时钟源选择 （T0CS 位） 112
相关的寄存器 113
溢出中断 113
预分频器 113
切换分配 113
预分频器。*请参见*预分频器， Timer0。
Timer2 121
PR2 寄存器 127
TMR2 到 PR2 匹配中断 127
工作原理 121
输出 122
相关的寄存器 122
中断 122
Timer1 115
16 位读 / 写模式 117
TMR1H 寄存器 115
TMR1L 寄存器 115
复位，使用特殊事件触发器输出 （CCP） 118
工作原理 116
相关的寄存器 119, 126
溢出中断 115
用作实时时钟 118
振荡器 115, 117

布线注意事项 118
低功耗选项 117
使用 Timer1 作为时钟源 117
中断 118
TQFP 封装和特殊功能 209
TSTFSZ 251
TXSTA 寄存器
BRGH 位 157
特殊 ICPORT 功能 209
特殊事件触发器。*请参见*比较 （CCP 模块）。
同步间隔字符自动唤醒 166
通用串行总线 59
RAM 136
存储器映射 136
UFRMH:UFRML 寄存器 136
地址寄存器 （UADDR） 136
电源 150
电源模式 148
仅自供电 148
仅总线供电 148
主要使用自供电的双重供电 148
端点控制 135
分层的框架 150
分类规范和驱动程序 151
概述 129, 150
固件和驱动程序 148
缓冲器描述符 137
BDnSTAT 寄存器 （CPU 模式） 138
BDnSTAT 寄存器 （SIE 模式） 140
不同缓冲模式中的分配 142
存储器映射 141
地址验证 140
寄存器汇总 142
乒乓缓冲 141
示例 137
所有权 137
状态和配置 137
字节计数 140
缓冲器描述符表 137
枚举 151
描述符 151
内部收发器 131
内部稳压器 133
乒乓缓冲配置 133
上拉电阻 133
使能眼图测试 133
输出使能监视器 133
速度 151
外部收发器 131
相关的寄存器 149
帧 150
振荡器要求 148
帧编号寄存器 136
中断 143
和 USB 事务 143
传输类型 150
状态和控制 130

U

USB
模块规范 279
内部稳压器规范 279
USB 的振荡器设置 27
USB。*请参见*通用串行总线。

W

WWW，在线支持 6
外部时钟输入 26

X

XORLW 251
XORWF 252
休眠
OSC1 和 OSC2 引脚的状态 32
休眠模式 37

Y

延时时序 45
引脚功能
MCLR/VPP/RE3 ..... 12, 16
NC/ICCK/ICPGC 21
NC/ICDT/ICPGD 21
NC/ICPORTS 21
NC/ICRST/ICVPP 21
OSC1/CLKI 12, 16
OSC2/CLKO/RA6 ..... 12, 16
RA0/AN0 ..... 13, 17
RA1/AN1 ..... 13, 17
RA2/AN2/VREF- 13, 17
RA3/AN3/VREF+ 13, 17
RA4/T0CKI/RCV 13, 17
RA5/AN4/HLVDIN 13, 17
RB0/AN12/INT0 ..... 14, 18
RB1/AN10/INT1 ..... 14, 18
RB2/AN8/INT2/VMO 14, 18
RB3/AN9/VPO 14, 18
RB4/AN11/KBI0 ..... 14, 18
RB5/KBI1/PGM 14, 18
RB6/KBI2/PGC 14, 18
RB7/KBI3/PGD 14, 18
RC0/T1OSO/T1CKI 15, 19
RC1/T1OSI/UOE 15, 19
RC2/CCP1 ..... 15, 19
RC4/D-/VM 15, 19
RC5/D+/VP 15, 19
RC6/TX/CK 15, 19
RC7/RX/DT 15, 19
RD0 20
RD1 20
RD2 20
RD3 20
RD4 20
RD5 20
RD6 20
RD7 20
RE0/AN5 21
RE1/AN6 21
RE2/AN7 21
VDD 15, 21
VSS 15, 21
VUSB 15, 21
硬件乘法器 83
工作原理 83
简介 83
性能比较 83
预分频器， Timer0 113
比率选择 （T0PS2:T0PS0 位） 113
分配 （PSA 位） 113
预分频器， Timer2 128
预分频器， WDT

 © 2007 Microchip Technology Inc.

比率选择 （T0PS2:T0PS0 位） 113
分配 （PSA 位） 113

Z

在线串行编程 （ICSP） 189, 209
在线调试器 209
增强型通用同步 / 异步收发器（EUSART）。*请参见* EUSART。
振荡器配置 23
EC 23
ECIO 23
ECPIO 23
ECPLL 23
HS 23
HSPLL 23
INTCKO 23
INTHS 23
INTIO 23
INTXT 23
XT 23
XTPLL 23
内部振荡器模块 27
振荡器模式和 USB 操作 24
振荡器起振定时器 （OST） 32, 45
振荡器切换 30
振荡器选择 189
振荡器转换 30
振荡器， Timer1 115
直接寻址 68
指令集 211
ADDLW 217
ADDWF 217
ADDWFC 218
ADDWF （立即数变址寻址模式） 259
ANDLW 218
ANDWF 219
BC 219
BCF 220
BN 220
BNC 221
BNN 221
BNOV 222
BNZ 222
BOV 225
BRA 223
BSF 223
BSF （立即数变址寻址模式） 259
BTFSC 224
BTFSS 224
BTG 225
BZ 226
CALL 226
CLRF 227
CLRWDT 227
COMF 228
CPFSEQ 228
CPFSGT 229
CPFSLT 229
DAW 230
DCFSNZ 231
DECF 230
DECFSZ 231
GOTO 232
INCF 232
INCFSZ 233
INFSNZ 233
IORLW 234
IORWF 234
LFSR 235
MOVF 235
MOVFF 236
MOVLB 236
MOVLW 237
MOVWF 237
MULLW 238
MULWF 238
NEGF 239
NOP 239
POP 240
PUSH 240
RCALL 241
RESET 241
RETFIE 242
RETLW 242
RETURN 243
RLCF 243
RLNCF 244
RRCF 244
RRNCF 245
SETF 245
SETF （立即数变址寻址模式） 259
SLEEP 246
SWAPF 248
SUBFWB 246
SUBLW 247
SUBWF 247
SUBWFB 248
TBLRD 249
TBLWT 250
TSTFSZ 251
XORLW 251
XORWF 252
标准指令集 211
操作码字段说明 212
通用格式 213
指令周期 57
时钟分配 57
指令流 / 流水线 57
直流和交流特性
图表 293
直流特性 276
掉电和供电电流 268
供电电压 267
中断 85
USB 85
中断的现场保护 97
中断源 189
A/D 转换完成 177
INTn 引脚 97
PORTB，电平变化中断 97
TMR0 97
TMR0 溢出 113
TMR1 溢出 115
TMR2 到 PR2 匹配 （PWM） 127
比较完成 （CCP） 125
捕捉完成 （CCP） 124
电平变化中断 （RB7:RB4） 101
中断，标志位
电平变化中断 （RB7:RB4）标志 （RBIF 位） 101
主清零复位 （MCLR） 43
转换注意事项 304
专用 ICD/ICSP 端口 209

状态寄存器 66

 © 2007 Microchip Technology Inc.

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、应用笔记和样本程序、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（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、 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，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://support.microchip.com获得网上技术支持。

读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。请填写以下信息，并从下面各方面提出您对本文档的意见。

致: TRC 经理　　总页数 _____

关于: 读者反馈

发自: 姓名 _____

公司 _____

地址 _____

国家 / 省份 / 城市 / 邮编 _____

电话: (_____) _____ - _____　　传真: (_____) _____ - _____

应用（选填）:

您希望收到回复吗？是 ____ 否____

器件: PIC18F2450/4450　　文献编号: DS39760A_CN

问题:

1. 本文档中哪些部分最有特色？

2. 本文档是否满足了您的软硬件开发要求？如何满足的？

3. 您认为本文档的组织结构便于理解吗？如果不便于理解，那么问题何在？

4. 您认为本文档应该添加哪些内容以改善其结构和主题？

5. 您认为本文档中可以删减哪些内容，而又不会影响整体使用效果？

6. 本文档中是否存在错误或误导信息？如果存在，请指出是什么信息及其具体页数。

7. 您认为本文档还有哪些方面有待改进？

© 2007 Microchip Technology Inc.

PIC18F2450/4450 产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

器件编号	X	/XX	XXX
器件	温度范围	封装	模式

器件	PIC18F2450(1)， PIC18F4450(1)， PIC18F2450T(2)， PIC18F4450(2)； VDD 范围为 4.2V 至 5.5V PIC18LF2450(1)， PIC18LF4450(1)， PIC18LF2450(2)， PIC18LF4450(2)； VDD 范围为 2.0V 至 5.5V
温度范围	I = -40°C 至 +85°C（工业级） E = -40°C 至 +125°C（扩展级）
封装	PT = TQFP（薄型正方扁平封装） SO = SOIC SP = 窄型塑封 DIP P = PDIP ML = QFN
模式	QTP、SQTP、代码或特殊要求 （其他情况空白）

示例：

a） PIC18LF4450-I/P 301 表示工业级温度， PDIP 封装，扩展级 VDD 范围， QTP 模式 #301。

b） PIC18LF2450-I/SO 表示工业级温度， SOIC 封装，扩展级 VDD 范围。

c） PIC18F4450-I/P 表示工业级温度，PDIP 封装，普通 VDD 范围。

注 1： F = 标准电压范围
LF = 宽电压范围

2： T = 卷带式，仅限 TQFP 封装。

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://support.microchip.com
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

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
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

科科莫 Kokomo
Kokomo, IN
Tel: 1-765-864-8360
Fax: 1-765-864-8387

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

加拿大多伦多 Toronto
Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Habour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 – 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 – 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 – 福州
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

中国 – 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 – 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 – 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 – 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 – 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 – 顺德
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

中国 – 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 – 西安
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

台湾地区 – 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 – 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 – 新竹
Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韩国 Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

12/08/06

© 2007 Microchip Technology Inc.