

PIC18F87J11 系列
数据手册

采用纳瓦技术的
64/80 引脚高性能
1 Mb 闪存单片机

© 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、 KEELOQ 徽标、 microID、 MPLAB、 PIC、 PICmicro、 PICSTART、 PRO MATE、 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、 Linear Active Thermistor、 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、 dsSPEAK、 ECAN、 ECONOMONITOR、 FanSense、 FlexROM、 fuzzyLAB、 In-Circuit Serial Programming、 ICSP、 ICEPIC、 Mindi、 MiWi、 MPASM、 MPLAB Certified 徽标、 MPLIB、 MPLINK、 PICkit、 PICDEM、 PICDEM.net、 PICLAB、 PICtail、 PowerCal、 PowerInfo、 PowerMate、 PowerTool、 REAL ICE、 rfLAB、 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 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC® MCU 与 dsPIC® DSC、 KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品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。

PIC18F87J11 系列

采用纳瓦技术的 64/80 引脚高性能 1 Mb 闪存单片机

灵活的振荡器结构：

- 4 种晶振模式，包括高精度 PLL
- 两种外部时钟模式，频率最高可达 48 MHz
- 内部振荡电路：
  - 提供 8 个可由用户选择的频率：从 31 kHz 到 8 MHz
  - 在和 PLL 结合使用时提供完整的时钟速率范围（从 31 kHz 到 32 MHz）
  - 用户可对该电路进行调节以补偿频率漂移
- 辅助振荡器使用 Timer1（工作频率为 32 kHz）
- 故障保护时钟监视器：
  - 当时钟停止时可使器件安全关闭

外设特点：

- PORTB 和 PORTC 上的高灌 / 拉电流峰 - 峰值为 25 mA/25 mA
- 4 个可编程外部中断
- 4 个输入电平变化中断
- 1 个 8/16 位定时器 / 计数器
- 2 个 8 位定时器 / 计数器
- 2 个 16 位定时器 / 计数器
- 2 个捕捉 / 比较 /PWM（CCP）模块
- 3 个增强型捕捉 / 比较 /PWM（ECCP）模块
  - 1 个、2 个或 4 个 PWM 输出
  - 可选择极性
  - 可编程死区时间
  - 自动关闭和自动重启
- 2 个主同步串行端口（Master Synchronous Serial Port，MSSP）模块支持 3 线 SPI（共 4 种模式）和 I^2C™ 主控和从动模式
- 2 个增强型 USART 模块：
  - 支持 RS-485、RS-232 和 LIN 1.2
  - 遇到起始位时自动唤醒
  - 自动波特率检测

外设特点（续）：

- 具有 16 条地址线的 8 位并行主端口 / 增强型并行从端口（Parallel Master Port/Enhanced Parallel Slave Port，PMP/EPSP）
- 2 个带输入多路开关的模拟比较器
- 多达 15 路通道的 10 位模数转换器模块（Analog-to-Digital Converter，A/D）：
  - 自动采集功能
  - 可在休眠模式下进行转换

外部存储器总线（仅限 80 引脚的器件）：

- 最大 2 MB 的寻址能力
- 8 位或 16 位接口
- 12 位、16 位和 20 位寻址模式

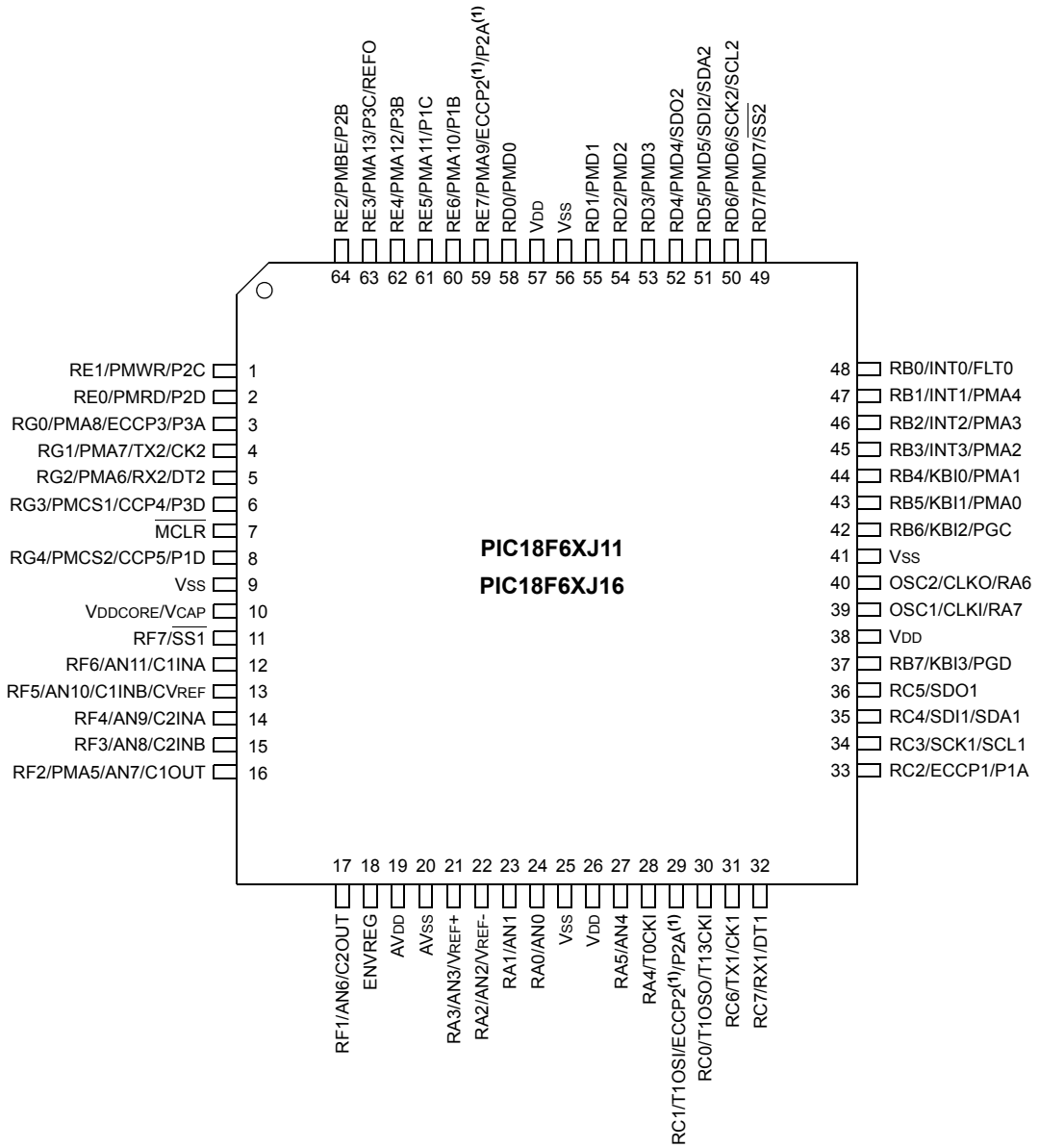
单片机的特殊性能：

- 低功耗、高速 CMOS 闪存技术
- 针对重入代码（Re-Entrant Code）进行了优化的 C 编译器架构
- 功耗管理特性：
  - 运行：CPU 打开，外设打开
  - 空闲：CPU 关闭，外设打开
  - 休眠：CPU 关闭，外设关闭
- 中断优先级
- 可在软件控制下自编程
- 8 x 8 单周期硬件乘法器
- 扩展的看门狗定时器（WDT）：
  - 从 4 ms 到 131s 的可编程周期
- 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）
- 工作电压范围：2.0V 到 3.6V
- 允许 5.5V 输入（仅限数字引脚）
- 片上 2.5V 稳压器
- 闪存程序存储器可承受 10,000 次擦 / 写，且数据保存时间长达 20 年

器件	闪存程序存储器（字节）	SRAM 数据存储器（字节）	I/O	10 位 A/D（通道数）	CCP/ECCP（PWM）	MSSP		EUSART	比较器	8/16 位定时器	外部总线	PMP/EPSP
							SPI	主器件 I^2C™					
PIC18F66J11	64 KB	3930	52	11	2/3	2	有	有	2	2	2/3	无	有
PIC18F66J16	96 KB	3930	52	11	2/3	2	有	有	2	2	2/3	无	有
PIC18F67J11	128 KB	3930	52	11	2/3	2	有	有	2	2	2/3	无	有
PIC18F86J11	64 KB	3930	68	15	2/3	2	有	有	2	2	2/3	有	有
PIC18F86J16	96 KB	3930	68	15	2/3	2	有	有	2	2	2/3	有	有
PIC18F87J11	128 KB	3930	68	15	2/3	2	有	有	2	2	2/3	有	有

引脚图

64 引脚 TQFP

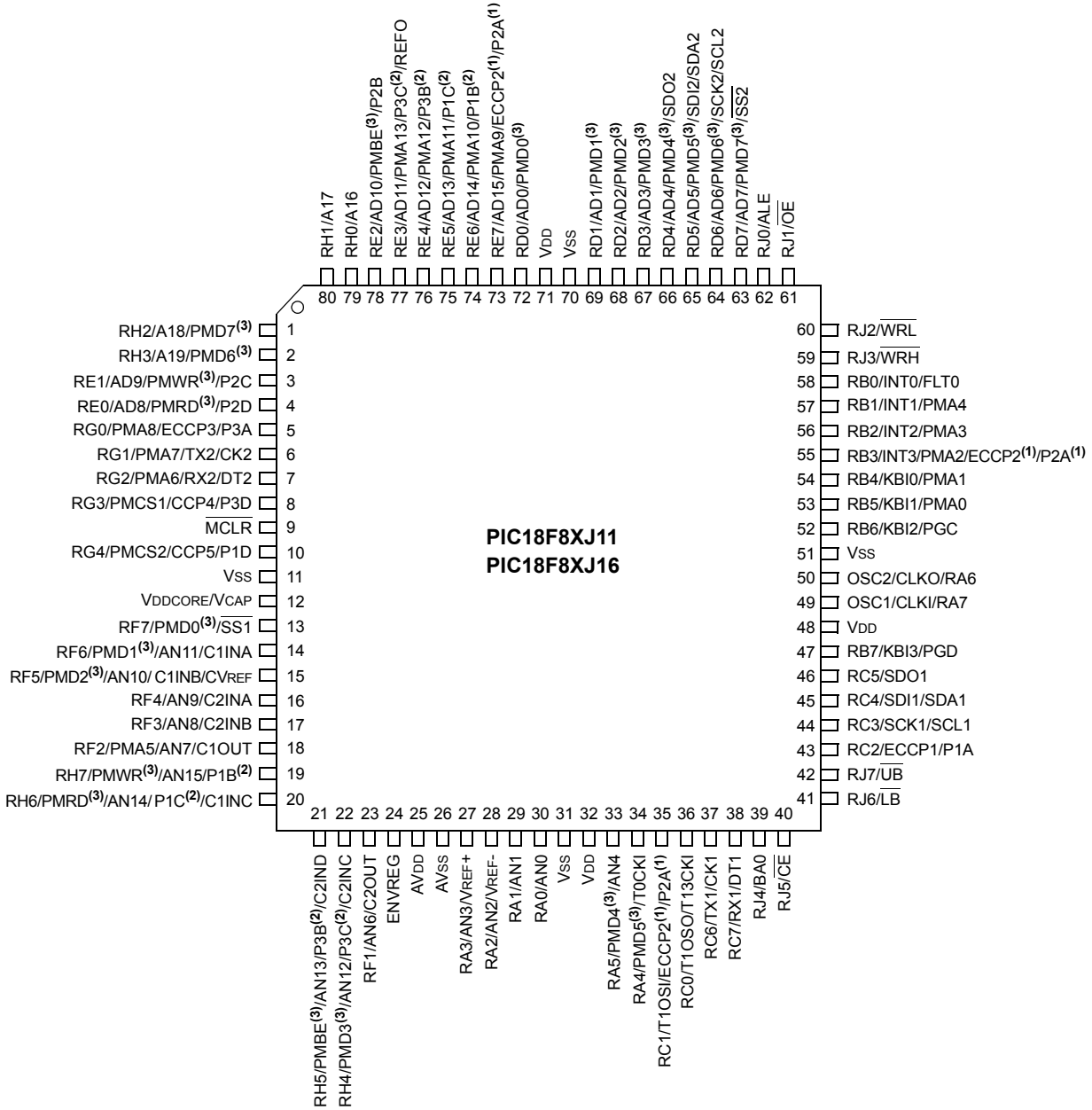


注 1： ECCP2/P2A 引脚布局取决于 CCP2MX 配置位的设置。

© 2007 Microchip Technology Inc.

引脚图 （续）

80 引脚 TQFP



注 **1：** ECCP2/P2A 引脚布局取决于 CCP2MX 配置位和处理器模式的设置。
2： P1B、 P1C、 P3B 和 P3C 引脚布局取决于 ECCPMX 配置位的设置。
3： PMP 引脚布局取决于 PMPMX 配置位的设置。

目录

1.0 器件概述 7
2.0 振荡器配置 31
3.0 功耗管理模式 41
4.0 复位 49
5.0 存储器构成 61
6.0 闪存程序存储器 87
7.0 外部存储器总线 97
8.0 8 x8 硬件乘法器 109
9.0 中断 111
10.0 I/O 端口 127
11.0 并行主端口 157
12.0 Timer0 模块 181
13.0 Timer1 模块 185
14.0 Timer2 模块 191
15.0 Timer3 模块 193
16.0 Timer4 模块 197
17.0 捕捉 / 比较 /PWM （CCP）模块 199
18.0 增强型捕捉 / 比较 /PWM （CCP）模块 207
19.0 主同步串口 （MSSP）模块 223
20.0 增强型通用同步异步收发器 （EUSART） 269
21.0 10 位模数转换器 （A/D）模块 291
22.0 比较器模块 301
23.0 比较器参考电压模块 309
24.0 CPU 的特殊性能 313
25.0 指令集综述 329
26.0 开发支持 379
27.0 电气特性 383
28.0 DC 和 AC 特性图表 423
29.0 封装信息 425
附录 A： 版本历史 429
附录 B： 器件差异 429
Microchip 网站 431
变更通知客户服务 431
客户支持 431
读者反馈表 432
索引 433
产品标识体系 445

 © 2007 Microchip Technology Inc.

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。

注：

 © 2007 Microchip Technology Inc.

1.0 器件概述

本文档包含以下器件的信息：

- PIC18F66J11
- PIC18F66J16
- PIC18F67J11
- PIC18F86J11
- PIC18F86J16
- PIC18F87J11

该系列是低压通用单片机，继承了所有 PIC18 单片机的主要传统优点，即出色的计算性能和丰富的外设集，并极具价格竞争力。这些特点使 PIC18F87J11 系列成为需要扩展外设功能集且成本是主要考虑因素的高性能应用的理想选择。

1.1 内核功能

1.1.1 纳瓦技术

PIC18F87J11 系列的所有器件都具有一系列能在工作时显著降低功耗的功能。主要包括以下几项：

- **备用运行模式：**通过将 Timer1 或内部 RC 振荡器作为控制器时钟源，可使代码执行时的功耗最大降低 90%。
- **多种空闲模式：**控制器还可在其 CPU 内核禁止而外设仍然运行的情况下运行。处于这些状态时，功耗可进一步降低，甚至降到只有正常工作需求的 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，使用户可将节能理念融入到他们的应用软件设计中。

1.1.2 振荡器选项和特点

PIC18F87J11 系列的所有器件均提供 4 种不同的振荡器选项，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：

- 2 种晶振模式，使用晶振或陶瓷谐振器。
- 2 种外部时钟模式，提供 4 分频时钟输出选项。
- 一个内部振荡模块，它提供一个 8 MHz 的时钟源和一个 INTRC 时钟源 （振荡频率约为 31 kHz，在温度和 V_{DD} 变化时频率保持稳定），以及一个包含 6 个可供用户选择的时钟频率的频率范围（从 125 kHz 到 4 MHz），总共 8 种时钟频率可供选择。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。
- 一个锁相环 （Phase Lock Loop， PLL）倍频器，可用于所有振荡器模式，并可提供从 16 MHz 至 48 MHz 范围内的时钟速率。

内部振荡器电路提供了一个稳定的参考源，为此系列器件增加了以下功能以使器件能稳定工作：

- **故障保护时钟监视器：**该选项持续监视主时钟源，将其与内部振荡器提供的参考信号作比较。如果发生了时钟故障，控制器会切换到内部振荡器，使器件可继续低速工作或安全地关闭应用。
- **双速启动：**该功能使得内部振荡器可在上电复位或从休眠模式唤醒时用作时钟源，直到主时钟源可正常工作为止。

1.1.3 扩展的存储器

PIC18F87J11 系列为应用代码提供了从 64 KB 到 128 KB 的充足代码空间。程序存储器的闪存单元可反复擦写多达 10,000 次。如果不刷新，保守地估计数据至少能保存 20 年以上。

闪存程序存储器可读写，且在正常工作状态下，PIC18F87J11 系列也为动态应用数据提供多达 3930 字节的充足数据 RAM 空间。

1.1.4 外部存储器总线

对于需要高于 128 KB 存储器的应用，PIC18F87J11 系列的 80 引脚的成员还可使用外部存储器总线（External Memory Bus， EMB）。这使控制器内部程序计数器可寻址高达 2 MB 的存储空间，从而能达到其他 8 位器件无法实现的数据访问级。这样器件就可具备额外的存储器选项，包括：

- 使用片上和外部存储器组合，上限为 2 MB
- 使用外部闪存存储器存储用于再编程应用代码或大数据表
- 使用外部 RAM 器件存储大量变量数据

1.1.5 扩展指令集

PIC18F87J11 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。此扩展可使用一个器件配置选项使能，是专为优化重入应用代码而设计的，这些代码最初是使用高级语言 （如 C）开发的。

1.1.6 快速移植

无论存储器大小如何，所有器件均共享同一组外设，这使其可随应用代码的增加和演化而平滑地移植。

整个系列使用一致的引脚排列方案还有助于向下一代更大的器件移植。在 64 引脚成员之间和 80 引脚成员之间移植，甚至是从64引脚成员向80引脚的移植都是如此。

PIC18F87J11 系列的引脚同其他 PIC18 系列器件（如 PIC18F87J10、PIC18F85J11、PIC18F8720 和 PIC18F8722）的引脚兼容。这为应用的演化提供了更多选择余地，使开发者能在保留相同功能集的同时，在 Microchip PIC18 系列中选择各种价位的器件。

1.2 其他特殊功能

- **通信：**PIC18F87J11 系列包含了一系列并行通信外设。这些器件都包括 2 个独立的增强型 USART 和 2 个主控 SSP 模块，具备 SPI 和 I^2C™（主控和从动）两种工作模式。此外，器件还有一个并行端口，且可配置为并行主端口或并行从端口。
- **CCP 模块：**本系列中所有的器件均包含 2 个捕捉 / 比较 /PWM （CCP）模块和 3 个增强型 CCP 模块，以在控制应用时得到最大的灵活性。用户可采用多达 4 个不同的时基以便能同时执行不同的操作。每个 ECCP 模块可提供多达 4 个 PWM 输出，共 12 个 PWM 输出。 ECCP 还提供许多有用的功能，包括极性选择、可编程死区时间、自动关闭和自动重启以及半桥和全桥输出模式。
- **10 位 A/D 转换器：**该模块包含了可编程采集时间，可以在不必等待一个采样周期时就可选择一个通道并启动一个转换，从而减少了代码开销。
- **扩展的看门狗定时器（Watchdog Timer，WDT）：**该增强的看门狗定时器添加了 16 位预分频器，允许扩展的超时范围，在整个工作电压和温度内延时范围保持稳定。请参见**第 27.0 节 “电气规范”**了解超时周期。

1.3 系列中各产品的具体信息

PIC18F87J11 系列器件具有 64 引脚和 80 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。这两类器件在以下三个方面存在差异：

1. 闪存程序存储器（3 种规格，范围从 PIC18FX6J11 器件的 64 KB 到 PIC18FX7J11 器件的 128 KB）。
2. I/O 端口（64 引脚器件上有 7 个双向端口，80 引脚器件上有 9 个双向端口）。
3. A/D 输入通道（64 引脚器件有 11 个，80 引脚器件有 15 个）。

该系列器件的其他功能都是相同的。表 1-1 和表 1-2 概述了这些功能。

表 1-3 和表 1-4 列举了所有器件的引脚排列。

© 2007 Microchip Technology Inc.

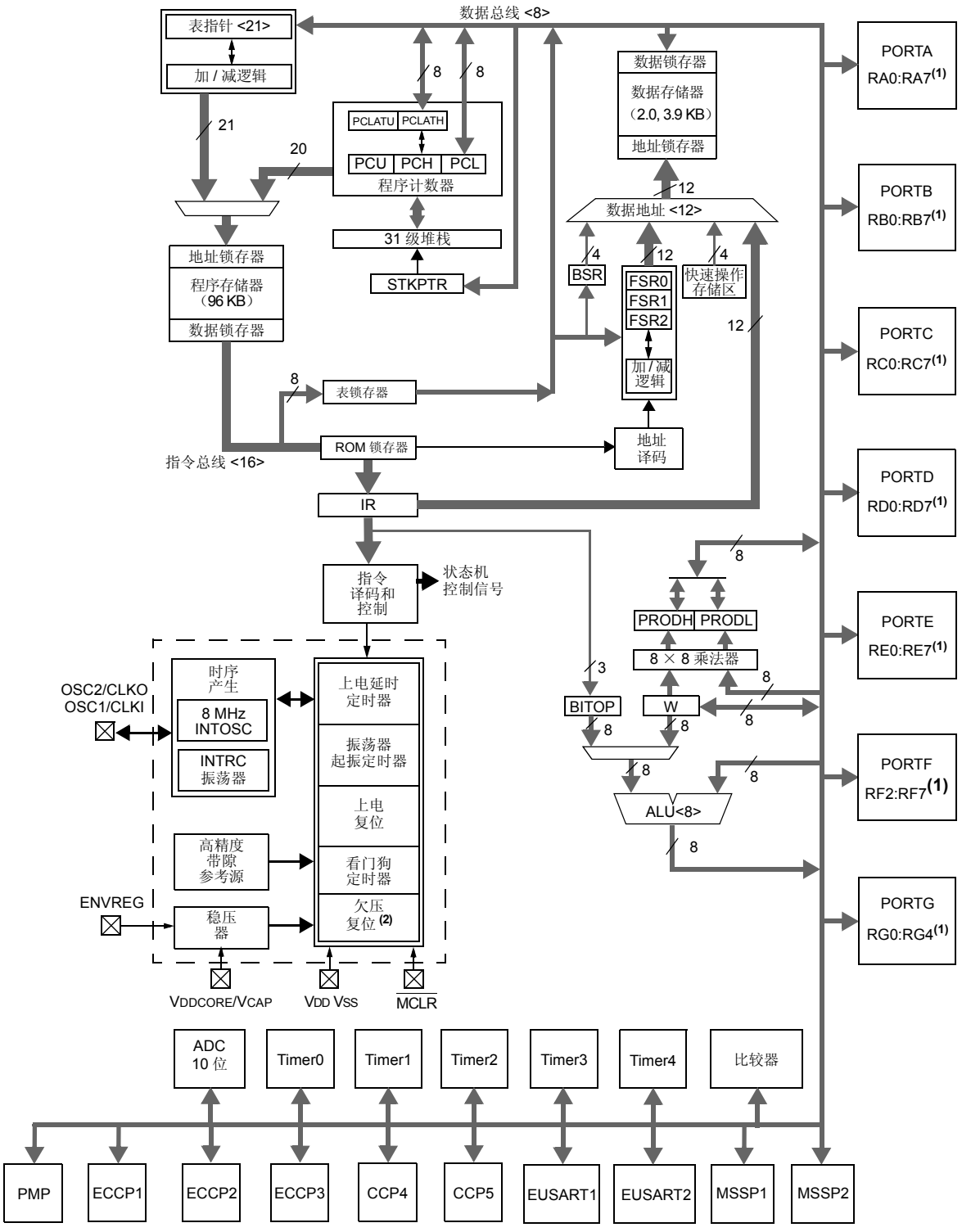
表 1-1： PIC18F6XJ1X 系列 （64 引脚器件）器件特性

特性	PIC18F66J11	PIC18F66J16	PIC18F67J11
工作频率	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
程序存储器 （字节）	64K	96K	128K
程序存储器 （指令）	32768	49152	65536
数据存储器 （字节）	3930	3930	3930
中断源	29		
I/O 端口	端口 A、 B、 C、 D、 E、 F 和 G		
定时器	5		
捕捉 / 比较 /PWM 模块	2		
增强型捕捉 / 比较 /PWM 模块	3		
串行通信	MSSP （2），增强型 USART （2）		
并行通信 （PMP）	有		
10 位模数转换模块	11 个输入通道		
复位 （和延时）	POR、 BOR、 RESET 指令，堆栈满、堆栈下溢、 $\overline{\text{MCLR}}$ 和 WDT （PWRT、 OST）		
指令集	75 条指令，启用扩展指令集后总共为 83 条		
封装	64 引脚 TQFP		

表 1-2： PIC18F8XJ1X 系列 （80 引脚器件）器件特性

特性	PIC18F86J11	PIC18F86J16	PIC18F87J11
工作频率	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
程序存储器 （字节）	64K	96K	128K
程序存储器 （指令）	32768	49152	65536
数据存储器 （字节）	3930	3930	3930
中断源	29		
I/O 端口	端口 A、 B、 C、 D、 E、 F、 G、 H 和 J		
定时器	5		
捕捉 / 比较 /PWM 模块	2		
增强型捕捉 / 比较 /PWM 模块	3		
串行通信	MSSP （2），增强型 USART （2）		
并行通信 （PMP）	有		
10 位模数转换模块	15 个输入通道		
复位 （和延时）	POR、 BOR、 RESET 指令，堆栈满、堆栈下溢、 $\overline{\text{MCLR}}$ 和 WDT （PWRT、 OST）		
指令集	75 条指令，启用扩展指令集后总共为 83 条		
封装	80 引脚 TQFP		

图 1-1：　　PIC18F6XJ1X （64 引脚）框图

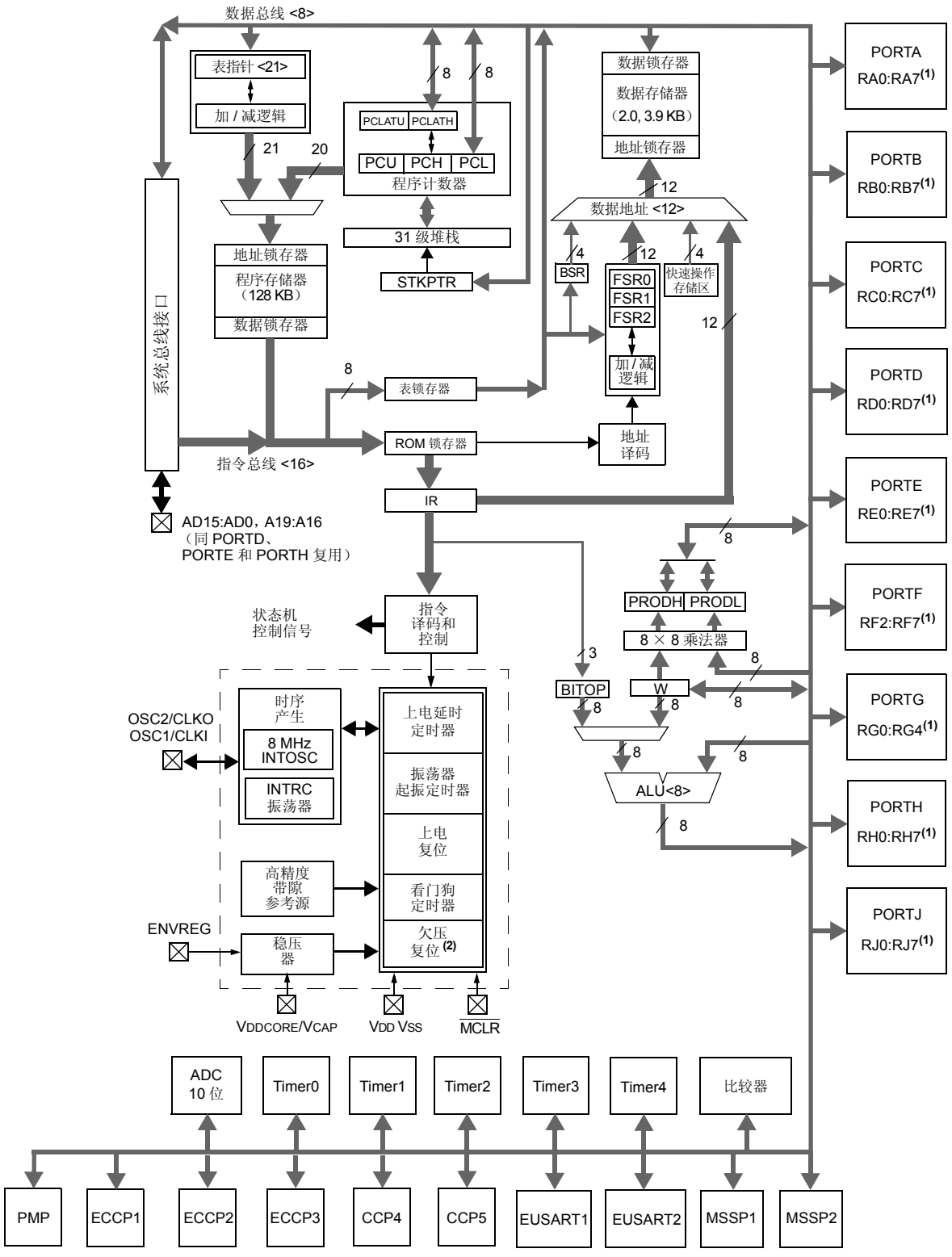


注　**1：** 请参见表 1-3 了解 I/O 端口引脚说明。

2： 当使能主板稳压器时提供 BOR 功能。

© 2007 Microchip Technology Inc.

图 1-2：　　PIC18F8XJ1X （80 引脚）框图



注　　1： 请参见表 1-4 了解 I/O 端口引脚说明。

2： 当使能主板稳压器时提供 BOR 功能。

表 1-3： PIC18F6XJ1X I/O 引脚排列说明

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
$\overline{MCLR}$	7	I	ST	主清零 （复位）输入。此引脚为低电平时，器件复位。
OSC1/CLKI/RA7	39			振荡器晶振或外部时钟输入。仅提供外部振荡器模式（EC/ECPLL 和 HS/HSPLL）。
OSC1		I	ST	主振荡器输入连接。 振荡器晶振输入或外部时钟源输入。 在 RC 模式配置时带 ST 缓冲器； 否则为 CMOS 缓冲器。
CLKI		I	CMOS	主时钟输入连接。 外部时钟源输入。总是与 OSC1 引脚功能复用 （见相关 OSC1/CLKI、 OSC2/CLKO 引脚信息）。
RA7		I/O	TTL	通用 I/O 引脚。仅提供 INTIO2 和 INTPLL2 振荡器模式。
OSC2/CLKO/RA6	40			振荡器晶振或时钟输出。仅提供外部振荡器模式（EC/ECPLL 和 HS/HSPLL）。
OSC2		O	-	主振荡器反馈输出连接。 振荡器晶振输出。在晶振模式下，连接晶振或谐振器。
CLKO		O	-	系统周期时钟输出 （FOSC/4）。 在 EC、 ECPLL、 INTIO1 和 INTPLL1 振荡器模式下，OSC2 引脚输出 CLKO，其频率为 OSC1 的四分频，该频率等于指令周期的倒数。
RA6		I/O	TTL	通用 I/O 引脚。仅提供 INTIO1 和 INTPLL1 振荡器模式。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

© 2007 Microchip Technology Inc.

表 1-3： PIC18F6XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
				PORTA 是双向 I/O 端口。
RA0/AN0	24			
RA0		I/O	TTL	数字 I/O。
AN0		I	模拟	模拟输入 0。
RA1/AN1	23			
RA1		I/O	TTL	数字 I/O。
AN1		I	模拟	模拟输入 1。
RA2/AN2/VREF-	22			
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 （低电平端）输入。
RA3/AN3/VREF+	21			
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 （高电平端）输入。
RA4/T0CKI	28			
RA4		I/O	ST	数字 I/O。
T0CKI		I	ST	Timer0 外部时钟源输入。
RA5/AN4	27			
RA5		I/O	TTL	数字 I/O。
AN4		I	模拟	模拟输入 4。
RA6	-	-	-	请参见 OSC2/CLKO/RA6 引脚
RA7	-	-	-	请参见 OSC1/CLKI/RA7 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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

表 1-3： PIC18F6XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
				PORTB 是双向 I/O 端口。 PORTB 在所有的输入端都可以软件编程为内部弱上拉。
RB0/FLT0/INT0	48			
RB0		I/O	TTL	数字 I/O。
FLT0		I	ST	ECCP1/2/3 故障输入。
INT0		I	ST	外部中断 0。
RB1/INT1/PMA4	47			
RB1		I/O	TTL	数字 I/O。
INT1		I	ST	外部中断 1。
PMA4		O	-	并行主端口地址。
RB2/INT2/PMA3	46			
RB2		I/O	TTL	数字 I/O。
INT2		I	ST	外部中断 2。
PMA3		O	-	并行主端口地址。
RB3/INT3/PMA2	45			
RB3		I/O	TTL	数字 I/O。
INT3		I	ST	外部中断 3。
PMA2		O	-	并行主端口地址。
RB4/KBI0/PMA1	44			
RB4		I/O	TTL	数字 I/O。
KBI0		I	TTL	电平变化中断引脚。
PMA1		I/O	-	并行主端口地址。
RB5/KBI1/PMA0	43			
RB5		I/O	TTL	数字 I/O。
KBI1		I	TTL	电平变化中断引脚。
PMA0		I/O	-	并行主端口地址。
RB6/KBI2/PGC	42			
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 ICS™ 编程时钟引脚。
RB7/KBI3/PGD	37			
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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

© 2007 Microchip Technology Inc.

表 1-3： PIC18F6XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
				PORTC 是双向 I/O 端口。
RC0/T1OSO/T13CKI	30			
RC0		I/O	ST	数字 I/O。
T1OSO		O	-	Timer1 振荡器输出。
T13CKI		I	ST	Timer1/Timer3 外部时钟源输入。
RC1/T1OSI/ECCP2/P2A	29			
RC1		I/O	ST	数字 I/O。
T1OSI		I	CMOS	Timer1 振荡器输入。
ECCP2(1)		I/O	ST	Capture2 输入 /Compare2 输出 /PWM2 输出。
P2A(1)		O	-	ECCP2 PWM 输出 A。
RC2/ECCP1/P1A	33			
RC2		I/O	ST	数字 I/O。
ECCP1		I/O	ST	Capture1 输入 /Compare1 输出 /PWM1 输出。
P1A		O	-	ECCP1 PWM 输出 A。
RC3/SCK1/SCL1	34			
RC3		I/O	ST	数字 I/O。
SCK1		I/O	ST	SPI 模式的同步串行时钟输入 / 输出。
SCL1		I/O	ST	I^2C™ 模式的同步串行时钟输入 / 输出。
RC4/SDI1/SDA1	35			
RC4		I/O	ST	数字 I/O。
SDI1		I	ST	SPI 数据输入。
SDA1		I/O	ST	I^2C 数据 I/O。
RC5/SDO1	36			
RC5		I/O	ST	数字 I/O。
SDO1		O	-	SPI 数据输出。
RC6/TX1/CK1	31			
RC6		I/O	ST	数字 I/O。
TX1		O	-	EUSART1 异步发送。
CK1		I/O	ST	EUSART1 同步时钟 （见相关 RX1/DT1 引脚信息）。
RC7/RX1/DT1	32			
RC7		I/O	ST	数字 I/O。
RX1		I	ST	EUSART1 异步接收。
DT1		I/O	ST	EUSART1 同步数据 （见相关 TX1/CK1 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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

表 1-3： PIC18F6XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
				PORTD 是双向 I/O 端口。
RD0/PMD0	58			
RD0		I/O	ST	数字 I/O。
PMD0		I/O	TTL	并行主端口数据。
RD1/PMD1	55			
RD1		I/O	ST	数字 I/O。
PMD1		I/O	TTL	并行主端口数据。
RD2/PMD2	54			
RD2		I/O	ST	数字 I/O。
PMD2		I/O	TTL	并行主端口数据。
RD3/PMD3	53			
RD3		I/O	ST	数字 I/O。
PMD3		I/O	TTL	并行主端口数据。
RD4/PMD4/SDO2	52			
RD4		I/O	ST	数字 I/O。
PMD4		I/O	TTL	并行主端口数据。
SDO2		O	-	SPI 数据输出。
RD5/PMD5/SDI2/SDA2	51			
RD5		I/O	ST	数字 I/O。
PMD5		I/O	TTL	并行主端口数据。
SDI2		I	ST	SPI 数据输入。
SDA2		I/O	ST	I^2C™ 数据 I/O。
RD6/PMD6/SCK2/SCL2	50			
RD6		I/O	ST	数字 I/O。
PMD6		I/O	TTL	并行主端口数据。
SCK2		I/O	ST	SPI 模式的同步串行时钟输入 / 输出。
SCL2		I/O	ST	I^2C 模式的同步串行时钟输入 / 输出。
RD7/PMD7/$\overline{SS2}$	49			
RD7		I/O	ST	数字 I/O。
PMD7		I/O	TTL	并行主端口数据。
$\overline{SS2}$		I	TTL	SPI 从动选择输入。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

© 2007 Microchip Technology Inc.

表 1-3： PIC18F6XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
				PORTE 是双向 I/O 端口。
RE0/PMRD/P2D	2			
RE0		I/O	ST	数字 I/O。
PMRD		I/O	-	并行主端口读选通。
P2D		O	-	ECCP2 PWM 输出 D。
RE1/PMWR/P2C	1			
RE1		I/O	ST	数字 I/O。
PMWR		I/O	-	并行主端口写选通。
P2C		O	-	ECCP2 PWM 输出 C。
RE2/PMBE/P2B	64			
RE2		I/O	ST	数字 I/O。
PMBE		O	-	并行主端口字节使能
P2B		O	-	ECCP2 PWM 输出 B。
RE3/PMA13/P3C/REFO	63			
RE3		I/O	ST	数字 I/O。
PMA13		O	-	并行主端口地址。
P3C		O	-	ECCP3 PWM 输出 C。
REFO		O	-	参考时钟输出。
RE4/PMA12/P3B	62			
RE4		I/O	ST	数字 I/O。
PMA12		O	-	并行主端口地址。
P3B		O	-	ECCP3 PWM 输出 B。
RE5/PMA11/P1C	61			
RE5		I/O	ST	数字 I/O。
PMA11		O	-	并行主端口地址。
P1C		O	-	ECCP1 PWM 输出 C。
RE6/PMA10/P1B	60			
RE6		I/O	ST	数字 I/O。
PMA10		O	-	并行主端口地址。
P1B		O	-	ECCP1 PWM 输出 B。
RE7/PMA9/ECCP2/P2A	59			
RE7		I/O	ST	数字 I/O。
PMA9		O	-	并行主端口地址。
ECCP2(2)		I/O	ST	Capture2 输入 /Compare2 输出 /PWM2 输出。
P2A(2)		O	-	ECCP2 PWM 输出 A。

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 模拟 = 模拟输入
I = 输入 O = 输出
P = 功耗 OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

表 1-3： PIC18F6XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
				PORTF 是双向 I/O 端口。
RF1/AN6/C2OUT	17			
RF1		I/O	ST	数字 I/O。
AN6		I	模拟	模拟输入 6。
C2OUT		O	-	Comparator2 输出。
RF2/PMA5/AN7/C1OUT	16			
RF2		I/O	ST	数字 I/O。
PMA5		O	-	并行主端口地址。
AN7		I	模拟	模拟输入 7。
C1OUT		O	-	Comparator1 输出。
RF3/AN8/C2INB	15			
RF3		I	ST	数字输入。
AN8		I	模拟	模拟输入 8。
C2INB		I	模拟	Comparator2 输入 B。
RF4/AN9/C2INA	14			
RF4		I	ST	数字输入。
AN9		I	模拟	模拟输入 8。
C2INA		I	模拟	Comparator2 输入 A。
RF5/AN10/C1INB/CVREF	13			
RF5		I	ST	数字输入。
AN10		I	模拟	模拟输入 10。
C1INB		I	模拟	Comparator1 输入 B。
CVREF		O	模拟	比较器参考电压输出。
RF6/AN11/C1INA	12			
RF6		I/O	ST	数字 I/O。
AN11		I	模拟	模拟输入 11。
C1INA		I	模拟	Comparator1 输入 A。
RF7/$\overline{SS1}$	11			
RF7		I/O	ST	数字 I/O。
$\overline{SS1}$		I	TTL	SPI 从动选择输入。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

© 2007 Microchip Technology Inc.

表 1-3： PIC18F6XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 64-TQFP	引脚类型	缓冲器类型	说明
				PORTG 是双向 I/O 端口。
RG0/PMA8/ECCP3/P3A	3			
RG0		I/O	ST	数字 I/O。
PMA8		O	-	并行主端口地址。
ECCP3		I/O	ST	Capture3 输入 /Compare3 输出 /PWM3 输出。
P3A		O	-	ECCP3 PWM 输出 A。
RG1/PMA7/TX2/CK2	4			
RG1		I/O	ST	数字 I/O。
PMA7		O	-	并行主端口地址。
TX2		O	-	EUSART2 异步发送。
CK2		I/O	ST	EUSART2 同步时钟 （见相关 RX2/DT2 引脚信息）。
RG2/PMA6/RX2/DT2	5			
RG2		I/O	ST	数字 I/O。
PMA6		O	-	并行主端口地址。
RX2		I	ST	EUSART2 异步接收。
DT2		I/O	ST	EUSART2 同步数据 （见相关 TX2/CK2 引脚信息）。
RG3/PMCS1/CCP4/P3D	6			
RG3		I/O	ST	数字 I/O。
PMCS1		O	-	并行主端口芯片选择 1。
CCP4		I/O	ST	Capture4 输入 /Compare4 输出 /PWM4 输出。
P3D		O	-	ECCP3 PWM 输出 D。
RG4/PMCS2/CCP5/P1D	8			
RG4		I/O	ST	数字 I/O。
PMCS2		O	-	并行主端口芯片选择 2。
CCP5		I/O	ST	Capture5 输入 /Compare5 输出 /PWM5 输出。
P1D		O	-	ECCP1 PWM 输出 D。
VSS	9, 25, 41, 56	P	-	逻辑电路和 I/O 引脚的参考地。
VDD	26, 38, 57	P	-	逻辑电路和 I/O 引脚的正电源。
AVss	20	P	-	模拟模块的参考地。
AVDD	19	P	-	模拟模块的正电源。
ENVREG	18	I	ST	使能片内稳压器。
VDDCORE/VCAP	10			内核逻辑功耗部件或外部滤波器电容连接。
VDDCORE		P	-	单片机内核逻辑电路正向电源 （禁止稳压器）。 外部过滤器电容连接 （使能稳压器）。
VCAP		P	-	

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位置 1 时对 ECCP2/P2A 进行默认分配。
2： 当 CCP2MX 配置位清零时对 ECCP2/P2A 进行备用分配。

表 1-4： PIC18F8XJ1X I/O 引脚排列说明

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	80-TQFP			
$\overline{\text{MCLR}}$	9	I	ST	主清零（复位）输入。此引脚为低电平时，器件复位。
OSC1/CLKI/RA7	49			振荡器晶振或外部时钟输入。仅提供外部振荡器模式（EC/ECPLL 和 HS/HSPLL）。
OSC1		I	ST	主振荡器输入连接。 振荡器晶振输入或外部时钟源输入。
CLKI		I	CMOS	在 RC 模式配置时带 ST 缓冲器；否则为 CMOS 缓冲器。 主时钟输入连接。 外部时钟源输入。总是与 OSC1 引脚功能复用。（见相关的 OSC1/CLKI、OSC2/CLKO 引脚信息）。
RA7		I/O	TTL	通用 I/O 引脚。仅提供 INTIO2 和 INTPLL2 振荡器模式。
OSC2/CLKO/RA6	50			振荡器晶振或时钟输出。仅提供外部振荡器模式（EC/ECPLL 和 HS/HSPLL）。
OSC2		O	-	主振荡器反馈输出连接。 振荡器晶振输出。在晶振模式下，连接晶振或谐振器。
CLKO		O	-	系统周期时钟输出（FOSC/4）。 在 EC、ECPLL、INTIO1 和 INTPLL1 振荡器模式下，OSC2 引脚输出 CLKO，其频率为OSC1 的四分频，其频率等于指令周期的倒数。
RA6		I/O	TTL	通用 I/O 引脚。仅提供 INTIO 和 INTPLL 振荡器模式。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路（无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配（扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配（CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配（ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配（单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配（ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零（通过编程）时，对 PMP 数据及控制引脚进行备用分配。

© 2007 Microchip Technology Inc.

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTA 是双向 I/O 端口。
RA0/AN0	30			
RA0		I/O	TTL	数字 I/O。
AN0		I	模拟	模拟输入 0。
RA1/AN1	29			
RA1		I/O	TTL	数字 I/O。
AN1		I	模拟	模拟输入 1。
RA2/AN2/VREF-	28			
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 （低电平端）输入。
RA3/AN3/VREF+	27			
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 （高电平端）输入。
RA4/PMD5/T0CKI	34			
RA4		I/O	ST	数字 I/O。
PMD5(7)		I/O	TTL	并行主端口数据。
T0CKI		I	ST	Timer0 外部时钟源输入。
RA5/PMD4/AN4	33			
RA5		I/O	TTL	数字 I/O。
PMD4(7)		I/O	TTL	并行主端口数据。
AN4		I	模拟	模拟输入 4。
RA6	-	-	-	请参见 OSC2/CLKO/RA6 引脚
RA7	-	-	-	请参见 OSC1/CLKI/RA7 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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTB 是双向 I/O 端口。 PORTB 在所有的输入端都可以软件编程为内部弱上拉。
RB0/FLT0/INT0	58			
RB0		I/O	TTL	数字 I/O。
FLT0		I	ST	ECCP1/2/3 故障输入。
INT0		I	ST	外部中断 0。
RB1/INT1/PMA4	57			
RB1		I/O	TTL	数字 I/O。
INT1		I	ST	外部中断 1。
PMA4		O	-	并行主端口地址。
RB2/INT2/PMA3	56			
RB2		I/O	TTL	数字 I/O。
INT2		I	ST	外部中断 2。
PMA3		O	-	并行主端口地址。
RB3/INT3/PMA2/ ECCP2/P2A	55			
RB3		I/O	TTL	数字 I/O。
INT3		I	ST	外部中断 3。
PMA2		O	-	并行主端口地址。
ECCP2(1)		I/O	ST	Capture2 输入 /Compare2 输出 /PWM2 输出。
P2A(1)		O	-	ECCP2 PWM 输出 A。
RB4/KBI0/PMA1	54			
RB4		I/O	TTL	数字 I/O。
KBI0		I	TTL	电平变化中断引脚。
PMA1		I/O	-	并行主端口地址。
RB5/KBI1/PMA0	53			
RB5		I/O	TTL	数字 I/O。
KBI1		I	TTL	电平变化中断引脚。
PMA0		I/O	-	并行主端口地址。
RB6/KBI2/PGC	52			
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	47			
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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

© 2007 Microchip Technology Inc.

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTC 是双向 I/O 端口。
RC0/T1OSO/T13CKI	36			
RC0		I/O	ST	数字 I/O。
T1OSO		O	-	Timer1 振荡器输出。
T13CKI		I	ST	Timer1/Timer3 外部时钟源输入。
RC1/T1OSI/ECCP2/P2A	35			
RC1		I/O	ST	数字 I/O。
T1OSI		I	CMOS	Timer1 振荡器输入。
ECCP2(2)		I/O	ST	Capture2 输入 /Compare2 输出 /PWM2 输出。
P2A(2)		O	-	ECCP2 PWM 输出 A。
RC2/ECCP1/P1A	43			
RC2		I/O	ST	数字 I/O。
ECCP1		I/O	ST	Capture1 输入 /Compare1 输出 /PWM1 输出。
P1A		O	-	ECCP1 PWM 输出 A。
RC3/SCK1/SCL1	44			
RC3		I/O	ST	数字 I/O。
SCK1		I/O	ST	SPI 模式的同步串行时钟输入 / 输出。
SCL1		I/O	ST	I^2C™ 模式的同步串行时钟输入 / 输出。
RC4/SDI1/SDA1	45			
RC4		I/O	ST	数字 I/O。
SDI1		I	ST	SPI 数据输入。
SDA1		I/O	ST	I^2C 数据 I/O。
RC5/SDO1	46			
RC5		I/O	ST	数字 I/O。
SDO1		O	-	SPI 数据输出。
RC6/TX1/CK1	37			
RC6		I/O	ST	数字 I/O。
TX1		O	-	EUSART1 异步发送。
CK1		I/O	ST	EUSART1 同步时钟 （见相关的 RX1/DT1 引脚信息）。
RC7/RX1/DT1	38			
RC7		I/O	ST	数字 I/O。
RX1		I	ST	EUSART1 异步接收。
DT1		I/O	ST	EUSART1 同步数据 （见相关的 TX1/CK1 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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTD 是双向 I/O 端口。
RD0/AD0/PMD0	72			
RD0		I/O	ST	数字 I/O。
AD0		I/O	TTL	外部存储器地址 / 数据 0。
PMD0(6)		I/O	TTL	并行主端口数据。
RD1/AD1/PMD1	69			
RD1		I/O	ST	数字 I/O。
AD1		I/O	TTL	外部存储器地址 / 数据 1。
PMD1(6)		I/O	TTL	并行主端口数据。
RD2/AD2/PMD2	68			
RD2		I/O	ST	数字 I/O。
AD2		I/O	TTL	外部存储器地址 / 数据 2。
PMD2(6)		I/O	TTL	并行主端口数据。
RD3/AD3/PMD3	67			
RD3		I/O	ST	数字 I/O。
AD3		I/O	TTL	外部存储器地址 / 数据 3。
PMD3(6)		I/O	TTL	并行主端口数据。
RD4/AD4/PMD4/SDO2	66			
RD4		I/O	ST	数字 I/O。
AD4		I/O	TTL	外部存储器地址 / 数据 4。
PMD4(6)		I/O	TTL	并行主端口数据。
SDO2		O	-	SPI 数据输出。
RD5/AD5/PMD5/ SDI2/SDA2	65			
RD5		I/O	ST	数字 I/O。
AD5		I/O	TTL	外部存储器地址 / 数据 5。
PMD5(6)		I/O	TTL	并行主端口数据。
SDI2		I	ST	SPI 数据输入。
SDA2		I/O	ST	I^2C™ 数据 I/O。
RD6/AD6/PMD6/ SCK2/SCL2	64			
RD6		I/O	ST	数字 I/O。
AD6		I/O	TTL	外部存储器地址 / 数据 6。
PMD6(6)		I/O	TTL	并行主端口数据。
SCK2		I/O	ST	SPI 模式的同步串行时钟输入 / 输出。
SCL2		I/O	ST	I^2C 模式的同步串行时钟输入 / 输出。
RD7/AD7/PMD7/$\overline{SS2}$	63			
RD7		I/O	ST	数字 I/O。
AD7		I/O	TTL	外部存储器地址 / 数据 7。
PMD7(6)		I/O	TTL	并行主端口数据。
$\overline{SS2}$		I	TTL	SPI 从动选择输入。

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　模拟 = 模拟输入
I = 输入　O = 输出
P = 功耗　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

© 2007 Microchip Technology Inc.

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTE 是双向 I/O 端口。
RE0/AD8/PMRD/P2D	4			
RE0		I/O	ST	数字 I/O。
AD8		I/O	TTL	外部存储器地址 / 数据 8。
PMRD(6)		I/O	-	并行主端口读选通。
P2D		O	-	ECCP2 PWM 输出 D。
RE1/AD9/PMWR/P2C	3			
RE1		I/O	ST	数字 I/O。
AD9		I/O	TTL	外部存储器地址 / 数据 9。
PMWR(6)		I/O	-	并行主端口写选通。
P2C		O	-	ECCP2 PWM 输出 C。
RE2/AD10/PMBE/P2B	78			
RE2		I/O	ST	数字 I/O。
AD10		I/O	TTL	外部存储器地址 / 数据 10。
PMBE(6)		O	-	并行主端口字节使能。
P2B		O	-	ECCP2 PWM 输出 B。
RE3/AD11/PMA13/P3C/REFO	77			
RE3		I/O	ST	数字 I/O。
AD11		I/O	TTL	外部存储器地址 / 数据 11。
PMA13		O	-	并行主端口地址。
P3C(3)		O	-	ECCP3 PWM 输出 C。
REFO		O	-	参考时钟输出。
RE4/AD12/PMA12/P3B	76			
RE4		I/O	ST	数字 I/O。
AD12		I/O	TTL	外部存储器地址 / 数据 12。
PMA12		O	-	并行主端口地址。
P3B(3)		O	-	ECCP3 PWM 输出 B。
RE5/AD13/PMA11/P1C	75			
RE5		I/O	ST	数字 I/O。
AD13		I/O	TTL	外部存储器地址 / 数据 13。
PMA11		O	-	并行主端口地址。
P1C(3)		O	-	ECCP1 PWM 输出 C。
RE6/AD14/PMA10/P1B	74			
RE6		I/O	ST	数字 I/O。
AD14		I/O	TTL	外部存储器地址 / 数据 14。
PMA10		O	-	并行主端口地址。
P1B(3)		O	-	ECCP1 PWM 输出 B。
RE7/AD15/PMA9/ECCP2/P2A	73			
RE7		I/O	ST	数字 I/O。
AD15		I/O	TTL	外部存储器地址 / 数据 15。
PMA9		O	-	并行主端口地址。
ECCP2(4)		I/O	ST	Capture2 输入 /Compare2 输出 /PWM2 输出。
P2A(4)		O	-	ECCP2 PWM 输出 A。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTF 是双向 I/O 端口。
RF1/AN6/C2OUT	23			
RF1		I/O	ST	数字 I/O。
AN6		I	模拟	模拟输入 6。
C2OUT		O	-	Comparator2 输出。
RF2/PMA5/AN7/C1OUT	18			
RF2		I/O	ST	数字 I/O。
PMA5		O	-	并行主端口地址。
AN7		I	模拟	模拟输入 7。
C1OUT		O	-	Comparator1 输出。
RF3/AN8/C2INB	17			
RF3		I	ST	数字输入。
AN8		I	模拟	模拟输入 8。
C2INB		I	模拟	Comparator2 输入 B。
RF4/AN9/C2INA	16			
RF4		I	ST	数字输入。
AN9		I	模拟	模拟输入 8。
C2INA		I	模拟	Comparator2 输入 A。
RF5/PMD2/AN10/ C1INB/CVREF	15			
RF5		I/O	ST	数字 I/O。
PMD2(7)		I/O	TTL	并行主端口地址。
AN10		I	模拟	模拟输入 10。
C1INB		I	模拟	Comparator1 输入 B。
CVREF		O	模拟	比较器参考电压输出。
RF6/PMD1/AN11/C1INA	14			
RF6		I/O	ST	数字 I/O。
PMD1(7)		I/O	TTL	并行主端口地址。
AN11		I	模拟	模拟输入 11。
C1INA		I	模拟	Comparator1 输入 A。
RF7/PMD0/$\overline{SS1}$	13			
RF7		I/O	ST	数字 I/O。
PMD0(7)		I/O	TTL	并行主端口地址。
$\overline{SS1}$		I	TTL	SPI 从动选择输入。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

© 2007 Microchip Technology Inc.

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTG 是双向 I/O 端口。
RG0/PMA8/ECCP3/P3A	5			
RG0		I/O	ST	数字 I/O。
PMA8		O	-	并行主端口地址。
ECCP3		I/O	ST	Capture3 输入 /Compare3 输出 /PWM3 输出。
P3A		O	-	ECCP3 PWM 输出 A。
RG1/PMA7/TX2/CK2	6			
RG1		I/O	ST	数字 I/O。
PMA7		O	-	并行主端口地址。
TX2		O	-	EUSART2 异步发送。
CK2		I/O	ST	EUSART2 同步时钟 （见相关 RX2/DT2 引脚信息）。
RG2/PMA6/RX2/DT2	7			
RG2		I/O	ST	数字 I/O。
PMA6		I/O	-	并行主端口地址。
RX2		I	ST	EUSART2 异步接收。
DT2		I/O	ST	EUSART2 同步数据 （见相关 TX2/CK2 引脚信息）。
RG3/PMCS1/CCP4/P3D	8			
RG3		I/O	ST	数字 I/O。
PMCS1		I/O	-	并行主端口芯片选择 1。
CCP4		I/O	ST	Capture4 输入 /Compare4 输出 /PWM4 输出。
P3D		O	-	ECCP3 PWM 输出 D。
RG4/PMCS2/CCP5/P1D	10			
RG4		I/O	ST	数字 I/O。
PMCS2		O	-	并行主端口芯片选择 2。
CCP5		I/O	ST	Capture5 输入 /Compare5 输出 /PWM5 输出。
P1D		O	-	ECCP1 PWM 输出 D。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTH 是双向 I/O 端口。
RH0/A16	79			
RH0		I/O	ST	数字 I/O。
A16		O	TTL	外部存储器地址 / 数据 16。
RH1/A17	80			
RH1		I/O	ST	数字 I/O。
A17		O	TTL	外部存储器地址 / 数据 17。
RH2/A18/PMD7	1			
RH2		I/O	ST	数字 I/O。
A18		O	TTL	外部存储器地址 / 数据 18。
PMD7(7)		I/O	TTL	并行主端口数据。
RH3/A19/PMD6	2			
RH3		I/O	ST	数字 I/O。
A19		O	TTL	外部存储器地址 / 数据 19。
PMD6(7)		I/O	TTL	并行主端口数据。
RH4/PMD3/AN12/ P3C/C2INC	22			
RH4		I/O	ST	数字 I/O。
PMD3(7)		I/O	TTL	并行主端口地址。
AN12		I	模拟	模拟输入 12。
P3C(5)		O	-	ECCP3 PWM 输出 C。
C2INC		I	模拟	Comparator2 输入 C。
RH5/PMBE/AN13/ P3B/C2IND	21			
RH5		I/O	ST	数字 I/O。
PMBE(7)		O	-	并行主端口字节使能。
AN13		I	模拟	模拟输入 13。
P3B(5)		O	-	ECCP3 PWM 输出 B。
C2IND		I	模拟	Comparator2 输入 D。
RH6/PMRD/AN14/ P1C/C1INC	20			
RH6		I/O	ST	数字 I/O。
PMRD(7)		I/O	-	并行主端口读选通。
AN14		I	模拟	模拟输入 14。
P1C(5)		O	-	ECCP1 PWM 输出 C。
C1INC		I	模拟	Comparator1 输入 C。
RH7/PMWR/AN15/P1B	19			
RH7		I/O	ST	数字 I/O。
PMWR(7)		I/O	-	并行主端口写选通。
AN15		I	模拟	模拟输入 15。
P1B(5)		O	-	ECCP1 PWM 输出 B。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 VDD）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

© 2007 Microchip Technology Inc.

表 1-4： PIC18F8XJ1X I/O 引脚排列说明 （续）

引脚名称	引脚号 80-TQFP	引脚类型	缓冲器类型	说明
				PORTJ 是双向 I/O 端口。
RJ0/ALE	62			
RJ0		I/O	ST	数字 I/O。
ALE		O	-	外部存储器地址锁存器使能。
RJ1/$\overline{OE}$	61			
RJ1		I/O	ST	数字 I/O。
$\overline{OE}$		O	-	外部存储器输出使能。
RJ2/$\overline{WRL}$	60			
RJ2		I/O	ST	数字 I/O。
$\overline{WRL}$		O	-	外部存储器写低控制引脚。
RJ3/$\overline{WRH}$	59			
RJ3		I/O	ST	数字 I/O。
$\overline{WRH}$		O	-	外部存储器写高控制引脚。
RJ4/BA0	39			
RJ4		I/O	ST	数字 I/O。
BA0		O	-	外部存储器字节地址 0 控制引脚。
RJ5/$\overline{CE}$	40			
RJ5		I/O	ST	数字 I/O
$\overline{CE}$		O	-	外部存储器芯片使能控制引脚。
RJ6/$\overline{LB}$	41			
RJ6		I/O	ST	数字 I/O。
$\overline{LB}$		O	-	外部存储器低字节控制引脚。
RJ7/$\overline{UB}$	42			
RJ7		I/O	ST	数字 I/O。
$\overline{UB}$		O	-	外部存储器高字节控制引脚。
V_{SS}	11, 31, 51, 70	P	-	逻辑电路和 I/O 引脚的参考地。
V_{DD}	32, 48, 71	P	-	逻辑电路和 I/O 引脚的正向电源。
AVss	26	P	-	模拟模块的参考地。
AV_{DD}	25	P	-	模拟模块的正向电源。
ENVREG	24	I	ST	使能片内稳压器。
V_{DDCORE}/V_{CAP}	12			内核逻辑功耗部件或外部滤波器电容连接。
V_{DDCORE}		P	-	单片机内核逻辑电路正向电源 （禁止稳压器）。
V_{CAP}		P	-	外部滤波器电容连接 （使能稳压器）。

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　　模拟 = 模拟输入
I = 输入　　O = 输出
P = 功耗　　OD = 漏极开路 （无 P 二极管接到 V_{DD}）

注 1： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （扩展单片机模式）。
2： 在所有工作模式下，对 ECCP2/P2A 进行默认分配 （CCP2MX 位置 1 时）。
3： 对 P1B/P1C/P3B/P3C 进行默认分配 （ECCPMX 配置位置 1）。
4： 当 CCP2MX 配置位清零时，对 ECCP2/P2A 进行备用分配 （单片机模式）。
5： 对 P1B/P1C/P3B/P3C 进行备用分配 （ECCPMX 配置位清零）。
6： 当 PMPMX 配置位置 1 时，对 PMP 数据及控制引脚进行默认分配。
7： 当 PMPMX 配置位清零 （通过编程）时，对 PMP 数据及控制引脚进行备用分配。

注：

 © 2007 Microchip Technology Inc.

2.0 振荡器配置

2.1 振荡器类型

PIC18F87J11 系列器件可在以下八种不同的振荡模式下工作：

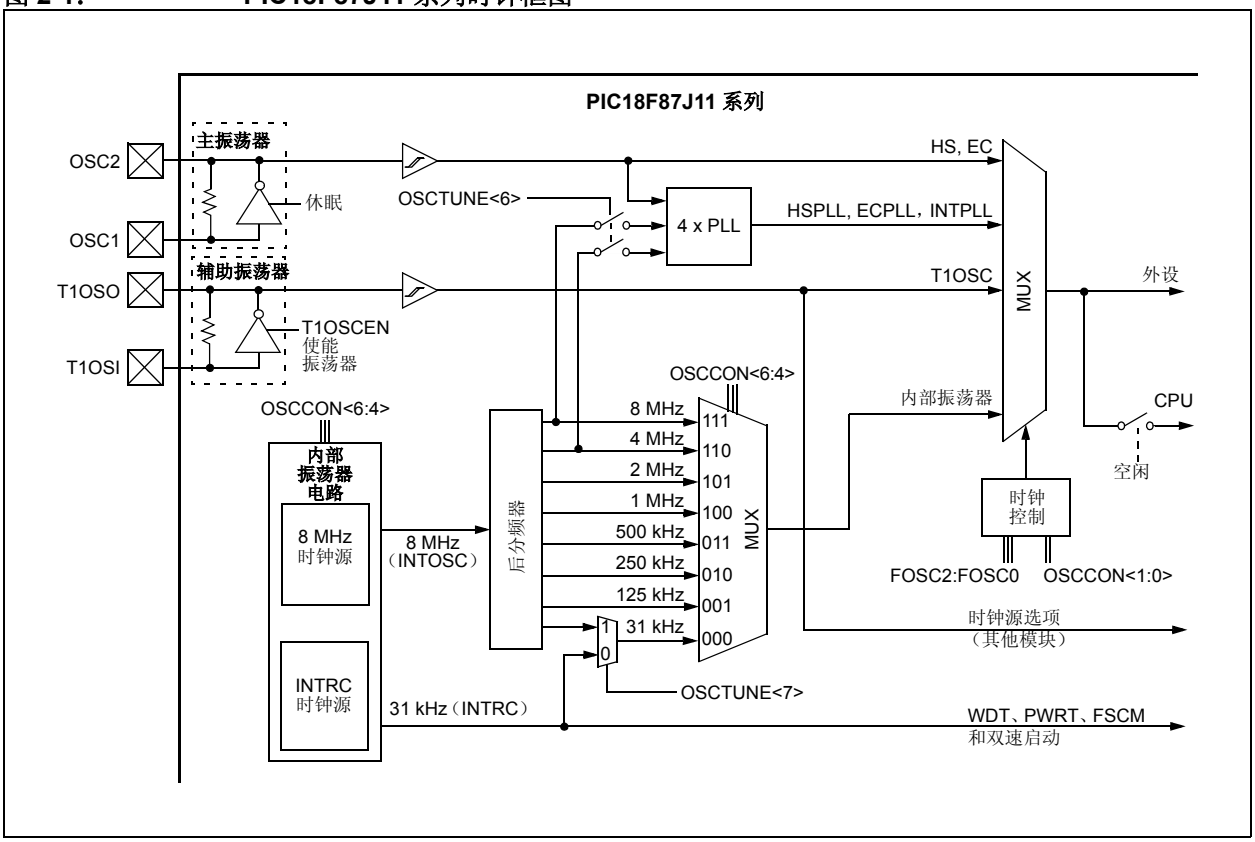
1. HS 高速晶振 / 谐振器模式
2. HSPLL 带软件 PLL 控制的高速晶振 / 谐振器模式
3. EC 带 FOSC/4 输出的外部时钟
4. ECPLL 带软件 PLL 控制的外部时钟
5. INTIO1 内部振荡器电路，通过 RA6 输出 FOSC/4 的信号，RA7 作为 I/O 引脚
6. INTIO2 内部振荡器电路，RA6 和 RA7 作为 I/O 引脚
7. INTPLL1 带软件 PLL 控制的内部振荡器电路，通过 RA6 输出 FOSC/4 的信号，RA7 作为 I/O 引脚
8. INTPLL2 带软件 PLL 控制的内部振荡器电路，RA6 和 RA7 作为 I/O 引脚

用户可以通过编程 FOSC2:FOSC0 配置位来选择所有模式。

另外，PIC18F87J11 系列器件可在软件控制下或在某些条件下自动在不同时钟源之间切换。这使用户能通过实时管理器件时钟速度而不必复位应用程序以达到节能的目的。

图 2-1 所示为 PIC18F87J11 系列器件的时钟源。

图 2-1： PIC18F87J11 系列时钟框图



2.2 控制寄存器

OSCCON 寄存器（寄存器 2-1）控制器件时钟运行的主要方面。此寄存器用于选择要使用的振荡器类型、调用的功耗管理模式以及 INTOSC 时钟源的输出频率。它还提供振荡器的状态。

OSCTUNE 寄存器（寄存器 2-2）控制内部振荡器电路的调整和运行。它还实现了 PLLEN 位，用于控制锁相环（Phase Locked Loop，PLL）的操作（见**第 2.4.3 节 “PLL 倍频器”**）。

寄存器 2-1： OSCCON：振荡器控制寄存器 (1)

R/W-0	R/W-1	R/W-1	R/W-0	R(2)	U-1	R/W-0	R/W-0
IDLEN	IRCF2(3)	IRCF1(3)	IRCF0(3)	OSTS	—	SCS1(5)	SCS0(5)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

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

bit 6-4 **IRCF2:IRCF0**：INTOSC 时钟源频率选择位 (3)
111 = 8 MHz （INTOSC 直接驱动时钟）
110 = 4 MHz （默认）
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz （来自 INTOSC/256 或 INTRC） (4)

bit 3 **OSTS**：振荡器起振定时器超时状态位 (2)
1 = 振荡器起振定时器（OST）超时；主振荡器正在运行
0 = 振荡器起振定时器（OST）正在运行；主振荡器尚未就绪

bit 2 **未实现位**：读为 1

bit 1-0 **SCS1:SCS0**：系统时钟选择位 (5)
11 = 内部振荡器电路
10 = 主振荡器
01 = Timer1 振荡器
00 = 默认主振荡器（由 FOSC2:FOSC0 配置位定义）

注 1：当 WDTCON<4> = 0 时，此处的默认（旧） SFR 可用。
2：复位状态取决于 IESO 配置位的状态。
3：如果器件时钟由内部振荡器提供，那么修改这些位将导致时钟频率立即切换。
4：由 INTSRC 位（OSCTUNE<7>）选择的时钟源，请参见正文。
5：修改这些位将导致时钟源立即切换。

© 2007 Microchip Technology Inc.

寄存器 2-2： OSCTUNE：振荡器调节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **INTSRC**：内部振荡器低频源选择位
1 = 来自 8 MHz INTOSC 源的 31.25 kHz 器件时钟 （使能 256 分频）
0 = 来自 INTRC 31 kHz 振荡器的31 kHz 器件时钟

bit 6 **PLLEN**：倍频器 PLL 使能位
1 = 使能 PLL
0 = 禁止 PLL

bit 5-0 **TUN5:TUN0**：快速 RC 振荡器 （INTOSC）频率调节位
011111 =最高频率
• •
• •
000001
000000 =中心频率。快速 RC 振荡器运行在校准后的频率上。
111111
• •
• •
100000 =最低频率

2.3 时钟源和振荡器切换

PIC18F87J11 系列器件实际上有三个独立的时钟源：

- 主振荡器
- 辅助振荡器
- 内部振荡器

主振荡器可作为器件的主要振荡器。它们是连接到 OSC1 引脚和 OSC2 引脚的任何外部振荡器，包括外部晶振和谐振器模式以及外部时钟模式。如果通过 FOSC2:FOSC0 配置位选择，内部振荡器时钟（31 kHz INTRC 源或 8 MHz INTOSC 源）均可被视为是主振荡器。特定的模式由 FOSC 配置位定义。这些模式的详细信息在**第 2.4 节 “外部振荡器模式”**中介绍。

辅助振荡器是未与 OSC1 或 OSC2 引脚连接的外部时钟源。这些时钟源即使在控制器处于功耗管理模式时仍可继续工作。PIC18F87J11 系列器件支持使用 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作为实时时钟 （RTC）等功能的时基。在**第 13.0 节 “Timer1 模块”**中将对 Timer1 振荡器作更详细的介绍。

除了在一些情况下作为主时钟源之外，**内部振荡器**还可作为功耗管理模式的时钟源。INTRC 源也可作为几种特殊功能的时钟源，例如 WDT 和故障保护时钟监视器。在**第 2.5 节 “内部振荡器电路”**中将对内部振荡器时钟作更详细的介绍。

PIC18F87J11 系列提供的功能包括允许将器件时钟源从主振荡器 （由器件的配置选定）切换到备用时钟源。当使能了备用时钟源时，可使用多种功耗管理工作模式。

2.3.1 时钟源选择

系统时钟选择位 SCS1:SCS0 （OSCCON<1:0>）用于选择时钟源。可用的时钟源包括主时钟（由 FOSC2:FOSC0 配置位定义）、辅助时钟 （Timer1 振荡器）和内部振荡器。在主要的时钟转换间隔后的一个或多个位被写入之后，时钟源发生改变。

OSTS （OSCCON<3>）和 T1RUN （T1CON<6>）位表明当前是哪个时钟源提供器件时钟。OSTS 位表明振荡器起振定时器（OST）已超时，主时钟是主时钟模式下的器件时钟源。T1RUN 位表明 Timer1 振荡器在辅助时钟模式中提供器件时钟。在功耗管理模式中的任何时间，这些位中都只有一个位会被置 1。如果这些位都没有置 1，则表示当前系统时钟源是 INTRC，或内部振荡模块刚刚起振且尚未稳定。

当执行 SLEEP 指令后，由 IDLEN 位决定器件进入休眠模式还是某个空闲模式。

第 3.0 节“功耗管理模式”将更详细地介绍在 OSCCON 寄存器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-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，则在执行 SLEEP 指令期间任何选择辅助时钟源的尝试都会被忽略。
	2：	建议在Timer1振荡器稳定工作之后再执行 SLEEP 指令，否则在 Timer1 振荡器起振时将等待一个较长的延时。

2.3.1.1 系统时钟选择与器件复位

由于 SCS 位在任何形式的复位时都被清零，这就意味着由FOSC2:FOSC0配置位定义的主振荡器在器件复位时作为主时钟源。这可以是内部振荡器本身，也可以是其他某一主时钟源（HS、EC、HSPLL、ECPLL1/2 或 INTPLL1/2）。

复位时内部振荡器电路（不带 PLL）为默认时钟的情况下，快速 RC 振荡器 （INTOSC）将作为器件时钟源。它将以 4 MHz 的频率起振；后分频器选择对应于 IRCF2:IRCF0 位复位值 （110）。

无论选择了哪个主振荡器，INTRC 总是在器件上电时被使能。它将作为时钟源，直到器件已经从存储器中装入了其配置值。此时 FOSC 配置位被读取，并选择了振荡器的工作模式。

注意主时钟源或内部振荡器在任何给定时刻都会有两种可能的 SCS1:SCS0 位设置选项。

2.3.2 振荡器转换

PIC18F87J11 系列器件包含了防止切换时钟源时发生时钟“毛刺”的电路。在时钟切换时，器件时钟会有短暂的停顿。停顿的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.

2.4 外部振荡器模式

2.4.1 晶体振荡器 / 陶瓷谐振器 （HS 模式）

在 HS 或 HSPLL 振荡器模式中，晶体振荡器或陶瓷谐振器被连接到 OSC1 和 OSC2 引脚以产生振荡。图 2-2 给出了引脚连接图。

振荡器的设计要求使用一个采用并联谐振工作方式的晶振。

注： 使用额定于串联谐振工作方式的晶体，可能会使振荡器产生的频率超出晶体制造厂商给出的参数范围。

表 2-1： 陶瓷谐振器的电容选择

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

上述电容值仅供设计参考。

要得到合适的振荡器工作状况，可能需要不同的电容值。用户应当测试振荡器在应用的预期 VDD 和温度下的性能。 请参见以下应用笔记以获取振荡器的详细信息：

- AN588， *“PIC ¨ Microcontroller Oscillator Design Guide ”*
- AN826， *“Crystal Oscillator Basics and Crystal Selection for rfPIC ¨ and PIC ¨ Devices ”*
- AN849， *“Basic PIC ¨ Oscillator Design ”*
- AN943， *“Practical PIC ¨ Oscillator Analysis and Design ”*
- AN949， *“Making Your Oscillator Work ”*

欲知更多信息，请参见表 2-2 下方的 “注”。

表 2-2： 晶体振荡器的电容选择

振荡器类型	**晶振频率**	**已测试的典型电容值：**	
		C1	**C2**
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。

要得到合适的振荡器工作状况，可能需要不同的电容值。用户应当测试振荡器在应用的预期 VDD 和温度下的性能。

请参见表 2-1 中所列的 Microchip 应用笔记以获取振荡器的详细信息。欲知更多信息，请参见本表下方的“注”。

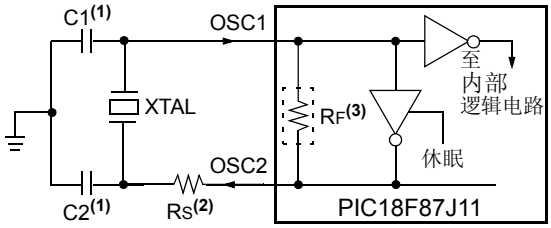
注 1： 较高的电容值可以增加振荡器的稳定性，但同时也会增长起振时间。

2： 因为每种谐振器 / 晶振都有其自身特点，用户应当向谐振器 / 晶振制造厂商询问外部元件的正确值。

3： 为避免对低驱动电平规格的晶振造成过驱动，可能会需要使用电阻 Rs。

4： 请总是验证在应用中的预期 VDD 和温度范围下的振荡器性能。

图 2-2： 晶振 / 陶瓷谐振器工作原理（HS 或 HSPLL 配置）



注 1： 如需了解 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需要一个串联电阻（RS）。

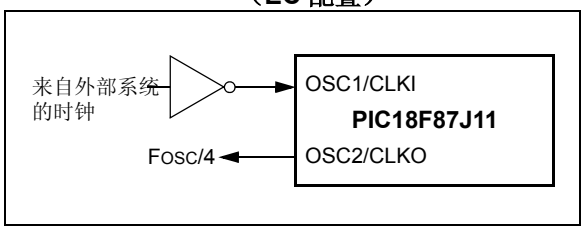
3： RF 的值随选定的振荡模式而变化。

2.4.2 外部时钟输入（EC 模式）

EC 和 ECPLL 振荡模式要求 OSC1 引脚连接一个外部时钟源。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。

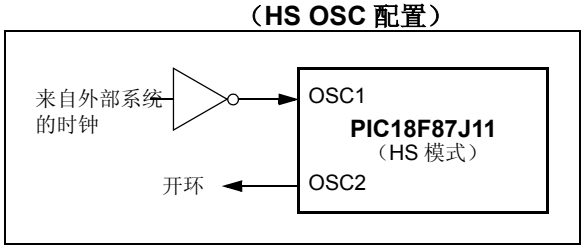
在 EC 振荡模式下，振荡器频率的 4 分频信号可由 OSC2 引脚输出。此信号可用于测试或同步其他逻辑单元。图 2-3 所示为 EC 振荡模式的引脚连接。

图 2-3： 外部时钟输入工作原理（EC 配置）



如图 2-4 所示，在 HS 模式下， OSC1 引脚也可以连接外部时钟源。在此配置中，OSC2 引脚无法得到 4 分频输出。该配置下的电流消耗在某种程度上要比在 EC 模式下的大，因为此模式下将使能内部振荡器反馈电路（在 EC 模式下，反馈电路被禁止）。

图 2-4： 外部时钟输入工作原理（HS OSC 配置）



2.4.3 PLL 倍频器

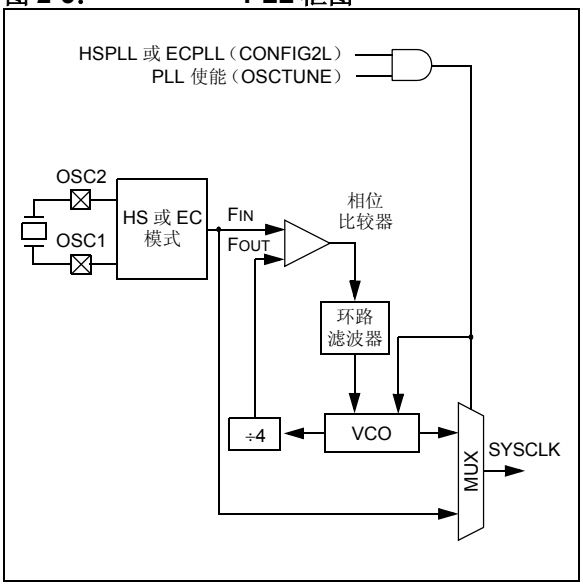
用户如果希望使用低频晶振电路或希望从晶体振荡器将器件频率调节至其最高额定频率，可以选择使用锁相环（PLL）电路。对于担心高频晶振引起 EMI 或需要来自内部振荡器更高速时钟的用户而言，这样做可能会有用。

2.4.3.1 HSPLL 和 ECPLL 模式

HSPLL 和 ESPLL 模式使器件能够有选择性地以外部振荡源的 4 倍速运行以产生最高为 48 MHz 的频率。

通过将 FOSC2:FOSC0 配置位编程为 111（对于 ECPLL）或 101（对于 HSPLL），可使能 PLL。此外，还必须将 PLLEN 位（OSCTUNE<6>）置 1。无论所选择的振荡器配置为何，清零 PLLEN 都将禁止 PLL。这使得通过软件控制应用程序时钟速度具有更大的灵活性。

图 2-5： PLL 框图



2.4.3.2 PLL 和 INTOSC

当内部振荡器电路被配置为主时钟源时，内部振荡器电路也可使用 PLL。在此配置下，PLL 用软件使能并产生最高为 32 MHz 的时钟输出。**第 2.5.2 节 “INTPLL 模式”**中描述了使用 PLL 的 INTOSC 的工作原理。

© 2007 Microchip Technology Inc.

2.5 内部振荡器电路

PIC18F87J11 系列器件包括可产生两种不同时钟信号的内部振荡器电路，两者都可以作为单片机的时钟源。这就可以避免在 OSC1 和 / 或 OSC2 引脚上使用外部振荡电路。

器件的主输出为快速 RC 振荡器，也称 INTOSC，是一个 8 MHz 时钟源，可用于直接驱动器件时钟。它还可以驱动后分频器，该分频器可提供从 31 kHz 到 4 MHz 的时钟频率。当选择 125 kHz 到 8 MHz 的时钟频率时，会使能 INTOSC。当选择 31 kHz 的时钟频率时，也可使能 INTOSC 输出，这取决于 INTSRC 位（OSCTUNE<7>）的设置。

另一个时钟源是内部 RC 振荡器 （INTRC），它提供了标称 31 kHz 的输出。如果选择 INTRC 作为器件的时钟源，它就被使能；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 24.0 节 “CPU 的特殊性能” 将对这些功能进行更详细的介绍。

通过配置 OSCCON 寄存器的 IRCF 位，可以选择时钟源频率（INTOSC 直接频率、INTOSC 后分频器频率或 INTRC 直接频率）。 单片机复位时的默认频率为 4 MHz。

2.5.1 INTIO 模式

使用内部振荡器作为时钟源可以至多省却两个外部振荡器引脚，而将其用作数字 I/O。目前有如下两种不同的配置，它们由 FOSC 配置位决定：

- 在 INTIO1 模式下， OSC2 引脚输出 FOSC/4 信号，而 OSC1 引脚则作为 RA7 （见图 2-6），用于数字输入和输出。
- 在 INTIO2 模式下，OSC1 作为 RA7，OSC2 作为 RA6 （见图 2-7），两者都用于数字输入和输出。

图 2-6： INTIO1 振荡器模式

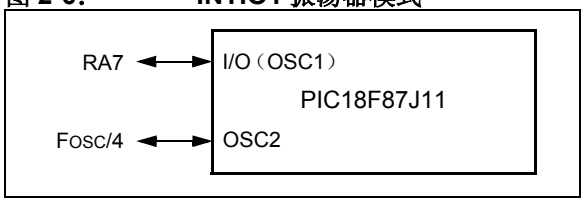
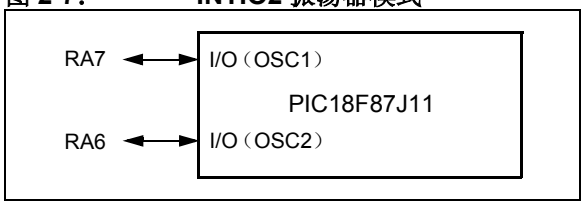


图 2-7： INTIO2 振荡器模式



2.5.2 INTPLL 模式

4 倍频锁相环可以和内部振荡器电路一起使用，以产生比一般内部振荡器时钟源更快的器件时钟速度。使能时，PLL 最高可以产生 16 MHz 或 32 MHz 的时钟速度。

PLL 的工作由软件控制。控制位 PLLEN（OSCTUNE<6>）用来使能或禁止其工作。当器件被配置为使用一种 INTPLL 模式作为主时钟源时 （FOSC2:FOSC0 = 011 或 010），仅 INTOSC 可使用 PLL。此外，仅当选定的输出频率是 4 MHz 或 8 MHz （OSCCON<6:4> = 111 或 110）时， PLL 才会工作。

与 INTIO 模式类似，有两种 INTPLL 模式可用：

- 在 INTPLL1 模式下， OSC2 引脚输出 FOSC/4 信号，而 OSC1 引脚则作为 RA7，用于数字输入和输出。从表面上看，此模式与 INTIO1 （图 2-6）相同。
- 在 INTPLL2 模式下， OSC1 作为 RA7， OSC2 作为 RA6，两者都用于数字输入和输出。从表面上看，此模式和 INTIO2 （图 2-7）相同。

2.5.3 内部振荡器输出频率和调节

内部振荡器电路出厂时进行了校准可产生 8 MHz 的 INTOSC 输出频率。在用户应用中，可以通过写 OSCTUNE 寄存器（寄存器 2-2）的 TUN5:TUN0（OSCTUNE<5:0>）对其进行调整。

当修改了 OSCTUNE 寄存器时，INTOSC 的频率将开始变为新的频率。振荡器将在 1 ms 内稳定下来。在此变动期间，代码会继续执行，且没有任何迹象表明发生了变动。

INTRC振荡器的工作独立于INTOSC源。INTOSC随着电压和温度变化而发生的变化不一定通过 INTRC 的变化反映出来，反之亦然。INTRC 的频率不受 OSCTUNE 的影响。

2.5.4 INTOSC 频率漂移

INTOSC 频率可能会随着 VDD 或温度的改变而发生漂移，这可能会以各种方式影响控制器运行。通过修改 OSCTUNE 寄存器的值可以调节 INTOSC 频率。根据器件的不同，这可能对 INTRC 时钟源频率没有影响。

调节 INTOSC 需要了解何时需要调节、应该在哪个方向进行调节，以及在某些情况下需要做多大的改变。这里描述了三种补偿技术。

2.5.4.1 用 EUSART 补偿

当 EUSART 开始产生帧错误，或者在异步模式下接收带有错误的数据时就可能需要进行调节。帧错误表示器件时钟频率过高。要对此进行调节，可以减小OSTUNE 寄存器中的值来降低时钟频率。另一方面，数据中有错误可能表示时钟速度过低。要进行补偿，可以增大 OSTUNE 寄存器中的值来提高时钟频率。

2.5.4.2 用定时器补偿

此技术可以将器件时钟的速度与某些参考时钟进行比较。可能要用两个定时器；一个为外设时钟提供时钟源，而另一个由一个固定的参考时钟（如 Timer1 振荡器）提供时钟源。

两个定时器都被清零，但由参考时钟提供时钟源的定时器产生中断。当中断发生时，时钟源在内部的定时器被读取且两个定时器都被清零。如果时钟源在内部的定时器的值比期望值大得多，则表示内部振荡器电路运行过快。要对此进行调整，请减小 OSCTUNE 寄存器中的值。

2.5.4.3 用捕捉模式下的 CCP 模块补偿

CCP 模块可以使用由内部振荡器电路提供时钟源的自由运行的 Timer1（或 Timer3）和已知周期的外部事件（例如，AC 电源频率）。第一个事件时间被捕捉到 CCPRxH:CCPRxL 寄存器中并记录下来，以备稍后使用。当第二个事件引起捕捉时，第一个事件的时间要从第二个事件的时间中减去。由于外部事件周期是已知的，即可计算出两个事件之间的时间差。

如果测量得到的时间比计算得到的时间大很多，则表示内部振荡器电路运行过快。要进行补偿，请减小 OSTUNE寄存器中的值。如果测量得到的时间比计算得到的时间小得多，则表示内部振荡器电路运行过慢。要进行补偿，请增大 OSTUNE 寄存器中的值。

2.6 参考时钟输出

除了在某一振荡器模式下的 FOSC/4 时钟输出之外，PIC18F87J11 系列中的器件时钟也能通过配置向端口引脚提供一个参考时钟输出。所有振荡器配置都提供该特性，它允许用户选择更大范围的时钟因数，以在应用中驱动外部器件。

参考时钟输出由 REFOCON 寄存器（寄存器 2-3）控制。将 ROON 位（REFOCON<7>）置 1 将使时钟信号可在 REFO（RE3）引脚上得到。RODIV3:RODIV0 位使用户能够选择 16 种不同的时钟分频值。

ROSSLP 和 ROSEL 位（REFOCON<5:4>）控制休眠模式下参考输出的可用性。ROSEL 位决定是 OSC1 和 OSC2 引脚上的振荡器还是当前系统时钟源上的振荡器用于参考时钟输出。ROSSLP 位决定器件在休眠模式下时参考时钟源是否能在 RE3 上得到。

要在休眠模式下使用参考时钟输出，ROSSLP 和 ROSEL 位都必须置 1。器件时钟必须配置为工作在 EC 或 HS 模式；否则，当器件进入休眠模式时，将关闭 OSC1 和 OSC2 上的振荡器。清零 ROSEL 位可使参考输出频率在任何时钟切换时随系统时钟的改变而改变。

REFOCON 寄存器为备用 SFR，且与 OSCCON 寄存器共享一个存储器地址。可通过置 1 REFOCON 寄存器中的 ADSHR 位（WDTCON<4>）来访问此存储器地址。

© 2007 Microchip Technology Inc.

寄存器 2-3：　　REFOCON：参考振荡器控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROON	—	ROSSLP	ROSEL(1)	RODIV3	RODIV2	RODIV1	RODIV0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　　**ROON：**参考振荡器输出使能位
1 = 参考振荡器输出在 REFO 引脚上可用
0 = 参考振荡器输出被禁止

bit 6　　**未实现位：**读为 0

bit 5　　**ROSSLP：**休眠模式下参考振荡器输出停止位
1 = 参考振荡器在休眠模式下继续运行
0 = 参考振荡器在休眠模式下被禁止

bit 4　　**ROSEL：**参考振荡器源选择位 (1)
1 = 主振荡器 （EC 或 HS）被用作基础时钟
0 = 系统时钟用作基础时钟；基础时钟反映任何器件时钟切换

bit 3-0　　**RODIV3:RODIV0：**参考振荡器除数选择位
1111 = 基础时钟值除以 32,768
1110 = 基础时钟值除以 16,384
1101 = 基础时钟值除以 8,192
1100 = 基础时钟值除以 4,096
1011 = 基础时钟值除以 2,048
1010 = 基础时钟值除以 1,024
1001 = 基础时钟值除以 512
1000 = 基础时钟值除以 256
0111 = 基础时钟值除以 128
0110 = 基础时钟值除以 64
0101 = 基础时钟值除以 32
0100 = 基础时钟值除以 16
0011 = 基础时钟值除以 8
0010 = 基础时钟值除以 4
0001 = 基础时钟值除以 2
0000 = 基础时钟值

注　1：　如果 ROSEL = 1， EC 或 HS 振荡器必须被配置为默认振荡器，其中 FOSC 配置位在休眠模式下维持时钟输出。

2.7 各种时钟源的功耗管理模式的影响

当选择了 PRI_IDLE 模式时，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。OSC1 引脚（以及由振荡器使用 OSC2 引脚）将会停止振荡。

在辅助时钟模式下（SEC_RUN和SEC_IDLE），Timer1 振荡器作为系统时钟源工作。如果需要使用 Timer1 或 Timer3 的时钟， Timer1 振荡器也可以在所有功耗管理模式下运行。

在 RC_RUN 和 RC_IDLE 模式下，由内部振荡器提供器件时钟源。可以直接用 31 kHz 的 INTRC 输出提供时钟源，或使能它来支持多种特殊功能，与功耗管理模式无关 （欲知更多有关 WDT、故障保护时钟监视器和双速启动的信息请参见**第 24.2 节“看门狗定时器（WDT）”**至**第 24.5 节 “故障保护时钟监视器”**）。

如果选择了休眠模式，所有的时钟源都会被停止。因为停止了所有晶体管切换电流，因此休眠模式下器件电流消耗最小 （仅泄漏电流）。

在休眠期间使能任何片上功能都会增加休眠时的电流消耗。需要使能 INTRC 来支持 WDT 工作。Timer1 振荡器可以用于为实时时钟（RTC）提供时钟源。不需要器件时钟源的其他功能也可工作 （例如 MSSP 从器件、PSP 和 INTx 引脚等）。**第 27.2 节“DC 特性：掉电和供电电流”**列出了会明显增加电流消耗的外设。

2.8 上电延时

有两个定时器控制上电延时，这样大部分应用都无需外接复位电路。上电延时可以确保在满足以下条件前器件保持复位状态：一般环境下器件电源供应稳定，并且主时钟已工作并稳定。如需更多有关上电延时的信息，请参见**第 4.6 节 “上电延时定时器 （PWRT）”**。

第一个定时器是上电延时定时器 （PWRT），在上电时提供固定的延时（参数 33，表 27-12）；它总是被使能的。

第二个定时器是振荡器起振定时器 （OST），用于在晶体振荡器稳定前使芯片保持在复位状态 （HS 模式）。OST 是通过允许振荡器为器件提供时钟源前等候 1024 个振荡周期来实现此延时的。

上电复位后，当控制器准备好执行指令时，有一个延时间隔 TCSD （参数 38，表 27-2）。

表 2-3： 休眠模式下 OSC1 和 OSC2 引脚状态

振荡器模式	OSC1 引脚	OSC2 引脚
EC，ECPLL	悬空，由外部时钟拉高	处于逻辑低电平 （时钟 /4 输出）
HS，HSPLL	处于静态电平时，反馈反相器被禁用	处于静态电平时，反馈反相器被禁用
INTOSC, INTPLL1/2	RA6 引脚为 I/O 功能，由 TRISA<6> 控制其方向	RA6 引脚为 I/O 功能，由 TRISA<7> 控制其方向

注： 欲知有关由于休眠和 $\overline{\text{MCLR}}$ 复位而产生的延时的信息，请参见**第 4.0 节 “复位”**。

© 2007 Microchip Technology Inc.

3.0 功耗管理模式

PIC18F87J11 系列器件提供了只需通过管理 CPU 和外设的时钟源就可以管理功耗的功能。一般而言，较低的时钟频率和由时钟源驱动的电路数目的减少会使功耗降低。为了在应用中管理功耗，提供了三种主要的工作模式：

- 运行模式
- 空闲模式
- 休眠模式

这些模式决定为器件的哪些部分提供时钟，有时还定义时钟速率。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电路）中的任意一种；而休眠模式则不使用时钟源。

功耗管理模式具有几种在以前的器件上提供的节约功耗的功能。其中一个功能是在其他 PIC18 器件上提供的时钟切换功能，允许控制器使用 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，器件的所有时钟均在此模式下停止工作。

3.1 选择功耗管理模式

选择功耗管理模式时，需要确定是否为 CPU 提供时钟以及选择何种时钟源。IDLEN 位（OSCCON<7>）控制是否为 CPU 提供时钟，而 SCS1:SCS0 位（OSCCON<1:0>）选择时钟源。表 3-1 总结了各个模式、位设置、时钟源和受影响的模块。

3.1.1 时钟源

SCS1:SCS0 位可以在功耗管理模式下选择三个时钟源中的一个。它们是：

- 主时钟，由 FOSC2:FOSC0 配置位定义。
- 辅助时钟 （Timer1 振荡器）
- 内部振荡器

3.1.2 进入功耗管理模式

从一种功耗管理模式转换到另一种功耗管理模式是通过装载 OSCCON 寄存器开始的。SCS1:SCS0 位选择时钟源并确定使用哪一种运行或空闲模式。更改这些位将导致立即切换到新的时钟源 （假定新的时钟源正在运行）。切换可能会引起时钟转换延时。**第 3.1.3 节 “时钟切换和状态指示”**及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需要设置所有的位，需要设置哪些配置位取决于当前的模式和将要切换到的模式。在执行 SLEEP 指令之前更改振荡器选择位或更改 IDLEN 位可完成多种模式之间的转换。如果已经正确地配置了 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<7,1:0>		模块时钟		可用时钟和振荡器
	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	10	提供时钟	提供时钟	主时钟——HS、 EC、 HSPLL、 ECPLL 和 INTOSC 内部振荡器电路； 这是正常的全功耗工作模式
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	11	提供时钟	提供时钟	内部振荡器模块 (2)
PRI_IDLE	1	10	关闭	提供时钟	主时钟—— HS、 EC、 HSPLL、 ECPLL 和 INTOSC
SEC_IDLE	1	01	关闭	提供时钟	辅助时钟——Timer1 振荡器
RC_IDLE	1	11	关闭	提供时钟	内部振荡器电路 (2)

注 1： 当执行 SLEEP 指令时， IDLEN 为设置值。

2： 包含 INTRC 和 INTOSC 后分频器 （内部振荡器电路）。

3.1.3 时钟切换和状态指示

在两个时钟源之间进行切换所需的时间长度是旧时钟源的两个周期与新时钟源的 3 到 4 个周期的和。此公式假设新时钟源是稳定的。

以下两个位表示当前的时钟源及其状态：OSTS（OSCCON<3>）和 T1RUN（T1CON<6>）。通常，在给定功耗管理模式下，这两个位中只有一个将被置 1。当将 OSTS 位置 1 时，主时钟提供器件时钟。当将 T1RUN 位置 1 时， Timer1 振荡器提供器件时钟。如果这两个位都不置 1， INTRC 为器件提供时钟。

注：	执行 SLEEP 指令不一定会使器件进入休眠模式。它充当触发指令，根据 IDLEN 位的设置，使控制器进入休眠模式和某种空闲模式。

3.1.4 多条 SLEEP 命令

使用 SLEEP 指令进入功耗管理模式时，其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，器件将在指令执行后进入由 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，内核和外设的时钟均处于正常工作状态。这些模式之间的差异就在于时钟源的不同。

3.2.1 PRI_RUN 模式

PRI_RUN 模式是单片机在正常情况下的全功耗执行模式。除非使能了双速启动（详细信息请参见**第 24.4 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 位可能被置 1（见**第 2.2 节“控制寄存器”**）。

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 和外设由 Timer1 振荡器提供时钟。这使用户可在使用高精度时钟源的情况下依然可获得较低的功耗。

通过将 SCS1:SCS0 位设置为 01 进入 SEC_RUN 模式。器件时钟源切换到 Timer1 振荡器（见图 3-1），关闭主振荡器， T1RUN 位（T1CON<6>）置 1 并且 OSTS 位清零。

© 2007 Microchip Technology Inc.

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 位被设置为 01 时 T1OSCEN 位未置 1，就不会进入 SEC_RUN 模式。如果 Timer1 振荡器被使能，但是未开始运行，器件时钟将被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。

从 SEC_RUN 模式转换到 PRI_RUN 模式时，在主时钟起振的同时，外设和 CPU 继续由 Timer1 振荡器提供时钟源。主时钟准备就绪后，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。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

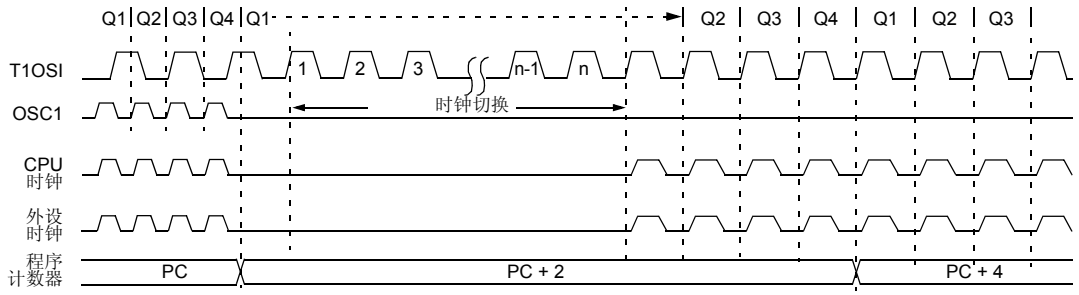
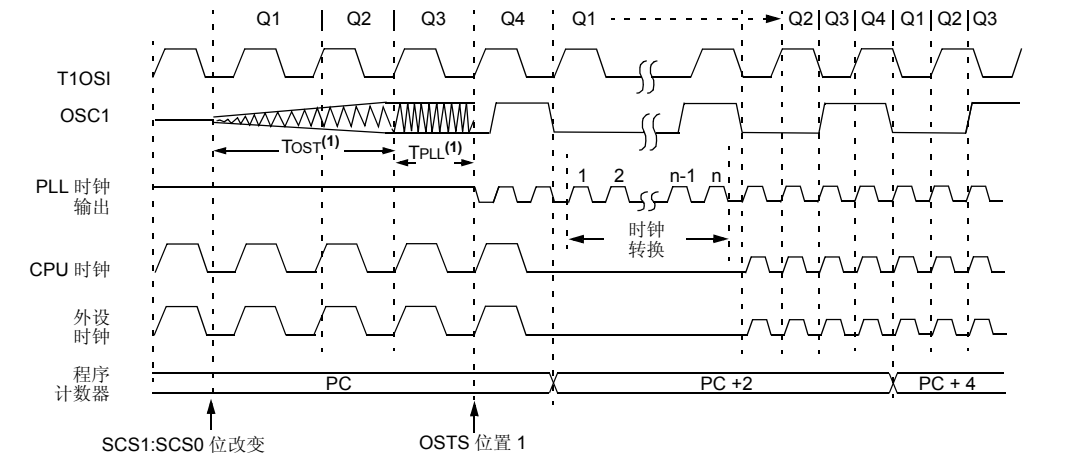


图 3-2： 从 SEC_RUN 模式到 PRI_RUN 模式 （HSPLL）的转换时序



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

3.2.3 RC_RUN 模式

RC_RUN 模式中，内部振荡器电路使用 INTOSC 复用器作为 CPU 和外设的时钟源，此时主时钟关闭。在使用 INTRC 时钟源时，此模式是所有运行模式中最节省功耗的运行模式，同时仍然在执行代码。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。

通过将 SCS<1:0> 设置为 11 可以进入此模式。当时钟源切换到内部振荡器模块（见图 3-3）时，主振荡器关闭， OSTS 位清零。

从 RC_RUN 模式转换到 PRI_RUN 模式时，在主时钟起振期间器件继续使用 INTOSC 作为时钟源。当主时钟准备就绪以后，时钟开始切换回主时钟（见图 3-4）。当时钟切换完成后， OSTS 位被置 1，主时钟提供器件时钟。切换不会影响 IDLEN 和 SCS 位。如果使能 WDT 或故障保护时钟监视器， INTRC 时钟源将继续运行。

图 3-3： 转换到 RC_RUN 模式的切换时序

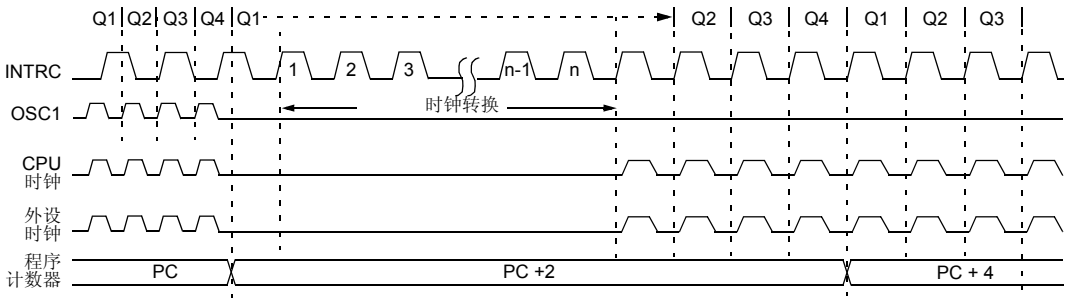
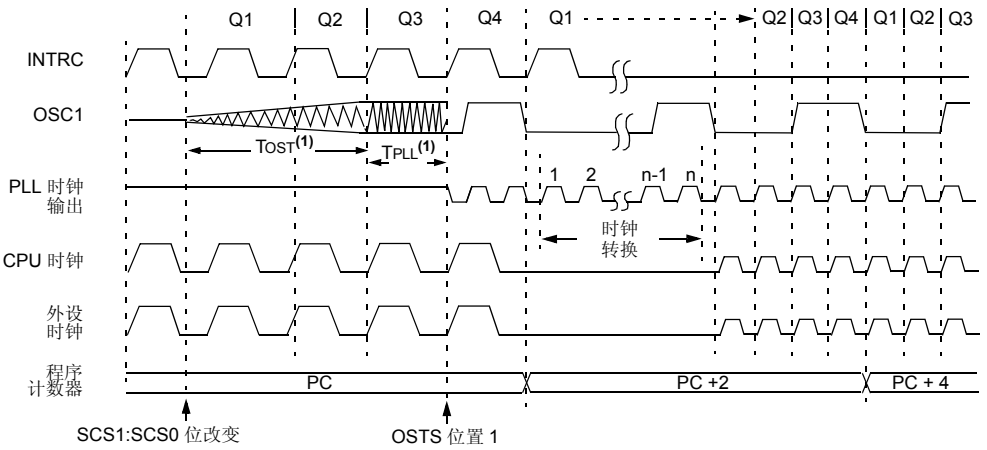


图 3-4： 从 RC_RUN 模式到 PRI_RUN 模式的转换时序



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

© 2007 Microchip Technology Inc.

3.3 休眠模式

功耗管理休眠模式与所有其他 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 指令即可进入该模式。这将关闭选定的振荡器（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无需时钟切换。这是因为单片机一旦进入休眠模式就不再需要时钟了。如果选择了 WDT，INTRC 源将继续运行。如果使能了 Timer1 振荡器， Timer1 也将继续运行。

在休眠模式中发生唤醒事件时（由于中断、复位或 WDT 超时），在由 SCS1:SCS0 位选定的时钟源准备就绪之前，器件将没有时钟源（见图 3-6），但是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电路作为时钟源（见**第 24.0 节“CPU 的特殊性能”**）。在这两种情况下，当主时钟提供器件时钟时， OSTS 位被置 1。唤醒事件不会影响 IDLEN 和 SCS 位。

3.4 空闲模式

空闲模式可使外设在继续工作时能够有选择地关闭单片机的 CPU。选择某种特定的空闲模式可使用户能进一步管理功耗。

如果在执行 SLEEP 指令时，IDLEN 位被置为 1，外设将使用由 SCS1:SCS0 位选择的时钟作为时钟源，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 时钟源将继续运行。如果使能了 Timer1 振荡器， Timer1 也将继续运行。

由于 CPU 没有执行指令，器件只能通过中断、WDT 超时或复位从任一空闲模式退出。当发生唤醒事件时，CPU 在延时一个 TCSD 间隔（表 27-12 中的参数 38）之后开始执行代码，在此延时期间 CPU 将为执行代码做好准备。当 CPU 开始执行代码时，它将沿用与当前空闲模式相同的时钟源。例如，当从 RC_IDLE 模式唤醒时，将使用内部振荡电路作为 CPU 和外设的时钟源（即 RC_RUN 模式）。唤醒事件不会影响 IDLEN 和 SCS 位。

当处于任何空闲模式或休眠模式时， WDT 超时将导致 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

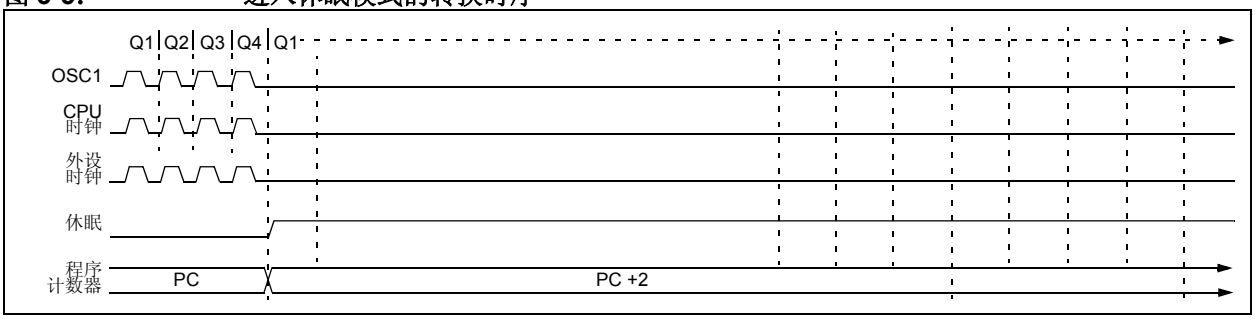
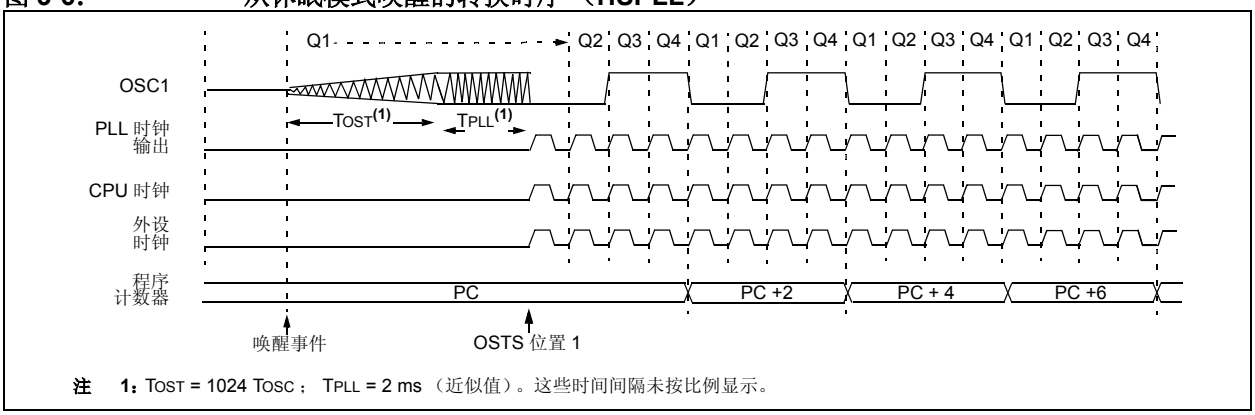


图 3-6： 从休眠模式唤醒的转换时序 （HSPLL）



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

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。对于那些对时间精度要求较高的应用来说，由于时钟源不需要“热身”或从其他振荡器转换，选用此模式可以使对时间要求较高的应用使用更精确的主时钟源以最快速度恢复器件运行。

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）。

当唤醒事件发生时，CPU 由主时钟源提供时钟。在唤醒事件和代码开始执行之间需要一段 TCSD 间隔的延时。需要这段时间以使 CPU 为执行指令做好准备。在唤醒之后，OSTS 位保持置 1。切换不会影响 IDLEN 和 SCS 位 （见图 3-8）。

3.4.2 SEC_IDLE 模式

在 SEC_IDLE 模式中， CPU 被禁止，但外设会继续将 Timer1 振荡器作为时钟源。可通过将 IDLEN 位置 1 并执行 SLEEP 指令从 SEC_RUN 进入此模式。如果器件处于其他运行模式，请先将 IDLEN 置 1，然后将 SCS1:SCS0 设置为 01 并执行 SLEEP。当时钟源切换到 Timer1 振荡器时，主振荡器关闭， OSTS 位清零，T1RUN 位置 1。

当唤醒事件发生时，外设继续将 Timer 振荡器作为时钟源。唤醒事件发生后经过一个 TCSD 的时间间隔， CPU 开始执行代码并使用 Timer1 振荡器作为其时钟源。唤醒事件不会影响 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指令并且不会进入 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

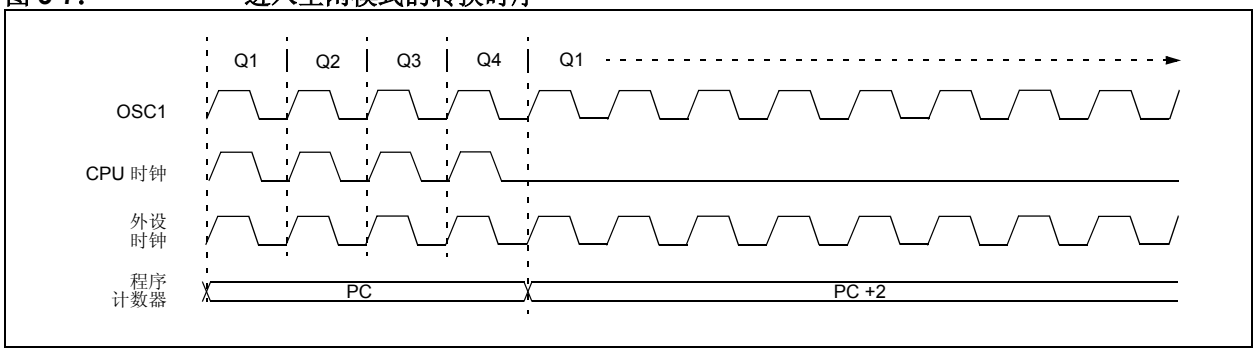
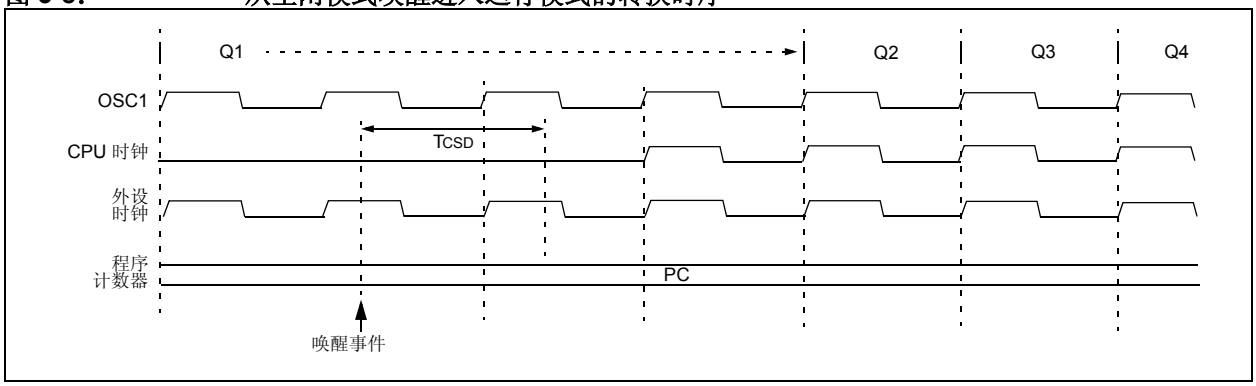


图 3-8： 从空闲模式唤醒进入运行模式的转换时序



 © 2007 Microchip Technology Inc.

3.4.3 RC_IDLE 模式

在 RC_IDLE 模式下，CPU 被禁止，但外设仍继续使用内部振荡器电路的 INTOSC 复用器作为时钟源。使用该模式可在空闲期间对功耗进行控制。

通过将IDLEN位置1并执行SLEEP指令可以从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。当时钟源切换到 INTOSC 时，主振荡器被关闭， OSTS 位被清零。

当唤醒事件发生时，外设继续将内部振荡器作为时钟源。在唤醒事件后的 TCSD 延时后，CPU 使用 INTRC 作为时钟源并开始执行代码。切换不会影响IDLEN和SCS位。如果 WDT 或故障保护时钟监视器被使能， INTRC 时钟源将继续运行。

3.5 退出空闲和休眠模式

中断、复位或 WDT 超时将触发退出休眠模式或任何空闲模式。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讨论了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 INTCON 或 PIE 寄存器中的中断源允许位置 1。当相应的中断标志位置 1 时，触发退出操作。

当通过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矢量处执行代码。否则代码就会顺序执行 （见**第 9.0 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 需要此延时来为执行代码做准备。在此延时后的第一个时钟周期恢复指令执行。

3.5.2 通过 WDT 超时退出

WDT 根据超时发生时器件所处的不同功耗管理模式会进行不同的操作。

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 24.2 节 “看门狗定时器 （WDT）”**）。

WDT 和后分频器将被以下任一事件清零：

- 执行 SLEEP 或 CLRWDT 指令
- 当前选定的时钟源失效 （故障保护时钟监视器使能时）

3.5.3 通过复位退出

通过复位从空闲或休眠模式退出，自动强制器件使用 INTRC 作为时钟源运行。

3.5.4 无需振荡器起振延时的退出

从某些功耗管理模式退出不会引起 OST 延时。有两种情形：

- 主时钟源一直工作的 PRI_IDLE 模式
- 主时钟源处于 EC 或 ECPLL 模式。

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 （RC、EC 和 INTIO 振荡器模式）。然而，当器件退出休眠和空闲模式时，在唤醒事件后仍需要一个固定的延时TCSD，以便让CPU为执行代码做好准备。指令将在此延时后的第一个时钟周期恢复执行。

注：

 © 2007 Microchip Technology Inc.

4.0 复位

PIC18F87J11 系列器件共有以下几种复位：

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) 配置不匹配 （CM）
f) 欠压复位 （BOR）
g) RESET 指令
h) 堆栈满复位
i) 堆栈下溢复位

本节将讨论由 $\overline{\text{MCLR}}$、POR 和 BOR 产生的各种复位以及各种起振定时器的操作。**第 5.1.6.4 节 “堆栈满和下溢复位”**将介绍堆栈复位事件。WDT 复位将在**第 24.2 节 “看门狗定时器 （WDT）”**中讨论。

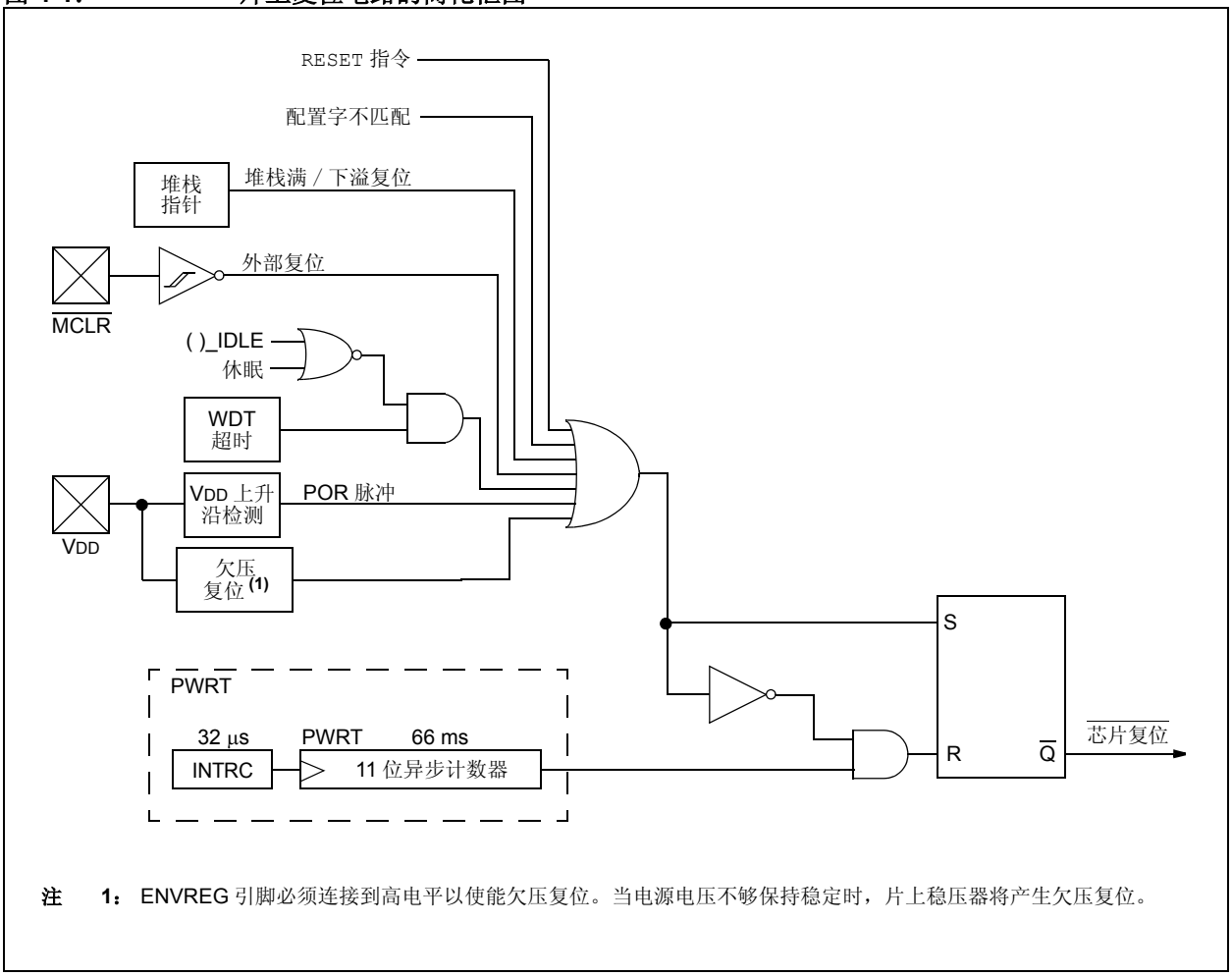
图 4-1 给出了片上复位电路的简化框图。

4.1 RCON 寄存器

可通过 RCON 寄存器 （寄存器 4-1）跟踪器件复位事件。该寄存器的低 5 位表示特定的复位事件是否已经发生。在大部分情况下，只有复位事件可以将这些位置 1，而且必须在此事件之后由应用程序将它们清零。读这 5 位标志位的状态可以知道刚刚发生过的复位的类型。**第 4.7 节 “寄存器的复位状态”**中对此进行了更详细地说明。

RCON 寄存器还有一个设置中断优先级的控制位 （IPEN）。**第 9.0 节 “中断”**将详细讨论中断优先级。

图 4-1： 片上复位电路的简化框图



注 1： ENVREG 引脚必须连接到高电平以使能欠压复位。当电源电压不够保持稳定时，片上稳压器将产生欠压复位。

寄存器 4-1：　　RCON：复位控制寄存器

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

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	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　**未实现**：读作为 0

bit 5　**$\overline{CM}$**：配置不匹配标志位
1 = 未发生配置不匹配复位
0 = 发生配置不匹配复位 （必须在发生配置不匹配复位后用软件置位）

bit 4　**$\overline{RI}$**：RESET 指令标志位
1 = 未执行 RESET 指令 （仅由固件置位）
0 = 已执行 RESET 指令，导致器件复位 （必须在发生欠压复位之后用软件置位）

bit 3　**$\overline{TO}$**：看门狗定时器超时标志位
1 = 通过上电、 CLRWDT 指令或 SLEEP 指令置位
0 = 发生了 WDT 超时

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

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

bit 0　**$\overline{BOR}$**：欠压复位状态位
1 = 未发生欠压复位 （仅由固件置位）
0 = 发生了欠压复位 （必须在欠压复位发生之后由软件置位）

注　1：建议在检测到上电复位后，将 $\overline{POR}$ 位置位，以便检测后续的上电复位。
2：如果禁用了片上稳压器，则 $\overline{BOR}$ 将始终为 0。更多信息，请参见**第 4.4.1 节 “检测 BOR”**。
3：欠压复位是指当 $\overline{BOR}$ 为 0 且 $\overline{POR}$ 为 1 时发生的复位 （假设在上电复位之后立即由软件置 1 $\overline{POR}$）。

© 2007 Microchip Technology Inc.

4.2 主复位（MCLR）

MCLR 引脚提供触发器件的外部硬复位的方法。将该引脚拉低可以产生复位信号。PIC18 扩展单片机器件在 MCLR 复位信号传输路径中有一个噪声滤波器，它可以检测并滤除小脉冲信号的干扰。

任何内部复位（包括 WDT 复位）都不能将 MCLR 引脚驱动为低电平。

4.3 上电复位（POR）

只要 VDD 超过设定的门限值之后，就会在片上产生上电复位脉冲。这使得在VDD达到满足器件正常工作的数值时，器件会初始化并启动。

要使用 POR 电路，需要将 MCLR 引脚通过一个电阻（阻值为 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 的最小上升速率在参数 D004 中指定。对于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，必须满足器件的工作参数（电压、频率和温度等），才能够确保其正常工作。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。

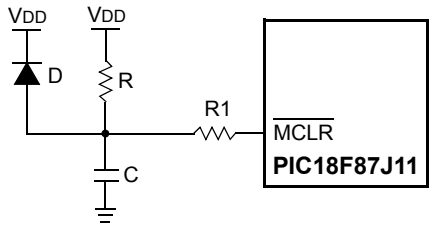
上电复位事件由 POR 位（RCON<1>）捕获。每当发生 POR 时，该位的状态就会被置为 0；任何其他的复位事件均不能更改它。任何硬件事件都不能将 POR 复位为 1。要捕获多个事件，用户必须在任何 POR 之后用软件手动地将该位复位为 1。

4.4 欠压复位（BOR）

当使能内部稳压器时（ENVREG 引脚连接到 VDD），PIC18F87J10 系列器件将具有简单的 BOR 功能。只要 VDD 低于 VBOR（参数 **D005**）的时间超过 TBOR（参数 35）就会复位器件。如果 VDD 降到 VBOR 以下的持续时间短于 TBOR，就不一定会发生复位。芯片将保持欠压复位状态，直至 VDD 电压上升到 VBOR 以上。

一旦发生了 BOR，上电延时定时器将使芯片在延时 TPWRT（参数 33）期间保持复位。如果上电延时定时器运行期间，VDD 电压降到 VBOR 以下，芯片将重新回到欠压复位状态，且将初始化上电延时定时器。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此操作。

图 4-2：　外部上电复位电路（VDD 缓慢上电）



注　1： 仅当 VDD 上电斜率过慢时，才需要外部上电复位电路。当 VDD 掉电时，二极管 D 有助于电容迅速放电。

2： 建议 R < 40 kΩ，以确保电阻 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 引脚击穿。

4.4.1 检测 BOR

任何欠压复位或上电复位事件的发生通常都会使 BOR 位复位至 0。因此仅通过读 BOR 的状态很难确定是否发生了欠压复位事件。更可靠的方法是同时检查 POR 和 BOR 的状态。假定在发生任何上电复位事件后，立即用软件将 POR 位置 1。如果 BOR 为 0 而 POR 为 1，就可以断定发生了欠压复位事件。

如果禁用了稳压器，也就欠禁用了压复位功能。在这种情况下，不能使用 BOR 位来确定是否发生了欠压复位事件。欠压复位事件仍然会清零 BOR 位。

4.5 配置不匹配（CM）

配置不匹配（Configuration Mismatch，CM）复位设计用于检测随机的存储器故障事件，并从上述事件中恢复。这些事件包括静电放电（ESD）事件，该事件能够导致整个器件内的大范围的单个位更改，并产生灾难性故障。

在 PIC18FXXJ 闪存器件中，通过比较器件配置寄存器（位于配置存储器空间）的值和互补影子寄存器的值，以在工作期间持续监测此配置寄存器。如果检测到两组寄存器不匹配，那么 CM 将自动复位。这些事件由 CM 位（RCON<5>）捕捉。只要发生 CM 事件，CM 位的状态就被置为 0；它不会因为任何其他复位事件而发生改变。

CM 复位操作类似于主清零复位、 RESET 指令、 WDT 超时或堆栈事件复位。与发生所有硬复位和电源复位事件一样，在器件重启时，器件配置字会从程序存储器的闪存配置字中被重新装载。

4.6 上电延时定时器 （PWRT）

PIC18F87J10 系列器件添加了片上上电延时定时器（PWRT）以帮助稳定上电复位过程。PWRT 总是使能的。其主要功能是确保在执行代码之前，器件的电压是稳定的。

PIC18F87J10 系列器件的上电延时定时器（PWRT）是一个 11 位计数器，使用 INTRC 时钟源作为时钟输入。产生约 2048 x 32 μs = 66 ms 的时间间隔。当 PWRT 计数时，器件保持在复位状态。

上电延时取决于 INTRC 时钟，并且由于温度和工艺的不同，不同芯片的延时也各不相同。详细信息，请参见 DC 参数 **33**。

4.6.1 延时序列

如果使能，在 POR 脉冲被清零后将调用 PWRT 延时。总延时将根据 PWRT 的状态而有所不同。图 4-3、图 4-4、图 4-5 和图 4-6 分别给出了不同的上电延时时序，其中上电延时定时器均被使能。

由于延时是由 POR 脉冲触发的，因此若 $\overline{\text{MCLR}}$ 保持足够长时间的低电平， PWRT 将超时。将 MCLR 电平拉高后程序将立即执行代码 （图 4-5）。这对于测试或同步多个并行工作的 PIC18FXXXX 器件来说非常有用。

图 4-3： 上电时的延时时序 （$\overline{\text{MCLR}}$ 连接到 VDD， VDD 上升时间 < TPWRT）

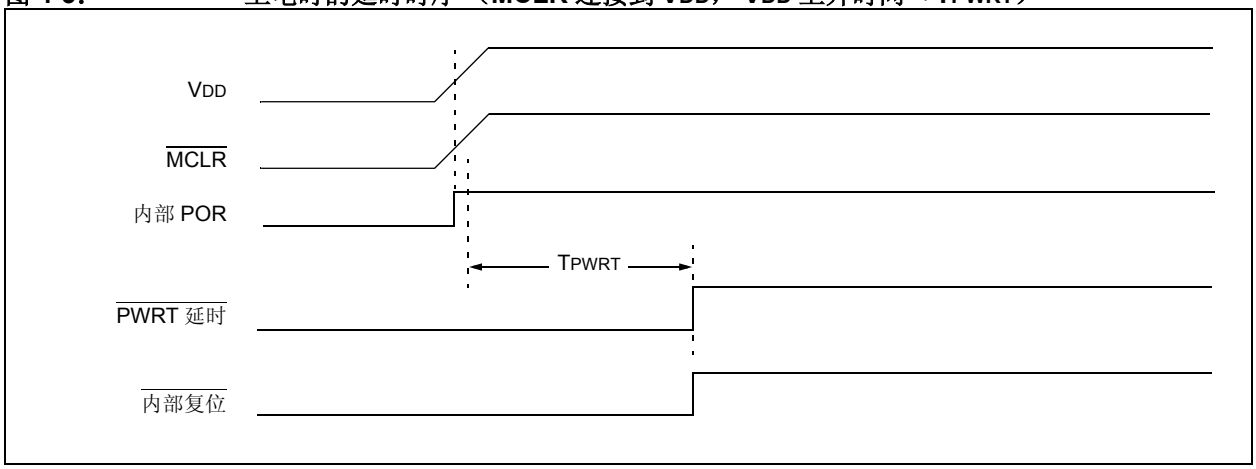
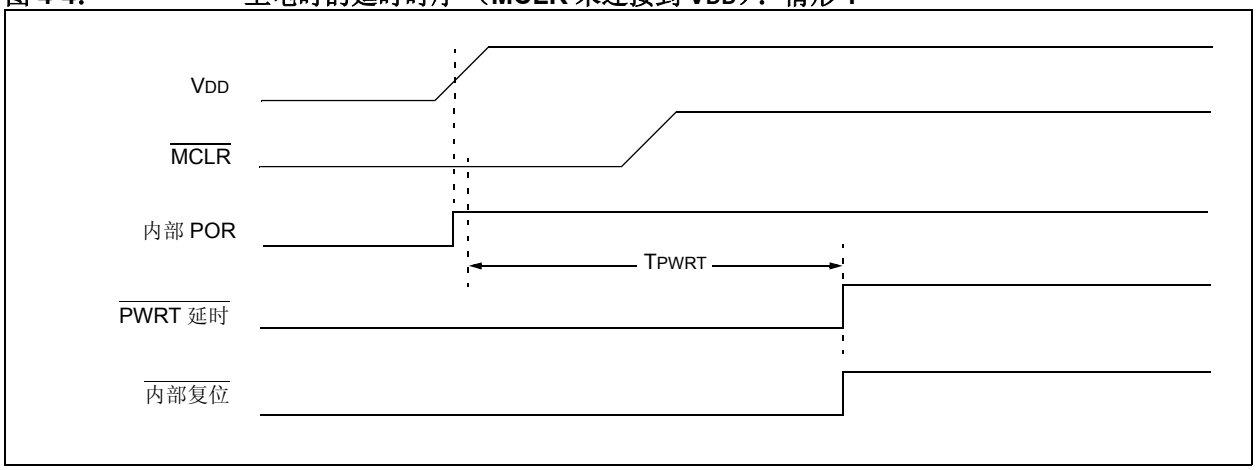


图 4-4： 上电时的延时时序 （$\overline{\text{MCLR}}$ 未连接到 VDD）：情形 1



© 2007 Microchip Technology Inc.

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

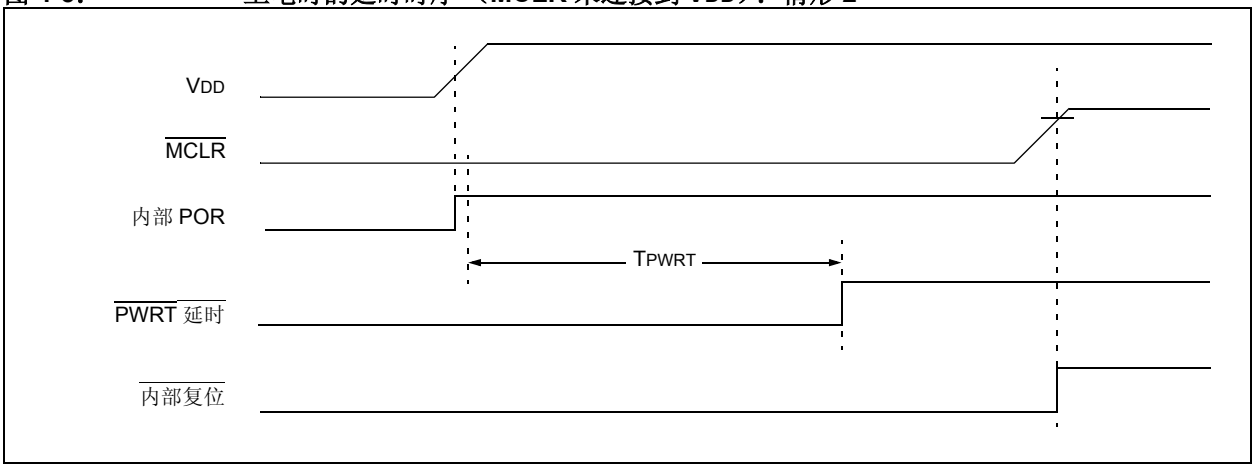
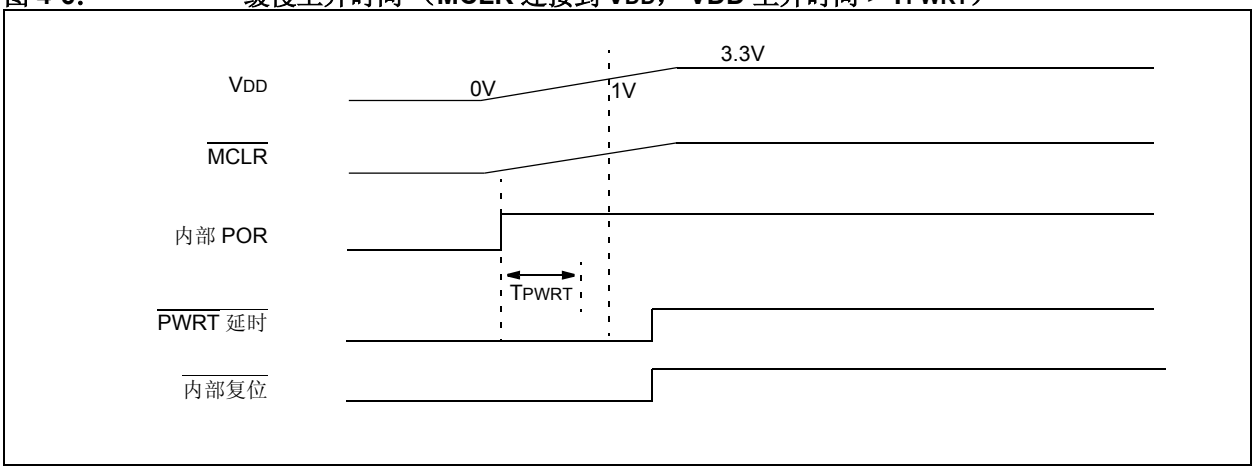


图 4-6：　　缓慢上升时间（$\overline{\text{MCLR}}$ 连接到 VDD，VDD 上升时间 > TPWRT）



4.7 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。而其他寄存器则根据不同的复位类型被强制为“复位状态”。

因为 WDT 唤醒被视为恢复正常工作，所以大多数寄存器不受 WDT 唤醒的影响。RCON 寄存器的状态位（CM、RI、TO、PD、POR 和 BOR）在不同的复位情形中分别被置位或清零，如表 4-1 中所示。可在软件中使用这些状态位判断复位的类型。

表 4-2 说明了所有特殊功能寄存器的复位状态。这些复位被分类为上电和欠压复位、主复位和 WDT 复位以及 WDT 唤醒复位。

表 4-1： RCON 寄存器的状态位、含义和初始状态

条件	程序计数器 (1)	RCON 寄存器						STKPTR 寄存器	
		CM	RI	TO	PD	POR	BOR	STKFUL	STKUNF
上电复位	0000h	1	1	1	1	0	0	0	0
RESET 指令	0000h	u	0	u	u	u	u	u	u
欠压复位	0000h	1	1	1	1	u	0	u	u
配置不匹配复位	0000h	0	u	u	u	u	u	u	u
功耗管理运行模式期间的 MCLR 复位	0000h	u	u	1	u	u	u	u	u
功耗管理空闲模式和休眠模式期间的 MCLR 复位	0000h	u	u	1	0	u	u	u	u
全功耗期间的 MCLR 复位	0000h	u	u	u	u	u	u	u	u
堆栈满复位 （STVREN=1）	0000h	u	u	u	u	u	u	1	u
堆栈下溢复位 （STVREN = 1）	0000h	u	u	u	u	u	u	u	1
堆栈下溢错误 （不是真正的复位， STVREN=0）	0000h	u	u	u	u	u	u	u	1
全功耗或功耗管理运行模式期间的 WDT 超时	0000h	u	u	0	u	u	u	u	u
功耗管理空闲或休眠模式期间的 WDT 超时	PC +2	u	u	0	0	u	u	u	u
通过中断从功耗管理模式退出	PC +2	u	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 位被置 1 时，向 PC 装入中断向量 （0008h 或 0018h）。

© 2007 Microchip Technology Inc.

表 4-2： 所有寄存器的初始化状态

寄存器	适用器件		上电复位和欠压复位	$\overline{\text{MCLR}}$ 复位、WDT 复位、RESET 指令、堆栈复位和 CM 复位	通过 WDT 或中断唤醒
TOSU	PIC18F6XJ1X	PIC18F8XJ1X	---0 0000	---0 0000	---0 uuuu(1)
TOSH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu(1)
TOSL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu(1)
STKPTR	PIC18F6XJ1X	PIC18F8XJ1X	00-0 0000	uu-0 0000	uu-u uuuu(1)
PCLATU	PIC18F6XJ1X	PIC18F8XJ1X	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	PC + 2(2)
TBLPTRU	PIC18F6XJ1X	PIC18F8XJ1X	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F6XJ1X	PIC18F8XJ1X	0000 000x	0000 000u	uuuu uuuu(3)
INTCON2	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu(3)
INTCON3	PIC18F6XJ1X	PIC18F8XJ1X	1100 0000	1100 0000	uuuu uuuu(3)
INDF0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTINC0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTDEC0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PREINC0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PLUSW0	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
FSR0H	PIC18F6XJ1X	PIC18F8XJ1X	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTINC1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTDEC1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PREINC1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PLUSW1	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
FSR1H	PIC18F6XJ1X	PIC18F8XJ1X	---- xxxx	---- uuuu	---- uuuu
FSR1L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F6XJ1X	PIC18F8XJ1X	---- 0000	---- 0000	---- uuuu

图注： u = 不变， x = 未知， - = 未实现位，读为 0， q = 取值依情况而定。
阴影单元格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： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断矢量 （0008h 或 0018h）。

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

4： 特定条件下的复位值，请参见表 4-1。

© 2007 Microchip Technology Inc.

表 4-2： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位和欠压复位	MCLR 复位、WDT 复位、RESET 指令、堆栈复位和 CM 复位	通过 WDT 或中断唤醒
INDF2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTINC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
POSTDEC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PREINC2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
PLUSW2	PIC18F6XJ1X	PIC18F8XJ1X	N/A	N/A	N/A
FSR2H	PIC18F6XJ1X	PIC18F8XJ1X	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	PIC18F6XJ1X	PIC18F8XJ1X	---x xxxx	---u uuuu	---u uuuu
TMR0H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TMR0L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F6XJ1X	PIC18F8XJ1X	0110 q100	0110 q100	0110 q10u
REFOCON	PIC18F6XJ1X	PIC18F8XJ1X	0-00 0000	u-uu uuuu	u-uu uuuu
CM1CON	PIC18F6XJ1X	PIC18F8XJ1X	0001 1111	uuuu uuuu	uuuu uuuu
CM2CON	PIC18F6XJ1X	PIC18F8XJ1X	0001 1111	uuuu uuuu	uuuu uuuu
RCON(4)	PIC18F6XJ1X	PIC18F8XJ1X	0-11 1100	0-qq qquu	u-qq qquu
TMR1H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ODCON1	PIC18F6XJ1X	PIC18F8XJ1X	---0 0000	---u uuuu	---u uuuu
TMR1L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ODCON2	PIC18F6XJ1X	PIC18F8XJ1X	---- --00	---- --uu	---- --uu
T1CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	u0uu uuuu	uuuu uuuu
ODCON3	PIC18F6XJ1X	PIC18F8XJ1X	---- --00	---- --uu	---- --uu
TMR2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PADCFG1	PIC18F6XJ1X	PIC18F8XJ1X	---- ---0	---- ---u	---- ---u
PR2	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	1111 1111
MEMCON	PIC18F6XJ1X	PIC18F8XJ1X	0-00 --00	0-00 --00	u-uu --uu
T2CON	PIC18F6XJ1X	PIC18F8XJ1X	-000 0000	-000 0000	-uuu uuuu
SSP1BUF	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP1ADD	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1MSK	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	uuuu uuuu	uuuu uuuu
SSP1STAT	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1CON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP1CON2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu

图注： u = 不变， x = 未知， - = 未实现位，读为 0， q = 取值依情况而定。
阴影单元格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： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断矢量 （0008h 或 0018h）。

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

4： 特定条件下的复位值，请参见表 4-1。

表 4-2： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位和欠压复位	MCLR 复位、WDT 复位、RESET 指令、堆栈复位和 CM 复位	通过 WDT 或中断唤醒
ADRESH	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
ADCON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
ANCON0	PIC18F6XJ1X	PIC18F8XJ1X	00-0 0000	uu-u uuuu	uu-u uuuu
ANCON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	uuuu uuuu	uuuu uuuu
WDTCON	PIC18F6XJ1X	PIC18F8XJ1X	0x-0 ---0	0x-u ---0	ux-u ---u
ECCP1AS	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
ECCP1DEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
CCPR1H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
ECCP2AS	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
ECCP2DEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
CCPR2H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
ECCP3AS	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
ECCP3DEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
CCPR3H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR3L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP3CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SPBRG1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
RCREG1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TXREG1	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXSTA1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0010	0000 0010	uuuu uuuu
RCSTA1	PIC18F6XJ1X	PIC18F8XJ1X	0000 000x	0000 000x	uuuu uuuu
SPBRG2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
RCREG2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TXREG2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TXSTA2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0010	0000 0010	uuuu uuuu
EECON2	PIC18F6XJ1X	PIC18F8XJ1X	---- ----	---- ----	---- ----
EECON1	PIC18F6XJ1X	PIC18F8XJ1X	--00 x00-	--00 u00-	--00 u00-

图注： u = 不变， x = 未知， - = 未实现位，读为 0， q = 取值依情况而定。
阴影单元格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： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断矢量 （0008h 或 0018h）。

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

4： 特定条件下的复位值，请参见表 4-1。

© 2007 Microchip Technology Inc.

表 4-2： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位和欠压复位	MCLR 复位、WDT 复位、RESET 指令、堆栈复位和 CM 复位	通过 WDT 或中断唤醒
IPR3	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
PIR3	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu(3)
PIE3	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
IPR2	PIC18F6XJ1X	PIC18F8XJ1X	111- 1111	111- 1111	uuu- uuuu
PIR2	PIC18F6XJ1X	PIC18F8XJ1X	000- 0000	000- 0000	uuu- uuuu(3)
PIE2	PIC18F6XJ1X	PIC18F8XJ1X	000- 0000	000- 0000	uuu- uuuu
IPR1	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
PIR1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu(3)
PIE1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
RCSTA2	PIC18F6XJ1X	PIC18F8XJ1X	0000 000x	0000 000x	uuuu uuuu
OSCTUNE	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
TRISJ	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISH	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISG	PIC18F6XJ1X	PIC18F8XJ1X	---1 1111	---1 1111	---u uuuu
TRISF	PIC18F6XJ1X	PIC18F8XJ1X	1111 111-	1111 111-	uuuu uuu-
TRISE	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISD	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
TRISA	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	uuuu uuuu
LATJ	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATH	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATG	PIC18F6XJ1X	PIC18F8XJ1X	---x xxxx	---u uuuu	---u uuuu
LATF	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxx-	uuuu uuu-	uuuu uuu-
LATE	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATD	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu

图注： u = 不变， x = 未知， - = 未实现位，读为 0， q = 取值依情况而定。
阴影单元格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： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断矢量 （0008h 或 0018h）。

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

4： 特定条件下的复位值，请参见表 4-1。

表 4-2： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位和欠压复位	MCLR 复位、WDT 复位、RESET 指令、堆栈复位和 CM 复位	通过 WDT 或中断唤醒
PORTJ	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTH	PIC18F6XJ1X	PIC18F8XJ1X	0000 xxxx	uuuu uuuu	uuuu uuuu
PORTG	PIC18F6XJ1X	PIC18F8XJ1X	000x xxxx	000u uuuu	uuuu uuuu
PORTF	PIC18F6XJ1X	PIC18F8XJ1X	x001 100-	xuuu uuu-	xuuu uuu-
PORTE	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	PIC18F6XJ1X	PIC18F8XJ1X	000x 0000	000u 0000	uuuu uuuu
SPBRGH1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
BAUDCON1	PIC18F6XJ1X	PIC18F8XJ1X	0100 0-00	0100 0-00	uuuu u-uu
SPBRGH2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
BAUDCON2	PIC18F6XJ1X	PIC18F8XJ1X	0100 0-00	0100 0-00	uuuu u-uu
TMR3H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	uuuu uuuu	uuuu uuuu
TMR4	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PR4	PIC18F6XJ1X	PIC18F8XJ1X	1111 1111	1111 1111	1111 1111
CVRCON	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
T4CON	PIC18F6XJ1X	PIC18F8XJ1X	-000 0000	-000 0000	-uuu uuuu
CCPR4H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR4L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP4CON	PIC18F6XJ1X	PIC18F8XJ1X	--00 0000	--00 0000	--uu uuuu
CCPR5H	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR5L	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP5CON	PIC18F6XJ1X	PIC18F8XJ1X	--00 0000	--00 0000	--uu uuuu
SSP2BUF	PIC18F6XJ1X	PIC18F8XJ1X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP2ADD	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP2MSK	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP2STAT	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
CMSTAT	PIC18F6XJ1X	PIC18F8XJ1X	---- --11	---- --11	---- --uu

图注： u = 不变， x = 未知， - = 未实现位，读为 0， q = 取值依情况而定。
阴影单元格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： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断矢量 （0008h 或 0018h）。

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

4： 特定条件下的复位值，请参见表 4-1。

© 2007 Microchip Technology Inc.

表 4-2： 所有寄存器的初始化状态 （续）

寄存器	适用器件		上电复位和欠压复位	MCLR 复位、WDT 复位、RESET 指令、堆栈复位和 CM 复位	通过 WDT 或中断唤醒
PMADDRH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT1H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMADDRL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT1L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN1H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN1L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMCONH	PIC18F6XJ1X	PIC18F8XJ1X	0-00 0000	0-00 0000	u-uu uuuu
PMCONL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMMODEH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMMODEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT2H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDOUT2L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN2H	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMDIN2L	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMEH	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMEL	PIC18F6XJ1X	PIC18F8XJ1X	0000 0000	0000 0000	uuuu uuuu
PMSTATH	PIC18F6XJ1X	PIC18F8XJ1X	00-- 0000	00-- 0000	uu-- uuuu
PMSTATL	PIC18F6XJ1X	PIC18F8XJ1X	10-- 1111	10-- 1111	uu-- uuuu

图注： u = 不变， x = 未知， - = 未实现位，读为 0， q = 取值依情况而定。
阴影单元格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： 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时， PC 装入中断矢量 （0008h 或 0018h）。

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

4： 特定条件下的复位值，请参见表 4-1。

© 2007 Microchip Technology Inc.

5.0 存储器构成

PIC18 闪存单片机有 2 种类型的存储器：

- 程序存储器
- 数据 RAM

在哈佛架构的器件中，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。

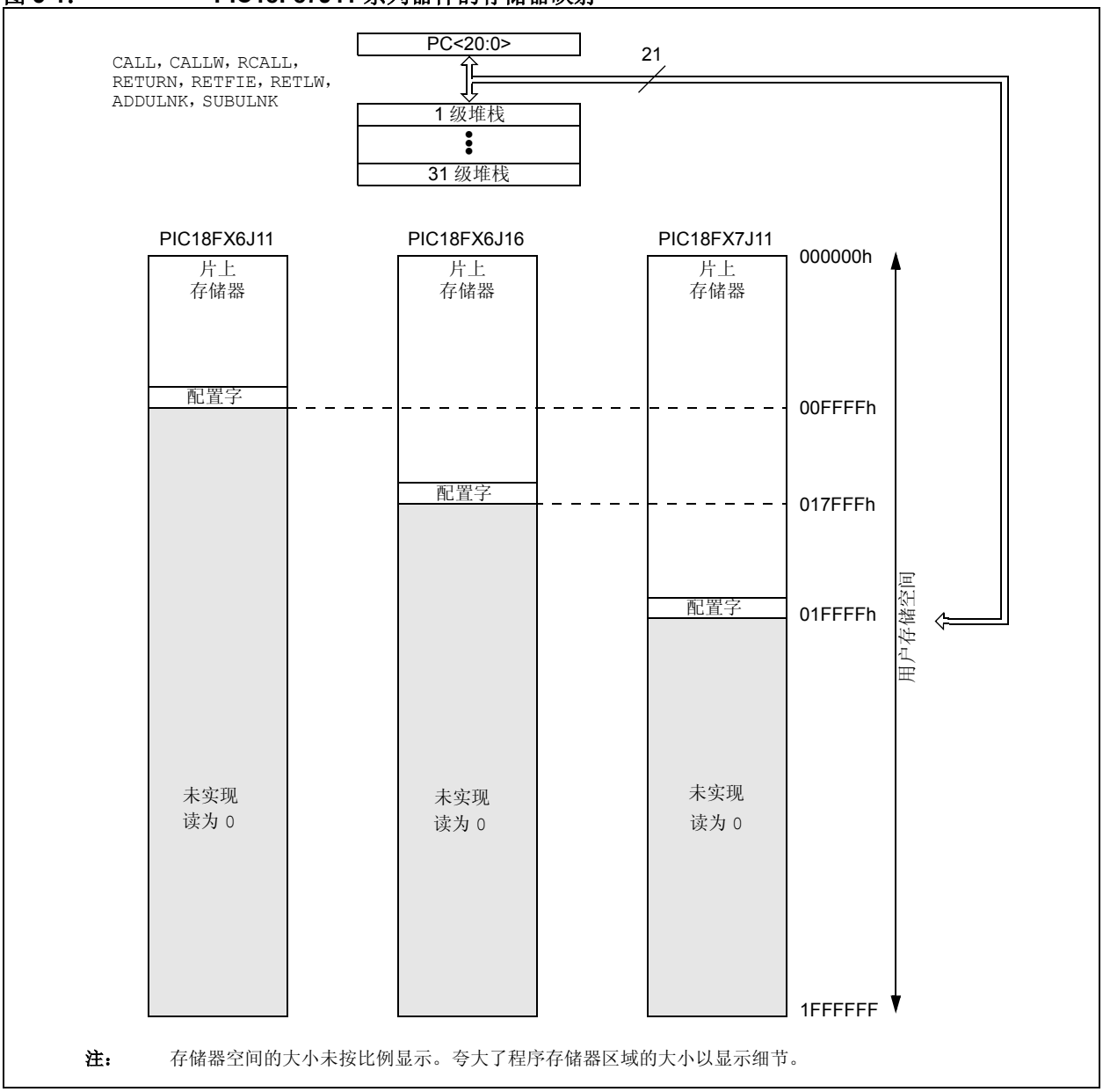
第 6.0 节“闪存程序存储器”提供了闪存程序存储器操作的详细信息。

5.1 程序存储器构成

PIC18 单片机实现了 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2 MB地址之间的存储单元将会返回全0（NOP指令）。

整个 PIC18F87J11 系列器件提供了 3 种不同的片上闪存程序存储空间：从 64 KB（至多 16,384 条单字指令）到 128 KB （65,536 条单字指令）。该系列的各个器件的程序存储器映射如图 5-3 所示。

图 5-1： PIC18F87J11 系列器件的存储器映射



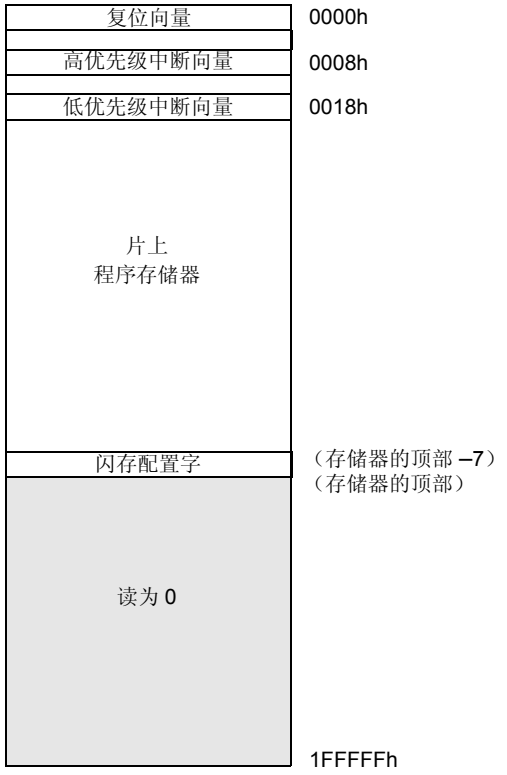
注： 存储器空间的大小未按比例显示。夸大了程序存储器区域的大小以显示细节。

5.1.1 存储器硬件编码向量

所有的 PIC18 器件在其程序存储器空间内共有 3 个硬件编码的返回向量。复位向量地址是在器件发生任何复位时程序计数器返回的默认值；它位于 0000h。

PIC18 器件还有两个中断向量地址，用于处理高优先级和低优先级中断。高优先级中断向量位于 0008h，低优先级中断向量位于 0018h。它们相对于程序存储器映射的位置如图 5-2 所示。

图 5-2： PIC18F87J11 系列器件的硬件编码向量和配置字单元



图注： （存储器的顶部）代表片上程序存储器空间的上边界（有关器件特定的参数见图 5-1）。阴影区域表示未用的存储空间。这些区域未按照比例绘制。

5.1.2 闪存配置字

由于 PIC18F87J11 系列器件没有固定的配置存储器，所以保留片上程序存储器顶部的 4 个字来保存配置信息。复位时，配置信息被复制到配置寄存器。

配置字根据数字大小由低到高顺序存储在程序存储器单元中，从最低地址开始存放 CONFIG1 的低字节，到 CONFIG4 的高字节结束。对于这些器件，只使用从 CONFIG1 到 CONFIG3 的配置字，保留 CONFIG4。PIC18F87J11 系列器件的闪存配置字的实际地址如表 5-1 所示。图 5-2 显示了闪存配置字以及其他的存储器向量在存储器映射中的位置。

第 24.1 节 “配置位” 中提供了有关器件配置字的更多详细信息。

表 5-1： PIC18F87J11 系列器件的闪存配置字

器件	程序存储器（KB）	配置字地址
PIC18F66J11	64	FFF8h 到 FFFFh
PIC18F86J11		
PIC18F66J16	96	17FF8h 到 17FFFh
PIC18F86J16		
PIC18F67J11	128	1FFF8h 到 1FFFFh
PIC18F87J11		

© 2007 Microchip Technology Inc.

5.1.3 PIC18F8XJ11/8XJ16 程序存储器模式

此系列中的80引脚器件可以对总共2 MB的程序存储器空间进行寻址。这是通过外部存储器总线实现的。控制器有两种不同的工作模式：

- 单片机（MC）
- 扩展单片机（EMC）

通过设置EMB配置位（CONFIG3L<5:4>）来决定程序存储器的模式，如寄存器5-1所示（欲知有关器件配置位的更多详细信息，请参见**第 24.1 节“配置位”**）。

程序存储器模式的工作方式如下：

- **单片机模式**只访问片上闪存存储器。尝试读片上存储器顶部以上的地址单元将会导致读为全0（相当于执行NOP指令）。

  单片机模式也是64引脚器件惟一可用的工作模式。

- **扩展单片机模式**允许将内部和外部程序存储器作为一个存储块进行访问。器件可以访问其整个的片上程序存储器；除此之外，器件可以访问的外部程序存储器最大限制为2 MB的程序存储器空间。按照需要自动在两个存储器之间切换执行。

EMB 配置位的设置还能控制外部存储器总线的地址总线宽度。在**第 7.0 节“外部存储总线”**中对此有更详细的论述。

在这两种模式下，单片机都能完全地访问数据RAM。

图5-3比较了不同程序存储器模式的存储器映射。表5-2中更充分地说明了片上和外部存储器访问限制之间的差异。

寄存器 5-1： CONFIG3L：配置寄存器 3 低字节

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT(1)	BW(1)	EMB1(1)	EMB0(1)	EASHFT(1)	—	—	—
bit 7							bit 0

图注：	WO = 一次写入位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
－n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **WAIT**：外部总线等待使能位 (1)
1 = 禁止外部总线上的等待状态
0 = 通过 MEMCON<5:4> 选择并使能外部总线上的等待状态

bit 6 **BW**：数据总线宽度选择位 (1)
1 = 16 位数据宽度模式
0 = 8 位数据宽度模式

bit 5-4 **EMB1:EMB0**：外部存储器总线配置位 (1)
11 = 单片机模式，禁止外部总线
10 = 扩展单片机模式， 12 位地址宽度的外部总线
01 = 扩展单片机模式， 16 位地址宽度的外部总线
00 = 扩展单片机模式， 20 位地址宽度的外部总线

bit 3 **EASHFT**：外部地址总线移位使能位 (1)
1 = 使能地址移位——外部地址总线被移位至从 000000h 开始
0 = 禁止地址移位——外部地址总线反映 PC 值

bit 2-0 **未实现**：读为 0

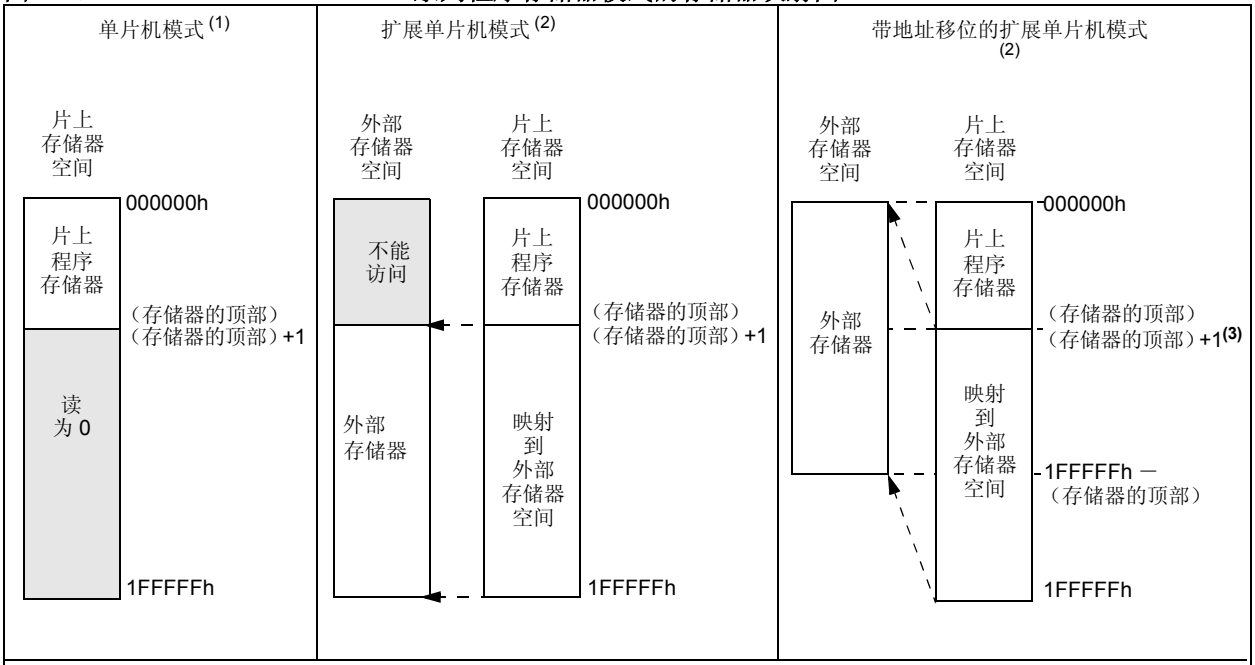
注 1： 仅在 80 引脚器件上实现。

5.1.4 扩展单片机模式和地址移位

默认情况下，处于扩展单片机模式的器件将指向外部存储器空间范围内的地址的程序计数器值直接放到外部地址总线上。实际上对单片机而言，低于片上存储器最高地址的外部存储器件的地址是不可用的。

为了避免这种情况，扩展单片机模式使用了一个地址移位功能以使能自动的地址转换。在此模式下，放在外部总线上的地址值减去片上程序存储器的大小并重新映射到从0000h开始的外部存储器地址。这样就完全将外部存储器件的存储器空间用作器件片上程序存储器的扩展。

图 5-3： PIC18F87J11 系列程序存储器模式的存储器映射图



图注： （存储器的顶部）代表片上程序存储器空间的上边界 （有关器件特定的参数见图 5-1）。阴影区域代表未实现或不能访问的区域，这取决于不同模式。

注 1： 此模式是 64 引脚器件惟一可用的模式，也是 80 引脚器件的默认模式。

2： 这些模式仅在 80 引脚器件上可用。

3： 只要 EASHFT 配置位被置 1，由程序存储器顶部开始的地址就被转换为从外部器件的 0000h 开始的地址。

表 5-2： PIC18F8X11/8616 各种程序存储器模式下的存储器访问

工作模式	内部程序存储器			外部程序存储器		
	执行程序	表读	表写	执行程序	表读	表写
单片机模式	可以	可以	可以	不可以	不可以	不可以
扩展单片机模式	可以	可以	可以	可以	可以	可以

© 2007 Microchip Technology Inc.

5.1.5 程序计数器

程序计数器（Program Counter，PC）指定要取出执行的指令地址。PC 内的地址为 21 位宽，并且保存在 3 个独立的 8 位寄存器中。其中的低字节称为 PCL 寄存器，该寄存器可读写。 高字节，即 PCH 寄存器，存储 PC<15:8> 位，不可直接读写。可以通过 PCLATH 寄存器更新 PCH 寄存器。最高字节称为 PCU。该寄存器存储 PC<20:16> 位，它也不能直接读写。通过 PCLATU 寄存器更新 PCU 寄存器。

通过任何写 PCL 的操作，可以将 PCLATH 和 PCLATU 的内容传送到程序计数器。同样，通过执行读 PCL 的操作，可以将程序计数器的两个高字节传送到 PCLATH 和 PCLATU。这对于计算 PC 的偏移量很有用（见**第 5.1.8.1 节 “计算 GOTO”**）

PC 在程序存储器中按字节寻址。为防止 PC 不能正确获取指令字，需要将 PCL 的最低有效位固定为 0。 PC 每次加 2 来连续寻址程序存储器中的指令。

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

5.1.6 返回地址堆栈

用于存放返回地址的堆栈允许保存最多 31 个程序调用地址和中断向量。当执行 CALL 或 RCALL 指令或响应中断时， PC 值被压入堆栈。而执行 RETURN、RETLW 或 RETFIE 指令（如果使能了扩展指令集，则还包括 ADDULNK 和 SUBULNK 指令）时， PC 值从堆栈弹出。PCLATU 和 PCLATH 不受任何 RETURN 或 CALL 指令的影响。

通过 21 位的 RAM 和 5 位的堆栈指针 （STKPTR）来实现 31 级的堆栈操作。堆栈既不占用程序存储空间也不占用数据存储空间。堆栈指针可以读写，并且通过栈顶的特殊文件寄存器可以读写栈顶地址。也可使用这些寄存器将数据压入堆栈，或将数据从堆栈弹出。

执行 CALL 类型指令引起进栈操作：堆栈指针首先加 1，并且将 PC 的内容写入堆栈指针指向的地址单元 （PC 已经指向 CALL 的下一条指令）。执行 RETURN 类型指令时，引起出栈操作：STKPTR 寄存器所指向的地址单元的内容被传送给 PC，然后堆栈指针减 1。

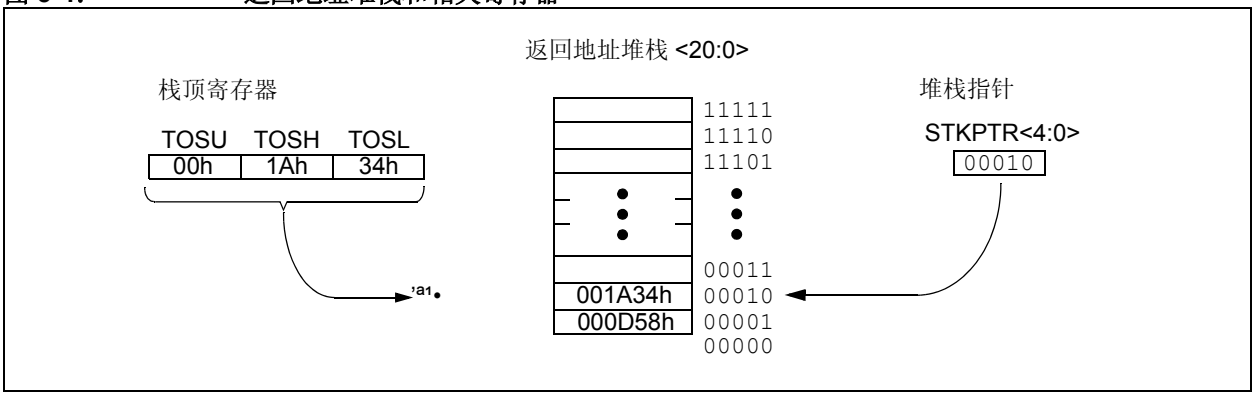
所有复位后，堆栈指针被初始化为 00000。堆栈指针值 00000 不指向任何 RAM 单元，它只是一个复位值。状态位表明堆栈是满、上溢还是下溢。

5.1.6.1 访问栈顶

只有栈顶 （Top-of-Stack， TOS）是可读写的。有 3 个寄存器 TOSU:TOSH:TOSL 用于保存 STKPTR 寄存器（图 5-4）所指向的堆栈单元的内容。这可以让用户在必要时实现软件堆栈。在 CALL、 RCALL 或中断 （如果使能了扩展指令集，则还包括 ADDULNK 和 SUBULNK 指令）时，软件可以通过读取 TOSU:TOSH:TOSL 寄存器来读取进栈值。这些值可以被置入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。

为防止对堆栈的意外操作，访问堆栈时用户必须禁止全局中断允许位。

图 5-4： 返回地址堆栈和相关寄存器



5.1.6.2 返回堆栈指针 （STKPTR）

STKPTR 寄存器（寄存器 5-2）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 之间的整数。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。

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

堆栈满时执行的操作由 STVREN（堆栈上溢复位使能）配置位的状态决定。（有关器件配置位的介绍，请参见**第 24.1 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，将 STKFUL 位置 1，并复位器件。SKFUL 位将保持置 1，而堆栈指针将被清零。

如果 STVREN 位被清零，第 31 次进栈时 STKFUL 位会被置 1，堆栈指针则加 1 变为 31。任何其他进栈都不会覆盖第 31 次进栈的值，并且 STKPTR 将保持为 31。

当堆栈弹出次数足够卸空堆栈时，下一次出栈会向 PC 返回一个零值，并将 STKUNF 位置 1，而堆栈指针则保持为 0。STKUNF 位将保持置 1，直到被软件清零或发生 POR。

注：	下溢会导致向 PC 返回一个零值，并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 SFR 的内容不受影响。

5.1.6.3 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值而不影响程序的正常执行是非常理想的。PIC18 指令集包括两条指令 PUSH 和 POP，它们允许在软件控制下对 TOS 进行操作。可以通过修改 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。先将堆栈指针加 1，并将当前 PC 值装入堆栈。

POP 指令通过将栈指针减 1 来丢弃当前的 TOS 值。然后前一个进栈值就成为 TOS 值。

寄存器 5-2： 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 = 只可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **STKFUL：**堆栈满标志位 (1)
1 = 堆栈满或上溢
0 = 堆栈未满或未上溢

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 **未实现：**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。

© 2007 Microchip Technology Inc.

5.1.6.4 堆栈满和下溢复位

通过将配置寄存器 1L 中的 STVREN 位置 1，允许在出现堆栈上溢和堆栈下溢条件时使器件复位。当 STVREN 位置 1 时，堆栈满或堆栈下溢条件会将相应的 STKFUL 或 STKUNF 位置 1，然后使器件复位。当 STVREN 位清零时，堆栈满或堆栈下溢条件会将相应的 STKFUL 或 STKUNF 位置 1，然后使器件复位。通过用户软件或上电复位使 STKFUL 或 STKUNF 位清零。

5.1.7 快速寄存器堆栈

为 STATUS、WREG 和 BSR 寄存器提供了快速寄存器堆栈具有从中断“快速返回”的功能。每个寄存器堆栈的深度仅为 1 级，并且不可读写。当处理器转入中断向量处执行指令时，此堆栈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 和 FAST 指令从中断返回，这些寄存器中的值会被装回工作寄存器。

如果同时允许了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。如果在处理低优先级中断时，发生了高优先级中断，则低优先级中断存储的堆栈寄存器值将被覆盖。在这些情况下，用户必须在低优先级中断期间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，则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。要将快速寄存器堆栈用于子程序调用，必须执行 CALLlabel,FAST 指令将 STATUS、WREG 和 BSR 寄存器的内容存入快速寄存器堆栈。在调用结束后执行 RETURN,FAST 指令，从快速寄存器堆栈中弹出并恢复这些寄存器的值。

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
```

5.1.8 程序存储器中的查找表

有些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，有两种方法可以实现查找表：

- 计算 GOTO
- 表读

5.1.8.1 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 指令可以组成一个查找表。在调用该表前，会先将查找表的偏移量装入 W 寄存器。被调用子程序的第一条指令是 ADDWF PCL 指令。接下去执行的是一条 RETLW nn 指令，它将数值 nn 返回给调用函数。

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

在这种方法中，每个指令单元只能存储一个数据字节，并且要求在返回地址堆栈还有空闲空间。

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
        .
        .
        .
```

5.1.8.2 表读

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

编程时，每个程序字可以存储 2 个字节的查找表数据。表指针寄存器 （TBLPTR）指定字节地址，而表锁存器（TABLAT）则存储从程序存储器读取的数据。一次只能从程序存储器读取一个字节。

在**第 6.1 节“表读和表写”**中将进一步讨论表读操作。

5.2 PIC18 指令周期

5.2.1 时钟机制

单片机时钟输入信号，无论来自内部或外部时钟源，都会在器件内部被 4 分频用来产生 4 个不重叠的正交时钟信号，即 Q1、Q2、Q3 和 Q4。单片机工作时，程序计数器在每个 Q1 递增，并在 Q4 期间从程序存储器取指并将指令锁存到指令寄存器中。指令的译码和执行在下一个 Q1 到 Q4 周期完成。图 5-5 所示为时钟和指令执行的流程图。

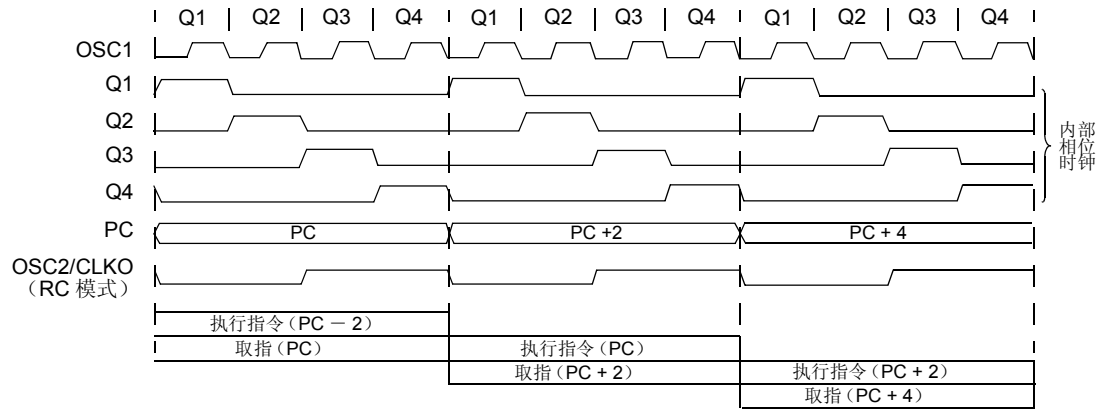
5.2.2 指令流 / 流水线

指令周期由 4 个 Q 周期组成：Q1 到 Q4。指令的取指和执行是以流水线的形式进行的，在一个指令周期进行取指，而在另一个指令周期译码并执行指令。但由于是流水线操作，因此每个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 指令），则需要两个指令周期才能完成该指令（见例 5-3）。

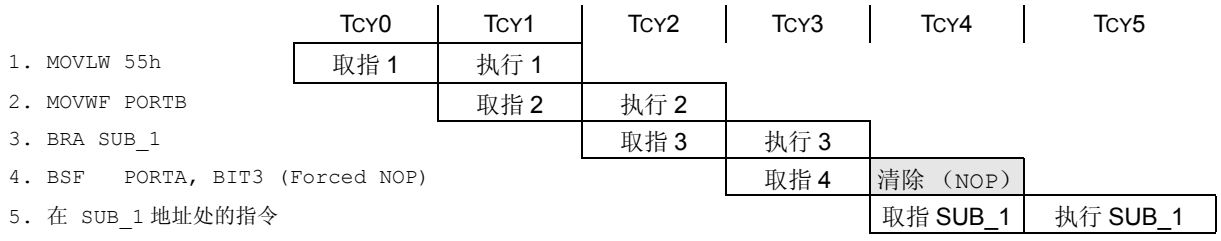
在 Q1 周期，程序计数器 （PC）递增，开始取指。

指令的执行过程如下：在 Q1 周期，将所取指令锁存到指令寄存器（Instruction Register， IR）。然后在 Q2、Q3 和 Q4 周期中进行指令的译码和执行。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 （写目标地址）。

图 5-5： 时钟 / 指令周期



例 5-3： 指令流水线流程图



除程序转移指令之外，所有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，然后执行指令，所以程序转移指令需要两个周期。

© 2007 Microchip Technology Inc.

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偶地址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 0 （见**第 5.1.5 节 “程序计数器”**）。

图 5-6 给出了指令字如何存储在程序寄存器中的示例。

CALL 和 GOTO 指令在指令中嵌入了程序存储器的绝对地址。指令总是存储为一个字长，因而指令所包含的数据为字地址。字地址会写入PC<20:1>，后者则在程序存储器中访问所需的字节地址。图 5-6 中的指令 2 给出了指令 “GOTO 0006h”在程序存储器中的编码过程。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。在转移指令中的偏移值代表单字指令数，PC 将以此作为偏移量跳转到指定的地址单元。**第 25.0 节“指令集综述”**提供了指令集的更多详情。

图 5-6： 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。在所有情况下，这些指令第二个字的 4 个最高有效位总是 1111，而其余 12 位是立即数数据，通常为一个数据存储器地址。

指令的 4 个最高有效位 1111 用于指定一个特殊的 NOP 指令。指令的正确执行顺序为：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并自行执行指令的第二个字，那么第一个字将作为 NOP 指令执行。如果双字指令跟在更改 PC 的条件指令后，就有必要执行此操作。例 5-4 说明了其执行过程。

注： 欲知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

5.3 数据存储器构成

注：	当使能 PIC18 扩展指令集时，数据存储器某些方面的操作会有所改变。如需更多信息，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。在数据存储器中，每个寄存器有 12 位地址，数据存储器可达 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。PIC18F87J11 系列器件实现了所有可用的存储区并为用户提供 3936 个字节的数据存储器。图 5-7 为器件的数据存储器构成。

数据存储器由特殊功能寄存器 （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，而 GPR 则用于在用户应用程序中存储数据和高速暂存操作。任何未用单元的读取值均为 0。

此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。本节后面的部分将讨论寻址模式。

为了确保能在一个周期内存取常用寄存器（SFR 和所选的 GPR），PIC18 器件设置了快速操作存储区。这是一个 256 字节的存储空间，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 5.3.2 节 “快速操作存储区”**提供了对快速操作 RAM 的详细说明。

5.3.1 存储区选择寄存器 （BSR）

存储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，这意味着不必要为每次读写操作提供整个地址。 PIC18 器件是使用 RAM区存储机制实现快速存取的。该机制将存储空间分成连续的 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 指令集中的大多数指令使用存储区指针，也就是存储区选择寄存器 （Bank Select Register， BSR） 。SFR 保存单元地址的 4 个最高有效位，而指令本身包括 8 个最低有效位。只使用 BSR 的低四位（BSR3:BSR0）而不是使用高 4 位，高四位的读取值始终为 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，可以将它看作是以存储区下边界为起点的偏移量。图 5-8 所示为 BSR 的值与数据存储器中存储区划分的关系。

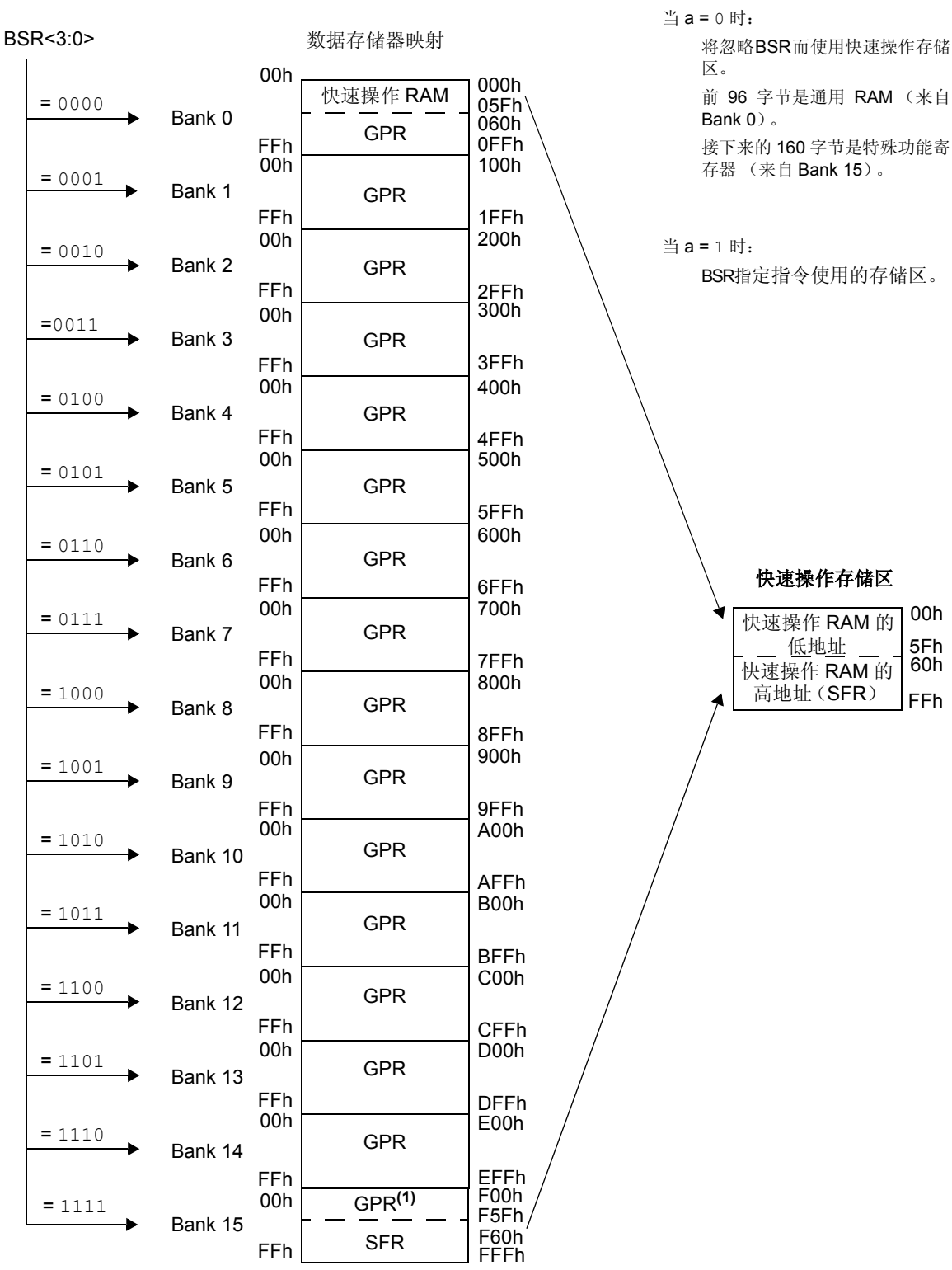
由于最多有 16 个寄存器共享同一个低位地址，用户必须非常小心以确保在执行数据读或写之前选择了正确的存储区。例如，当 BSR 为 0Fh 时将程序数据写入 8 位地址 F9h 将导致程序计数器复位。

当选择存储区时，只有实际可使用的存储区可以被读写。对未用的存储区的写入将被忽略，而读未用的存储区会返回 0。虽然是这样，状态寄存器仍然会受到影响。图 5-7 中的数据存储器映射图指出了可使用的存储区。

在 PIC18 内核指令集中，只有 MOVFF 指令完全地指定源寄存器和目标寄存器的完整 12 位地址。此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。

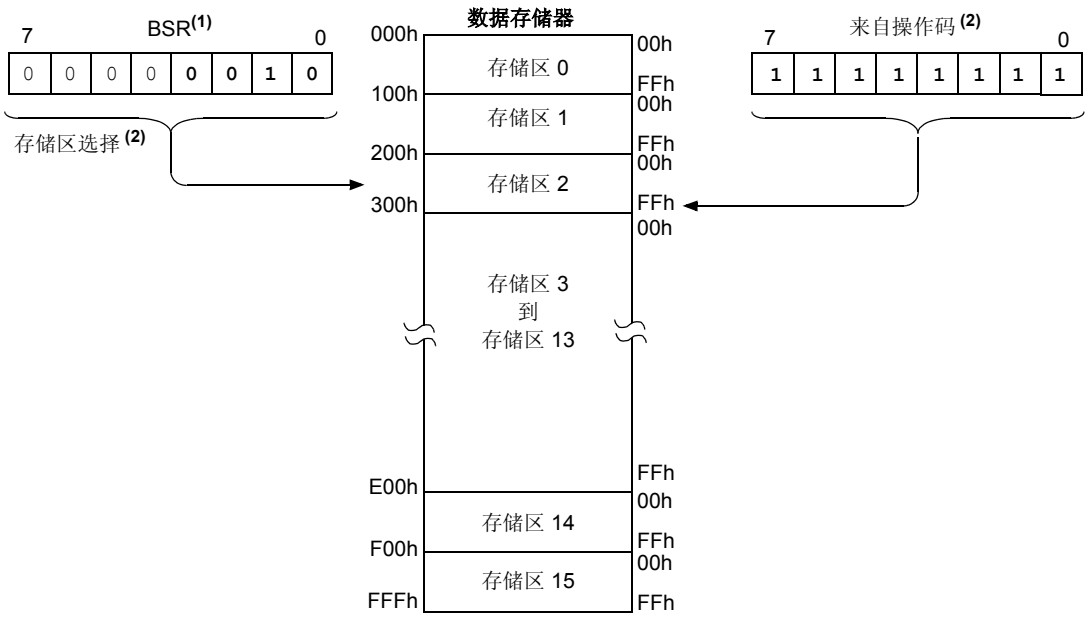
© 2007 Microchip Technology Inc.

图 5-7：　　PIC18F87J11 系列器件的数据存储器映射



注　1： F5Ah 到 F5Fh 的地址也供 SFR 使用，但不是快速操作 RAM 的一部分。用户须始终使用完整地址或载入正确的 BSR 值来访问这些寄存器。

图 5-8：　　使用存储区选择寄存器 （直接寻址）



注　1： 指令的快速操作 RAM 位强制忽略选定的存储区 （由 BSR<3:0> 选定）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。

5.3.2　快速操作存储区

使用 BSR 和指令内嵌的 8 位地址可以使用户对整个数据存储空间进行寻址，这同时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，验证和 / 或更改 BSR 可能会降低代码的执行效率。

为了连续访问大多数常用的数据存储单元，必须为数据存储器配置快速操作存储区，这使得用户无需指定 BSR 即可访问被映射的存储区。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。低半部分就是 “快速操作 RAM”，由 GPR 组成。高地址的那一半被映射为器件的 SFR。这两个区域可以在快速操作存储区中连续映射并且可以用 8 位地址进行线性寻址 （图 5-7）。

快速操作存储区供包括快速操作 RAM 位 （指令中的 “a” 参数）的 PIC18 内核指令使用。当 “a” 等于 1 时，指令使用 BSR 和包含在操作码中的 8 位地址来对数据存储器进行寻址。但是当 “a” 为 0 时，指令被强制使用快速操作存储区地址映射； BSR 的当前值被忽略。

该 “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，而无需首先更新 BSR。这意味着用户可以更有效地对 8 位地址为 60h 及以上的 SFR 进行取值和操作。地址低于 60h 的快速操作 RAM 非常适合于存储那些用户可能需要快速存取的数据值（如直接计算结果或常用程序变量）。快速操作 RAM 还可以实现更快速、代码效率更高的现场保护和变量切换。

当使能扩展指令集 （XINST 配置位 = 1）时，快速操作存储区的映射有略有不同。**第 5.6.3 节 “在立即数变址寻址模式下映射快速操作存储区”** 中对此有更详细的讨论。

5.3.3　通用寄存器

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

© 2007 Microchip Technology Inc.

5.3.4 特殊功能寄存器

特殊功能寄存器（Special Function Registers，SFR）是CPU和外设模块用来控制所需的器件操作的寄存器。这类寄存器以静态 RAM 的形式实现。SFR 起始于数据存储器的顶部（FFFh）并且向下扩展到 Bank 15 的上半部分（F80h 到 FFFh）。表 5-3、表 5-4 和表 5-5 列出了这些寄存器。

SFR 可分为两类：一类与“内核”器件功能（ALU、复位和中断）有关，另一类与外设功能有关。在相关的章节中将对复位和中断寄存器进行说明，而本节后面的部分将对ALU的状态寄存器进行说明。与外设功能部件的工作相关的寄存器在该外设的章节说明。

SFR 常分布在外设中，用来控制外设的功能。未使用的 SFR 单元不可用，它们的读取值为 0。

注：	地址 F5Ah 到 F5Fh 不属于快速操作存储区的一部分。这些寄存器必须始终通过存储区选择寄存器进行访问。

表 5-3： PIC18F87J11 系列器件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	地址	名称
FFFh	TOSU	FDFh	INDF2(1)	FBFh	ECCP1AS	F9Fh	IPR1	F7Fh	SPBRGH1	F5Fh	PMDIN2H
FFEh	TOSH	FDEh	POSTINC2(1)	FBEh	ECCP1DEL	F9Eh	PIR1	F7Eh	BAUDCON1	F5Eh	PMDIN2L
FFDh	TOSL	FDDh	POSTDEC2(1)	FBDh	CCPR1H	F9Dh	PIE1	F7Dh	SPBRGH2	F5Dh	PMEH
FFCh	STKPTR	FDCh	PREINC2(1)	FBCh	CCPR1L	F9Ch	RCSTA2	F7Ch	BAUDCON2	F5Ch	PMEL
FFBh	PCLATU	FDBh	PLUSW2(1)	FBBh	CCP1CON	F9Bh	OSCTUNE	F7Bh	TMR3H	F5Bh	PMSTATH
FFAh	PCLATH	FDAh	FSR2H	FBAh	ECCP2AS	F9Ah	TRISJ(2)	F7Ah	TMR3L	F5Ah	PMSTATL
FF9h	PCL	FD9h	FSR2L	FB9h	ECCP2DEL	F99h	TRISH(2)	F79h	T3CON	F59h	—
FF8h	TBLPTRU	FD8h	STATUS	FB8h	CCPR2H	F98h	TRISG	F78h	TMR4	F58h	—
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	CCPR2L	F97h	TRISF	F77h	PR4(3)	F57h	—
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	CCP2CON	F96h	TRISE	F76h	T4CON	F56h	—
FF5h	TABLAT	FD5h	T0CON	FB5h	ECCP3AS	F95h	TRISD	F75h	CCPR4H	F55h	—
FF4h	PRODH	FD4h	Ñ	FB4h	ECCP3DEL	F94h	TRISC	F74h	CCPR4L	F54h	—
FF3h	PRODL	FD3h	OSCCON(3)	FB3h	CCPR3H	F93h	TRISB	F73h	CCP4CON	F53h	—
FF2h	INTCON	FD2h	CM1CON	FB2h	CCPR3L	F92h	TRISA	F72h	CCPR5H	F52h	—
FF1h	INTCON2	FD1h	CM2CON	FB1h	CCP3CON	F91h	LATJ(2)	F71h	CCPR5L	F51h	—
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRG1	F90h	LATH(2)	F70h	CCP5CON	F50h	—
FEFh	INDF0(1)	FCFh	TMR1H(3)	FAFh	RCREG1	F8Fh	LATG	F6Fh	SSP2BUF	F4Fh	—
FEEh	POSTINC0(1)	FCEh	TMR1L(3)	FAEh	TXREG1	F8Eh	LATF	F6Eh	SSP2ADD	F4Eh	—
FEDh	POSTDEC0(1)	FCDh	T1CON(3)	FADh	TXSTA1	F8Dh	LATE	F6Dh	SSP2STAT	F4Dh	—
FECh	PREINC0(1)	FCCh	TMR2(3)	FACh	RCSTA1	F8Ch	LATD	F6Ch	SSP2CON1	F4Ch	—
FEBh	PLUSW0(1)	FCBh	PR2(3)	FABh	SPBRG2	F8Bh	LATC	F6Bh	SSP2CON2	F4Bh	—
FEAh	FSR0H	FCAh	T2CON	FAAh	RCREG2	F8Ah	LATB	F6Ah	CMSTAT	F4Ah	—
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	TXREG2	F89h	LATA	F69h	PMADDRH(4)	F49h	—
FE8h	WREG	FC8h	SSP1ADD	FA8h	TXSTA2	F88h	PORTJ(2)	F68h	PMADDRL(4)	F48h	—
FE7h	INDF1(1)	FC7h	SSP1STAT	FA7h	EECON2	F87h	PORTH(2)	F67h	PMDIN1H	F47h	—
FE6h	POSTINC1(1)	FC6h	SSP1CON1	FA6h	EECON1	F86h	PORTG	F66h	PMDIN1L	F46h	—
FE5h	POSTDEC1(1)	FC5h	SSP1CON2	FA5h	IPR3	F85h	PORTF	F65h	PMCONH	F45h	—
FE4h	PREINC1(1)	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	PMCONL	F44h	—
FE3h	PLUSW1(1)	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	PMMODEH	F43h	—
FE2h	FSR1H	FC2h	ADCON0(3)	FA2h	IPR2	F82h	PORTC	F62h	PMMODEL	F42h	—
FE1h	FSR1L	FC1h	ADCON1(3)	FA1h	PIR2	F81h	PORTB	F61h	PMDOUT2H	F41h	—
FE0h	BSR	FC0h	WDTCON	FA0h	PIE2	F80h	PORTA	F60h	PMDOUT2L	F40h	—

注 **1：** 这不是物理寄存器。
2： 该寄存器在 64 引脚上不可用。
3： 该寄存器与另一寄存器共用一个地址（见关于备用寄存器的表 5-4 中的描述）。
4： PMADDRH/L 和 PMDOUT1H/L 寄存器对共用一个地址。PMADDR 用在主控模式下，PMDOUT1 用在从动模式下。

5.3.4.1 共享地址 SFR

在SFR存储区的多个单元中，使用同一个地址可访问两个不同的硬件寄存器。在这些情况下，一个标准 PIC18 SFR 系列（如 OSCCON 和 T1CON 等）的“传统”寄存器与一个备用寄存器共享地址。备用寄存器与外设的增强型配置选项相关联，或者与标准 PIC18 SFR 映射图中未包括的新器件功能相关联。表 5-4 中提供了共享的寄存器地址以及与之关联的寄存器的完整列表。

通过软件方式将 WDTCON 寄存器中的 ADSHR 位置 1 来使能对备用寄存器（见寄存器 5-3）的访问权。若有需要，必须通过手动将 ADSHR 置 1 或清零来访问备用寄存器或传统寄存器。因为比特位在被改动之前保持不变，所以在写入任何共享SFR地址之前用户都应当总是先验证 ADSHR 的状态。

5.3.4.2 由工作条件决定的 SFR

在共享地址 SFR 之外，还有一些寄存器共用 SFR 空间中的相同地址，但它们不通过 ADSHR 位进行访问。相反，这些寄存器的定义和使用取决于与之关联的外设的运行模式。它们是：

- SSPxADD 和 SSPxMSK：有两个独立的硬件寄存器通过一个 SFR 地址进行访问。正在访问的寄存器由 MSSP 模块的运行模式决定。更多的细节见**第 19.4.3.4 节 “7 位地址屏蔽模式”**。
- PMADDRH/L 和 PMDOUT2H/L：在此情况下，这些已命名的缓冲区对实际上是相同的物理寄存器。PMP 模块的运行模式决定了寄存器的功能。更多详情请参见**第 11.1.2 节 “数据寄存器”**。

表 5-4： PIC18F87J11 系列器件的共享 SFR 地址

地址		名称	地址		名称	地址		名称
FD3h	(D)	OSCCON	FCDh	(D)	T1CON	FC2h	(D)	ADCON0
	(A)	REFOCON		(A)	ODCON3		(A)	ANCON1
FCFh	(D)	TMR1H	FCCh	(D)	TMR2	FC1h	(D)	ADCON1
	(A)	ODCON1		(A)	PADCFG1		(A)	ANCON0
FCEh	(D)	TMR1L	FCBh	(D)	PR2	F77h	(D)	PR4
	(A)	ODCON2		(A)	MEMCON(1)		(A)	CVRCON

图注： (D) = 默认 SFR，仅在 ADSHR = 0 时才可访问； (A) = 备用 SFR，仅在 ADSHR = 1 时才可访问。
注 1： 仅在 80 引脚器件上才有。

寄存器 5-3： WDTCON：看门狗定时器控制寄存器

R/W-0	R-x	U-0	R/W-0	U-0	U-0	U-0	U-0
REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
— n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **REGSLP**：稳压器低功耗运行使能位
位操作的更多细节参见寄存器 24-9。

bit 6 **LVDSTAT**：LVD 状态位
1 = V_{DDCORE} > 2.45V
0 = V_{DDCORE} < 2.45V

bit 5 **未实现**：读为 0

bit 4 **ADSHR**：共享地址 SFR 选择位
1 = 选择备用 SFR
0 = 选择默认（传统） SFR

bit 3-1 **未实现**：读为 0

bit 0 **SWDTEN**：软件控制看门狗定时器使能位
位操作的更多细节参见寄存器 24-9。

© 2007 Microchip Technology Inc.

表 5-5： 寄存器汇总 （PIC18F87J11 系列）

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	55, 65
TOSH	栈顶高字节 （TOS<15:8>）								0000 0000	55, 65
TOSL	栈顶低字节 （TOS<7:0>）								0000 0000	55, 65
STKPTR	STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0	00-0 0000	55, 66
PCLATU	—	—	bit 21(1)	PC<20:16> 的保持寄存器					---0 0000	55, 65
PCLATH	PC<15:8> 的保持寄存器								0000 0000	55, 65
PCL	PC 低位字节 （PC<7:0>）								0000 0000	55, 65
TBLPTRU	—	—	bit 21	程序存储器表指针最高字节 （TBLPTR<20:16>）					--00 0000	55, 96
TBLPTRH	程序存储器表指针高字节 （TBLPTR<15:8>）								0000 0000	55, 96
TBLPTRL	程序存储器表指针低字节 （TBLPTR<7:0>）								0000 0000	55, 96
TABLAT	程序存储器表锁存器								0000 0000	55, 96
PRODH	乘积寄存器高字节								xxxx xxxx	55, 109
PRODL	乘积寄存器低字节								xxxx xxxx	55, 109
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	55, 113
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	55, 113
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	55, 113
INDF0	使用 FSR0 的内容寻址数据存储器——FSR0 的值不变 （非物理寄存器）								N/A	55, 82
POSTINC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后增 （非物理寄存器）								N/A	55, 83
POSTDEC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后减 （非物理寄存器）								N/A	55, 83
PREINC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后减 （非物理寄存器）								N/A	55, 83
PLUSW0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后减 （非物理寄存器）， FSR0 的偏移量由 W 寄存器提供								N/A	55, 83
FSR0H	—	—	—	—	间接数据存储器地址指针 0 的高字节				---- xxxx	55, 82
FSR0L	间接数据存储器地址指针 0 的低字节								xxxx xxxx	55, 82
WREG	工作寄存器								xxxx xxxx	55, 67
INDF1	使用 FSR1 的内容寻址数据存储器——FSR1 的值不变 （非物理寄存器）								N/A	55, 82
POSTINC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值后增 （非物理寄存器）								N/A	55, 83
POSTDEC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值后减 （非物理寄存器）								N/A	55, 83
PREINC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值前增 （非物理寄存器）								N/A	55, 83
PLUSW1	使用 FSR1 的内容寻址数据存储器——FSR1 的值后减 （非物理寄存器）， FSR1 的偏移量由 W 寄存器提供								N/A	55, 83
FSR1H	—	—	—	—	间接数据存储器地址指针 1 的高字节				---- xxxx	55, 82
FSR1L	间接数据存储器地址指针 1 的低字节								xxxx xxxx	55, 82
BSR	—	—	—	—	存储区选择寄存器 （BSR）				---- 0000	55, 70
INDF2	使用 FSR2 的内容寻址数据存储器——FSR2 的值不变 （非物理寄存器）								N/A	56, 82
POSTINC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值后增 （非物理寄存器）								N/A	56, 83
POSTDEC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值后减 （非物理寄存器）								N/A	56, 83
PREINC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值前增 （非物理寄存器）								N/A	56, 83
PLUSW2	使用 FSR2 的内容寻址数据存储器——FSR2 的值后减 （非物理寄存器）， FSR2 的偏移量由 W 寄存器提供								N/A	56, 83

图注： x = 未知， u = 不变， - = 未实现， q = 取值视情况而定。**粗体**表示共享访问 SFR。

注 1： PC 的第 21 位仅在串行编程模式下可用。

2： 当 WDTCON<4> = 0 时，此地址的默认 （传统） SFR 有效。

3： 配置 SFR 在此地址处与默认 SFR 重叠，仅在 WDTCON<4> = 1 时才可访问配置 SFR。

4： 当双速启动被使能时复位值为 0，被禁止时复位值为 1。

5： 仅当 SSPxCON2<3:0> = 1001 时， SSPxMSK 寄存器才可用。

6： 当 MSSP 模块运行在 I^2C™ 从动模式下时，替换这些位的名称和定义。详情见**第 19.4.3.2 节 “地址屏蔽模式”**

7： 这些位和 / 或寄存器仅在 80 引脚器件上可用；在其他器件上，它们未实现且读为 0。复位值如 80 引脚器件所示。

8： 仅当选定了振荡器模式 （FOSC2 配置位 = 0）时，这些位才可用；在其他情况下，它们未实现。

9： PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L 寄存器对共享物理寄存器及地址，但根据模块的运行模式不同，其功能也会有所不同。更多信息，请参见**第 11.1.2 节 “数据寄存器”**。

表 5-5： 寄存器汇总 （PIC18F87J11 系列） （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）
FSR2H	—	—	—	—	间接数据存储器地址指针 2 的高字节				---- xxxx	56, 82
FSR2L	间接数据存储器地址指针 2 的低字节								xxxx xxxx	56, 82
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	56, 80
TMR0H	Timer0 寄存器高字节								0000 0000	56, 183
TMR0L	Timer0 寄存器低字节								xxxx xxxx	56, 183
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	56, 182
OSCCON(2)/	IDLEN	IRCF2	IRCF1	IRCF0	OSTS(4)	—	SCS1	SCS0	0110 q100	56, 32
REFOCON(3)	ROON	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	0-00 0000	56, 39
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111	56, 302
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111	56, 302
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0-11 1100	54, 56, 125
TMR1H(2)/	Timer1 寄存器高字节								xxxx xxxx	56, 186
ODCON1(3)	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	---0 0000	56, 129
TMR1L(2)/	Timer1 寄存器低字节								xxxx xxxx	56, 186
ODCON2(3)	—	—	—	—	—	—	U2OD	U1OD	---- --00	56, 129
T1CON (2)/	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	56, 186
ODCON3(3)	—	—	—	—	—	—	SPI2OD	SPI1OD	---- --00	56, 129
TMR2(2)/	Timer2 寄存器								0000 0000	56, 191
PADCFG1(3)	—	—	—	—	—	—	—	PMPTTL	---- ---0	56, 130
PR2(2)/	Timer2 周期寄存器								1111 1111	56, 191
MEMCON(3,7)	EDBIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	56, 98
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	56, 191
SSP1BUF	MSSP1 接收缓冲器 / 发送寄存器								xxxx xxxx	56, 224, 233
SSP1ADD/	MSSP1 地址寄存器 （I²C™ 从动模式）， MSSP1 波特率重载寄存器 （I²C 主控模式）								0000 0000	56, 233
SSP1MSK(5)	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	0000 0000	56, 240
SSP1STAT	SMP	CKE	$\overline{D/A}$	P	S	$\overline{R/W}$	UA	BF	0000 0000	56, 224, 234
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	56, 225, 235
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN/	SEN	0000 0000	56, 236, 268
	GCEN	ACKSTAT	ADMSK5(6)	ADMSK4(6)	ADMSK3(6)	ADMSK2(6)	ADMSK1(6)	SEN		
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	56, 291
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	57, 291
ADCON0(2)/	VCFG1	VCFG0	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	0000 0000	57, 291
ANCON1(3)	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	0000 0000	57, 293
ADCON1(2)/	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0000 0000	57, 292
ANCON0(3)	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	00-0 0000	57, 293
WDTCON	REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN	0x-0 ---0	57, 321

图注： x = 未知， u = 不变， - = 未实现， q = 取值视情况而定。**粗体**表示共享访问 SFR。

注 1： PC 的第 21 位仅在串行编程模式下可用。

2： 当 WDTCON<4> = 0 时，此地址的默认 （传统） SFR 有效。

3： 配置 SFR 在此地址处与默认 SFR 重叠，仅在 WDTCON<4> = 1 时才可访问配置 SFR。

4： 当双速启动被使能时复位值为 0，被禁止时复位值为 1。

5： 仅当 SSPxCON2<3:0> = 1001 时， SSPxMSK 寄存器才可用。

6： 当 MSSP 模块运行在 I²C™ 从动模式下时，替换这些位的名称和定义。详情见**第 19.4.3.2 节 “地址屏蔽模式”**

7： 这些位和 / 或寄存器仅在 80 引脚器件上可用；在其他器件上，它们未实现且读为 0。复位值如 80 引脚器件所示。

8： 仅当选定了振荡器模式 （FOSC2 配置位 = 0） 时，这些位才可用；在其他情况下，它们未实现。

9： PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L 寄存器对共享物理寄存器及地址，但根据模块的运行模式不同，其功能也会有所不同。更多信息，请参见**第 11.1.2 节 “数据寄存器”**。

© 2007 Microchip Technology Inc.

表 5-5：　寄存器汇总 （PIC18F87J11 系列）（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）
ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	57, 222
ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000	57, 222
CCPR1H	捕捉 / 比较 /PWM 寄存器 1 的高字节								xxxx xxxx	57, 222
CCPR1L	捕捉 / 比较 /PWM 寄存器 1 的低字节								xxxx xxxx	57, 222
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	57, 222
ECCP2AS	ECCP2ASE	ECCP2AS2	ECCP2AS1	ECCP2AS0	PSS2AC1	PSS2AC0	PSS2BD1	PSS2BD0	0000 0000	57, 222
ECCP2DEL	P2RSEN	P2DC6	P2DC5	P2DC4	P2DC3	P2DC2	P2DC1	P2DC0	0000 0000	57, 222
CCPR2H	捕捉 / 比较 /PWM 寄存器 2 的高字节								xxxx xxxx	57, 222
CCPR2L	捕捉 / 比较 /PWM 寄存器 2 的低字节								xxxx xxxx	57, 222
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000	57, 222
ECCP3AS	ECCP3ASE	ECCP3AS2	ECCP3AS1	ECCP3AS0	PSS3AC1	PSS3AC0	PSS3BD1	PSS3BD0	0000 0000	57, 222
ECCP3DEL	P3RSEN	P3DC6	P3DC5	P3DC4	P3DC3	P3DC2	P3DC1	P3DC0	0000 0000	57, 222
CCPR3H	捕捉 / 比较 /PWM 寄存器 1 的高字节								xxxx xxxx	57, 222
CCPR3L	捕捉 / 比较 /PWM 寄存器 1 的低字节								xxxx xxxx	57, 222
CCP3CON	P3M1	P3M0	DC3B1	DC3B0	CCP3M3	CCP3M2	CCP3M1	CCP3M0	0000 0000	57, 222
SPBRG1	EUSART1 波特率发生器寄存器的低字节								0000 0000	57, 273
RCREG1	EUSART1 接收寄存器								0000 0000	57, 281, 282
TXREG1	EUSART1 发送寄存器								xxxx xxxx	57, 279, 280
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	57, 279
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	57, 281
SPBRG2	EUSART2 波特率发生器寄存器的低字节								0000 0000	57, 273
RCREG2	EUSART2 接收寄存器								0000 0000	57, 281, 282
TXREG2	EUSART2 发送寄存器								0000 0000	57, 279, 280
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	57, 279
EECON2	程序存储器控制寄存器 2（非物理寄存器）								---- ----	57, 88
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	--00 x00-	57, 88
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	1111 1111	57, 122
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	0000 0000	57, 116
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	0000 0000	58, 119
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	111- 1111	58, 122
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	000- 0000	58, 116
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	000- 0000	58, 119
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	1111 1111	58, 122
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	58, 116
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	58, 119
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	58, 281
OSCTUNE	INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000 0000	58, 33

图注： x = 未知， u = 不变， - = 未实现， q = 取值视情况而定。**粗体**表示共享访问 SFR。

注 1： PC 的第 21 位仅在串行编程模式下可用。

2： 当 WDTCON<4> = 0 时，此地址的默认 （传统） SFR 有效。

3： 配置 SFR 在此地址处与默认 SFR 重叠，仅在 WDTCON<4> = 1 时才可访问配置 SFR。

4： 当双速启动被使能时复位值为 0，被禁止时复位值为 1。

5： 仅当 SSPxCON2<3:0> = 1001 时， SSPxMSK 寄存器才可用。

6： 当 MSSP 模块运行在 I^2C™ 从动模式下时，替换这些位的名称和定义。详情见**第 19.4.3.2 节 “地址屏蔽模式”**

7： 这些位和 / 或寄存器仅在 80 引脚器件上可用；在其他器件上，它们未实现且读为 0。复位值如 80 引脚器件所示。

8： 仅当选定了振荡器模式 （FOSC2 配置位 = 0）时，这些位才可用；在其他情况下，它们未实现。

9： PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L 寄存器对共享物理寄存器及地址，但根据模块的运行模式不同，其功能也会有所不同。更多信息，请参见**第 11.1.2 节 “数据寄存器”**。

表 5-5： 寄存器汇总 （PIC18F87J11 系列）（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）
TRISJ(7)	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	1111 1111	58, 155
TRISH(7)	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	1111 1111	58, 153
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	---1 1111	58, 150
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	1111 111-	58, 147
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	1111 1111	58, 144
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	58, 141
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	58, 138
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	58, 135
TRISA	TRISA7(8)	TRISA6(8)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	58, 132
LATJ(7)	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	xxxx xxxx	58, 155
LATH(7)	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	xxxx xxxx	58, 153
LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	---x xxxx	58, 150
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	xxxx xxx-	58, 147
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx xxxx	58, 144
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx	58, 141
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx	58, 138
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	58, 135
LATA	LATA7(8)	LATA6(8)	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx xxxx	58, 132
PORTJ(7)	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	xxxx xxxx	59, 155
PORTH(7)	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	0000 xxxx	59, 153
PORTG	RDPU	REPU	RJPU(7)	RG4	RG3	RG2	RG1	RG0	000x xxxx	59, 150
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	x000 000-	59, 147
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx xxxx	59, 144
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	59, 141
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	59, 138
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	59, 135
PORTA	RA7(8)	RA6(8)	RA5	RA4	RA3	RA2	RA1	RA0	000x 0000	59, 132
SPBRGH1	EUSART1 波特率发生器寄存器的高字节								0000 0000	59, 273
BAUDCON1	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	0100 0-00	59, 273
SPBRGH2	EUSART2 波特率发生器寄存器的高字节								0000 0000	59, 273
BAUDCON2	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	0100 0-00	59, 273
TMR3H	Timer3 寄存器高字节								xxxx xxxx	59, 198
TMR3L	Timer3 寄存器低字节								xxxx xxxx	59, 198
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	0000 0000	59, 198
TMR4	Timer4 寄存器								0000 0000	59, 197
PR4(2)/	Timer4 周期寄存器								1111 1111	59, 198
CVRCON(3)	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	59, 310
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	-000 0000	59, 197

图注： x = 未知， u = 不变， - = 未实现， q = 取值视情况而定。**粗体**表示共享访问 SFR。

注 1： PC 的第 21 位仅在串行编程模式下可用。

2： 当 WDTCON<4> = 0 时，此地址的默认 （传统） SFR 有效。

3： 配置 SFR 在此地址处与默认 SFR 重叠，仅在 WDTCON<4> = 1 时才可访问配置 SFR。

4： 当双速启动被使能时复位值为 0，被禁止时复位值为 1。

5： 仅当 SSPxCON2<3:0> = 1001 时， SSPxMSK 寄存器才可用。

6： 当 MSSP 模块运行在 I²C™ 从动模式下时，替换这些位的名称和定义。详情见**第 19.4.3.2 节 “地址屏蔽模式”**

7： 这些位和 / 或寄存器仅在 80 引脚器件上可用；在其他器件上，它们未实现且读为 0。复位值如 80 引脚器件所示。

8： 仅当选定了振荡器模式 （FOSC2 配置位 = 0）时，这些位才可用；在其他情况下，它们未实现。

9： PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L 寄存器对共享物理寄存器及地址，但根据模块的运行模式不同，其功能也会有所不同。更多信息，请参见**第 11.1.2 节 “数据寄存器”**。

© 2007 Microchip Technology Inc.

表 5-5： 寄存器汇总 （PIC18F87J11 系列）（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）
CCPR4H	捕捉 / 比较 /PWM 寄存器 4 的高字节								xxxx xxxx	59, 200
CCPR4L	捕捉 / 比较 /PWM 寄存器 4 的低字节								xxxx xxxx	59, 200
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	--00 0000	59, 200
CCPR5H	捕捉 / 比较 /PWM 寄存器 5 的高字节								xxxx xxxx	59, 200
CCPR5L	捕捉 / 比较 /PWM 寄存器 5 的低字节								xxxx xxxx	59, 200
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	--00 0000	59, 200
SSP2BUF	MSSP2 接收缓冲器 / 发送寄存器								xxxx xxxx	59, 224, 233
SSP2ADD/	MSSP2 地址寄存器 （I²C™ 从动模式）， MSSP2 波特率重载寄存器 （I²C 主控模式）								0000 0000	59, 233
SSP2MSK(5)	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	0000 0000	59, 240
SSP2STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	59, 224, 234
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	59, 225, 235
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN/	SEN	0000 0000	59, 236, 268
	GCEN	ACKSTAT	ADMSK5(6)	ADMSK4(6)	ADMSK3(6)	ADMSK2(6)	ADMSK1(6)	SEN		
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	---- --11	59, 303
PMADDRH /	CS2	CS1	并行主端口地址高字节						0000 0000	60, 164
PMDOUT1H(9)	并行端口输出数据高字节 （Buffer 1）								0000 0000	60, 167
PMADDRL/	并行主端口地址低字节								0000 0000	60, 164
PMDOUT1L(9)	并行端口输出数据低字节 （Buffer 0）								0000 0000	60, 164
PMDIN1H	并行端口输入数据高字节 （Buffer 1）								0000 0000	60, 164
PMDIN1L	并行端口输入数据低字节 （Buffer 0）								0000 0000	60, 164
PMCONH	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	0-00 0000	60, 158
PMCONL	CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP	0000 0000	60, 159
PMMODEH	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	0000 0000	60, 160
PMMODEL	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	0000 0000	60, 161
PMDOUT2H	并行端口输出数据高字节 （Buffer 3）								0000 0000	60, 164
PMDOUT2L	并行端口输出数据低字节 （Buffer 2）								0000 0000	60, 164
PMDIN2H	并行端口输入数据高字节 （Buffer 3）								0000 0000	60, 164
PMDIN2L	并行端口输入数据低字节 （Buffer 2）								0000 0000	60, 164
PMEH	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	0000 0000	60, 161
PMEL	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000 0000	60, 162
PMSTATH	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	00-- 0000	60, 162
PMSTATL	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	10-- 1111	60, 163

图注： x = 未知， u = 不变， - = 未实现， q = 取值视情况而定。**粗体**表示共享访问 SFR。

注 1： PC 的第 21 位仅在串行编程模式下可用。

2： 当 WDTCON<4> = 0 时，此地址的默认 （传统） SFR 有效。

3： 配置 SFR 在此地址处与默认 SFR 重叠，仅在 WDTCON<4> = 1 时才可访问配置 SFR。

4： 当双速启动被使能时复位值为 0，被禁止时复位值为 1。

5： 仅当 SSPxCON2<3:0> = 1001 时， SSPxMSK 寄存器才可用。

6： 当 MSSP 模块运行在 I²C™ 从动模式下时，替换这些位的名称和定义。详情见**第 19.4.3.2 节 “地址屏蔽模式”**

7： 这些位和 / 或寄存器仅在 80 引脚器件上可用；在其他器件上，它们未实现且读为 0。复位值如 80 引脚器件所示。

8： 仅当选定了振荡器模式 （FOSC2 配置位 = 0）时，这些位才可用；在其他情况下，它们未实现。

9： PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L 寄存器对共享物理寄存器及地址，但根据模块的运行模式不同，其功能也会有所不同。更多信息，请参见**第 11.1.2 节 “数据寄存器”**。

5.3.5 状态寄存器

状态寄存器，如寄存器 5-4 所示，包含 ALU 的算术运算状态。状态寄存器和任何其他寄存器一样，可以作为任何指令的操作数。如果一条影响 Z、DC、C、OV 或 N 位的指令的目标寄存器是状态寄存器，则会禁止对这 5 位进行写操作。

根据器件逻辑，这些位会被置 1 或清零。所以当执行一条把 STATUS 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 会将 Z 位置 1，而保留其余位不变。然后状态寄存器将读出 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 STATUS 寄存器中的 Z、C、DC、OV 或 N 位。

关于其他不影响任何状态位的指令，请参见表 25-2 和表 25-3 中的指令集综述。

注： 在减法运算中，C 和 DC 位分别作为借位和半借位标志位。

寄存器 5-4： 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 = 未实现位，读为 0	
— n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 **未实现：** 读为 0

bit 4 **N：** 负标志位
此位用于有符号数的算术运算 （通过 2 进制补码方式进行）。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：** 溢出标志位
此位用于有符号数的算术运算 （通过 2 进制补码方式进行）。表明运算结果超出了 7 位二进制数的范围，溢出导致了符号位 （bit 7）发生改变。
1 = 有符号数的算术运算发生溢出
0 = 没有发生溢出

bit 2 **Z：** 全零标志位
1 = 算术运算或逻辑运算结果为零
0 = 算术运算或逻辑运算结果非零

bit 1 **DC：** 半进位 / 借位 标志位 (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 标志位 (2)
用于 ADDWF、 ADDLW、SUBLW 和 SUBWF 指令：
1 = 结果的最高有效位发生了进位
0 = 结果的最高有效位未发生进位

注 1： 对于半借位，极性是相反的。减法指令通过加上第二个操作数的 2 进制补码来实现。对于移位指令 （RRF 和 RLF），此位的值来自源寄存器的第 4 位或第 3 位。

2： 对于借位，极性是相反的。减法指令通过加上第二个操作数的 2 进制补码来实现。对于移位指令 （RRF 和 RLF），此位的值来自源寄存器的高位或低位。

© 2007 Microchip Technology Inc.

5.4 数据寻址模式

注： 当使能 PIC18 扩展指令集时， PIC18 内核指令集中的某些指令的执行会改变。更多信息，请参见**第 5.6 节“数据存储器和扩展指令集”**。

虽然只能用一种方法（即通过程序计数器）对程序存储器进行寻址，但是可以用多种方法来对数据存储空间进行寻址。对于大多数指令，寻址模式是固定的。其他指令最多可以使用 3 种模式，这取决于使用哪些操作数以及是否使能了扩展指令集。

寻址模式有：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

当使能扩展指令集（XINST 配置位 = 1）时，可以使用另一种寻址模式，即使用立即数作为偏移量进行变址寻址的模式。**第 5.6.1节“用立即数偏移量进行变址寻址”**中对其操作有更详细的讨论。

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。这种寻址模式就是固有寻址。例如 SLEEP、RESET 和 DAW 指令。

其他指令的工作方式与此类似但需要操作码中有直接的参数。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，它们分别将立即数值移入 W 寄存器或从中移出。其他的立即数寻址指令，例如 CALL 和 GOTO，包括一个 20 位的程序存储器地址。

5.4.2 直接寻址

直接寻址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，面向位和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 的某个存储区中的寄存器地址（**第 5.3.3 节“通用寄存器”**）或快速操作存储区（**第 5.3.2 节“快速操作存储区”**）中作为指令的数据源的单元地址。

快速操作 RAM 位“a”决定地址的解析方式。当“a”为 1 时， BSR（**第 5.3.1 节“存储区选择寄存器（BSR）”**）的内容和地址一起用于确定寄存器完整的 12 位地址。当“a”为 0 时，此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 寻址有时候也被称为直接强制寻址模式。

有几个指令，比如 MOVFF，在操作码中包含完整的 12 位地址。在这些情况下， BSR 完全被忽略。

操作结果的目标寄存器由目标位“d”确定。当“d”为 1 时，结果被存储到源寄存器并覆盖它原来的内容。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不需要在指令中给出一个固定的地址。这是通过使用文件选择寄存器（File Select Register， FSR）指向被读取或写入的单元实现的。由于FSR本身作为特殊功能寄存器位于 RAM 中，所以也可以在程序控制下直接对它们进行操作。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）对寄存器进行间接寻址。这可以实现使用循环的高效代码，如例 5-5 中清零整个 RAM 存储区的操作。它还允许用户在数据存储器中执行变址寻址和其他针对程序存储器堆栈指针的操作。

例 5-5： 使用间接寻址将 RAM（BANK 1）清零的方法

```
        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
```

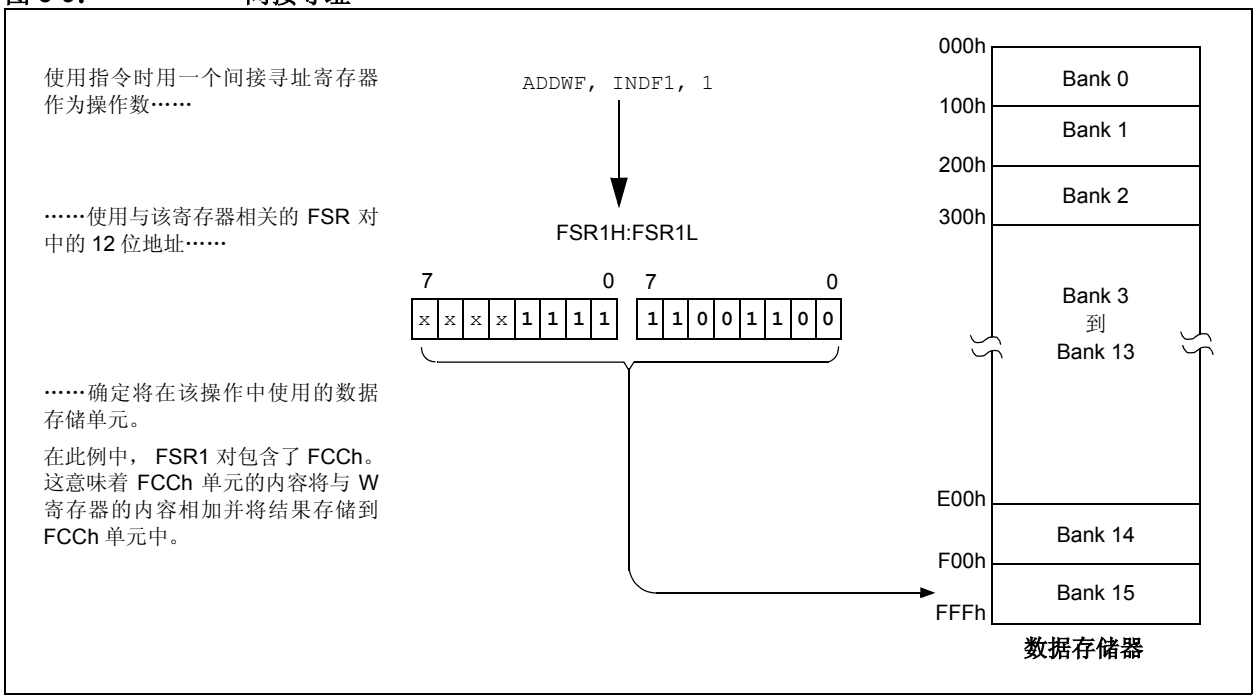
5.4.3.1 FSR 寄存器和 INDF 操作数

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 寄存器的高四位未使用，所以每对 FSR 只保存一个 12 位二进制数，从而可以线性寻址整个数据存储空间。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（从 INDF0 到 INDF2）完成的。这些操作数可被视作“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相应的 FSR 寄存器对。例如，读 INDF1 就是读 FSR1H:FSR1L 指向的地址单元中的数据。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 操作数只是使用指针的一种较方便的方法。

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。因此 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 5-9： 间接寻址



 © 2007 Microchip Technology Inc.

5.4.3.2 FSR 寄存器与 POSTINC、POSTDEC、 PREINC 和 PLUSW

除了 INDF 操作数外，每个 FSR 寄存器对还有另外 4 个间接操作数。与 INDF 相同，它们都是“虚拟”寄存器，不能直接读写。访问这些寄存器实际上是访问相关的一对 FSR 寄存器，并对其存储的值执行特定的操作。这些寄存器是：

- POSTDEC：访问 FSR 值，然后在指令操作后自动将其减 1
- POSTINC：访问 FSR 值，然后在指令操作后自动将其加 1
- PREINC：将 FSR 值加 1，然后在指令操作中使用它
- PLUSW：让 W 寄存器的有符号的值 （范围从 -127 到 128）与 FSR 的值相加，然后在指令操作中使用这个新值

在此上下文中，访问 INDF 寄存器将使用 FSR 寄存器中的值而不更改它们。类似地，访问 PLUSW 寄存器会用 W 寄存器中的值作为 FSR 的偏移值；实际上在该操作中W寄存器中的值也不作更改。访问其他的虚拟寄存器将更改 FSR 寄存器的值。

用 POSTDEC、 POSTINC 和 PREINC 对 FSR 进行操作将影响整个寄存器对，例如：FSRnL 寄存器从 FFh 计满返回到 00h，会向 FSRnH 寄存器进位。另外，这些操作的结果将不会更改状态寄存器中的任何标志位的值 （如 Z、 N 和 OV 等）。

可以使用 PLUSW 寄存器在数据存储器空间中实现一种变址寻址形式。通过设置 W 寄存器中的值，用户可以访问指针地址加上固定偏移值后的地址。 在有些应用中，这可以用来实现一些强大的程序控制结构，诸如数据存储器内部的软件堆栈。

5.4.3.3 对 FSR 的间接寻址

将其他FSR或虚拟寄存器作为目标寄存器的间接寻址操作是一类特殊情况。例如，使用一个 FSR 指向某个虚拟寄存器将会导致操作失败。举一个具体的例子，假定 FSR0H:FSR0L 包含了 FE7h，它是 INDF1 的地址。尝试读取 INDF1 的值时如果使用 INDF0 作为操作数，将返回 00h。尝试写 INDF1 时如果使用 INDF0 作为操作数，将导致执行一条 NOP 指令。

另一方面，使用虚拟寄存器写FSR对的结果可能与预期的不同。在有些情况下，值将被写入 FSR 对但是不会对该值进行加减。因此，对 INDF2 或 POSTDEC2 进行写操作会将同一个值写入 FSR2H:FSR2L。

因为 FSR 是映射到 SFR 空间的物理寄存器，所以可以通过所有的直接操作对它们进行操作。当使用这些寄存器时，用户应该小心处理，尤其是当它们的代码使用间接寻址时。

同理，在所有其他SFR上通常也允许使用间接寻址的操作。用户在进行此类操作时应该特别小心，以免不慎更改可能影响器件操作的设置。

5.5 程序存储器和扩展指令集

使用扩展指令集不会影响程序存储器的操作。

使能扩展指令集会向现有的PIC18指令集添加5个新的双字命令： ADDFSR、 CALLW、 MOVSF、 MOVSS 和 SUBFSR。**第 5.2.4 节“双字指令”**中说明这些指令的执行方式。

5.6 数据存储器和扩展指令集

使能 PIC18 扩展的指令集 （XINST 配置位 = 1）明显的更改了数据存储器及其寻址的方式。特别是许多 PIC18 内核指令使用快速操作存储区的方式会有所不同。这是由于扩展的指令集引入了对数据存储空间的新的寻址模式。

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 用立即数偏移量进行变址寻址

使能 PIC18 扩展的指令集将更改使用 FSR2 寄存器对在快速操作 RAM 内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（即面向位和字节的指令）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。

当使用扩展指令集时，这种寻址模式有以下要求：

- 强制使用快速操作存储区 （a = 0）；并且
- 指针地址参数小于或等于 5Fh。

在这些条件下，指令的指针地址不被认为是地址的低字节 （与直接寻址中的 BSR 一起使用）或快速操作存储区中的 8 位地址。相反，该值被解析为由 FSR2 指定的地址指针的偏移量。该偏移量与 FSR2 的内容相加以获得操作的目标地址。

5.6.2 受立即数变址寻址模式影响的指令

任何使用直接寻址的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。只能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不使用快速操作存储区（快速操作 RAM 位为 1）或包含 60h 以上的地址，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 5-10 给出了当使能了扩展的指令集时，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面向字节或位的指令的用户，应该注意此模式下汇编语法的改变。**第 25.2.1 节 “扩展指令的语法”**中对此有更详细的讨论。

© 2007 Microchip Technology Inc.

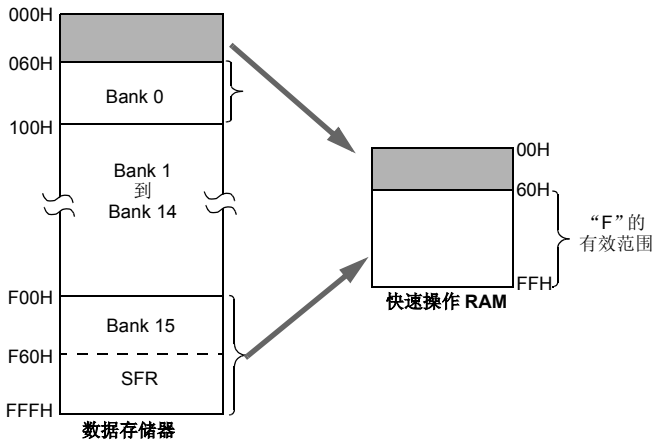
图 5-10：　　针对位和针对字节的指令的寻址方式对比 （使能了扩展指令集）

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 和 FFFh 之间的单元地址。这实际上是从 060h 到 FFFh （Bank 15）的数据存储单元。

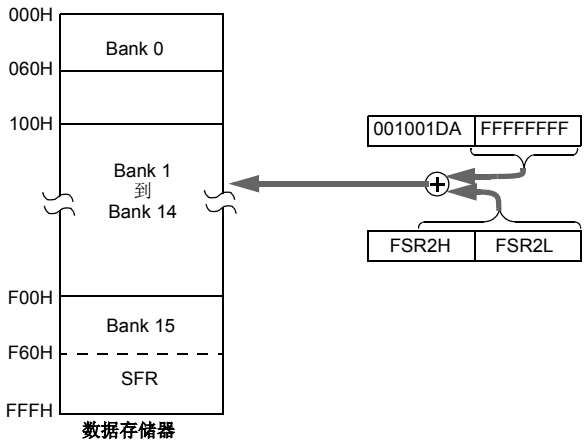
不可用此模式寻址地址低于 060h 的单元。



当 a = 1 且 f ≤ 5Fh 时：

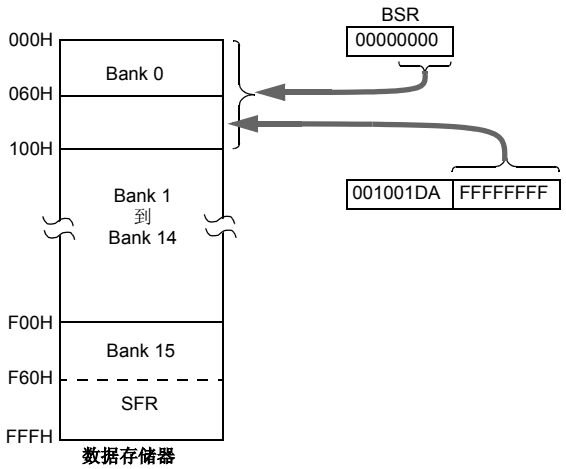
该指令以立即数变址寻址模式执行。“f”被解析为 FSR2 中地址值的偏移量。这两个值相加以获得指令的目标寄存器的地址。此地址可以在数据存储器空间的任何地方。

注意在此模式下，正确的语句是：
`ADDWF [k], d`
其中“k”与“f”相同。



当 a = 1（f 的所有值）：

指令以直接模式执行 （也被称为直接长地址寻址模式）。“f”被解析为数据存储空间的 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空间的任何位置。



5.6.3 在立即数变址寻址模式下映射快速操作存储区

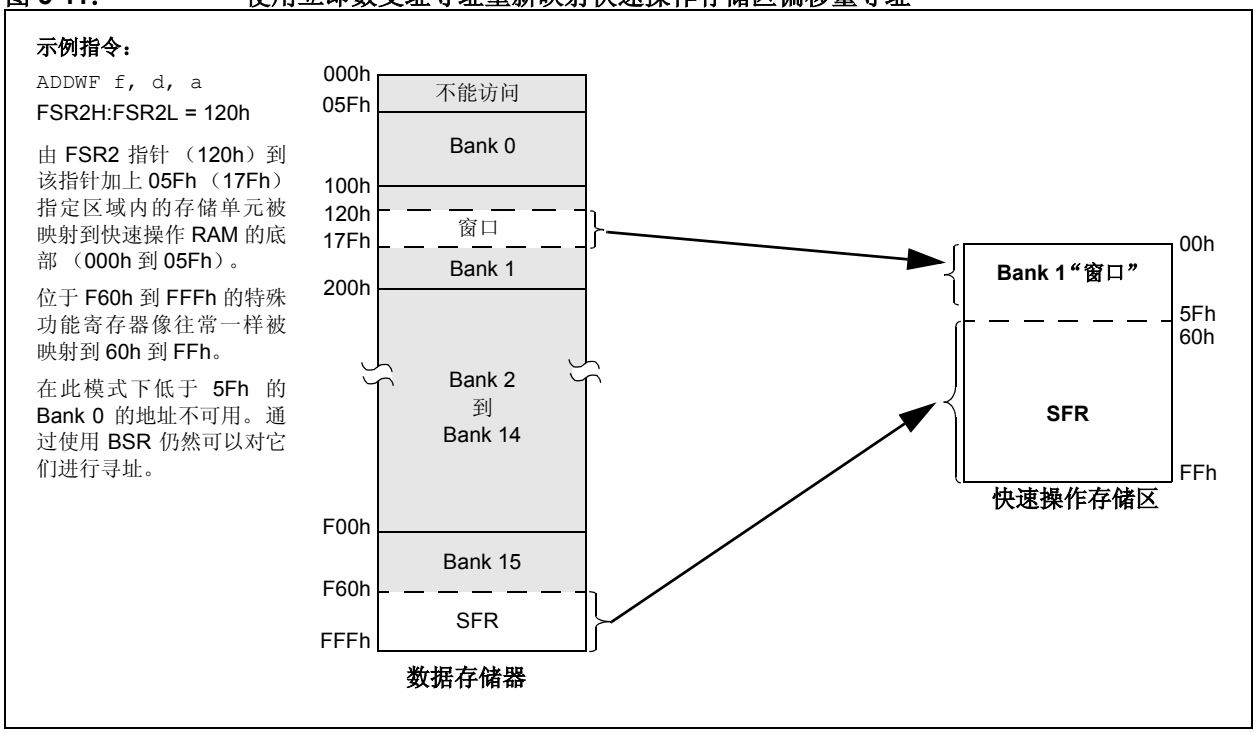
使用立即数变址寻址模式能改变快速操作 RAM 前 96 个单元地址（00h 到 5Fh）的映射方式。此模式映射 Bank 0 的内容和由用户定义的可以位于数据存储器空间中任何地方的“窗口”内容，而不是仅仅包含 Bank 0 底部的内容。FSR2 的值定义映射到窗口的地址的下边界，而上边界则由 FSR2 加 95 （5Fh）决定。地址为 5Fh 以上的快速操作 RAM 的映射方法如前所述（见**第 5.3.2 节“快速操作存储区”**）。图 5-11 中给出了在这种寻址模式下重新映射快速操作存储区的示例。

快速操作存储区的重新映射*仅*适用于使用立即数变址寻址模式的操作。使用 BSR（快速操作 RAM 位为“1”）的操作将像以前一样继续使用直接寻址。任何明确使用间接文件操作数（包括 FR2）进行的间接或变址操作都将继续以标准的间接寻址模式进行操作。任何使用快速操作存储区，但包括大于 05Fh 的寄存器地址的指令仍将使用直接寻址和常规的快速操作存储区映射。

5.6.4 立即数变址寻址模式中的 BSR 偏移模式

虽然当扩展指令集使能时会重新映射快速操作存储区，但 BSR 的操作将保持不变。直接寻址，使用 BSR 来选择数据存储区，操作方式与以前描述的相同。

图 5-11： 使用立即数变址寻址重新映射快速操作存储区偏移量寻址



 © 2007 Microchip Technology Inc.

6.0 闪存程序存储器

在整个 VDD 范围内，闪存程序存储器在正常工作状态下都是可读写并可擦除的。

对程序存储器的读操作按照每次 1 个字节来执行。对程序存储器的写操作按照每次 64 字节或者是 2 字节块来执行。对程序存储器的擦除操作也按照每次 1024 字节块执行。不允许用户代码执行批量擦除操作。

写或擦除程序存储器将停止取指操作，直到写或擦除操作完成为止。在写或擦除期间不能访问程序存储器，因此无法执行代码。内部编程定时器可终止程序存储器的写入和擦除操作。

写入程序存储器的值不必是有效指令。执行存有无效指令的程序存储器单元会导致执行 NOP 指令。

6.1 表读和表写

为了读取和写入程序存储器，有两种操作可以让处理器在程序存储空间和数据 RAM 之间移动字节。

- 表读 （TBLRD）
- 表写 （TBLWT）

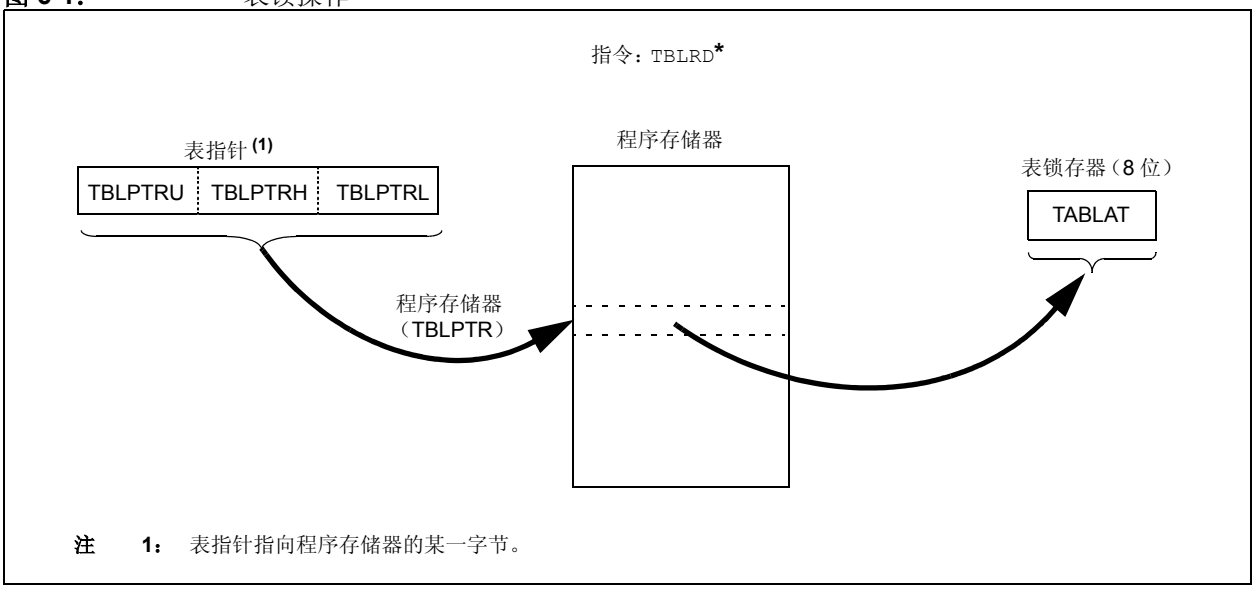
程序存储空间为 16 位宽，而数据 RAM 空间为 8 位宽。表读和表写操作通过一个 8 位寄存器 （TABLAT）在这两个存储空间之间移动数据。

表读操作从程序存储器获取数据并将其存入数据 RAM 空间。图 6-1 显示了程序存储器和数据 RAM 之间的表读操作。

表写操作将数据存储空间中的数据存入程序存储器中的保持寄存器。**第 6.5 节 “写闪存程序存储器”**将详细介绍将保持寄存器中的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 之间的表写操作。

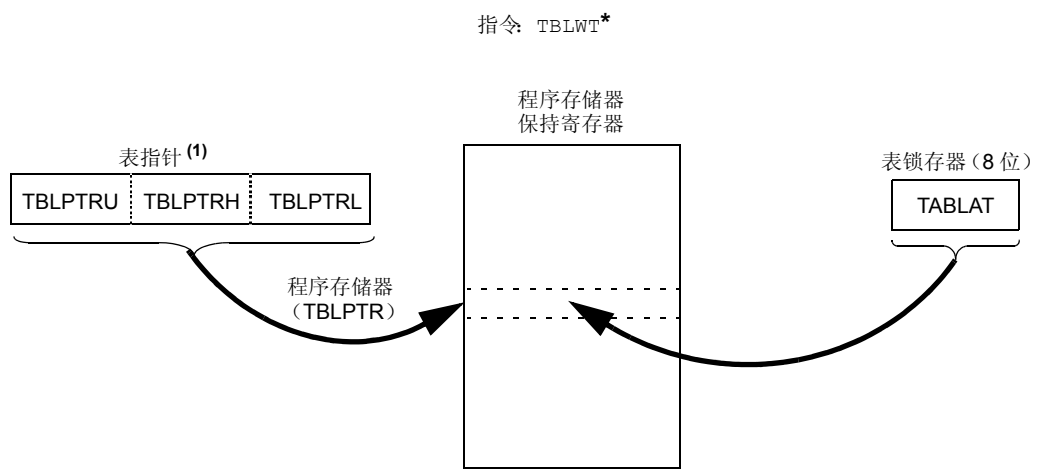
表操作以字节为单位进行。一个包含数据而非程序指令的表块不需要以字为单位。因此，表块可在任何字节地址处起始和结束。如果使用表写操作将可执行代码写入程序存储器，程序指令必须按字对齐。

图 6-1： 表读操作



注 1： 表指针指向程序存储器的某一字节。

图 6-2：　　表写操作



注　1： 表指针实际指向的是 8 个保持寄存器中的某一个，其地址由 TBLPTRL<5:0> 决定。**第 6.5 节 “写闪存程序存储器”**中将讨论将数据物理写入程序存储器阵列的过程。

6.2　控制寄存器

TBLRD 和 TBLWT 指令要用到几个控制寄存器。其中包括：

- 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）是访问存储器的控制寄存器。EECON2 寄存器不是物理寄存器，它专用于存储器的擦写顺序。读取 EECON2 得到的全部是 0。

若将 WPROG 位置 1，则允许用户通过 WR 命令进行单字（2 个字节）操作。若清零该位，WR 命令将运行一个 64 字节块。

若将 FREE 位置 1，则允许程序存储器擦除操作。当 FREE 位置 1 时，擦除操作在下一个 WR 命令时开始。当 FREE 位清零时，则只使能写操作。

将 WREN 位置 1 将允许写操作。上电时将清零 WREN 位。在 WR 位置 1 时，WRERR 位将由硬件置 1，当内部编程定时器超时并且写操作完成时，清零 WRERR 位。

注：	如果在正常工作期间，读到 WRERR 为 1，则表明写操作因复位而提早终止或进行了非法的写操作。

控制位 WR 用于启动写操作。该位只能用软件置 1 而不能清零。写操作完成后，由硬件将其清零。

© 2007 Microchip Technology Inc.

寄存器 6-1：　　EECON1：EEPROM 控制寄存器 1

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

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6　**未实现：**读为 0

bit 5　**WPROG：**一字宽编程位
1 = 在下一个 WR 命令时编程 64 字节
0 = 在下一个 WR 命令时编程 2 字节

bit 4　**FREE：**闪存行擦除使能位
1 = 在下一个 WR 命令时擦除 TBLPTR 寻址的程序存储器行 （擦除操作完成时清零）
0 = 仅执行写操作

bit 3　**WRERR**：闪存程序错误标志位 (1)
1 = 写操作提早终止 （由于正常工作中自定时编程期间的任何复位，或非法写入）
0 = 写操作完成

bit 2　**WREN：**闪存程序写使能位
1 = 允许向闪存程序寄存器的写周期
0 = 内存位闪存程序寄存器的写周期

bit 1　**WR：**写控制位
1 = 启动程序存储器擦除或写周期（该操作是自定时的，一旦写入完成该位即由硬件清零。软件只能将 WR 位置 1 而不能清零）。
0 = 写周期完成

bit 0　**未实现：**读为 0

注　1： 发生 WRERR 时，不会清零 EEPGD 和 CFGS 位。这样就可对错误状态进行跟踪。

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）在程序存储器中寻址字节。该TBLPTR由三个 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 位可使器件寻址至多 2 MB 的程序存储空间。第 22 位则允许访问器件 ID、用户 ID 以及配置位。

TBLRD和TBLWT指令使用表指针寄存器TBLPTR。利用表操作的 4 种方法之一，这些指令可以更新 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 表指针边界

TBLPTR 用于闪存程序存储器的读取、写入和擦除。

当执行 TBLRD 时，TBLPTR 的所有 22 位决定要从程序存储器读入 TABLAT 的字节。

当执行 TBLWT 时，表指针寄存器的低 7 位（TBLPTR<6:0>）决定要写入的是 64 个程序存储器保持寄存器的哪一个。当开始定时写入程序存储器时 （通过 WR 位），表指针寄存器的高 12 位（TBLPTR<21:10>）决定要写入哪一个 1024 字节的程序存储块。如需更多详情，请参见**第 6.5 节 “写闪存程序存储器”**。

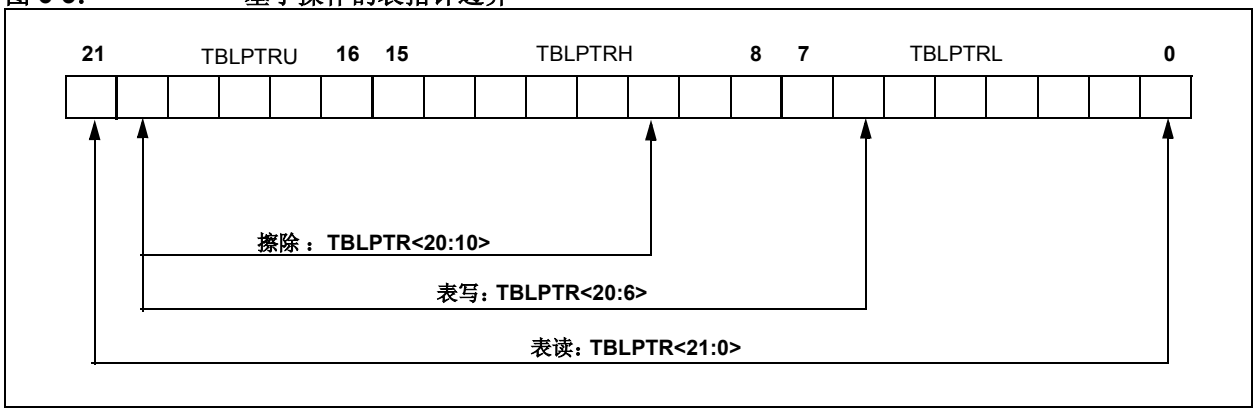
当擦除程序存储器时，表指针的 12 个 MSb 指向将要擦除的 1024 字节块。最低有效位被忽略。

图 6-3 说明了基于闪存程序存储器操作的相关 TBLPTR 边界。

表 6-1： 使用 TBLRD 和 TBLWT 指令执行表指针操作

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： 基于操作的表指针边界



 © 2007 Microchip Technology Inc.

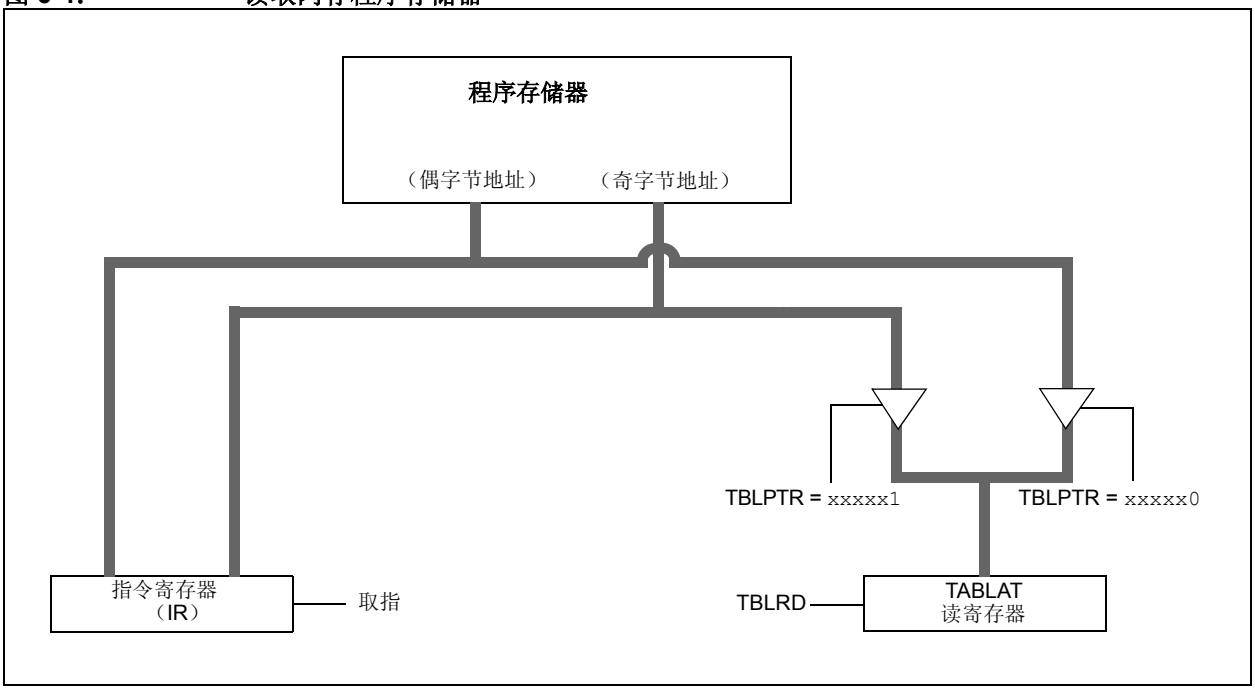
6.3 读取闪存程序存储器

TBLRD 指令用于从程序存储器获取数据并存入数据 RAM。读表操作每次从程序存储器读取一个字节。

TBLPTR 指向程序存储空间内的某个字节。执行 TBLRD 会将指向的字节存入 TABLAT。另外，可以自动修改 TBLPTR 以进行下次读表操作。

内部程序存储器通常是以字为单位构成的。由地址的最低有效位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：　　读取闪存程序存储器



例 6-1：　　读取一个闪存程序存储器字

```
            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
            MOVWF     WORD_ODD
```

6.4 擦除闪存程序存储器

最小擦除数据块大小为 512 字 （即 1024 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。闪存存阵列不支持字擦除。

当从单片机启动擦除操作时，将擦除程序存储器的一个 1024 字节块 。TBLPTR<21:10> 的高 12 位指向将被擦除的数据块。TBLPTR<9:0> 被忽略。

擦除操作由 EECON1 寄存器控制。必须将 WREN 位置 1 以使能写操作。置 1 FREE 位以选择擦除操作。作为保护机制，必须采用 EECON2 的写操作启动序列。

对于擦除内部闪存，长写周期是必要的。在长写周期中，指令执行暂停。内部编程定时器将终止长周期写操作。

6.4.1 闪存程序存储器擦除顺序

擦除内部程序存储器单元块的步骤如下：

1. 将要擦除的行地址载入表指针。
2. 设置 WREN 和 FREE 位（EECON1<2,4>）以执行擦除操作。
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. 在擦除操作的 Tiw（见参数 D133A）期间，CPU 将停止工作。
8. 重新允许中断。

例 6-2： 擦除闪存程序存储器行

```
                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
                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

最小编程块为 32 个字（即 64 字节）。也支持一字或两字节编程。

表写操作用于在内部装入编程闪存存储器所需的保持寄存器。表写操作使用 64 个保持寄存器进行编程 。

由于表锁存器 （TABLAT）只有单字节大小，因此每次编程操作都必须执行 64 次 TBLWT 指令（如果 WPROG = 0）。由于只写入保持寄存器，所有表写操作实际上都是短周期写操作。在更新完 64 个寄存器后，必须写入EECON1寄存器以便使用长写操作开始编程。

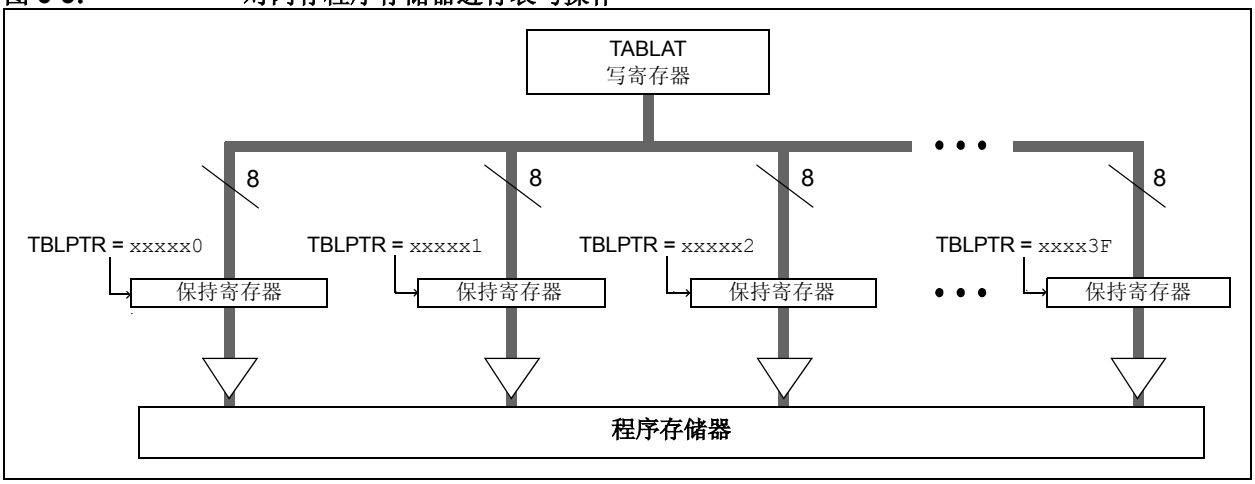
对于编程内部闪存而言，长写周期是必须的。在长写周期中，指令暂停执行，内部编程定时器将终止长写操作。

片上定时器控制写操作的时间，写 / 擦除电压由可在器件电压范围内运行的片上电荷泵产生。

注 1： 与早期的 PIC18 闪存器件不同，PIC18F87J11 系列器件在写操作发生后并不会复位保持寄存器。 必须在编程序列之前清零或改写保持寄存器。

2： 为了保证程序寄存器单元的耐用性，在两次擦除操作不应对某一闪存字节编程多于一次。在尝试第二次修改目标单元的内容之前 ，需要对目标行进行行擦除或对整个存储器进行批量擦除操作。

图 6-5： 对闪存程序存储器进行表写操作



6.5.1 闪存程序存储器写操作序列

对内部程序存储器编程的事件顺序应为：

1. 将 1024 个字节读入 RAM。
2. 必要时更新 RAM 中的数据值。
3. 把要擦除的目标地址载入表指针。
4. 执行行擦除操作。
5. 把要写入的第一个字节的地址装入表指针，表指针减 1。
6. 通过自动递增将 64 个字节写入保持寄存器。
7. 将 WREN 位 （EECON1<2>）置 1 以使能字节写。
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. 在写操作期间， CPU 将停止工作 TIW 时间 （见参数 D133A）。
13. 重新允许中断。
14. 重复从 6 到 13 的步骤直到 1024 个字节完全写入程序寄存器。
15. 校验存储器 （表读）。

下一页中的例 6-3 给出了所需的代码示例。

注： 在置 1 WR 位前，表指针地址需要在保持寄存器中的 64 字节地址范围内。

例 6-3：　写闪存程序存储器

```
                    MOVLW   CODE_ADDR_UPPER      ; Load TBLPTR with the base address
                    MOVWF   TBLPTRU              ; of the memory block, minus 1
                    MOVLW   CODE_ADDR_HIGH
                    MOVWF   TBLPTRH
                    MOVLW   CODE_ADDR_LOW
                    MOVWF   TBLPTRL
ERASE_BLOCK
                    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
                    MOVLW   0AAh
                    MOVWF   EECON2               ; write 0AAh
                    BSF     EECON1, WR           ; start erase (CPU stall)
                    BSF     INTCON, GIE          ; re-enable interrupts
                    MOVLW   D'16'
                    MOVWF   WRITE_COUNTER        ; Need to write 16 blocks of 64 to write
                                                 ; one erase block of 1024
RESTART_BUFFER
                    MOVLW   D'64'
                    MOVWF   COUNTER
                    MOVLW   BUFFER_ADDR_HIGH     ; point to buffer
                    MOVWF   FSR0H
                    MOVLW   BUFFER_ADDR_LOW
                    MOVWF   FSR0L
FILL_BUFFER
                    ...                          ; read the new data from I2C, SPI,
                                                 ; PSP, USART, etc.
WRITE_BUFFER
                    MOVLW   D’64                 ; number of bytes in holding register
                    MOVWF   COUNTER
WRITE_BYTE_TO_HREGS
                    MOVFF   POSTINC0, WREG       ; 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_BYTE_TO_HREGS
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)
                    BSF     INTCON, GIE          ; re-enable interrupts
                    BCF     EECON1, WREN         ; disable write to memory

                    DECFSZ  WRITE_COUNTER        ; done with one write cycle
                    BRA     RESTART_BUFFER       ; if not done replacing the erase block
```


© 2007 Microchip Technology Inc.

6.5.2 闪存程序寄存器写序列 （字编程）

PIC18F87J11 系列器件具有允许编程一个字 （两个字节）的功能。该功能在 WPROG 位置 1 时使能。若已擦除了此存储单元，则需要使用以下序列来使能该功能：

1. 用要写入的数据的地址装载表指针寄存器。
2. 将 2 个字节写入保持寄存器并执行表写操作。
3. 置 1 WREN 位（EECON1<2>）以使能字节写。
4. 禁止中断。
5. 向 EECON2 写入 55h。
6. 向 EECON2 写入 0AAh。
7. 将 WR 位置 1, 这将开始写周期。
8. 在写操作期间， CPU 将停止工作 TIW 时间 （见参数 D133A ）。
9. 重新允许中断。

例 6-4： 单字写闪存程序存储器

```
                MOVLW    CODE_ADDR_UPPER      ; Load TBLPTR with the base address
                MOVWF    TBLPTRU
                MOVLW    CODE_ADDR_HIGH
                MOVWF    TBLPTRH
                MOVLW    CODE_ADDR_LOW
                MOVWF    TBLPTRL

                MOVLW    DATA0
                MOVWF    TABLAT
                TBLWT*+
                MOVLW    DATA1
                MOVWF    TABLAT
                TBLWT*


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

6.5.3 写校验

根据具体的应用，好的编程习惯一般需要使用原始值对写入值进行校验。在过多的写入可能会导致写入的位数接近规范极限值的应用中，就应该采用写校验。

6.5.4 写操作意外终止

如果由于意外事件（如掉电或意外复位）终止了写操作，就应该对刚刚编程的存储单元进行验证，如有必要，还要重新进行编程。如果写入操作在正常操作过程中被 MCLR 复位或 WDT 超时复位中断，用户就可以根据需要检测 WRERR 位并重写该单元。

6.6 代码保护时的闪存程序存储器操作

如需了解更多有关闪存程序存储器代码保护的信息，请参见**第 24.6 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>）					55
TBPLTRH	程序存储器表指针次高字节 （TBLPTR<15:8>）								55
TBLPTRL	程序存储器表指针低字节 （TBLPTR<7:0>）								55
TABLAT	程序存储器表锁存器								55
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
EECON2	程序存储控制寄存器 2（非物理寄存器）								57
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	57

图注： **— =** 未实现，读为 0。访问闪存程序存储器时不使用阴影单元。

© 2007 Microchip Technology Inc.

7.0 外部存储器总线

注： 64 引脚器件内未实现外部存储器总线。

外部存储器总线（External Memory Bus，EMB）使器件能像访问程序存储器或数据存储器一样访问外部存储器件（如闪存、EPROM 和 SRAM 等）。外部存储器总线同时支持 8 位和 16 位数据宽度模式以及三种高达 20 位的地址宽度模式。

该总线包含 28 个引脚，复用 4 个 I/O 端口。三个端口（PORTD、PORTE 和 PORTH）同地址 / 数据总线共复用 20 根线， PORTJ 同总线控制信号复用。

表 7-1 中罗列了总线引脚并介绍了他们的功能。

表 7-1： PIC18F87J11 系列外部总线－ I/O 端口功能

名称	端口	位	外部存储器总线功能
RD0/AD0	PORTD	0	地址位 0 或数据位 0
RD1/AD1	PORTD	1	地址位 1 或数据位 1
RD2/AD2	PORTD	2	地址位 2 或数据位 2
RD3/AD3	PORTD	3	地址位 3 或数据位 3
RD4/AD4	PORTD	4	地址位 4 或数据位 4
RD5/AD5	PORTD	5	地址位 5 或数据位 5
RD6/AD6	PORTD	6	地址位 6 或数据位 6
RD7/AD7	PORTD	7	地址位 7 或数据位 7
RE0/AD8	PORTE	0	地址位 8 或数据位 8
RE1/AD9	PORTE	1	地址位 9 或数据位 9
RE2/AD10	PORTE	2	地址位 10 或数据位 10
RE3/AD11	PORTE	3	地址位 11 或数据位 11
RE4/AD12	PORTE	4	地址位 12 或数据位 12
RE5/AD13	PORTE	5	地址位 13 或数据位 13
RE6/AD14	PORTE	6	地址位 14 或数据位 14
RE7/AD15	PORTE	7	地址位 15 或数据位 15
RH0/A16	PORTH	0	地址位 16
RH1/A17	PORTH	1	地址位 17
RH2/A18	PORTH	2	地址位 18
RH3/A19	PORTH	3	地址位 19
RJ0/ALE	PORTJ	0	地址锁存使能 （ALE）控制引脚
RJ1/$\overline{OE}$	PORTJ	1	输出使能 （$\overline{OE}$）控制引脚
RJ2/$\overline{WRL}$	PORTJ	2	写低位 （$\overline{WRL}$）控制引脚
RJ2/$\overline{WRL}$	PORTJ	3	写高位 （$\overline{WRH}$）控制引脚
RJ4/BA0	PORTJ	4	字节地址位 0 （BA0）
RJ5/$\overline{CE}$	PORTJ	5	芯片使能 （$\overline{CE}$）控制引脚
RJ6/$\overline{LB}$	PORTJ	6	低字节使能 （$\overline{LB}$）控制引脚
RJ7/$\overline{UB}$	PORTJ	7	高字节使能 （$\overline{UB}$）控制引脚

注： 为避免混淆，这里只显示 I/O 端口和外部总线分配。在一些引脚上可能还有 1 个或多个其他的复用功能。

7.1 外部存储器总线控制

MEMCON 寄存器控制（寄存器 7-1）用于控制接口的工作模式。 MEMCON 寄存器在所有的程序存储器工作模式（除单片机模式）下可用。在单片机模式中，寄存器被禁止且不能写入。

EBDIS 位（MEMCON<7>）控制总线的工作模式和相关端口的功能。将 EBDIS 位清零会使能接口和禁止端口的 I/O 功能及其他复用这些引脚的任何功能。将该位置 1 将使能 I/O 端口和其他复用这些引脚的功能，但当需要外部存储器操作时，接口的优先级高于引脚上的任何功能。默认情况下，外部总线总是被使能的并禁止其他所有的 I/O。

对EBDIS位的操作也受当前所用的程序存储器模式的影响。**第 7.5 节“程序存储模式和外部存储总线”**中详细讨论了 EBDIS 位的操作。

WAIT 位支持给外部存储操作增加等待状态。**第 7.3 节“等待状态”**中详细讨论了这些位的用法。

WM 位选择总线在 16 位数据宽度模式下使用的特定工作模式。**第 7.6 节 “16 位数据宽度模式”**中对此进行了更详细的讨论。当选择使用 8 位数据宽度模式时这些位不起作用。

MEMCON 寄存器（见寄存器 7-1）与 PR2 共用存储器空间，同时可根据 WDTCON 寄存器（见寄存器 24-9）中的 ADSHR 位所赋的值对它们进行交替选择。

寄存器 7-1： MEMCON：外部存储器总线控制寄存器

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
bit7							bit0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
— n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **EBDIS**：外部总线禁止位
1 = 当单片机访问外部存储器时使能外部总线；否则，所有的外部总线驱动器映射为 I/O 端口
0= 一直使能外部总线，禁止 I/O 端口

bit 6 **未实现**：读为 0

bit 5-4 **WAIT1:WAIT0**：表读和表写总线周期等待计数位
11 = 表读和表写将等待 0TCY
10 = 表读和表写将等待 1TCY
01 = 表读和表写将等待 2TCY
00 = 表读和表写将等待 3TCY

bit 3-2 **未实现**：读为 0

bit 1-0 **WM1:WM0**：16 位数据总线宽度的 TBLWT 操作选择位
1x = 字写模式：TABLAT 作为字输出，当写 TABLAT 时 $\overline{WRH}$ 有效
01= 字节选择模式：TABLAT 数据复制 MSB 和 LSB，$\overline{WRH}$ 和（$\overline{UB}$ 或 $\overline{LB}$）将有效
00 = 字节写模式：TABLAT 数据复制 MSB 和 LSB， $\overline{WRH}$ 或 $\overline{WRL}$ 将有效

© 2007 Microchip Technology Inc.

7.2 地址和数据宽度

PIC18F87J11 系列器件能在相同的存储总线上为不同的地址和数据宽度独立进行配置。两者均由 CONFIG3L 寄存器中的配置位设置。作为配置位意味着这些选项只能通过对器件编程才能进行配置，而不能由软件控制。

BW 位选择 8 位或 16 位数据总线宽度。该位置 1（默认）将选择 16 位数据宽度。

EMB1:EMB0 位决定程序存储器工作模式和地址总线宽度。可用的总线宽度有：20 位、16 位、12 位和单片机模式（禁止外部总线）。选择 16 位或 12 位宽度会使相应的高位线用于 I/O 功能；这些引脚不再受 EBDIS 位设置的影响。例如，选择 16 位地址模式（EMB1:EMB0 = 01）会禁止 A19:A16 并允许 PORTH<3:0> 工作不受总线中断控制。使用较小的地址宽度使用户可为特定的设计定制存储总线以适应外部存储空间的大小，同时留出引脚用于专用 I/O 操作。

因为 EMB 位能禁止引脚用于存储总线操作，所以选择一个至少与数据宽度相等的地址宽度尤为重要。如果 12 位地址宽度用于 16 位数据宽度，那么总线上数据的高 4 位将不可用。

地址和数据宽度的所有组合要求在相同的线上复用地址和数据信息。表 7-2 总结了地址和数据复用以及使用短地址宽度时可用的 I/O 端口。

7.2.1 外部总线上的地址移位

默认情况下，PC 值就是外部总线上的地址。也就是说，对单片机而言，低于片内存储器最高地址的外部存储器件的地址是不可用的。要访问这些物理存储单元，单片机和外部存储器间的连接逻辑必须设法译址。

为简化接口，外部总线提供了一种扩展的单片机模式，它能自动进行地址移位。EASHFT 配置位控制该功能。将该位置 1 会使总线上的实际地址发生偏移，偏移量为单片机的片内程序存储器大小，并设置总线的起始地址为 0000h。这样器件就可以访问外部存储器的所有物理地址。

7.2.2 21 位寻址

作为 20 位地址宽度操作的扩展，外部存储器总线也能寻址整个 2MB 的存储空间。将总线地址位 0（BA0）控制线用作地址的最低位可以完成上述功能。$\overline{UB}$ 和 $\overline{LB}$ 控制信号还可能同某些存储器件一起使用在 16 位宽的数据字内选择高字节和低字节。

在 8 位和某些 16 位数据宽度模式中可使用该寻址模式。**第 7.6.3 节“16 位字节选择模式”**和**第 7.7 节“8 位数据宽度模式”**中提供了更多详细信息。

表 7-2： 不同地址和数据宽度的地址和数据线

数据宽度	地址宽度	复用的数据和地址线（和相应端口）	仅地址线（和相应端口）	可用作 I/O 的端口
8 位	12 位	AD7:AD0（PORTD<7:0>）	AD11:AD8（PORTE<3:0>）	（PORTE<7:4>），PORTH 的全部引脚
	16 位		AD15:AD8（PORTE<7:0>）	PORTH 的全部引脚
	20 位		A19:A16, AD15:AD8（PORTE<3:0>），（PORTE<7:0>）	—
16 位	16 位	AD15:AD0（PORTD<7:0>），（PORTE<7:0>）	—	PORTH 的全部引脚
	20 位		A19:A16（PORTE<3:0>）	—

7.3 等待状态

我们通常假定外部存储器件将工作在单片机时钟速率下，但实际情况往往并非如此。事实上，许多器件读或写数据所需的时间长于表读或表写操作允许的时间。

为补偿时间差，在使用总线时可以配置外部存储器总线来给表操作添加一个固定的延时。设置 WAIT 配置位使能等待状态。当使能时，通过 WAIT1:WAIT0 位（MEMCON<5:4>）设置延时长度。延时长度是单片机指令周期时间的倍数，添加在表操作指令周期后。延时范围为 0 至 3 个 TCY（默认值）。

7.4 端口引脚弱上拉

除高位地址线 A19:A16 外，与外部存储器总线相关的引脚均配有弱上拉。PORTG 寄存器的高 3 位（PORTG<7:5>）控制上拉。 高三位分别命名为 RDPU、REPU 和 RJPU，分别控制 PORTD、PORTE 和 PORTJ 上的弱上拉。将其中某位置 1 会使能相应端口的上拉。默认情况下，器件复位时禁止所有的上拉。

在扩展单片机模式下，端口上拉可用于当总线暂时被禁止时 （EBDIS =1）在外部总线上保存存储器状态。

7.5 程序存储模式和外部存储器总线

同时使用片内和外部程序存储器时，PIC18F87J11 系列器件能工作在两种程序存储模式中的任意一种模式下。端口引脚的复用功能取决于选定的程序存储模式和 EBDIS 位的设置。

在**单片机模式**下，总线无效，引脚只具有端口功能。不允许写 MEMCOM 寄存器。EBDIS 的复位值（0）被忽略， EMB 引脚充当 I/O 端口。

在**扩展单片机模式**下，外部程序存储总线共享引脚的 I/O 端口功能。当器件在外部程序存储空间上进行取指或表读 / 表写操作时，引脚将具有外部总线功能。如果器件只是对内部程序存储单元进行取指和访问，EBDIS 控制位将把引脚从外部存储功能改为 I/O 端口功能。当 EBDIS = 0 时，引脚作为外部总线使用。当 EBDIS = 1 时，引脚作为 I/O 端口使用。

当 EBDIS =1 时，器件对外部存储器进行取指或访问，引脚将切换为外部总线功能。如果在外部存储器执行的程序将 EBDIS 位置 1，那么置 1 动作将被延时，直到程序转到内部存储器为止。那时，引脚将从外部总线功能改为 I/O 端口功能。

当 EBDIS =0 时，如果器件在内部存储器外执行，存储总线的地址 / 数据和控制引脚将失效。它们进入以下状态 有效的地址 / 数据引脚处于三态； CE、OE、WRH、WRL、UB 和 LB 信号为 1， ALE 和 BA0 为 0。注意只有与当前地址宽度关联的引脚被强制为三态；其他引脚继续作为 I/O 使用。例如在 16 位地址宽度情况下，只有 AD<15:0>（PORTD 和 PORTE）受到影响；而 A19:A16（PORTH<3:0>）继续作为 I/O 使用。

在所有的外部存储模式下，总线的优先级高于其他任何与总线共享引脚的外设。这包括并行主端口和串行通信模块，这两者的优先级高于 I/O 端口。

7.6 16 位数据宽度模式

在 16 位数据宽度模式下，外部存储器接口能以三种不同的配置连接到外部存储器：

- 16 位字节写
- 16 位字写
- 16 位字节选择

MEMCON 寄存器中的 WM1:WM0 （MEMCON<1:0>）决定了使用的配置。这三种不同的配置使设计人员在使用带16位数据的8位和16位器件时具有最大的灵活性。

对于所有的 16 位模式，地址锁存使能 （ALE）引脚表明在外部存储接口总线上地址位 AD<15:0> 可用。在地址锁存之后，输出使能信号 （OE）将立即使程序存储器的两个字节形成一个 16 位的指令字。在单片机访问外部存储器（不论读或写）的任何时间内芯片使能信号（CE）始终有效。只要器件一进入休眠模式，信号就变为无效 （变为高电平）。

在字节选择模式下，JEDEC标准闪存存储器将需要BA0用于字节地址线和一条 I/O 线，以在字节和字模式间进行选择。其他 16 位模式不需要 BA0。JEDEC 标准静态 RAM 存储器将使用 UB 或 LB 信号选择字节模式。

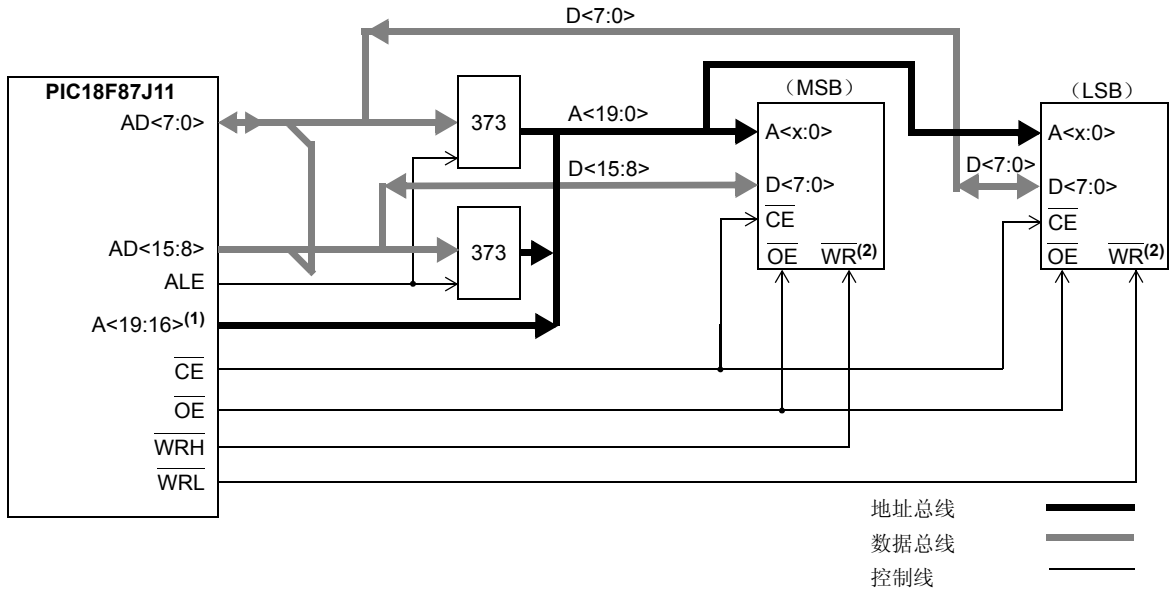
© 2007 Microchip Technology Inc.

7.6.1 16 位字节写模式

图 7-1 举例说明了 PIC18F87J11 系列器件的 16 位字节写模式。该模式用于将 2 个独立的 8 位存储器相连以进行 16 位的操作。通常包括基本 EPROM 和闪存器件。允许对字节宽的外部存储器进行表写操作。

在 TBLWT 指令周期内， TABLAT 数据在 AD15:AD0 总线的高低字节中。TBLPTR 的 LSb 选通相应的 WRH 或 WRL 控制线。

图 7-1： 16 位字节写模式示例



注 1： 高位地址线只用于 20 位宽度地址。

2： 该信号只应用于表写操作。请参见**第 6.1 节 “表读和表写”**。

7.6.2 16 位字写模式

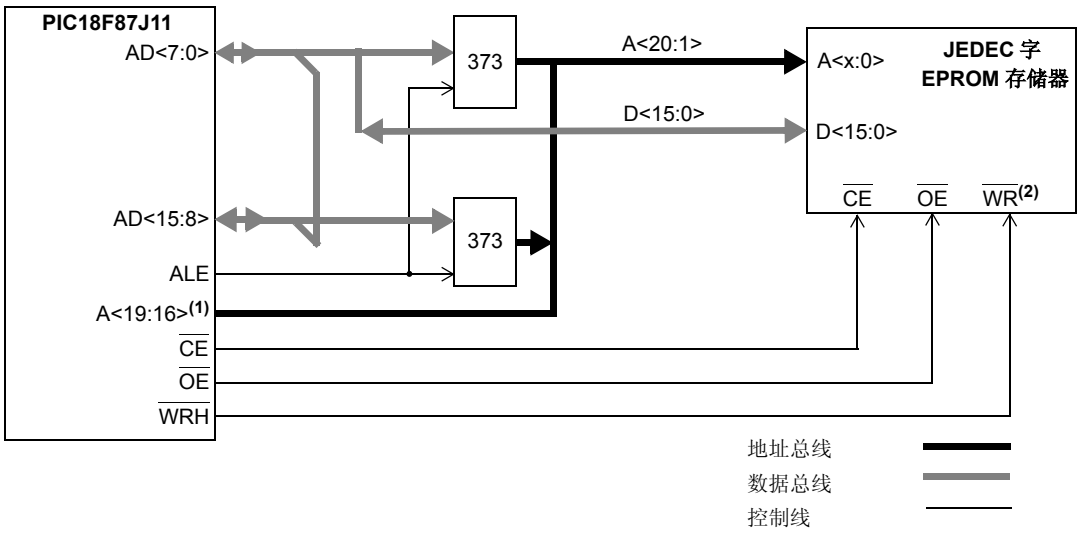
图 7-2 举例说明了 PIC18F87J11 系列器件的 16 位字写模式。该模式用于字宽存储器，包括一些 EPROM 和闪存类型的存储器。该模式允许从所有的 16 位存储器取操作码和表读，并向所有的字宽外部存储器进行表写。这种方式把 TBLWT 周期分为写入偶地址和写入奇地址。

在写入偶地址（TBLPTR<0> = 0）的 TBLWT 周期中，TABLAT 数据转移到保持锁存器中，外部地址数据总线在总线周期的数字部分呈现为三态。没有有效写信号。

在写入奇地址（TBLPTR<0> = 1）的 TBLWT 周期中，TABLAT 中的数据出现在 AD15:AD0 总线的高字节上。保持锁存器的内容保存在 AD15:AD0 总线的低字节上。

$\overline{WRH}$ 信号选通每个写周期，不使用 $\overline{WRL}$ 引脚。BA0 引脚上的信号表示 TBLPTR 的 LSb，但该引脚是悬空的。相反，$\overline{UB}$ 和 $\overline{LB}$ 信号有效，可用于同时选择奇偶地址的两个字节。这种方法的明显缺陷是表写必须在特定字边界上成对完成，以便正确地写一个字存储单元。

图 7-2： 16 位字写模式示例



注 1： 高位地址线只用于 20 位宽度地址。

2： 该信号只应用于表写操作。请参见**第 6.1 节 “表读和表写”**。

© 2007 Microchip Technology Inc.

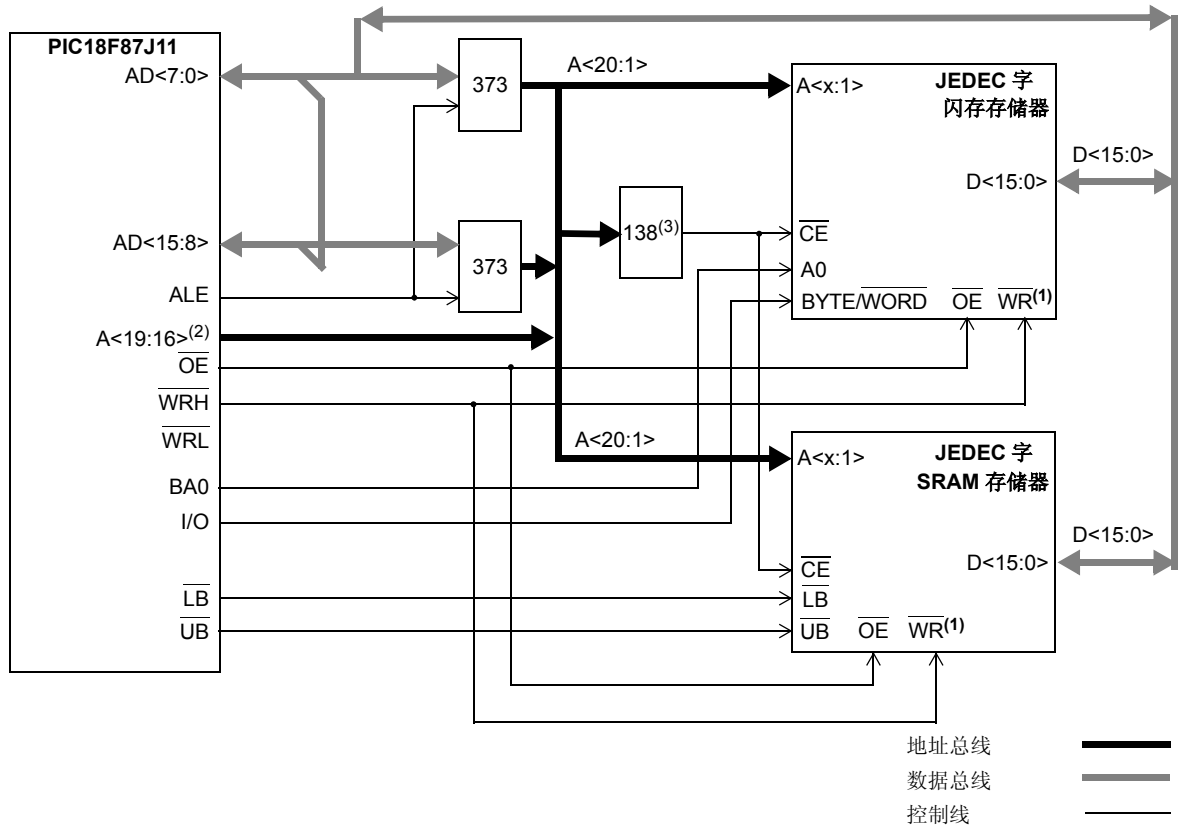
7.6.3 16 位字节选择模式

图 7-3 举例说明了 16 位字节选择模式。该模式允许对带字节选择功能的字宽外部存储器进行表写操作。这个通常包括字宽闪存和 SRAM 器件。

在 TBLWT 周期内，TABLAT 中的数据保存在 AD15:AD0 总线的高低字节中。WRH 信号选通每个写周期； WRL 没有使用。BA0 或 UB/LB 信号用于根据 TBLPTR 寄存器中的最低有效位选择要写的字节。

闪存和SRAM器件使用不同的控制信号组合来实现字节选择模式。JEDEC 标准闪存存储器要求控制器 I/O 端口引脚连接到存储器的 BYTE/WORD 引脚，以提供选择信号。它们还使用控制器的BA0 信号作为字节地址。在另一方面，JEDEC 标准静态 RAM 存储器将使用 UB 或 LB 信号选择字节模式。

图 7-3： 16 位字节选择模式示例



注 1： 该信号只应用于表写操作。请参见**第 6.1 节 “表读和表写”**。

2： 高位地址线只用于 20 位宽度地址。

3： 只有当访问多个存储器件时才需要取消复用。

7.6.4 16 位模式时序

在外部存储器总线上出现的控制信号在不同的操作模式下是不同的。图7-4和图7-5所示为典型的信号时序图。

图 7-4： 执行 TBLRD 的外部存储器总线时序 （扩展单片机模式）

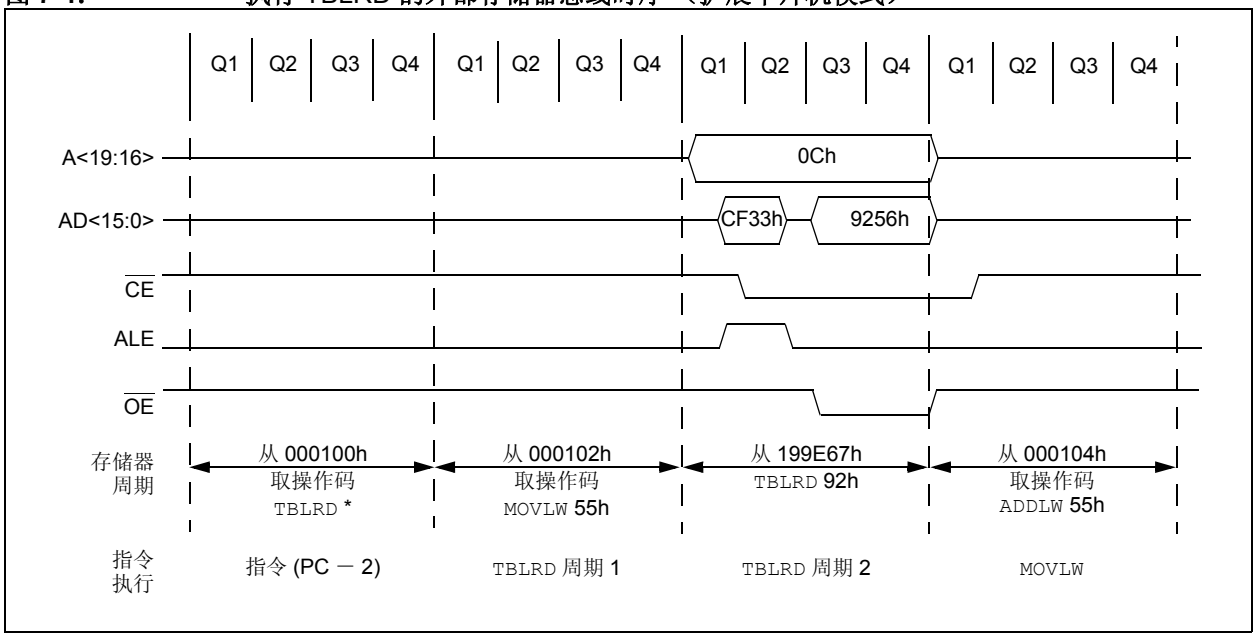
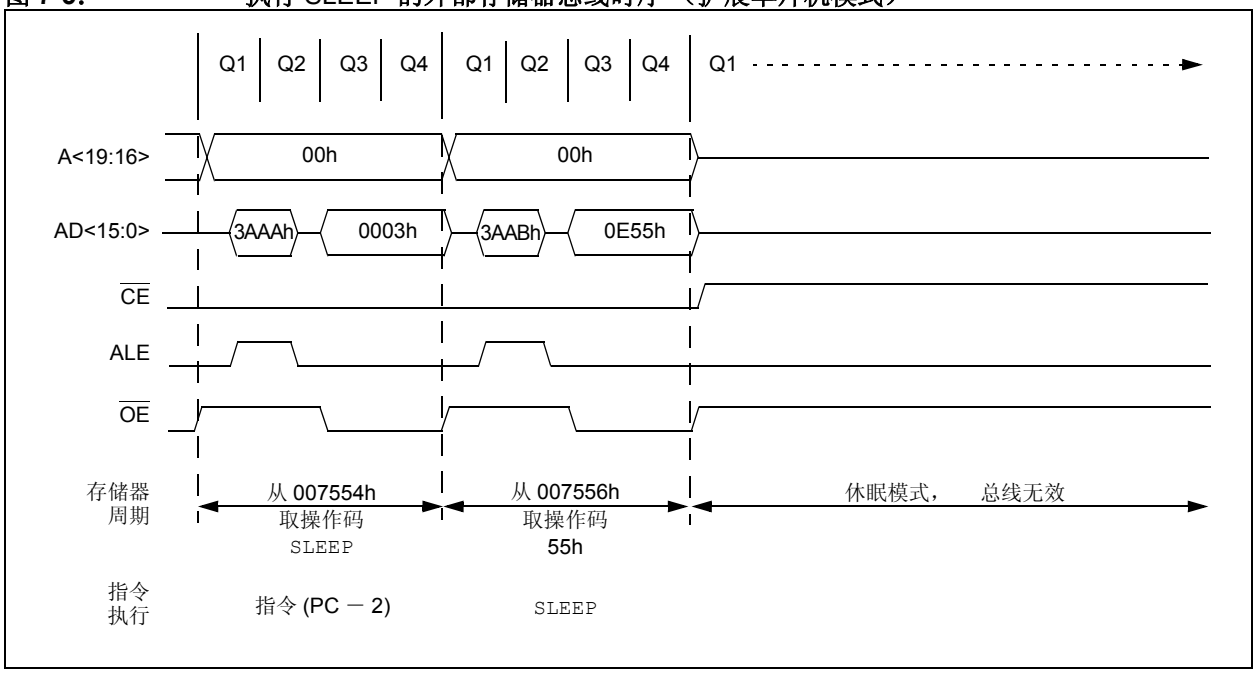


图 7-5： 执行 SLEEP 的外部存储器总线时序 （扩展单片机模式）



 © 2007 Microchip Technology Inc.

7.7 8 位数据宽度模式

在 8 位数据宽度模式下，外部存储器总线只工作在复用模式下，即数据总线的低 8 位数据共享。

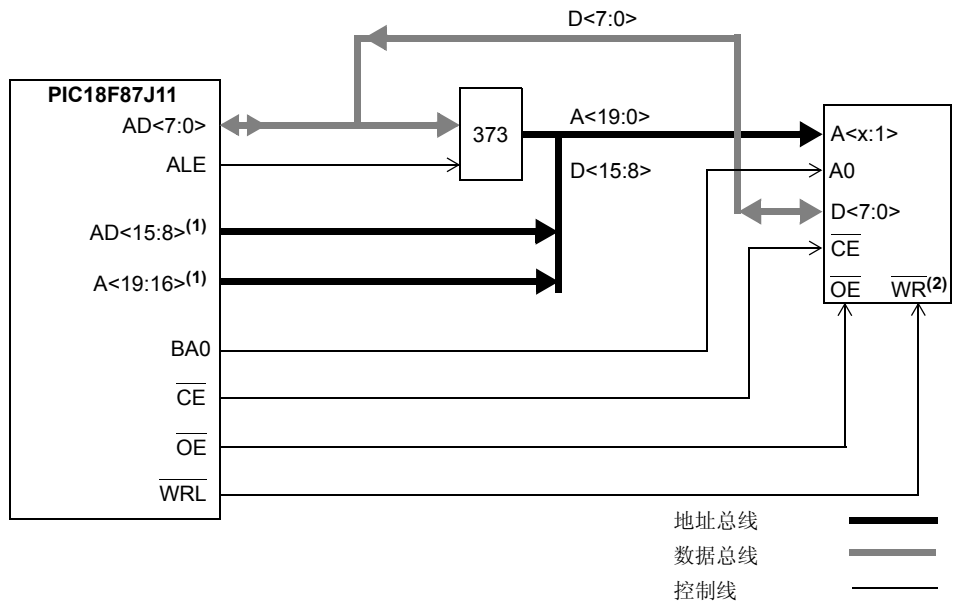
图 7-6 举例说明了 80 引脚器件的 8 位复用模式。该模式用于将 1 个 8 位存储器连接进行 16 位的操作。在共享数据 / 地址总线上取 2 个 8 位字节指令，这两个字节在一个指令周期 （TCY）内顺序取指。所以，设计人员必须选择根据基于 1/2 TCY （2 倍指令速率）的时序计算选择外部存储器件。为了选择合适的存储器速度，必须考虑到连接逻辑往返延时时间以及建立和保持时间。

地址锁存使能 （Address Latch Enable， ALE）引脚表明在外部存储接口总线上地址位 AD<15:0> 可用。输出使能信号 （$\overline{OE}$）将使能程序存储器的一个字节作为指令周期的一部分，然后会更改 BA0，并使能第二个字节以形成 16 位的指令字。在该模式下地址的最低有效位 BA0 必须连接到存储器件。在单片机访问外部存储器（不论读或写）的任何时间内芯片使能信号（$\overline{CE}$）始终有效。只要器件一进入休眠模式，信号就变为无效（变为高电平）。

通常包括基本 EPROM 和闪存器件。允许对字节宽的外部存储器进行表写操作。

在 TBLWT 指令周期内，TABLAT 数据保存在 AD15:AD0 总线的高低字节中。TBLPTR 的 LSb 用于选通 BA0 控制线上相应的电平。

图 7-6： 8 位复用模式示例



注 1： 高位地址位只用于 20 位宽度地址。AD 高字节用于除 8 位以外的所有地址。

2： 该信号只应用于表写操作。请参见**第 6.1 节 “表读和表写”**。

7.7.1 8 位模式时序

在外部存储器总线上出现的控制信号在不同的操作模式下是不同的。图7-7和图7-8所示为典型的信号时序图。

图 7-7： 执行 TBLRD 的外部存储器总线时序 （扩展单片机模式）

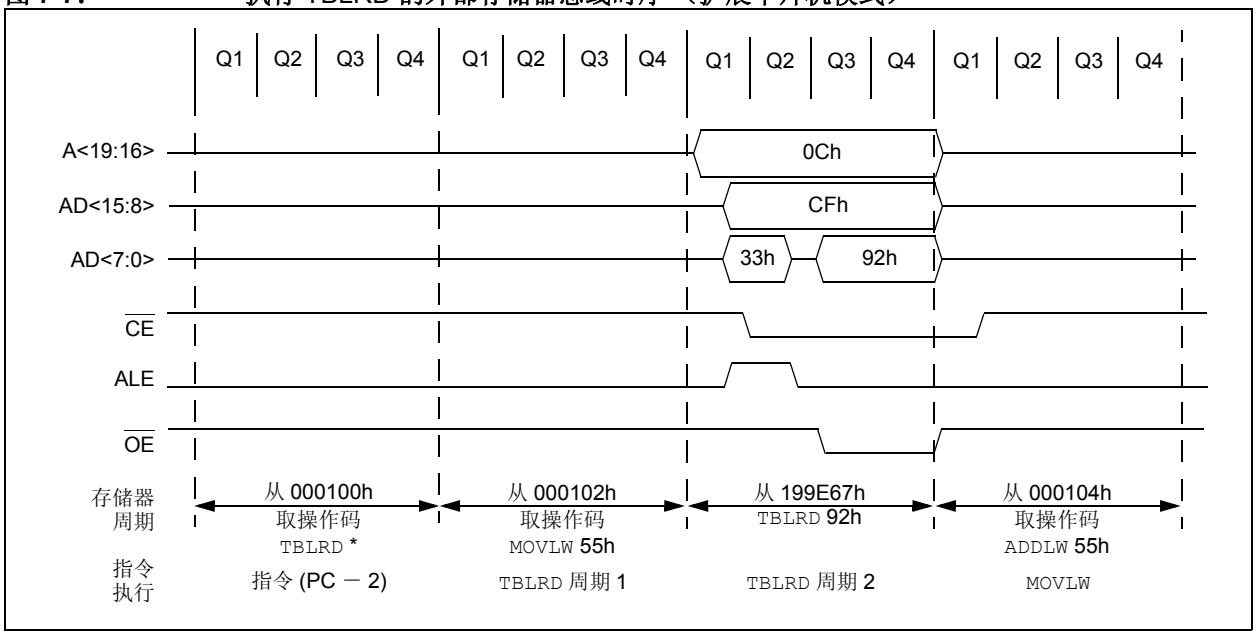
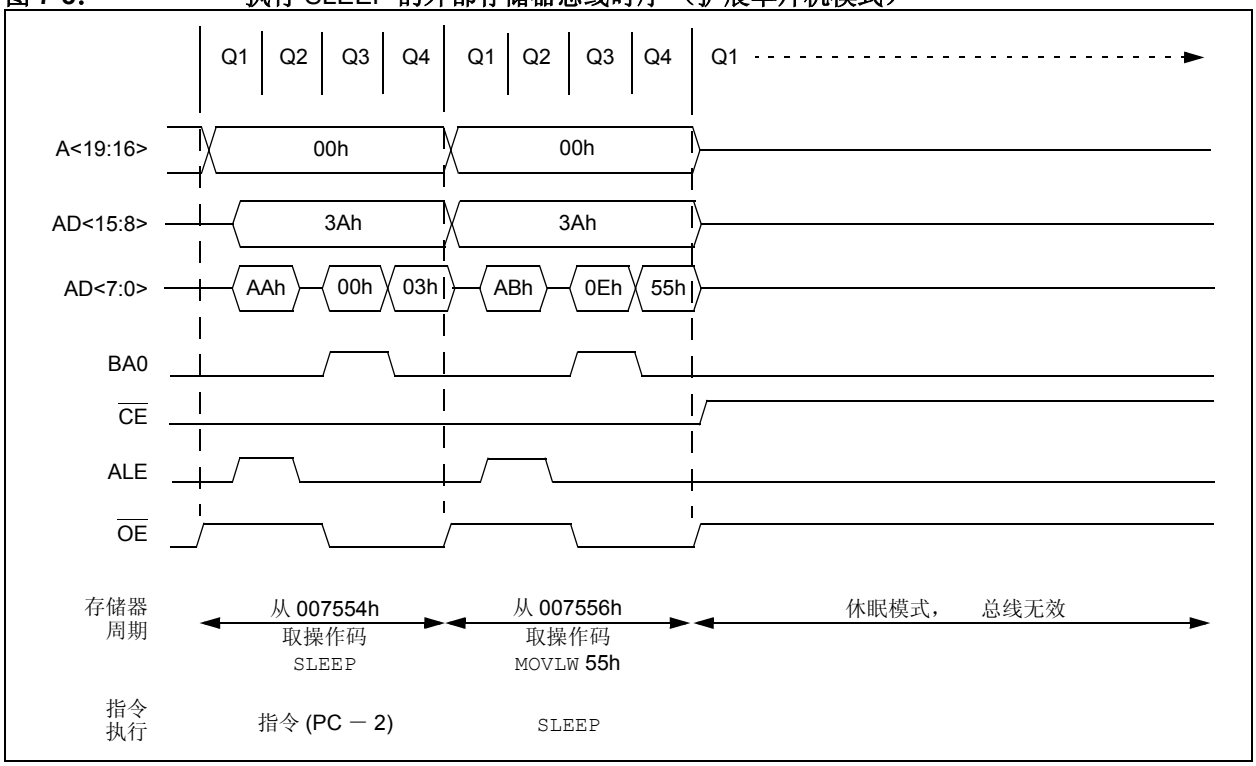


图 7-8： 执行 SLEEP 的外部存储器总线时序 （扩展单片机模式）



© 2007 Microchip Technology Inc.

7.8 工作在功耗管理模式下

在所有功耗管理运行模式下，外部总线都能正常运行。如果选择一个低速的时钟源，总线操作将运行在该速度下。在这些情况下，如果使能了等待状态并把它添加到外部存储操作中，可能会导致对外部存储器访问时间过长。如果希望工作在低功耗运行模式下，用户应该在应用中调整在低时钟速度下访问存储器的次数。

在休眠和空闲模式下，单片机内核不需访问数据，总线操作暂停。外部总线的状态被冻结，地址 / 数据引脚和大多数控制引脚保持在进入该模式前的状态。惟一的变化可能是 CE、LB 和 UB 引脚，它们保持为逻辑高电平。

注：

 © 2007 Microchip Technology Inc.

8.0 8 x 8 硬件乘法器

8.1 简介

所有 PIC18 器件均包含一个 8 x 8 硬件乘法器（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 1 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 8-1 给出了各种硬件和软件乘法运算的比较，包括存储器空间和执行时间。

8.2 工作原理

例 8-1 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 8-2 给出了执行 8 x 8 有符号乘法运算的指令序列。要弄清参数的符号位，必须检查每个乘数的最高有效位（MSb），并做相应的减法。

例 8-1： 8x8 无符号乘法程序

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

例 8-2： 8x8 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 8-1： 各种乘法运算的性能比较

程序	乘法实现方法	程序存储空间（字）	周期数（最多）	时间		
				48 MHz 时	10 MHz 时	4 MHz 时
8 × 8 无符号	软件乘法	13	69	5.7 μs	27.6 μs	69 μs
	硬件乘法	1	1	83.3 ns	400 ns	1 μs
8 × 8 有符号	软件乘法	33	91	7.5 μs	36.4 μs	91 μs
	硬件乘法	6	6	500 ns	2.4 μs	6 μs
16 × 16 无符号	软件乘法	21	242	20.1 μs	96.8 μs	242 μs
	硬件乘法	28	28	2.3 μs	11.2 μs	28 μs
16 × 16 有符号	软件乘法	52	254	21.6 μs	102.6 μs	254 μs
	硬件乘法	35	40	3.3 μs	16.0 μs	40 μs

例 8-3 给出了一个 16 x 16 无符号乘法运算的指令序列。公式 8-1 为所使用的算法。32 位结果存储在 4 个寄存器（RES3:RES0）中。

公式 8-1： 16x16 无符号乘法算法

RES3:RES0 = ARG1H:ARG1L • ARG2H:ARG2L
= (ARG1H • ARG2H • 2^{16}) +
(ARG1H • ARG2L • 2^{8}) +
(ARG1L • ARG2H • 2^{8}) +
(ARG1L • ARG2L)

例 8-3： 16x16 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 8-4 给出了 16 x 16 有符号乘法的指令序列。公式 8-2 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 8-2： 16x16 有符号乘法算法

RES3:RES0 = ARG1H:ARG1L • ARG2H:ARG2L
= (ARG1H • ARG2H • 2^{16}) +
(ARG1H • ARG2L • 2^{8}) +
(ARG1L • ARG2H • 2^{8}) +
(ARG1L • ARG2L) +
(-1 • ARG2H<7> • ARG1H:ARG1L • 2^{16}) +
(-1 • ARG1H<7> • ARG2H:ARG2L • 2^{16})

例 8-4： 16x16 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.

9.0 中断

PIC18F87J10 系列器件具有多个中断源及中断优先级功能，该中断优先级功能可以为大多数中断源分配高优先级或低优先级。高优先级中断向量地址为 0008h，低优先级中断向量地址为 0018h。高优先级中断事件将中断所有可能正在处理的低优先级中断。

有 13 个寄存器用于控制中断操作。这些寄存器是：

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1、 PIR2 和 PIR3
- PIE1、 PIE2 和 PIE3
- IPR1、 IPR2 和 IPR3

建议使用 MPLAB® IDE 提供的 Microchip 头文件命名这些寄存器中的位。这样可使汇编器 / 编译器能够自动识别指定寄存器内这些位的位置。

通常，用三个位来控制中断源的操作。它们是：

- **标志位**表明发生了中断事件
- **使能位**在标志位置 1 时，使程序执行跳转到中断向量地址
- **优先级位**用于选择高优先级或低优先级

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 位可允许全局中断。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、允许位以及相应的全局中断允许位均被置 1 时，中断将立即转到地址 0008h 或 0018h，转到哪个地址取决于优先级位的设置。通过对应的允许位可以禁止某个中断。

当 IPEN 位被清零 （默认状态）时，中断优先级功能被禁止，此时中断与 PIC16 中档器件相兼容。在兼容模式下，各个中断源的中断优先级位均不起作用。INTCON<6> 是 PEIE 位，该位可允许 / 禁止所有的外设中断源。INTCON<7> 是 GIE 位，该位可允许 / 禁止所有的中断源。在兼容模式下，所有中断均跳转到地址 0008h。

当响应中断时，全局中断允许位会被清零以禁止其他中断。如果 IPEN 位被清零，该位就是 GIE 位。如果使用中断优先级，该位就是 GIEH 位或者 GIEL 位。高优先级中断源能中断低优先级中断。处理高优先级中断时，低优先级中断将不被处理。

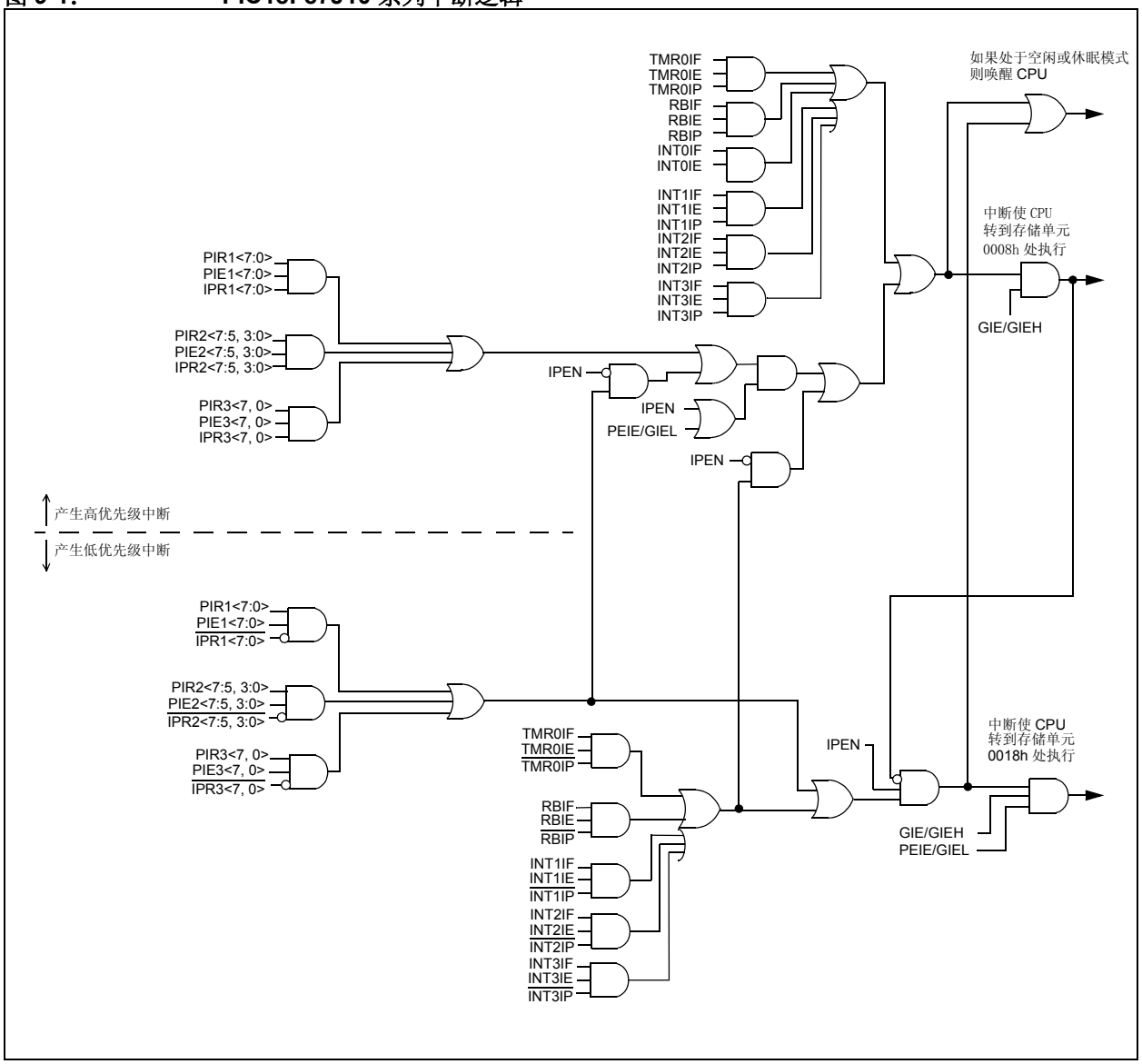
返回地址被压入堆栈，并将中断向量地址 （0008h 或 0018h）装入 PC 中。一旦进入中断服务程序，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

执行 “从中断返回” 指令 RETFIE，退出中断程序并置位 GIE 位 （若使用中断优先级，则为 GIEH 或 GIEL 位），该位重新允许中断。

对于外部中断事件，诸如 INTx 引脚或 PORTB 输入电平变化中断，中断响应延时将为 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。各中断标志位的置位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，都不要使用 MOVFF 指令来修改任何中断控制寄存器。否则可能引起单片机执行出错。

图 9-1：　　PIC18F87J10 系列中断逻辑



© 2007 Microchip Technology Inc.

9.1 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： 当有中断条件产生时，无论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。用户软件应在允许中断之前，确保先将该中断标志位清零。此功能可用软件进行查询。

寄存器 9-1： 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 = 未实现位，读为 0	
-n = 上电复位时的值	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 = 允许所有未屏蔽中断
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 引脚中至少有一位的状态发生了变化 （必须用软件清零）
0 = RB7:RB4 引脚状态没有变化

注 1： 不匹配的情况会不断将此位置 1。对 PORTB 进行读操作，将结束不匹配的情况，从而将该位清零。

寄存器 9-2：　　INTCON2：中断控制寄存器 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	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 上拉功能
0 = 按各个端口锁存值使能 PORTB 上拉功能

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

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

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

bit 3　**INTEDG3**：外部中断 3 边沿选择位
1 = 上升沿中断
0 = 下降沿中断

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

bit 1　**INT3IP**：INT3 外部中断优先级位
1 = 高优先级
0 = 低优先级

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：　当有中断条件产生时，无论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之前，确保先将该中断标志位清零。这个功能可用软件进行查询。

© 2007 Microchip Technology Inc.

寄存器 9-3：　　INTCON3：中断控制寄存器 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	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　**INT3IE**：INT3 外部中断允许位
1 = 允许 INT3 外部中断
0 = 禁止 INT3 外部中断

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　**INT3IF**：INT3 外部中断标志位
1 = 发生了 INT3 外部中断（必须用软件清零）
0 = 未发生 INT2 外部中断

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：　当有中断条件产生时，无论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之前，确保先将该中断标志位清零。这个功能可用软件进行查询。

9.2 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 3 个外设中断请求（标志）寄存器（PIR1、PIR2 和 PIR3）。

注 1： 当有中断条件产生时，无论相应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之前，确保先将该中断标志位清零；同时在响应该中断后，也应该将该中断标志位清零。

寄存器 9-4： PIR1：外设中断请求 （标志）寄存器 1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **PMPIF**：并行从端口读 / 写中断标志位
1 = 发生了读或写操作 （必须用软件清零）
0 = 未发生读或写操作

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

bit 5 **RC1IF**：EUSART1 接收中断标志位
1 = EUSART1 接收缓冲器 RCREG1 已满 （当读 RCREG1 时清零）
0 = EUSART1 接收缓冲器为空

bit 4 **TX1IF**：EUSART1 发送中断标志位
1 = EUSART1 发送缓冲器 TXREG1 为空 （当写 TXREG1 时清零）
0 = EUSART1 发送缓冲器已满

bit 3 **SSP1IF**：MSSP1 中断标志位
1 = 完成发送 / 接收 （必须用软件清零）
0 = 等待发送 / 接收

bit 2 **CCP1IF**：ECCP1 中断标志位
捕捉模式：
1 = 发生了 TMR1/TMR3 寄存器捕捉（必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器捕捉
比较模式：
1 = 发生了 TMR1/TMR3 寄存器比较匹配 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器比较匹配
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 匹配（必须用软件清零）
0 = TMR2 与 PR2 不匹配

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

© 2007 Microchip Technology Inc.

寄存器 9-5：　　PIR2：外设中断请求 （标志）寄存器 2

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　**OSCFIF**：振荡器故障中断标志位
1 = 器件振荡器发生故障，改由 INTOSC 提供时钟输入 （必须用软件清零）
0 = 器件时钟正常运行

bit 6　**CM2IF**：比较器 2 中断标志位
1 = 比较器输入已改变 （必须用软件清零）
0 = 比较器输入未改变

bit 5　**CM1IF**：比较器 1 中断标志位
1 = 比较器输入已改变 （必须用软件清零）
0 = 比较器输入未改变

bit 4　**未实现**：读为 0

bit 3　**BCL1IF**：总线冲突中断标志位 （MSSP1 模块）
1 = 发生了总线冲突 （必须用软件清零）
0 = 未发生总线冲突

bit 2　**LVDIF**：低压检测中断标志位
1 = 产生了低压条件 （必须用软件清零）
0 = V_{DDCORE} 未降到低压跳变点下 （约 2.45V）

bit 1　**TMR3IF**：TMR3 溢出中断标志位
1 = TMR3 寄存器溢出 （必须用软件清零）
0 = TMR3 寄存器未溢出

bit 0　**CCP2IF**：ECCP2 中断标志位
捕捉模式：
1 = 发生了 TMR1/TMR3 寄存器捕捉 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器捕捉
比较模式：
1 = 发生了 TMR1/TMR3 寄存器的比较匹配 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器的比较匹配
PWM 模式：
在此模式下未使用。

寄存器 9-6：　　PIR3：外设中断请求 （标志）寄存器 3

R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　**SSP2IF**：MSSP2 中断标志位
1 = 完成发送 / 接收 （必须用软件清零）
0 = 等待发送 / 接收

bit 6　**BCL2IF**：总线冲突中断标志位 （MSSP2 模块）
1 = 发生了总线冲突 （必须用软件清零）
0 = 未发生总线冲突

bit 5　**RC2IF**：EUSART2 接收中断标志位
1 = EUSART2 接收缓冲器 RCREG2 已满 （当读取 RCREG1 时清零）
0 = EUSART2 接收缓冲器为空

bit 4　**TX2IF**：EUSART2 发送中断标志位
1 = EUSART2 发送缓冲器 TXREG1 为空 （当写 TXREG1 时清零）
0 = EUSART2 发送缓冲器已满

bit 3　**TMR4IF**：TMR4 与 PR4 匹配中断标志位
1 = TMR4 与 PR4 发生了匹配（必须用软件清零）
0 = TMR4 与 PR4 未发生匹配

bit 2　**CCP5IF**：CCP5 中断标志位
捕捉模式：
1 = 发生了 TMR1/TMR3 寄存器捕捉 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器捕捉
比较模式：
1 = 发生了 TMR1/TMR3 寄存器的比较匹配 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器的比较匹配
PWM 模式：
在此模式下未使用。

bit 1　**CCP4IF**：CCP4 中断标志位
捕捉模式：
1 = 发生了 TMR1/TMR3 寄存器捕捉 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器捕捉
比较模式：
1 = 发生了 TMR1/TMR3 寄存器的比较匹配 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器比较匹配
PWM 模式：
在此模式下未使用。

bit 0　**CCP3IF**：ECCP3 中断标志位
捕捉模式：
1 = 发生了 TMR1/TMR3 寄存器捕捉 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器捕捉
比较模式：
1 = 发生了 TMR1/TMR3 寄存器的比较匹配 （必须用软件清零）
0 = 未发生 TMR1/TMR3 寄存器的比较匹配
PWM 模式：
在此模式下未使用。

© 2007 Microchip Technology Inc.

9.3 PIE 寄存器

PIE 寄存器包含各外设的中断允许位。根据外设中断源的数量，有 3 个外设中断允许寄存器（PIE1、PIE2 和 PIE3）。当 IPEN = 0 时，要允许任何外设中断就必须将 PEIE 位置 1。

寄存器 9-7： PIE1：外设中断允许寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIE	ADIE	RC1IE	TX1IE	SSP1IF	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **PMPIE**：并行主端口读 / 写中断允许位
1 = 允许 PM 读 / 写中断
0 = 禁止 PM 读 / 写中断

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 **RC1IE**：EUSART1 接收中断允许位
1 = 允许 EUSART1 接收中断
0 = 禁止 EUSART1 接收中断

bit 4 **TX1IE**：EUSART1 发送中断允许位
1 = 允许 EUSART1 发送中断
0 = 禁止 EUSART1 发送中断

bit 3 **SSP1IE**：MSSP1 中断允许位
1 = 允许 MSSP1 中断
0 = 禁止 MSSP1 中断

bit 2 **CCP1IE**：ECCP1 中断允许位
1 = 允许 ECCP1 中断
0 = 禁止 ECCP1 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

寄存器 9-8：　　PIE2：外设中断允许寄存器 2

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　**OSCFIE**：振荡器故障中断允许位
1 = 允许
0 = 禁止

bit 6　**CM2IE**：比较器 2 中断标志位
1 = 允许
0 = 禁止

bit 5　**CM1IE**：比较器 1 中断允许位
1 = 允许
0 = 禁止

bit 4　**未实现**：读为 0

bit 3　**BCL1IE**：总线冲突中断允许位 （MSSP1 模块）
1= 允许
0= 禁止

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

bit 1　**TMR3IE**：TMR3 溢出中断允许位
1 = 允许
0 = 禁止

bit 0　**CCP2IE**：ECCP2 中断允许位
1 = 允许
0 = 禁止

© 2007 Microchip Technology Inc.

寄存器 9-9：　　PIE3：外设中断允许寄存器 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　**SSP2IE**：MSSP2 中断允许位
1 = 允许
0 = 禁止

bit 6　**BCL2IE**：总线冲突中断允许位（MSSP2 模块）
1 = 允许
0 = 禁止

bit 5　**RC2IE**：EUSART2 接收中断允许位
1 = 允许
0 = 禁止

bit 4　**TX2IE**：EUSART2 发送中断允许位
1 = 允许
0 = 禁止

bit 3　**TMR4IE**：TMR4 与 PR4 匹配中断允许位
1 = 允许
0 = 禁止

bit 2　**CCP5IE**：CCP5 中断允许位
1 = 允许
0 = 禁止

bit 1　**CCP4IE**：CCP4 中断允许位
1 = 允许
0 = 禁止

bit 0　**CCP3IE**：ECCP3 中断允许位
1 = 允许
0 = 禁止

9.4 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 3 个外设中断优先级寄存器（IPR1、IPR2 和 IPR3）。使用优先级位需要将中断优先级使能（IPEN）位置 1。

寄存器 9-10： IPR1：外设中断优先级寄存器 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **PMPIP**：并行主端口读 / 写中断优先级位
1 = 高优先级
0 = 低优先级

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 **RC1IP**：EUSART1 接收中断优先级位
1 = 高优先级
0 = 低优先级

bit 4 **TX1IP**：EUSART1 发送中断优先级位
1 = 高优先级
0 = 低优先级

bit 3 **SSP1IP**：MSSP1 中断优先级位
1 = 高优先级
0 = 低优先级

bit 2 **CCP1IP**：ECCP1 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 9-11：　　IPR2：外设中断优先级寄存器 2

R/W-1	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　**OSCFIP**：振荡器故障中断优先级位
1 = 高优先级
0 = 低优先级

bit 6　**CM2IP**：模拟比较器 2 中断优先级位
1 = 高优先级
0 = 低优先级

bit 5　**C12IP**：模拟比较器 1 中断优先级位
1 = 高优先级
0 = 低优先级

bit 4　**未实现**：读为 0

bit 3　**BCL1IP**：总线冲突中断优先级位 （MSSP1 模块）
1 = 高优先级
0 = 低优先级

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

bit 1　**TMR3IP**：TMR3 溢出中断优先级位
1 = 高优先级
0 = 低优先级

bit 0　**CCP2IP**：ECCP2 中断优先级位
1 = 高优先级
0 = 低优先级

寄存器 9-12： IPR3：外设中断优先级寄存器 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **SSP2IP**：MSSP2 中断优先级位
1 = 高优先级
0 = 低优先级

bit 6 **BCL2IP**：总线冲突中断优先级位 （MSSP2 模块）
1 = 高优先级
0 = 低优先级

bit 5 **RC2IP**：EUSART2 接收中断优先级位
1 = 高优先级
0 = 低优先级

bit 4 **TX2IP**：EUSART2 发送中断优先级位
1 = 高优先级
0 = 低优先级

bit 3 **TMR4IE**：TMR4 与 PR4 匹配中断优先级位
1 = 高优先级
0 = 低优先级

bit 2 **CCP5I**：CCP5 中断优先级位
1 = 高优先级
0 = 低优先级

bit 1 **CCP4IP**：CCP4 中断优先级位
1 = 高优先级
0 = 低优先级

bit 0 **CCP3IP**：ECCP3 中断优先级位
1 = 高优先级
0 = 低优先级

© 2007 Microchip Technology Inc.

9.5 RCON 寄存器

RCON 寄存器包含几个位，可以用它们来确定器件上次复位或从空闲或休眠模式被唤醒的原因。RCON还包含一位可以使能中断优先级 （IPEN）。

寄存器 9-13： RCON：复位控制寄存器

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

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	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 **未实现**：读为 0

bit 5 **$\overline{CM}$**：配置不匹配标志位
关于位操作的详细信息，请参见寄存器 4-1。

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

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

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

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

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

9.6 INTx 引脚中断

RB0/INT0、RB1/INT1、RB2/INT2 和 RB3/INT3 引脚上的外部中断是边沿触发的。如果 INTCON2 寄存器中对应的 INTEDGx 位被置 1 （= 1），则该中断为上升沿触发；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，对应的标志位 INTxIF 就被置 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 INTxIF 标志位清零。

如果 INTxIE 位在进入功耗管理模式前被置 1，则所有外部中断（INT0、INT1、INT2 和 INT3）均能把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跳转到中断向量处执行程序。

INT1、INT2 和 INT3 的中断优先级由中断优先级位 INT1IP（INTCON3<6>）、INT2IP（INTCON3<7>）和 INT3IP（INTCON2<1>）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。

9.7 TMR0 中断

在 8 位模式下（默认设置），TMR0 寄存器的溢出（FFh → 00h）将会使 TMR0IF 标志位置 1。在 16 位模式下，TMR0H:TMR0L 这对寄存器溢出（FFFFh → 0000h）将会使 TMR0IF 置 1。将允许位 TMR0IE（INTCON<5>）置 1 或清零可允许或禁止该中断。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 12.0节“Timer0 模块”**。

9.8 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。将允许位 RBIE（INTCON<3>）置 1 或清零可允许或禁止该中断。PORTB 电平变化的中断优先级由中断优先级位 RBIP （INTCON2<0>）中的值决定。

9.9 中断的现场保护

在中断期间，PC 的返回地址被压入堆栈。另外，将 WREG、STATUS 以及 BSR 寄存器的值压入快速返回堆栈。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”**），则用户可能需要在进入中断服务程序前，保存 WREG、STATUS 以及 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 9-1 所示为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 9-1： 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
```


© 2007 Microchip Technology Inc.

10.0 I/O 端口

根据选定的器件和使能的功能的不同，最多有 9 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。通常，当外设使能时，其复用的引脚就无法作为通用 I/O 引脚使用。

每个端口都有 3 个映射的存储寄存器进行工作：

- 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 （输出锁存寄存器）

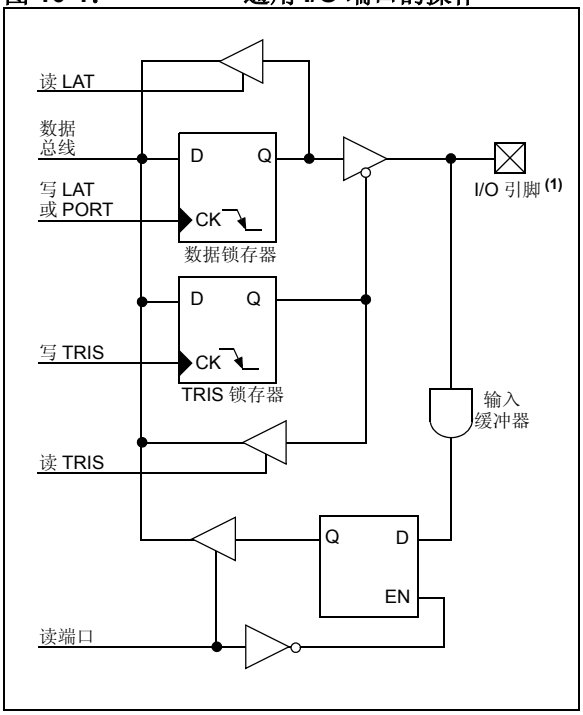
读 PORT 寄存器即是读引脚的当前状态，而写 PORT 寄存器则是写入输出锁存 （LAT）寄存器。

将 TRIS 位置 1 （= 1）可以让相应的 PORT 引脚作为输入引脚 （即，将相应的输出驱动器置于高阻态）。清零 TRIS 位 （= 0）可以让相应的 PORT 引脚作为输出引脚 （即，将相应的 LAT 位的内容置于所选择的引脚上）。

在对 I/O 引脚驱动值进行读－修改－写的操作时会用到输出锁存器 （LAT 寄存器）。LAT 寄存器上的读 - 修改 - 写操作将读写 PORT 寄存器的锁存输出值。

图 10-1 所示为通用 I/O 端口的简化模型，没有给出到其他外设的接口。

图 10-1： 通用 I/O 端口的操作



10.1 I/O 端口引脚功能

在开发应用程序时，必须考虑到端口引脚的功能。某些引脚上的输出驱动电平强度比其他引脚要高。同样，某些引脚可以接受高于 VDD 的输入电平。

10.1.1 输入引脚和电压注意事项

用作器件输入的引脚的电压容差取决于该引脚的输入功能。仅用作数字输入的引脚能够接受高达 5.5V 的 DC 电压，这个电平值是数字逻辑电路的典型电平值。相反，还具有任何模拟输入功能（例如 A/D 和比较器输入）的引脚只可承受最高为 VDD 的电压值。应该避免在这些引脚上施加超过 VDD 的电压。

表 10-1 汇总了这些输入功能。更多详情，请参见**第 27.0 节 “电气规范”**。

表 10-1： 输入电平

端口或引脚	可接受的输入电平	说明
PORTA<7:0>	VDD	只能承受 VDD 输入电平。
PORTC<1:0>		
PORTF<6:1>		
PORTH<7:4>(1)		
PORTB<7:0>	5.5V	可承受高于 VDD 的输入电平，可用于大部分标准逻辑电路。
PORTC<7:2>		
PORTD<7:0>		
PORTE<7:0>		
PORTF<7>		
PORTG<4:0>		
PORTH<3:0>(1)		
PORTJ<7:0>(1)		

注 1： 此端口在 64 引脚器件上不可用。

10.1.2 引脚输出驱动电平

用作数字 I/O 时，各引脚组的输出引脚驱动能力是不同的，这样可以满足不同应用的需求。通常，根据驱动能力可将输出引脚划分为三类。

PORTB 和 PORTC 以及 PORTA<7:6> 都是为驱动较高的电流负载（例如 LED）而设计的。PORTD、PORTE 和 PORTJ 能够驱动与外部存储器器件有关的数字电路；它们还可以驱动具有较低电流要求的 LED。PORTF、PORTG 和 PORTH，以及 PORTA<5:0> 具有最低的驱动电平，但是能够驱动带有高输入阻抗的常规数字电路负载。

表10-2汇总了端口输出功能。更多详情，请参见**第 27.0 节 “电气规范”**中的 **“绝对最大值”**。

表 10-2： 输出驱动电平

端口	驱动电平	说明
PORTA	最低	用作指示作用。
PORTF		
PORTG		
PORTH(1)		
PORTD	中等	足够高的驱动电平，用于与外部存储器接口以及指示作用。
PORTE		
PORTJ(1)		
PORTB	高	适合直接输出 LED 驱动电平。
PORTC		

注 1： 此端口在 64 引脚器件上不可用。

10.1.3 上拉配置

有四种I/O端口（PORTB、PORTD、PORTE和PORTJ）在所有引脚上实现了可配置的弱上拉功能。它们属于内部上拉功能，允许将浮动的数字输入信号拉至一致的电平，而无需使用外部电阻。

每个端口都有一个单独的位来使能上拉功能：PORTB 的 RBPU（INTCON2<7>），其他端口的为：RDPU、REPU 和 RJPU （PORTG<7:5>）。

10.1.4 漏极开路输出

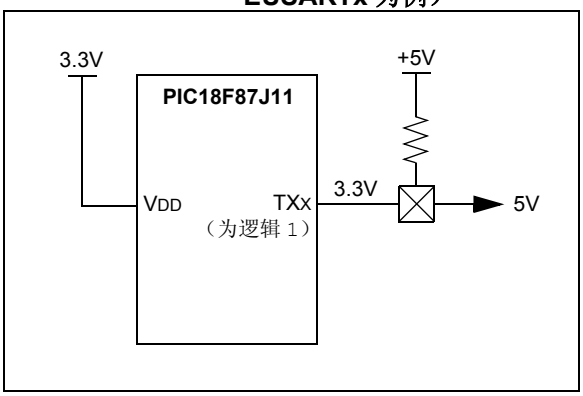
部分外设的输出引脚也配备有可配置的漏极开路输出选项。这将允许这些外设与工作在高电平下的外部数字逻辑电路进行通信，而无需使用电平转换器。

可在与 EUSART、 MSSP 模块 （在 SPI 模式下）以及 CCP 和 ECCP 模块的数据和时钟输出密切相关的端口引脚上实现漏极开路选项 。可通过在 ODCON 寄存器（寄存器 10-1、寄存器 10-2 和寄存器 10-3）中将相应模块的漏极开路控制位置 1 来使能漏极开路选项。此选项的配置将针对各个端口 （这些外设复用了这些端口）进行更加详细的讨论。

所有ODCON寄存器都位于SFR配置空间中，并与Timer1寄存器共享相同的 SFR 地址（更多详情，请参见**第 5.3.4.1 节 “共享地址 SFR”**）。通过将 ADSHR 位（WDTCON<4>）置 1 可访问 ODCON 寄存器。

当需要漏极开路选项时，输出引脚还必须通过一个由用户提供的外部上拉电阻连接到高电平，仅数字引脚上可达到 5V （图 10-2）。当输出数字逻辑高电平信号时，它必须上拉至高电平。

图 10-2： 使用漏极开路输出 （以 EUSARTx 为例）



10.1.5 TTL 输入缓冲器选项

许多数字 I/O 端口都使用施密特触发器（ST）输入缓冲器。虽然此种类型的缓冲器与很多类型的输入兼容得很好，但是一些应用可能还需要 TTL 电平信号来与外部逻辑器件相连。 EMB 和并行主端口 （Parallel Master Port， PMP）就是属于这样的情况，它们极可能需要连接到 TTL 电平逻辑或存储器器件。

PMP 的输入可通过 PADCFG1 寄存器中的 PMPTTL 位配置为 TTL 缓冲器 （寄存器 10-4）。将此位置 1 可配置 PMP 的所有数据和控制输入引脚，以使用 TTL 缓冲器。默认情况下，这些PMP输入使用端口的ST缓冲器。

与 ODCON 寄存器一样，PADCFG1 寄存器也位于 SFR 配置空间中；它与 TMR2 寄存器共享同一存储器地址。通过将 ADSHR 位 （WDTCON<4>）置 1 可访问 PADCFG1 寄存器。

© 2007 Microchip Technology Inc.

寄存器 10-1： ODCON1：外设漏极开路控制寄存器 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-5 **未实现：**读为 0

bit 4-3 **CCP5OD:CCP4OD：**CCPx 漏极开路输出使能位
1 = 使能 CCPx 引脚 （捕捉 /PWM 模式）上的漏极开路输出
0 = 禁止漏极开路输出

bit 2-0 **ECCP3OD:ECCP1OD：**ECCPx 漏极开路输出使能位
1 = 使能 ECCPx 引脚 （捕捉模式）上的漏极开路输出
0 = 禁止漏极开路输出

寄存器 10-2： ODCON2：外设漏极开路控制寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	U2OD	U1OD
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-2 **未实现：**读为 0

bit 1-0 **U2OD:U1OD：**EUSARTx 漏极开路输出使能位
1 = 使能 TXx 引脚上的漏极开路输出
0 = 禁止漏极开路输出

寄存器 10-3： ODCON3：外设漏极开路控制寄存器 3

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPI2OD	SPI1OD
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-2 **未实现：**读为 0

bit 1-0 **SPI2OD:SPI1OD：**SPI 漏极开路输出使能位
1 = 使能 SDOx 引脚上的漏极开路输出
0 = 禁止漏极开路输出

寄存器 10-4：　PADCFG1：I/O 填充配置控制寄存器

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

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-1　**未实现：**读为 0

bit 0　**PMPTTL：**PMP 模块 TTL 输入缓冲器选择位
1 = PMP 模块使用 TTL 输入缓冲器
0 = PMP 模块使用施密特输入缓冲器

10.2　PORTA、 TRISA 和 LATA 寄存器

PORTA 是 8 位宽的双向端口。它可用作 6 位或 7 位端口，这取决于所选的振荡器模式。相应的数据方向寄存器和输出锁存寄存器分别为 TRISA 和 LATA。

RA4 引脚与 Timer0 模块时钟输入引脚复用成为了 RA4/T0CKI 引脚；此引脚可被复用为并行主端口数据引脚（在 80 引脚器件中）。其他 PORTA 引脚与模拟 VREF+ 和 VREF- 输入引脚复用。 通过清零或置 1 ANCON0 寄存器中相应的 PCFG 控制位，可选择将 RA<5,3:0> 引脚作为 A/D 转换器输入引脚工作。

注　1：RA5（RA5/PMD4/AN4）在所有器件中被复用为模拟输入引脚，而在 80 引脚器件中被复用为并行主端口数据引脚。

2：RA5和RA3:RA0在任何复位时被配置为模拟输入引脚并读为 0。RA4 被配置为数字输入引脚。

RA4/T0CKI 引脚是施密特触发器输入。而其他的 PORTA 引脚具有 TTL 输入电平和完全的 CMOS 输出驱动器。

即使在 PORTA 引脚被用作模拟输入的时候，TRISA 寄存器仍然控制 PORTA 引脚的方向。在将它们用作模拟输入时，用户必须确保 TRISA 寄存器中的位保持为置位状态。

OSC2/CLKO/RA6 和 OSC1/CLKI/RA7 通常用作到外部（主）振荡器电路（HS 和 HSPLL 振荡器模式）或外部时钟输入 （EC 和 ECPLL 振荡器模式）的外部电路连接。在这些情况下，RA6 和 RA7 不可用作数字 I/O，并且其相应的 TRIS 和 LAT 位读为 0。

对于 INTIO 和 INTPLL 振荡器模式 （FOSC2 配置位为 0），根据所选的振荡器模式，RA7或RA6和RA7将自动用作数字 I/O。在这些情况下，当 RA6 未被配置为数字 I/O 时，它将提供一个 FOSC/4 的时钟输出。表 10-3 列出了按振荡器模式划分的 RA6 和 RA7 可能有的配置。对于这些引脚，当将其配置为 I/O 引脚时仅定义了相应的 PORTA、 TRISA 和 LATA 位。

表 10-3：　INTIO 和 INTPLL 模式下 RA7:RA6 的功能

振荡器模式（FOSC2:FOSC0 配置）	RA6	RA7
INTPLL1 （011）	CLKO	I/O
INTPLL2 （010）	I/O	I/O
INTIO1 （001）	CLKO	I/O
INTIO2 （000）	I/O	I/O

图注：　CLKO = FOSC/4 时钟输出；
I/O = 数字端口。

例 10-1：　初始化 PORTA

```
CLRF   PORTA         ; Initialize PORTA by
                     ; clearing output
                     ; data latches
CLRF   LATA          ; Alternate method to
                     ; clear  data latches
BSF    WDTCON,ADSHR  ; Enable write/read to
                     ; the shared SFR
MOVLW  1Fh           ; Configure A/D
MOVWF  ANCON0        ; for digital inputs
BCF    WDTCON,ADSHR  ; Disable write/read
                     ; to the shared SFR
MOVLW  0CFh          ; Value used to
                     ; initialize
                     ; data direction
MOVWF  TRISA         ; Set RA<3:0> as inputs,
                     ; RA<5:4> as outputs
```


© 2007 Microchip Technology Inc.

表 10-4： PORTA 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RA0/AN0	RA0	0	O	DIG	LATA<0> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<0> 数据输入；使能模拟输入时被禁止。
	AN0	1	I	ANA	A/D 输入通道 0。 POR 时默认为输入配置；不影响数字输出。
RA1/AN1	RA1	0	O	DIG	LATA<1> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<1> 数据输入；使能模拟输入时被禁止。
	AN1	1	I	ANA	A/D 输入通道 1。 POR 时默认为输入配置；不影响数字输出。
RA2/AN2/VREF-	RA2	0	O	DIG	LATA<2> 数据输出；不受模拟输入影响。当使能 CVREF 输出时禁止。
		1	I	TTL	PORTA<2> 数据输入。当使能模拟功能时禁止；当使能 CVREF 输出时禁止。
	AN2	1	I	ANA	A/D 输入通道 2。 POR 时默认为输入配置；不受模拟输出的影响。
	VREF-	1	I	ANA	A/D 低参考电压输入。
RA3/AN3/VREF+	RA3	0	O	DIG	LATA<3> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<3> 数据输入；使能模拟输入时禁止。
	AN3	1	I	ANA	A/D 输入通道 3。 POR 时默认为输入配置。
	VREF+	1	I	ANA	A/D 高参考电压输入。
RA4/PMD5/ T0CKI/	RA4	0	O	DIG	LATA<4> 数据输出。
		1	I	ST	PORTA<4> 数据输入；发生 POR 时为默认配置。
	PMD5(1)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输出。
	T0CKI	x	I	ST	Timer0 时钟输入。
RA5/PMD4/AN4	RA5	0	O	DIG	LATA<5> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<5> 数据输入；使能模拟输入时禁止。
	PMD4(1)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输出。
	AN4	1	I	ANA	A/D 输入通道 4。发生 POR 时为默认配置。
OSC2/CLKO/ RA6	OSC2	x	O	ANA	主振荡器反馈输出连接 （HS 和 HSPLL 模式）。
	CLKO	x	O	DIG	系统周期时钟输出 FOSC/4 （EC、 ECPLL、 INTIO1 和 INTPLL1 振荡器模式下）。
	RA6	0	O	DIG	LATA<6> 数据输出；在 FOSC2 配置位置 1 时禁止。
		1	I	TTL	PORTA<6> 数据输入；在 FOSC2 配置位置 1 时禁止。
OSC1/CLKI/ RA7	OSC1	x	I	ANA	主振荡器输入连接 （HS 和 HSPLL 模式）。
	CLKI	x	I	ANA	主外部时钟源输入 （EC 和 ECPLL 模式）。
	RA7	0	O	DIG	LATA<7> 数据输出；在 FOSC2 配置位置 1 时禁止。
		1	I	TTL	PORTA<7> 数据输入；在 FOSC2 配置位置 1 时禁止。

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 = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 PMPMX 配置位为 0 时为 PMP 的备用分配，仅在 80 引脚器件上可用。

表 10-5：　　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	复位值所在的页
PORTA	RA7(1)	RA6(1)	RA5	RA4	RA3	RA2	RA1	RA0	60
LATA	LATA7(1)	LATA6(1)	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	59
TRISA	TRISA7(1)	TRISA6(1)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	59
ANCON0(2)	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	58

图注：　— = 未实现位，读为 0。 PORTA 不使用阴影单元。

注　1： 仅在特定的振荡器模式下实现 （FOSC2 配置位 = 0）；否则读为 0。

2： 配置 SFR 的地址与默认 SFR 重叠；仅在 WDTCON<4> = 1 时才可访问配置 SFR。

© 2007 Microchip Technology Inc.

10.3 PORTB、TRISB 和 LATB 寄存器

PORTB 是 8 位宽的双向端口。相应的数据方向寄存器为 TRISB。PORTB 上的所有引脚都仅仅是数字的，并且可以接受高达 5.5V 的电压。

PORTB 的每个引脚都有内部弱上拉电路。单个控制位可以开启所有上拉电路。可以通过将 RBPU 位清零（INTCON2<7>）来完成此操作。当 PORTB 端口的引脚配置为输出时，其弱上拉电路会自动切断。此弱上拉电路在上电复位时被禁止。

四个 PORTB 引脚（RB7:RB4）有电平变化中断功能。只有配置为输入的引脚会导致此类中断发生（即，当 RB7:RB4 中的任何一个引脚被配置为输出时，该引脚不再具有电平变化的中断功能）。当前 RB7:RB4 输入引脚上的电平与上次读 PORTB 时锁存的旧值进行比较。RB7:RB4 输出的“不匹配”值一起进行或运算，产生 RB 端口电平变化中断，使用标志位 RBIF（INTCON<0>）表示。

此中断可以将器件从功耗管理模式唤醒。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（使用 MOVFF（ANY），PORTB 指令时除外）。这将结束不匹配状态。
b) 清零标志位 RBIF。

不匹配状态将会持续地将 RBIF 标志位置 1。而读 PORTB 将结束不匹配状态，并且允许将 RBIF 标志位清零。

对于按键唤醒以及仅使用 PORTB 的电平变化触发中断功能的操作，建议使用电平变化中断功能来实现。在使用电平变化触发中断功能时，建议不要查询 PORTB 的状态。

对于 80 引脚的器件来说，可将 RB3 配置为 ECCP2 模块的备用外设引脚并且通过清零 CCP2MX 配置位将 PWM 的输出电流增加为 2A。这种配置仅应用于在扩展单片机模式下工作的 80 引脚器件。如果此器件处于单片机模式，ECCP2 的备用分配为 RE7。根据其他 ECCP2 配置，用户必须确保为要进行的操作将 TRISB<3> 位相应的置 1。RB1、RB2、RB3、RB4 和 RB5 端口都与并行主端口地址复用。

例 10-2： 初始化 PORTB

```
CLRF    PORTB   ; Initialize PORTB by
                ; clearing output
                ; data latches
CLRF    LATB    ; Alternate method to clear
                ; output data latches
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 10-6： PORTB 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RB0/INT0/FLT0	RB0	0	O	DIG	LATB<0> 数据输出。
		1	I	TTL	PORTB<0> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	INT0	1	I	ST	外部中断 0 输入。
	FLT0	1	I	ST	增强型 PWM 错误输入 （ECCP1 模块）；用软件使能。
RB1/INT1/ PMA4	RB1	0	O	DIG	LATB<1> 数据输出。
		1	I	TTL	PORTB<1> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	INT1	1	I	ST	外部中断 1 输入。
	PMA4	x	O	—	并行主端口地址输出。
RB2/INT2/ PMA3	RB2	0	O	DIG	LATB<2> 数据输出。
		1	I	TTL	PORTB<2> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	INT2	1	I	ST	外部中断 2 输入。
	PMA3	x	O	—	并行主端口地址输出。
RB3/INT3/ PMA2/ECCP2/ P2A	RB3	0	O	DIG	LATB<3> 数据输出。
		1	I	TTL	PORTB<3> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	INT3	1	I	ST	外部中断 3 输入。
	PMA2	x	O	—	并行主端口地址输出。
	ECCP2(1)	0	O	DIG	ECCP2 比较输出和 CCP2 PWM 输出；优先于端口数据。
		1	I	ST	ECCP2 捕捉输入。
	P2A(1)	0	O	DIG	ECCP2 增强型 PWM 输出，通道 A。可能在增强型 PWM 关闭事件期间被配置为三态。优先于端口数据。
RB4/KBI0/ PMA1	RB4	0	O	DIG	LATB<4> 数据输出。
		1	I	TTL	PORTB<4> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	KBI0		I	TTL	引脚电平变化时中断。
	PMA1	x	O	—	并行主端口地址输出。
RB5/KBI1/ PMA0	RB5	0	O	DIG	LATB<5> 数据输出。
		1	I	TTL	PORTB<5> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	KBI1		I	TTL	引脚电平变化时中断。
	PMA0	x	O	—	并行主端口地址输出。
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> 数据输出。
		1	I	TTL	PORTB<6> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	KBI2	1	I	TTL	引脚电平变化时中断。
	PGC	x	I	ST	串行执行 ICSP （ICSP™）的时钟输入和 ICD 操作 (2)。
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> 数据输出。
		1	I	TTL	PORTB<7> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时为弱上拉电路。
	KBI3	1	I	TTL	引脚电平变化时中断。
	PGD	x	O	DIG	串行执行 ICSP 的数据输出和 ICD 操作 (2)。
		x	I	ST	串行执行 ICSP 的数据输入和 ICD 操作 (2)。

图注： O = 输出， I = 输入， DIG = 数字输出， ST = 施密特缓冲器输入， TTL = TTL 缓冲器输入。
x = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 CCP2MX 配置位被清零时，为 ECCP2/P2A 的备用配置 （扩展单片机模式，仅用于 80 引脚器件）。默认配置为 RC1。

2： 当使能 ICSP™ 或 ICD 时，所有其他的引脚功能均被禁止。

© 2007 Microchip Technology Inc.

表 10-7： 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	复位值所在的页
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	60
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	59
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	59
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	56
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	56
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	56

图注： PORTB 模块未使用阴影单元。

10.4 PORTC、 TRISC 和 LATC 寄存器

PORTC 是 8 位宽的双向端口。PORTC 引脚（RC2 到 RC7）只能作为数字引脚并且可以接受高达 5.5V 的输入电压。

PORTC 与 ECCP、 MSSP 和 EUSART 外设功能复用（表 10-8）。这些引脚具有施密特触发器输入缓冲器。ECCP、SPI 和 EUSART 的引脚还可被配置为漏极开路输出，无论这些功能是否有效。可通过将 ODCON 寄存器中的 SPIxOD、ECCPxOD 和 UxOD 控制位置 1 来选择漏极开路配置（更多详情，请参见**第 10.1.3 节 “上拉配置”**）。

RC1 通常被配置为 ECCP2 的默认外设引脚。ECCP2 的分配由配置位 CCP2MX 控制（默认状态，CCP2MX = 1）。

当外设功能使能时，应考虑到每个 PORTC 引脚的 TRIS 位方向设置。有些外设使能时，会改写相应引脚的 TRIS 位方向设置而将引脚定义为输出，而另一些外设使能时，也会改写相应引脚的 TRIS 位方向设置，但却将引脚定义为输入。用户应该参考相应的外设章节来正确设置 TRIS 位。

注：	这些引脚在任何器件复位时都被配置为数字输入引脚。

外设改写会影响 TRISC 寄存器的内容。尽管外设器件可能会改写一个或多个引脚，读 TRISC 总是会返回当前内容。

例 10-3： 初始化 PORTC

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


© 2007 Microchip Technology Inc.

表 10-8： PORTC 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RC0/T1OSO/ T13CKI	RC0	0	O	DIG	LATC<0> 数据输出。
		1	I	ST	PORTC<0> 数据输入。
	T1OSO	x	O	ANA	Timer1 振荡器输出；当 Timer1 振荡器使能时被使能。禁止数字 I/O。
	T13CKI	1	I	ST	Timer1/Timer3 计数器输入。
RC1/T1OSI/ ECCP2/P2A	RC1	0	O	DIG	LATC<1> 数据输出。
		1	I	ST	PORTC<1> 数据输入。
	T1OSI	x	I	ANA	Timer1 振荡器输入；当 Timer1 振荡器使能时被使能。禁止数字 I/O。
	ECCP2(1)	0	O	DIG	ECCP2 比较输出和 ECCP2 PWM 输出；优先于端口数据。
		1	I	ST	ECCP2 捕捉输入。
	P2A(1)	0	O	DIG	ECCP2 增强型 PWM 输出，通道 A。可能在增强型 PWM 关闭事件期间被配置为三态。优先于端口数据。
RC2/ECCP1/ P1A	RC2	0	O	DIG	LATC<2> 数据输出。
		1	I	ST	PORTC<2> 数据输入。
	ECCP1	0	O	DIG	ECCP1 比较输出和 ECCP1 PWM 输出；优先于端口数据。
		1	I	ST	ECCP1 捕捉输入。
	P1A	0	O	DIG	ECCP1 增强型 PWM 输出，通道 A。可能在增强型 PWM 关闭事件期间被配置为三态。优先于端口数据。
RC3/SCK1/ SCL1	RC3	0	O	DIG	LATC<3> 数据输出。
		1	I	ST	PORTC<3> 数据输入。
	SCK1	0	O	DIG	SPI 时钟输出 （MSSP1 模块）；优先于端口数据。
		1	I	ST	SPI 时钟输入 （MSSP1 模块）。
	SCL1	0	O	DIG	I^2C™ 时钟输出 （MSSP1 模块）；优先于端口数据。
		1	I	ST	I^2C 时钟输入 （MSSP1 模块）；输入类型取决于模块设置。
RC4/SDI1/ SDA1	RC4	0	O	DIG	LATC<4> 数据输出。
		1	I	ST	PORTC<4> 数据输入。
	SDI1	1	I	ST	SPI 数据输入 （MSSP1 模块）。
	SDA1	1	O	DIG	I^2C 数据输出 （MSSP1 模块）；优先于端口数据
		1	I	ST	I^2C 数据输入 （MSSP1 模块）；输入类型取决于模块设置。
RC5/SDO1	RC5	0	O	DIG	LATC<5> 数据输出。
		1	I	ST	PORTC<5> 数据输入。
	SDO1	0	O	DIG	SPI 数据输出 （MSSP1 模块）；优先于端口数据。
RC6/TX1/CK1	RC6	0	O	DIG	LATC<6> 数据输出。
		1	I	ST	PORTC<6> 数据输入。
	TX1	1	O	DIG	同步串行数据输出 （EUSART1 模块）；优先于端口数据。
	CK1	1	O	DIG	同步串行数据输入 （EUSART1 模块）。用户必须配置为输入。
		0	O	ST	同步串行时钟输入 （EUSART1 模块）。
RC7/RX1/DT1	RC7	1	I	DIG	LATC<7> 数据输出。
		x	O	ST	PORTC<7> 数据输入。
	RX1	1	I	ST	异步串行接收数据输入 （EUSART1 模块）。
	DT1	0	O	DIG	同步串行数据输出 （EUSART1 模块）；优先于端口数据。
		1	I	ST	同步串行数据输入 （EUSART1 模块）。用户必须配置为输入。

图注： O = 输出， I = 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 = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 CCP2MX 配置位置 1 时，为 ECCP2/P2A 的默认分配。

表 10-9： 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	复位值所在的页
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	60
LATC	LATC7	LATBC6	LATC5	LATCB4	LATC3	LATC2	LATC1	LATC0	59
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	59

© 2007 Microchip Technology Inc.

10.5 PORTD、TRISD 和 LATD 寄存器

PORTD 是 8 位宽的双向端口。PORTD 上的所有引脚都仅是数字引脚，并且可以接受高达 5.5V 的电压。

PORTD 上的所有引脚都是使用施密特触发器输入缓冲器实现的。每个引脚都被单独地设置为输入或输出。

注： 这些引脚在任何器件复位时都被配置为数字输入引脚。

在 80 引脚的器件上，PORTD 与系统总线复用作为外部存储器接口的一部分。I/O 端口和其他功能只有在通过将 EBDIS 位（MEMCON<7>）位置 1 来禁止此接口后才可用。当使能接口时，PORTD 是复用的地址 / 数据总线（AD7:AD0）的低位字节。还将改写 TRISD 位。

PORTD 还与并行主端口数据引脚的数据功能复用。在此模式下，并行主端口优先于其他数字 I/O（但不优先于外部存储器总线）。当 PMPMX = 1 时这种复用才有效。当激活并行主端口时，输入缓冲器为 TTL。更多详情，请参见**第 11.0 节 “并行主端口”**。

PORTD 的每个引脚都有内部弱上拉电路。可以通过清零RDPU位（PORTG<7>）来关闭上拉电路。当PORTB端口的引脚配置为输出时，其弱上拉电路会自动关闭。在所有器件复位时上拉电路被禁止。

例 10-4： 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 10-10： PORTD 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RD0/AD0/ PMD0	RD0	0	O	DIG	LATD<0> 数据输出。
		1	I	ST	PORTD<0> 数据输入。
	AD0(2)	x	O	DIG	外部存储器接口，地址 / 数据 bit 0 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 0 输入 (1)。
	PMD0(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
RD1/AD1/ PMD1	RD1	0	O	DIG	LATD<1> 数据输出。
		1	I	ST	PORTD<1> 数据输入。
	AD1(2)	x	O	DIG	外部存储器接口，地址 / 数据 bit 1 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 1 输入 (1)。
	PMD1(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
RD2/AD2/ PMD2	RD2	0	O	DIG	LATD<2> 数据输出。
		1	I	ST	PORTD<2> 数据输入。
	AD2(2)	x	O	DIG	外部存储器接口，地址 / 数据 bit 2 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 2 输入 (1)。
	PMD2(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
RD3/AD3/ PMD3	RD3	0	O	DIG	LATD<3> 数据输出。
		1	I	ST	PORTD<3> 数据输入。
	AD3(2)	x	O	DIG	外部存储器接口，地址 / 数据 bit 3 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 3 输入 (1)。
	PMD3(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
RD4/AD4/ PMD4/SDO2	RD4	0	O	DIG	LATD<4> 数据输出。
		1	I	ST	PORTD<4> 数据输入。
	AD4(2)	x	O	DIG	外部存储器接口，地址 / 数据 bit 4 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 4 输入 (1)。
	PMD4(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
	SDO2	0	O	DIG	SPI 数据输出 （MSSP2 模块）；优先于端口数据。
RD5/AD5/ PMD5/SDI2/ SDA2	RD5	0	O	DIG	LATD<5> 数据输出。
		1	I	ST	PORTD<5> 数据输入。
	AD5(2)	x	O	DIG	外部存储器接口，地址 / 数据 bit 5 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 5 输入 (1)。
	PMD5(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
	SDI2	1	I	ST	SPI 数据输入 （MSSP2 块）。
	SDA2	1	O	DIG	I^2C™ 数据输出 （MSSP2 模块）；优先于端口数据。
		1	I	ST	I^2C 数据输入 （MSSP2 模块）；输入类型取决于模块设置。

图注： O = 输出， I = 输入， DIG = 数字输出， ST = 施密特缓冲器输入， TTL = TTL 缓冲器输入。
x = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 外部存储器接口 I/O 优先于所有其他数字和 PMP I/O。
2： 仅在 80 引脚器件上可用。
3： PMP 的默认配置 （PMPMX 配置位 = 1）。

© 2007 Microchip Technology Inc.

表 10-10： PORTD 功能 （续）

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RD6/AD6/ PMD6/SCK2/ SCL2	RD6	0	O	DIG	LATD<6> 数据输出。
		1	I	ST	PORTD<6> 数据输入。
	AD6(2)	x	O	DIG-3	外部存储器接口，地址 / 数据 bit 6 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 6 输入 (1)。
	PMD6(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
	SCK2	0	O	DIG	SPI 时钟输出 （MSSP2 模块）；优先于端口数据。
		1	I	ST	SPI 时钟输入 （MSSP2 模块）。
	SCL2	0	O	DIG	I²C™ 时钟输出 （MSSP2 模块）；优先于端口数据。
		1	I	ST	I²C 时钟输入 （MSSP2 块）；输入类型取决于模块设置。
RD7/AD7/ PMD7/$\overline{SS2}$	RD7	0	O	DIG	LATD<7> 数据输出。
		1	I	ST	PORTD<7> 数据输入。
	AD7(2)	x	O	DIG	外部存储器接口，地址 / 数据 bit 7 输出 (1)。
		x	I	TTL	外部存储器接口，数据 bit 7 输入 (1)。
	PMD7(3)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
	$\overline{SS2}$	x	I	TTL	MSSP2 模块的从动选择输入。

图注： O = 输出， I = 输入， DIG = 数字输出， ST = 施密特缓冲器输入， TTL = TTL 缓冲器输入。
x = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 外部存储器接口 I/O 优先于所有其他数字和 PMP I/O。
2： 仅在 80 引脚器件上可用。
3： PMP 的默认配置 （PMPMX 配置位 = 1）。

表 10-11： 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 所在的页
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	60
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	59
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	59
PORTG	RDPU	REPU	RJPU(1)	RG4	RG3	RG2	RG1	RG0	60

图注： PORTD 未使用阴影单元。

注 1： 在 64 引脚器件上未使用；读为 0。

10.6 PORTE、 TRISE 和 LATE 寄存器

PORTE 是 8 位宽的双向端口。PORTE 上的所有引脚都仅仅是数字的，并且可以接受高达 5.5V 的电压。

POTE 上的所有引脚都是使用施密特触发器输入缓冲器实现的。每个引脚都被单独地设置为输入或输出。

注：	这些引脚在任何器件复位时都被配置为数字输入引脚。

在 80 引脚的器件上，PORTE 与系统总线复用作为外部存储器接口的一部分。 I/O 端口和其他功能只有在通过将 EBDIS 位（MEMCON<7>）位置 1 而禁止此接口时可用。当使能接口时，PORTE 是复用的地址 / 数据总线（AD15:AD8）的高位字节。还将改写 TRISE 位。

PORTE 的每个引脚都有内部弱上拉电路。单个控制位可以关闭所有上拉电路。 可以通过清零 REPU 位（PORTG<6>）来关闭上拉电路。当 PORTB 端口的引脚配置为输出时，其弱上拉电路会自动关闭。发生任何器件复位时，弱上拉电路会被禁止。

PORTE 也可与以下功能复用：ECCP1 模块和 ECCP3 的增强型 PWM 输出 B 和 C 以及 ECCP2 模块的输出 B、C 和 D。对于所有器件来说，它们的默认分配是在 PORTE<6:0> 上。在 80 引脚器件上，由 ECCPMX 配置位控制 ECCP1 和 ECCP3 的输出复用。清零这些位将把 P1B/P1C 和 P3B/P3C 的输出重新分配到 PORTH。

对于在单片机模式工作的器件，可以将引脚 RE7 配置为 ECCP2 模块和增强型 PWM 输出 2A 的备用外设引脚。可以通过清零 CCP2MX 配置位来完成以上配置。

PORTE 还可与并行主端口地址线复用。当 PMPMX = 0 时， RE1 和 RE0 与控制信号 PMWR 和 PMRD 复用。

RE3 还可被配置为来自系统时钟的参考时钟输出（Reference Clock Output，REFO）。更多详情，请参见**第 2.6 节 “参考时钟输出”**。

例 10-5： 初始化 PORTE

```
CLRF    PORTE   ; Initialize PORTE by
                ; clearing output
                ; data latches
CLRF    LATE    ; Alternate method to clear
                ; output data latches
MOVLW   03h     ; Value used to initialize
                ; data direction
MOVWF   TRISE   ; Set RE<1:0> as inputs
                ; RE<7:2> as outputs
```


© 2007 Microchip Technology Inc.

表 10-12： PORTE 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RE0/AD8/ PMRD/P2D	RE0	0	O	DIG	LATE<0> 数据输出。
		1	I	ST	PORTE<0> 数据输入。
	AD8(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 8 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 8 输入 (2)。
	PMRD(5)	x	O	DIG	并行主端口读选通引脚。
		x	I	TTL	并行主端口读引脚。
	P2D	0	O	DIG	ECCP2 增强型 PWM 输出，通道 D；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
RE1/AD9/ PMWR/P2C	RE1	0	O	DIG	LATE<1> 数据输出。
		1	I	ST	PORTE<1> 数据输入。
	AD9(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 9 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 9 输入 (2)。
	PMWR(5)	x	O	DIG	并行主端口写选通引脚。
		x	I	TTL	并行主端口写引脚。
	P2C	0	O	DIG	ECCP2 增强型 PWM 输出，通道 C；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
RE2/AD10/ PMBE/P2B	RE2	0	O	DIG	LATE<2> 数据输出。
		1	I	ST	PORTE<2> 数据输入。
	AD10(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 10 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 10 输入 (2)。
	PMBE(5)	x	O	DIG	并行主端口字节使能。
	P2B	0	O	DIG	ECCP2 增强型 PWM 输出，通道 B；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
RE3/AD11/ PMA13/P3C/ REFO	RE3	0	O	DIG	LATE<3> 数据输出。
		1	I	ST	PORTE<3> 数据输入。
	AD11(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 11 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 11 输入 (2)。
	PMA13	x	O	DIG	并行主端口地址。
	P3C(1)	0	O	DIG	ECCP3 增强型 PWM 输出，通道 C；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
	REFO	x	O	DIG	参考输出时钟。
RE4/AD12/ PMA12/P3B	RE4	0	O	DIG	LATE<4> 数据输出。
		1	I	ST	PORTE<4> 数据输入。
	AD12(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 12 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 12 输入 (2)。
	PMA12	x	O	DIG	并行主端口地址。
	P3B(1)	0	O	DIG	ECCP3 增强型 PWM 输出，通道 B；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。

图注： O = 输出， I = 输入， DIG = 数字输出， ST = 施密特缓冲器输入， TTL = TTL 缓冲器输入。
x = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 ECCPMX 配置位置 1 时， P1B/P1C 和 P3B/P3C 的默认分配 （仅用于 80 引脚器件）。
2： 外部存储器接口 I/O 优先于所有其他数字和 PMP I/O。
3： 仅在 80 引脚器件上可用。
4： 当 ECCP2MX 配置位被清零时，为 ECCP2/P2A 的备用分配 （单片机模式中的所有器件适用）。
5： PMP 的默认配置 （PMPMX 配置位 = 1）。

表 10-12： PORTE 功能 （续）

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RE5/AD13/ PMA11/P1C	RE5	0	O	DIG	LATE<5> 数据输出。
		1	I	ST	PORTE<5> 数据输入。
	AD13(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 13 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 13 输入 (2)。
	PMA11	x	O	DIG	并行主端口地址。
	P1C(1)	0	O	DIG	ECCP1 增强型 PWM 输出，通道 C ；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
RE6/AD14/ PMA10/P1B	RE6	0	O	DIG	LATE<6> 数据输出。
		1	I	ST	PORTE<6> 数据输入。
	AD14(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 14 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 14 输入 (2)。
	PMA10	x	O	DIG	并行主端口地址。
	P1B(1)	0	O	DIG	ECCP1 增强型 PWM 输出，通道 B ；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
RE7/AD15/ PMA9/ECCP2/ P2A	RE7	0	O	DIG	LATE<7> 数据输出。
		1	I	ST	PORTE<7> 数据输入。
	AD15(3)	x	O	DIG	外部存储器接口，地址 / 数据 bit 15 输出 (2)。
		x	I	TTL	外部存储器接口，数据 bit 15 输入 (2)。
	PMA9	x	O	DIG	并行主端口地址。
	ECCP2(4)	0	O	DIG	ECCP2 比较输出和 ECCP2 PWM 输出；优先于端口数据。
		1	I	ST	ECCP2 捕捉输入。
	P2A(4)	0	O	DIG	ECCP2 增强型 PWM 输出，通道 A ；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。

图注： O = 输出， I = 输入， DIG = 数字输出， ST = 施密特缓冲器输入， TTL = TTL 缓冲器输入。
x = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 ECCPMX 配置位置 1 时， P1B/P1C 和 P3B/P3C 的默认分配 （仅用于 80 引脚器件）。
2： 外部存储器接口 I/O 优先于所有其他数字和 PMP I/O。
3： 仅在 80 引脚器件上可用。
4： 当 ECCP2MX 配置位被清零时，为 ECCP2/P2A 的备用分配 （单片机模式中的所有器件适用）。
5： PMP 的默认配置 （PMPMX 配置位 = 1）。

表 10-13： 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 所在的页
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	60
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	59
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	59
PORTG	RDPU	REPU	RJPU(1)	RG4	RG3	RG2	RG1	RG0	60

图注： PORTE 模块未使用阴影单元。

注 1： 在 64 引脚器件上未使用；读为 0。

© 2007 Microchip Technology Inc.

10.7 PORTF、 LATF 和 TRISF 寄存器

PORTF 是 7 位宽的双向端口。只有 PORTF 的引脚 7 没有模拟输入；它是惟一可以接受高达 5.5V 的电压的引脚。

PORTF 上的所有引脚都是使用施密特触发器输入缓冲器实现的。每个引脚都被单独地设置为输入或输出。

PORTF 与模拟外设功能复用。RF1 至 RF6 还可用作 A/D 转换器的模拟输入通道。可通过将 CMCON 寄存器中的相应位置1，以将所有引脚用作比较器输入或输出。要将 RF<6:3> 用作数字输入，还必须关闭比较器。

注 1： 当器件复位时，引脚 RF6:RF1 被配置为模拟输入并读为 0。

2： 要将 PORTF 配置为数字 I/O，需要将 ANCON0 和 ANCON1 中的相应位置 1。

当配置位 PMPMX = 0 时，PORTF 与并行主端口数据引脚复用。此复用仅对 80 引脚器件有效。

例 10-6： 初始化 PORTF

```
CLRF   PORTF        ; Initialize PORTF by
                    ; clearing output
                    ; data latches
CLRF   LATF         ; Alternate method to
                    ; clear output latches
BSF    WDTCON,ADSHR ; Enable write/read to
                    ; the shared SFR
MOVLW  C0h          ; make RF1:RF2 digital
MOVWF  ANCON0       ;
MOVLW  0Fh          ; make RF<6:3> digital
MOVWF  ANCON1       ;
BCF    WDTCON,ADSHR ; Disable write/read to
                    ; the shared SFR
MOVLW  CEh          ;
MOVWF  TRISF        ; Set RF5:RF4 as outputs,
                    ; RF<7:6>,<3:1> as inputs
```

表 10-14： PORTF 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RF1/AN6/ C2OUT	RF1	0	O	DIG	LATF<1> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<1> 数据输入；使能模拟输入时被禁止。
	AN6	1	I	ANA	A/D 输入通道 6。发生 POR 时为默认配置。
	C2OUT	x	O	DIG	比较器 2 输出。
RF2/PMA5/ AN7//C1OUT	RF2	0	O	DIG	LATF<2> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<2> 数据输入；使能模拟输入时被禁止。
	PMA5	x	O	DIG	并行主端口地址。
	AN7	1	I	ANA	A/D 输入通道 7。发生 POR 时为默认配置。
	C1OUT	x	O	DIG	比较器 1 输出。
RF3/AN8/ C2INB	RF3	0	O	DIG	LATF<3> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<3> 数据输入；使能模拟输入时被禁止。
	AN8	1	I	ANA	A/D 输入通道 8。发生 POR 时为默认配置。
	C2INB	x	I	ANA	比较器 2 输入 B。
RF4/AN9/ C2INA	RF4	0	O	DIG	LATF<4> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<4> 数据输入；使能模拟输入时被禁止。
	AN9	1	I	ANA	A/D 输入通道 9。发生 POR 时为默认配置。
	C2INA	x	I	ANA	比较器 2 输入 A。
RF5/PMD2/ AN10/C1INB/ CVREF	RF5	0	O	DIG	LATF<5> 数据输出；不受模拟输入影响。当使能 CVREF 输出时禁止。
		1	I	ST	PORTF<5> 数据输入；使能模拟输入时被禁止。当使能 CVREF 输出时禁止。
	PMD2(1)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
	AN10	1	I	ANA	A/D 输入通道 10 和比较器 C1+ 输入。 POR 时默认为输入配置。
	C1INB	x	I	ANA	比较器 1 输入 B。
	CVREF	x	O	ANA	比较器参考电压输出。使能此功能将禁止数字 I/O。
RF6/PMD1/ AN11/C1INA	RF6	0	O	DIG	LATF<6> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<6> 数据输入；使能模拟输入时被禁止。
	PMD1(1)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
	AN11	1	I	ANA	A/D 输入通道 11 和比较器 C1- 输入。POR 时默认为输入配置；不影响数字输出。
	C1INA	x	I	ANA	比较器 1 输入 A。
RF7/PMD0/ SS1	RF7	0	O	DIG	LATF<7> 数据输出。
		1	I	ST	PORTF<7> 数据输入。
	PMD0(1)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
	SS1	1	I	TTL	MSSP1 模块的从动选择输入。

图注： O = 输出， I = 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 = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 PMPMX 配置位为 0 时为 PMP 的备用分配，仅在 80 引脚器件上可用。

© 2007 Microchip Technology Inc.

表 10-15：　　与 PORTF 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	复位值所在的页：
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	60
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	59
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	59
ANCON0(1)	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	58
ANCON1(1)	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	58

图注：　— = 未实现位，读为 0。PORTF 未使用阴影单元。

注　1： 配置 SFR 的地址与默认 SFR 重叠；仅在 WDTCON<4> = 1 时才可访问配置 SFR。

10.8 PORTG、 TRISG 和 LATG 寄存器

PORTG 是 5 位宽的双向端口。 PORTG 上的所有引脚都仅是数字引脚，并且可以接受高达 5.5V 的电压。

PORTG 与 EUSART2 功能复用（表 10-16）。PORTG 引脚都有施密特触发输入缓冲器。PORTG 还与并行主端口的地址和控制功能复用。

当外设功能使能时，应考虑到每个 PORTG 引脚的 TRIS 位方向设置。有些外设使能时，会改写相应引脚的 TRIS 位方向设置而将引脚定义为输出，而另一些外设使能时，也会改写相应引脚的 TRIS 位方向设置，但却将引脚定义为输入。用户应该参考相应的外设章节来正确设置 TRIS 位。引脚改写值不载入 TRIS 寄存器。这将允许 TRIS 寄存器不会造成改写外设的读－修改－写操作。

虽然此端口只有 5 位宽，但是还是会实现 PORTG<7:5> 位。它们将用于控制与外部存储器总线（PORTD、PORTE 和 PORTJ）相关的 I/O 端口上的弱上拉电路。将这些位置 1 将使能弱上拉电路。由于这些位不是与 I/O 端口相关联的控制位，所以不会执行相应的 TRISG 和 LATG。

例 10-7： 初始化 PORTG

```
CLRF    PORTG   ; Initialize PORTG by
                ; clearing output
                ; data latches
CLRF    LATG    ; Alternate method to clear
                ; output data latches
MOVLW   04h     ; Value used to initialize
                ; data direction
MOVWF   TRISG   ; Set RG1:RG0 as outputs
                ; RG2 as input
                ; RG4:RG3 as outputs
```


© 2007 Microchip Technology Inc.

表 10-16： PORTG 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RG0/PMA8/ ECCP3/P3A	RG0	0	O	DIG	LATG<0> 数据输出。
		1	I	ST	PORTG<0> 数据输入。
	PMA8	x	O	DIG	并行主端口地址。
	ECCP3		O	DIG	ECCP3 比较和 PWM 输出，优先于端口数据。
			I	ST	ECCP3 捕捉输入。
	P3A	0	O	DIG	ECCP3 增强型 PWM 输出，通道 A；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
RG1/PMA7/ TX2/CK2	RG1	0	O	DIG	LATG<1> 数据输出。
		1	I	ST	PORTG<1> 数据输入。
	PMA7	x	O	DIG	并行主端口地址。
	TX2	1	O	DIG	同步串行数据输出 （EUSART2 模块）；优先于端口数据。
	CK2	1	O	DIG	同步串行数据输入 （EUSART2 模块）。用户必须配置为输入。
		1	I	ST	同步串行时钟输入 （EUSART2 模块）。
RG2/PMA6/ RX2/DT2	RG2	0	O	DIG	LATG<2> 数据输出。
		1	I	ST	PORTG<2> 数据输入。
	PMA6	x	O	DIG	并行主端口地址。
	RX2	1	I	ST	异步串行接收数据输入 （EUSART2 模块）。
	DT2	1	O	DIG	同步串行数据输出 （EUSART2 模块）；优先于端口数据。
		1	I	ST	同步串行数据输入 （EUSART2 模块）。用户必须配置为输入。
RG3/PMCS1/ CCP4/P3D	RG3	0	O	DIG	LATG<3> 数据输出。
		1	I	ST	PORTG<3> 数据输入。
	PMCS1	x	O	DIG	并行主端口地址片选 1。
		x	I	TTL	并行主端口地址片选 1 输入。
	CCP4	0	O	DIG	CCP4 比较输出和 CCP4 PWM 输出；优先于端口数据。
		1	I	ST	CCP4 捕捉输入。
	P3D	0	O	DIG	ECCP3 增强型 PWM 输出，通道 D；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
RG4/PMCS2/ CCP5/P1D	RG4	0	O	DIG	LATG<4> 数据输出。
		1	I	ST	PORTG<4> 数据输入。
	PMCS2	x	O	DIG	并行主端口地址片选 2。
	CCP5	0	O	DIG	CCP5 比较输出和 CCP5 PWM 输出；优先于端口数据。
		1	I	ST	CCP5 捕捉输入。
	P1D	0	O	DIG	ECCP1 增强型 PWM 输出，通道 D；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。

图注： O = 输出， I = 输入， DIG = 数字输出， ST = 施密特缓冲器输入， TTL = TTL 缓冲器输入。
x = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

表 10-17：　　与 PORTG 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 所在的页
PORTG	RDPU	REPU	RJPU(1)	RG4	RG3	RG2	RG1	RG0	60
LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	59
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	59

图注：　　— = 未实现位，读为 0。 PORTG 未使用阴影单元。

注　1： 在 64 引脚器件上未使用，读为 0。

© 2007 Microchip Technology Inc.

10.9 PORTH、 LATH 和 TRISH 寄存器

注：	PORTH 仅在 80 引脚器件上可用。

PORTH 是 8 位宽的双向 I/O 端口。PORTH 引脚 <3:0> 仅是数字引脚，并且可以接受高达 5.5V 的电压。

PORTH 上的所有引脚都是使用施密特触发器输入缓冲器实现的。每个引脚都被单独地设置为输入或输出。

当使能外部存储器接口时， 4 个 PORTH 引脚被用作该接口的高位地址线。来自接口的地址输出优先于其他数字 I/O。还将改写相应的 TRISH 位。PORTH 引脚（RH4 到 RH7）与模拟转换器输入复用。可通过在 ADCON1 寄存器中将相应位置 1 或清零来选择让这些引脚作为模拟输入来工作。RH2 至 RH6 与并行主端口复用，并且 RH4 至 RH6 被复用为比较器输入。

也可以将 ECCP1 和 ECCP3 模块的 PORTH 配置为备用的增强型 PWM 输出通道 B 和 C。可以通过清零 ECCPMX 配置位来完成上述配置。

例 10-8： 初始化 PORTH

```
CLRF   PORTH        ; Initialize PORTH by
                    ; clearing output
                    ; data latches
CLRF   LATH         ; Alternate method to
                    ; clear output latches
BSF    WDTCON,ADSHR ; Enable write/read to
                    ; the shared SFR
MOVLW  F0h          ; Configure PORTH as
MOVWF  ANCON1       ; digital I/O
BCF    WDTCON,ADSHR ; Disable write/read to
                    ; the shared SFR
MOVLW  0CFh         ; Value used to initialize
                    ; data direction
MOVWF  TRISH        ; Set RH<3:0> as inputs
                    ; RH<5:4> as outputs
                    ; RH<7:6> as inputs
```

表 10-18： PORTH 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RH0/A16	RH0	0	O	DIG	LATH<0> 数据输出。
		1	I	ST	PORTH<0> 数据输入。
	A16	x	O	DIG	外部存储器接口，地址线 16。优先于端口数据。
RH1/A17	RH1	0	O	DIG	LATH<1> 数据输出。
		1	I	ST	PORTH<1> 数据输入。
	A17	x	O	DIG	外部存储器接口，地址线 17。优先于端口数据。
RH2/A18/ PMD7	RH2	0	O	DIG	LATH<2> 数据输出。
		1	I	ST	PORTH<2> 数据输入。
	A18	x	O	DIG	外部存储器接口，地址线 18。优先于端口数据。
	PMD7(2)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
RH3/A19/ PMD6	RH3	0	O	DIG	LATH<3> 数据输出。
		1	I	ST	PORTH<3> 数据输入。
	A19	x	O	DIG	外部存储器接口，地址线 19。优先于端口数据。
	PMD6(2)	x	O	DIG	并行主端口数据输出。
		x	I	TTL	并行主端口数据输入。
RH4/PMD3/ AN12/P3C/ C2INC	RH4	0	O	DIG	LATH<4> 数据输出。
		1	I	ST	PORTH<4> 数据输入。
	PMD3(2)	x	I	TTL	并行主端口数据输出。
		x	O	DIG	并行主端口数据输入。
	AN12		I	ANA	A/D 输入通道 12。 POR 时默认为输入配置；不受数字输出的影响。
	P3C(1)	0	O	DIG	ECCP3 增强型 PWM 输出，通道 C ；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
	C2INC	x	I	ANA	比较器 2 输入 C。
RH5/PMBE/ AN13/P3B/ C2IND	RH5	0	O	DIG	LATH<5> 数据输出。
		1	I	ST	PORTH<5> 数据输入。
	PMBE(2)	x	O	DIG	并行主端口数据字节使能。
	AN13		I	ANA	A/D 输入通道 13。 POR 时默认为输入配置；不受数字输出的影响。
	P3B(1)	0	O	DIG	ECCP3 增强型 PWM 输出，通道 B ；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
	C2IND	x	I	ANA	比较器 2 输入 D。
RH6/PMRD/ AN14/P1C/ C1INC	RH6	0	O	DIG	LATH<6> 数据输出。
		1	I	ST	PORTH<6> 数据输入。
	PMRD(2)	x	O	DIG	并行主端口读选通。
		x	I	TTL	并行主端口读入。
	AN14		I	ANA	A/D 输入通道 14。 POR 时默认为输入配置；不受数字输出的影响。
	P1C(1)	0	O	DIG	ECCP1 增强型 PWM 输出，通道 C ；优先于端口和 PMP 数据。可能在增强型 PWM 关闭事件期间被配置为三态。
	C1INC	0	O	ANA	比较器 1 输入 C。

图注： O = 输出， I = 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 = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 ECCPMX 配置位被清零时，为 P1B/P1C 和 P3B/P3C 的备用分配。默认分配是 PORTE<6:3>。

2： 当 PMPMX 配置位为 0 时， PMP 的备用分配，仅在 80 引脚器件上可用。

© 2007 Microchip Technology Inc.

表 10-18： PORTH 功能 （续）

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RH7/PMWR/ AN15/P1B	RH7	0	O	DIG	LATH<7> 数据输出。
		1	I	ST	PORTH<7> 数据输入。
	PMWR(2)	x	O	DIG	并行主端口写选通。
		x	I	TTL	并行主端口写入。
	AN15		I	ANA	A/D 输入通道 15。 POR 时默认为输入配置；不受数字输出的影响。
	P1B(1)	0	O	DIG	ECCP1 增强型 PWM 输出，通道 B ；优先于端口和 PMP 数据。可能在增强型 PWM关闭事件期间被配置为三态。

图注： O = 输出， I = 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 = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

注 1： 当 ECCPMX 配置位被清零时，为 P1B/P1C 和 P3B/P3C 的备用分配。默认分配是 PORTE<6:3>。

2： 当 PMPMX 配置位为 0 时， PMP 的备用分配，仅在 80 引脚器件上可用。

表 10-19： 与 PORTH 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 所在的页
PORTH(1)	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	59
LATH(1)	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	60
TRISH(1)	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	59
ANCON1(2)	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	58

图注： PORTH 模块未使用阴影单元。

注 1： 在 64 引脚器件上未使用；读为 0。

2： 配置 SFR 的地址与默认 SFR 重叠；仅在 WDTCON<4> = 1 时才可访问配置 SFR。

10.10 PORTJ、 TRISJ 和 LATJ 寄存器

注：	PORTJ 仅在 80 引脚器件上可用。

PORTJ 是 8 位宽的双向端口。PORTJ 上的所有引脚都仅仅是数字的，并且可以接受高达 5.5V 的电压。

PORTJ 上的所有引脚都是使用施密特触发器输入缓冲器实现的。每个引脚都被单独地设置为输入或输出。

注：	这些引脚在任何器件复位时都被配置为数字输入引脚。

当启用外部存储器接口时，所有的 PORTJ 引脚都用于控制接口的输出。当通过清零 EBDIS 控制位（MEMCON<7>）而使能此接口时，这将自动发生。还将改写 TRISJ 位。

PORTJ 的每个引脚都有内部弱上拉电路。单个控制位可以关闭所有上拉电路。可以通过清零 RJPU 位（PORTG<5>）来关闭上拉电路。当 PORTB 端口的引脚配置为输出时，其弱上拉电路会自动关闭。发生任何器件复位时，弱上拉电路会被禁止。

例 10-9： 初始化 PORTJ

```
CLRF    PORTJ   ; Initialize PORTG by
                ; clearing output
                ; data latches
CLRF    LATJ    ; Alternate method to clear
                ; output data latches
MOVLW   0CFh    ; Value used to initialize
                ; data direction
MOVWF   TRISJ   ; Set RJ3:RJ0 as inputs
                ; RJ5:RJ4 as output
                ; RJ7:RJ6 as inputs
```


© 2007 Microchip Technology Inc.

表 10-20： PORTJ 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RJ0/ALE	RJ0	0	O	DIG	LATJ<0> 数据输出。
		1	I	ST	PORTJ<0> 数据输入。
	ALE	x	O	DIG	外部存储器接口地址锁存器使能控制输出；优先于数字 I/O。
RJ1/$\overline{OE}$	RJ1	0	O	DIG	LATJ<1> 数据输出。
		1	I	ST	PORTJ<1> 数据输入。
	$\overline{OE}$	x	O	DIG	外部存储器接口输出使能控制输出；优先于数字 I/O。
RJ2/$\overline{WRL}$	RJ2	0	O	DIG	LATJ<2> 数据输出。
		1	I	ST	PORTJ<2> 数据输入。
	$\overline{WRL}$	x	O	DIG	外部存储器总线写低字节控制；优先于数字 I/O。
RJ3/$\overline{WRH}$	RJ3	0	O	DIG	LATJ<3> 数据输出。
		1	I	ST	PORTJ<3> 数据输入。
	$\overline{WRH}$	x	O	DIG	外部存储器接口写高字节控制输出；优先于数字 I/O。
RJ4/BA0	RJ4	0	O	DIG	LATJ<4> 数据输出。
		1	I	ST	PORTJ<4> 数据输入。
	BA0	x	O	DIG	外部存储器接口字节地址 0 控制输出；优先于数字 I/O。
RJ5/$\overline{CE}$	RJ5	0	O	DIG	LATJ<5> 数据输出。
		1	I	ST	PORTJ<5> 数据输入。
	$\overline{CE}$	x	O	DIG	外部存储器接口芯片使能控制输出；优先于数字 I/O。
RJ6/$\overline{LB}$	RJ6	0	O	DIG	LATJ<6> 数据输出。
		1	I	ST	PORTJ<6> 数据输入。
	$\overline{LB}$	x	O	DIG	外部存储器接口低字节使能控制输出；优先于数字 I/O。
RJ7/$\overline{UB}$	RJ7	0	O	DIG	LATJ<7> 数据输出。
		1	I	ST	PORTJ<7> 数据输入。
	$\overline{UB}$	x	O	DIG	外部存储器接口高字节使能控制输出；优先于数字 I/O。

图注： O = 输出， I= 输入， DIG = 数字输出， ST = 施密特缓冲器输入，
x = 任意值 （TRIS 位不影响端口方向或在此选项中被忽略）。

表 10-21： 与 PORTJ 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	复位值所在的页
PORTJ(1)	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	60
LATJ(1)	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	59
TRISJ(1)	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	59
PORTG	RDPU	REPU	RJPU(1)	RG4	RG3	RG2	RG1	RG0	60

图注： PORTJ 模块未使用阴影单元。

注 1： 在 64 引脚器件上未使用；读为 0。

注：

 © 2007 Microchip Technology Inc.

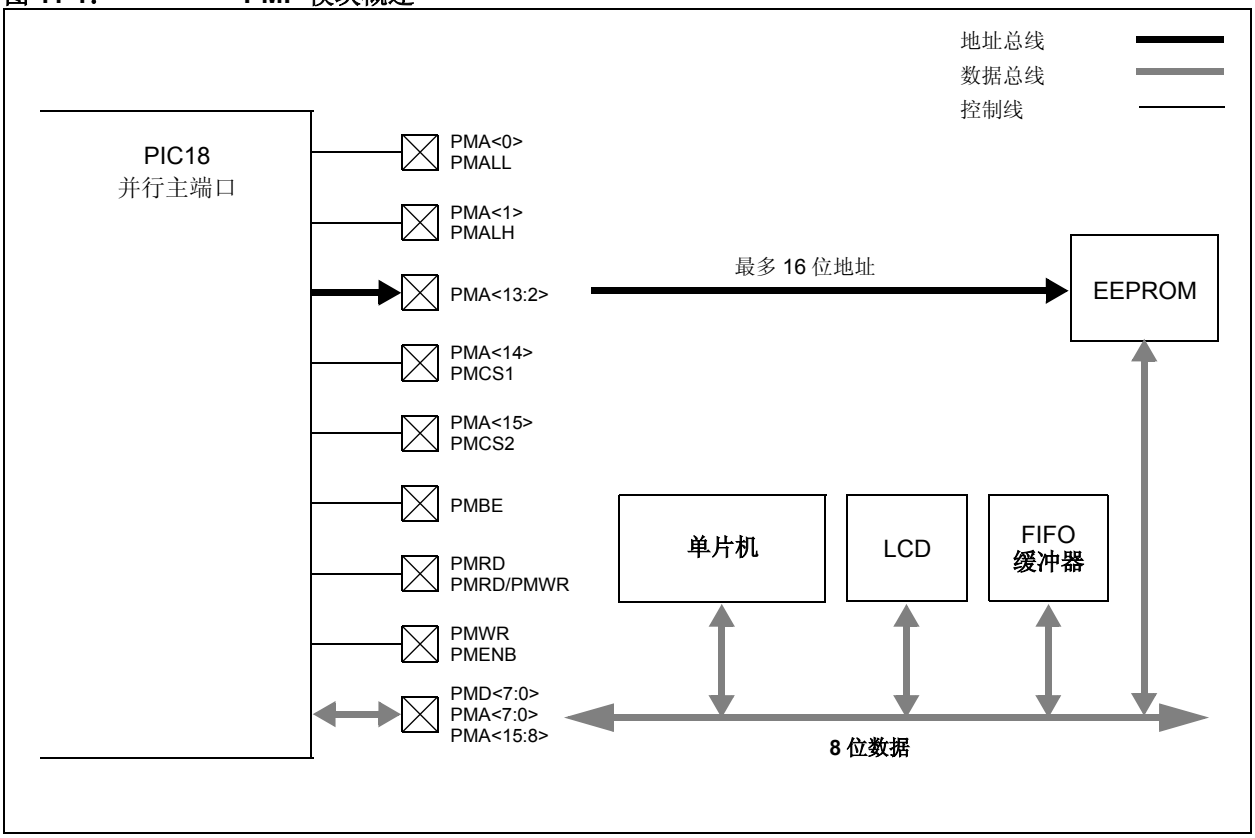
11.0 并行主端口

并行主端口模块（Parallel Master Port，PMP）是一个并行 8 位 I/O 模块，专用于与各种并行器件，如通信外设、LCD、外部存储器件及单片机进行通信。由于并行外设的接口变化很大，所以 PMP 具有高度的可配置性。PMP 模块可以被配置为并行主端口或并行从端口。

PMP 模块的主要特性包括：

- 最多 16 根可编程地址线
- 最多 2 根片选线信号线
- 可编程选通选项
  - 单独的读写选通，或；
  - 带有使能选通的读 / $\overline{\text{写}}$选通
- 地址自动递增 / 自动递减
- 可编程地址 / 数据复用
- 控制信号的可编程极性
- 支持传统的并行从端口
- 增强的并行从动支持
  - 地址支持
  - 4 KB 深的自动递增缓冲器
- 可编程等待状态
- 可选择的输入电平

图 11-1： PMP 模块概述



11.1 模块寄存器

PMP 模块共有 14 个特殊功能寄存器，及一个用于设置配置选项的额外寄存器。其中 8 个寄存器用于控制，而其余 6 个寄存器用于 PMP 数据传输。

11.1.1 控制寄存器

这 8 个 PMP 控制寄存器是：

- PMCONH 和 PMCONL
- PMMODEH 和 PMMODEL
- PMSTATL 和 PMSTATH
- PMEH 和 PMEL

PMCON 寄存器（寄存器 11-1 和寄存器 11-2）控制模块的基本工作，包括启动或停止模块。还配置地址复用及控制选通配置。

PMMODE 寄存器（寄存器 11-3 和寄存器 11-4）配置各种主从工作模式、数据宽度和中断的产生。

PMEH 和 PMEL 寄存器（寄存器 11-5 和寄存器 11-6）在硬件（I/O 引脚）级配置模块工作。

PMSTAT 寄存器（寄存器 11-7 和寄存器 11-8）提供模块输入和输出缓冲器的状态标志，具体状态取决于所在的工作模式。

寄存器 11-1： PMCONH：并行端口控制高字节寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **PMPEN**：并行主端口使能位
1 = 使能 PMP
0 = 禁止 PMP，不执行片外访问

bit 6 **未实现**：读为 0

bit 5 **PSIDL**：空闲模式停止位
1 = 器件进入空闲模式后，模块停止工作
0 = 在空闲模式下模块继续工作

bit 4-3 **ADRMUX1:ADRMUX0**：地址 / 数据复用选择位
11 = 保留
10 = 所有 16 位地址与 PMD<7:0> 引脚复用
01 = 地址的低 8 位与 PMD<7:0> 引脚复用，高 8 位与 PMA<15:8> 引脚复用
00 = 地址和数据使用独立的引脚

bit 2 **PTBEEN**：字节使能端口使能位（16 位主控模式）
1 = 使能 PMBE 端口
0 = 禁止 PMBE 端口

bit 1 **PTWREN**：写使能选通端口使能位
1 = 使能 PMWR/PMENB 端口
0 = 禁止 PMWR/PMENB 端口

bit 0 **PTRDEN**：读 / 写选通端口使能位
1 = 使能PMRD/PMWR 端口
0 = 禁止PMRD/PMWR 端口

© 2007 Microchip Technology Inc.

寄存器 11-2：　　PMCONL：并行端口控制低字节寄存器

R/W-0	R/W-0	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6　**CSF1:CSF0**：片选功能位
11 = 保留
10 = PMCS1 和 PMCS2 用作片选功能
01 = PMCS2 用作片选功能， PMCS1 用作地址的 bit 14 （PMADDRH 地址的 bit 6）
00 = PMCS2 和 PMCS1 分别用作地址的 bit 15 和 bit 14 （PMADDRH 地址的 bit 7 和 bit 6）

bit 5　**ALP**：地址锁存极性位 (1)
1 = 高电平有效 （PMALL 和 PMALH）
0 = 低电平有效 （PMALL 和 PMALH）

bit 4　**CS2P**：片选 2 极性位 (1)
1 = 高电平有效 （PMCS2）
0 = 低电平有效 （PMCS2）

bit 3　**CS1P**：片选 1 极性位 (1)
1 = 高电平有效 （PMCS1/PMCS）
0 = 低电平有效 （PMCS1/PMCS）

bit 2　**BEP**：字节使能极性位
1 = 字节使能高电平有效 （PMBE）
0 = 字节使能低电平有效 （PMBE）

bit 1　**WRSP**：写选通极性位
对于从动模式和主控模式 2 （PMMODEH<1:0> = 00,01,10）：
1 = 写选通高电平有效 （PMWR）
0 = 写选通低电平有效 （PMWR）
对于主控模式 1 （PMMODEH<1:0> = 11）：
1 = 使能选通高电平有效 （PMENB）
0 = 使能选通低电平有效 （PMENB）

bit 0　**RDSP**：读选通极性位
对于从动模式和主控模式 2 （PMMODEH<1:0> = 00,01,10）：
1 = 读选通高电平有效 （PMRD）
0 = 读选通低电平有效 （PMRD）
对于主控模式 1 （PMMODEH<1:0> = 11）：
1 = 读 / 写选通高电平有效 （PMRD/PMWR）
0 = 读 / 写选通低电平有效 （PMRD/PMWR）

注　1：当对应的引脚用作地址线时，这些位无效。

寄存器 11-3： PMMODEH：并行端口模式高字节寄存器

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **BUSY**：忙位 （仅用于主控模式）
1 = 端口忙
0 = 端口不忙

bit 6-5 **IRQM1:IRQM0**：中断请求模式位
11 = 当读取读缓冲器 3 或写入写缓冲器 3 时产生中断 （缓冲的 PSP 模式）
或当 PMA<1:0> = 11 时读或写操作时产生中断 （仅可寻址 PSP 模式）
10 = 不产生中断，处理器暂停
01 = 在读 / 写周期结束时产生中断
00 = 不产生中断

bit 4-3 **INCM1:INCM0**：递增模式位
11 = PSP 读和写缓冲器自动递增 （仅传统 PSP 模式）
10 = DDR<15,13:0> 在每个读写周期递减
01 = ADDR<15,13:0> 在每个读写周期递增
00 = 地址不会递增或递减

bit 2 **MODE16**：8/16 位模式位
1 = 16 位模式：数据寄存器为 16 位，读或写数据寄存器需要两次 8 位数据的传输
0 = 8 位模式：数据寄存器为 8 位，读或写数据寄存器要一次 8 位数据的传输

bit 1-0 **MODE1:MODE0**：并行端口模式选择位
11 = 主控模式 1 （PMCSx、 PMRD/PMWR、 PMENB、 PMBE、 PMA<x:0> 和 PMD<7:0>）
10 = 主控模式 2 （PMCSx、 PMRD、 PMWR、 PMBE、 PMA<x:0> 和 PMD<7:0>）
01 = 增强 PSP，控制信号 （PMRD、 PMWR、 PMCS、 PMD<7:0> 和 PMA<1:0>）
00 = 传统并行从端口，控制信号 （PMRD、 PMWR、 PMCS 和 PMD<7:0>）

© 2007 Microchip Technology Inc.

寄存器 11-4： PMMODEL：并行端口模式低字节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAITB1(1)	WAITB0(1)	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1(1)	WAITE0(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **WAITB1:WAITB0：**从数据建立到执行读 / 写的等待状态配置位 (1)
11 = 数据等待时间为 4 个 TCY；地址复用时间为 4 个 TCY
10 = 数据等待时间为 3 个 TCY；地址复用时间为 3 个 TCY
01 = 数据等待时间为 2 个 TCY；地址复用时间为 2 个 TCY
00 = 数据等待时间为 1 个 TCY；地址复用时间为 1 个 TCY

bit 5-2 **WAITM3:WAITM0：**从执行读操作到字节使能选通的等待状态配置位
1111 = 额外等待 15 个 TCY
...
0001 = 额外等待 1 个 TCY
0000 = 无额外等待周期（操作被强制在 1 个 TCY 内完成）

bit 1-0 **WAITE1:WAITE0：**选通后数据保持的等待状态配置位 (1)
11 = 等待 4 个 TCY
10 = 等待 3 个 TCY
01 = 等待 2 个 TCY
00 = 等待 1 个 TCY

注 1： 只要 WAITM3:WAITM0 = 0000，就忽略 WAITB 和 WAITE 位。

寄存器 11-5： PMEH：并行端口使能高字节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **PTEN15:PTEN14：**PMCSx 选通使能位
1 = PMA15 和 PMA14 用作 PMA<15:14> 或分别用作 PMCS2 和 PMCS1
0 = PMA15 和 PMA14 用作端口 I/O

bit 5-0 **PTEN13:PTEN8：**PMP 地址端口使能位
1 = PMA<13:8> 用作 PMP 地址线
0 = PMA<13:8> 用作端口 I/O

寄存器 11-6：　PMEL：并行端口使能低字节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2　**PTEN7:PTEN2：**PMP 地址端口使能位
1 = PMA<7:2> 用作 PMP 地址线
0 = PMA<7:2> 用作端口 I/O

bit 1-0　**PTEN1:PTEN0：**PMALH/PMALL 选通使能位
1 = PMA1 和 PMA0 用作 PMA<1:0> 或分别用作 PMALH 和 PMALL
0 = PMA1 和 PMA0 用作端口 I/O

寄存器 11-7：　PMSTATH：并行端口状态高字节寄存器

R-0	R/W-0	U-0	U-0	R-0	R-0	R-0	R-0
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　**IBF：**输入缓冲器满状态位
1 = 所有可写输入缓冲器寄存器为满
0 = 某些或全部可写输入缓冲器寄存器为空

bit 6　**IBOV：**输入缓冲器溢出状态位
1 = 企图写入一个已满的输入字节寄存器 （必须由软件清零）
0 = 没有发生溢出

bit 5-4　**未实现：**读为 0

bit 3-0　**IB3F:IB0F：**输入缓冲器状态满位
1 = 输入缓冲器中包含未读取的数据 （读缓冲器将清零该位）
0 = 输入缓冲器中没有任何未读取的数据

© 2007 Microchip Technology Inc.

寄存器 11-8：　　PMSTATL：并行端口状态低字节寄存器

R-1	R/W-0	U-0	U-0	R-1	R-1	R-1	R-1
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　**OBE：**输出缓冲器空状态位
1 = 所有可读的输出缓冲寄存器为空
0 = 某些或全部可读的输出缓冲寄存器为满

bit 6　**OBUF：**输出缓冲器下溢状态位
1 = 读取了一个为空的输出字节寄存器（必须由软件清零）
0 = 没有发生下溢

bit 5-4　**未实现：**读为 0

bit 3-0　**OBnE：**输出缓冲器 n 空状态位
1 = 输出缓冲器为空（向缓冲器写数据将清零该位）
0 = 输出缓冲器中包含未发送的数据

11.1.2 数据寄存器

PMP 模块使用 6 个寄存器从单片机传入或传出数据。它们组合成三对以允许 16 位数据操作：

- PMDIN1H 和 PMDIN1L
- PMDIN2H 和 PMDIN2L
- PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L
- PMDOUT2H 和 PMDOUT2L

PMDIN1 寄存器用于在从动模式下输入数据，并在主控模式下输入和输出数据。PMDIN2 寄存器用于在选择从动模式下缓冲输入的数据。

PMADDRx/PMDOUT1x 寄存器实际上是一个寄存器对，该寄存器对的名称和功能由模块的操作模式决定。在主控模式中，这两个寄存器用作 PMADDRH 和 PMADDRL 寄存器，且包含任何输入和输出数据的地址。在从动模式中，这两个寄存器用作 PMDOUT1H 和 PMDOUT1L，并用于输出数据。

PMADDRH 与 PMADDRL 不同，PMADDRH 还具有受限的 PMP 控制功能。 如果模块在选择主控模式配置下工作，则寄存器的高两位可以用于确定片选信号的操作。如果片选信号不可用，则 PMADDR 将仅用作保存地址的高 8 位。PMADDRH 中各位的功能如寄存器 11-9 中所示。

PMDOUT2H 和 PMDOUT2L 寄存器只用于缓冲的从动模式，并作为输出数据的缓冲器。

11.1.3 引脚配置控制寄存器

除了模块级别配置选项之外，PMP 模块还可以在 I/O 引脚上配置为与电相关的操作。该选项允许用户选择与 PMP 共用的数字 I/O 引脚上的普通施密特触发器输入缓冲器，或使用 TTL 级兼容缓冲器。缓冲器配置由 PADCFG1 寄存器中的 PMPTTL 位控制。

PADCFG1 寄存器是其中一个共享地址 SFR，其地址与 TMR2 寄存器相同。通过置 1 ADSHR（WDTCON<4>）可以访问 PADCFG1。更多信息，请参见**第 5.3.4.1 节“共享地址 SFR”**。

寄存器 11-9： PMADDRH：并行端口地址寄存器，高字节 （仅主控模式）(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CS2	CS1	ADDR<13:8>					
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置位	0 = 清零	x = 未知

bit 7 **CS2**：片选 2 位
如果 PMCON<7:6> = 10 或 01：
1 = 片选 2 有效
0 = 片选 2 无效
如果 PMCON<7:6> = 11 或 00：
位用作 ADDR<15>。

bit 6 **CS1**：片选 1 位
如果 PMCON<7:6> = 10：
1 = 片选 1 有效
0 = 片选 1 无效
如果 PMCON<7:6> = 11 或 0x：
位用作 ADDR<14>。

bit 5-0 **ADDR13:ADDR0**：目标地址位

注 1： 在增强型从动模式中， PMADDRH 用作 PMDOUT1H （输出数据缓冲寄存器之一）。

© 2007 Microchip Technology Inc.

11.1.4 PMP 复用选项 （80 引脚器件）

在默认情况下，PMP 和外部存储器总线复用 PORTD 和 PORTE 上相同 I/O 引脚的某些信号。在某些应用中可能需要在别处也可以加载 PMP 信号。对于这些实例，可将 80 引脚器件配置为对不同的 I/O 端口复用 PMP。PMP 的配置由 PMPMX 配置位的设置决定；在默认情况下，PMP 和 EMB 模块共享 PORTD 和 PORTE。可选引脚配置如表 11-1 所示。

表 11-1： 80 引脚器件的 PMP 引脚复用

PMP 功能	引脚分配	
	PMPMX = 1	**PMPMX =** 0
PMD0	PORTD<0>	PORTF<7>
PMD1	PORTD<1>	PORTF<6>
PMD2	PORTD<2>	PORTF<5>
PMD3	PORTD<3>	PORTH<4>
PMD4	PORTD<4>	PORTA<5>
PMD5	PORTD<5>	PORTA<4>
PMD6	PORTD<6>	PORTH<3>
PMD7	PORTD<7>	PORTH<2>
PMBE	PORTE<2>	PORTH<5>
PMWR	PORTE<1>	PORTH<7>
PMRD	PORTE<0>	PORTH<6>

11.2 从端口模式

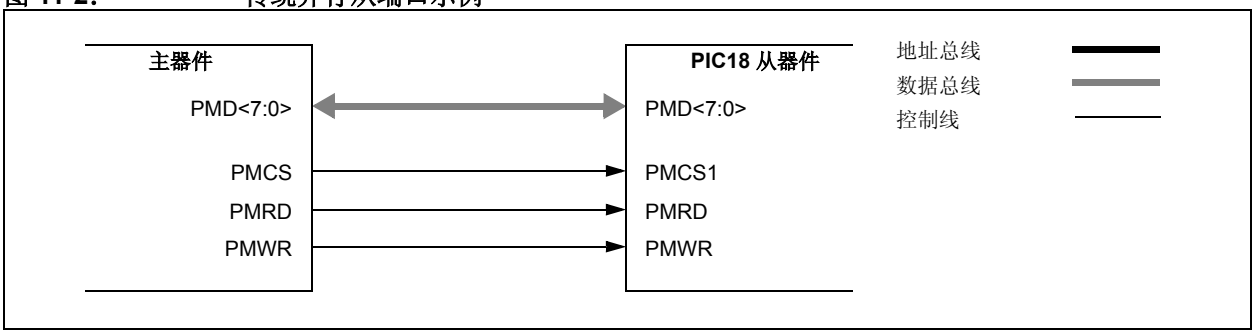
通过使用PMMODEH寄存器中的MODE1:MODE0位配置模块工作的主模式。此设置决定此模块是在从动模式还是在主控模式下工作，并决定控制引脚的使用。

11.2.1 传统模式 （PSP）

在传统模式 （PMMODEH<1:0> = 00 且 PMPEN = 1）下，该模块被配置为并行从端口，相应的使能模块引脚为模块所专用。在该模式下，外部器件（如其他单片机或微处理器）可通过 8 位数据总线 （PMD<7:0>）、读（PMRD）、写（PMWR)）和片选（PMCS）输入异步地读写数据。此器件用作总线上的从器件并响应读 / 写控制信号。

图11-2所示为并行从端口的连接。如果片选信号有效并产生写选通 （PMCS = 1 且 PMWR = 1），则来自 PMD<7:0> 的数据会被捕捉到 PMDIN1L 寄存器中。

图 11-2： 传统并行从端口示例



11.2.1.1 写从端口

如果片选信号有效并产生写选通（PMCS = 1 且 PMWR = 1），则会将来自 PMD<7:0> 的数据捕捉到 PMDIN1L 寄存器。写操作结束时 PMPIF 和 IBF 标志位会被置 1。写模式下的控制信号时序如图 11-3 所示。控制信号的极性是可配置的。

11.2.1.2 读从端口

如果片选信号有效且产生写选通（PMCS = 1 且 PMRD = 1），则 PMDOUTL1 寄存器（PMDOUTL1<7:0>）中的数据将被传送到 PMD<7:0>。读模式下的控制信号时序如图 11-4 中所示。

图 11-3： 并行从端口写操作波形图

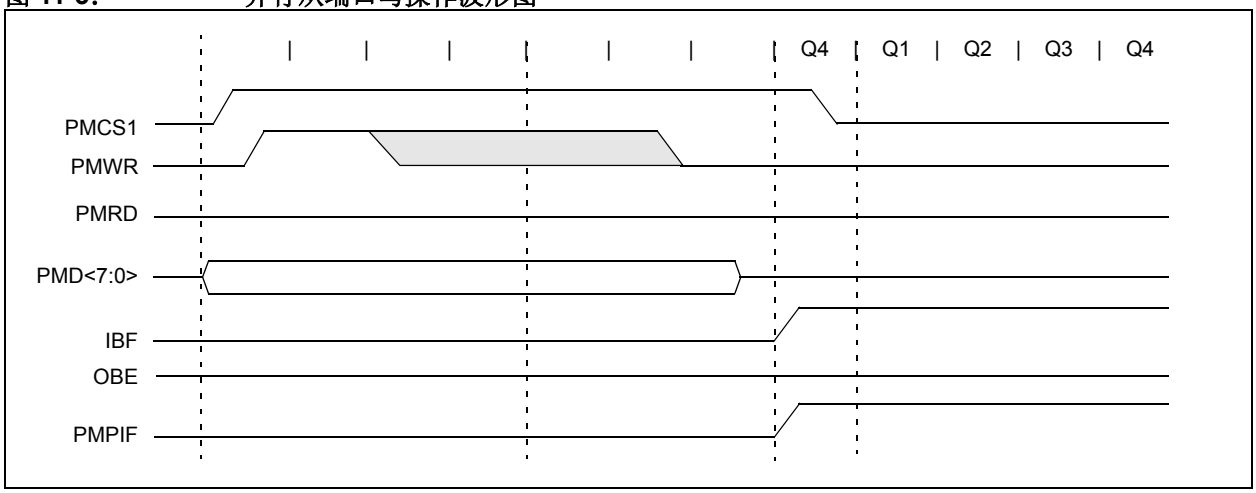
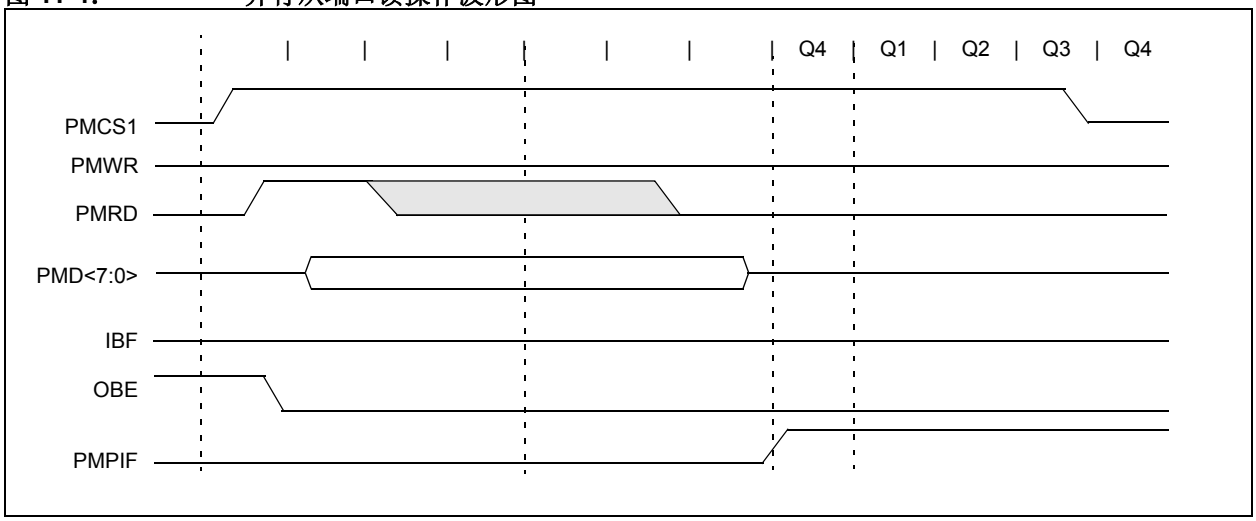


图 11-4： 并行从端口读操作波形图



© 2007 Microchip Technology Inc.

11.2.2 缓冲并行从端口模式

缓冲并行从端口模式在功能上与传统的并行从端口模式基本相同，只有一种例外：它实现了 4 级读写缓冲器操作。通过将 PMMODE 寄存器中的 INCM 位置 1 可使能缓冲 PSP 模式。如果将 INCM<1:0> 位设置为 11，则 PMP 模块将用作缓冲并行从端口。

激活缓冲模式后，PMDIN1L、PMDIN1H、PMDIN2L 和 PMDIN2H 寄存器将作为写缓冲器，而 PMDOUT1L、PMDOUT1H、PMDOUT2L 和 PMDOUT2H 寄存器将作为读缓冲器。缓冲器编号为 0 至 3，PMDIN1L 到 PMDIN2H 的低字节为读缓冲器，而 PMDOUT1L 到 PMDOUT2H 为写缓冲器。

11.2.2.1 读从端口

对于读操作，字节按顺序发送，从每个读选通的缓冲器 0（PMDOUT1L<7:0>）开始到缓冲器 3（PMDOUT2H<7:0>）结束。该模块通过内部指针跟踪要读取的缓冲器。每个缓冲器在 PMSTATL 寄存器中有一个相应的读状态位 OBxE。当缓冲器中包含尚未写入总线的数据时，该位被清零，当数据写入总线时该位被置 1。如果当前读取的缓冲器单元为空，则将产生缓冲器下溢，且缓冲器上溢标志位 OBUF 置 1。如果所有 4 个 OBxE 状态位都被置 1，则输出缓冲器空标志位（OBE）也将被置 1。

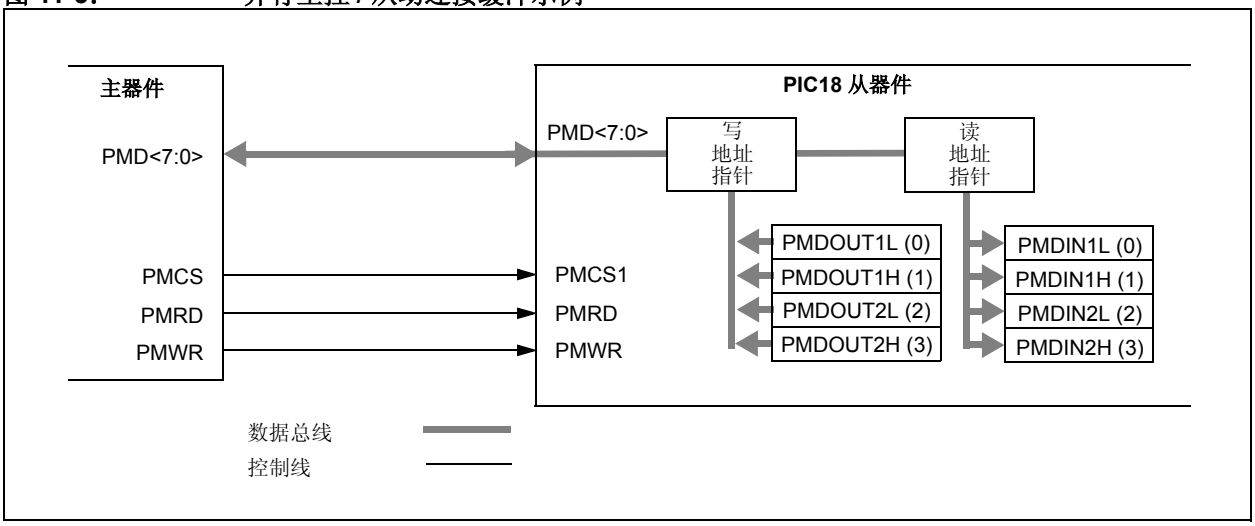
11.2.2.2 写从端口

对于写操作，数据按顺序存储，从缓冲器 0（PMDIN1L<7:0>）开始到缓冲器 3（PMDIN2H<7:0）结束。和读操作一样，模块内部指针保持指向下一个要写的缓冲器。

输入缓冲器在 PMSTATH 寄存器中具有自己的写状态位 IBxF。当缓冲器包含尚未读取的输入数据，则该位置 1，当数据已读取时，则该位清零。该标志位在写选通时被置 1。如果相关的 IBxF 位置 1 时对缓冲器执行写操作，则缓冲器上溢标志位 IBOV 被置 1，缓冲器中的任何输入地数据将会丢失。如果所有 4 个 IBxF 标志位被置 1，则输入缓冲器满标志位（IBF）被置 1。

在缓冲从动模式中，可以配置该模块以在每次读或写选通（IRQM1:IRQM0 = 01）时产生中断。可以配置该模块以在读取读缓冲器 3 或写入写缓冲器 3 时产生一个中断，即每四个读写选通（RQM1:IRQM0 = 11）信号产生一个中断。如果输入数据的每四个字节发生一个中断，则可以读取所有的输入缓冲器寄存器以清零 IBxF 标志位。如果这些标志位未被清零，则可能会产生溢出。

图 11-5：　　并行主控 / 从动连接缓冲示例



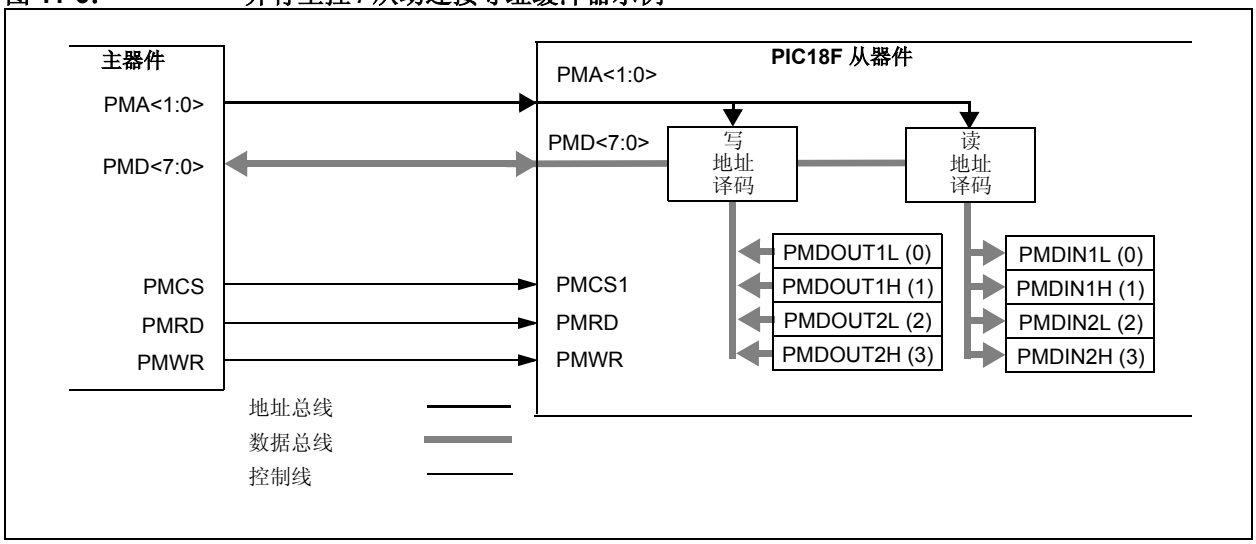
11.2.3 可寻址的并行从端口模式

在可寻址的并行从端口模式中（PMMODEH<1:0> = 01），模块被配置为具有两个额外的输入 PMA<1:0>（分别为地址线 1 和 0）。这使其可以像固定的读写寄存器一样直接寻址 4 字节缓冲器空间。和缓冲传统模式一样，数据从 PMDOUT1L、PMDOUT1H、PMDOUT2L 和 PMDOUT2H 中读出，且读入 PMDIN1L、PMDIN1H、PMDIN2L 和 PMDIN2H。表11-2给出了输入和输出寄存器的输入地址的缓冲器寻址。

表 11-2： 从动模式缓冲器寻址

PMADDR <1:0>	输出寄存器（缓冲器）	输入寄存器（缓冲器）
00	PMDOUT1L (0)	PMDIN1L (0)
01	PMDOUT1H (1)	PMDIN1H (1)
10	PMDOUT2L (2)	PMDIN2L (2)
11	PMDOUT2H (3)	PMDIN2H (3)

图 11-6： 并行主控 / 从动连接寻址缓冲器示例



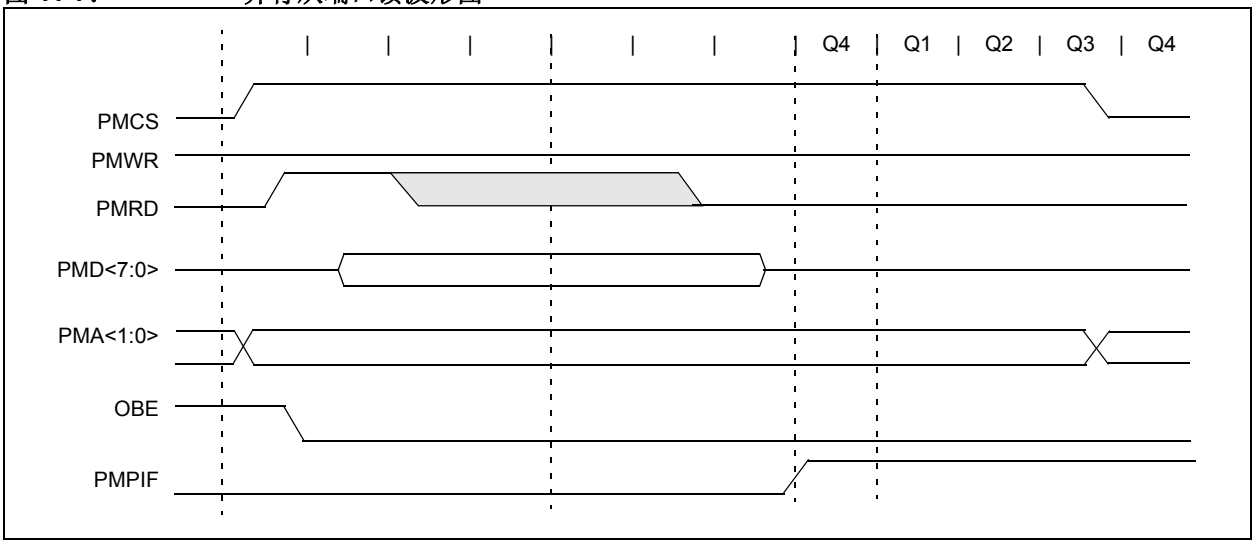
© 2007 Microchip Technology Inc.

11.2.3.1 读从端口

如果片选信号有效并产生读选通（PMCS = 1 且 PMRD = 1），四个输出字节数据之一被送到 PMD<7:0>。读哪个字节取决于 ADDR<1:0> 上的 2 位地址。表 11-2 给出了相应的输出寄存器及其对应的地址。

当读取输出缓冲器时，相应的 OBxE 位被置 1。OBE 标志位在所有缓冲器为空时置 1。如果任何缓冲器已为空（OBxE = 1），则对该寄存器的下一次读取操作将产生 OBUF 事件。

图 11-7： 并行从端口读波形图

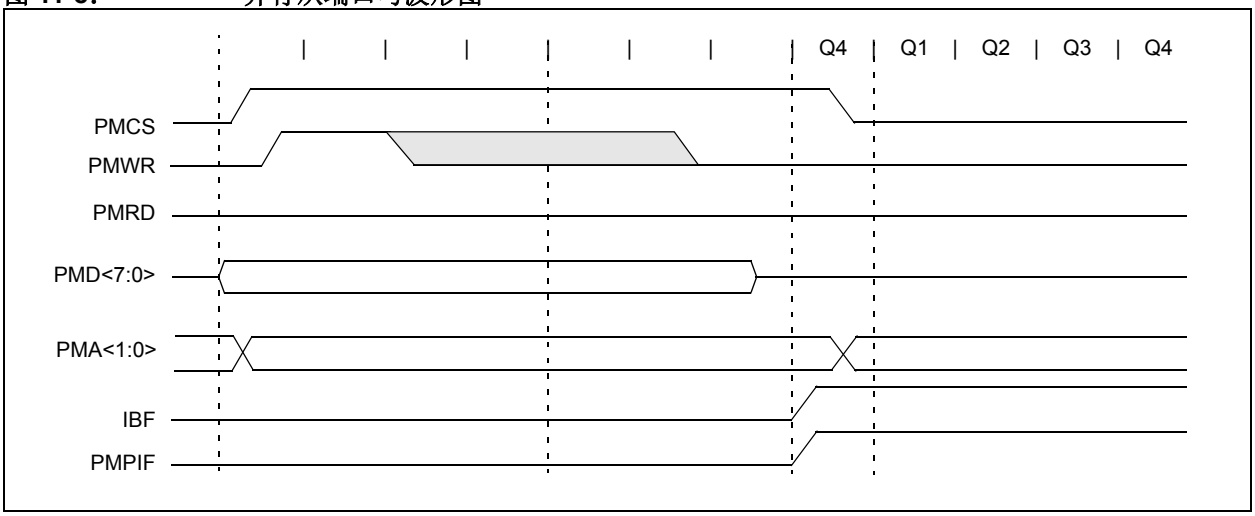


11.2.3.2 写从端口

如果片选信号有效并产生写选通（PMCS = 1 且 PMWR = 1），则 PMD<7:0> 的数据被捕捉到四个输入缓冲器字节之一。写哪个字节取决于ADDRL<1:0>上的 2 位地址。表 11-2 给出了相应的输入寄存器及其对应的地址。

当写输入缓冲器时，相应的 IBxF 位被置 1。写完所有缓冲器时， IBF 标志位被置 1。如果已对某个缓冲器执行了写操作（IBxF = 1），则对该缓冲器的下一次写选通将产生 OBUF 事件，且字节将被丢弃。

图 11-8： 并行从端口写波形图



11.3 主端口模式

在主控模式中， PMP 模块提供了 8 位数据总线、最多 16 位地址及操作各种外部并行器件（例如存储器件、外设和从动单片机）的所有必需的控制信号。要使 PMP 工作在主控模式下，必须使能模块（PMPEN = 1），且必须将模式设置为两种可用的主控模式之一（PMMODEH<1:0> = 10 或 11）。

由于许多并行器件具有多种控制方式，PMP模块设计得非常灵活，以适应多种配置要求。部分特性包括：

- 8 位数据总线上的 8 和 16 位数据模式
- 可配置的地址 / 数据复用
- 最多 2 根片选线
- 最多 16 根可选地址线
- 地址自动递增和自动递减
- 所有控制线都可以选择极性
- 在读 / 写周期的不同阶段可配置等待状态

11.3.1 PMP 和 I/O 引脚控制

有多个控制位用于配置模块中的控制和地址信号的出现或消失。它们是 PTBEEN、TWREN、PTRDEN 和 PTEN<15:0>。这些位使用户可保存其他功能的引脚，并可灵活地控制外部地址。当这些位中的任一位被置 1 时，相应的引脚将具备相关的功能；被清零时，相应的引脚将恢复到定义的 I/O 端口功能。

将 PTEN 位置 1 将使能相关的引脚作为地址引脚并驱动 PMADDR 寄存器中的相应数据。清零 PTENx 位将强制引脚恢复到原始 I/O 功能。

对于配置为片选 （PMCS1 或 PMCS2）的引脚，其相应的 PTENx 位置 1，当未执行读或写操作时，片选引脚将驱动未激活的数据（具有 CS1P 和 CS2P 位定义的极性 ）PTEN0 和 PTEN1 位还控制 PMALL 和 PMALH 信号。当使用复用功能时，将使能相应的地址锁存信号。

11.3.2 读 / 写控制

PMP 模块支持两种不同的读 / 写发送信号方式。在主控模式 1 下，读和写选通合并为一根信号控制线 PMRD/PMWR。第二个控制线PMENB决定何时执行读或写操作。在主控模式 2 中，在独立的引脚上提供了独立的读和写选通 （PMRD 和 PMWR）。

可以将所有控制信号（PMRD、PMWR、PMBE、PMENB、PMAL 和 PMCSx）单独配置为正极性或负极性。此配置由PMCONL 寄存器中单独的位控制。注意，共用同一输出引脚的控制信号的极性 （如， PMWR 和 PMENB）由同一个位控制；该配置取决于所使用的主端口模式。

11.3.3 数据宽度

PMP 支持 8 位和 16 位数据宽。数据宽度由 MODE16 位选择（PMMODEH<2>）。由于进出模块大的 数据路径仅为 8 位宽，16 位操作总是以复用形式处理，先传送数据的最低有效字节。要区分数据字节，通过字节使能（PMBE）控制选通在数据的最高有效字节传送到数据线时发出信号。

11.3.4 地址复用

在任一主控模式中 （PMMODEH<1:0> = 1x），用户都可以将地址总线配置为与数据总线复用。这可以通过 ADRMUX1:ADRMUX0 位（PMCONH<4:3>）实现。有三种地址复用模式可供使用；这些模式的典型引脚排列配置如图 11-9、图 11-10 和图 11-11 所示。

在多路分解模式中（PMCONH<4:3> = 00），数据和地址信息是完全分开的。数据位位于 PMD<7:0>，而地址位位于 PMADDRH<7:0> 和 PMADDRL<7:0> 上。

在部分复用模式中（PMCONH<4:3> = 01），地址的低 8 位与 PMD<7:0> 上的数据引脚复用。地址的高 8 位不受影响，位于 PMADDRH<7:0> 上。PMA0 引脚用作地址锁器，并接到地址锁存器低电平 （PMALL）使能选通。读写时序在一个完整的 CPU 周期内展开，其间地址位于 PMD<7:0> 引脚上。

在完全的复用模式中（PMCONH<4:3> = 10），地址的全部 16 位与 PMD<7:0> 上的数据引脚复用。PMA0 和 PMA1 引脚分别用于接地址锁存器低电平（PMALL）使能和地址锁存器高电平 （PMALH）使能选通。读写时序在两个完整的 CPU 周期内展开。在第一个周期内，地址的低 8 位位于 PMD<7:0> 引脚上， PMALL 选通有效。在第二个周期内，地址的高 8 位位于 PMD<7:0> 引脚上， PMALH 选通有效。如果将高地址位配置为片选引脚，则相应的地址位被自动强制为 0。

© 2007 Microchip Technology Inc.

图 11-9：　　多路分解寻址模式 （独立的读写选通，两个片选）

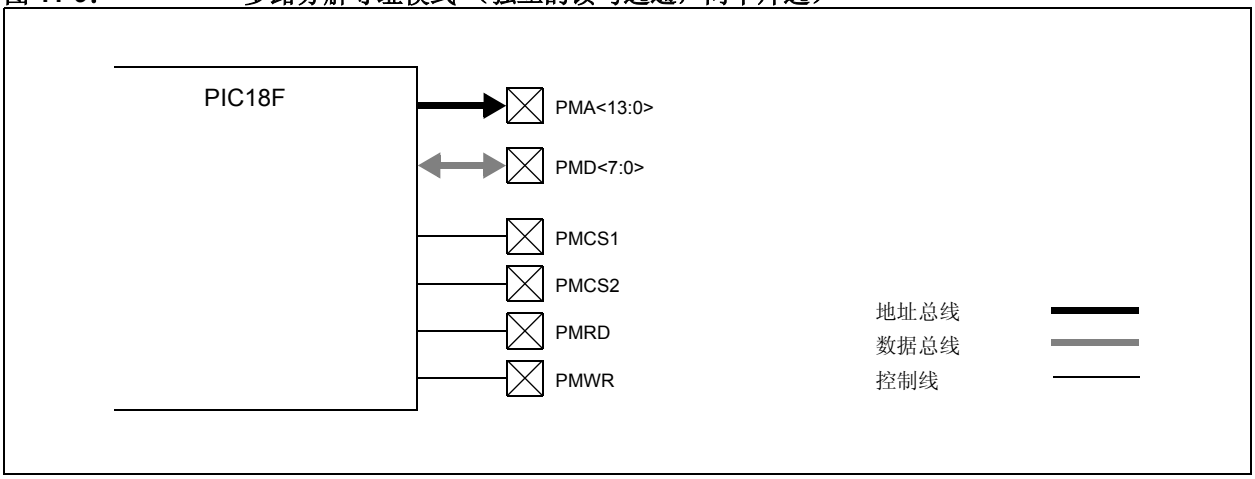


图 11-10：　　部分复用的寻址模式 （独立的读写选通，两个片选）

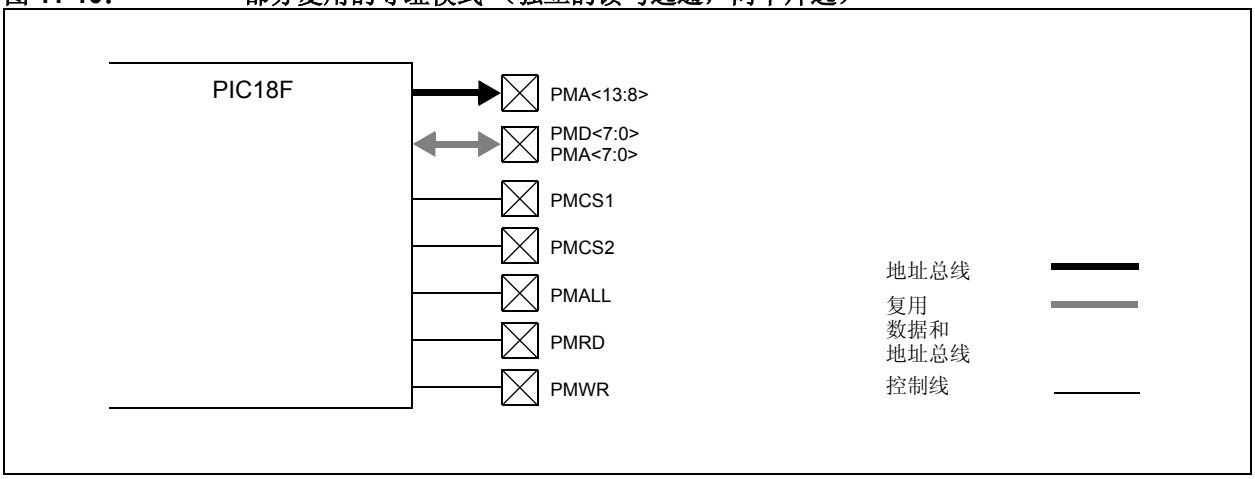
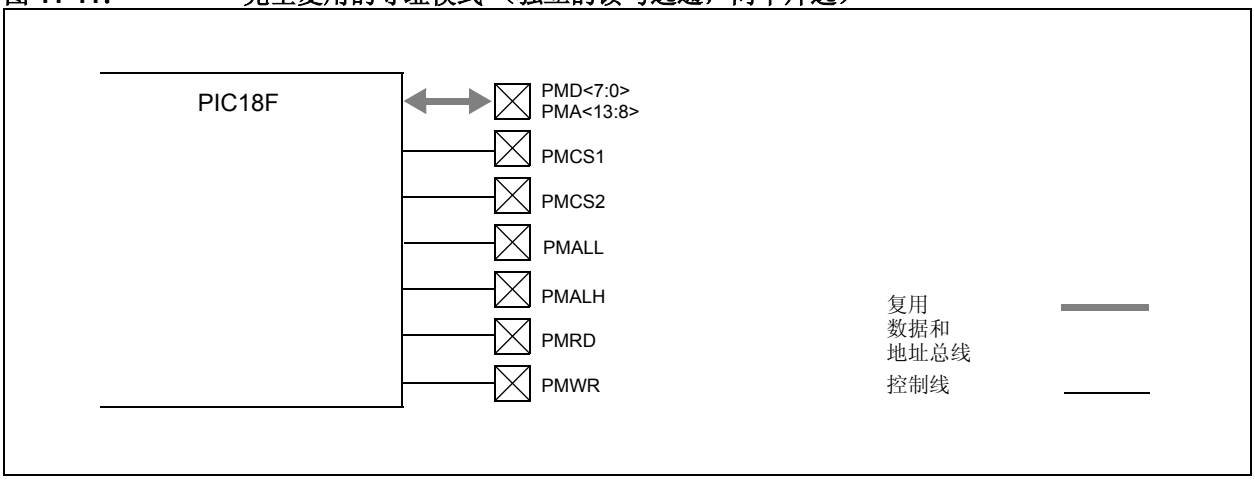


图 11-11：　　完全复用的寻址模式 （独立的读写选通，两个片选）



11.3.5 片选功能

PMP 主控模式下最多有两根片选信号线 PMCS1 和 PMCS2。这两根片选信号线与地址总线的两个最高有效位（PMADDRH<6> 和 PMADDRH<7>）复用。当引脚被配置为片选时，该引脚地址不会自动递增 / 递减。片选信号的功能通过片选功能位（PMCONL <7:6>）进行配置。

11.3.6 自动递增 / 递减

当模块工作在其中一个主控模式下时，INCM 位（PMMODEH<3:4>）控制地址值的变化。可以将地址配置为在每个读写操作之后自动递增或递减。在每完成一次操作后地址都会递增一次，且 BUSY 位变为 0。如果片选信号被禁止并被配置为地址位，则这些位将进行递增和递减操作；否则， CS2 和 CS1 位的值将不受影响。

11.3.7 等待状态

在主控模式中，用户可以通过配置模块等待状态控制读写周期和地址周期。使用 PMMODEL 寄存器中相应的 WAITBx、 WAITMx 和 WAITEx 位可以配置周期的三个部分，开始、中间和末尾。

WAITB1:WAITB0 位 （PMMODEL<7:6>）用于设置模式 10 下 PMRD/PMWT 选通之前，或模式 11 下 PMENB 选通之前数据建立等待的周期数。WAITM3:WAITM0 位（PMMODEL<5:2>）用于设置模式 10 下 PMRD/PMWT 选通后，或模式 11 下 PMENB 选通后等待的周期数。当等待状态被设置为 0 时，则 WAITB 和 WAITE 无效。WAITE1:WAITE0 位 （PMMODEL<1:0>）用于设置模式 10 下 PMRD/PMWT 选通后，或模式 11 下 PMENB 选通后数据保持时间等待的周期数。

11.3.8 读操作

要对并行主端口执行读操作，用户需要读 PMDIN1L 寄存器。这使PMP在片选线和地址总线上输出期望的值。然后选通线 （PMRD）被选通。所读取的数据被放到 PMDIN1L 寄存器中。

如果使能 16 位模式 （MODE16 = 1），则读 PMDIN1L 寄存器的低字节将启动两次总线读操作。第一个读取的字节被放到 PMDIN1L 寄存器中，第二个读取的字节被放到 PMDIN1H 寄存器中。

注意，从 PMDIN1L 寄存器读取的数据实际上是来自先前读取操作中读取的值。因此，用户执行的第一次读取操作是伪读操作，该读操作启动第一次总线读操作并填充读寄存器。同样，只有在检测到 BUSY 位为低电平之后，所请求的读取值才就绪。因此，在背对背读取操作中，两次从寄存器中读取的数据都是一样的。寄存器的下一次读取将产生新值。

11.3.9 写操作

要向并行总线执行写操作，用户需要写 PMDIN1L 寄存器。这使模块首先在片选线和地址总线上输出期望的值。PMDIN1L寄存器的写操作数据被放置在PMD<7:0>数据总线上。然后选通写操作线（PMWR）。如果使能 16 位模式 （MODE16 = 1），则写 PMDIN1L 寄存器将启动两次总线写操作。第一次写操作将由 PMDIN1L 中包含的数据组成，而第二次写操作将包含 PMDIN1H 寄存器。

11.3.10 并行主端口状态

11.3.10.1 BUSY 位

除了 PMP 中断，还提供了 BUSY 位来表明模块的状态。该位仅用在主控模式下。正在进行读或写操作时，除了操作的最后一个 CPU 周期，全部 BUSY 位都被置 1。如果请求一个单周期读或写操作，则 BUSY 位将永远不会有效。这就允许背对背传输。当该位被置 1 时，将忽略用户的任何启动新操作请求 （即，读或写 PMDIN1L 寄存器的低字节均不会启动读或写操作）。

11.3.10.2 中断

当主控模式下允许 PMP 模块中断时，模块将在每完成一个读或写周期时产生中断；否则，可通过 BUSY 位查询模块的状态。

© 2007 Microchip Technology Inc.

11.3.11 主控模式时序

本部分包含与常用主控模式配置选项相对应的许多时序示例。这些选项从 8 位到 16 位数据、从完全多路分解的地址到完全复用的地址，以及等待状态。

图 11-12： 读写时序， 8 位数据，多路分解地址

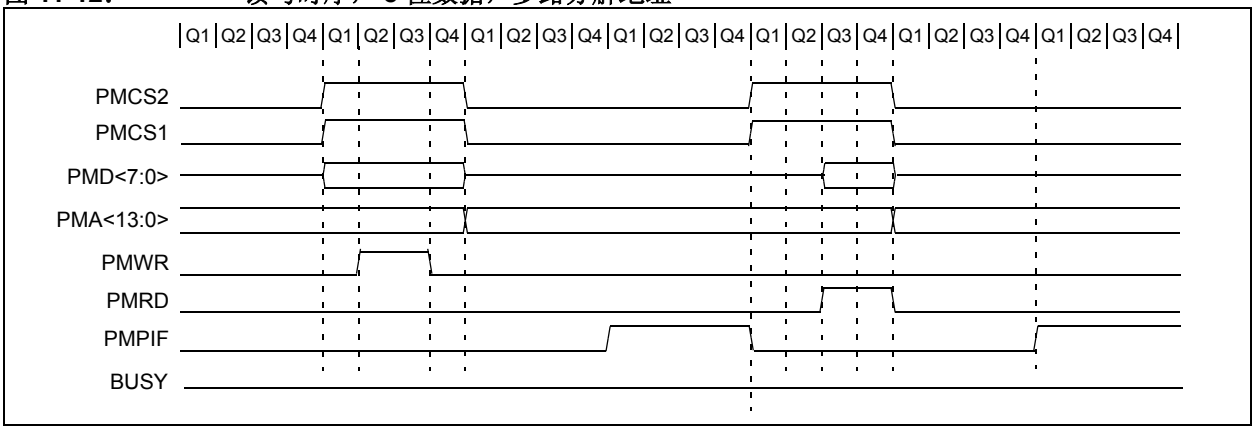


图 11-13： 读时序， 8 位数据，部分复用地址

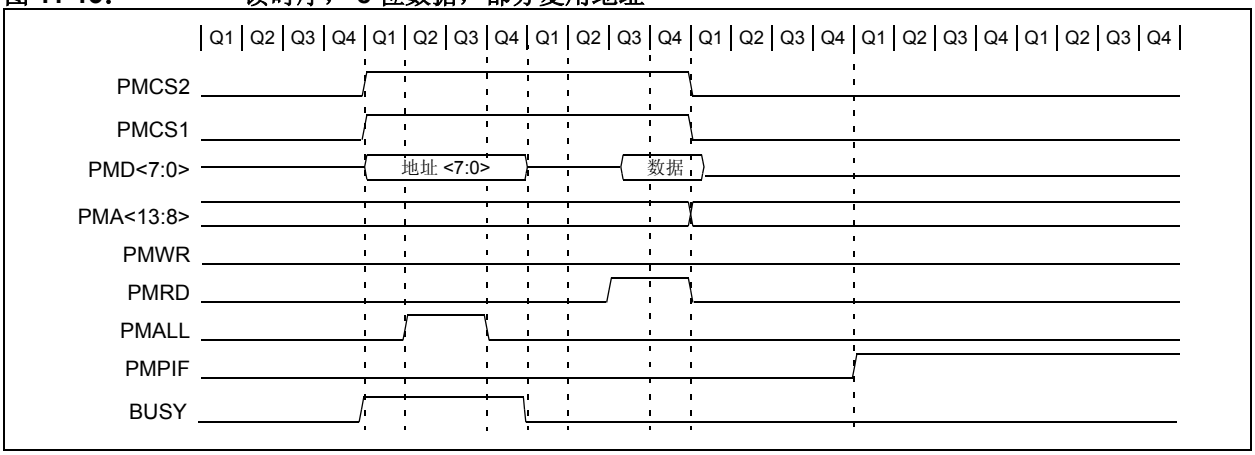


图 11-14： 读时序， 8 位数据，使能等待状态，部分复用地址

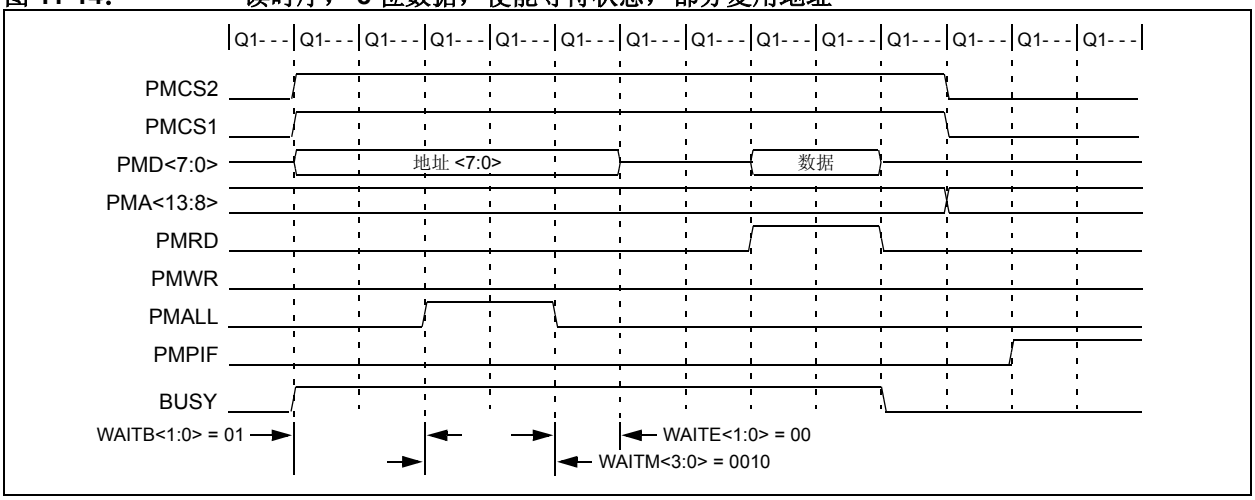


图 11-15：　　写时序， 8 位数据，部分复用地址

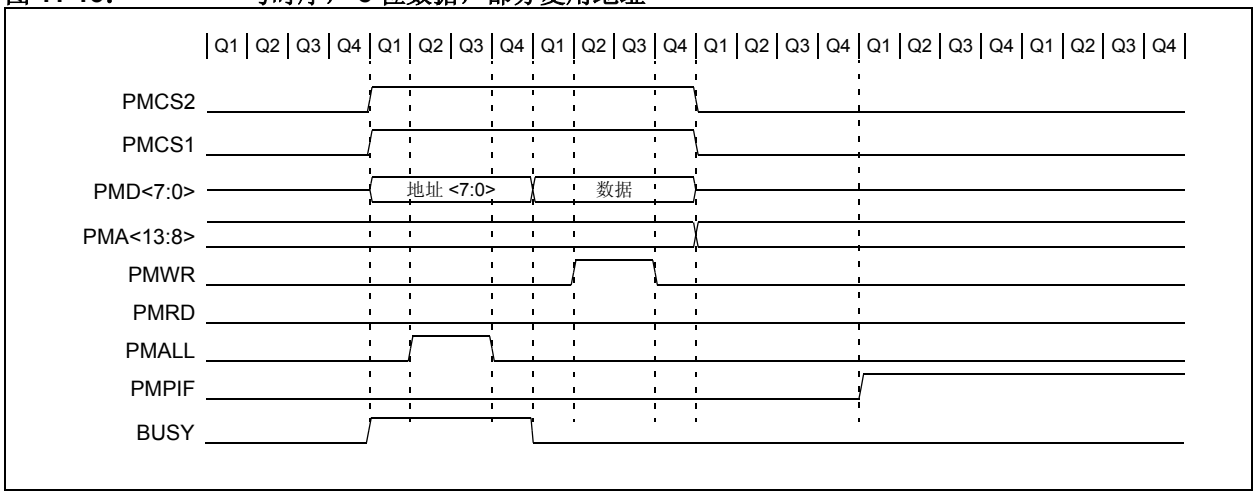


图 11-16：　　写时序， 8 位数据，使能等待状态，部分复用地址

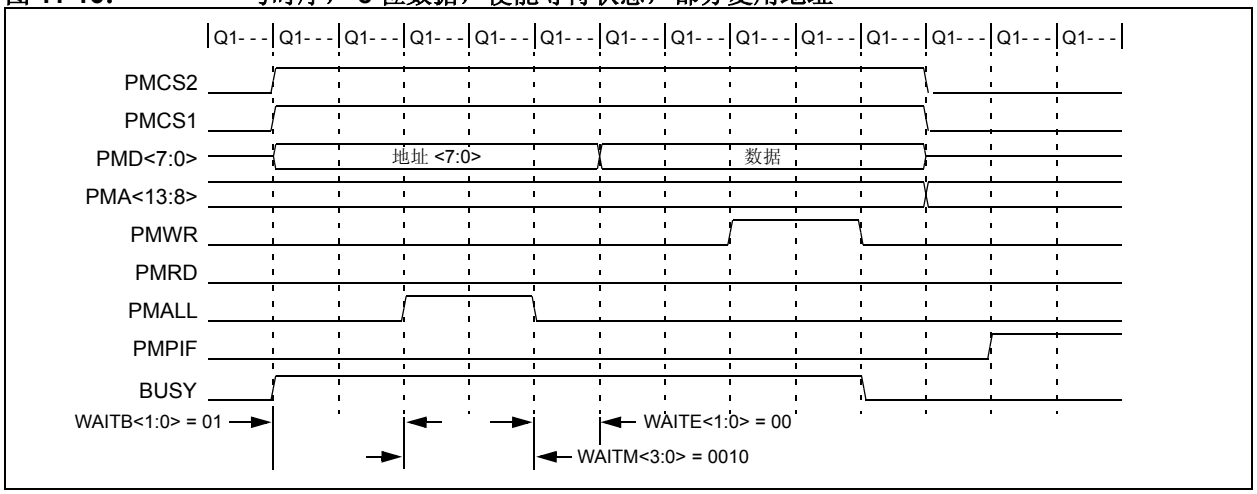
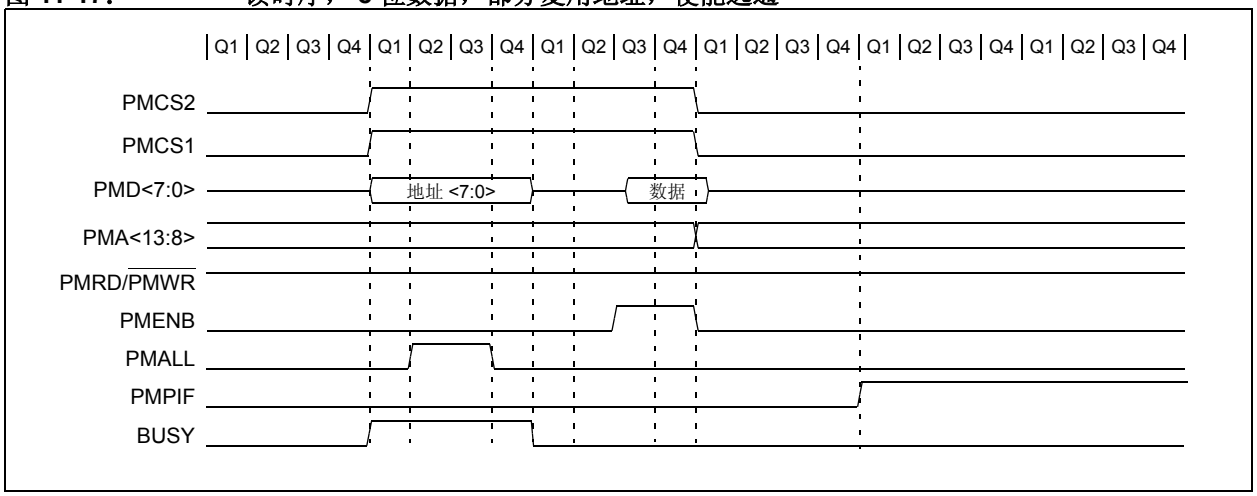


图 11-17：　　读时序， 8 位数据，部分复用地址，使能选通



© 2007 Microchip Technology Inc.

图 11-18：　　写时序， 8 位数据，部分复用地址，使能选通

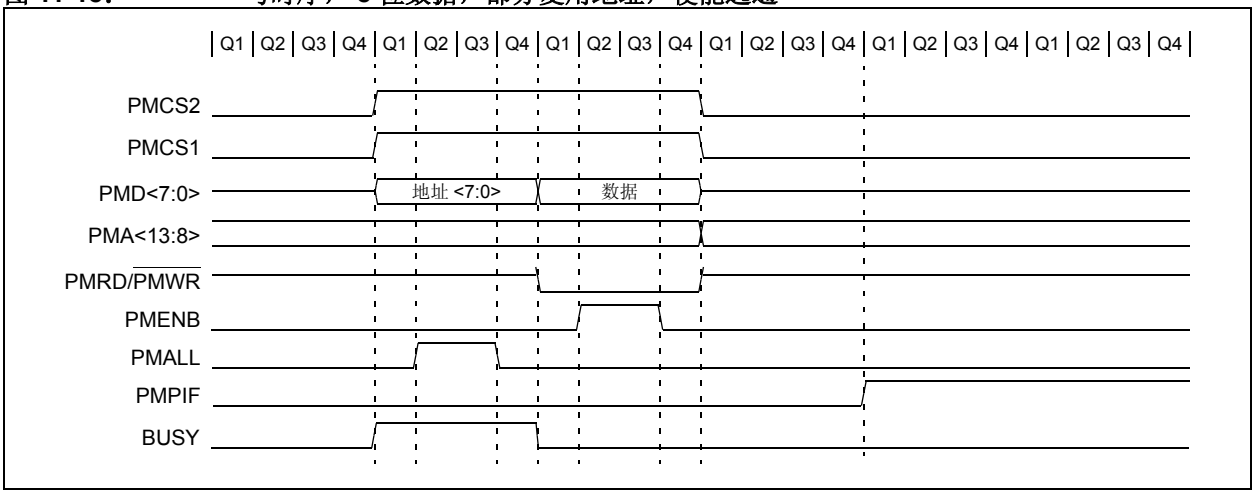


图 11-19：　　读时序， 8 位数据，部分复用 16 位地址

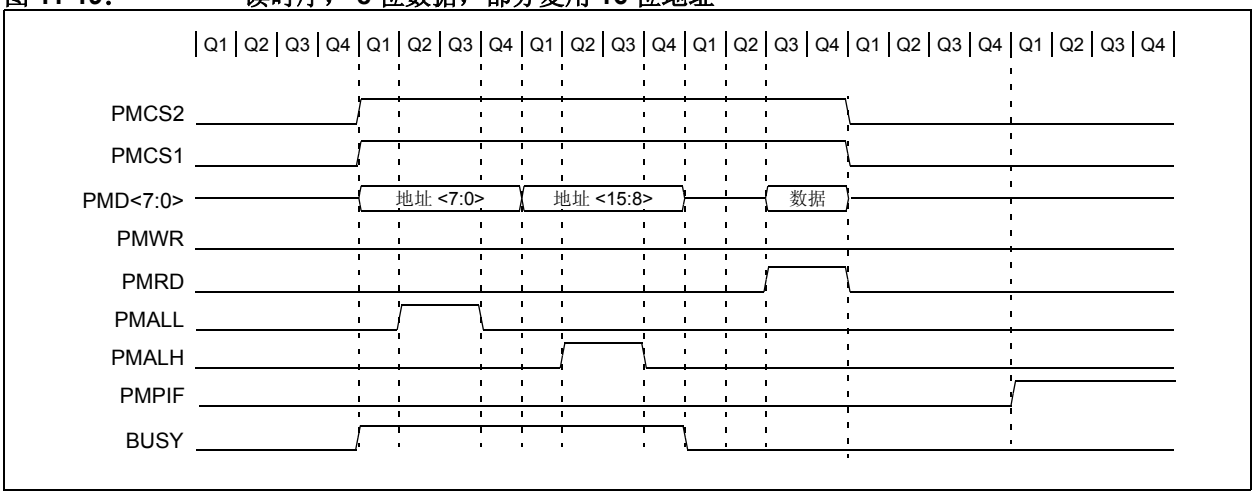


图 11-20：　　写时序， 8 位数据，完全复用 16 位地址

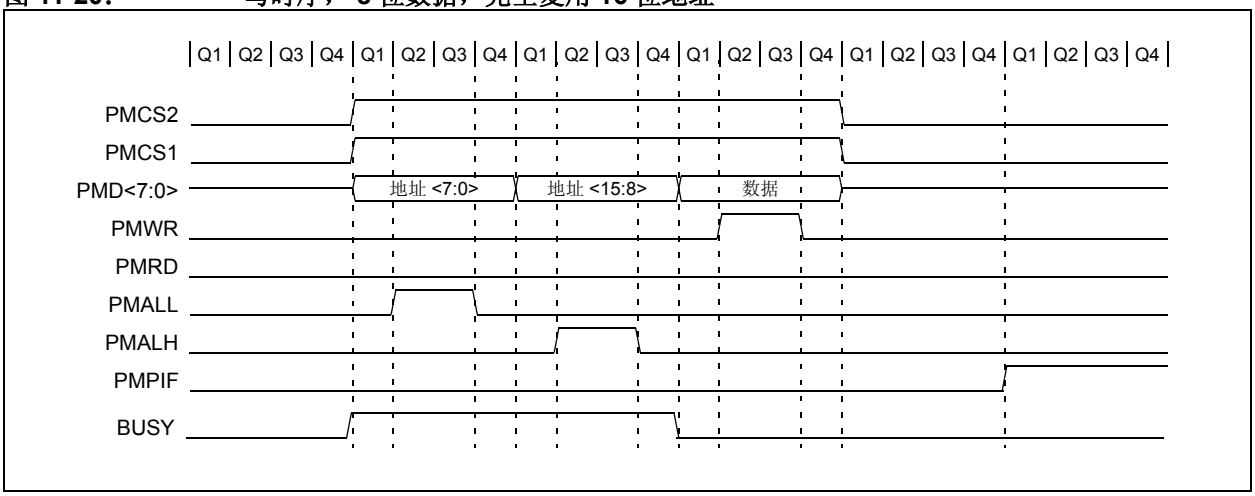


图 11-21：　　读时序， 16 位数据，多路分解地址

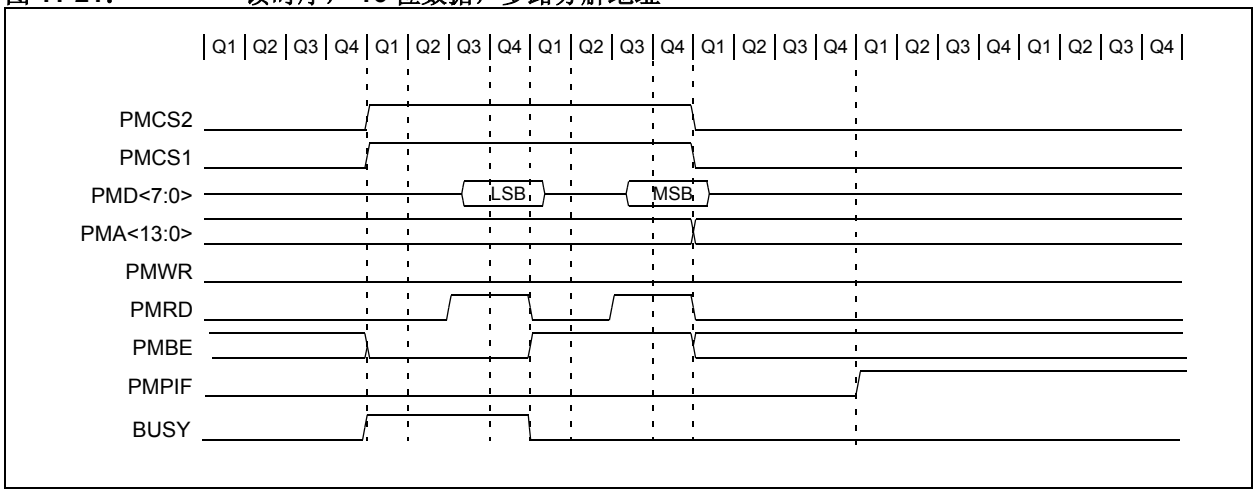


图 11-22：　　写时序， 16 位数据，多路分解地址

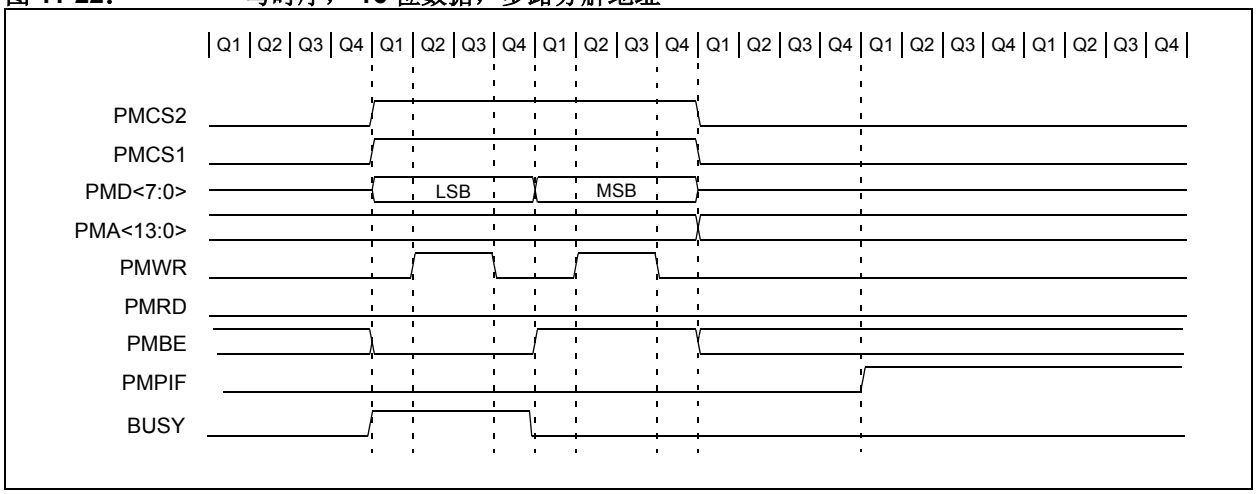
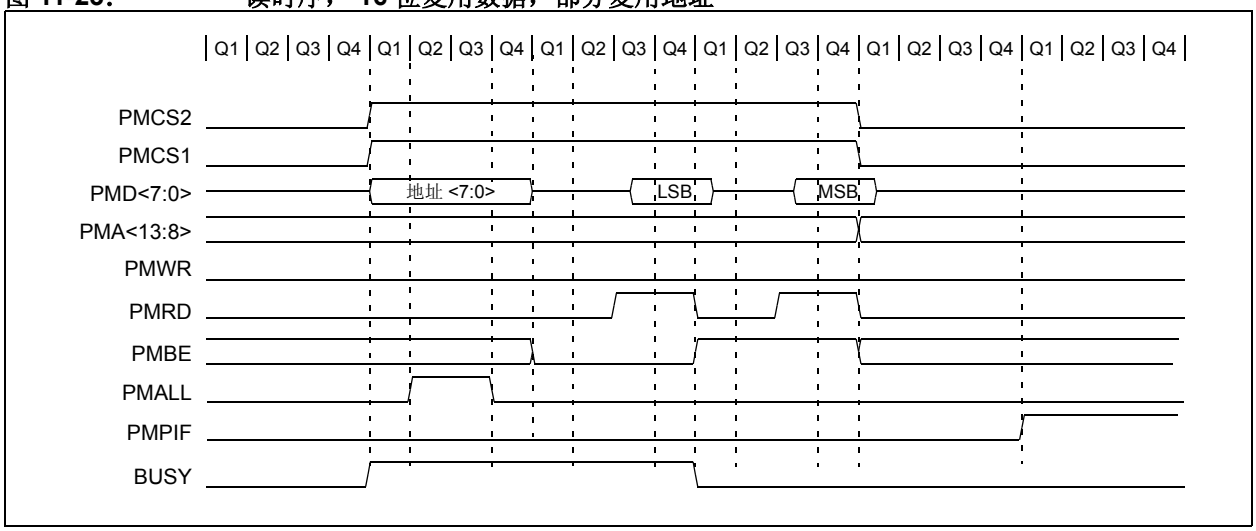


图 11-23：　　读时序， 16 位复用数据，部分复用地址



© 2007 Microchip Technology Inc.

图 11-24：　　写时序， 16 位复用数据，部分复用地址

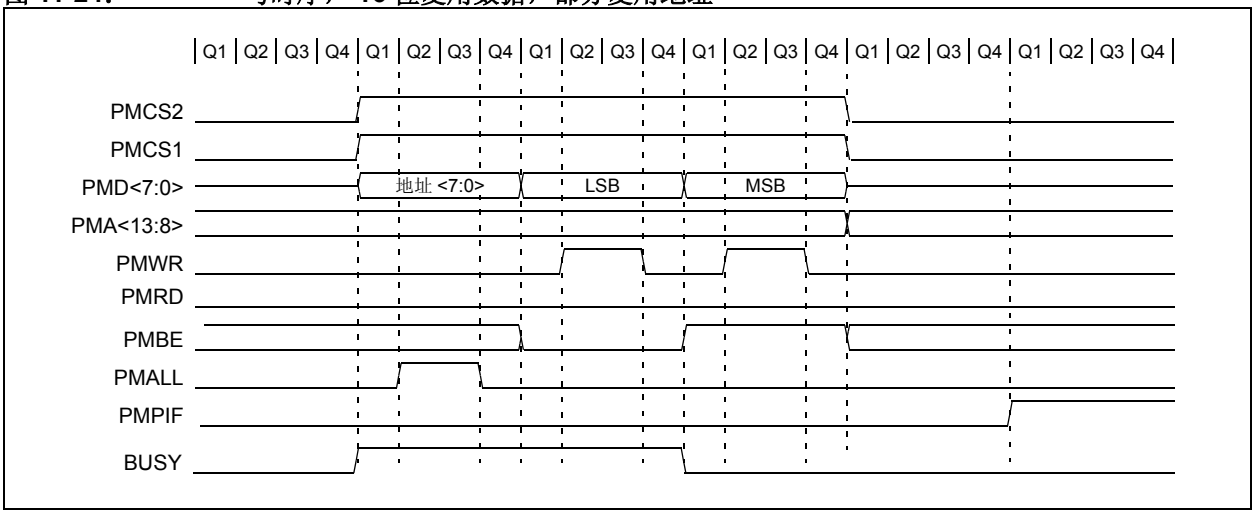


图 11-25：　　读时序， 16 位复用数据，完全复用的 16 位地址

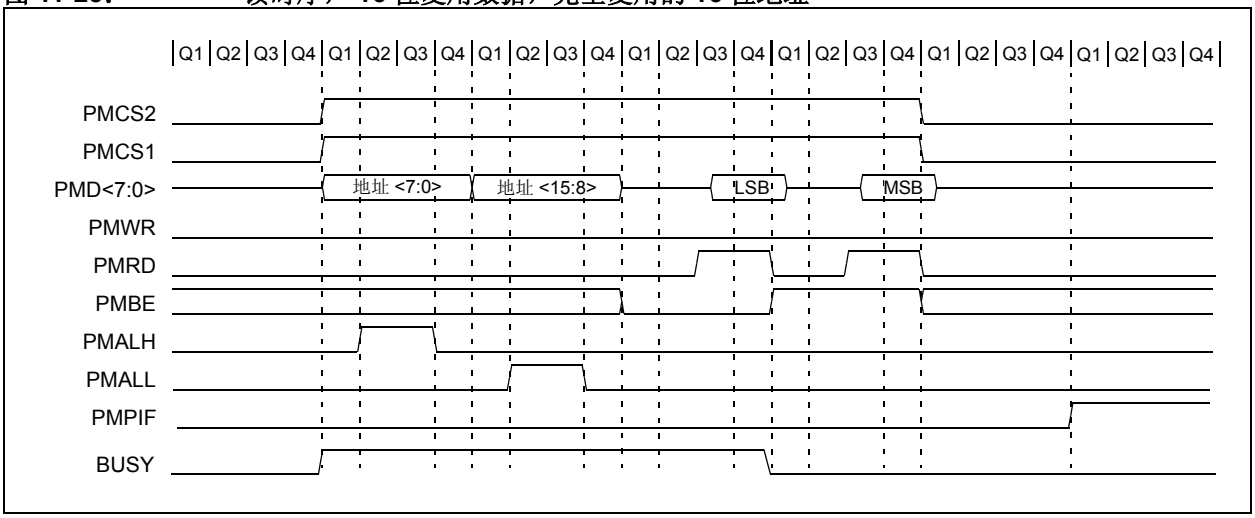
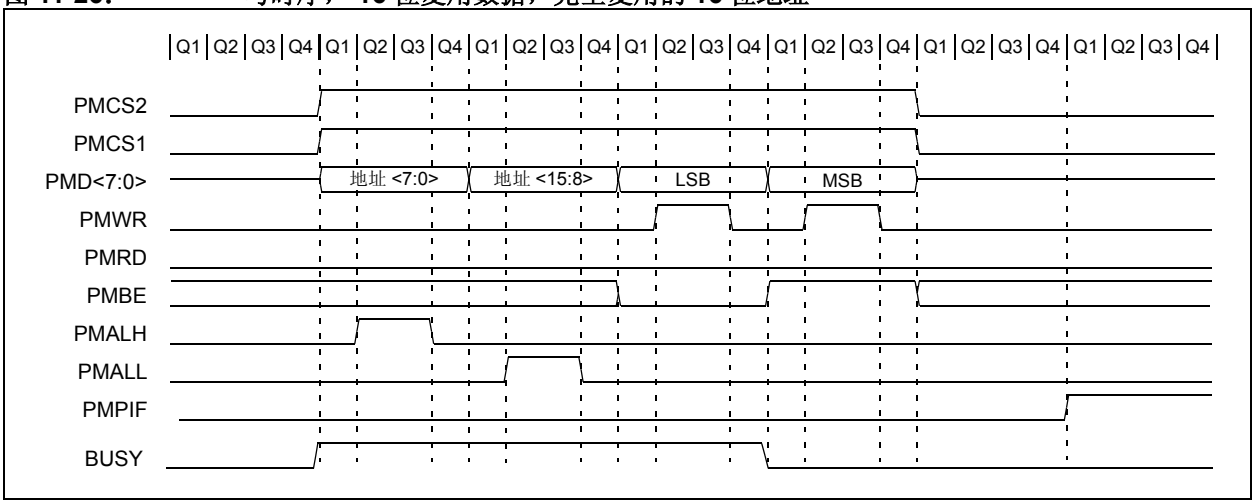


图 11-26：　　写时序， 16 位复用数据，完全复用的 16 位地址



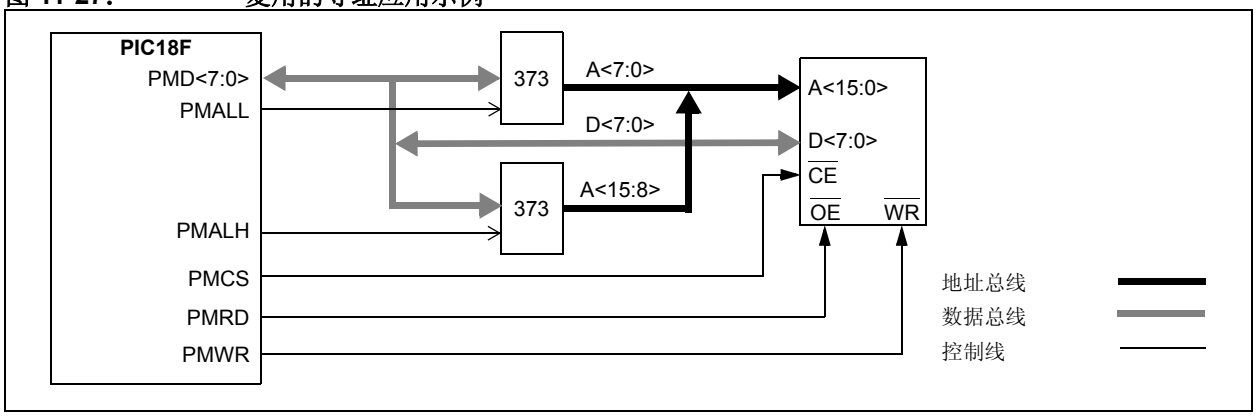
11.4 应用示例

本节介绍了一些可使用 PMP 模块的应用。

11.4.1 复用的存储器或外设

图 11-27 所示为存储器和其他可寻址外设在完全复用模式下的连接。因此，从单片机角度来看，该模式最能节省引脚。但是为了实现此配置，需要使用一些外设锁存器来保持地址。

图 11-27： 复用的寻址应用示例



11.4.2 部分复用的存储器或外设

部分复用需要使用较多引脚；但是增加几个引脚可实现一些额外功能。

图 11-28 给出了与外部锁存器部分复用的存储器或外设的示例。如果外设有如图 11-29 所示的内部锁存器，则除了外设之外不再需要额外的电路。

图 11-28： 部分复用的寻址应用示例

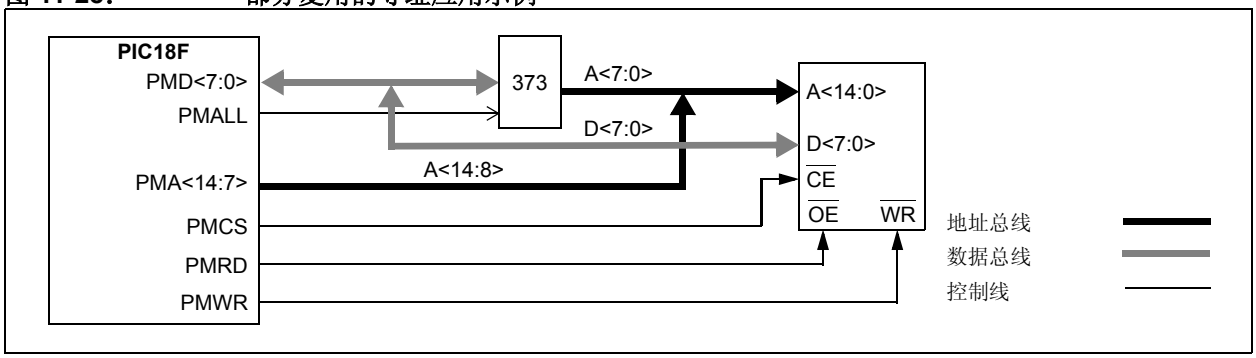
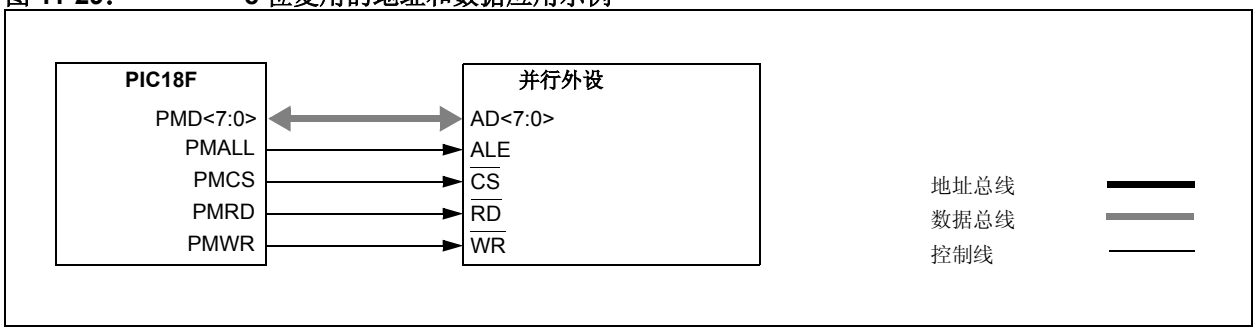


图 11-29： 8 位复用的地址和数据应用示例



© 2007 Microchip Technology Inc.

11.4.3 并行 EEPROM 示例

图 11-30 所示为将并行 EEPROM 连接到 PMP 的示例。图 11-31 对此稍作改动，配置为与一个 EEPROM 进行 16 位数据通信。

图 11-30： 并行 EEPROM 示例 （最多 15 位地址， 8 位数据）

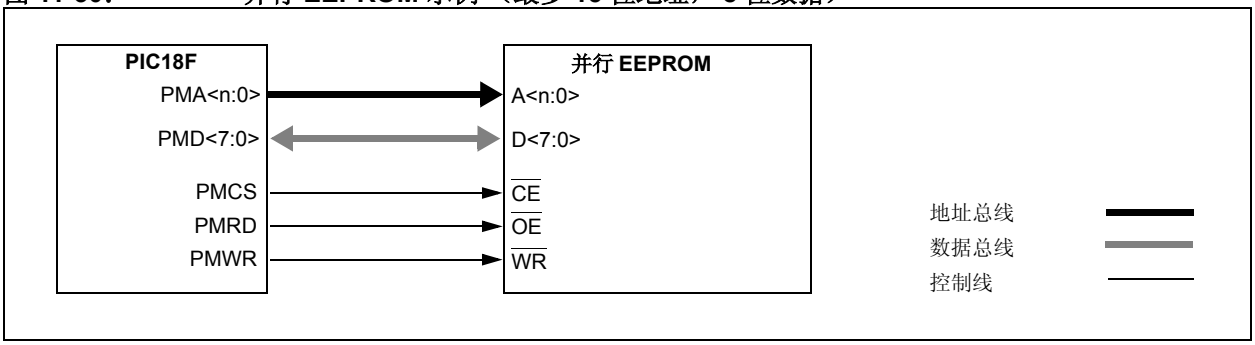
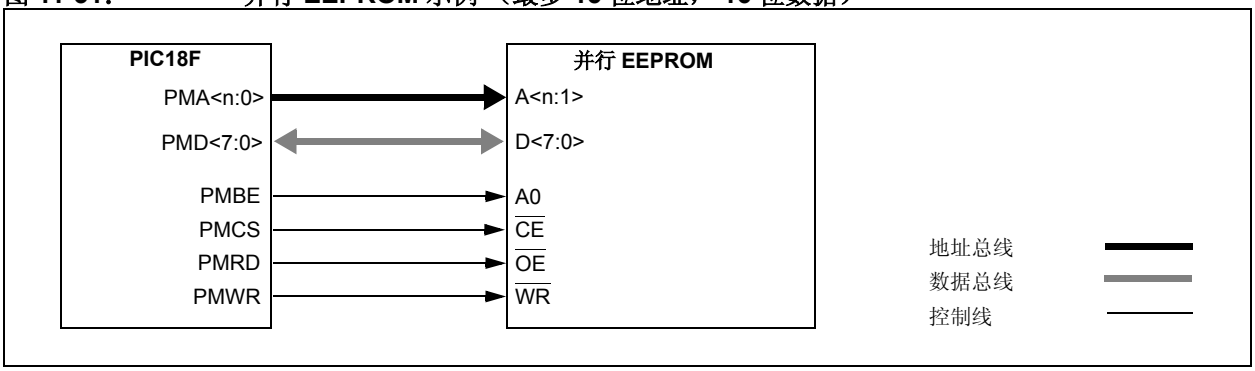


图 11-31： 并行 EEPROM 示例 （最多 15 位地址， 16 位数据）



11.4.4 LCD 控制器示例

可以将 PMP 模块配置为连接典型的 LCD 控制器接口，如图 11-32 所示。在此例中，由于常见的 LCD 显示需要高电平有效控制，因此 PMP 模块被配置为控制信号高电平有效。

图 11-32： LCD 控制示例 （字节模式操作）

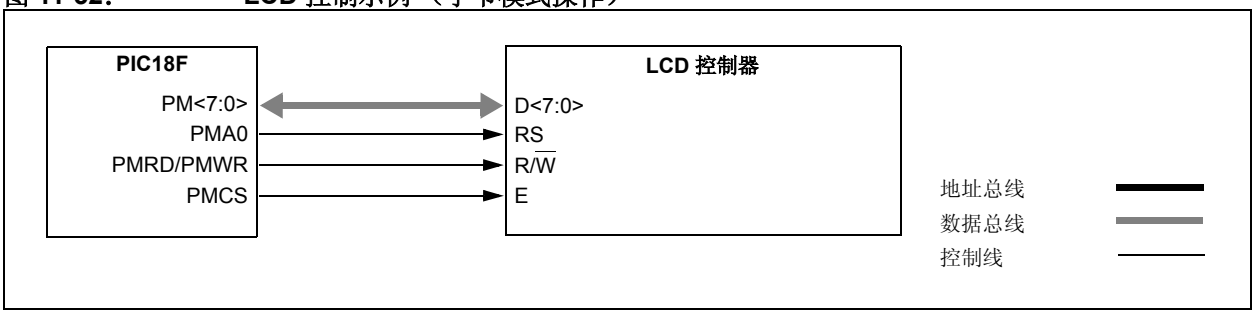


表 11-3：　　与 PMP 模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PMCONH：	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	60
PMCONL	CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP	60
PMADDRH/	CS2	CS1	并行主端口地址，高字节						60
PMDOUT1H(1)	并行端口输出数据高字节 （缓冲器 1）								60
PMADDRL/	并行主端口地址低字节								60
PMDOUT1L(1)	并行端口输出数据低字节 （缓冲器 0）								60
PMDOUT2H	并行端口输出数据高字节 （缓冲器 3）								60
PMDOUT2L	并行端口输出数据低字节 （缓冲器 2）								60
PMDIN1H	并行端口输入数据高字节 （缓冲器 1）								60
PMDIN1L	并行端口输入数据低字节 （缓冲器 0）								60
PMDIN2H	并行端口输入数据高字节 （缓冲器 3）								60
PMDIN2L	并行端口输入数据低字节 （缓冲器 2）								60
PMMODEH	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	60
PMMODEL	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	60
PMEH	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	60
PMEL	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	60
PMSTATH	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	60
PMSTATL	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	60
PADCFG1(2)	—	—	—	—	—	—	—	PMPTTL	56

图注：　— = 未实现，读为 0。在 PMP 运行期间不使用阴影单元。

注　1： PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L 寄存器对共享物理寄存器和地址，但具有不同的功能，且这些功能由模块的操作模式所决定。

2： 配置 SFR 的地址与默认 SFR 的地址重叠；且仅当 WDTCON<4> = 1 时才可访问配置 SFR。

© 2007 Microchip Technology Inc.

12.0 TIMER0 模块

Timer0 模块具有以下特性：

- 可通过软件选择，作为 8 位或 16 位定时器 / 计数器运行
- 可读写的寄存器
- 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 12-1）控制模块工作的所有方面，包括预分频比的选择。该寄存器是可读写的。

图 12-1 所示为 8 位模式下 Timer0 模块的简化框图，而图 12-2 所示为 16 位模式下 Timer0 模块的简化框图。

寄存器 12-1： 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 = 未实现位，读为 0	
-n = 上电复位时的值	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 引脚上的传输信号作为时钟源
0 = 内部指令周期时钟 （CLKO）作为时钟源

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 时钟输入信号来自预分频器的输出。

bit 2-0 **T0PS2:T0PS0**：Timer0 预分频器选择位
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

12.1 Timer0 工作原理

Timer0 既可用作定时器亦可用作计数器。具体模式由 T0CS 位（T0CON<5>）选择。在定时器模式下（T0CS = 0），除非选择了不同的预分频值，否则，默认情况下在每个时钟周期该模块都会递增（见**第 12.3 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，它将不再递增。用户可通过将调整值写入 TMR0 寄存器达到上述目的。

通过将 T0CS 位置 1 选择计数器模式。在计数器模式下，Timer0 可在 RA4/T0CKI 引脚信号的每个上升沿或下降沿递增。触发递增的边沿由 Timer0 时钟源边沿选择位 T0SE（T0CON<4>）决定；清零此位选择上升沿。下面讨论外部时钟输入的限制条件。

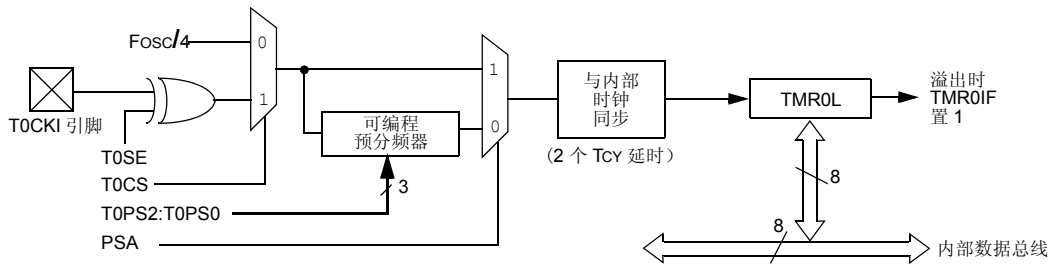
可以使用外部时钟源来驱动 Timer0；但是必须满足一定的要求以确保外部时钟与内部相位时钟（TOSC）同步。在同步之后，定时器 / 计数器仍需要一定的延时才会引发递增操作。

12.2 Timer0 的 16 位读写模式

在 16 位模式下，TMR0H 并非 Timer0 的高字节。它实际上是被缓存的 Timer0 的高字节，不可以直接读写（见图 12-2）。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（在分高、低字节两次连续读取的情况下，由于可能存在进位，因此需要验证读到字节的有效性）。

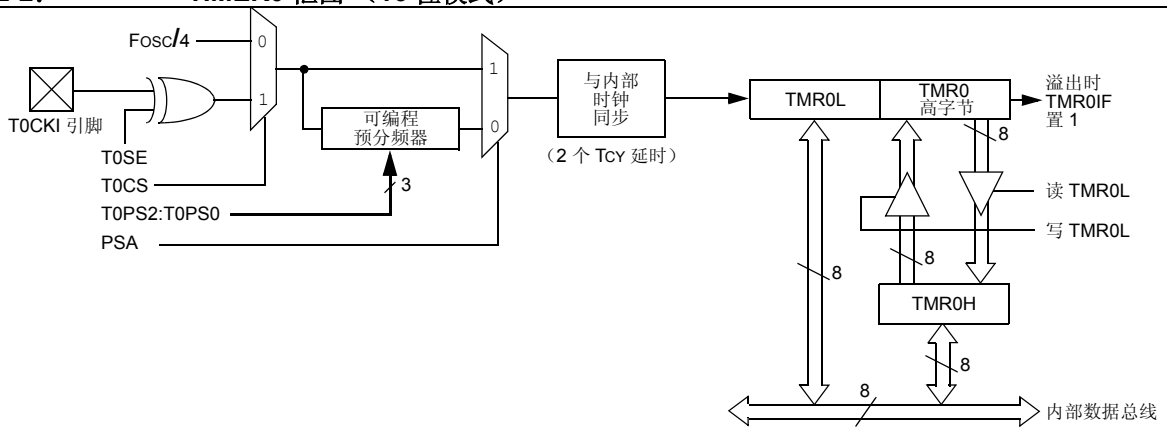
同样，可以使用 TMR0H 缓冲寄存器写入 Timer0 的高字节。在写入 TMR0L 的同时，使用 TMR0H 的内容更新 Timer0 的高字节。这样一次就可以完成 Timer0 全部 16 位的更新。

图 12-1： 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 12-2： 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.

12.3 预分频器

Timer0 模块的预分频器为一个 8 位计数器。不可直接读写该预分频器。其值通过 PSA 和 T0PS2:T0PS0 位（T0CON<3:0>）来设置，此位决定预分频器的分配和预分频比。

将 PSA 位清零可将预分频器分配给 Timer0 模块。在分配预分频器时，预分频比可以从 1:2 到 1:256 之间进行选择，该预分频比以 2 的次幂递增。

将预分频器分配给 Timer0 模块时，所有对 TMR0L 寄存器进行写入操作的指令（如 CLRF TMR0、MOVWF TMR0 和 BSF TMR0 等）都将清零预分频器的计数值。

注：	当将预分频器分配给Timer0时写入TMR0L会将预分频器的计数值清零，但不会改变预分频器的分配。

12.3.1 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。

12.4 Timer0 中断

8 位模式下，TMR0 寄存器发生从 FFh 到 00h 的溢出，或 16 位模式下，TMR0 寄存器发生从 FFFFh 到 0000h 的溢出，都将产生 TMR0 中断。这种溢出会将标志位 TMR0IF 置 1。清零 TMR0IE 位（INTCON<5>）可屏蔽此中断。在重新使能该中断前，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 12-1： 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	复位值所在的页
TMR0L	Timer0 寄存器的低字节								56
TMR0H	Timer0 寄存器的高字节								56
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	56
TRISA	TRISA7(1)	TRISA6(1)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58

图注： — = 未实现，读为 0。 Timer0 不使用阴影单元。

注 1： 这些位仅在选择振荡器模式（FOSC2 配置位 = 0）时才可用；否则，它们都处于未用状态。

注：

 © 2007 Microchip Technology Inc.

13.0 TIMER1 模块

Timer1 定时器 / 计数器模块具有以下特征：

- 可通过软件选择，作为 16 位定时器或计数器运行
- 可读写的 8 位寄存器 （TMR1H 和 TMR1L）
- 可选择使用器件时钟作为 （内部或外部）时钟源或选择 Timer1 振荡器作为内部时钟源
- 溢出时中断
- 在触发 ECCPx 特殊事件时复位
- 器件时钟状态标志位 （T1RUN）

图 13-1 所示为 Timer1 模块的简化框图。图 13-2 所示为此模块在读 / 写模式下的工作原理框图。

该模块自身具有低功耗振荡器，可提供额外的时钟选择。Timer1 振荡器也可用作单片机处于功耗管理工作时的低功耗时钟源。

Timer1 还可为应用提供实时时钟 （RTC）功能，而仅需增加极少的外部元件和代码开销。

Timer1 由 T1CON 控制寄存器 （寄存器 13-1）控制。Timer1 振荡器使能位 （T1OSCEN）也在该寄存器中。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 13-1： T1CON：TIMER1 控制寄存器 (1)

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 = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **RD16**：16 位读 / 写模式使能位
1 = 使能通过一次 16 位操作对 Timer1 寄存器进行读写
0 = 使能通过两次 8 位操作对 Timer1 寄存器进行读写

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 = 不与外部时钟输入同步
0 = 与外部时钟输入同步
当 TMR1CS = 0 时：
忽略此位。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/T13CKI 引脚上的外部时钟 （上升沿触发计数）
0 = 内部时钟 （FOSC/4）

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

注 1： 当 WDTCON<4> = 0 时，此处的默认 （旧） SFR 可用。

13.1 Timer1 工作原理

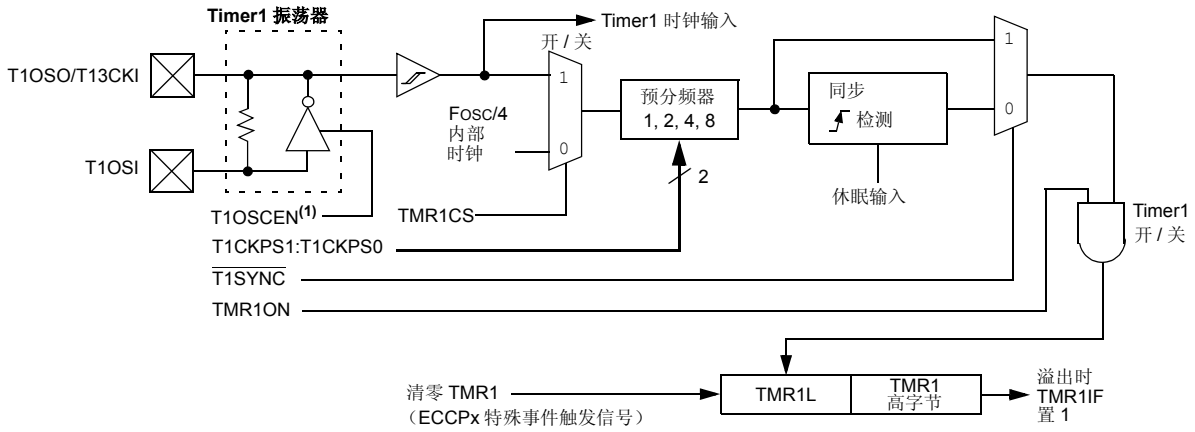
Timer1 可工作在以下模式：

- 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 TMR1CS 位置 1 时，Timer1 在 Timer1 外部时钟输入信号或 Timer1 振荡器（如果使能）输出信号的每个上升沿递增。

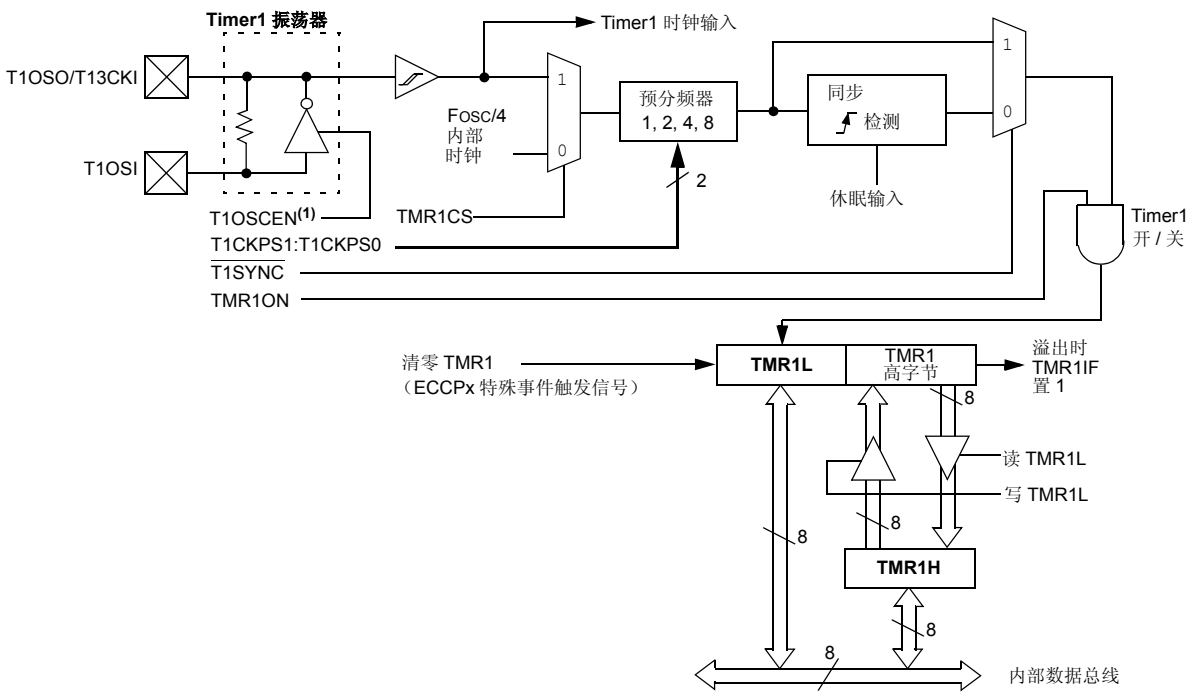
当使能 Timer1 时，RC1/T1OSI 和 RC0/T1OSO/T13CKI 引脚变为输入引脚。这意味着 TRISC<1:0> 的值被忽略并且这些引脚的读取值为 0。

图 13-1：　　TIMER1 框图



注　1：当使能位 T1OSCEN 清零时，将关闭振荡器反相器和反馈电阻以降低功耗。

图 13-2：　　TIMER1 框图（16 位读 / 写模式）



注　1：当使能位 T1OSCEN 清零时，将关闭振荡器反相器和反馈电阻以降低功耗。

© 2007 Microchip Technology Inc.

13.2 Timer1 16 位读 / 写模式

可将 Timer1 配置为 16 位读写模式（见图 13-2）。当 RD16 控制位（T1CON<7>）置 1 时，TMR1H 的地址被映射到 Timer1 高字节的缓冲寄存器。从 TMR1L 的读操作将把 Timer1 的高字节内容装入 Timer1 高字节缓冲寄存器。这种方式使用户可以精确地读取 Timer1 的全部 16 位，而无需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。

对 Timer1 的高字节进行写操作也必须通过 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。这样允许用户将所有的 16 位值一次写入 Timer1 的高字节和低字节。

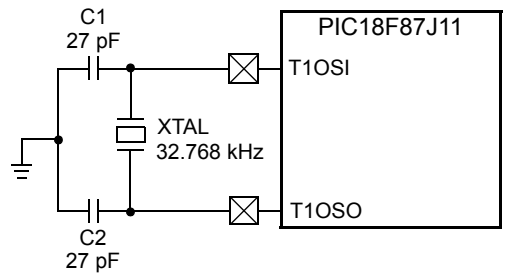
在这一模式下不能直接读写 Timer1 的高字节。所有读写都必须通过 Timer1 高字节缓冲寄存器进行。写入 TMR1H 不会清零 Timer1 预分频器。只有在写 TMR1L 时才会清零该预分频器。

13.3 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。通过将 Timer1 振荡器使能位 T1OSCEN（T1CON<3>）置 1 可使能该振荡电路。此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 13-3 所示为典型的 LP 振荡电路。表 13-1 为 Timer1 振荡器的电容选择。

用户必须提供软件延时来确保 Timer1 振荡器的正常起振。

图 13-3： TIMER1 LP 振荡器的外部元件



注： 如需了解电容选择的更多信息，请参见表 13-1 下的“注”。

表 13-1： TIMER振荡器(2,3,4)的电容选择

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值。

2： 选用较大的电容值虽然可以提高振荡器的稳定性，但同时也会延长起振时间。

3： 由于谐振器 / 晶振的特性各不相同，因此用户应当向谐振器 / 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。

13.3.1 使用 TIMER1 作为时钟源

Timer1 振荡器也可用作功耗管理模式下的时钟源。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，此时 CPU 和外设都以 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。欲知更多详情，请参见**第 3.0 节“功耗管理模式”**。

无论 Timer1 振荡器何时提供时钟源，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，那么查询 T1RUN 位可以确定时钟源是 Timer1 振荡器还是其他时钟源。

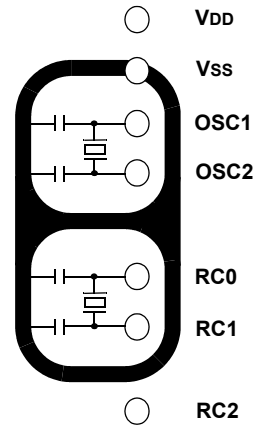
13.3.2 TIMER1 振荡器布线注意事项

Timer1 振荡电路在工作期间消耗极少的功率。由于此振荡器的低功耗特性，因此它对在极小范围内快速变化的信号也可能比较敏感。

如图 13-3 所示，振荡电路应该尽可能靠近单片机。除了 VSS 或 VDD 外，在振荡电路附近不应有其他电路。

如果必须要在该振荡器附近布置高频电路（如输出比较模式或 PWM 模式下的 ECCP1 引脚，或使用 OSC2 引脚的主振荡器），当在单面 PCB 板或外加接地层的 PCB 板上使用时，在该振荡电路周围布置接地保护环（如图 13-4 所示）可能会有所帮助。

图 13-4：　　有接地保护环的振荡电路



注：未按比例绘制。

13.4 Timer1 中断

TMR1 寄存器对（TMR1H:TMR1L）从 0000h 开始递增，一直到 FFFFh，然后溢出从 0000h 重新开始计数。如果允许 Timer1 中断，该中断就会在溢出时产生，由中断标志位 TMR1IF（PIR1<0>）表示。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。

13.5 使用 ECCPx 特殊事件触发信号来复位 Timer1

如果 ECCP1 或 ECCP2 被配置为使用 Timer1 并且在比较模式（CCPxM3:CCPxM0 = 1011）下产生特殊事件触发信号，该信号将复位 Timer3。如果使能了 A/D 模块，来自 ECCP2 的触发信号还将启动 A/D 转换（欲知更多信息，请参见**第 18.2.1 节“特殊事件触发器”**）。

为了利用这一功能，必须将 Timer1 配置为定时器或同步计数器。在这种情况下，CCPRxH:CCPRxL 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。

如果对 Timer1 的写操作和特殊事件触发同时发生，则写操作优先。

注：	来自 ECCPx 模块的特殊事件触发信号不会将中断标志位 TMR1IF（PIR1<0>）置 1。

13.6 使用 Timer1 作为实时时钟

为 Timer1 添加外部 LP 振荡器（如**第 13.3 节“Timer1 振荡器”**中所述），可以为用户应用提供 RTC 功能。只需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实现这一功能。当器件工作于休眠模式下并使用电池或超大容量电容作为电源时，RTC 功能选项可以完全避免使用独立的 RTC 器件和备用电池。

应用代码程序 RTCisr（如例 13-1 所示），给出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。将一对 TMR1 寄存器的值不断加 1 直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。当秒计数器溢出时，分计数器递增而小时计数器则会在分计数器溢出时加 1。

由于这对寄存器为 16 位宽，因此直接使用 32.768 kHz 时钟将其计数到溢出需要 2 秒。要使溢出按要求的 1 秒间隔进行，就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，因为这样做可能会引起多个周期的累积错误。

要使此方法精确，Timer1 必须工作在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 振荡器也必须使能并始终保持运行。

13.7 异步计数器模式下的注意事项

在 Timer1 中断和对 TMR1 寄存器的更新后，Timer1 模块使用其时钟源的下降沿，以在上升沿触发下一个寄存器更新。如果更新在时钟输入变为低电平后才完成，则不计算下一个上升沿。

如果应用程序在定时器输入变为低电平之前对 TMR1 进行了可靠更新，那么无需其他操作。否则，在递增 Timer1 之后要执行一个调整更新。要做到这一点，可通过在中断服务程序内监视 TMR1L 直到它发生递增，然后在时钟变为低电平时或 1.5 个时钟源周期后更新 TMR1H:TMR1L 寄存器对。假定 Timer1 正被用作实时时钟，时钟源是一个 32.768 kHz 晶体振荡器。在这种情况下，1.5 个时钟周期为 15.25 μs。

© 2007 Microchip Technology Inc.

例 13-1 中所示的实时时钟应用程序代码为 Timer1 的典型 ISR，同时还给出了在要求的间隔内未完成可靠更新时需要使用的可选代码。

例 13-1： 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     T1CON             ; Asynchronous operation, external oscillator
            CLRF      secs              ; Initialize timekeeping registers
            CLRF      mins              ;
            MOVLW     .12
            MOVWF     hours
            BSF       PIE1, TMR1IE      ; Enable Timer1 interrupt
            RETURN
RTCisr
                                        ; Insert the next 4 lines of code when TMR1
                                        ; can not be reliably updated before clock pulse goes low
            BTFSC     TMR1L,0           ; wait for TMR1L to become clear
            BRA       $-2               ; (may already be clear)
            BTFSS     TMR1L,0           ; wait for TMR1L to become set
            BRA       $-2               ; TMR1 has just incremented
                                        ; If TMR1 update can be completed before clock pulse goes low
                                        ; Start ISR here
            BSF       TMR1H, 7          ; Preload for 1 sec overflow
            BCF       PIR1, TMR1IF      ; Clear interrupt flag
            INCF      secs, F           ; Increment seconds
            MOVLW     .59               ; 60 seconds elapsed?
            CPFSGT    secs
            RETURN                      ; No, done
            CLRF      secs              ; Clear seconds
            INCF      mins, F           ; Increment minutes
            MOVLW     .59               ; 60 minutes elapsed?
            CPFSGT    mins
            RETURN                      ; No, done
            CLRF      mins              ; clear minutes
            INCF      hours, F          ; Increment hours
            MOVLW     .23               ; 24 hours elapsed?
            CPFSGT    hours
            RETURN                      ; No, done
            CLRF      hours             ; Reset hours
            RETURN                      ; Done
```

表 13-2： TIMER1 作为定时器 / 计数器时相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
TMR1L(1)	Timer1 寄存器的低字节								56
TMR1H(1)	Timer1 寄存器的高字节								56
T1CON(1)	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	56

图注： Timer1 模块不使用阴影单元。

注 1： 当 WDTCON<4> = 0 时，此处的默认 （旧） SFR 可用。

注：

 © 2007 Microchip Technology Inc.

14.0 TIMER2 模块

Timer2 模块具有以下特征：

- 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 至 1:16）
- 当 TMR2 与 PR2 匹配时中断
- 可选作为 MSSP 模块的移位时钟

此模块由 T2CON 寄存器（寄存器 14-1）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，以使功耗最低。

图 14-1 所示为模块的简化框图。

14.1 Timer2 工作原理

在正常工作中， TMR2 从 00h 开始递增计数，每个时钟周期 （Fosc/4）计数一次 4 位的计数器 / 预分频器提供了对时钟输入不分频、 4 分频和 16 分频三种预分频选项，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 的值都会与其周期寄存器 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，并驱动输出计数器 / 后分频器 （见**第 14.2 节 “Timer2 中断”**）。

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

- 对 TMR2 寄存器进行写操作
- 对 T2CON 寄存器进行写操作
- 任何方式的器件复位 （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 14-1： 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 = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **未实现：**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 = 禁止 Timer2

bit 1-0 **T2CKPS1:T2CKPS0：**Timer2 时钟预分频比选择位
00 = 预分频比为 1
01 = 预分频比为 4
1x = 预分频比为 16

14.2 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 TMR2 匹配中断，由其标志位 TMR2IF （PIR1<1>）表示。可以通过将 TMR2 匹配中断允许位 TMR2IE （PIE1<1>）置 1 来允许此中断。

可以通过后分频器控制位 T2OUTPS3:T2OUTPS0（T2CON<6:3>）在 16 个后分频比选项（从 1:1 到 1:16）中进行选择。

14.3 Timer2 输出

TMR2 的未经分频的输出主要用于 ECCPx/CCPx 模块，它用作 ECCPx/CCPx 模块在 PWM 模式下工作时的时基。

还可选择将 Timer2 用作 MSSP 模块在 SPI 模式下工作时的移位时钟源。**第 19.0 节“主控同步串口（MSSP）模块”**中提供了更多信息。

图 14-1： TIMER2 框图

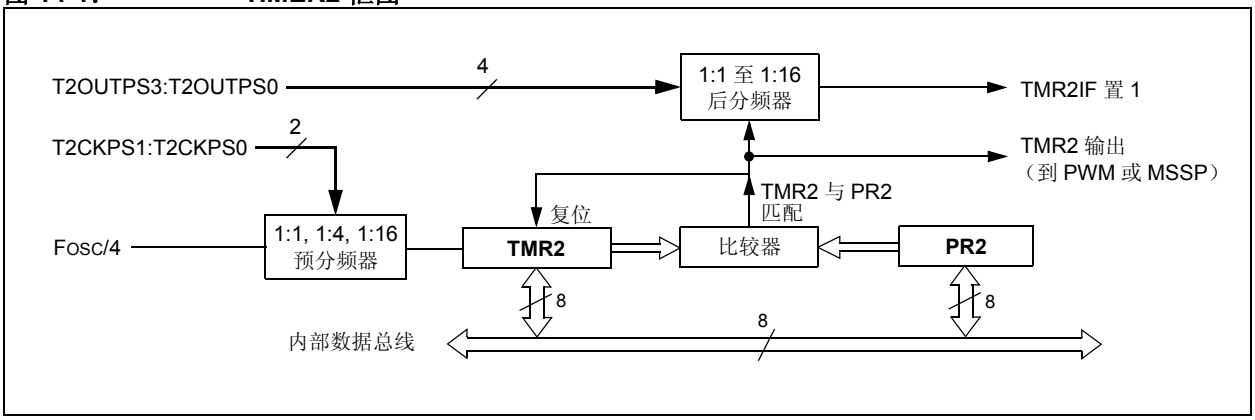


表 14-1： TIMER2 作为定时器 / 计数器时相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
TMR2(1)	Timer2 寄存器								56
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	56
PR2(1)	Timer2 周期寄存器								56

图注： - = 未实现位，读为 0。 Timer2 模块不使用阴影单元

注 1： 当 WDTCON<4> = 0 时，此处的默认 （旧） SFR 可用。

© 2007 Microchip Technology Inc.

15.0 TIMER3 模块

Timer3 定时器 / 计数器模块具有以下特征：

- 可通过软件选择，作为 16 位定时器或计数器运行
- 可读写的 8 位寄存器 （TMR3H 和 TMR3L）
- 可选择使用器件时钟作为 （内部或外部）时钟源或选择 Timer1 振荡器作为内部时钟源
- 溢出时中断
- 模块在触发 ECCPx 特殊事件时复位

图 15-1 所示为 Timer3 模块的简化框图。图 15-2 所示为此模块在读 / 写模式下的工作原理框图。

Timer3 模块由 T3CON 寄存器 （寄存器 15-1）控制。此寄存器还可为 CCP 和 ECCP 模块选择时钟源。更多信息请参见**第 17.1.1 节 “CCP 模块和定时器资源”**。

寄存器 15-1： T3CON：TIMER3 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-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **RD16**：16 位读 / 写模式使能位
1 = 使能通过一次 16 位操作对 Timer3 寄存器进行读写
0 = 使能通过两次 8 位操作对 Timer3 寄存器进行读写

bit 6,3 **T3CCP2:T3CCP1**：ECCPx/CCPx 的时钟源 （是 Timer3 还是 Timer1）使能位
11 = Timer3 和 Timer4 作为所有 ECCPx/CCPx 模块的时钟源
10 = Timer3 和 Timer4 作为 ECCP3、 CCP4 和 CCP5 的时钟源；
Timer1 和 Timer2 作为 ECCP1 和 ECCP2 的时钟源
01 = Timer3 和 Timer4 作为 ECCP2、 ECCP3、 CCP4 和 CCP5 的时钟源；
Timer1 和 Timer2 作为 ECCP1 的时钟源
00 = Timer1 和 Timer2 作为所有 ECCPx/CCPx 模块的时钟源

bit 5-4 **T3CKPS1:T3CKPS0**：Timer3 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 2 **$\overline{\text{T3SYNC}}$**：Timer3 外部时钟输入同步控制位
（不适用于器件时钟来自 Timer1/Timer3 的场合。）
当 TMR3CS=1 时：
1 = 不与外部时钟输入同步
0 = 与外部时钟输入同步
当 TMR3CS=0 时：
忽略此位。当 TMR3CS=0 时， Timer3 使用内部时钟。

bit 1 **TMR3CS**：Timer3 时钟源选择位
1 = 使用 Timer1 振荡器或 T13CKI 引脚信号作为外部时钟输入 （在第一个下降沿之后的上升沿开始计数）
0 = 内部时钟 （FOSC/4）

bit 0 **TMR3ON**：Timer3 使能位
1 = 使能 Timer3
0 = 停止 Timer3

15.1 Timer3 工作原理

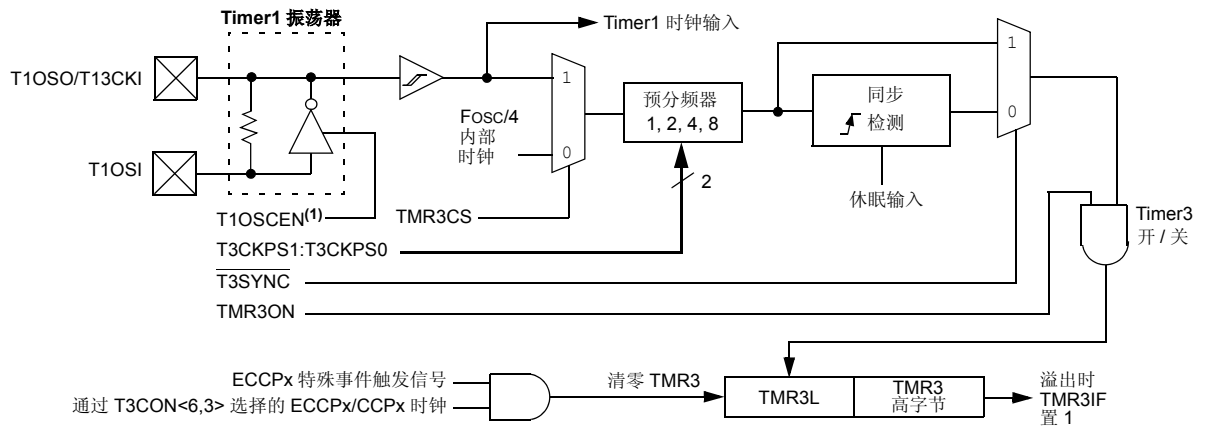
Timer3 有三种工作模式：

- 定时器
- 同步计数器
- 异步计数器

工作模式由时钟选择位 TMR3CS（T3CON<1>）决定。当 TMR3CS 清零 （= 0）时， Timer3 在每个内部指令周期（FOSC/4）递增。当 TMR3CS 位置 1 时， Timer3 在 Timer1 外部时钟输入信号或 Timer1 振荡器（如果使能）输出信号的每个上升沿递增。

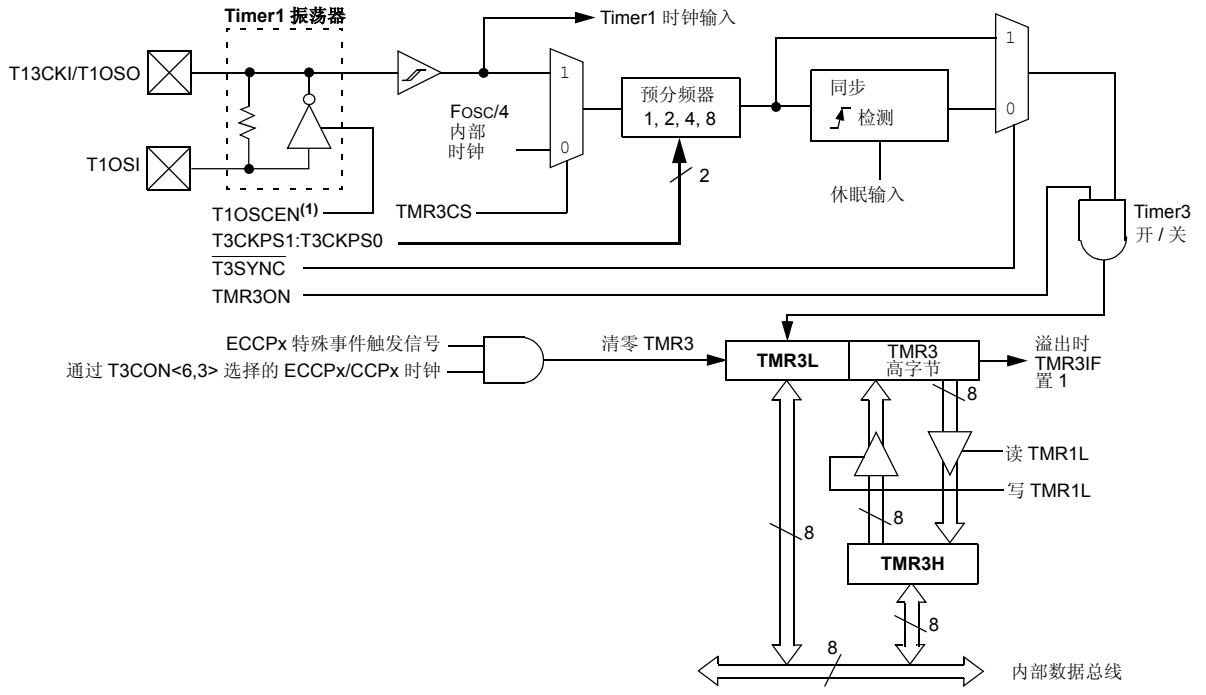
当使能 Timer1 时，RC1/T1OSI 和 RC0/T1OSO/T13CKI 引脚变为输入引脚。这意味着 TRISC<1:0> 的值被忽略并且这些引脚将读为 0。

图 15-1： TIMER3 框图



注 1：当使能位 T1OSCEN 清零时，将关闭振荡器反相器和反馈电阻以降低功耗。

图 15-2： TIMER3 框图 （16 位读 / 写模式）



注 1：当使能位 T1OSCEN 清零时，将关闭振荡器反相器和反馈电阻以降低功耗。

© 2007 Microchip Technology Inc.

15.2 Timer3 16 位读 / 写模式

可将 Timer3 配置为 16 位读写模式（见图 15-2）。当 RD16 控制位（T3CON<7>）置 1 时，TMR3H 的地址被映射到 Timer3 的高字节缓冲寄存器。从 TMR3L 的读操作将把 Timer3 的高字节内容装入 Timer3 高字节缓冲寄存器。这种方式使用户可以精确地读取 Timer1 的全部 16 位，而无需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。

对 Timer3 的高字节进行写操作也必须通过 TMR3H 缓冲寄存器进行。在写入 TMR3L 的同时，使用 TMR3H 的内容更新 Timer3 的高字节。这样允许用户将所有的 16 位值一次写入 Timer3 的高字节和低字节。

在这一模式下不能直接读写 Timer3 的高字节。所有读写都必须通过 Timer3 高字节缓冲寄存器进行。

写入 TMR3H 不会清零 Timer3 预分频器。只有在写入 TMR3L 时才会清零该预分频器。

15.3 使用 Timer1 振荡器作为 Timer3 的时钟源

Timer1 内部振荡器可用作 Timer3 的时钟源。通过将 T1OSCEN（T1CON<3>）位置 1，可使能 Timer1 振荡器。要将它用作 Timer3 的时钟源还必须将 TMR3CS 位置 1。如前文所述，这样做也会将 Timer3 配置为在振荡器源的每个上升沿递增。

在**第 13.0 节 “Timer1 模块”**中对 Timer1 振荡器进行了描述。

15.4 Timer3 中断

TMR3 寄存器对（TMR3H:TMR3L）从 0000h 开始递增直到 FFFFh，然后溢出返回 0000h。如果允许 Timer3 中断，该中断就会在溢出时产生，由中断标志位 TMR3IF（PIR2<1>）表示。可以通过对 Timer3 中断允许位 TMR3IE（PIE2<1>）置 1 或清零来允许或禁止该中断。

15.5 使用 ECCPx 特殊事件触发信号来复位 Timer3

如果 ECCP1 或 ECCP2 被配置为使用 Timer3 并且在比较模式（CCPxM3:CCPxM0 = 1011）下产生特殊事件触发信号，该信号将复位 Timer3。如果使能了 A/D 模块，来自 ECCP2 的触发信号还将启动 A/D 转换（欲知更多信息，请参见**第 18.2.1 节 “特殊事件触发器”**）。

为了利用这一功能，必须将 Timer3 配置为定时器或同步计数器。在这种情况下，CCPRxH:CCPRxL 这对寄存器实际上变成了 Timer3 的周期寄存器。

如果 Timer3 在异步计数器模式下运行，复位操作可能不起作用。

如果对 Timer3 的写操作和 ECCPx 模块的特殊事件触发信号同时发生，则写操作优先。

注：	来自 ECCPx 模块的特殊事件触发信号不会将中断标志位 TMR3IF（PIR2<0>）置 1。

表 15-1： TIMER3 作为定时器 / 计数器时相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
TMR3L	Timer3 寄存器的低字节								59
TMR3H	Timer3 寄存器的高字节								59
T1CON(1)	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	56
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON	59

图注： - = 未实现位，读为 0。Timer3 模块不使用阴影单元

注 1： 当 WDTCON<4> = 0 时，此处的默认（旧）SFR 可用。

注：

 © 2007 Microchip Technology Inc.

16.0 TIMER4 模块

Timer4 定时器模块具有以下特征：

- 8 位定时器寄存器 （TMR4）
- 8 位周期寄存器 （PR4）
- 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 至 1:16）
- 当 TMR4 与 PR4 匹配时中断

Timer4 具有如寄存器 16-1 所示的控制寄存器。可以通过清零控制位 TMR4ON （T4CON<2>）关闭 Timer4，以使功耗最低。此寄存器还控制对 Timer4 的预分频值和后分频值的选择。图 16-1 所示为 Timer4 模块的简化框图。

16.1 Timer4 工作原理

Timer4 可以作为 ECCPx/CCPx 模块在 PWM 模式下的 PWM 时基。 TMR4 寄存器是可读写的，任何方式的器件复位都会使之清零。输入时钟（Fosc/4）有三种预分频比选项，分别是 1:1、 1:4 或 1:16，可通过控制位 T4CKPS1:T4CKPS0（T4CON<1:0>）选择。TMR4 的匹配输出通过 4 位后分频器 （分频比在 1:1 到 1:16 之间）产生 TMR4 中断，由标志位 TMR4IF （PIR3<3>）表示。

当发生以下情况中的任何一种时，预分频器和后分频计数器均会清零：

- 对 TMR4 寄存器进行写操作
- 对 T4CON 寄存器进行写操作
- 任何方式的器件复位 （上电复位、 $\overline{\text{MCLR}}$ 复位、看门狗定时器复位或欠压复位）

写 T4CON 时 TMR4 不会清零。

寄存器 16-1： T4CON：TIMER4 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
—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **未实现：**读为 0

bit 6-3 **T4OUTPS3:T4OUTPS0：**Timer4 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 **TMR4ON：**Timer4 使能位
1 = 打开 Timer4
0 = 关闭 Timer4

bit 1-0 **T4CKPS1:T4CKPS0：**Timer4 时钟预分频比选择位
00 = 预分频比为 1
01 = 预分频比为 4
1x = 预分频比为 16

16.2 Timer4 中断

Timer4 模块具有一个 8 位周期寄存器 PR4，该寄存器是可读写的。Timer4 从 00h 开始递增，直到与 PR4 匹配为止，然后在下一个计数周期复位为 00h。复位时，PR4 寄存器初始化为 FFh。

16.3 TMR4 的输出

TMR4 的输出（在通过预分频器之前）只能用作 ECCPx/CCPx 模块的 PWM 时基。它无法像 Timer2 输出那样用作 MSSP 模块的波特率时钟。

图 16-1： TIMER4 框图

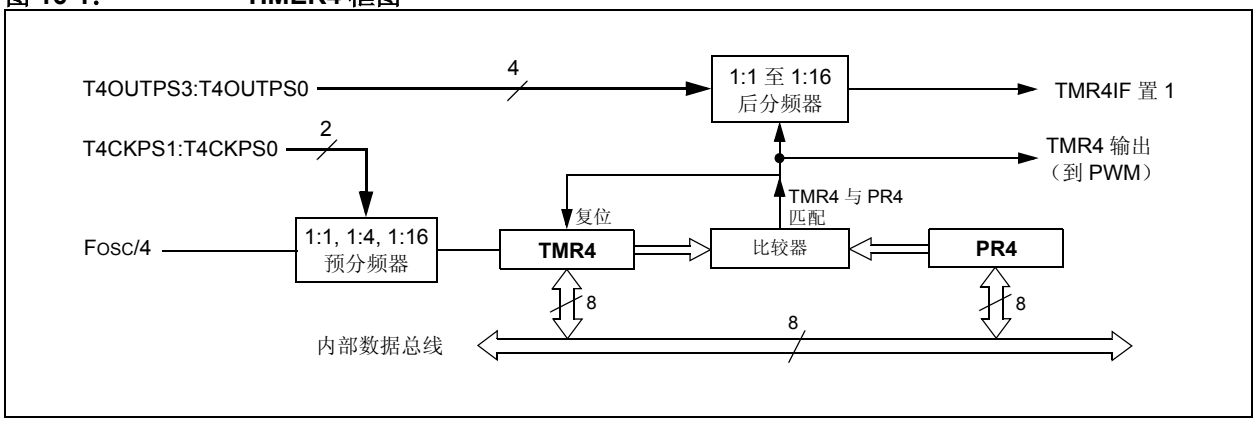


表 16-1： TIMER4 作为定时器 / 计数器时相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
TMR4	Timer4 寄存器								59
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	59
PR4	Timer4 周期寄存器								59

图注： - = 未实现位，读为 0。 Timer4 模块不使用阴影单元。

© 2007 Microchip Technology Inc.

17.0 捕捉 / 比较 /PWM （CCP）模块

PIC18F87J11 系列器件的产品都有五个 CCP（捕捉 / 比较 /PWM）模块。其中两个模块 （CCP4 和 CCP5）实现了标准捕捉、比较和脉宽调制（Pulse-Width Modulation，PWM）模式，本节将讨论这两个模块。其他三个模块（ECCP1、ECCP2 和 ECCP3）实现了标准捕捉和比较模式，以及增强型 PWM 模式。**第 18.0 节“增强型捕捉 / 比较 /PWM（ECCP）模块”**将讨论这三个模块。

每个 CCP/ECCP 模块都包含一个 16 位寄存器，它可用作 16 位捕获寄存器、 16 位比较寄存器或 PWM 主 / 从占空比寄存器。为了明了起见，以下章节中所描述的所有 CCP 模块操作都是针对 CCP4 的，但同样也适用于 CCP5。

本节所描述的捕捉和比较操作适用于所有标准和增强型 CCP 模块。**第 17.4 节“PWM 模式”**中所描述的 PWM 模式下的操作仅适用于 CCP4 和 CCP5。

注：	在本节和**第 18.0 节“增强型捕捉 / 比较 /PWM（ECCP）模块”**中，在提到与特定 CCP 模块相关的寄存器和位名称时，一般会使用 “x” 或 “y” 代替特定的模块编号。 因此，“CCPxCON”可能指 ECCP1、ECCP2、ECCP3、 CCP4 或 CCP5 的控制寄存器。

寄存器 17-1： CCPxCON：CCPx 控制寄存器 （CCP4 模块或 CCP5 模块）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现**：读为 0

bit 5-4 **CCPx<X:Y>**：CCPx 模块的 PWM 占空比 bit 1 和 bit 0

捕捉模式：未用。

比较模式：未用。

PWM 模式：这两位是 10 位 PWM 占空比的两个最低有效位 （bit 1 和 bit 0）。占空比的 8 个最高有效位在 CCPRxL 中。

bit 3-0 **CCPxM3:CCPxM0**：CCPx 模块模式选择位

0000 = 禁止捕捉 / 比较 /PWM （复位 CCPx 模块）
0001 = 保留
0010 = 比较模式，匹配时输出电平翻转 （CCPxIF 位置 1）
0011 = 保留
0100 = 捕捉模式：每个下降沿捕捉
0101 = 捕捉模式：每个上升沿捕捉
0110 = 捕捉模式：每 4 个上升沿捕捉
0111 = 捕捉模式：每 16 个上升沿捕捉
1000 = 比较模式：初始化 CCPx 引脚为低电平；比较匹配时，强制 CCPx 引脚为高电平（CCPxIF 位置 1）
1001 = 比较模式：初始化 CCPx 引脚为高电平；比较匹配时，强制 CCPx 引脚为低电平（CCPxIF 位置 1）
1010 = 比较模式：比较匹配时产生软件中断 （CCPxIF 位置 1， CCPx 引脚反映 I/O 状态）
1011 = 比较模式：CCPx 匹配时触发特殊事件、复位定时器并启动 A/D 转换 （CCPxIF 位置 1）
11xx = PWM 模式

17.1 CCP 模块配置

每个捕捉 / 比较 /PWM 模块均与一个控制寄存器（通常为 CCPxCON）和一个数据寄存器（CCPRx）相对应。数据寄存器由两个 8 位寄存器顺序组成：CCPRxL（低字节）和 CCPRxH（高字节）。所有寄存器都可读写。

17.1.1 CCP 模块和定时器资源

ECCP/CCP 模块根据所选模式决定使用 Timer1、Timer2、Timer3 或 Timer4。Timer1 和 Timer3 适用于捕捉或比较模式下的模块，而 Timer2 和 Timer4 适用于 PWM 模式下的模块。

表 17-1： CCP 模式——定时器资源

CCP 模式	定时器资源
捕捉	Timer1 或 Timer3
比较	Timer1 或 Timer3
PWM	Timer2 或 Timer4

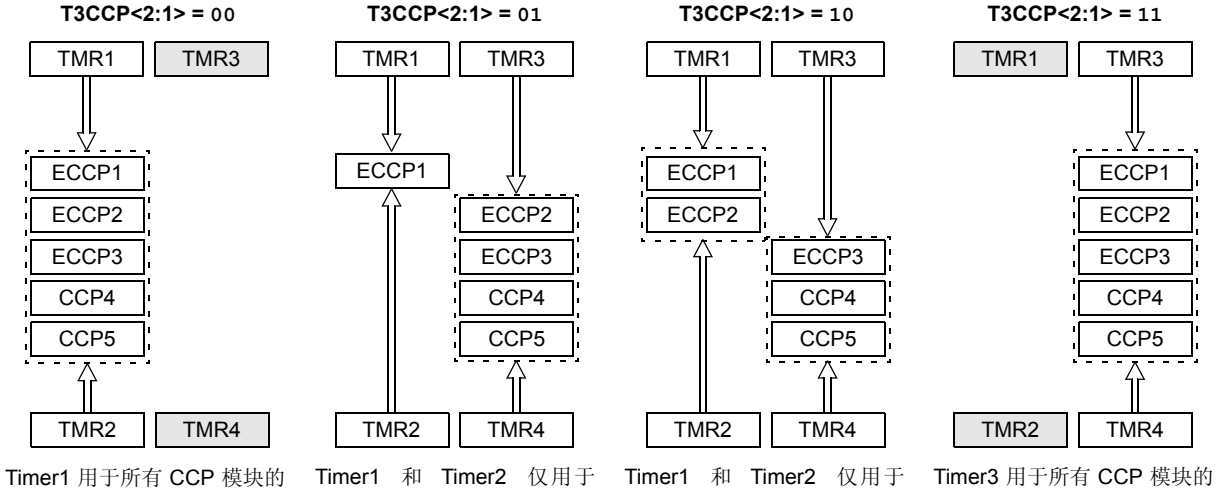
将哪个特定定时器分配给 CCP 模块是由 T3CON 寄存器（第 193 页的寄存器 15-1）中的 “CCP 的时钟源”使能位决定。根据所选配置，一次最多可激活四个定时器，具有相同配置的模块 （捕捉 / 比较或 PWM）共享定时器资源。图 17-1 所示为可能的配置。

17.1.2 漏极开路输出选项

器件工作在输出模式 （如比较或 PWM 模式）时，可选择将 CCP 引脚驱动器配置为漏极开路输出。此功能允许通过外部上拉电阻将 CCP 引脚上的电平拉为高电平，并允许漏极开路输出与外部电路通信，而无需额外的电平转换器。更多信息，请参见**第 10.1.4 节 “漏极开路输出”**。

漏极开路输出选项由 ODCON1 寄存器中的相应位控制。通过置 1 适当的位来配置相应模块的引脚以执行漏极开路操作。ODCON1 存储器与 TMR1H 共用同一地址空间。可通过置 1 WDTCON 寄存器 （WDTCON<4>）中的 ADSHR 位来访问 ODCON1 寄存器。

图 17-1： ECCPx/CCPx 和定时器互连配置



Timer1 用于所有 CCP 模块的所有捕捉和比较操作。Timer2 用于所有 CCP 模块的 PWM 操作。模块可共用这两个定时器资源中的任何一个作为公共时基。

Timer3 和 Timer4 不可用。

Timer1 和 Timer2 仅用于 ECCP1 的捕捉和比较或 PWM 操作 （取决于选定的模式）。

所有其他模块使用 Timer3 或 Timer4。如果模块处于捕捉 / 比较或 PWM 模式，则模块可共用这两个定时器资源之一作为公共时基。

Timer1 和 Timer2 仅用于 ECCP1 和 ECCP2 的捕捉和比较或 PWM 操作 （取决于为每个模块选定的模式）。如果这两个模块均处于捕捉 / 比较或 PWM 模式，则它们可共用一个定时器作为公共时基。

其他模块使用 Timer3 或 Timer4。如果模块处于捕捉 / 比较或 PWM 模式，则模块可共用这两个定时器资源之一作为公共时基。

Timer3 用于所有 CCP 模块的所有捕捉和比较操作。Timer4 用于所有 CCP 模块的 PWM 操作。模块可共用这两个定时器资源中的任何一个作为公共时基。

Timer1 和 Timer2 不可用。

© 2007 Microchip Technology Inc.

17.2 捕捉模式

在捕捉模式下，当相应的 CCP 引脚发生捕捉事件时，CCPRxH:CCPRxL 这对寄存器即捕捉 TMR1 或 TMR3 寄存器的 16 位值。事件定义如下：

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

事件由模式选择位 CCPxM3:CCPxM0（CCPxCON<3:0>）选择。当发生捕捉时，中断请求标志位 CCPxIF 置 1；该位必须用软件清零。如果在寄存器 CCPRx 中的值被读取之前发生了另一个捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

17.2.1 CCP 引脚配置

在捕捉模式下，通过置 1 相应的 TRIS 方向位可将 CCP 引脚配置为输入。

注：	如果 RG4/CCP5 被配置为输出，则对端口的写操作可产生捕捉条件。

17.2.2 TIMER1/TIMER3 模式选择

用于捕捉功能的定时器（Timer1 和 / 或 Timer3）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无法进行捕捉操作。用于每个 CCP 模块的定时器在 T3CON 寄存器中进行选择（见**第 17.1 节 “CCP 模块配置”**）。

17.2.3 软件中断

当捕捉模式改变时，可能会产生错误捕捉中断。用户应保持 CCPxIE 中断允许位清零，以避免错误中断。还应在操作模式发生任何改变之后清零中断标志位 CCPxIF。

17.2.4 CCP 预分频器

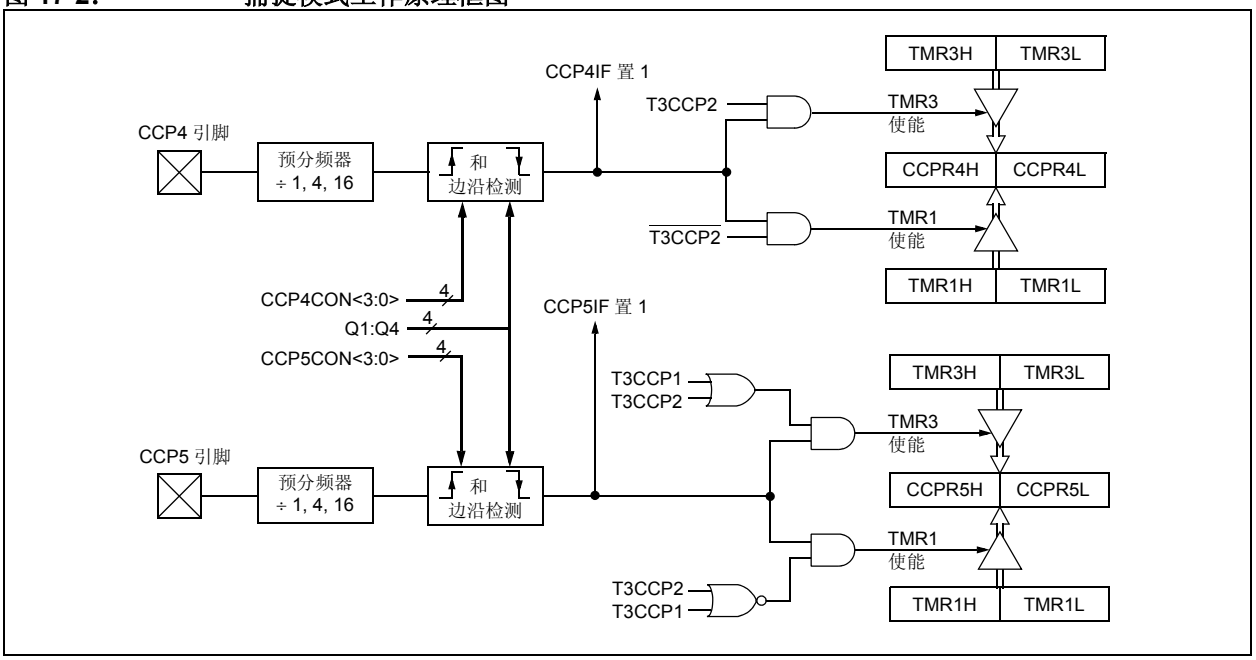
捕捉模式下有四种预分频器设置。它们作为操作模式的一部分由模式选择位（CCPxM3:CCPxM0）指定。每当关闭 CCP 模块或禁止捕捉模式时，预分频计数器就将被清零。这意味着任何复位都将清零预分频计数器。

在两个捕捉预分频器之间切换会产生中断。且不会清零预分频计数器；因此，切换后第一次捕捉可能来自一个非零预分频器。例 17-1 是切换捕捉预分频器时建议采用的方法。该例也清零预分频计数器且不会产生“假”中断。

例 17-1：　改变捕捉预分频比值（以 CCP5 为例）

```
CLRF   CCP5CON      ; Turn CCP module off
MOVLW  NEW_CAPT_PS  ; Load WREG with the
                    ; new prescaler mode
                    ; value and CCP ON
MOVWF  CCP5CON      ; Load CCP5CON with
                    ; this value
```

图 17-2：　捕捉模式工作原理框图



17.3 比较模式

在比较模式下，16位CCPRx寄存器的值将不断与TMR1或TMR3寄存器对的值相比较。当两者匹配时，CCP引脚可能会出现以下几种情况：

- 输出高电平
- 输出低电平
- 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 （CCPxM3:CCPxM0）的值。同时，中断标志位 CCPxIF 置 1。

17.3.1 CCP 引脚配置

用户必须通过清零相应的 TRIS 位，将 CCP 引脚配置为输出。

注：	清零CCP5CON寄存器会将RG4比较输出锁存器 （取决于器件配置）强制为默认的低电平。这不是 PORTB 或 PORTC I/O 数据锁存器。

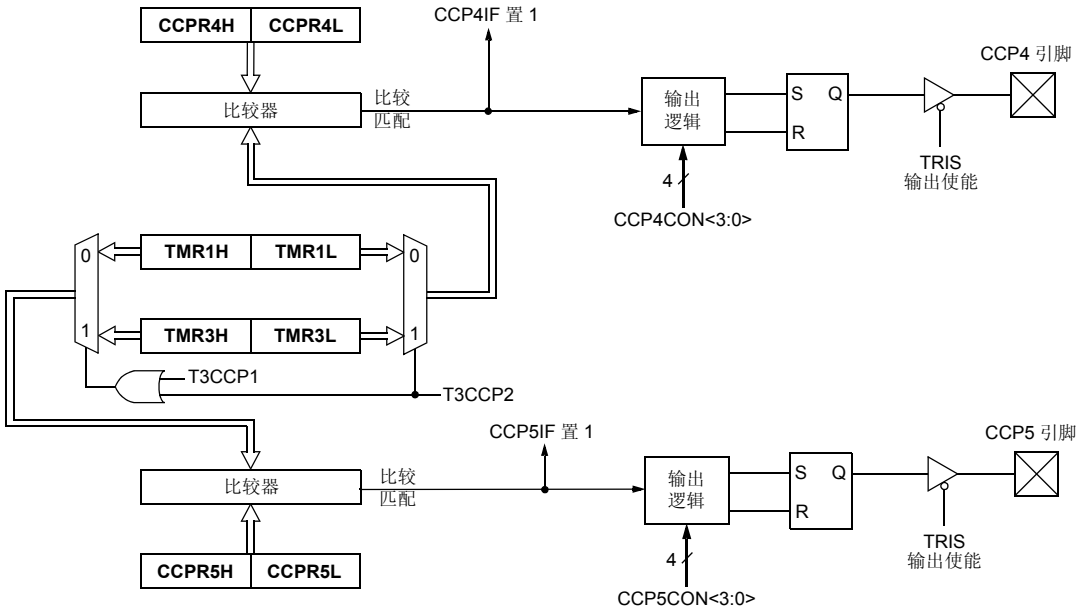
17.3.2 TIMER1/TIMER3 模式选择

如果CCP模块正在使用比较功能，Timer1和/或Timer3必须运行在定时器模式或同步计数器模式。在异步计数器模式下，无法进行比较操作。

17.3.3 软件中断模式

选定产生软件中断模式 （CCPxM3:CCPxM0 = 1010）时，相应的 CCP 引脚不受影响。只有在将 CCPxIE 位置 1 允许中断时，才会产生 CCP 中断。

图 17-3： 比较模式工作原理框图



© 2007 Microchip Technology Inc.

表 17-2： 与捕捉、比较、 TIMER1 和 TIMER3 相关的寄存器

名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	56
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	58
TMR1L(1)	Timer1 寄存器的低字节								56
TMR1H(1)	Timer1 寄存器的高字节								56
ODCON1(2)	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	56
T1CON(1)	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	56
TMR3H	Timer3 寄存器的高字节								59
TMR3L	Timer3 寄存器的低字节								59
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	59
CCPR4L	捕捉 / 比较 /PWM 寄存器 4 的低字节								59
CCPR4H	捕捉 / 比较 /PWM 寄存器 4 的高字节								59
CCPR5L	捕捉 / 比较 /PWM 寄存器 5 的低字节								59
CCPR5H	捕捉 / 比较 /PWM 寄存器 5 的高字节								59
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	59
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	59

图注： — = 未实现，读为 0。捕捉 / 比较、 Timer1 或 imer3 不使用阴影单元。

注 1： 当 WDTCON<4> = 0 时，此地址处的默认 （传统） SFR 可用。

2： 配置 SFR 的地址与默认 SFR 重叠；仅当 WDTCON<4> = 1 时才可访问配置 SFR。

17.4 PWM 模式

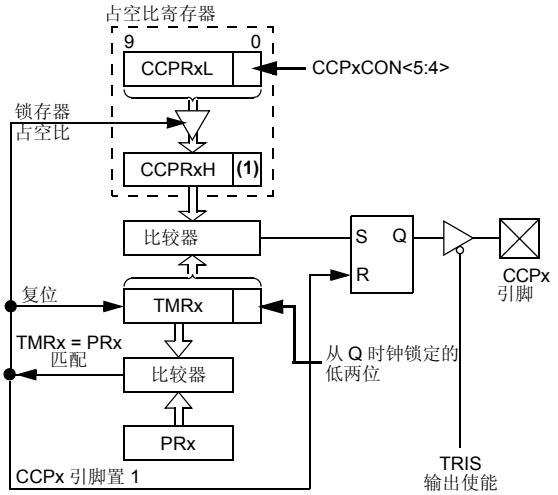
在脉宽调制（PWM）模式下，CCP 引脚可产生分辨率高达 10 位的 PWM 输出。由于 CCP4 和 CCP5 引脚与 PORTG 数据锁存器复用，因此必须清零相应的 TRISG 位才能使 CCP4 或 CCP5 引脚作为输出引脚。

注：	清零 CCP4CON 或 CCP5CON 寄存器会将 RG3 或 RG4 输出锁存器（取决于器件配置）强制为默认的低电平。这不是 PORTG I/O 数据锁存器。

图 17-4 所示为 CCP 模块在 PWM 模式时的简化框图。

如需了解设置 CCP 模块以进行 PWM 操作的详细步骤，请参见**第 17.4.3 节 “PWM 操作的设置”**。

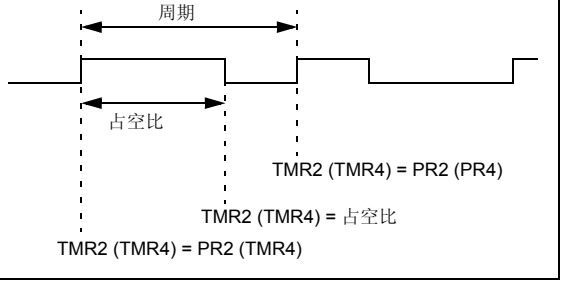
图 17-4： PWM 简化框图



注 1： 占空比寄存器的低两位位于此模块硬件中的第 2 位锁存器中。它在物理上独立于 CCPRx 寄存器。

PWM 输出信号（图 17-5）包含一个时基（周期）和一段输出高电平的时间（占空比）。PWM 信号的频率是周期的倒数（1/ 周期）。

图 17-5： PWM 输出



17.4.1 PWM 周期

PWM 周期可通过写 PR2（PR4）寄存器来指定。使用公式 17-1 计算 PWM 周期：

公式 17-1：

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

PWM 频率定义为 1/[PWM 周期]。

当 TMR2（TMR4）等于 PR2（PR4）时，在下一个计数周期中会发生以下三个事件：

- TMR2（TMR4）被清零
- CCP 引脚被置 1（例外情况：如果 PWM 占空比 = 0%， CCP 引脚将不被置 1）
- PWM 占空比从 CCPRxL 锁存到 CCPRxH

注：	确定 PWM 频率时不使用 Timer2 和 Timer4 后分频器（见**第 14.0 节 “Timer2 模块”**和**第 16.0 节“Timer4 模块”**）。使用后分频器时，其伺服更新速率的频率可与 PWM 输出频率不同。

17.4.2 PWM 占空比

PWM 占空比可通过写 CCPRxL 寄存器和 CCPxCON<5:4> 位来指定。最高分辨率可达 10 位。CCPRxL 包含占空比的高 8 位，而 CCPxCON<5:4> 包含其低 2 位。由 CCPRxL:CCPxCON<5:4> 表示这个 10 位值。公式 17-2 用于计算 PWM 占空比。

公式 17-2：

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

可在任意时刻写入 CCPRxL 和 CCPxCON<5:4>，但直至 PR2（PR4）与 TMR2（TMR4）发生匹配时（即周期结束）时，才会将占空比的值锁存到 CCPRxH 中。在 PWM 模式下， CCPRxH 为只读寄存器。

© 2007 Microchip Technology Inc.

CCPRxH 寄存器和一个 2 位的内部锁存器用于为 PWM 占空比提供双重缓冲。这种双重缓冲结构极其重要，可以避免 PWM 切换占空比的时候产生毛刺。

当 CCPRxH 和 2 位锁存器的值与 TMR2（TMR4）（连有内部 Q 时钟或 TMR2（TMR4）预分频器的 2 位）匹配时， CCP 引脚被清零。

对于给定的 PWM 频率，其最大 PWM 分辨率 （位）可以由公式 17-3 计算：

公式 17-3：

$$\text{PWM 分辨率（最大）} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ 位}$$

注：	如果 PWM 占空比的值大于 PWM 周期，CCP 引脚将不会被清零。

17.4.3 PWM 操作的设置

在为 CCP 模块配置 PWM 工作模式时应该遵循以下步骤：

1. PWM 周期可通过写 PR2（PR4）寄存器来指定。
2. PWM 占空比可通过写 CCPRxL 寄存器和 CCPxCON<5:4> 位来指定。
3. 通过清零相应的 TRIS 位将 CCP 引脚配置为输出引脚。
4. 通过写 T2CON （T4CON）来设置 TMR2 （TMR4）预分频值并使能 Timer2 （Timer4）。
5. 配置 CCPx 模块使之工作于 PWM 模式。

表 17-3： 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 或 16）	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最高分辨率 （位）	10	10	10	8	7	6.58

表 17-4： 与 PWM、 TIMER2 和 TIMER4 相关的寄存器

名称	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	56
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	58
TMR2(1)	Timer2 寄存器								56
PR2(1)	Timer2 周期寄存器								56
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	56
TMR4	Timer4 寄存器								59
PR4	Timer4 周期寄存器								59
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	59
CCPR4L	捕捉 / 比较 /PWM 寄存器 4 的低字节								59
CCPR4H	捕捉 / 比较 /PWM 寄存器 4 的高字节								59
CCPR5L	捕捉 / 比较 /PWM 寄存器 5 的低字节								59
CCPR5H	捕捉 / 比较 /PWM 寄存器 5 的高字节								59
CCP4CON	—	—	DC4B1	DC4B0	CCP4M3	CCP4M2	CCP4M1	CCP4M0	59
CCP5CON	—	—	DC5B1	DC5B0	CCP5M3	CCP5M2	CCP5M1	CCP5M0	59
ODCON1(2)	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	56

图注： — = 未实现，读为 0。 PWM、 Timer2 或 imer4 不使用阴影单元。

注 1： WDTCON<4> = 0 时，此地址处的默认 （传统） SFR 可用。

2： 配置 SFR 的地址与默认 SFR 重叠；仅当 WDTCON<4> = 1 时才可访问配置 SFR。

© 2007 Microchip Technology Inc.

18.0 增强型捕捉 / 比较 /PWM （ECCP）模块

PIC18F87J11 系列器件实现了 3 个带有增强型 PWM 功能的标准 CCP 模块。这些增强的功能包括 2 个或 4 个输出通道、用户可选的极性、死区控制以及自动关闭和重启。**第 18.4 节 “增强型 PWM 模式”**中有对增强功能的详细讨论。ECCP模块的捕捉、比较和单输出PWM功能与标准 CCP 模块中描述的相同。

增强型 CCP 模块的控制寄存器如寄存器 18-1 所示。该寄存器与 CCP4CON/CCP5CON 寄存器的不同之处在于它使用了两个最高有效位来控制 PWM 功能。

除了通过增强型CCPxCON寄存器可获得扩展的模式范围以外，ECCP 模块还有两个与增强型 PWM 操作和自动关闭功能相关的寄存器。它们是：

- ECCPxDEL （ECCPx PWM 延时寄存器）
- ECCPxAS （ECCPx 自动关闭控制寄存器）

寄存器 18-1： CCPxCON：ECCPx 控制寄存器 （ECCP1/ECCP2/ECCP3）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **PxM1:PxM0**：增强型 PWM 输出配置位

如果 CCPxM3:CCPxM2 = 00、 01 或 10：

xx = PxA 被指定为捕捉 / 比较输入 / 输出； PxB、 PxC 和 PxD 被指定为端口引脚

如果 CCPxM3:CCPxM2 = 11：

00 = 单输出：PxA 调制输出； PxB、 PxC 和 PxD 被指定为端口引脚
01 = 全桥正向输出：P1D 调制输出； P1A 有效； P1B 和 P1C 无效
10 = 半桥输出：P1A 和 P1B 为带死区控制的调制输出； P1C 和 P1D 被指定为端口引脚
11 = 全桥反向输出：P1B 调制输出； P1C 有效； P1A 和 P1D 无效

bit 5-4 **DCxB1:DCxB0**：PWM 占空比 bit 1 和 bit 0

捕捉模式：

未用。

比较模式：

未用。

PWM 模式：

这些位是 10 位 PWM 占空比的两个 LSb。占空比的 8 个 MSb 在 CCPRxL 中。

bit 3-0 **CCPxM3:CCPxM0**：增强型 CCPx 模块模式选择位

0000 = 捕捉 / 比较 /PWM 关闭 （复位 ECCPx 模块）
0001= 保留
0010 = 比较模式，匹配时交替输出
0011 = 捕捉模式
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：ECCPx 引脚初始化为低电平；比较匹配时输出高电平 （CCPxIF 置 1）
1001 = 比较模式：ECCPx 引脚初始化为高电平；比较匹配时输出低电平 （CCPxIF 置 1）
1010 = 比较模式：仅产生软件中断， ECCPx 引脚恢复到 I/O 状态
1011 = 比较模式：触发特殊事件 （ECCPx 复位 TMR1 或 TMR3，将 CCPxIF 位置 1，
如果使能了 A/D 模块， ECCPx 触发信号还会启动 A/D 转换）(1)
1100 = PWM 模式：PxA 和 PxC 为高电平有效； PxB 和 PxD 为高电平有效
1101 = PWM 模式：PxA 和 PxC 为高电平有效； PxB 和 PxD 为低电平有效
1110 = PWM 模式：PxA 和 PxC 为低电平有效； PxB 和 PxD 为高电平有效
1111 = PWM 模式：PxA 和 PxC 为低电平有效； PxB 和 PxD 为低电平有效

注 1： 只有 ECCP1 和 ECCP2 实现了该功能；其作用如同 ECCP3 中这些位设置为 1010 一样。

18.1 ECCP 输出和配置

每个增强型 CCP 模块可能至多有 4 路 PWM 输出，这取决于选定的工作模式。这些输出被指定为从 PxA 到 PxD，与多个 I/O 引脚复用。有些 ECCP 引脚的分配是固定的，而其他引脚的分配根据器件配置而改变。控制分配可变的引脚的位如下：

- CCP2MX 配置位
- ECCPMX 配置位 （仅 80 引脚器件）
- 程序存储器工作模式，由 EMB 配置位设置 （仅 80 引脚器件）

表 18-1、表 18-2 和表 18-3 汇总了增强型 CCP 模块的引脚分配。要将 I/O 引脚配置为 PWM 输出，必须通过分别设置 PxMx （CCPxCON<7:6>）和 CCPxMx 位（CCPxCON<3:0>），来选择适当的 PWM 模式。还必须将对应端口引脚的相应 TRIS 方向位设置为输出。

18.1.1 ECCP1/ECCP3 输出和程序存储器模式

在 80 引脚器件中，使用扩展单片机模式对工作在增强型 PWM 模式下的 ECCP1 和 ECCP3 有间接影响。默认情况下，PWM 输出 P1B/P1C 和 P3B/P3C 与 PORTE 引脚以及外部存储器总线的高字节复用。扩展单片机模式下，外部存储器总线上有活动时，会忽略增强型 CCP 输出并使它们不可用。因此，当器件处于扩展单片机模式并使用默认引脚配置时，ECCP1 和 ECCP3 只能用于兼容 （单输出） PWM 模式。

这种配置的例外情况是当为外部总线选定了 12 位地址宽度时 （EMB1:EMB0 配置位 = 01）。在这种情况下，即使外部总线上有活动， PORTE 的高位引脚仍然继续作为数字 I/O 引脚。P1B/P1C 和 P3B/P3C 仍可被用作增强型 PWM 输出。

如果在增强型单片机运行期间需要使用更多的 PWM 输出，P1B/P1C 和 P3B/P3C 输出可被重新分配给 PORTH 的高位。可以通过清零 ECCPMX 配置位来实现上述配置。

18.1.2 ECCP2 输出和程序存储器模式

对于 80 引脚器件，器件的程序存储器模式（**第 5.1.3 节 “PIC18F8xJ11/8XJ16 程序存储器模式”** ）还会影响该模块的引脚复用。

ECCP2 输入 / 输出（ECCP2/P2A）可以与 3 个引脚中的一个复用。所有器件的默认分配 （CCP2MX 配置位 1）是 RC1。清零 CCP2MX 会将 ECCP2/P2A 重新分配给 RE7。

80 引脚器件还有一个选择。当这些器件在单片机模式下运行时，上述复用选项仍然适用。在扩展单片机模式下，清零 CCP2MX 会将 ECCP2/P2A 重新分配给 RB3。

改变 ECCP2 的引脚分配不会自动改变有关配置端口引脚的要求。用户必须总是确认已为 ECCP2 操作正确地配置了相应的 TRIS 寄存器。

18.1.3 ECCP1 和 ECCP3 对 CCP4 和 CCP5 的使用

只有 ECCP2 模块有 4 个可用的专用输出引脚。假定这些引脚不需要被用作 I/O 端口或其他复用功能，它们随时可以使用，而无需任何其他 CCP 模块的介入。

ECCP1 和 ECCP3 只有 3 个专用的输出引脚：ECCPx/PxA、PxB 和 PxC。只要这些模块被配置为四输出 PWM 模式时，通常用作 CCP4 或 CCP5 的引脚分别成为 ECCP3 和 ECCP1 的 PxD 输出引脚。 CCP4 和 CCP5 模块保持工作，但是其输出被覆盖。

18.1.4 ECCP 模块和定时器资源

与标准 CCP 模块相同， ECCP 模块可以使用 Timer1、Timer2、 Timer3 或 Timer4，具体选用哪个定时器取决于选定的模式。Timer1 和 Timer3 在捕捉或比较模式下可用于该模块，而 Timer2 和 Timer4 则在 PWM 模式下可用于该模块。**第 18.1.4 节 “ECCP 模块和定时器资源”** 中提供了更多有关定时器资源的详细信息。

18.1.5 漏极开路输出选项

当工作在比较 PWM 模式或标准 PWM 模式时，可选择配置 ECCP 引脚驱动器为漏极开路输出。该功能使得引脚电平值可通过外部上拉电阻拉高，也使得输出无需使用额外的电平转换器即可用于外部电路通信。欲知更多信息，请参见**第 10.1.4 节 “漏极开路输出”**。

漏极开路输出选项通过 ODCON1 寄存器中的位进行控制。将相应位置 1 可配置相应模块的对应引脚为漏极开路输出。ODCON1 寄存器与 TMR1H 共享地址空间。ODCON1 寄存器可通过将 WDTCON 寄存器中的 ADSHR 位 （WDTCON<4>）置 1 来进行访问。

© 2007 Microchip Technology Inc.

表 18-1： ECCP1 的引脚配置

ECCP 模式	CCP1CON 配置	RC2	RE6	RE5	RG4	RH7	RH6
所有 PIC18F6XJ1X 器件：							
兼容的 CCP	00xx 11xx	ECCP1	RE6	RE5	RG4/CCP5	N/A	N/A
双输出 PWM	10xx 11xx	P1A	P1B	RE5	RG4/CCP5	N/A	N/A
四输出 PWM(1)	x1xx 11xx	P1A	P1B	P1C	P1D	N/A	N/A
PIC18F8XJ1X 器件， ECCPMX = 0，单片机模式：							
兼容的 CCP	00xx 11xx	ECCP1	RE6/AD14	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
双输出 PWM	10xx 11xx	P1A	RE6/AD14	RE5/AD13	RG4/CCP5	P1B	RH6/AN14
四输出 PWM(1)	x1xx 11xx	P1A	RE6/AD14	RE5/AD13	P1D	P1B	P1C
PIC18F8XJ1X 器件， ECCPMX = 1，扩展单片机模式， 16 位或 20 位地址宽度：							
兼容的 CCP	00xx 11xx	ECCP1	RE6/AD14	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
PIC18F8XJ1X 器件， ECCPMX = 1，单片机模式或扩展单片机模式， 12 位地址宽度：							
兼容的 CCP	00xx 11xx	ECCP1	RE6/AD14	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
双输出 PWM	10xx 11xx	P1A	P1B	RE5/AD13	RG4/CCP5	RH7/AN15	RH6/AN14
四输出 PWM(1)	x1xx 11xx	P1A	P1B	P1C	P1D	RH7/AN15	RH6/AN14

图注： x = 任意值， N/A = 不可用。阴影单元表示在给定模式下 ECCP1 不使用的引脚分配。

注 1： 当 ECCP1 处于四输出 PWM 模式时， CCP5 的输出被 P1D 覆盖；其他情况下， CCP5 完全正常工作。

表 18-2： ECCP2 的引脚配置

ECCP 模式	CCP2CON 配置	RB3	RC1	RE7	RE2	RE1	RE0
所有器件， CCP2MX = 1，任一工作模式：							
兼容的 CCP	00xx 11xx	RB3/INT3	ECCP2	RE7	RE2	RE1	RE0
双输出 PWM	10xx 11xx	RB3/INT3	P2A	RE7	P2B	RE1	RE0
四输出 PWM	x1xx 11xx	RB3/INT3	P2A	RE7	P2B	P2C	P2D
所有器件， CCP2MX = 0，单片机模式：							
兼容的 CCP	00xx 11xx	RB3/INT3	RC1/T1OS1	ECCP2	RE2	RE1	RE0
双输出 PWM	10xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	RE1	RE0
四输出 PWM	x1xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	P2C	P2D
PIC18F8XJ1X 器件 , CCP2MX = 0，扩展单片机模式：							
兼容的 CCP	00xx 11xx	ECCP2	RC1/T1OS1	RE7/AD15	RE2/$\overline{CS}$	RE1/$\overline{WR}$	RE0/$\overline{RD}$
双输出 PWM	10xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	RE1/$\overline{WR}$	RE0/$\overline{RD}$
四输出 PWM	x1xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	P2C	P2D

图注： x = 任意值。阴影单元表示在给定模式下 ECCP2 不使用的引脚分配。

表 18-3： ECCP3 的引脚配置

ECCP 模式	CCP3CON 配置	RG0	RE4	RE3	RG3	RH5	RH4
PIC18F6XJ1X 器件：							
兼容的 CCP	00xx 11xx	ECCP3	RE4	RE3	RG3/CCP4	N/A	N/A
双输出 PWM	10xx 11xx	P3A	P3B	RE3	RG3/CCP4	N/A	N/A
四输出 PWM(1)	x1xx 11xx	P3A	P3B	P3C	P3D	N/A	N/A
PIC18F8XJ1X 器件， ECCPMX = 0，单片机模式：							
兼容的 CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RG3/CCP4	RH7/AN15	RH6/AN14
双输出 PWM	10xx 11xx	P3A	RE6/AD14	RE5/AD13	RG3/CCP4	P3B	RH6/AN14
四输出 PWM(1)	x1xx 11xx	P3A	RE6/AD14	RE5/AD13	P3D	P3B	P3C
PIC18F8XJ1X 器件， ECCPMX = 1，扩展单片机模式， 16 位或 20 位地址宽度：							
兼容的 CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RG3/CCP4	RH7/AN15	RH6/AN14
PIC18F8XJ1X 器件， ECCPMX = 1，单片机模式或扩展单片机模式， 12 位地址宽度：							
兼容的 CCP	00xx 11xx	ECCP3	RE4/AD12	RE3/AD11	RG3/CCP4	RH5/AN13	RH4/AN12
双输出 PWM	10xx 11xx	P3A	P3B	RE3/AD11	RG3/CCP4	RH5/AN13	RH4/AN12
四输出 PWM(1)	x1xx 11xx	P3A	P3B	P3C	P3D	RH5/AN13	RH4/AN12

图注： x = 任意值， N/A = 不可用。阴影单元表示在给定模式下 ECCP3 不使用的引脚分配。

注 1： 当 ECCP3 处于四输出 PWM 模式时， CCP4 的输出被 P1D 覆盖；其他情况下， CCP4 完全正常工作。

18.2 捕捉和比较模式

除了要在下面讨论的特殊事件触发器的操作，ECCP 模块的捕捉和比较模式在操作上与 CCP4 相同。**第 17.2 节 “捕捉模式”**和**第 17.3 节 “比较模式”**详细讨论了这些操作。

18.2.1 特殊事件触发器

ECCP1 和 ECCP2 模块具有一个内部硬件触发器，在比较模式下，当 CCPRx 寄存器对和选定的定时器发生匹配时会产生触发信号。触发信号可以用来启用某个操作。通过将 CCPxCON<3:0> 设置为 “1011”选择此模式。

ECCP1 或 ECCP2 的特殊事件触发输出会复位 TMR1 或 TMR3 寄存器对，这取决于当前选定的定时器资源。这将使 CCPRx 寄存器对被用作 Timer1 或 Timer3 的 16 位可编程周期寄存器。另外，如果 A/D 模块被使能，则 ECCP2 特殊事件触发信号还将启动 A/D 转换。

没有为 ECCP3、CCP4 或 CCP5 实现特殊事件触发器。为这些模块选择特殊事件触发模式与选择带软件中断的比较模式（CCPxM3:CCPxM0 = 1010）作用相同。

注：	来自 ECCP2 模块的特殊事件触发信号不会将 Timer1 或 Timer3 中断标志位置 1。

18.3 标准 PWM 模式

当配置为单输出模式时， ECCP 模块的工作方式与**第 17.4 节 “PWM 模式”**中描述的 PWM 模式下的标准 CCP 模块相同。如表 18-1 到表 18-3 所述，这种模式有时也被称为 “兼容的 CCP”模式。

注：	当设置单输出 PWM 操作时，用户可以自行决定使用**第 17.4.3 节 “PWM 操作的设置”**或**第 18.4.9 节 “设置 PWM 操作”**中说明的步骤。后者更加常用但是只适用于单输出或多输出 PWM。

© 2007 Microchip Technology Inc.

18.4 增强型 PWM 模式

增强型 PWM 模式提供了更多的 PWM 输出选项以适合更广泛的控制应用。该模块与标准 CCP 模块向后兼容，提供至多 4 路输出，分别指定为 PxA 到 PxD。用户还能够选择信号的极性（高电平有效或低电平有效）。通过分别设置 CCPxCON 寄存器的 PxM1:PxM0（CCPxCON<7:6>）和 CCPxM3:CCPxM0 位（CCPxCON<3:0>）可以配置模块的输出模式和极性。

为清晰起见，本节中增强型 PWM 模式的操作描述普遍与 ECCP1 和 TMR2 模块相关。控制寄存器的名称均根据 ECCP1 给出。 3 个增强型模块和 2 个定时器资源可以互换使用并且工作方式相同。通过在 T3CON 中选择适当的位可以为 PWM 操作选择 TMR2 或 TMR4。

图 18-1 所示为 PWM 操作的简化框图。所有的控制寄存器都是双重缓冲的，并且在一个新的 PWM 周期的开始时（Timer2 复位时的周期边界）被装载，以防止输出出现毛刺。但 ECCPx PWM 延时寄存器 ECCPxDEL 例外，该寄存器在占空比边界或周期边界（选择两者中首先出现的边界）被装载。由于缓冲，模块将不会立即启动，而要等到分配的定时器复位为止。这意味着增强型 PWM 波形并不完全与标准的 PWM 波形吻合，而是偏移一个完整的指令周期（4 TOSC）。

如前所述，用户必须手动为输出配置相应的 TRIS 位。

18.4.1 PWM 周期

PWM 周期可以通过写 PR2 寄存器来指定。可使用以下公式来计算 PWM 周期：

公式 18-1：

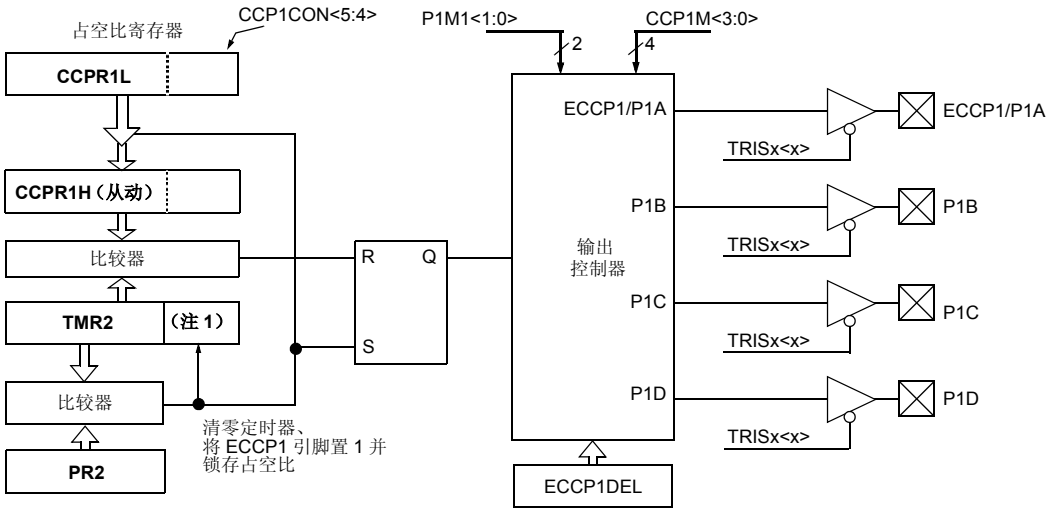
$$\text{PWM 周期} = [(\text{PR2}) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 预分频值})$$

PWM 频率定义为 1 / [PWM 周期]。当 TMR2 等于 PR2 时，在下一个计数周期中会发生以下三个事件：

- TMR2 被清零
- ECCP1 引脚被置 1（如果 PWM 占空比 = 0%，ECCP1 引脚将不会被置 1）
- PWM 占空比从 CCPR1L 复制到 CCPR1H

注： 确定 PWM 频率时不使用 Timer2 后分频器（见**第 14.0 节 “Timer2 模块”**）。使用后分频器可得到与PWM输出频率不同的伺服更新速率。

图 18-1： 增强型 PWM 模块的简化框图



注： 8 位 TMR2 寄存器结合一个 2 位内部 Q 时钟或预分频器的 2 位产生 10 位时基。

18.4.2 PWM 占空比

PWM 占空比可通过写入 CCPR1L 寄存器和 CCP1CON<5:4> 位来指定。最高分辨率可达 10 位。CCPR1L 包含 8 个 MSb，CCP1CON<5:4> 包含两个 LSb。由 CCPR1L:CCP1CON<5:4> 表示该 10 位值。PWM 占空比由以下公式计算。

公式 18-2：

PWM 占空比 = (CCPR1L:CCP1CON<5:4>) •
TOSC • (TMR2 预分频值)

可以在任何时候写入 CCPR1L 和 CCP1CON<5:4>，但直到 PR2 与 TMR2 中的值匹配时（即当周期结束时），占空比的值才被复制到 CCPR1H。在 PWM 模式中，CCPR1H 是只读寄存器。

CCPR1H 寄存器和一个 2 位的内部锁存器用于为 PWM 占空比提供双重缓冲。这种双重缓冲结构极其重要，可以避免 PWM 操作过程中产生毛刺。当 CCPR1H 和 2 位锁存值与 TMR2 结合内部 2 位 Q 时钟或 TMR2 预分频器的 2 位匹配时， ECCP1 引脚被清零。对于给定的 PWM 频率，其最大分辨率 （位）由以下公式计算：

公式 18-3：

$$\text{PWM 分辨率（最大）} = \frac{\log\left(\frac{FOSC}{FPWM}\right)}{\log(2)} \text{ 位}$$

注： 如果 PWM 占空比的值大于 PWM 周期，ECCP1 引脚将不会被清零。

18.4.3 PWM 输出配置

CCP1CON 寄存器中的 P1M1:P1M0 位可以实现以下 4 种配置：

- 单输出
- 半桥输出
- 全桥输出，正向模式
- 全桥输出，反向模式

单输出模式是在**第 18.4 节 “增强型 PWM 模式”**中讨论的标准 PWM 模式。在接下来的各节中将详细介绍半桥和全桥输出模式。

图 18-2 中汇总了各种配置下输出的大体关系。

表 18-4： 40MHz 时的 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 或 16）	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最高分辨率 （位）	10	10	10	8	7	6.58

© 2007 Microchip Technology Inc.

图 18-2：　　PWM 输出关系 （高电平有效状态）

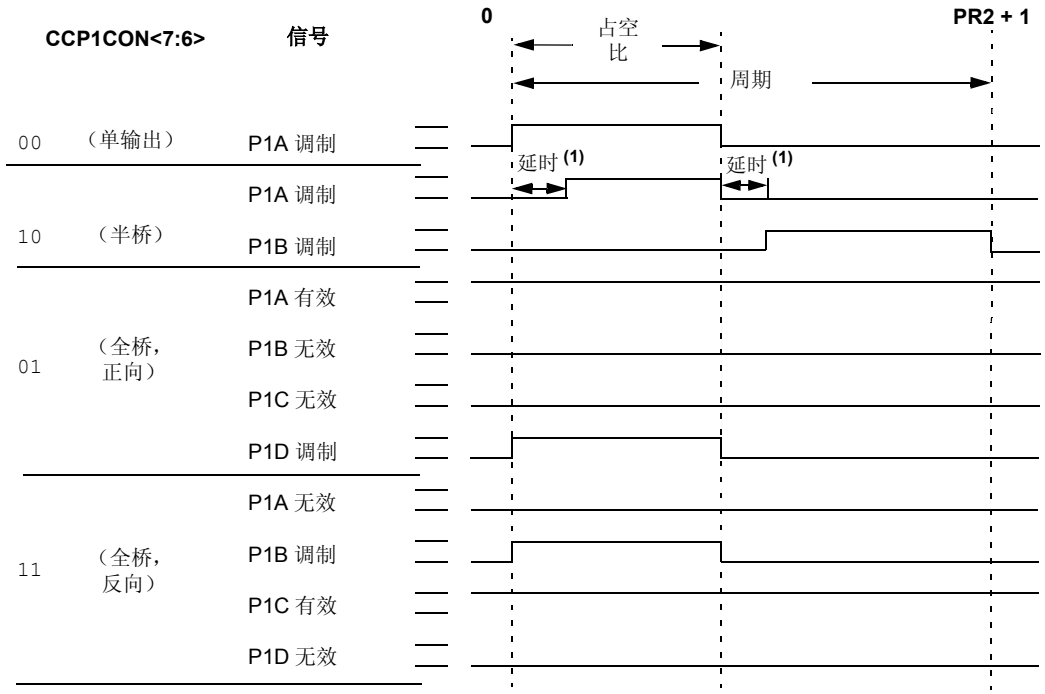
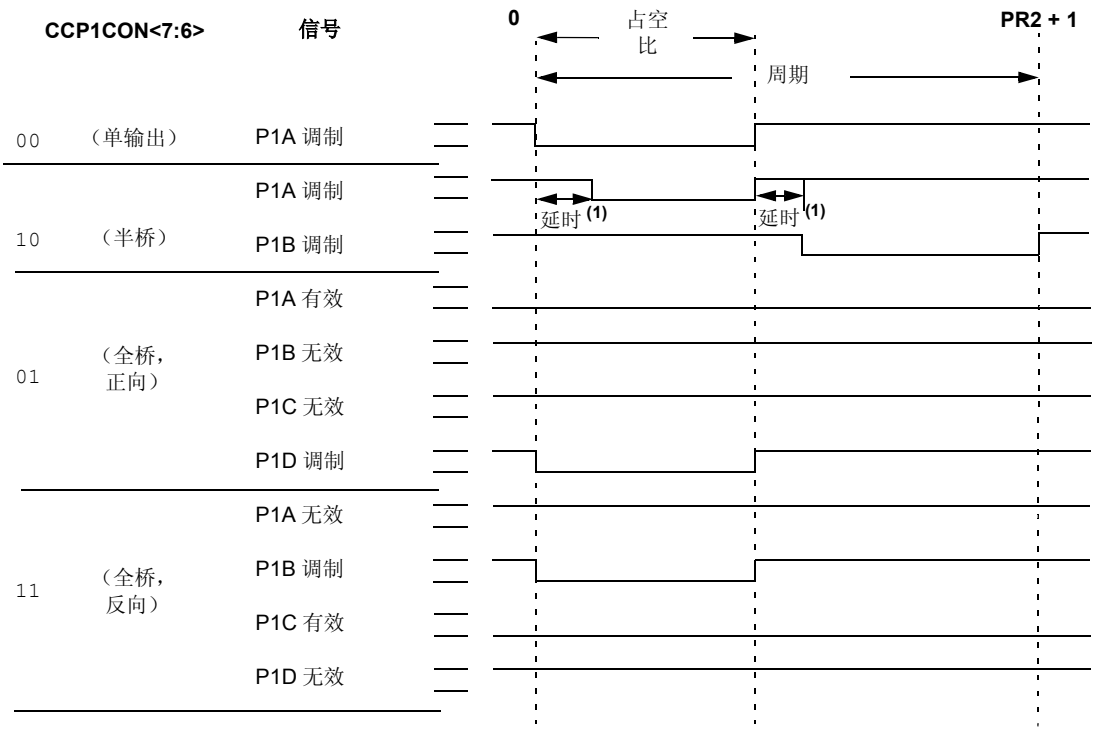


图 18-3：　　PWM 输出关系 （低电平有效状态）



关系：

- 周期 =4 * Tosc * (PR2 + 1) * (TMR2 预分频值)
- 占空比 = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 预分频值)
- 延时 = 4 * Tosc * (ECCP1DEL<6:0>)

注　　1： 使用 ECCP1DEL 寄存器设置死区延时 （见**第 18.4.6 节 “可编程死区延时”**）。

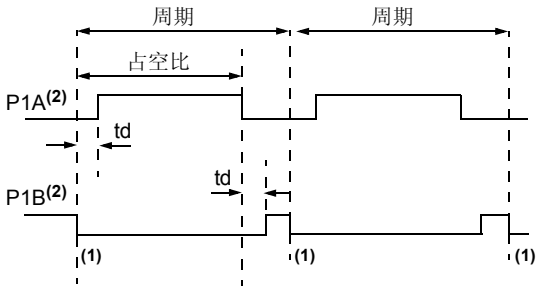
18.4.4 半桥模式

在半桥输出模式下，两个引脚用作输出端来驱动推挽式负载。PWM输出信号在P1A引脚上输出，而互补的PWM输出信号在 P1B 引脚上输出 （图 18-4）。如图 18-5 所示，此模式可用于半桥应用或那些使用 2 个 PWM 信号来调制 4 个功率开关的全桥应用。

在半桥输出模式下，可编程的死区延时可用来防止在半桥功率器件中产生贯通电流。P1DC6:P1DC0 位的值可用来设置输出被驱动为有效之前的指令周期数。如果该值大于占空比，在整个周期内对应的输出将保持无效。有关死区延时操作的更多详细信息，请参见**第 18.4.6 节“可编程死区延时”**。

由于 P1A 和 P1B 输出与 PORTC<2> 和 PORTE<6> 数据锁存器复用，必须清零 TRISC<2> 和 TRISE<6> 位以将 P1A 和 P1B 配置为输出。

图 18-4： 半桥 PWM 输出

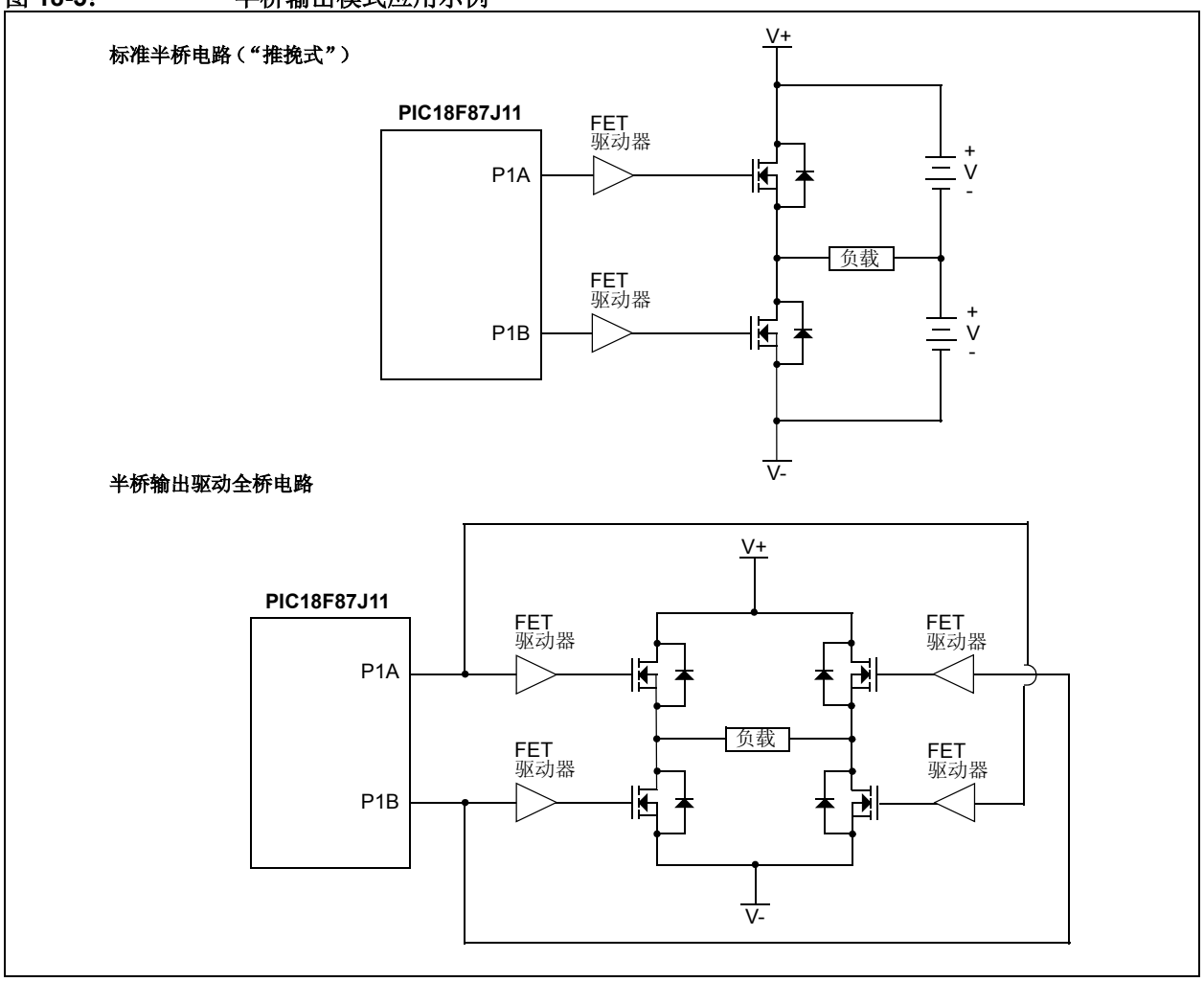


td = 死区延时

注 1： 此时， TMR2 寄存器的值与 PR2 寄存器的值相等。

2： 输出信号显示为高电平有效。

图 18-5： 半桥输出模式应用示例



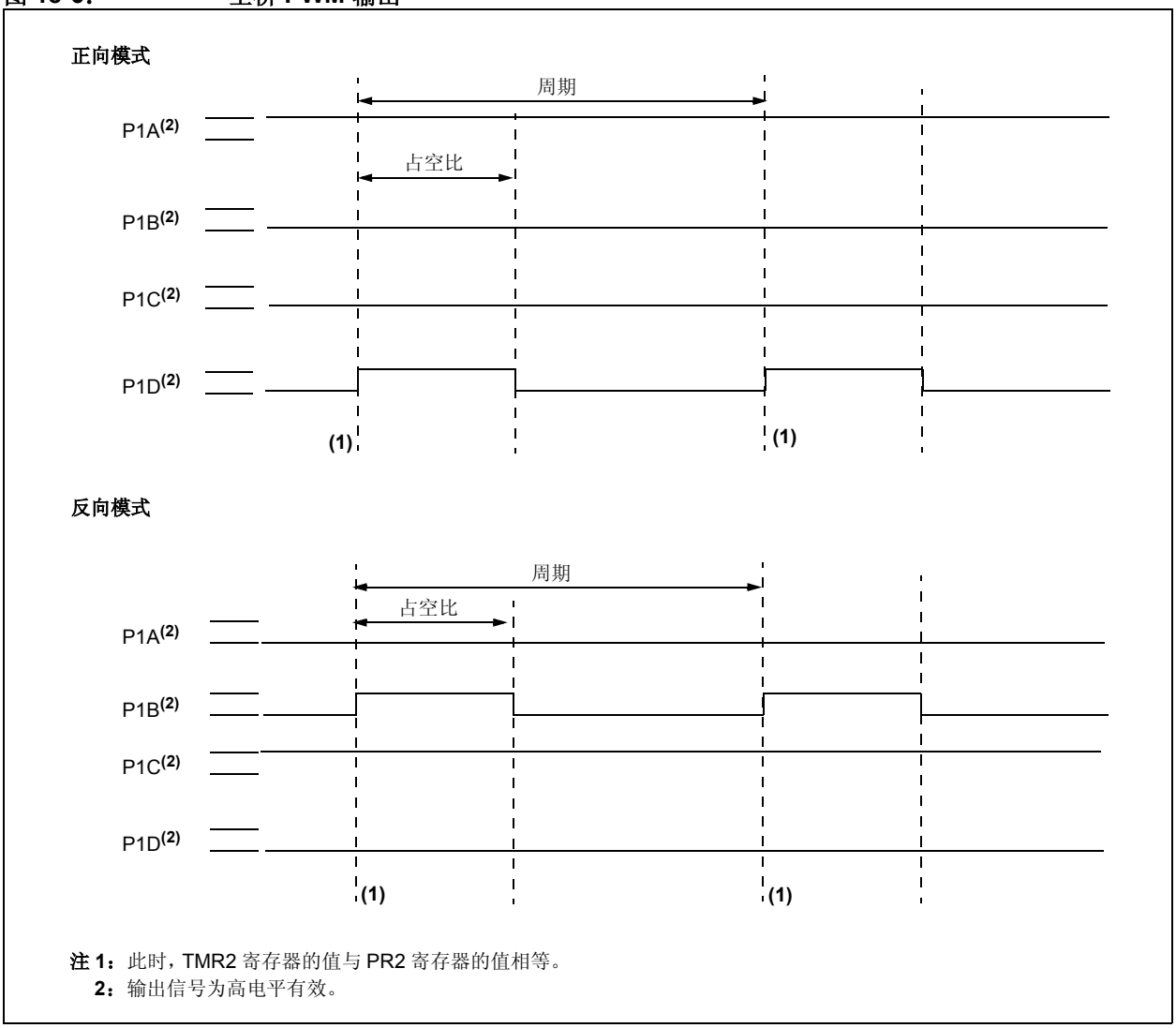
© 2007 Microchip Technology Inc.

18.4.5 全桥模式

在全桥输出模式下， 4 个引脚被用作输出；但是，一次只能有 2 个输出有效。在正向模式下，引脚 P1A 连续有效而引脚 P1D 为调制输出。在反向模式下，引脚 P1C 连续有效而引脚 P1B 为调制输出。这些在图 18-6 中进行了说明。

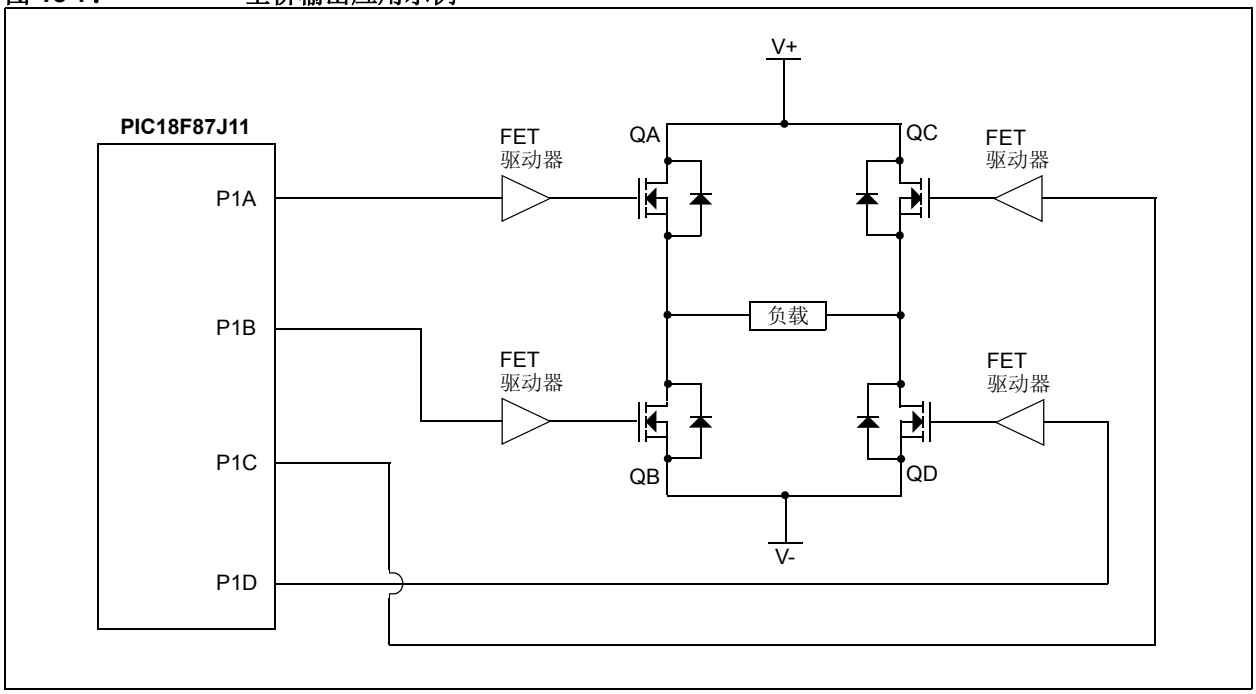
P1A、P1B、P1C 和 P1D 输出与表 18-1、表 18-2 和表 18-3 中描述的端口引脚复用。必须清零 TRIS 位以使 P1A、 P1B、 P1C 和 P1D 引脚成为输出引脚。

图 18-6： 全桥 PWM 输出



注 1： 此时，TMR2 寄存器的值与 PR2 寄存器的值相等。
2： 输出信号为高电平有效。

图 18-7：　　全桥输出应用示例



18.4.5.1　全桥输出模式下方向的更改

在全桥输出模式下，CCP1CON 寄存器中的 P1M1 位使用户能控制负载中电流为正向还是反向。当应用固件更改此方向控制位时，模块将在下一个 PWM 周期采用新的方向。

就在当前 PWM 周期结束之前，调制输出（P1B 和 P1D）被置于它们的无效状态，而未调制的输出（P1A 和 P1C）被切换为输出反向驱动电流。这发生在下一个 PWM 周期开始前的一段时间间隔内（4 TOSC *(Timer2预分频值)）。Timer2 预分频器的分频值将是 1、4 或 16，这取决于 T2CKPS 位（T2CON<1:0>）的值。从切换未调制输出到下一个周期开始之间的这段间隔内，调制输出（P1B 和 P1D）保持无效。此关系如图 18-8 所示。

注意，在全桥输出模式下， ECCP1 模块不提供任何死区延时。通常，由于在任何时间只调制一个输出，所以不需要死区延时。然而有一种情形可能需要死区延时。当以下两个条件都成立时，出现此情形：

1. 当输出信号的占空比接近或等于 100% 时，PWM 输出的方向发生改变。
2. 功率开关（包括功率器件和驱动器电路）的关闭时间大于打开时间。

图 18-9 所示为在占空比接近 100% 时，PWM 方向从正向更改为反向的示例。在时间 t1 时，输出 P1A 和 P1D 变为无效，而输出 P1C 变为有效。在此示例中，由于功率器件的关闭时间比打开时间长，贯通电流可能在时间段 “t” 内流过功率器件 QC 和 QD （见图 18-7）。如果 PWM 方向从反向更改为正向，功率器件 QA 和 QB 将出现相同的现象。

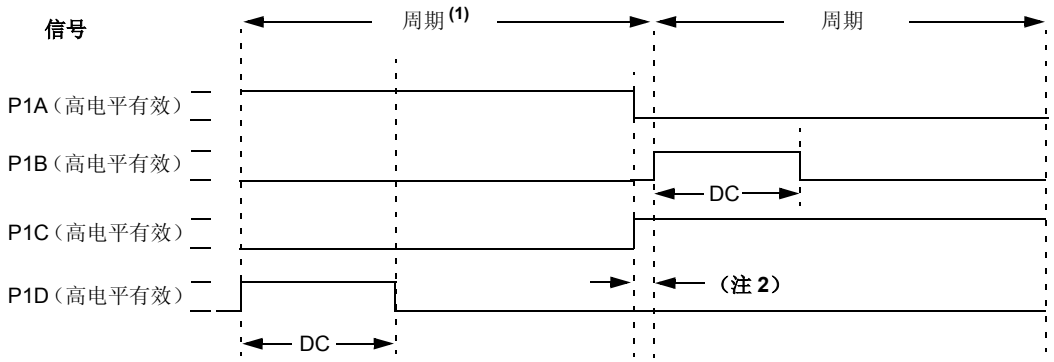
如果应用需要更改高占空比 PWM 的方向，必须满足以下要求之一：

1. 在更改方向前的那个 PWM 周期缩小 PWM 的占空比。
2. 使用可使开关元件的关断速度比导通速度更快的开关驱动器。

可能还存在防止贯通电流的其他方法。

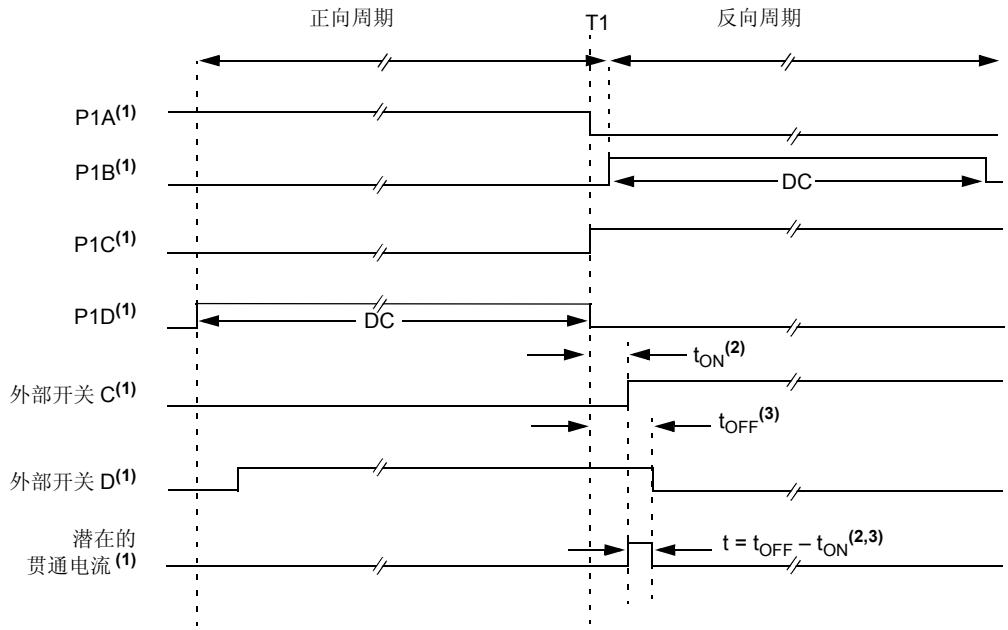
© 2007 Microchip Technology Inc.

图 18-8：　　PWM 方向更改



注　1： 可在 PWM 周期内的任何时间写 ECCP1 控制寄存器中的方向位 （CCP1CON<7>）。
2： 当更改方向时， P1A 和 P1C 上的信号在当前 PWM 周期末尾之前的间隔内进行切换，间隔可以是 4 TOSC、 16 TOSC 或 64 TOSC，这取决于 Timer2 的预分频值。在此期间被调制的 P1B 和 P1D 信号为无效。

图 18-9：　　当占空比接近 100% 时 PWM 更改方向



注　1： 所示的所有信号为高电平有效。
2： t_{ON} 是功率开关 QC 及其驱动器的导通延时。
3： t_{OFF} 是功率开关 QD 及其驱动器的关断延时。

18.4.6 可编程死区延时

在半桥应用中，始终以 PWM 频率的调制信号驱动功率开关，通常关断功率开关比导通它需要更长的时间。如果上方和下方的功率开关同时切换（一个导通，而另一个关断），两个开关可能会在一段短时间内都处于导通状态，直到一个开关完全关断为止。在这很短的间隔内，很高的电流（*贯通电流*）可能流过两个功率开关，从而使半桥供电电路短路。为了避免在切换期间流过这种潜在的破坏性贯通电流，一般延时打开其中的一个功率开关以使另一个开关完全关断。

在半桥输出模式下，可数字编程的死区延时可用来避免贯通电流破坏半桥功率开关。该延时在信号从无效状态转换到有效状态时发生（见图 18-4）。ECCPxDEL 寄存器（寄存器 18-2）的低 7 位根据单片机指令周期（TCY 或 4 TOSC）设置延时。

18.4.7 增强型 PWM 自动关闭

当 ECCP1 被编程为任何一种增强型 PWM 模式时，可以将有效输出引脚配置为自动关闭。当发生关闭事件时，自动关闭会立即将增强型 PWM 输出引脚置于定义的关闭状态。

关闭事件可以由两个比较器模块中的任意一个或FLT0引脚（或者以上 3 者的任意组合）触发。比较器可以用来监视与桥路中监视到的电流成比例的电压输入。如果电压超过门限值，比较器将切换状态并触发关闭。另外，FLT0引脚上的数字低电平信号也能触发关闭。通过不选择任何自动关闭源，可以禁止自动关闭功能。通过使用 ECCP1AS2:ECCP1AS0 位（ECCP1AS<6:4>）选择将使用的自动关闭源。

关闭发生后输出引脚被逐一置于它们的关闭状态，关闭状态由PSS1AC1:PSS1AC0和PSS1BD1:PSS1BD0位（ECCP1AS3:ECCP1AS0）指定。 每对引脚（P1A/P1C 和 P1B/P1D）可被设置为驱动高电平、驱动低电平或三态（不驱动）。还需将 ECCP1ASE 位（ECCP1AS<7>）置 1 以便将增强型 PWM 输出保持在它们的关闭状态。

当关闭事件发生时，ECCP1ASE 被硬件置 1。如果不使能自动重启，当关闭源清除后， ECCP1ASE 位将被固件清零。如果使能自动重启，当自动关闭源清除后，ECCP1ASE 位将被自动清零。

如果当 PWM 周期开始时 ECCP1ASE 位置 1，PWM 输出将在这整个 PWM 周期内保持在它们的关闭状态。当 ECCP1ASE 位清零时，PWM 输出将在下一个 PWM 周期的开头返回到正常操作。

注：	当关闭条件有效时，禁止写 ECCP1ASE 位。

寄存器 18-2： ECCPxDEL：ECCPx PWM 延时寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **PxRSEN**：PWM 重启使能位
1 = 自动关闭时，一旦关闭事件清除， ECCPxASE 位立即自动清零； PWM 自动重启
0 = 自动关闭时，必须用软件清零 ECCPxASE 位以重启 PWM

bit 6-0 **PxDC6:PxDC0**：PWM 延时计数位
延时时间，以 FOSC/4 （4 * TOSC）周期为单位，为 PWM 信号变为有效的预定时间和实际时间之差。

© 2007 Microchip Technology Inc.

寄存器 18-3： ECCPxAS：ECCPx 自动关闭控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **ECCPxASE**：ECCPx 自动关闭事件状态位
0 = ECCPx 正在输出
1 = 已发生了关闭事件； ECCPx 输出处于关闭状态

bit 6-4 **ECCPxAS2:ECCPxAS0**：ECCPx 自动关闭源选择位
000 = 禁止自动关闭
001 = 比较器 1 输出
010 = 比较器 2 输出
011 = 比较器 1 或 2
100 = FLT0
101 = FLT0 或比较器 1
110 = FLT0 或比较器 2
111 = FLT0、比较器 1 或比较器 2

bit 3-2 **PSSxAC1:PSSxAC0**：引脚 A 和 C 的关闭状态控制位
00 = 将引脚 A 和 C 驱动为 0
01 = 将引脚 A 和 C 驱动为 1
1x = 引脚 A 和 C 为三态

bit 1-0 **PSSxBD1:PSSxBD0**：引脚 B 和 D 的关闭状态控制位
00 = 将引脚 B 和 D 驱动为 0
01 = 将引脚 B 和 D 驱动为 1
1x = 引脚 B 和 D 为三态

18.4.7.1 自动关闭和自动重启

可以将自动关闭功能配置为在关闭事件后允许模块自动重启。通过将 ECCP1DEL 寄存器中的 P1RSEN 位（ECCP1DEL<7>）置 1 可使能此功能。

在 P1RSEN = 1（图 18-10）的关闭模式下，只要关闭源继续存在， ECC1PASE 位就将保持置 1。当关闭条件清除时，ECCP1ASE 位清零。若 P1RSEN = 0（图 18-11），一旦出现关闭条件，ECCP1ASE 位将保持置 1，直到它被固件清零为止。一旦 ECCP1ASE 被清零，增强型 PWM 将在下一个 PWM 周期的开头恢复正常操作。

注：	当关闭条件有效时，禁止写 ECCP1ASE 位。

如果自动关闭源是某个比较器，则关闭条件为输出的电平，而与 P1RSEN 位的设置无关。只要关闭源继续存在， ECCP1ASE 位就不能被清零。

通过将 1 写入 ECCP1ASE 位可强制进入自动关闭模式。

18.4.8 启动注意事项

当在 PWM 模式下使用 ECCP1 模块时，应用硬件必须在 PWM 输出引脚上使用适当的上拉和 / 或下拉电阻。当单片机从复位状态恢复时，所有的 I/O 引脚都处于高阻态。外部电路必须将功率开关器件保持在关闭（OFF）状态直到单片机用适当的信号电平驱动 I/O 引脚，或激活 PWM 输出为止。

CCP1M1:CCP1M0 位（CCP1CON<1:0>）允许用户为每对 PWM 输出引脚（P1A/P1C 和 P1B/P1D）选择 PWM 输出信号是高电平有效还是低电平有效。必须在 PWM 引脚被配置为输出之前选择 PWM 输出的极性。建议不要在 PWM 引脚被配置为输出时更改极性配置，因为这可能造成对应用电路的损害。

当初始化 PWM 模块时，P1A、P1B、P1C 和 P1D 的输出锁存器可能处于不正确的状态。使能 ECCP1 模块同时将 PWM 引脚使能为输出可能导致对应用电路的损害。必须将 ECCP1 模块使能为正确的输出模式并且在配置PWM引脚为输出之前完成一个完整的PWM周期。当第2 个 PWM 周期开始时 TMR2IF 位会置 1，从而表明经过了一个完整的 PWM 周期。

图 18-10： PWM 自动关闭（P1RSEN = 1，使能自动重启）

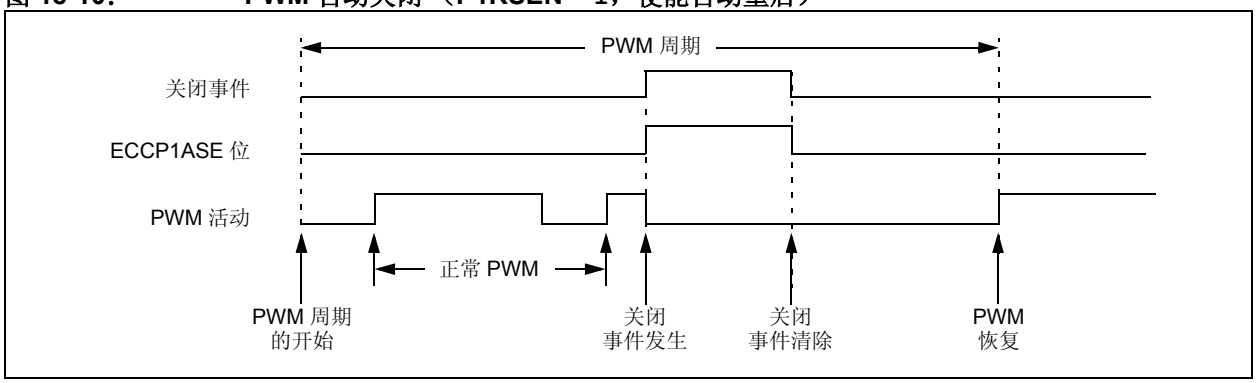
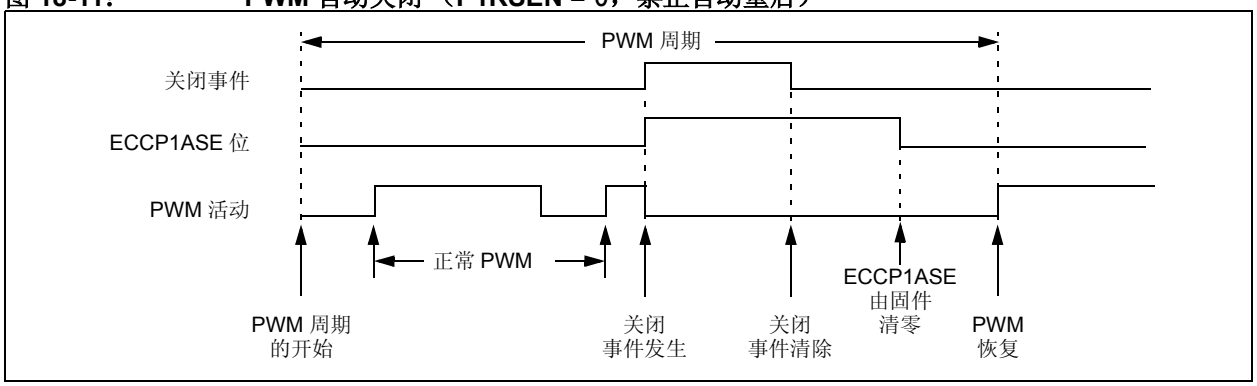


图 18-11： PWM 自动关闭（P1RSEN = 0，禁止自动重启）



© 2007 Microchip Technology Inc.

18.4.9 设置 PWM 操作

在将 ECCP 模块配置为 PWM 操作模式时应该执行以下步骤：

1. 通过将相应的 TRIS 位置 1，将 PWM 引脚 PxA 和 PxB （若使用 PxC 和 PxD，还包括这两个引脚）配置为输入。
2. 通过装载 PR2 （PR4）寄存器设置 PWM 周期。
3. 将 ECCP 模块配置为所需的 PWM 模式，通过用相应的值装载 CCPxCON 寄存器来完成该配置：
   - 通过 PxM1:PxM0 位选择一种可用的输出配置和方向。
   - 通过 CCPxM3:CCPxM0 位选择 PWM 输出信号的极性。
4. 可通过装载 CCPRxL 寄存器和 CCPxCON<5:4> 位来设置 PWM 占空比。
5. 对于自动关闭：
   - 禁止自动关闭； ECCPxASE = 0
   - 配置自动关闭源
   - 等待运行条件
6. 对于半桥输出模式，通过用相应的值装载 ECCPxDEL<6:0> 来设置死区延时。
7. 如果需要自动关闭操作，则装载 ECCPxAS 寄存器：
   - 使用 ECCPxAS2:ECCPxAS0 位选择自动关闭源。
   - 使用 PSSxAC1:PSSxAC0 和 PSSxBD1:PSSxBD0 位选择 PWM 输出引脚的关闭状态。
   - 将 ECCPxASE 位 （ECCPxAS<7>）置 1。
8. 如果需要自动重启操作，则将 PxRSEN 位（ECCPxDEL<7>）置 1。
9. 配置并启动 TMRn （TMR2 或 TMR4）：
   - 通过清零 TMRnIF 位 （Timer2 的 PIR1<1> 或 Timer4 的 PIR3<3>）来清零 TMRn 中断标志位。
   - 通过装载 TnCKPS 位 （TnCON<1:0>）来设置 TMRn 预分频值。
   - 通过将 TMRnON 位 （TnCON<2>）置 1 来使能 Timer2 （或 Timer4）。
10. 在新的 PWM 周期开始后，使能 PWM 输出：
    - 等待直到 TMRn 溢出为止 （TMRnIF 位置 1）。
    - 通过清零对应的 TRIS 位，将 ECCPx/PxA、PxB、 PxC 和 / 或 PxD 引脚使能为输出。
    - 将 ECCPxASE 位 （ECCPxAS<7>）清零。

18.4.10 复位的影响

上电复位和后续的复位会将所有端口强制为输入模式，并强制 ECCP 寄存器进入它们的复位状态。

这会将增强型 CCP 模块强制复位为与标准 CCP 模块兼容的状态。

表 18-5：　　与 ECCP 模块和 TIMER1 到 TIMER4 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	56
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	58
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	58
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	58
TRISH(1)	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	58
TMR1L(3)	Timer1 寄存器的低字节								56
TMR1H(3)	Timer1 寄存器的高字节								56
ODCON1(4)	—	—	—	CCP5OD	CCP4OD	ECCP3OD	ECCP2OD	ECCP1OD	56
T1CON(3)	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	56
TMR2(3)	Timer2 寄存器								56
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	56
PR2(3)	Timer2 周期寄存器								56
TMR3L	Timer3 寄存器的低字节								59
TMR3H	Timer3 寄存器的高字节								59
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	59
TMR4	Timer4 寄存器								59
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	59
PR4(3)	Timer4 周期寄存器								59
CCPRxL(2)	捕捉 / 比较 /PWM 寄存器 x 的低字节								57
CCPRxH(2)	捕捉 / 比较 /PWM 寄存器 x 的高字节								57
CCPxCON(2)	PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	57
ECCPxAS(2)	ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0	57
ECCPxDEL(2)	PxRSEN	PxDC6	PxDC5	PxDC4	PxDC3	PxDC2	PxDC1	PxDC0	57

图注：　— = 未实现位，读为 0。 ECCP 工作期间不使用阴影单元。

注　1： 只在 80 引脚器件上可用。

2： 所有增强型 CCP 模块中具有此名称的相同寄存器的通用称谓，其中 “x”标识了各个模块 （ECCP1、 ECCP2 或 ECCP3）。具有同一个通用名称的所有寄存器的位分配和复位值相同。

3： 当 WDTCON<4> = 0 时，此地址的默认 （传统） SFR 可用。

4： 配置 SFR 的地址与默认 SFR 重叠，仅当 WDTCON<4> = 1 时才能访问配置 SFR。

© 2007 Microchip Technology Inc.

19.0 主控同步串口 （MSSP）模块

19.1 主控 SSP （MSSP）模块概述

主控同步串口 （MSSP）模块是用于同其他外设或单片机器件进行通信的串行接口。这些外设包括串行EEPROM、移位寄存器、显示驱动器和 A/D 转换器等。MSSP 模块可以工作在以下两种模式：

- 串行外设接口 （SPI）
- I^2C™
  - 全主控模式
  - 从动模式 （广播地址呼叫）

I^2C 接口通过硬件支持下列模式：

- 主控模式
- 多主机模式
- 具有 5 位和 7 位地址屏蔽的从动模式 （10 位和 7 位寻址的地址屏蔽）

PIC18F87J11 系列的所有器件具有两个MSSP模块，称为 MSSP1 和 MSSP2。这两个模块彼此独立运行。

注：	本节是对 MSSP 模块的所有工作模式的通用参考，这些工作模式可以通过转换同样应用于 MSSP1 或 MSSP2。需要的时候，寄存器名称和模块 I/O 信号使用通用标识符“x”表示使用数字来区分某个特定模块。控制位名称不是个别的。

19.2 控制寄存器

每个 MSSP 模块都有 3 个相关的控制寄存器，包括一个状态寄存器（SSPxSTAT）和两个控制寄存器（SSPCON1 和 SSPCON2）。根据 MSSP 模块是在 SPI 模式还是 I^2C 模式下工作，这些寄存器的用途及它们各自的配置位将完全不同。

下面各章节提供了更多详细信息。

注：	在具有多个 MSSP模块的器件中，需要特别注意 SSPxCON 寄存器的名称。SSP1CON1 和 SSP1CON2 控制同一模块工作的不同方面，而 SSP1CON1 和 SSP2CON1 控制两个不同模块的相同功能。

19.3 SPI 模式

SPI 模式允许同时同步发送和接收 8 位数据。支持 SPI 的所有四种模式。通常使用以下三个引脚来完成通信：

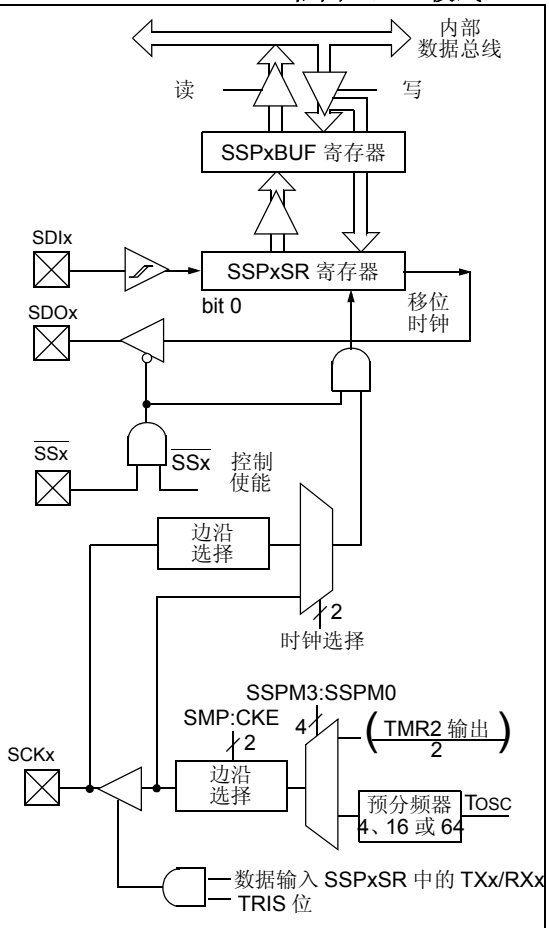
- 串行数据输出 （SDOx） – RC5/SDO1 或 RD4/SDO2
- 串行数据输入 （SDIx） – RC4/SDI1/SDA1 或 RD5/SDI2/SDA2
- 串行时钟 （SCKx） – RC3/SCK1/SCL1 或 RD6/SCK2/SCL2

此外，当工作在从动模式下可以使用第 4 个引脚：

- 从动选择 （$\overline{SSx}$） – RF7/$\overline{SS1}$ 或 RD7/$\overline{SS2}$

图 19-1 给出了 MSSP 模块在 SPI 模式下工作时的框图。

图 19-1： MSSP 框图 （SPI 模式）



注： 为了简洁起见，此框图中仅使用 I/O 端口的名称。复用功能的完整列表请参见正文。

19.3.1 寄存器

MSSP 模块有四个寄存器用于 SPI 模式。这些寄存器包括：

- MSSPx 控制寄存器 1 （SSPxCON1）
- MSSPx 状态寄存器 （SSPxSTAT）
- 串行接收 / 发送缓冲寄存器 （SSPxBUF）
- MSSP 移位寄存器 （SSPxSR）——不能直接访问

SSPxCON1 和 SSPxSTAT 是在 SPI 模式下工作的控制寄存器和状态寄存器。SSPxCON1 寄存器是可读写的。SSPxSTAT 的低 6 位是只读的。SSPxSTAT 的高 2 位是可读写的。

SSPxSR 是用来将数据移入或移出的移位寄存器。SSPxBUF 是缓冲寄存器，可用于数据字节的读取和写入。

接收时，SSPxSR 和 SSPxBUF 共同构成一个双重缓冲接收器。当 SSPxSR 接收到一个完整的字节后，该字节被送入 SSPxBUF 寄存器，同时将中断标志位 SSPxIF 置 1。

在发送过程中， SSPxBUF 并不是双重缓冲的。 对 SSPxBUF 的写操作将同时写入 SSPxBUF 和 SSPxSR。

寄存器 19-1： SSPxSTAT：MSSPx 状态寄存器 （SPI 模式）

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE(1)	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **SMP**：采样位
SPI 主控模式：
1 = 在数据输出时间结束时采样输入数据
0 = 在数据输出时间的中间采样输入数据
SPI 从动模式：
当 SPI 在从动模式下使用时，必须将 SMP 清零。

bit 6 **CKE**：SPI 时钟选择位 (1)
1 = 当时钟从有效状态转换到空闲状态时开始发送
0 = 当时钟从空闲状态转换到有效状态时开始发送

bit 5 **D/$\overline{A}$**：数据 / $\overline{地址}$位
仅在 I^2C 模式下使用。

bit 4 **P**：停止位
仅在 I^2C 模式下使用。当禁止 MSSPx 模块 （SSPEN 清零）时该位被清零。

bit 3 **S**：启动位
仅在 I^2C 模式下使用。

bit 2 **R/$\overline{W}$**：读 / $\overline{写}$信息位
仅在 I^2C 模式下使用。

bit 1 **UA**：更新地址位
仅在 I^2C 模式下使用。

bit 0 **BF**：缓冲器满状态位 （仅用于接收模式）
1 = 接收完成， SSPxBUF 满
0 = 接收未完成， SSPBUF 空

注 1： 时钟状态的极性由 CKP 位 （SSPxCON1<4>）设置。

© 2007 Microchip Technology Inc.

寄存器 19-2： SSPxCON1：MSSPx 状态寄存器 1 （SPI 模式）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **WCOL**：写冲突检测位
1 = 正在发送前一个字时，又有数据写入 SSPxBUF 寄存器 （必须用软件清零）
0 = 未发生冲突

bit 6 **SSPOV**：接收溢出指示位 (1)
SPI 从动模式：
1 = 当SSPxBUF中仍保存有前一数据时，又接收到一个新字节。如果溢出，SSPxSR中的数据会丢失。溢出只发生在从动模式下。即使只是发送数据，也必须读 SSPxBUF，以避免溢出标志位置 1 （必须由软件清零）。
0 = 未发生溢出

bit 5 **SSPEN**：主控同步串口使能位 (2)
1 = 使能串口并将 SCKx、 SDOx、 SDIx、 $\overline{\text{SSx}}$ 配置为串口引脚
0 = 禁止串口并将上述引脚配置为 I/O 端口引脚

bit 4 **CKP**：时钟极性选择位
1 = 空闲状态时，时钟为高电平
0 = 空闲状态时，时钟为低电平

bit 3-0 **SSPM3:SSPM0**：主控同步串口模式选择位 (3)
0101 = SPI 从动模式，时钟 = SCKx 引脚，禁止 $\overline{\text{SSx}}$ 引脚控制功能，可将 $\overline{\text{SSx}}$ 用作 I/O 引脚
0100 = SPI 从动模式，时钟 = SCKx 引脚，使能 $\overline{\text{SSx}}$ 引脚控制功能
0011 = SPI 主控模式，时钟 = TMR2 输出 /2
0010 = SPI 主控模式，时钟 = FOSC/64
0001 = SPI 主控模式，时钟 = FOSC/16
0000 = SPI 主控模式，时钟 = FOSC/4

注 1： 在主控模式下，溢出位不会被置 1，因为每次接收 （和发送）新数据都是通过写 SSPxBUF 寄存器启动的。
2： 当该位使能时，必须将相应的引脚正确地配置为输入或输出。
3： 此处没有列出的位组合为保留或仅在 I^2C 模式下使用。

19.3.2 工作原理

当初始化 SPI 时，需要指定几个选项。这可通过编程相应的控制位 （SSPxCON1<5:0> 和 SSPxSTAT<7:6>）来指定。这些控制位用于设置以下选项：

- 主控模式 （SCKx 作为时钟输出）
- 从动模式 （SCKx 作为时钟输入）
- 时钟极性 （SCKx 处于空闲状态）
- 输入数据的采样相位 （数据输出时间的中间或末端）
- 时钟边沿 （在 SCKx 的上升沿 / 下降沿输出数据）
- 时钟速率 （仅主控模式下）
- 从动选择模式 （仅从动模式下）

MSSP 模块由一个发送 / 接收移位寄存器 （SSPxSR）和一个缓冲寄存器 （SSPxBUF）组成。SSPxSR 将数据移入 / 移出器件，最高有效位在前。在新数据接收完毕前，SSPxBUF 保存上次写入 SSPxSR 的数据。一旦 8 位数据接收完毕，该字节就被移入 SSPxBUF 寄存器。然后，缓冲器满检测位 BF（SSPxSTAT<0>）和中断标志位 SSPxIF 均被置 1。这种双重缓冲数据接收方式（SSPxBUF）允许在读取刚接收的数据之前，就开始接收下一个字节。在数据发送 / 接收期间，任何试图写 SSPxBUF 寄存器的操作均无效，并将写冲突检测位 WCOL （SSPxCON1<7>）置 1。用户必须用软件将 WCOL 位清零才能判断以后对 SSPxBUF 寄存器的写入是否成功。

为确保应用软件能有效地接收数据，应该在要发送的下一数据字节写入 SSPxBUF 之前，读取 SSPxBUF 中现有的数据。缓冲器满位 BF （SSPxSTAT<0>）用于表示何时 SSPxBUF 载入了已接收的数据 （发送完成）。当 SSPxBUF 中的数据被读取后，BF 位即被清零。如果 SPI 仅仅作为一个发送器，则不必理会接收的数据。通常可用 MSSP 中断来判断发送或接收何时完成。如果不使用中断来处理数据的收发，用软件查询方法同样可确保不会发生写冲突。例 19-1 显示装载 SSPxBUF（SSPxSR）进行数据发送的过程。

用户不能直接读写 SSPxSR 寄存器，只能通过寻址 SSPxBUF 寄存器来访问它。此外， SSPxSTAT 寄存器指示各种状态。

19.3.3 开漏输出选项

可选择将 SDOx 输出和 SCKx 时钟引脚的驱动器配置为开漏输出。此功能允许通过外部上拉电阻将 SCKx 时钟引脚上的电平拉至更高的电平，还允许 SDOx 输出与外部电路通信，而无需额外的电平移位器。更多信息，请参见**第 10.1.4 节 “漏极开路输出”**。

开漏输出选项由 SPI2OD 和 SPI1OD 位（ODCON3<1:0>）控制。置 1 SPIxOD 位可将相应模块的 SDOx 和 SCKx 引脚配置为开漏操作。

ODCON3 寄存器与 T1CON 寄存器共享同一地址。通过将 WDTCON 寄存器中的 ADSHR 位（WDTCON<4>）置 1，可访问 ODCON3 寄存器。

例 19-1： 装载 SSP1BUF （SSP1SR）寄存器

```
LOOP    BTFSS   SSP1STAT, BF    ;Has data been received (transmit complete)?
        BRA     LOOP            ;No
        MOVF    SSP1BUF, W      ;WREG reg = contents of SSP1BUF
        MOVWF   RXDATA          ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W       ;W reg = contents of TXDATA
        MOVWF   SSP1BUF         ;New data to xmit
```


© 2007 Microchip Technology Inc.

19.3.4 使能 SPI I/O

要使能串行端口，MSSP 使能位 SSPEN（SSPxCON1<5>）必须置 1。要复位或重新配置 SPI 模式，应先将 SSPEN 位清零，重新初始化SSPxCON 寄存器，然后将 SSPEN 位置 1。这样就将 SDIx、SDOx、SCKx 和 SSx 引脚配置为串口引脚。要让上述引脚充当串口引脚，必须按如下说明正确设置其中一些引脚的数据方向位（在 TRIS 寄存器中）：

- SDIx 由 SPI 模块自动控制
- SDOx 必须将 TRISC<5> 或 TRISD<4> 位清零
- SCKx（主控模式）必须将 TRISC<3> 或 TRISD<6> 位清零
- SCKx（主控模式）必须将 TRISC<3> 或 TRISD<6> 位清零
- $\overline{SSx}$ 必须将 TRISF<7> 或 TRISD<7> 位置 1

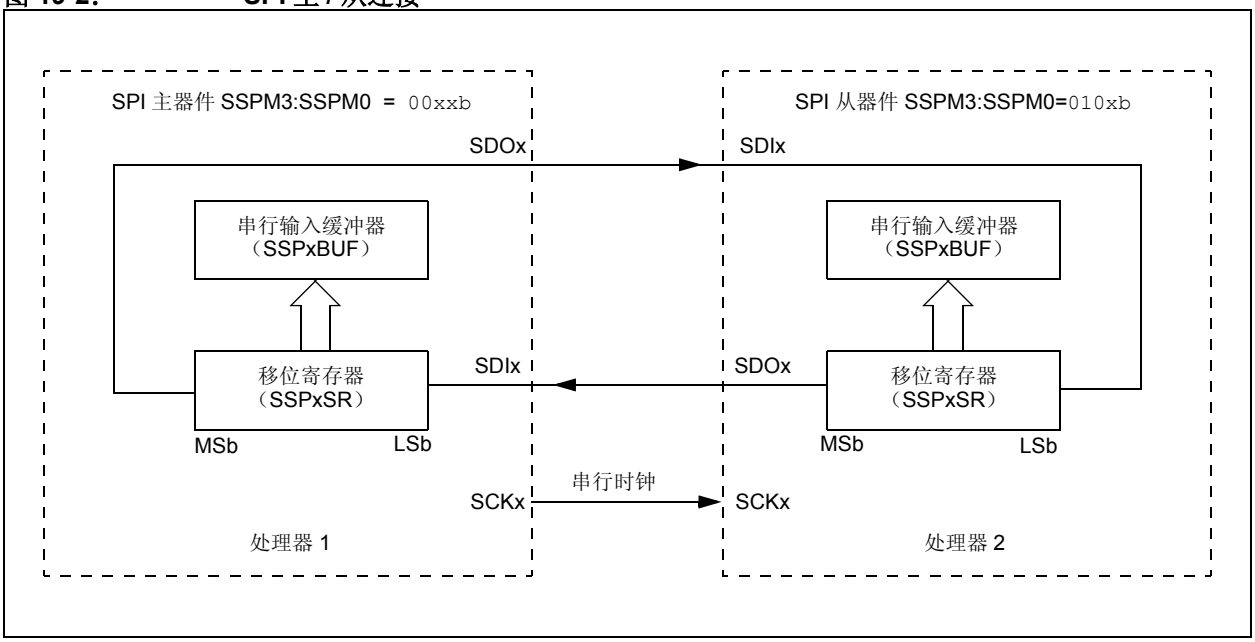
对于任何不需要的串口功能，可通过将对应的数据方向（TRIS）寄存器设置为相反值来屏蔽。

19.3.5 典型连接

图 19-2 给出了两个单片机之间的典型连接。主器件（处理器 1）通过发送 SCKx 信号来启动数据传输。在两个处理器的移位寄存器之间，数据在编程设定的时钟边沿被传送，并在相反的时钟边沿被锁存。必须将两个处理器的时钟极性（CKP）设定为相同，这样两个控制器就可以同时收发数据。数据是否有效取决于应用软件。这就导致以下三种数据传输情形：

- 主器件发送数据——从器件发送无效（Dummy）数据
- 主器件发送数据——从器件发送数据
- 主器件发送无效数据——从器件发送数据

图 19-2： SPI 主 / 从连接



19.3.6 主控模式

由于主器件控制 SCKx 信号，所以可随时启动数据传输。主器件根据软件协议确定从器件（图 19-2 中的处理器 1）应在何时广播数据。

在主控模式下，数据一旦写入 SSPxBUF 寄存器就开始发送或接收。如果SPI仅作为接收器，则可以禁止SDOx输出（通过编程将其设置为输入）。SSPxSR 寄存器按设定的时钟速率连续移入 SDIx 引脚上的信号。每收到一个字节，就将其装入 SSPxBUF 寄存器，就像接收普通字节一样（中断和状态位相应置 1）。这在以“线路活动监控”（Line Activity Monitor）方式工作的接收器应用中很有用。

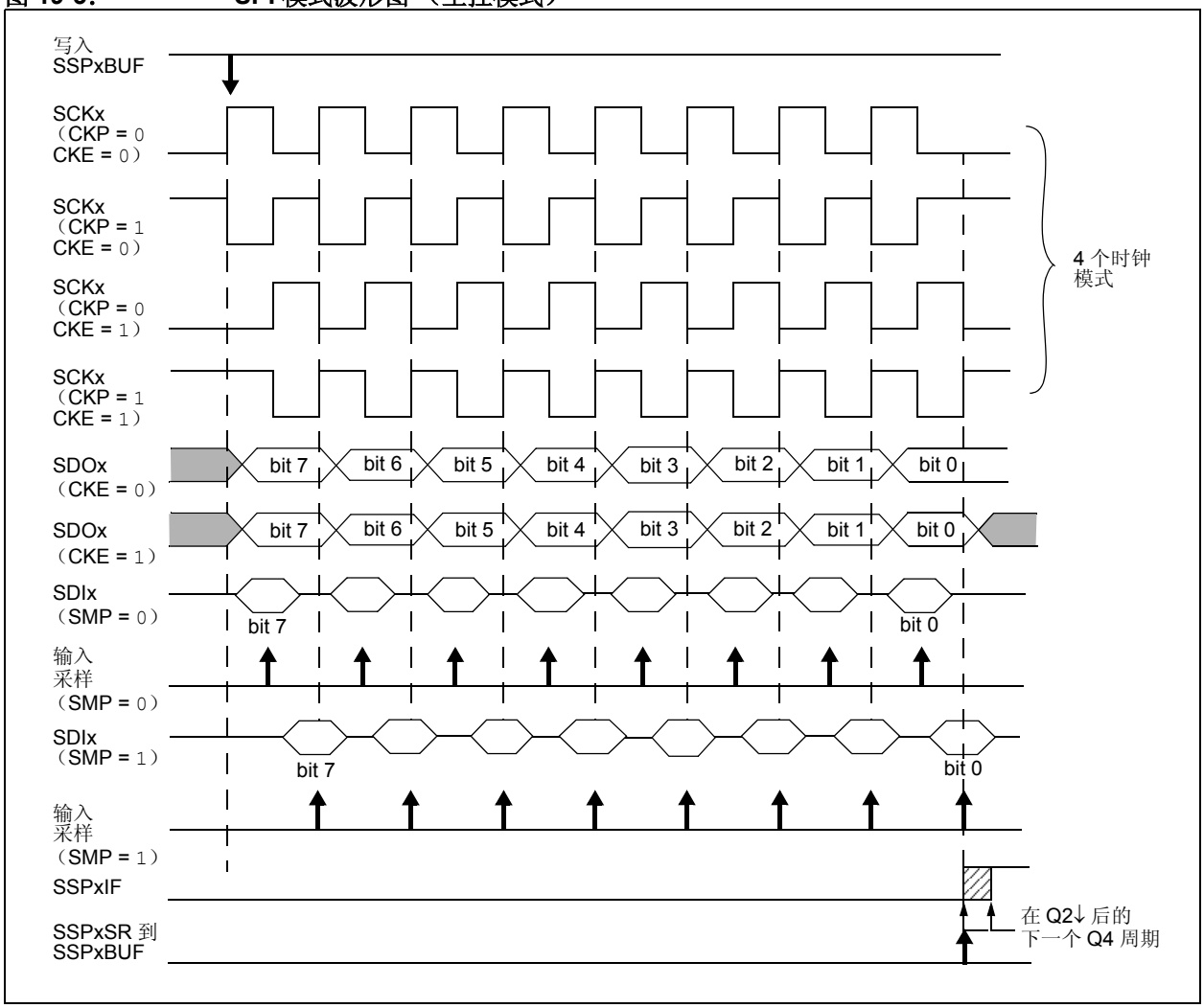
时钟极性可通过对 CKP 位（SSPxCON<4>）进行适当的编程来选择。图 19-3、图 19-5 和图 19-6 将给出 SPI 通信的波形图，其中最先发送最高有效位。在主控模式下，SPI 时钟速率（比特率）可由用户编程设定为下面某种速率：

- FOSC/4 （或 TCY）
- FOSC/16 （或 4 TCY）
- FOSC/64 （或 16 TCY）
- Timer2 输出 /2

这样允许数据速率最大为 10.00 Mbps （时钟速率为 40 MHz）。

图 19-3 给出了主控模式的波形图。当 CKE 位置 1 时，SDOx 数据在 SCKx 出现时钟边沿前一直有效。图中所示输入采样的变化由 SMP 位的状态决定。图中给出了接收到的数据装入 SSPxBUF 的时间。

图 19-3： SPI 模式波形图 （主控模式）



 © 2007 Microchip Technology Inc.

19.3.7 从动模式

在从动模式下，当 SCKx 引脚上有外部时钟脉冲时启动发送和接收数据。当最后一位数据被锁存后，中断标志位 SSPxIF 置 1。

在从动模式下，外部时钟由 SCKx 引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。

在休眠状态下，从器件仍可发送 / 接收数据。当收到一个字节时，将唤醒休眠状态下的器件。

19.3.8 从动选择同步

$\overline{SSx}$ 引脚允许同步从动模式。SPI 必须处于从动模式，并使能 $\overline{SSx}$ 引脚控制功能（SSPxCON1<3:0> = 04h）。当 $\overline{SSx}$ 引脚为低电平时，将使能数据的发送和接收，同时驱动 SDOx 引脚。当 $\overline{SSx}$ 引脚为高电平时，即使是在数据的发送过程中，SDOx 引脚也不再被驱动，而是变成悬空输出。根据应用需要，可在 SDOx 引脚上外接上拉 / 下拉电阻。

注 1： 当 SPI 处于从动模式，并使能 $\overline{SSx}$ 引脚控制功能（SSPxCON1<3:0> = 0100）时，如果 $\overline{SSx}$ 引脚置为 VDD 电平将使 SPI 模块复位。

2： 如果 SPI 工作在从动模式并且 CKE 置 1，则必须使能 $\overline{SSx}$ 引脚控制功能。

当 SPI 模块复位后，位计数器被强制为 0。这可以通过强制将SSx引脚拉为高电平或将SSPEN位清零来实现。

将 SDOx 引脚和 SDIx 引脚相连，可以仿真二线制通信。当 SPI 需要作为接收器工作时，SDOx 引脚可以被配置为输入端。这样就禁止了从 SDOx 发送数据。由于 SDIx 不会引起总线冲突，因而总是可以将其保留为输入（SDI 功能）。

图 19-4： 从动同步波形图

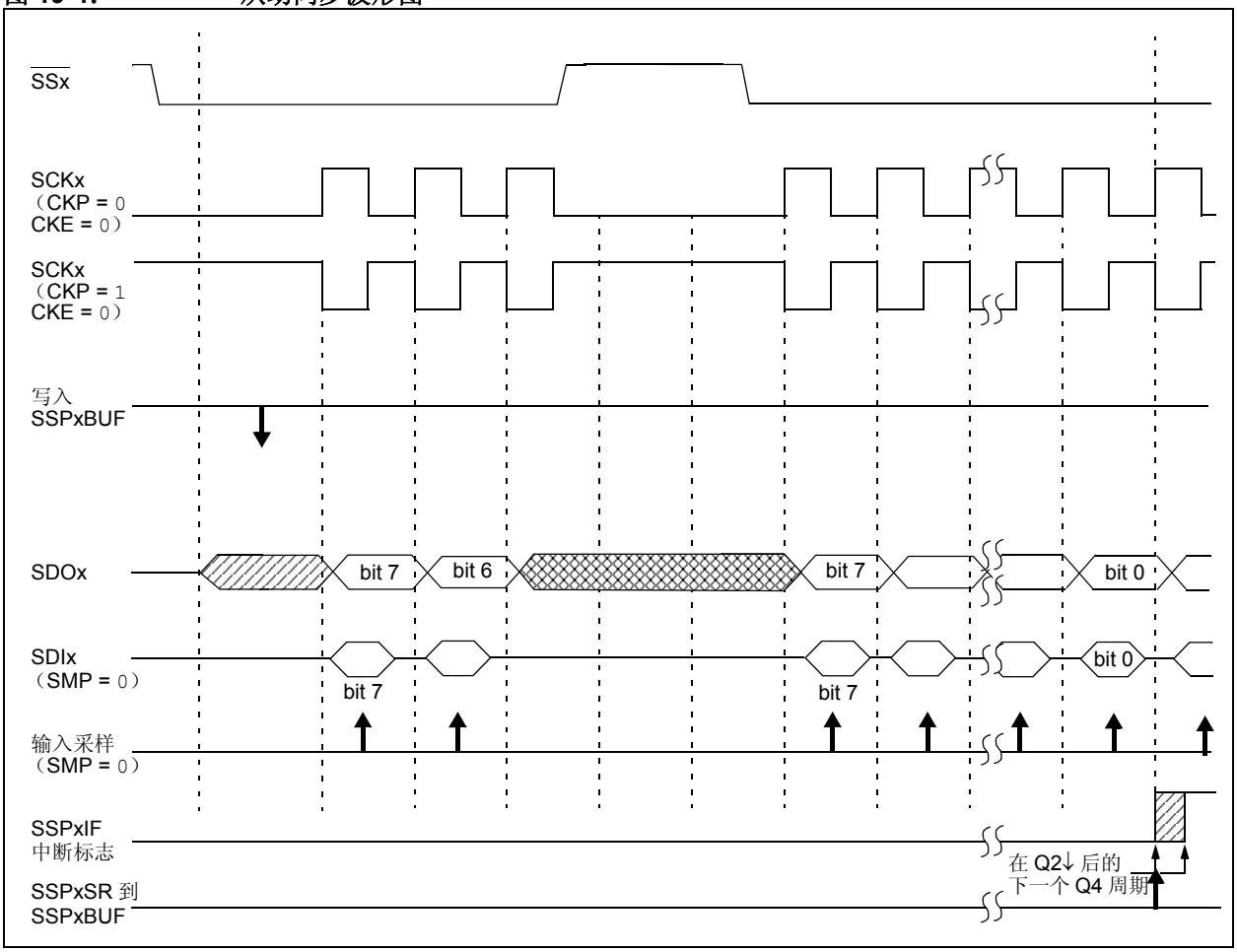


图 19-5： SPI 模式波形图 （从动模式且 CKE = 0）

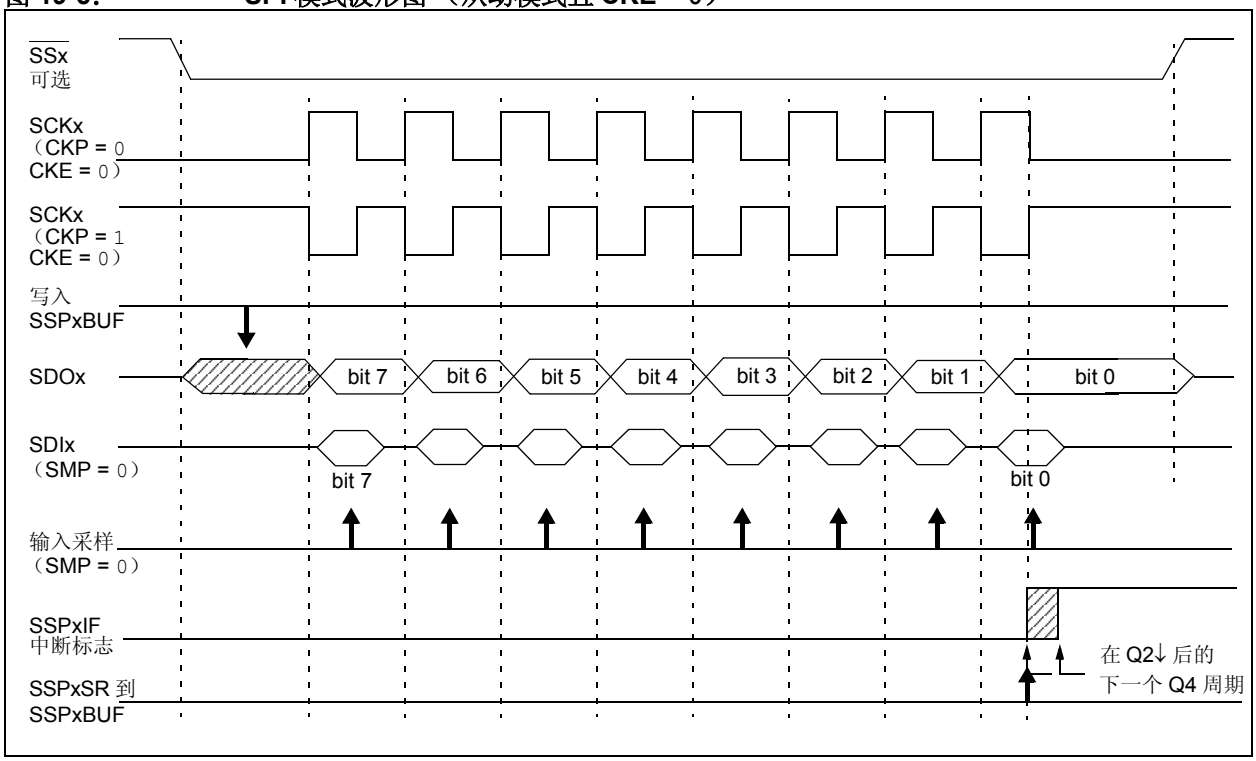
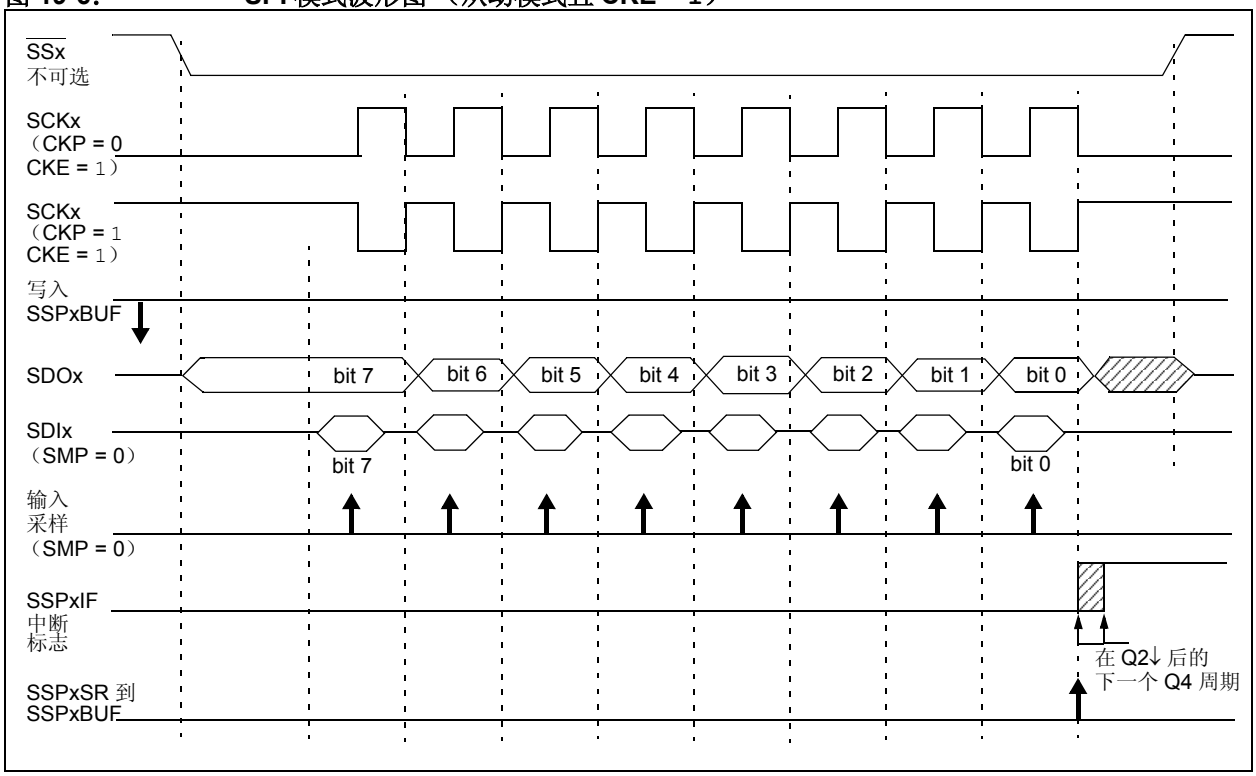


图 19-6： SPI 模式波形图 （从动模式且 CKE = 1）



 © 2007 Microchip Technology Inc.

19.3.9 在功耗管理模式下的操作作

在 SPI 主控模式下，模块时钟速度与全功耗模式下的不同。处于休眠模式时，所有时钟都停止。

在空闲模式下，为外设提供一个时钟。该时钟可以来自于主时钟源、辅助时钟源（Timer1 振荡器）或 INTOSC 时钟源。更多信息，请参见**第 2.3 节 “时钟源和振荡器切换”**。

在大多数情况下，主器件为 SPI 数据提供的时钟速率并不重要。但是，每个系统都应该评估此因素。

如果允许了 MSSP 中断，那么当主器件发送完数据时这些中断可以将控制器从休眠模式或某种空闲模式唤醒。如果不想从休眠或空闲模式退出，应禁止 MSSP 中断。

表 19-1： SPI 总线模式

标准 SPI 模式术语	控制位状态	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

还有一个 SMP 位用来控制何时采样数据。

19.3.12 SPI 时钟速度和模块的相互关系

由于 MSSP1 和 MSSP2 是独立的模块，所以它们能够以不同的数据速率同时运行。将 SSPxCON1 寄存器的 SSPM3:SSPM0 位置 1 可确定相应模块的速率。

在主控模式下，这两个模块都将 Timer2 用作时基，这是一个特例。这种情况下，Timer2 模块操作上的任何更改都将对这两个MSSP模块造成同等影响。如果每个模块需要不同的比特率，用户应该为另一个模块选择其他三种时基中的一种。

如果选择了休眠模式，所有模块的时钟都将停止，并且在器件被唤醒前，发送 / 接收将保持此停滞状态。当器件返回到运行模式后，该模块将恢复发送和接收数据。

在 SPI 从动模式下， SPI 发送 / 接收移位寄存器与器件异步工作。这可以使器件处于任何功耗管理模式下，而且数据仍可被移入 SPI 发送 / 接收移位寄存器。当 8 位全部接收完后， MSSP 中断标志位将置 1，并且如果允许中断的话，将唤醒器件。

19.3.10 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

19.3.11 总线模式兼容性

表 19-2： 与 SPI 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	58
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	—	—	58
SSP1BUF	MSSP1 接收缓冲器 / 发送寄存器								56
SSPxCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	56, 59
SSPxSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	56, 59
SSP2BUF	MSSP2 接收缓冲器 / 发送寄存器								59
ODCON3(1)	—	—	—	—	—	—	SPI2OD	SPI1OD	56

图注： SPI 模式下的 MSSP 模块不使用阴影单元。

注 1： 配置 SFR 的地址与默认 SFR 重叠；仅在 WDTCON<4> = 1 时才可访问配置 SFR。

© 2007 Microchip Technology Inc.

19.4 I²C™ 模式

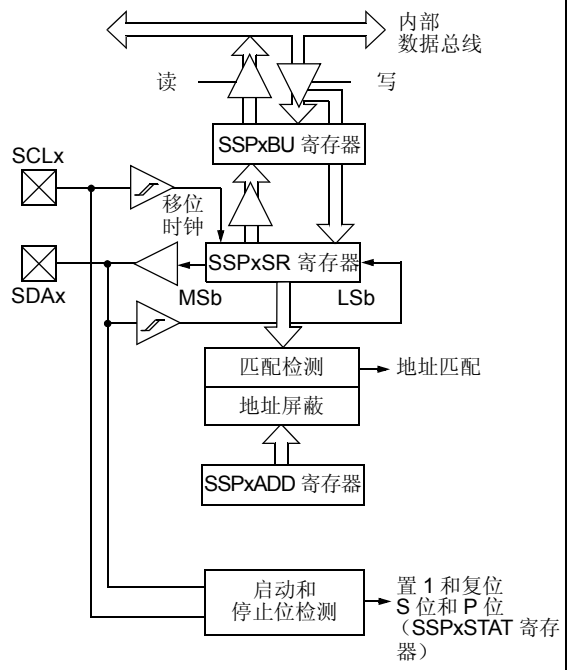
MSSP 模块工作在 I²C 模式时，可以实现所有主控和从动功能 （包括广播呼叫支持），并且用硬件提供启动位和停止位的中断来判断总线何时空闲 （多主机功能）。MSSP 模块实现标准模式规范以及 7 位和 10 位寻址。

有两个引脚用于数据传输：

- 串行时钟 （SCKx）——RC3/SCK1/SCL1 或 RD6/SCK2/SCL2
- 串行数据 （SDAx）——RC4/SDI1/SDA1 或 RD5/SDI2/SDA2

用户必须通过 TRIS 位将这些引脚配置为输入引脚。

图 19-7： MSSP 框图 （I²C™ 模式）



注： 为了简便起见，此图仅使用端口 I/O 名称。复用功能的完整列表请参见正文。

19.4.1 寄存器

MSSP 模块有 6 个寄存器用于 I²C 操作。这些寄存器包括：

- MSSPx 控制寄存器 1 （SSPxCON1）
- MSSPx 控制寄存器 2 （SSPxCON2）
- MSSPx 状态寄存器 （SSPxSTAT）
- 串行接收 / 发送缓冲寄存器 （SSPxBUF）
- MSSPx 移位寄存器 （SSPxSR） － 不能直接访问
- MSSPx 地址寄存器 （SSPxADD）
- I²C 从动地址屏蔽寄存器 （SSPxMSK）

SSPxCON1、SSPxCON2 和 SSPxSTAT是工作在 I²C 模式下的控制和状态寄存器。SSPxCON1 和 SSPxCON2 寄存器是可读写的。SSPxSTAT 的低 6 位是只读的，而高 2 位是可读写的。

SSPxSR 是用来将数据移入或移出的移位寄存器。SSPxBUF是缓冲寄存器，用于数据字节的写入和读出。

在 I²C 从动模式下配置 MSSP 时，SSPxADD 将保存从器件的地址。在主控模式下配置 MSSP 时， SSPxADD 的低 7 位保存波特率发生器的重载值。

当 SSPxMSK 模块配置为 7 位地址屏蔽模式时，它将保存从动地址屏蔽值。当 SSPxMSK 作为一个单独的寄存器时，它将与 SSPxADD 共享同一 SFR 地址；只有在专门将 SSPM3:SSPM 位置 1 以允许访问时才可访问 SSPxMSK。欲知更多详情，请参见**第 19.4.3.4 节 “7 位地址屏蔽模式”**。

接收时，SSPxSR 和 SSPxBUF 共同构成一个双重缓冲接收器。当 SSPxSR 接收到一个完整的字节后，该字节被送入 SSPxBUF 寄存器，同时将中断标志位 SSPxIF 置 1。

在发送过程中， SSPxBUF 并不是双重缓冲的。 对SSPxBUF的写操作将同时写入SSPxBUF和SSPxSR。

寄存器 19-3： SSPxSTAT：MSSPx 状态寄存器 （I²C™ 模式）

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P(1)	S(1)	R/$\overline{W}$(2,3)	UA	BF
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **SMP**：变化率控制位
在主控或从动模式下：
1 = 标准速度模式下禁止变化率控制 （100 kHz 和 1 MHz）
0 = 高速模式下使能变化率控制 （400 kHz）

bit 6 **CKE**：SMBus 选择位
在主控或从动模式下：
1 = 使能 SMBus 特定输入
0 = 禁止 SMBus 特定输入

bit 5 **D/$\overline{A}$**：数据 / $\overline{地址}$位
在主控模式下：
保留。
在从动模式下：
1 = 表示上一个接收或发送的字节是数据
0 = 表示上一个接收或发送的字节是地址

bit 4 **P**：停止位 (1)
1 = 表示上一次检测到停止位
0 = 表示上一次未检测到停止位

bit 3 **S**：启动位 (1)
1 = 表示上一次检测到启动位
0 = 表示上一次未检测到启动位

bit 2 **R/$\overline{W}$**：读 / $\overline{写}$信息位 (2,3)
在从动模式下：
1 = 读
0 = 写
在主控模式下：
1 = 正在发送
0 = 未发送

bit 1 **UA**：更新地址位 （仅在 10 位从动模式下）
1 = 表示用户需要更新 SSPxADD 寄存器中的地址
0 = 不需要更新地址

bit 0 **BF**：缓冲器满状态位
在发送模式下：
1 = SSPxBUF 已满
0 = SSPxBUF 为空
在接收模式下：
1 = SSPxBUF 已满 （不包括 $\overline{ACK}$ 位和停止位）
0 = SSPxBUF 为空 （不包括 ACK 位和停止位）

注 1： 当 SSPEN 被清零时该位在复位时清零。
2： 该位用来保存在最近一次地址匹配后的 R/$\overline{W}$ 位信息。该位仅在从地址匹配开始到下一个启动位、停止位或非 $\overline{ACK}$ 位之间有效。
3： 该位与 SEN、 RSEN、 PEN、 RCEN 或 ACKEN 进行 “或” 运算的结果表示 MSSPx 是否处于活动模式。

© 2007 Microchip Technology Inc.

寄存器 19-4： SSPxCON1：MSSPx 控制寄存器 1 （I²C™ 模式）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **WCOL**：写冲突检测位

在主控发送模式下：

1 = 当 I²C 不满足启动发送数据的条件时，试图写 SSPxBUF 寄存器 （必须用软件清零）

0 = 未发生冲突

在从动发送模式下：

1 = 正在发送前一个字时，又有数据写入 SSPxBUF 寄存器 （必须用软件清零）

0 = 未发生冲突

在接收模式下 （主控或从动模式）：

此位被忽略。

bit 6 **SSPOV**：接收溢出指示位

在接收模式下：

1 = SSPxBUF 寄存器仍保存有前一字节时，接收到一个新的字节 （必须用软件清零）

0 = 未发生溢出

在发送模式下：

此位被忽略。

bit 5 **SSPEN**：主控同步串口使能位 (1)

1 = 使能串口并将 SDAx 和 SCLx 引脚配置为串口引脚。

0 = 禁止串口并将这些引脚配置为 I/O 端口引脚

bit 4 **CKP**：SCKx 释放控制位

在从动模式下：

1 = 释放时钟

0 = 保持时钟低电平 （延长低电平时间），用来确保数据建立时间

在主控模式下：

在此模式下未使用。

bit 3-0 **SSPM3:SSPM0**：主控同步串口模式选择位 (2)

1111 = I²C 从动模式， 10 位地址，并允许启动位和停止位中断

1110 = I²C 从动模式， 7 位地址，并允许启动位和停止位中断

1011 = I²C 由固件控制的主控模式 （从器件空闲）

1001 = 将 SSPMSK 寄存器的内容载入 SSPADD SFR 地址 (3,4)

1000 = I²C 主控模式，时钟 = FOSC/(4 * (SSPxADD + 1))

0111 = I²C 从动模式， 10 位地址

0110 = I²C 从动模式， 7 位地址

注 1： 当该位被使能时，必须将 SDAx 和 SCLx 引脚配置为输入引脚。

2： 此处未列出的位组合为保留或只用于 SPI 模式。

3： 当 SSPM3:SSPM0 = 1001 时，任何对 SSPxADD SFR 地址的读写操作实际访问的是 SSPxMSK 寄存器。

4： 仅在选择 7 位地址屏蔽模式时此模式才可用 （MSSPMSK 配置位为 1）。

寄存器 19-5： SSPxCON2：MSSPx 控制寄存器 2 （I²C™ 主控模式）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT(1)	ACKEN(2)	RCEN(2)	PEN(2)	RSEN(2)	SEN(2)
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **GCEN**：广播呼叫使能位
不用于主控模式。

bit 6 **ACKSTAT**：应答状态位 （仅用于主控发送模式）
1 = 未收到来自从器件的应答
0 = 收到来自从器件的应答

bit 5 **ACKDT**：应答数据位 （仅用于主控接收模式） (1)
1 = 无应答
0 = 应答

bit 4 **ACKEN**：应答序列使能位 (2)
1 = 在 SDAx 和 SCLx 引脚上发起应答序列，并发送 ACKDT 数据位。由硬件自动清零。
0 = 应答序列空闲

bit 3 **RCEN**：接收使能位 （仅用于主控接收模式） (2)
1 = 使能 I²C 接收模式
0 = 接收空闲

bit 2 **PEN**：停止条件使能位 (2)
1 = 在 SDAx 和 SCLx 引脚上发起停止条件。由硬件自动清零。
0 = 停止条件空闲

bit 1 **RSEN**：重复启动条件使能位 (2)
1 = 在 SDAx 和 SCLx 引脚上发起重复启动条件。由硬件自动清零。
0 = 重复启动条件空闲

bit 0 **SEN**：启动条件使能位 (2)
1 = 在 SDAx 和 SCLx 引脚上发起启动条件。由硬件自动清零。
0 = 启动条件空闲

注 1： 用户在接收结束时发起一个应答序列，同时发送该值。
2： 如果 I²C 模块处于活动状态，则这些位不能被置 1 （没有缓存），并且不能对 SSPxBUF 进行写操作 （或者禁止写 SSPxBUF）。

© 2007 Microchip Technology Inc.

寄存器 19-6： SSPxCON2：MSSPx 控制寄存器 2 （I²C™ 从动模式）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ADMSK5	ADMSK4	ADMSK3	ADMSK2	ADMSK1	SEN(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7 **GCEN**：广播呼叫使能位
1 = 当 SSPSR 接收到广播呼叫地址 （0000h）时允许中断
0 = 禁止广播呼叫地址

bit 6 **ACKSTAT**：应答状态位
不用于从动模式。

bit 5-2 **ADMSK5:ADMSK2**：从动地址屏蔽选择位 （5 位地址屏蔽模式）
1 = 使能屏蔽 SSPxADD 的相应位
0 = 禁止屏蔽 SSPxADD 的相应位

bit 1 **ADMSK1**：从动地址最低有效位屏蔽选择位
在 7 位寻址模式下：
1 = 仅使能屏蔽 SSPxADD<1>
0 = 仅禁止屏蔽 SSPxADD<1>
在 10 位寻址模式下：
1 = 使能屏蔽 SSPxADD<1:0>
0 = 禁止屏蔽 SSPxADD<1:0>

bit 0 **SEN**：延长使能位 (1)
1 = 为从动发送和从动接收 （已使能延长）使能时钟延长
0 = 禁止时钟延长

注 1： 如果 I²C 模块处于活动状态，则这些位不能被置 1 （没有缓存），并且不能对 SSPBUF 进行写操作 （或者禁止写 SSPxBUF）。

寄存器 19-7： SSPxMSK：I²C™ 从动地址屏蔽寄存器 （7 位屏蔽模式）(1)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0(2)
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-0 **MSK7:MSK0**：从动地址屏蔽选择位
1 = 使能屏蔽 SSPxADD 的相应位
0 = 禁止屏蔽 SSPxADD 的相应位

注 1： 此寄存器与 SSPxADD 共享同一 SFR 地址，并且仅在选择 MSSPx 工作模式时才可寻址。详细情况，请参见**第 19.4.3.4 节 “7 位地址屏蔽模式”**。
2： MSK0 不用作 7 位寻址模式下的屏蔽位。

19.4.2 工作原理

通过将 MSSP 使能位 SSPEN（SSPxCON1<5>）置 1，可使能 MSSP 模块。

SSPxCON1 寄存器用于控制 I^2C 的工作模式。可通过设置模式选择位 （SSPxCON1<3:0>）选择以下几种 I^2C 模式之一：

- I^2C 主控模式，时钟
- I^2C 从动模式 （7 位地址）
- I^2C 从动模式 （10 位地址）
- I^2C 从动模式 （7 位地址），允许启动位和停止位中断
- I^2C 从动模式 （10 位地址），允许启动位和停止位中断
- I^2C 固件控制的主控模式，从器件空闲

通过将相应的 TRISC 或 TRIS 位置 1，将 SCLx 和 SDAx 引脚编程为输入引脚；在 SSPEN 位置 1 时选择任何 I^2C 模式，将强制上述引脚漏极开路。要确保此模块的正常工作，必须为 SCLx 和 SDAx 引脚提供外接上拉电阻。

19.4.3 从动模式

在从动模式下， SCLx 引脚和 SDAx 引脚必须被配置为输入（TRISC<4:3> 置 1）。必要时（从发送器）MSSP 模块将使用输出数据改写输入状态。

I^2C 从动模式硬件总是在地址匹配时产生中断。地址屏蔽允许此硬件为多个地址 （7 位寻址模式下最多 31 个地址， 10 位寻址模式下最多 63 个地址）产生中断。用户也可以通过模式选择位选择启动位或停止位中断。

当地址匹配时或在地址匹配后传送的数据被接收时，硬件会自动产生一个应答（$\overline{ACK}$）脉冲，并把当前 SSPxSR 寄存器中接收到的数据装入 SSPxBUF 寄存器。

只要满足下列条件之一，MSSP 模块就不会产生此 $\overline{ACK}$ 脉冲：

- 在接收到传输的数据前，缓冲器满标志位 BF （SSPxSTAT<0>）置 1。
- 在接收到传输的数据前，溢出位 SSPOV （SSPxCON1<6>）置 1。

在这种情况下， SSPxSR 寄存器的值不会载入 SSPxBUF，但是 SSPxIF 位会置 1。BF 位是通过读取 SSPxBUF 寄存器清零的，而 SSPOV 位必须通过软件来清零。

为确保正常工作， SCLx 时钟输入必须满足最小高低电平时间要求。关于 I^2C 规范的高电平和低电平时间以及对 MSSP 模块的具体要求，请参见时序参数 100 和 101。

19.4.3.1 寻址

一旦使能了 MSSP 模块，它就会等待启动条件出现。在启动信号出现后，8 位数据被移入 SSPxSR 寄存器。在时钟 （SCLx）线的上升沿采样所有的输入位。在第 8 个时钟（SCLx）脉冲的下降沿，寄存器 SSPxSR<7:1> 的值会和 SSPxADD 地址寄存器的值进行比较。如果地址匹配，并且 BF 位和 SSPOV 位被清零，则会发生下列事件：

1. SSPxSR 寄存器值装入 SSPxBUF 寄存器。
2. 缓冲器满标志位 BF 置 1。
3. 产生 $\overline{ACK}$ 脉冲。
4. 在第 9 个 SCLx 脉冲的下降沿，MSSP 中断标志位 SSPxIF 被置 1（如果允许中断，则产生中断）。

在 10 位寻址模式下，从器件需要接收两个地址字节。第一个地址字节的高 5 位指定这是否为 10 位地址。R/W 位 （SSPxSTAT<2>）必须指定写操作，这样从器件才能接收到第二个地址字节。对于 10 位地址，第一个字节应该是 `11110 A9 A8 0`，其中 `A9` 和 `A8` 是地址的两个最高有效位。10 位地址传输的操作步骤如下，其中 7-9 步是针对从发送器而言的。

1. 接收地址的第一个 （高）字节 （地址匹配时 SSPxIF 位、 BF 位和 UA 位置 1）。
2. 用地址的第二个（低）字节更新 SSPxADD 寄存器 （UA 位清零并释放 SCLx 线）。
3. 读 SSPxBUF 寄存器 （BF 位清零）并将标志位 SSPxIF 清零。
4. 接收地址的第二个（低）字节（SSPxIF 位、BF 位和 UA 位置 1）。
5. 使用地址的第一个（高）字节更新 SSPxADD 寄存器。如果匹配的话就释放 SCLx 线，这将清零 UA 位。
6. 读 SSPxBUF 位（BF 位清零）并将标志位 SSPxIF 清零。
7. 接收重复启动条件。
8. 接收地址的第一个（高）字节（SSPxIF 位和 BF 位置 1）。
9. 读 SSPxBUF 寄存器 （BF 位清零）并将标志位 SSPxIF 清零。

© 2007 Microchip Technology Inc.

19.4.3.2 地址屏蔽模式

屏蔽地址位将导致该位变为“无关位”。若屏蔽了一个地址位，将应答两个地址并会引起中断。可一次屏蔽多个地址位，这样便扩展了被应答地址的数量。

I^2C 从器件的工作状态与是否屏蔽地址无关。但是，当使用了地址屏蔽时， I^2C 从器件可应答多个地址并会引起中断。当出现这种情况时，有必要通过检查SSPxBUF来确定由哪个地址引起中断。

PIC18F87J11 系列器件能够在 I^2C 从器件工作中使用两种地址屏蔽模式：5 位地址屏蔽和 7 位地址屏蔽模式。屏蔽模式是在配置器件时使用MSSPMSK配置位进行选择。默认的器件配置是 7 位地址屏蔽模式。

这两种屏蔽模式又进一步支持 7 位和 10 位地址的地址屏蔽。屏蔽模式和地址的组合提供不同的应答地址范围。

当两种屏蔽模式功能几乎以相同的方式工作时，它们使用地址屏蔽的方式是不同的。

19.4.3.3 5 位地址屏蔽模式

从其名称中可以看出，5 位地址屏蔽模式使用最多 5 位的地址屏蔽功能来创建应答的地址范围，它使用的是输入地址的 1 至 5 位。这允许该模块在使用 7 位地址时最多应答 31 个地址，或在使用 10 位地址时最多应答 63 个地址（见例 19-2）。在设定 MSSPMSK 配置位（0）时选择此屏蔽模式。

此模式下的地址屏蔽位包含在 SSPxCON 寄存器中，该位可阻止 SSPxCON 寄存器作为 I^2C 从动模式下的控制寄存器使用（寄存器 19-6）。在 7 位屏蔽模式下，地址屏蔽位 ADMSK<5:1>（SSPxCON2<5:1>）将屏蔽 SSPxADD 寄存器中的相应地址位。对于任何被置 1 的 ADMSK 位（ADMSK<n> = 1），将忽略相应的地址位（SSPxADD<n> = x）。对于发出地址应答的模块，仅在地址不具备有效地址屏蔽时才匹配。

在 10 位地址屏蔽模式下， ADMSK<5:2> 位将屏蔽 SSPxADD 寄存器中的相应地址位。此外，ADMSK1 同时还屏蔽该地址的低 2 位（SSPxADD<1:0>）。对于任何有效的 ADMSK 位（ADMSK<n> = 1），将忽略相应的地址位（SSPxADD<n> = x）。另外还应注意，尽管在10位地址屏蔽模式下，高地址位重用了 SSPxADD 寄存器的部分位，但是地址屏蔽位不得与这部分位交互，而只影响低地址位。

注 **1**：ADMSK1 屏蔽该地址的低 2 位。
2：该地址的高 2 位不受地址屏蔽的影响。

例 19-2： 5 位屏蔽模式下的地址屏蔽示例

7 位寻址模式：

SSPADD<7:1>= A0h (1010000) （SSPADD<0> 假定为 0）

ADMSK<5:1> = 00111

被应答的地址：A0h、 A2h、 A4h、 A6h、 A8h、 AAh、 ACh 和 AEh

10 位寻址模式：

SSPADD<7:0> = A0h (10100000) （本例忽略了该地址的高 2 位，因为它们不受地址屏蔽的影响）

ADMSK<5:1> = 00111

被应答的地址：A0h、A1h、A2h、A3h、A4h、A5h、A6h、A7h、A8h、A9h、AAh、ABh、ACh、ADh、AEh 和 AFh

19.4.3.4 7 位地址屏蔽模式

与 5 位地址屏蔽不同，7 位地址屏蔽模式最多可屏蔽 8 位（在 10 位地址中）来定义可被应答的地址范围，且屏蔽的是输入地址的低 8 位。这允许该模块在使用 7 位地址时最多应答 127 个不同的地址，或在使用 10 位地址时最多应答 255 个地址（见例 19-3）。此模式是该模块的默认配置，并且在未编程 MSSPMSK（1）时进行选择。

7 位地址屏蔽模式的地址屏蔽位包含在 SSPxMSK 寄存器中，而不是在 SSPxCON2 寄存器中。SSPxMSK 是该模块内的一个独立的硬件寄存器，但是不可直接寻址。另外，它与 SSPxADD 寄存器共享 SFR 空间中的一个地址。要访问 SSPxMSK 寄存器，必须选择 MSSP 模式1001（SSPCON1<3:0> = 1001），然后对SSPxADD 单元进行读写操作。

要使用 7 位地址屏蔽模式，必须在选择 I^2C 从动地址模式之前用一个值来初始化 SSPxMSK。因此，执行操作的序列为：

1. 选择 SSPxADD 访问模式（SSPxCON2<3:0> = 1001）
2. 将屏蔽值写入相应的 SSPADD 寄存器地址（对于 MSSP1 为 FC8h，对于 MSSP2 为 F6Eh）。
3. 设置适合的 I^2C 从动模式（对于 10 位寻址，SSPxCON2<3:0> = 0111；对于 7 位寻址，SSPxCON2<3:0> = 0110）。

将 SSPxMSK 中的屏蔽位置 1 或清零，会让 5 位地址屏蔽模式下的 ADMSK 位的工作方式完全相反。也就是说，将 SSPxMSK 中的某个位清零会导致屏蔽相应的地址位，而将该位置 1 要求在该位置出现匹配。出现任何复位条件时 SSPxMSK 将被复位为全 1，因此，在向 SSPxMSK 写入屏蔽值之前，对标准 MSSP 操作无任何影响。

对于 7 位寻址模式，SSPxMSK<7:1> 位将屏蔽 SSPxADD 寄存器中的相应地址位。对于任何处于有效状态的 SSPxMSK 位（SSPxMSK<n> = 0），将忽略相应的 SSPxADD 地址位（SSPxADD<n> = x）。对于发出地址应答的模块，仅在地址不具备有效地址屏蔽时才匹配。

对于 10 位寻址模式，SSPxMSK<7:0> 位将屏蔽 SSPxADD 寄存器中的相应地址位。对于任何处于有效的 SSPxMSK 位（= 0），将忽略相应的 SSPxADD 地址位（SSPxADD<n> = x）。

注：	该地址的高 2 位不受地址屏蔽的影响。

例 19-3： 7 位屏蔽模式下的地址屏蔽示例

7 位寻址模式：

SSPxADD<7:1> = 1010 000

SSPxMSK<7:1> = 1111 001

被应答的地址 = A8h、 A6h、 A4h 和 A0h

10 位寻址模式：

SSPxADD<7:0> = 1010 0000（本例忽略了高 2 位，因为它们不受任何影响）

SSPxMSK<5:1> = 1111 0

被应答的地址 = A8h、 A6h、 A4h 和 A0h

© 2007 Microchip Technology Inc.

19.4.3.5 接收

当地址字节的R/W位清零且地址匹配时，SSPxSTAT寄存器的R/W位清零。接收的地址被装入SSPxBUF寄存器，且SDAx数据线保持低电平（ACK）。

当存在地址字节溢出条件时，则不会产生应答脉冲（ACK）。溢出条件是指BF位（SSPSTAT<0>）置1，或者SSPOV位（SSPCON1<6>）置1。

每个数据传输字节都会产生一个MSSP中断。必须用软件将中断标志位SSPxIF清零。使用SSPxSTAT寄存器可以确定该字节的状态。

如果SEN被使能（SSPxCON2<0> = 1），SCLx将在每个数据传输之后保持为低电平（低电平时钟延长）。必须通过将CKP位（SSPxCON1<4>）置1才能释放时钟。详细情况，请参见**第 19.4.4 节 “时钟延长”**。

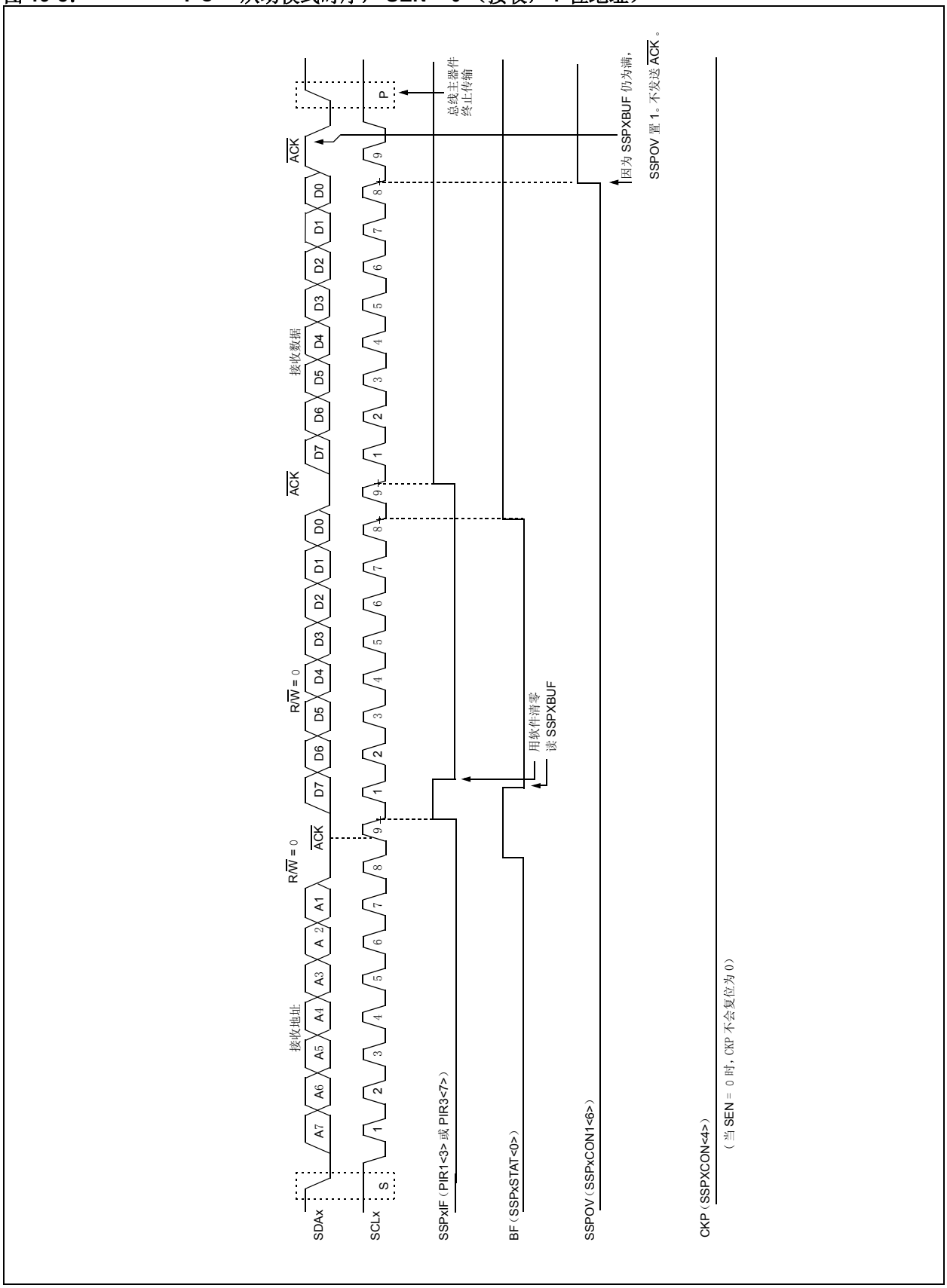
19.4.3.6 发送

当接收的地址字节的R/W位置1且地址匹配时，SSPxSTAT寄存器的R/W位置1。接收到的地址被装入SSPxBUF寄存器。ACK脉冲在第9位上发送，同时无论SEN的值为何，SCLx引脚保持低电平（如需了解更多详细信息，请参见**第 19.4.4 节 “时钟延长”**）。通过延长时钟，主器件在从器件准备好发送数据时，才能发送另一个时钟脉冲。发送的数据必须被装入SSPBUF寄存器，同时也被装入SSPSR寄存器。然后，通过将CKP位（SSPxCON1<4>）置1来使能SCLx引脚。8个数据位在SCL时钟输入的下降沿被移出。这可确保在SCLx为高电平期间SDAx信号是有效的（如图19-10）。

主接收器的ACK脉冲将在SCLx输入第9个脉冲的上升沿被锁存。如果SDAx线为高电平（无ACK），那么表示数据传输已完成。在这种情况下，如果从器件锁存了ACK，将复位从动逻辑（复位SSPxSTAT寄存器），同时从器件监视下一个启动位的出现。如果SDAx线为低电平（ACK），则必须将下一个要发送的数据装入SSPxBUF寄存器。同样，必须通过将CKP位置1来使能SCLx引脚。

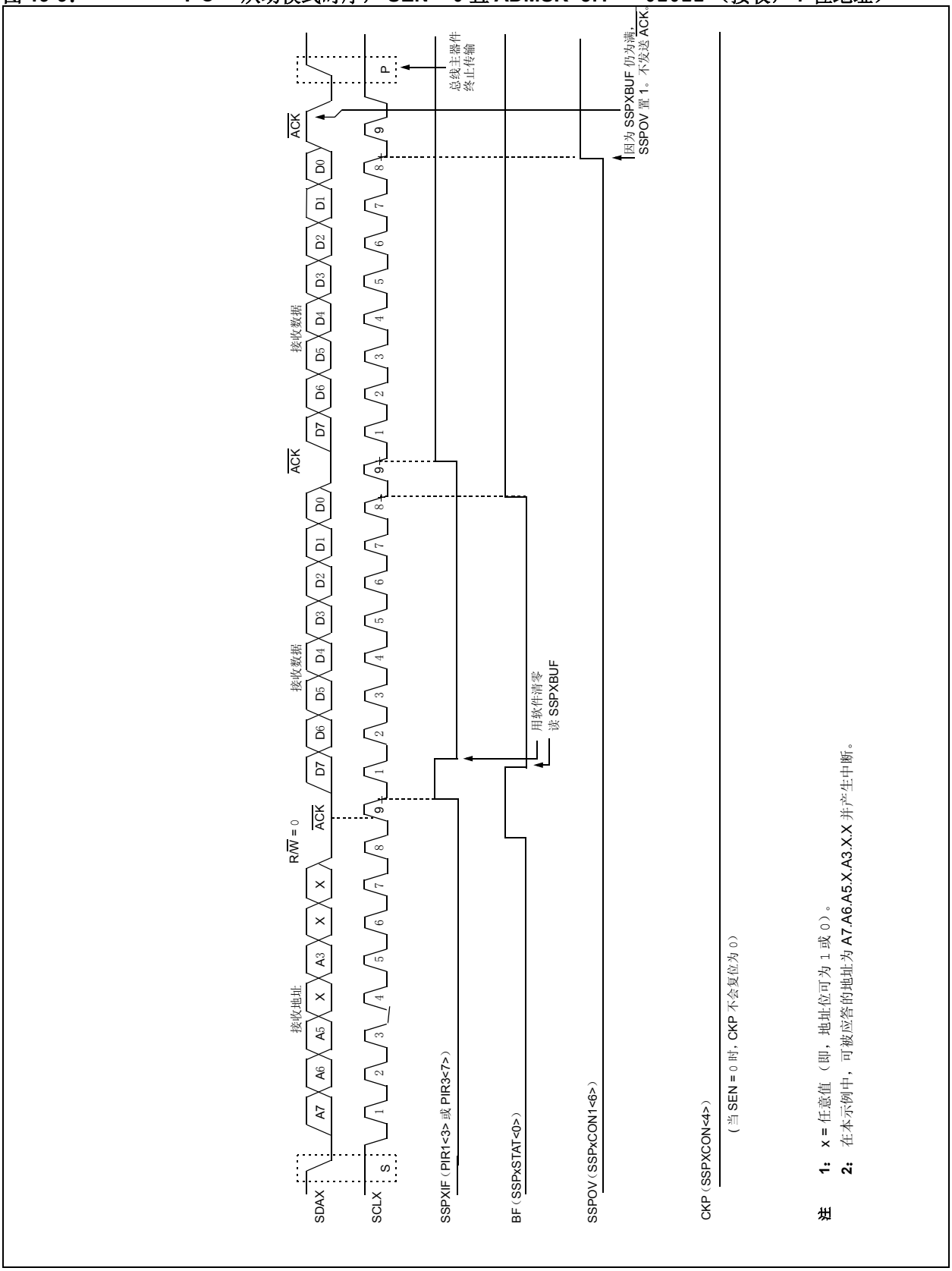
每个数据传输字节都会产生一个MSSP中断。SSPxIF位必须用软件清零，SSPxSTAT状态寄存器用于确定字节的状态。SSPxIF位在第9个时钟脉冲的下降沿被置1。

图 19-8： I^2C™ 从动模式时序， SEN = 0 （接收， 7 位地址）



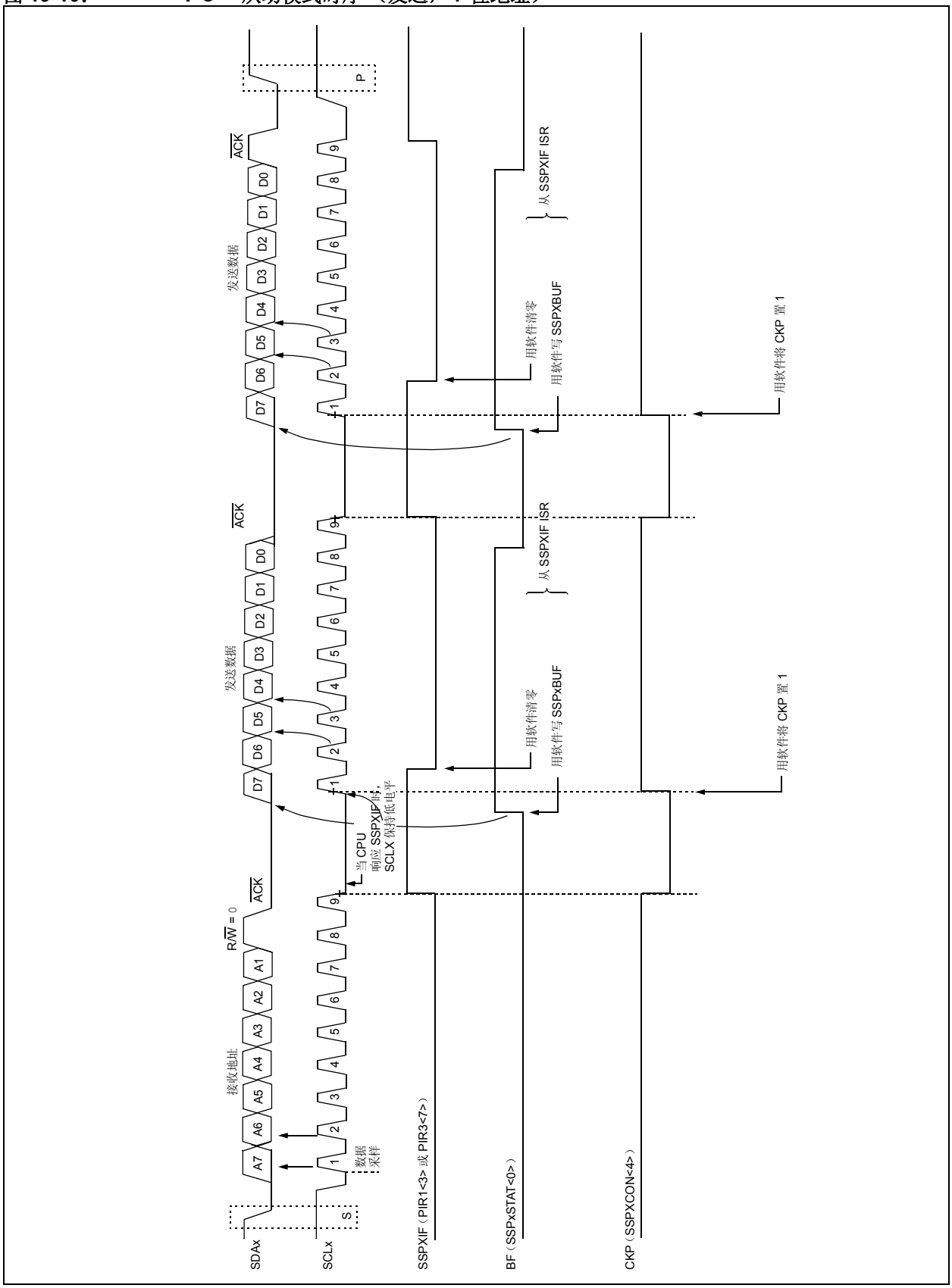
© 2007 Microchip Technology Inc.

图 19-9： I^2C™ 从动模式时序， SEN = 0 且 ADMSK<5:1> = 01011 （接收， 7 位地址）



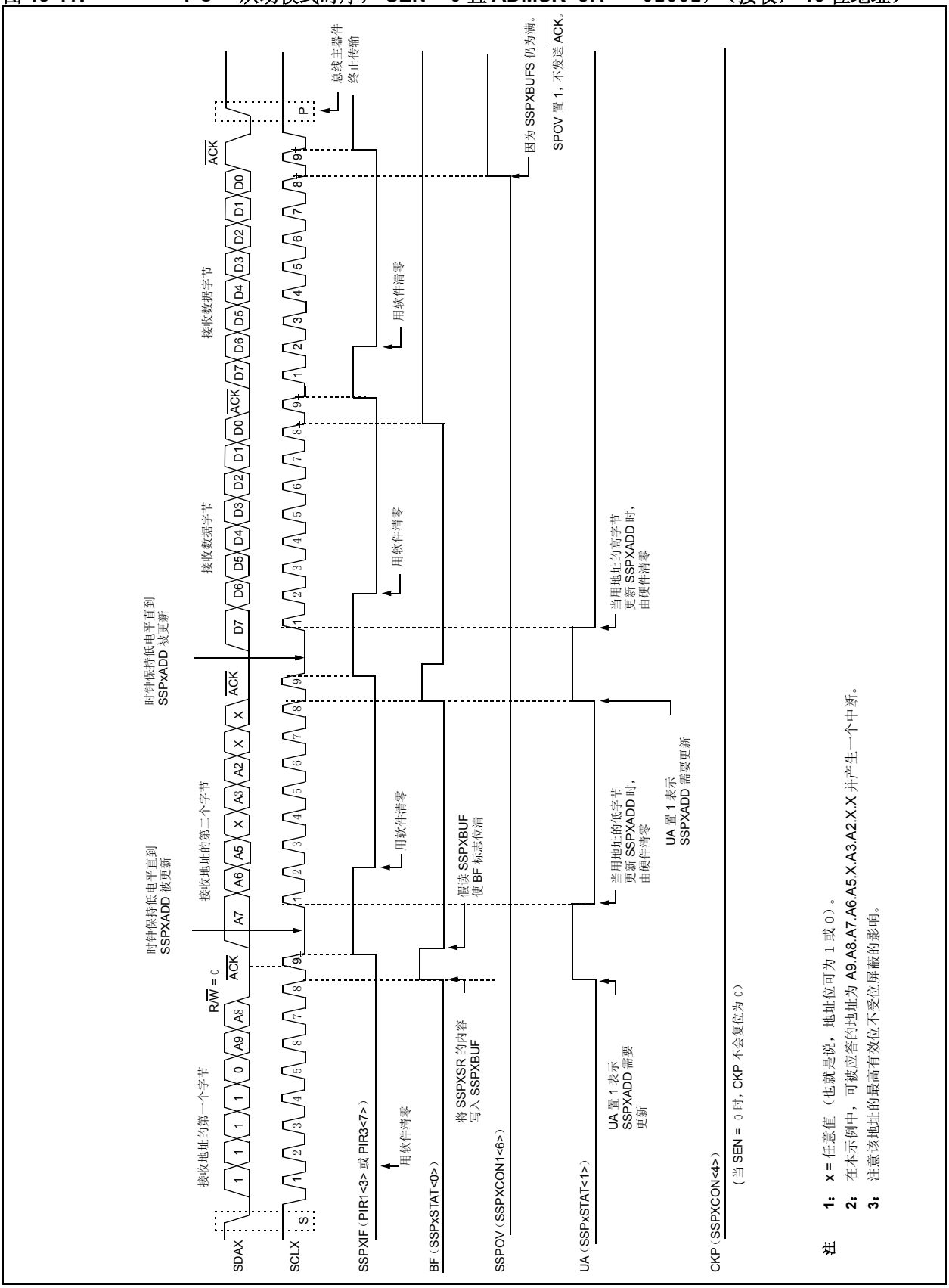
注： 1： x = 任意值 （即，地址位可为 1 或 0）。
2： 在本示例中，可被应答的地址为 A7.A6.A5.X.A3.X.X 并产生中断。

图 19-10： I^2C™ 从动模式时序 （发送， 7 位地址）



 © 2007 Microchip Technology Inc.

图 19-11：　　I²C™ 从动模式时序， SEN = 0 且 ADMSK<5:1> = 01001，（接收， 10 位地址）

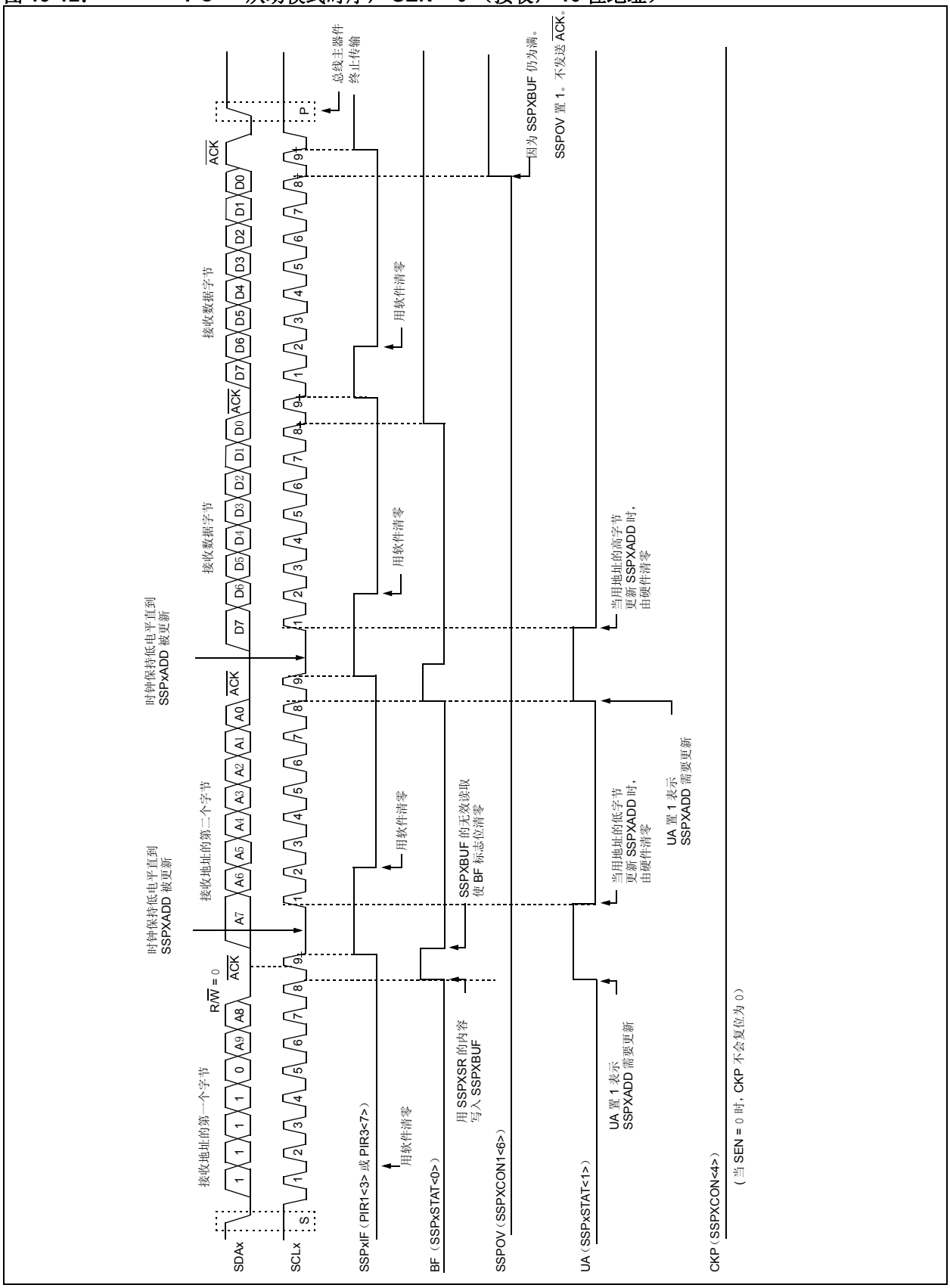


注　1： x = 任意值（也就是说，地址位可为 1 或 0）。

2： 在本示例中，可被应答的地址为 A9.A8.A7.A6.A5.X.A3.A2.X.X 并产生一个中断。

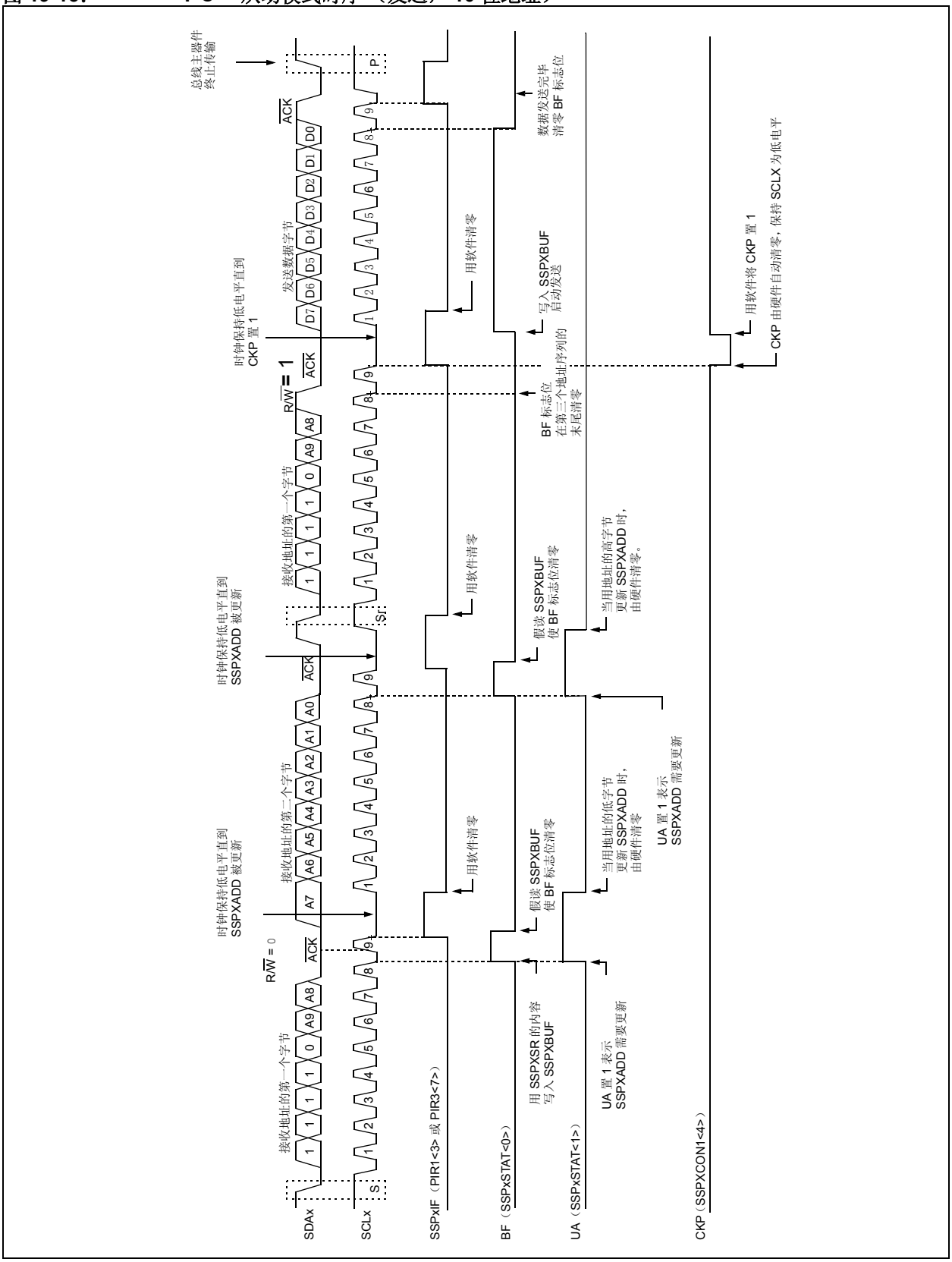
3： 注意该地址的最高有效位不受位屏蔽的影响。

图 19-12： I^2C™ 从动模式时序， SEN = 0 （接收， 10 位地址）



 © 2007 Microchip Technology Inc.

图 19-13：　　I²C™ 从动模式时序 （发送， 10 位地址）



19.4.4 时钟延长

7位和10位从动模式都在发送序列中实现了自动时钟延长。

SEN 位（SSPxCON2<0>）允许在接收过程中使能时钟延长。将 SEN 置 1 会导致在每个数据接收序列的末尾将 SCLx 引脚保持在低电平。

19.4.4.1 7 位从动接收模式的时钟延长（SEN = 1）

在 7 位从动接收模式下，如果在 $\overline{ACK}$ 序列末尾的第 9 个时钟的下降沿 BF 位被置 1，则 SSPxCON1 寄存器中的 CKP 位就会自动清零，强制 SCLx 输出保持低电平。CKP 位被清零会将 SCLx 线拉为低电平。在允许继续接收之前，必须在用户的中断服务程序中将 CKP 位置 1。保持 SCLx 线为低电平期间，用户可以在主器件发起另一个接收序列之前，有时间响应 ISR 并读取 SSPxBUF 的内容。这将防止发生缓冲器溢出（见图 19-15）。

注 1： 如果用户在第 9 个时钟的下降沿前读取了 SSPxBUF 的内容，使得 BF 位被清零，CKP 位就不会被清零，也不会产生时钟延长。

2： 无论 BF 位的状态为何，CKP 位都可以用软件置 1。在下一个接收序列开始之前，用户在 ISR 中清零 BF 位时应该小心，以避免溢出。

19.4.4.2 10 位从动接收模式的时钟延长（SEN = 1）

在 10 位从动接收模式下，在地址序列中会自动发生时钟延长，但是 CKP 位不会被清零。在这期间，如果 UA 位在第 9 个时钟之后被置 1，就会启动时钟延长。UA 位在接收到 10 位地址的高字节后被置 1，然后接收 10 位地址的第二个字节并清零 R/$\overline{W}$ 位。在更新 SSPxADD 时释放时钟线。如 7 位模式中描述的那样，在每个数据接收序列中都会发生时钟延长。

注： 如果用户在第9个时钟的下降沿出现之前查询 UA 位，并通过更新 SSPxADD 寄存器清零 UA 位，而且在此之前用户没有读取 SSPxBUF 寄存器使 BF 位清零，则 CKP 位的电平仍然不会被拉低。基于 BF 位状态的时钟延长仅在数据序列中出现，不会出现在地址序列中。

19.4.4.3 7 位从动发送模式的时钟延长

如果 BF 位被清零，7 位从动发送模式将通过在第 9 个时钟的下降沿出现后清零 CKP 位，来实现时钟延长。无论 SEN 位的状态为何，这种情况都会发生。

用户的中断服务程序必须先将 CKP 位置 1 才可以继续发送。保持 SCLx 线为低电平期间，用户在主器件发起另一个发送序列之前，将有时间响应 ISR 并装入 SSPxBUF 的内容（见图 19-10）。

注 1： 如果用户在第 9 个时钟的下降沿之前就装入 SSPxBUF 的内容，CKP 位就不会被清零，也不会发生时钟延长。

2： 无论 BF 位的状态为何，CKP 位都可以用软件置 1。

19.4.4.4 10 位从动发送模式的时钟延长

在 10 位从动发送模式下，在前两个地址序列中由 UA 位的状态来控制时钟延长，正如同 10 位从动接收模式一样。前两个地址后跟着第三个地址序列，该地址序列包含 10 位地址的高位和被置 1 的 R/$\overline{W}$ 位。在执行完第三个地址序列后，UA 位不置 1，此时模块配置为发送模式，BF 标志位控制时钟延长，正如 7 位从动发送模式一样（见图 19-13）。

© 2007 Microchip Technology Inc.

19.4.4.5 时钟同步和 CKP 位

当 CKP 位清零时，SCLx 输出被强制为 0。然而，清零 CKP 位不会将 SCLx 输出拉为低电平，除非已经采样到 SCLx 输出为低电平。因此，CKP 位将不会把 SCLx 线拉为低电平，直到外部 I^2C 主器件已将 SCLx 线拉低。SCLx 输出将保持低电平，直到 CKP 位置 1 且 I^2C 总线上的所有其他器件将 SCLx 电平拉高为止。这可以确保对 CKP 位的写操作不会违反 SCLx 的最小高电平时间要求（见图 19-14）。

图 19-14：　　时钟同步时序

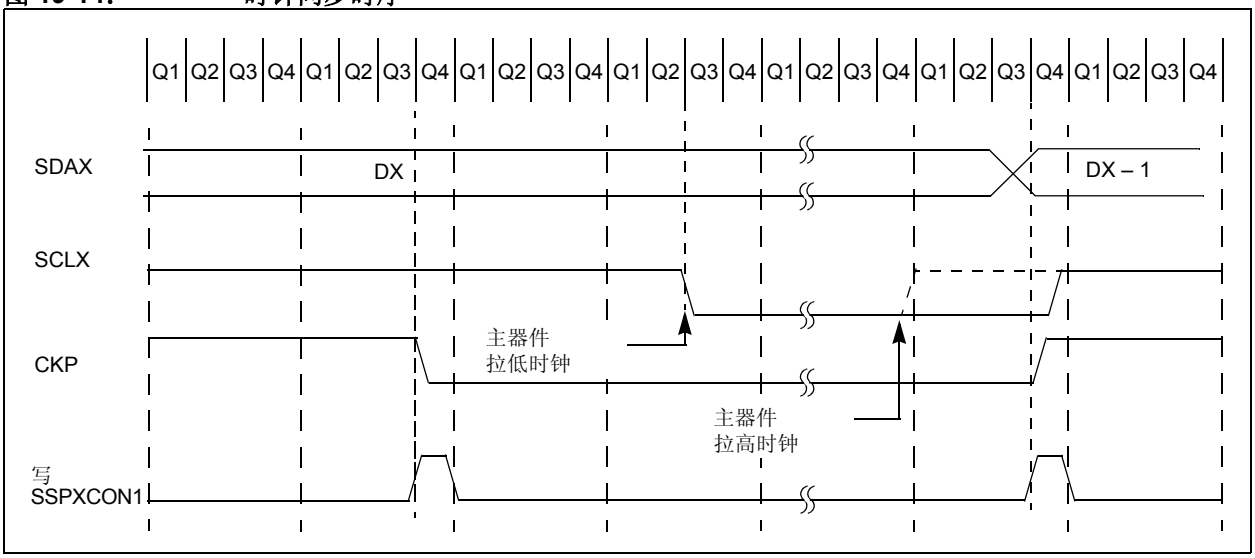
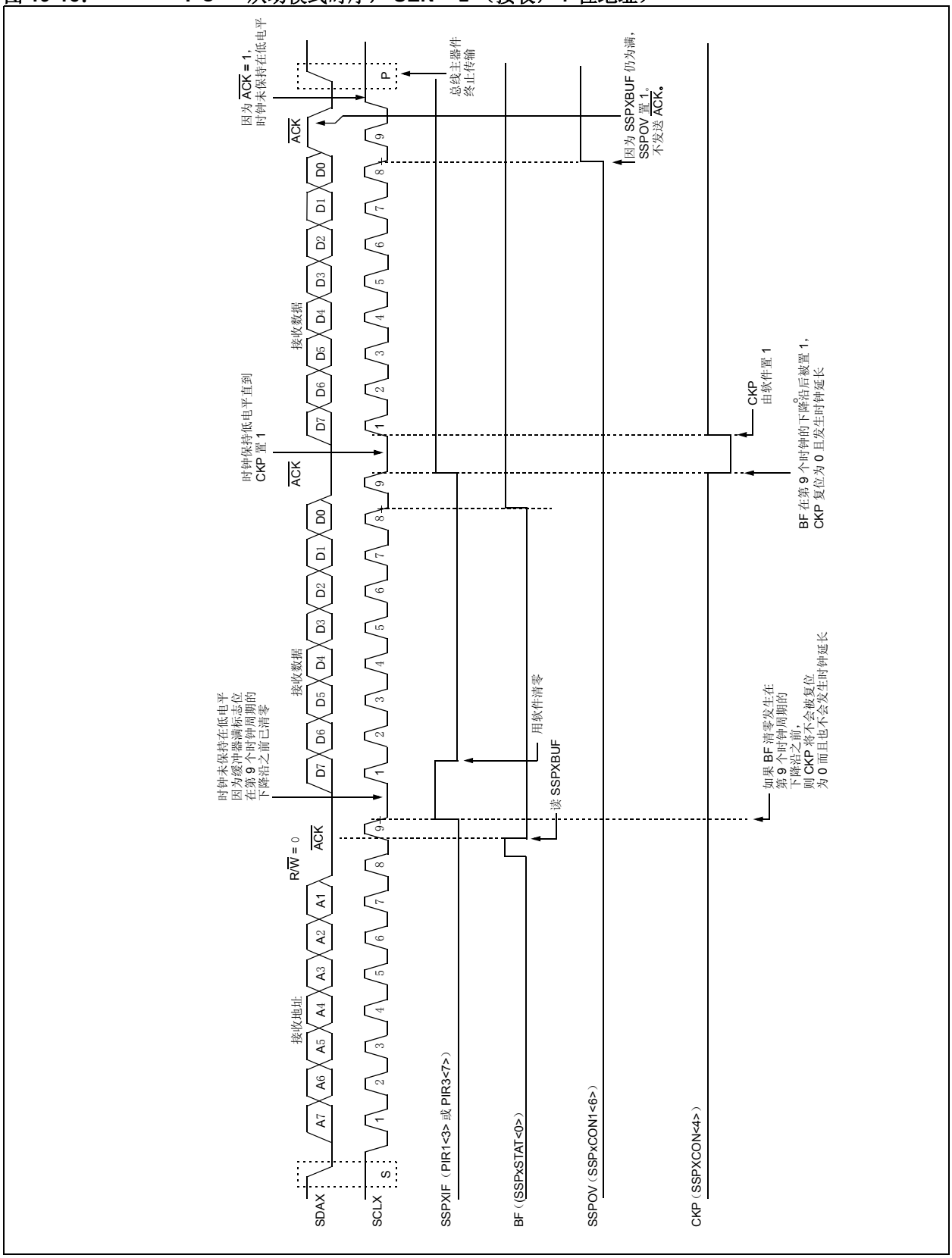
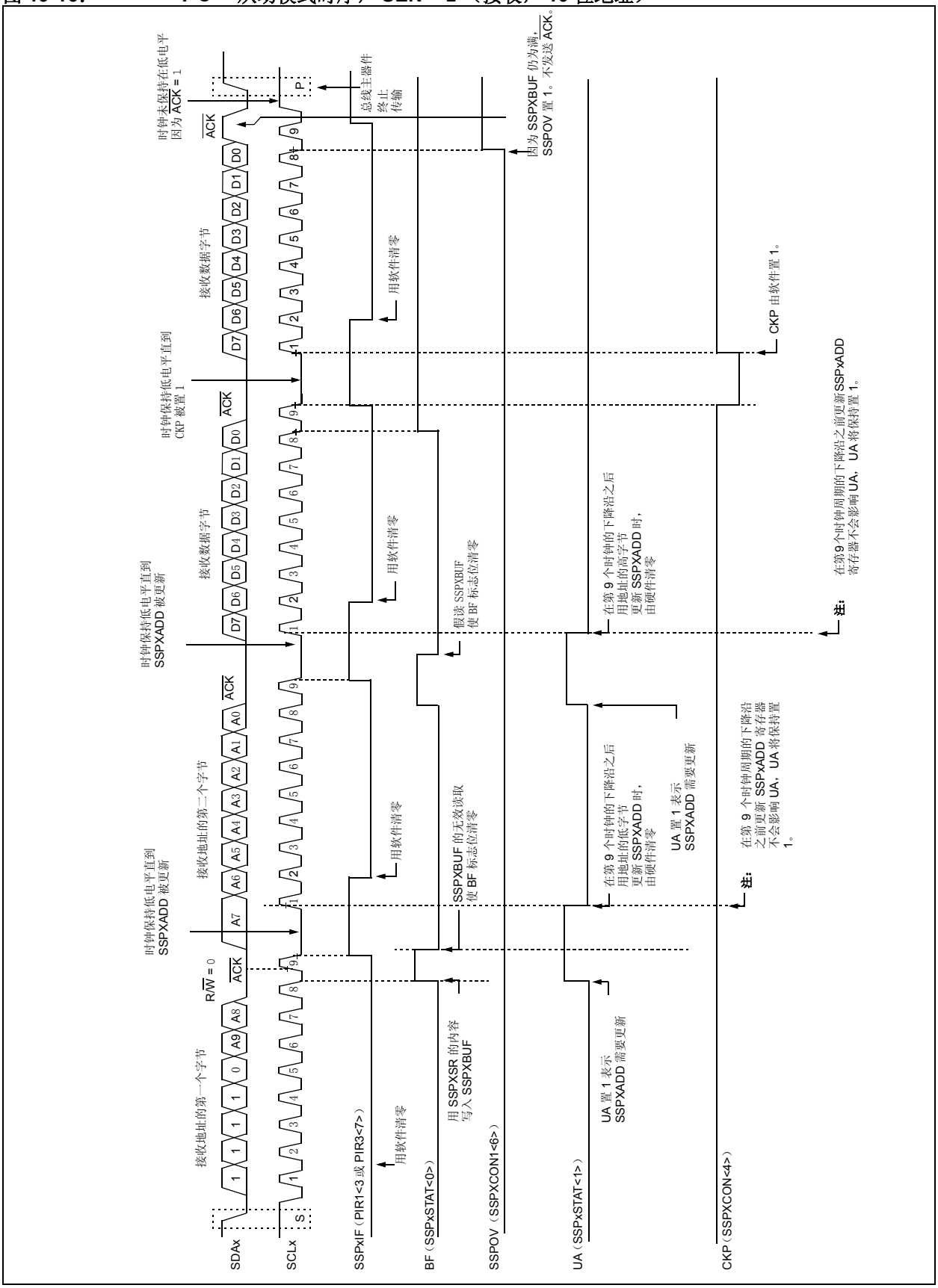


图 19-15： I^2C™ 从动模式时序， SEN = 1 （接收， 7 位地址）



 © 2007 Microchip Technology Inc.

图 19-16： I^2C™ 从动模式时序， SEN = 1 （接收， 10 位地址）



19.4.5 支持广播呼叫地址

在 I²C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用此地址时，理论上所有的器件都应该发送一个应答信号来响应。

广播呼叫地址是由 I²C 协议为特定目的而保留的 8 个地址之一。它由全 0 组成，且 R/W = 0。

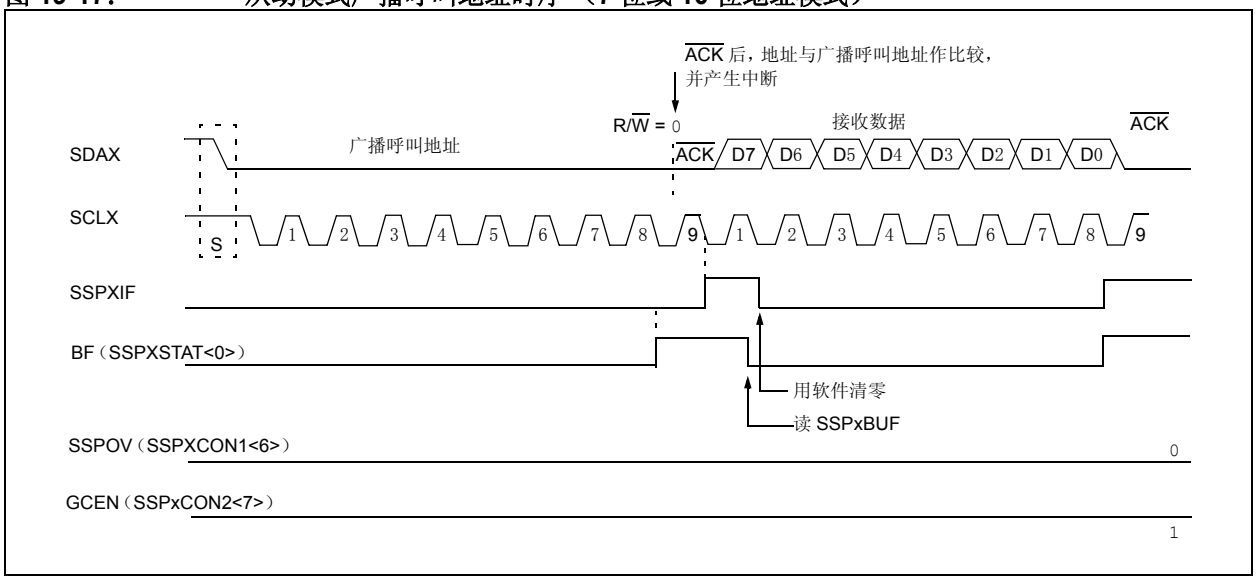
当使能广播呼叫使能位 GCEN（SSPxCON2<7> 置 1）时，即可识别广播呼叫地址。检测到启动位后， 8 位数据会被移入 SSPxSR，同时将该地址与 SSPxADD 进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

如果与广播呼叫地址匹配， SSPxSR 的值将被传输到 SSPxBUF， BF 标志位（第 8 位）置 1，并且 SSPxIF 中断标志位在第 9 位 （ACK 位）的下降沿置 1。

当响应中断时，可以通过读取 SSPxBUF 的内容来检查中断源。该值可以用于判断是特定器件的地址还是一个广播呼叫地址。

在 10 位寻址模式下，需要更新 SSPxADD 来匹配地址的后半部分，同时 UA 位 （SSPxSTAT<1>）置 1。如果 GCEN 位置 1 时采样到广播呼叫地址，同时从器件被配置为 10 位寻址模式，则不再需要地址的后半部分，也不会将 UA 位置 1，从器件将在应答后开始接收数据（见图 19-17）。

图 19-17： 从动模式广播呼叫地址时序 （7 位或 10 位地址模式）



© 2007 Microchip Technology Inc.

19.4.6 主控模式

通过将 SSPxCON1 中的相应 SSPM 位置 1 和清零，同时将 SSPEN 位置 1，可以使能主控模式。在主控模式下，如果 TRIS 位置 1，则 SCLx 和 SDAx 信号线由 MSSP 硬件控制。

主控模式通过在检测到启动和停止条件时产生中断来工作。停止（P）位和启动（S）位在复位时或禁止 MSSP 模块时清零。当 P 位置 1 时，可以取得 I^2C 总线的控制权；否则总线处于空闲状态，且 P 位和 S 位都为零。

在固件控制的主控模式下，用户代码根据启动和停止位条件执行所有的 I^2C 总线操作。

一旦使能主控模式，用户即可选择以下 6 项操作：

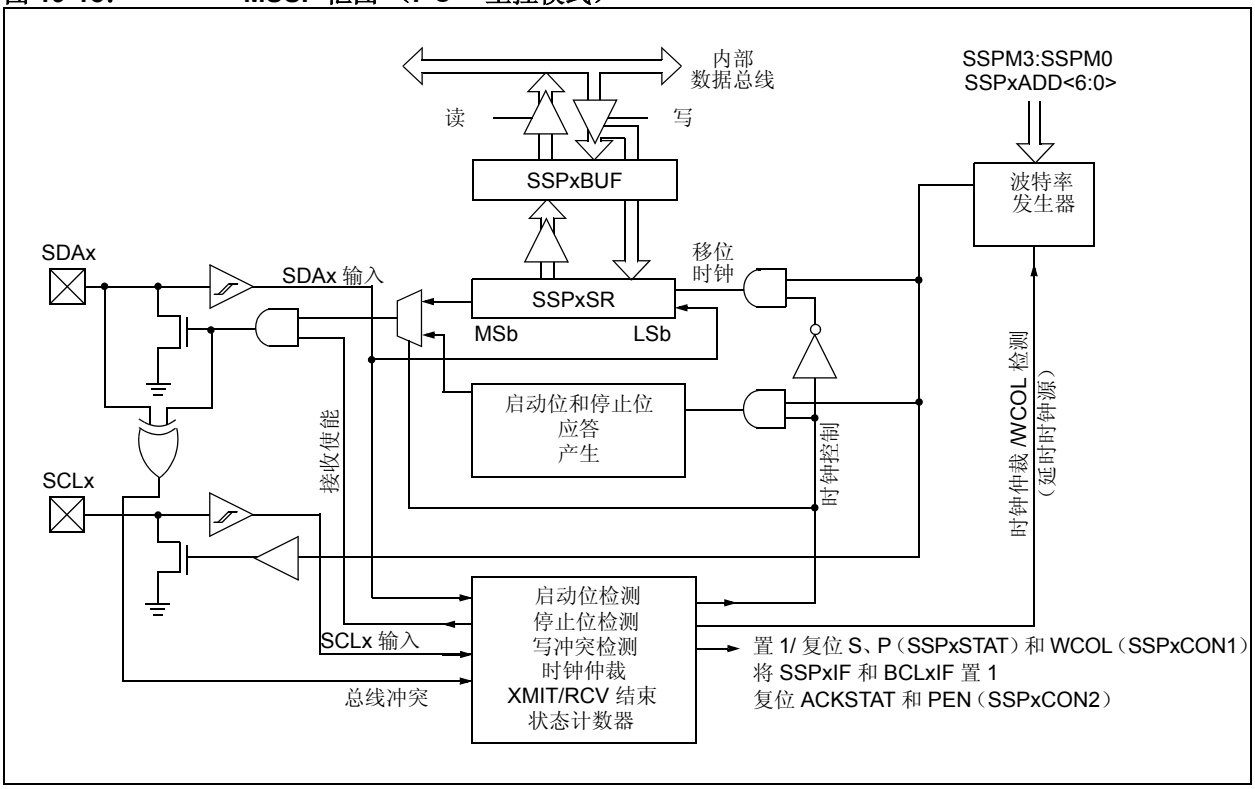
1. 在 SDAx 和 SCLx 上产生一个启动条件。
2. 在 SDAx 和 SCLx 上产生一个重复启动条件。
3. 写入 SSPxBUF 寄存器，启动数据 / 地址的发送。
4. 配置 I^2C 端口接收数据。
5. 在接收到数据字节后产生应答信号。
6. 在 SDAx 和 SCLx 上产生停止条件。

注：	当被配置为 I^2C 主控模式时，MSSP 模块不允许事件排队。例如，在启动条件结束前，不允许用户立即写SSPxBUF寄存器以启动传输。这种情况下，将不会写 SSPxBUF，WCOL 位将被置 1，这表明没有发生对 SSPxBUF 的写操作。

下列事件会使 SSP 中断标志位 SSPxIF 置 1（如果允许 SSP 中断，则产生中断）：

- 启动信号
- 停止信号
- 数据传输字节发送 / 接收
- 应答发送
- 重复启动

图 19-18： MSSP 框图 （I^2C™ 主控模式）



19.4.6.1 I²C 主控模式工作原理

主器件产生所有串行时钟脉冲和启动 / 停止条件，以停止条件或重复启动条件结束传输过程。因为重复启动条件也是下一次串行传输的开始，因此不会释放I²C总线。

在主控发送器模式下，串行数据通过 SDAx 输出，而串行时钟由 SCLx 输出。发送的第一个字节包括接收器件的从器件地址（7 位）和读 / 写（R/W）位。在这种情况下，R/W 位将为逻辑 0。一次发送 8 位串行数据。每发送一个字节，会收到一个应答位。输出启动和停止条件，表明串行传输的开始和结束。

在主控接收模式下，发送的第一个字节包括发送器件的从器件地址（7 位）和 R/W 位。在这种情况下，R/W 位将为逻辑 1。因此，发送的第一个字节是一个 7 位从器件地址，后面跟 1 表示接收。串行数据通过 SDAx 接收，而串行时钟由 SCLx 输出。每次接收 8 位串行数据。每接收到一个字节，都会发送一个应答位。启动和停止信号分别表明发送的开始和结束。

在 I²C 模式下，在 SPI 模式工作中使用的波特率发生器被用于将 SCLx 时钟频率设置为 100 kHz、400 kHz 或 1 MHz。更多详细信息，请参见**第 19.4.7节“波特率”**。

下面是一个典型的发送顺序：

1. 用户通过将启动使能位 SEN（SSPxCON2<0>）置 1，产生启动条件。
2. SSPxIF 置 1。在进行下一步操作前， MSSP 模块将等待所需的启动时间。
3. 用户将从器件地址装入 SSPxBUF 进行发送。
4. 地址从 SDAx 引脚移出，直到发送完所有 8 位地址数据。
5. MSSP 模块移入来自从器件的 ACK 位，并将它的值写入 SSPxCON2 寄存器 （SSPxCON2<6>）。
6. MSSP 模块在第 9 个时钟周期的末尾将 SSPxIF 位置 1，产生一个中断。
7. 用户将 8 位数据装入 SSPxBUF。
8. 数据从 SDAx 引脚移出，直到发送完所有 8 位为止。
9. MSSP 模块移入来自从器件的 ACK 位，并将它的值写入 SSPxCON2 寄存器 （SSPxCON2<6>）。
10. MSSP 模块在第 9 个时钟周期的末尾将 SSPxIF 位置 1，产生一个中断。
11. 用户通过将停止使能位 PEN（SSPxCON2<2>）置 1，产生停止条件。
12. 一旦停止条件完成，将产生一个中断。

© 2007 Microchip Technology Inc.

19.4.7 波特率

在 I²C 主控模式下，波特率发生器（Baud Rate Generator，BRG）的重载值位于SSPxADD寄存器的低 7 位（图 19-19）。当发生对 SSPxBUF 的写操作时，波特率发生器将自动开始计数。BRG 会递减计数至 0，然后停止，直到再次发生重载。BRG 计数会在每个指令周期（TCY）中的 Q2 和 Q4 时钟周期上进行两次减计数。在 I²C 主控模式下，会自动重载 BRG。

若完成了指定操作（即，在发送最后一个数据位后发送 ACK），内部时钟将自动停止计数，SCLx 引脚将保持在其最后的状态。

表 19-3 显示了不同的指令周期下的时钟频率以及装入 SSPxADD 的 BRG 值。

19.4.7.1 波特率和模块的相互关系

因为MSSP1和MSSP2是独立的模块，所以它们能以不同的波特率同时在 I²C 主控模式下工作。这是通过为每个模块使用不同的 BRG 重载值实现的。

由于此模式的基本时钟源来自系统时钟，对系统时钟的任何更改都会同等程度地影响这两个模块。通过更改BRG重载值可以将一个或两个波特率改回到前一个值。

图 19-19： 波特率发生器框图

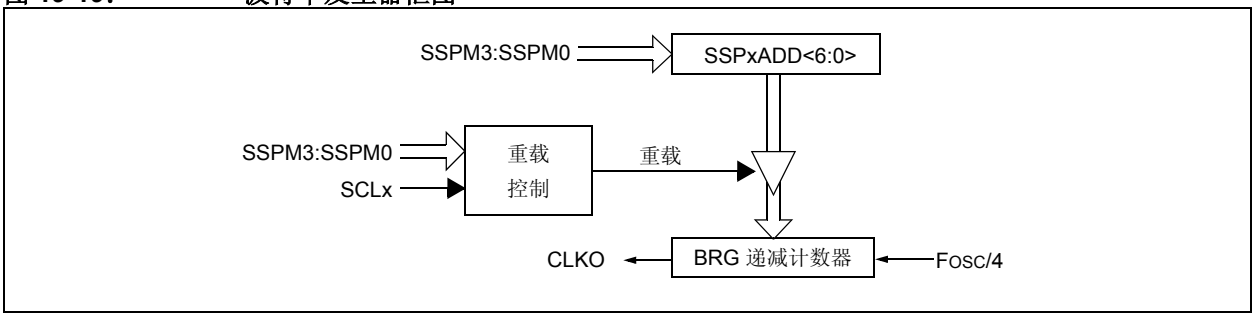


表 19-3： 带 BRG 的 I²C 时钟速率

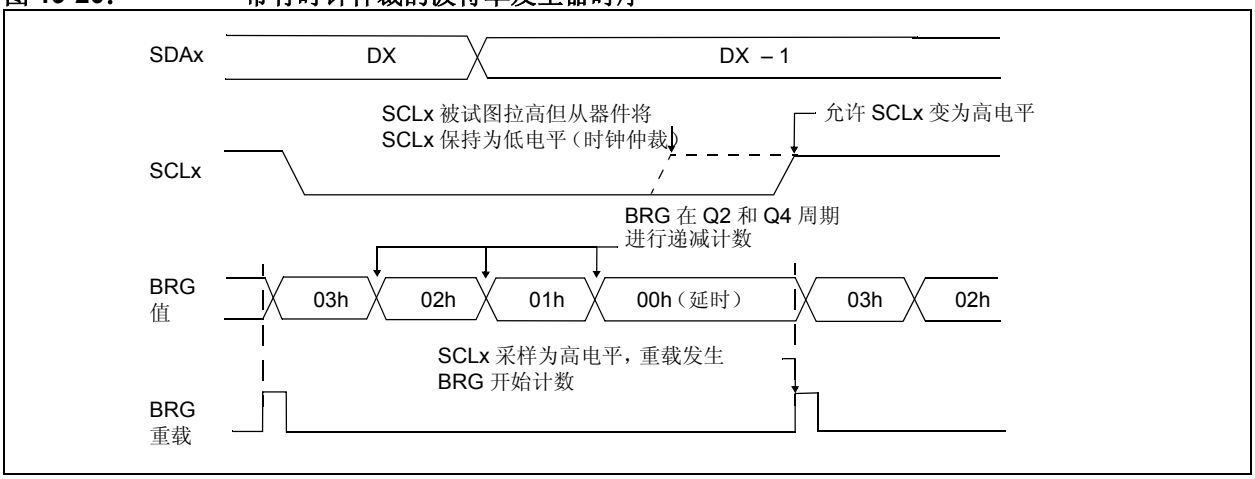
FOSC	FCY	FCY * 2	BRG 值	FSCL（两次 BRG 计满返回）
40 MHz	10 MHz	20 MHz	18h	400 kHz(1)
40 MHz	10 MHz	20 MHz	1Fh	312.5 kHz
40 MHz	10 MHz	20 MHz	63h	100 kHz
16 MHz	4 MHz	8 MHz	09h	400 kHz(1)
16 MHz	4 MHz	8 MHz	0Ch	308 kHz
16 MHz	4 MHz	8 MHz	27h	100 kHz
4 MHz	1 MHz	2 MHz	02h	333 kHz(1)
4 MHz	1 MHz	2 MHz	09h	100 kHz
4 MHz	1 MHz	2 MHz	00h	1 MHz(1)

注 1： 尽管 I²C 接口并不完全符合 400 kHz I²C 规范（该规范适用于大于 100 kHz 的频率），但在需要较高频率的应用场合可以谨慎使用。

19.4.7.2 时钟仲裁

如果在任何接收、发送或重复启动 / 停止条件过程中，主器件不再控制 SCLx 引脚（允许 SCLx 引脚悬空为高电平），就会发生时钟仲裁。如果允许 SCLx 引脚悬空为高电平，波特率发生器（BRG）将暂停计数直到实际采样到 SCLx 引脚为高电平为止。当 SCLx 引脚采样为高电平时，会将 SSPxADD<6:0> 的内容重新装入波特率发生器并使之开始计数。这可以保证当外部器件将时钟拉低时，SCLx 始终在至少一个 BRG 计满返回周期内保持高电平 （图 19-20）。

图 19-20： 带有时钟仲裁的波特率发生器时序



 © 2007 Microchip Technology Inc.

19.4.8 I²C 主控模式启动信号时序

要产生启动条件，用户应将启动使能位 SEN（SSPxCON2<0>）置 1。当 SDAx 和 SCLx 引脚都采样为高电平时，波特率发生器重新装入 SSPxADD<6:0> 的内容并开始计数。当波特率发生器发生超时 （TBRG）时，如果 SCLx 和 SDAx 都采样为高电平时，则 SDAx 引脚被驱动为低电平。当 SCLx 为高电平时，将 SDAx 驱动为低电平即为启动条件，并使 S 位（SSPxSTAT<3>）置 1。随后波特率发生器重新装入 SSPxADD<6:0> 的内容并恢复计数。当波特率发生器再次发生超时 （TBRG）时， SEN 位（SSPxCON2<0>）将自动由硬件清零。波特率发生器暂停工作，SDAx 线保持低电平，启动条件完成。

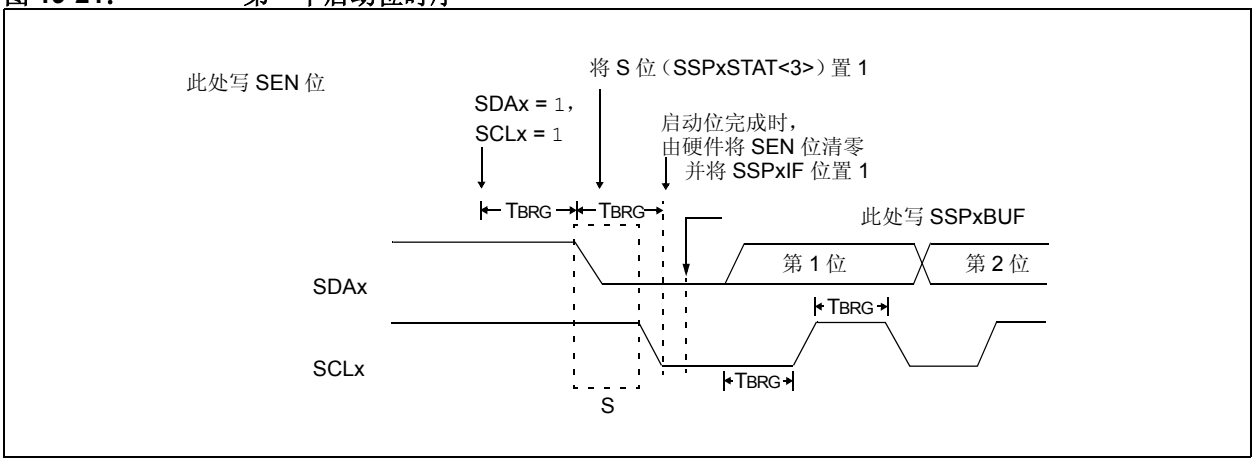
注： 如果在启动条件开始时，SDAx 和 SCLx 引脚已经被采样为低电平，或者在启动条件期间，SCLx 在 SDAx 线被驱动为低电平之前已经采样为低电平，则将产生总线冲突，总线冲突中断标志位 BCLxIF 置 1，启动条件中止， I²C 模块复位到空闲状态。

19.4.8.1 WCOL 状态标志位

在启动时序进行中时，如果用户写SSPxBUF，则WCOL 位被置 1，同时缓冲器内容不变 （写操作无效）。

注： 由于不允许事件排队，在启动条件完成之前，不能对 SSPxCON2 的低 5 位进行写操作。

图 19-21： 第一个启动位时序



19.4.9 I²C 主控模式重复启动条件时序

将 RSEN 位（SSPxCON2<1>）编程为高电平，并且 I²C逻辑模块处于空闲状态时，就会产生重复启动条件。当 RSEN 位置 1 时，SCLx 引脚被拉为低电平。当 SCLx 引脚采样为低电平时，波特率发生器装入 SSPxADD<5:0> 的内容，并开始计数。在一个波特率发生器计数周期（TBRG）内 SDAx 引脚被释放（其引脚电平被拉高）。当波特率发生器超时时，如果 SDAx 被采样为高电平， SCLx 引脚将被拉高。当 SCLx 被采样为高电平时，波特率发生器重新装入 SSPxADD<6:0> 的内容并开始计数。SDAx 和 SCLx 必须在一个计数周期 TBRG 内始终被采样为高电平。在接下来的一个 TBRG 周期内， SDAx 引脚被拉为低电平（SDAx = 0），同时 SCLx 为高电平。然后 RSEN 位（SSPxCON2<1>）将自动清零，波特率发生器不会重载， SDAx 引脚保持低电平。一旦在 SDAx 和 SCLx 引脚上检测到启动条件，S 位（SSPxSTAT<3>）将被置 1。直到波特率发生器超时后， SSPxIF 位才会置 1。

注 1： 有任何其他事件在进行时，编程设置对 RSEN 无效。

2： 在重复启动条件发生期间，下列事件将会导致总线冲突：

- 当 SCLx 由低电平变为高电平时，采样到 SDAx 为低电平。
- 在 SDAx 被拉低之前，SCLx 变为低电平。这表示另一个主器件正尝试发送数据 1。

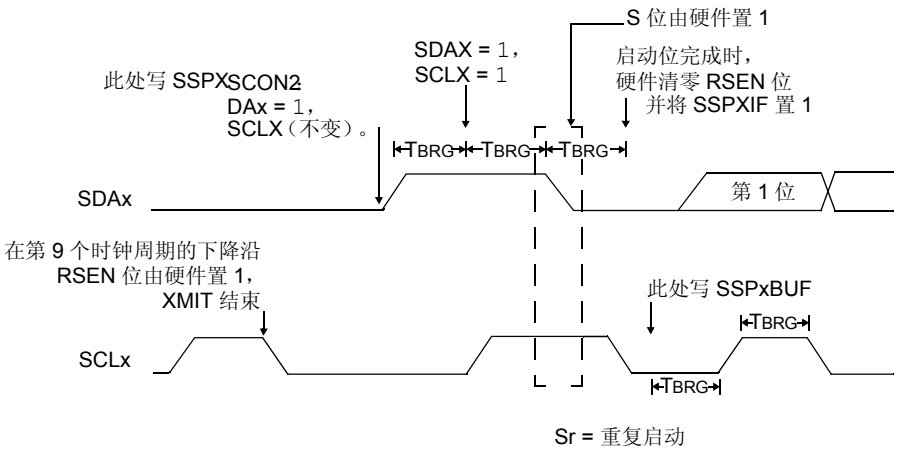
一旦 SSPxIF 位被置 1，用户便可以在 7 位地址模式下将 7 位地址，或者在 10 位地址模式下写入默认的第一个地址字节写入 SSPxBUF。当发送完第一个 8 位并接收到一个 ACK 后，用户可以发送另外 8 位地址（10 位地址模式下）或 8 位数据（7 位地址模式下）。

19.4.9.1 WCOL 状态标志位

在重复启动时序进行中，如果用户写 SSPxBUF，则 WCOL 被置 1，同时缓冲器内容不变（写操作无效）。

注： 由于不允许事件排队，在重复启动条件完成之前，不能对 SSPxCON2 的低 5 位进行写操作。

图 19-22： 重复启动条件波形



© 2007 Microchip Technology Inc.

19.4.10 I²C 主控模式下的发送

发送一个数据字节、一个 7 位地址或一个 10 位地址的另一半，都可通过写一个值到SSPxBUF寄存器来实现。该操作将使缓冲器满标志位 BF 置 1，波特率发生器开始计数，同时启动下一次发送。在出现 SCLx 的下降沿后 （见数据保持时间规范参数 106），地址 / 数据的每一位被移出至 SDAx 引脚。在一个波特率发生器计满返回计数周期（TBRG）内，SCLx 保持低电平。数据应该在 SCLx 被释放为高电平前保持有效（见数据建立时间规范参数 107）。当 SCLx 引脚被释放为高电平时，它将在一个 TBRG 内保持为高电平。在此期间以及 SCLx 的下一个下降沿之后的一段时间内， SDAx 引脚上的数据必须保持稳定。在第 8 位数据被移出（第 8 个时钟周期的下降沿）之后， BF 标志位清零，同时主器件释放 SDAx。此时如果发生地址匹配或是数据被正确接收，被寻址的从器件将在第 9 位以一个 ACK 位做出响应。ACK的状态在第9个时钟周期的下降沿写入ACKDT位。主器件接收到应答之后，应答状态位 ACKSTAT 会被清零；如果未收到应答，则该位被置 1。第 9 个时钟周期之后，SSPxIF 位会置 1，主控时钟（波特率发生器）暂停，直到下一个数据字节装入 SSPxBUF。SCLx 引脚保持低电平，并且 SDAx 保持不变 （图 19-23）。

在写 SSPxBUF 之后，地址的每一位在 SCLx 的下降沿被移出，直至所有的 7 位地址和 R/W 位都被移出。在第 8 个时钟的下降沿，主器件将 SDAx 引脚拉为高电平，以允许从器件发出应答响应。在第 9 个时钟的下降沿，主器件通过采样 SDAx 引脚来判断地址是否被从器件识别。 ACK 位的状态被装入 ACKSTAT 状态位（SSPxCON2<6>）。在发送地址的第 9 个时钟下降沿之后，SSPxIF 标志位置 1，BF 标志位清零，波特率发生器关闭直到下一次写SSPxBUF，同时SCLx引脚保持低电平并且允许 SDAx 引脚悬空。

19.4.10.1 BF 状态标志位

在发送模式下， BF 位 （SSPxSTAT<0>）在 CPU 写 SSPxBUF 时置 1，在所有 8 位数据被移出后清零。

19.4.10.2 WCOL 状态标志位

如果用户在发送过程中（即，SSPxSR 仍在移出数据字节时）写 SSPxBUF，则 WCOL 置 1，且在写 SSPxBUF 之后的两个 TCY 后此缓冲器的内容不变 （未发生写操作）。如果在两个 TCY 内重新写 SSPxBUF，则 WCOL 位置 1 并且更新 SSPxBUF。这可能导致错误的传输。

在每次写 SSPxBUF 之后，用户都应验证 WCOL 位是否清零以确保正确传输。在所有情况下，WCOL 都必须用软件清零。

19.4.10.3 ACKSTAT 状态标志位

在发送模式下，当从器件发送应答响应（$\overline{\text{ACK}}$ = 0）时，ACKSTAT 位 （SSPxCON2<6>）清零；当从器件没有应答 （$\overline{\text{ACK}}$ = 1）时，该位置 1。从器件在识别出其地址 （包括广播呼叫地址）或正确接收数据后，会发送一个应答。

19.4.11 I²C 主控模式接收

通过编程接收使能位 RCEN（SSPxCON2<3>）使能主控模式接收。

注：	RCEN 位置 1 前， MSSP 模块必须处于空闲状态，否则对 RCEN 位的置 1 将无效。

波特率发生器开始计数，每次计满返回时， SCLx 引脚的状态发生改变 （由高变低或由低变高），数据被移入 SSPxSR。第 8 个时钟的下降沿之后，接收使能标志位自动清零，SSPxSR 的内容装入 SSPxBUF，BF 标志位置 1， SSPxIF 标志位置 1，波特率发生器暂停计数，SCLx 保持为低电平。此时 MSSP 处于空闲状态，等待下一条命令。当 CPU 读缓冲器时，BF 标志位将自动清零。通过将应答时序使能位 ACKEN（SSPxCON2<4>）置 1，用户可以在接收结束后发送应答位。

19.4.11.1 BF 状态标志位

接收时，当将地址或数据字节从 SSPxSR 装入 SSPxBUF 时， BF 位置 1。在读 SSPxBUF 寄存器时，BF 位清零。

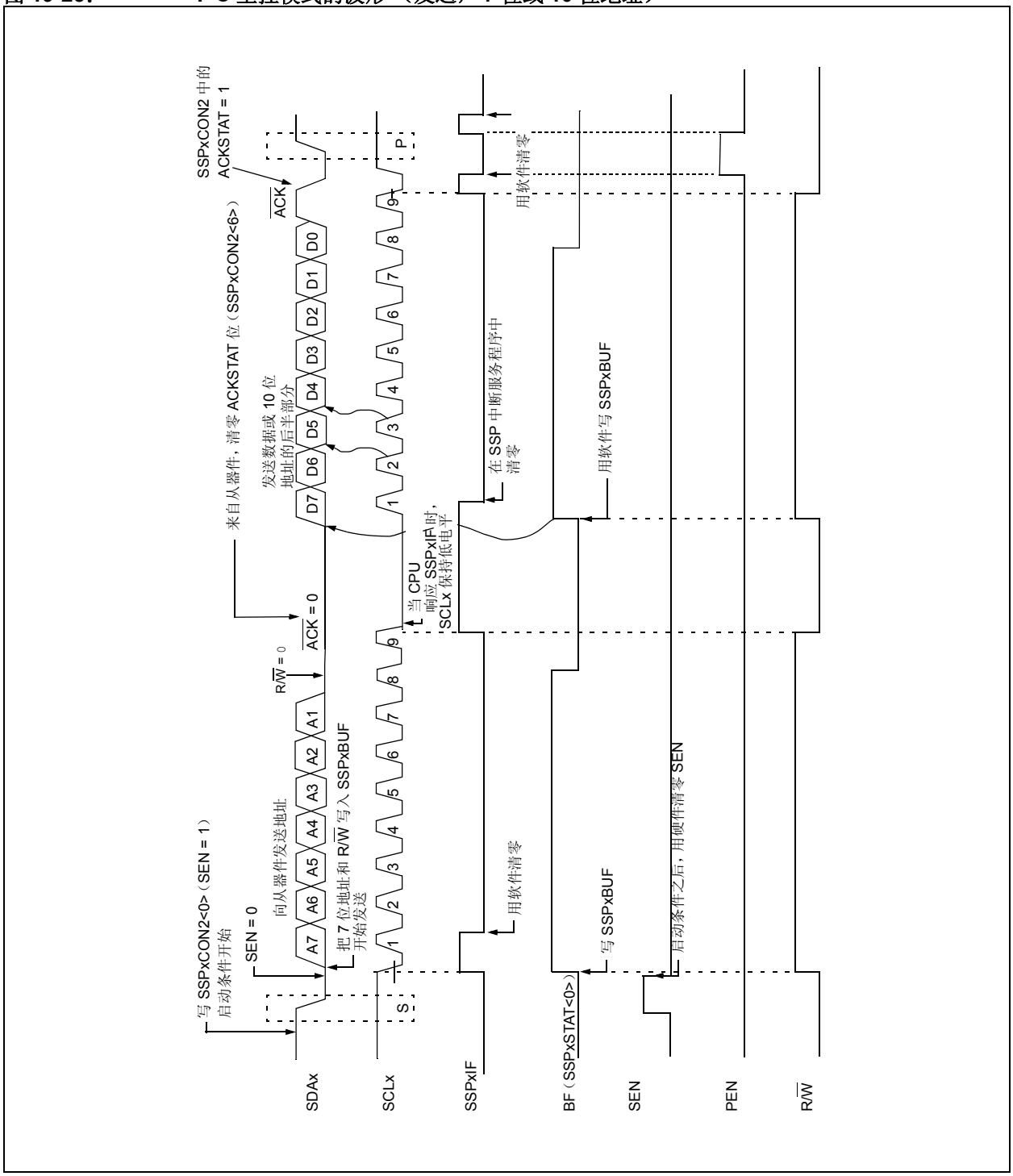
19.4.11.2 SSPOV 状态标志位

接收时，当 SSPxSR 接收到 8 位数据时，SSPOV 位置 1， BF 标志位已经在上一次接收时置 1。

19.4.11.3 WCOL 状态标志位

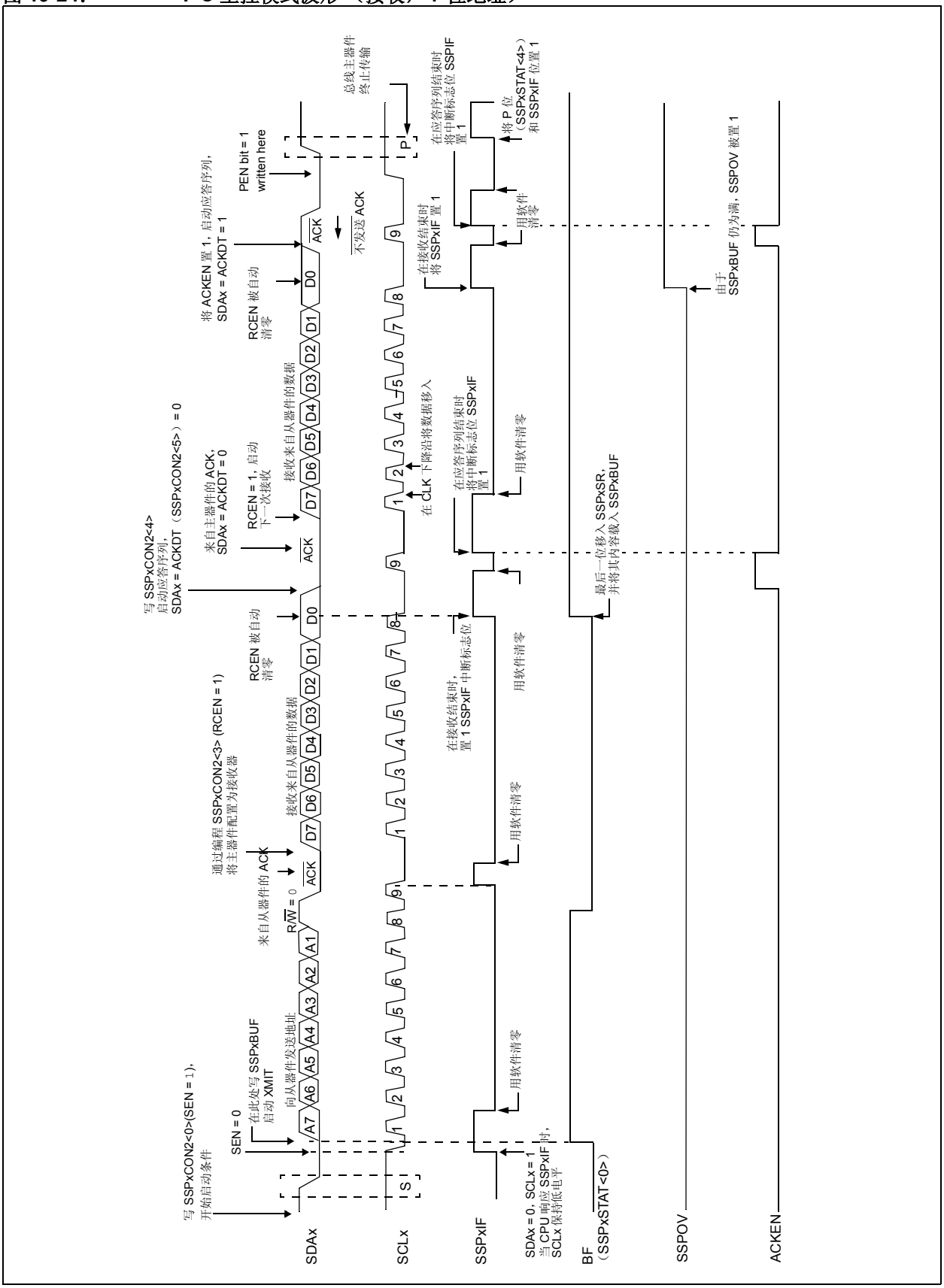
如果用户在接收过程中（即，SSPxSR 仍在移入数据字节时）写 SSPxBUF，则 WCOL 位置 1，且缓冲器内容不变 （未发生写操作）。

图 19-23： I^2C 主控模式的波形 （发送， 7 位或 10 位地址）



 © 2007 Microchip Technology Inc.

图 19-24：　　I^2C 主控模式波形 （接收， 7 位地址）



19.4.12 应答序列时序

将应答序列使能位 ACKEN（SSPxCON2<4>）置 1 即可使能应答序列。当该位被置 1 时，SCLx 引脚被拉低，应答数据位的内容出现在 SDAx 引脚上。如果用户希望产生一个应答，则应该将 ACKDT 位清零。如果不希望产生应答，应在应答序列开始前将 ACKDT 位置 1。然后波特率发生器进行一个计满返回周期（TBRG）的计数，SCLx 引脚电平被拉高。当 SCLx 引脚被采样为高电平时（时钟仲裁），波特率发生器进行一个 TBRG 周期的计数，然后将 SCLx 引脚拉低。在这之后，ACKEN 位自动清零，波特率发生器关闭，且 MSSP 模块进入空闲模式（图 19-25）。

19.4.12.1 WCOL 状态标志位

如果用户在应答序列进行过程中试图写 SSPxBUF，则 WCOL 将置 1 并且缓冲器的内容不会改变（未发生写操作）。

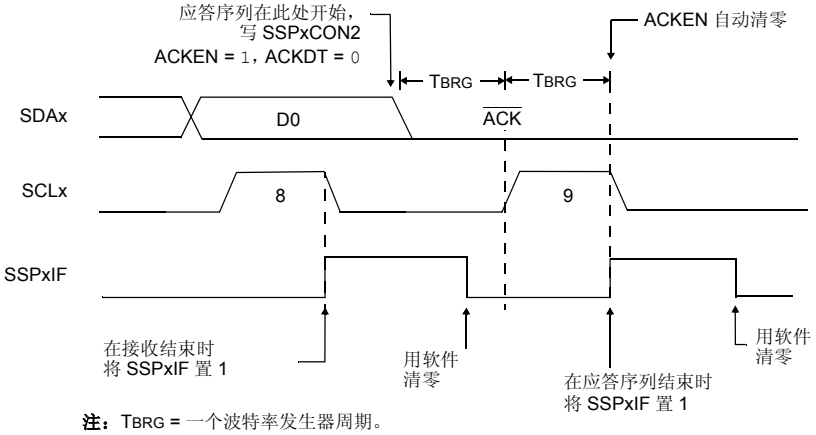
19.4.13 停止条件时序

将停止序列使能位 PEN（SSPxCON2<2>）置 1，在接收 / 发送结束后， SDAx 引脚上将产生停止位。在接收 / 发送结束时，SCLx 引脚在第 9 个时钟的下降沿后保持低电平。当 PEN 位置 1 时，主器件将 SDAx 线置为低电平。当 SDAx 线被采样为低电平时，重载波特率发生器并递减计数至 0。当波特率发生器发生超时，SCLx 引脚被拉为高电平，在一个 TBRG（波特率发生器计满返回周期）之后， SDAx 引脚将被拉高。当 SDAx 引脚被采样为高电平且 SCLx 也是高电平时， P 位（SSPxSTAT<4>）置 1。一个 TBRG 之后， PEN 位清零，同时 SSPxIF 位置 1（图 19-26）。

19.4.13.1 WCOL 状态标志位

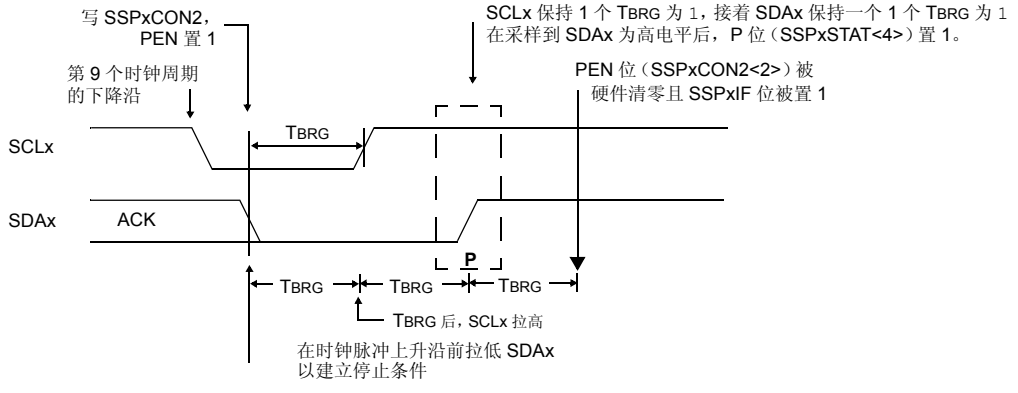
如果用户在停止序列过程中试图写 SSPxBUF，则 WCOL 位将置 1，缓冲器的内容不会改变（未发生写操作）。

图 19-25： 应答序列波形



注： TBRG = 一个波特率发生器周期。

图 19-26： 停止条件接收或发送模式



注： TBRG = 一个波特率发生器周期。

© 2007 Microchip Technology Inc.

19.4.14 休眠工作方式

在休眠模式下，I^2C 模块能够接收地址或数据，并且在地址匹配或字节传输完成后，将唤醒处理器（如果允许 MSSP 中断）。

19.4.15 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

19.4.16 多主机模式

在多主机模式下，在检测到启动和停止条件时产生中断，这可用于判断总线何时空闲。停止（P）位和启动（S）位在复位时或禁止 MSSP 模块时清零。当 P 位（SSPxSTAT<4>）置 1，或 P 位和 S 位都为零且总线空闲时，可以取得 I^2C 总线的控制权。当总线忙时，一旦出现停止条件，允许 MSSP 中断的情况下将产生中断。

在多主机模式下，必须一直监视 SDAx 线来进行仲裁，查看信号电平是否为期望的输出电平。此检查由硬件实现，其结果保存在 BCLxIF 位中。

可能导致仲裁失败的情况是：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

19.4.17 多主机通信、总线冲突与总线仲裁

多主机模式是通过总线仲裁来支持的。当主器件将地址 / 数据位输出到 SDAx 引脚时，如果主器件通过将 SDAx 引脚悬空为高电平以在 SDAx 上输出 1，而另一个主器件输出 0，就会发生总线仲裁。如果 SDAx 引脚上期望的数据是1，而实际在SDAx引脚上采样到的数据是 0 ，则表示发生了总线冲突。主器件将把总线冲突中断标志位 BCLxIF 置 1，并将 I^2C 端口复位到空闲状态（图 19-27）。

如果在发送过程中发生总线冲突，则会暂停发送，并且清零 BF 标志位，SDAx 和 SCLx 线被拉高同时将 SSPxBUF 置于可写状态。当执行完总线冲突中断服务程序后，如果 I^2C 总线空闲，用户可通过发出启动条件恢复通信。

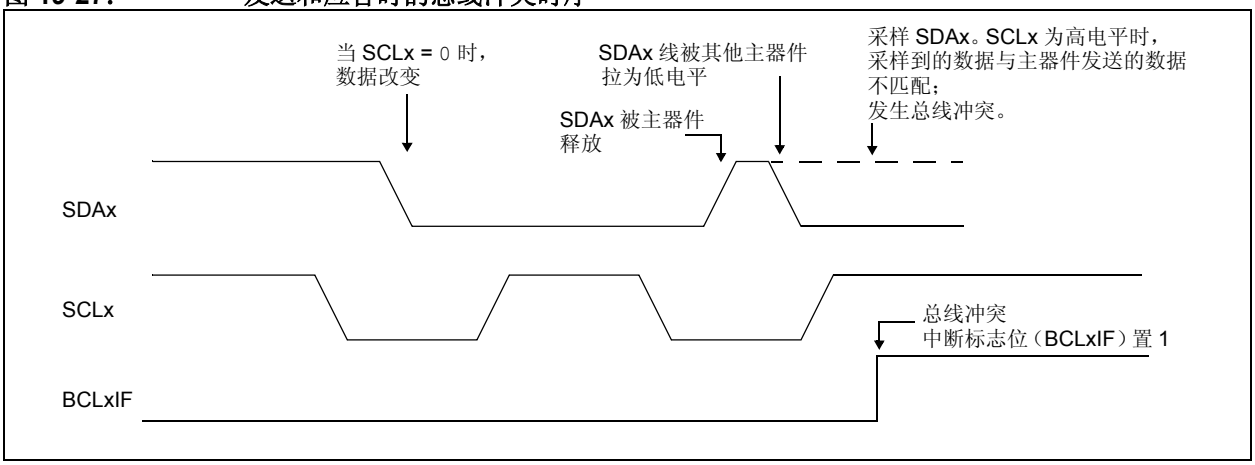
如果在启动、重复启动、停止或应答条件进行的过程中发生总线冲突，那么上述条件将被中止，SDAx 和 SCLx 线被拉高，SSPxCON2 寄存器中的对应控制位清零。当执行完总线冲突中断服务程序后，如果 I^2C 总线空闲，用户可通过发出启动条件恢复通信。

主器件将继续监视 SDAx 和 SCLx 引脚。如果出现停止条件，SSPxIF 位将被置 1。

无论发生总线冲突时发送的进度如何，写入 SSPxBUF 都会重新从第一个数据位开始发送数据。

在多主机模式下，通过在检测到启动和停止条件时产生中断可以确定总线何时空闲。SSPxSTAT 寄存器中的 P 位置 1 时，或总线空闲且 S 和 P 位清零时，可以获取 I^2C 总线的控制权。

图 19-27： 发送和应答时的总线冲突时序



19.4.17.1 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

a) 在启动条件开始时， SDAx 或 SCLx 被采样为低电平 （图 19-28）。

b) SDAx 被拉低之前， SCLx 被采样为低电平（图 19-29）。

在启动条件期间， SDAx 和 SCLx 引脚都会被监视。

如果 SDAx 引脚已经是低电平，或 SCLx 引脚已经是低电平，则：

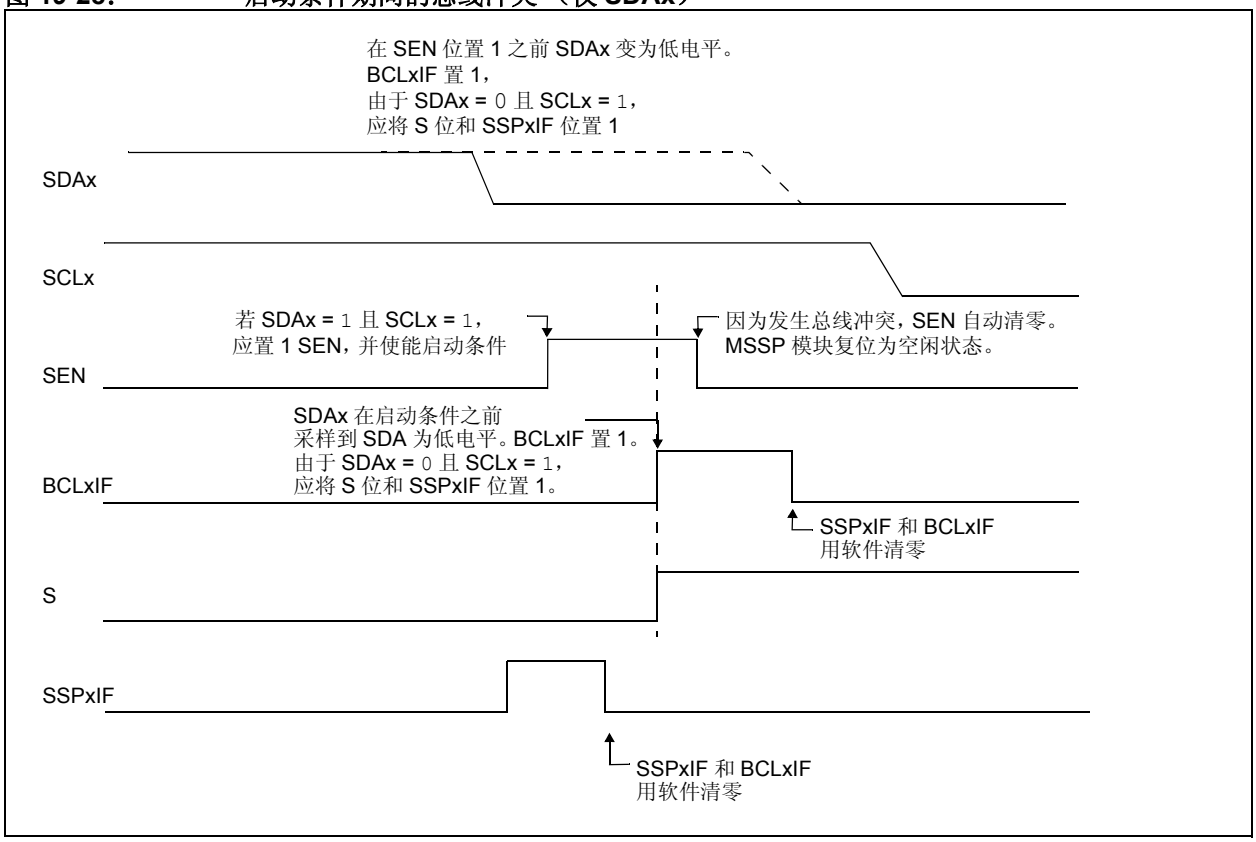
- 中止启动条件，
- BCLxIF 标志位置 1，且
- MSSP 模块复位为空闲状态 （图 19-28）。

启动条件从 SDAx 和 SCLx 引脚被拉高开始。当 SDAx 引脚被采样为高电平时，波特率发生器装入 SSPxADD<6:0> 的内容并向下计数至 0。如果在 SDAx 为高电平时，SCLx 引脚被采样为低电平，则发生总线冲突，因为这表示另一台主器件在启动条件期间试图发送数据 1。

如果 SDAx 引脚在该计数周期内采样为低电平，则 BRG 复位，同时 SDAx 线保持原值 （图 19-30）。但是，如果 SDAx 引脚采样为 1，SDAx 引脚将在 BRG 计数结束时被置为低电平。接着，重载波特率发生器并使之向下计数至 0。在此期间，如果采样到 SCLx 引脚为 0，则不会发生总线冲突。在 BRG 计数结束时，SCLx 引脚被拉为低电平。

注：	在启动信号期间不太可能发生总线冲突，因为两个总线主器件不可能精确地在同一时刻发出启动条件。因此一个主器件将总是先于另一个主器件将 SDAx 拉低。由于必须允许两个主器件对启动条件后的第一个地址进行仲裁，因此上述情况不会引起总线冲突。如果地址是相同的，必须继续对数据部分、重复启动或停止条件进行仲裁。

图 19-28： 启动条件期间的总线冲突 （仅 SDAx）



 © 2007 Microchip Technology Inc.

图 19-29：　　启动条件期间的总线冲突 （SCLx = 0）

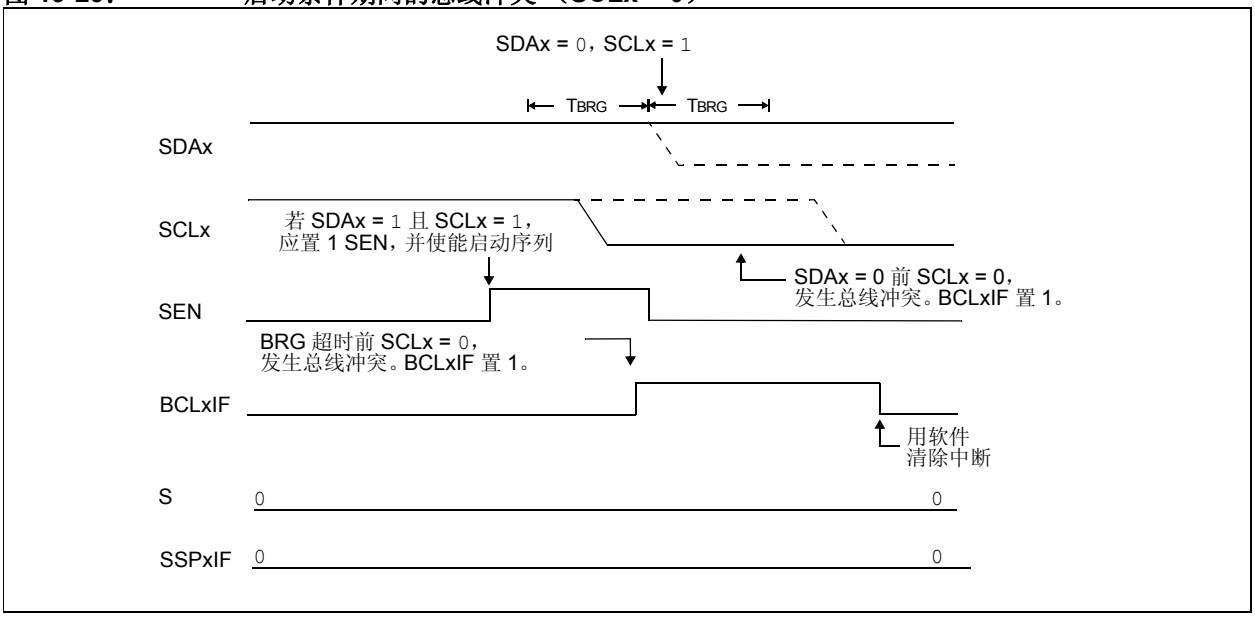
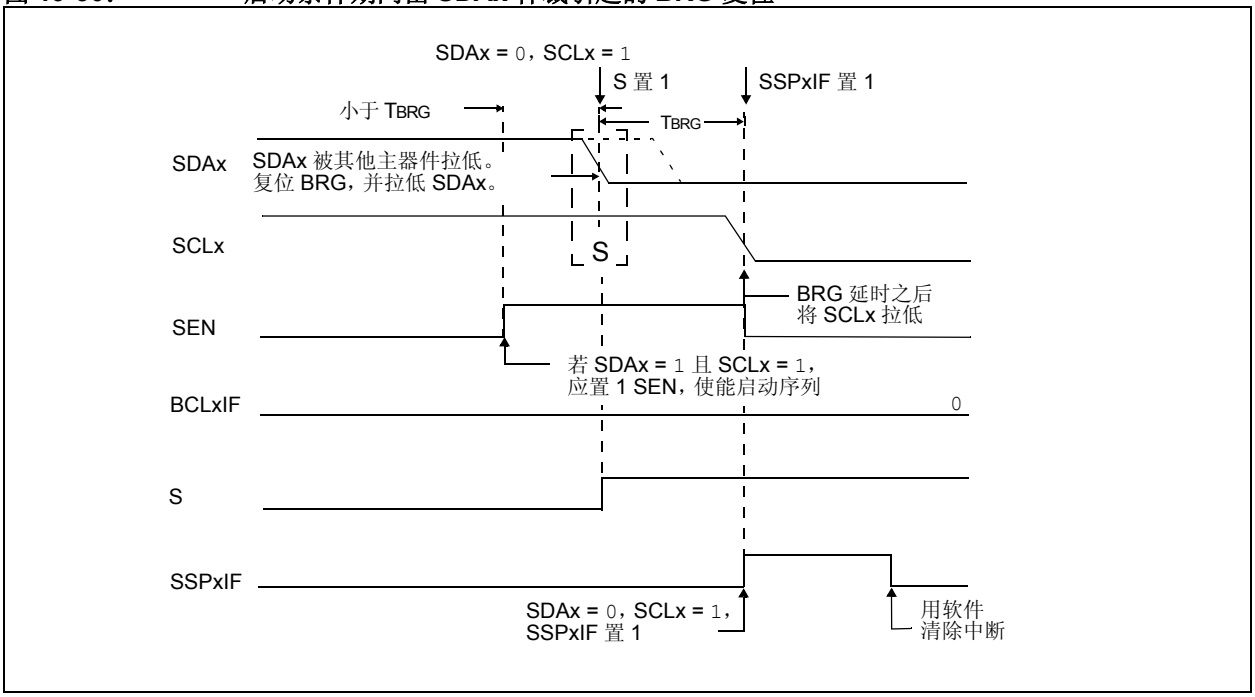


图 19-30：　　启动条件期间由 SDAx 仲裁引起的 BRG 复位



19.4.17.2 重复启动条件期间的总线冲突

在重复启动条件期间，以下事件会导致发生总线冲突：

a) 在SCLx由低电平变为高电平的过程中，在SDAx上采样到低电平。

b) 在 SDAx 被拉为低电平之前， SCLx 变为低电平，表示另一个主器件正试图发送数据 1。

当用户拉高 SDAx 且允许该引脚悬空为高电平时，波特率发生器载入 SSPxADD<6:0> 中的值并向下计数至 0。接着 SCLx 引脚被置为高电平，当 SCLx 采样到高电平时，对 SDAx 引脚进行采样。

如果 SDAx 为低电平，则表示已发生了总线冲突 （即另一个主器件正试图发送数据 0，图 19-31）。如果 SDAx 被采样为高电平，则重载 BRG 并使之开始计数。如果 SDAx 在 BRG 超时之前从高电平变为低电平，则不会发生总线冲突，因为两个主器件不可能精确地在同一时刻将 SDAx 拉低。

如果 SCLx 在 BRG 超时之前从高电平变为低电平，但 SDAx 尚未变为低电平，那么将发生总线冲突。此情况说明，在重复启动条件期间另一个主器件正试图发送数据 1 （见图 19-32）。

如果在 BRG 超时结束时 SCLx 和 SDAx 都仍然是高电平，则 SDAx 引脚被拉低，重载 BRG 并使之开始计数。在计数结束时，无论 SCLx 引脚的状态为何，SCLx 引脚都被拉低，重复启动条件完成。

图 19-31： 重复启动条件期间的总线冲突 （情形 1）

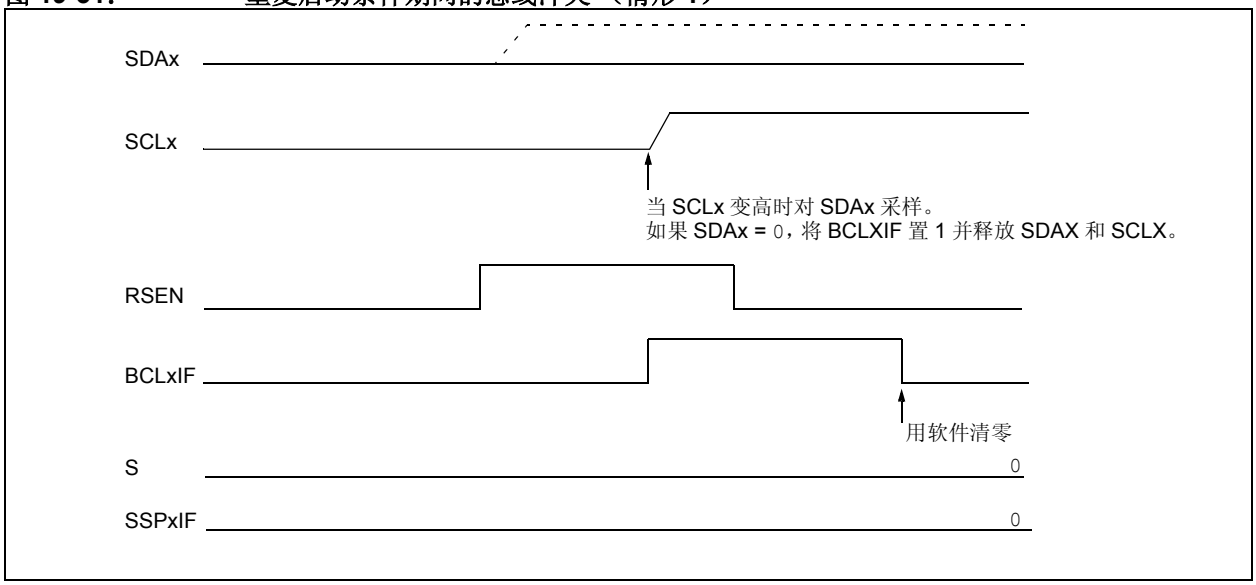
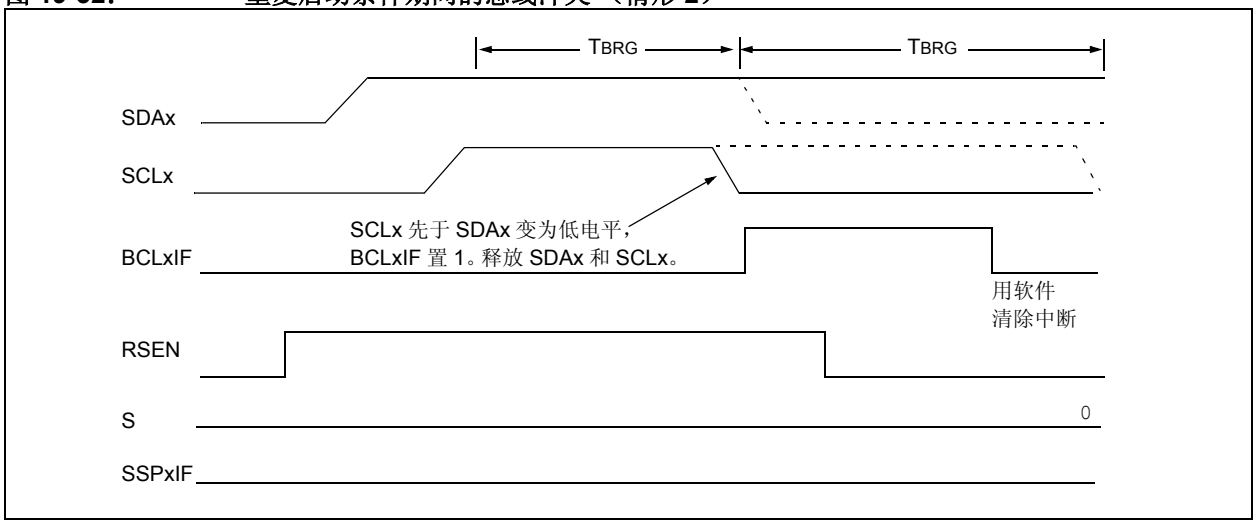


图 19-32： 重复启动条件期间的总线冲突 （情形 2）



© 2007 Microchip Technology Inc.

19.4.17.3 停止条件期间的总线冲突

以下事件会导致停止条件期间发生总线冲突：

a) SDAx 已被拉高并允许悬空为高电平之后，SDAx 在 BRG 超时后被采样为低电平。

b) SCLx 引脚被拉高之后，SCLx 在 SDAx 变成高电平之前被采样为低电平。

停止条件从 SDAx 被拉低开始。当 SDAx 被采样为低电平时， SCLx 引脚就可以悬空。当引脚被采样为高电平时（时钟仲裁），波特率发生器被载入 SSPxADD<6:0> 的值并向下计数到 0。BRG 超时后，对 SDAx 采样。如果采样到 SDAx 为低电平，则表示已发生总线冲突，这是因为另一个主器件正试图发送数据 0（图 19-33）。如果 SCLx 引脚在允许 SDAx 悬空为高电平前被采样为低电平，也会发生总线冲突，原因同上（图 19-34）。

图 19-33： 停止条件期间的总线冲突（情形 1）

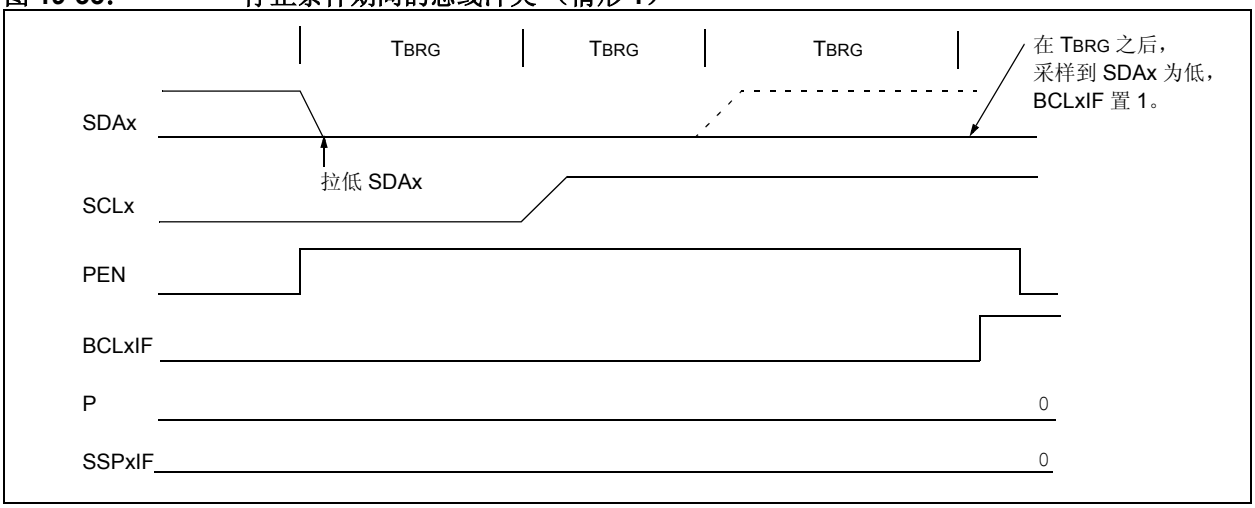


图 19-34： 停止条件期间的总线冲突（情形 2）

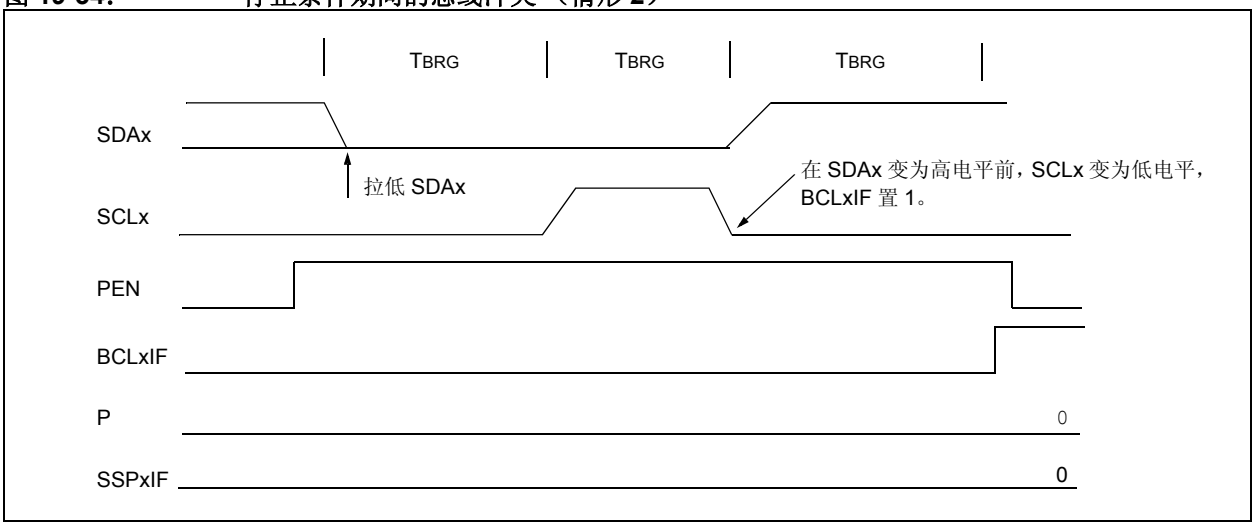


表 19-4： 与 I²C™ 工作模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	58
SSP1BUF	MSSP1 接收缓冲器 / 发送寄存器								56
SSP1ADD	MSSP1 地址寄存器 （I²C™ 从动模式）， MSSP1 波特率重载寄存器 （I²C 主控模式）								56
SSP1MSK(1)	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	56
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	56
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	56
	GCEN	ACKSTAT	ADMSK5(2)	ADMSK4(2)	ADMSK3(2)	ADMSK2(2)	ADMSK1(2)	SEN	
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	56
SSP2BUF	MSSP2 接收缓冲器 / 发送寄存器								59
SSP2ADD	MSSP2 地址寄存器 （I²C 从动模式）， MSSP2 波特率重载寄存器 （I²C 主控模式）								59
SSP2MSK(1)	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	59
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	59
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	59
	GCEN	ACKSTAT	ADMSK5(2)	ADMSK4(2)	ADMSK3(2)	ADMSK2(2)	ADMSK1(2)	SEN	
SSP2STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	59

图注： — = 未实现，读为 0。在 I²C™ 模式下 MSSP 模块未用阴影单元。

注 1： SSPxMSK 与 SSPxADD 在 SFR 中共享同一地址，只有在 7 位屏蔽模式下的某种 I²C™ 从动工作模式下才可访问 SSPxMSK。更多详细信息，请参见**第 19.4.3.4 节 “7 位地址屏蔽模式”**。

2： 只有在 I²C 从动模式下工作时才可使用备用位定义。

© 2007 Microchip Technology Inc.

20.0 增强型通用同步 / 异步收发器（EUSART）

增强型通用同步 / 异步收发器（Enhanced Universal Synchronous Asynchronous Receiver Transmitter，EUSART）模块是两个串行 I/O 模块之一（一般也将 EUSART 称为串行通信接口或 SCI）。可以将 EUSART 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 USART 模块实现了更多的功能，包括自动波特率检测和校准以及在接收到“同步间隔”字符和发送 12 位间隔字符时自动唤醒。这些功能使 EUSART 模块成为局域互连网络（Local Interconnect Network，LIN）总线系统非常理想的选择。

PIC18F87J11 系列的所有器件都配备有两个独立的 EUSART 模块，分别称为 EUSART1 和 EUSART2。可将这两个模块配置为以下几种工作模式：

- 带有以下功能的全双工异步模式：
  - 接收到间隔字符自动唤醒
  - 自动波特率校准
  - 12 位间隔字符发送
- 时钟极性可选的半双工同步主模式
- 时钟极性可选的半双工同步从模式

EUSART1 和 EUSART2 的引脚分别与 PORTC（RC6/TX1/CK1 和 RC7/RX1/DT1）和 PORTG（RG1/TX2/CK2 和 RG2/RX2/DT2）的功能复用。将这些引脚配置为 EUSART 的步骤如下：

- 对于 EUSART1：
  - SPEN（RCSTA1<7>）位必须置 1
  - TRISC<7> 位必须置 1
  - TRISC<6> 位必须清 0 以使该模块工作于异步和同步主模式
  - TRISC<6> 位必须置 1 以使该模块工作于同步从模式
- 对于 EUSART2：
  - SPEN（RCSTA2<7>）位必须置 1
  - TRISG<2> 位必须置 1
  - TRISG<1> 位必须清 0 以使该模块工作于异步和同步主模式
  - TRISC<6> 位必须置 1 以使该模块工作于同步从模式

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 模块的操作是通过 3 个寄存器控制的：

- 发送状态和控制寄存器（TXSTAx）
- 接收状态和控制寄存器（RCSTAx）
- 波特率控制寄存器（BAUDCONx）

在下页的寄存器 20-1、寄存器 20-2 和寄存器 20-3 中分别对这些寄存器进行了详细说明。

注： 在本节中，凡是涉及与特定 EUSART 模块相关的寄存器和位的名称一般都采用以“x”代替特定模块编号的方式。因此，“RCSTAx”可能指 EUSART1 的接收状态寄存器，也可能指 EUSART2 的接收状态寄存器。

寄存器 20-1： TXSTAx：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 = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **CSRC**：时钟源选择位
异步模式：
任意值。
同步模式：
1 = 主模式 （时钟来自内部 BRG）
0 = 从模式 （时钟来自外部时钟源）

bit 6 **TX9**：9 位发送使能位
1 = 选择 9 位发送
0 = 选择 8 位发送

bit 5 **TXEN**：发送使能位 (1)
1 = 使能发送
0 = 禁止发送

bit 4 **SYNC**：EUSART 模式选择位
1 = 同步模式
0 = 异步模式

bit 3 **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 （在完成时用硬件清零）
0 = “同步间隔”字符发送完成
同步模式：
任意值。

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
该位可以是地址 / 数据位或奇偶校验位。

注 1： 同步模式下 SREN/CREN 的优先级高于 TXEN。

© 2007 Microchip Technology Inc.

寄存器 20-2： RCSTAx：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 = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **SPEN**：串口使能位
1 = 使能串口 （配置 RXx/DTx 和 TXx/CKx 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：
任意值。
同步主模式：
1 = 使能单字节接收
0 = 禁止单字节接收
此位在接收完成后清零。
同步从模式：
任意值。

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、允许中断并装载接收缓冲器
0 = 禁止地址检测、接收所有字节并且第 9 位可用作奇偶校验位
9 位异步模式 （RX9 = 0）：
任意值。

bit 2 **FERR**：帧错误位
1 = 帧错误 （可以通过读 RCREGx 寄存器更新并接收下一个有效字节）
0 = 没有帧错误

bit 1 **OERR**：溢出错误位
1 = 溢出错误 （可以通过清零 CREN 位清零）
0 = 没有溢出错误

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，其值必须由用户固件计算得到。

寄存器 20-3： BAUDCONx：波特率控制寄存器

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

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **ABDOVF**：自动波特率采集进位状态位
1 = 在自动波特率检测模式下发生了 BRG 进位 （必须在软件中清零）
0 = 没有发生 BRG 进位

bit 6 **RCIDL**：接收操作空闲状态位
1 = 接收操作空闲
0 = 接收操作有效

bit 5 **DTRXP**：数据 / 接收极性选择位
异步模式：
1 = 接收数据 （RXx）翻转 （低电平有效）
0 = 接收数据 （RXx）不翻转 （高电平有效）
同步模式：
1 = 数据 （DTx）翻转 （低电平有效）
0 = 数据 （DTx）不翻转 （高电平有效）

bit 4 **SCKP**：同步时钟极性选择位
异步模式：
1 = 发送空闲状态 （TXx）为低电平
0 = 发送空闲状态 （TXx）为高电平
同步模式：
1 = 时钟空闲状态 （CKx）为高电平
0 = 时钟空闲状态 （CKx）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——SPBRGHx 和 SPBRGx
0 = 8 位波特率发生器——仅 SPBRGx （兼容模式），忽略 SPBRGHx 的值

bit 2 **未实现**：读为 0

bit 1 **WUE**：唤醒使能位
异步模式：
1 = EUSART 将继续采样 RXx 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 = 不监视 RXx 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.

20.1 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 （BAUDCONx<3>）置 1 可选择 16 位模式。

SPBRGHx:SPBRGx 这对寄存器控制独立运行定时器的周期。在异步模式下， BRGH （TXSTAx<2>）位和 BRG16（BAUDCONx<3>）位也控制波特率。在同步模式下，会忽略 BRGH 位。表 20-1 所示为不同 EUSART 模式的波特率计算公式，但仅适用于主模式 （由内部产生时钟信号）。

给定目标波特率和 FOSC 的情况下，可以使用表 20-1 中的公式计算 SPBRGHx:SPBRGx 寄存器的最接近整数值，并根据该值确定波特率误差。例 20-1 给出了一个计算示例。表 20-2 中给出了各种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 减少波特率误差，或者在快速振荡频率条件下实现低波特率都可能是很有益的。

向 SPBRGHx:SPBRGx 寄存器写入新值会导致 BRG 定时器复位 （或清零）。这可以确保 BRG 无需等待定时器溢出就可以输出新的波特率。

20.1.1 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与先前不同的频率下。这可能需要调整 SPBRGx 寄存器对中的值。

20.1.2 采样

择多检测电路对 RXx 引脚（RC7/RX1/DT1 或者 RG2/RX2/DT2）采样三次，以判定 RXx 引脚上出现的是高电平还是低电平。

表 20-1： 波特率公式

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

图注： x = 任意值， n = SPBRGHx:SPBRGx 寄存器对的值

例 20-1：　　计算波特率误差

针对工作在异步模式下，工作频率 FOSC 为 16MHz，采用 8 位 BRG，目标波特率为 9600 bps 的器件：

目标波特率　　=　FOSC/(64 ([SPBRGHx:SPBRGx] + 1))

求解 SPBRGHx:SPBRGx 的值：

X　=　((FOSC/ 目标波特率)/64) － 1

=　((16000000/9600)/64) － 1

=　[25.042] = 25

计算得到的波特率　=　16000000/(64 (25 + 1))

=　9615

误差　=　(计算得到的波特率－目标波特率)/ 目标波特率

=　(9615 － 9600)/9600 = 0.16%

表 20-2：　　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 所在的页
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								59
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								59

图注：　— = 未实现位，读为 0。 BRG 未使用阴影单元。

© 2007 Microchip Technology Inc.

表 20-3： 异步模式下的波特率

目标波特率（Kbps）	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（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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	—	—	—

目标波特率（Kbps）	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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	—	—	—	—	—	—

目标波特率（Kbps）	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（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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	—	—	—

目标波特率（Kbps）	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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	—	—	—	—	—	—

表 20-3： 异步模式下的波特率 （续）

目标波特率（Kbps）	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（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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	—	—	—

目标波特率（Kbps）	SYNC = 0, BRGH = 0, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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	—	—	—	—	—	—

目标波特率（Kbps）	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（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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

目标波特率（Kbps）	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（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 进制）	实际波特率（Kbps）	误差 %	SPBRG 值（10 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	—	—	—	—	—	—

© 2007 Microchip Technology Inc.

20.1.3 自动波特率检测

增强型 USART 模块支持自动检测和校准波特率。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过程（图 20-1）。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 RXx 输入信号提供时钟源，而是由 RXx 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周期 （以位数表示）。

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所造成的影响，在接收低位和高位的时间内都要进行测量。在起始位后，SPBRGx 使用预先选择的时钟源，在 RXx 引脚上的第一个上升沿开始计数。在 RXx 引脚传输了 8 个位，或在检测到第 5 个上升沿后，会将在相应 BRG 周期内累计的值保存在 SPBRGHx:SPBRGx 寄存器对中。一旦检测到第 5 个上升沿（应将其视为停止位），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 （BAUDCONx<7>）有所反映。当 BRG 计满返回时，该位由硬件置 1，用户也可用软件将其置 1 或清零。在发生计满返回事件后，继续保持 ABD 模式， ABDEN 位保持置 1 （图 20-2）。

当校准波特率周期时，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。无论 BRG16 位的设置如何， SPBRGx 和 SPBRGHx 都将被用作一个 16 位计数器。通过检查 SPBRGHx 寄存器中的值是否为 00h，用户可以验证在 8 位模式下是否发生了进位。参见表 20-4 可获取 BRG 计数器的时钟速率。

当产生 ABD 序列时，EUSART 状态机保持在空闲状态。一旦在 RXx 上检测到第 5 个上升沿，中断标志位 RCxIF 就会置 1。需要读取 RCREGx 中的值，来清除中断标志位 RCxIF。应丢弃 RCREGx 的内容。

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： 由用户判断进入的字符波特率是否处于所选 BRG 时钟源范围内。由于误码率的原因，某些振荡频率和 EUSART 波特率的组合是无法实现的。在使用自动波特率检测功能时，必须考虑系统总体时序和通信波特率。

表 20-4： BRG 计数器时钟速率

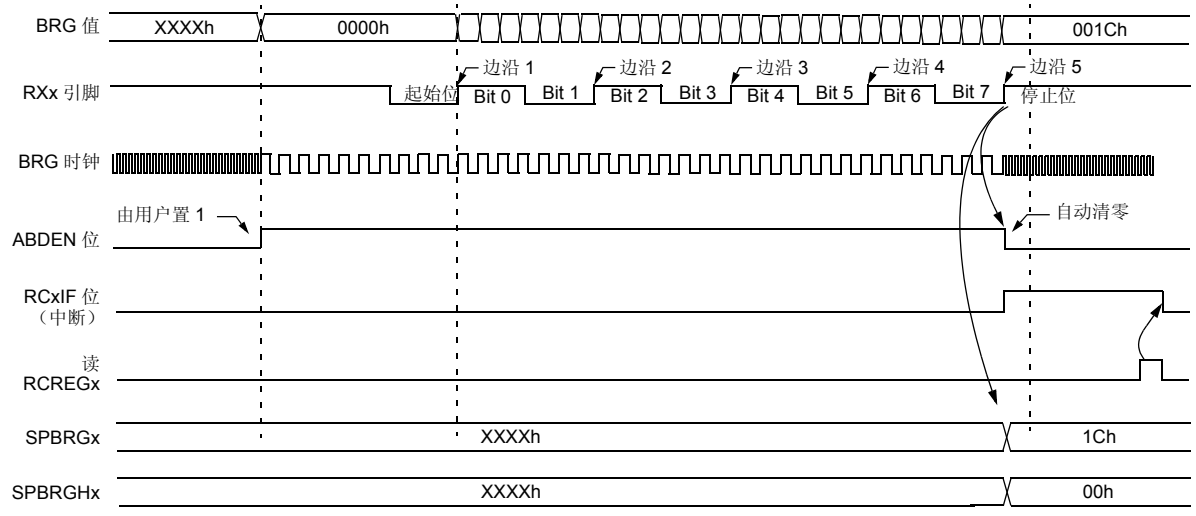
BRG16	BRGH	BRG 计数器时钟
0	0	FOSC/512
0	1	FOSC/128
1	0	FOSC/128
1	1	FOSC/32

注： 在产生 ABD 序列时，不管 BRG16 的设置如何， SPBRGx 和 SPBRGHx 均被用作一个 16 位计数器。

20.1.3.1 ABD 和 EUSART 发送

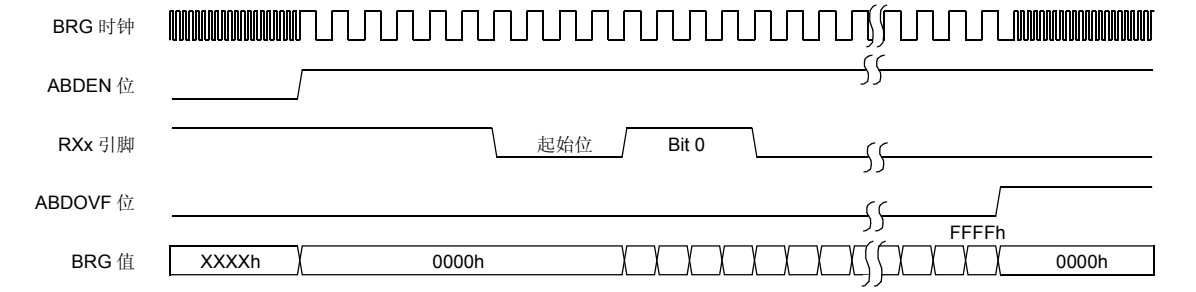
由于在 ABD 采集期间 BRG 时钟是反向的，因此在 ABD 期间不能使用 EUSART 发送器。这意味着只要 ABDEN 位置 1，就不能写入 TXREGx。用户还应确保在发送期间 ABDEN 不变为置 1 状态，否则可能会导致无法预料的 EUSART 操作。

图 20-1：　　自动波特率计算



注：　必须将 EUSART 模块配置为异步模式，并且 WUE = 0 才能产生正确的 ABD 时序。

图 20-2：　　BRG 上溢时序



© 2007 Microchip Technology Inc.

20.2 EUSART 异步模式

通过将 SYNC 位 （TXSTAx<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， 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 首先发送和接收的是最低有效位。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和BRG16位 （TXSTAx<2> 和 BAUDCONx<3>）的设置值产生两种不同的波特率时钟，频率分别为移比特率的 16 倍和 64 倍。硬件不支持奇偶校验，但可以用软件实现，校验位保存在第 9 个数据位。

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

20.2.1 EUSART 异步发送器

图 20-3 显示了 EUSART 发送器的框图。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 TXREGx 中获取数据，而 TXREGx 寄存器中的数据由软件装入。直到前一次装入的数据的停止位已被发送， 才会向 TSR 寄存器装入新数据。一旦停止位发送完毕，TXREGx 寄存器中的新数据（如果有的话）就会被装入 TSR。

一旦 TXREGx 寄存器向 TSR 寄存器传输了数据 （在 1 个 TCY 内发生）， TXREGx 寄存器就为空，同时标志位 TXxIF 置 1。可以通过将中断允许位 TXxIE 置 1 或清零来允许 / 禁止该中断。不管 TXxIF 的状态如何，只要中断发生，TXxIE 就会置 1 并且不能用软件清零。TXxIF 也不会在装载 TXREGx 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 TXREGx 装入新数据后立即查询 TXxIF，会返回无效结果。

标志位 TXxIF 表示的是 TXREGx 寄存器的状态，而另一个位 TRMT （TXSTAx<1>）则表示 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，用户只能对其进行查询。

注 1： TSR 寄存器并未映射到数据存储器中，因此用户不能访问它。

2： 当使能位 TXEN 置 1 时，标志位 TXxIF 也置 1。

设置异步发送的操作步骤如下：

1. 选择合适的波特率，对SPBRGHx:SPBRGx寄存器进行初始化。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 使能异步串口。
3. 如果需要中断，将中断允许位 TXxIE 置 1。
4. 若需要发送 9 位数据，请将发送位 TX9 置 1。发送的第 9 位可以是地址位也可以是数据位。
5. 通过将 TXEN 位置 1 使能发送，此操作同时也会将 TXxIF 位置 1。
6. 如果选择发送9位数据，将第9位数据装入TX9D位。
7. 将数据装入 TXREGx 寄存器 （开始发送）。
8. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）置 1。

图 20-3： EUSART 发送框图

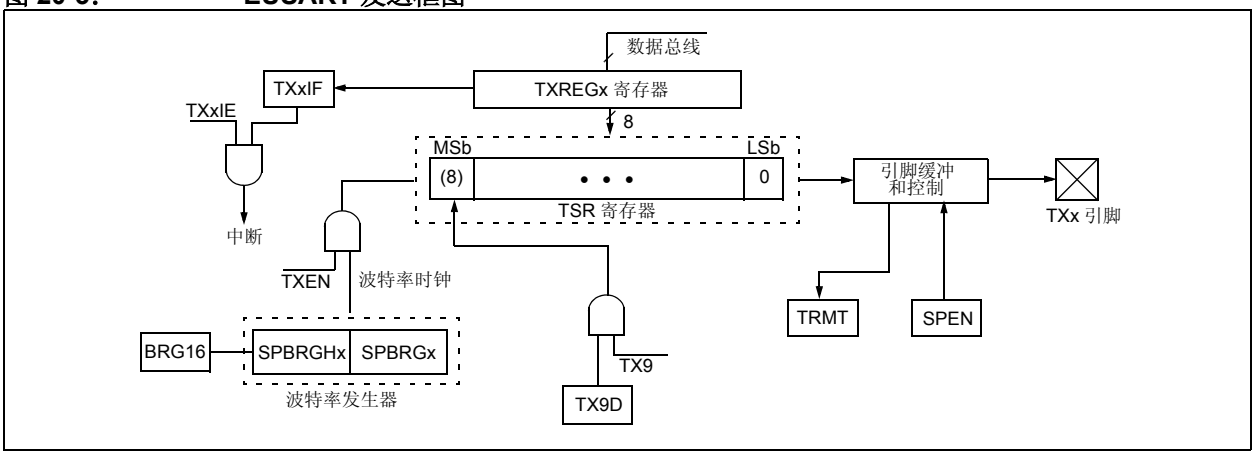


图 20-4：　　异步发送

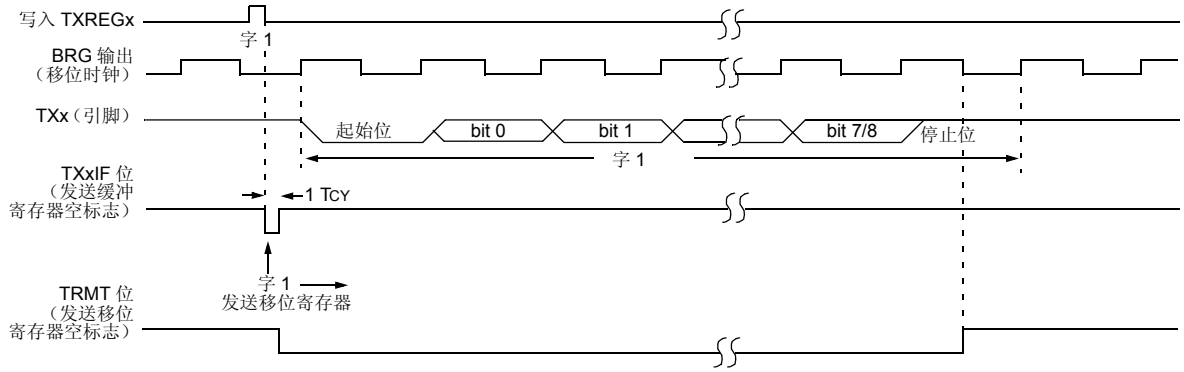
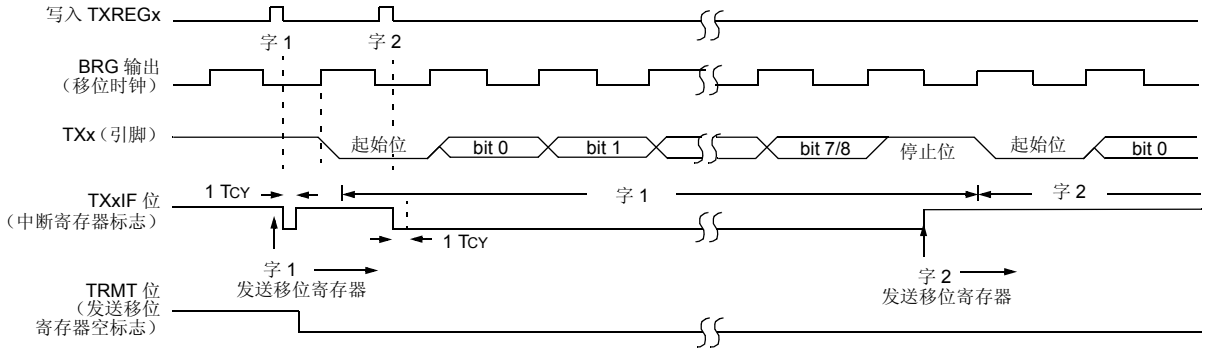


图 20-5：　　异步发送 （背靠背）



注：　此时序图表示的是两次连续的发送。

表 20-5：　　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	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
TXREGx	EUSARTx 发送寄存器								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								59
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								59

图注：　— = 未实现单元，读为 0。异步发送不使用阴影单元。

© 2007 Microchip Technology Inc.

20.2.2 EUSART 异步接收器

图 20-6 显示了接收器的框图。在 RXx 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以 16 倍波特率为工作频率的高速移位器，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 系统中。

设置异步接收操作的步骤如下：

1. 选择合适的波特率，对 SPBRGHx:SPBRGx 寄存器进行初始化。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 使能异步串口。
3. 如果需要中断，将允许位 RCxIE 置 1。
4. 若需要接收 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 RCxIF 将置 1，此时如果允许位 RCxIE 已置 1，将会产生一个中断。
7. 读 RCSTAx 寄存器以获取第 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. 通过读 RCREGx 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. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

20.2.3 设置带有地址检测功能的 9 位模式

此模式通常用于 RS-485 系统中。按如下步骤设置带有地址检测功能的异步接收操作：

1. 选择合适的波特率，对 SPBRGHx:SPBRGx 寄存器进行初始化。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 使能异步串口。
3. 若需要中断，请将 RCEN 位置 1 并使用 RCxIP 位设置优先级。
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 RCxIF 位将被置 1。此时如果 RCxIE 和 GIE 位已置 1，还将响应中断。
8. 读 RCSTAx 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（如果有的话）。
9. 读 RCREGx 以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. 如果已经找到了器件，将 ADDEN 位清零，允许接收到的所有数据进入接收缓冲器，并中断 CPU。

图 20-6： EUSART 接收框图

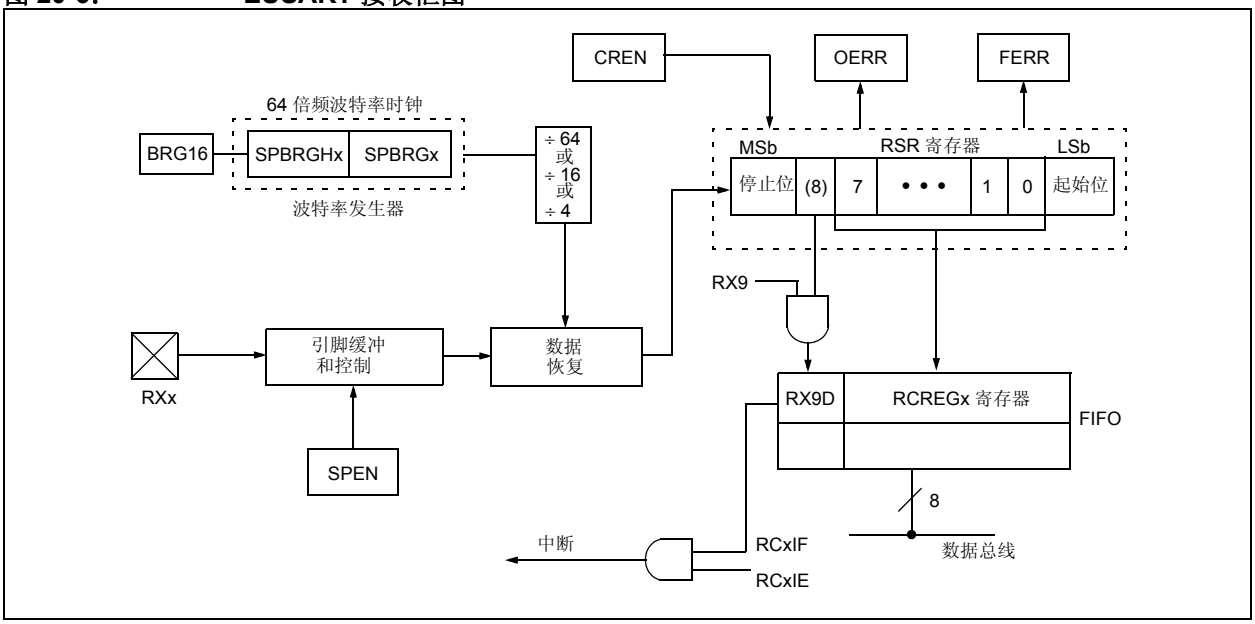
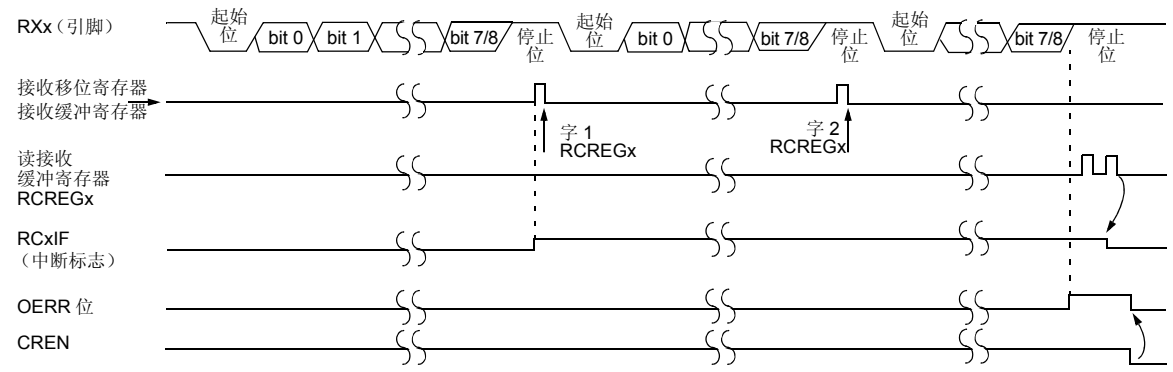


图 20-7：　　异步接收



注： 此时序图显示了在 RXx 输入引脚上顺序接收的 3 个字。在第 3 个字后读取 RCREGx（接收缓冲器）会使 OERR（溢出）位置 1。

表 20-6：　　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	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREGx	EUSARTx 接收寄存器								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								59
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								59

图注： — = 未实现单元，读为 0。异步接收不使用阴影单元。

20.2.4　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，波特率发生器处于不活动状态，并且无法进行正确的字节接收。自动唤醒功能允许在 RXx/DTx 线上有活动时唤醒控制器，该功能需要 EUSART 工作在异步模式下。

通过将 WUE 位（BAUDCONx<1>）置 1，使能自动唤醒功能。该功能启用后，将禁止 RXx/DTx 上的典型接收操作，且 EUSART 保持在空闲状态并监视唤醒事件（与 CPU 运行模式无关）。唤醒事件是指 RXx/DTx 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 RCxIF 中断。在正常工作模式下，中断会与 Q 时钟同步产生（图 20-8）；如果器件处于休眠模式，则两者不同步（图 20-9）。通过读 RCREGx 寄存器可清除中断条件。

唤醒事件后，当 RXx 线上出现由低电平向高电平的转换时，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。

© 2007 Microchip Technology Inc.

20.2.4.1 使用自动唤醒时需特别注意的事项

因为自动唤醒功能是通过检测RXx/DTx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，为了确保正确的传输，必须首先发送全 0 字符。对于标准的 RS-232 器件，这可以是 00h （8 位），而对于 LIN 总线则是 000h （12 位）。

另外还必须考虑振荡器起振时间，尤其在采用起振延时较长的振荡器 （即， HS 或 HSPLL 模式）的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，以便使振荡器有充足的时间起振并保证 EUSART 正确初始化。

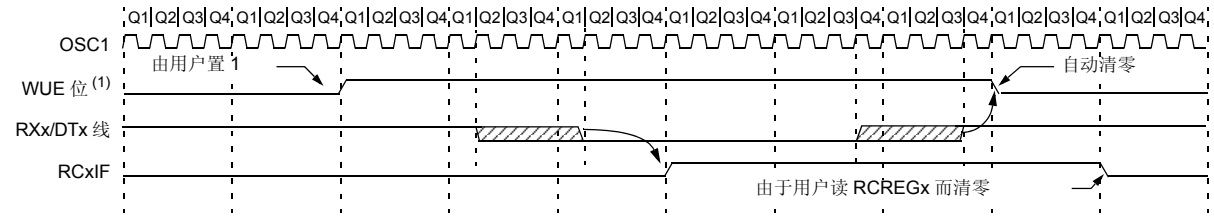
20.2.4.2 使用 WUE 位时的特别注意事项

用 WUE 和 RCxIF 事件的时序来判断接收数据的有效性可能会引起混淆。如前所述，将 WUE 位置 1 会使 EUSART 进入空闲模式。唤醒事件会产生一个接收中断，并将 RCxIF 位置 1。此后当 RXx/DTx 出现上升沿时 WUE 位清零。然后可以通过读 RCREGx 寄存器清除中断条件。一般情况下， RCREGx 中的数据是无效数据，应该丢弃。

WUE 位清零 （或仍然置 1）且 RCxIF 标志位置 1 并不能表明 RCREGx 中数据是完整的。用户还应该考虑使用固件验证是否完整地接收了数据。

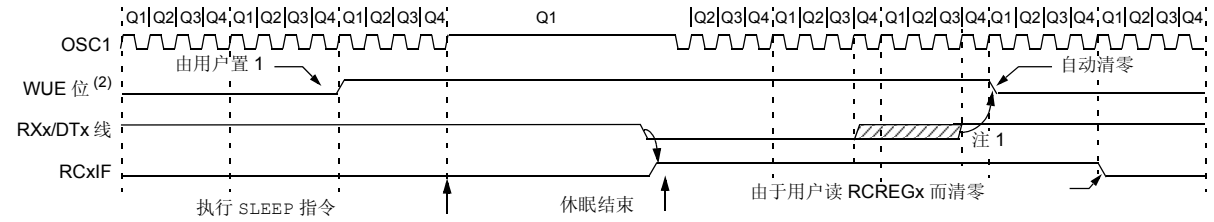
要确保没有丢失有效数据，应检查RCIDL位来验证是否还在接收数据。如果不再进行接收，则可将 WUE 位置 1，使器件立即进入休眠模式。

图 20-8： 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 20-9： 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，在振荡器就绪之前， WUE 位可以被自动清零。此过程与 Q 时钟出现与否无关。

2： WUE 位置 1 时， EUSART 保持在空闲状态。

20.2.5 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（TXSTAx<3> 和 TXSTAx<5>）置 1，就会发送帧间隔字符。请注意写入 TXREGx 的数据值会被忽略，并会发送全 0。

在发送了停止位后，硬件会自动将 SENDB 位清零。这样用户可以在发送完间隔字符（在 LIN 规范中通常是同步字符）后将下一个要发送的字节预先装入发送 FIFO。

请注意发送间隔字符时写入 TXREGx 的数据值会被忽略。写入仅仅是为了启动正确的序列。

如其在正常发送过程中一样，TRMT 位表明发送正在进行还是处于空闲状态。关于间隔字符时序，请参见图 20-10。

20.2.5.1 间隔和同步发送序列

下列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 TXREGx，启动发送（该值会被忽略）。
4. 将 55h 写入 TXREGx，以便把同步字符装入发送 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 TXxIF 指出 TXREGx 为空以后，下一个数据字节会被写入 TXREGx。

20.2.6 接收间隔字符

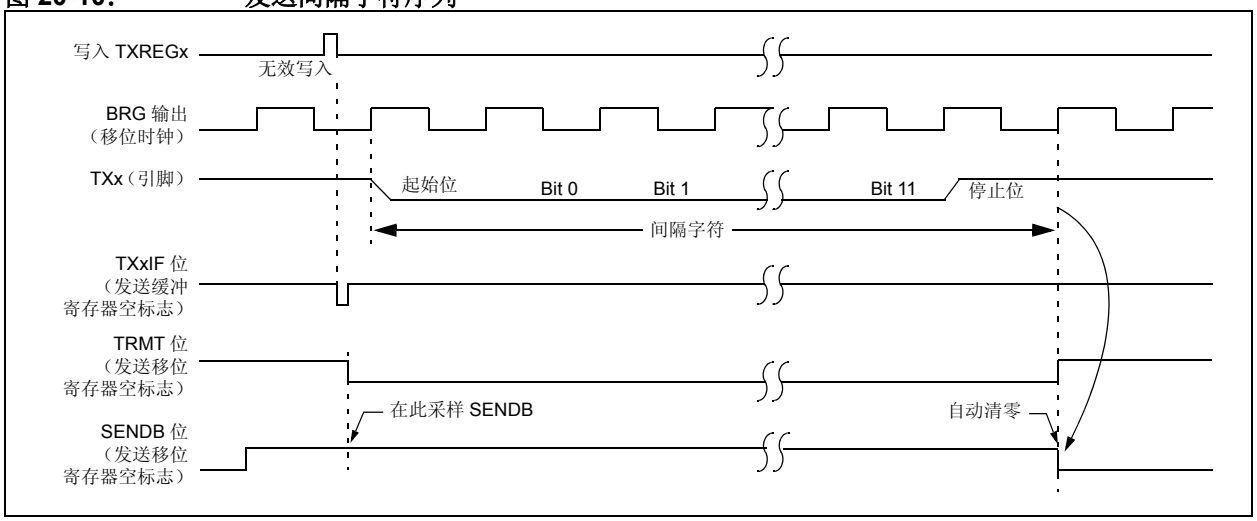
增强型 USART 模块接收间隔字符有两种方法。

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。

第二种方法使用**第 20.2.4 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 RXx/DTx 引脚上电平的下两次跳变，产生一个 RCxIF 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 TXxIF 中断时马上将 ABDEN 位置 1。

图 20-10： 发送间隔字符序列



© 2007 Microchip Technology Inc.

20.3 EUSART 同步主模式

将 CSRC 位（TXSTAx<7>）置 1 可以进入同步主模式。在此模式中，数据以半双工方式（即发送和接收不同时进行）发送。发送数据时，禁止接收，反之亦然。将 SYNC 位（TXSTAx<4>）置 1 可进入同步模式。此外，应将使能位 SPEN （RCSTAx<7>）置 1，把 TXx 和 RXx 引脚分别配置为 CKx（时钟）和 DTx（数据）线。

主模式意味着处理器在 CKx 时钟线上发送主控时钟信号。时钟极性是通过 SCKP 位 （BAUDCONx<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 CKx 设为低电平。此选项支持将本模块与 Microwire 器件配合使用。

20.3.1 EUSART 同步主发送

图 20-3 显示了 EUSART 发送器的框图。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 TXREGx 中获取数据，而 TXREGx 寄存器中的数据由软件装入。在前一次装入数据的最后一位发送完成后，才会向 TSR 寄存器装入新数据。一旦最后一位发送完成，就会将 TXREGx 寄存器中的新数据 （如果有的话）装入 TSR。

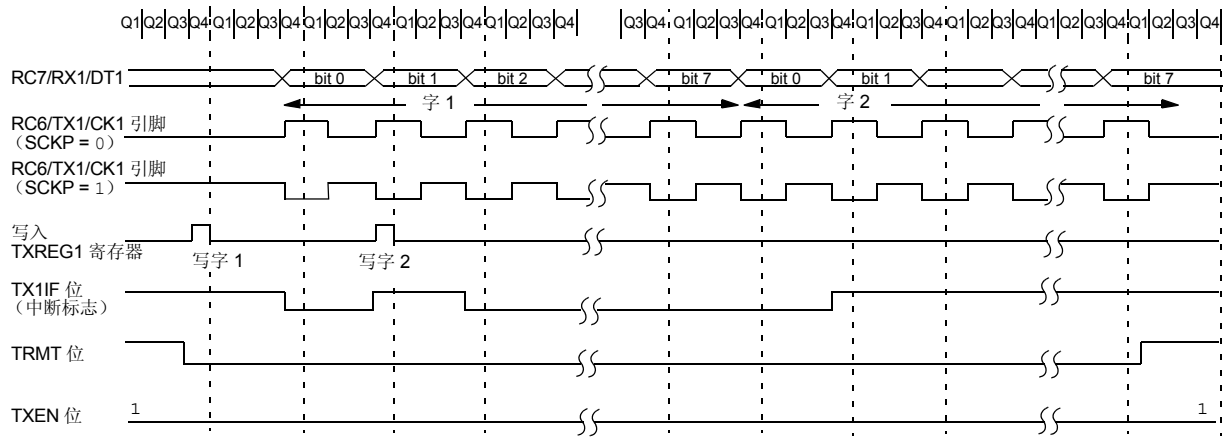
一旦 TXREG 寄存器向 TSR 寄存器传输了数据 （在 1 个 TCY 内发生）， TXREG 寄存器就为空，同时标志位 TXxIF 置 1。可以通过将中断允许位 TXxIE 置 1 或清零来允许 / 禁止该中断。不管 TXxIE 的状态如何，只要中断发生，TXxIF 就会置 1 并且不能用软件清零。只有在新数据写入 TXREGx 寄存器时， TXxIF 才会复位。

TXxIF 标志位表示的是 TXREGx 寄存器的状态，而另一个标志位 TRMT （TXSTAx<1>）则表示 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，因此要判断 TSR 寄存器是否为空，用户只能对此位进行查询。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。

设置同步主发送操作的步骤如下：

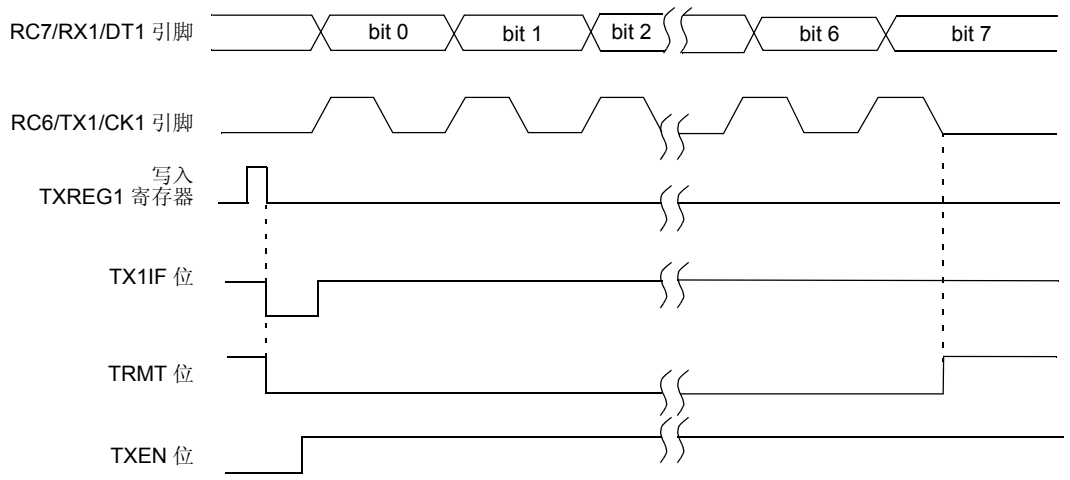
1. 选择合适的波特率，对 SPBRGHx:SPBRGx 寄存器进行初始化。按需要将 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC、SPEN 和 CSRC 位置 1，使能同步主控串口。
3. 如果需要中断，将中断允许位 TXxIE 置 1。
4. 若需要发送 9 位数据，将 TX9 位置 1。
5. 将 TXEN 位置 1，使能发送。
6. 如果选择发送 9 位数据，将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREG 寄存器，启动发送。
8. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）置 1。

图 20-11： 同步发送



注： 同步主模式 （SPBRGx = 0），连续发送两个 8 位字节。该示例同样适用于 EUSART2 （RG1/TX2/CK2 和 RG2/RX2/DT2）。

图 20-12：　　同步发送 （由 TXEN 位控制）



注： 该示例同样适用于 EUSART2 （RG1/TX2/CK2 和 RG2/RX2/DT2）。

表 20-7：　　与同步主发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
TXREGx	EUSARTx 发送寄存器								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	ó	WUE	ABDEN	59
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								59
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								59

图注：　　**—** = 未实现 （读为 0）。同步主发送不使用阴影单元。

© 2007 Microchip Technology Inc.

20.3.2 EUSART 同步主接收

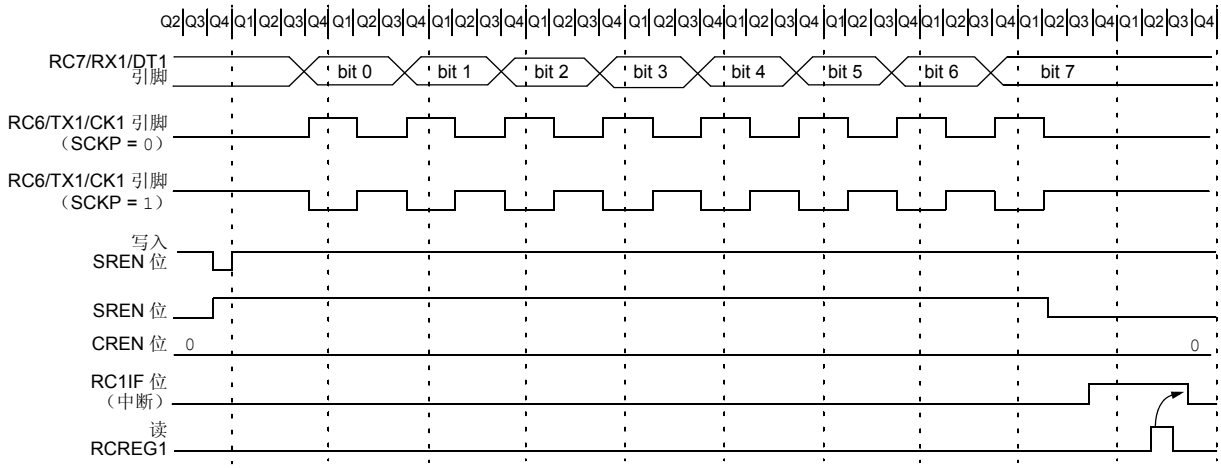
一旦选择了同步模式，只要将单字节接收使能位 SREN（RCSTAx<5>）或连续接收使能位 CREN（RCSTAx<4>）置 1，即可使能接收。在时钟的下降沿采样 RXx 引脚上的数据。

如果将使能位 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 具有优先权。

设置同步主接收操作的步骤如下：

1. 选择合适的波特率，对 SPBRGHx:SPBRGx 寄存器进行初始化。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，使能同步主控串口。
3. 确保将 CREN 和 SREN 位清零。
4. 如果需要中断，将允许位 RCxIE 置 1。
5. 若需要接收 9 位数据，将 RX9 位置 1。
6. 若需要单字节接收，将 SREN 位置 1； 若需要连续接收，将 CREN 位置 1。
7. 当接收完成时中断标志位 RCxIF 将置 1，此时如果允许位 RCxIE 已置 1，还将产生一个中断。
8. 读 RCSTAx 寄存器以获取第 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. 通过读 RCREGx 寄存器来读取接收到的 8 位数据。
10. 如果发生错误，通过将 CREN 位清零来清除错误。
11. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）置 1。

图 20-13： 主模式同步接收 （由 SREN 位控制）



注： 所示时序图为 SREN = 1 并且 BRGH = 0 时的同步主模式。该示例同样适用于 EUSART2 （RG1/TX2/CK2 和 RG2/RX2/DT2）。

表 20-8： 与同步主接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREGx	EUSARTx 接收寄存器								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								59
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								59

图注： — = 未用 （读为 0）。同步主接收不使用阴影单元。

20.4 EUSART 同步从模式

将 CSRC（TXSTAx<7>）位清零可进入同步从模式。此模式与同步主模式的区别在于移位时钟由 CKx 引脚上的外部时钟提供（主模式下由内部时钟提供）。这使得器件能在任何低功耗模式下发送或接收数据。

20.4.1 EUSART 同步从发送

除了休眠模式以外，同步主模式和从模式的工作方式是完全相同的。

如果向 TXREGx 寄存器写入两个字，然后执行 SLEEP 指令，则将发生以下事件：

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 TXREGx 寄存器中。
c) 不会将标志位 TXxIF 置 1。
d) 当第一个字移出 TSR 后， TXREGx 寄存器将把第二个字送入 TSR，同时将标志位 TXxIF 置 1。
e) 如果中断允许位 TXxIE 已置 1，中断将把器件从休眠状态唤醒。如果允许全局中断，程序将会跳转到中断向量处执行。

设置同步从发送操作的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步从串口。
2. 将 CREN 和 SREN 位清零。
3. 如果需要中断，将中断允许位 TXxIE 置 1。
4. 若需要发送 9 位数据，将 TX9 位置 1。
5. 将中断允许位 TXEN 置 1，使能发送。
6. 如果选择发送 9 位数据，将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREGx 寄存器，开始发送。
8. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）置 1。

表 20-9： 与同步从发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
TXREGx	EUSARTx 发送寄存器								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								59
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								59

图注： — = 未实现 （读为 0）。同步从发送不使用阴影单元。

© 2007 Microchip Technology Inc.

20.4.2 EUSART 同步从接收

除了休眠模式、空闲模式以及在从模式下忽略 SREN 位以外，同步主和同步从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，那么在该低功耗模式下可以接收到一个数据字。一接收到该数据字，RSR寄存器就将数据传送到RCREGx寄存器。如果 RCxIE 允许已置 1，则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 位清零使能同步主串口。
2. 如果需要中断，将允许位 RCxIE 置 1。
3. 若需要接收 9 位数据，将 RX9 位置 1。
4. 将使能位 CREN 置 1，使能接收。
5. 当接收完成时，RCxIF 标志位将被置 1。如果允许位 RCxIE 已置 1，还将产生一个中断。
6. 读 RCSTAx 寄存器以获取第 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. 通过读 RCREGx 寄存器来读取接收到的 8 位数据。
8. 如果发生错误，通过将 CREN 位清零来清除错误。
9. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 （INTCON<7:6>）置 1。

表 20-10： 与同步从接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	58
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	58
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	58
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREGx	EUSARTx 接收寄存器								57
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCONx	ABDOVF	RCIDL	DTRXP	SCKP	BRG16	—	WUE	ABDEN	59
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								59
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								59

图注： **— =** 未实现 （读为 0）。同步从接收不使用阴影单元。

注：

 © 2007 Microchip Technology Inc.

21.0 10 位模数转换器 （A/D）模块

64 引脚器件的模数（A/D）转换器模块具有 11 路输入，而 80 引脚器件的模数转换器模块则具有 15 路输入。A/D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 6 个寄存器：

- A/D 控制寄存器 0 （ADCON0）
- A/D 控制寄存器 1 （ADCON1）
- A/D 端口配置寄存器 2 （ANCON0）
- A/D 端口配置寄存器 1 （ANCON1）
- A/D 结果寄存器 （ADRESH 和 ADRESL）

如寄存器 21-1 所示，A/D 模块的工作由 ADCON0 寄存器控制。如寄存器 21-2 所示，由 ADCON1 寄存器配置 A/D 时钟源，编程采集时间和对齐方式。

寄存器 21-1： ADCON0：A/D 控制寄存器 0(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VCFG1	VCFG0	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-6 **VCFG1**：参考电压配置位 （VREF- 参考电压源）
1 = VREF- （AN2）
0 = AVSS

bit **VCFG0**：参考电压配置位 （VREF+ 参考电压源）
1 = VREF+ （AN3）
0 = AVDD

bit 5-2 **CHS3:CHS0**：模拟通道选择位
0000 = 通道 00 （AN0）
0001 = 通道 01 （AN1）
0010 = 通道 02 （AN2）
0011 = 通道 03 （AN3）
0100 = 通道 04 （AN4）
0101 = 未使用
0110 = 通道 06 （AN6）
0111 = 通道 07 （AN7）
1000 = 通道 08 （AN8）
1001 = 通道 09 （AN9）
1010 = 通道 10 （AN10）
1011 = 通道 11 （AN11）
1100 = 通道 12 （AN12）(2,3)
1101 = 通道 13 （AN13）(2,3)
1110 = 通道 14 （AN14）(2,3)
1111 = 通道 15 （AN15）(2,3)

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： 当 WDTCON<4> = 0 时，此处的默认 （旧） SFR 可用。
2： 这些通道在 64 引脚器件上未用。
3： 在未用通道上执行转换会返回随机值。

寄存器 21-2： ADCON1：A/D 控制寄存器 1(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	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 **ADCAL**：A/D 校准位
1 = 在下一次 A/D 转换时进行校准
0 = 正常的 A/D 转换器操作 （没有进行转换）

bit 5-3 **ACQT2:ACQT0**：A/D 采集时间选择位
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 RC 振荡器）(2)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC （时钟来自 A/D RC 振荡器）(2)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

注 1： 当 WDTCON<4> = 0 时，此处的默认 （旧） SFR 可用。
2： 如果选择了 A/D FRC 时钟源，在 A/D 时钟启动之前会添加一个 TCY （指令周期）的延时。这允许在开始转换之前执行 SLEEP 指令。

ANCON0 和 ANCON1 寄存器用于配置与每个模拟通道相关的 I/O 引脚的操作。将任何一个 PCFG 位置 1 可将相应引脚配置为以仅数字 I/O 运行。清零任何一个 PCFG 位可将相应引脚配置为 A/D 转换器或比较器模块的模拟输入；禁止所有数字外设，且数字输入读为 0。通常，与模拟输入复用的 I/O 引脚，在器件复位时默认为模拟功能。

ANCON0 和 ANCON1 为共享地址的 SFR，并且与 ADCON1 和 ADCON0 寄存器使用相同的地址。通过将 ADSHR 位（WDTCON<4>）置 1 可访问 ANCON 寄存器。更多信息，请参见**第 5.3.4.1 节“共享地址 SFR”**。

寄存器 21-3： ANCON0：A/D 端口配置寄存器 2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-6 **PCFG7:PCFG6：**模拟端口配置位 （AN7 和 AN6）
1 = 引脚配置为数字端口
0 = 引脚配置为模拟通道；数字输入被禁止且读为 0

bit 5 **未实现：**读为 0

bit 4-0 **PCFG4:PCFG0：**模拟端口配置位 （AN4 至 AN0）
1 = 引脚配置为数字端口
0 = 引脚配置为模拟通道；数字输入被禁止且读为 0

寄存器 21-4： ANCON1：A/D 端口配置寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15(1)	PCFG14(1)	PCFG13(1)	PCFG12(1)	PCFG11	PCFG10	PCFG9	PCFG8
bit 7							bit 0

图注：			
R= 可读位	W= 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x= 未知

bit 7-0 **PCFG15:PCFG8：**模拟端口配置位 （AN15 至 AN8）
1 = 引脚配置为数字端口
0 = 引脚配置为模拟通道；数字输入被禁止且读为 0

注 1： 只有在 80 引脚器件中 AN12 到 AN15 才可用。对于 64 引脚器件，虽然为这些通道实现了相应的 PCFGx 位，但是这些位实际上不起任何作用。

可用软件选择模拟参考电压为器件的正电源电压和负电源电压（AVDD 和 AVSS）或 RA3/AN3/VREF+ 引脚和 RA2/AN2/VREF 引脚上的电平。

A/D 转换器具备可在休眠模式下工作的独特特性。要使 A/D 转换器在休眠模式下运行， A/D 转换时钟必须来自于 A/D 模块内部的 RC 振荡器。

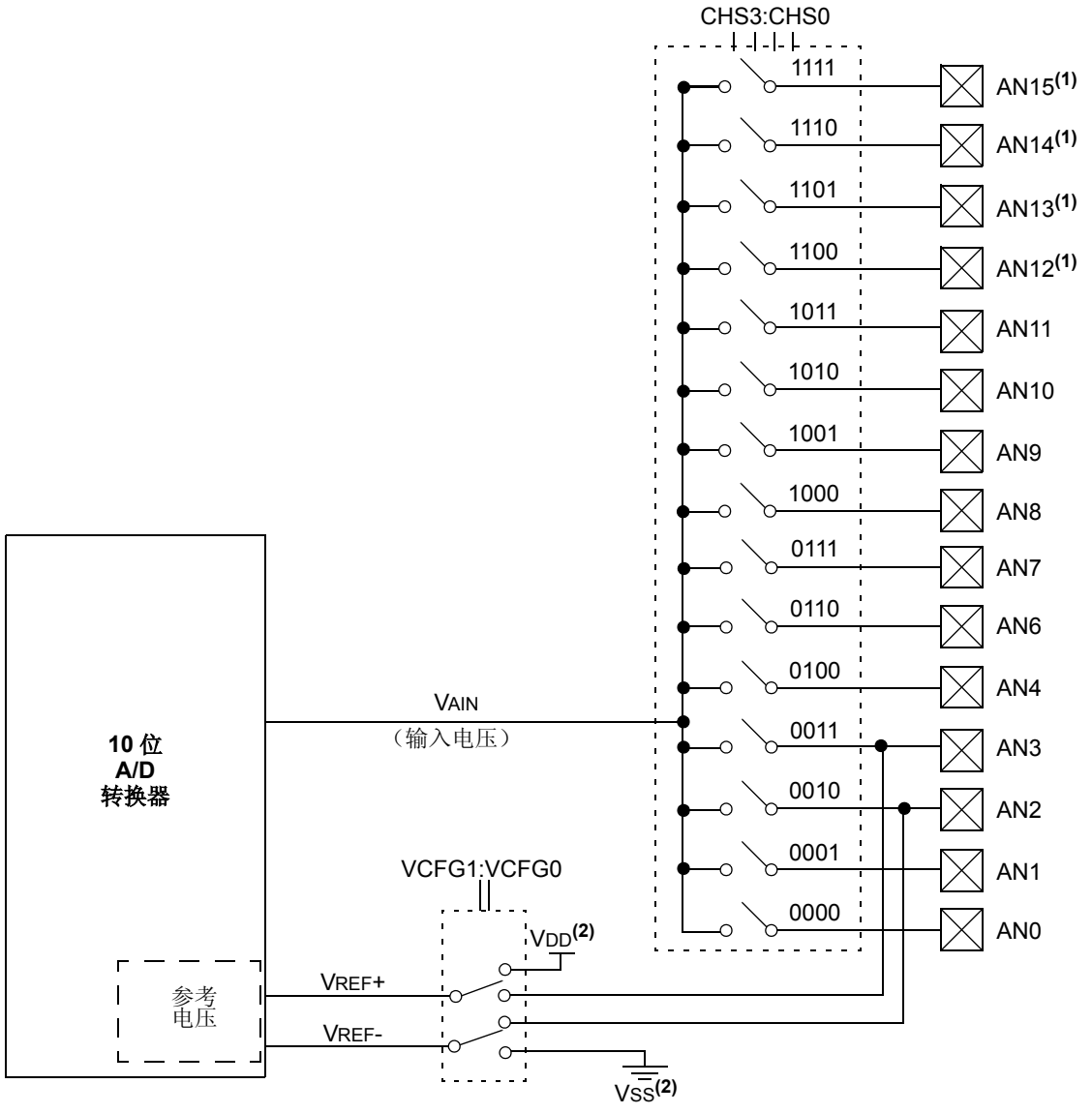
采样和保持输出输入到转换器，转换器采用逐次逼近法得到转换结果。

可以将每个与 A/D 转换器相关的端口引脚配置为模拟输入或数字 I/O。 ADRESH 和 ADRESL 寄存器包含 A/D 转换的结果。当 A/D 转换完成时，转换结果被载入 ADRESH:ADRESL 寄存器对， GO/DONE 位（ADCON0<1>）被清零且 A/D 中断标志位 ADIF 置 1。

器件复位强制所有寄存器进入复位状态。同时迫使 A/D 模块关闭并中止任何正在进行的转换。上电复位时，ADRESH:ADRESL 寄存器对中的值保持不变。上电复位后，这两个寄存器中的值不确定。

图 21-1 给出了 A/D 模块的框图。

图 21-1： A/D 框图



注 1：在 64 引脚器件上不存在通道 AN12 到 AN15。
2：I/O 引脚有连接到 VDD 和 VSS 的保护二极管。

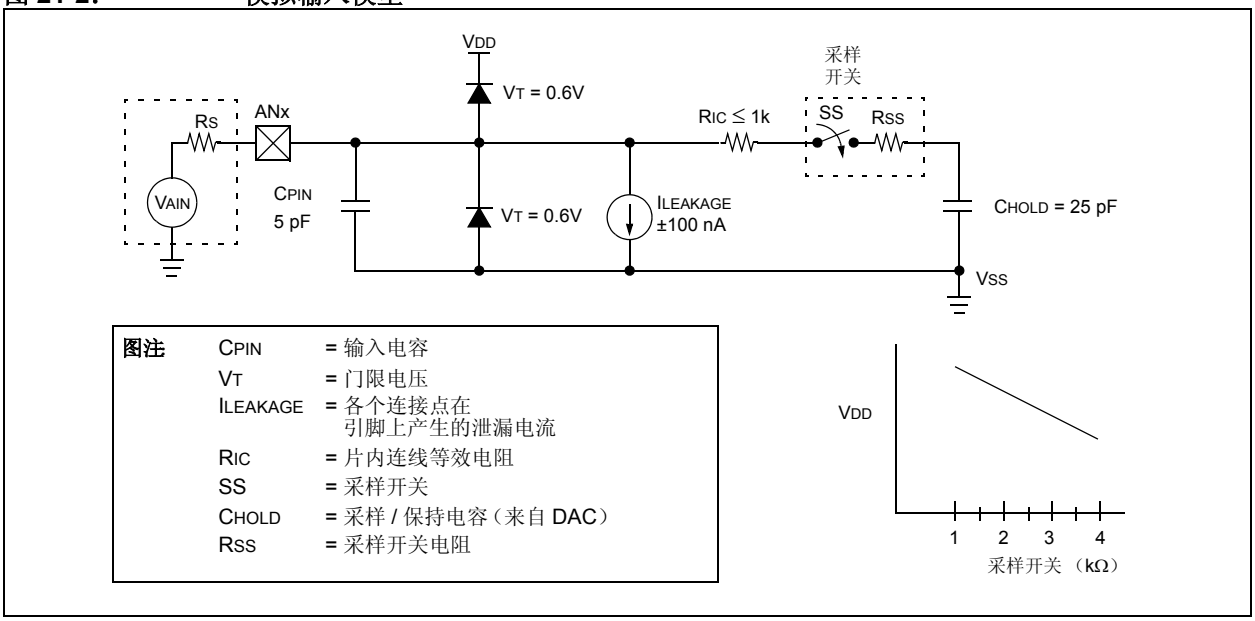
在根据需要配置好 A/D 模块之后，必须在转换开始之前对选定的通道进行采集。必须将模拟输入通道对应的 TRIS 位选择为输入。采集时间的确定请参见**第 21.1 节“A/D 采集要求”**。在采集完成之后， A/D 转换即可开始。可编程采集时间在 GO/DONE 位置 1 和转换实际开始的时间之间进行。

按照以下步骤进行 A/D 转换：

1. 配置 A/D 模块：
   - 将所需的 ADC 引脚配置为使用 ANCON0 和 ANCON1 的模拟引脚
   - 将参考电压设置为使用 ADCON0
   - 选择 A/D 输入通道 （ADCON0）
   - 选择 A/D 采集时间 （ADCON1）
   - 选择 A/D 转换时钟 （ADCON1）
   - 打开 A/D 模块 （ADCON0）
2. 需要时，配置 A/D 中断：
   - 清零 ADIF 位
   - 将 ADIE 位置 1
   - 将 GIE 位置 1
3. 如果需要，等待所需的采集时间。
4. 启动转换：
   - 将 GO/DONE 位 （ADCON0<1>）置 1
5. 等待 A/D 转换完成，通过以下两种方法之一可判断转换是否完成：
   - 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，请根据需要返回步骤 1 或步骤 2。每位的 A/D 转换时间定义为 TAD。在下一次采集开始前至少需要等待 2 个 TAD。

图 21-2： 模拟输入模型



21.1 A/D 采集要求

为了使 A/D 转换器达到规定精度，必须让充电保持电容（CHOLD）充满至输入通道的电平。图 21-2 给出了模拟输入模型。源阻抗（RS）和内部采样开关（RSS）阻抗直接影响给电容CHOLD充电所需要的时间。采样开关（RSS）阻抗随器件电压（VDD）不同而改变。源阻抗将影响模拟输入的偏置电压（由于引脚上的泄漏电流）。**模拟信号源的最大阻抗推荐值为 2.5 kΩ**。在选择（改变）了模拟输入通道之后，在转换开始之前，通道必须被采样，采样时间必须大于最小采集时间。

注：	当开始转换时，要将保持电容与输入引脚断开。

要计算最小采集时间，可使用公式 21-1。该公式假定的误差为 1/2 LSb（即 A/D 转换需要 1024 步）。1/2 LSb 的误差是 A/D 模块达到规定分辨率所能允许的最大误差。

公式 21-3 显示了所需的最小采集时间 TACQ 的计算过程。该计算基于对应用系统的以下假定：

CHOLD	=	25 pF
Rs	=	2.5 kΩ
转换误差	≤	1/2 LSb
VDD	=	3V → Rss = 2 kΩ
温度	=	85° C （系统最大值）

公式 21-1： 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 21-2： 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 21-3： 计算所需的最小采集时间

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$

$$T_{AMP} = 0.2\ \mu s$$

$$T_{COFF} = (Temp - 25°C)(0.02\ \mu s/°C)$$
$$(85°C - 25°C)(0.02\ \mu s/°C)$$
$$1.2\ \mu s$$

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。

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S) \ln(1/2048)\ \mu s$$
$$-(25\ pF)\ (1\ k\Omega + 2\ k\Omega + 2.5\ k\Omega) \ln(0.0004883)\ \mu s$$
$$1.05\ \mu s$$

$$T_{ACQ} = 0.2\ \mu s + 1.05\ \mu s + 1.2\ \mu s$$
$$2.45\ \mu s$$

21.2 选择和配置自动采集时间

每当 GO/$\overline{\text{DONE}}$ 位置 1 时，用户就可利用 ADCON1 寄存器选择采集时间。

当 GO/$\overline{\text{DONE}}$ 位置 1 时，采样停止并启动转换。用户有责任确保在选择所需要的输入通道和 GO/$\overline{\text{DONE}}$ 置 1 之间经过了必需的采集时间。这在 ACQT2:ACQT0 位（ADCON1<5:3>）保持在复位状态（000）时发生，并与不提供可编程采集时间的器件兼容。

如果需要，可将 ACQT 位置 1 来选择 A/D 模块的可编程采集时间。当 GO/$\overline{\text{DONE}}$ 位置 1 时，A/D 模块继续对输入进行采样，采样时间为所选择的采集时间，然后自动开始转换。由于采集时间已被编程，因此 GO/$\overline{\text{DONE}}$ 位会立即置 1 而不需要在选择通道以后等待一个采集时间。

在这两种情况下，当转换完成时，GO/$\overline{\text{DONE}}$位被清零、ADIF 标志位被置 1 且 A/D 开始再次采样当前选定的通道。如果采集时间已被编程，采集时间结束或转换开始都不会有任何指示。

21.3 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 转换时钟源。

TAD 有以下 7 种可能的选择：

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- 内部 RC 振荡器

要实现正确的 A/D 转换， A/D 转换时钟 （TAD）必须尽可能小，但它必须大于最小 TAD （如需了解更多信息，请参见表 27-30 中的参数 130）。

表 21-1 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 21-1： TAD 与器件工作频率关系表

AD 时钟源 (TAD)		最大器件频率
工作原理	ADCS2:ADCS0	
2 TOSC	000	2.86 MHz
4 TOSC	100	5.71 MHz
8 TOSC	001	11.43 MHz
16 TOSC	101	22.86 MHz
32 TOSC	010	40.00 MHz
64 TOSC	110	40.00 MHz
RC(2)	x11	1.00 MHz(1)

注 1： RC 源的典型 TAD 时间为 4 μs。

2： 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所允许的范围。

21.4 配置模拟端口引脚

ADCON0、ANCON1、TRISA、TRISF 和 TRISH 寄存器控制 A/D 端口引脚的操作。若希望端口引脚作为模拟输入，则必须将相应的 TRIS 位置 1 （输入）。如果将 TRIS 位清零 （输出），则将转换该引脚的数字输出电平 （VOH 或 VOL）。

A/D 转换操作与 CHS3:CHS0 位以及 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转换模拟输入信号。配置为数字输入的引脚上将模拟电平精确转换为数字电平。

2： 定义为数字输入的引脚上的模拟电平，可能会导致数字输入缓冲器消耗的电流超出器件规范。

21.5 A/D 转换

图 21-3 显示了在 GO/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 21-4 显示了在 GO/DONE 位置 1，ACQT2:CQT0 位被设置为 010，且在转换开始之前选择了 4 TAD 的采集时间后 A/D 转换器的工作状态。

在转换期间将 GO/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 寄存器将仍然保存上一次的转换结果（即上一次写入 ADRESH:ADRESL 寄存器中的值）。

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，将自动开始对选定通道进行采集。

注：	**不能**用启动 A/D 模块的同一指令将 GO/DONE 位置 1。

21.6 ECCP2 触发器的使用

可以通过 ECCP2 模块的“特殊事件触发信号”启动 A/D 转换。这要求将 CCP2M3:CCP2M0 位（CCP2CON<3:0>）编程为 1011，且使能 A/D 模块（ADON 位置 1）。发生触发事件时，GO/DONE 位将被置 1，启动 A/D 采集和转换，并将 Timer1（或 Timer3）计数器复位为 0。复位 Timer1（或 Timer3）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 的内容移到目标单元）。在“特殊事件触发信号”将 GO/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，并通过手动或者选择相应的 TACQ 时间来设定最小采样时间。

如果未使能 A/D 模块（ADON 清零），则 A/D 模块将忽略“特殊事件触发信号”，但它仍会将 Timer1（或 Timer3）计数器复位。

图 21-3： A/D 转换 TAD 周期（ACQT2:ACQT0 = 000， TACQ = 0）

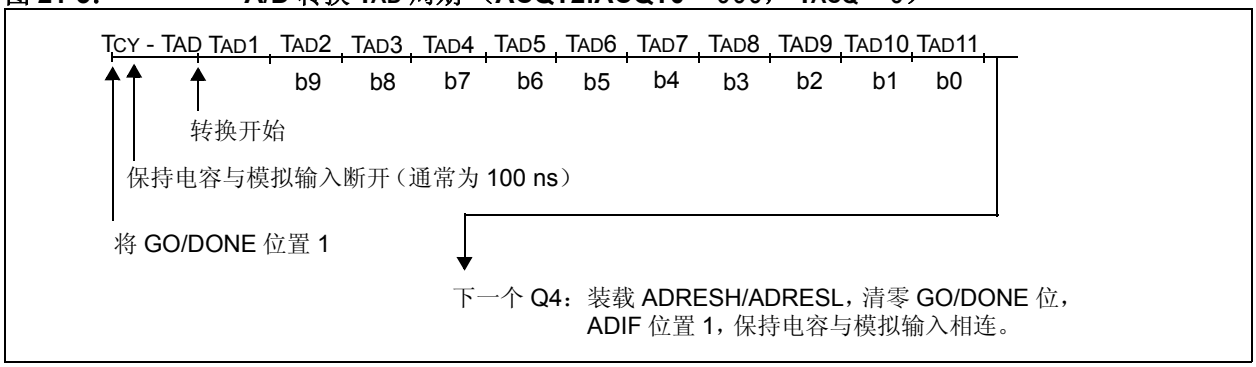
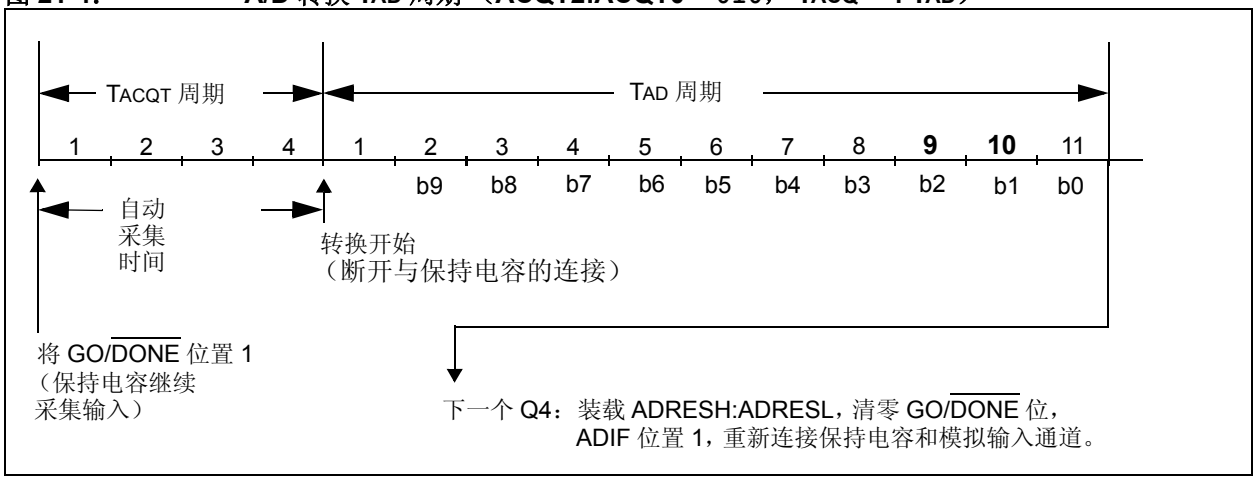


图 21-4： A/D 转换 TAD 周期（ACQT2:ACQT0 = 010， TACQ = 4 TAD）



21.7 A/D 转换器校准

PIC18F87J11 系列器件中的 A/D 转换器具有自校准功能，它可以补偿模块中所产生的任何偏移。校准过程是自动进行的，并可通过将 ADCAL 位（ADCON0<6>）置 1 来开始校准。下一次 GO/DONE 位置 1 时，此模块将执行“无效”转换（即，在输入通道上读不到任何数据）并将转换结果存储在内部以补偿偏移。因此，后续的偏移都将得到补偿。例 21-1 给出了一个校准程序示例。

在校准过程中假设器件处于相对稳态工作状态。如果使用 A/D 校准，则应在每次器件复位之后或在工作条件发生其他重要变化时执行。

21.8 在功耗管理模式下的工作方式

在功耗管理模式下，自动采集时间和 A/D 转换时钟的选择的一部分是由时钟源和频率决定的。

如果要在器件处于功耗管理模式时进行 A/D 转换，ADCON2 中的 ACQT2:ACQT0 和 ADCS2:ADCS0 位就应该根据将使用的功耗管理模式时钟进行更新。在进入功耗管理模式之后（两种功耗管理运行模式中的一种），就会开始 A/D 采集或转换。采集或转换开始以后，器件仍应继续使用与功耗管理模式相同的时钟源直到转换完成。如果需要的话，在转换期间也可以将器件置于相应的功耗管理空闲模式。

如果功耗管理模式时钟频率小于 1 MHz，就应该选择 A/D RC 时钟源。

在休眠模式下工作也需要选择 A/D RC 时钟源。如果 ACQT2:ACQT0 位被置为 000 且转换开始，转换将被延时一个指令周期以允许执行 SLEEP 指令并进入休眠模式。OSCCON 寄存器中的 IDLEN 和 SCS 位必须在转换开始之前清零。

例 21-1： A/D 校准程序示例

```
        BSF     WDTCON,ADSHR        ;Enable write/read to the shared SFR
        BCF     ANCON0,PCFG0        ;Make Channel 0 analog
        BCF     WDTCON,ADSHR        ;Disable write/read to the shared SFR
        BSF     ADCON0,ADON         ;Enable A/D module
        BSF     ADCON1,ADCAL        ;Enable Calibration
        BSF     ADCON0,GO           ;Start a dummy A/D conversion
CALIBRATION                         ;
        BTFSC   ADCON0,GO           ;Wait for the dummy conversion to finish
        BRA     CALIBRATION         ;
        BCF     ADCON1,ADCAL        ;Calibration done, turn off calibration enable
                                    ;Proceed with the actual A/D conversion
```

表 21-2： A/D 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 所在的页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IF	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PMPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	58
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
ADRESH	A/D 结果寄存器的高字节								57
ADRESL	A/D 结果寄存器的低字节								57
ADCON0(2)	VCFG1	VCFG0	CHS3	CHS3	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	57
ANCON0(3)	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57
ADCON1(2)	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	57
ANCON1(3)	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	57
PORTA	RA7(4)	RA6(4)	RA5	RA4	RA3	RA2	RA1	RA0	59
TRISA	TRISA7(4)	TRISA6(4)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	59
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	58
PORTH(1)	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	59
TRISH(1)	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	58

图注： — = 未实现，读为 0。 A/D 转换不使用阴影单元。

注 1： 64 引脚器件上不存在该寄存器。

2： 当 WDTCON<4> = 0 时，此处的默认 （旧） SFR 可用。

3： 配置 SFR 的地址与默认 SFR 重叠；仅在 WDTCON<4> = 1 时才可访问配置 SFR。

4： 这些位仅在选择振荡器模式 （FOSC2 配置位 = 0）时才可用；否则，它们都处于未用状态。

22.0 比较器模块

模拟比较器模块包含两个比较器，可以用多种方式对它们进行配置。比较器的输入可以是模拟输入和两个内部参考电压。数字输出可从引脚电平获取也可通过控制寄存器读取。比较器还具有多路输出和中断事件生成功能。图 22-1 所示为该模块的通用单比较器。

该模块的主要功能有：

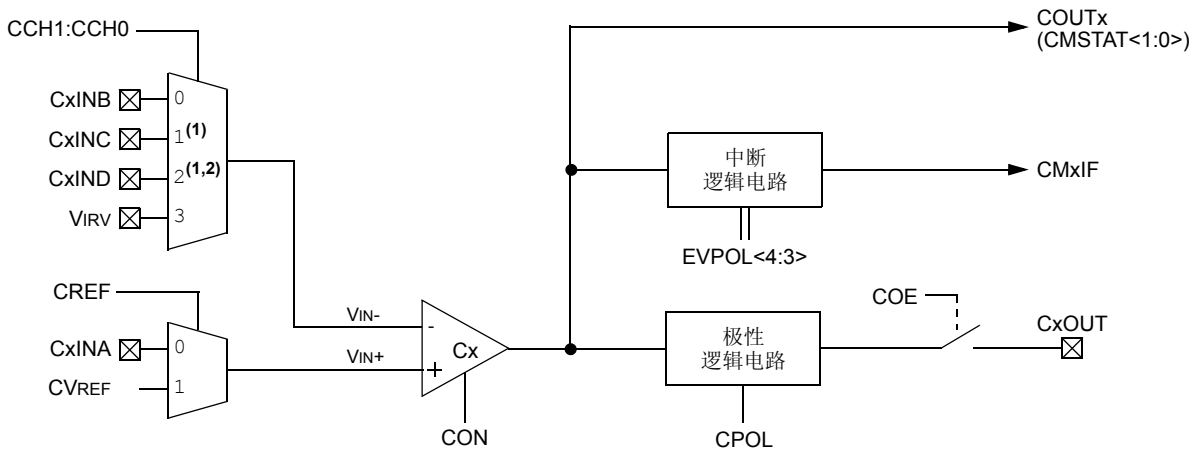
- 独立的比较器控制
- 可编程的输入配置
- 输出至引脚和寄存器电平
- 可编程的输出极性
- 对每个具有可配置的电平变化中断的比较器，可产生独立中断

22.1 寄存器

CMxCON 寄存器（寄存器 22-1）选择各比较器的输入和输出配置以及产生中断的设置。

CMSTAT 寄存器（寄存器 22-2）提供比较器的输出结果。寄存器中的这些位是只读的。

图 22-1： 比较器简化框图



注 1： 仅在 80 引脚的器件中可用。

2： 仅在比较器 2 中实现。

寄存器 22-1： CMxCON：比较器 x 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **CON**：比较器使能位
1 = 使能比较器
0 = 禁止比较器

bit 6 **COE**：比较器输出使能位
1 = 比较器在 CxOUT 引脚上进行输出
0 = 仅在内部进行比较器输出

bit 5 **CPOL**：比较器输出极性选择位
1 = 比较器输出反相
0 = 比较器输出正相

bit 4-3 **EVPOL1:EVPOL0**：中断极性选择位
11 = 输出发生任何变化时产生中断 (1)
10 = 在输出电平的下降沿产生中断
01 = 在输出电平的上升沿产生中断
00 = 禁止产生中断

bit 2 **CREF**：比较器参考选择位 （同相输入）
1 = 同相输入与内部 CV_{REF} 电压连接
0 = 同相输入与 CxINA 引脚连接

bit 1-0 **CCH1:CCH0**：比较器通道选择位
11 = 比较器的反相输入与 V_{IRV} 连接
10 = 比较器的反相输入与 CxIND 引脚连接 (2)
01 = 比较器的反相输入与 CxINC 引脚连接 (2)
00 = 比较器的反相输入与 CxINB 引脚连接

注 1： 每次选择此模式时， CMxIF 会自动置位，且初次配置完成后该位必须由应用程序清零。
2： 仅在 80 引脚的器件中可用。

© 2007 Microchip Technology Inc.

寄存器 22-2：　　CMSTAT：比较器输出状态寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R-1	R-1
—	—	—	—	—	—	COUT2	COUT1
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2　**未实现位：**读为 0

bit 1-0　**COUT2:COUT1：**比较器 x 状态位

如果 CPOL = 0 （极性相同）：

1 = 比较器 $V_{IN+} > V_{IN-}$

0 = 比较器 $V_{IN+} < V_{IN-}$

如果 CPOL = 1 （极性相反）：

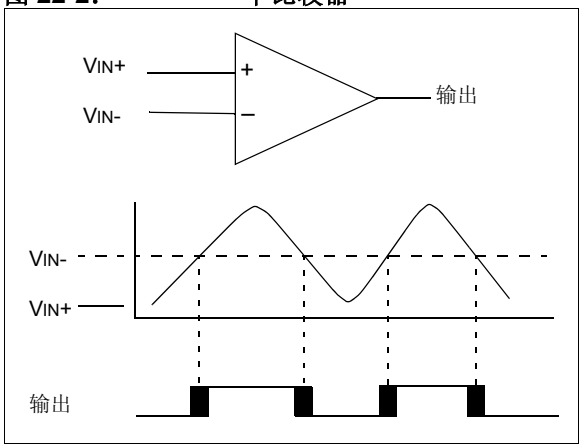
1 = 比较器 $V_{IN+} < V_{IN-}$

0 = 比较器 $V_{IN+} > V_{IN-}$

22.2 比较器工作原理

图22-2所示为单个比较器以及模拟输入电平和数字输出之间的关系。如果 VIN+ 上的模拟输入小于 VIN- 上的模拟输入，那么比较器将输出数字低电平。当 VIN+ 上的模拟输入高于 VIN- 上的模拟输入时，比较器输出数字高电平。图 22-2 中比较器输出的阴影部分表示因输入失调和响应时间所造成的不确定区。

图 22-2： 单比较器



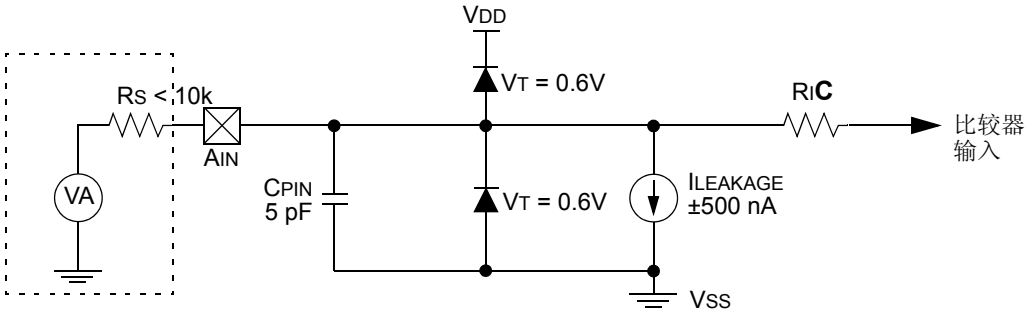
22.3 比较器响应时间

响应时间是指比较器从选定一个新的参考电压或输入源到输出达到一个有效电平时的最短时间。比较器的响应时间与参考电压的稳定时间不同。因此，在确定对比较器输入变化的总体响应时必须同时考虑以上两个时间。否则，应使用比较器延时的最大值（见**第 27.0 节“电气规范”**）。

22.4 模拟输入连接注意事项

图 22-3 是一个简化的模拟输入电路。由于模拟引脚和数字输出端相连，因而它们与 VDD 及 VSS 之间加有反向偏置二极管。这样，就把模拟输入电压限制在 VSS 和 VDD 之间。一旦输入电压在任一方向上超过 0.6V，就会有一个二极管正偏使输入电压被钳位。模拟信号源的最大阻抗值推荐为 10 kΩ。任何连接到模拟输入引脚上的外部元件 （如电容器、齐纳二极管等），应具有极小的泄漏电流。

图 22-3： 比较器模拟输入模型



图注：

CPIN	=	输入电容
VT	=	门限电压
ILEAKAGE	=	各结点在引脚上引起的泄漏电流
RIC	=	内部连线等效电阻
RS	=	源阻抗
VA	=	模拟电压

© 2007 Microchip Technology Inc.

22.5 比较器控制和配置

各比较器最多具有 8 个可能的输入组合：最多 4 个外部模拟输入，两个内部参考电源压之一。

两个比较器均可选择CxINA引脚上的信号或同相通道上的比较器参考电压（CVREF）。该电压与 CxINB、CxINC、CxIND 或单片机反相通道上的固定内部参考电压（VIRV，标称值为 1.2V ）进行比较。比较器输入和输出与固定 I/O 引脚连接，如表 22-1 所示。图 22-1 所示为可用的配置及其相应的位设置。

表 22-1： 比较器输入和输出

比较器	输入或输出	I/O 引脚
1	C1INA (VIN+)	RF6
	C1INB (VIN-)	RF5
	C1INC (VIN-)(1)	RH6(1)
	C1OUT	RF2
2	C2INA(VIN+)	RF4
	C2INB(VIN-)	RF3
	C2INC(VIN-)(1)	RH4(1)
	C2IND(VIN-)(1)	RH5(1)
	C2OUT	RF1

注 1： 仅在 80 引脚的器件中存在。

22.5.1 比较器使能和输入选择

将 CMxCON 寄存器（CMxCON<7>）的 CON 位置 1 将使能比较器工作。清零 CON 位将禁止比较器工作，从而可最大程度地降低电流消耗。

CMxCON 寄存器中的CCH1:CCH0位（CMxCON<1:0>）将三个模拟输入引脚之一或内部参考电压（VIRV）分配给比较器的 VIN- 引脚。根据比较器的工作模式，可选择使用外部或内部参考电压。将 VIN- 上的模拟信号和 VIN+ 上的模拟信号做比较，并相应的调整比较器的数字输出。

当 CREF = 0 （CMxCON<2>）且 VIN+ 连接到 CxINA 引脚时，使用外部参考电压。当使用外部参考电压时，可以将比较器模块配置为使用外部参考源。参考信号必须介于 VSS 和 VDD 之间，且可被施加到比较器的任一引脚上。

比较器模块也可以选择使用比较器参考电压模块内部产生的参考电压（CVREF）。在**第 23.0 节“比较器参考电压模块”**中详细介绍了该模块。仅当 CREF = 1 时，才可使用比较器参考电压模块。该模式下，内部参考电压被施加到比较器的 VIN+ 引脚上。

注：	必须通过置 1 相应的 TRISF 或 TRISH 位、ANCON1 寄存器中相应的 PCFG 位，将 CCH1:CH0 所选择的比较器输入引脚配置为输入。

22.5.1.1 64 引脚和 80 引脚器件的比较器配置

PIC18F87J11 系列器件中，两个比较器的 C 和 D 输入通道均与 PORTH 中的引脚相连，不能将其重新分配给备用模拟输入。因此，64 引脚器件为每个比较器提供共 4 种不同的配置。而 80 引脚器件为比较器 1 提供 6 种配置，为比较器 2 提供 8 种配置。图 22-1 中的配置附有脚注，说明了何种情况下这些配置不可用。

22.5.2 比较器使能和输出选择

通过 CMSTAT 寄存器可读取比较器输出。CMSTAT<0> 读取比较器 1 的输出，CMSTAT<1> 读取比较器 2 的输出。这些位是只读的。

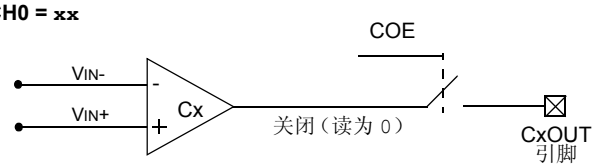
通过置位 COE 位（CMxCON<6>），比较器的输出也可以直接输出到 RF1 和 RF2 I/O 引脚。使能时，引脚的输出路径中的多路复用器切换到比较器的输出。在该模式下，TRISF<1:2> 位仍作为 RF1 和 RF2 引脚的数字输出使能位。

默认情况下，当 VIN+ 上的电压高于 VIN- 上的电压时，比较器输出逻辑高电平。使用CPOL位（CMxCON<5>）可以反转比较器输出极性。

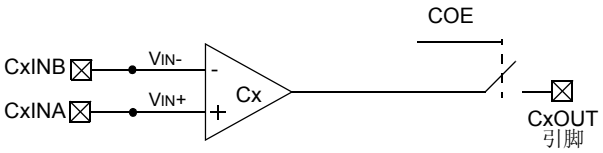
每个比较器的不确定区的大小与输入失调电压和响应时间有关，在电气规范里对此作出了说明，请参见**第 22.2 节 “比较器工作原理”**。

图 22-4：　　比较器 I/O 配置

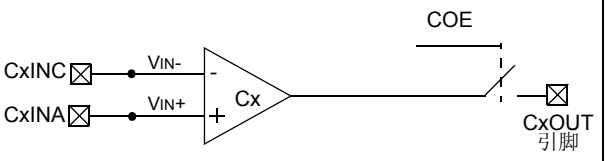
比较器关闭
CON = 0, **CREF =** x, **CCH1:CCH0 =** xx



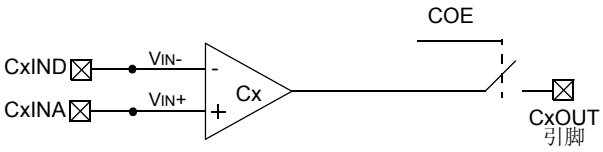
比较器比较 CxINB > CxINA
CON = 1, **CREF =** 0, **CCH1:CCH0 =** 00



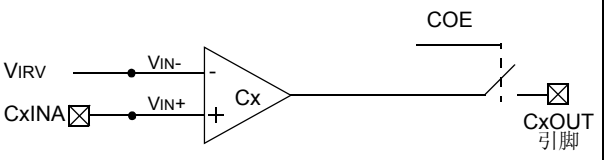
比较器比较 CxINC > CxINA(1)
CON = 1, **CREF =** 0, **CCH1:CCH0 =** 01



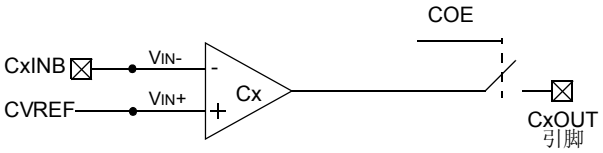
比较器比较 CxIND > CxINA(1,2)
CON = 1, **CREF =** 0, **CCH1:CCH0 =** 10



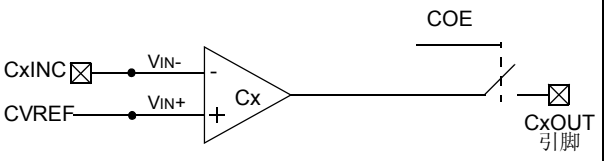
比较器比较 VIRV > CxINA
CON = 1, **CREF =** 0, **CCH1:CCH0 =** 11



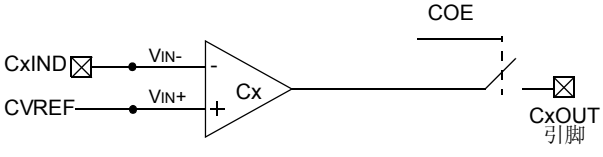
比较器比较 CxINB > CVREF
CON = 1, **CREF =** 1, **CCH1:CCH0 =** 00



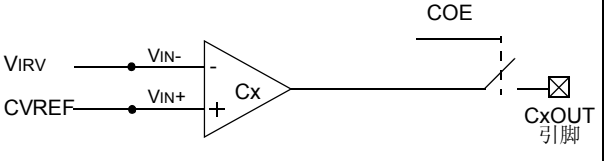
比较器比较 CxINC > CVREF(1)
CON = 1, **CREF =** 1, **CCH1:CCH0 =** 01



比较器比较 CxIND > CVREF(1,2)
CON = 1, **CREF =** 1, **CCH1:CCH0 =** 10



比较器比较 VIRV > CVREF
CON = 1, **CREF =** 1, **CCH1:CCH0 =** 11



图注： VIRV = 固定间隔参考电压 （标称值 1.2V）， CVREF = 比较器参考电压模块输出。
除非另行声明，否则可对所有封装尺寸的比较器 1 和 2 进行配置。

注　1： 配置仅在 80 引脚器件上可用。
2： 配置仅在比较器 2 （80 引脚器件）中可用。

© 2007 Microchip Technology Inc.

22.6 比较器中断

当发生以下任何事件之一时，比较器中断标志位置 1：

- 比较器输出电平的上升沿
- 比较器输出电平的下降沿
- 比较器输出电平发生任何变化

比较器中断选择通过CMxCON寄存器（CMxCON<4:3>）中的 EVPOL1:EVPOL0 位完成。

为提供最大灵活性，可使用 CMxCON 寄存器（CMxCON<5>）中的 CPOL 位对比较器的输出取反。其作用相当于反转特定模式下的比较器的反相和同相输入。

在比较器输出电平的上升沿或下降沿产生中断。此中断产生模式由 CMxCON 寄存器的 EVPOL<1:0> 确定。如果 EVPOL<1:0> = 01 或 10，则在比较器输出电平的上升沿或下降沿产生中断。一旦产生中断，则要求由软件清零中断标志位。

当 EVPOL<1:0> = 11 时，任何一个比较器的输出电平一旦发生了变化，就会将该比较器的中断标志位置 1。从 CMSTAT<1:0> 读取数据时，需要用软件来保持输出位的状态信息以判断实际发生的变化。 CMxIF 位（PIR2<6:5>）是比较器中断标志位。CMxIF 位必须通过清零复位。 因为也可把 1 写入该寄存器，所以可以启动模拟中断。表 22-2 所示为与比较器输入电压和 EVPOL 位设置相关的中断。

必须将 CMxIE 位（PIE2<6:5>）和 PEIE 位（INTCON<6>）置 1 以允许中断。此外，也必须置位 GIE （INTCON<7>）位。如果这些位中的任何一位被清零，虽然当有中断条件产生时 CMxIF 位仍会置 1，但却仍然禁止中断。

表 22-2： 产生比较器中断

CPOL	EVPOL<1:0>	比较器 输入电平变化	COUTx 转换	中断 产生
0	00	VIN+ > VIN-	上升沿	无
		VIN+ < VIN-	下降沿	无
	01	VIN+ > VIN-	上升沿	有
		VIN+ < VIN-	下降沿	无
	10	VIN+ > VIN-	上升沿	无
		VIN+ < VIN-	下降沿	有
	11	VIN+ > VIN-	上升沿	有
		VIN+ < VIN-	下降沿	有
1	00	VIN+ > VIN-	下降沿	无
		VIN+ < VIN-	上升沿	无
	01	VIN+ > VIN-	下降沿	无
		VIN+ < VIN-	上升沿	有
	10	VIN+ > VIN-	下降沿	有
		VIN+ < VIN-	上升沿	无
	11	VIN+ > VIN-	下降沿	有
		VIN+ < VIN-	上升沿	有

22.7 休眠模式下比较器工作原理

当比较器处于运行模式而器件处于休眠模式时，比较器仍能继续工作，此时如果允许中断，则中断同样有效。在允许中断时，中断会把器件从休眠模式唤醒。每个工作的比较器会消耗额外的电流。若要把休眠模式下的功耗减少到最小，可在进入休眠状态前关闭比较器（CON = 0）。器件从休眠模式唤醒时，CMxCON 寄存器的内容不受影响。

22.8 复位的影响

器件复位强制 CMxCON 寄存器进入复位状态。从而强制两个比较器和参考电压进入关闭状态。

表 22-3： 与比较器模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页：
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR2	OSCFIF	CM2IF	CM1IF	—	BCL1IF	LVDIF	TMR3IF	CCP2IF	58
PIE2	OSCFIE	CM2IE	CM1IE	—	BCL1IE	LVDIE	TMR3IE	CCP2IE	58
IPR2	OSCFIP	CM2IP	CM1IP	—	BCL1IP	LVDIP	TMR3IP	CCP2IP	58
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	56
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	56
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	56
CVRCON(2)	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	59
ANCON1(2)	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57
ANCON0(2)	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	59
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	58
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	58
PORTH(1)	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	59
TRISH(1)	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	58

图注： — = 未实现，读为 0。 A/D 转换不使用阴影单元。

注 1： 这些寄存器在 64 引脚器件上未实现。

2： 配置 SFR 的地址与默认 SFR 重叠；仅当 WDTCON<4> = 1 时才可访问配置 SFR。

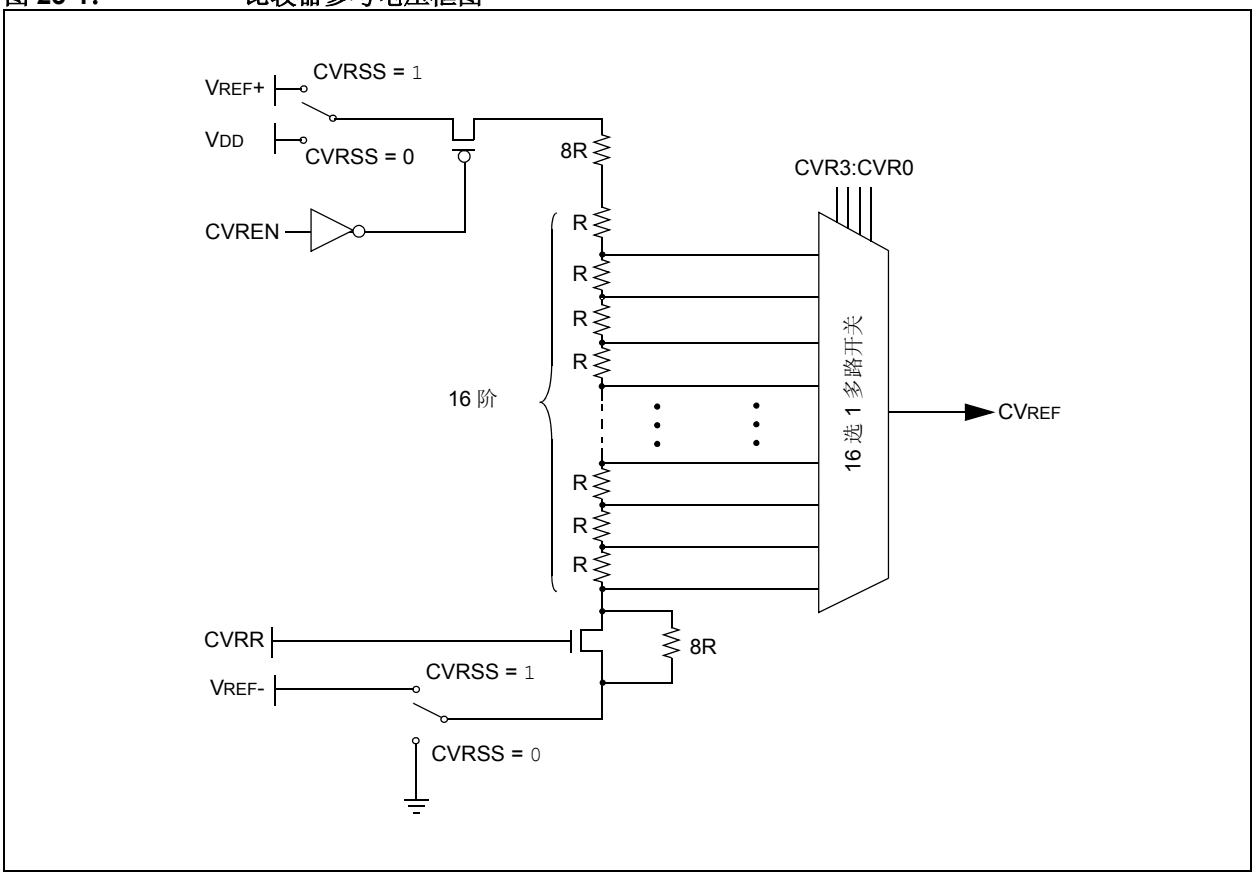
© 2007 Microchip Technology Inc.

23.0 比较器参考电压模块

比较器参考电压模块是一个 16 阶的梯形电阻网络，提供可选择的参考电压。虽然它的主要作用是给模拟比较器提供参考电压，但也可独立于模拟比较器使用。

图 23-1 为模块的框图。分割后的梯形电阻可提供两种量程范围的 CVREF 值，并且还具有断电功能以在不使用参考电压时降低功耗。模块的供电参考电压由器件 VDD/VSS 或外部参考电压提供。

图 23-1： 比较器参考电压框图



23.1 配置比较器参考电压

参考电压模块由 CVRCON 寄存器（寄存器 23-1）控制。比较器参考电压提供两种量程范围的输出电压，每种范围都具有16个不同的电压级。CVRR位（CVRCON<5>）选择要用的电压量程范围。这两种量程范围的主要区别在于 CVREF 选择位 （CVR3:CVR0）选定的步长不同，其中一个范围具有更高的分辨率。下面是计算比较器参考电压输出值的公式：

如果 CVRR = 1：
CVREF = ((CVR3:CVR0)/24) x (CVRSRC)

如果 CVRR = 0：
CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) x (CVRSRC)

比较器参考供电电压可以来自 VDD 和 VSS，或者与 RA2 和 RA3 复用的外部 VREF+ 和 VREF-。 CVRSS 位（CVRCON<4>）选择电压源。

在改变 CVREF 输出值时，必须考虑比较器参考电压的稳定时间 （见**第 27.0 节 “电气规范”**中的表 27-3）。

CVRCON 寄存器为共享地址 SFR，并使用相同的地址作为 PR4 寄存器。通过置 1 ADSHR 位 （WDTCON<4>）可访问 CVRCON 寄存器。

寄存器 23-1： CVRCON：比较器参考电压控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE(1)	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **CVREN**：比较器参考电压使能位
1 = 开启 CVREF 电路
0 = 关闭 CVREF 电路

bit 6 **CVROE**：比较器 VREF 输出使能位 (1)
1 = 在 RF5/AN10/C1INB/ 引脚上输出 CVREF 电压电平。
0 = CVREF 电压与 RF5/AN10/C1INB/CVREF 引脚断开

bit 5 **CVRR**：比较器 VREF 范围选择位
1 = 0 到 0.667CVRSRC，步长为 CVRSRC/24 （低压范围）
0 = 0.25CVRSRC 到 0.75CVRSRC，步长为 CVRSRC/32 （高压范围）

bit 4 **CVRSS**：比较器 VREF 源选择位
1 = 比较器参考电压源， CVRSRC = (VREF+) – (VREF-)
0 = 比较器参考电压源， CVRSRC = AVDD – AVSS

bit 3-0 **CVR3:CVR0**：比较器 VREF 值选择位 （0 ≤ (CVR3:CVR0) ≤ 15）
当 CVRR = 1 时：
CVREF=((CVR3:CVR0)/24) • (CVRSRC)
当 CVRR = 0 时：
CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) • (CVRSRC)

注 1： CVROE 的优先级高于 TRISF<5> 位。

© 2007 Microchip Technology Inc.

23.2 参考电压精度 / 误差

由于模块构造的原因，模块无法实现满量程参考电压。梯形电阻网络顶部和底部的晶体管（图 23-1）使 CVREF 无法达到参考电压源的满幅值。参考电压来自于参考电压源；因此， CVREF 输出电平会随参考电压源一起波动。测试所得的参考电压绝对精度，请参见**第 27.0 节“电气规范”**。

23.3 连接注意事项

参考电压模块独立于比较器模块工作。如果 CVROE 位置 1，参考电压发生器的输出可能会连到 RF5 引脚。使能参考电压输出到 RA2 引脚 （如果该引脚被配置为数字输入）将会增加 电流消耗。使能 CVRSS 时，将 RF5 用作数字输出引脚也将增加电流消耗。

RF5 引脚可用作简单 D/A 输出，但是其驱动能力有限。由于这种有限的电流驱动能力，因此当 VREF 有外部连接时必须在参考电压输 出上使用缓冲器。图 23-2 举例说明了这一缓冲技术。

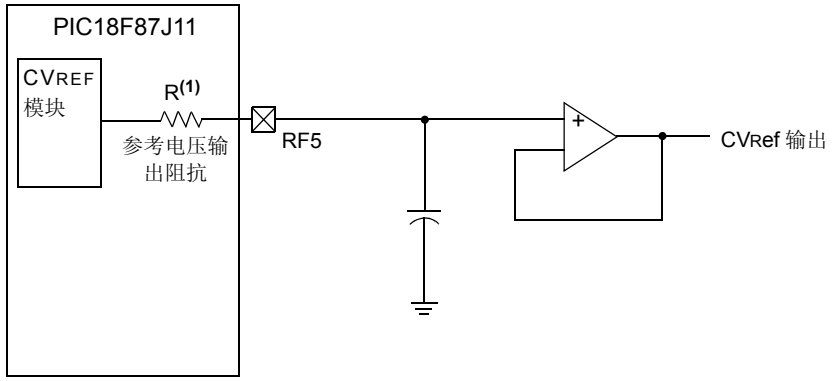
23.4 休眠模式下的工作

当中断或看门狗定时器超时而唤醒器件时， CVRCON 寄存器内容不受影响。为了最大限度地降低休眠模式下的电流消耗，应关闭参考电压模块。

23.5 复位的影响

通过清零 CVREN （CVRCON<7>）使器件复位从而禁止参考电压模块。通过清零 CVROE，复位还将参考模块与 RA2 引脚断开，并且通过清零 CVRR，复位还可以选择高电压量程范围。同时 CVR 值选择位也被清零。

图 23-2： 比较器参考电压输出缓冲示例



注 1： R 取决于参考电压配置位 CVRCON<5> 和 CVRCON<3:0>。

表 23-1： 与比较器参考电压相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页
CVRCON(2)	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	59
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	56
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	56
TRISA	TRISA7(1)	TRISA6(1)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	58
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	58
ANCON0(2)	PCFG7	PCFG6	—	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	57
ANCON1(2)	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	57

图注： — = 未实现，读为 0。比较器参考电压未使用阴影单元。

注 1： 这些位仅在选择振荡器模式下 （FOSC2 配置位 = 0）可用；其他模式下均未实现。

2： 配置 SFR 的地址与默认 SFR 重叠 ； 仅在 WDTCON<4> = 1 时才可访问配置 SFR。

注：

 © 2007 Microchip Technology Inc.

24.0 CPU 的特殊性能

PIC18F87J11 系列器件包含的功能旨在最大限度地提高系统可靠性，并通过减去外部元件将成本降到最低。这些寄存器包括：

- 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
- 在线串行编程

根据具体应用对频率、功耗、精度和成本的要求来配置振荡器。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经完整地讨论了器件的复位和中断。PIC18F87J11 系列器件除了为复位提供上电延时定时器和振荡器起振定时器外，还具有可配置的看门狗定时器，该定时器由软件控制。

器件自带的 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有益功能。 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。双速启动使得代码几乎可在起振发生时立即执行，此时主时钟源正在完成其自身的起振延时。

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。

24.1 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中从 300000h 开始的单元中。在表 24-2 中给出了完整的列表。从寄存器 24-1 到寄存器 24-6 详细解释了各配置位的功能。

24.1.1 配置 PIC18F87J11 系列器件的注意事项

与早期 PIC18 单片机不同， PIC18F87J11 系列器件不使用非易失性存储寄存器存储配置信息。配置字节以易失性存储方式实现，这意味着在器件每次上电时都必须对配置数据进行编程。

配置数据存储在片上程序寄存器空间顶部的 4 个字中，这些字被称为闪存配置字。配置数据按照表 24-2 中相同的次序存储在程序存储器中，CONFIG1L 位于地址最低的单元，CONFIG3H 位于地址最高的单元。在器件上电时，这些数据自动装载到正确的配置寄存器中。

当为这些器件创建应用程序时，用户应始终为配置数据特别分配闪存配置字，以确保在编译代码时程序代码不会存储在这个地址上。

上电复位时，用于配置位的易失性存储单元始终复位为 1。对于所用其他类型的复位事件，将保存和使用先前已编程的值，而无需从程序寄存器中重载。

程序存储器中 CONFIG1H、CONFIG2H 和 CONFIG3H 的高 4 位也应该设置为 1111。这使得这些配置字被远程事件意外执行时，被当作一条 NOP 指令。因为配置位在对应单元中是未实现的，因此向这些单元写入 1 不会影响器件工作。

为了避免在代码执行期间配置被意外更改，所有的可编程配置位只可被写入一次。在上电周期中对某一位运行初始化后，就不能再次写入了。改变器件的配置需要对器件重新上电。

表 24-1： 将闪存配置字映射到配置寄存器

配置字节	代码空间地址	配置寄存器地址
CONFIG1L	XXXF8h	300000h
CONFIG1H	XXXF9h	300001h
CONFIG2L	XXXFAh	300002h
CONFIG2H	XXXFBh	300003h
CONFIG3L	XXXFCh	300004h
CONFIG3H	XXXFDh	300005h
CONFIG4L(1)	XXXFEh	300006h
CONFIG4H(1)	XXXFFh	300007h

注 1： 在 PIC18F87J11 系列器件中未实现。

表 24-2： 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 / 未编程的值 (1)
300000h	CONFIG1L	$\overline{\text{DEBUG}}$	XINST	STVREN	—	—	—	—	WDTEN	111- ---1
300001h	CONFIG1H	— (2)	— (2)	— (2)	— (2)	—	CP0	—	—	1111 -111
300002h	CONFIG2L	IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0	11-- -111
300003h	CONFIG2H	— (2)	— (2)	— (2)	— (2)	WDTPS3	WDTPS2	WDTPS1	WDTPS0	1111 1111
300004h	CONFIG3L	RA6(3)	BW(3)	EMB1(3)	EMB0(3)	EASHFT(3)	—	—	—	1111 1---
300005h	CONFIG3H	— (2)	— (2)	— (2)	— (2)	MSSPMSK	PMPMX(3)	ECCPMX(3)	CCP2MX	1111 1111
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxx0 0000(4)
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0100 00xx(4)

图注： x = 未知， u = 不变， - = 未实现。阴影单元表示未实现，读为 0。

注 1： 这些值反映出厂时或上电复位后未编程状态。在所有其他复位后，配置字节保持先前的编程状态。

2： 程序存储器中这些位的值应始终为 1。这样可确保如果意外地执行了这些单元，将会执行 NOP 指令。

3： 仅在 80 引脚器件中实现了。

4： 请参见寄存器 24-7 和寄存器 24-8 了解 DEVID 的值。这些寄存器为只读寄存器，用户不能对其进行编程。

© 2007 Microchip Technology Inc.

寄存器 24-1： CONFIG1L：配置寄存器 1 的低字节 （字节地址为 300000h）

R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0	U-0	R/WO-1
$\overline{\text{DEBUG}}$	XINST	STVREN	—	—	—	—	WDTEN
bit 7							bit 0

图注：			
R = 可读位	WO = 一次性写入位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

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 **STVREN**：堆栈上溢 / 下溢复位使能位
1 = 使能堆栈上溢 / 下溢复位
0 = 禁止堆栈上溢 / 下溢复位

bit 4-1 **未实现**：读为 0

bit 0 **WDTEN**：看门狗定时器使能位
1 = 使能 WDT
0 = 使能 WDT （由 SWDTEN 位控制）

寄存器 24-2： CONFIG1H：配置寄存器 1 的高字节 （字节地址为 300001h）

U-1	U-1	U-1	U-1	U-0	R/WO-1	U-1	U-1
—	—	—	—	—	CP0	—	—
bit 7							bit 0

图注：			
R = 可读位	WO = 一次性写入位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-3 **未实现**：读为 0

bit 2 **CP0**：代码保护位
1 = 程序存储器未受代码保护
0 = 程序存储器受代码保护

bit 1-0 **未实现**：读为 0

寄存器 24-3： CONFIG2L：配置寄存器 2 的低字节 （字节地址为 300002h）

R/WO-1	R/WO-1	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1
IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0
bit 7							bit 0

图注：			
R = 可读位	WO = 一次性写入位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **IESO**：双速启动 （内部 / 外部振荡器切换）控制位
1 = 使能双速启动
0 = 禁止双速启动

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 5-3 **未实现**：读为 0

bit 2-0 **FOSC2:FOSC0**：振荡器选择位
111 = EC 振荡器，使能 PLL， RA6 用作 CLKO （ECPLL）
110 = EC 振荡器， RA6 用作 CLKO （EC）
101 = HS 振荡器，使能 PLL （HSPLL）
100 = HS 振荡器 （HS）
011 = 内部振荡器，使能 PLL， RA6 用作 CLKO， RA7 用作端口功能 （INTPLL1）
010 = 内部振荡器，使能 PLL， RA6 用作 CLKO， RA7 用作端口功能 （INTPLL2）
001 = 内部振荡器电路， RA6 用作 CLKO 引脚， RA7 用作端口功能 （INTIO1）
000 = 内部振荡器电路， RA6 和 RA7 均用作端口功能 （INTIO2）

© 2007 Microchip Technology Inc.

寄存器 24-4： CONFIG2H：配置寄存器 2 的高字节 （字节地址为 300003h）

U-1	U-1	U-1	U-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
—	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

图注：			
R = 可读位	WO = 一次性写入位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **未实现**：读为 0

bit 3-0 **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

寄存器 24-5： CONFIG3L：配置寄存器 3 的低字节 （字节地址为 300004h）

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
RA6(1)	BW(1)	EMB1(1)	EMB0(1)	EASHFT(1)	—	—	—
bit 7							bit 0

图注：			
R = 可读位	WO = 一次性写入位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **WAIT**：外部总线等待使能位 (1)
1 = 禁止外部总线等待
0 = 使能外部总线等待，可通过 MEMCON<5:4> 进行选择

bit 6 **BW**：数据总线宽度选择位 (1)
1 = 16 位数据宽模式
0 = 8 位数据宽模式

bit 5-4 **EMB1:EMB0**：外部存储器总线配置位 (1)
11 = 单片机模式，禁止外部总线
10 = 扩展的单片机模式，外部总线的地址宽度为 12 位
01 = 扩展的单片机模式，外部总线的地址宽度为 16 位
00 = 扩展的单片机模式，外部总线的地址宽度为 20 位

bit 3 **EASHFT**：外部地址总线移位使能位 (1)
1 = 使能地址移位——外部地址总线平移到从 000000h 开始的地方
0 = 禁止地址移位——外部地址总线反映 PC 值

bit 2-0 **未实现**：读为 0

注 1： 仅在 80 引脚器件上实现了。

© 2007 Microchip Technology Inc.

寄存器 24-6： CONFIG3H：配置寄存器 3 的高字节 （字节地址为 300005h）

U-1	U-1	U-1	U-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
—	—	—	—	MSSPMSK	PMPMX(1)	ECCPMX(1)	CCP2MX
bit 7							bit 0

图注：			
R = 可读位	WO = 一次性写入位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **未实现：**读为 0

bit 3 **MSSPMSK：**MSSP 地址屏蔽模式选择位
1 = 使能 7 位地址屏蔽模式
0 = 使能 5 位地址屏蔽模式

bit 2 **PMPMX：**PMP 引脚复用位 (1)
1 = PMP 数据和控制线复用为同一引脚，以作为外部存储器总线 （PORTD 和 PORTE）
0 = PMP 数据和控制线复用为其他引脚分配 （PORTA、 PORTF 和 PORTH）

bit 1 **ECCPMX：**ECCPx 复用位 (1)
1 = ECCP1 输出 （P1B/P1C）与 RE6 和 RE5 复用；
ECCP3 输出 （P3B/P3C）与 RE4 和 RE3 复用
0 = ECCP1 输出 （P1B/P1C）与 RH7 和 RH6 复用；
ECCP3 输出 （P3B/P3C）与 RH5 和 RH4 复用

bit 0 **CCP2MX：**ECCP2 复用位
1 = ECCP2/P2A 与 RC1 复用
0 = 在单片机模式下，ECCP2/P2A 与 RE7 复用（所有器件），或在扩展单片机模式下与 RB 复用（仅 80 引脚器件）

注 1： 仅在 80 引脚器件中实现了。

寄存器 24-7：　DEVID1：PIC18F87J11 系列器件的器件 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 = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5　**DEV2:DEV0：**器件 ID 位
完整列表，请参见寄存器 24-8。

bit 4-0　**REV4:REV0：**版本 ID 位
这些位用于表明器件版本。

寄存器 24-8：　DEVID2：PIC18F87J11 系列器件的器件 ID 寄存器 2

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0　**DEV10:DEV3：**器件 ID 位

DEV10:DEV3（DEVID2<7:0>）	DEV2:DEV0（DEVID1<7:5>）	器件
0100 0100	010	PIC18F66J11
0100 0100	011	PIC18F66J16
0100 0100	100	PIC18F67J11
0100 0100	111	PIC18F86J11
0100 0101	000	PIC18F86J16
0100 0101	001	PIC18F87J11

© 2007 Microchip Technology Inc.

24.2 看门狗定时器 （WDT）

对于 PIC18F87J11 系列器件， WDT 由 INTRC 振荡器驱动。当使能 WDT 时，也将使能时钟源。WDT 超时溢出周期的标称值为 4 ms，其稳定性与 INTRC 振荡器相同。

4 ms 的 WDT 超时溢出周期与 16 位的后分频比值相乘。通过配置寄存器 2H 中 WDTPS 位控制一个多路开关以对 WDT 后分频器的输出进行选择，可获得的超时溢出周期范围为 4 ms 至 135s（2.25 分钟，取决于电压、温度和 WDT 分频器的输出）。当执行 SLEEP 或 CLRWDT 指令时，或发生时钟故障 （主时钟或 Timer1 振荡器）时， WDT 和分频器被清零。

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。

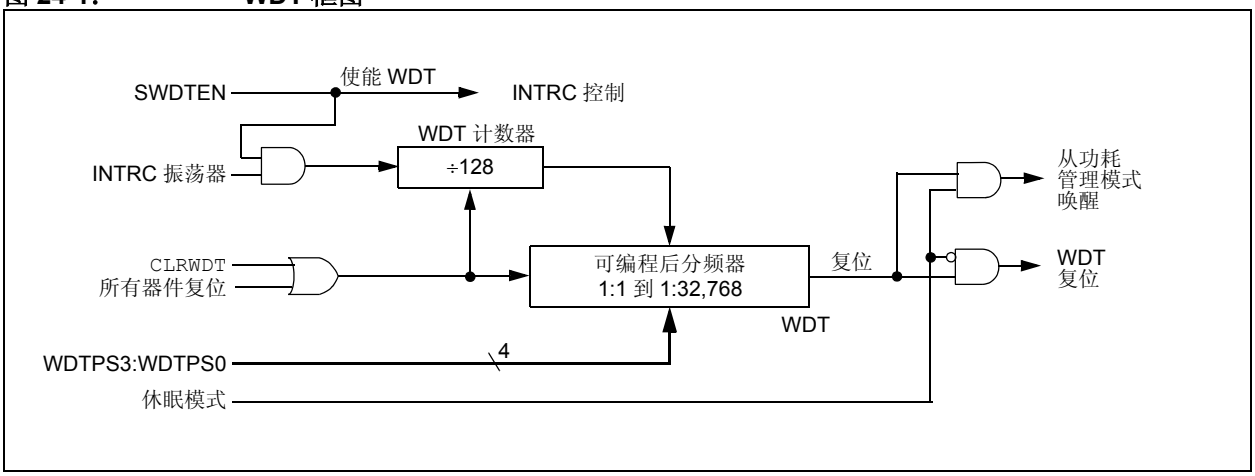
24.2.1 控制寄存器

WDTCON 寄存器（寄存器 24-9）是可读写寄存器。SWDTEN 位使能或者禁止 WDT 操作。这使得软件能够改写 WDTEN 配置位，且仅当由配置位禁止 WDT 操作时才能使能 WDT。

ADSHR位选择当前选择的且可访问的SFR。更多详情，请参见**第 5.3.4.1 节 “共享地址 SFR”**。

LVDSTAT 是只读状态位，它能够不断的更新并提供关于 VDDCORE 电流的信息。仅在使能了片上稳压器时此位才有效。

图 24-1： WDT 框图



寄存器 24-9： WDTCON：看门狗定时器控制寄存器

R/W-0	R-x	U-0	R/W-0	U-0	U-0	U-0	U-0
REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN(1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **REGSLP**：稳压器低电压操作使能位。
1 = 当器件进入休眠状态时片上稳压器进入低功耗操作。
0 = 即使在休眠状态下片上稳压器处于活跃状态。

bit 6 **LVDSTAT**：LVD 状态位
1 = V_{DDCORE} > 2.45V
0 = V_{DDCORE} < 2.45V

bit 5 **未实现**：读为 0

bit 4 **ADSHR**：共享地址 SFR 选择位
欲知位操作的详细信息，请参见寄存器 5-3。

bit 3-1 **未实现**：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： 当配置位 WDTEN 使能时该位不起作用。

表 24-3： 看门狗定时器的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在的页
RCON	IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	56
WDTCON	REGSLP	LVDSTAT	—	ADSHR	—	—	—	SWDTEN	57

图注： — = 未实现位，读为 0。看门狗定时器不使用阴影单元。

© 2007 Microchip Technology Inc.

24.3 片上稳压器

所有的 PIC18F87J11 系列器件使用 2.5V 标称电压为其内核数字逻辑电路供电。对于需要工作在较高的典型电压（如 3.3V）的设计，PIC18F87J11 系列的所有器件包含允许器件从 VDD 运行其内核逻辑的片上稳压器。

通过 ENVREG 引脚控制稳压器。将 VDD 连接到该引脚将使能稳压器，然后稳压器通过其他 VDD 引脚向内核供电。当使能稳压器时，低 ESR 滤波电容必须和 VDDCORE/VCAP 引脚（图 24-2）相连。这有助于保持稳压器的稳定性。**第 27.3 节“直流特性：PIC18F87J11 系列（工业级）”**中给出了滤波电容的推荐值。

如果 ENVREG 连至 VSS，将禁止稳压器。在这种情况下，独立的 2.5V 标称值的内核逻辑电压必须通过 VDDCORE/VCAP 引脚向器件供电，从而将 I/O 引脚驱动为一个较高的电平，通常为 3.3V。另外，VDDCORE/VCAP 和 VDD 引脚也可以连接在一起，使器件工作在较低的标称电压下。可能的配置请参见图 24-2。

24.3.1 稳压器跟踪模式和低电压检测

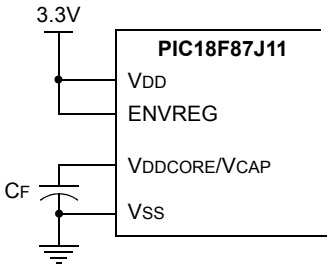
当使能片上稳压器时，它向数字内核逻辑提供一个固定的 2.5V 标称电压。稳压器可通过约 2.5V 的 VDD 直到器件 VDDMAX 来提供此电压。但不能将低于 2.5V 的 VDD 升压到此标称电压。为了防止“欠压”条件，当电压降至很低时，稳压器将进入跟踪模式。在跟踪模式下，稳压器的输出低于 VDD，典型压降为 100 mV。

片上稳压器包含一个简单的低压检测（Low-Voltage Detect，LVD）电路。如果VDD下降到不能维持VDDCORE上的近 2.45V 电压，该电路就会将低电压检测中断标志 LVDIF（PIR2<2>）置 1。这可用来产生中断并使应用进入低压工作模式，或引发顺序关闭。低压检测仅在稳压器被使能时可用。

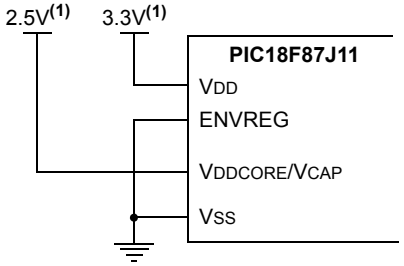
低压检测中断是边沿触发的。仅在 VDDCORE 的每个下降沿处置 1 中断标志。固件能清零中断标志，但在 VDDCORE 回升至阈值 2.45 以上再降至其下后即可产生新中断。器件复位时，中断标志将复位为 0，即使 VDDCORE 低于 2.45V。当稳压器使能时，可通过查询 WDTCON 寄存器的 LVDSTAT 位来确定 VDDCORE 的电流。

图 24-2： 片上稳压器连接

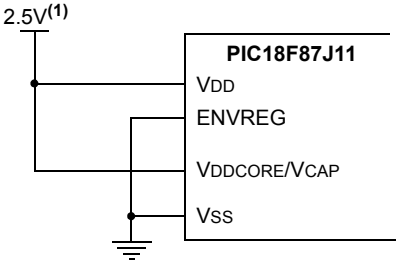
使能稳压器（ENVREG 与 VDD 相连）：



禁止稳压器（ENVREGL 与地相连）：



禁止稳压器（VDD 与 VDDCORE 相连）：



注 1： 上述电压为典型的工作电压。有关 VDD 和 VDDCORE 的全部工作电压范围，请参见第 **节 27.1“直流特性：电源电压”**。

24.3.2 片上稳压器和 BOR

当片上稳压器使能时，PIC18F87J11 系列器件也具有一个简单的欠压功能。如果向稳压器提供的电压不足以维持一个稳定的电平，那么稳压器复位电路将产生欠压复位。通过 BOR 标志位 （RCON<0>）来捕捉该事件。

在**第 4.4 节“欠压复位（BOR）”**和**第 4.4.1 节“检测 BOR”**中详细描述了欠压复位工作原理。在**第 27.1 节“直流特性：供电电压 PIC18F87J11 系列（工业级）”**中指定了欠压电平值。

24.3.3 上电要求

片上稳压器的设计应满足器件的上电要求。如果应用不使用稳压器，那么必须严格遵守上电条件。上电时，V_{DDCORE} 不能比 V_{DD} 高出 0.3V 以上。

24.3.4 休眠模式下的操作

当使能时，片上稳压器消耗的电流总是稍高于 I_{DD}。包括当器件处于休眠模式下，即使内核数字逻辑不需要电源的情况下消耗的电流。要在对电源资源要求高的应用中提供额外的节能，可将稳压器配置为只要器件进入休眠模式就自动禁止。该功能由 REGSLP 位（寄存器 24-9 中的 WDTCON<7>）控制。置 1 该位将在休眠模式下禁止稳压器并将其电流消耗将到最低。

通过将 REGSLP 位置 1 可以获得充足的休眠模式功耗，但是为了确保稳压器有足够的时间来稳定，将增加器件的唤醒时间。当产生低压检测条件时，REGSLP 位将由硬件自动清零。

24.4 双速启动

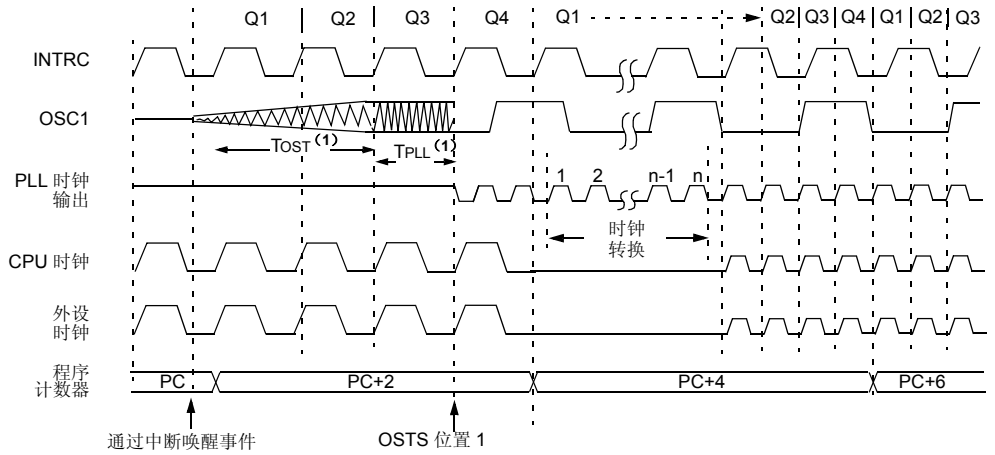
双速启动功能允许单片机在主时钟源可用之前使用 INTRC 振荡器作为时钟源，从而帮助器件最大限度地缩短从振荡器起振到代码执行之间的延时。通过将 IESO 配置位置 1 可使能该功能。

仅当主振荡模式为 HS 或 HSPLL（基于晶振的模式）时才可使用双速启动。因为 EC 和 ECLL 模式不需要振荡器起振定时器延时，因此应禁止双速启动。

一旦使能双速启动，器件复位和从休眠模式唤醒时，在上电延时定时器 （使能上电延时复位后）发生延时后，器件都将被配置成使用内部振荡电路作为时钟源。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，器件就自动换到 PRI_RUN 模式。

在其他所有的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。

图 24-3： 双速启动时钟转换的时序图 （从 INTRC 切换到 HSPLL）



注 1： T_{OST} = 1024 T_{OSC}； T_{PLL} = 2 ms （近似值）。这些时间间隔未按比例显示。

© 2007 Microchip Technology Inc.

24.4.1 使用双速启动时的特殊注意事项

当在双速启动中使用 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。这就使应用能短暂地唤醒器件，执行“日常事务”子程序，并在器件开始使用主时钟源前返回休眠状态。

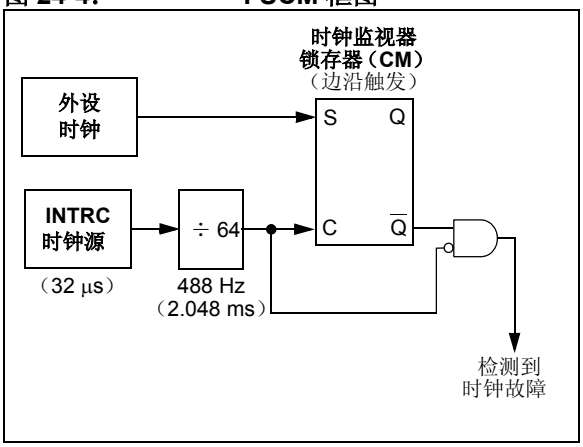
用户代码还能通过检查 OSTS 位（OSCCON<3>）的状态来确定主时钟源是否正在为系统提供时钟。若该位置 1，则表示主振荡器正在为系统提供时钟。否则，表示当器件从复位或休眠模式唤醒期间由内部振荡电路为系统提供时钟。

24.5 故障保护时钟监视器

故障保护时钟监视器（FSCM）可使单片机在发生外部时钟故障时，自动将系统时钟切换到内部振荡电路以保持器件继续运行。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 24-4 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。外设器件时钟和采样时钟作为时钟监视器锁存器（CM）的输入。CM 在系统时钟源的下降沿被置 1，在采样时钟的上升沿被清零。

图 24-4： 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 24-5）。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电路（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。

切换过程中，对于时序要求较高的应用，内部振荡电路的后分频频率可能不够稳定。在这些情况下，最好选择另一种时钟配置并进入其他功耗管理模式。可以尝试部分恢复或执行受控的关闭。请参见**第 3.1.4 节“多条 Sleep 命令”**和**第 24.4.1 节“使用双速启动时的特殊注意事项”**获得更多详细信息。

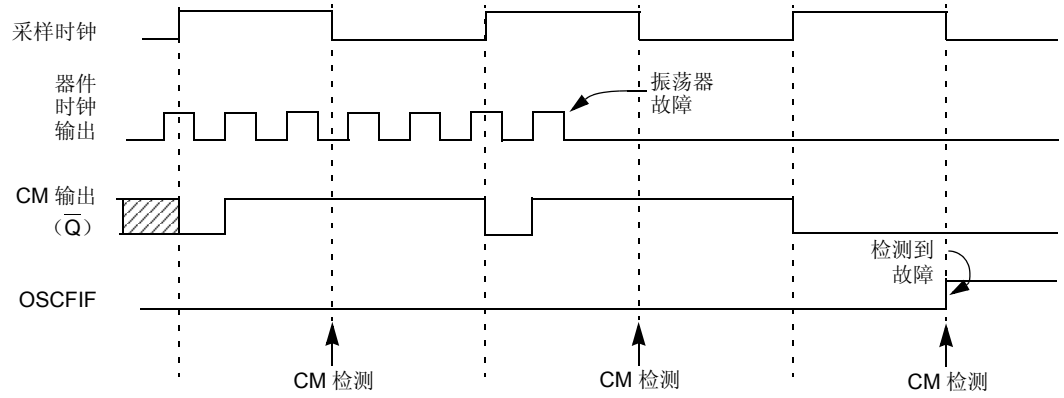
FSCM 只能检测出主时钟或辅助时钟源故障。如果内部振荡电路发生故障，将无法检测到任何故障，当然也不可能采取任何措施。

24.5.1 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，当使能 FSCM 时，禁止 WDT 对 INTRC 振荡器的运行没有影响。

如前所述，当监测到时钟故障时，时钟源转换为 INTRC 时钟；这可能意味着代码执行速度会发生很大的变化。如果用小分频值的时钟源使能 WDT，时钟速率的下降将允许 WDT 发生超时并在随后使器件复位。由于这个原因，故障保护事件也会清零 WDT 和后分频器，从而使 WDT 从运行速度发生变化的时刻开始重新计数，因而减少发生错误超时的可能性。

图 24-5：　　FSCM 时序图



注：　通常，器件时钟的频率比采样时钟频率高很多。为明了起见，本例选择了差异较大的频率。

24.5.2　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终止故障保护状态。发生复位时，控制器启动在配置寄存器 2H 中指定的主时钟源（伴有如 OST 或 PLL 定时器等振荡器模式所需的起振延时）。INTRC 多路开关将在主时钟源就绪前提供系统时钟（类似于双速启动）。当主时钟源可用时，系统时钟源将切换回主时钟（OSCCON 寄存器中的 OSTS 位置 1 以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。

在起振期间，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振荡器作为时钟源。OSCCON 寄存器将保持复位状态直到进入功耗管理模式为止。

24.5.3　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。在功耗管理模式下将恢复对功耗管理时钟源的故障保护监视。

如果在功耗管理运行期间发生了振荡器故障，随后将会发生的事件取决于是否允许振荡器故障中断。如果使能（OSCFIF = 1），代码执行将以 INTRC 多路开关作为时钟源。并不会自动转回到发生故障的时钟源。

如果禁止该中断，休眠模式下故障所导致的中断将使 CPU 开始执行指令，此时将由 INTRC 时钟源提供时钟。

24.5.4　POR 或从休眠中唤醒

FSCM 旨在在器件退出上电复位（POR）或低功耗休眠模式后的任一时刻都可以检测到振荡器故障。当系统主时钟为 EC 或 INTRC 模式时，监视会在这些事件发生后立即开始。

对于 HS 或 HSPLL 模式，情况会有所不同。由于这类振荡器需要的起振时间可能比FCSM采样时钟的时间长很多，因此可能会检测到假时钟故障。为了避免这一情况，内部振荡电路会被自动配置为器件时钟并一直工作直到主时钟稳定下来为止（OST 和 PLL 定时器已完成延时）。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：	防止在发生POR或从休眠状态唤醒时发生假中断的逻辑电路同样也将阻止在发生这些事件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标记任何振荡器故障中断。

正如**第 24.4.1 节“使用双速启动时的特殊注意事项”**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并进入另一功耗管理模式。当选择新的功耗管理模式时，主时钟将被禁止。

© 2007 Microchip Technology Inc.

24.6 程序校验和代码保护

对于 PIC18F87J11 系列中所有器件，将片上程序存储器空间视为一个存储区。该存储区的代码保护由 CP0 配置位控制。该位阻止对程序存储空间的外部读写。但对正常执行的模式没有直接影响。

24.6.1 保护配置寄存器保护

有两种方法保护配置寄存器使其免遭破坏性改写或读取。主要的保护方式是配置位的一次写入功能，该功能阻止对在上电周期内完成编程的位再次进行配置。要阻止不可预见的事件，由于电池故障（如 ESD 事件）产生的配置位更改将导致奇偶校验错误并触发器件复位。对于用户来说，此为配置匹配复位。

配置寄存器的数据来自程序存储器中的闪存配置字。当 CP0 位置 1 时，也将保护器件配置的源数据。

24.7 在线串行编程

PIC18F87J11 系列单片机可以在最终应用电路中进行串行编程。只需要 5 根线即可完成这一操作，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。这允许用户使用未编程器件制造电路板，仅在产品交付前才对单片机进行编程。从而可使固件最新或按定制编写固件。

24.8 在线调试器

将 $\overline{\text{DEBUG}}$ 配置位清零可使能在线调试功能。这一功能允许使用 MPLAB® IDE 进行一些简单的调试。当使能了单片机的这项功能时，某些资源就不再是通用的了。表 24-4 显示了后台调试器所需的资源。

表 24-4： 调试器资源

I/O 引脚：	RB6， RB7
堆栈：	2 级
程序存储器：	512 字节
数据存储器：	10 字节

注：

 © 2007 Microchip Technology Inc.

25.0 指令集综述

PIC18F87J11 系列器件具有一个含有 75 个 PIC18 内核指令的标准指令集，以及一个含有 8 个新指令（优化递归和软件堆栈代码）的扩展指令集组成。本节后面部分将讨论该扩展指令集。

25.1 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。

每条单字指令都是一个 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，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 25-2 中的 PIC18 指令集汇总列示了**字节**、**位**、**立即数**和**控制** 操作。表 25-1 给出了对操作码字段的说明。

大部分**字节**操作类指令含有三种操作数：

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。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的编号。

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，执行指令需要两个指令周期，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 个振荡周期组成。因此，对于频率为 4 MHz 的振荡器，其正常的指令执行时间为 1 μs。如果条件测试为 true 或指令执行改变了程序计数器值，则该指令的执行时间为 2 μs。双字转移指令（如果为 true）的执行需要 3 μs。

图 25-1 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 25-2）列出了可被Microchip MPASM™ 汇编器识别的标准指令。

第 25.1.1 节“标准指令集”中对每条指令进行了介绍。

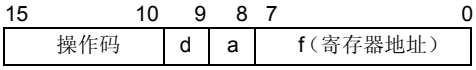
表 25-1： 操作码字段说明

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 寄存器指定的 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**位标志位、**辅助进**位标志位、**全**零标志位、**溢**出标志位和**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 寄存器或指定的寄存器单元。
f	8 位寄存器地址 （00h 到 FFh）或 2 位 FSR 标识符 （0h 到 3h）。
f_s	12 位寄存器地址 （000h 到 FFFh）。这是源地址。
f_d	12 位寄存器地址 （000h 到 FFFh）。这是目标地址。
GIE	全局中断允许位。
k	立即数字段、常数或者标号 （可能是 8 位、 12 位或 20 位的值）。
标号	标号名称。
mm * *+ *- +*	表读和表写指令的 TBLPTR 寄存器模式。只和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	程序计数器的低字节。
PCH	程序计数器的高字节。
PCLATH	程序计数器的高字节锁存器。
PCLATU	程序计数器的最高字节锁存器。
$\overline{PD}$	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：将某些寄存器的值存入影子寄存器或把影子寄存器的值载入某些寄存器 （快速模式）
TBLPTR	21 位表指针 （指向程序存储单元）。
TABLAT	8 位表锁存器。
$\overline{TO}$	超时溢出位。
TOS	栈顶。
u	未使用或不变。
WDT	看门狗定时器。
WREG	工作寄存器 （累加器）。
x	忽略 （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 New）。

© 2007 Microchip Technology Inc.

图 25-1：　　指令的通用格式

面向字节的文件寄存器操作　　**指令示例**



ADDWF MYREG, W, B

d = 0 表示结果存入 WREG 寄存器
d = 1 表示结果存入文件寄存器（f）
a = 0 强制使用快速操作存储区
a = 1 根据 BSR 选择存储区
f = 8 位文件寄存器地址

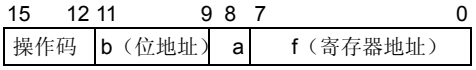
字节到字节的传送操作（双字）



MOVFF MYREG1, MYREG2

f = 12 位文件寄存器地址

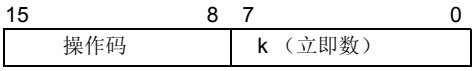
面向位的文件寄存器操作



BSF MYREG, bit, B

b = 占 3 位，表示文件寄存器（f）中位的位置
a = 0 强制使用快速操作存储区
a = 1 根据 BSR 选择存储区
f = 8 位文件寄存器地址

立即数操作类指令

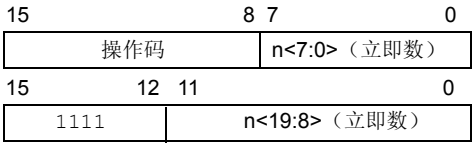


MOVLW 7Fh

k = 8 位立即数的值

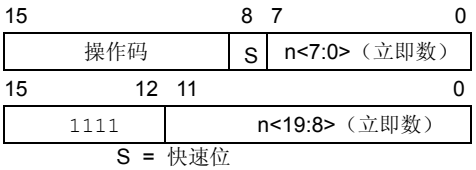
控制操作类指令

CALL、GOTO 和跳转类操作指令



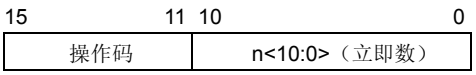
GOTO 标号

n = 20 位立即数的值

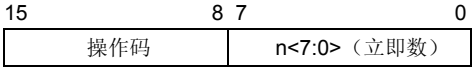


CALL MYFUNC

S = 快速位



BRA MYFUNC



BC MYFUNC

表 25-2： PIC18F87J11 系列指令集

助记符，操作数	说明	周期数	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 与 f 作逻辑与运算	1	0001 01da ffff ffff	Z,N	1, 2
CLRF f, a	f 清零	1	0110 101a ffff ffff	Z	2
COMF f, d, a	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 与 f 作逻辑或运算	1	0001 00da ffff ffff	Z,N	1, 2
MOVF f, d, a	传送 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 f_d （目标）地址装入第二个字	2	1100 ffff ffff ffff 1111 ffff ffff ffff	无	
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	f 取补	1	0110 110a ffff ffff	C,DC,Z,OV,N	
RLCF f, d, a	对 f 执行带进位的循环左移	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	对 f 执行带进位的循环右移	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	f 置 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 与 f 作逻辑异或运算	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，虽然其数据锁存器中的值为 1，但此时外部器件将该引脚驱动为低电平，则被写回数据总线的数据值将是 0。

2： 当对 TMR0 寄存器 （以及其他适用的寄存器）执行该指令时 （并且 d = 1），如果已对其分配了预分频器，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单元内存储的都是合法的指令。

© 2007 Microchip Technology Inc.

表 25-2： PIC18F87J11 系列指令集 （续）

助记符，操作数		说明	周期数	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	检测 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	检测 f 中的某位，为 1 则跳过	1 (2 或 3)	1010 bbba ffff ffff	无	3, 4
BTG	f, b, a	将 f 中的某位取反	1	0111 bbba ffff ffff	无	1, 2
控制操作类指令						
BC	n	进位则转移	1 (2)	1110 0010 nnnn nnnn	无	
BN	n	为负则转移	1 (2)	1110 0110 nnnn nnnn	无	
BNC	n	无进位则转移	1 (2)	1110 0011 nnnn nnnn	无	
BNN	n	不为负则转移	1 (2)	1110 0111 nnnn nnnn	无	
BNOV	n	不溢出则转移	1 (2)	1110 0101 nnnn nnnn	无	
BNZ	n	不为零则转移	1 (2)	1110 0001 nnnn nnnn	无	
BOV	n	溢出则转移	1 (2)	1110 0100 nnnn nnnn	无	
BRA	n	无条件转移	2	1101 0nnn nnnn nnnn	无	
BZ	n	为零则转移	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	—	清零看门狗定时器	1	0000 0000 0000 0100	$\overline{TO}$ 和 $\overline{PD}$	
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	—	弹出返回堆栈栈顶 (TOS) 的内容	1	0000 0000 0000 0110	无	
PUSH	—	压入返回堆栈栈顶 (TOS)	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	所有	
RETFIE	s	中断返回使能	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	$\overline{TO}$ 和 $\overline{PD}$	

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，虽然其数据锁存器中的值为 1，但此时外部器件将该引脚驱动为低电平，则被写回数据总线的数据值将是 0。

2： 当对 TMR0 寄存器 （以及其他适用的寄存器）执行该指令时 （并且 d = 1），如果已对其分配了预分频器，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单元内存储的都是合法的指令。

表 25-2： PIC18F87J11 系列指令集 （续）

助记符，操作数		说明	周期数	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	立即数与 WREG 作逻辑与运算	1	0000 1011 kkkk kkkk	Z,N	
IORLW	k	立即数与 WREG 作逻辑或运算	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	立即数与 WREG 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	立即数与 WREG 进行逻辑异或运算	1	0000 1010 kkkk kkkk	Z,N	
数据存储器 ↔ 程序存储器操作类指令						
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，虽然其数据锁存器中的值为 1，但此时外部器件将该引脚驱动为低电平，则被写回数据总线的数据值将是 0。

2： 当对 TMR0 寄存器 （以及其他适用的寄存器）执行该指令时 （并且 d = 1），如果已对其分配了预分频器，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单元内存储的都是合法的指令。

© 2007 Microchip Technology Inc.

25.1.1 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 的内容与 8 位立即数 K 相加，结果保存在 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： 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) → 目标寄存器

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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) → 目标寄存器

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、进位标志位与数据存储单元 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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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 立即数与 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 作逻辑与运算。结果保存在 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 与 f 作逻辑与运算**

语法： ANDWF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： (W) .AND. (f) → 目标寄存器

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 的内容作逻辑与运算。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 ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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 ≤ n ≤ 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，那么程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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，那么 程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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 ≤ 1277

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，那么程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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，那么程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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，那么程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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，那么程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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： 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，所以新地址将为所以新地址将为 PC + 2 + 2n。该指令是一条双周期指令。

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 中的位 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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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 中的位 b 为 0，则跳过下一条指令。即在位 b 为 0 时，丢弃下一条指令 （执行当前指令期间取指）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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见**第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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双字指令：

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 中的位 b 为 1，则跳过下一条指令。即在位 b 为 1 时，丢弃下一条指令 （执行当前指令期间取指）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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见**第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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双字指令：

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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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。因为 PC 要先递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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，那么程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

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 2 MB 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 ≤ f ≤ 255
a ∈ [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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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。而且还会复位 WDT 的后分频器。状态位 $\overline{TO}$ 和 $\overline{PD}$ 被置 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 **将 f 取补**

语法： COMF f {,d {,a}}

操作数： 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： $\bar{f} \rightarrow$ 目标寄存器

受影响的状态位： N 和 Z

机器码：

0001	11da	ffff	ffff

说明： 将寄存器 f 的内容取补。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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见**第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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，转而执行一条 NOP 指令，从而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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见**第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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

如果被跳过的指令后面跟有双字指令：

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： $0 \le f \le 255$
$a \in [0,1]$

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： 通过执行无符号减法，将数据存储单元 f 的内容与 W 的内容作比较。

如果 f 的内容大于 WREG 的内容，则所取的指令会被丢弃，转而执行一条 NOP 指令，从而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 且使能了扩展的指令集，只要 $f \le 95$ （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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

如果被跳过的指令后面跟有双字指令：

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： $0 \le f \le 255$
$a \in [0,1]$

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： 通过执行无符号减法，将数据存储单元 f 的内容与 W 的内容作比较。

如果 f 的内容小于 W 的内容，则所取的指令被丢弃并执行一条 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）。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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

如果被跳过的指令后面跟有双字指令：

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] 那么，
(W<3:0>) + 6 → W<3:0> ；
否则
(W<3:0>) → W<3:0>

如果 [W<7:4> > 9] 或 [C = 1] 那么，
(W<7:4>) + 6 → W<7:4>，
C = 1 ；
否则
(W<7:4>) → 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 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 → 目标寄存器

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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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 → 目标寄存器，
结果为 0 时跳过

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，则丢弃已经取指的指令而执行一条 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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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双字指令：

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 → 目标寄存器，
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 丢弃已经取指的指令而执行一条 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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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双字指令：

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： $0 \le k \le 1048575$

操作： $k \rightarrow PC<20:1>$

受影响的状态位： 无

机器码：

第一个字 （k<7:0>）	1110	1111	k_7kkk	$kkkk_0$
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>	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： $0 \le f \le 255$

$d \in [0,1]$

$a \in [0,1]$

操作： (f)+ 1 → 目标寄存器

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 且使能了扩展的指令集，只要 $f \le 95$ （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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 → 目标寄存器，
如果结果为 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，则丢弃已经取指的指令而执行一条 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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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双字指令：

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 → 目标寄存器，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 丢弃已经取指的指令而执行一条 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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)
注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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双字指令：

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 **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 作逻辑或运算。结果保存在 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) → 目标寄存器

受影响的状态位： N 和 Z

机器码：

0001	00da	ffff	ffff

说明： 将 W 与寄存器 f 的内容作逻辑或运算。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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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 ≤ f ≤ 2
0 ≤ k ≤ 4095

操作： k → 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 载入由 f 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	写立即数 k 的 MSB 到 FSRfH
译码	读立即数 k 的 LSB	处理数据	将立即数 k 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 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作： f → 目标寄存器

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 的内容送入目标单元。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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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 **将源寄存器的内容送入目标寄存器**

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”的内容送入目标寄存器“f_d”。
源寄存器“f_s”可以是 4096 字节数据空间（000h 到 FFFh）中的任何地址，目标寄存器“f_d”也可以是 000h 到 FFFh 中的任何地址。

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送入外设寄存器（如发送缓冲器或 I/O 端口）的场合非常有用。

MOVFF 指令中不能使用 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 源寄存器 f	处理数据	空操作
译码	空操作 非无效读取	空操作	写 目标寄存器 f

示例： MOVFF REG1, REG2

执行指令前
REG1 = 33h
REG2 = 11h
执行指令后
REG1 = 33h
REG2 = 33h

MOVLB **将立即数送入 BSR 的低半字节**

语法： MOVLW 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 寄存器中的数据送入寄存器 f。
f 可以是 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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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 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 存储高字节。

W 的内容不变。

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，但不会反映到相应的标志位。

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 的内容做无符号的乘法运算。运算的 16 位结果存储在 PRODH:PRODL 寄存器对中。其中 PRODH 存储高字节。W 和 f 的内容都不变。

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，但不会反映到相应的标志位。

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见**第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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 （向下 1 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 （向下一级） = 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 1 KB）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。该指令是一条双周期指令。

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： 此指令可提供一种用软件实现 $\overline{\text{MCLR}}$ 复位的方法。

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 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）的内容保持不变。

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>) → 目标寄存器 <n + 1>,
(f<7>) → C,
(C) → 目标寄存器 <0>

受影响的状态位： C、N 和 Z

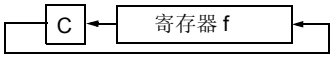
机器码：

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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。.



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>) → 目标寄存器 <n+1>,
(f<n>) → 目标寄存器 <0>

受影响的状态位： N 和 Z

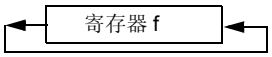
机器码：

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。.



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>) → 目标寄存器 <n – 1>,
(f<0>) → C,
(C) → 目标寄存器 <7>

受影响的状态位： C、 N 和 Z

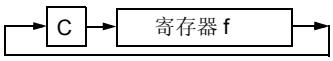
机器码：

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。.



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>) → 目标寄存器 <n − 1>,
(f<n>) → 目标寄存器 <7>

受影响的状态位： N 和 Z

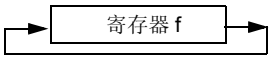
机器码：

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。



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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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 → 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 ∈ [0,1]
a ∈ [0,1]

操作： (W) - (f) - ($\overline{C}$) → 目标寄存器

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，结果存储在寄存器 f 中 （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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见**第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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。结果保存在 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) → 目标寄存器

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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 ；（二进制补码）
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) - ($\overline{C}$) → 目标寄存器

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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)
; [二进制补码]
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>) → 目标寄存器 <7:4>，
(f<7:4>) → 目标寄存器 <3:0>

受影响的状态位： 无

机器码：

0011	10da	ffff	ffff

说明： 将寄存器 f 的高半字节和低半字节互相交换。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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

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）指向程序存储器中的每个字节。TBLPTR 的寻址范围为 2 MB。

TBLPTR<0> = 0：程序存储器字的最低有效字节

TBLPTR<0> = 1：程序存储器字的最高有效字节

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。（关于对闪存存储器编程的更多详情，请参见**第 5.0 节“存储器构成”**。）

TBLPTR（一个 21 位指针）指向程序存储器中的每个字节。TBLPTR 的寻址范围为 2 MB。TBLPTR 的 LSb 选择要访问的程序存储器单元。

TBLPTR<0> = 0：程序存储器字的最低有效字节

TBLPTR<0> = 1：程序存储器字的最高有效字节

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 且使能了扩展的指令集，只要 f ≤ 95 （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节 “立即数变址寻址模式中面向字节和位的指令”**。

指令字数： 1

指令周期数： 1(2)

注： 如果跳过的指令后面跟有双字指令，则执行该指令需要 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双字指令：

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 **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。结果保存在 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 **W 与 f 作逻辑异或运算**

语法： XORWF f {,d {,a}}

操作数： $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

操作： (W) .XOR. (f) → 目标寄存器

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 且使能了扩展的指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 **第 25.2.3 节“立即数变址寻址模式中面向字节和位的指令”**。

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.

25.2 扩展的指令集

除了PIC18指令集的75个标准指令之外，PIC18F87J11系列器件还提供针对内核 CPU 功能的可选扩展指令。这些新增的功能包括 8 个额外的指令，它们可以实现间接和变址寻址操作，以及许多标准 PIC18 指令执行立即数变址寻址。

在未编程器件上扩展指令集的这些额外功能在默认情况下是允许的。用户必须在编程期间正确置 1 或清零 XINST 配置位，以允许或禁用这些功能。

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。 其中的两条指令 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 25-3 提供了扩展指令集中的指令汇总。**第 25.2.2 节“扩展的指令集”**对这些指令进行了详细说明。表 25-1（第 330 页）提供了标准和扩展的 PIC18 指令集的操作码字段说明。

注： 扩展的指令集和变址立即数偏置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。

25.2.1 扩展指令的语法

大部分扩展指令都使用变址参数，同时使用一个指针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，它就会标出一个错误。

当使能扩展的指令集时，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。欲知更多信息，请参见**第 25.2.3.1 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 25-3： PIC18 指令集的扩展

助记符，操作数		说明	指令周期数	16 位指令字 MSb			LSb	受影响的状态位
ADDFSR	f, k	将立即数与 FSR 相加	1	1110	1000	ffkk	kkkk	无
ADDULNK	k	将立即数与 FSR 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）数值装入第一个字	2	1110	1011	0zzz	zzzz	无
		将 f_d （目标）数值装入第二个字		1111	ffff	ffff	ffff	
MOVSS	z_s, z_d	将 z_s （源）数值装入第一个字	2	1110	1011	1zzz	zzzz	无
		将 z_d （目标）数值装入第二个字		1111	xxxx	xzzz	zzzz	
PUSHL	k	将立即数保存在 FSR2 后，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

25.2.2 扩展的指令集

ADDFSR **FSR 的内容与立即数相加**

语法： ADDFSR f, k

操作数： $0 \le k \le 63$
$f \in [0, 1, 2]$

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： $0 \le k \le 63$

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 装入 PC，执行一条 RETURN 指令。

执行该指令需要两个周期；在第二个周期执行一条 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，并同时取出下一条指令。

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 zzzz	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）中的任何位置。

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 非无效读取	空操作	写目标寄存器 f

示例： MOVSF [05h], REG2

执行指令前
FSR2 = 80h
85h 单元内容 = 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： 将源寄存器的内容送入目标寄存器。通过将 FSR2 中的值分别加上 7 位立即数偏移量 z_s 或 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 字节数据存储空间（000h 到 FFFh）中的任意单元。

MOVSS 指令不能使用 PCL、TOS、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，指令将作为一条 NOP 指令执行。

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 **将立即数保存在 FSR2，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 指定的数据存储单元。操作完后 FSR2 减 1。

该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： 用寄存器 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 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)

25.2.3 立即数变址寻址模式中面向字节和位的指令

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。

一旦使能扩展的指令集，除了可以使用 8 条新命令之外，还可以使用将立即数作为偏移量的变址寻址模式（**第 5.6.1 节“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。

当禁用扩展的指令集时，被嵌入在操作码中的地址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）。当使能扩展的指令集且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，即所有面向字节或位的指令，或者几乎半数的 PIC18 内核指令——在使能了扩展的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 25.2.3.1 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，在使能了扩展的指令集时，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。

25.2.3.1 标准 PIC18 命令的扩展指令语法

当使能了扩展的指令集时，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。因为在扩展的指令集中，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（禁止扩展的指令集时）正好相反。在变址寻址模式中，声明快速操作 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的指令集的语言支持。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。

25.2.4 使能扩展的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的指令集，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。当使能了扩展的指令集时，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 PIC18F87J11 系列时，代码类型是非常重要的。在使用扩展的指令集时，用 C 语言编写的代码较长的重入应用程序会运行良好。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) → 目标寄存器

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

25.2.5 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 PIC18F87J11 系列器件的扩展指令集。包括 MPLAB C18 C 语言编译器、MPASM 汇编器和 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 将把该器件的默认配置位自动置 1。在禁用扩展的指令集和立即数变址寻址模式时，XINST 配置位的默认设置是 0。在编程过程中必须将 XINST 位置 1 才能正确地利用扩展的指令集开发应用程序。

要使用扩展的指令集开发软件，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.

26.0 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 谙叻抡嫫 ˜
- 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

26.1 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。

26.2 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

26.3 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 和 PIC24 系列单片机及 dsPIC30F 和 dsPIC33 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。

26.4 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

26.5 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

26.6 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.

26.7 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。

26.8 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® 闪存 MCU 和 dsPIC® DSC 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。

26.9 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。

26.10 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。

26.11 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。

26.12 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。

26.13 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 鲍 sPICDEM 菔 æ/ 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）。

© 2007 Microchip Technology Inc.

27.0 电气规范

绝对最大值 (†)

偏置电压下的环境温度 ..... -40°C 至 +100°C
储存温度 ..... -65°C 至 +150°C
任何数字输入引脚或 $\overline{MCLR}$ 引脚 （ VDD 除外）相对于 VSS 的电压 ..... -0.3V 至 6.0V
组合的数字和模拟引脚相对于 VSS 的电压 ..... -0.3V 至（VDD + 0.3V）
VDDCORE 相对于 VSS 的电压 ..... -0.3V 至 2.75V
VDD 相对于 VSS 的电压 ..... -0.3V 至 4.0V
总功耗 **（注 1）** 1.0W
VSS 引脚最大输出电流 ..... 300 mA
VDD 引脚最大输入电流 ..... 250 mA
输入钳位电流 IIK （VI < 0 或 VI > VDD）**（注 2）** ..... ±0 mA
输出钳位电流 IOK （VO < 0 或 VO > VDD）**（注 2）** ..... ±0 mA
PORTB 和 PORTC 中任何 I/O 引脚的最大输出灌电流 ..... 25 mA
PORTD、 PORTE 和 PORTJ 中任何 I/O 引脚的最大输出灌电流 ..... 8 mA
PORTA、 PORTF、 PORTG 和 PORTH 中任何 I/O 引脚的最大输出灌电流 ..... 2 mA
PORTB 和 PORTC 中任何 I/O 引脚的最大输出拉电流 ..... 25 mA
PORTD、 PORTE 和 PORTJ 中任何 I/O 引脚的最大输出拉电流 ..... 8 mA
PORTA、 PORTF、 PORTG 和 PORTH 中任何 I/O 引脚的最大输出拉电流 ..... 2 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： 功耗按如下公式计算：
$Pdis = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL}) + \sum (V_{TPOUT} \times I_{TPOUT})$

2： 不使用钳位二极管。

† 注意： 如果器件工作条件超过上述 “绝对最大值”，可能会对器件造成永久性损坏。上述值仅为运行条件极大值，我们不建议器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其稳定性会受到影响。

图 27-1： PIC18F87J11 系列电压－频率关系图，使能稳压器 （工业级）

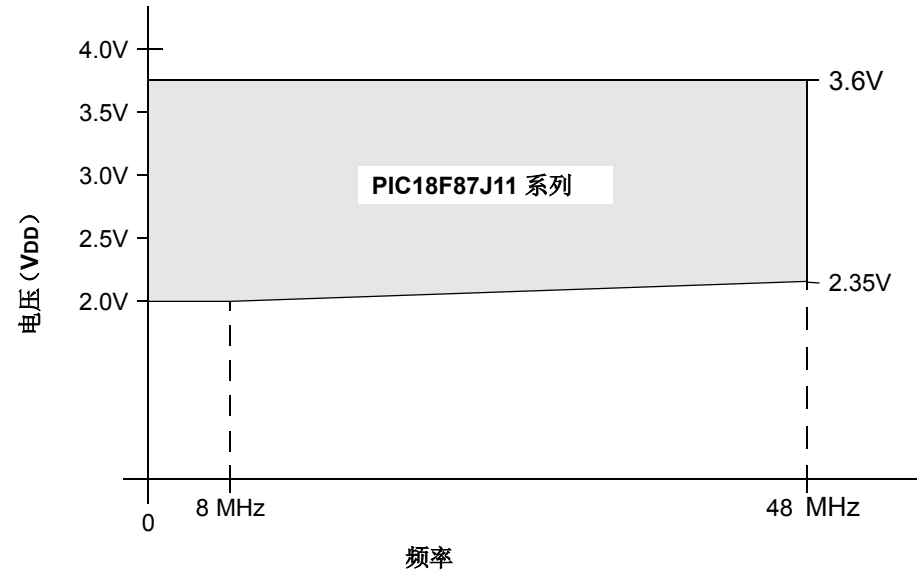
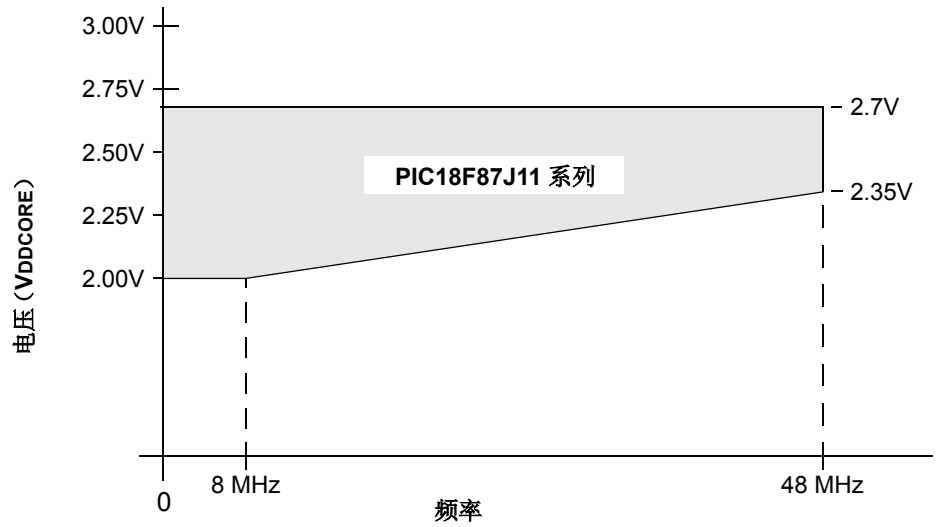


图 27-2： PIC18F87J11 系列电压－频率关系图，禁止稳压器 （工业级） (1)



注 1： 当禁止片上稳压器时， VDD 和 VDDCORE 必须满足以下条件：VDDCORE ≤ VDD ≤ 3.6V。

© 2007 Microchip Technology Inc.

27.1 直流特性： 供电电压 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）				
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	VDDCORE 2.0	— —	3.6 3.6	V V	ENVREG 连接到 VSS ENVREG 连接到 VDD
D001B	VDDCORE	单片机内核的外部电源	2.0	—	2.7	V	ENVREG 连接到 VSS
D001C	AVDD	模拟供电电压	VDD – 0.3	—	VDD + 0.3	V	
D001D	AVSS	模拟地电位	VSS – 0.3	—	VSS + 0.3	V	
D002	VDR	RAM 数据保持电压 (1)	1.5	—	—	V	
D003	VPOR	VDD 上电复位电压	—	—	0.7	V	详细信息请参见第 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	详细信息请参见第 4.3 节 “上电复位 （POR）”。
D005	VBOR	欠压复位电压	—	1.8	—	V	

注 1： 该电压是在休眠模式或器件复位状态下，在不丢失 RAM 数据前提下的最小 VDD。

27.2 直流规范： 掉电电流和供电电流 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）				
参数编号	器件	典型值	最大值	单位	条件	
	掉电电流 （IPD） (1)					
	所有器件	0.5	1.4	μA	-40°C	VDD = 2.0V(4) （休眠模式）
		0.5	1.4	μA	+25°C	
		5.5	10.2	μA	+85°C	
	所有器件	0.6	1.5	μA	-40°C	VDD = 2.5V(4) （休眠模式）
		0.6	1.5	μA	+25°C	
		6.8	12.6	μA	+85°C	
	所有器件	2.9	7	μA	-40°C	VDD = 3.3V(5) （休眠模式）
		3.6	7	μA	+25°C	
		9.6	19	μA	+85°C	

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。

在正常工作模式下，所有 IDD 测量的测试条件为：

OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；

$\overline{MCLR}$ = VDD ； 根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

© 2007 Microchip Technology Inc.

27.2 直流规范： 掉电电流和供电电流 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD）(2,3)						
	所有器件	5	14.2	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 31 kHz （**RC_RUN** 模式，内部振荡源）
		5.5	14.2	μA	+25°C		
		10	19.0	μA	+85°C		
	所有器件	6.8	16.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		7.6	16.5	μA	+25°C		
		14	22.4	μA	+85°C		
	所有器件	37	84	μA	-40°C	VDD = 3.3V(5)	
		51	84	μA	+25°C		
		72	108	μA	+85°C		
	所有器件	0.43	0.82	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz （**RC_RUN** 模式，内部振荡源）
		0.47	0.82	mA	+25°C		
		0.52	0.95	mA	+85°C		
	所有器件	0.52	0.98	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.57	0.98	mA	+25°C		
		0.63	1.10	mA	+85°C		
	所有器件	0.59	0.96	mA	-40°C	VDD = 3.3V(5)	
		0.65	0.96	mA	+25°C		
		0.72	1.18	mA	+85°C		
	所有器件	0.88	1.45	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz （**RC_RUN** 模式，内部振荡源）
		1	1.45	mA	+25°C		
		1.1	1.58	mA	+85°C		
	所有器件	1.2	1.72	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		1.3	1.72	mA	+25°C		
		1.4	1.85	mA	+85°C		
	所有器件	1.3	1.87	mA	-40°C	VDD = 3.3V(5)	
		1.4	1.87	mA	+25°C		
		1.5	1.96	mA	+85°C		

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。

在正常工作模式下，所有 IDD 测量的测试条件为：

OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；

MCLR = VDD ；根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

27.2 直流规范： 掉电电流和供电电流 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD）（续）(2,3)						
	所有器件	3	9.4	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 31 kHz （RC_IDLE 模式，内部振荡源）
		3.3	9.4	μA	+25°C		
		8.5	17.2	μA	+85°C		
	所有器件	4	10.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		4.3	10.5	μA	+25°C		
		10.3	19.5	μA	+85°C		
	所有器件	34	82	μA	-40°C	VDD = 3.3V(5)	
		48	82	μA	+25°C		
		69	105	μA	+85°C		
	所有器件	0.33	0.75	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz （RC_IDLE 模式，内部振荡源）
		0.37	0.75	mA	+25°C		
		0.41	0.84	mA	+85°C		
	所有器件	0.39	0.78	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.42	0.78	mA	+25°C		
		0.47	0.91	mA	+85°C		
	所有器件	0.43	0.82	mA	-40°C	VDD = 3.3V(5)	
		0.48	0.82	mA	+25°C		
		0.54	0.95	mA	+85°C		
	所有器件	0.53	0.98	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz （RC_IDLE 模式，内部振荡源）
		0.57	0.98	mA	+25°C		
		0.61	1.12	mA	+85°C		
	所有器件	0.63	1.14	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.67	1.14	mA	+25°C		
		0.72	1.25	mA	+85°C		
	所有器件	0.7	1.27	mA	-40°C	VDD = 3.3V(5)	
		0.76	1.27	mA	+25°C		
		0.82	1.45	mA	+85°C		

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下，所有 IDD 测量的测试条件为：
OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；
MCLR = VDD ； 根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

© 2007 Microchip Technology Inc.

27.2 直流规范： 掉电电流和供电电流 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD）（续）(2,3)						
	所有器件	0.17	0.35	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz （PRI_RUN 模式，EC 振荡器）
		0.18	0.35	mA	+25°C		
		0.20	0.42	mA	+85°C		
	所有器件	0.29	0.52	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.31	0.52	mA	+25°C		
		0.34	0.61	mA	+85°C		
	所有器件	0.59	1.1	mA	-40°C	VDD = 3.3V(5)	
		0.44	0.85	mA	+25°C		
		0.42	0.85	mA	+85°C		
	所有器件	0.70	1.25	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz （PRI_RUN 模式，EC 振荡器）
		0.75	1.25	mA	+25°C		
		0.79	1.36	mA	+85°C		
	所有器件	1.10	1.7	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		1.10	1.7	mA	+25°C		
		1.12	1.82	mA	+85°C		
	所有器件	1.55	1.95	mA	-40°C	VDD = 3.3V(5)	
		1.47	1.89	mA	+25°C		
		1.54	1.92	mA	+85°C		
	所有器件	9.9	14.8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 48 MHz （PRI_RUN 模式，EC 振荡器）
		9.5	14.8	mA	+25°C		
		10.1	15.2	mA	+85°C		
	所有器件	13.3	19.2	mA	-40°C	VDD = 3.3V(5)	
		12.2	18.7	mA	+25°C		
		12.1	18.7	mA	+85°C		

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。

在正常工作模式下，所有 IDD 测量的测试条件为：

OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；

MCLR = VDD ；根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

27.2 直流规范： 掉电电流和供电电流 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD）（续）(2,3)						
	所有器件	2.8	3.8	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz，16 MHz 内部（**PRI_RUN HSPLL** 模式）
		3.0	3.8	mA	+25°C		
		3.2	4.1	mA	+85°C		
	所有器件	4.5	5.2	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		4.4	5.2	mA	+25°C		
		4.5	5.2	mA	+85°C		
	所有器件	5.7	6.7	mA	-40°C	VDD = 3.3V(5)	
		5.5	6.3	mA	+25°C		
		5.3	6.3	mA	+85°C		
	所有器件	10.8	13.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 12 MHz，48 MHz 内部（**PRI_RUN HSPLL** 模式）
		10.8	13.5	mA	+25°C		
		9.9	13.0	mA	+85°C		
	所有器件	13.4	17.1	mA	-40°C	VDD = 3.3V(5)	
		12.3	16.2	mA	+25°C		
		11.2	15.5	mA	+85°C		

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下，所有 IDD 测量的测试条件为：
OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；
MCLR = VDD ； 根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

© 2007 Microchip Technology Inc.

27.2 直流规范： 掉电电流和供电电流 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD）（续）(2,3)						
	所有器件	0.10	0.26	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz （**PRI_IDLE** 模式，EC 振荡器）
		0.07	0.18	mA	+25°C		
		0.09	0.22	mA	+85°C		
	所有器件	0.25	0.48	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.13	0.30	mA	+25°C		
		0.10	0.26	mA	+85°C		
	所有器件	0.45	0.68	mA	-40°C	VDD = 3.3V(5)	
		0.26	0.45	mA	+25°C		
		0.30	0.54	mA	+85°C		
	所有器件	0.36	0.60	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz （**PRI_IDLE** 模式，EC 振荡器）
		0.33	0.56	mA	+25°C		
		0.35	0.56	mA	+85°C		
	所有器件	0.52	0.81	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.45	0.70	mA	+25°C		
		0.46	0.70	mA	+85°C		
	所有器件	0.80	1.15	mA	-40°C	VDD = 3.3V(5)	
		0.66	0.98	mA	+25°C		
		0.65	0.98	mA	+85°C		
	所有器件	5.2	6.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 48 MHz （**PRI_IDLE** 模式，EC 振荡器）
		4.9	5.9	mA	+25°C		
		3.4	4.5	mA	+85°C		
	所有器件	6.2	8.4	mA	-40°C	VDD = 3.3V(5)	
		5.9	7.5	mA	+25°C		
		5.8	7.5	mA	+85°C		

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。

在正常工作模式下，所有 IDD 测量的测试条件为：

OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；

MCLR = VDD ；根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

27.2 直流规范： 掉电电流和供电电流 PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数编号	器件	典型值	最大值	单位	条件		
	供电电流 （IDD）（续）(2,3)						
	所有器件	18	35	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) （**SEC_RUN** 模式，Timer1 作为时钟源）
		19	35	µA	+25°C		
		28	49	µA	+85°C		
	所有器件	20	45	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		21	45	µA	+25°C		
		32	61	µA	+85°C		
	所有器件	0.06	0.11	mA	-40°C	VDD = 3.3V(5)	
		0.07	0.11	mA	+25°C		
		0.09	0.15	mA	+85°C		
	所有器件	14	28	µA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) （**SEC_IDLE** 模式，Timer1 作为时钟源
		15	28	µA	+25°C		
		24	43	µA	+85°C		
	所有器件	15	31	µA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		16	31	µA	+25°C		
		27	50	µA	+85°C		
	所有器件	0.05	0.10	mA	-40°C	VDD = 3.3V(5)	
		0.06	0.10	mA	+25°C		
		0.08	0.14	mA	+85°C		

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下，所有 IDD 测量的测试条件为：
OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；
MCLR = VDD ； 根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

© 2007 Microchip Technology Inc.

27.2 直流规范： 掉电电流和供电电流
PIC18F87J11 系列 （工业级）

PIC18F87J11 系列 （工业级）		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数编号	器件	典型值	最大值	单位	条件		
D022	模块差分电流 （ΔIWDT， ΔIOSCB， ΔIAD）						
	看门狗定时器	2.1	7.0	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	
		2.2	7.0	μA	+25°C		
		4.3	9.5	μA	+85°C		
		3.0	8.0	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		3.1	8.0	μA	+25°C		
		5.5	10.4	μA	+85°C		
		5.9	12.1	μA	-40°C	VDD = 3.3V	
		6.2	12.1	μA	+25°C		
		6.9	13.6	μA	+85°C		
D025 (ΔIOSCB)	Timer1 振荡器	14	24	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	Timer1 为 32 kHz(3)
		15	24	μA	+25°C		
		23	36	μA	+85°C		
		17	26	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	Timer1 为 32 kHz(3)
		18	26	μA	+25°C		
		25	38	μA	+85°C		
		19	35	μA	-40°C	VDD = 3.3V	Timer1 为 32 kHz(3)
		21	35	μA	+25°C		
		28	44	μA	+85°C		
D026 (ΔIAD)	A/D 转换器	3.0	10.0	μA	-40°C至+85°C	VDD = 2.0V, VDDCORE = 2.0V(4)	A/D 启动，但不进行转换
		3.0	10.0	μA	-40°C至+85°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		3.2	11.0	μA	-40°C至+85°C	VDD = 3.3V	

图注： TBD = 待定

注 1： 休眠模式下的掉电电流不是由振荡器类型决定的。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 或者 VSS，以及禁止所有会带来新增电流的功能部件 （比如 WDT、 Timer1 振荡器和 BOR 等）时测得的。

2： 供电电流主要是由工作电压、频率和模式决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在正常工作模式下，所有 IDD 测量的测试条件为：
OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD ；
MCLR = VDD ；根据具体应用使能或禁止 WDT。

3： 标准低成本 32 kHz 晶振的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶振成本要高很多。

4： 禁止稳压器 （ENVREG = 0，连接到 VSS）。

5： 使能稳压器 （ENVREG = 1，连接到 VDD， REGSLP = 1）。

27.3 直流特性： PIC18F87J11 系列 （工业级）

直流特性			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）			
参数编号	符号	特性	最小值	最大值	单位	条件
	VIL	**输入低电压**				
		所有 I/O 端口：				
D030		带 TTL 缓冲器	VSS	0.15 VDD	V	
D031		带施密特触发缓冲器	VSS	0.2 VDD	V	
D032		$\overline{MCLR}$	VSS	0.2 VDD	V	
D033		OSC1	VSS	0.3 VDD	V	HS 和 HSPLL 模式
D033A		OSC1	VSS	0.2 VDD	V	EC 和 ECPLL 模式
D034		T13CKI	VSS	0.3	V	
	VIH	**输入高电压**				
		具有模拟功能的 I/O 端口：				
D040		带 TTL 缓冲器	0.25 VDD + 0.8V	VDD	V	VDD < 3.3V
D041		带施密特触发缓冲器	0.8 VDD	VDD	V	
		仅用作数字功能的 I/O 端口：				
		带 TTL 缓冲器	0.25 VDD + 0.8V	5.5	V	VDD < 3.3V
			2.0	5.5	V	3.3V ≤ VDD ≤ 3.6V
		带施密特触发缓冲器	0.8 VDD	5.5	V	
D042		$\overline{MCLR}$	0.8 VDD	VDD	V	
D043		OSC1	0.7 VDD	VDD	V	HS 和 HSPLL 模式
D043A		OSC1	0.8 VDD	VDD	V	EC 和 ECPLL 模式
D044		T13CKI	1.6	VDD	V	
	IIL	**输入泄漏电流 (1,2)**				
D060		I/O 端口	—	±1	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态
D061		$\overline{MCLR}$	—	±1	μA	Vss ≤ VPIN ≤ VDD
D063		OSC1	—	±5	μA	Vss ≤ VPIN ≤ VDD
	IPU	**弱上拉电流**				
D070	IPURB	PORTB 弱上来电流	30	240	μA	VDD = 3.3V， VPIN = VSS

注 1： 负电流定义为自引脚流出的电流。

© 2007 Microchip Technology Inc.

27.3 直流特性： PIC18F87J11 系列 （工业级）

直流特性			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）			
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：				
		PORTA、 PORTF、 PORTG 和PORTH	—	0.4	V	IOL = 2 mA， VDD = 3.3V， -40°C 至 +85°C
		PORTD、 PORTE 和 PORTJ	—	0.4	V	IOL = 3.4 mA， VDD = 3.3V， -40°C 至 +85°C
		PORTB 和 PORTC	—	0.4	V	IOL = 3.4 mA， VDD = 3.3V， -40°C 至 +85°C
D083		OSC2/CLKO （EC 和 ECPLL 模式）	—	0.4	V	IOL = 1.6 mA， VDD = 3.3V， -40°C 至 +85°C
	VOH	**输出高电压 (1)**				
D090		I/O 端口：			V	
		PORTA、 PORTF、 PORTG 和PORTH	2.4	—	V	IOH = -2 mA， VDD = 3.3V， -40°C 至 +85°C
		PORTD、 PORTE 和 PORTJ	2.4	—	V	IOH = -2 mA， VDD = 3.3V， -40°C 至 +85°C
		PORTB 和 PORTC	2.4	—	V	IOH = -2 mA， VDD = 3.3V， -40°C 至 +85°C
D092		OSC2/CLKO （INTOSC、 EC 和 ECPLL 模式）	2.4	—	V	IOH = -1 mA， VDD = 3.3V， -40°C 至 +85°C
		输出引脚上的容性负载规范				
D100(4)	COSC2	OSC2 引脚	—	15	pF	当使用外部时钟驱动 OSC1 时，处于 HS 模式
D101	CIO	所有 I/O 引脚和 OSC2	—	50	pF	要满足交流时序规范
D102	CB	SCLx 和 SDAx	—	400	pF	I2C™ 规范

注 1： 负电流定义为自引脚流出的电流。

表 27-1： 存储器编程要求

直流规范			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）				
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
		闪存程序存储器					
D130	EP	存储单元耐用性	1K	10K	—	E/W	-40°C 至 +85°C
D131	VPR	用于读的 VDD	VMIN	—	3.6	V	VMIN = 最小工作电压
D132B	VPEW	用于自定时操作的 VDD	VMIN	—	3.6	V	VMIN = 最小工作电压
D133A	TIW	自定时的写周期时间	—	2.8	—	ms	
D134	TRETD	保存时间	20	—	—	年	假如没有违反其他规范
D135	IDDP	编程期间的供电电流	—	10	14	mA	
D1xxx	TWE	每个擦除周期的写入次数	—	—	1		

† 除非另外声明，“典型值”栏中的数据均为 3.3V， 25°C 下的值。这些参数仅作为设计参考，未经测试。

© 2007 Microchip Technology Inc.

表 27-2： 比较器规范

工作条件：除非另外声明，否则均为 3.0V < VDD < 3.6V， -40°C < TA < +85°C							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D300	VIOFF	输入失调电压	—	±5.0	±10	mV	
D301	VICM	输入共模电压 *	0	—	AVDD – 1.5	V	
	VIRV	内部参考电压	—	±1.2(2)	—	V	±1.2%
D302	CMRR	共模抑制比 *	55	—	—	dB	
300	TRESP	响应时间 (1)*	—	150	400	ns	
301	TMC2OV	比较器模式变为输出有效 *	—	—	10	µs	

* 这些参数仅为特征值，未经测试。

注 1： 响应时间是在比较器的一个输入端电压为 (VDD – 1.5)/2，而另一个输入端从 VSS 跳变到 VDD 时测得的。

2： 容限为 ±1.2%。

表 27-3： 参考电压规范

工作条件：除非另外声明，否则均为 3.0V < VDD < 3.6V， -40°C < TA < +85°C							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D310	VRES	分辨率	VDD/24	—	VDD/32	LSb	
D311	VRAA	绝对精度	—	—	1/2	LSb	
D312	VRUR	单位电阻值 （R）	—	2k	—	Ω	
310	TSET	稳定时间 (1)	—	—	10	µs	

注 1： 稳定时间是在 CVRR = 1 并且 CVR3:CVR0 位从 0000 跳变到 1111 时测得的。

表 27-4： 内部稳压器规范

工作条件：除非另外声明，否则均为 -40°C < TA < +85°C							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	VRGOUT	稳压器输出电压 *	—	2.5	—	V	
	CF	外部滤波器电容值 *	4.7	10	—	µF	电容必须低于 ESR

* 这些参数仅为特征值，未经测试。尚未给这些规范分配 参数编号。

27.4 交流 （时序）规范

27.4.1 时序参数符号

可根据以下任一格式来创建时序参数符号：

1. TppS2ppS
2. TppS
3. TCC:ST （仅用于 I²C 规范）
4. Ts （仅用于 I²C 规范）

T			
F	频率	T	时间

小写字母 （pp）及其含意：

pp			
cc	CCP1	osc	OSC1
ck	CLKO	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ 或 $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	数据输入	t0	T0CKI
io	I/O 端口	t1	T13CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

大写字母及其含意：

S			
F	下降	P	周期
H	高	R	上升
I	无效 （高阻态）	V	有效
L	低	Z	高阻态
仅用于 I²C			
AA	输出通道	High	高
BUF	总线空闲	Low	低

TCC:ST （仅用于 I²C 规范）

CC			
HD	保持	SU	建立
ST			
DAT	保持数据输入	STO	停止条件
STA	启动条件		

© 2007 Microchip Technology Inc.

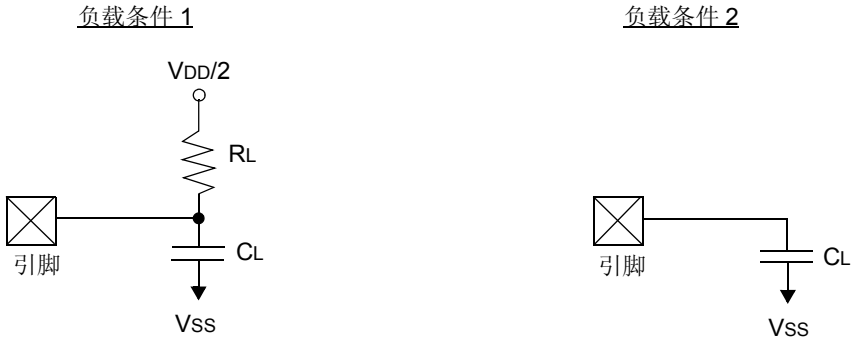
27.4.2 时序条件

表 27-5中指定的温度和电压适用于所有时序。图 27-3 规定了时序规范的负载条件。

表 27-5： 温度和电压规范——交流

交流规范	**标准工作条件（除非另外声明）** 工作温度 -40°C ≤ TA ≤ +85°C （工业级） **第 27.1 节**和**第 27.3 节**中描述了工作电压 VDD 的范围。

图 27-3： 器件时序规范的负载条件



RL = 464Ω

CL = 50 pF 适用于除 OSC2/CLKO/RA6 之外的所有引脚包括作为输出的 D 和 E 端口

CL = 15 pF 适用于 OSC2/CLKO/RA6

27.4.3 时序图和规范

图 27-4： 外部时钟时序

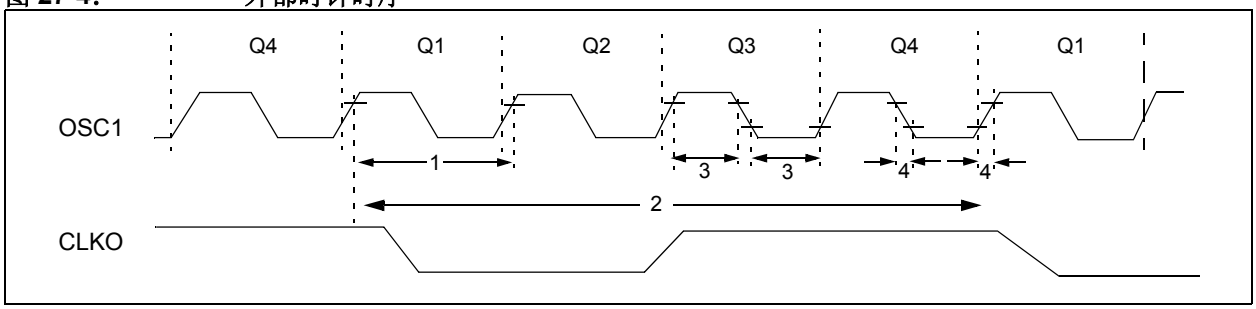


表 27-6： 外部时钟时序 要求

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)	DC	48	MHz	EC 振荡器模式
			DC	12		ECPLL 振荡器模式
		振荡器频率 (1)	4	25	MHz	HS 振荡器模式
			4	12		HSPLL 振荡器模式
1	TOSC	外部 CLKI 周期 (1)	20.8	—	ns	EC 振荡器模式
			83.3	—		ECPLL 振荡器模式
		振荡器周期 (1)	40.0	250	ns	HS 振荡器模式
			83.3	250		HSPLL 振荡器模式
2	TCY	指令周期时间 (1)	83.3	—	ns	TCY = 4/FOSC （工业级）
3	TOSL, TOSH	外部时钟输入 （OSC1）的高电平或低电平时间	10	—	ns	HS 振荡器模式
4	TOSR, TOSF	外部时钟输入 （OSC1）的上升或下降时间	—	7.5	ns	HS 振荡器模式

注 1： 对于除 PLL 的所有配置来说，指令周期时间 （TCY）等于输入振荡器时基周期的 4 倍。所有值均为在特定的振荡器模式下，器件在标准工作条件下执行代码时获得的特征数据。超过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”（没有时钟）。

© 2007 Microchip Technology Inc.

表 27-7： PLL 时钟时序规范 （VDD = 2.15V 至 3.6V）

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	—	12	MHz	
F11	FSYS	片上 VCO 系统频率	16	—	48	MHz	
F12	t_{rc}	PLL 起振时间 （锁定时间）	—	—	2	ms	
F13	ΔCLK	CLKO 稳定时间 （抗抖动）	-2	—	+2	%	

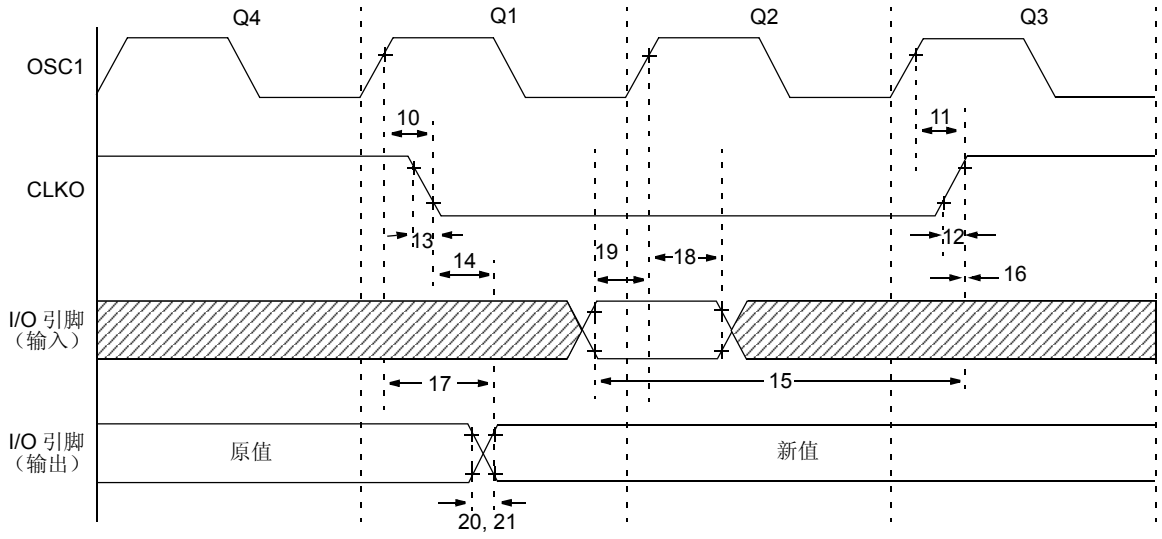
† “典型值”栏中的数据均在 3.3V， 25°C 下测得 （除非另外声明）。这些参数仅供设计参考，未经测试。

表 27-8： 内部 RC 精度 （INTOSC 和 INTRC 时钟源）

参数编号	器件	最小值	典型值	最大值	单位	条件	
	在频率 = 8 MHz、 4 MHz、 2 MHz、 1 MHz、 500 kHz、 250 kHz、 125 kHz、 31 kHz(1) 时的 INTOSC 精度						
	所有器件	-2	+/-1	2	%	+25°C	VDD = 2.7-3.3V
		-5	—	5	%	-10°C 至 +85°C	VDD = 2.0-3.3V
		-10	+/-1	10	%	-40°C 至 +85°C	VDD = 2.0-3.3V
	频率 = 31 kHz(1) 时的 INTRC 精度						
	所有器件	21.7	—	40.3	kHz		

注 1： 31 kHz 时钟所采用的精度规范取决于在给定时间时提供此时钟的时钟源 。当 INTSRC （OSCTUNE<7>）为 1 时，使用 INTOSC 精度规范。当 NTSRC 为 0 时，使用 INTRC 精度规范。

图 27-5：　　CLKO 和 I/O 时序



注：　　请参见 图 27-3 了解负载条件。

表 27-9：　　CLKO 和 I/O 时序 要求

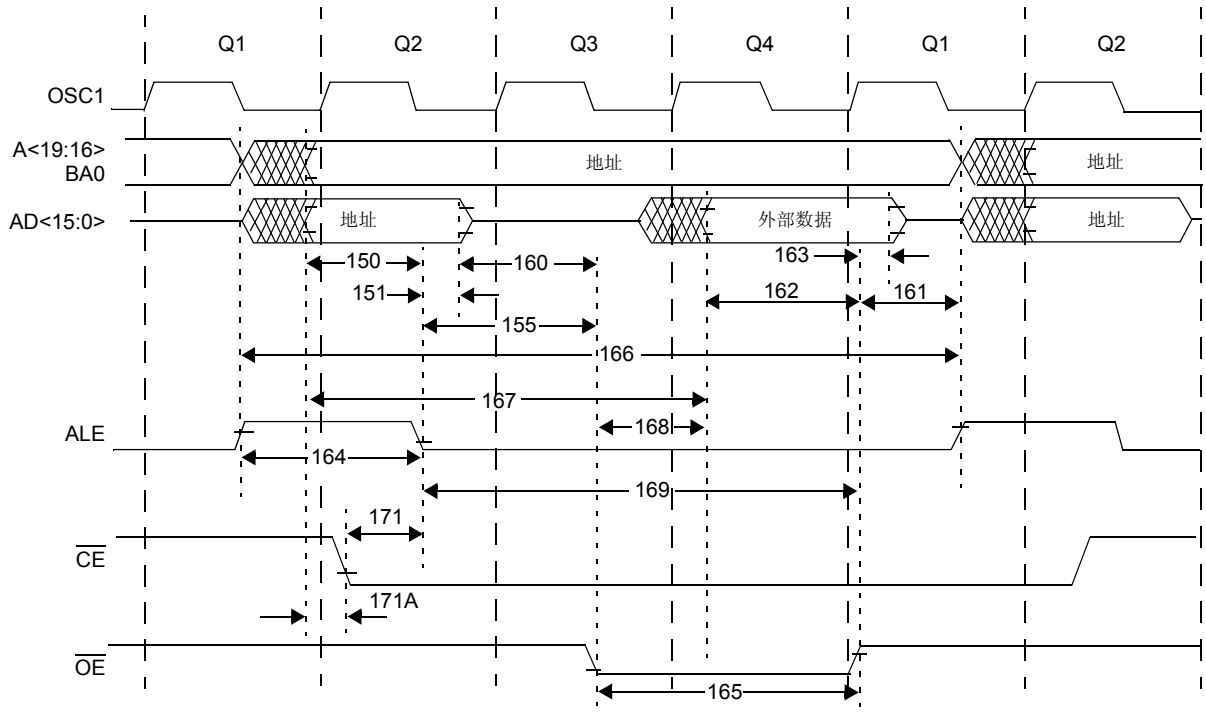
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
10	TOSH2CKL	OSC1 ↑ 到 CLKO ↓	—	75	200	ns	（注 1）
11	TOSH2CKH	OSC1 ↑ 到 CLKO ↑	—	75	200	ns	（注 1）
12	TCKR	CLKO 上升时间	—	15	30	ns	（注 1）
13	TCKF	CLKO 下降时间	—	15	30	ns	（注 1）
14	TCKL2IOV	CLKO ↓ 至端口输出有效	—	—	0.5 TCY + 20	ns	
15	TIOV2CKH	在 CLKO ↑ 之前端口输入有效	0.25 TCY + 25	—	—	ns	
16	TCKH2IOI	在 CLKO ↑ 之后端口输入保持	0	—	—	ns	
17	TOSH2IOV	OSC1 ↑ （Q1 周期）至端口输出有效	—	50	150	ns	
18	TOSH2IOI	OSC1 ↑（Q2 周期）至端口输入无效（I/O 输入保持时间）	100	—	—	ns	
19	TIOV2OSH	端口输入有效至 OSC1 ↑（I/O 输入建立时间）	0	—	—	ns	
20	TIOR	端口输出上升时间	—	—	6	ns	
21	TIOF	端口输出下降时间	—	—	5	ns	
22†	TINP	INTx 引脚高电平和低电平时间	TCY	—	—	ns	
23†	TRBP	RB7:RB4 改变 INTx 高电平和低电平时间	TCY	—	—	ns	

† 这些参数是与内部时钟边沿无关的异步事件。

注　1： 测量是在 EC 模式下进行的，其中 CLKO 输出为 4 x TOSC。

© 2007 Microchip Technology Inc.

图 27-6：　　程序存储器读时序图

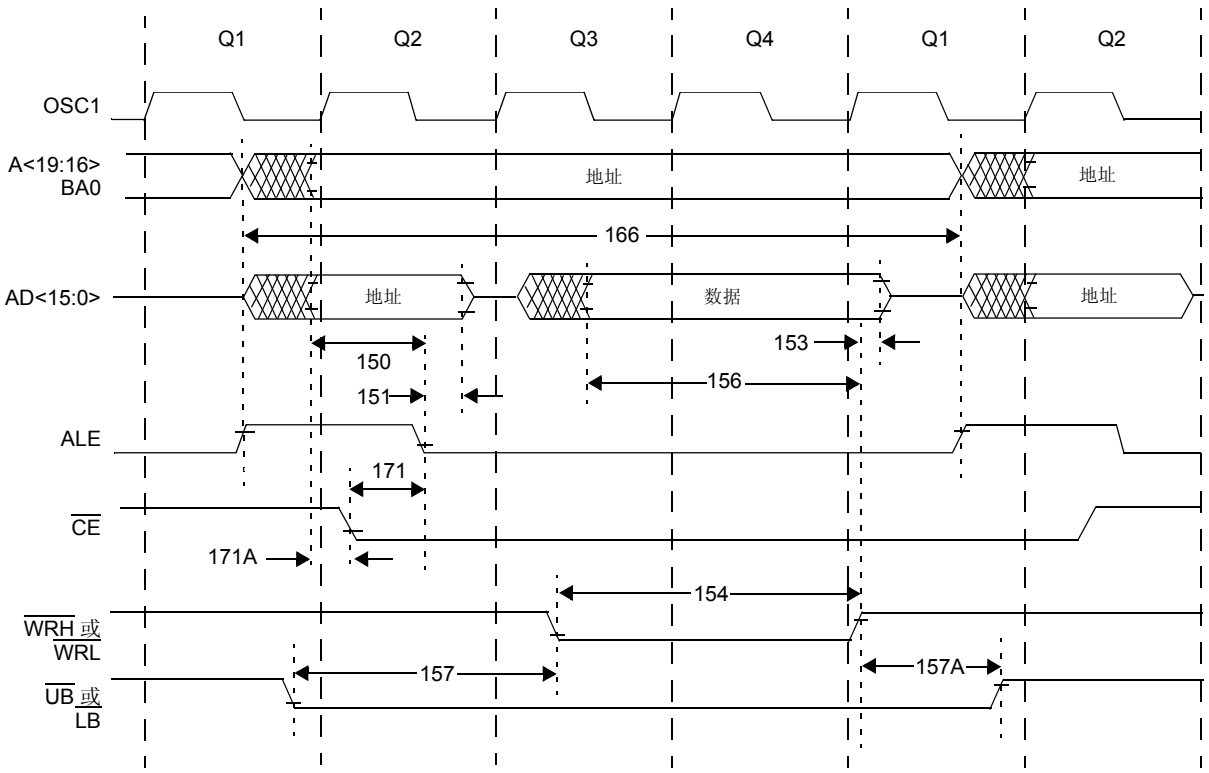


工作条件：除非另外声明，否则均为 2.0V < VCC < 3.6V，-40°C < TA < +125°C。

表 27-10：　　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
150	TadV2alL	地址输出有效至 ALE ↓ （地址建立时间）	0.25 TCY – 10	—	—	ns
151	TalL2adI	ALE ↓ 至地址输出无效 （地址保持时间）	5	—	—	ns
155	TalL2oeL	ALE ↓ 至 $\overline{OE}$ ↓	10	0.125 TCY	—	ns
160	TadZ2oeL	AD 高阻态至 $\overline{OE}$ ↓ （总线空闲至 $\overline{OE}$）	0	—	—	ns
161	ToeH2adD	$\overline{OE}$ ↑ 至 AD 驱动	0.125 TCY – 5	—	—	ns
162	TadV2oeH	在 $\overline{OE}$ ↑ 之前最低有效数据有效（数据建立时间）	20	—	—	ns
163	ToeH2adI	$\overline{OE}$ ↑ 至数据输入无效 （数据保持时间）	0	—	—	ns
164	TalH2alL	ALE 脉冲宽度	—	0.25 TCY	—	ns
165	ToeL2oeH	$\overline{OE}$ 脉冲宽度	0.5 TCY – 5	0.5 TCY	—	ns
166	TalH2alH	ALE ↑ 至 ALE ↑ （周期时间）	—	TCY	—	ns
167	Tacc	地址有效至数据有效	0.75 TCY – 25	—	—	ns
168	Toe	$\overline{OE}$ ↓ 至数据有效		—	0.5 TCY – 25	ns
169	TalL2oeH	ALE ↓ 至 $\overline{OE}$ ↑	0.625 TCY – 10	—	0.625 TCY + 10	ns
171	TalH2csL	芯片使能有效至 ALE ↓	0.25 TCY – 20	—	—	ns
171A	TubL2oeH	AD 有效至芯片使能有效	—	—	10	ns

图 27-7：　　程序存储器写时序图



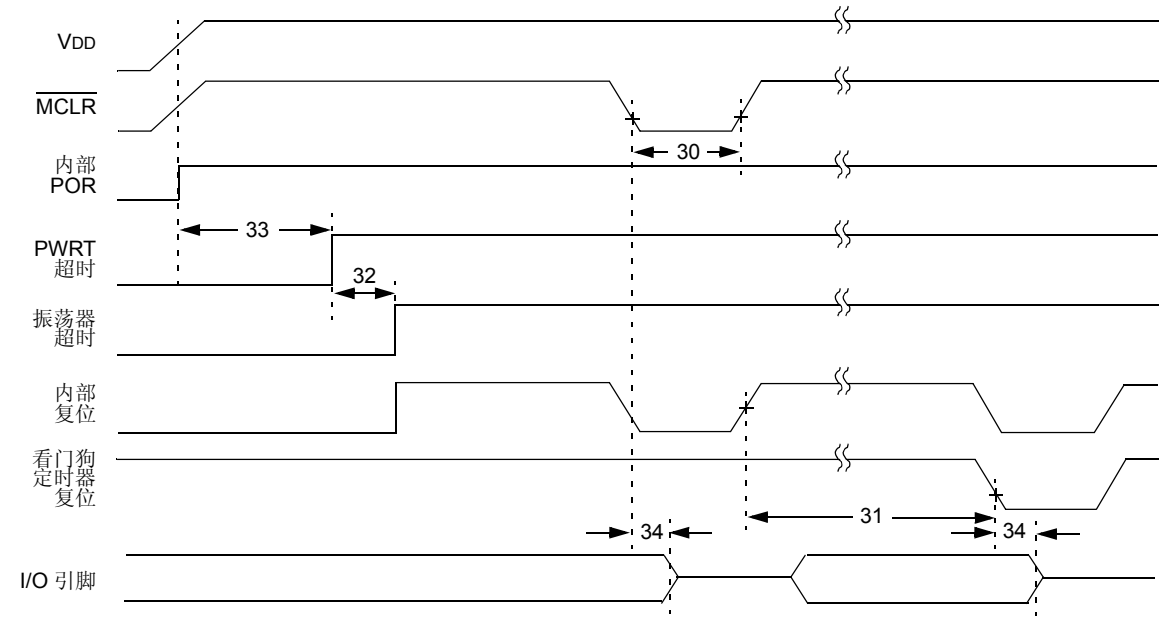
工作条件：除非另外声明，否则均为 2.0V < VCC < 3.6V，-40°C < TA < +125°C。

表 27-11：　　程序存储器写时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位
150	TadV2alL	地址输出有效至 ALE ↓（地址建立时间）	0.25 TCY – 10	—	—	ns
151	TalL2adI	ALE ↓ 至地址输出无效（地址保持时间）	5	—	—	ns
153	TwrH2adI	$\overline{WRn}$ ↑ 至数据输出无效（数据保持时间）	5	—	—	ns
154	TwrL	$\overline{WRn}$ 脉冲宽度	0.5 TCY – 5	0.5 TCY	—	ns
156	TadV2wrH	在 $\overline{WRn}$ ↑ 之前数据有效（数据建立时间）	0.5 TCY – 10	—	—	ns
157	TbsV2wrL	在 $\overline{WRn}$ ↓ 之前字节选择有效（字节选择建立时间）	0.25 TCY	—	—	ns
157A	TwrH2bsI	$\overline{WRn}$ ↑ 至字节选择无效（字节选择保持时间）	0.125 TCY – 5	—	—	ns
166	TalH2alH	ALE ↑ 至 ALE ↑（周期时间）	—	TCY	—	ns
171	TalH2csL	芯片使能有效至 ALE ↓	0.25 TCY – 20	—	—	ns
171A	TubL2oeH	AD 有效至芯片使能有效	—	—	10	ns

© 2007 Microchip Technology Inc.

图 27-8：　　复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序



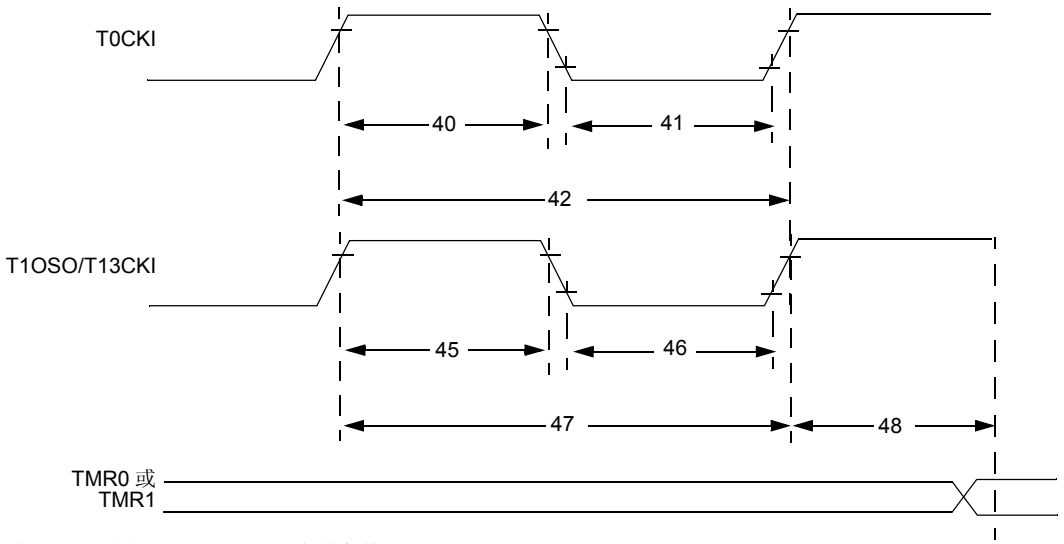
注：　请参见 图 27-3 了解负载条件。

表 27-12：　　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	—	—	TCY	（注 1）
31	TWDT	看门狗定时器超时周期 （无后分频器）	3.4	4.0	4.6	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	45.8	65.5	85.2	ms	
34	TIOZ	自 $\overline{\text{MCLR}}$ 低电平和看门狗定时器复位起 I/O 处于高阻态的时间	—	2	—	µs	
38	TCSD	CPU 启动时间	—	200	—	µs	

注　1：　要确保器件复位， $\overline{\text{MCLR}}$ 必须至少保持 2 TCY 或 400 µs （取较低值）的低电平时间。

表 27-13： TIMER0 和 TIMER1 外部时钟时序



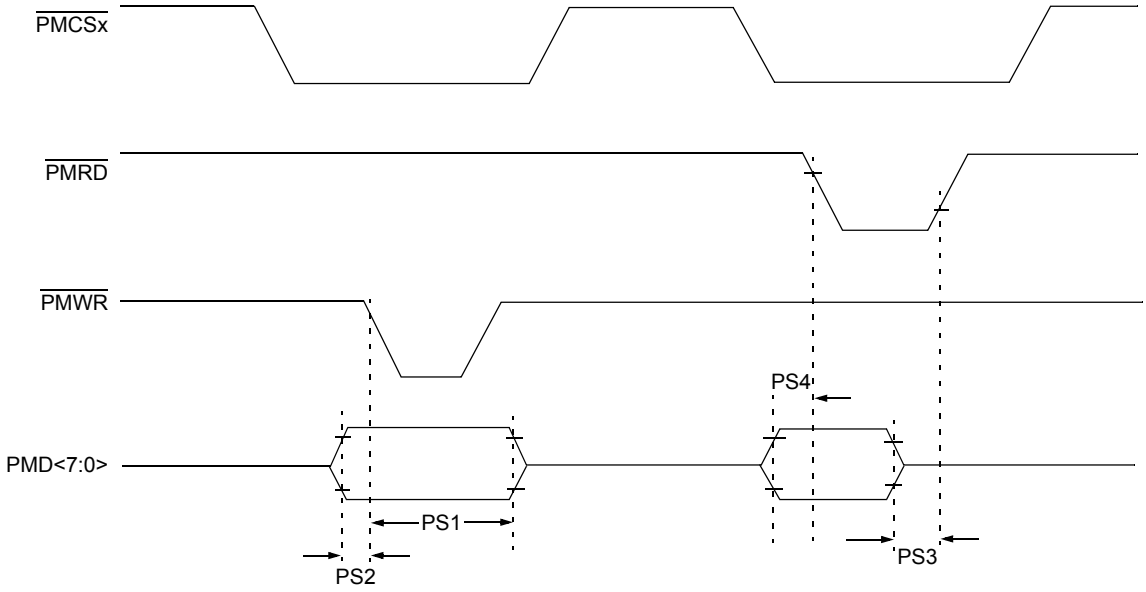
注： 请参见 图 27-3 了解负载条件。

表 27-14： 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、 ... 或 256)
45	TT1H	T13CKI 高电平时间	同步，无预分频器	0.5 TCY + 20	—	ns	
			同步，有预分频器	10	—	ns	
			异步	30	—	ns	
46	TT1L	T13CKI 低电平时间	同步，无预分频器	0.5 TCY + 5	—	ns	
			同步，有预分频器	10	—	ns	
			异步	30	—	ns	
47	TT1P	T13CKI 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 或 8）
			异步	60	—	ns	
	FT1	T13CKI 振荡器输入频率范围		DC	50	kHz	
48	TCKE2TMRI	从外部 T13CKI 时钟沿到定时器递增的延时		2 TOSC	7 TOSC	—	

© 2007 Microchip Technology Inc.

图 27-9：　　并行从端口时序

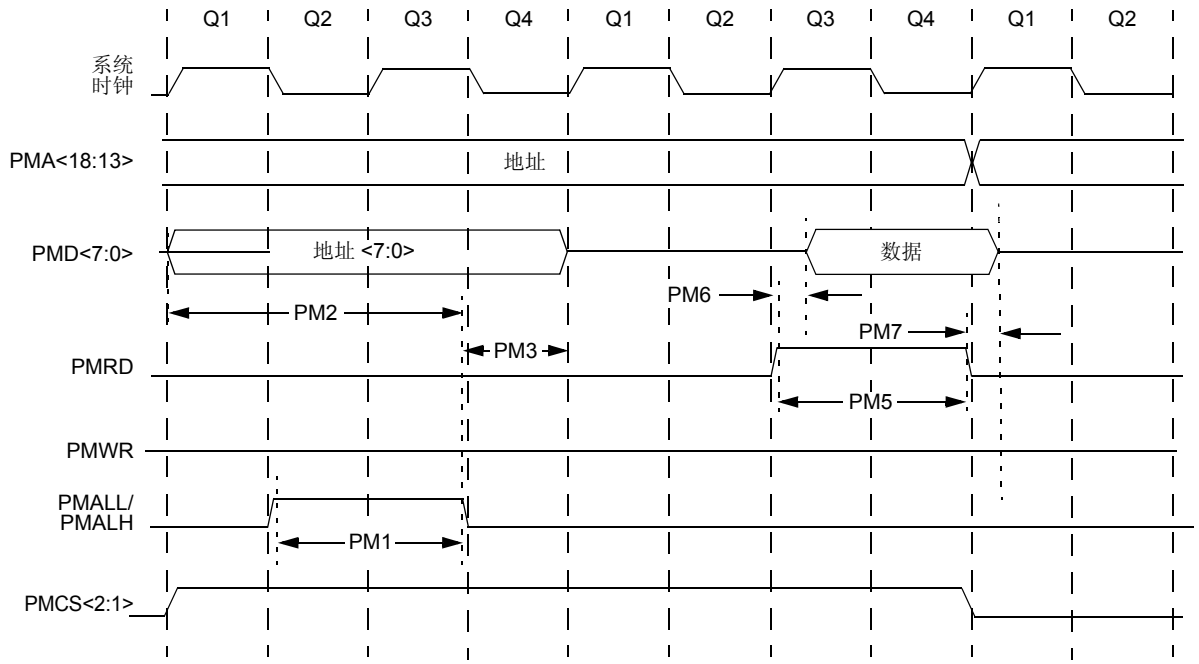


注：　请参见 图 27-3 了解负载条件。

表 27-15：　　并行从端口要求

参数编号	符号	特性	最小值	最大值	单位	条件
PS1	TdtV2wrH	在 $\overline{\text{PMWR}}$ 或 $\overline{\text{PMCSx}}$ 无效之前数据输入有效（建立时间）	20	—	ns	
PS2	TwrH2dtI	$\overline{\text{PMWR}}$ 或 $\overline{\text{PMCSx}}$ 无效至数据输入无效（保持时间）	20	—	ns	
PS3	TrdL2dtV	$\overline{\text{PMRD}}$ 和 $\overline{\text{PMCSx}}$ 有效至数据输出有效	—	80	ns	
PS4	TrdH2dtI	$\overline{\text{PMRD}}$ 有效和 $\overline{\text{PMCSx}}$ 无效至数据输出无效	10	30	ns	

图 27-10：　　并行主端口 读时序图



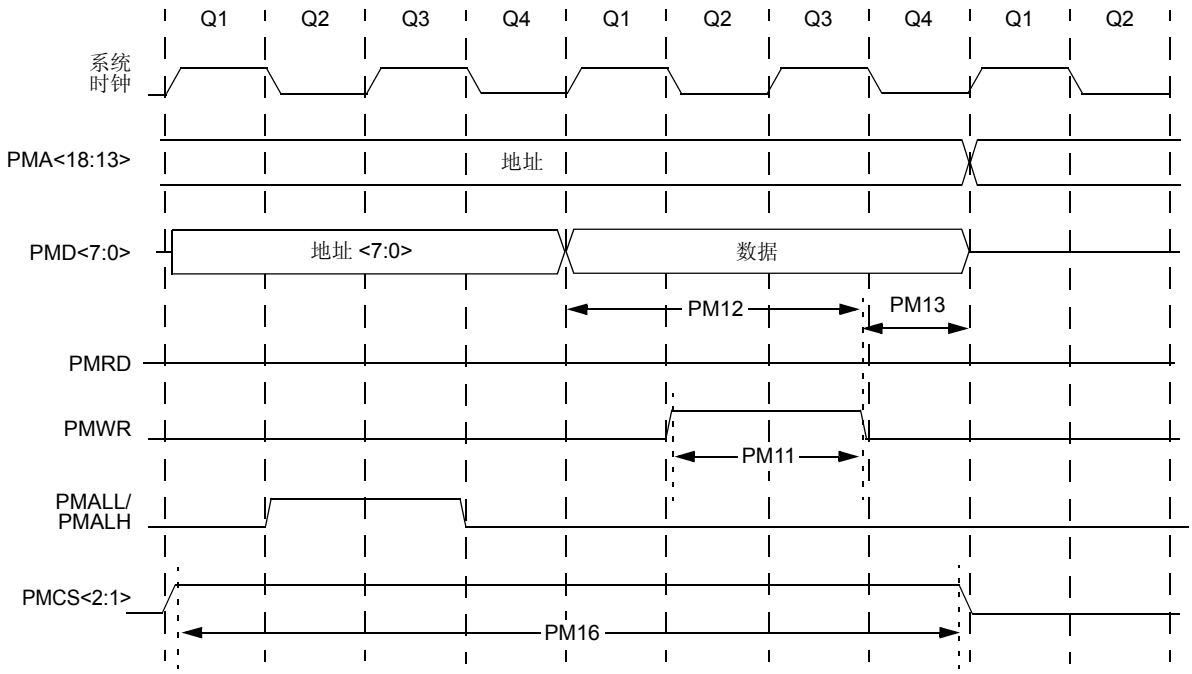
工作条件：除非另外声明，否则均为 2.0V < VCC < 3.6V，-40°C < TA < +85°C。

表 27-16：　　并行主端口读时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位
PM1		PMALL/PMALH 脉冲宽度	—	0.5 TCY	—	ns
PM2		地址输出有效至 PMALL/PMALH 无效（地址建立时间）	—	0.75 TCY	—	ns
PM3		PMALL/PMALH 无效至地址数据无效（地址保持时间）	—	0.25 TCY	—	ns
PM5		PMRD 脉冲宽度	—	0.5 TCY	—	ns
PM6		PMRD 或 PMENB 有效至数据输入有效（数据建立时间）	—	—	—	ns
PM7		PMRD 或 PMENB 无效至数据输入无效（数据保持时间）	—	—	—	ns

© 2007 Microchip Technology Inc.

图 27-11：　　并行主端口 写时序图

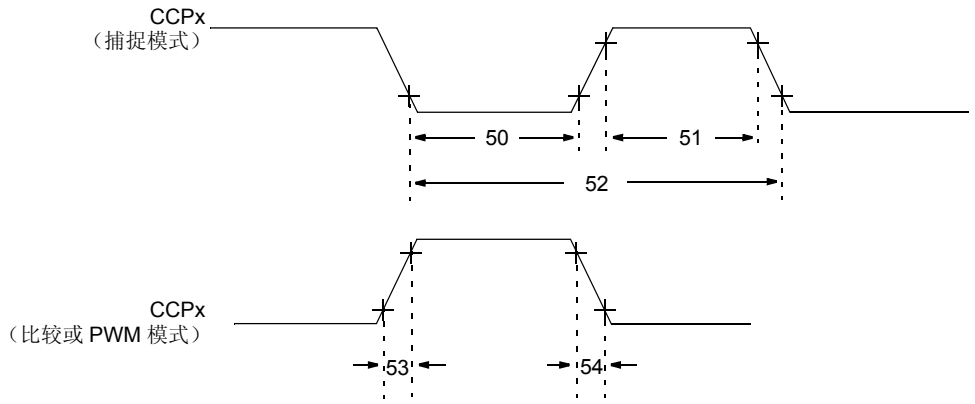


工作条件：除非另外声明，否则均为 2.0V < VCC < 3.6V，-40°C < TA < +85°C。

表 27-17：　　并行主端口写时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位
PM11		PMWR 脉冲宽度	—	0.5 TCY	—	ns
PM12		在 PMWR 或 PMENB 变为无效前数据输出有效 （数据建立时间）	—	—	—	ns
PM13		PMWR 或 PMEMB 无效至数据输出无效（数据保持时间）	—	—	—	ns
PM16		PMCSx 脉冲宽度	TCY – 5	—	—	ns

图 27-12： 捕捉 / 比较 /PWM 时序 （包括 ECCP 模块）



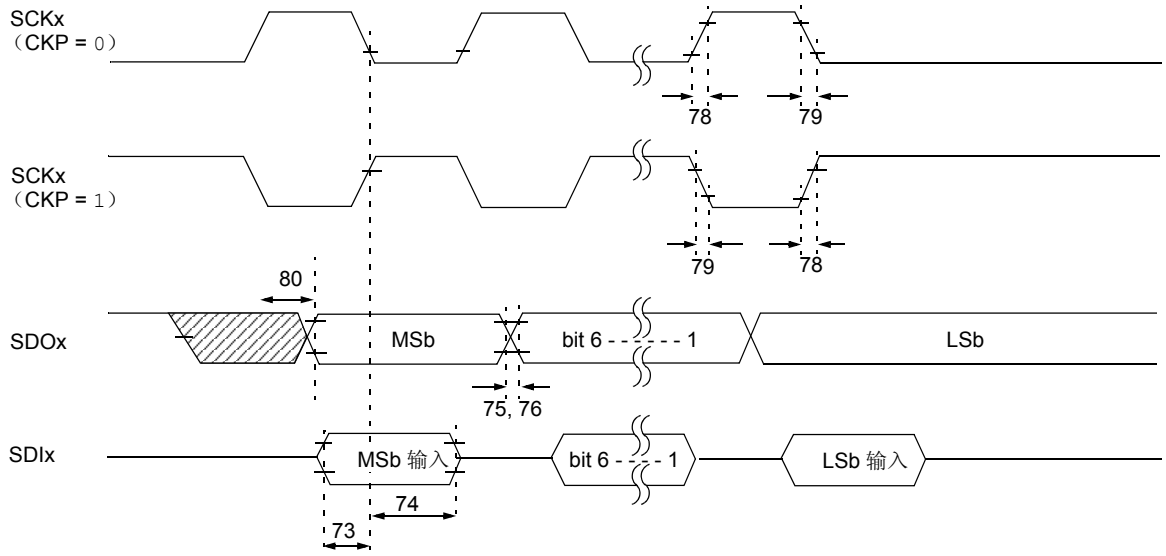
注： 请参见图 27-3 了解负载特性。

表 27-18： 捕捉 / 比较 /PWM 要求 （包括 ECCP 模块）)

参数编号	符号	特性		最小值	最大值	单位	条件
50	TccL	CCPx 输入低电平时间	无预分频器	0.5 TCY + 20	—	ns	
			有预分频器	10	—	ns	
51	TccH	CCPx 输入高电平时间	无预分频器	0.5 TCY + 20	—	ns	
			有预分频器	10	—	ns	
52	TccP	CCPx 输入周期		3 TCY + 40 N	—	ns	N = 预分频值（1、 4 或 16）
53	TccR	CCPx 输出上升时间		—	25	ns	
54	TccF	CCPx 输出下降时间		—	25	ns	

© 2007 Microchip Technology Inc.

图 27-13：　　SPI 主控模式时序示例 （CKE = 0）

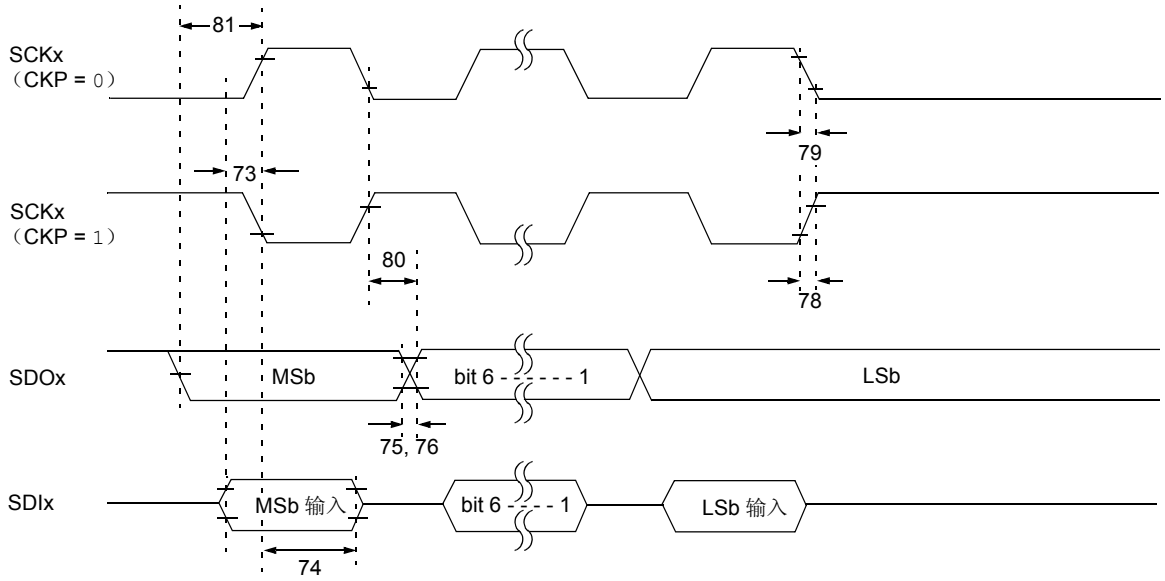


注：　请参见图 27-3 了解负载条件。

表 27-19：　　SPI 模式要求示例 （主控模式， CKE = 0）

参数编号	符号	特性	最小值	最大值	单位	条件
73	TDIV2SCH, TDIV2SCL	SDIx 数据输入至 SCKx 边沿的建立时间	100	—	ns	
73A	TB2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿	1.5 TCY + 40	—	ns	
75	TDOR	SDOx 数据输出上升时间	—	25	ns	
76	TDOF	SDOx 数据输出下降时间	—	25	ns	
78	TSCR	SCKx 输出上升时间	—	25	ns	
79	TSCF	SCKx 输出下降时间	—	25	ns	
80	TSCH2DOV, TSCL2DOV	SCKx 边沿后 SDOx 数据输出有效	—	50	ns	

图 27-14： SPI 主控模式时序示例 （CKE = 1）



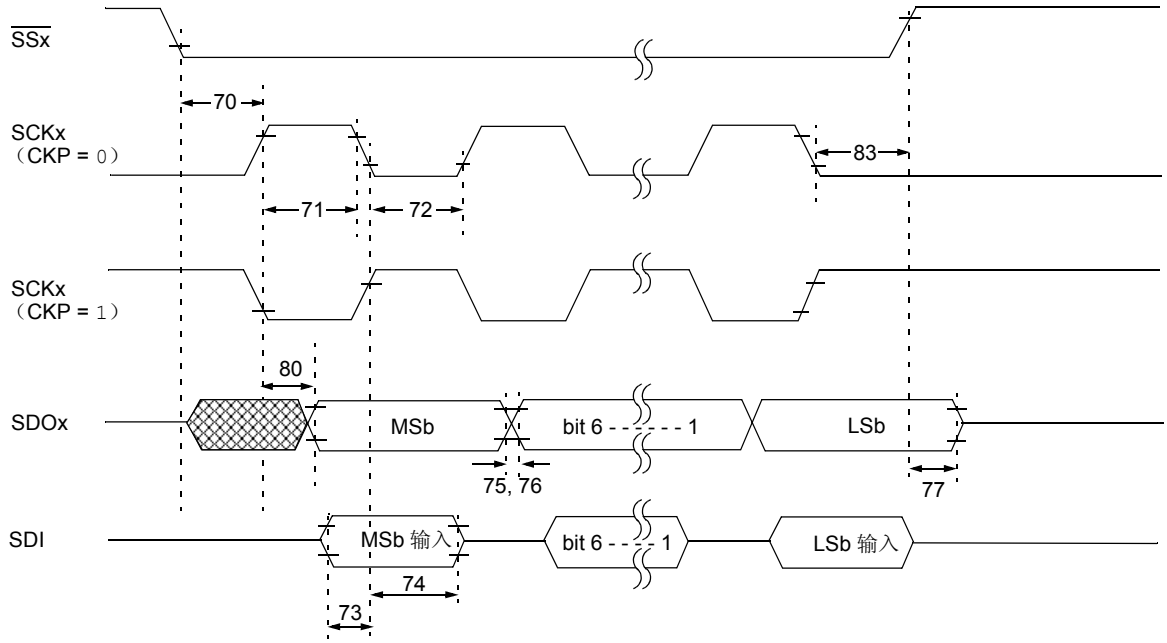
注： 请参见图 27-3 了解负载条件。

表 27-20： SPI 模式要求示例 （主控模式， CKE = 1）

参数编号	符号	特性	最小值	最大值	单位	条件
73	TDIV2SCH, TDIV2SCL	SDIx 数据输入至 SCKx 边沿的建立时间	100	—	ns	
74	TSCH2DIL, TSCL2DIL	SDIx 数据输入至 SCKx 边沿的保持时间	100	—	ns	
75	TDOR	SDOx 数据输出上升时间	—	25	ns	
76	TDOF	SDOx 数据输出下降时间	—	25	ns	
78	TSCR	SCKx 输出上升时间	—	25	ns	
79	TSCF	SCKx 输出下降时间	—	25	ns	
80	TSCH2DOV, TSCL2DOV	SCKx 边沿后 SDOx 数据输出有效	—	50	ns	
81	TDOV2SCH, TDOV2SCL	SDOx 数据输出建立至 SCKx 边沿	TCY	—	ns	

© 2007 Microchip Technology Inc.

图 27-15：　　SPI 从动模式时序示例 （CKE = 0）



注：　请参见 图 27-3 了解负载特性。

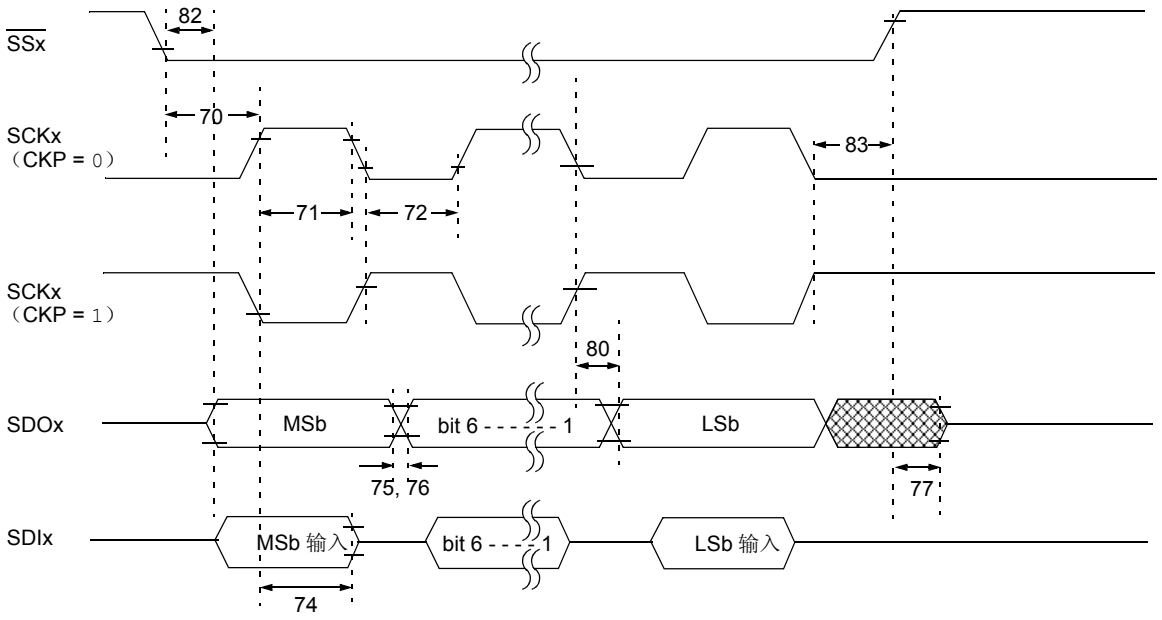
表 27-21：　　SPI 模式要求示例 （从动模式时序，CKE = 0）

参数编号	符号	特性		最小值	最大值	单位	条件
70	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ 至 SCKx ↓ 或 SCKx ↑ 输入		3 TCY	—	ns	
70A	TssL2WB	$\overline{SSx}$ ↓ 至写入 SSPxBUF		3 TCY	—	ns	
71	TscH	SCKx 输入高电平时间	连续	1.25 TCY + 30	—	ns	
71A			单字节	40	—	ns	（注 1）
72	TscL	SCKx 输入低电平时间	连续	1.25 TCY + 30	—	ns	
72A			单字节	40	—	ns	（注 1）
73	TdiV2scH, TdiV2scL	SDIx 数据输入至 SCKx 边沿的建立时间		100	—	ns	
73A	Tb2b	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿		1.5 TCY + 40	—	ns	（注 2）
74	TscH2diL, TscL2diL	SDIx 数据输入至 SCKx 边沿的保持时间		100	—	ns	
75	TdoR	SDOx 数据输出上升时间		—	25	ns	
76	TdoF	SDOx 数据输出下降时间		—	25	ns	
77	TssH2doZ	$\overline{SSx}$ ↑ 至 SDOx 输出高阻态		10	50	ns	
80	TscH2doV, TscL2doV	SCKx 边沿后 SDOx 数据输出有效		—	50	ns	
83	TscH2ssH, TscL2ssH	SCKx 边沿后至 $\overline{SSx}$ ↑		1.5 TCY + 40	—	ns	

注　1： 要求使用参数 #73A。

2： 仅当使用参数 #71A 和 #72A 时。

图 27-16：　　SPI 从动模式时序示例 （CKE = 1）



注：　请参见 图 27-3 了解负载特性。

表 27-22：　　SPI 从动模式要求示例 （CKE = 1）

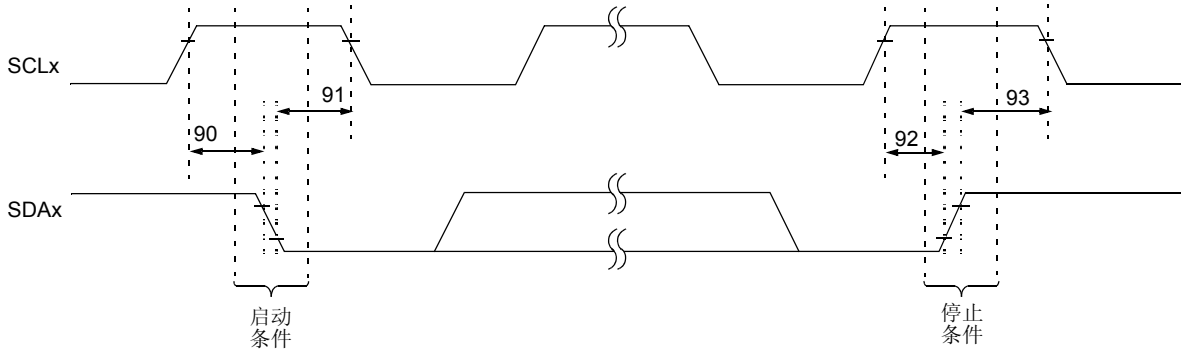
参数编号	符号	特性		最小值	最大值	单位	条件
70	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ 至 SCKx ↓ 或 SCKx ↑ 输入		3 TCY	—	ns	
70A	TssL2WB	$\overline{SSx}$ ↓ 至写入 SSPxBUF		3 TCY	—	ns	
71	TscH	SCKx 输入高电平时间	连续	1.25 TCY + 30	—	ns	
71A			单字节	40	—	ns	（注 1）
72	TscL	SCKx 输入低电平时间	连续	1.25 TCY + 30	—	ns	
72A			单字节	40	—	ns	（注 1）
73A	TB2B	字节 1 的最后一个时钟边沿至字节 2 的第一个时钟边沿		1.5 TCY + 40	—	ns	（注 2）
74	TscH2diL, TscL2diL	SDIx 数据输入至 SCKx 边沿的保持时间		100	—	ns	
75	TdoR	SDOx 数据输出上升时间		—	25	ns	
76	TdoF	SDO 数据输出下降时间		—	25	ns	
77	TssH2doZ	$\overline{SSx}$ ↑ 至 SDOx 输出高阻态		10	50	ns	
80	TscH2doV, TscL2doV	SCKx 边沿后 SDOx 数据输出有效		—	50	ns	
82	TssL2doV	$\overline{SSx}$ ↓ 边沿后 SDOx 数据输出有效		—	50	ns	
83	TscH2ssH, TscL2ssH	SCKx 边沿后至 $\overline{SSx}$ ↑		1.5 TCY + 40	—	ns	

注　1： 要求使用参数 #73A。

2： 仅当使用参数 #71A 和 #72A 时。

© 2007 Microchip Technology Inc.

图 27-17：　　I²C™ 总线 启动 / 停止位时序

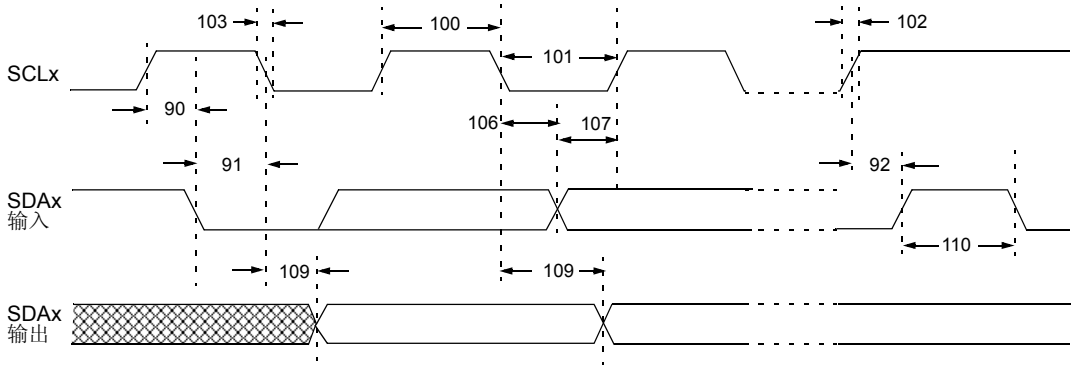


注： 请参见 图 27-3 了解负载特性。

表 27-23：　　I²C™ 总线启动 / 停止位要求 （从动模式）

参数编号	符号	特性		最小值	最大值	单位	条件
90	TSU:STA	启动条件建立时间	100 kHz mode	4700	—	ns	仅与重复启动条件相关
			400 kHz mode	600	—		
91	THD:STA	启动条件保持时间	100 kHz mode	4000	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz mode	600	—		
92	TSU:STO	停止条件建立时间	100 kHz mode	4700	—	ns	
			400 kHz mode	600	—		
93	THD:STO	停止条件保持时间	100 kHz mode	4000	—	ns	
			400 kHz mode	600	—		

图 27-18：　　I²C™ 总线数据时序



注： 请参见图 27-3 了解负载特性。

表 27-24： I²C™ 总线数据要求 （从动模式）

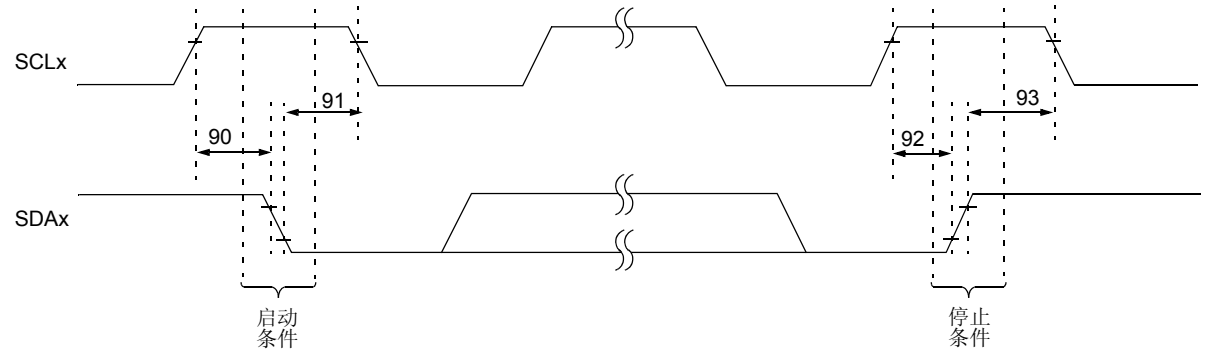
参数编号	符号	特性		最小值	最大值	单位	条件
100	THIGH	时钟高电平时间	100 kHz 模式	4.0	—	μs	
			400 kHz 模式	0.6	—	μs	
			MSSP 模块	1.5 TCY	—		
101	TLOW	时钟低电平时间	100 kHz 模式	4.7	—	μs	
			400 kHz 模式	1.3	—	μs	
			MSSP 模块	1.5 TCY	—		
102	TR	SDAx 和 SCLx 下降时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	20 + 0.1 CB	300	ns	CB 值在 10 到 400 pF 之间
103	TF	SDAx 和 SCLx 上升时间	100 kHz 模式	—	300	ns	
			400 kHz 模式	20 + 0.1 CB	300	ns	CB 值在 10 到 400 pF 之间
90	TSU:STA	启动条件建立时间	100 kHz 模式	4.7	—	μs	仅与重复启动条件相关
			400 kHz 模式	0.6	—	μs	
91	THD:STA	启动条件保持时间	100 kHz 模式	4.0	—	μs	这个周期后产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs	
106	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
107	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	**（注 2）**
			400 kHz 模式	100	—	ns	
92	TSU:STO	停止条件建立时间	100 kHz 模式	4.7	—	μs	
			400 kHz 模式	0.6	—	μs	
109	TAA	时钟输出有效时间	100 kHz 模式	—	3500	ns	**（注 1）**
			400 kHz 模式	—	—	ns	
110	TBUF	总线空闲时间	100 kHz 模式	4.7	—	μs	在启动一个新的传输前总线保持空闲的时间
			400 kHz 模式	1.3	—	μs	
D102	CB	总线的容性负载		—	400	pF	

注 1： 为避免意外产生启动或停止条件，作为发送器的器件必须提供此内部最小延时时间，以覆盖 SCLx 下降沿的未定义区域（最小值 300 ns）

2： 在标准模式的 I²C 总线系统中可以使用快速模式的 I²C™ 总线器件，但必须满足 TSU:DAT ≥ 250 ns 的要求。如果快速模式器件没有延长 SCLx 信号的低电平时间，将自动满足此条件。如果该器件延长了 SCLx 信号的低电平时间，它必须将下一个数据位输出到 SDAx 线， SCLx 线被释放前， TR max. + TSU:DAT = 1000 + 250 = 1250 ns （根据标准模式 I²C 规范）。

© 2007 Microchip Technology Inc.

图 27-19： MSSP I²C™ 总线启动 / 停止位时序波形



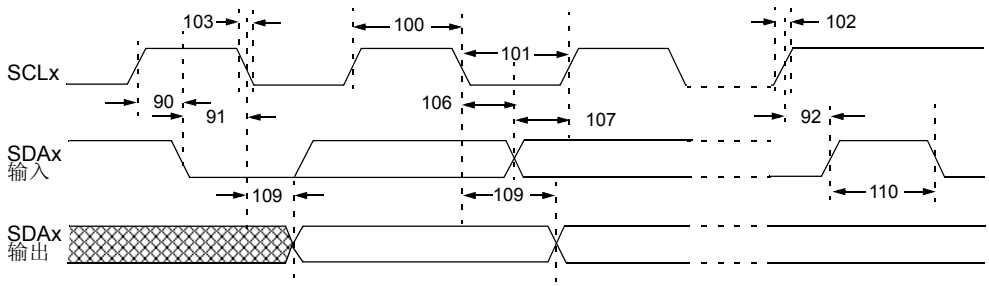
注： 请参见 图 27-3 了解负载条件。

表 27-25： MSSP I²C™ 总线启动 / 停止位要求

参数编号	符号	特性		最小值	最大值	单位	条件
90	TSU:STA	启动条件建立时间	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	仅与重复启动条件相关
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
91	THD:STA	启动条件保持时间	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
92	TSU:STO	停止条件建立时间	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
93	THD:STO	停止条件保持时间	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		

注 1： 对于所有 I²C™ 引脚，最小引脚电容为 10 pF。

图 27-20： MSSP I²C™ 总线数据时序



注： 请参见 图 27-3 了解负载条件。

表 27-26： MSSP I²C™ 总线数据要求

参数编号	符号	特性		最小值	最大值	单位	条件
100	THIGH	时钟高电平时间	100 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			400 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			1 MHz 模式 (1)	2(TOSC)(BRG + 1)	—	ms	
101	TLOW	时钟低电平时间	100 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			400 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			1 MHz 模式 (1)	2(TOSC)(BRG + 1)	—	ms	
102	TR	SDAx 和 SCLx 上升时间	100 kHz 模式	—	1000	ns	CB 值在 10 到 400 pF 之间
			400 kHz 模式	20 + 0.1 CB	300	ns	
			1 MHz 模式 (1)	—	300	ns	
103	TF	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns	CB 值在 10 到 400 pF 之间
			400 kHz 模式	20 + 0.1 CB	300	ns	
			1 MHz 模式 (1)	—	100	ns	
90	TSU:STA	启动条件建立时间	100 kHz 模式	2(TOSC)(BRG + 1)	—	ms	仅与重复启动条件相关
			400 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			1 MHz 模式 (1)	2(TOSC)(BRG + 1)	—	ms	
91	THD:STA	启动条件保持时间	100 kHz 模式	2(TOSC)(BRG + 1)	—	ms	这个周期后产生第一个时钟脉冲
			400 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			1 MHz 模式 (1)	2(TOSC)(BRG + 1)	—	ms	
106	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	ms	
			1 MHz 模式 (1)	TBD	—	ns	
107	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	（注 2）
			400 kHz 模式	100	—	ns	
			1 MHz 模式 (1)	TBD	—	ns	
92	TSU:STO	停止条件建立时间	100 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			400 kHz 模式	2(TOSC)(BRG + 1)	—	ms	
			1 MHz 模式 (1)	2(TOSC)(BRG + 1)	—	ms	
109	TAA	时钟输出有效时间	100 kHz 模式	—	3500	ns	
			400 kHz 模式	—	1000	ns	
			1 MHz 模式 (1)	—	—	ns	
110	TBUF	总线空闲时间	100 kHz 模式	4.7	—	ms	在启动一个新的传输前总线保持空闲的时间
			400 kHz 模式	1.3	—	ms	
			1 MHz 模式 (1)	TBD	—	ms	
D102	CB	总线的容性负载		—	400	pF	

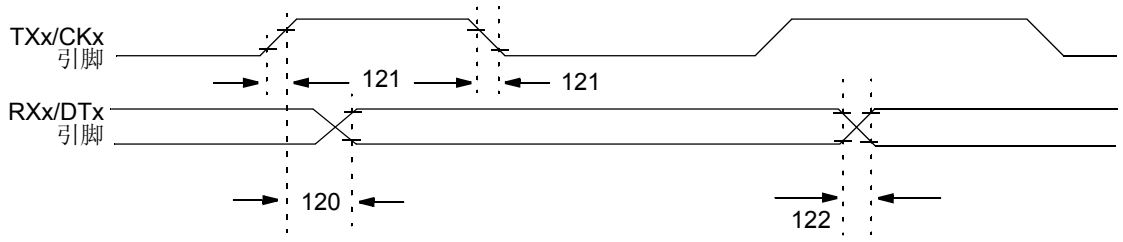
图注： TBD = 待定

注 1： 对于所有 I²C™ 引脚的最大引脚电容为 10 pF。

2： 在标准模式的 I²C 总线系统中，可以使用快速模式的 I²C 总线器件，但必须满足参数 #107 ≥ 250 ns 的要求。如果快速模式器件没有延长 SCLx 信号的低电平周期，则将自动满足此条件。如果该器件延长了 SCLx 信号的低电平周期，它必须将下一个数据位输出到 SDAx 线，在 SCLx 线被释放前，参数 #102 + 参数 #107 = 1000 + 250 = 1250 ns （100 kHz 模式下）。

© 2007 Microchip Technology Inc.

图 27-21： EUSART 同步 发送 （主控 / 从动）时序

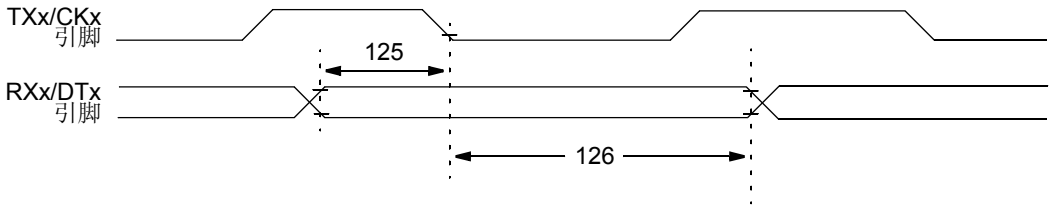


注： 请参见图 27-3 了解负载条件。

表 27-27： 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） 从时钟高电平至数据输出有效	—	40	ns	
121	TCKRF	时钟输出信号的上升时间和下降时间 （主控模式）	—	20	ns	
122	TDTRF	数据输出信号的上升时间和下降时间	—	20	ns	

图 27-22： EUSART 同步接收 （主控 / 从动）时序



注： 请参见图 27-3 了解负载条件。

表 27-28： 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 CKx↓ 之前数据保持时间 （DTx 保持时间）	10	—	ns	
126	TCKL2DTL	在 CKx↓ 之后数据保持时间 （DTx 保持时间）	15	—	ns	

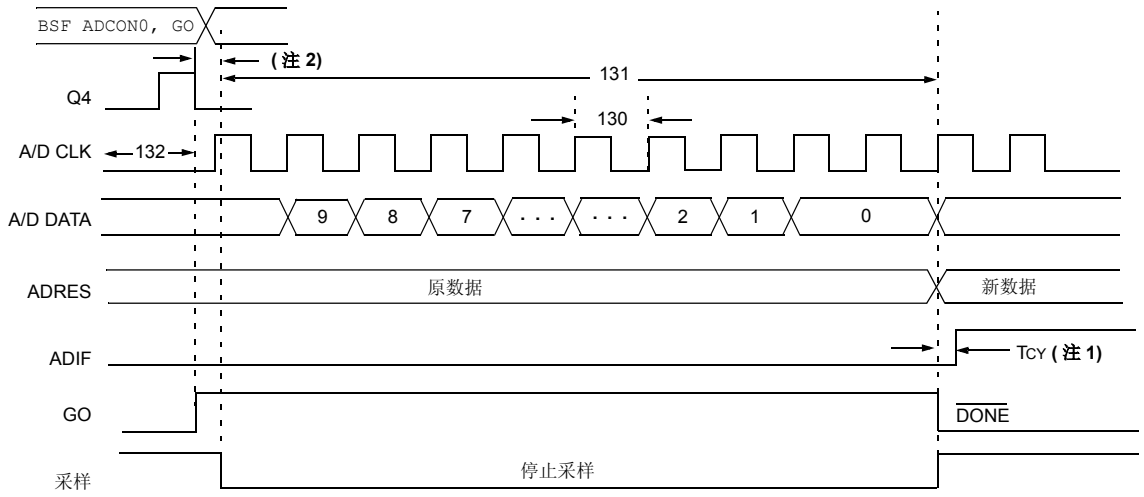
表 27-29： A/D 转换器规范： PIC18F87J11 系列 （工业级）

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	—	—	<±3	LSb	ΔVREF ≥ 3.0V
A07	EGN	增益误差	—	—	<±3	LSb	ΔVREF ≥ 3.0V
A10	—	单调性	保证 (1)			—	VSS ≤ VAIN ≤ VREF
A20	ΔVREF	参考电压范围 （VREFH – VREFL）	2.0 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	模拟信号源阻抗建议值	—	—	2.5	kΩ	
A50	IREF	VREF 输入电流 (2)	— —	— —	5 150	μA μA	在采集 VAIN 期间 在 A/D 转换期间

注 1： A/D 转换结果不会因输入电压的增加而减小，且不会丢失编码。

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。

图 27-23： A/D 转换时序



注 1： 如果选择 RC 作为 A/D 时钟源，在 A/D 时钟启动前要加上一个 TCY 时间。用于执行 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断开。

© 2007 Microchip Technology Inc.

表 27-30： 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	0.7	25.0(1)	μs	基于 TOSC， VREF ≥ 3.0V
			TBD	1	μs	A/D RC 模式
131	TCNV	转换时间（不包括采集时间）（**注 2**）	11	12	TAD	
132	TACQ	采集时间 （**注 3**）	1.4	—	μs	-40°C 至 +85°C
135	TSWC	转换 → 采样的切换时间	—	（**注 4**）		
TBD	TDIS	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），保持电容采集一个 “新” 输入电压所需的时间。在输入通道上的源阻抗 （RS）为 50Ω。

4： 在器件时钟的下一个周期上。

注：

 © 2007 Microchip Technology Inc.

28.0 DC 和 AC 特性图表

目前尚未提供特性图表。

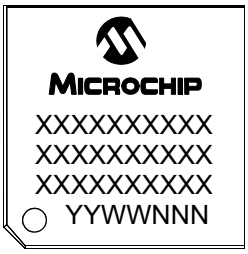
注：

 © 2007 Microchip Technology Inc.

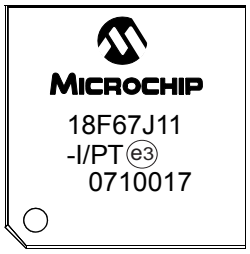
29.0 封装信息

29.1 封装标识信息

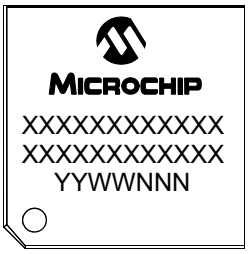
64 引脚 TQFP



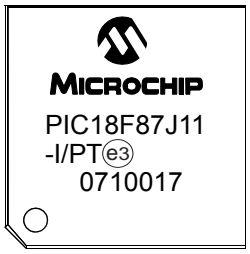
示例



80 引脚 TQFP



示例



图注：	XX...X	客户信息
	Y	年份代码 （公历年的最后一位数字）
	YY	年份代码 （公历年的最后两位数字）
	WW	星期代码 (1 月 1 日的星期代码为 01)
	NNN	按字母数字排序的追踪代码
	(e3)	无铅 JEDEC 雾锡 （Sn）标识符
	*	该封装不含铅。无铅 JEDEC 标识符 （(e3)） 能在此类封装的外封装上找到。

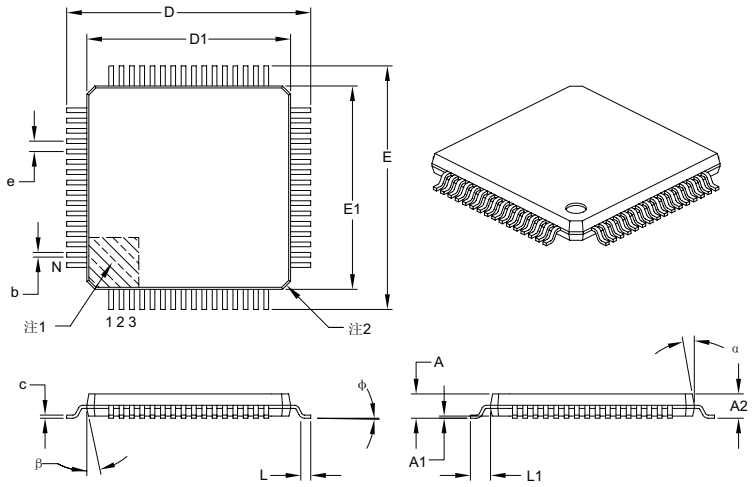
注： 若 Microchip 器件编号未在一行中完全标出，它将换行继续标出，因此会限制客户信息的可用字符数。

29.2 封装详细信息

以下部分将介绍各种封装的技术细节。

64 引脚塑封薄型正方形封装 （PT）——10x10x1 mm 主体， 2.00 mm 占位 [TQFP]

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



	单位	毫米		
	尺寸范围	最小	正常	最大
引脚数	N		64	
引脚间距	e		0.50 BSC	
总高度	A	-	-	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	-	0.15
底足长度	L	0.45	0.60	0.75
底足占位	L1		1.00 REF	
底足倾角	φ	0°	3.5°	7°
总宽度	E		12.00 BSC	
总长度	D		12.00 BSC	
塑模封装宽度	E1		10.00 BSC	
塑模封装长度	D1		10.00 BSC	
引脚厚度	c	0.09	-	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶部椎度	α	11°	12°	13°
塑模底部椎度	β	11°	12°	13°

注：

1. 引脚1的外观定位特性可能变化，但必须处于阴影区内。
2. 四角处的倒角是可选的；尺寸可能变化。
3. 尺寸D1和E1不包括塑模的毛边或突起。塑模的毛边或突起每侧不得超过0.25毫米。
4. 尺寸及公差依照ASME Y14.5M。

BSC： 基本尺寸。所示为理论精确值，无公差。

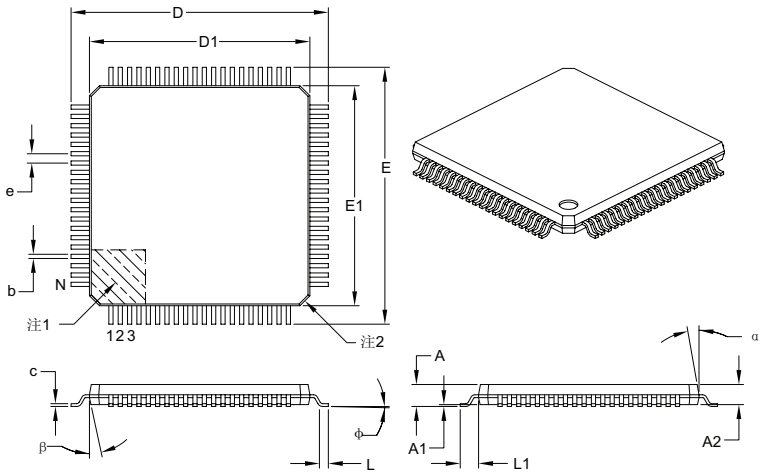
REF： 参考尺寸，通常无公差，仅供参考。

Microchip 图号：C04-085B

© 2007 Microchip Technology Inc.

80 引脚塑封薄型正方形封装 （PT）——12x12x1 mm 主体， 2.00 mm 占位 [TQFP]

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



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	80		
引脚间距	e	0.50 BSC		
总高度	A	-	-	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	-	0.15
底足长度	L	0.45	0.60	0.75
底足占位	L1	1.00 REF		
底足倾角	ϕ	0°	3.5°	7°
总宽度	E	14.00 BSC		
总长度	D	14.00 BSC		
塑模封装宽度	E1	12.00 BSC		
塑模封装长度	D1	12.00 BSC		
引脚厚度	c	0.09	-	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶剖椎度	α	11°	12°	13°
塑模底部椎度	β	11°	12°	13°

注：

1. 引脚1的外观定位特性可能变化，但必须位于阴影区域内。
2. 四角的倒角是可选的；尺寸可能变化。
3. 尺寸D1和E1不包含塑模的毛边或突起。塑模的毛边或突起每侧不得超过0.25毫米。
4. 尺寸和公差依照ASME Y14.5M。

BSC： 基本尺寸。所示为理论精确值，无公差。
REF： 参考尺寸，通常无公差，仅供参考。

Microchip 图号：C04-092B

注：

 © 2007 Microchip Technology Inc.

附录 A： 版本历史

版本 A （2007 年 1 月）

PIC18F87J11 系列器件数据手册的最初版本。

版本 B （2007 年 2 月）

更新了“DC特性”节的掉电电流和供电电流表中的值。

附录 B： 器件差异

表 B-1 为本数据手册中所列器件间的差异。

表 B-1： PIC18F87J11 系列各产品间的差异

功能特性	PIC18F66J11	PIC18F66J16	PIC18F67J11	PIC18F86J11	PIC18F86J16	PIC18F87J11
程序存储器	64K	96K	128K	64K	96K	128K
程序存储器 （指令）	32764	49148	65532	32764	49148	65532
I/O 端口	端口 A、 B、 C、 D、 E、 F 和 G			端口 A、 B、 C、 D、 E、 F、 G、 H 和 J		
EMB	无			有		
10 位 ADC 模块	11 路输入通道			15 路输入通道		
封装	64 引脚 TQFP			80 引脚 TQFP		

注：

 © 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、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

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

欲注册，请登录 Microchip 网站 www.microchip.com，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: PIC18F87J11 系列　　文献编号: DS39778B_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.

索引

A

A/D 291
  A/D 转换器中断，配置 295
  ADCAL 位 299
  ADRESH 寄存器 294
  采集要求 296
  ECCP2 触发器的使用 298
  校准 299
  配置模拟端口引脚 297
  配置模块 295
  特殊事件触发器 （ECCP） 210, 298
  相关的寄存器 300
  在功耗管理模式下的工作方式 299
  自动采集时间 297
  转换 298
  转换器规范 421
  转换时钟 （TAD） 297
  转换要求 422
  转换状态 （GO/DONE 位） 294
AC （时序）规范 398
  参数符号 398
  器件时序规范的负载条件 399
  时序条件 399
  温度和电压规范 399
ACKSTAT 259
ACKSTAT 状态标志位 259
ADCAL 位 299
ADCON0 寄存器
  GO/DONE 位 294
ADDFSR 372
ADDLW 335
ADDULNK 372
ADDWF 335
ADDWFC 336
ADRESL 寄存器 294
ANDLW 336
ANDWF 337

B

版本历史 431
保护配置寄存器 327
BC 337
BCF 338
BF 259
BF 状态标志位 259
比较 （CCP 模块） 202
  CCPRx 寄存器 202
  软件中断 202
  相关的寄存器 203
  引脚配置 202
  Timer1/Timer3 模式选择 202
比较 （ECCP 模块） 210
  特殊事件触发信号 298
  特殊事件触发器 210
比较 （ECCPx 模块）
  特殊事件触发信号 195
比较器 301
  参考
    响应时间 304
  单个比较器 304
  复位的影响 308
  工作原理 304
  控制 305
  模拟输入连接注意事项 304
  配置 305
  使能和输出选择 305
  使能和输入选择 305
  相关的寄存器 308
  休眠期间的工作 308
  中断 307
比较器参考电压模块 309
  复位的影响 311
  精度和误差 311
  连接注意事项 311
  配置 310
  相关的寄存器 311
  休眠模式下的工作 311
比较器规范 397
编程，器件指令 329
变更通知客户服务 433
表读 / 表写 67
表指针操作 （表） 90
并行主端口 （PMP） 157
  从端口模式 165
  控制寄存器 158
  数据寄存器 164
  相关的寄存器 180
  应用示例 178
  主端口模式 170
BN 338
BNC 339
BNN 339
BNOV 340
BNZ 340
BOV 343
波特率发生器 255
BRA 341
BRG。*请参见*波特率发生器。
BSF 341
BTFSC 342
BTFSS 342
BTG 343
捕捉 / 比较 /PWM （CCP） 199
  CCP 模式和定时器资源 200
  CCPRxH 寄存器 200
  CCPRxL 寄存器 200
  定时器互连配置 200
  模块配置 200
捕捉 / 比较 /PWM （CCP）
  比较模式。*请参见*比较。
  捕捉模式。*请参见*捕捉。
捕捉 （CCP 模块） 201
  CCP 引脚配置 201
  CCPRxH:CCPRxL 寄存器 201
  相关的寄存器 203
  软件中断 201
  预分频器 201
  Timer1/Timer3 模式选择 201
捕捉 （ECCP 模块） 210
BZ 344

C

C 编译器
  MPLAB C18 380
  MPLAB C30 380
参考电压规范 397
参考时钟输出 38

CALL 344
CALLW 373
程序校验和代码保护 327
程序存储器
ALU
状态 80
查找表 67
存储器硬编码向量 62
存储器映射图 61
模式 64
硬编码向量和配置字单元 62
复位向量 62
扩展指令集 84
模式 63
存储器访问 （表） 64
单片机 63
扩展单片机 63
扩展单片机 （地址移位） 64
闪存配置字 62
指令 69
双字 69
中断向量 62
程序计数器 65
PCL、 PCH 和 PCU 寄存器 65
PCLATH 和 PCLATU 寄存器 65
CLRF 345
CLRWDT 345
从动选择 （SSx） 223
COMF 346
CPFSEQ 346
CPFSGT 347
CPFSLT 347
CPU 的特殊性能 313

D

DAW 348
代码保护 313
代码示例
A/D 校准程序 299
擦除闪存程序存储器行 92
初始化 PORTA 130
初始化 PORTB 133
初始化 PORTC 136
初始化 PORTD 139
初始化 PORTE 142
初始化 PORTF 145
初始化 PORTG 148
初始化 PORTH 151
初始化 PORTJ 154
单字写闪存程序存储器 95
读取一个闪存程序存储器内字 91
改变捕捉预分频比值 201
将 STATUS、WREG 和 BSR 寄存器的值保存在 RAM 中 126
快速寄存器堆栈 67
使用偏移量计算 GOTO 67
使用间接寻址将 RAM （Bank 1）清零的方法 81
使用 Timer1 中断服务实现实时时钟 189
写闪存程序存储器 94
装载 SSP1BUF （SSP1SR）寄存器 226
16 × 16 有符号乘法程序 110
16 × 16 无符号乘法程序 110
8 × 8 有符号乘法程序 109
8 × 8 无符号乘法程序 109
DC 和 AC 特性图表 425
DCFSNZ 349
DECF 348
DECFSZ 349
串行时钟， SCKx 223
串行数据输出 （SDOx） 223
串行数据输入 （SDIx） 223
串行外设接口。*请参见* SPI 模式。
存储器构成 61
程序存储器 61
数据存储器 70
电气规范 383
读者反馈表 434
对标准 PIC 指令的影响 376
堆栈满 / 下溢复位 67

E

ECCP
标准 PWM 模式 210
捕捉和比较模式 210
相关的寄存器 222
增强型 PWM 模式 211
ENVREG 引脚 323
EUSART
波特率发生器
在功耗管理模式下工作 273
波特率发生器 （BRG） 273
波特率，异步模式 275
波特率误差，计算 274
采样 273
高波特率选择位 （BRGH 位） 273
相关的寄存器 274
自动波特率检测 277
异步模式 279
发送器 279
接收器 281
设置带有地址检测功能的 9 位模式 281
12 位间隔传送与接收 284
同步间隔字符自动唤醒 282
相关寄存器，发送 280
相关寄存器，接收 282
同步从动模式 288
发送 288
接收 289
相关寄存器，发送 288
相关寄存器，接收 289
同步主控模式 285
发送 285
接收 287
相关寄存器，发送 286
相关寄存器，接收 287

F

封装 427
标识 427
详细信息 428
FSCM。*请参见*故障保护时钟监视器。
复位 49, 313
堆栈满复位 49
堆栈下溢复位 49
功耗管理模式下的 MCLR 复位 49
看门狗定时器 （WDT）复位 49
配置不匹配 （CM） 49
RESET 指令 49
欠压复位 （BOR） 49, 313
上电复位 （POR） 49, 313
上电延时定时器 （PWRT） 313
振荡器起振定时器 （OST） 313

 © 2007 Microchip Technology Inc.

正常工作状态下的 MCLR 复位 49

G

各种时钟源的功耗管理模式的影响 40
GOTO 350
功耗管理模式 41
多条休眠命令 42
和 EUSART 的工作原理 273
和 SPI 模式的工作原理 231
进入 41
空闲模式 45
PRI_IDLE 46
RC_IDLE 47
SEC_IDLE 46
时钟切换和状态指示 42
时钟源 41
退出空闲和休眠模式 47
通过复位退出 47
通过中断退出 47
通过 WDT 超时退出 47
休眠模式 45
休眠模式
OSC1 和 OSC2 引脚状态 40
选择 41
运行模式 42
PRI_RUN 42
RC_RUN 44
SEC_RUN 42
总结 （表） 41
公式
A/D 采集时间 296
A/D 最小充电时间 296
计算所需的最小采集时间 296
故障保护时钟监视器 313, 325
功耗管理模式下的中断 326
上电复位或从休眠中唤醒 326
退出 326
振荡器故障期间的 WDT 325
固件指令 329

H

汇编器
MPASM 汇编器 380

I

I/O 端口 127
漏极开路输出 128
输入上拉配置 128
引脚功能 127
I/O 引脚排列说明
PIC18F6XJ1X （64-TQFP） 12
PIC18F8XJ1X （80-TQFP） 20
I²C 模式 （MSSP）
波特率发生器 255
串行时钟 （RC3/SCKx/SCLx） 241
从动模式 238
地址屏蔽模式
5 位 239
7 位 240
发送 241
接收 241
寻址 238
带有 BRG 的 I²C 时钟速率 255
读 / 写位信息 （R/W 位） 238, 241
多主机模式 263
多主机通信、总线冲突与总线仲裁 263
复位的影响 263
工作原理 238
寄存器 233
休眠工作方式 263
时钟延长 248
10 位从动发送模式 248
10 位从动接收模式 （SEN = 1） 248
7 位从动发送模式 248
7 位从动接收模式 （SEN = 1） 248
时钟仲裁 256
时钟同步和 CKP 位 249
停止条件时序 262
相关的寄存器 268
应答序列时序 262
支持广播呼叫地址 252
主控模式 253
发送 259
工作原理 254
接收 259
启动条件时序 257
重复启动条件时序 258
总线冲突
停止条件期间 267
在重复启动条件期间 266
INCF 350
INCFSZ 351
INFSNZ 351
INTCON 寄存器 113
RBIF 位 133
INTOSC， INTRC。*请参见*内部振荡器电路。
IORLW 352
IORWF 352
IPR 寄存器 122

J

计算 GOTO 67
寄存器 72
ADCON0 （A/D 控制 1） 292
ADCON0 （A/D 控制寄存器 0） 291
ANCON0 （A/D 端口配置寄存器 2） 293
ANCON1 （A/D 端口配置寄存器 1） 293
BAUDCONx （波特率控制寄存器） 272
CCPxCON （捕捉 / 比较 /PWM 控制） 199
CCPxCON （ECCPx 控制） 207
CMSTAT （比较器输出状态寄存器） 303
CMxCON （比较器 x 控制） 302
CONFIG1H （配置寄存器 1 的高字节） 315
CONFIG1L （配置寄存器 1 的低字节） 315
CONFIG2H （配置寄存器 2 的高字节） 317
CONFIG3H （配置寄存器 3 的高字节） 319
CONFIG3L （配置寄存器 3 的低字节） 318
CONFIG3L （配置寄存器 3 低字节） 63
CVRCON （比较器参考电压控制寄存器） 310
DEVID1 （器件 ID 寄存器 1） 320
DEVID2 （器件 ID 寄存器 2） 320
ECCPxAS （ECCPx 自动关闭控制） 219
ECCPxDEL （ECCPx PWM 延时） 218
EECON1 数据 EEPROM 控制寄存器 1） 89
INTCON （中断控制寄存器） 113
INTCON2 （中断控制寄存器 2） 114
INTCON3 （中断控制寄存器 3） 115
IPR1 （外设中断优先级寄存器 1） 122
IPR2 （外设中断优先级寄存器 2） 123
IPR3 （外设中断优先级寄存器 3） 124
MEMCON （外部存储器总线控制） 98

ODCON1 （外设漏极开路控制寄存器 1） 129
ODCON2 （外设漏极开路控制寄存器 2） 129
ODCON3 （外设漏极开路控制寄存器 3） 129
OSCCON （振荡器控制寄存器） 32
OSCTUNE （振荡器调节寄存器） 33
PADCFG1 （I/O 填充配置控制寄存器） 130
PIE1 （外设中断允许寄存器 1） 119
PIE2 （外设中断允许寄存器 2） 120
PIE3 （外设中断允许寄存器 3） 121
PIR1 （外设中断请求 （标志）寄存器 1） 116
PIR2 （外设中断请求 （标志）寄存器 2） 117
PIR3 （外设中断请求 （标志）寄存器 3） 118
PMADDRH （并行端口地址高字节，仅主控模式）. 164
PMCONH （并行端口控制高字节） 158
PMCONL （并行端口控制低字节） 159
PMEH （并行端口使能高字节） 161
PMEL （并行端口使能低字节） 162
PMMODEH （并行端口模式高字节） 160
PMMODEL （并行端口模式低字节） 161
PMSTAT （并行端口状态低字节） 163
PMSTAT （并行端口状态高字节） 162
RCON （复位控制） 50, 125
RCSTAx （接收状态和控制寄存器） 271
REFOCON （参考振荡器控制寄存器） 39
SSPCON2 （MSSPx 控制 2， I2C 主控模式） 236
SSPCON2 （MSSPx 控制寄存器 2， I2C 从动模式）237
SSPxCON1 （MSSPx 控制寄存器 1， I2C 模式） ...235
SSPxCON1 （MSSPx 控制寄存器 1， SPI 模式）...225
SSPxMSK （I2C 从动地址屏蔽） 237
SSPxSTAT （MSSPx 状态， I2C 模式） 234
SSPxSTAT （MSSPx 状态寄存器， SPI 模式） 224
STATUS 80
STKPTR （堆栈指针） 66
T0CON （Timer0 控制寄存器） 181
T1CON （Timer1 控制寄存器） 185
T2CON （Timer2 控制寄存器） 191
T3CON （Timer3 控制寄存器） 193
T4CON （Timer4 控制寄存器） 197
TXSTAx （发送状态和控制寄存器） 270
WDTCON （看门狗定时器控制） 74
寄存器汇总 75–79
间隔字符 （12 位）传送与接收 284
间接寻址 82
校准 （A/D 转换器） 299
晶体振荡器 / 陶瓷谐振器 35
绝对最大值 383

K

开发支持 379
勘误表 5
看门狗定时器 （WDT） 313, 321
编程注意事项 321
控制寄存器 321
相关的寄存器 322
振荡器故障期间 325
客户支持 433
客户通知服务 433
快速寄存器堆栈 67
框图
A/D 294
比较器 301
比较器参考电压模块 309
比较器模拟输入模型 304
比较器 I/O 配置 306
比较模式工组原理 202
表写操作 88
表读操作 87
并行主控 / 从动连接寻址缓冲器 168
并行主控 / 从动连接缓冲 167
并行 EEPROM （最多 15 位地址， 16 位数据） 179
并行 EEPROM （最多 15 位地址， 8 位数据） 179
波特率发生器 255
捕捉模式工作原理 201
部分复用寻址模式 171
部分复用的寻址应用 178
参考电压输出缓冲示例 311
传统并行从端口 165
读闪存程序存储器 91
对闪存程序存储器进行表写操作 93
多路分解寻址模式 171
EUSART 发送 279
EUSART 接收 281
复用的寻址应用 178
故障保护时钟监视器 325
看门狗定时器 321
LCD 控制 179
模拟输入模型 295
MSSP （I2C 主控模式） 253
MSSP （SPI 模式） 223
MSSPx （I2C 模式） 233
片上复位电路 49
片内稳压器连接 323
PIC18F6XJ1X （64 引脚） 10
PIC18F8XJ1X （80 引脚） 11
PLL 36
PMP 模块 157
PWM 工作原理 （简化） 204
器件时钟 31
Timer1 186
Timer1 （16 位读 / 写模式） 186
Timer2 192
Timer3 194
Timer3 （16 位读 / 写模式） 194
Timer4 198
通用 I/O 端口的操作 127
外部上电复位电路 （VDD 慢速上电） 51
完全复用的寻址模式 171
增强型 PWM 211
中断逻辑 112
16 位字写模式 102
16 位字节选择模式 103
16 位字节写模式 101
16 位模式下的 Timer0 182
8 位复用的地址和数据应用 178
8 位复用模式示例 105
8 位模式下的 Timer0 182
扩展的指令集
ADDFSR 372
ADDULNK 372
CALLW 373
MOVSF 373
MOVSS 374
PUSHL 374
SUBFSR 375
SUBULNK 375

L

LFSR 353
立即数变址寻址
和标准的 PIC18 指令 376
立即数变址寻址模式 376
漏极开路输出 128

 © 2007 Microchip Technology Inc.

M

脉宽调制。*请参见* PWM（CCP 模块）和 PWM（ECCP 模块）。
模数转换器。*请参见* A/D。
默认系统时钟 34
MOVF 353
MOVFF 354
MOVLB 354
MOVLW 355
MOVSF 373
MOVSS 374
MOVWF 355
MPLAB ASM30 汇编器、链接器和库管理器 380
MPLAB ICD 2 在线调试器 381
MPLAB ICE 2000 高性能通用在线仿真器 381
MPLAB PM3 器件编程器 381
MPLAB REAL ICE 在线仿真器系统 381
MPLAB 集成开发环境软件 379
MPLINK 目标链接器 /MPLIB 目标库管理器 380
MSSP
    $\overline{\text{ACK}}$ 脉冲 238, 241
    模块概述 223
    I^2C 模式。*请参见* I^2C 模式。
    SPI 主 / 从连接 227
MULLW 356
MULWF 356

N

NEGF 357
内部振荡器电路 37
    INTIO 模式 37
    INTOSC 频率漂移 38
    INTOSC 输出频率 38
    INTPLL 模式 37
    调节 38
内部互连模块。*请参见* I^2C。
内部稳压器规范 397
内部 RC 振荡电路
    与 WDT 一起使用 321
内核功能
    快速移植 8
    扩展的存储器 7
    扩展指令集 7
    纳瓦技术 7
    外部存储器总线 7
    振荡器选项和特点 7
NOP 357

P

配置不匹配复位 （CM） 51
配置位 313
PICSTART 2 开发编程器 382
PICSTART Plus 开发编程器 382
PIE 寄存器 119
PIR 寄存器 116
PLL 36
    HSPLL 和 ECPLL 振荡器模式 36
    与 INTOSC 一起使用 36
POP 358
PORTA
    LATA 寄存器 130
    PORTA 寄存器 130
    TRISA 寄存器 130
    相关的寄存器 132
PORTB
    LATB 寄存器 133
    PORTB 寄存器 133
    RB7:RB4 电平变化中断标志 （RBIF 位） 133
    TRISB 寄存器 133
    相关的寄存器 135
PORTC
    LATC 寄存器 136
    PORTC 寄存器 136
    RC3/SCKx/SCLx 引脚 241
    TRISC 寄存器 136
    相关的寄存器 138
PORTD
    LATD 寄存器 139
    PORTD 寄存器 139
    TRISD 寄存器 139
    相关的寄存器 141
PORTE
    LATE 寄存器 142
    PORTE 寄存器 142
    TRISE 寄存器 142
    相关的寄存器 144
PORTF
    LATF 寄存器 145
    PORTF 寄存器 145
    TRISF 寄存器 145
    相关的寄存器 147
PORTG
    LATG 寄存器 148
    PORTG 寄存器 148
    TRISG 寄存器 148
    相关的寄存器 150
PORTH
    LATH 寄存器 151
    PORTH 寄存器 151
    TRISH 寄存器 151
    相关的寄存器 153
PORTJ
    LATJ 寄存器 154
    PORTJ 寄存器 154
    TRISJ 寄存器 154
    相关的寄存器 155
PRI_IDLE 模式 46
PRI_RUN 模式 42
PUSH 358
PUSHL 374
PWM （CCP 模块）
    操作设置 205
    频率 / 分辨率示例 205
    PR2/PR4 寄存器 204
    TMR2 （TMR4）与 PR2 （PR4）匹配 204
    相关的寄存器 206
    占空比 204
    周期 204
PWM （CCP 模块）
    TMR2 与 PR2 匹配 211
    TMR4 与 PR4 匹配 197
PWM （ECCP 模块） 211
    半桥输出模式应用示例 214
    半桥模式 214
    CCPR1H:CCPR1L 寄存器 211
    复位的影响 221
    可编程死区延时 218
    频率 / 分辨率示例 212
    启动注意事项 219
    全桥输出应用示例 216
    全桥输出模式下的方向更改 216

全桥模式 215
设置 PWM 操作 221
输出关系 （低电平有效） 213
输出关系 （高电平有效） 213
输出配置 212
增强型 PWM 自动关闭 218
占空比 212
周期 211

Q

Q 时钟 205, 212
器件差异 431
器件概述 7
特性 （64 引脚器件） 9
特性 （80 引脚器件） 9
系列中各产品的具体信息 8
欠压复位。*请参见*欠压复位。
欠压复位 （BOR）
以及片内稳压器 324
欠压复位 （BOR） 51
检测 51
在休眠模式中禁止 51

R

RAM。*请参见*数据存储器。
RC_IDLE 模式 47
RC_RUN 模式 44
RCALL 359
RCON 寄存器
初始化期间的位状态 54
RESET 359
RETFIE 360
RETLW 360
RETURN 361
RLCF 361
RLNCF 362
RRCF 362
RRNCF 363
软件模拟器 （MPLAB SIM） 380

S

SCKx 223
SDIx 223
SDOx 223
SEC_IDLE 模式 46
SEC_RUN 模式 42
SETF 363
闪存程序存储器 87
表读和表写 87
表指针边界 90
擦除 92
擦除顺序 92
代码保护时的操作 96
读 91
基于操作的表指针
边界 90
控制寄存器 88
EECON1 和 EECON2 88
TABLAT （表锁存器）寄存器 90
TBLPTR （表指针）寄存器 90
相关的寄存器 96
写 93
写校验 96
意外终止 96
写操作顺序 93
写序列 （字编程） 95
闪存配置字 313
上电定时器 （PWRT） 52
上电复位。*请参见*上电复位。
上电复位 （POR） 51
上电延时 40
上电延时定时器 （PWRT） 40
延时序列 52
上拉配置 128
时钟源 33
复位时的默认系统时钟 34
使用 OSCCON 寄存器进行选择 34
时序图
A/D 转换 421
半桥 PWM 输出 214
并行从端口 407
并行从端口写操作 166, 169
并行从端口读操作 166, 169
并行主控读 408
并行主端口写 409
BRG 溢出时序 278
捕捉 / 比较 /PWM （包括 ECCP 模块） 410
程序存储器写 404
程序存储器读 403
从动同步 229
从休眠模式唤醒的转换时序 （HSPLL） 45
从空闲模式唤醒进入运行模式的转换时序 46
从 RC_RUN 模式到 PRI_RUN 模式的转换时序 44
从 SEC_RUN 模式到 PRI_RUN 模式 （HSPLL）的转换时序 43
重复启动信号 258
重复启动条件期间的总线冲突 （情形 1） 266
重复启动条件期间的总线冲突 （情形 2） 266
CLKO 和 I/O 402
带有时钟仲裁的波特率发生器 256
到 RC_RUN 模式的切换时序 44
第一个启动位时序 257
读， 16 位复用数据，部分复用地址 176
读， 16 位复用数据，完全复用 16 位地址 177
读， 16 位数据，多路分解地址 176
读， 8 位数据，部分复用地址，使能选通 174, 175
读， 8 位数据，完全复用 16 位地址 175
读写时序， 8 位数据，多路分解地址 173
读时序， 8 位数据，部分复用地址 173
读时序， 8 位数据，使能等待状态，部分复用地址 . 173
EUSART 同步发送 （主控 / 从动） 420
EUSART 同步接收 （主控 / 从动） 420
发送和应答时的总线冲突 263
发送间隔字符时序 284
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） 405
故障保护时钟监视器 326
缓慢上升时间 （MCLR 连接到 VDD， VDD 上升时间 > TPWRT） 53
^{2}C 总线启动 / 停止位 415
I^{2}C 总线数据 416
I^{2}C 从动模式 （10 位接收， SEN = 0） 246
I^{2}C 从动模式（10 位接收，SEN = 0，ADMSK = 01011） 245
I^{2}C 从动模式 （7 位接收， SEN = 0） 242
I^{2}C 从动模式（7 位接收，SEN = 0，ADMSK = 01011） 243
I^{2}C 从动模式 （7 位接收， SEN = 1） 250
I^{2}C 从动模式广播呼叫地址时序（7 位或 10 位地址模式） 252
I^{2}C 应答序列 262

 © 2007 Microchip Technology Inc.

I²C 主控模式 （7 位接收） 261
I²C 主控模式 （7 位或 10 位地址发送） 260
I²C 停止条件接收或发送模式 262
IMSSP I² 总线启动 / 停止位 418
进入休眠模式的转换时序 45
进入空闲模式的转换时序 46
进入 SEC_RUN 模式的转换时序 43
MSSP I²C 总线数据 418
PWM 方向更改 217
PWM 输出 204
PWM 自动关闭 （P1RSEN = 0，禁止自动重启） 220
PWM 自动关闭 （P1RSEN = 1，使能自动重启） 220
启动条件期间的总线冲突 （仅 SDAx） 264
启动条件期间由 SDAx 仲裁引起的 BRG 复位 265
全桥 PWM 输出 215
上电时的延时时序（MCLR 连接到 VDD, VDD 上升时间 < TPWRT） 52
上电时的延时时序 （MCLR 未连接到 VDD），情形 1 . 52
上电时的延时时序 （MCLR 未连接到 VDD），情形 2 . 53
时钟 / 指令周期 68
时钟同步 249
双速启动时钟转换 （从 INTRC 切换到 HSPLL） 324
SPI 主控模式示例 （CKE = 1） 412
SPI 从动模式 （CKE = 0） 413
SPI 从动模式示例 （CKE = 1） 414
SPI 主控模式示例 （CKE = 0） 411
SPI 模式 （从动模式， CKE=0） 230
SPI 模式 （从动模式， CKE=1） 230
SPI 模式 （主控模式） 228
Timer0 和 Timer1 外部时钟 406
停止条件期间的总线冲突 （情形 1） 267
停止条件期间的总线冲突 （情形 2） 267
同步发送 285
同步发送 （由 TXEN 位控制） 286
写， 16 位复用数据，部分复用地址 177
写， 16 位复用数据，完全复用 16 位地址 177
写， 8 位数据，部分复用地址 174
写， 8 位数据，多路分解地址 176
写， 8 位数据，使能等待状态，部分复用地址 174
写， 8 位数据，完全复用 16 位地址 175
外部时钟 （除 PLL 之外的所有模式） 400
休眠模式下的自动唤醒位 （WUE） 283
异步发送 280
异步发送 （背对背） 280
异步接收 282
在占空比接近 100% 时 PWM 更改方向 217
正常工作模式下的自动唤醒位 （WUE）时序 283
执行 SLEEP 的外部存储器总线 （扩展单片机模式） 104, 106
执行 TBLRD 的外部存储器总线 （扩展单片机模式） 104, 106
主控模式同步接收 （由 SREN 位控制） 287
自动波特率计算 278
时序图和规范
CLKO 和 I/O 要求 402, 403
并行从端口要求 407
并行主端口写 409
并行主端口读要求 408
捕捉 / 比较 /PWM 要求 （包括 ECCP 模块） 410
程序存储器写要求 404
EUSART 同步发送要求 420
EUSART 同步接收要求 420
复位、看门狗定时器 （WDT）、振荡器起振定时器 （OST）、上电延时定时器 （PWRT）和欠压复位 405
I²C 总线启动 / 停止位要求 （从动模式） 415
I²C 总线数据要求 （从动模式） 417
MSSP I²C 总线启动 / 停止位要求 418
MSSP I²C 总线数据要求 419
内部 RC 精度 （INTOSC 和 INTRC 时钟源） 401
PLL 时钟 401
SPI 从动模式要求示例 （CKE = 1） 414
SPI 模式要求示例 （从动模式， CKE = 0） 413
SPI 模式要求示例 （主控模式， CKE = 0） 411
Timer0 和 Timer1 外部时钟要求 406
外部时钟要求 400
数据寻址模式 81
变址立即数偏移 84
固有和立即数寻址 81
间接寻址 81
立即数变址寻址 84
BSR 86
受影响的指令 84
映射快速操作存储区 86
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） 85
直接寻址 81
数据存储器 70
存储器映射图 71
特殊功能寄存器 73
存储区选择寄存器 （BSR） 70
快速操作存储区 72
扩展指令集 84
特殊功能寄存器 73
地址共用 74
通用寄存器 72
双字指令
示例情形 69
双速启动 313, 324
SLEEP 364
SPI 模式 （MSSP） 223
从动选择 223
从动选择同步 229
从动模式 229
串行时钟 223
串行数据输出 223
串行数据输入 223
典型连接 227
复位的影响 231
工作原理 226
SPI 时钟 228
SSPxBUF 寄存器 228
SSPxSR 寄存器 228
时钟速度，模块相互关系 231
使能 SPI I/O 227
相关的寄存器 232
在功耗管理模式下的工作方式 231
主 / 从连接 227
主控模式 228
总线模式兼容性 231
SSPOV 259
SSPOV 状态标志位 259
SSPxSTAT 寄存器
R/W 位 238, 241
SSx 223
SUBFSR 375
SUBFWB 364
SUBLW 365
SUBULNK 375
SUBWF 365

SUBWFB 366
所有寄存器的初始化状态 55–60
SWAPF 366

T

TBLRD 367
TBLWT 368
特殊功能寄存器
  共享的寄存器 74
特殊事件触发器。*请参见*比较 （ECCP 模块）
Timer0 181
  工作原理 182
  时钟源选择 （T0CS 位） 182
  时钟源边沿选择 （T0SE 位） 182
  相关的寄存器 183
  溢出时中断 183
  预分频比选择 （T0PS2:T0PS0 位） 183
  预分频器 183
    切换分配 183
  预分频器。*请参见*预分频器， Timer0。
  预分频器分配 （PSA 位） 183
  16 位读写模式 182
Timer1 185
  16 位读 / 写模式 187
  复位，使用 ECCPx 特殊事件触发信号 188
  工作原理 186
  TMR1H 寄存器 185
  TMR1L 寄存器 185
  相关的寄存器 189
  异步计数器模式下的注意事项 188
  用作时钟源 187
  用作实时时钟 188
  特殊事件触发器 （ECCP） 210

  振荡器 185, 187
    布线注意事项 187
  振荡器，作为辅助时钟 33
  中断 188
Timer2 191
  工作原理 191
  PR2 寄存器 211
  输出 192
  TMR2 与 PR2 匹配中断 211
  相关的寄存器 192
  中断 192
Timer3 193
  16 位读 / 写模式 195
  工作原理 194
  特殊事件触发信号 （ECCPx） 195
  TMR3H 寄存器 193
  TMR3L 寄存器 193
  相关的寄存器 195
  溢出时中断 193, 195
  振荡器 193, 195
Timer4 197
  工作原理 197
  后分频器。*请参见*后分频器， Timer4。
  PR4 寄存器 197
  输出 198
  TMR4 与 PR4 匹配中断 197, 198
  TMR4 寄存器 197
  相关的寄存器 198
  预分频器。*请参见*预分频器， Timer4。
同步中断字符自动唤醒 282
TSTFSZ 369
TXSTAx 寄存器
  BRGH 位 273

V

VDDCORE/VCAP 引脚 323

W

外部存储器总线 97
  程序存储模式 100
    单片机 100
    扩展的单片机 100
  等待状态 100
  地址和数据线使用方法 （表格） 99
  地址和数据宽度 99
  地址移位 99
  端口引脚上的弱上拉 100
  工作在功耗管理模式下 107
  I/O 端口功能 97
  控制 98
  16 位数据宽度模式 100
  16 位字写模式 102
  16 位字节选择模式 103
  16 位字节写模式 101
  16 位模式时序 104
  8 位数据宽度模式 105
  8 位模式时序 106
外部振荡器模式
  HS 35
  外部时钟输入 （EC 模式） 36
WCOL 257, 258, 259, 262
WCOL 状态标志位 257, 258, 259, 262
稳压器 （片内） 323
  上电要求求 324
  休眠模式下的操作 324
WWW 在线支持 5

X

XORLW 369
XORWF 370

Y

引脚功能
  AVDD 29
  AVDD 19
  AVSS 29
  AVSS 19
  ENVREG 19, 29
  MCLR 12, 20
  OSC1/CLKI/RA7 ..... 12, 20
  OSC2/CLKO/RA6 ..... 12, 20
  RA0/AN0 ..... 13, 21
  RA1/AN1 ..... 13, 21
  RA2/AN2/VREF- 13, 21
  RA3/AN3/VREF+ 13, 21
  RA4/PMD5/T0CKI 21
  RA4/T0CKI 13
  RA5/AN4 13
  RA5/PMD4/AN4 21
  RA6 ..... 13, 21
  RA7 ..... 13, 21
  RB0/FLT0/INT0 ..... 14, 22
  RB1/INT1/PMA4 ..... 14, 22
  RB2/INT2/PMA3 ..... 14, 22
  RB3/INT3//PMA2/ECCP2/P2A 22
  RB3/INT3/PMA2 14
  RB4/KBI0/PMA1 ..... 14, 22
  RB5/KBI1/PMA0 ..... 14, 22

 © 2007 Microchip Technology Inc.

RB6/KBI2/PGC 14, 22
RB7/KBI3/PGD 14, 22
RC0/T1OSO/T13CKI 15, 23
RC1/T1OSI/ECCP2/P2A 15, 23
RC2/ECCP1/P1A 15, 23
RC3/SCK1/SCL1 ..... 15, 23
RC4/SDI1/SDA1 ..... 15, 23
RC5/SDO1 ..... 15, 23
RC6/TX1/CK1 ..... 15, 23
RC7/RX1/DT1 ..... 15, 23
RD0/AD0/PMD0 24
RD0/PMD0 16
RD1/AD1/PMD1 24
RD1/PMD1 16
RD2/AD2/PMD2 24
RD2/PMD2 16
RD3/AD3/PMD3 24
RD3/PMD3 16
RD4/AD4/PMD4/SDO2 24
RD4/PMD4/SDO2 16
RD5/AD5/PMD5/SDI2/SDA2 24
RD5/PMD5/SDI2/SDA2 16
RD6/AD6/PMD6/SCK2/SCL2 24
RD6/PMD6/SCK2/SCL2 16
RD7/AD7/PMD7/SS2 24
RD7/PMD7/SS2 16
RE0/AD8/PMRD/P2D 25
RE0/PMRD/P2D 17
RE1/AD9/PMWR/P2C 25
RE1/PMWR/P2C 17
RE2/AD10/PMBE/P2B 25
RE2/PMBE/P2B 17
RE3/AD11/PMA13/P3C/REFO 25
RE3/PMA13/P3C/REFO 17
RE4/AD12/PMA12/P3B 25
RE4/PMA12/P3B 17
RE5/AD13/PMA11/P1C 25
RE5/PMA11/P1C 17
RE6/AD14/PMA10/P1B 25
RE6/PMA10/P1B 17
RE7/AD15/PMA9/ECCP2/P2A 25
RE7/PMA9/ECCP2/P2A 17
RF1/AN6/C2OUT 18, 26
RF2/PMA5/AN7/C1OUT 18, 26
RF3/AN8/C2INB 18, 26
RF4/AN9/C2INA 18, 26
RF5/AN10/C1INB/CVREF 18
RF5/PMD2/AN10/C1INB/CVREF 26
RF6/AN11/C1INA 18
RF6/PMD1/AN11/C1INA 26
RF7/PMD0/SS1 26
RF7/SS1 18
RG0/PMA8/ECCP3/P3A 19, 27
RG1/PMA7/TX2/CK2 ..... 19, 27
RG2/PMA6/RX2/DT2 ..... 19, 27
RG3/PMCS1/CCP4/P3D 19, 27
RG4/PMCS2/CCP5/P1D 19, 27
RH0/A16 28
RH1/A17 28
RH2/A18/PMD7 28
RH3/A19/PMD6 28
RH4/PMD3/AN12/P3C/C2INC 28
RH5/PMBE/AN13/P3B/C2IND 28
RH6/PMRD/AN14/P1C/C1INC 28
RH7/PMWR/AN15/P1B 28
RJ0/ALE 29
RJ1/OE 29
RJ2/WRL 29
RJ3/WRH 29
RJ4/BA0 29
RJ5/CE 29
RJ6/LB 29
RJ7/UB 29
VDD 29
VDD 19
VDDCORE/VCAP 19, 29
VSS 29
VSS 19
硬件乘法器 109
工作原理 109
性能比较 （表格） 109
8 × 8 乘法算法 109
预分频器
Timer2 212
预分频器， Timer0 183
预分频器， Timer2 （Timer4） 205

Z

在线串行编程 （ICSP） 313, 327
在线调试器 327
增强型捕捉 / 比较 /PWM （ECCP） 207
捕捉模式。*请参见*捕捉 （ECCP 模块）。
定时器资源 208
ECCP1/ECCP3 输出和程序存储器模式 208
ECCP1 的引脚配置 209
ECCP1 和 ECCP3 对 CCP4 和 CCP5 的使用 208
ECCP2 的引脚配置 209
ECCP2 输出和程序存储器模式 208
ECCP3 的引脚配置 210
PWM 模式。*请参见* PWM （ECCP 模块）。
输出和配置 208
增强型通用同步 / 异步收发器（EUSART）。*请参见* EUSART。
振荡器选择 313
振荡器， Timer1 ..... 185, 195
振荡器， Timer3 193
振荡器配置 31
EC 31
ECPLL 31
HS 31
HSPLL 31
INTIO1 31
INTIO2 31
INTPLL1 31
INTPLL2 31
内部振荡器电路 37
振荡器起振定时器 （OST） 40
振荡器切换 33
振荡器转换 34
直接寻址 82
直流特性
掉电电流和供电电流 386
中断 111
期间的现场保护 126
INTx 引脚 126
PORTB 电平变化中断 126
TMR0 126
中断，标志位
电平变化中断 （RB7:RB4）标志 （RBIF 位） 133
中断源 313
A/D 转换完成 295
比较完成 （CCP） 202

捕捉完成 （CCP） 201
电平变化中断 （RB7:RB4） 133
TMR0 溢出 183
TMR2 与 PR2 匹配 （PWM） 211
TMR3 溢出 193, 195
TMR4 与 PR4 匹配 198
TMR4 与 PR4 匹配 （PWM） 197
指令周期 68
时钟机制 68
指令流 / 流水线 68
指令集 329
ADDLW 335
ADDWF 335
ADDWF （立即数变址寻址模式） 377
ADDWFC 336
ANDLW 336
ANDWF 337
BC 337
BCF 338
标准指令 329
BN 338
BNC 339
BNN 339
BNOV 340
BNZ 340
BOV 343
BRA 341
BSF 341
BSF （立即数变址寻址模式） 377
BTFSC 342
BTFSS 342
BTG 343
BZ 344
CALL 344
操作码字段说明 330
CLRF 345
CLRWDT 345
COMF 346
CPFSEQ 346
CPFSGT 347
CPFSLT 347
DAW 348
DCFSNZ 349
DECF 348
DECFSZ 349
GOTO 350
INCF 350
INCFSZ 351
INFSNZ 351
IORLW 352
IORWF 352
扩展指令 371
使用 MPLAB IDE 工具 378
使能时的注意事项 376
语法 371
LFSR 353
MOVF 353
MOVFF 354
MOVLB 354
MOVLW 355
MOVWF 355
MULLW 356
MULWF 356
NEGF 357
NOP 357
POP 358
PUSH 358
RCALL 359
RESET 359
RETFIE 360
RETLW 360
RETURN 361
RLCF 361
RLNCF 362
RRCF 362
RRNCF 363
SETF 363
SETF （立即数变址寻址模式） 377
SLEEP 364
SUBFWB 364
SUBLW 365
SUBWF 365
SUBWFB 366
SWAPF 366
TBLRD 367
TBLWT 368
通用格式 331
TSTFSZ 369
XORLW 369
XORWF 370
主复位 （MCLR） 51
主控同步串口 （MSSP）。*请参见* MSSP。

 © 2007 Microchip Technology Inc.

注：

注：

 © 2007 Microchip Technology Inc.

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销售办事处联系。

器件编号	X	/XX	XXX
器件	温度范围	封装	编程模式

器件	PIC18F66J11/66J16/67J11(1), PIC18F86J11/86J16/87J11(1), PIC18F66J11/66J16/67J11(2), PIC18F86J11/86J16/87J11T(2)
温度范围	I = -40°C 到 +85°C（工业级）
封装	PT = TQFP（薄型正方扁平封装）
编程模式	QTP、SQTP、代码或特殊要求 （其他情况空白）

示例：

a） PIC18F87J11-I/PT 301 表示工业级温度，TQFP 封装， QTP 模式 #301。

b） PIC18F66J16-I/PT 表示卷带式，工业级温度，TQFP 封装。

注 1： F = 标准电压范围

2： T = 卷带式封装

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
Harbour 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 – 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 – 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 – 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 – 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 – 深圳
Tel: 86-755-8203-2660
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: 86-29-8833-7252
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 - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - 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

10/05/07

© 2007 Microchip Technology Inc.

1.0 器件概述 7
2.0 振荡器配置 31
3.0 功耗管理模式 41
4.0 复位 49
5.0 存储器构成 61
6.0 闪存程序存储器 87
7.0 外部存储器总线 97
8.0 8 × 8 硬件乘法器 109
9.0 中断 111
10.0 I/O 端口 127
11.0 并行主端口 157
12.0 Timer0 模块 181
13.0 Timer1 模块 185
14.0 Timer2 模块 191
15.0 Timer3 模块 193
16.0 Timer4 模块 197
17.0 捕捉 / 比较 /PWM （CCP）模块 199
18.0 增强型捕捉 / 比较 /PWM （ECCP）模块 207
19.0 主控同步串口 （MSSP）模块 223
20.0 增强型通用同步 / 异步收发器 （EUSART） 269
21.0 10 位模数转换器 （A/D）模块 291
22.0 比较器模块 301
23.0 比较器参考电压模块 309
24.0 CPU 的特殊性能 313
25.0 指令集综述 329
26.0 开发支持 379
27.0 电气规范 383
28.0 DC 和 AC 特性图表 425
29.0 封装信息 427
Appendix A: 版本历史 431
Appendix B: 器件差异 431
TMicrochip 网站 433
变更通知客户服务 433
客户支持 433
..... 433
读者反馈表 434
产品标识体系 445