



PIC18F46J50

数据手册

采用 nanoWatt XLP 技术的
28/44 引脚低功耗
高性能 USB 单片机

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更安全的产品之一。
- 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、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rfPIC 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL 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、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、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。

© 2009 -2011, Microchip Technology Inc. 版权所有。

ISBN: 978-1-61341-585-6

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-16949:2009 认证。Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外, Microchip 在开发系统的设计和和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

采用 nanoWatt XLP 技术的 28/44 引脚低功耗、高性能 USB 单片机

nanoWatt XLP 功耗管理功能，实现超低功耗：

- 深度休眠模式：CPU 关闭、外设关闭，电流低至 13 nA，采用 RTCC 时电流低至 850 nA，并能够实现以下功能：
 - 可由外部触发信号、可编程 WDT 超时或 RTCC 闹钟唤醒
 - 超低功耗唤醒（Ultra Low-Power Wake-up, ULPWU）
- 休眠模式：CPU 关闭、外设关闭、SRAM 工作，可实现快速唤醒，电流典型值低至 105 nA
- 空闲模式：CPU 关闭、外设工作，电流典型值低至 2.3 μ A
- 运行模式：CPU 工作、外设工作，电流典型值低至 6.2 μ A
- 使用 RTCC 的 Timer1 振荡器 32 kHz 时的典型电流消耗为 1 μ A
- 看门狗定时器：2V 时的典型电流消耗为 0.8 μ A

单片机特性：

- 低功耗、高速 CMOS 闪存技术
- 其架构针对 C 编译器进行了优化，用于可重入代码
- 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
- 扩展型看门狗定时器（Watchdog Timer, WDT）：
 - 可编程周期从 4 ms 到 131s
- 通过两个引脚进行单电源在线串行编程（In-Circuit Serial Programming™, ICSP™）
- 通过两个引脚进行带有 3 个断点的在线调试（In-Circuit Debug, ICD）
- 工作电压范围为 2.0V 到 3.6V
- 片上 2.5V 稳压器
- 闪存程序存储器至少可承受 10,000 次擦写并具有最少 20 年的数据保存时间

通用串行总线（USB）特性：

- 符合 USB V2.0 规范
- 全速（12 Mbps）和低速（1.5 Mbps）工作
- 支持控制、中断、同步和批量传输
- 支持最多 32 个端点（两个方向各 16 个）
- USB 模块能将器件上的任何 RAM 存储单元用作 USB 端点缓冲区
- 片上 USB 收发器，无需晶振即可工作

灵活的振荡器结构：

- 用于 USB 的高精度（典型值为 $\pm 0.15\%$ ）内部振荡器
- 两种外部时钟模式，频率最高为 48 MHz（12 MIPS）
- 低功耗 31 kHz 内部 RC 振荡器
- 可调节的内部振荡器（31 kHz 到 8 MHz，采用 PLL 时最多 48 MHz）
- 辅助振荡器使用 Timer1（工作频率为 32 kHz）
- 故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM）：
 - 当时钟停止时可使器件安全关闭
- 双速振荡器启动
- 可编程的参考时钟输出发生器

外设特点：

- 外设引脚选择：
 - 允许许多个外设独立映射 I/O
 - 连续的硬件完整性检查和安全互锁可防止意外更改配置
- 硬件实时时钟和日历（Real-Time Clock and Calendar, RTCC）：
 - 提供时钟、日历和闹钟功能
- 高灌 / 拉电流：25 mA/25 mA（PORTB 和 PORTC）
- 引脚可耐受 5.5V 输入（仅限只能用作数字功能的引脚）
- 4 个可编程外部中断
- 4 个输入电平变化中断
- 两个增强型捕捉 / 比较 / PWM（Enhanced Capture/Compare/PWM, ECCP）模块：
 - 1、2 或 4 路 PWM 输出
 - 可选择的极性
 - 可编程的死区
 - 自动关闭和自动重启
 - 脉冲转向（steering）控制
- 两个主同步串行口（Master Synchronous Serial Port, MSSP）模块，支持 3 线 SPI（所有 4 种模式）和 I²C™ 主 / 从模式
- 全双工主 / 从 SPI DMA 引擎
- 8 位并行主端口 / 增强型并行从端口
- 两个带输入多路选择的轨到轨模拟比较器
- 最多 13 路通道的 10 位模数（Analog-to-Digital, A/D）转换器模块：
 - 自动采集功能
 - 可在休眠模式下进行转换
 - 自校准
- 高 / 低压检测模块
- 充电时间测量单元（Charge Time Measurement Unit, CTMU）：
 - 支持容性触摸屏和容性开关的触摸传感
 - 为流量测量和简单温度检测提供高精度高分辨率时间测量
- 两个增强型 USART 模块：
 - 支持 RS-485、RS-232 和 LIN/J2602
 - 接收到启动位时自动唤醒
- 自动波特率检测

PIC18F46J50 系列

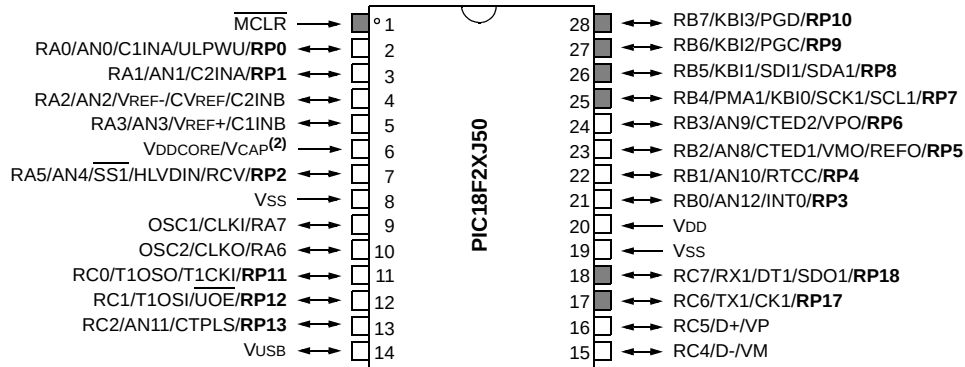
PIC18F/LF ⁽¹⁾ 器件	引脚数	程序存储器 (字节)	SRAM (字节)	可重映射引脚数	8/16 位 定时器	ECCP/ (PWM)	EUSART		MSSP		10 位 A/D (通道数)	比较器	深度休眠	PMP/PSP	CTMU	RTCC	USB
									带 DMA 的 SPI	I ² C™							
PIC18F24J50	28	16K	3776	16	2/3	2	2	2	有	有	10	2	有	无	有	有	有
PIC18F25J50	28	32K	3776	16	2/3	2	2	2	有	有	10	2	有	无	有	有	有
PIC18F26J50	28	64K	3776	16	2/3	2	2	2	有	有	10	2	有	无	有	有	有
PIC18F44J50	44	16K	3776	22	2/3	2	2	2	有	有	13	2	有	有	有	有	有
PIC18F45J50	44	32K	3776	22	2/3	2	2	2	有	有	13	2	有	有	有	有	有
PIC18F46J50	44	64K	3776	22	2/3	2	2	2	有	有	13	2	有	有	有	有	有
PIC18LF24J50	28	16K	3776	16	2/3	2	2	2	有	有	10	2	无	无	有	有	有
PIC18LF25J50	28	32K	3776	16	2/3	2	2	2	有	有	10	2	无	无	有	有	有
PIC18LF26J50	28	64K	3776	16	2/3	2	2	2	有	有	10	2	无	无	有	有	有
PIC18LF44J50	44	16K	3776	22	2/3	2	2	2	有	有	13	2	无	有	有	有	有
PIC18LF45J50	44	32K	3776	22	2/3	2	2	2	有	有	13	2	无	有	有	有	有
PIC18LF46J50	44	64K	3776	22	2/3	2	2	2	有	有	13	2	无	有	有	有	有

注 1: 第 1.3 节 “系列中各器件的详细说明”、第 4.6 节 “深度休眠模式” 和 第 27.3 节 “片上稳压器” 详细描述了本系列器件中 PIC18F 和 PIC18LF 两类型号的功能差异。

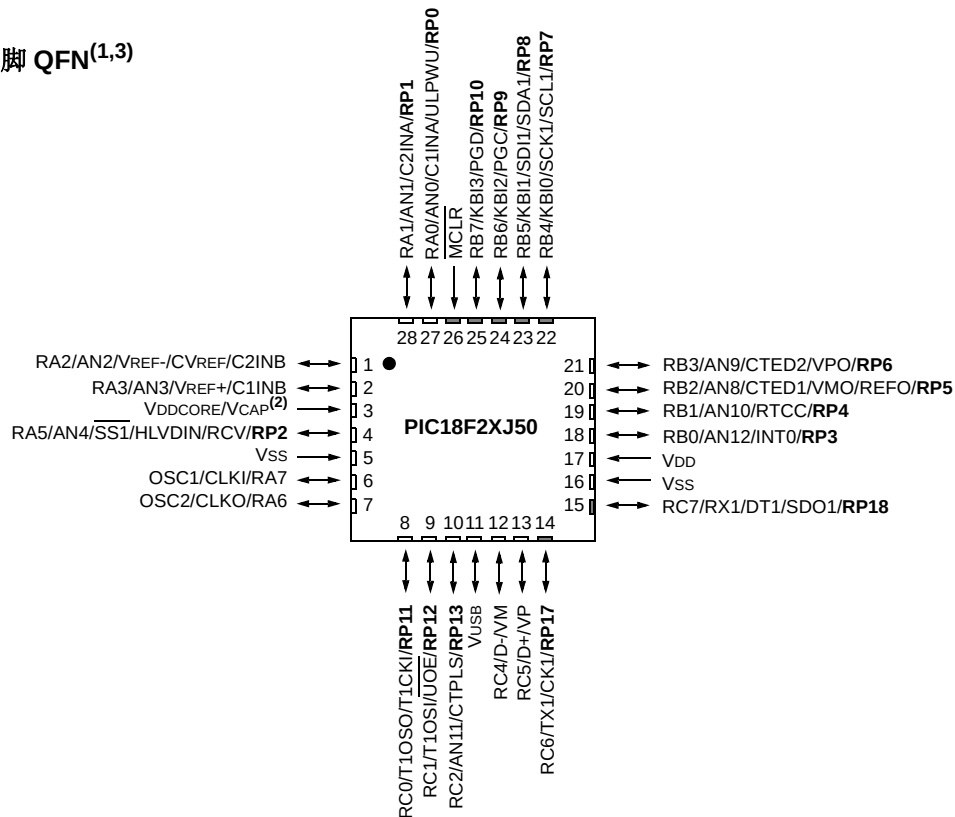
引脚图

28 引脚 SPDIP/SOIC/SSOP⁽¹⁾

■ = 最高可耐受 5.5V 电压的引脚



28 引脚 QFN^(1,3)

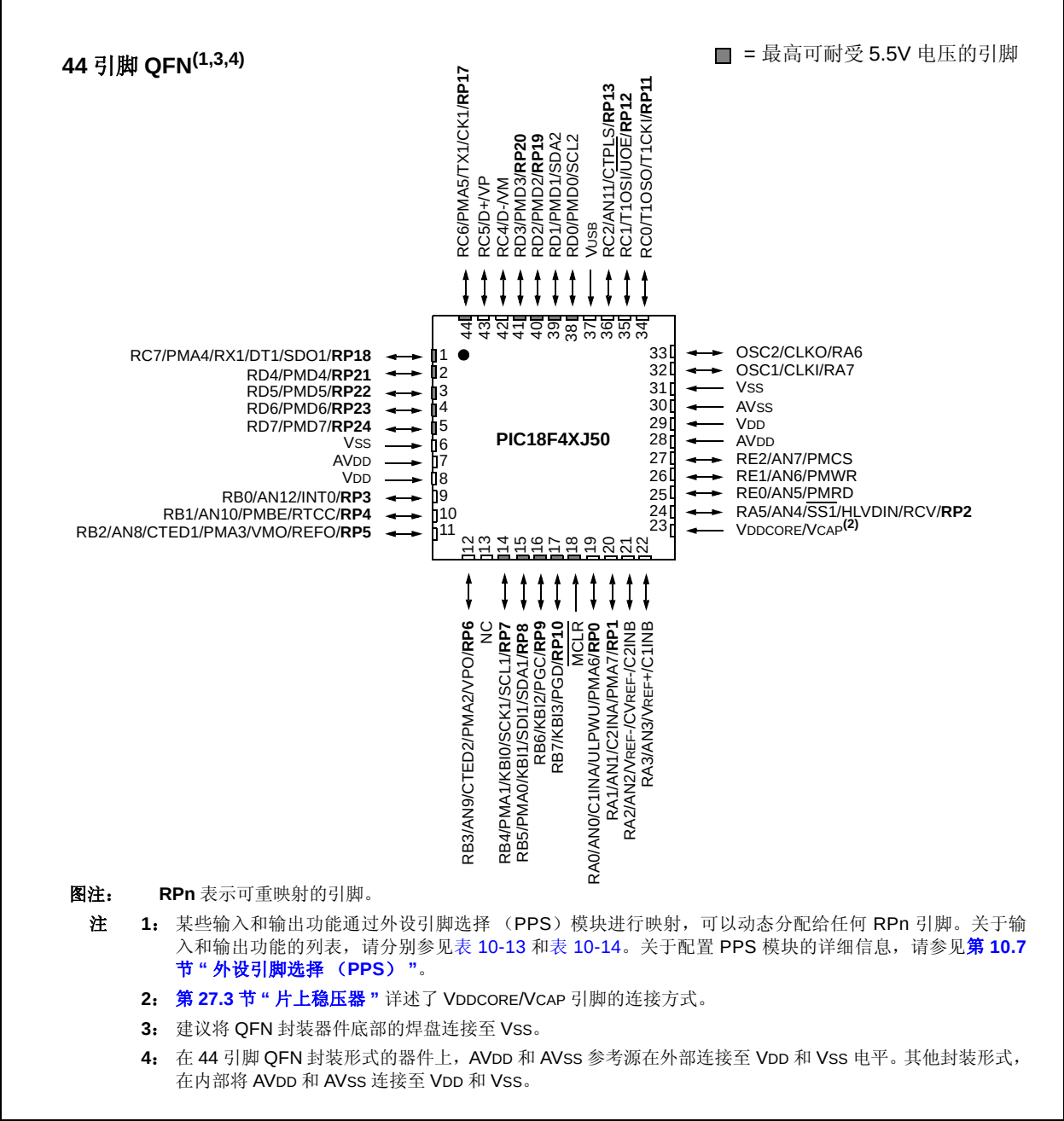


图注: RPn 表示可重映射的引脚。

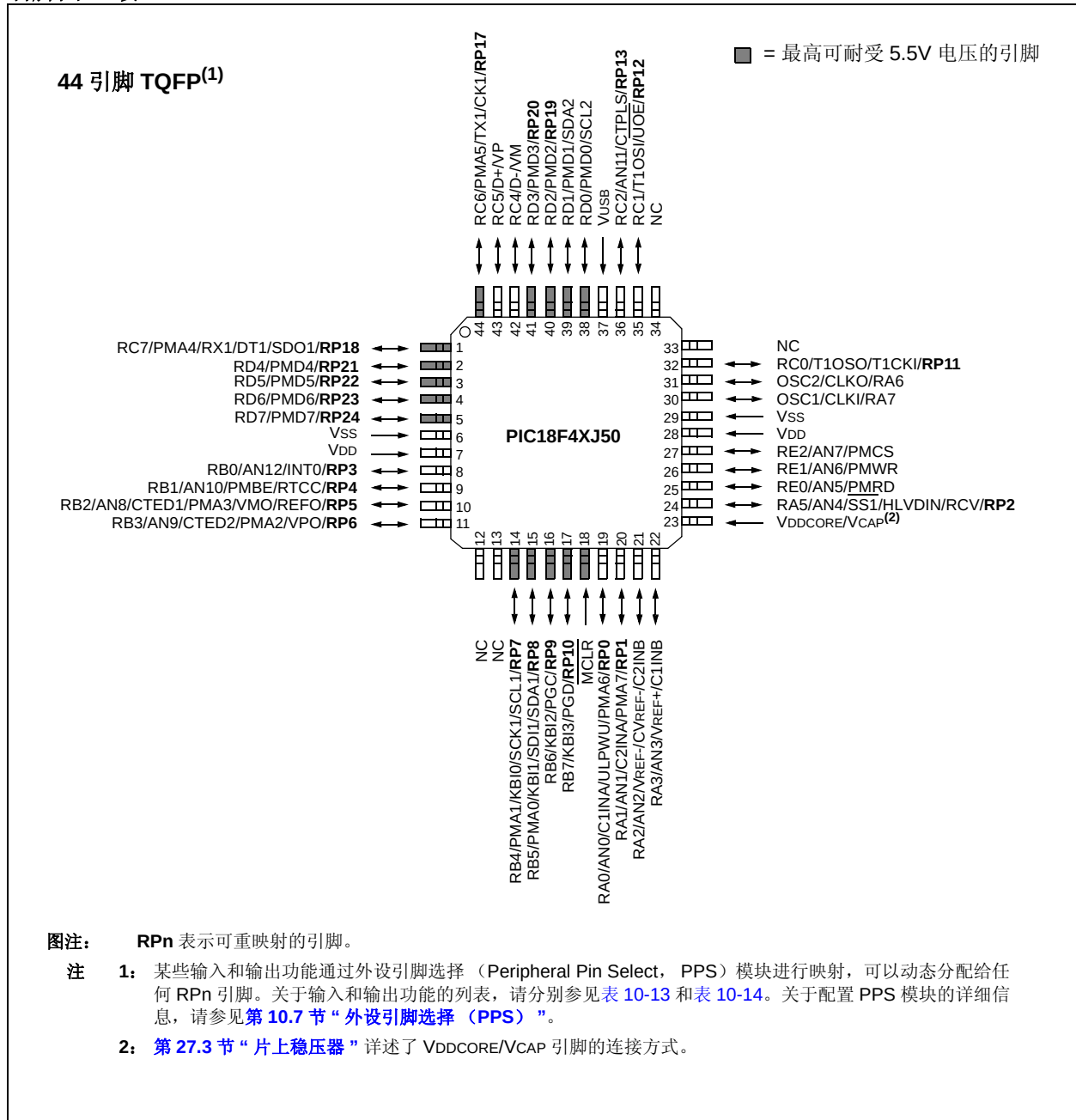
- 注
- 1: 某些输入和输出功能通过外设引脚选择 (Peripheral Pin Select, PPS) 模块进行映射, 可以动态分配给任何 RPn 引脚。关于输入和输出功能的列表, 请分别参见表 10-13 和表 10-14。关于配置 PPS 模块的详细信息, 请参见第 10.7 节“外设引脚选择 (PPS)”。
 - 2: 第 27.3 节“片上稳压器”详述了 VDDCORE/VCAP 引脚的连接方式。
 - 3: 建议将 QFN 封装器件底部的焊盘连接至 Vss。

PIC18F46J50 系列

引脚图（续）



引脚图 (续)



PIC18F46J50 系列

目录

1.0	器件概述	11
2.0	PIC18FJ 单片机入门指南	29
3.0	振荡器配置	35
4.0	低功耗模式	47
5.0	复位	63
6.0	存储器构成	77
7.0	闪存程序存储器	103
8.0	8 x 8 硬件乘法器	113
9.0	中断	115
10.0	I/O 端口	131
11.0	并行主端口 (PMP)	169
12.0	Timer0 模块	195
13.0	Timer1 模块	199
14.0	Timer2 模块	211
15.0	Timer3 模块	213
16.0	Timer4 模块	223
17.0	实时时钟和日历 (RTCC)	225
18.0	增强型捕捉 / 比较 / PWM (ECCP) 模块	245
19.0	主同步串行口 (MSSP) 模块	269
20.0	增强型通用同步 / 异步收发器 (EUSART)	323
21.0	10 位模数转换器 (A/D) 模块	347
22.0	通用串行总线 (USB)	357
23.0	比较器模块	385
24.0	比较器参考电压模块	391
25.0	高 / 低压检测 (HLVD)	395
25.0	充电时间测量单元 (CTMU)	401
26.0	CPU 的特殊功能	417
27.0	指令集汇总	435
28.0	开发支持	485
29.0	电气特性	489
30.0	封装信息	531
附录 A:	版本历史	545
附录 B:	器件差异	545
索引		547
Microchip 网站		559
变更通知客户服务		559
客户支持		559
读者反馈表		560
产品标识体系		561

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。

PIC18F46J50 系列

注:

1.0 器件概述

本文档包含以下器件的具体信息：

- PIC18F24J50
- PIC18F25J50
- PIC18F26J50
- PIC18F44J50
- PIC18F45J50
- PIC18F46J50
- PIC18LF24J50
- PIC18LF25J50
- PIC18LF26J50
- PIC18LF44J50
- PIC18LF45J50
- PIC18LF46J50

本系列是新面世的低电压通用串行总线（USB）单片机产品系列，在保留所有 PIC18 单片机的主要传统优点（即出色的计算性能以及丰富的功能集）的同时，性价比极高。这些特性使得 PIC18F46J50 系列成为许多高性能、低成本应用的理想选择。

1.1 内核特性

1.1.1 纳瓦技术

PIC18F46J50 系列的所有器件具有一系列能在工作时显著降低功耗的功能。主要功能包含：

- **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降低大约 90%。
- **多种空闲模式：**控制器还可在其 CPU 内核禁止而外设仍然工作的情况下工作。处于这些状态时，功耗能降得更低，只有正常工作时的 4%。
- **动态模式切换：**在器件工作期间可由用户代码调用该功耗管理模式，允许用户将节能的理念融入到其应用软件设计中。

1.1.2 通用串行总线（USB）

PIC18F46J50 系列器件具有全功能的 USB 通信模块，该模块带有符合 USB 规范 2.0 版的内置收发器。该模块支持所有支持的数据传输类型的低速和全速通信。

1.1.3 振荡器选项和特性

PIC18F46J50 系列的所有器件可提供 5 个不同的振荡器选项，使用户在开发应用硬件时有很大的选择范围。这些选项包括：

- 两种晶振模式，使用晶振或陶瓷谐振器。
- 两种外部时钟模式，提供四分频时钟输出选项。
- 一个内部振荡器模块，它提供一个 8 MHz 的时钟源和一个 INTRC 时钟源（振荡频率大约为 31 kHz，温度和 VDD 变化时频率保持稳定），以及用户可选的 6 种时钟频率（从 125 kHz 到 4 MHz），因此共有 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，可在高速晶振、外部和内部振荡器模式下使用，可使时钟速度最高达到 48 MHz。
- 双时钟操作，允许 USB 模块依靠高频振荡器运行，而单片机的其余部分采用其他频率的时钟。

内部振荡器模块还提供了一个稳定的参考源，增加了以下功能以使 PIC18F46J50 系列器件更可靠地工作：

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

1.1.4 扩展存储器

PIC18F46J50 系列为应用程序代码提供了充足的空间——从 16 KB 到 64 KB 的代码空间。程序存储器的闪存单元经评测最多可承受 10000 次擦写。在不刷新的情况下，数据保存时间保守地估计在 20 年以上。

正常工作期间，闪存程序存储器是可读写的。PIC18F46J50 系列还为动态应用数据提供了充足的空间——最大 3.8 KB 的数据 RAM。

PIC18F46J50 系列

1.1.5 扩展指令集

PIC18F46J50 系列在 PIC18 指令集的基础上实现了可选择的扩展，添加了 8 个新指令和一个变址寻址模式。此扩展可以使用一个器件配置选项使能，它是为优化原来采用高级语言（如 C）开发的可重入应用程序代码而专门设计的。

1.1.6 移植方便

无论存储器容量如何，所有器件都共享相同的丰富外设，使应用的扩展和升级工作变得轻松而容易。

整个系列的引脚排列设计一致也有助于向引脚更多的器件移植。

PIC18F46J50 系列的引脚还与其他 PIC18 系列（如 PIC18F4550、PIC18F2450 和 PIC18F45J10）的引脚兼容。这为应用的更新换代拓展了空间，使开发人员能在保留相同功能集的同时在 Microchip PIC18 系列中选择不同价位的器件。

1.2 其他特殊功能

- **通信：**PIC18F46J50 系列具有一系列串行和并行通信外设，包括符合 USB 规范 2.0 版的全功能 USB 通信模块。该器件还包含两个独立的增强型 USART 和两个主同步串行口（MSSP）模块，可以支持串行外设接口（Serial Peripheral Interface, SPI）和 I²C™（主 / 从）工作模式。器件还具有一个并行端口，可以配置作为并行主端口（Parallel Master Port, PMP）或作为并行从端口（Parallel Slave Port, PSP）。
- **ECCP 模块：**本系列的所有器件都具有 2 个增强型捕捉 / 比较 / PWM（ECCP）模块，以最大程度提高控制应用的灵活性。在同一时间，最多可以使用 4 种不同时间基来执行几项不同的操作。每个 ECCP 模块最多可提供 4 个 PWM 输出，从而总共可提供 8 个 PWM 输出。ECCP 还提供许多有用功能，包括极性选择、可编程死区、自动关闭和重启，以及半桥和全桥输出模式。

- **10 位 A/D 转换器：**该模块具备可编程采集时间，从而不必在选择通道和启动转换之间等待一个采样周期，因而减少了代码开销。
- **扩展型看门狗定时器（WDT）：**该增强型版本融入了一个 16 位预分频器，可以提供在不同工作电压和温度下保持稳定的扩展超时范围。超时周期请参见第 30.0 节“电气特性”。

1.3 系列中各器件的详细说明

PIC18F46J50 系列器件提供 28 引脚和 44 引脚封装形式。图 1-1 和图 1-2 分别为这两类器件的框图。这些器件在以下两个方面存在差异：

- 闪存程序存储器（三种容量：PIC18FX4J50 器件为 16 KB，PIC18FX5J50 器件为 32 KB，PIC18FX6J50 器件为 64 KB）
- I/O 端口数（28 引脚器件有 3 个双向端口，44 引脚器件有 5 个双向端口）

本系列器件的所有其他功能都是相同的。表 1-1 和表 1-1 中总结了这些功能。

PIC18F2XJ50 器件的引脚排列在表 1-3 中列出。PIC18F4XJ50 器件的引脚排列在表 1-4 中列出。

PIC18F46J50 系列器件提供了片上稳压器，用于为内核提供正确的电压。部件编号标有“F”的器件（例如 PIC18F46J50）都使能了稳压器。

这些器件可以依靠 V_{DD} 上的 2.15V-3.6V 电压工作，但 V_{DDCORE} 引脚应通过低 ESR 电容连接到 V_{SS}。部件编号标有“LF”的器件（例如 PIC18LF46J50）未使能稳压器。对于“LF”器件，必须为 V_{DDCORE} 引脚提供 2.0V-2.7V 的外部电源，而对 V_{DD} 可以提供 2.0V-3.6V 的电压（V_{DDCORE} 决不能超出 V_{DD}）。

关于内部稳压器的更多详细信息，请参见第 27.3 节“片上稳压器”。

PIC18F46J50 系列

表 1-1: PIC18F2XJ50 器件特性 (28 引脚器件)

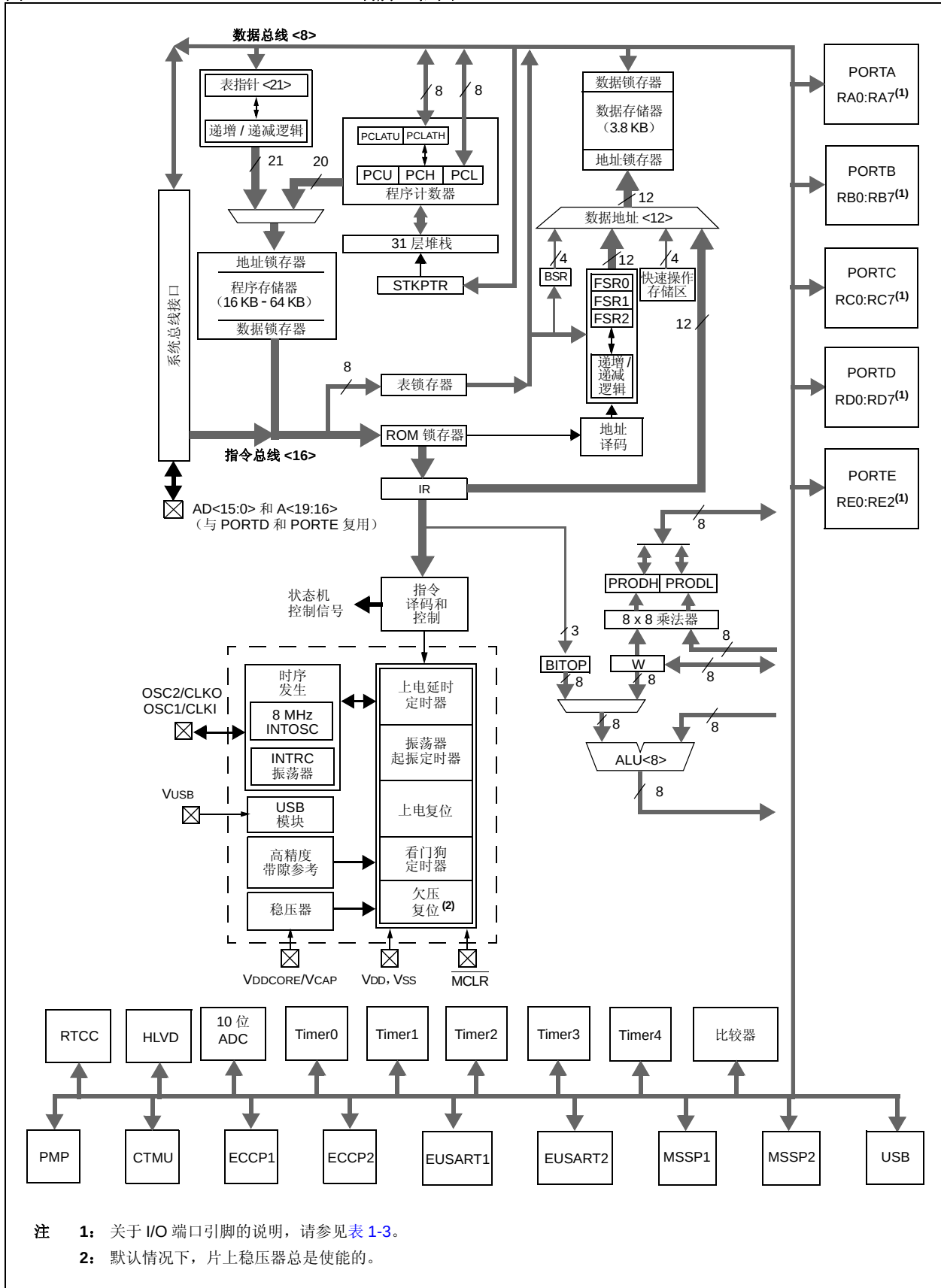
特性	PIC18F24J50	PIC18F25J50	PIC18F26J50
工作频率	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
程序存储器 (字节)	16K	32K	64K
程序存储器 (指令)	8,192	16,384	32,768
数据存储器 (字节)	3.8K	3.8K	3.8K
中断源	30		
I/O 端口	端口 A、B 和 C		
定时器	5		
增强型捕捉 / 比较 / PWM 模块	2		
串行通信	MSSP (2)、增强型 USART (2) 和 USB		
并行通信 (PMP/PSP)	无		
10 位模数转换模块	10 路输入通道		
复位 (和延时)	POR、BOR、RESET 指令、堆栈满、堆栈下溢、 $\overline{\text{MCLR}}$ 和 WDT (PWRT 和 OST)		
指令集	75 条指令, 使能扩展指令集后总共为 83 条指令		
封装	28 引脚 QFN、SOIC、SSOP 和 SPDIP (300 mil)		

表 1-2: PIC18F4XJ50 器件特性 (44 引脚器件)

特性	PIC18F44J50	PIC18F45J50	PIC18F46J50
工作频率	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
程序存储器 (字节)	16K	32K	64K
程序存储器 (指令)	8,192	16,384	32,768
数据存储器 (字节)	3.8K	3.8K	3.8K
中断源	30		
I/O 端口	端口 A、B、C、D 和 E		
定时器	5		
增强型捕捉 / 比较 / PWM 模块	2		
串行通信	MSSP (2)、增强型 USART (2) 和 USB		
并行通信 (PMP/PSP)	有		
10 位模数转换模块	13 路输入通道		
复位 (和延时)	POR、BOR、RESET 指令、堆栈满、堆栈下溢、 $\overline{\text{MCLR}}$ 和 WDT (PWRT 和 OST)		
指令集	75 条指令, 使能扩展指令集后总共为 83 条指令		
封装	44 引脚 QFN 和 TQFP		



图 1-2: PIC18F4XJ50 (44 引脚) 框图



PIC18F46J50 系列

表 1-3: PIC18F2XJ50 I/O 引脚说明

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	28-SPDIP/ SSOP/ SOIC	28-QFN			
MCLR	1	26	I	ST	主复位输入。此引脚为低电平有效的器件复位输入端。
OSC1/CLKI/RA7	9	6	I	ST	晶振或外部时钟输入。
OSC1					晶振输入或外部时钟源输入。配置为 RC 模式时为 ST 缓冲器输入；否则为 CMOS 缓冲器输入。主振荡器输入连接。
CLKI			I	CMOS	外部时钟源输入；总是与 OSC1 引脚功能相关联（见相关的 OSC1/CLKI 引脚信息）。
RA7 ⁽¹⁾			I/O	TTL	数字 I/O。
OSC2/CLKO/RA6	10	7	O	—	晶振或时钟输出。
OSC2			O	—	晶振输出。在晶振模式下，该引脚与晶振或谐振器相连。
CLKO			O	—	主振荡器反馈输出连接。
RA6 ⁽¹⁾			I/O	TTL	在 RC 模式下，OSC2 引脚输出 CLKO 信号，其频率是 OSC1 引脚上信号频率的 4 分频，该频率等于指令周期的倒数。
					数字 I/O。

图注:

TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

I = 输入

P = 电源

DIG = 数字输出

CMOS = CMOS 兼容输入或输出

Analog = 模拟输入

O = 输出

OD = 漏极开路（没有 P 型二极管连接到 VDD）

I²C™ = 漏极开路，特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能，则将禁止 RA7 和 RA6。

表 1-3: PIC18F2XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	28-SPDIP/ SSOP/ SOIC	28-QFN			
RA0/AN0/C1INA/ULPWU/RP0 RA0 AN0 C1INA ULPWU RP0	2	27	I/O I I I I/O	DIG Analog Analog Analog DIG	PORTA 是双向 I/O 端口。 数字 I/O。 模拟输入 0。 比较器 1 的输入 A。 超低功耗唤醒输入。 可重映射外设引脚 0 输入 / 输出。
RA1/AN1/C2INA/RP1 RA1 AN1 C2INA RP1	3	28	I O I I/O	DIG Analog Analog DIG	数字 I/O。 模拟输入 1。 比较器 2 的输入 A。 可重映射外设引脚 1 输入 / 输出。
RA2/AN2/VREF-/CVREF/C2INB RA2 AN2 VREF- CVREF C2INB	4	1	I/O I O I I	DIG Analog Analog Analog Analog	数字 I/O。 模拟输入 2。 A/D 参考电压 (低电压) 输入。 比较器参考电压输出。 比较器 2 的输入 B。
RA3/AN3/VREF+/C1INB RA3 AN3 VREF+ C1INB	5	2	I/O I I I	DIG Analog Analog Analog	数字 I/O。 模拟输入 3。 A/D 参考电压 (高电压) 输入。 比较器 1 的输入 B。
RA5/AN4/SS1/HLVDIN/ RCV/RP2 RA5 AN4 SS1 HLVDIN RCV RP2	7	4	I/O I I I I I/O	DIG Analog TTL Analog Analog DIG	数字 I/O。 模拟输入 4。 SPI 从选择输入。 低压检测 (LVD) 输入。 外部 USB 收发器 RCV 输入。 可重映射外设引脚 2 输入 / 输出。
RA6 ⁽¹⁾ RA7 ⁽¹⁾					见 OSC2/CLKO/RA6 引脚信息。 见 OSC1/CLKI/RA7 引脚信息。

图注: TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

I = 输入

P = 电源

DIG = 数字输出

CMOS = CMOS 兼容输入或输出

Analog = 模拟输入

O = 输出

OD = 漏极开路 (没有 P 型二极管连接到 VDD)

I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

PIC18F46J50 系列

表 1-3: PIC18F2XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	28-SPDIP/ SSOP/ SOIC	28-QFN			
RB0/AN12/INT0/RP3 RB0 AN12 INT0 RP3	21	18	I/O I I I/O	DIG Analog ST DIG	PORTB 是双向 I/O 端口。PORTB 的所有输入端都可软件编程为内部弱上拉。 数字 I/O。 模拟输入 12。 外部中断 0。 可重映射外设引脚 3 输入 / 输出。
RB1/AN10/RTCC/RP4 RB1 AN10 RTCC RP4	22	19	I/O I O I/O	DIG Analog DIG DIG	数字 I/O。 模拟输入 10。 实时时钟日历 (RTCC) 输出。 可重映射外设引脚 4 输入 / 输出。
RB2/AN8/CTED1/VMO/ REFO/RP5 RB2 AN8 CTED1 VMO REFO RP5	23	20	I/O I I O O I/O	DIG Analog ST DIG DIG DIG	数字 I/O。 模拟输入 8。 CTMU 边沿 1 输入。 外部 USB 收发器 D- 数据输出。 参考输出时钟。 可重映射外设引脚 5 输入 / 输出。
RB3/AN9/CTED2/VPO/RP6 RB3 AN9 CTED2 VPO RP6	24	21	I/O I I/O O I	DIG Analog ST DIG DIG	数字 I/O。 模拟输入 9。 CTMU 边沿 2 输入。 外部 USB 收发器 D+ 数据输出。 可重映射外设引脚 6 输入 / 输出。

图注:

TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
I = 输入
P = 电源
DIG = 数字输出

CMOS = CMOS 兼容输入或输出
Analog = 模拟输入
O = 输出
OD = 漏极开路 (没有 P 型二极管连接到 VDD)
I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

表 1-3: PIC18F2XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	28-SPDIP/ SSOP/ SOIC	28-QFN			
RB4/KBI0/SCK1/SCL1/RP7	25	22	I/O	DIG	PORTB (续) 数字 I/O。 电平变化中断引脚。 同步串行时钟输入 / 输出。 I ² C 时钟输入 / 输出。 可重映射外设引脚 7 输入 / 输出。 数字 I/O。 电平变化中断引脚。 SPI 数据输入。 I ² C™ 数据输入 / 输出。 可重映射外设引脚 8 输入 / 输出。 数字 I/O。 电平变化中断引脚。 ICSP™ 时钟输入。 可重映射外设引脚 9 输入 / 输出。 数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP 编程数据引脚。 可重映射外设引脚 10 输入 / 输出。
RB4			I	TTL	
KBI0			I/O	DIG	
SCK1			I/O	I ² C	
SCL1			I/O	DIG	
RP7			I/O	DIG	
RB5/KBI1/SDI1/SDA1/RP8	26	23	I/O	DIG	
RB5			I	TTL	
KBI1			I	ST	
SDI1			I/O	I ² C	
SDA1			I/O	DIG	
RP8			I/O	DIG	
RB6/KBI2/PGC/RP9	27	24	I/O	DIG	
RB6			I	TTL	
KBI2			I	ST	
PGC			I/O	DIG	
RP9			I/O	DIG	
RB7/KBI3/PGD/RP10	28	25	I/O	DIG	
RB7			I	TTL	
KBI3			I/O	ST	
PGD			I/O	DIG	
RP10			I/O	DIG	

图注: TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

I = 输入

P = 电源

DIG = 数字输出

CMOS = CMOS 兼容输入或输出

Analog = 模拟输入

O = 输出

OD = 漏极开路 (没有 P 型二极管连接到 VDD)

I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

PIC18F46J50 系列

表 1-3: PIC18F2XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	28-SPDIP/ SSOP/ SOIC	28-QFN			
RC0/T1OSO/T1CKI/RP11 RC0 T1OSO T1CKI RP11	11	8	I/O O I I/O	ST Analog ST DIG	PORTC 是双向 I/O 端口。 数字 I/O。 Timer1 振荡器输出。 Timer1 外部数字时钟输入。 可重映射外设引脚 11 输入 / 输出。
RC1/T1OSI/ <u>UOE</u> /RP12 RC1 T1OSI UOE RP12	12	9	I/O I O I/O	ST Analog DIG DIG	数字 I/O。 Timer1 振荡器输入。 外部 USB 收发器 NOE 输出。 可重映射外设引脚 12 输入 / 输出。
RC2/AN11/CTPLS/RP13 RC2 AN11 CTPLS RP13	13	10	I/O I O I/O	ST Analog DIG DIG	数字 I/O。 模拟输入 11。 CTMU 脉冲发生器输出。 可重映射外设引脚 13 输入 / 输出。
RC4/D-/VM RC4 D- VM	15	12	I I/O I	TTL — TTL	数字输入。 USB 总线负信号线输入 / 输出。 外部 USB 收发器 FM 输入。
RC5/D+/VP RC5 D+ VP	16	13	I I/O I	TTL DIG TTL	数字输入。 USB 总线正信号线输入 / 输出。 外部 USB 收发器 VP 输入。
RC6/TX1/CK1/RP17 RC6 TX1 CK1 RP17	17	14	I/O O I/O I/O	ST DIG ST DIG	数字 I/O。 EUSART1 异步发送。 EUSART1 同步时钟 (见相关的 RX1/DT1 引脚信息)。 可重映射外设引脚 17 输入 / 输出。
RC7/RX1/DT1/SDO1/RP18 RC7 RX1 DT1 SDO1 RP18	18	15	I/O I I/O O I/O	ST ST ST DIG DIG	数字 I/O。 异步串行接收数据输入。 同步串行数据输入 / 输出。 SPI 数据输出。 可重映射外设引脚 18 输入 / 输出。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
I = 输入
P = 电源
DIG = 数字输出

CMOS = CMOS 兼容输入或输出
Analog = 模拟输入
O = 输出
OD = 漏极开路 (没有 P 型二极管连接到 VDD)
I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

表 1-3: PIC18F2XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	28-SPDIP/ SSOP/ SOIC	28-QFN			
VSS1	8	5	P	—	逻辑和 I/O 引脚的参考地。
VSS2	19	16	—	—	
VDD	20	17	P	—	外设数字逻辑和 I/O 引脚的正电源。
VDDCORE/VCAP	6	3	—	—	内核逻辑电源或外部滤波电容连接。
VDDCORE			P	—	单片机内核逻辑的正电源 (禁止稳压器)。
VCAP			P	—	外部滤波电容连接 (使能稳压器)。
VUSB	14	11	P	—	USB 电压输入引脚。

图注:

TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

I = 输入

P = 电源

DIG = 数字输出

CMOS = CMOS 兼容输入或输出

Analog = 模拟输入

O = 输出

OD = 漏极开路 (没有 P 型二极管连接到 VDD)

I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

PIC18F46J50 系列

表 1-4: PIC18F4XJ50 I/O 引脚说明

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	44-QFN	44-TQFP			
MCLR	18	18	I	ST	主复位输入；此引脚为低电平有效的器件复位输入端。 晶振或外部时钟输入。 晶振输入或外部时钟源输入。配置为 RC 模式时为 ST 缓冲器输入；否则为 CMOS 缓冲器输入。主振荡器输入连接。 外部时钟源输入；总是与 OSC1 引脚功能相关（见相关的 OSC1/CLKI 引脚信息）。 数字 I/O。 晶振或时钟输出。 晶振输出。在晶振模式下，该引脚与晶振或谐振器相连。 主振荡器反馈输出连接。 在 RC 模式下，OSC2 引脚输出 CLKO 信号，其频率是 OSC1 引脚上信号频率的 4 分频，该频率等于指令周期的倒数。 数字 I/O。
OSC1/CLKI/RA7 OSC1	32	30	I	ST	
CLKI			I	CMOS	
RA7 ⁽¹⁾			I/O	TTL	
OSC2/CLKO/RA6 OSC2	33	31	O	—	
CLKO			O	—	
RA6 ⁽¹⁾			I/O	TTL	

图注:

TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

I = 输入

P = 电源

DIG = 数字输出

CMOS = CMOS 兼容输入或输出

Analog = 模拟输入

O = 输出

OD = 漏极开路（没有 P 型二极管连接到 VDD）

I²C™ = 漏极开路，特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能，则将禁止 RA7 和 RA6。

表 1-4: PIC18F4XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	44-QFN	44-TQFP			
RA0/AN0/C1INA/ULPWU/PMA6/RP0	19	19	I/O I I I O I/O	DIG Analog Analog Analog DIG DIG	PORTA 是双向 I/O 端口。 数字 I/O。 模拟输入 0。 比较器 1 的输入 A。 超低功耗唤醒输入。 并行主端口数字输出。 可重映射外设引脚 0 输入 / 输出。
RA1/AN1/C2INA/PMA7/RP1	20	20	I O I O I/O	DIG Analog Analog DIG DIG	数字 I/O。 模拟输入 1。 比较器 2 的输入 A。 并行主端口数字输出。 可重映射外设引脚 1 输入 / 输出。
RA2/AN2/VREF-/CVREF/C2INB	21	21	I/O I O I I	DIG Analog Analog Analog Analog	数字 I/O。 模拟输入 2。 A/D 参考电压 (低电压) 输入。 比较器参考电压输出。 比较器 2 的输入 B。
RA3/AN3/VREF+/C1INB	22	22	I/O I I I	DIG Analog Analog Analog	数字 I/O。 模拟输入 3。 A/D 参考电压 (高电压) 输入。 比较器 1 的输入 B。
RA5/AN4/SS1/HLVDIN/RCV/RP2	24	24	I/O I I I I I/O	DIG Analog TTL Analog Analog DIG	数字 I/O。 模拟输入 4。 SPI 从选择输入。 低压检测 (LVD) 输入。 外部 USB 收发器 RCV 输入。 可重映射外设引脚 2 输入 / 输出。
RA6 ⁽¹⁾ RA7 ⁽¹⁾					见 OSC2/CLKO/RA6 引脚信息。 见 OSC1/CLKI/RA7 引脚信息。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
I = 输入
P = 电源
DIG = 数字输出

CMOS = CMOS 兼容输入或输出
Analog = 模拟输入
O = 输出
OD = 漏极开路 (没有 P 型二极管连接到 VDD)
I²CTM = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

PIC18F46J50 系列

表 1-4: PIC18F4XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	44-QFN	44-TQFP			
RB0/AN12/INT0/RP3 RB0 AN12 INT0 RP3	9	8	I/O I I I/O	DIG Analog ST DIG	PORTB 是双向 I/O 端口。PORTB 的所有输入端都可软件编程为内部弱上拉。 数字 I/O。 模拟输入 12。 外部中断 0。 可重映射外设引脚 3 输入 / 输出。
RB1/AN10/PMBE/RTCC/RP4 RB1 AN10 PMBE RTCC RP4	10	9	I/O I O O I/O	DIG Analog DIG DIG DIG	数字 I/O。 模拟输入 10。 并行主端口字节使能。 实时时钟日历 (RTCC) 输出。 可重映射外设引脚 4 输入 / 输出。
RB2/AN8/CTED1/PMA3/VMO/REFO/RP5 RB2 AN8 CTED1 PMA3 VMO REFO RP5	11	10	I/O I I O O O I/O	DIG Analog ST DIG DIG DIG DIG	数字 I/O。 模拟输入 8。 CTMU 边沿 1 输入。 并行主端口地址。 外部 USB 收发器 D- 数据输出。 参考输出时钟。 可重映射外设引脚 5 输入 / 输出。
RB3/AN9/CTED2/PMA2/VPO/RP6 RB3 AN9 CTED2 PMA2 VPO RP6	12	11	I/O I I O O I/O	DIG Analog ST DIG DIG DIG	数字 I/O。 模拟输入 9。 CTMU 边沿 2 输入。 并行主端口地址。 外部 USB 收发器 D+ 数据输出。 可重映射外设引脚 6 输入 / 输出。

图注:

TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
I = 输入
P = 电源
DIG = 数字输出

CMOS = CMOS 兼容输入或输出
Analog = 模拟输入
O = 输出
OD = 漏极开路 (没有 P 型二极管连接到 VDD)
I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

表 1-4: PIC18F4XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	44-QFN	44-TQFP			
PORTB (续)					
RB4/PMA1/KBI0/SCK1/SCL1/RP7	14	14	I/O	DIG	数字 I/O。
RB4			I/O	DIG	并行主端口地址。
PMA1			I	TTL	电平变化中断引脚。
KBI0			I/O	DIG	同步串行时钟输入 / 输出。
SCK1			I/O	I ² C	I ² C™ 时钟输入 / 输出。
SCL1			I/O	DIG	可重映射外设引脚 7 输入 / 输出。
RP7					
RB5/PMA0/KBI1/SDI1/SDA1/RP8	15	15	I/O	DIG	数字 I/O。
RB5			I/O	DIG	并行主端口地址。
PMA0			I	TTL	电平变化中断引脚。
KBI1			I	ST	SPI 数据输入。
SDI1			I/O	I ² C	I ² C™ 数据输入 / 输出。
SDA1			I/O	DIG	可重映射外设引脚 8 输入 / 输出。
RP8					
RB6/KBI2/PGC/RP9	16	16	I/O	DIG	数字 I/O。
RB6			I	TTL	电平变化中断引脚。
KBI2			I	ST	ICSP™ 时钟输入。
PGC			I/O	DIG	可重映射外设引脚 9 输入 / 输出。
RP9					
RB7/KBI3/PGD/RP10	17	17	I/O	DIG	数字 I/O。
RB7			I	TTL	电平变化中断引脚。
KBI3			I/O	ST	在线调试器和 ICSP 编程数据引脚。
PGD			I/O	DIG	可重映射外设引脚 10 输入 / 输出。
RP10					

图注: TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

I = 输入

P = 电源

DIG = 数字输出

CMOS = CMOS 兼容输入或输出

Analog = 模拟输入

O = 输出

OD = 漏极开路 (没有 P 型二极管连接到 VDD)

I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

PIC18F46J50 系列

表 1-4: PIC18F4XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	44-QFN	44-TQFP			
RC0/T1OSO/T1CKI/RP11 RC0 T1OSO T1CKI RP11	34	32	I/O O I I/O	ST Analog ST DIG	PORTC 是双向 I/O 端口。 数字 I/O。 Timer1 振荡器输出。 Timer1/Timer3 外部时钟输入。 可重映射外设引脚 11 输入 / 输出。
RC1/T1OSI/ $\overline{\text{UOE}}$ /RP12 RC1 T1OSI $\overline{\text{UOE}}$ RP12	35	35	I/O I O I/O	ST Analog DIG DIG	数字 I/O。 Timer1 振荡器输入。 外部 USB 收发器 NOE 输出。 可重映射外设引脚 12 输入 / 输出。
RC2/AN11/CTPLS/RP13 RC2 AN11 CTPLS RP13	36	36	I/O I O I/O	ST Analog DIG DIG	数字 I/O。 模拟输入 11。 CTMU 脉冲发生器输出。 可重映射外设引脚 13 输入 / 输出。
RC4/D-/VM RC4 D- VM	42	42	I O I	TTL — TTL	数字输入。 USB 总线负信号线输入 / 输出。 外部 USB 收发器 FM 输入。
RC5/D+/VP RC5 D+ VP	43	43	I I/O I	TTL DIG TTL	数字输入。 USB 总线正信号线输入 / 输出。 外部 USB 收发器 VP 输入。
RC6/PMA5/TX1/CK1/RP17 RC6 PMA5 TX1 CK1 RP17	44	44	I/O O O I/O I/O	ST DIG DIG ST DIG	数字 I/O。 并行主端口地址。 EUSART1 异步发送。 EUSART1 同步时钟 (见相关的 RX1/DT1 引脚信息)。 可重映射外设引脚 17 输入 / 输出。
RC7/PMA4/RX1/DT1/SDO1/RP18 RC7 PMA4 RX1 DT1 SDO1 RP18	1	1	I/O O I I/O O I/O	ST DIG ST ST DIG DIG	数字 I/O。 并行主端口地址。 EUSART1 异步接收。 EUSART1 同步数据输入 / 输出 (见相关的 TX1/CK1 引脚信息)。 SPI 数据输出。 可重映射外设引脚 18 输入 / 输出。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
I = 输入
P = 电源
DIG = 数字输出
CMOS = CMOS 兼容输入或输出
Analog = 模拟输入
O = 输出
OD = 漏极开路 (没有 P 型二极管连接到 VDD)
I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

表 1-4: PIC18F4XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	44-QFN	44-TQFP			
RD0/PMD0/SCL2 RD0 PMD0 SCL2	38	38	I/O I/O I/O	ST DIG DIG	PORTD 是双向 I/O 端口。 数字 I/O。 并行主端口数据。 I ² C™ 数据输入 / 输出。
RD1/PMD1/SDA2 RD1 PMD1 SDA2	39	39	I/O I/O I/O	ST DIG DIG	数字 I/O。 并行主端口数据。 I ² C 数据输入 / 输出。
RD2/PMD2/RP19 RD2 PMD2 RP19	40	40	I/O I/O I/O	ST DIG DIG	数字 I/O。 并行主端口数据。 可重映射外设引脚 19 输入 / 输出。
RD3/PMD3/RP20 RD3 PMD3 RP20	41	41	I/O I/O I/O	ST DIG DIG	数字 I/O。 并行主端口数据。 可重映射外设引脚 20 输入 / 输出。
RD4/PMD4/RP21 RD4 PMD4 RP21	2	2	I/O I/O I/O	ST DIG DIG	数字 I/O。 并行主端口数据。 可重映射外设引脚 21 输入 / 输出。
RD5/PMD5/RP22 RD5 PMD5 RP22	3	3	I/O I/O I/O	ST DIG DIG	数字 I/O。 并行主端口数据。 可重映射外设引脚 22 输入 / 输出。
RD6/PMD6/RP23 RD6 PMD6 RP23	4	4	I/O I/O I/O	ST DIG DIG	数字 I/O。 并行主端口数据。 可重映射外设引脚 23 输入 / 输出。
RD7/PMD7/RP24 RD7 PMD7 RP24	5	5	I/O I/O I/O	ST DIG DIG	数字 I/O。 并行主端口数据。 可重映射外设引脚 24 输入 / 输出。

图注: TTL = TTL 兼容输入

ST = CMOS 电平的施密特触发器输入

I = 输入

P = 电源

DIG = 数字输出

CMOS = CMOS 兼容输入或输出

Analog = 模拟输入

O = 输出

OD = 漏极开路 (没有 P 型二极管连接到 VDD)

I²C™ = 漏极开路, 特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能, 则将禁止 RA7 和 RA6。

PIC18F46J50 系列

表 1-4: PIC18F4XJ50 I/O 引脚说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	说明
	44-QFN	44-TQFP			
RE0/AN5/PMRD RE0 AN5 PMRD	25	25	I/O I I/O	ST Analog DIG	PORTC 是双向 I/O 端口。 数字 I/O。 模拟输入 5。 并行主端口输入 / 输出。
RE1/AN6/PMWR RE1 AN6 PMWR	26	26	I/O I I/O	ST Analog DIG	数字 I/O。 模拟输入 6。 并行主端口写选通。
RE2/AN7/PMCS RE2 AN7 PMCS	27	27	I/O I O	ST Analog —	数字 I/O。 模拟输入 7。 并行主端口片选。
Vss1	6	6	P	—	逻辑和 I/O 引脚的参考地。
Vss2	31	29	—	—	
AVss1	30	—	P	—	模拟模块的参考地。
VDD1	8	7	P	—	外设数字逻辑和 I/O 引脚的正电源。
VDD2	29	28	P	—	
VDDCORE/VCAP VDDCORE VCAP	23	23	P P	— —	内核逻辑电源或外部滤波电容连接。 单片机内核逻辑的正电源（禁止稳压器）。 外部滤波电容连接（使能稳压器）。
AVDD1	7	—	P	—	模拟模块的正电源。
AVDD2	28	—	—	—	模拟模块的正电源。
VUSB	37	37	P	—	USB 电压输入引脚。

图注:

TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
I = 输入
P = 电源
DIG = 数字输出

CMOS = CMOS 兼容输入或输出
Analog = 模拟输入
O = 输出
OD = 漏极开路（没有 P 型二极管连接到 VDD）
I²C™ = 漏极开路，特定于 I²C

注 1: 如果 OSC1 和 OSC2 用于时钟功能，则将禁止 RA7 和 RA6。

2.0 PIC18FJ 单片机入门指南

2.1 基本连接要求

在开始使用 PIC18F46J50 系列 8 位单片机进行开发之前，需要注意最基本的器件引脚连接要求。

必须始终连接以下引脚：

- 所有 VDD 和 VSS 引脚（见第 2.2 节“电源引脚”）
- 所有 AVDD 和 AVSS 引脚（如果存在）（不论是否使用模拟器件功能）（见第 2.2 节“电源引脚”）
- MCLR 引脚（见第 2.3 节“主复位（MCLR）引脚”）
- VCAP/VDDCORE 引脚（见第 2.4 节“稳压器引脚（VCAP/VDDCORE）”）

如果在最终应用中使用了以下引脚，则也必须连接它们：

- PGC/PGD 引脚，用于进行在线串行编程（ICSP™）和调试（见第 2.5 节“ICSP 引脚”）
- OSCI 和 OSCO 引脚（使用外部振荡器源时）（见第 2.6 节“外部振荡器引脚”）

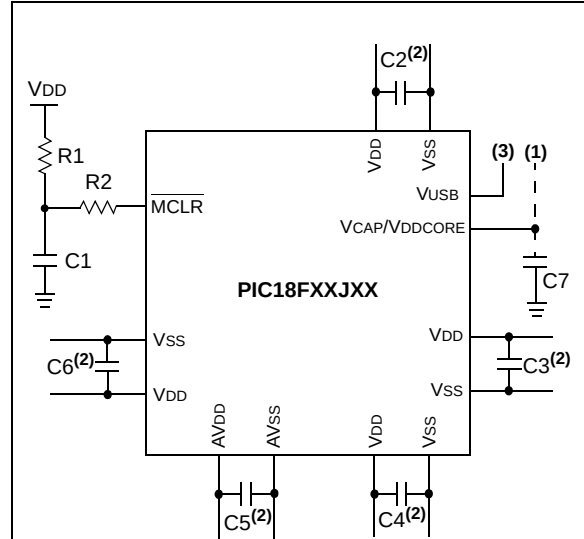
此外，可能还需要连接以下引脚：

- VREF+/VREF- 引脚（在实现模拟模块的外部参考电压时使用）

注： 不论是否使用任何模拟模块，都必须始终连接 AVDD 和 AVSS 引脚。

图 2-1 中显示了最基本的连接要求。

图 2-1: 建议的最基本连接



关键（所有值均为建议值）：

C1 至 C6: 0.1 μ F, 20V 陶瓷电容

C7: 10 μ F, 6.3V 或更高电压的钽电容，或者 10V 或更高电压的陶瓷电容

R1: 10 k Ω

R2: 100 Ω 至 470 Ω

- 注**
- 1: 关于 VCAP/VDDCORE 引脚连接的说明，请参见第 2.4 节“稳压器引脚（VCAP/VDDCORE）”。
 - 2: 所给出的示例针对的是具有 5 个 VDD/VSS 和 AVDD/AVSS 引脚对的 PIC18F 器件。其他器件的引脚对可能增多或减少，请相应地调整去耦电容的数量。
 - 3: 请参见第 22.2.2.1 节“内部收发器”。

PIC18F46J50 系列

2.2 电源引脚

2.2.1 去耦电容

需要在每对电源引脚（例如，VDD/VSS 和 AVDD/AVSS）上使用去耦电容。

使用去耦电容时，需要考虑以下标准：

- **电容的类型和电容值：**建议使用参数为 $0.1\ \mu\text{F}$ （ $100\ \text{nF}$ ）、 $10\text{-}20\text{V}$ 的电容。该电容应具有低 ESR，谐振频率为 $200\ \text{MHz}$ 或更高。建议使用陶瓷电容。
- **在印制电路板上的放置：**去耦电容应尽可能靠近引脚。建议将电容与器件放置在电路板的同一层。如果空间受到限制，可以使用过孔将电容放置在 PCB 的另一层，但请确保从引脚到电容的走线长度不超出 0.25 英寸（ 6 毫米）。
- **高频噪声处理：**如果电路板遇到高频噪声（频率高于数十 MHz ），则另外添加一个陶瓷电容，与上述去耦电容并联。第二个电容的电容值可以介于 $0.001\ \mu\text{F}$ 至 $0.01\ \mu\text{F}$ 之间。请将第二个电容放置在靠近每个主去耦电容的位置。在高速电路设计中，需要考虑尽可能靠近电源和接地引脚放置一对这样的电容（例如， $0.1\ \mu\text{F}$ 电容与 $0.001\ \mu\text{F}$ 电容并联构成一对）。
- **最大程度提高性能：**对于从电源电路开始的电路板布线，需要将电源和返回走线先连接到去耦电容，然后再与器件引脚连接。这可以确保去耦电容是电源链中的第一个元件。同等重要的是尽可能减小电容和电源引脚之间的走线长度，从而降低 PCB 走线电感。

2.2.2 槽路电容

对于电源走线长度超出 6 英寸的电路板，建议对集成电路（包括单片机）使用容量较大的储能电容来提供本地电源。该电容的电容值应根据连接电源与器件的走线电阻和应用中的器件的最大电流确定。也就是说，选择的电容需要满足器件的可接受电压骤降要求。典型值的范围为 $4.7\ \mu\text{F}$ 至 $47\ \mu\text{F}$ 。

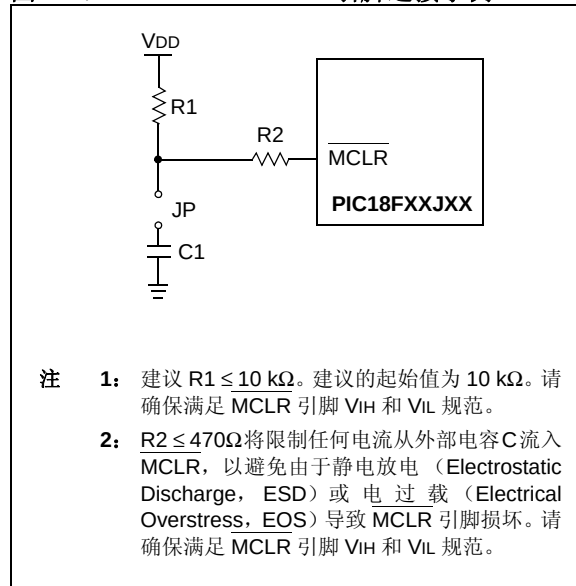
2.3 主复位（MCLR）引脚

MCLR 引脚提供两种特定的器件功能：器件复位，以及器件编程和调试。如果最终应用中不需要进行编程和调试，则只需直接连接 VDD 即可。添加其他元件有助于提高应用抵抗由于电压骤降导致意外复位的能力。图 2-1 给出了一种典型配置。根据应用的需求，还可以实现其他电路设计。

在编程和调试过程中，必须考虑到引脚上可能会增加的电阻和电容。器件编程器和调试器会驱动 MCLR 引脚。因此，特定电平（ V_{IH} 和 V_{IL} ）和快速信号跳变一定不能受到不利影响。所以，需要根据应用和 PCB 需求来调整 R1 和 C1 的具体值。例如，在编程和调试操作期间，建议通过使用跳线将电容 C1 与 MCLR 引脚隔离（图 2-2）。对于正常的运行时操作，可以将跳线放回原处。

与 MCLR 引脚关联的所有元件都应放置在距离该引脚 0.25 英寸（ 6 毫米）的范围内。

图 2-2: MCLR 引脚连接示例



2.4 稳压器引脚（VCAP/VDDCORE）

在“F”器件上，需要在 VCAP/VDDCORE 引脚上使用低 ESR (< 5Ω) 电容，以稳定稳压器的输出电压。在“F”器件上，VCAP/VDDCORE 引脚一定不能与 VDD 或任何其他电压源连接。VCAP/VDDCORE 引脚只能通过一个 10 μF 的电容接地。可以使用陶瓷电容或钽电容。表 2-1 列出了一些适用电容的示例。

设计人员可以根据图 2-3 来评估候选器件的 ESR 等效值。建议走线长度不要超出 0.25 英寸（6 毫米）。更多信息，请参见第 30.0 节“电气特性”。

在“LF”器件上，内部内核稳压器被禁止。在这些器件上，VCAP/VDDCORE 引脚必须与外部电路板上的适用 VDDCORE 电压源连接。关于允许的 VDDCORE 电压范围，请参见第 30.0 节“电气特性”。对于提供 VCAP/VDDCORE 电压的电源，应采取良好的电源旁路措施。

建议在 VCAP/VDDCORE 与地之间使用一个 0.1 μF 的陶瓷电容，并使其尽量靠近 VCAP/VDDCORE 和 VSS 引脚。

图 2-3: 建议的 VCAP 的频率与 ESR 性能

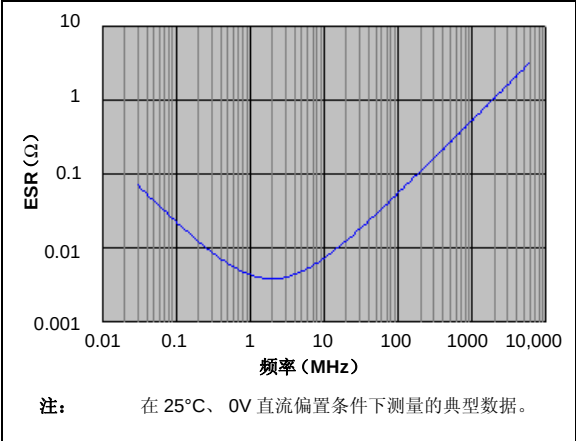


表 2-1: 适用的等效电容

制造商	部件编号	标称电容	基本公差	额定电压	温度范围
TDK	C3216X7R1C106K	10 μF	±10%	16V	-55 至 +125°C
TDK	C3216X5R1C106K	10 μF	±10%	16V	-55 至 +85°C
Panasonic	ECJ-3YX1C106K	10 μF	±10%	16V	-55 至 +125°C
Panasonic	ECJ-4YB1C106K	10 μF	±10%	16V	-55 至 +85°C
Murata	GRM32DR71C106KA01L	10 μF	±10%	16V	-55 至 +125°C
Murata	GRM31CR61C106KC31L	10 μF	±10%	16V	-55 至 +85°C

PIC18F46J50 系列

2.4.1 陶瓷电容的注意事项

近年来，大容量低电压表面贴装陶瓷电容已成为一种极具成本效益的电容，其容量最高可达数十微法。低 ESR、小物理尺寸和一些其他特性使陶瓷电容对于许多应用类型都极具吸引力。

陶瓷电容适用于此类单片机的 VDDCORE 稳压器。但是，在选择电容时需要谨慎，以确保它可以在应用的期望工作范围内维持足够的电容。

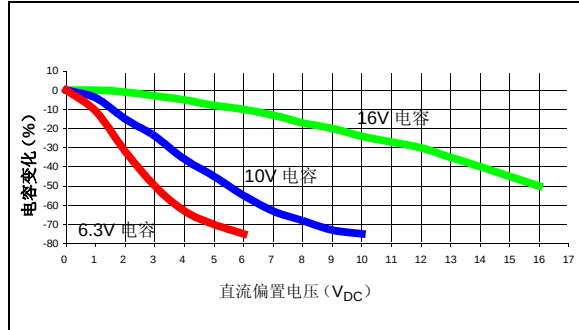
典型的低成本 10 μ F 陶瓷电容提供了 X5R、X7R 和 Y5V 等介电规格（其他类型也有，但较不常见）。这些电容类型的初始公差规范通常规定为 $\pm 10\%$ 至 $\pm 20\%$ （X5R 和 X7R）或 $-20\%/+80\%$ （Y5V）。但是，这些电容在应用电路中产生的有效电容还会因为其他一些因素而变化，例如所施加的直流偏置电压和温度。因此，整个电路的总误差会远高于初始公差规范。

X5R 和 X7R 电容通常具有良好的温度稳定性（例如：误差在很宽的温度范围为 $\pm 15\%$ ，欲知具体参数，请参见制造商的数据手册）。但 Y5V 在温度变化时的电容变化范围通常很糟糕，为 $+22\%/ -82\%$ 。因此，一个标称值为 10 μ F 的 Y5V 型电容可能无法提供足够的总电容来满足 VDDCORE 稳压器稳定性和瞬态响应的最低要求。因此，如果应用必须在很宽的温度范围内工作，则建议不要对 VDDCORE 稳压器使用 Y5V 电容。

除了温度的影响之外，根据电容上所施加直流电压的不同，大容量陶瓷电容的有效电容也会显著变化。这种影响可能会非常显著，但常常被忽视或并不总是记入文档。

图 2-4 给出了 X7R 型和 Y5V 型电容的典型直流偏置电压 — 电容关系图。

图 2-4: 直流偏置电压 — 电容特性



在选择用于 VDDCORE 稳压器的陶瓷电容时，建议选择高额定电压的电容，以使工作电压仅为电容最高额定电压的一小部分。例如，对于 2.5V 的 VDDCORE 电压，选择额定电压为 16V 的陶瓷电容。表 2-1 列出了所建议的电容。

2.5 ICSP 引脚

PGC 和 PGD 引脚用于进行在线串行编程（ICSP™）和调试。建议尽可能减小 ICSP 连接器与器件 ICSP 引脚之间的走线长度。如果 ICSP 连接器会遇到 ESD 事件，则建议添加一个串联电阻，电阻值为几十欧姆，不要超出 100 Ω 。

建议不要在 PGC 和 PGD 引脚上连接上拉电阻、串联二极管和电容，因为它们会影响与器件的编程器 / 调试器通信。如果应用需要此类分立元件，则在编程和调试期间应将它们从电路中去掉。或者，请参见相应器件闪存编程规范中的交流 / 直流特性与时序要求信息，了解关于容性负载限制、引脚输入高电压（V_{IH}）和输入低电压（V_{IL}）要求的信息。

对于器件仿真，请确保烧写至器件中的软件所选择的通信通道（即 PGCx/PGDx 引脚）符合 ICSP 到 Microchip 调试器 / 仿真器工具的物理连接。

关于可用的 Microchip 开发工具连接要求的更多信息，请参见第 29.0 节“开发支持”。

2.6 外部振荡器引脚

许多单片机都有至少两个振荡器可供选择：高频主振荡器和低频辅助振荡器（详情请参见第 3.0 节“振荡器配置”）。

振荡器电路与器件应放置在电路板的同一层。请将振荡器电路放置在靠近相应振荡器引脚的位置，电路元件与引脚之间的距离不要超出 0.5 英寸（12 毫米）。负载电容应靠近振荡器自身，位于电路板的同一层。

请在振荡器电路周围使用接地灌铜区，以将其与周围电路隔离。接地灌铜区应与 MCU 地直接连接。不要在接地灌铜区内安排任何信号走线或电源走线。此外，如果使用双面电路板，请避免在电路板上晶振所在位置的背面有任何走线。

图 2-5 给出了一些布线建议。直插式封装可以采用可完全容纳振荡器引脚的单面布线来处理。对于引脚排列紧密的器件，单面布局则可能无法始终完全地容纳所有引脚和元件。一种适合的解决方案是将含有保护走线的部分连接到反面的接地层。在所有情形中，保护走线都必须回接到地。

在规划应用的走线和 I/O 分配时，需要确保相邻端口引脚和其他邻近振荡器的信号是无害的（即无高频，无短暂上升和下降，以及无其他类似噪声）。

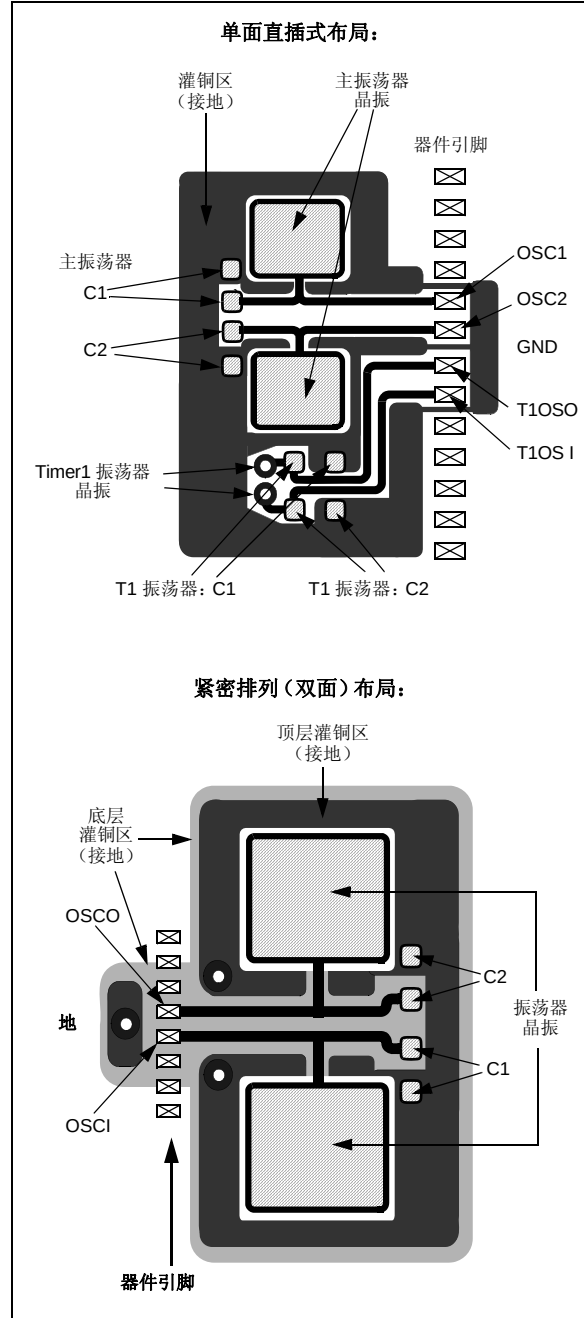
关于振荡器电路的更多信息和设计指南，请参见 Microchip 公司网站（www.microchip.com）上提供的以下应用笔记：

- AN826, “Crystal Oscillator Basics and Crystal Selection for *rfPIC™* and *PICmicro®* Devices”
- AN849, “Basic *PIC®* Oscillator Design”
- AN943, “Practical *PIC®* Oscillator Analysis and Design”
- AN949, “Making Your Oscillator Work”

2.7 未用 I/O

未用 I/O 引脚应配置为输出，并驱动为逻辑低电平状态。或者，将未用引脚通过一个 1 kΩ 至 10 kΩ 的电阻与 V_{SS} 连接，并将输出驱动为逻辑低电平。

图 2-5: 振荡器电路的建议布线方式



PIC18F46J50 系列

注:

3.0 振荡器配置

3.1 概述

PIC18F46J50 系列器件包含不同于通用 PIC18F 器件的振荡器和单片机时钟系统。此外 USB 模块有独特的稳定时钟源要求，因此必须提供同时符合 USB 低速和全速规范的独立时钟源。

PIC18F46J50 系列具有附加的预分频器和后分频器，增加它们是为了提供一系列广泛的振荡器频率。图 3-1 给出了振荡器结构的概图。

PIC18 增强型单片机中使用的其他振荡器特性（例如内部振荡器模块和时钟切换）仍保持不变。它们将在本章中稍后讨论。

3.1.1 振荡器控制

PIC18F46J50 系列器件中振荡器的工作通过 3 个配置寄存器和 2 个控制寄存器控制。配置寄存器 CONFIG1L、CONFIG1H 和 CONFIG2L 用于选择振荡器模式、PLL 预分频比和 CPU 分频比选项。作为配置位，它们在器件编程时被置 1，并保留该配置直到对器件重新编程。

OSCCON 寄存器（寄存器 3-2）选择有效时钟模式；它主要用于控制功耗管理模式下的时钟切换。它的使用在第 3.5.1 节“振荡器控制寄存器”中讨论。

OSCTUNE 寄存器（寄存器 3-1）用于微调 INTOSC 频率源，以及选择用于驱动几个特殊功能部件的低频时钟源。OSCTUNE 寄存器还可用于激活或禁止锁相环（PLL）。它的使用如第 3.2.5.1 节“OSCTUNE 寄存器”中所述。

3.2 振荡器类型

PIC18F46J50 系列器件可以在 8 种不同的振荡器模式下工作。用户可以通过设定 FOSC<2:0> 配置位来选择表 3-1 中列出的一种模式。对于会在引脚 RA6 上产生时钟输出（CLKO）的振荡器模式，输出频率将为外设时钟频率的四分之一。在处于休眠模式时，时钟输出会停止，但在空闲模式期间，时钟输出会继续（见图 3-1）。

表 3-1: 振荡器模式

模式	说明
ECPLL	外部时钟输入模式，可以用软件使能或禁止 PLL，RA6 上产生 CLKO，对 RA7 施加外部时钟信号。
EC	外部时钟输入模式，始终禁止 PLL，RA6 上产生 CLKO，对 RA7 施加外部时钟信号。
HSPLL	高速晶振 / 谐振器模式，可以用软件使能或禁止 PLL，晶振 / 谐振器连接在 RA6 和 RA7 之间。
HS	高速晶振 / 谐振器模式，始终禁止 PLL，晶振 / 谐振器连接在 RA6 和 RA7 之间。
INTOSCPLLO	内部振荡器模式，可以用软件使能或禁止 PLL，RA6 上产生 CLKO，RA7 用于端口功能，内部振荡器模块用于产生主时钟源和经过后分频的内部时钟。
INTOSCPLL	内部振荡器模式，可以用软件使能或禁止 PLL，RA6 和 RA7 用于端口功能，内部振荡器模块用于产生主时钟源和经过后分频的内部时钟。
INTOSCO	内部振荡器模式，始终禁止 PLL，RA6 上产生 CLKO，RA7 用于端口功能，INTOSC 后分频器的输出同时用作经过后分频的内部时钟和主时钟源。
INTOSC	内部振荡器模式，始终禁止 PLL，RA6 和 RA7 用于端口功能，INTOSC 后分频器的输出同时用作经过后分频的内部时钟和主时钟源。

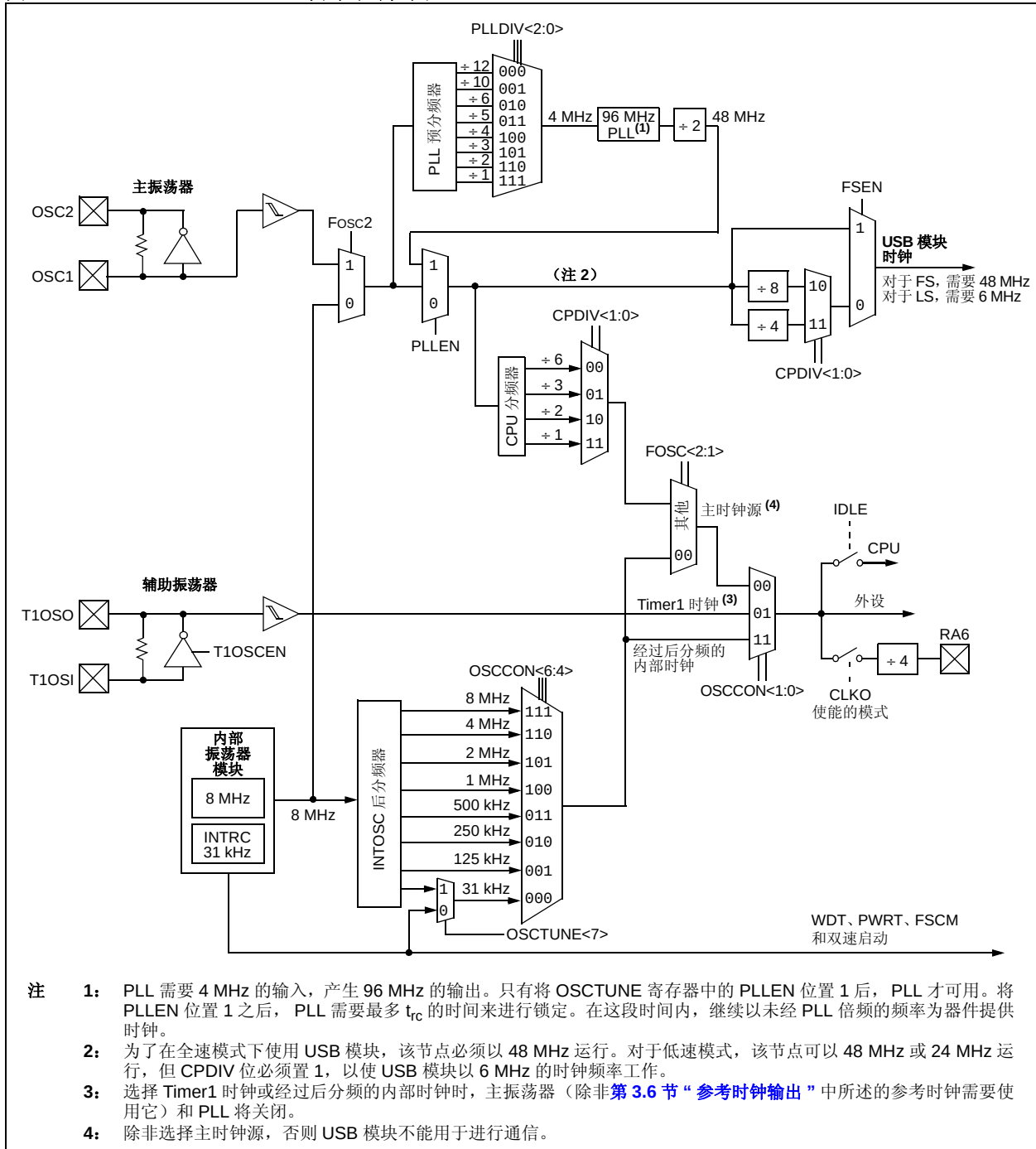
PIC18F46J50 系列

3.2.1 振荡器模式和 USB 操作

由于 USB 模块的独特要求，需要不同的时钟工作方式。为了使用 USB 模块，必须在内部为 USB 模块提供固定的 6 MHz 或 48 MHz 时钟，以分别工作于低速或全速模式。单片机内核的时钟频率不需要与 USB 模块相同。

器件提供了由 MUX、时钟分频器和固定 96 MHz 输出 PLL 组成的网络，可用于产生各种单片机内核和 USB 模块频率。图 3-1 可以帮助您了解 PIC18F46J50 系列器件的振荡器结构。

图 3-1: PIC18F46J50 系列时钟框图



3.2.2 晶振 / 陶瓷谐振器

在 HS 和 HSPLL 振荡器模式下，将晶振或陶瓷谐振器连接在 OSC1 和 OSC2 引脚之间来产生振荡信号。图 3-2 显示了引脚连接方式。

振荡器的设计要求使用并联谐振晶体。

注： 使用串联谐振晶体，可能会使振荡器产生的频率超出晶体制造厂商所给的规范。

图 3-2: 晶振 / 陶瓷谐振器工作原理 (HS 或 HSPLL 配置)

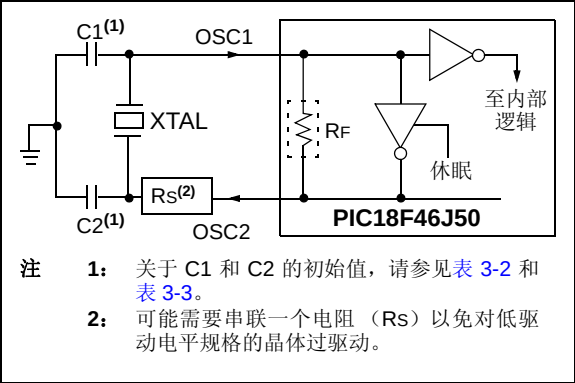


表 3-2: 陶瓷谐振器的电容选择

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。

已使用下面列出的谐振器在基本的启动和操作过程中对这些电容值作了测试。这些值未经优化。

要达到理想的振荡器工作状态，可能需要不同的电容值。用户应在应用要求的 VDD 和温度范围下测试振荡器的性能。

更多信息，请参见表 3-3 下方的“注”。

使用的谐振器:	
4.0 MHz	
8.0 MHz	
16.0 MHz	

表 3-3: 晶振的电容选择

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
	16 MHz	18 pF	18 pF

上述电容值仅供设计参考。

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

要达到理想的振荡器工作状态，可能需要不同的电容值。用户应在应用要求的 VDD 和温度范围下测试振荡器的性能。

更多信息，请参见本表下方的“注”。

使用的晶振:	
4 MHz	
8 MHz	
16 MHz	

注 1: 电容值越大，振荡器的稳定性越高，但同时起振时间也越长。

注 2: 因为每种谐振器 / 晶振都有其自身特性，用户应当向谐振器 / 晶振制造厂商询问外部元件的适当值。

注 3: 可能需要使用电阻 Rs 以避免对低驱动规格的晶体造成过驱动。

注 4: 应始终验证振荡器在应用要求的 VDD 和温度范围下的性能。

内部后分频器允许用户选择不同于晶振或谐振器频率的时钟频率。分频比是由 CPDIV 配置位决定的。用户可以选择振荡器频率的时钟频率，或该频率的 1/2、1/3 或 1/6。

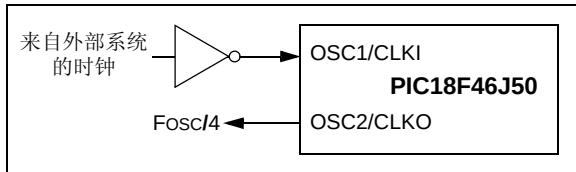
PIC18F46J50 系列

3.2.3 外部时钟输入

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

在 EC 振荡器模式下, 由 OSC2 引脚输出振荡器频率的 4 分频信号。在 ECPLL 振荡器模式下, 由 OSC2 引脚输出 PLL 输出的 4 分频信号。此信号可用于测试或同步其他逻辑。图 3-3 显示了 EC 振荡器模式的引脚连接方式。

图 3-3: 外部时钟输入工作原理
(EC 和 ECPLL 配置)



3.2.4 PLL 倍频器

PIC18F46J50 系列器件包含一个 PLL 电路。这是专为具有低速振荡器的 USB 应用提供的, 也可用作单片机时钟源。

PLL 可以在 HSPLL、ECPLL、INTOSCPLL 和 INTOSCPLLO 振荡器模式下使能, 方法是将 PLEN 位 (OSCTUNE<6>) 置 1。它设计为从固定的 4 MHz 输入产生固定的 96 MHz 参考时钟。输出随后可进行分频, 用于 USB 和单片机内核时钟。由于 PLL 具有固定的频率输入和输出, 有 8 个预分频比选项用来匹配振荡器输入频率到 PLL。该预分频器使 PLL 可以与晶振、谐振器和外部时钟 (频率是 4 MHz 的整数倍) 配合使用。例如, 可以在预分频器 3 分频模式下使用 12 MHz 的晶振来驱动 PLL。

此外, 还有一个 CPU 分频器, 它可用于从 PLL 产生单片机时钟。这就允许 USB 外设和单片机使用相同的振荡器输入, 但仍允许在不同的时钟速度下工作。CPU 分频器可以按照分频因子 1、2、3 或 6 来降低输入频率。

3.2.5 内部振荡器模块

PIC18F46J50 系列器件包含一个可产生两种不同时钟信号的内部振荡器模块; 这两种信号均可用作单片机的时钟源。有了内部振荡器可无需在 OSC1 和 / 或 OSC2 引脚上使用外部振荡器电路。

主输出 (INTOSC) 是一个 8 MHz 的时钟源, 可用于直接驱动器件时钟。它还可以驱动 INTOSC 后分频器, 该分频器可提供从 31 kHz 到 8 MHz 的时钟频率。此外, 可以将 INTOSC 与 PLL 配合使用, 产生最高 48 MHz 的时钟频率。

另一个时钟源是内部 RC 振荡器 (INTRC), 它提供了标称值为 31 kHz 的输出。如果选择 INTRC 作为器件的时钟源, 它就会被使能。当使能以下任一功能时, 也将自动使能 INTRC:

- 上电延时定时器
- 故障保护时钟监视器
- 看门狗定时器
- 双速启动

第 27.0 节 “CPU 的特殊功能” 将详细讨论以上功能。

通过配置 OSCCON 寄存器 (第 43 页) 的 IRCF 位, 可以选择时钟源频率 (INTOSC 直接频率、INTRC 直接频率或 INTOSC 后分频器频率)。

3.2.5.1 OSCTUNE 寄存器

内部振荡器的输出已在出厂前经过校准，但仍可以在用户应用中调整。这是通过写 OSCTUNE 寄存器（寄存器 3-1）完成的。在整个调节范围内调节灵敏度保持不变。

当修改了 OSCTUNE 寄存器后，INTOSC 的频率将开始改变为新的频率。在此转变期间，代码会继续执行。是否已发生频率转变并无明确的指示。

OSCTUNE 寄存器还包含 INTSRC 位。当选取 31 kHz 频率时，用户可通过 INTSRC 位选择用作时钟源的内部振荡器。在第 3.5.1 节“振荡器控制寄存器”中对此进行了更详细的说明。

以一种 PLL 类型的振荡器模式（例如，INTOSCPLL）运行时，OSCTUNE 寄存器中包含的 PLEN 位可用于使能或禁止内部 96 MHz PLL。名称中未包含“PLL”的振荡器模式不能与 PLL 一起使用。在这些模式下，无论 PLEN 位的设置如何，始终禁止 PLL。

当配置为一种使能 PLL 的模式时，如果将 PLEN 位置 1，器件时钟不会立即切换为 PLL 输出。PLL 需要最多 t_{TC} （电气参数）时间进行启动和锁定，在此期间，器件继续使用原时钟工作。当 PLL 输出就绪时，单片机内核将自动切换为 PLL 产生的频率。

3.2.5.2 内部振荡器输出频率和漂移

出厂时已校准了内部振荡器模块使之能够产生 8.0 MHz 的 INTOSC 输出频率。但是，此频率可能会随着 V_{DD} 或温度的改变而发生漂移，这一点可能会以各种方式影响控制器的工作。

低频 INTRC 振荡器的工作独立于 INTOSC 时钟源。电压和温度变化导致的任何 INTOSC 变化并不一定会使 INTRC 变化，反之亦然。

3.2.5.3 INTOSC 漂移补偿

通过修改 OSCTUNE 寄存器中的值可以调节 INTOSC 的频率。这不会对 INTRC 时钟源的频率造成影响。

调节 INTOSC 时钟源需要了解何时调节、调节的方向以及在某些情况下的调整量。当使用 EUSART 时，例如在 EUSART 开始产生帧错误，或者在异步模式下接收数据有错误时，可能需要进行调节。帧错误表示器件时钟频率太高；要对此进行调节，可以减小 OSCTUNE 中的值来降低时钟频率。另一方面，数据中有错误可能表明时钟速度太低；要对此进行补偿，可以增大 OSCTUNE 中的值来提高时钟频率。

此外，还可以基于参考时钟来验证器件时钟速度。可能要用到两个定时器：一个由外设时钟提供时钟源，而另一个由一个固定的参考源（如 Timer1 振荡器）提供时钟源。两个定时器都被清零，但由参考源提供时钟信号的定时器产生中断。当中断发生时，使用内部时钟源的定时器值被读取且两个定时器均被清零。如果使用内部时钟源的定时器的值大于期望值，则表示内部振荡器模块运行过快。要对此进行调整，需减小 OSCTUNE 寄存器中的值。

最后，ECCP 模块可以使用由内部振荡器模块提供时钟信号的自由运行 Timer1（或 Timer3）和已知周期的外部事件（即交流电源频率）。在 CCPRxH:CCPRxL 寄存器中捕捉并记录第一个事件的时间以备以后使用。当第二个事件导致捕捉时，要用第二个事件的时间减去第一个事件的时间。由于外部事件的周期是已知的，因此可以计算两个事件之间的时间差。

如果测得的时间比计算得到的时间大，则表示内部振荡器模块运行过快；要对此进行补偿，需减小 OSCTUNE 寄存器中的值。如果测得的时间比计算得到的时间小，则表示内部振荡器模块运行过慢；要对此进行补偿，需增大 OSCTUNE 寄存器中的值。

PIC18F46J50 系列

寄存器 3-1: OSCTUNE: 振荡器调节寄存器 （位于快速操作存储区，地址 F9Bh）

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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	INTSRC: 内部振荡器低频时钟源选择位 1 = 来自 8 MHz INTOSC 时钟源的 31.25 kHz 器件时钟 （使能 256 分频） 0 = 直接来自 INTRC 内部振荡器的 31 kHz 器件时钟
bit 6	PLLEN: 倍频器使能位 1 = 使能 96 MHz PLL 0 = 禁止 96 MHz PLL
bit 5-0	TUN<5:0>: 频率调节位 011111 = 最高频率 011110 . . . 000001 000000 = 中心频率；振荡器模块运行在经过校准的频率下 111111 . . . 100000 = 最低频率

3.3 USB 的振荡器设置

当 PIC18F46J50 系列器件用于 USB 连接时，必须为 USB 模块提供 6 MHz 或 48 MHz 的时钟，以分别工作于低速或全速模式。这要求选择振荡器频率和对器件编程时预先计划好。

表 3-5 中给出了与 USB 操作兼容的所有可能的振荡器配置。

3.3.1 低速操作

低速模式的 USB 时钟来自主振荡器或 96 MHz PLL。为了使 USB 模块以低速模式工作，必须为 USB 模块提供 6 MHz 的时钟。由于时钟分频器在 PIC18F46J50 系列

中的实现方式的原因，单片机内核必须以 24 MHz 运行，以使 USB 模块获得低速 USB 操作所需的 6 MHz 时钟。有几种时钟方案可用于满足这两种必需条件。关于可用于低速 USB 操作的可能组合，请参见表 3-4 和表 3-5。

表 3-4: 低速 USB 的时钟

时钟输入	CPU 时钟	CPDIV<1:0>	USB 时钟
48	24	10	48/8 = 6 MHz
24	24	11	24/4 = 6 MHz

表 3-5: USB 操作的振荡器配置选项

输入振荡器频率	PLL 分频 (PLLDIV<2:0>)	时钟模式 (FOSC<2:0>)	MCU 时钟分频 (CPDIV<1:0>)	单片机 时钟频率
48 MHz	N/A	EC	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
48 MHz	+12 (000)	ECPLL	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
40 MHz	+10 (001)	ECPLL	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
24 MHz	+6 (010)	ECPLL	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
24 MHz	N/A	EC	无 (11)	24 MHz
			+2 (10)	12 MHz
			+3 (01)	8 MHz
			+6 (00)	4 MHz
20 MHz	+5 (011)	ECPLL	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
16 MHz	+4 (100)	HSPLL 和 ECPLL	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
12 MHz	+3 (101)	HSPLL 和 ECPLL	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
8 MHz	+2 (110)	HSPLL、ECPLL、 INTOSCPLL 和 INTOSCPLLO	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz
4 MHz	+1 (111)	HSPLL 和 ECPLL	无 (11)	48 MHz
			+2 (10)	24 MHz
			+3 (01)	16 MHz
			+6 (00)	8 MHz

图注：所有时钟频率（除 24 MHz 外）都专门与全速 USB 操作（48 MHz 的 USB 时钟）关联。**粗体**文字用于强调与低速 USB 操作（系统时钟 24 MHz，USB 时钟 6 MHz）兼容的时钟选择。

PIC18F46J50 系列

3.4 INTOSC 提供 USB 时钟

PIC18F46J50 系列中所有器件所包含的 8 MHz INTOSC 的精度都很高。当此振荡器与 96 MHz PLL 配合使用时，可用来产生 USB 模块的时钟。INTOSC 的高精度使得应用能够满足低速 USB 信号传输速率规范。

3.5 时钟源与振荡器切换

与早期的 PIC18 增强型器件一样，PIC18F46J50 系列包含允许将器件时钟源从主振荡器切换到备用低频时钟源的功能。PIC18F46J50 系列器件提供了两个备用时钟源。当使能备用时钟源时，可以使用多种功耗管理工作模式。

基本上，这些器件都有 3 种时钟源：

- 主振荡器
- 辅助振荡器
- 内部振荡器模块

主振荡器包括外部晶振和谐振器模式、外部时钟模式以及内部振荡器模块。特定的模式由 FOSC<2:0> 配置位定义。这些模式的详细信息已在本章前面的内容中做过介绍。

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

PIC18F46J50 系列器件将 Timer1 振荡器作为辅助振荡器。此振荡器（在所有功耗管理模式中）通常是实时时钟（Real-Time Clock, RTC）等功能的时基。大部分情况下，在 RC0/T1OSO/T1CKI/RP11 和 RC1/T1OSI/UOE/RP12 引脚之间接有一个 32.768 kHz 的时钟晶振。与 HS 振荡器模式电路类似，在每个引脚与地之间均接有负载电容。将在第 13.5 节“Timer1 振荡器”中详细讨论 Timer1 振荡器。

除了作为主时钟源之外，**经过后分频的内部时钟**还可以作为功耗管理模式的时钟源。INTRC 时钟源也可用作几种特殊功能部件（例如 WDT 和故障保护时钟监视器（FSCM））的时钟源。

3.5.1 振荡器控制寄存器

OSCCON 寄存器（寄存器 3-2）控制全功耗模式和功耗管理模式下器件时钟工作的多个方面。

系统时钟选择位 SCS<1:0> 用于选择时钟源。可用的时钟源包括主时钟（由 FOSC<2:0> 配置位定义）、辅助时钟（Timer1 振荡器）和经过后分频的内部时钟。当写入一个或多个位之后，接着是一段很短的时钟转换间隔，然后时钟源会立即改变。发生所有形式的复位时 SCS 位都会被清零。

内部振荡器频率选择位 IRCF<2:0> 用于选择经过后分频的内部时钟线上的频率输出。这些频率可以是 INTRC 时钟源的频率、INTOSC 时钟源的频率（8 MHz）或 INTOSC 后分频器产生的几个频率（31 kHz 到 4 MHz）之一。如果器件时钟由经过后分频的内部时钟提供，改变这些位的状态会使内部振荡器输出立即改变。器件复位时，INTOSC 后分频器的默认输出频率设置为 4 MHz。

当选取 31 kHz 的输出频率（IRCF<2:0> = 000）时，用户可以选择用作时钟源的内部振荡器。这通过使用 OSCTUNE 寄存器中的 INTSRC 位（OSCTUNE<7>）完成。将该位置 1，将通过使能 INTOSC 后分频器的 256 分频输出选择 INTOSC 作为 31.25 kHz 时钟源。清零 INTSRC 位选择 INTRC（标称值为 31 kHz）作为时钟源。

此选项使用户能选择可调节且更精确的 INTOSC 作为时钟源，同时以非常低的时钟速度运行以节省功耗。无论 INTSRC 的设置如何，INTRC 总是作为 WDT 和 FSCM 之类功能部件的时钟源。

OSTS 和 T1RUN 位指示当前提供器件时钟的是哪一个时钟源。OSTS 位置 1 表明振荡器起振定时器（Oscillator Start-up Timer, OST）已超时且主时钟在主时钟模式下提供器件时钟。T1RUN 位（T1CON<6>）置 1 表明 Timer1 振荡器正在辅助时钟模式下提供器件时钟。在功耗管理模式下，任何时候这些位中只有一位会置 1。如果这些位都没有置 1，则表示当前时钟源是 INTRC，或内部振荡器模块刚刚起振而尚未稳定。

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

第 4.0 节“低功耗模式”更详细地讨论了 OSCCON 寄存器中标志位和控制位的使用。

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

3.5.2 振荡器转换

PIC18F46J50 系列器件包含在时钟源切换时防止时钟产生“毛刺”的电路。在切换时钟时, 器件时钟会有短暂的停顿。该停顿的时间长度是旧时钟源的两个周期与新时钟源的三到四个周期的和。此公式假设新时钟源是稳定的。

第 4.1.2 节“进入功耗管理模式”详细讨论了时钟转换。

寄存器 3-2: OSCCON: 振荡器控制寄存器 (位于快速操作存储区, 地址 FD3h)

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

图注:

R = 可读位

W = 可写位

U = 未实现位

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **IDLEN:** 空闲使能位

1 = 执行 SLEEP 指令后器件进入空闲模式

0 = 执行 SLEEP 指令后器件进入休眠模式

bit 6-4 **IRCF<2:0>:** 内部振荡器频率选择位

111 = 8 MHz (由 INTOSC 直接驱动时钟)

110 = 4 MHz⁽²⁾

101 = 2 MHz

100 = 1 MHz

011 = 500 kHz

010 = 250 kHz

001 = 125 kHz

000 = 31 kHz (来自 INTOSC/256 或直接来自 INTRC) ⁽³⁾

bit 3 **OSTS:** 振荡器起振延时状态位 ⁽¹⁾

1 = 振荡器起振定时器延时已结束; 主振荡器正在运行

0 = 振荡器起振定时器延时正在进行; 主振荡器尚未就绪

bit 2 **未实现:** 读为 1

bit 1-0 **SCS<1:0>:** 系统时钟选择位

11 = 经过后分频的内部时钟 (来自 INTRC/INTOSC)

10 = 保留

01 = Timer1 振荡器 ⁽⁴⁾

00 = 主时钟源 (当 FOSC<2:0> = 001 或 000 时的 INTOSC 后分频器输出)

00 = 主时钟源 (FOSC<2:0> 为其他值时的 CPU 分频器输出)

注 1: 双速启动使能时复位值为 0, 禁止时为 1。

2: 复位时 INTOSC 的默认输出频率 (4 MHz)。

3: 由 INTSRC 位 (OSCTUNE<7>) 选择的时钟源。

4: 首先, 应用固件应将 T1OSCEN 位置 1 使能 Timer1 振荡器。

PIC18F46J50 系列

3.6 参考时钟输出

除了某些振荡器模式中的外设时钟 4 分频输出外，PIC18F46J50 系列中的器件时钟也可以配置为向端口引脚提供参考时钟输出信号。该功能在所有振荡器配置中都可用，允许用户选择更大范围的时钟因数来驱动应用中的外部器件。

该参考时钟输出由 REFOCON 寄存器（寄存器 3-3）控制。将 ROON 位（REFOCON<7>）置 1 将使 REFO（RB2）引脚上输出时钟信号。RODIV<3:0> 位允许选择 16 个不同的时钟分频比选项。

ROSSLP 和 ROSEL 位（REFOCON<5:4>）控制休眠模式下参考输出的可用性。ROSEL 位决定用 OSC1 和 OSC2 上的振荡器还是当前系统时钟源作为参考时钟输出。ROSSLP 位决定器件处于休眠模式时 RB2 上的参考源是否可用。

要在休眠模式下使用参考时钟输出，ROSSLP 和 ROSEL 位都必须置 1。器件时钟也必须配置为 EC 或 HS 模式；否则，在器件进入休眠模式时，OSC1 和 OSC2 上的振荡器将会掉电。清零 ROSEL 位允许参考输出频率在任何时钟切换期间随着系统时钟的改变而改变。

寄存器 3-3: REFOCON: 参考振荡器控制寄存器（位于快速操作存储区，地址 F3Dh）

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	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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7ROON: 参考振荡器输出使能位
1 = 在 REFO 引脚上使能参考振荡器
0 = 禁止参考振荡器
- bit 6未实现: 读为 0
- bit 5ROSSLP: 参考振荡器输出在休眠模式下停止的位
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禁止
- bit 4ROSEL: 参考振荡器源选择位
1 = 主振荡器晶振 / 谐振器用作基本时钟 (1)
0 = 系统时钟 (Fosc) 用作基本时钟; 基本时钟反映器件的任何时钟切换
- bit 3-0RODIV<3:0>: 参考振荡器分频比选择位
1111 = 基本时钟值被 32,768 分频
1110 = 基本时钟值被 16,384 分频
1101 = 基本时钟值被 8,192 分频
1100 = 基本时钟值被 4,096 分频
1011 = 基本时钟值被 2,048 分频
1010 = 基本时钟值被 1,024 分频
1001 = 基本时钟值被 512 分频
1000 = 基本时钟值被 256 分频
0111 = 基本时钟值被 128 分频
0110 = 基本时钟值被 64 分频
0101 = 基本时钟值被 32 分频
0100 = 基本时钟值被 16 分频
0011 = 基本时钟值被 8 分频
0010 = 基本时钟值被 4 分频
0001 = 基本时钟值被 2 分频
0000 = 基本时钟值

注 1: 必须使用 FOSC<2:0> 位使能晶振; 在休眠模式下，晶振会保持工作。

3.7 功耗管理模式对各种时钟源的影响

当选取 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。除非使能了 USB 模块，否则 OSC1 引脚（以及可能由振荡器使用的 OSC2 引脚）将会停止振荡。

在辅助时钟模式（SEC_RUN 和 SEC_IDLE）下，Timer1 振荡器作为器件时钟源工作。如果需要，Timer1 振荡器也可以运行在所有功耗管理模式下为 Timer1 或 Timer3 提供时钟。

在内部振荡器模式（RC_RUN 和 RC_IDLE）下，由内部振荡器模块提供器件时钟源。无论是哪种功耗管理模式，31 kHz 的 INTRC 输出均可被直接用来提供时钟并且可被使能来支持多种特殊的功能部件（关于 WDT、FSCM 和双速启动的更多信息，请参见第 27.2 节“看门狗定时器（WDT）”、第 27.4 节“双速启动”和第 27.5 节“故障保护时钟监视器”）。8 MHz 的 INTOSC 输出可被直接用来为器件提供时钟，或者也可先由后分频器进行分频再用作器件时钟。如果直接由 INTRC 输出提供时钟，则会禁止 INTOSC 输出。

如果选择了休眠模式，所有不再需要的时钟源都会被停止。因为休眠模式下所有晶体管中都没有开关电流，因此除了深度休眠外，休眠模式能实现最低的器件电流消耗（仅泄漏电流）。

当使能 USB 模块、并以全功耗模式工作时，不应调用休眠模式。在选择休眠模式之前，应将 USB 模块置于挂起状态。方法是将 UCON 寄存器中的 SUSPND 位置 1。

在休眠模式下使能任何片上功能部件的工作都将增加休眠模式期间的电流消耗。需要使用 INTRC 来支持 WDT 工作。Timer1 振荡器可用来为 RTC 提供时钟源。不需要器件时钟源的其他功能部件也可以工作（即，MSSP 从器件、PMP 和 INTx 引脚等）。在第 30.2 节“直流特性：PIC18F46J50 系列的掉电电流和供电电流（工业级）”中列出了可能显著增加电流消耗的外设。

3.8 上电延时

由两个定时器控制上电延时，这样大多数应用都无需外接复位电路。上电延时可以确保在器件电源稳定（常规环境下）和主时钟稳定工作之前器件保持在复位状态。关于上电延时的更多信息，请参见第 5.6 节“上电延时定时器（PWRT）”。

第一个定时器是上电延时定时器（Power-up Timer，PWRT），它在上电时提供固定的延时（表 30-14 中的参数 33）。

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

POR 之后有一个 TcSD 间隔的延时（表 30-14 中的参数 38），在延时期间控制器为执行指令做准备。此延时与任何其他延时并行发生。当将内部振荡器或 EC 模式之一用作主时钟源时，这可能是唯一的延时。

PIC18F46J50 系列

注:

4.0 低功耗模式

PIC18F46J50 系列器件可以通过控制 CPU 和外设的时钟来管理功耗。一般来说，降低时钟频率和减少使用时钟的电路数量可以降低功耗。

对于应用中的功耗管理，主要的工作模式有：

- 运行模式
- 空闲模式
- 休眠模式
- 深度休眠模式

此外，还有用于在 RA0 上产生电平变化中断的超低功耗唤醒（Ultra Low-Power Wake-up，ULPWU）模式。

这些模式定义了需要为器件的哪些部分提供时钟以及时钟的速度。

- 运行模式和空闲模式可以使用三种时钟源（主时钟源、辅助时钟源或内部振荡器模块）中的任意一种。
- 休眠模式则不使用时钟源。

RA0 上的 ULPWU 模式允许缓慢下降的电压能够在 RA0 上产生电平变化中断，而不消耗很大的电流。请参见第 4.7 节“超低功耗唤醒”。

功耗管理模式含有几种在以前的 PIC® 器件上已提供的节能特性，例如时钟切换、ULPWU 和休眠模式。除此之外，PIC18F46J50 系列器件还增加了一种新的功耗管理模式，即深度休眠模式。

4.1 选择功耗管理模式

选择功耗管理模式之前需要先做出两个决定：

- 是否为 CPU 提供时钟源？
- 如果提供，使用何种时钟源？

IDLEN 位（OSCCON<7>）控制是否为 CPU 提供时钟源，而 SCS<1:0> 位（OSCCON<1:0>）选择时钟源。表 4-1 总结了各个模式下的位设置、时钟源和受影响的模块。

4.1.1 时钟源

SCS<1:0> 位允许为功耗管理模式在三个时钟源中任选其一。它们是：

- 主时钟源 —— 由 FOSC<2:0> 配置位定义
- Timer1 时钟 —— 由辅助振荡器提供
- 经过分频的内部时钟 —— 来自内部振荡器模块

4.1.2 进入功耗管理模式

可以通过装载 OSCCON 寄存器从一种时钟源切换到另一种时钟源。SCS<1:0> 位用于选择时钟源。

更改这些位会导致立即切换到一个新的时钟源（假定新时钟源正在运行）。此切换可能还会引起时钟转换延时。第 4.1.3 节“时钟转换和状态指示”及其后续章节将会讨论这些延时。

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 位的状态决定。

更改功耗管理模式并不总是要求设置所有这些位，而是取决于当前的模式和将要切换到的模式。通过在发出 SLEEP 指令之前更改振荡器选择位、IDLEN 位或 DSEN 位可完成多种模式转换。

如果已经正确配置了 IDLEN 和 DSEN 位，可能只需执行 SLEEP 指令就可切换到所需的模式。

PIC18F46J50 系列

表 4-1: 低功耗模式

模式	DSCONH<7>	OSCCON<7,1:0>		模块时钟		可用时钟和振荡器源
	DSEN ⁽¹⁾	IDLEN ⁽¹⁾	SCS<1:0>	CPU	外设	
休眠	0	0	N/A	关闭	关闭	可选择是否使能 Timer1 振荡器和 / 或 RTCC
深度休眠	1	0	N/A	关闭 ⁽²⁾	关闭	RTCC 可以使用 Timer1 振荡器或内部低功耗 RC 振荡器不间断地运行
PRI_RUN	0	N/A	00	提供时钟	提供时钟	正常的全功耗执行模式；主时钟源（由 FOSC<2:0> 定义）
SEC_RUN	0	N/A	01	提供时钟	提供时钟	辅助时钟 ——Timer1 振荡器
RC_RUN	0	N/A	11	提供时钟	提供时钟	经过后分频的内部时钟
PRI_IDLE	0	1	00	关闭	提供时钟	主时钟源（由 FOSC<2:0> 定义）
SEC_IDLE	0	1	01	关闭	提供时钟	辅助时钟 ——Timer1 振荡器
RC_IDLE	0	1	11	关闭	提供时钟	经过后分频的内部时钟

注 1: IDLEN 和 DSEN 在执行 SLEEP 指令时反映其值。
2: 深度休眠会关断稳压器以实现超低功耗。更多信息，请参见第 4.6 节“深度休眠模式”。

4.1.3 时钟转换和状态指示

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

以下两位用于指明当前的时钟源及其状态：OSTS（OSCCON<3>）和 T1RUN（T1CON<6>）。一般来说，在给定的功耗管理模式下，这两个位中只有一个位会置 1。当 OSTS 位置 1 时，表明由主时钟提供器件时钟。当 T1RUN 位置 1 时，表明由 Timer1 振荡器提供时钟源。如果这些位均不置 1，则由 INTRC 为器件提供时钟信号。

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

4.1.4 多条 SLEEP 命令

使用 SLEEP 指令调用功耗管理模式时，具体进入何种模式由该指令执行时 IDLEN 和 DSEN 位的设置决定。如果执行了另一条 SLEEP 指令，器件将进入由此时 IDLEN 和 DSEN 位指定的功耗管理模式。如果 IDLEN 或 DSEN 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。

4.2 运行模式

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

4.2.1 PRI_RUN 模式

PRI_RUN 模式是单片机的正常全功耗执行模式。除非使能了双速启动（详情请参见第 27.4 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 位置 1（见第 3.5.1 节“振荡器控制寄存器”）。

4.2.2 SEC_RUN 模式

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

通过将 SCS<1:0> 位设置为 01 可以进入 SEC_RUN 模式。器件时钟源被切换到 Timer1 振荡器（见图 4-1），主振荡器被关闭，T1RUN 位（T1CON<6>）被置 1 并且 OSTS 位被清零。

注： Timer1 振荡器应该在进入 SEC_RUN 模式之前就已经运行了。如果 SCS<1:0> 位被设置为 01 时，T1OSCEN 位未置 1，则不会进入 SEC_RUN 模式。如果使能了 Timer1 振荡器，但还没有运行，器件时钟将会延时直到该振荡器起振。在这种情况下，最初的振荡器工作很不稳定，可能会导致无法预料的结果。

在从 SEC_RUN 模式转换到 PRI_RUN 模式时，外设和 CPU 继续使用 Timer1 振荡器作为时钟源，直到主时钟启动。当主时钟就绪以后，时钟切换回主时钟（见图 4-2）。

当时钟切换完成后，T1RUN 位被清零，OSTS 位被置 1 并且由主时钟提供时钟源。这种唤醒不会影响 IDLEN 和 SCS 位；Timer1 振荡器继续运行。

图 4-1: 进入 SEC_RUN 模式的转换时序

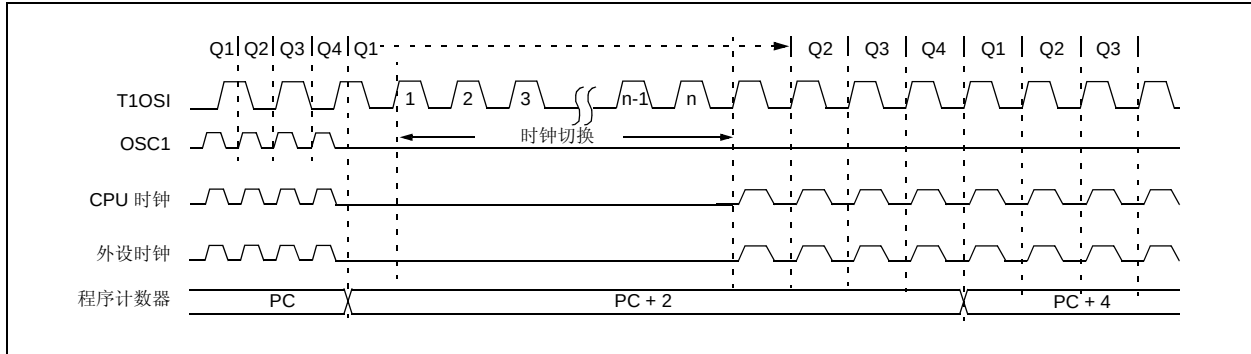
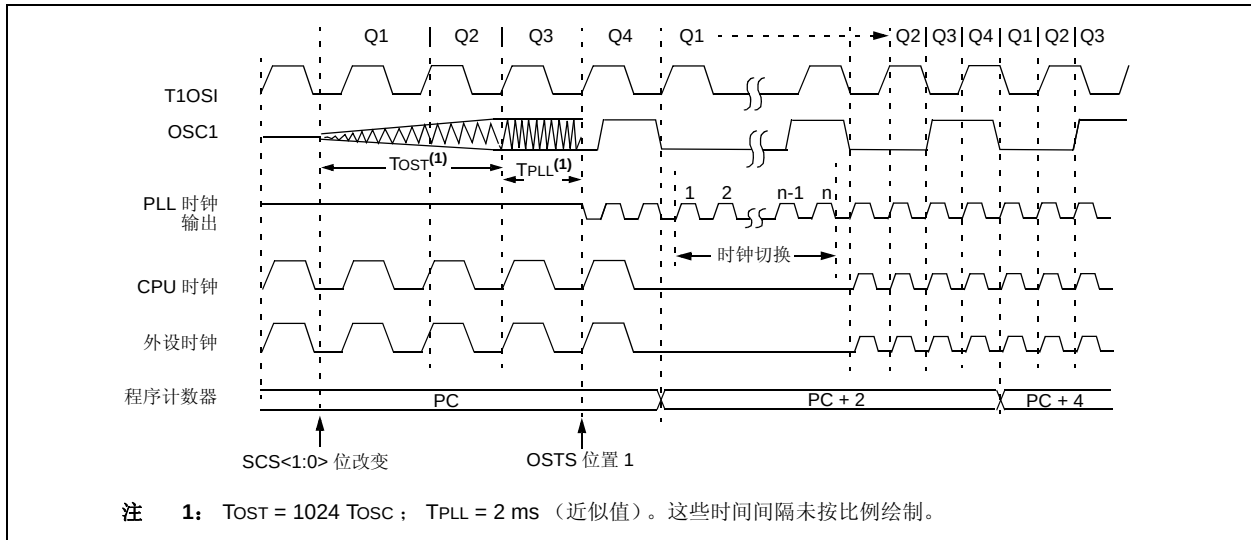


图 4-2: 从 SEC_RUN 模式切换到 PRI_RUN 模式的转换时序 (HSPLL)



PIC18F46J50 系列

4.2.3 RC_RUN 模式

在 RC_RUN 模式下，将内部振荡器作为 CPU 和外设的时钟源；主时钟关闭。此模式是在代码执行期间所有运行模式中最节省功耗的模式。它非常适用于对时序不是很敏感或者不是一直需要高速时钟的应用。

通过将 SCS<1:0> 位（OSCCON<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时（见图4-3），主振荡器将被关闭并且OSTS位被清零。

在从 RC_RUN 模式转换到 PRI_RUN 模式时，在主时钟处于启动状态时，器件将继续使用 INTOSC 模块作为时钟源。当主时钟就绪以后，时钟切换回主时钟（见图4-4）。当时钟切换完成后，OSTS 位被置 1 并且由主时钟提供器件时钟。这种切换不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或 FSCM，INTRC 时钟源将继续运行。

图 4-3: 到 RC_RUN 模式的转换时序

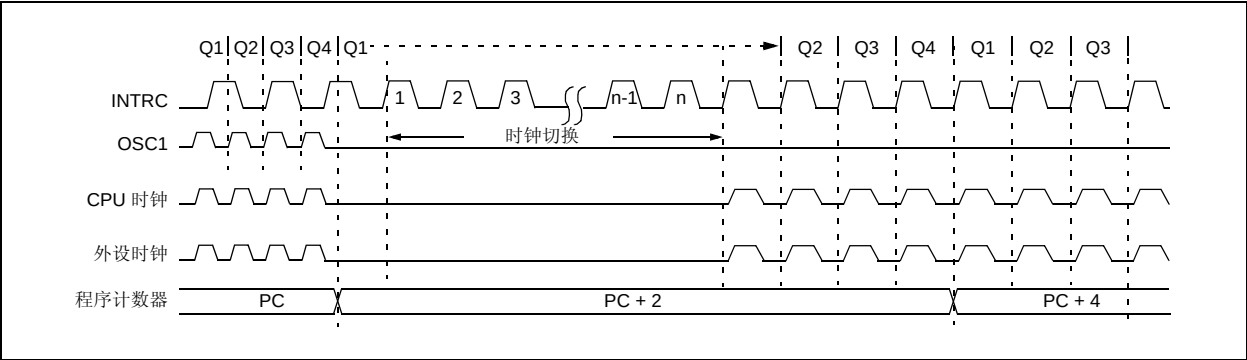
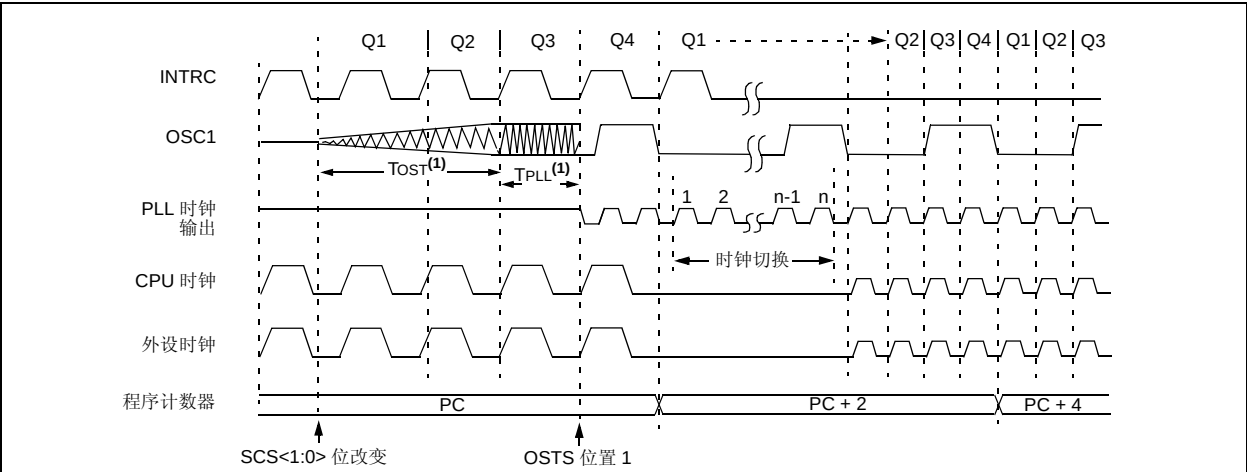


图 4-4: 从 RC_RUN 模式切换到 PRI_RUN 模式的转换时序



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

4.3 休眠模式

功耗管理休眠模式和所有其他 PIC 器件提供的传统休眠模式相同。通过清零 IDLEN 位（器件复位时的默认状态）并执行 SLEEP 指令即可进入此模式。这将关闭选定的振荡器（见图 4-5），并将所有的时钟源状态位清零。从任何其他模式进入休眠模式不需要切换时钟。这是因为控制器一旦进入休眠模式就不需要时钟了。如果选择了 WDT，INTRC 时钟源将继续工作。如果使能了 Timer1 振荡器，它也将继续运行。

当在休眠模式下发生唤醒事件（通过中断、复位或 WDT 超时）时，在时钟源（通过 SCS<1:0> 位选择）就绪之前器件将没有时钟源（见图 4-6），或者如果使能了双速启动或 FSCM，它将使用内部振荡器作为时钟源（见第 27.0 节“CPU 的特殊功能”）。在这两种情况下，当由主时钟提供器件时钟时，OSTS 位将置 1。唤醒不会影响 IDLEN 和 SCS 位。

图 4-5: 进入休眠模式的转换时序

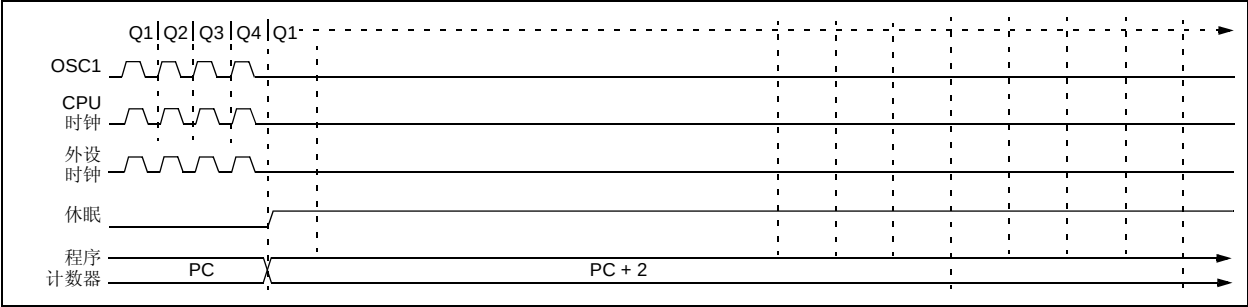
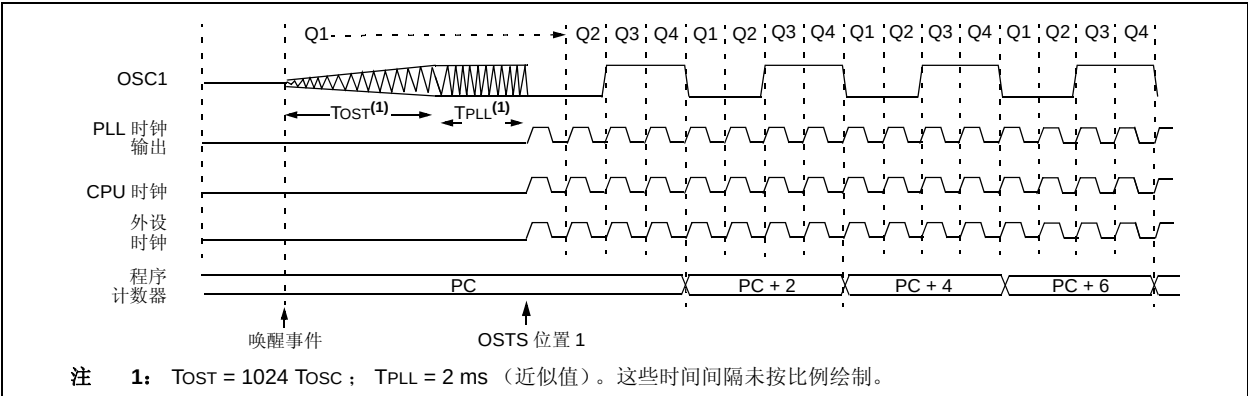


图 4-6: 从休眠模式唤醒的转换时序（HSPLL）



4.4 空闲模式

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

如果在执行 SLEEP 指令时，IDLEN 位被置 1，外设将使用由 SCS<1:0> 位选择的时钟源，而 CPU 没有时钟源。时钟源状态位不受影响。将 IDLEN 置 1 并执行 SLEEP 指令可以从给定的运行模式快速切换到相应的空闲模式。

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

由于 CPU 不在执行指令，器件只能通过中断、WDT 超时或复位从空闲模式退出。当 CPU 开始执行代码时，它将沿用当前空闲模式所使用的时钟源。例如，当从 RC_IDLE 模式唤醒时，将使用内部振荡器模块为 CPU 和外设提供时钟（即 RC_RUN 模式）。唤醒不会影响 IDLEN 和 SCS 位。

当处于任何空闲或休眠模式时，WDT 超时会导致 WDT 唤醒并进入当前由 SCS<1:0> 位指定的运行模式。

4.4.1 PRI_IDLE 模式

在三种低功耗空闲模式中，只有该模式不会禁止主器件时钟。由于时钟源不需要“预热”或是从其他振荡器转换过来，对于时序敏感的应用，选用此模式可以使用较精确的主时钟源以最快的速度恢复器件运行。

可以通过将 IDLEN 位置 1 并执行 SLEEP 指令以实现从 PRI_RUN 模式进入 PRI_IDLE 模式。如果器件处于另一种运行模式，首先将 IDLEN 置 1，然后将 SCS 位设

置为 00 并执行 SLEEP。虽然 CPU 已被禁止，但外设仍可继续使用由 FOSC<1:0> 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 4-7）。

当发生唤醒事件时，由主时钟源为 CPU 提供时钟。在唤醒之后，OSTS 位保持置 1 状态。这种唤醒不会影响 IDLEN 和 SCS 位（见图 4-8）。

4.4.2 SEC_IDLE 模式

在 SEC_IDLE 模式下，CPU 被禁止，但外设继续将 Timer1 振荡器作为时钟源。可以通过将 IDLEN 位置 1 并执行 SLEEP 指令从 SEC_RUN 模式进入此模式。如果器件处于另一种运行模式，首先将 IDLEN 置 1，然后将 SCS<1:0> 设置为 01 并执行 SLEEP。当时钟源切换到 Timer1 振荡器时，主振荡器被关闭（除非有其他外设仍然需要使用它），OSTS 位被清零并且 T1RUN 位被置 1。

当唤醒事件发生时，外设继续将 Timer1 振荡器作为时钟源。唤醒事件发生后，CPU 开始执行代码并使用 Timer1 振荡器作为其时钟源。这种唤醒不会影响 IDLEN 和 SCS 位。Timer1 振荡器继续运行（见图 4-8）。

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 位未置 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 4-7： 进入空闲模式的转换时序

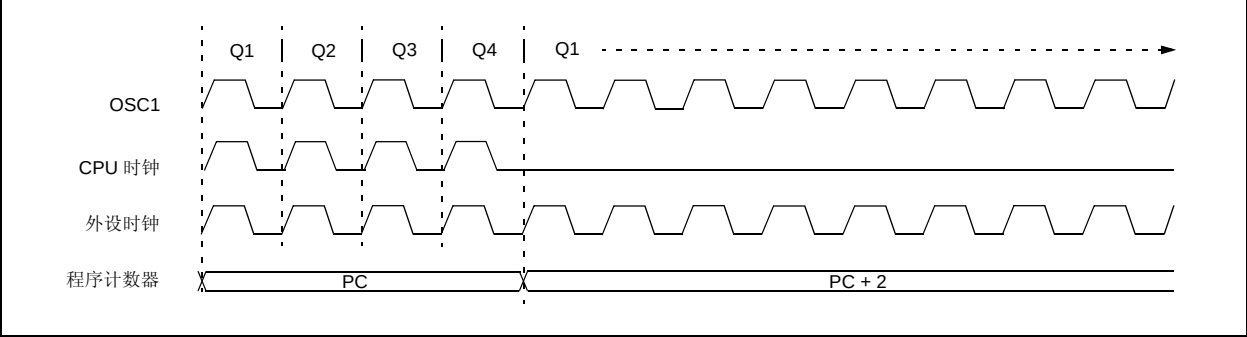
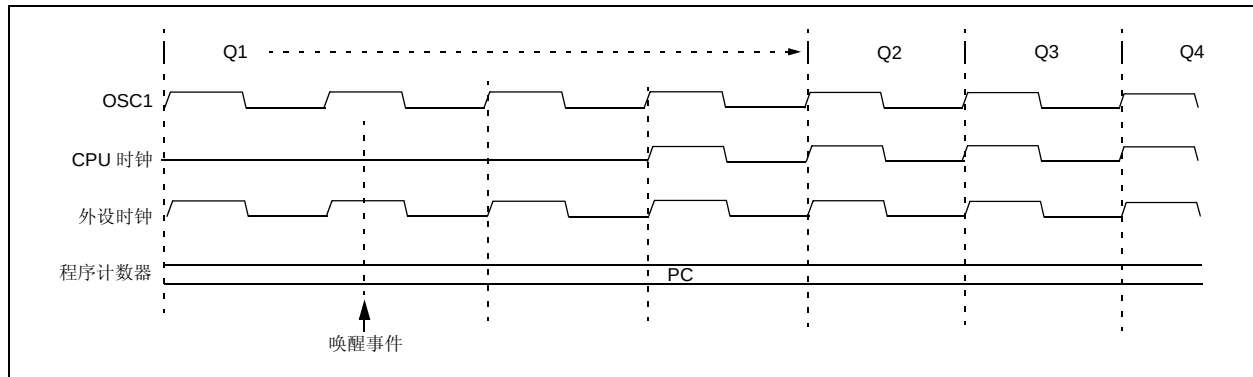


图 4-8: 从空闲模式唤醒进入运行模式的转换时序。



4.4.3 RC_IDLE 模式

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

可以通过将 IDLEN 位置 1 并执行 SLEEP 指令从 RC_RUN 模式进入此模式。如果器件处于另一种运行模式，首先将 IDLEN 置 1，然后清零 SCS 位并执行 SLEEP。当时钟源切换到 INTOSC 模块时，主振荡器被关闭并且 OSTS 位被清零。

当唤醒事件发生时，外设继续将内部振荡器模块作为时钟源。唤醒事件发生后，CPU 使用 INTRC 作为其时钟源开始执行代码。唤醒不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或 FSCM，INTRC 时钟源将继续运行。

4.5 退出空闲和休眠模式

由中断、复位或 WDT 超时触发从休眠模式或任何空闲模式的退出。本节将讨论从功耗管理模式退出的触发方式。在每种功耗管理模式章节中我们还将讨论其时钟子系统的操作（见第 4.2 节“运行模式”、第 4.3 节“休眠模式”和第 4.4 节“空闲模式”）。

4.5.1 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，必须通过将对应 INTCON 或 PIE 寄存器中的中断源允许位置 1 来允许中断源。当相应的中断标志位置 1 时，触发退出操作。

在通过中断从空闲或休眠模式退出时，如果 GIE/GIEH 位（INTCON<7>）置 1，程序将跳转到中断向量处执行代码。否则，代码将继续执行，不进行跳转（见第 9.0 节“中断”）。

4.5.2 通过 WDT 超时退出

根据 WDT 超时发生时器件所处的不同功耗管理模式会引发不同的操作。

如果器件不在执行代码（所有空闲模式和休眠模式），超时将导致从功耗管理模式退出（见第 4.2 节“运行模式”和第 4.3 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 27.2 节“看门狗定时器（WDT）”）。

WDT 和后分频器可由以下任一事件清零：

- 执行 SLEEP 或 CLRWDI 指令
- 当前选定的时钟源失效（如果使能了 FSCM）

4.5.3 通过复位退出

通过复位退出空闲或休眠模式会自动强制器件使用 INTRC 运行。

PIC18F46J50 系列

4.5.4 在没有振荡器起振延时的情况下退出

从某些功耗管理模式退出不需要 OST 延时。有以下两种情形：

- PRI_IDLE 模式（主时钟源不停止）以及主时钟源为 EC 模式
- PRI_IDLE 模式以及主时钟源为 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（EC）。

4.6 深度休眠模式

深度休眠模式无需使用外部开关来切断器件电源，即可使器件进入最低功耗状态。在深度休眠期间，片上 VDDCORE 稳压器会掉电，实际上就会断开单片机内核逻辑的电源。

注： 由于深度休眠模式是通过关闭片上 VDDCORE 稳压器来使单片机掉电，所以只有器件系列中的 PIC18FXXJ 成员可以使用深度休眠功能。器件系列的 PIC18LFXXJ 成员不提供片上稳压器，所以它们不支持深度休眠。

在支持深度休眠的器件上，进入深度休眠模式的方式为：

- 将 REGSLP（WDTCON<7>）位置 1
- 清零 IDLEN 位
- 清零 GIE 位
- 将 DSEN 位（DSCONH<7>）置 1
- 在将 DSEN 置 1 后立即执行 SLEEP 指令（两者之间没有延时和中断）

为了最大程度降低意外进入深度休眠模式的可能性，硬件会在 DSEN 位置 1 两个指令周期后将其清零。因此，为了进入深度休眠模式，将 DSEN 置 1 之后，必须在紧接的指令周期中执行 SLEEP 指令。如果执行 Sleep 指令时 DSEN 未置 1，器件将进入常规的休眠模式。

在深度休眠期间，单片机的内核逻辑电路会掉电，以降低泄漏电流。因此，在深度休眠期间，单片机的大多数外设和功能变为不可用。但是，有一些特定的外设和功能使用单片机的 VDD 电源轨直接供电，因而可以在深度休眠模式下继续工作。

进入深度休眠模式会清零 DSWAKEL 寄存器。但是，如果在进入深度休眠模式之前使能了实时时钟和日历（RTCC），它将继续不间断地工作。

器件具有专用的欠压复位（DSBOR）和看门狗定时器复位（DSWDT），用于监视深度休眠模式下的电压和超时事件。DSBOR 和 DSWDT 独立于用于其他功耗管理模式（运行、空闲和休眠）的标准 BOR 和 WDT。

在深度休眠模式下，当发生唤醒事件（通过 MCLR 复位、RTCC 闹钟、INT0 中断、ULPWU 或 DSWDT）时，器件将退出深度休眠模式并执行上电复位（POR）。当器件从复位释放时，代码将在器件的复位向量处继续执行。

4.6.1 准备深度休眠

因为处于深度休眠模式时，VDDCORE 可能会降低低于 SRAM 保持电压，所以在深度休眠模式下，SRAM 数据可能会丢失。因为退出深度休眠模式会产生 POR，所以大多数特殊功能寄存器（SFR）将复位为其默认的 POR 值。

对于在整个深度休眠周期中需要保存少量数据的应用，可以将数据保存在通用寄存器 DSGPR0 和 DSGPR1 中。在器件处于深度休眠模式时，这些寄存器的内容会被保护，并在整个深度休眠进入和唤醒序列中保持有效。

4.6.2 深度休眠期间的 I/O 引脚

在深度休眠期间，通用 I/O 引脚将保持它们原先的状态。

在进入深度休眠模式之前配置为输入的引脚（TRIS 位置 1）在深度休眠期间将保持高阻态。

在进入深度休眠模式之前配置为输出的引脚（TRIS 位清零）在深度休眠期间将保持为输出引脚。处于该模式时，这些引脚驱动的输出电平将由进入深度休眠模式时其相应的 LAT 位决定。

当器件唤醒时，I/O 引脚的操作取决于唤醒源的类型。

如果器件是由 RTCC 闹钟、INT0 中断、DSWDT 或 ULPWU 事件唤醒，所有 I/O 引脚将继续维持它们原先的状态，即使器件已经完成 POR 序列，并再次执行应用程序代码。配置为输入的引脚在深度休眠期间将保持高阻态，配置为输出的引脚将继续驱动其原先的值。唤醒之后，TRIS 和 LAT 寄存器将复位，但 I/O 引脚将仍然维持其原先的状态。如果固件修改了 I/O 引脚的 TRIS 和 LAT 值，它们不会立即切换到新配置的状态。在固件清零 RELEASE 位（DSCONL<0>）之后，I/O 引脚将被“释放”。这会导致 I/O 引脚切换为由其相应的 TRIS 和 LAT 位值配置的状态。

如果使能了深度休眠 BOR（DSBOR）电路，并且 V_{DD} 降低到低于 DSBOR 和 V_{DD} 轨 POR 门限值，I/O 引脚将被立即释放，类似于清零 RELEASE 位。所有原先的状态信息都将丢失，包括通用寄存器 DSGPR0 和 DSGPR1 的内容。关于此情形的更多详细信息，请参见第 4.6.5 节“深度休眠欠压复位（DSBOR）”。

如果在深度休眠期间发生 $\overline{\text{MCLR}}$ 复位事件，I/O 引脚也会被自动释放，但在这种情况下，DSGPR0 和 DSGPR1 的内容将保持有效。

在所有其他深度休眠唤醒情形中，应用固件需要清零 RELEASE 位，以重新配置 I/O 引脚。

4.6.3 深度休眠唤醒源

可以通过 $\overline{\text{MCLR}}$ 、POR、RTCC、INT0 I/O 引脚中断、DSWDT 或 ULPWU 事件将器件从深度休眠模式唤醒。在唤醒之后，器件会执行 POR。当器件从复位释放时，代码将在器件的复位向量处开始执行。

软件可以通过读取 DS 位（WDTCON<3>）来确定唤醒是否是由于从深度休眠模式退出而导致的。如果该位置 1，说明 POR 是由于深度休眠退出而导致的。DS 位必须用软件手动清零。

软件可以通过读取 DSWAKEH 和 DSWAKEL 寄存器来确定唤醒事件源。在应用固件完成使用 DSWAKEH 和 DSWAKEL 状态寄存器时，再次进入深度休眠模式之前不需要手动清零各个位。在进入深度休眠模式时，这些寄存器会自动被清零。

4.6.3.1 唤醒事件注意事项

仅在处理器彻底处于深度休眠模式下时，才监视深度休眠唤醒事件。如果唤醒事件在器件进入深度休眠模式前发生，则 DSWAKE 寄存器中将不会反映该事件的状态。如果在进入深度休眠前，唤醒源有效，则 CPU 将跳转到中断向量处执行代码（如果唤醒源具有中断位，并且已完全允许相应的中断），或通过继续执行 SLEEP 指令后的指令而放弃进入深度休眠（如果未允许中断的话）。在这样的情况下，应在 SLEEP 指令后安排一段唤醒事件处理程序，以对事件进行处理，并在必要时重新尝试进入深度休眠。

若在器件处于深度休眠模式期间，有多个唤醒源同时使能，则将仅检测第一个有效的唤醒源，并记录在 DSWAKEH/DSWAKEL 状态寄存器中。

4.6.4 深度休眠看门狗定时器（DSWDT）

深度休眠模式具有自己的专用 WDT（DSWDT），可通过 DSWDTPS<3:0> 位设置后分频比来配置 2.1 ms 至 25.7 天的超时。

DSWDT 可以使用 INTRC 或 T1OSC/T1CKI 输入来提供时钟。如果要将 T1OSC/T1CKI 源与晶振配合使用，则在进入深度休眠模式之前，需要将 T1CON 寄存器中的 T1OSCEN 位置 1。参考时钟源通过 DSWDTOSC 位进行配置。

DSWDT 通过 DSWDTEN 位进行使能。进入深度休眠模式时会自动清零 DSWDT。更多信息，请参见第 27.0 节“CPU 的特殊功能”。

PIC18F46J50 系列

4.6.5 深度休眠欠压复位 (DSBOR)

深度休眠模块含有专用的深度休眠 BOR (DSBOR) 电路。可选择通过 DSBOR 配置位来使能该电路。

DSBOR 电路会监视 VDD 电源轨电压。[第 5.4 节“欠压复位 \(BOR\)”](#) 说明了 DSBOR 电路的工作。

4.6.6 RTCC 外设和深度休眠

RTCC 可以在深度休眠模式期间不间断地工作。通过配置闹钟，它可以将器件从深度休眠模式唤醒。

RTCC 时钟源由 RTCOSC 位 (CONFIG3L<1>) 进行配置。可用的参考时钟源有 INTRC 和 T1OSC/T1CKI。如果使用 INTRC，RTCC 的精度将直接依赖于 INTRC 容差。关于配置 RTCC 外设的更多信息，请参见[第 17.0 节“实时时钟和日历 \(RTCC\)”](#)。

4.6.7 典型的深度休眠序列

本节给出使用深度休眠模式的典型序列。可选步骤会有说明，并且在过程末尾的注释中还会给出附加信息。

1. 使能 DSWDT (可选)。(1)
2. 配置 DSWDT 时钟源 (可选)。(2)
3. 使能 DSBOR (可选)。(1)
4. 使能 RTCC (可选)。(3)
5. 配置 RTCC 外设 (可选)。(3)
6. 配置 ULPWU 外设 (可选)。(4)
7. 允许 INT0 中断 (可选)。
8. 通过写入 DSGPR0 和 DSGPR1 寄存器来现场保护 SRAM 数据 (可选)。
9. 将 REGSLP 位 (WDTCON<7>) 置 1 并清零 IDLEN 位 (OSCCON<7>)。
10. 若采用 RTCC 闹钟作为唤醒源，则等待 RTCSYNC 位 (RTCCFG<4>) 清零。
11. 通过将 DSEN 位 (DSCONH<7>) 置 1 并发出 SLEEP 指令来进入深度休眠模式。这两条指令必须背对背执行。
12. 发生唤醒事件时，器件将执行上电复位序列。代码将在器件的复位向量处继续执行。
13. 通过读取深度休眠位 DS (WDTCON<3>) 确定器件是否是从深度休眠模式退出。如果上次是从深度休眠模式退出，该位将置 1。

14. 清零深度休眠位 DS (WDTCON<3>)。
15. 通过读取 DSWAKEH 和 DSWAKEL 寄存器来确定唤醒源。
16. 通过读取 DSBOR 位 (DSCONL<1>) 确定在深度休眠模式期间是否发生了 DSBOR 事件。
17. 读取 DSGPR0 和 DSGPR1 现场保护寄存器 (可选)。
18. 清零 RELEASE 位 (DSCONL<0>)。

- 注**
- 1: DSWDT 和 DSBOR 通过器件的配置位进行使能。更多信息，请参见[第 27.1 节“配置位”](#)。
 - 2: DSWDT 和 RTCC 时钟源通过器件的配置位进行选择。更多信息，请参见[第 27.1 节“配置位”](#)。
 - 3: 更多信息，请参见[第 17.0 节“实时时钟和日历 \(RTCC\)”](#)。
 - 4: 关于配置该外设的更多信息，请参见[第 4.7 节“超低功耗唤醒”](#)。

4.6.8 深度休眠故障检测

如果在深度休眠期间，器件受到异常工作条件（例如静电放电 (Electrostatic Discharge, ESD) 事件）影响，供深度休眠模块使用的内部电路状态可能会损坏。如果发生这种情况，器件可能会出现非预期的行为，例如无法唤醒。

为了防止发生此类情况，深度休眠模块中包含了自动自监视功能。在深度休眠期间，将持续地监视关键的内部节点，以检测可能产生的故障条件（通常情况下不会产生）。如果检测到故障条件，电路会将 DSFLT 状态位 (DSWAKEL<7>) 置 1，并自动将单片机从深度休眠唤醒，产生 POR。

在深度休眠期间，故障检测电路始终使能，在进入深度休眠模式之前不需要任何特别配置。

4.6.9 深度休眠模式寄存器

寄存器4-1至寄存器4-6中列出了深度休眠模式寄存器。

寄存器 4-1: DSCONH: 深度休眠控制高字节寄存器（位于普通存储区，地址 F4Dh）

R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DSEN ⁽¹⁾	—	—	—	—	r	DSULPEN	RTCWDIS
bit 7							bit 0

图注:	r = 保留位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	0 = 清零
	x = 未知

- bit 7 **DSEN:** 深度休眠使能位 ⁽¹⁾
 1 = 在执行 SLEEP 命令时进入深度休眠模式
 0 = 在执行 SLEEP 命令时进入休眠模式
- bit 6-3 **未实现:** 读为 0
- bit 2 **保留:** 始终向该位写入 0
- bit 1 **DSULPEN:** 超低功耗唤醒模块使能位
 1 = 在深度休眠模式下使能 ULPWU 模块
 0 = 在深度休眠模式下禁止 ULPWU 模块
- bit 0 **RTCWDIS:** RTCC 唤醒禁止位
 1 = 禁止从 RTCC 唤醒
 0 = 使能从 RTCC 唤醒

注 1: 为了进入深度休眠模式，在将 DSEN 置 1 之后必须立即执行 Sleep 指令。

寄存器 4-2: DSCONL: 深度休眠控制低字节寄存器（位于普通存储区，地址 F4Ch）

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾
—	—	—	—	—	ULPWDIS	DSBOR	RELEASE
bit 7							bit 0

图注:	
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	0 = 清零
	x = 未知

- bit 7-3 **未实现:** 读为 0
- bit 2 **ULPWDIS:** 超低功耗唤醒禁止位
 1 = 禁止 ULPWU 唤醒源
 0 = 使能 ULPWU 唤醒源（还必须设置 DSULPEN = 1）
- bit 1 **DSBOR:** 深度休眠 BOR 事件状态位
 1 = 在深度休眠模式期间使能了 DSBOR，并且 V_{DD} 降低到低于 DSBOR 门限值，但仍高于 V_{DSBOR}
 0 = 在深度休眠模式期间禁止了 DSBOR，或者 V_{DD} 并未降低到低于 DSBOR 门限值
- bit 0 **RELEASE:** I/O 引脚状态释放位
 从深度休眠模式唤醒时，I/O 引脚维持它们原先的状态。清零该位将释放 I/O 引脚，并允许其相应的 TRIS 和 LAT 位控制它们的状态。

注 1: 这是最初施加 V_{DD} 时的值。

PIC18F46J50 系列

寄存器 4-3: DSGPR0: 深度休眠持久通用寄存器 0（位于普通存储区，地址 F4Eh）

R/W-xxxx ⁽¹⁾	
深度休眠持久通用位	
bit 7	bit 0

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

bit 7-0 深度休眠持久通用位
即使处于深度休眠模式也保留内容。

注 1: 保持所有寄存器位，除非：不处于深度休眠时，VDDCORE 降低到低于正常的 BOR 门限值；或者，处于深度休眠模式时，使能了专用 DSBOR，并且 VDD 降低到低于 DSBOR 门限值，或者 DSBOR 被使能或禁止，但 VDD 直接下降回接近 VSS。

寄存器 4-4: DSGPR1: 深度休眠持久通用寄存器 1（位于普通存储区，地址 F4Fh）

R/W-xxxx ⁽¹⁾	
深度休眠持久通用位	
bit 7	bit 0

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

bit 7-0 深度休眠持久通用位
即使处于深度休眠模式也保留内容。

注 1: 保持所有寄存器位，除非：不处于深度休眠时：VDDCORE 降低到低于正常的 BOR 门限值；或者，处于深度休眠模式时：使能了专用 DSBOR，并且 VDD 降低到低于 DSBOR 门限值；或者 DSBOR 被使能或禁止，但 VDD 直接下降回接近 VSS。

寄存器 4-5: DSWAKEH: 深度休眠唤醒高字节寄存器 (位于普通存储区, 地址 F4Bh)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DSINT0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-1 **未实现:** 读为 0

bit 0 **DSINT0:** 电平变化中断位
 1 = 电平变化中断在深度休眠期间被置为有效
 0 = 电平变化中断在深度休眠期间未被置为有效

寄存器 4-6: DSWAKEL: 深度休眠唤醒低字节寄存器 (位于普通存储区, 地址 F4Ah)

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1
DSFLT	—	DSULP	DSWDT	DSRTC	DSMCLR	—	DSPOR
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **DSFLT:** 深度休眠故障检测位
 1 = 在深度休眠期间检测到深度休眠故障
 0 = 在深度休眠期间未检测到深度休眠故障

bit 6 **未实现:** 读为 0

bit 5 **DSULP:** 超低功耗唤醒状态位
 1 = 在深度休眠期间发生了超低功耗唤醒事件
 0 = 在深度休眠期间未发生超低功耗唤醒事件

bit 4 **DSWDT:** 深度休眠看门狗定时器超时位
 1 = 深度休眠看门狗定时器在深度休眠期间超时
 0 = 深度休眠看门狗定时器在深度休眠期间未超时

bit 3 **DSRTC:** 实时时钟和日历闹钟位
 1 = 实时时钟 / 日历在深度休眠期间触发了一次闹钟
 0 = 实时时钟 / 日历在深度休眠期间未触发闹钟

bit 2 **DSMCLR:** $\overline{\text{MCLR}}$ 事件位
 1 = $\overline{\text{MCLR}}$ 引脚在深度休眠模式期间被置为有效
 0 = $\overline{\text{MCLR}}$ 引脚在深度休眠模式期间未被置为有效

bit 1 **未实现:** 读为 0

bit 0 **DSPOR:** 上电复位事件位
 1 = VDD 电源 POR 电路曾经工作, 并检测到 POR 事件 ⁽¹⁾
 0 = VDD 电源 POR 电路未工作, 或者曾经工作, 但未检测到 POR 事件

注 1: 不同于该寄存器中的其他位, 该位可以在不处于深度休眠模式时置 1。

PIC18F46J50 系列

4.7 超低功耗唤醒

RA0 上的超低功耗唤醒（ULPWU）功能允许缓慢下降的电压能够产生电平变化中断，同时不消耗很大的电流。

执行以下步骤来使用该功能：

1. 配置可重映射输出引脚，以输出 ULPOUT 信号。
2. 将 INTx 输入电平变化中断功能映射到 ULPOUT 输出功能使用的引脚上。或者在步骤 1 中，将 ULPOUT 配置为输出至 PORTB 电平变化中断引脚。
3. 将 RA0 引脚配置为输出，并将它设为 1，通过此方式对 RA0 上的电容进行充电。
4. 允许在步骤 2 中选择的相应引脚的电平变化中断（通过 PIE 位进行操作）。
5. 通过将 RA0 配置为输入来停止对电容充电。
6. 将 WDTCON 寄存器中的 ULPEN 和 ULPSINK 位置 1，使电容放电。
7. 配置休眠模式。
8. 进入休眠模式。

当 RA0 上的电压下降到低于 V_{IL} 时，将产生中断，从而将器件唤醒并执行下一条指令。

该功能提供了一种定期将器件从休眠模式唤醒的低功耗技术。超时时间取决于 RA0 上 RC 电路的放电时间。

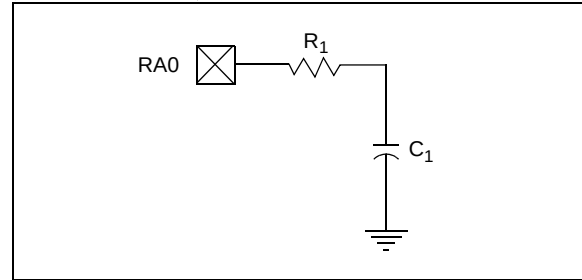
当 ULPWU 模块导致器件从休眠模式唤醒时，WDTCON <ULPLVL> 位会置 1。当 ULPWU 模块导致器件从深度休眠模式唤醒时，DSULP (DSWAKEL<5>) 位会置 1。软件可以在唤醒时检查这些位，以确定唤醒源。此外，在休眠模式下，可重映射输出功能 ULPWU 会将该位的值输出到某个 RPN 引脚，用于从外部检测唤醒事件。

关于初始化 ULPWU 模块的信息，请参见例 4-1。

注： 关于模块相关位的定义，请参见第 27.2 节“看门狗定时器 (WDT)”中的 WDTCON 寄存器，以及 DSWAKEL 寄存器（寄存器 4-6）。

RA0 和外部电容之间的串联电阻为 RA0/AN0/C1INA/ULPWU/RP0 引脚提供过流保护，同时允许使用软件校准超时时间（见图 4-9）。

图 4-9: 串联电阻



可使用一个定时器测量电容的充放电时间。然后调节充电时间，以提供所需的中断延时。该技术将对温度、电压和元件精度的影响进行补偿。ULPWU 外设还可以配置为简单的可编程低压检测（Low-Voltage Detect, LVD）或温度传感器。

注： 更多信息，请参见 AN879, “Using the Microchip Ultra Low-Power Wake-up Module” 应用笔记（DS00879）。

例 4-1: 超低功耗唤醒初始化

```

//*****
//Configure a remappable output pin with interrupt capability
//for ULPWU function (RP21 => RD4/INT1 in this example)
//*****
RPOR21 = 13;// ULPWU function mapped to RP21/RD4
RPINR1 = 21;// INT1 mapped to RP21 (RD4)

//*****
//Charge the capacitor on RA0
//*****
TRISAbits.TRISA0 = 0;
LATAbits.LATA0 = 1;
for(i = 0; i < 10000; i++) Nop();

//*****
//Stop Charging the capacitor on RA0
//*****
TRISAbits.TRISA0 = 1;

//*****
//Enable the Ultra Low Power Wakeup module
//and allow capacitor discharge
//*****
WDTCONbits.ULPEN = 1;
WDTCONbits.ULPSINK = 1;

//*****
//Enable Interrupt for ULPW
//*****
//For Sleep
//(assign the ULPOUT signal in the PPS module to a pin
//which has also been assigned an interrupt capability,
//such as INT1)
INTCON3bits.INT1IF = 0;
INTCON3bits.INT1IE = 1;

//*****
//Configure Sleep Mode
//*****
//For Sleep
OSCCONbits.IDLEN = 0;

//For Deep Sleep
OSCCONbits.IDLEN = 0; // enable deep sleep
DSCONHbits.DSEN = 1; // Note: must be set just before executing Sleep();
//*****
//Enter Sleep Mode
//*****
Sleep();
// for sleep, execution will resume here
// for deep sleep, execution will restart at reset vector (use WDTCONbits.DS to detect)

```

PIC18F46J50 系列

注

5.0 复位

PIC18F46J50 系列器件有以下几种不同类型的复位：

- 上电复位（POR）
- 正常工作期间的 MCLR 复位
- 功耗管理模式下的 MCLR 复位
- 看门狗定时器（WDT）复位（执行程序期间）
- 配置不匹配（Configuration Mismatch, CM）
- 欠压复位（BOR）
- RESET 指令
- 堆栈满复位
- 堆栈下溢复位
- 深度休眠复位

本节讨论了由 MCLR、POR 和 BOR 产生的复位，并涉及各种起振定时器的工作方式。

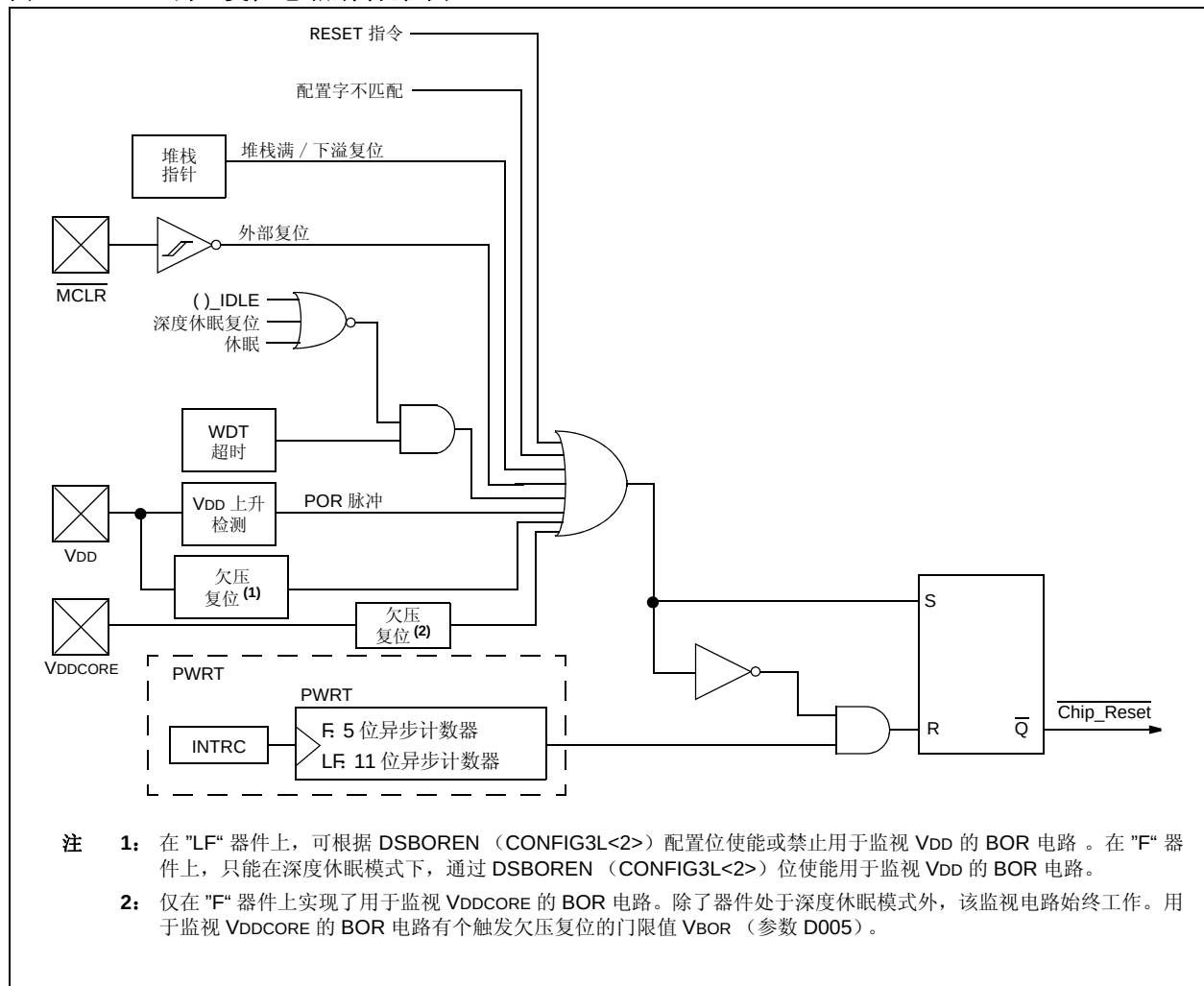
关于 WDT 复位的信息，请参见第 27.2 节“看门狗定时器（WDT）”。关于堆栈复位事件的信息，请参见第 6.1.4.4 节“堆栈满和下溢复位”；关于深度休眠模式的信息，请参见第 4.6 节“深度休眠模式”。

图 5-1 给出了片上复位电路的简化框图。

5.1 RCON 寄存器

通过 RCON 寄存器（寄存器 5-1）跟踪器件复位事件。该寄存器的低 5 位表明是否已经发生了特定的复位事件。在大多数情况下，只能通过事件将这些位置 1，而且必须在事件发生后由应用程序将它们清零。需要读取所有这些标志位的状态来确定刚发生的复位的类型。在第 5.7 节“寄存器的复位状态”中对此进行了更详细的说明。

图 5-1: 片上复位电路的简化框图



PIC18F46J50 系列

寄存器 5-1: RCON: 复位控制寄存器 (位于快速操作存储区, 地址 FD0h)

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	CM	RI	TO	PD	POR	BOR
bit 7							bit 0

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 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	CM: 配置不匹配标志位 1 = 未发生配置不匹配复位 0 = 发生了配置不匹配复位 (发生配置不匹配复位后必须用软件置 1)
bit 4	RI: RESET 指令标志位 1 = 未执行 RESET 指令 (只能由固件置 1) 0 = 执行了 RESET 指令, 导致器件复位 (发生欠压复位后必须用软件置 1)
bit 3	TO: 看门狗超时标志位 1 = 通过上电、CLRWDT 指令或 SLEEP 指令置 1 0 = 发生了 WDT 超时
bit 2	PD: 掉电检测标志位 1 = 通过上电或 CLRWDT 指令置 1 0 = 通过执行 SLEEP 指令置 1
bit 1	POR: 上电复位状态位 1 = 未发生上电复位 (只能由固件置 1) 0 = 发生了上电复位 (发生上电复位后必须用软件置 1)
bit 0	BOR: 欠压复位状态位 1 = 未发生欠压复位 (只能由固件置 1) 0 = 发生了欠压复位 (发生欠压复位后必须用软件置 1)

注	1: 建议在检测到上电复位后, 将 POR 位置 1, 以便继续检测后续的上电复位。
	2: 如果禁止了片上稳压器, BOR 则总是保持为 0。更多信息, 请参见第 5.4.1 节 “检测 BOR”。
	3: 当 BOR 为 0 并且 POR 为 1 时 (假定在上电复位之后立即用软件将 POR 设为 1), 可以说已发生了欠压复位。

5.2 主复位 ($\overline{\text{MCLR}}$)

主复位 ($\overline{\text{MCLR}}$) 引脚提供了触发器件外部硬复位的方法。将该引脚拉低可以产生复位信号。PIC18 扩展单片机器件在 $\overline{\text{MCLR}}$ 复位路径中有一个噪声滤波器，该滤波器可以检测并滤除小的干扰脉冲。

任何内部复位，包括 WDT 复位，均不能将 $\overline{\text{MCLR}}$ 引脚驱动为低电平。

5.3 上电复位 (POR)

只要当 VDD 上升到高于某个门限时，就会在片上产生 POR 条件。这使得 VDD 达到满足器件正常工作的数值时，器件会以初始化状态启动。

为了利用 POR 电路，需要将 $\overline{\text{MCLR}}$ 引脚通过一个电阻（阻值范围为 1 k Ω 到 10 k Ω ）连接到 VDD。这样可以省去产生 POR 延时通常所需的外部 RC 元件。

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

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

5.4 欠压复位 (BOR)

PIC18F46J50 系列中的“F”器件具有两种类型的 BOR 电路：一种监视 VDDCORE，另一种监视 VDD。每次只能有一个 BOR 电路在工作。处于正常运行模式、空闲或常规休眠模式时，监视 VDDCORE 的 BOR 电路工作，如果 VDDCORE 降低到低于 VBOR（参数 D005），该电路将导致器件保持在 BOR 状态。当 VDDCORE 回升到高于 VBOR 时，器件将保持在复位状态，直到上电延时定时器延时结束，计时周期为 TPWRT（参数 33）。

在深度休眠操作期间，片上内核稳压器会被禁止，并允许 VDDCORE 降至 VSS。如果通过 DSBOREN 位 (CONFIG3L<2> = 1) 使能了深度休眠 BOR 电路，它将会监视 VDD。如果 VDD 降低到低于 VDSBOR 门限值，器件将保持在与 POR 类似的复位状态。所有寄存器将恢复为它们的上电复位值，并且 DSGPR0 和 DSGPR1 保持寄存器的内容将丢失。此外，如果在深度休眠期间，有任何 I/O 引脚已被配置为输出，这些引脚将为三态，并且器件将不再保持在深度休眠模式。一旦 VDD 电压恢复至 VDSBOR 门限值以上，以及内核稳压器使得

VDDCORE 电压高于 VBOR，器件将开始重新正常执行代码，但 WDTCON 寄存器中的 DS 位不会置 1。器件的行为与给器件的所有部分硬上电类似。

在“LF”器件上（PIC18LF46J50 除外），由于内部内核稳压器禁止，VDDCORE BOR 电路始终禁止。虽说不能监视 VDDCORE，本系列中的 PIC18LF 器件仍能使用 VDD BOR 电路来监视 VDD 低于 VDSBOR 门限值的情况。通过将 DSBOREN 设置为 0 可禁止 VDD BOR 电路。

在“LF”器件上，当 DSBOREN = 1 时，或在“F”器件上，器件处于深度休眠且 DSBOREN = 1 时使能 VDD BOR 电路。VDD BOR 电路使能后，正常工作期间的功耗极低（典型值为 40 nA），电压比 VDD 高约 2.3V。当 VDD BOR 电路使能时，VDD 跌落到 DSBOR 门限值以下会使器件开始消耗更多电流（增加量的典型值为 50 μ A），因为电路的内部逻辑此时上电。为了更准确地检测 VDD 电平，势必会需要更高的电流。然后，VDD 降至在 VDSBOR 门限值以下前，器件不会进入复位状态。

5.4.1 检测 BOR

在发生任何 VDDCORE BOR 或 POR 事件时，BOR 位总是复位为 0。因此只通过读 BOR 位的状态很难确定是否发生过欠压复位事件。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，POR 位被立即用软件复位为 1。如果 BOR 为 0 同时 POR 为 1，那么就可以断定已经发生了欠压复位事件。

如果禁止了稳压器（LF 器件），也会禁止 VDDCORE BOR 功能。在这种情况下，不能使用 BOR 位来确定欠压复位事件。上电复位事件仍会将 BOR 位清零。

PIC18F46J50 系列

5.5 配置不匹配（CM）

配置不匹配（CM）复位旨在用于检测随机存储器损坏事件，并尝试从事件中恢复。这些事件包括静电放电（ESD）事件，该事件会导致整个器件中大范围的单个位单元内容改变，并导致灾难性的故障。

在 PIC18FXXJ 闪存器件中，在工作期间会持续监视器件配置寄存器（位于配置存储空间中），即将它们的值和与之配对的影子寄存器进行比较。如果在两组寄存器之间检测到不匹配，则会自动发生 CM 复位。这些事件由 CM 位（RCON<5>）捕捉。每当发生 CM 事件时，该位的状态就会被设为 0；任何其他复位事件均不能改变它。

CM 复位的行为类似于 $\overline{\text{MCLR}}$ 、RESET 指令、WDT 超时或堆栈事件复位。类似于所有硬复位和电源复位事件，在器件重启时，会从程序存储器中的闪存配置字重新装入器件配置字。

5.6 上电延时定时器（PWRT）

PIC18F46J50 系列器件具有片上 PWRT，用于帮助稳定 POR 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。

PIC18F46J50 系列器件的上电延时定时器（PWRT）是一个 5 位计数器，它使用 INTRC 时钟源作为时钟输入。该定时器可产生大约 $32 \times 32 \mu\text{s} = 1 \text{ ms}$ 的时间间隔。PWRT 计数期间，器件保持在复位状态。

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

5.6.1 延时时序

在 POR 脉冲被清除后，启动 PWRT 延时。总延迟时间将取决于 PWRT 的状态。图 5-2、图 5-3、图 5-4 和图 5-5 都说明了在使能 PWRT 时的延时时序。

由于延时是由 POR 脉冲触发的，因此如果 $\overline{\text{MCLR}}$ 保持足够长时间的低电平，PWRT 延时将结束。将 $\overline{\text{MCLR}}$ 升为高电平时，如果有时钟源可用，则将立即开始执行（图 5-4）。这对于测试或同步多个并行工作的 PIC18F 器件是非常有用的。

图 5-2: 上电延时时序（ $\overline{\text{MCLR}}$ 连接到 VDD，VDD 电压上升时间 < TPWRT）

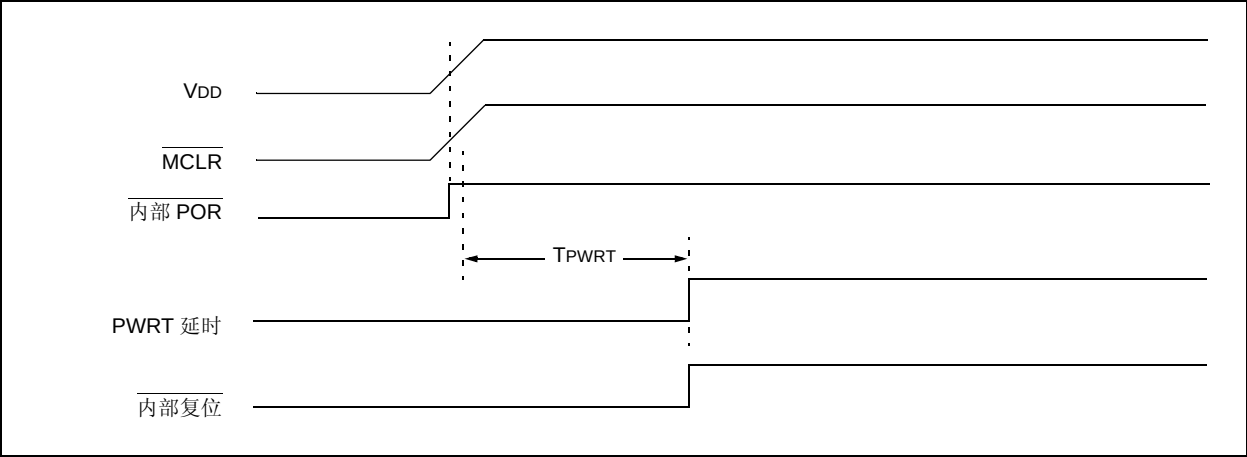


图 5-3: 上电延时时序 ($\overline{\text{MCLR}}$ 未连接到 VDD): 情形 1

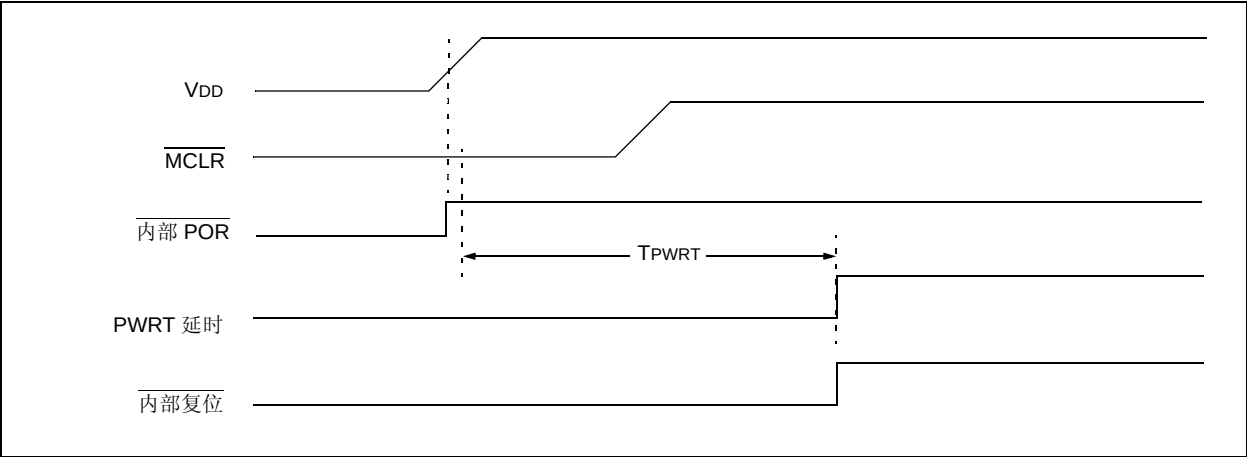


图 5-4: 上电延时时序 ($\overline{\text{MCLR}}$ 未连接到 VDD): 情形 2

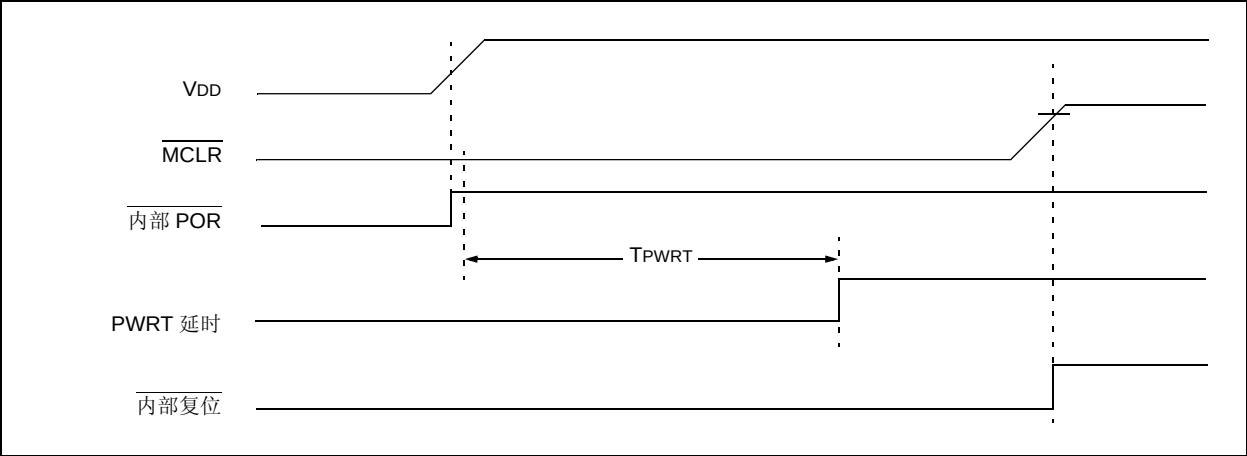
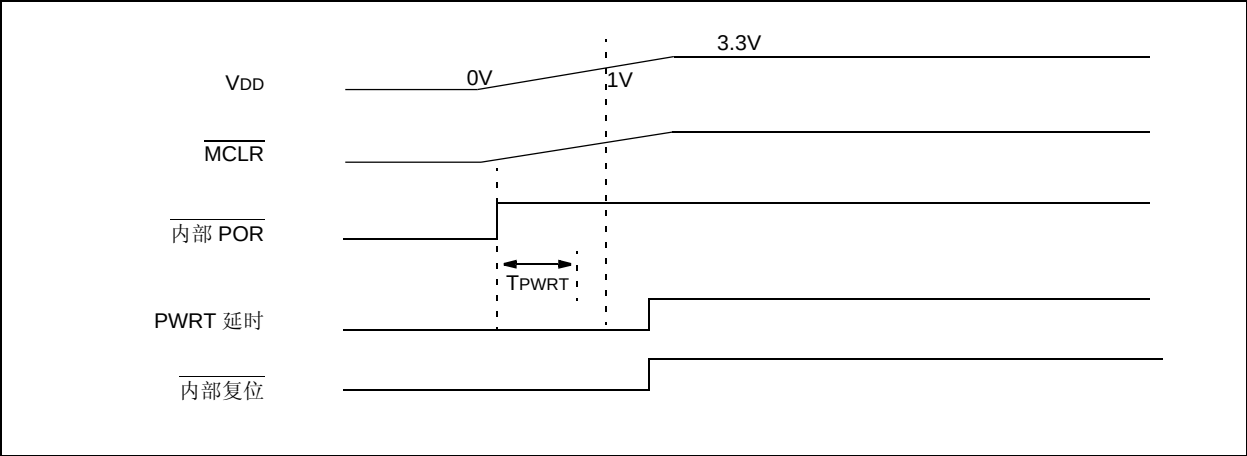


图 5-5: 缓慢上升时间 ($\overline{\text{MCLR}}$ 连接到 VDD , VDD 电压上升时间 $> \text{TPWRT}$)



PIC18F46J50 系列

5.7 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”。

大多数寄存器不受 WDT 唤醒的影响，因为这被视为是正常工作的恢复。如表 5-1 所示，RCON 寄存器中的状

态位（ $\overline{\text{CM}}$ 、 $\overline{\text{RI}}$ 、 $\overline{\text{TO}}$ 、 $\overline{\text{PD}}$ 、 $\overline{\text{POR}}$ 和 $\overline{\text{BOR}}$ ）在不同的复位情形下会分别被置 1 或清零。可在软件中使用这些位判断复位的性质。

表 5-2 描述了所有特殊功能寄存器的复位状态。可以将这些复位状态分类为 POR 和 BOR、MCLR 和 WDT 复位，以及 WDT 唤醒。

表 5-1: RCON 寄存器的状态位、含义以及初始化状态

条件	程序计数器 ⁽¹⁾	RCON 寄存器						STKPTR 寄存器	
		$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	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
功耗管理运行模式下的 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下的 MCLR 复位	0000h	u	u	1	0	u	u	u	u
全功耗执行期间的 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下的 WDT 超时	0000h	u	u	0	u	u	u	u	u
功耗管理空闲或休眠模式下的 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，PC 装入中断向量（0008h 或 0018h）。

表 5-2: 所有寄存器的初始化状态

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
TOSU	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu ⁽¹⁾
TOSH	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu ⁽¹⁾
TOSL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu ⁽¹⁾
STKPTR	PIC18F2XJ50	PIC18F4XJ50	00-0 0000	uu-0 0000	uu-u uuuu ⁽¹⁾
PCLATU	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	PC + 2 ⁽²⁾
TBLPTRU	PIC18F2XJ50	PIC18F4XJ50	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F2XJ50	PIC18F4XJ50	0000 000x	0000 000u	uuuu uuuu ⁽³⁾
INTCON2	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu ⁽³⁾
INTCON3	PIC18F2XJ50	PIC18F4XJ50	1100 0000	1100 0000	uuuu uuuu ⁽³⁾
INDF0	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
POSTINC0	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
POSTDEC0	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
PREINC0	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
PLUSW0	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
FSR0H	PIC18F2XJ50	PIC18F4XJ50	---- 0000	---- 0000	---- uuuu
FSR0L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
POSTINC1	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
POSTDEC1	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
PREINC1	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
PLUSW1	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
FSR1H	PIC18F2XJ50	PIC18F4XJ50	---- 0000	---- 0000	---- uuuu
FSR1L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F2XJ50	PIC18F4XJ50	---- 0000	---- 0000	---- uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
- 5: 在 PIC18F2XJ50 器件上未实现。
- 6: 在 “LF” 器件上未实现。

PIC18F46J50 系列

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
INDF2	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
POSTINC2	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
POSTDEC2	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
PREINC2	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
PLUSW2	PIC18F2XJ50	PIC18F4XJ50	N/A	N/A	N/A
FSR2H	PIC18F2XJ50	PIC18F4XJ50	---- 0000	---- 0000	---- uuuu
FSR2L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	PIC18F2XJ50	PIC18F4XJ50	---x xxxx	---u uuuu	---u uuuu
TMR0H	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TMR0L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F2XJ50	PIC18F4XJ50	0110 q100	0110 q100	uuuu q1uu
CM1CON	PIC18F2XJ50	PIC18F4XJ50	0001 1111	0001 1111	uuuu uuuu
CM2CON	PIC18F2XJ50	PIC18F4XJ50	0001 1111	0001 1111	uuuu uuuu
RCON ⁽⁴⁾	PIC18F2XJ50	PIC18F4XJ50	0-11 11qq	0-qq qquu	u-qq qquu
TMR1H	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	PIC18F2XJ50	PIC18F4XJ50	0000 0000	uuuu uuuu	uuuu uuuu
TMR2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PR2	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
T2CON	PIC18F2XJ50	PIC18F4XJ50	-000 0000	-000 0000	-uuu uuuu
SSP1BUF	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP1ADD	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
SSP1MSK	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
SSP1STAT	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
SSP1CON1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
SSP1CON2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
ADRESH	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
ADCON1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
WDTCON	PIC18F2XJ50	PIC18F4XJ50	1qq- q000	1qq- 0000	uqq- uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
- 5: 在 PIC18F2XJ50 器件上未实现。
- 6: 在 “LF” 器件上未实现。

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
PSTR1CON	PIC18F2XJ50	PIC18F4XJ50	00-0 0001	00-0 0001	uu-u uuuu
ECCP1AS	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
ECCP1DEL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
CCPR1H	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PSTR2CON	PIC18F2XJ50	PIC18F4XJ50	00-0 0001	00-0 0001	uu-u uuuu
ECCP2AS	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
ECCP2DEL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
CCPR2H	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
CTMUCONH	PIC18F2XJ50	PIC18F4XJ50	0-00 000-	0-00 000-	u-uu uuu-
CTMUCONL	PIC18F2XJ50	PIC18F4XJ50	0000 00xx	0000 00xx	uuuu uuuu
CTMUICON	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
SPBRG1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
RCREG1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TXREG1	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	0000 0000	uuuu uuuu
TXSTA1	PIC18F2XJ50	PIC18F4XJ50	0000 0010	0000 0010	uuuu uuuu
RCSTA1	PIC18F2XJ50	PIC18F4XJ50	0000 000x	0000 000x	uuuu uuuu
SPBRG2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
RCREG2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TXREG2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TXSTA2	PIC18F2XJ50	PIC18F4XJ50	0000 0010	0000 0010	uuuu uuuu
EECON2	PIC18F2XJ50	PIC18F4XJ50	----	----	----
EECON1	PIC18F2XJ50	PIC18F4XJ50	--00 x00-	--00 q00-	--00 u00-
IPR3	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
PIR3	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
PIE3	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
IPR2	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
PIR2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
PIE2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
- 5: 在 PIC18F2XJ50 器件上未实现。
- 6: 在 “LF” 器件上未实现。

PIC18F46J50 系列

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
IPR1	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
PIR1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
PIE1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
RCSTA2	PIC18F2XJ50	PIC18F4XJ50	0000 000x	0000 000x	uuuu uuuu
OSCTUNE	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
T1GCON	PIC18F2XJ50	PIC18F4XJ50	0000 0x00	0000 0x00	uuuu uxuu
RTCVALH	PIC18F2XJ50	PIC18F4XJ50	0xxx xxxx	0uuu uuuu	0uuu uuuu
RTCVALL	PIC18F2XJ50	PIC18F4XJ50	0xxx xxxx	0uuu uuuu	0uuu uuuu
T3GCON	PIC18F2XJ50	PIC18F4XJ50	0000 0x00	uuuu uxuu	uuuu uxuu
TRISE ⁽⁵⁾	—	PIC18F4XJ50	---- -111	---- -111	---- -uuu
TRISD ⁽⁵⁾	—	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F2XJ50	PIC18F4XJ50	11-- -111	11-- -111	uu-- -uuu
TRISB	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
TRISA	PIC18F2XJ50	PIC18F4XJ50	111- 1111	111- 1111	uuu- uuuu
ALRMCFG	PIC18F2XJ50	PIC18F4XJ50	0000 0000	uuuu uuuu	uuuu uuuu
ALMRPT	PIC18F2XJ50	PIC18F4XJ50	0000 0000	uuuu uuuu	uuuu uuuu
ALRMVALH	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
ALRMVALL	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATE ⁽⁵⁾	—	PIC18F4XJ50	---- -xxx	---- -uuu	---- -uuu
LATD ⁽⁵⁾	—	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F2XJ50	PIC18F4XJ50	xx-- -xxx	uu-- -uuu	uu-- -uuu
LATB	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA	PIC18F2XJ50	PIC18F4XJ50	xxx- xxxx	uuu- uuuu	uuu- uuuu
DMACON1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
DMACON2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
HLVDCON	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PORTE ⁽⁵⁾	—	PIC18F4XJ50	00-- -xxx	uu-- -uuu	uu-- -uuu
PORTD ⁽⁵⁾	—	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F2XJ50	PIC18F4XJ50	xxxx -xxx	uuuu -uuu	uuuu -uuu
PORTB	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	PIC18F2XJ50	PIC18F4XJ50	xxx- xxxx	uuu- uuuu	uuu- uuuu
SPBRGH1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注
- 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
 - 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
 - 5: 在 PIC18F2XJ50 器件上未实现。
 - 6: 在 “LF” 器件上未实现。

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
BAUDCON1	PIC18F2XJ50	PIC18F4XJ50	0100 0-00	0100 0-00	uuuu u-uu
SPBRGH2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
BAUDCON2	PIC18F2XJ50	PIC18F4XJ50	0100 0-00	0100 0-00	uuuu u-uu
TMR3H	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F2XJ50	PIC18F4XJ50	0000 0000	uuuu uuuu	uuuu uuuu
TMR4	PIC18F2XJ50	PIC18F4XJ50	0000 0000	uuuu uuuu	uuuu uuuu
PR4	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
T4CON	PIC18F2XJ50	PIC18F4XJ50	-000 0000	-000 0000	-uuu uuuu
SSP2BUF	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP2ADD	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
SSP2MSK	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
SSP2STAT	PIC18F2XJ50	PIC18F4XJ50	1111 1111	1111 1111	uuuu uuuu
SSP2CON1	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
CMSTAT	PIC18F2XJ50	PIC18F4XJ50	---- --11	---- --11	---- --uu
PMADDRH ⁽⁵⁾	—	PIC18F4XJ50	-000 0000	-000 0000	-uuu uuuu
PMDOUT1H ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMADDRL ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMDOUT1L ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMDIN1H ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMDIN1L ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TXADDRL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TXADDRH	PIC18F2XJ50	PIC18F4XJ50	---- 0000	---- 0000	---- uuuu
RXADDRL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
RXADDRH	PIC18F2XJ50	PIC18F4XJ50	---- 0000	---- 0000	---- uuuu
DMABCL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
DMABCH	PIC18F2XJ50	PIC18F4XJ50	---- --00	---- --00	---- --uu
UCON	PIC18F2XJ50	PIC18F4XJ50	-0x0 000-	-0x0 000-	-uuu uu- -
USTAT	PIC18F2XJ50	PIC18F4XJ50	-xxx xxx-	-xxx xxx-	-uuu uu- -
UEIR	PIC18F2XJ50	PIC18F4XJ50	0--0 0000	0--0 0000	u--u uuuu
UIR	PIC18F2XJ50	PIC18F4XJ50	-000 0000	-000 0000	-uuu uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
- 5: 在 PIC18F2XJ50 器件上未实现。
- 6: 在 “LF” 器件上未实现。

PIC18F46J50 系列

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
UFRMH	PIC18F2XJ50	PIC18F4XJ50	---- -xxx	---- -xxx	---- -uuu
UFRML	PIC18F2XJ50	PIC18F4XJ50	xxxx xxxx	xxxxx xxxx	uuuu uuuu
PMCONH ⁽⁵⁾	—	PIC18F4XJ50	0--0 0000	0--0 0000	u--u uuuu
PMCONL ⁽⁵⁾	—	PIC18F4XJ50	000- 0000	000- 0000	uuu- uuuu
PMMODEH ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMMODEL ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMDOUT2H ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMDOUT2L ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMDIN2H ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMDIN2L ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMEH ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMEL ⁽⁵⁾	—	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
PMSTATH	—	PIC18F4XJ50	00-- 0000	00-- 0000	uu-- uuuu
PMSTATL	—	PIC18F4XJ50	10-- 1111	10-- 1111	uu-- uuuu
CVRCON	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
TCLKCON	PIC18F2XJ50	PIC18F4XJ50	--0 --00	--0 --uu	--u --uu
DSGPR1 ⁽⁶⁾	PIC18F2XJ50	PIC18F4XJ50	uuuu uuuu	uuuu uuuu	uuuu uuuu
DSGPR0 ⁽⁶⁾	PIC18F2XJ50	PIC18F4XJ50	uuuu uuuu	uuuu uuuu	uuuu uuuu
DSCONH ⁽⁶⁾	PIC18F2XJ50	PIC18F4XJ50	0--- -000	0--- -uuu	u--- -uuu
DSCONL ⁽⁶⁾	PIC18F2XJ50	PIC18F4XJ50	---- -000	---- -u00	---- -uuu
DSWAKEH ⁽⁶⁾	PIC18F2XJ50	PIC18F4XJ50	---- ---0	---- ---0	---- ---u
DSWAKEL ⁽⁶⁾	PIC18F2XJ50	PIC18F4XJ50	0-00 00-1	0-00 00-0	u-uu uu-u
ANCON1	PIC18F2XJ50	PIC18F4XJ50	00-0 0000	00-0 0000	uu-u uuuu
ANCON0	PIC18F2XJ50	PIC18F4XJ50	0000 0000	0000 0000	uuuu uuuu
ODCON1	PIC18F2XJ50	PIC18F4XJ50	---- --00	---- --uu	---- --uu
ODCON2	PIC18F2XJ50	PIC18F4XJ50	---- --00	---- --uu	---- --uu
ODCON3	PIC18F2XJ50	PIC18F4XJ50	---- --00	---- --uu	---- --uu
RTCCFG	PIC18F2XJ50	PIC18F4XJ50	0-00 0000	u-uu uuuu	u-uu uuuu
RTCCAL	PIC18F2XJ50	PIC18F4XJ50	0000 0000	uuuu uuuu	uuuu uuuu
REFOCON	PIC18F2XJ50	PIC18F4XJ50	0-00 0000	0-00 0000	u-uu uuuu
PADCFG1	PIC18F2XJ50	PIC18F4XJ50	---- -000	---- -000	---- -uuu
UCFG	PIC18F2XJ50	PIC18F4XJ50	00-0 0000	00-0 0000	uu-u uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注
- 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
 - 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
 - 5: 在 PIC18F2XJ50 器件上未实现。
 - 6: 在 “LF” 器件上未实现。

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
UADDR	PIC18F2XJ50	PIC18F4XJ50	-000 0000	-uuu uuuu	-uuu uuuu
UEIE	PIC18F2XJ50	PIC18F4XJ50	0--0 0000	0--0 0000	u--u uuuu
UIE	PIC18F2XJ50	PIC18F4XJ50	-000 0000	-000 0000	-uuu uuuu
UEP15	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP14	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP13	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP12	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP11	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP10	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP9	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP8	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP7	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP6	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP5	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP4	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP3	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP2	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP1	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
UEP0	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
PPSCON	PIC18F2XJ50	PIC18F4XJ50	---- --0	---- --0	---- --u
RPINR24	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR23	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR22	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR21	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR17	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR16	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR13	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR12	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR8	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR7	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR6	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR4	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
- 5: 在 PIC18F2XJ50 器件上未实现。
- 6: 在 “LF” 器件上未实现。

PIC18F46J50 系列

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位, 从深度休眠唤醒	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或 中断唤醒器件
RPINR3	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR2	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPINR1	PIC18F2XJ50	PIC18F4XJ50	---1 1111	---1 1111	---u uuuu
RPOR24	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR23	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR22	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR21	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR20	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR19	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR18	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR17	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR13	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR12	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR11	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR10	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR9	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR8	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR7	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR6	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR5	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR4	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR3	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR2	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR1	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu
RPOR0	PIC18F2XJ50	PIC18F4XJ50	---0 0000	---0 0000	---u uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 值取决于具体条件。

- 注 1: 当器件被中断唤醒且 GIEH (若为低优先级, 则为 GIEL) 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存储单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位被置 1 时, PC 装入中断向量 (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: 具体条件下的复位值, 请参见表 5-1。
- 5: 在 PIC18F2XJ50 器件上未实现。
- 6: 在 “LF” 器件上未实现。

6.0 存储器构成

PIC18 闪存单片机有两种类型的存储器：

- 程序存储器
- 数据 RAM

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

第 7.0 节“闪存程序存储器”提供了关于闪存程序存储器操作的更多信息。

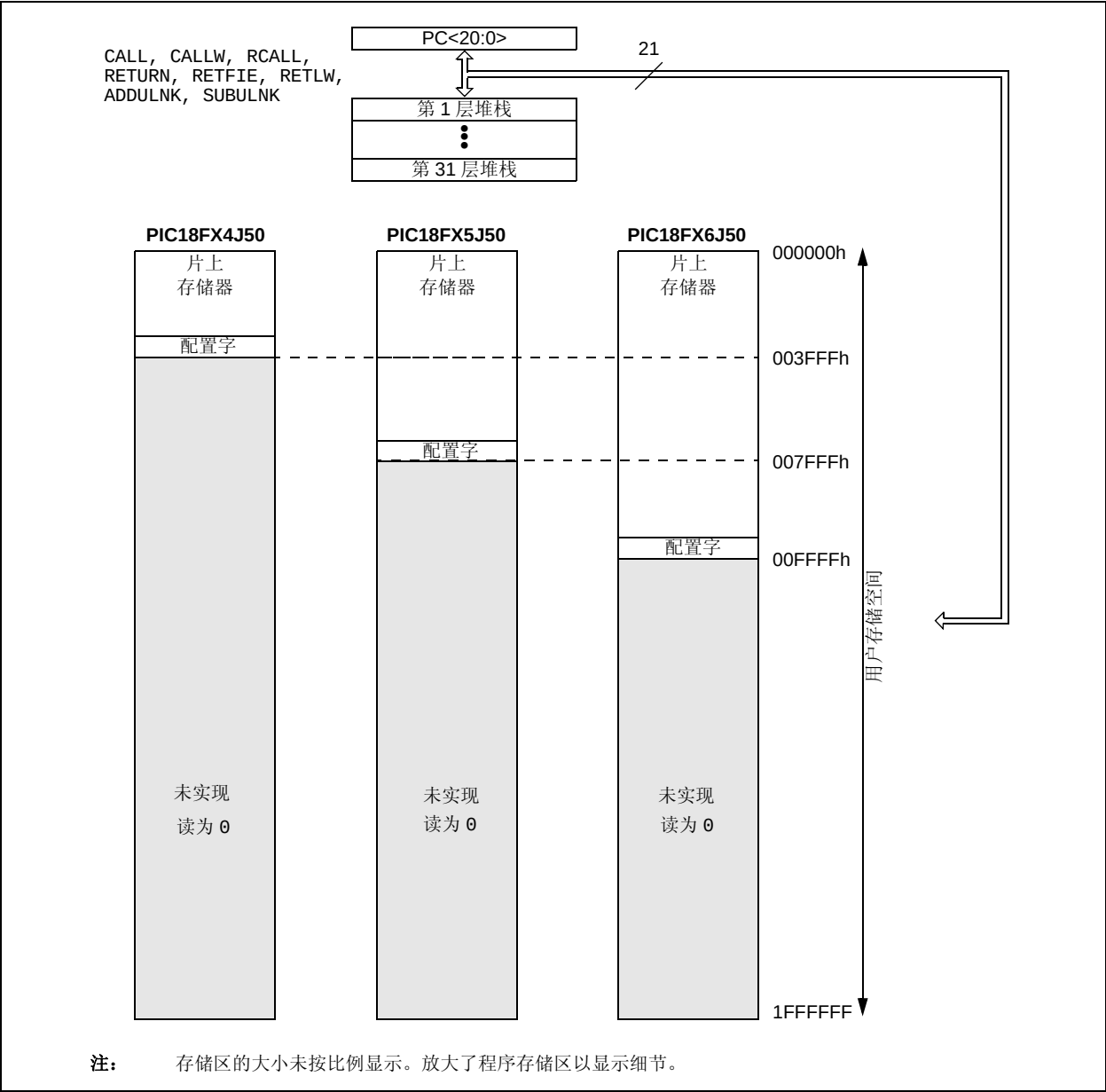
6.1 程序存储器构成

PIC18 单片机具有 21 位程序计数器，可以对 2 MB 的程序存储空间进行寻址。访问物理实现存储器的上边界和 2 MB 地址之间的存储单元会返回全 0（NOP 指令）。

PIC18F46J50 系列提供了各种容量的片上闪存程序存储器，从 16 KB（最多 8,192 条单字指令）到 64 KB（32,768 条单字指令）。

图 6-1 给出了系列中各器件的程序存储器映射。

图 6-1: PIC18F46J50 系列器件的存储器映射



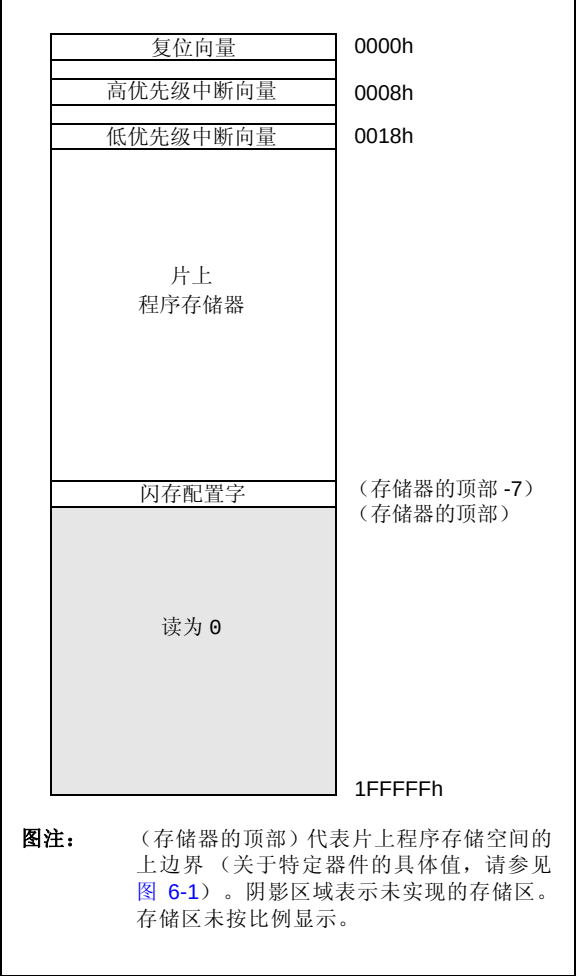
PIC18F46J50 系列

6.1.1 存储器硬编码向量

所有的 PIC18 器件在它们的程序存储空间内共有 3 个硬编码的返回向量。复位向量地址是在器件发生任何复位时程序计数器返回到的默认值；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。图 6-2 给出了它们在程序存储器映射中的相对位置。

图 6-2: PIC18F46J50 系列器件的硬编码向量和配置字单元



6.1.2 闪存配置字

由于 PIC18F46J50 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，该配置信息被复制到配置寄存器。

配置字按数字编号顺序存储在程序存储单元中，从最低地址开始存放 CONFIG1 的低字节，到 CONFIG4 的高字节结束。

表 6-1 给出了 PIC18F46J50 系列器件的闪存配置字的实际地址。图 6-2 给出了闪存配置字以及其他存储器向量在存储器映射中的位置。

第 27.1 节“配置位”中提供了有关器件配置字的更多详细信息。

表 6-1: PIC18F46J50 系列器件的闪存配置字

器件	程序存储器 (KB)	配置字地址
PIC18F24J50	16	3FF8h 到 3FFFh
PIC18F44J50		
PIC18F25J50	32	7FF8h 到 7FFFh
PIC18F45J50		
PIC18F26J50	64	FFF8h 到 FFFFh
PIC18F46J50		

6.1.3 程序计数器

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

PCLATH 和 PCLATU 的内容通过执行写 PCL 的任何操作被传送到程序计数器。同样，程序计数器的两个高字节通过读 PCL 的操作被传送到 PCLATH 和 PCLATU。这对于 PC 的计算偏移量很有用（见第 6.1.6.1 节“计算 GOTO”）。

PC 是按字节寻址程序存储器的。为了防止 PC 不能与字节指令对齐，需要将 PCL 的最低有效位（Least Significant bit，LSb）固定取值为 0。PC 每次加 2 来寻址存储器中的顺序指令。

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

6.1.4 返回地址堆栈

返回地址堆栈允许最多 31 个程序调用和中断的任意组合。当执行 CALL、RCALL 指令或响应中断时，PC 值会被压入堆栈。而在执行 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 值会从堆栈弹出。PCLATU 和 PCLATH 不受 RETURN 或 CALL 指令的影响。

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

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

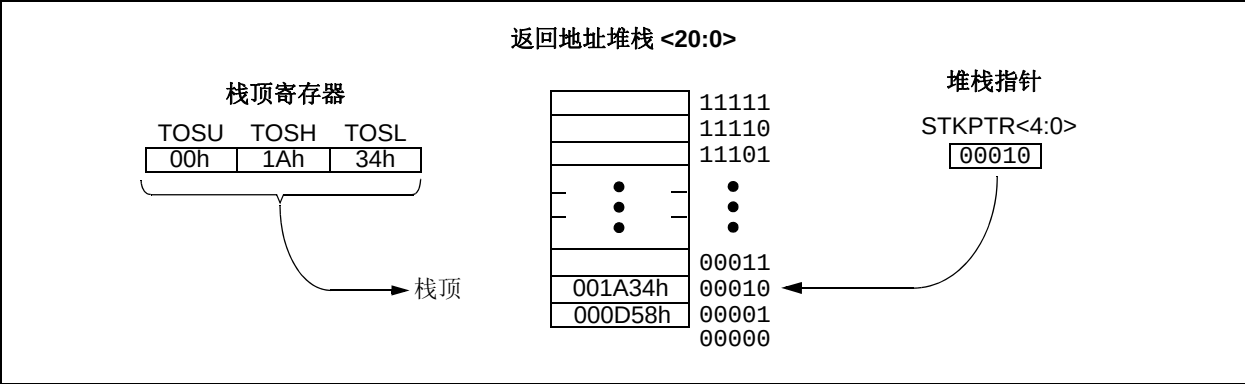
所有复位后，堆栈指针均会初始化为 00000。堆栈指针值 00000 不指向任何 RAM 单元；它仅仅是一个复位值。状态位表明堆栈是已满、上溢还是下溢。

6.1.4.1 访问栈顶

只可读写返回地址堆栈的栈顶（Top-of-Stack，TOS）。有三个寄存器 TOSU:TOSH:TOSL 用于保存由 STKPTR 寄存器所指向的堆栈单元的内容（图 6-3）。这可以让用户在必要时实现软件堆栈。在 CALL、RCALL 或响应中断（如果使能了扩展指令集，则还包括 ADDULNK 和 SUBULNK 指令）后，软件可以通过读取 TOSU:TOSH:TOSL 寄存器来读取压入堆栈的值。这些值可以被存放到由用户定义的软件堆栈。返回时，软件将这些值存回 TOSU:TOSH:TOSL 并执行返回。

为防止意外的堆栈操作，访问堆栈时用户必须禁止全局中断允许位。

图 6-3: 返回地址堆栈和相关的寄存器



PIC18F46J50 系列

6.1.4.2 返回堆栈指针（STKPTR）

STKPTR 寄存器（寄存器 6-1）包含堆栈指针值、STKFUL（堆栈满）和 STKUNF（堆栈下溢）状态位。堆栈指针值可为 0 到 31 范围内的值。向堆栈压入值前，堆栈指针加 1；而从堆栈弹出值后，堆栈指针减 1。复位时，堆栈指针值为零。用户可以读写堆栈指针的值。实时操作系统（Real-Time Operating System, RTOS）可以利用此特性对返回堆栈进行维护。

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

由堆栈溢出复位使能（STVREN）配置位的状态决定堆栈满时将执行的操作。

关于器件配置位的说明，请参见第 27.1 节“配置位”。

如果 STVREN 置 1（默认），第 31 次压栈将把（PC+2）值压入堆栈，从而将 STKFUL 位置 1 并复位器件。STKFUL 位将保持置 1，而堆栈指针将被清零。

如果 STVREN 清零，第 31 次压栈时 STKFUL 位将会置 1，堆栈指针值递增到 31。任何其他压栈操作都不会覆盖第 31 次进栈的值，并且 STKPTR 将保持为 31。

寄存器 6-1: STKPTR: 堆栈指针寄存器（位于快速操作存储区，地址 FFCh）

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

图注:	C = 可清零位
R = 可读位	W = 可写位
-n = POR 时的值	U = 未实现位，读为 0
	1 = 置 1
	0 = 清零
	x = 未知

bit 7 STKFUL: 堆栈满标志位⁽¹⁾

1 = 堆栈满或溢出
0 = 堆栈未满也未溢出

bit 6 STKUNF: 堆栈下溢标志位⁽¹⁾

1 = 发生了堆栈下溢
0 = 未发生堆栈下溢

bit 5 未实现: 读为 0

bit 4-0 SP<4:0>: 堆栈指针地址位

注 1: 通过用户软件或 POR 清零 bit 7 和 bit 6。

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零。STKUNF 位将保持置 1，直到由软件清零或发生 POR 为止。

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

6.1.4.3 PUSH 和 POP 指令

由于栈项是可以读写的，因此将值压入堆栈或从堆栈弹出而不影响程序的正常执行是必需的。PIC18 指令集包含两条指令 PUSH 和 POP，使用这两条指令可在软件控制下对 TOS 执行操作。然后就可以修改 TOSU、TOSH 和 TOSL，将数据或返回地址压入堆栈。

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

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

6.1.4.4 堆栈满和下溢复位

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

6.1.5 快速寄存器堆栈（FRS）

为 STATUS、WREG 和 BSR 寄存器提供的快速寄存器堆栈（Fast Register Stack，FRS）具有从中断“快速返回”的功能。此堆栈只有一层深且不可读写。当处理器转入中断向量处执行时，它装入对应寄存器的当前值。所有中断源都会将值压入堆栈寄存器。如果使用 RETFIE 和 FAST 指令从中断返回，这些寄存器中的值就会被装回工作寄存器。

如果同时允许低优先级和高优先级中断，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在堆栈寄存器中的值将被覆盖。在这种情况下，在处理低优先级中断时用户必须用软件保存关键寄存器的值。

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

例 6-1 给出了一个在子程序调用和返回期间使用 FRS 的源代码示例。

例 6-1: 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
```

6.1.6 程序存储器中的查找表

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

- 计算 GOTO
- 表读

6.1.6.1 计算 GOTO

计算 GOTO 是通过向 PC 加一个偏移量来实现的。例 6-2 中给出了一个示例。

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

偏移量值（WREG 中）指定 PC 应该增加的字节数，其值应该为 2 的倍数（LSb = 0）。

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

例 6-2: 使用偏移量值的计算 GOTO

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

6.1.6.2 表读

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

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（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。

第 7.1 节“表读与表写”将进一步讨论表读操作。

6.2 PIC18 指令周期

6.2.1 时钟机制

来自内部或外部时钟源的单片机时钟输入都将在内部被四分频以产生四个互不重叠的正交时钟信号（Q1、Q2、Q3 和 Q4）。在单片机内部 PC 在每个 Q1 递增；在 Q4 期间，从程序存储器取指令并将指令锁存到指令寄存器（Instruction Register, IR）中。指令的译码和执行在下一个 Q1 到 Q4 周期完成。图 6-4 所示为时钟和指令执行流程。

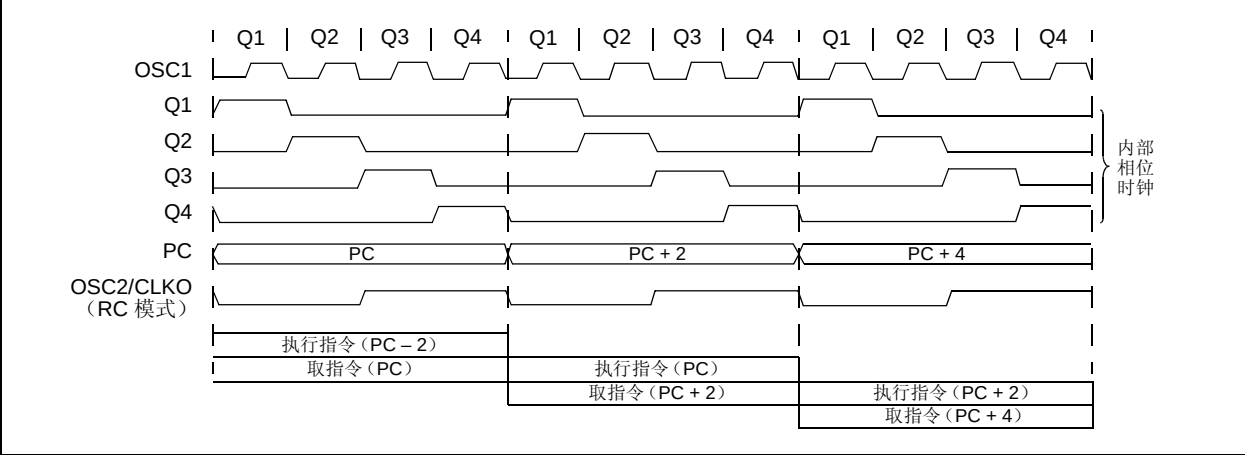
6.2.2 指令流 / 流水线

一个指令周期由 Q1 到 Q4 四个 Q 周期组成。取指令和执行指令是以流水线方式进行的，用一个指令周期来取指令，而用另一个指令周期译码和执行指令。但由于是流水线操作，所以每条指令的等效执行时间都是一个指令周期。如果某条指令改变了 PC（如 GOTO），则需要两个指令周期才能完成该指令（例 6-3）。

取指周期中：PC 在 Q1 周期递增 1，开始取指令。

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

图 6-4: 时钟 / 指令周期



例 6-3: 指令流水线流程

	Tcy0	Tcy1	Tcy2	Tcy3	Tcy4	Tcy5
1. MOVLW 55h	取指 1	执行 1				
2. MOVWF LATB		取指 2	执行 2			
3. BRA SUB_1			取指 3	执行 3		
4. BSF LATA, 3 (Forced NOP)				取指 4	清除 (NOP)	
5. Instruction @ address SUB_1					取指 SUB_1	执行 SUB_1

注：除程序跳转指令外，所有的指令都是单周期指令。由于程序跳转指令将导致一条已取指令从流水线清除，与此同时取新指令，然后执行指令，所以程序跳转指令需要两个周期。

6.2.3 程序存储器中的指令

程序存储器按字节寻址。指令以 2 字节或 4 字节的形式存储在程序存储器中。指令字的最低有效字节（Least Significant Byte，LSB）始终存储在地址为偶数的程序存储单元中（LSb = 0）。要保证与指令边界对齐，PC 必须以 2 为单位递增，并且 LSb 总是读为 0（见第 6.1.3 节“程序计数器”）。

图 6-5 给出了指令字存储在程序存储器中的一个示例。

CALL 和 GOTO 指令在指令中嵌入了程序存储器的绝对地址。由于指令总是按字边界存储，因而指令所包含的数据为一个字地址。字地址会写入 PC<20:1>，用于访问程序存储器中的目标字节。图 6-5 中的指令 2 给出了指令 GOTO 0006h 在程序存储器中的译码过程。程序跳转指令也采取同样的方式对相对地址偏移量进行译码。存储在跳转指令中的偏移量代表单字指令数，PC 将以此作为偏移量跳转到指定的地址单元。第 28.0 节“指令集汇总”提供了指令集的更多详细信息。

图 6-5: 程序存储器中的指令

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

6.2.4 双字指令

标准的 PIC18 指令集有 4 条双字指令: CALL、MOVFF、GOTO 和 LSFR。这些指令第二个字的高 4 位（Most Significant bit, MSb）均为 1111；其他 12 位是立即数数据，通常为一个数据存储单元地址。

指令的高 4 位为 1111，用于指定一条特殊形式的 NOP 指令。指令顺序执行的正确顺序为：执行完第一个字之后立即按顺序访问并使用第二个字中的数据。如果由于

某些原因跳过了第一个字而自动执行指令的第二个字，那么将作为一条 NOP 指令执行。如果双字指令跟在修改 PC 的条件指令后，就有必要执行此操作。例 6-4 给出了它的执行过程。

注：关于扩展指令集中的双字指令信息，请参见第 6.5 节“程序存储器和扩展指令集”。

例 6-4: 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

PIC18F46J50 系列

6.3 数据存储器构成

注： 当使能了 PIC18 扩展指令集时，数据存储器某些方面的操作会有所改变。更多信息，请参见第 6.6 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，每个寄存器都有 12 位地址，允许数据存储器实现为最大 4096 个字节。存储空间最多被分为 16 个存储区，每个存储区包含 256 个字节。PIC18F46J50 系列实现了所有可用的存储区，并提供 3.8 KB 的数据存储区供用户使用。图 6-6 给出了器件的数据存储器构成。

数据存储器由特殊功能寄存器（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指示，GPR 则用于用户应用程序的数据存储和中间结果暂存。任何未实现存储单元均读为 0。

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

为确保能在一个周期中访问常用寄存器（某些 SFR 和 GPR），PIC18 器件实现了一个快速操作存储区。该存储区是一个 256 字节的存储空间，它可实现对某些 SFR 和 GPR Bank 0 的低地址单元的快速访问，而无需使用 BSR。第 6.3.3 节“快速操作存储区”提供了对于快速操作 RAM 的详细说明。

6.3.1 USB RAM

PIC18F46J50 系列器件上实现的所有 3.8 KB GPR 可由单片机内核和串行接口引擎（Serial Interface Engine, SIE）同时访问。SIE 使用专用的 USB DMA 引擎将所有传入的数据包（OUT/SETUP）直接存储到主系统数据存储器中。

对于 IN（输入）数据包，SIE 可以直接读取通用 SRAM 的内容，并使用它创建发送给主机的 USB 数据包。

注： IN（输入）和 OUT（输出）始终是针对 USB 主机的角度而言。

SRAM Bank 4（400h-4FFh）是独特的。除了可由单片机内核和 USB 模块访问之外，SIE 还使用 Bank 4 的一部分作为特殊功能寄存器（SFR）。这些 SFR 组成缓冲描述符表（Buffer Descriptor Table, BDT）。

当使能 USB 模块时，BDT 寄存器用于控制每个已使能端点的 USB DMA 操作的行为。用于 BDT 的 SRAM 单元的准确数量取决于使能了多少个端点，以及使用的是哪种 USB 乒乓模式。更多详细信息，请参见第 22.3 节“USB RAM”。

当禁止 USB 模块时，这些 SRAM 单元的行为与所有其他 GPR 单元相同。当禁止 USB 模块时，这些单元可以用作通用寄存器。

6.3.2 存储区选择寄存器

容量较大的数据存储器需要高效的寻址机制，以便对所有地址进行快速访问。理想状况下，这意味着不必为每次读写操作提供完整地址。PIC18 器件是使用 RAM 存储区分区机制实现快速访问的。这种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，或通过 8 位的低字节地址和 4 位存储区指针直接寻址每个存储单元。

PIC18 指令集中的大部分指令都使用存储区指针，也就是存储区选择寄存器（Bank Select Register, BSR）。SFR 保存单元地址的高 4 位，而指令本身则包括单元地址的低 8 位。只使用 BSR 的低 4 位（BSR<3:0>），不使用高 4 位；高 4 位始终读为 0 且不能被写入。可以通过使用 MOVLB 指令直接装入 BSR。

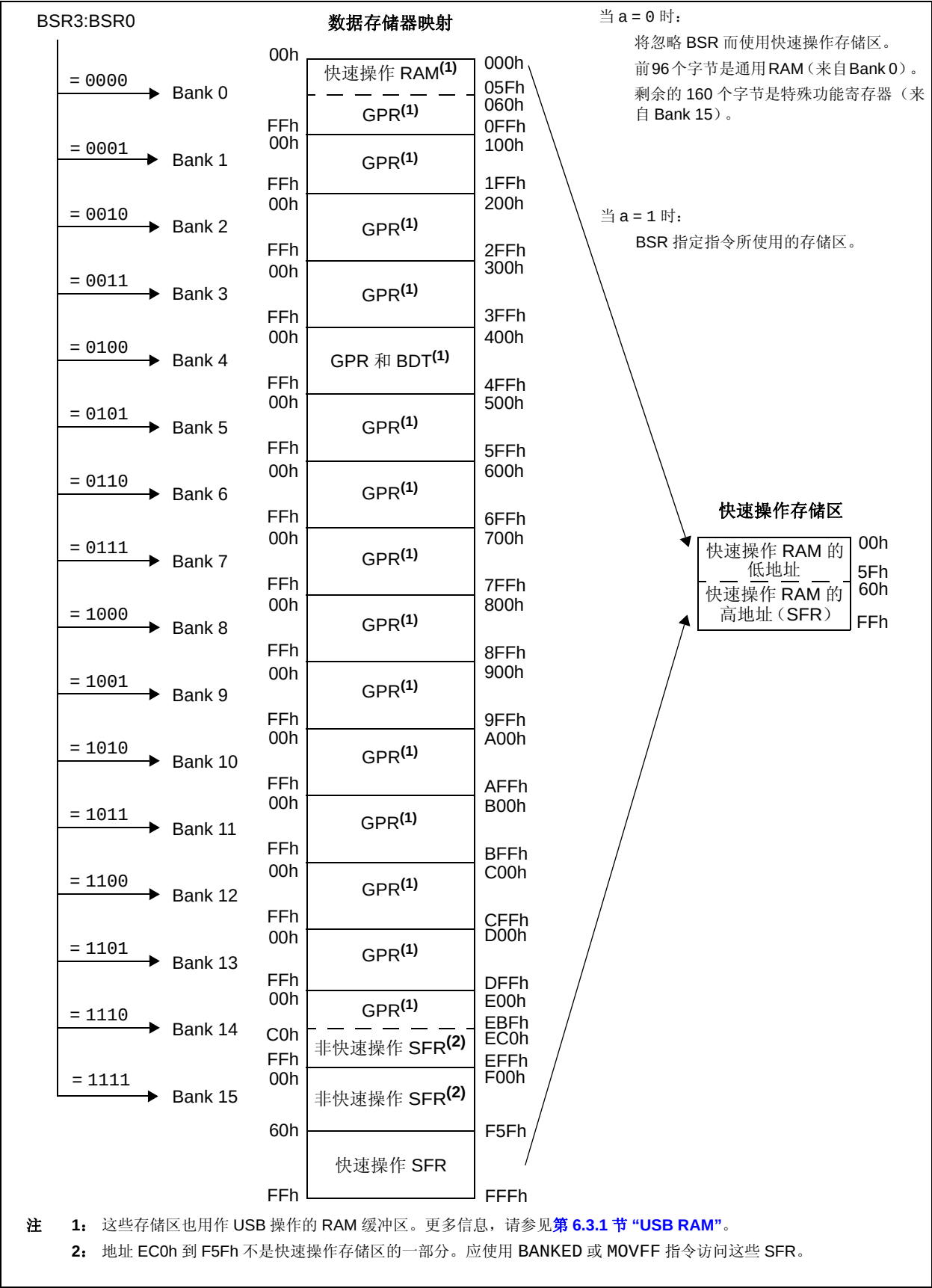
BSR 的值代表数据存储器中的存储区，指令中的 8 位指向存储区中的存储单元，可以将它看作距离存储区下边界的偏移量。图 6-7 显示了 BSR 的值与数据存储器中的存储区之间的关系。

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

当选择存储区时，只有已实现的存储区才可以读写。对未实现存储区进行的写操作将被忽略，而读这些存储区会返回 0。虽然是这样，STATUS 寄存器仍然会受到影响，好像操作是成功的。图 6-6 中的数据存储器映射指出了已实现的存储区。

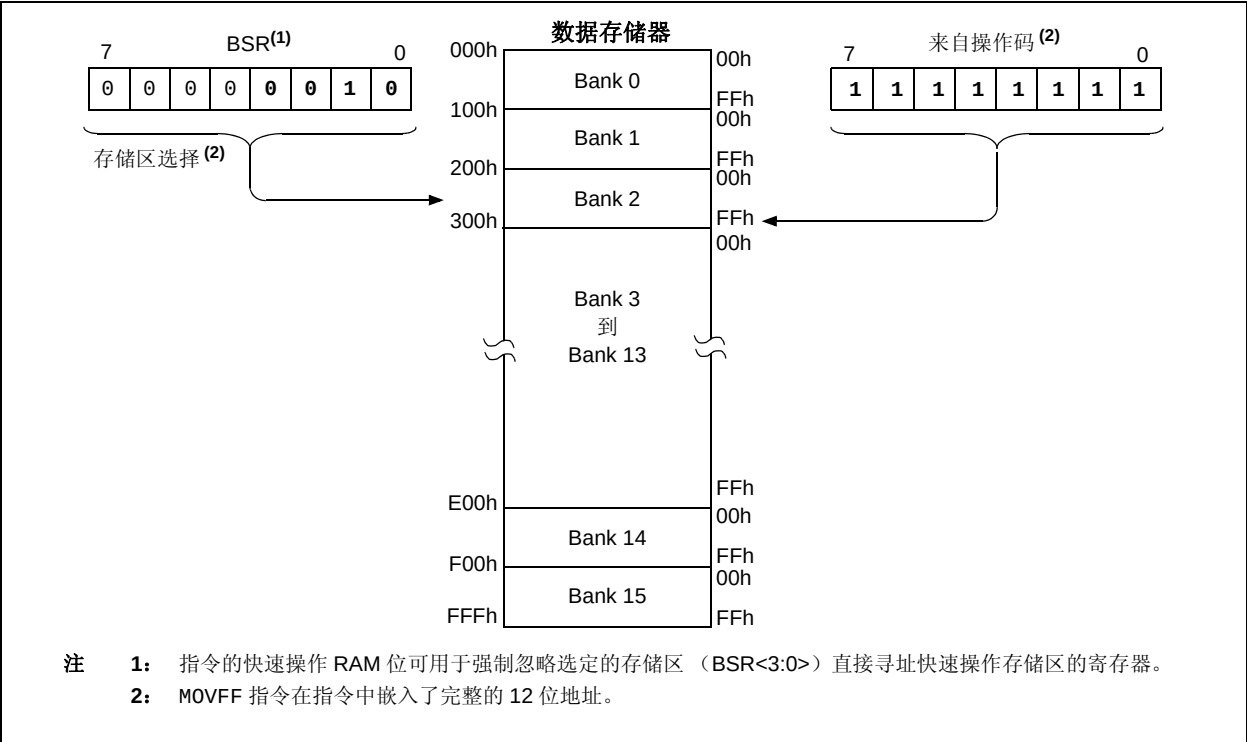
在核心 PIC18 指令集中，只有 MOVFF 指令指定源寄存器和目标寄存器的完整 12 位地址。该指令在执行时完全忽略 BSR。所有其他指令仅包含作为操作数的低位地址，而且必须使用 BSR 或快速操作存储区来寻址目标寄存器。

图 6-6: PIC18F46J50 系列器件的数据存储器映射



PIC18F46J50 系列

图 6-7: 使用存储区选择寄存器（直接寻址）



6.3.3 快速操作存储区

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

为了提高访问大多数常用数据存储单元的效率，现为数据存储器配置了快速操作存储区，这样可以允许用户访问被映射的存储区而无需指定 BSR。快速操作存储区由 Bank 0 的前 96 个字节 (00h-5Fh) 和 Bank 15 的后 160 个字节 (60h-FFh) 组成。地址较低的部分被称为“快速操作 RAM”，由 GPR 组成。地址较高的部分则被映射为器件的 SFR。这两个区域被连续地映射到快速操作存储区并且可以用一个 8 位地址进行线性寻址 (图 6-6)。

包括快速操作 RAM 位 (指令中的 “a” 参数) 的核心 PIC18 指令使用快速操作存储区。当 “a” 等于 1 时，指令使用 BSR 和包含在操作码中的 8 位地址对数据存储器寻址。当 “a” 为 0 时，强制指令使用位于快速操作存储区，地址：映射，此时完全忽略 BSR 的当前值。

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

使能扩展指令集 (XINST 配置位 = 1) 时的快速操作存储区的映射略有不同。在第 6.6.3 节“在立即数变址模式下映射快速操作存储区”中对此进行了更详细的讨论。

6.3.4 通用寄存器文件

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

6.3.5 特殊功能寄存器

SFR 是 CPU 和外设模块用来控制所需器件操作的寄存器。这些寄存器以静态 RAM 的形式实现。SFR 从数据存储器的顶部（FFFh）开始向下，它占用了 Bank 15 上面的大部分单元空间（F40h 到 FFFh）。表 6-2、表 6-3 和表 6-4 提供了这些寄存器的列表。

可以将 SFR 归类为两组：与“内核”器件功能（ALU、复位和中断）相关的寄存器和与外设功能相关的寄存器。复位和中断寄存器在相应章节中进行讨论，本章后

面的部分将对 ALU 的 STATUS 寄存器进行说明。与外设操作相关的寄存器将在该外设的章节中进行说明。

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

注： 位于 EC0h 和 F5Fh 之间的 SFR 不是快速操作存储区的一部分。应使用 BANKED 指令（使用 BSR）或 MOVFF 指令访问这些单元。在 MPLAB® C18 中编程时，编译器将自动使用相应的寻址模式。

表 6-2：快速操作存储区的特殊功能寄存器映射

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	PSTR1CON	F9Fh	IPR1	F7Fh	SPBRGH1
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	ECCP1AS	F9Eh	PIR1	F7Eh	BAUDCON1
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	ECCP1DEL	F9Dh	PIE1	F7Dh	SPBRGH2
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR1H	F9Ch	RCSTA2	F7Ch	BAUDCON2
FFBh	PCLATU	FDBh	PLUSW2 ⁽¹⁾	FBBh	CCPR1L	F9Bh	OSCTUNE	F7Bh	TMR3H
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP1CON	F9Ah	T1GCON	F7Ah	TMR3L
FF9h	PCL	FD9h	FSR2L	FB9h	PSTR2CON	F99h	RTCVALH	F79h	T3CON
FF8h	TBLPTRU	FD8h	STATUS	FB8h	ECCP2AS	F98h	RTCVALL	F78h	TMR4
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP2DEL	F97h	T3GCON	F77h	PR4
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	CCPR2H	F96h	TRISE	F76h	T4CON
FF5h	TABLAT	FD5h	T0CON	FB5h	CCPR2L	F95h	TRISD	F75h	SSP2BUF
FF4h	PRODH	FD4h	— ⁽⁵⁾	FB4h	CCP2CON	F94h	TRISC	F74h	SSP2ADD ⁽³⁾
FF3h	PRODL	FD3h	OSCCON	FB3h	CTMUCONH	F93h	TRISB	F73h	SSP2STAT
FF2h	INTCON	FD2h	CM1CON	FB2h	CTMUCONL	F92h	TRISA	F72h	SSP2CON1
FF1h	INTCON2	FD1h	CM2CON	FB1h	CTMUICON	F91h	ALRMCFG	F71h	SSP2CON2
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRG1	F90h	ALRMRPT	F70h	CMSTAT
FEFh	INDF0 ⁽¹⁾	FCFh	TMR1H	FAFh	RCREG1	F8Fh	ALRMVALH	F6Fh	PMADDRH ^(2,4)
FEEh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FAEh	TXREG1	F8Eh	ALRMVALL	F6Eh	PMADDRL ^(2,4)
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXSTA1	F8Dh	LATE ⁽²⁾	F6Dh	PMDIN1H ⁽²⁾
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	RCSTA1	F8Ch	LATD ⁽²⁾	F6Ch	PMDIN1L ⁽²⁾
FEBh	PLUSW0 ⁽¹⁾	FCBh	PR2	FABh	SPBRG2	F8Bh	LATC	F6Bh	TXADDRL
FEAh	FSR0H	FCAh	T2CON	FAAh	RCREG2	F8Ah	LATB	F6Ah	TXADDRH
FE9h	FSR0L	FC9h	SSP1BUF	FA9h	TXREG2	F89h	LATA	F69h	RXADDRL
FE8h	WREG	FC8h	SSP1ADD ⁽³⁾	FA8h	TXSTA2	F88h	DMACON1	F68h	RXADDRH
FE7h	INDF1 ⁽¹⁾	FC7h	SSP1STAT	FA7h	EECON2	F87h	— ⁽⁵⁾	F67h	DMABCL
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSP1CON1	FA6h	EECON1	F86h	DMACON2	F66h	DMABCH
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSP1CON2	FA5h	IPR3	F85h	HLVDCON	F65h	UCON
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE ⁽²⁾	F64h	USTAT
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD ⁽²⁾	F63h	UEIR
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	UIR
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	UFRMH
FE0h	BSR	FC0h	WDTCON	FA0h	PIE2	F80h	PORTA	F60h	UFRML

- 注**
- 这不是实际存在的寄存器。
 - 该寄存器在 28 引脚器件上不可用。
 - SSPxADD 和 SSPxMSK 共用相同的地址。
 - PMADDRH 和 PMDOUTH 共用相同的地址，PMADDRL 和 PMDOUTL 共用相同的地址。PMADDRx 在主模式下使用，PMDOUTx 在从模式下使用。
 - 保留：不要写入该单元。

PIC18F46J50 系列

表 6-3: 非快速操作存储区的特殊功能寄存器映射

地址	名称	地址	名称	地址	名称	地址	名称	地址	名称
F5Fh	PMCONH	F3Fh	RTCCFG	F1Fh	—	EFFh	PPSCON	EDFh	—
F5Eh	PMCONL	F3Eh	RTCCAL	F1Eh	—	EFEh	RPINR24	EDEh	RPOR24 ⁽⁴⁾
F5Dh	PMMODEH	F3Dh	REFOCON	F1Dh	—	EFDh	RPINR23	EDDh	RPOR23 ⁽⁴⁾
F5Ch	PMMODEL	F3Ch	PADCFG1	F1Ch	—	EFCh	RPINR22	EDCh	RPOR22 ⁽⁴⁾
F5Bh	PMDOUT2H	F3Bh	—	F1Bh	—	EFBh	RPINR21	EDBh	RPOR21 ⁽⁴⁾
F5Ah	PMDOUT2L	F3Ah	—	F1Ah	—	EFAh	—	EDAh	RPOR20 ⁽⁴⁾
F59h	PMDIN2H	F39h	UCFG	F19h	—	EF9h	—	ED9h	RPOR19 ⁽⁴⁾
F58h	PMDIN2L	F38h	UADDR	F18h	—	EF8h	—	ED8h	RPOR18
F57h	PMEH	F37h	UEIE	F17h	—	EF7h	RPINR17	ED7h	RPOR17
F56h	PMEL	F36h	UIE	F16h	—	EF6h	RPINR16	ED6h	—
F55h	PMSTATH	F35h	UEP15	F15h	—	EF5h	—	ED5h	—
F54h	PMSTATL	F34h	UEP14	F14h	—	EF4h	—	ED4h	—
F53h	CVRCON	F33h	UEP13	F13h	—	EF3h	RPINR13	ED3h	RPOR13
F52h	TCLKCON	F32h	UEP12	F12h	—	EF2h	RPINR12	ED2h	RPOR12
F51h	—	F31h	UEP11	F11h	—	EF1h	—	ED1h	RPOR11
F50h	—	F30h	UEP10	F10h	—	EF0h	—	ED0h	RPOR10
F4Fh	DSGPR1	F2Fh	UEP9	F0Fh	—	EEFh	—	ECFh	RPOR9
F4Eh	DSGPR0	F2Eh	UEP8	F0Eh	—	EEEh	RPINR8	ECEh	RPOR8
F4Dh	DSCONH	F2Dh	UEP7	F0Dh	—	EEDh	RPINR7	ECDh	RPOR7
F4Ch	DSCONL	F2Ch	UEP6	F0Ch	—	EECh	RPINR6	ECCh	RPOR6
F4Bh	DSWAKEH	F2Bh	UEP5	F0Bh	—	EEBh	—	ECBh	RPOR5
F4Ah	DSWAKEL	F2Ah	UEP4	F0Ah	—	EEAh	RPINR4	ECAh	RPOR4
F49h	ANCON1	F29h	UEP3	F09h	—	EE9h	RPINR3	EC9h	RPOR3
F48h	ANCON0	F28h	UEP2	F08h	—	EE8h	RPINR2	EC8h	RPOR2
F47h	—	F27h	UEP1	F07h	—	EE7h	RPINR1	EC7h	RPOR1
F46h	—	F26h	UEP0	F06h	—	EE6h	—	EC6h	RPOR0
F45h	—	F25h	—	F05h	—	EE5h	—	EC5h	—
F44h	—	F24h	—	F04h	—	EE4h	—	EC4h	—
F43h	—	F23h	—	F03h	—	EE3h	—	EC3h	—
F42h	ODCON1	F22h	—	F02h	—	EE2h	—	EC2h	—
F41h	ODCON2	F21h	—	F01h	—	EE1h	—	EC1h	—
F40h	ODCON3	F20h	—	F00h	—	EE0h	—	EC0h	—

注 1: 该寄存器在 28 引脚器件上不可用。

6.3.5.1 上下文定义的 SFR

有几个寄存器共用 SFR 空间中的相同地址。寄存器的定义和使用取决于其关联外设的工作模式。这些寄存器是：

- **SSPxADD** 和 **SSPxMSK**：它们是两个独立的硬件寄存器，通过单个 SFR 地址进行访问。MSSP 模块的工作模式决定哪一个寄存器被访问。更多详细信息，请参见第 19.5.3.4 节“7 位地址掩码模式”。

- **PMADDRH/L** 和 **PMDOUT2H/L**：在这种情况下，这些指定的缓冲寄存器对实际上是相同的物理寄存器。并行主端口（PMP）模块的工作模式决定寄存器具有哪种功能。更多详细信息，请参见第 11.1.2 节“数据寄存器”。

表 6-4: 寄存器文件汇总（PIC18F46J50 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	69, 81
TOSH	栈顶高字节 (TOS<15:8>)								0000 0000	69, 79
TOSL	栈顶低字节 (TOS<7:0>)								0000 0000	69, 79
STKPTR	STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0	00-0 0000	69, 79
PCLATU	—	—	bit 21 ⁽¹⁾	PC<20:16> 的保持寄存器					---0 0000	69, 79
PCLATH	PC<15:8> 的保持寄存器								0000 0000	69, 79
PCL	PC 低字节 (PC<7:0>)								0000 0000	69, 79
TBLPTRU	—	—	bit 21	程序存储器表指针最高字节 (TBLPTR<20:16>)					--00 0000	69, 112
TBLPTRH	程序存储器表指针高字节 (TBLPTR<15:8>)								0000 0000	69, 112
TBLPTRL	程序存储器表指针低字节 (TBLPTR<7:0>)								0000 0000	69, 112
TABLAT	程序存储器表锁存器								0000 0000	69, 112
PRODH	乘积寄存器的高字节								xxxx xxxx	69, 113
PRODL	乘积寄存器的低字节								xxxx xxxx	69, 113
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	69, 117
INTCON2	RBPUP	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	69, 117
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	69, 117
INDF0	使用 FSR0 的内容寻址数据存储器——FSR0 的值不变 (不是实际存在的寄存器)								N/A	69, 98
POSTINC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后递增 (不是实际存在的寄存器)								N/A	69, 99
POSTDEC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值后递减 (不是实际存在的寄存器)								N/A	69, 99
PREINC0	使用 FSR0 的内容寻址数据存储器——FSR0 的值预递增 (不是实际存在的寄存器)								N/A	69, 99
PLUSW0	使用 FSR0 的内容寻址数据存储器——FSR0 的值预递增 (不是实际存在的寄存器), FSR0 偏移量的值由 W 寄存器提供								N/A	69, 99
FSR0H	—	—	—	—	间接数据存储器地址指针 0 的高字节				---- 0000	69, 98
FSR0L	间接数据存储器地址指针 0 的低字节								xxxx xxxx	69, 98
WREG	工作寄存器								xxxx xxxx	69, 81
INDF1	使用 FSR1 的内容寻址数据存储器——FSR1 的值不变 (不是实际存在的寄存器)								N/A	69, 98
POSTINC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值后递增 (不是实际存在的寄存器)								N/A	69, 99
POSTDEC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值后递减 (不是实际存在的寄存器)								N/A	69, 99
PREINC1	使用 FSR1 的内容寻址数据存储器——FSR1 的值预递增 (不是实际存在的寄存器)								N/A	70, 99
PLUSW1	使用 FSR1 的内容寻址数据存储器——FSR1 的值预递增 (不是实际存在的寄存器), FSR1 偏移量的值由 W 寄存器提供								N/A	69, 99

图注： x = 未知，u = 不变，— = 未实现，q = 值取决于具体条件，r = 保留。**粗体**表示共用的快速操作 SFR。

- 1: PC 的 bit 21 仅在串行编程（Serial Programming，SP）模式下可用。
- 2: 双速启动使能时复位值为 0，禁止时为 1。
- 3: 仅当 SSPxCON2<3:0> = 1001 时才能访问 SSPxMSK 寄存器。
- 4: 当 MSSP 模块工作在 I²C™ 从模式下时这些位的备用名称和定义。详情请参见第 19.5.3.2 节“地址掩码模式”。
- 5: 这些位和 / 或寄存器仅在 44 引脚器件上可用；在其他器件上，它们未实现并读为 0。所示为 44 引脚器件的复位值。
- 6: PMADDRH/PMOUT1H 和 PMADDRL/PMOUT1L 寄存器对共用相同的物理寄存器和地址，但具有由模块工作模式所决定的不同功能。更多信息，请参见第 11.1.2 节“数据寄存器”。
- 7: 仅当相应的引脚没有被配置为用于主振荡器功能时，才实现 TRISA6 和 TRISA7 位。

PIC18F46J50 系列

表 6-4: 寄存器文件汇总 (PIC18F46J50 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)：
FSR1H	—	—	—	—	间接数据存储器地址指针 1 的高字节				---- 0000	69, 98
FSR1L	间接数据存储器地址指针 1 的低字节								xxxx xxxx	69, 98
BSR	—	—	—	—	存储区选择寄存器				---- 0000	69, 84
INDF2	使用 FSR2 的内容寻址数据存储器——FSR2 的值不变（不是实际存在的寄存器）								N/A	69, 98
POSTINC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值后递增（不是实际存在的寄存器）								N/A	70, 99
POSTDEC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值后递减（不是实际存在的寄存器）								N/A	70, 99
PREINC2	使用 FSR2 的内容寻址数据存储器——FSR2 的值预递增（不是实际存在的寄存器）								N/A	70, 99
PLUSW2	使用 FSR2 的内容寻址数据存储器——FSR2 的值预递增（不是实际存在的寄存器），FSR2 偏移量的值由 W 寄存器提供								N/A	70, 99
FSR2H	—	—	—	—	间接数据存储器地址指针 2 的高字节				---- 0000	70, 98
FSR2L	间接数据存储器地址指针 2 的低字节								xxxx xxxx	70, 98
STATUS	—	—	—	N	OV	Z	DC	C	--x xxxx	70, 96
TMR0H	Timer0 寄存器的高字节								0000 0000	70, 203
TMR0L	Timer0 寄存器的低字节								xxxx xxxx	70, 203
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	70, 196
OSCCON	IDLEN	IRCF2	IRCF1	IRCF0	OSTS ⁽²⁾	—	SCS1	SCS0	0110 q-00	70, 43
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111	70, 391
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	0001 1111	70, 391
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	0-11 1100	68, 70, 129
TMR1H	Timer1 寄存器的高字节								xxxx xxxx	70, 203
TMR1L	Timer1 寄存器的低字节								xxxx xxxx	70, 203
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	RD16	TMR1ON	0000 0000	70, 203
TMR2	Timer2 寄存器								0000 0000	70, 211
PR2	Timer2 周期寄存器								1111 1111	70, 211
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	70, 211
SSP1BUF	MSSP1 接收缓冲区 / 发送寄存器								xxxx xxxx	70, 288, 322
SSP1ADD	MSSP1 地址寄存器（I ² C™ 从模式），MSSP1 波特率重载寄存器（I ² C 主模式）								0000 0000	70, 293
SSP1MSK ⁽⁴⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	70, 295
SSP1STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	70, 270, 289
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	70, 270, 290
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	70, 270, 291
	GCEN	ACKSTAT	ADMSK5 ⁽⁴⁾	ADMSK4 ⁽⁴⁾	ADMSK3 ⁽⁴⁾	ADMSK2 ⁽⁴⁾	ADMSK1 ⁽⁴⁾	SEN		
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	70, 356
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	70, 356
ADCON0	VCFG1	VCFG0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	69, 347
ADCON1	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0000 0000	70, 347
WDTCON	REGSLP	LVDSTAT	ULPLVL	—	DS	ULPEN	ULPSINK	SWDTEN	1qx- q000	70, 427
PSTR1CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA	00-0 0001	70, 265
ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	0000 0000	70

图注: x = 未知, u = 不变, — = 未实现, q = 值取决于具体条件, r = 保留。**粗体**表示共用的快速操作 SFR。

- 注**
- 1: PC 的 bit 21 仅在串行编程 (Serial Programming, SP) 模式下可用。
 - 2: 双速启动使能时复位值为 0, 禁止时为 1。
 - 3: 仅当 SSPxCON2<3:0> = 1001 时才能访问 SSPxMSK 寄存器。
 - 4: 当 MSSP 模块工作在 I²C™ 从模式时这些位的备用名称和定义。详情请参见 [第 19.5.3.2 节“地址掩码模式”](#)。
 - 5: 这些位和 / 或寄存器仅在 44 引脚器件上可用; 在其他器件上, 它们未实现并读为 0。所示为 44 引脚器件的复位值。
 - 6: PMADDRH/PMDDOUT1H 和 PMADDRL/PMDDOUT1L 寄存器对共用相同的物理寄存器和地址, 但具有由模块工作模式所决定的不同功能。更多信息, 请参见 [第 11.1.2 节“数据寄存器”](#)。
 - 7: 仅当相应的引脚没有被配置为用于主振荡器功能时, 才实现 TRISA6 和 TRISA7 位。

表 6-4: 寄存器文件汇总 (PIC18F46J50 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):
ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	0000 0000	71
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								xxxx xxxx	71
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								xxxx xxxx	71
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	71
PSTR2CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA	00-0 0001	71, 265
ECCP2AS	ECCP2ASE	ECCP2AS2	ECCP2AS1	ECCP2AS0	PSS2AC1	PSS2AC0	PSS2BD1	PSS2BD0	0000 0000	71
ECCP2DEL	P2RSEN	P2DC6	P2DC5	P2DC4	P2DC3	P2DC2	P2DC1	P2DC0	0000 0000	71
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								xxxx xxxx	71
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								xxxx xxxx	71
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	0000 0000	71
CTMUCONH	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	—	0-00 000-	71
CTMUCONL	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000 00xx	71
CTMUICON	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	0000 0000	71
SPBRG1	EUSART1 波特率发生器寄存器的低字节								0000 0000	71, 327
RCREG1	EUSART1 接收寄存器								0000 0000	71, 336, 328
TXREG1	EUSART1 发送寄存器								xxxx xxxx	71, 336, 335
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	71, 333
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	71, 336
SPBRG2	EUSART2 波特率发生器寄存器的低字节								0000 0000	71, 327
RCREG2	EUSART2 接收寄存器								0000 0000	71, 336, 338
TXREG2	EUSART2 发送寄存器								0000 0000	71, 333, 335
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	71, 333
EECON2	程序存储器控制寄存器 2 (不是实际存在的寄存器)								---- ----	71, 104
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	--00 x00-	71, 104
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	1111 1111	71, 126
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	0000 0000	71, 120
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	0000 0000	71, 123
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	1111 1111	71, 126
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	0000 0000	71, 120
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	0000 0000	71, 123
IPR1	PMPIP ⁽⁵⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	1111 1111	71, 126
PIR1	PMPIF ⁽⁵⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	71, 120
PIE1	PMPIE ⁽⁵⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	71, 123
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	72, 336
OSCTUNE	INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000 0000	72, 39
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ T1DONE	T1GVAL	T1GSS1	T1GSS0	0000 0x00	201
RTCVALH	RTCC 值寄存器窗口的高字节, 基于 RTCPTR<1:0>								0xxx xxxx	72, 231
RTCVALL	RTCC 值寄存器窗口的低字节, 基于 RTCPTR<1:0>								0xxx xxxx	72, 231

图注: x = 未知, u = 不变, — = 未实现, q = 值取决于具体条件, r = 保留。粗体表示共用的快速操作 SFR。

- 注**
- 1: PC 的 bit 21 仅在串行编程 (Serial Programming, SP) 模式下可用。
 - 2: 双速启动使能时复位值为 0, 禁止时为 1。
 - 3: 仅当 SSPxCON2<3:0> = 1001 时才能访问 SSPxMSK 寄存器。
 - 4: 当 MSSP 模块工作在 I²C™ 从模式下时这些位的备用名称和定义。详情请参见第 19.5.3.2 节 “地址掩码模式”。
 - 5: 这些位和 / 或寄存器仅在 44 引脚器件上可用; 在其他器件上, 它们未实现并读为 0。所示为 44 引脚器件的复位值。
 - 6: PMADDRH/PMDOU1H 和 PMADDRL/PMDOU1L 寄存器对共用相同的物理寄存器和地址, 但具有由模块工作模式所决定的不同功能。更多信息, 请参见第 11.1.2 节 “数据寄存器”。
 - 7: 仅当相应的引脚没有被配置为用于主振荡器功能时, 才实现 TRISA6 和 TRISA7 位。

PIC18F46J50 系列

表 6-4: 寄存器文件汇总 (PIC18F46J50 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):
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ T3DONE	T3GVAL	T3GSS1	T3GSS0	0000 0x00	72, 214
TRISE	—	—	—	—	—	TRISE2	TRISE1	TRISE0	---- -111	72
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	72, 146
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	11-- -111	72, 143
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	72, 139
TRISA	TRISA7 ⁽⁷⁾	TRISA6 ⁽⁷⁾	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	qq1- 1111	72, 136
ALRMCFG	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	0000 0000	72, 229
ALRMRPT	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000 0000	72, 230
ALRMVALH	闹钟值寄存器窗口的高字节, 基于 ALRMPTR<1:0>								xxxx xxxx	72, 234
ALRMVALL	闹钟值寄存器窗口的低字节, 基于 ALRMPTR<1:0>								xxxx xxxx	72, 234
LATE	—	—	—	—	—	LATE2	LATE1	LATE0	---- -xxx	72, 149
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx	72, 147
LATC	LATC7	LATC6	—	—	—	LATC2	LATC1	LATC0	xxxx -xxx	72, 142
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx	72, 142
LATA	LATA7	LATA6	LATA5	—	LATA3	LATA2	LATA1	LATA0	xxx- xxxx	72, 142
DMACON1	SSCON1	SSCON0	TXINC	RXINC	DUPLEX1	DUPLEX0	DLYINTEN	DMAEN	0000 0000	72, 282
DMATXBUF	SPI DMA 发送缓冲区								xxxx xxxx	72
DMACON2	DLYCYC3	DLYCYC2	DLYCYC1	DLYCYC0	INTLVL3	INTLVL2	INTLVL1	INTLVL0	0000 0000	72, 283
HLVDCON	VDIRMAG	BGVST	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0000 0000	72
PORTE	RDPUR	REPU	—	—	—	RE2	RE1	RE0	00-- -xxx	72, 132
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	72, 132
PORTC	RC7	RC6	RC5	RC4	—	RC2	RC1	RC0	xxxx -xxx	72, 132
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	72, 132
PORTA	RA7	RA6	RA5	—	RA3	RA2	RA1	RA0	xxx- xxxx	72, 356
SPBRGH1	EUSART1 波特率发生器寄存器的高字节								0000 0000	72, 327
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	0100 0-00	72, 327
SPBRGH2	EUSART2 波特率发生器寄存器的高字节								0000 0000	72, 327
BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	0100 0-00	72, 327
TMR3H	Timer3 寄存器的高字节								xxxx xxxx	73, 197
TMR3L	Timer3 寄存器的低字节								xxxx xxxx	73, 197
T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	T3SYNC	RD16	TMR3ON	0000 0000	73, 197
TMR4	Timer4 寄存器								0000 0000	73, 223
PR4	Timer4 周期寄存器								1111 1111	73, 197
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	-000 0000	73, 223
SSP2BUF	MSSP2 接收缓冲区 / 发送寄存器								xxxx xxxx	73, 288, 322
SSP2ADD/	MSSP2 地址寄存器 (I ² C™ 从模式), MSSP2 波特率重载寄存器 (I ² C 主模式)								0000 0000	73, 288
SSP2MSK ⁽⁴⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	73, 295
SSP2STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	73, 270, 310
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	73, 270, 322

图注: x = 未知, u = 不变, — = 未实现, q = 值取决于具体条件, r = 保留。粗体表示共用的快速操作 SFR。

- 注
- 1: PC 的 bit 21 仅在串行编程 (Serial Programming, SP) 模式下可用。
 - 2: 双速启动使能时复位值为 0, 禁止时为 1。
 - 3: 仅当 SSPxCON2<3:0> = 1001 时才能访问 SSPxMSK 寄存器。
 - 4: 当 MSSP 模块工作在 I²C™ 从模式时这些位的备用名称和定义。详情请参见第 19.5.3.2 节 “地址掩码模式”。
 - 5: 这些位和 / 或寄存器仅在 44 引脚器件上可用; 在其他器件上, 它们未实现并读为 0。所示为 44 引脚器件的复位值。
 - 6: PMADDRH/PMOUT1H 和 PMADDRH/PMOUT1L 寄存器对共用相同的物理寄存器和地址, 但具有由模块工作模式所决定的不同功能。更多信息, 请参见第 11.1.2 节 “数据寄存器”。
 - 7: 仅当相应的引脚没有被配置为用于主振荡器功能时, 才实现 TRISA6 和 TRISA7 位。

表 6-4: 寄存器文件汇总 (PIC18F46J50 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):
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	73, 270, 322
	GCEN	ACKSTAT	ADMSK5 ⁽⁴⁾	ADMSK4 ⁽⁴⁾	ADMSK3 ⁽⁴⁾	ADMSK2 ⁽⁴⁾	ADMSK1 ⁽⁴⁾	SEN		
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	---- - -11	73, 389
PMDOUT1H ^(5,6)	—	CS1	并行主端口地址的高字节						-000 0000	73, 177
PMADDRL/	并行端口输出数据的高字节 (缓冲区 1)								0000 0000	73, 180
PMDOUT1L ^(5,6)	并行主端口地址的低字节								0000 0000	73, 176
PMDIN1H ⁽⁵⁾	并行端口输出数据的低字节 (缓冲区 0)								0000 0000	73, 177
PMDIN1L ⁽⁵⁾	并行端口输入数据的高字节 (缓冲区 1)								0000 0000	73, 177
PMDOUT1H ^(5,6)	并行端口输入数据的低字节 (缓冲区 0)								0000 0000	73, 177
TXADDRL	SPI DMA 发送数据指针的低字节								xxxx xxxx	73, 284
TXADDRH	—	—	—	—	SPI DMA 发送数据指针的高字节				---- xxxx	73, 284
RXADDRL	SPI DMA 接收数据指针的低字节								xxxx xxxx	73, 284
RXADDRH	—	—	—	—	SPI DMA 接收数据指针的高字节				---- xxxx	73, 284
DMABCL	SPI DMA 字节计数的低字节								xxxx xxxx	73, 284
DMABCH	—	—	—	—	—	—	SPI DMA 字节计数的高字节		---- --xx	73, 284
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	-0x0 000-	73, 359
USTAT	—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	—	-xxx xxx-	73, 363
UEIR	BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0--0 0000	73, 376
UIR	—	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	-000 0000	73, 373
UFRMH	—	—	—	—	—	FRM10	FRM9	FRM8	---- -xxx	73, 365
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	xxxx xxxx	73, 365
PMCONH ⁽⁵⁾	PMPEN	—	—	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	0--0 0000	73, 170
PMCONL ⁽⁵⁾	CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP	000- 0000	73, 171
PMMODEH ⁽⁵⁾	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	0000 0000	74, 172
PMMODEL ⁽⁵⁾	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	0000 0000	74, 173
PMDOUT2H ⁽⁵⁾	并行端口输出数据的高字节 (缓冲区 3)								0000 0000	74, 176
PMDOUT2L ⁽⁵⁾	并行端口输出数据的低字节 (缓冲区 2)								0000 0000	74, 176
PMDIN2H ⁽⁵⁾	并行端口输入数据的高字节 (缓冲区 3)								0000 0000	74, 176
PMDIN2L ⁽⁵⁾	并行端口输入数据的低字节 (缓冲区 2)								0000 0000	74, 176
PMEH ⁽⁵⁾	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	0000 0000	74, 174
PMEL ⁽⁵⁾	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000 0000	74, 174
PMSTATH ⁽⁵⁾	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	00-- 0000	74, 175
PMSTATL ⁽⁵⁾	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	10-- 1111	74, 175
CVRCON	CVREN	CVROE	CVRR	r	CVR3	CVR2	CVR1	CVR0	0000 0000	74, 392
TCLKCON	—	—	—	T1RUN	—	—	T3CCP2	T3CCP1	---0 --00	202
DSGPR1	深度休眠持久通用寄存器 (即使在深度休眠模式下也能保留的内容)								uuuu uuuu	58
DSGPR0	深度休眠持久通用寄存器 (即使在深度休眠模式下也能保留的内容)								uuuu uuuu	58
DSCONH	DSEN	—	—	—	—	r	DSULPEN	RTCWDIS	0--- -000	57
DSCONL	—	—	—	—	—	ULPWDIS	DSBOR	RELEASE	---- -000	57
DSWAKEH	—	—	—	—	—	—	—	DSINT0	---- ---0	59
DSWAKEL	DSFLT	—	DSULP	DSWDT	DSRTC	DSMCLR	—	DSPOR	0-00 00-1	59

图注: x = 未知, u = 不变, — = 未实现, q = 值取决于具体条件, r = 保留。粗体表示共用的快速操作 SFR。

- 注**
- 1: PC 的 bit 21 仅在串行编程 (Serial Programming, SP) 模式下可用。
 - 2: 双速启动使能时复位值为 0, 禁止时为 1。
 - 3: 仅当 SSPxCON2<3:0> = 1001 时才能访问 SSPxMSK 寄存器。
 - 4: 当 MSSP 模块工作在 I²C™ 从模式下时这些位的备用名称和定义。详情请参见第 19.5.3.2 节“地址掩码模式”。
 - 5: 这些位和 / 或寄存器仅在 44 引脚器件上可用; 在其他器件上, 它们未实现并读为 0。所示为 44 引脚器件的复位值。
 - 6: PMADDRH/PMOUT1H 和 PMADDR1/PMOUT1L 寄存器对共用相同的物理寄存器和地址, 但具有由模块工作模式所决定的不同功能。更多信息, 请参见第 11.1.2 节“数据寄存器”。
 - 7: 仅当相应的引脚没有被配置为用于主振荡器功能时, 才实现 TRISA6 和 TRISA7 位。

PIC18F46J50 系列

表 6-4: 寄存器文件汇总 (PIC18F46J50 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):
ANCON1	VBGEN	r	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	00-0 0000	74, 348
ANCON0	PCFG7 ⁽⁵⁾	PCFG6 ⁽⁵⁾	PCFG5 ⁽⁵⁾	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000	74, 347
ODCON1	—	—	—	—	—	—	ECCP20D	ECCP10D	---- --00	74, 134
ODCON2	—	—	—	—	—	—	U2OD	U1OD	---- --00	74, 134
ODCON3	—	—	—	—	—	—	SPI2OD	SPI1OD	---- --00	74, 135
RTCCFG	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTTR1	RTCPTTR0	0-00 0000	74, 227
RTCCAL	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000 0000	74, 228
REFOCON	ROON	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	0-00 0000	74, 44
PADCFG1	—	—	—	—	—	RTSECSEL1	RTSECSEL0	PMP TTL	---- -000	74, 135
UCFG	UTEYE	UOEMON	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0	00-0 0000	74, 360
UADDR	—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	-000 0000	74, 365
UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0--0 0000	74, 377
UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	-000 0000	74, 375
UEP15	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP14	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP13	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP12	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP11	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP10	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP9	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP8	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP7	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	74, 364
UEP6	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	75, 364
UEP5	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	75, 364
UEP4	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	75, 364
UEP3	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	75, 364
UEP2	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	75, 364
UEP1	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	75, 364
UEP0	—	—	—	EPHSK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	--0 0000	75, 364
PPSCON	—	—	—	—	—	—	—	IOLOCK	---- --0	155
RPINR24	—	—	—	输入功能 FLT0 到输入引脚映射位					---1 1111	75, 160
RPINR23	—	—	—	输入功能 SS2 到输入引脚映射位					---1 1111	75, 160
RPINR22	—	—	—	输入功能 SCK2 到输入引脚映射位					---1 1111	75, 160
RPINR21	—	—	—	输入功能 SDI2 到输入引脚映射位					---1 1111	75, 159
RPINR17	—	—	—	输入功能 CK2 到输入引脚映射位					---1 1111	75, 159
RPINR16	—	—	—	输入功能 RX2DT2 到输入引脚映射位					---1 1111	75
RPINR13	—	—	—	输入功能 T3G 到输入引脚映射位					---1 1111	75
RPINR12	—	—	—	输入功能 T1G 到输入引脚映射位					---1 1111	75, 158
RPINR8	—	—	—	输入功能 IC2 到输入引脚映射位					---1 1111	75, 158
RPINR7	—	—	—	输入功能 IC1 到输入引脚映射位					---1 1111	75, 157
RPINR6	—	—	—	输入功能 T3CKI 到输入引脚映射位					---1 1111	75, 157
RPINR4	—	—	—	输入功能 T0CKI 到输入引脚映射位					---1 1111	75, 157

图注: x = 未知, u = 不变, — = 未实现, q = 值取决于具体条件, r = 保留。粗体表示共用的快速操作 SFR。

- 注
- 1: PC 的 bit 21 仅在串行编程 (Serial Programming, SP) 模式下可用。
 - 2: 双速启动使能时复位值为 0, 禁止时为 1。
 - 3: 仅当 SSPxCON2<3:0> = 1001 时才能访问 SSPxMSK 寄存器。
 - 4: 当 MSSP 模块工作在 I²C™ 从模式时这些位的备用名称和定义。详情请参见第 19.5.3.2 节“地址掩码模式”。
 - 5: 这些位和 / 或寄存器仅在 44 引脚器件上可用; 在其他器件上, 它们未实现并读为 0。所示为 44 引脚器件的复位值。
 - 6: PMADDRH/PMDOU1H 和 PMADDRH/PMDOU1L 寄存器对共用相同的物理寄存器和地址, 但具有由模块工作模式所决定的不同功能。更多信息, 请参见第 11.1.2 节“数据寄存器”。
 - 7: 仅当相应的引脚没有被配置为用于主振荡器功能时, 才实现 TRISA6 和 TRISA7 位。

表 6-4: 寄存器文件汇总（PIC18F46J50 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):
RPINR3	—	—	—	输入功能 INT3 到输入引脚映射位					---1 1111	76, 156
RPINR2	—	—	—	输入功能 INT2 到输入引脚映射位					---1 1111	76
RPINR1	—	—	—	输入功能 INT1 到输入引脚映射位					---1 1111	76, 156
RPOR24 ⁽⁵⁾	—	—	—	可重映射引脚 RP24 输出信号选择位					---0 0000	76, 168
RPOR23 ⁽⁵⁾	—	—	—	可重映射引脚 RP23 输出信号选择位					---0 0000	76, 167
RPOR22 ⁽⁵⁾	—	—	—	可重映射引脚 RP22 输出信号选择位					---0 0000	76, 167
RPOR21 ⁽⁵⁾	—	—	—	可重映射引脚 RP21 输出信号选择位					---0 0000	76, 167
RPOR20 ⁽⁵⁾	—	—	—	可重映射引脚 RP20 输出信号选择位					---0 0000	76, 166
RPOR19 ⁽⁵⁾	—	—	—	可重映射引脚 RP19 输出信号选择位					---0 0000	76, 166
RPOR18	—	—	—	可重映射引脚 RP18 输出信号选择位					---0 0000	76, 166
RPOR17	—	—	—	可重映射引脚 RP17 输出信号选择位					---0 0000	76, 165
RPOR13	—	—	—	可重映射引脚 RP13 输出信号选择位					---0 0000	76, 165
RPOR12	—	—	—	可重映射引脚 RP12 输出信号选择位					---0 0000	76, 165
RPOR11	—	—	—	可重映射引脚 RP11 输出信号选择位					---0 0000	76, 164
RPOR10	—	—	—	可重映射引脚 RP10 输出信号选择位					---0 0000	76, 164
RPOR9	—	—	—	可重映射引脚 RP9 输出信号选择位					---0 0000	76, 164
RPOR8	—	—	—	可重映射引脚 RP8 输出信号选择位					---0 0000	76, 163
RPOR7	—	—	—	可重映射引脚 RP7 输出信号选择位					---0 0000	76, 163
RPOR6	—	—	—	可重映射引脚 RP6 输出信号选择位					---0 0000	76, 163
RPOR5	—	—	—	可重映射引脚 RP5 输出信号选择位					---0 0000	76, 162
RPOR4	—	—	—	可重映射引脚 RP4 输出信号选择位					---0 0000	76, 162
RPOR3	—	—	—	可重映射引脚 RP3 输出信号选择位					---0 0000	76, 162
RPOR2	—	—	—	可重映射引脚 RP2 输出信号选择位					---0 0000	76, 161
RPOR1	—	—	—	可重映射引脚 RP1 输出信号选择位					---0 0000	76, 161
RPOR0	—	—	—	可重映射引脚 RP0 输出信号选择位					---0 0000	76, 161

图注: x = 未知, u = 不变, — = 未实现, q = 值取决于具体条件, r = 保留。**粗体**表示共用的快速操作 SFR。

注

- 1: PC 的 bit 21 仅在串行编程（Serial Programming, SP）模式下可用。
- 2: 双速启动使能时复位值为 0，禁止时为 1。
- 3: 仅当 SSPxCON2<3:0> = 1001 时才能访问 SSPxMSK 寄存器。
- 4: 当 MSSP 模块工作在 I²C™ 从模式下时这些位的备用名称和定义。详情请参见第 19.5.3.2 节“地址掩码模式”。
- 5: 这些位和 / 或寄存器仅在 44 引脚器件上可用；在其他器件上，它们未实现并读为 0。所示为 44 引脚器件的复位值。
- 6: PMADDRH/PMDO1H 和 PMADDRL/PMDO1L 寄存器对共用相同的物理寄存器和地址，但具有由模块工作模式所决定的不同功能。更多信息，请参见第 11.1.2 节“数据寄存器”。
- 7: 仅当相应的引脚没有被配置为用于主振荡器功能时，才实现 TRISA6 和 TRISA7 位。

PIC18F46J50 系列

6.3.6 STATUS 寄存器

寄存器 6-2 中所示的 STATUS 寄存器包含 ALU 的算术运算状态。和任何其他寄存器一样，STATUS 寄存器可以作为任何指令的操作数。如果一条影响 Z、DC、C、OV 或 N 位的指令以 STATUS 寄存器作为目标寄存器，则会禁止对这 5 位进行写操作。

这些位根据器件逻辑被置 1 或清零。因此，当执行一条把 STATUS 寄存器作为目标寄存器的指令后，运行结果可能与预想的不同。例如，CLRF STATUS 将 Z 位置 1

并保持其余位不变。然后 STATUS 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 STATUS 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。

关于其他不会影响状态位的指令，请参见表 28-2 和表 28-3 中的指令集汇总。

注： 在减法运算中，C 和 DC 位分别作为借位位和半借位位。

寄存器 6-2: STATUS 寄存器（位于快速操作存储区，地址 FD8h）

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 = POR 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 此位用于有符号的算术运算（以二进制补码方式进行）。它可以表示结果是否为负（ALU MSB = 1）。 1 = 结果为负 0 = 结果为正
bit 3	OV： 溢出标志位 此位用于有符号的算术运算（以二进制补码方式进行）。它表明运算结果溢出了 7 位二进制数的范围，溢出导致符号位（bit 7）发生改变。 1 = 有符号算术运算中发生溢出（本次运算） 0 = 未发生溢出
bit 2	Z： 全零标志位 1 = 算术运算或逻辑运算的结果为零 0 = 算术运算或逻辑运算的结果不为零
bit 1	DC： 半进位 / 借位位 ⁽¹⁾ 对于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令： 1 = 结果的第 4 个低位发生了进位 0 = 结果的第 4 个低位未发生进位
bit 0	C： 进位 / 借位位 ⁽²⁾ 对于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令： 1 = 结果的最高位发生了进位 0 = 结果的最高位未发生进位

- 注** 1: 对于借位，极性是相反的。减法是通过加上第二个操作数的二进制补码来执行的。对于移位指令（RRF 和 RLF），此位来自源寄存器的 bit 4 或 bit 3。
- 2: 对于借位，极性是相反的。减法是通过加上第二个操作数的二进制补码来执行的。对于移位指令（RRF 和 RLF），此位来自源寄存器的最高位或最低位。

6.4 数据寻址模式

注： 当使能 PIC18 扩展指令集时，核心 PIC18 指令集中某些指令的执行方式会发生改变。更多信息，请参见第 6.6 节“数据存储器和扩展指令集”。

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

这些寻址模式为：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

当使能了扩展指令集（XINST 配置位 = 1）时，还可使用另外一种寻址模式，即立即数变址寻址模式。第 6.6.1 节“使用立即数偏移量进行变址寻址”将更详细讨论它的操作。

6.4.1 固有寻址和立即数寻址

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，要么仅隐式地针对一个寄存器进行操作。此寻址模式就是固有寻址。例如指令 SLEEP、RESET 和 DAW。

其他指令的工作方式与此类似，但需要操作码中有其他显式的参数。由于需要一些立即数作为参数，这种寻址模式被称为立即数寻址。例如 ADDLW 和 MOVLW，它们分别向 W 寄存器加或移入立即数值。其他立即数寻址指令，例如 CALL 和 GOTO，它们包括一个 20 位的程序存储器地址。

6.4.2 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。这些选项由指令附带的参数指定。

在核心 PIC18 指令集中，针对位和针对字节的指令默认情况下使用直接寻址。所有这些指令都包含某个 8 位的立即数地址作为其 LSB。此地址指定数据 RAM 的某个存

储区中寄存器的地址（第 6.3.4 节“通用寄存器文件”）或快速操作存储区（第 6.3.3 节“快速操作存储区”）中作为指令数据源的单元地址。

快速操作 RAM 位“a”决定地址的解析方式。当“a”为 1 时，BSR（第 6.3.2 节“存储区选择寄存器”）的内容将和指令中的直接地址一起用于确定寄存器的完整 12 位地址。当“a”为 0 时，此直接地址将被解析为快速操作存储区中的一个寄存器。使用快速操作 RAM 的寻址模式有时也被称为直接强制寻址模式。

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

保存操作结果的目标寄存器由目标位“d”确定。当“d”为 1 时，结果被回源寄存器并覆盖原来的内容。当“d”为 0 时，结果被存储在 W 寄存器中。没有“d”参数的指令的目标寄存器隐含在指令中，这些指令的目标寄存器是在操作的目标寄存器或 W 寄存器。

6.4.3 间接寻址

间接寻址允许用户访问数据存储器中的单元而无需在指令中给出一个固定的地址。这种寻址模式是通过使用文件选择寄存器（File Select Register, FSR）作为指向要读写单元的指针实现的。由于 FSR 本身作为 SFR 位于 RAM 中，因此也可在程序控制中直接操作它们。这使得 FSR 对于在数据存储器中实现诸如表和数组等数据结构时非常有用。

也可以使用间接文件操作数（Indirect File Operand, INDF）进行寄存器间接寻址。这种操作允许自动递增、递减或偏移指针，从而自动操作指针的值。它通过使用循环提高代码执行效率，如例 6-5 所示的清零整个 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的堆栈指针操作。

例 6-5: 如何使用间接寻址清零 RAM (BANK 1)

	LFSR	FSR0, 0x100 ;
NEXT	CLRF	POSTINC0 ; Clear INDF
		; register then
		; inc pointer
	BTFSS	FSR0H, 1 ; All done with
		; Bank1?
	BRA	NEXT ; NO, clear next
CONTINUE		; YES, continue

PIC18F46J50 系列

6.4.3.1 FSR 寄存器和 INDF 操作数 (INDF)

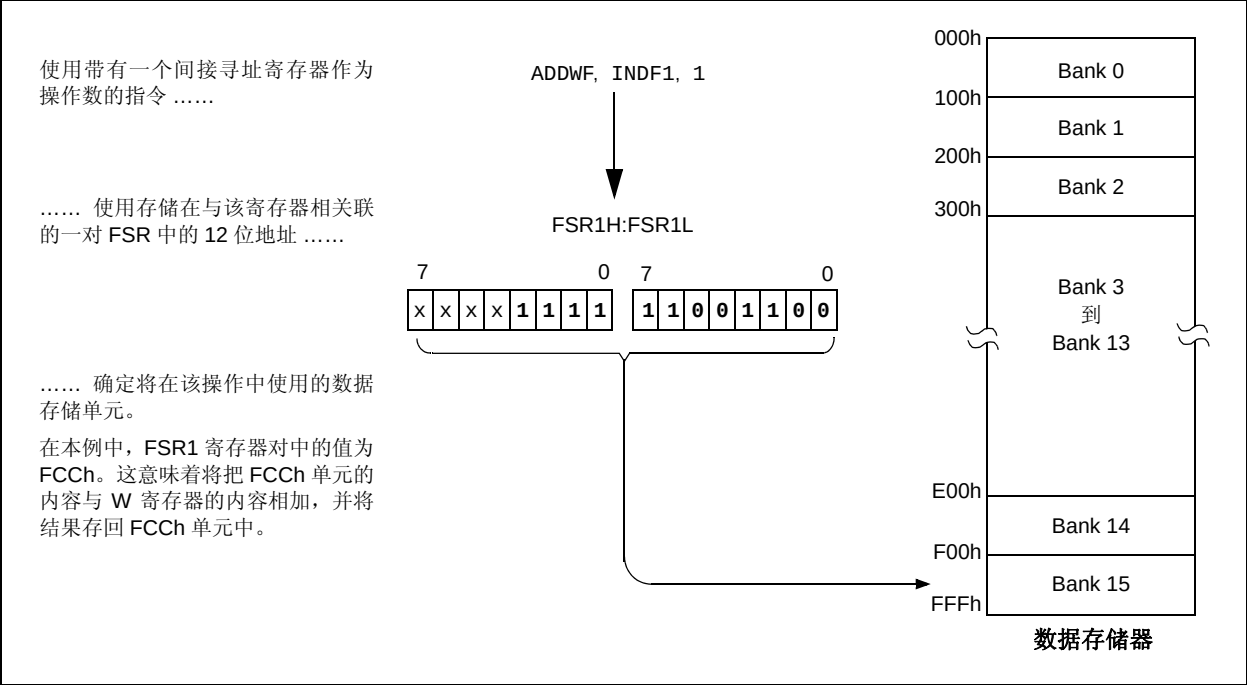
间接寻址的核心是三组寄存器: FSR0、FSR1 和 FSR2。每组寄存器都含有一对 8 位寄存器: FSRnH 和 FSRnL。FSRnH 寄存器的高 4 位未使用, 因此每对 FSR 只保存一个 12 位值, 从而可以线性寻址数据存储器的整个空间。因此, FSR 寄存器对被用作数据存储器的地址指针。

间接寻址是通过一组 INDF 操作数 (INDF0 到 INDF2) 完成的。这些操作数可被看作“虚拟”寄存器: 它们被映射到 SFR 空间而不是物理实现的。对特定的 INDF 寄

存器执行读或写操作实际上访问的是与之对应的一对 FSR 寄存器。例如, 读 INDF1 就是读 FSR1H:FSR1L 指向地址中的数据。使用 INDF 寄存器作为操作数的指令实际上使用相应FSR的内容作为指向指令目标地址的指针。INDF 操作数只是使用指针的一种简便方法。

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

图 6-8: 间接寻址



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

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

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

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

使用 POSTDEC、POSTINC 和 PREINC 对 FSR 进行操作会影响整对寄存器：即，FSRnL 寄存器从 FFh 溢出到 00h 并向 FSRnH 寄存器进位。但这些操作的结果不会更改 STATUS 寄存器中的任何标志位（如 Z、N 和 OV 等）。

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

6.4.3.3 通过 FSR 对其他 FSR 进行操作

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

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

由于 FSR 是映射到 SFR 空间中的物理寄存器，所以可以通过所有直接寻址来操作它们。用户在使用这些寄存器时应特别小心，尤其是在代码使用间接寻址时。

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

6.5 程序存储器和扩展指令集

程序存储器的操作不受扩展指令集的影响。

使能扩展指令集会将 5 条额外的双字命令添加到现有的 PIC18 指令集中：即 ADDFSR、CALLW、MOVSF、MOVSS 和 SUBFSR。这些指令如第 6.2.4 节“双字指令”中所述执行。

6.6 数据存储器和扩展指令集

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

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

6.6.1 使用立即数偏移量进行变址寻址

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

使用扩展指令集时，这种寻址模式有如下要求：

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

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

6.6.2 受立即数变址寻址模式影响的指令

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

此外，如果针对字节和针对位的指令不使用快速操作存储区（快速操作 RAM 位为 1）或包含 60h 以上的文件地址，它们也不受影响。符合这些条件的指令会像以前一样执行。图 6-9 给出了当使能扩展指令集时，各种寻址模式之间的对比。

那些想要在立即数变址寻址模式中使用针对位或针对字节的指令的用户，应该注意此模式下汇编语法的改变。在第 28.2.1 节“扩展指令的语法”中对此进行了更详细的说明。

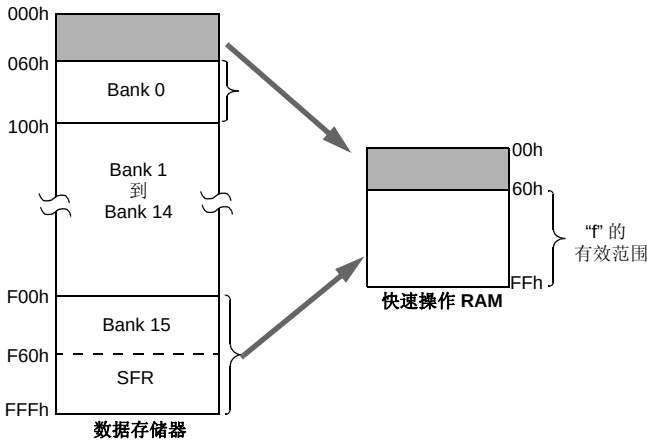
图 6-9: 针对位和针对字节的指令的寻址模式对比（使能了扩展指令集）

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

当 a = 0 且 f ≥ 60h 时:

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

不可用此模式寻址地址低于 060h 的单元。

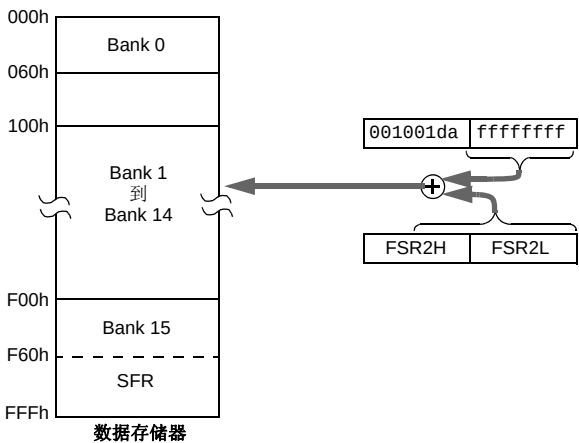


当 a = 0 且 f ≤ 5Fh 时:

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

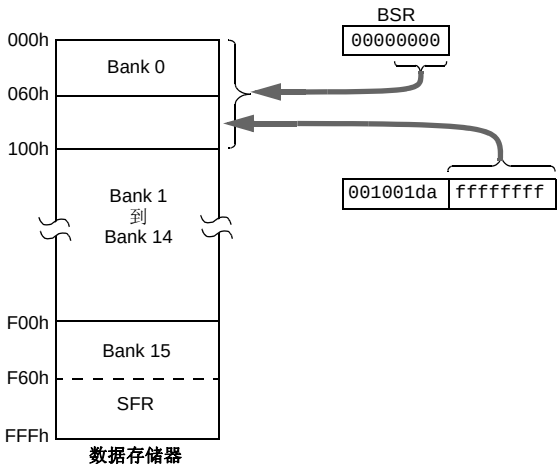
注意在此模式中，正确的语法如下:

ADDWF [k], d
其中“k”就是“f”。



当 a = 1 (f 可为任何值) 时:

指令以直接寻址模式（也称为直接长地址寻址模式）执行。“f”被解析为数据存储空间的 16 个存储区之一中的一个单元地址。存储区由存储区选择寄存器（BSR）指定。此地址可以位于数据存储空间的任何已实现存储区中。



PIC18F46J50 系列

6.6.3 在立即数变址模式下映射快速操作存储区

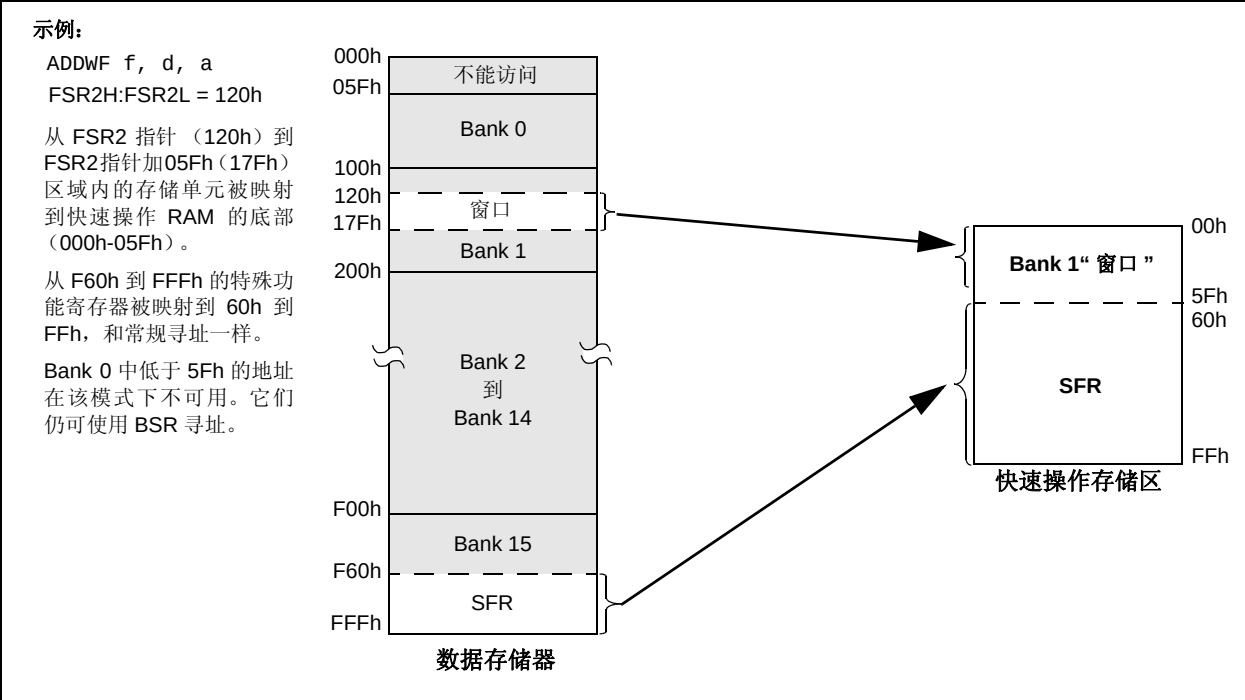
使用立即数变址寻址模式能有效改变快速操作 RAM 低地址单元（00h 到 5Fh）的映射方式。此模式映射 Bank 0 的内容和由用户定义的、可位于数据存储空间中任何地方的“窗口”内容，而不仅仅映射 Bank 0 底部的内容。FSR2 的值定义映射到窗口的地址的下边界，而上边界则由 FSR2 加 95（5Fh）决定。地址为 5Fh 以上的快速操作 RAM 的映射方法如前所述（见第 6.3.3 节“快速操作存储区”）。图 6-10 给出了在此寻址模式下重映射快速操作存储区的示例。

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

6.6.4 立即数变址模式中的 BSR

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

图 6-10: 使用立即数变址寻址模式重映射快速操作存储区



7.0 闪存程序存储器

正常操作期间，闪存程序存储器都是完全可以读写和擦除的。

读程序存储器时，每次读取一个字节。写程序存储器时，每次写一个 64 字节或 2 字节的块。擦除程序存储器时，每次擦除一个 1024 字节的块。用户代码不能执行批量擦除操作。

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

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

7.1 表读与表写

为了读写程序存储器，有两个操作可供处理器在程序存储空间和数据 RAM 之间传送字节：

- 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传送数据。

表读操作从程序存储器获取数据并将其放入数据 RAM 空间。图 7-1 显示了程序存储器和数据 RAM 之间的一次表读操作。

表写操作将数据存储空间中的数据存储到程序存储器的保持寄存器中。第 7.5 节“写闪存程序存储器”详细介绍了将保持寄存器的内容写入程序存储器的过程。图 7-2 显示了程序存储器和数据 RAM 之间的一次表写操作。

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

图 7-1： 表读操作

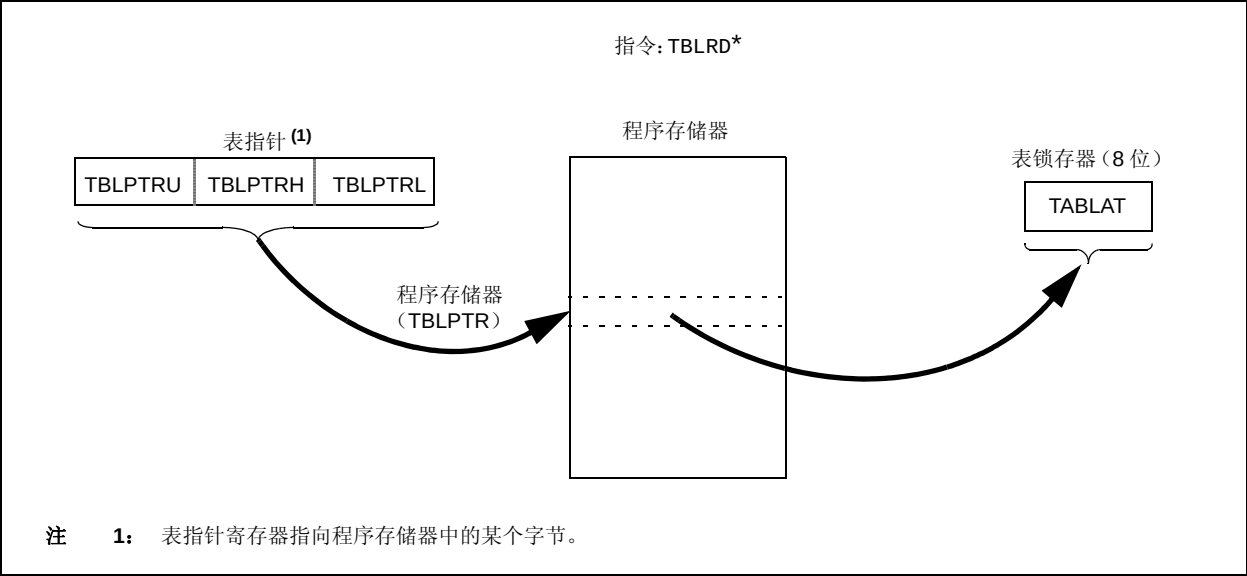
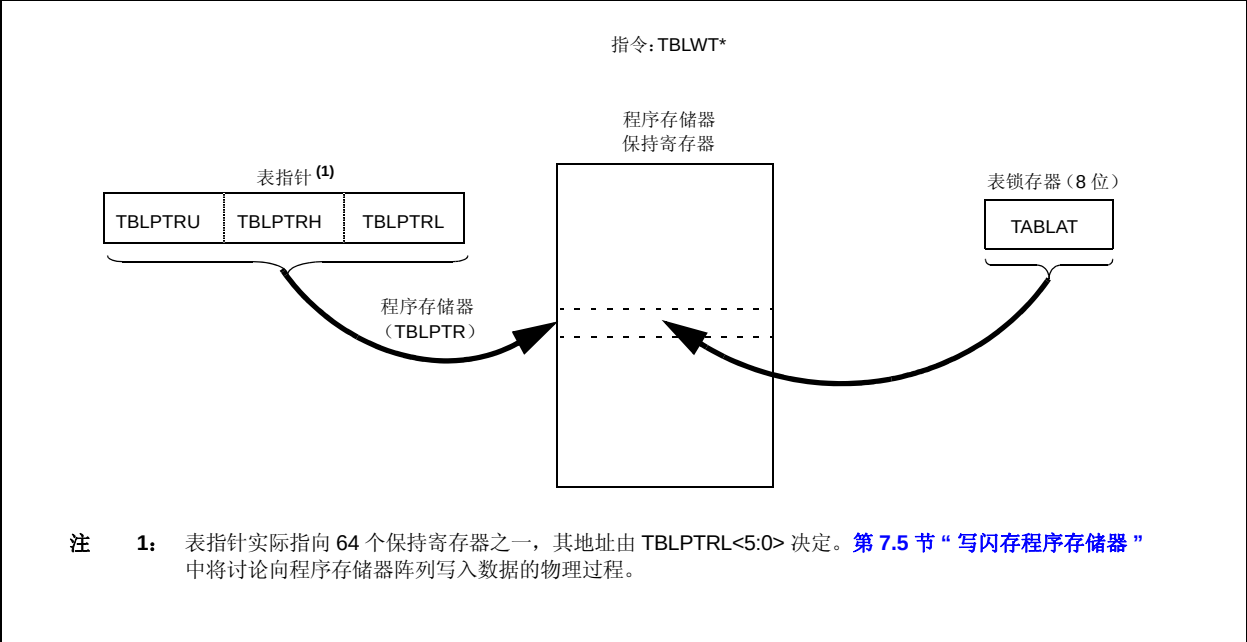


图 7-2: 表写操作



7.2 控制寄存器

TBLRD和TBLWT指令要用到几个控制寄存器。这些寄存器包括:

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

7.2.1 EECON1 和 EECON2 寄存器

EECON1 寄存器 (寄存器 7-1) 是存储器访问的控制寄存器。EECON2 寄存器不是实际存在的寄存器, 专用于存储器的擦写操作。读 EECON2 将得到全 0。

如果 WPROG 位置 1, 则执行 WR 命令时, 允许按字编程 2 个字节。如果该位清零, 则执行 WR 命令时, 将编程 64 字节的块。

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

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

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

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

寄存器 7-1: EECON1: EEPROM 控制寄存器 1（位于快速操作存储区，地址 FA6h）

U-0	U-0	R/W-0	R/W-0	R/W-x	R/W-0	R/S-0	U-0
—	—	WPROG	FREE	WRERR	WREN	WR	—
bit 7							bit 0

图注:	S = 可置 1 位（无法用软件清零）		
R = 可读位	W = 可写位		U = 未实现位，读为 0
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7-6 未实现：读为 0
- bit 5 **WPROG**: 单字宽编程位
 1 = 在执行下一条 **WR** 命令时编程 2 个字节
 0 = 在执行下一条 **WR** 命令时编程 64 个字节
- bit 4 **FREE**: 闪存擦除使能位
 1 = 在下一条 **WR** 命令时执行擦除操作（擦除操作完成后由硬件清零）
 0 = 仅执行写操作
- bit 3 **WRERR**: 闪存程序存储器错误标志位
 1 = 写操作提早终止（由于正常操作中自定时编程期间的任何复位，或不合法的写操作）
 0 = 写操作完成
- bit 2 **WREN**: 闪存程序存储器写使能位
 1 = 允许对闪存程序存储器的写周期
 0 = 禁止对闪存程序存储器的写周期
- bit 1 **WR**: 写控制位
 1 = 启动程序存储器的擦除周期或写周期
 （操作是自定时的，一旦写操作完成，该位即由硬件清零。用软件只能将 **WR** 位置 1（不能清零）。）
 0 = 写周期完成
- bit 0 未实现：读为 0

PIC18F46J50 系列

7.2.2 表锁存寄存器 (TABLAT)

表锁存器 (TABLAT) 是映射到特殊功能寄存器 (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。

7.2.3 表指针寄存器 (TBLPTR)

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

TBLRD 和 TBLWT 指令要使用表指针寄存器 TBLPTR。这些指令可以基于表操作以 4 种方法之一更新 TBLPTR。

表 7-1 提供了这些操作。这些操作只会影响 TBLPTR 的低 21 位。

7.2.4 表指针边界

TBLPTR 用于读、写和擦除闪存程序存储器。

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

当执行 TBLWT 时, 表指针寄存器的低 7 位 (LSb) (TBLPTR<6:0>) 决定要写入程序存储器的哪个保持寄存器 (共有 64 个)。当程序存储器的定时写入 (通过 WR 位) 开始时, TBLPTR 的高 12 位 (MSb) (TBLPTR<21:10>) 决定要写入哪个程序存储器块 (每块 1024 字节)。更多详细信息, 请参见第 7.5 节 “写闪存程序存储器”。

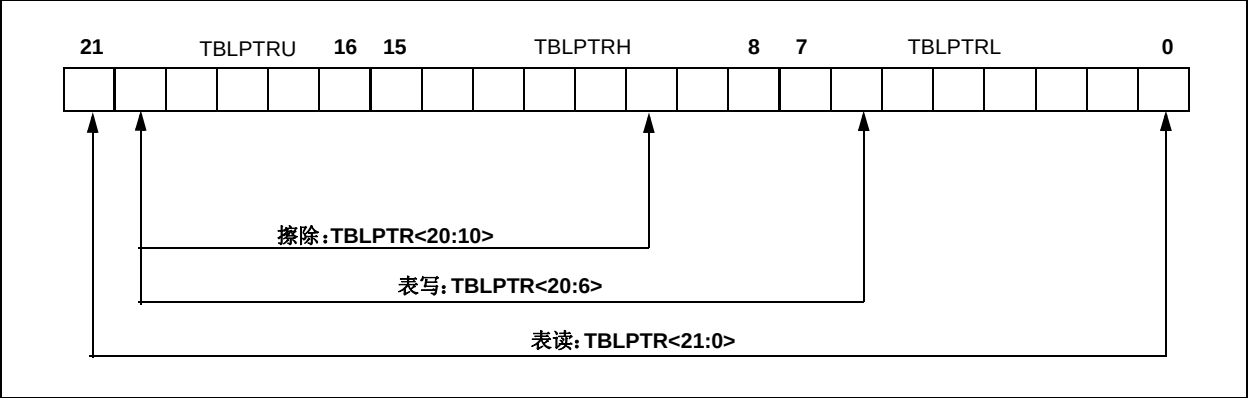
当执行擦除程序存储器时, 表指针寄存器的高 12 位指向将要擦除的 1024 字节块。LSb 被忽略。

图 7-3 说明了基于闪存程序存储器操作的 TBLPTR 相关边界。

表 7-1: 执行 TBLRD 和 TBLWT 指令的表指针操作

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

图 7-3: 基于操作的表指针边界



7.3 读闪存程序存储器

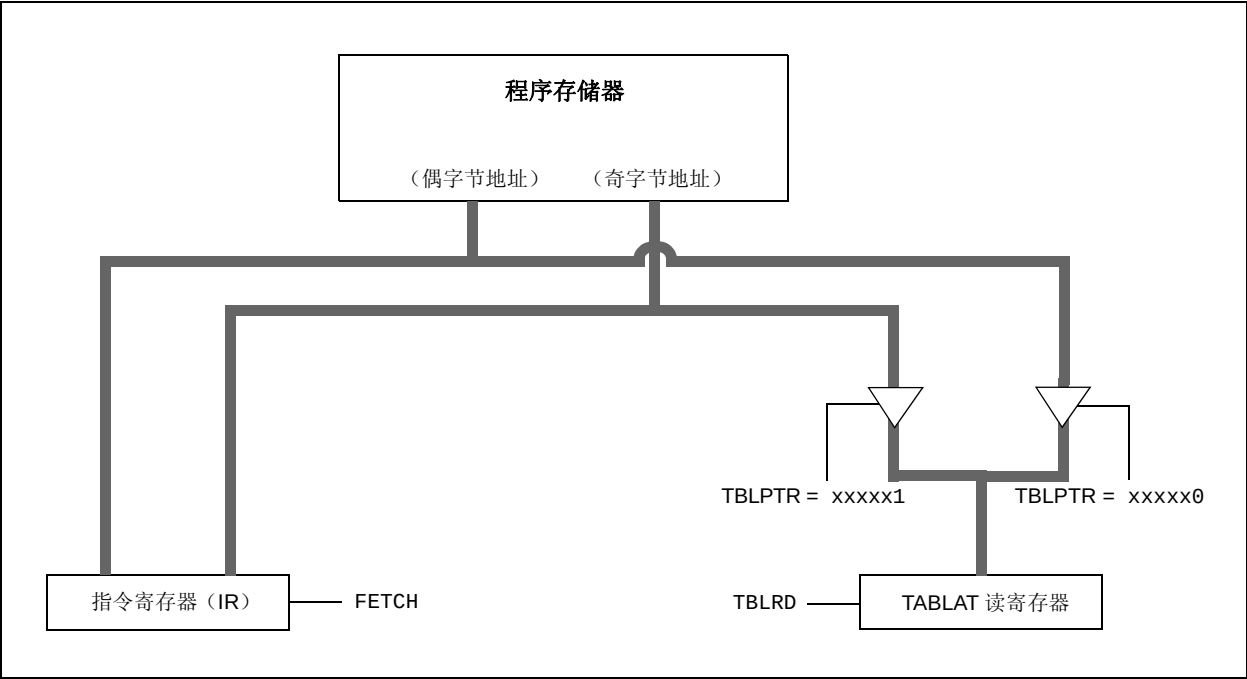
TBLRD 指令用于从程序存储器读取数据并放入数据 RAM。表读操作每次从程序存储器读取一个字节。

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

内部程序存储器通常以字为单位进行组织。由地址的 LSB 来选择字的高字节和低字节。

图 7-4 显示了内部程序存储器和 TABLAT 之间的接口。

图 7-4: 读闪存程序存储器



例 7-1: 读闪存程序存储器的一个字

```
MOV LW CODE_ADDR_UPPER ; Load TBLPTR with the base
MOV WF TBLPTRU ; address of the word
MOV LW CODE_ADDR_HIGH
MOV WF TBLPTRH
MOV LW CODE_ADDR_LOW
MOV WF TBLPTRL

READ_WORD

TBLRD*+ ; read into TABLAT and increment
MOV F TABLAT, W ; get data
MOV WF WORD_EVEN
TBLRD*+ ; read into TABLAT and increment
MOV F TABLAT, W ; get data
MOV WF WORD_ODD
```

7.4.1 闪存程序存储器擦除序列

擦除内部程序存储器块的过程如下:

1. 将要擦除的行地址装入表指针寄存器。
2. 将 WREN 和 FREE 位 (EECON1<2,4>) 置 1 以使能擦除操作。
3. 禁止中断。
4. 将 0x55 写入 EECON2。
5. 将 0xAA 写入 EECON2。
6. 将 WR 位置 1；这将开始擦除周期。
7. CPU 在擦除期间 (T_{IE}，见参数 D133B) 将会停止工作。
8. 重新允许中断。

6. 将 WR 位置 1；这将开始擦除周期。

7. CPU 在擦除期间 (T_{IE}, 见参数 D133B) 将会停止工作。

8. 重新允许中断。

	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, WREN	; enable write to memory
	BSF	EECON1, FREE	; enable Erase operation
	BCF	INTCON, GIE	; disable interrupts
必需的 序列	MOVLW	0x55	
	MOVWF	EECON2	; write 0x55
	MOVLW	0xAA	
	MOVWF	EECON2	; write 0xAA
	BSF	EECON1, WR	; start erase (CPU stall)
	BSF	INTCON, GIE	; re-enable interrupts

7.5 写闪存程序存储器

编程块大小为 32 个字或 64 字节。还支持一次编程一个字或 2 字节。

在内部使用表写指令将需要写入闪存的内容装入保持寄存器中。表写操作使用 64 个保持寄存器进行编程。

由于表锁存器 (TABLAT) 只是单字节寄存器，所以每次编程操作 TBLWT 指令可能需要执行 64 次 (如果 WPROG = 0)。因为只写保持寄存器，所以所有的表写操作实际上都是短写。更新 64 个保持寄存器后，必须写 EECON1 寄存器，以便启动长写周期开始编程操作。

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

由片上定时器控制写入的时间。写入 / 擦除电压由片上的电荷泵产生，该电荷泵可以在器件的电压范围内工作。

注

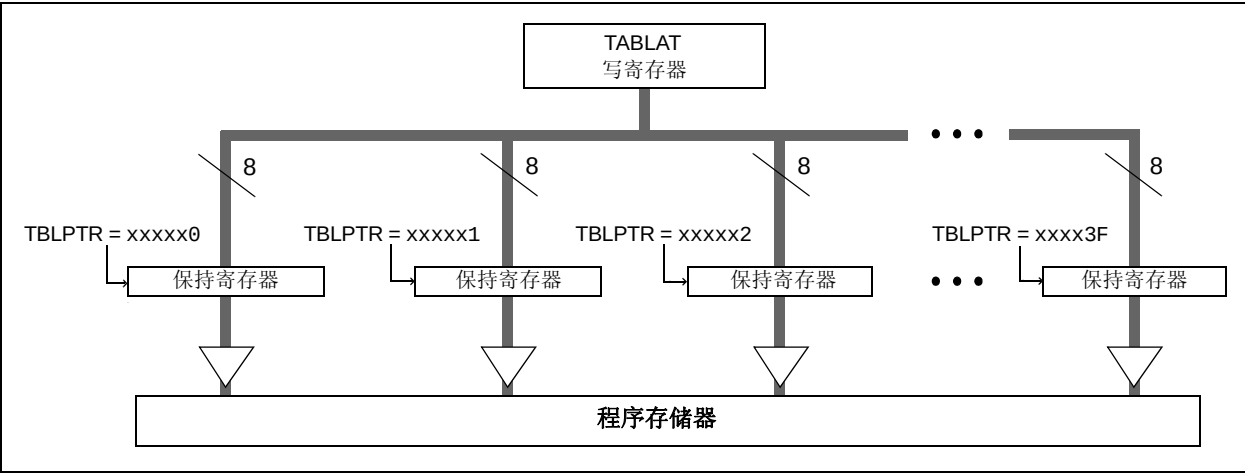
1:

与早期的 PIC® 器件不同，PIC18F46J50 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:

为了确保程序存储单元的耐擦写次数，在两次擦除操作之间不应每个闪存字节编程多于一次。在第二次尝试修改目标单元内容之前，必须对其所在页执行擦除操作或整个存储器的批量擦除。

图 7-5: 对闪存程序存储器的表写操作



7.5.1 闪存程序存储器写序列

对内部程序存储单元编程的过程如下：

1. 将 1024 字节读入 RAM。
2. 必要时更新 RAM 中的数据值。
3. 将要擦除的地址装入表指针寄存器。
4. 执行擦除过程。
5. 将要写入的第一个字节的地址装入表指针寄存器，表指针递减 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 以使能字节写操作。

8. 禁止中断。
9. 将 0x55 写入 EECON2。
10. 将 0xAA 写入 EECON2。
11. 将 WR 位置 1。这将开始写周期。
12. CPU 在写入期间 (T_{tw}, 见参数 D133A) 将会停止工作。
13. 重新允许中断。
14. 重复步骤 6 到 13，直到将所有 1024 个字节都写入程序存储器中。
15. 校验存储器 (表读)。

下一页中的例 7-3 给出了所需代码的示例。

注:

在将 WR 位置 1 前，表指针地址必须处于保持寄存器中的 64 字节预期地址范围内。

PIC18F46J50 系列

例 7-3: 写闪存程序存储器

ERASE_BLOCK	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	
	BSF	EECON1, WREN	; enable write to memory
	BSF	EECON1, FREE	; enable Erase operation
	BCF	INTCON, GIE	; disable interrupts
	MOVLW	0x55	
RESTART_BUFFER	MOVWF	EECON2	; write 0x55
	MOVLW	0xAA	
	MOVWF	EECON2	; write 0xAA
	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
	MOVLW	D'64'	
	MOVWF	COUNTER	
FILL_BUFFER	MOVLW	BUFFER_ADDR_HIGH	; point to buffer
	MOVWF	FSR0H	
	MOVLW	BUFFER_ADDR_LOW	
	MOVWF	FSR0L	
	...		; read the new data from I2C, SPI,
			; PSP, USART, etc.
	MOVLW	D'64'	; number of bytes in holding register
	MOVWF	COUNTER	
	MOVFF	POSTINC0, WREG	; get low byte of buffer data
	MOVWF	TABLAT	; present data to table latch
PROGRAM_MEMORY	TBLWT+*		; write data, perform a short write
			; to internal TBLWT holding register.
	DECFSZ	COUNTER	; loop until buffers are full
	BRA	WRITE_BYTE_TO_HREGS	
	BSF	EECON1, WREN	; enable write to memory
	BCF	INTCON, GIE	; disable interrupts
	MOVLW	0x55	
	MOVWF	EECON2	; write 0x55
	MOVLW	0xAA	
	MOVWF	EECON2	; write 0xAA
必需的序列	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

7.5.2 闪存程序存储器写序列（字编程）

PIC18F46J50系列器件具有允许编程单个字（2个字节）的功能。当 WPROG 位置 1 时，该功能使能。如果存储单元已经被擦除，需要通过以下序列来使能该功能：

1. 向表指针寄存器中装入要写入的数据的地址。
（它必须是偶地址。）

2. 通过执行表写指令将 2 个字节写入保持寄存器。
（第二个表写不执行后递增操作。）
3. 将 WREN 位（EECON1<2>）置 1 以使能写操作；将 WPROG 位（EECON1<5>）置 1 以选择字写模式。

4. 禁止中断。

5. 将 0x55 写入 EECON2。

6. 将 0xAA 写入 EECON2。

7. 将 WR 位置 1；这将开始写周期。

8. CPU 在写入期间（T_{iw}，见参数 D133A）将会停止工作。

9. 重新允许中断。

例 7-4: 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	; The table pointer must be loaded with an even address
	MOVWF	TBLPTRL	
	MOVLW	DATA0	; LSB of word to be written
	MOVWF	TABLAT	
	TBLWT*+		
	MOVLW	DATA1	; MSB of word to be written
	MOVWF	TABLAT	
	TBLWT*		; The last table write must not increment the table pointer!The table pointer needs to point to the MSB before starting the write operation.
PROGRAM_MEMORY			
	BSF	EECON1, WPROG	; enable single word write
	BSF	EECON1, WREN	; enable write to memory
	BCF	INTCON, GIE	; disable interrupts
必需的序列	MOVLW	0x55	
	MOVWF	EECON2	; write 0x55
	MOVLW	0xAA	
	MOVWF	EECON2	; write 0xAA
	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

PIC18F46J50 系列

7.5.3 写校验

根据具体应用，将写入存储器的值对照原始值进行校验是一个很好的编程习惯。在应用中，如果某些位的写次数接近规定极限值，就应该进行写校验。

7.5.4 意外终止写操作

如果由于意外事件（如掉电或意外复位）终止了写操作，应该对刚刚编程的存储单元进行校验，如有必要，

还要重新进行编程。如果写操作在正常操作期间因 MCLR 复位或 WDT 超时复位而中断，用户可以检查 WRERR 位以确定是否需要重写该单元。

7.6 代码保护期间闪存程序存储器的操作

关于闪存程序存储器代码保护的详细信息，请参见第 27.6 节“程序校验和代码保护”。

表 7-2: 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>）					69
TBPLTRH	程序存储器表指针高字节（TBLPTR<15:8>）								69
TBLPTRL	程序存储器表指针低字节（TBLPTR<7:0>）								69
TABLAT	程序存储器表锁存器								69
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
EECON2	程序存储器控制寄存器 2（不是实际存在的寄存器）								71
EECON1	—	—	WPROG	FREE	WRERR	WREN	WR	—	71

图注： — = 未实现，读为 0。闪存程序存储器访问期间不使用阴影单元。

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（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 STATUS 寄存器中的任何标志。

通过硬件执行乘法运算只需要一个指令周期。硬件乘法器具有更高的计算吞吐量并缩短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 给出了硬件和软件乘法运算的比较，包括所需存储器空间和执行时间。

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参数时，实现该运算仅需一条指令。

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

例 8-1: 8 x 8 无符号乘法程序

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

例 8-2: 8 x 8 有符号乘法程序

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

表 8-1: 各种乘法运算的性能比较

程序	乘法实现方法	程序存储器 (字)	周期数 (最大)	时间		
				48 MHz 时	10 MHz 时	4 MHz 时
8 x 8 无符号	无硬件乘法	13	69	5.7 μs	27.6 μs	69 μs
	硬件乘法	1	1	83.3 ns	400 ns	1 μs
8 x 8 有符号	无硬件乘法	33	91	7.5 μs	36.4 μs	91 μs
	硬件乘法	6	6	500 ns	2.4 μs	6 μs
16 x 16 无符号	无硬件乘法	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 x 16 有符号	无硬件乘法	52	254	21.6 μs	102.6 μs	254 μs
	硬件乘法	35	40	3.3 μs	16.0 μs	40 μs

PIC18F46J50 系列

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 (RES<3:0>) 中。

公式 8-1: 16 x 16 无符号乘法算法

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

例 8-3: 16 x 16 无符号乘法程序

```

MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L ->
                      ; PRODH:PRODL

MOVFF PRODH, RES1
MOVFF PRODL, RES0

MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H ->
                      ; PRODH:PRODL

MOVFF PRODH, RES3
MOVFF PRODL, RES2

MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F

MOVF ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F

```

例 8-4 给出了一个 16 x 16 有符号乘法运算的指令序列。
公式 8-2 为所使用的算法。32 位结果存储在 4 个寄存器 (RES<3:0>) 中。要弄清参数的符号位，必须检查每个参数对的最高有效位 (MSb)，并做相应的减法。

公式 8-2: 16 x 16 有符号乘法算法

$$\begin{aligned} \text{RES3:RES0} &= \text{ARG1H:ARG1L} \cdot \text{ARG2H:ARG2L} \\ &= (\text{ARG1H} \cdot \text{ARG2H} \cdot 2^{16}) + \\ &\quad (\text{ARG1H} \cdot \text{ARG2L} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2H} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2L}) + \\ &\quad (-1 \cdot \text{ARG2H} < 7 > \cdot \text{ARG1H:ARG1L} \cdot 2^{16}) + \\ &\quad (-1 \cdot \text{ARG1H} < 7 > \cdot \text{ARG2H:ARG2L} \cdot 2^{16}) \end{aligned}$$

例 8-4: 16 x 16 有符号乘法程序

```

MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L ->
                      ; PRODH:PRODL

MOVFF PRODH, RES1
MOVFF PRODL, RES0

MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H ->
                      ; PRODH:PRODL

MOVFF PRODH, RES3
MOVFF PRODL, RES2

MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F

MOVF ARG1H, W
MULWF ARG2L          ; ARG1H * ARG2L ->
                      ; PRODH:PRODL

MOVF PRODL, W
ADDWF RES1, F        ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F
CLRF WREG
ADDWFC RES3, F

BTFSS ARG2H, 7        ; ARG2H:ARG2L neg?
BRA SIGN_ARG1         ; no, check ARG1
MOVF ARG1L, W
SUBWF RES2
MOVF ARG1H, W
SUBWFB RES3

SIGN_ARG1
BTFSS ARG1H, 7        ; ARG1H:ARG1L neg?
BRA CONT_CODE         ; no, done
MOVF ARG2L, W
SUBWF RES2
MOVF ARG2H, W
SUBWFB RES3

CONT_CODE
:

```

9.0 中断

PIC18F46J50 系列器件具有多个中断源及一个中断优先级功能，该功能可以给大多数中断源分配高优先级或者低优先级。高优先级中断向量位于 0008h，低优先级中断向量位于 0018h。高优先级中断事件可以中断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。这使得汇编器 / 编译器能够自动存放指定寄存器中的这些位。

通常，中断源有 3 个位用于控制其操作。这些位分别是：

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

通过将 IPEN 位（RCON<7>）置 1，可使能中断优先级功能。当使能中断优先级时，有 2 个全局中断允许位。将 GIEH 位（INTCON<7>）置 1，可允许所有优先级位置 1（高优先级）的中断。将 GIEH 和 GIEL 位（INTCON<7:6>）置 1，可允许优先级位清零（低优先级）的中断。当中断标志位、允许位及相应的全局中断允许位均被置 1 时，中断将根据优先级位的设置立即跳转到地址 0008h 或 0018h。也可以通过设置相应的中断允许位来禁止单个中断。

当 IPEN 位清零（默认状态）时，便会禁止中断优先级功能，此时中断是与 PIC® 中档器件兼容的。在兼容模式下，各个中断源的中断优先级位不起作用。INTCON<6> 是 PEIE 位，用于允许 / 禁止所有的外设中断源。INTCON<7> 是 GIE 位，用于允许 / 禁止所有中断源。在兼容模式下，所有中断均跳转到地址 0008h。

当响应中断时，全局中断允许位由硬件自动清零以禁止后续中断。清零后的 IPEN 位就是 GIE 位。如果使用了中断优先级，这个位就是 GIEH 位（若中断被配置为高优先级中断）或者 GIEL 位（若中断被配置为低优先级中断）。在执行中断服务程序时，应用程序固件不应尝试重新允许已被硬件清零的 GIEH 或 GIEL 位。高优先级中断源会中断低优先级中断。在处理高优先级中断时，低优先级中断将不被处理。

中断发生时，返回地址被压入堆栈，中断向量地址（0008h 或 0018h）被装入 PC。只要在中断服务程序（Interrupt Service Routine, ISR）中，就可以通过查询中断标志位来确定中断源。从中断处理程序返回前，必须用软件将中断标志位或对应的 PIE_x 允许位清零，以避免递归中断。

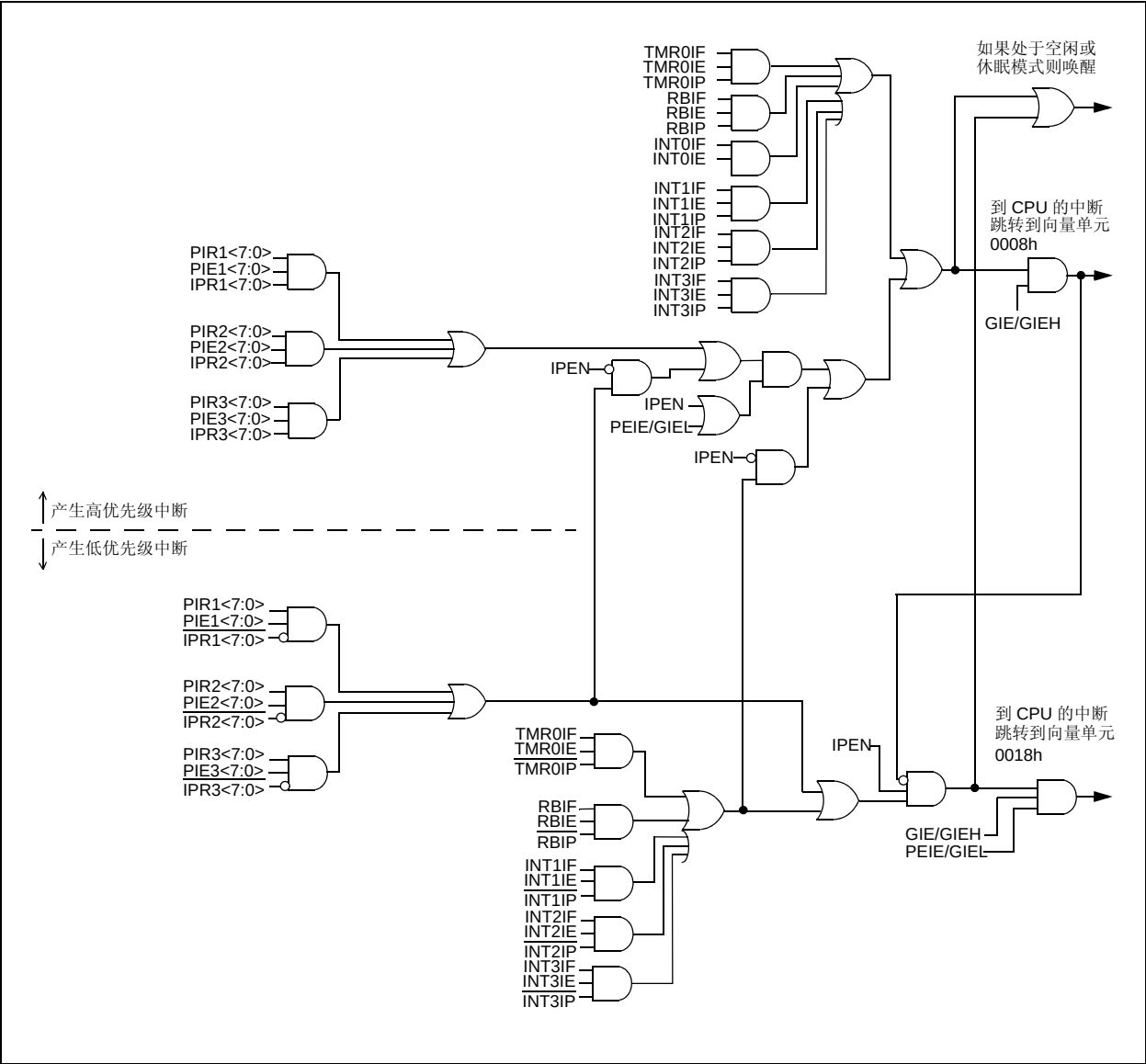
执行“从中断返回”指令 RETFIE 将退出中断服务程序，同时将 GIE 位（若使用中断优先级则为 GIEH 或 GIEL 位）置 1，从而重新允许中断。

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

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

PIC18F46J50 系列

图 9-1: PIC18F46J50 系列中断逻辑



9.1 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，包含各个中断允许位、优先级位和标志位。

注： 当中断条件产生时，不管相应的中断允许位或全局中断允许位的状态如何，中断标志位都将置 1。用户软件应在允许一个中断前，先将相应的中断标志位清零。这样做允许用软件查询中断标志位。

寄存器 9-1: INTCON: 中断控制寄存器（位于快速操作存储区，地址 FF2h）

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 = POR 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 = 允许所有未被屏蔽的中断

0 = 禁止所有中断

当 IPEN = 1 时:

1 = 允许所有高优先级中断（若 GIEL 也置 1 的话，还能允许低优先级中断）

0 = 禁止所有中断

bit 6 **PEIE/GIEL:** 外设 / 低优先级中断允许位

当 IPEN = 0 时:

1 = 允许所有未被屏蔽的外设中断（若 GIE 也置 1 的话）

0 = 禁止所有外设中断

当 IPEN = 1 时:

1 = 允许所有低优先级的中断（若 GIEH 也置 1 的话）

0 = 禁止所有低优先级的中断

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 = RB<7:4> 引脚中至少有一个引脚的电平状态发生了改变（必须用软件清零）

0 = 所有 RB<7:4> 引脚的电平状态都没有改变

注 1: 不匹配条件将继续把该位置 1。读取 PORTB 并等待 1 个 Tcy 将结束不匹配条件，并允许将该位清零。

PIC18F46J50 系列

寄存器 9-2: **INTCON2: 中断控制寄存器 2**（位于快速操作存储区，地址 **FF1h**）

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 = POR 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 0 = 根据各端口 TRIS 值使能 PORTB 上拉
bit 6	INTEDG0 : 外部中断 0 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
bit 5	INTEDG1 : 外部中断 1 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
bit 4	INTEDG2 : 外部中断 2 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
bit 3	INTEDG3 : 外部中断 3 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
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:	当中断条件产生时，不管相应的中断允许位或全局中断允许位的状态如何，中断标志位都将置 1。用户软件应在允许一个中断前，先将相应的中断标志位清零。这样做允许用软件查询中断标志位。

寄存器 9-3: INTCON3: 中断控制寄存器 3 (位于快速操作存储区, 地址 FF0h)

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 = POR 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 INT3 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:	当中断条件产生时, 不管相应的中断允许位或全局中断允许位的状态如何, 中断标志位都将置 1。用户软件应在允许一个中断前, 先将相应的中断标志位清零。这样做允许用软件查询中断标志位。

PIC18F46J50 系列

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，有三个外设中断请求（标志）寄存器（PIR1、PIR2 和 PIR3）。

- 注 1:** 当中断条件产生时，不管相应的中断允许位或全局中断允许位 GIE（INTCON<7>）的状态如何，中断标志位都将置 1。
- 2:** 用户软件应在允许中断前和处理完中断后，将相应的中断标志位清零。

寄存器 9-4: PIR1: 外设中断请求（标志）寄存器 1（位于快速操作存储区，地址 F9Eh）

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPIF ⁽¹⁾	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 = POR 时的值 1 = 置 1 0 = 清零 x = 未知

- bit 7 **PMPIF:** 并行主端口读 / 写中断标志位 ⁽¹⁾
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 = 一次 A/D 转换已完成（必须用软件清零）
0 = A/D 转换未完成
- bit 5 **RC1IF:** EUSART1 接收中断标志位
1 = EUSART1 接收缓冲区 RCREG1 已满（读取 RCREG1 时清零）
0 = EUSART1 接收缓冲区为空
- bit 4 **TX1IF:** EUSART1 发送中断标志位
1 = EUSART1 发送缓冲区 TXREG1 为空（写入 TXREG1 时清零）
0 = EUSART1 发送缓冲区已满
- bit 3 **SSP1IF:** 主同步串口 1 中断标志位
1 = 发送 / 接收已完成（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 **TMR2IF:** TMR2 与 PR2 匹配中断标志位
1 = TMR2 与 PR2 发生匹配（必须用软件清零）
0 = TMR2 与 PR2 未发生匹配
- bit 0 **TMR1IF:** TMR1 溢出中断标志位
1 = TMR1 寄存器已溢出（必须用软件清零）
0 = TMR1 寄存器未溢出

注 1: 这些位在 28 引脚器件上未实现。

寄存器 9-5: PIR2: 外设中断请求 (标志) 寄存器 2 (位于快速操作存储区, 地址 FA1h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	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 = POR 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 作为时钟输入 (必须用软件清零)
0 = 器件时钟正常工作
- 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 **USBIF:** USB 中断标志位
1 = USB 已请求中断 (必须用软件清零)
0 = 无 USB 中断请求
- bit 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 **HLVDIF:** 高 / 低压检测 (HLVD) 中断标志位
1 = 发生了高 / 低压条件 (必须用软件清零)
0 = 未发生 HLVD 事件
- bit 1 **TMR3IF:** TMR3 溢出中断标志位
1 = TMR3 寄存器已溢出 (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。

PIC18F46J50 系列

寄存器 9-6: **PIR3: 外设中断请求（标志）寄存器 3**（位于快速操作存储区，地址 FA4h）

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF
bit 7							bit 0

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

- bit 7

SSP2IF: 主同步串口 2 中断标志位
1 = 发送 / 接收已完成（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 接收缓冲区 RCREG2 已满（读取 RCREG2 时清零）
0 = EUSART2 接收缓冲区为空
- bit 4

TX2IF: EUSART2 发送中断标志位
1 = EUSART2 发送缓冲区 TXREG2 为空（写入 TXREG2 时清零）
0 = EUSART2 发送缓冲区已满
- bit 3

TMR4IF: TMR4 与 PR4 匹配中断标志位
1 = TMR4 与 PR4 发生匹配（必须用软件清零）
0 = TMR4 与 PR4 未发生匹配
- bit 2

CTMUIF: 充电时间测量单元中断标志位
1 = 发生了 CTMU 事件（必须用软件清零）
0 = 未发生 CTMU 事件
- bit 1

TMR3GIF: Timer3 门控事件中断标志位
1 = Timer3 门控事件已完成（必须用软件清零）
0 = Timer3 门控事件未完成
- bit 0

RTCCIF: RTCC 中断标志位
1 = 发生了 RTCC 中断（必须用软件清零）
0 = 未发生 RTCC 中断

9.3 PIE 寄存器

PIE 寄存器包含各外设中断的允许位。根据外设中断源的数量，有三个外设中断允许寄存器（PIE1、PIE2 和 PIE3）。当 IPEN = 0 时，要允许任一外设中断，必须将 PEIE 位置 1。

寄存器 9-7: PIE1: 外设中断允许寄存器 1（位于快速操作存储区，地址 F9Dh）

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 ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 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⁾
 1 = 允许 PMP 读 / 写中断
 0 = 禁止 PMP 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:** 主同步串口 1 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

注 1: 这些位在 28 引脚器件上未实现。

PIC18F46J50 系列

寄存器 9-8: **PIE2: 外设中断允许寄存器 2**（位于快速操作存储区，地址 **FA0h**）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	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 = POR 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 中断允许位
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

USBIE: USB 中断允许位
1 = 允许
0 = 禁止
- bit 3

BCL1IE: 总线冲突中断允许位（MSSP1 模块）
1 = 允许
0 = 禁止
- bit 2

HLVDIE: 高 / 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

寄存器 9-9: **PIE3: 外设中断允许寄存器 3**（位于快速操作存储区，地址 FA3h）

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	CTMUIE	TMR3GIE	RTCCIE
bit 7							bit 0

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

bit 7	SSP2IE: 主同步串口 2 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	CTMUIE: 充电时间测量单元（CTMU）中断允许位 1 = 允许 0 = 禁止
bit 1	TMR3GIE: Timer3 门控中断允许位 1 = 允许 0 = 禁止
bit 0	RTCCIE: RTCC 中断允许位 1 = 允许 0 = 禁止

PIC18F46J50 系列

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，有三个外设中断优先级寄存器（IPR1、IPR2 和 IPR3）。使用优先级位时，要求将中断优先级使能（IPEN）位置 1。

寄存器 9-10: IPR1: 外设中断优先级寄存器 1（位于快速操作存储区，地址 F9Fh）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PMP1P ⁽¹⁾	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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	PMP1P: 并行主端口读 / 写中断优先级位 ⁽¹⁾ 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: 主同步串口中断优先级位（MSSP1 模块） 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

注 1: 这些位在 28 引脚器件上未实现。

寄存器 9-11: IPR2: 外设中断优先级寄存器 2 (位于快速操作存储区, 地址 FA2h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	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 = POR 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: 比较器 2 中断优先级位 1 = 高优先级 0 = 低优先级
bit 5	C12IP: 比较器 1 中断优先级位 1 = 高优先级 0 = 低优先级
bit 4	USBIP: USB 中断优先级位 1 = 高优先级 0 = 低优先级
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	HLVDIP: 高 / 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

PIC18F46J50 系列

寄存器 9-12: IPR3: 外设中断优先级寄存器 3 (位于快速操作存储区, 地址 FA5h)

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	CTMUIP	TMR3GIP	RTCCIP
bit 7							bit 0

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

bit 7	SSP2IP: 主同步串口 2 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	CTMUIP: 充电时间测量单元 (CTMU) 中断优先级位 1 = 高优先级 0 = 低优先级
bit 1	TMR3GIP: Timer3 门控中断优先级位 1 = 高优先级 0 = 低优先级
bit 0	RTCCIP: RTCC 中断优先级位 1 = 高优先级 0 = 低优先级

9.5 RCON 寄存器

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

寄存器 9-13: RCON: 复位控制寄存器（位于快速操作存储区，地址 FD0h）

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	CM	RI	TO	PD	POR	BOR
bit 7							bit 0

图注:			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 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	CM: 配置不匹配标志位 位操作的详细信息，请参见 寄存器 5-1 。
bit 4	RI: RESET 指令标志位 位操作的详细信息，请参见 寄存器 5-1 。
bit 3	TO: 看门狗定时器超时标志位 位操作的详细信息，请参见 寄存器 5-1 。
bit 2	PD: 掉电检测标志位 位操作的详细信息，请参见 寄存器 5-1 。
bit 1	POR: 上电复位状态位 位操作的详细信息，请参见 寄存器 5-1 。
bit 0	BOR: 欠压复位状态位 位操作的详细信息，请参见 寄存器 5-1 。

PIC18F46J50 系列

9.6 INTx 引脚中断

INT0、INT1、INT2 和 INT3 引脚上的外部中断都是边沿触发的。如果 INTCON2 寄存器中相应的 INTEDGx 位被置 1 (= 1)，则为上升沿触发；如果该位被清零，则为下降沿触发。当 INTx 引脚上出现一个有效边沿时，相应的标志位 INTxIF 被置 1。通过清零相应的允许位 INTxIE，可禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将中断标志位 INTxIF 清零。

如果 INTxIE 位在进入功耗管理模式前被置 1，则所有的外部中断 (INT0、INT1、INT2 和 INT3) 均能将处理器从休眠和空闲管理模式唤醒。如果 GIEH (若配置为低优先级中断，为 GIEL) 位被置 1，则处理器在从休眠或空闲模式唤醒之后，会跳转到中断向量处执行程序。INT0 中断会将器件从深度休眠模式唤醒，但处理器不会跳转到中断向量处执行，而是从上电复位向量处开始执行程序。

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 (默认模式) 下，TMR0 寄存器的溢出 (FFh → 00h) 会使 TMR0IF 标志位置 1。在 16 位模式下，TMR0H:TMR0L 寄存器对的溢出 (FFFFh → 0000h)

会使 TMR0IF 标志位置 1。可以通过置 1/ 清零允许位 TMR0IE (INTCON<5>) 来允许 / 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。可以通过置 1/ 清零允许位 RBIE (INTCON<3>) 来允许 / 禁止该中断。PORTB 电平变化中断的优先级由中断优先级位 RBIP (INTCON2<0>) 中的值决定。

9.9 中断的现场保护

在中断期间，PC 的返回地址被保存在堆栈中。另外，WREG、STATUS 和 BSR 寄存器的值被压入快速返回堆栈。如果未使用从中断快速返回功能 (见第 6.3 节 “数据存储器构成”)，那么用户可能需要在进入中断服务程序前，保存 WREG、STATUS 和 BSR 寄存器的值。根据用户的具体应用，还可能需要保存其他寄存器的值。例 9-1 在执行中断服务程序期间，保存并恢复 WREG、STATUS 和 BSR 寄存器的值。

例 9-1: 将 STATUS、WREG 和 BSR 寄存器的值保存在 RAM 中

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

10.0 I/O 端口

根据选定的器件和使能的功能，最多有 5 个端口可供使用。I/O 端口的一些引脚与器件上外设功能的一个备用功能复用。通常而言，当某个外设使能时，其相关引脚可能不能用作通用 I/O 引脚。

每个端口都有三个工作寄存器。这些寄存器是：

- TRIS 寄存器（数据方向寄存器）
- 端口寄存器（读取器件引脚的电平）
- LAT 寄存器（数据锁存器）

与模拟功能（ANx 引脚）复用的引脚也具有与之相关联的 ANCON 寄存器位。

TRISx 寄存器用于控制哪些引脚应配置为数字输出（使能输出缓冲器），以及哪些引脚应保留为高阻态。向 TRIS 位写入 0 会将指定引脚配置为数字输出。向 TRIS 位写入 1 会禁止输出驱动器，因而引脚可以用作数字或模拟输入。0 类似于字母 O（Output（输出）的首字母），而 1 类似于字母 I（Input（输入）的首字母），这样可以便于记忆。

PORTx 寄存器可以用于读取从外部送到已配置为数字输入的引脚上的逻辑电平。当某个引脚配置为数字输入时，如果外部施加的电压大于该引脚的 V_{IH} 电压，则相应的端口位将读为 1。如果外部施加的电压小于 V_{IL} ，则 PORTx 位将读为 0。如果 I/O 引脚与模拟功能（ANx 引脚）复用，则还必须将相应的 PCFG 位（位于相应的 ANCONx 寄存器中）置 1，以正确读取外部施加给引脚的电压。具体信息，请参见下文对 ANCONx 寄存器的描述。

如果应用程序固件对 PORTx 寄存器进行写操作，这将导致相应的 LATx 寄存器发生更新。通常建议不要对 PORTx 寄存器执行读 - 修改 - 写指令（例如：BTG、BSF 和 BCF）。如果应用程序固件希望更改已配置为数字输出（TRIS 位 = 0）的引脚的输出状态，建议固件改为使用相应的 LATx 寄存器。

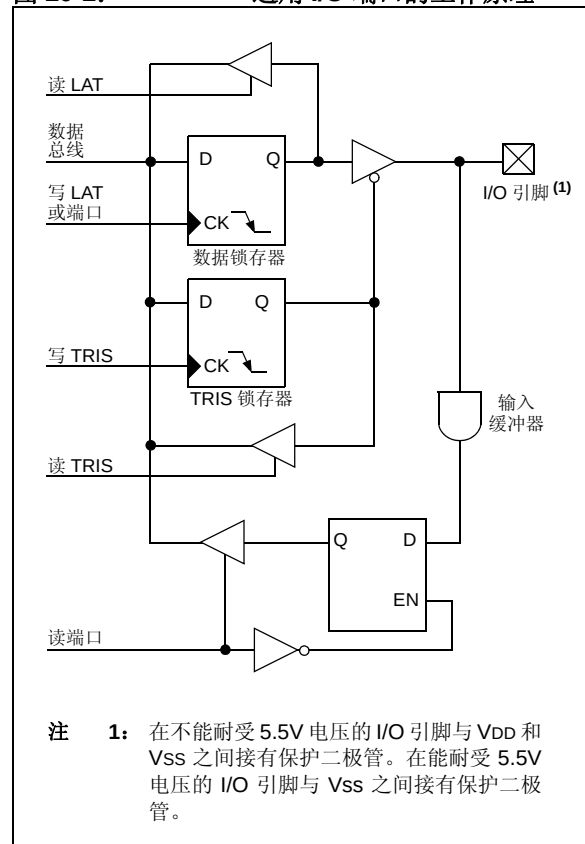
当某个引脚配置为数字输出（TRIS 位 = 0）时，LATx 寄存器将保存输出到该引脚上的数字值。向 LATx 位写入 1 会将输出引脚驱动为逻辑高电平状态。类似地，向 LAT 位写入 0 会将输出引脚驱动为逻辑低电平状态。对于 LATx 寄存器，可以安全地执行所有类型的读、写和读 - 修改 - 写指令。

ANCONx 寄存器用于将带有 ANx 模拟功能的引脚配置为数字输入或模拟输入模式。将 ANCONx 寄存器中的某个 PCFG 位置 1 会使能数字输入缓冲器，使 PORTx 寄存器的读操作可以正确地反映外部施加给数字输入引脚的电压。如果 PCFG 位清零，则数字输入缓冲器被禁止，以消除在对引脚施加值位于 V_{DD} 量程中间的模拟电压时产生的 CMOS 输入缓冲器跨导电流。这使得可以向引脚施加模拟输入电压（介于 V_{DD} 和 V_{SS} 之间），而不会增加器件的电流消耗。如果 ANCONx 寄存器中相应的 PCFG 位未置 1，则无论实际施加的外部电压如何，这都会导致该引脚的 PORTx 寄存器位读为 0。

在上电时，ANCONx 寄存器的默认状态是将 ANx 引脚配置为模拟模式（数字输入缓冲器关闭）。因此，要使用 ANx 引脚作为数字输入，应用程序固件必须先更新 ANCONx 寄存器。关于 ANCONx 寄存器的更多详细信息，请参见第 21.0 节“10 位模数转换器 (A/D) 模块”。

图 10-1 给出了通用 I/O 端口的简化模型，没有给出与其他外设的接口。

图 10-1: 通用 I/O 端口的工作原理



PIC18F46J50 系列

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能力。某些引脚上的输出驱动能力比其他引脚要高。同样，某些引脚可以承受高于 VDD 的输入电压。

10.1.1 引脚输出驱动能力

通用输出缓冲器使用 CMOS 晶体管实现，以支持轻负载时的轨到轨输出。满足各种应用需求的引脚组的输出引脚驱动能力是不同的。PORTB 和 PORTC 是为驱动较高的负载（例如 LED）而设计的。除此之外，所有其他端口都是为驱动较小的负载而设计的，通常只用作指示作用。表 10-1 汇总了输出功能。更多信息，请参见第 30.0 节“电气特性”。

表 10-1: 输出驱动能力

端口	驱动能力	说明
PORTA (RA6 除外)	最低	用作指示作用。
PORTD		
PORTE		
PORTB	高	适合以高电平驱动 LED。
PORTC		
PORTA<6>		

10.1.2 输入引脚和电压注意事项

用作器件输入的引脚的耐压能力取决于引脚的输入功能。仅用作数字输入的引脚能够承受最高 5.5V 的直流电压；这个电压值是数字逻辑电路的典型耐压值。而具有模拟输入功能的引脚只能承受最高为 VDD 的电压值。应避免在这些引脚上施加超过 VDD 的电压。表 10-2 汇总了输入能力。更多信息，请参见第 30.0 节“电气特性”。

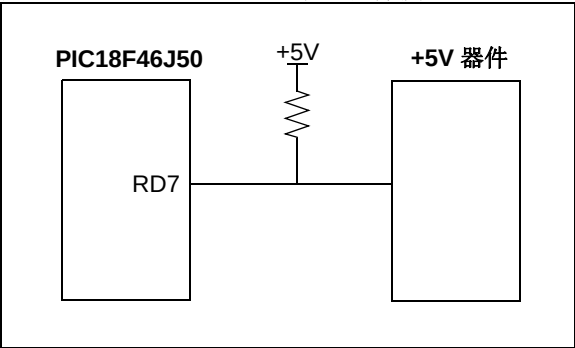
表 10-2: 输入电压

端口或引脚	可承受的最高输入电压	说明
PORTA<7:0>	VDD	只能承受小于等于 VDD 的输入电压。
PORTB<3:0>		
PORTC<2:0>		
PORTE<2:0>		
PORTB<7:4>	5.5V	可承受高于 VDD 的输入电压，可用于大部分标准逻辑电路。
PORTC<7:6>		
PORTD<7:0>		
PORTC<5:4>	(USB)	根据 USB 规范设计。

10.1.3 与 5V 系统的接口

虽然 PIC18F46J50 系列的 VDDMAX 是 3.6V，这些器件仍然能够与 5V 系统接口，即使目标系统的 VIH 高于 3.6V。这通过以下方式实现：为端口引脚连接一个上拉电阻（图 10-2），清零该引脚的 LAT 位，并对相应的 TRIS 位（图 10-1）进行操作，使之允许将线路上拉为高电平或将引脚驱动为低电平。只有最高可承受 5.5V 电压的端口引脚可用于此类接口（见第 10.1.2 节“输入引脚和电压注意事项”）。

图 10-2: +5V 系统硬件接口



例 10-1: 与 +5V 系统通信

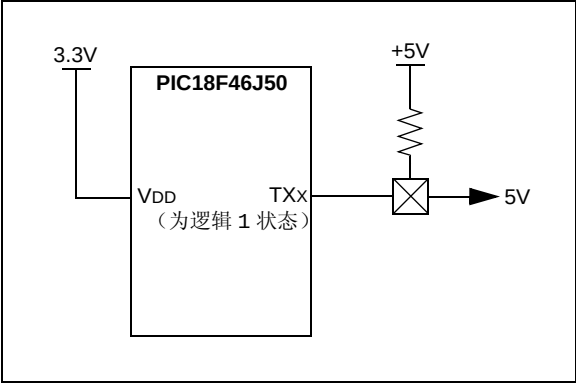
```
BCF LATD, 7 ; set up LAT register so
               ; changing TRIS bit will
               ; drive line low
BCF TRISD, 7 ; send a 0 to the 5V system
BSF TRISD, 7 ; send a 1 to the 5V system
```

10.1.4 漏极开路输出

几个外设的输出引脚还配备了一个可配置的漏极开路输出选项。这使外设可以与工作在较高电压下的外部数字逻辑通信，而无需使用电平转换器。

漏极开路选项在与 EUSART、MSSP 模块（在 SPI 模式下）和 ECCP 模块的数据和时钟输出相关的端口引脚上实现。通过将 ODCON 寄存器（寄存器 10-1、寄存器 10-2 和寄存器 10-3）中相应模块的漏极开路控制位置 1，可以有选择地使能该选项。将在讨论与这些外设复用的各个端口时讨论端口引脚的漏极开路配置。通过 PPS 模块进行映射的输出功能也可能使用漏极开路选项。漏极开路功能将遵循 PPS 模块中的 I/O 引脚分配。当需要漏极开路选项时，输出引脚也必须通过用户提供的外部上拉电阻连接到较高电压，最高为 5.5V（图 10-3）。当输出数字逻辑高电平信号时，它被上拉至较高电压。

图 10-3: 使用漏极开路输出（以 USART 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。虽然这种缓冲器形式对于许多输入类型都很有效，一些应用可能需要使用 TTL 电平信号来与外部逻辑器件进行接口。对于并行主端口（PMP）尤其如此，并行主端口有可能会与 TTL 电平逻辑或存储器器件进行接口。使用 PADCFG1 寄存器（寄存器 10-4）中的 PMPTTL 位，可选择将 PMP 的输入配置为使用 TTL 缓冲器。将该位置 1 时，PMP 的所有数据和控制输入引脚将配置为使用 TTL 缓冲器。默认情况下，这些 PMP 输入使用端口的 ST 缓冲器。

PIC18F46J50 系列

寄存器 10-1: **ODCON1: 外设漏极开路控制寄存器 1**（位于普通存储区，地址 F42h）

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7-2 未实现：读为 0
- bit 1 **ECCP2OD: ECCP2 漏极开路输出使能位**
1 = 使能漏极开路功能
0 = 禁止漏极开路功能
- bit 0 **ECCP1OD: ECCP1 漏极开路输出使能位**
1 = 使能漏极开路功能
0 = 禁止漏极开路功能

寄存器 10-2: **ODCON2: 外设漏极开路控制寄存器 2**（位于普通存储区，地址 F41h）

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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7-2 未实现：读为 0
- bit 1 **U2OD: USART2 漏极开路输出使能位**
1 = 使能漏极开路功能
0 = 禁止漏极开路功能
- bit 0 **U1OD: USART1 漏极开路输出使能位**
1 = 使能漏极开路功能
0 = 禁止漏极开路功能

寄存器 10-3: ODCON3: 外设漏极开路控制寄存器 3（位于普通存储区，地址 F40h）

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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-2 未实现：读为 0

bit 1 **SPI2OD**: SPI2 漏极开路输出使能位

1 = 使能漏极开路功能

0 = 禁止漏极开路功能

bit 0 **SPI1OD**: SPI1 漏极开路输出使能位

1 = 使能漏极开路功能

0 = 禁止漏极开路功能

寄存器 10-4: PADCFG1: 焊盘配置控制寄存器 1（位于普通存储区，地址 F3Ch）

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	RTSECSEL1 ⁽¹⁾	RTSECSEL0 ⁽¹⁾	PMPTTL
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-3 未实现：读为 0

bit 2-1 **RTSECSEL<1:0>**: RTCC 秒时钟输出选择位 ⁽¹⁾

11 = 保留；不要使用

10 = 选择 RTCC 引脚输出 RTCC 源时钟（可以是 INTRC、T1OSC 或 T1CKI，取决于 RTCOSC（CONFIG3L<1>）和 T1OSCEN（T1CON<3>）位设置）

01 = 选择 RTCC 引脚输出 RTCC 秒时钟

00 = 选择 RTCC 引脚输出 RTCC 闹钟脉冲

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 模块使用施密特触发器输入缓冲器

注 1: 要使能实际 RTCC 输出，需要将 RTCOE（RTCCFG<2>）位置 1。

PIC18F46J50 系列

10.2 PORTA、TRISA 和 LATA 寄存器

PORTA 是一个 7 位宽的双向端口。它也可用作一个 5 位或 6 位端口，取决于所选的振荡器模式。将 TRISA 某位置 1 (= 1) 时，会将 PORTA 的相应引脚设为输入（即，使相应的输出驱动器呈高阻态）。将 TRISA 某位清零 (= 0) 时，会将 PORTA 的相应引脚设为输出（即，将输出锁存器中的内容输出到选中引脚）。

读 PORTA 寄存器将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。

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

大部分 PORTA 引脚与模拟（ANx）功能复用。要将支持模拟功能的引脚用作数字输入，必须将 ANCON0 寄存器中相应的 PCFG 位置 1。

通过设置 CMxCON 寄存器中的相应位并将引脚配置为模拟输入，还可以将 RA0 至 RA3 引脚用作比较器输入。

注：上电复位（POR）时，RA5 和 RA<3:0> 被配置为模拟输入且读为 0。

所有 PORTA 引脚都具有全 CMOS 输出驱动器。
TRISA 寄存器控制着 PORTA 引脚的方向，即使它们被用作模拟输入。

例 10-2: 初始化 PORTA

CLRf	LATA	;Clearing the PORTA latches ;will cause the pins to drive ;low if configured as outputs
MOVLW	0x1F	;Configure AN0-AN4 pins
MOVFF	WREG,ANCON0	;for digital input mode
MOVLW	0xCF	;Example value used to ;initialize data direction
MOVWF	TRISA	;Set RA<3:0> as inputs ;RA4 is unimplemented ;RA5 as output ;RA6 and RA7 as inputs ;(unless overridden by osc settings)

表 10-3: PORTA I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RA0/AN0/C1INA/ ULPWU/PMA6/ RP0	RA0	1	I	TTL	PORTA<0> 数据输入；当使能模拟输入时被禁止。
		0	O	DIG	LATA<0> 数据输出；不受模拟输入影响。
	AN0	1	I	ANA	A/D 输入通道 0 和比较器 C1- 输入。POR 时的默认输入配置；不影响数字输出。
	C1INA	1	I	ANA	比较器 1 的输入 A。
	ULPWU	1	I	ANA	超低功耗唤醒输入。
	PMA6 ⁽¹⁾	0	O	DIG	并行主端口地址。
	RP0	1	I	ST	可重映射外设引脚 0 输入。
		0	O	DIG	可重映射外设引脚 0 输出。
RA1/AN1/C2INA/ PMA7/RP1	RA1	1	I	TTL	PORTA<1> 数据输入；当使能模拟输入时被禁止。
		0	O	DIG	LATA<1> 数据输出；不受模拟输入影响。
	AN1	1	I	ANA	A/D 输入通道 1 和比较器 C2- 输入。POR 时的默认输入配置；不影响数字输出。
	C2INA	1	I	ANA	比较器 1 的输入 A。
	PMA7 ⁽¹⁾	0	O	DIG	并行主端口地址。
	RP1	1	I	ST	可重映射外设引脚 1 输入。
		0	O	DIG	可重映射外设引脚 1 输出。
RA2/AN2/ VREF-/CVREF/ C2INB	RA2	0	O	DIG	LATA<2> 数据输出；不受模拟输入影响。当使能 CVREF 输出时被禁止。
		1	I	TTL	PORTA<2> 数据输入。当使能模拟功能时被禁止；当使能 CVREF 输出时被禁止。
	AN2	1	I	ANA	A/D 输入通道 2 和比较器 C2+ 输入。POR 时的默认输入配置；不受模拟输出影响。
	VREF-	1	I	ANA	A/D 和比较器低参考电压输入。
	CVREF	x	O	ANA	比较器参考电压输出。使能该功能将禁止数字 I/O。
	C2INB	1	I	ANA	比较器 2 的输入 B。
		0	O	ANA	CTMU 脉冲发生器充电器输出，用于 C2INB 比较器的输入。
RA3/AN3/VREF+/ C1INB	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；当使能模拟输入时被禁止。
	AN3	1	I	ANA	A/D 输入通道 3 和比较器 C1+ 输入。POR 时的默认输入配置。
	VREF+	1	I	ANA	A/D 和比较器高参考电压输入。
	C1INB	1	I	ANA	比较器 1 的输入 B。

图注: DIG = 数字电平输出；TTL = TTL 输入缓冲器；ST = 施密特触发器输入缓冲器；ANA = 模拟电平输入 / 输出；x = 无关位（TRIS 位不影响端口方向或在此可忽略）

注 1: 该位仅在 44 引脚器件上可用。

PIC18F46J50 系列

表 10-3: PORTA I/O 汇总 (续)

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RA5/AN4/SS1/ HLVDIN/RCV/ RP2	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；当使能模拟输入时被禁止。
	AN4	1	I	ANA	A/D 输入通道 4。POR 时的默认配置。
	SS1	1	I	TTL	MSSP1 的从选择输入。
	HLVDIN	1	I	ANA	高 / 低压检测外部跳变点参考输入。
	RCV	1	I	TTL	外部 USB 收发器 RCV 输入。
	RP2	1	I	ST	可重映射外设引脚 2 输入。
		0	O	DIG	可重映射外设引脚 2 输出。
OSC2/CLKO/ RA6	OSC2	x	O	ANA	主振荡器反馈输出连接（HS 模式）。
	CLKO	x	O	DIG	RC 和 EC 振荡器模式下的系统周期时钟输出（Fosc/4）。
	RA6	1	I	TTL	PORTA<6> 数据输入。
		0	O	DIG	LATA<6> 数据输出。
OSC1/CLKI/RA7	OSC1	1	I	ANA	主振荡器输入连接。
	CLKI	1	I	ANA	主时钟输入连接。
	RA7	1	I	TTL	PORTA<6> 数据输入。
		0	O	DIG	LATA<6> 数据输出。

图注: DIG = 数字电平输出；TTL = TTL 输入缓冲器；ST = 施密特触发器输入缓冲器；ANA = 模拟电平输入 / 输出；x = 无关位 (TRIS 位不影响端口方向或在此可忽略)

注 1: 该位仅在 44 引脚器件上可用。

表 10-4: 与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTA	RA7	RA6	RA5	—	RA3	RA2	RA1	RA0	87
LATA	LAT7	LAT6	LAT5	—	LAT3	LAT2	LAT1	LAT0	92
TRISA	TRIS7	TRIS6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	92
ANCON0	PCFG7 ⁽¹⁾	PCFG6 ⁽¹⁾	PCFG5 ⁽¹⁾	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	90
CMxCON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	90
CVRCON	CVREN	CVROE	CVRR	r	CVR3	CVR2	CVR1	CVR0	93

图注: — = 未实现，读为 0，r = 保留。PORTA 不使用阴影单元。

注 1: 这些位仅在 44 引脚器件上可用。

10.3 PORTB、TRISB 和 LATB 寄存器

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

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

PORTB 的每个引脚都具有内部弱上拉功能。通过一个控制位即可接通所有上拉。这可以通过清零 RBPU 位（INTCON2<7>）来实现。当端口引脚被配置为输出时，其弱上拉功能会自动关闭。POR 时会禁止上拉。集成的弱上拉由半导体结构构成，与分立电阻相类似，但又稍有不同。在空载 I/O 引脚上，弱上拉用于提供逻辑高电平指示，但不需要自始至终将引脚上拉至 VDD 电平。

注： POR 时，默认情况下 RB<3:0> 位被配置为模拟输入且读为 0；RB<7:4> 位则被配置为数字输入。

PORTB 的 4 个引脚（RB<7:4>）具有电平变化中断功能。仅当将这些引脚配置为输入时，才可使用此中断功能（即当 RB<7:4> 中的任何一个引脚被配置为输出时，该引脚将不再具有电平变化中断功能）。将输入引脚（RB<7:4>）上的电平与 PORTB 上次读入锁存器的旧值进行比较。对 RB<7:4> 上的“不匹配”输出进行逻辑或运算，产生 RB 端口电平变化中断，并将标志位 RBIF（INTCON<0>）置 1。

该中断可将器件从休眠模式或任何空闲模式唤醒。应用软件可通过以下步骤清零中断标志：

- 1. 读或写 PORTB（MOVFF (ANY), PORTB 指令除外）。
- 2. 等待一个指令周期（例如执行 NOP 指令）。
- 3. 清零标志位 RBIF。

不匹配条件会继续将标志位 RBIF 置 1。读 PORTB 将结束不匹配条件，并允许在一个指令周期的延时后将标志位 RBIF 清零。

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

RB5 引脚与 Timer0 模块的时钟输入以及比较器输出之一复用，成为 RB5/PMA0/KBI1/SDI1/SDA1/RP8 引脚。

例 10-3: 初始化 PORTB

MOVLW	0x08	; Initialize output data
MOVWF	LATB	; latch values for digital
		; output pins.
MOVLB	0x0F	; ANCONx registers are
		; not in access bank
BSF	ANCON1, PCFG12, BANKED	; Configure RB0/AN12 for digital input mode
BCF	ANCON1, PCFG10, BANKED	; Configure RB1/AN10 for analog input mode
MOVLW	0xC3	; RB0 configured as digital input
MOVWF	TRISB	; RB1 configured as analog input
		; RB2 configured as output low
		; RB3 configured as output high
		; RB4 configured as output low
		; RB5 configured as output low
		; RB6 configured as digital input
		; RB7 configured as digital input

PIC18F46J50 系列

表 10-5: PORTB I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RB0/AN12/ INT0/RP3	RB0	1	I	TTL	PORTB<0> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。当使能模拟输入时被禁止。(1)
		0	O	DIG	LATB<0> 数据输出；不受模拟输入影响。
	AN12	1	I	ANA	A/D 输入通道 12。(1)
	INT0	1	I	ST	外部中断 0 输入。
	RP3	1	I	ST	可重映射外设引脚 3 输入。
		0	O	DIG	可重映射外设引脚 3 输出。
RB1/AN10/ PMBE/RTCC/ RP4	RB1	1	I	TTL	PORTB<1> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。当使能模拟输入时被禁止。(1)
		0	O	DIG	LATB<1> 数据输出；不受模拟输入影响。
	AN10	1	I	ANA	A/D 输入通道 10。(1)
	PMBE ⁽³⁾	0	O	DIG	并行主端口字节使能输出。
	RTCC	0	O	DIG	实时时钟 / 日历输出。
	RP4	1	I	ST	可重映射外设引脚 4 输入。
		0	O	DIG	可重映射外设引脚 4 输出。
RB2/AN8/ CTED1/PMA3/ VMO/REFO/ RP5	RB2	1	I	TTL	PORTB<2> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。当使能模拟输入时被禁止。(1)
		0	O	DIG	LATB<2> 数据输出；不受模拟输入影响。
	AN8	1	I	ANA	A/D 输入通道 8。(1)
	CTED1	1	I	ST	CTMU 边沿 1 输入。
	PMA3 ⁽³⁾	0	O	DIG	并行主端口地址。
	VMO	0	O	DIG	外部 USB 收发器 D- 数据输出。
	REFO	0	O	DIG	参考输出时钟。
	RP5	1	I	ST	可重映射外设引脚 5 输入。
		0	O	DIG	可重映射外设引脚 5 输出。
RB3/AN9/ CTED2/PMA2/ VPO/RP6	RB3	0	O	DIG	LATB<3> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTB<3> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。当使能模拟输入时被禁止。(1)
	AN9	1	I	ANA	A/D 输入通道 9。(1)
	CTED2	1	I	ST	CTMU 边沿 2 输入。
	PMA2 ⁽³⁾	0	O	DIG	并行主端口地址。
	VPO	0	I	DIG	外部 USB 收发器 D+ 数据输出。
	RP6	1	I	ST	可重映射外设引脚 6 输入。
		0	O	DIG	可重映射外设引脚 6 输出。

图注: DIG = 数字电平输出；TTL = TTL 输入缓冲器；ST = 施密特触发器输入缓冲器；ANA = 模拟电平输入 / 输出；x = 无关位（TRIS 位不影响端口方向或在此可忽略）

- 注 1: POR 时，默认情况下，引脚被配置为模拟输入。要将这些引脚用作数字输入，需要先将 ANCONx 寄存器中的相应位置 1。
- 2: 当使能 ICSP™ 或 MPLAB® ICD 时，禁止所有其他引脚功能。
- 3: 该功能仅在 44 引脚器件上可用。

表 10-5: PORTB I/O 汇总 (续)

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RB4/PMA1/ KBI0/SCK1/ SCL1/RP7	RB4	0	O	DIG	LATB<4> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTB<4> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。当使能模拟输入时被禁止。(1)
	PMA1 ⁽³⁾	1	I	ST/TTL	并行从端口地址输入。
		0	O	DIG	并行主端口地址输出。
	KBI0	1	I	TTL	电平变化中断引脚。
	SCK1	1	I	ST	SPI 时钟输入 (MSSP1 模块)。
		0	O	DIG	SPI 时钟输出 (MSSP1 模块)。
	SCL1	1	I	I ² C/ SMBus	I ² C™ 时钟输入 (MSSP1 模块)。
		0	O	I ² C	I ² C 时钟输出 (MSSP1 模块)。
	RP7	1	I	ST	可重映射外设引脚 7 输入。
		0	O	DIG	可重映射外设引脚 7 输出。
RB5/PMA0/ KBI1/SDI1/ SDA1/RP8	RB5	0	O	DIG	LATB<5> 数据输出。
		1	I	TTL	PORTB<5> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。
	KBI1	1	I	TTL	电平变化中断引脚。
	PMA0 ⁽³⁾	1	I	ST/TTL	并行从端口地址输入。
		0	O	DIG	并行主端口地址输出。
	SDI1	1	I	ST	SPI 数据输入 (MSSP1 模块)。
	SDA1	1	I	I ² C/ SMBus	I ² C 数据输入 (MSSP1 模块)。
		0	O	I ² C	I ² C™/SMBus。
	RP8	1	I	ST	可重映射外设引脚 8 输入。
		0	O	DIG	可重映射外设引脚 8 输出。
RB6/KBI2/ PGC/RP9	RB6	0	O	DIG	LATB<6> 数据输出。
		1	I	TTL	PORTB<6> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。
	KBI2	1	I	TTL	电平变化中断引脚。
	PGC	x	I	ST	供 ICSP 和 ICD 工作使用的串行执行 (ICSP™) 时钟输入。(2)
	RP9	1	I	ST	可重映射外设引脚 9 输入。
		0	O	DIG	可重映射外设引脚 9 输出。

图注: DIG = 数字电平输出；TTL = TTL 输入缓冲器；ST = 施密特触发器输入缓冲器；ANA = 模拟电平输入 / 输出；x = 无关位 (TRIS 位不影响端口方向或在此可忽略)

注 1: POR 时，默认情况下，引脚被配置为模拟输入。要将这些引脚用作数字输入，需要先将 ANCONx 寄存器中的相应位置 1。

2: 当使能 ICSP™ 或 MPLAB® ICD 时，禁止所有其他引脚功能。

3: 该功能仅在 44 引脚器件上可用。

PIC18F46J50 系列

表 10-5: PORTB I/O 汇总 (续)

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RB7/KBI3/ PGD/RP10	RB7	0	O	DIG	LATB<7> 数据输出。
		1	I	TTL	PORTB<7> 数据输入；当 $\overline{\text{RBPU}}$ 位清零时使能弱上拉。
	KBI3	1	I	TTL	电平变化中断引脚。
	PGD	x	O	DIG	供 ICSP 和 ICD 工作使用的串行执行数据输出。(2)
		x	I	ST	供 ICSP 和 ICD 工作使用的串行执行数据输入。(2)
	RP10	1	I	ST	可重映射外设引脚 10 输入。
		0	O	ST	可重映射外设引脚 10 输出。

- 图注: DIG = 数字电平输出; TTL = TTL 输入缓冲器; ST = 施密特触发器输入缓冲器; ANA = 模拟电平输入 / 输出; x = 无关位 (TRIS 位不影响端口方向或在此可忽略)
- 注 1: POR 时, 默认情况下, 引脚被配置为模拟输入。要将这些引脚用作数字输入, 需要先将 ANCONx 寄存器中的相应位置 1。
- 2: 当使能 ICSP™ 或 MPLAB® ICD 时, 禁止所有其他引脚功能。
- 3: 该功能仅在 44 引脚器件上可用。

表 10-6: 与 PORTB 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	92
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	92
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	92
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	89
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	89
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	89
ADCON0	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	90

图注: — = 未实现, 读为 0。PORTB 不使用阴影单元。

10.4 PORTC、TRISC 和 LATC 寄存器

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

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

PORTC 与几种外设功能复用（见表 10-7）。这些引脚配有施密特触发器输入缓冲器。

当使能外设功能时，应小心定义每个 PORTC 引脚的 TRIS 位。有些外设会改写 TRIS 位的设置，将引脚重新定义为输出引脚或输入引脚。用户应查阅相应的外设章节来获取更多信息。

引脚 RC4 和 RC5 与 USB 模块复用。根据模块的配置，它们可用作片上 USB 收发器的差分数据线或来自外部 USB 收发器的数据输入。用作通用输入时，RC4 和 RC5 输入缓冲器都依赖于施加到 VUSB 引脚的电压值，而不是与所有其他通用 I/O 引脚一样依赖于 VDD。因此，如果要使用 RC4 或 RC5 通用输入功能，则不应将 VUSB 引脚悬空。

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

注： 上电复位时，PORTC 引脚（RC2、RC4 和 RC5 除外）都被配置为数字输入。RC2 默认为模拟输入（由 ANCON1 寄存器控制）。要将引脚 RC4 和 RC5 用作数字输入，必须禁止 USB 模块（UCON<3> = 0）并且必须禁止片上 USB 收发器（UCFG<3> = 1）。内部 USB 收发器在 POR 时是使能的。

外设对引脚的改写会影响 TRISC 寄存器的内容。尽管外设可能会改写一个或多个引脚，读 TRISC 总是会返回其当前的内容。

例 10-4: 初始化 PORTC

CLRF	LATC	; Initialize output data ; latch values for logic ; output low value.
MOVLB	0x0F	; ANCONx registers are ; not in access bank ;Configure RC2/AN11 for digital input mode
BSF	ANCON1, PCFG11, BANKED	;Disable USB transceiver to use RC4/RC5 as ;general purpose inputs
BCF	UCON, USBEN	;Disable USB module
BSF	UCFG, UTRDIS	;Disable USB transceiver
MOVLW	0x3F	; RC0 configured as digital input
MOVWF	TRISC	; RC1 configured as digital input ; RC2 configured as digital input ; RC4 configured as digital input ; RC5 configured as digital input ; RC6 configured as digital output ; RC7 configured as digital output

PIC18F46J50 系列

表 10-7: PORTC I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RC0/T1OSO/ T1CKI/RP11	RC0	1	I	ST	PORTC<0> 数据输入。
		0	O	DIG	LATC<0> 数据输出。
	T1OSO	x	O	ANA	Timer1 振荡器输出；当使能 Timer1 振荡器时被使能。禁止数字 I/O。
	T1CKI	1	I	ST	Timer1 数字时钟输入。
	RP11	1	I	ST	可重映射外设引脚 11 输入。
		0	O	DIG	可重映射外设引脚 11 输出。
RC1/T1OSI/ UOE/RP12	RC1	1	I	ST	PORTC<1> 数据输入。
		0	O	DIG	LATC<1> 数据输出。
	T1OSI	x	I	ANA	Timer1 振荡器输入；当使能 Timer1 振荡器时被使能。禁止数字 I/O。
	UOE	0	O	DIG	外部 USB 收发器 NOE 输出。
	RP12	1	I	ST	可重映射外设引脚 12 输入。
		0	O	DIG	可重映射外设引脚 12 输出。
RC2/AN11/ CTPLS/RP13	RC2	1	I	ST	PORTC<2> 数据输入。
		0	O	DIG	PORTC<2> 数据输出。
	AN11	1	I	ANA	A/D 输入通道 11。
	CTPLS	0	O	DIG	CTMU 脉冲发生器输出。
	RP13	1	I	ST	可重映射外设引脚 13 输入。
		0	O	DIG	可重映射外设引脚 13 输出。
RC4/D-/VM	RC4	x	I	TTL	PORTC<4> 数据输入。
	D-	x	I	XCVR	USB 总线负信号线输入。
		x	O	XCVR	USB 总线负信号线输出。
	VM	1	I	TTL	外部 USB 收发器 VP 输入。
RC5/D+/VP	RC5	x	I	TTL	PORTC<5> 数据输入。
	D+	x	I	XCVR	USB 总线正信号线输入。
		x	O	XCVR	USB 总线正信号线输出。
	VP	1	I	TTL	外部 USB 收发器 VP 输入。

图注： DIG = 数字电平输出； TTL = TTL 输入缓冲器； ST = 施密特触发器输入缓冲器； ANA = 模拟电平输入 / 输出； I²C/SMB = I²C/SMBus 输入缓冲器； x = 无关位（TRIS 位不影响端口方向或在此可忽略）

注 1: 该功能仅在 44 引脚器件上可用。

表 10-7: PORTC I/O 汇总 (续)

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RC6/PMA5/ TX1/CK1/RP17	RC6	1	I	ST	PORTC<6> 数据输入。
		0	O	DIG	LATC<6> 数据输出。
	PMA5 ⁽¹⁾	0	O	DIG	并行主端口地址。
	TX1	0	O	DIG	异步串行发送数据输出 (EUSART 模块)；优先于端口数据。用户必须将其配置为输出。
		1	I	ST	同步串行时钟输入 (EUSART 模块)。
	CK1	0	O	DIG	同步串行时钟输出 (EUSART 模块)；优先于端口数据。
		1	I	ST	可重映射外设引脚 17 输入。
	RP17	0	O	DIG	可重映射外设引脚 17 输出。
RC7/PMA4/ RX1/DT1/ SDO1/RP18	RC7	1	I	ST	PORTC<7> 数据输入。
		0	O	DIG	LATC<7> 数据输出。
	PMA4 ⁽¹⁾	0	O	DIG	并行主端口地址。
	RX1	1	I	ST	异步串行接收数据输入 (EUSART 模块)。
	DT1	1	I	ST	同步串行数据输入 (EUSART 模块)。用户必须将其配置为输入。
		0	O	DIG	同步串行数据输出 (EUSART 模块)；优先于端口数据。
	SDO1	0	O	DIG	SPI 数据输出 (MSSP1 模块)。
	RP18	1	I	ST	可重映射外设引脚 18 输入。
		0	O	DIG	可重映射外设引脚 18 输出。

图注: DIG = 数字电平输出; TTL = TTL 输入缓冲器; ST = 施密特触发器输入缓冲器; ANA = 模拟电平输入 / 输出; I²C/SMB = I²C/SMBus 输入缓冲器; x = 无关位 (TRIS 位不影响端口方向或在此可忽略)

注 1: 该功能仅在 44 引脚器件上可用。

表 10-8: 与 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTC	RC7	RC6	RC5	RC4	—	RC2	RC1	RC0	92
LATC	LATC7	LATC6	—	—	—	LATC2	LATC1	LATC0	92
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	92

PIC18F46J50 系列

10.5 PORTD、TRISD 和 LATD 寄存器

注： PORTD 仅在 44 引脚器件上可用。

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

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

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

注： POR 时，这些引脚被配置为数字输入。

例 10-5: 初始化 PORTD

```
CLRF    LATD        ;Initialize output data
                        ;levels for output pins
MOVLW   0x7F        ;Example value used to
                        ;initialize data direction
MOVWF   TRISD        ;RD0-RD6 as inputs
                        ;RD7 as output
```

PORTD 的每个引脚都具有内部弱上拉功能。通过一个控制位即可接通所有上拉。这可以通过清零 RDPUL 位（PORTE<7>）来实现。当端口引脚被配置为输出时，其弱上拉功能会自动关闭。POR 时禁止上拉。集成的弱上拉由半导体结构构成，与分立电阻相类似，但又稍有不同。在空载 I/O 引脚上，弱上拉用于提供逻辑高电平指示，但不需要自始至终将引脚上拉至 VDD 电平。

请注意，上拉可用于任意一组功能，类似于 PORTB 上的上拉功能。

表 10-9: PORTD I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RD0/PMD0/ SCL2	RD0	1	I	ST	PORTD<0> 数据输入。
		0	O	DIG	LATD<0> 数据输出。
	PMD0	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	SCL2	1	I	I ² C/ SMB	I ² C™ 时钟输入（MSSP2 模块）；输入类型取决于模块设置。
		0	O	I ² C	I ² C 时钟输出（MSSP2 模块）；优先于端口数据。
RD1/PMD1/ SDA2	RD1	1	I	ST	PORTD<1> 数据输入。
		0	O	DIG	LATD<1> 数据输出。
	PMD1	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	SDA2	1	I	I ² C/ SMB	I ² C 数据输入（MSSP2 模块）；输入类型取决于模块设置。
		0	O	I ² C	I ² C 数据输出（MSSP2 模块）；优先于端口数据。
RD2/PMD2/ RP19	RD2	1	I	ST	PORTD<2> 数据输入。
		0	O	DIG	LATD<2> 数据输出。
	PMD2	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	RP19	1	I	ST	可重映射外设引脚 19 输入。
		0	O	DIG	可重映射外设引脚 19 输出。

图注： DIG = 数字电平输出； TTL = TTL 输入缓冲器； ST = 施密特触发器输入缓冲器； ANA = 模拟电平输入 / 输出； I²C/SMB = I²C/SMBus 输入缓冲器； x = 无关位（TRIS 位不影响端口方向或在此可忽略）

表 10-9: PORTD I/O 汇总 (续)

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RD3/PMD3/ RP20	RD3	1	I	DIG	PORTD<3> 数据输入。
		0	O	DIG	LATD<3> 数据输出。
	PMD3	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	RP20	1	I	ST	可重映射外设引脚 20 输入。
		0	O	DIG	可重映射外设引脚 20 输出。
RD4/PMD4/ RP21	RD4	1	I	ST	PORTD<4> 数据输入。
		0	O	DIG	LATD<4> 数据输出。
	PMD4	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	RP21	1	I	ST	可重映射外设引脚 21 输入。
		0	O	DIG	可重映射外设引脚 21 输出。
RD5/PMD5/ RP22	RD5	1	I	ST	PORTD<5> 数据输入。
		0	O	DIG	LATD<5> 数据输出。
	PMD5	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	RP22	1	I	ST	可重映射外设引脚 22 输入。
		0	O	DIG	可重映射外设引脚 22 输出。
RD6/PMD6/ RP23	RD6	1	I	ST	PORTD<6> 数据输入。
		0	O	DIG	LATD<6> 数据输出。
	PMD6	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	RP23	1	I	ST	可重映射外设引脚 23 输入。
		0	O	DIG	可重映射外设引脚 23 输出。
RD7/PMD7/ RP24	RD7	1	I	ST	PORTD<7> 数据输入。
		0	O	DIG	LATD<7> 数据输出。
	PMD7	1	I	ST/TTL	并行主端口数据输入。
		0	O	DIG	并行主端口数据输出。
	RP24	1	I	ST	可重映射外设引脚 24 输入。
		0	O	DIG	可重映射外设引脚 24 输出。

图注: DIG = 数字电平输出; TTL = TTL 输入缓冲器; ST = 施密特触发器输入缓冲器; ANA = 模拟电平输入 / 输出; I²C/SMB = I²C/SMBus 输入缓冲器; x = 无关位 (TRIS 位不影响端口方向或在此可忽略)

表 10-10: 与 PORTD 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTD ⁽¹⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	92
LATD ⁽¹⁾	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	92
TRISD ⁽¹⁾	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	92

图注: — = 未实现, 读为 0。PORTD 不使用阴影单元。

注 1: 这些寄存器在 28 引脚器件上不可用。

PIC18F46J50 系列

10.6 PORTE、TRISE 和 LATE 寄存器

注： PORTE 仅在 44 引脚器件上可用。

根据选定的特定 PIC18F46J50 系列器件，PORTE 可通过两种不同的方式实现。

对于 44 引脚器件，PORTE 是一个 3 位宽的端口。3 个引脚（RE0/AN5/PMRD、RE1/AN6/PMWR 和 RE2/AN7/PMCS）可被单独配置为输入或输出。这些引脚配有施密特触发器输入缓冲器。当被选择作为模拟输入时，这些引脚将读为 0。

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

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

注： POR 时，RE<2:0> 被配置为模拟输入。

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

例 10-6: 初始化 PORTE

```
CLRF    LATE           ;Initialize LATE output
                        ;latch values
MOVLB   0x0F           ;ANCON registers not
                        ;in access bank
BSF     ANCON0,PCFG5    ;RE0/AN5 as digital
BSF     ANCON0,PCFG6    ;RE1/AN6 as digital
MOVLW   0x03           ;Example value used to
                        ;initialize data direction
MOVWF   TRISE           ;RE0, RE1 as inputs
                        ;RE2 as output
```

PORTE 的每个引脚都具有内部弱上拉功能。通过一个控制位即可接通所有上拉。这可以通过清零 REPU 位（PORTE<6>）来实现。当端口引脚被配置为输出时，其弱上拉功能会自动关闭。POR 时禁止上拉。集成的弱上拉由半导体结构构成，与分立电阻相类似，但又稍有不同。在空载 I/O 引脚上，弱上拉用于提供逻辑高电平指示，但不需要自始至终将引脚上拉至 VDD 电平。

请注意，上拉可用于任意一组功能，类似于 PORTB 上的上拉功能。

寄存器 10-5: PORTE 寄存器

R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
RDPU	REPU	—	—	—	RE2	RE1	RE0
bit 7							bit 0

图注：

R = 可读位	W = 可写位	U = 未实现位，读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

- bit 7 **RDPU:** PORTD 上拉使能位
1 = 根据各 TRIS 值使能 PORTD 上拉
0 = 禁止所有 PORTD 上拉
- bit 6 **REPU:** PORTE 上拉使能位
1 = 根据各 TRIS 值使能 PORTE 上拉
0 = 禁止所有 PORTE 上拉
- bit 5-3 **未实现:** 读为 0
- bit 2-0 **RE<2:0>:** PORTE 数据输入位

表 10-11: PORTE I/O 汇总

引脚	功能	TRIS 设置	I/O	I/O 类型	说明
RE0/AN5/ PMRD	RE0	1	I	ST	PORTE<0> 数据输入；当使能模拟输入时被禁止。
		0	O	DIG	LATE<0> 数据输出；不受模拟输入影响。
	AN5	1	I	ANA	A/D 输入通道 5；POR 时的默认输入配置。
	PMRD	1	I	ST/TTL	并行主端口 io_rd_in。
		0	O	DIG	并行主端口读选通。
RE1/AN6/ PMWR	RE1	1	I	ST	PORTE<1> 数据输入；当使能模拟输入时被禁止。
		0	O	DIG	LATE<1> 数据输出；不受模拟输入影响。
	AN6	1	I	ANA	A/D 输入通道 6；POR 时的默认输入配置。
	PMWR	1	I	ST/TTL	并行主端口 io_wr_in。
		0	O	DIG	并行主端口写选通。
RE2/AN7/ PMCS	RE2	1	I	ST	PORTE<2> 数据输入；当使能模拟输入时被禁止。
		0	O	DIG	LATE<2> 数据输出；不受模拟输入影响。
	AN7	1	I	ANA	A/D 输入通道 7；POR 时的默认输入配置。
	PMCS	0	O	DIG	并行主端口字节使能。

图注： DIG = 数字电平输出； TTL = TTL 输入缓冲器； ST = 施密特触发器输入缓冲器； ANA = 模拟电平；
I = 输入； O = 输出； P = 电源

表 10-12: 与 PORTE 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTE ⁽¹⁾	RDPU	REPU	—	—	—	RE2	RE1	RE0	92
LATE ⁽¹⁾	—	—	—	—	—	LATE2	LATE1	LATE0	92
TRISE ⁽¹⁾	—	—	—	—	—	TRISE2	TRISE1	TRISE0	92
ANCON0	PCFG7 ⁽²⁾	PCFG6 ⁽²⁾	PCFG5 ⁽²⁾	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	94

图注： — = 未实现，读为 0。PORTE 不使用阴影单元。

- 注 1: 这些寄存器在 28 引脚器件上不可用。
2: 这些位仅在 44 引脚器件上可用。

10.7 外设引脚选择 (PPS)

通用器件的主要挑战是提供尽可能多的外设功能部件，同时将其与 I/O 引脚功能的冲突降到最小。在类似于 PIC18F46J50 系列的低引脚数器件上，这一挑战更为严峻。在需要多个外设复用一個引脚的应用中，要在应用程序代码中进行变通比较困难，换句话说彻底重新设计可能是惟一的选择。

外设引脚选择 (PPS) 功能提供了这些选择的替代方法，使得用户可以在较宽的 I/O 引脚范围内选择和配置外设功能部件。通过增加特定器件上可用的引脚配置选项，用户可以让单片机更好地适合他们的整个应用，而不是通过修改应用来适应器件。

PPS 功能对固定的一部分数字 I/O 引脚进行操作。用户可以将许多数字外设的输入和 / 或输出独立地映射到这些 I/O 引脚中的任何一个。PPS 通过软件来执行，通常不需要对器件进行再编程。一旦建立外设引脚选择，就同时包含了硬件保护，以防止对外设映射的意外或错误更改。

10.7.1 可用的引脚

PPS 功能可在最多 22 个引脚的范围内使用。可用引脚的数目取决于特定器件及其引脚数。支持 PPS 功能的引脚在它们的引脚全称中包含名称“RPn”，其中“RP”表示可重映射的外设，“n”是可重映射的引脚编号。请参见表 1-1 了解每种封装中提供的引脚配置选项。

10.7.2 可用的外设

PPS 管理的外设都是仅数字外设。这些外设包括一般串行通信 (UART 和 SPI)、通用定时器时钟输入、与定时器相关的外设 (输入捕捉和输出比较) 以及外部中断输入。还包括比较器模块的输出，因为这些是离散的数字信号。

PPS 模块不适用于 I²C、电平变化通知输入、RTCC 闹钟输出或带模拟输入的外设。此外，MSSP1 和 EUSART1 模块不通过 PPS 模块进行映射。

引脚选择和非引脚选择外设之间的主要差异在于引脚选择外设与默认的 I/O 引脚无关。必须始终在使用外设前将其分配给特定的 I/O 引脚。相反，非引脚选择外设始终在默认引脚上可用，假设该外设工作且与其他外设没有冲突。

10.7.2.1 外设引脚选择功能优先级

当引脚可选择外设对于指定 I/O 引脚有效时，它的优先级高于所有其他数字 I/O 和与该引脚相关的数字通信外设。优先级与被映射外设的类型无关。引脚选择外设的优先级永远不会高于与该引脚相关的任何模拟功能。

10.7.3 控制外设引脚选择

PPS 功能由两组特殊功能寄存器 (SFR) 控制：一组映射外设输入，另一组映射外设输出。因为它们是分别控制的，所以可以不受限制地将特定外设的输入和输出 (如果外设同时具有输入和输出) 配置在任何可选择的功能引脚上。

外设与外设可选择引脚之间的关系用两种不同的方式进行处理，取决于被映射的是输入还是输出。

10.7.3.1 输入映射

PPS 选项的输入在外设基础上进行映射；即，与外设相关的控制寄存器指示要被映射的引脚。RPINRx 寄存器用于配置外设输入映射（见寄存器 10-7 至寄存器 10-21）。每个寄存器包含一个 5 位位域，该位域与引脚可选择外

设之一相关。用适当的 5 位值编程给定外设的位域，会将具有该值的 RPn 引脚映射到该外设。对于任何给定的器件，任何位域的值的有效范围与器件所支持的外设引脚选择的数目相对应。

表 10-13: 可选择的输入源（将输入映射到功能）⁽¹⁾

输入名称	功能名称	寄存器	配置位
外部中断 1	INT1	RPINR1	INTR1R<4:0>
外部中断 2	INT2	RPINR2	INTR2R<4:0>
外部中断 3	INT3	RPINR3	INTR3R<4:0>
Timer0 外部时钟输入	T0CKI	RPINR4	T0CKR<4:0>
Timer3 外部时钟输入	T3CKI	RPINR6	T3CKR<4:0>
输入捕捉 1	CCP1	RPINR7	IC1R<4:0>
输入捕捉 2	CCP2	RPINR8	IC2R<4:0>
Timer1 门控输入	T1G	RPINR12	T1GR<4:0>
Timer3 门控输入	T3G	RPINR13	T3GR<4:0>
EUSART2 异步接收 / 同步接收	RX2/DT2	RPINR16	RX2DT2R<4:0>
EUSART2 异步时钟输入	CK2	RPINR17	CK2R<4:0>
SPI2 数据输入	SDI2	RPINR21	SDI2R<4:0>
SPI2 时钟输入	SCK2IN	RPINR22	SCK2R<4:0>
SPI2 从选择输入	SS2IN	RPINR23	SS2R<4:0>
PWM 故障输入	FLT0	RPINR24	OCFAR<4:0>

注 1: 除非另外声明，否则所有输入都使用施密特触发器输入缓冲器。

PIC18F46J50 系列

10.7.3.2 输出映射

与输入相比，PPS 选项的输出在引脚基础上进行映射。在这种情况下，与特定引脚相关的控制寄存器指示要被映射的外设输出。RPORx 寄存器用于控制输出映射。位域的值与外设之一相对应，并且该外设的输出被映射到引脚（见表 10-14）。

输出映射的外设列表也包含 00000 的空值，这是由于映射技术造成的。这允许任何给定的引脚保持与任何引脚可选择外设的输出之间的未连接状态。

表 10-14: 可选择的输出源（将功能映射到输出）

功能	输出功能编号 ⁽¹⁾	输出名称
NULL	0	NULL ⁽²⁾
C1OUT	1	比较器 1 的输出
C2OUT	2	比较器 2 的输出
TX2/CK2	5	EUSART2 异步发送 / 异步时钟输出
DT2	6	EUSART2 同步发送
SDO2	9	SPI2 数据输出
SCK2	10	SPI2 时钟输出
SSDMA	12	SPI DMA 从选择
ULPOUT	13	超低功耗唤醒事件
CCP1/P1A	14	ECCP1 比较或 PWM 输出，通道 A
P1B	15	ECCP1 增强型 PWM 输出，通道 B
P1C	16	ECCP1 增强型 PWM 输出，通道 C
P1D	17	ECCP1 增强型 PWM 输出，通道 D
CCP2/P2A	18	ECCP2 比较或 PWM 输出
P2B	19	ECCP2 增强型 PWM 输出，通道 B
P2C	20	ECCP2 增强型 PWM 输出，通道 C
P2D	21	ECCP2 增强型 PWM 输出，通道 D

注 1: 分配给 RP<4:0> 引脚的值对应于外设输出功能编号。
注 2: NULL 功能在器件复位时被分配给所有 RPn 输出，并禁止 RPn 输出功能。

10.7.3.3 映射限制

PPS 的控制机制相当灵活。除了防止两个物理引脚配置为同一功能输入或两个功能输出配置为同一引脚导致的信号争用的系统模块，没有硬件强制的锁定。这种灵活性达到以下程度：允许一个输入驱动多个外设，或一个功能输出驱动多个输出引脚。

10.7.4 控制配置更改

由于可以在运行时更改外设的重映射，因此需要对外设重映射设置一些限制条件以防意外更改配置。PIC18F 器件具有 3 个功能可防止对外设映射的更改：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位重映射锁定

10.7.4.1 控制寄存器锁定

在正常工作时，不允许写入 RPINRx 和 RPORx 寄存器。尝试写入操作看似正常执行，但实际上寄存器的内容保持不变。要更改这些寄存器，必须用硬件进行解锁。寄存器锁定由 IOLOCK 位（PPSCON<0>）控制。将 IOLOCK 置 1 可防止对控制寄存器的写操作；将 IOLOCK 清零则允许写操作。

要置 1 或清零 IOLOCK，必须执行特定的命令序列：

1. 将 55h 写入 EECON2<7:0>。
2. 将 AAh 写入 EECON2<7:0>。
3. 通过一次操作将 IOLOCK 清零（或置 1）。

IOLOCK 会保持一种状态直到被更改。这允许对所有的 PPS 寄存器这样进行配置：在对所有控制寄存器进行更新前先写执行一个解锁序列，然后用第二个锁定序列锁定。

10.7.4.2 连续状态监视

除了防止直接写操作，RPINRx 和 RPORx 寄存器的内容一直由影子寄存器通过硬件进行监视。如果任何寄存器发生了意外更改（例如 ESD 或其他外部事件引起的干扰），将会触发配置不匹配复位。

10.7.4.3 配置位引脚选择锁定

为了进一步确保安全，可以将器件配置为防止对 RPINRx 和 RPORx 寄存器进行多于一次写会话。IOL1WAY（CONFIG3H<0>）配置位会阻止 IOLOCK 位在置 1 后被清零。如果 IOLOCK 保持置 1，则不会执行寄存器解锁过程，且不能写入 PPS 控制寄存器。清零该位并重新使能外设映射的惟一方法是执行器件复位。

在默认（未编程）状态下，IOL1WAY 被置 1，将用户限制为只能进行一次写会话。对 IOL1WAY 编程可允许用户（通过对解锁序列的正确使用）对 PPS 寄存器不受限制的访问。

10.7.5 外设引脚选择的注意事项

在设计中使用控制外设引脚选择功能有一些可能被忽略的注意事项。对于几个只能作为可重映射外设的常见外设尤其如此。

主要的注意事项是在器件的默认（复位）状态下，PPS 在默认引脚上不可用。由于所有 RPINRx 寄存器复位为 11111，所有 RPORx 寄存器复位为 00000，因此所有 PPS 输入连接到 RP31，而所有 PPS 输出处于未连接状态。

注：	在尝试将 PPS 输入连接到 RP31 时，要将寄存器复位为此值，RP31 不一定要存在于器件上。

这种情况要求用户在执行任何其他应用程序代码前，必须用适当的外设配置初始化器件。由于 IOLOCK 位复位到解锁状态，因此在器件复位结束后不必执行解锁序列。

然而，基于应用安全考虑，在写入控制寄存器后最好将 IOLOCK 置 1 并锁定配置。

解锁序列是时序敏感的。因此，建议将解锁序列作为汇编语言程序执行，并临时禁止中断。如果应用程序是用 C 语言或其他高级语言编写的，则解锁序列应通过写行内汇编代码来执行。

PIC18F46J50 系列

选择配置需要查看所有 PPS 及其引脚分配，尤其是那些不会在应用中使用的外设。在所有情况下，必须完全禁止未用的引脚可选择外设。未用的外设应将它们的输入分配给未用的 RPN 引脚功能。带有未用 RPN 功能的 I/O 引脚应被配置为 NULL 外设输出。

外设到特定引脚的分配不会自动执行引脚的 I/O 电路的任何其他配置。理论上，这意味着将引脚可选择输出加到引脚，当驱动输出时，引脚可能会意外驱动现有的外设输入。用户必须熟悉共用同一个可重映射引脚的其他固定外设的行为，了解何时使能或禁止它们。为安全起见，共用同一个引脚的固定数字外设在不使用时应被禁止。

根据这些概念，配置特定外设的可重映射引脚不会自动开启该外设功能。必须将外设特别配置为工作并使能，好像是连接到固定引脚一样。这部分在应用程序代码中的位置（紧跟器件复位和外设配置，或在主应用程序内）取决于外设及其在应用中的使用。

最后的注意事项是，PPS 功能既不会改写模拟输入，也不会将带模拟功能的引脚重新配置为数字 I/O。如果器件复位时引脚被配置为模拟输入，则使用 PPS 时必须明确将其重新配置为数字 I/O。

例 10-7 给出了使用 EUSART2 进行带流控制的双向通信的配置。使用了以下输入和输出功能：

- 输入功能 RX2
- 输出功能 TX2

例 10-7: 配置 EUSART2 输入和输出功能

```
*****
;
; Unlock Registers
*****
; PPS registers are in
; BANK 14

MOVLB 0x0E
BCF   INTCON, GIE   ; Disable interrupts
                        ; for unlock sequence

MOVLW 0x55
MOVWF EECON2, 0
MOVLW 0xAA
MOVWF EECON2, 0

                        ; Turn off PPS Write Protect
BCF   PPSCON, IOLOCK, BANKED

*****
; Configure Input Functions
; (See Table 9-13)
*****
; Assign RX2 To Pin RP0
*****
MOVLW 0x00
MOVWF RPINR16, BANKED

*****
; Configure Output Functions
; (See Table 9-14)
*****
; Assign TX2 To Pin RP1
*****
MOVLW 0x05
MOVWF RPOR1, BANKED

*****
; Lock Registers
*****
MOVLW 0x55
MOVWF EECON2, 0
MOVLW 0xAA
MOVWF EECON2, 0

; Write Protect PPS (if desired)
BSF   PPSCON, IOLOCK, BANKED
```

注： 如果配置位 IOL1WAY = 1，则 IOLOCK 位置 1 后，该配置位就不能清零，以防止之后 RP 寄存器发生改变。在任何器件复位时，IOLOCK 位会清零，恢复为 0。

10.7.6 外设引脚选择寄存器

PIC18F46J50 系列器件共实现了 37 个寄存器用于 44 引脚器件的可重映射外设配置。28 引脚器件有 31 个寄存器用于可重映射外设配置。

注： 仅当 IOLOCK (PPSCON<0>) = 0 时，才能更改输入寄存器和输出寄存器的值。具体的命令序列，请参见例 10-7。

寄存器 10-6: PPSCON: 外设引脚选择输入寄存器 0 (位于普通存储区, 地址 EFFh) (1)

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

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

bit 7-1 **未实现:** 读为 0
bit 0 **IOLOCK:** I/O 锁定使能位
 1 = I/O 锁定工作, RPORx 和 RPINRx 寄存器被写保护
 0 = I/O 锁定不工作, 可以更改引脚配置

注 1: 仅当 IOLOCK (PPSCON<0>) = 0 时, 才能更改寄存器的值。

PIC18F46J50 系列

寄存器 10-7: **RPINR1: 外设引脚选择输入寄存器 1**（位于普通存储区，地址 EE7h）

U-0	U-0	U-0	$\overline{R/\overline{W}}-1$	$\overline{R/\overline{W}}-1$	$\overline{R/\overline{W}}-1$	$\overline{R/\overline{W}}-1$	$\overline{R/\overline{W}}-1$
—	—	—	INTR1R4	INTR1R3	INTR1R2	INTR1R1	INTR1R0
bit 7			bit 0				

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **INTR1R<4:0>**: 将外部中断 1（INT1）分配给对应 RPn 引脚的位

寄存器 10-8: **RPINR2: 外设引脚选择输入寄存器 2**（位于普通存储区，地址 EE8h）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INTR2R4	INTR2R3	INTR2R2	INTR2R1	INTR2R0
bit 7			bit 0				

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **INTR2R<4:0>**: 将外部中断 2（INT2）分配给对应 RPn 引脚的位

寄存器 10-9: **RPINR3: 外设引脚选择输入寄存器 3**（位于普通存储区，地址 EE9h）

U-0	U-0	U-0	$\overline{R/W-1}$	$\overline{R/W-1}$	$\overline{R/W-1}$	$\overline{R/W-1}$	$\overline{R/W-1}$
—	—	—	INTR3R4	INTR3R3	INTR3R2	INTR3R1	INTR3R0
bit 7			bit 0				

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **INTR3R<4:0>**: 将外部中断 3（INT3）分配给对应 RPn 引脚的位

寄存器 10-10: RPINR4: 外设引脚选择输入寄存器 4（位于普通存储区，地址 EEAh）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T0CKR4	T0CKR3	T0CKR2	T0CKR1	T0CKR0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **T0CKR<4:0>**: 将 Timer0 外部时钟输入（T0CKI）分配给对应 RPn 引脚的位

寄存器 10-11: RPINR6: 外设引脚选择输入寄存器 6（位于普通存储区，地址 EECh）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **T3CKR<4:0>**: 将 Timer3 外部时钟输入（T3CKI）分配给对应 RPn 引脚的位

寄存器 10-12: RPINR7: 外设引脚选择输入寄存器 7（位于普通存储区，地址 EEDh）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **IC1R<4:0>**: 将输入捕捉 1（ECCP1）分配给对应 RPn 引脚的位

PIC18F46J50 系列

寄存器 10-13: **RPINR8: 外设引脚选择输入寄存器 8** (位于普通存储区, 地址 **EEEh**)

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 7							bit 0

图注: $\overline{\text{R/W}}$ = 可读 / 可写位 (如果 $\text{IOLOCK} = 0$)
R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现: 读为 0
bit 4-0 **IC2R<4:0>**: 将输入捕捉 2 (ECCP2) 分配给对应 RPn 引脚的位

寄存器 10-14: **RPINR12: 外设引脚选择输入寄存器 12** (位于普通存储区, 地址 **EF2h**)

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T1GR4	T1GR3	T1GR2	T1GR1	T1GR0
bit 7							bit 0

图注: $\overline{\text{R/W}}$ = 可读 / 可写位 (如果 $\text{IOLOCK} = 0$)
R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现: 读为 0
bit 4-0 **T1GR<4:0>**: 将 Timer1 门控输入 (T1G) 分配给对应 RPn 引脚的位

寄存器 10-15: **RPINR13: 外设引脚选择输入寄存器 13** (位于普通存储区, 地址 **EF3h**)

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T3GR4	T3GR3	T3GR2	T3GR1	T3GR0
bit 7							bit 0

图注: $\overline{\text{R/W}}$ = 可读 / 可写位 (如果 $\text{IOLOCK} = 0$)
R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现: 读为 0
bit 4-0 **T3GR<4:0>**: 将 Timer3 门控输入 (T3G) 分配给对应 RPn 引脚的位

寄存器 10-16: RPINR16: 外设引脚选择输入寄存器 16（位于普通存储区，地址 EF6h）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	RX2DT2R4	RX2DT2R3	RX2DT2R2	RX2DT2R1	RX2DT2R0
bit 7							bit 0

图注: $\overline{R/W}$ = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RX2DT2R<4:0>**: 将 EUSART2 同步 / 异步接收（RX2/DT2）分配给对应 RPn 引脚的位

寄存器 10-17: RPINR17: 外设引脚选择输入寄存器 17（位于普通存储区，地址 EF7h）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	CK2R4	CK2R3	CK2R2	CK2R1	CK2R0
bit 7							bit 0

图注: $\overline{R/W}$ = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **CK2R<4:0>**: 将 EUSART2 时钟输入（CK2）分配给对应 RPn 引脚的位

寄存器 10-18: RPINR21: 外设引脚选择输入寄存器 21（位于普通存储区，地址 EFBh）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

图注: $\overline{R/W}$ = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **SDI2R<4:0>**: 将 SPI2 数据输入（SDI2）分配给对应 RPn 引脚的位

PIC18F46J50 系列

寄存器 10-19: **RPINR22: 外设引脚选择输入寄存器 22**（位于普通存储区，地址 EFCh）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零			x = 未知	

bit 7-5 未实现：读为 0
bit 4-0 **SCK2R<4:0>**: 将 SPI2 时钟输入（SCK2）分配给对应 RPn 引脚的位

寄存器 10-20: **RPINR23: 外设引脚选择输入寄存器 23**（位于普通存储区，地址 EFDh）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零			x = 未知	

bit 7-5 未实现：读为 0
bit 4-0 **SS2R<4:0>**: 将 SPI2 从选择输入（SS2IN）分配给对应 RPn 引脚的位

寄存器 10-21: **RPINR24: 外设引脚选择输入寄存器 24**（位于普通存储区，地址 EFEh）

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零			x = 未知	

bit 7-5 未实现：读为 0
bit 4-0 **OCFAR<4:0>**: 将 PWM 故障输入（FLT0）分配给对应 RPn 引脚的位

寄存器 10-22: RPOR0: 外设引脚选择输出寄存器 0（位于普通存储区，地址 EC6h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP0R<4:0>**: 将外设输出功能分配给 RP0 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-23: RPOR1: 外设引脚选择输出寄存器 1（位于普通存储区，地址 EC7h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP1R<4:0>**: 将外设输出功能分配给 RP1 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-24: RPOR2: 外设引脚选择输出寄存器 2（位于普通存储区，地址 EC8h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP2R<4:0>**: 将外设输出功能分配给 RP2 输出引脚的位（请参见表 10-14 了解外设功能编号）

PIC18F46J50 系列

寄存器 10-25: RPOR3: 外设引脚选择输出寄存器 3 (位于普通存储区, 地址 EC9h)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位 (如果 IOLOCK = 0)						
R = 可读位	W = 可写位		U = 未实现位, 读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零			x = 未知	

bit 7-5 未实现: 读为 0
bit 4-0 RP3R<4:0>: 将外设输出功能分配给 RP3 输出引脚的位 (请参见表 10-14 了解外设功能编号)

寄存器 10-26: RPOR4: 外设引脚选择输出寄存器 4 (位于普通存储区, 地址 ECAh)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位 (如果 IOLOCK = 0)						
R = 可读位	W = 可写位		U = 未实现位, 读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零			x = 未知	

bit 7-5 未实现: 读为 0
bit 4-0 RP4R<4:0>: 将外设输出功能分配给 RP4 输出引脚的位 (请参见表 10-14 了解外设功能编号)

寄存器 10-27: RPOR5: 外设引脚选择输出寄存器 5 (位于普通存储区, 地址 ECBh)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP5R4	RP5R3	RP5R2	RP5R1	RP5R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位 (如果 IOLOCK = 0)						
R = 可读位	W = 可写位		U = 未实现位, 读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零			x = 未知	

bit 7-5 未实现: 读为 0
bit 4-0 RP5R<4:0>: 将外设输出功能分配给 RP5 输出引脚的位 (请参见表 10-14 了解外设功能编号)

寄存器 10-28: RPOR6: 外设引脚选择输出寄存器 6 (位于普通存储区, 地址 ECCh)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
bit 7							bit 0

图注: R/W = 可读 / 可写位 (如果 IOLOCK = 0)
R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现: 读为 0

bit 4-0 **RP6R<4:0>**: 将外设输出功能分配给 RP6 输出引脚的位 (请参见表 10-14 了解外设功能编号)

寄存器 10-29: RPOR7: 外设引脚选择输出寄存器 7 (位于普通存储区, 地址 ECDh)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 7							bit 0

图注: R/W = 可读 / 可写位 (如果 IOLOCK = 0)
R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现: 读为 0

bit 4-0 **RP7R<4:0>**: 将外设输出功能分配给 RP7 输出引脚的位 (请参见表 10-14 了解外设功能编号)

寄存器 10-30: RPOR8: 外设引脚选择输出寄存器 8 (位于普通存储区, 地址 ECEh)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
bit 7							bit 0

图注: R/W = 可读 / 可写位 (如果 IOLOCK = 0)
R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现: 读为 0

bit 4-0 **RP8R<4:0>**: 将外设输出功能分配给 RP8 输出引脚的位 (请参见表 10-14 了解外设功能编号)

PIC18F46J50 系列

寄存器 10-31: **RPOR9: 外设引脚选择输出寄存器 9**（位于普通存储区，地址 **ECFh**）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **RP9R<4:0>**：将外设输出功能分配给 RP9 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-32: **RPOR10: 外设引脚选择输出寄存器 10**（位于普通存储区，地址 **ED0h**）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **RP10R<4:0>**：将外设输出功能分配给 RP10 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-33: **RPOR11: 外设引脚选择输出寄存器 11**（位于普通存储区，地址 **ED1h**）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **RP11R<4:0>**：将外设输出功能分配给 RP11 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-34: RPOR12: 外设引脚选择输出寄存器 12（位于普通存储区，地址 ED2h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP12R<4:0>**: 将外设输出功能分配给 RP12 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-35: RPOR13: 外设引脚选择输出寄存器 13（位于普通存储区，地址 ED3h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP13R<4:0>**: 将外设输出功能分配给 RP13 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-36: RPOR17: 外设引脚选择输出寄存器 17（位于普通存储区，地址 ED7h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP17R<4:0>**: 将外设输出功能分配给 RP17 输出引脚的位（请参见表 10-14 了解外设功能编号）

PIC18F46J50 系列

寄存器 10-37: **RPOR18: 外设引脚选择输出寄存器 18**（位于普通存储区，地址 ED8h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **RP18R<4:0>**: 将外设输出功能分配给 RP18 输出引脚的位（请参见表 10-14 了解外设功能编号）

寄存器 10-38: **RPOR19: 外设引脚选择输出寄存器 19**（位于普通存储区，地址 ED9h）⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **RP19R<4:0>**: 将外设输出功能分配给 RP19 输出引脚的位（请参见表 10-14 了解外设功能编号）

注 1: RP19 引脚在 28 引脚器件上不可用。

寄存器 10-39: **RPOR20: 外设引脚选择输出寄存器 20**（位于普通存储区，地址 EDAh）⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）						
R = 可读位	W = 可写位		U = 未实现位，读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零		x = 未知		

bit 7-5 未实现：读为 0
bit 4-0 **RP20R<4:0>**: 将外设输出功能分配给 RP20 输出引脚的位（请参见表 10-14 了解外设功能编号）

注 1: RP20 引脚在 28 引脚器件上不可用。

寄存器 10-40: RPOR21: 外设引脚选择输出寄存器 21（位于普通存储区，地址 EDBh）⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP21R<4:0>**：将外设输出功能分配给 RP21 输出引脚的位（请参见表 10-14 了解外设功能编号）

注 1: RP21 引脚在 28 引脚器件上不可用。

寄存器 10-41: RPOR22: 外设引脚选择输出寄存器 22（位于普通存储区，地址 EDCh）⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP22R<4:0>**：将外设输出功能分配给 RP22 输出引脚的位（请参见表 10-14 了解外设功能编号）

注 1: RP22 引脚在 28 引脚器件上不可用。

寄存器 10-42: RPOR23: 外设引脚选择输出寄存器 23（位于普通存储区，地址 EDDh）⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
bit 7							bit 0

图注: R/W = 可读 / 可写位（如果 IOLOCK = 0）
R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-5 未实现：读为 0

bit 4-0 **RP23R<4:0>**：将外设输出功能分配给 RP23 输出引脚的位（请参见表 10-14 了解外设功能编号）

注 1: RP23 引脚在 28 引脚器件上不可用。

PIC18F46J50 系列

寄存器 10-43: **RPOR24: 外设引脚选择输出寄存器 24**（位于普通存储区，地址 EDEh）⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
bit 7							bit 0

图注:	R/W = 可读 / 可写位（如果 IOLOCK = 0）		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7-5 未实现：读为 0
- bit 4-0 **RP24R<4:0>**：将外设输出功能分配给 RP24 输出引脚的位（请参见表 10-14 了解外设功能编号）
- 注 1: RP24 引脚在 28 引脚器件上不可用。

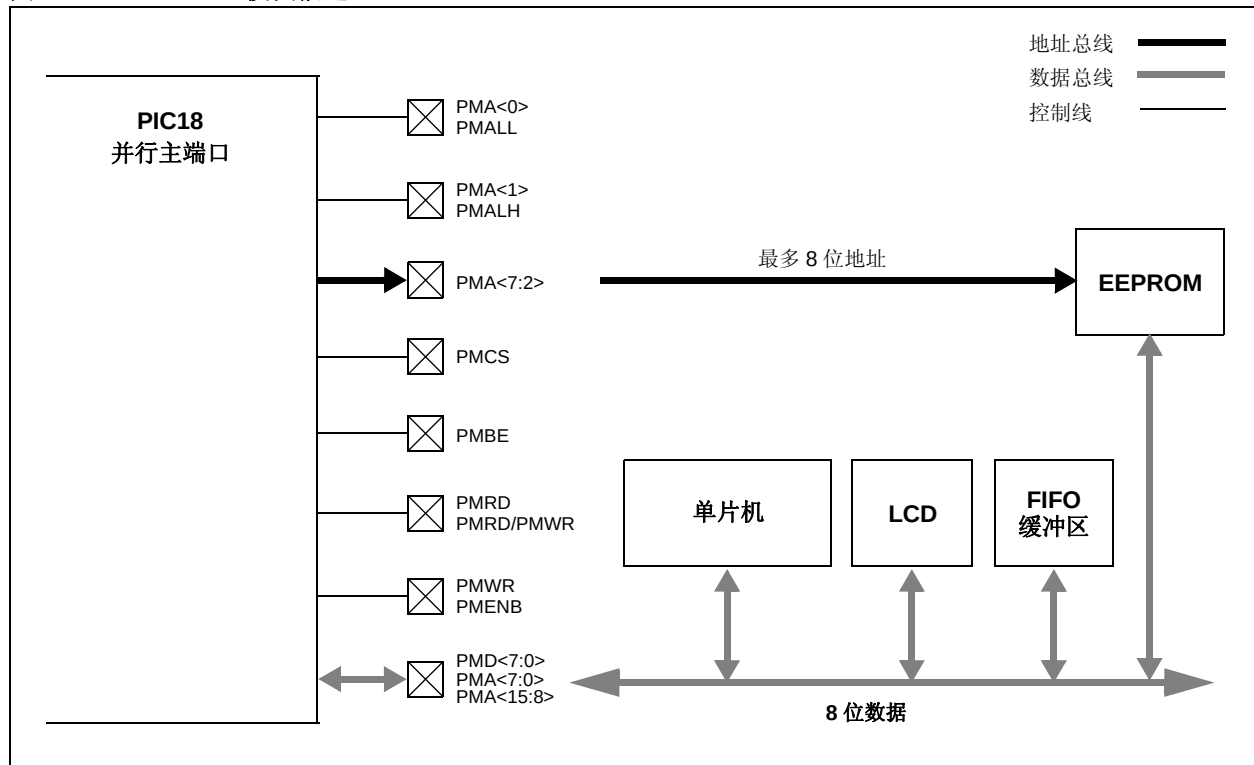
11.0 并行主端口（PMP）

并行主端口（PMP）模块是一个 8 位并行 I/O 模块，专用于与通信外设、LCD、外部存储器件及单片机等许多种并行器件进行通信。由于并行外设的接口的多样化，PMP 具有高度可配置性。PMP 模块可配置为用作 PMP 或并行从端口（PSP）。

PMP 模块的主要特性包括：

- 当采用数据 / 地址复用线时，最多可寻址 16 位
- 最多 8 根可编程地址线
- 1 根片选线
- 可编程选通选项：
 - 单独读和写选通，或；
 - 带使能选通的读 / 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
- 支持传统并行从端口
- 增强型并行从动支持：
 - 地址支持
 - 4 字节深的自动递增缓冲区
- 可编程等待状态
- 可选择的输入电平

图 11-1: PMP 模块概述



PIC18F46J50 系列

11.1 模块寄存器

PMP 模块共有 14 个特殊功能寄存器（SFR）用于进行操作，另外还有一个寄存器用于设置配置选项。其中，8 个寄存器用于进行控制，6 个寄存器用于 PMP 数据传输。

11.1.1 控制寄存器

8 个 PMP 控制寄存器包括：

- PMCONH 和 PMCONL
- PMMODEH 和 PMMODEL
- PMSTATL 和 PMSTATH
- PMEHL 和 PMEL

PMCON 寄存器（寄存器 11-1 和寄存器 11-2）用于控制基本的模块操作，包括开启和关闭模块。它们还可用于配置地址复用和控制选通配置。

PMMODE 寄存器（寄存器 11-3 和寄存器 11-4）用于配置各种主模式与从模式、数据宽度和中断产生。

PMEH 和 PMEL 寄存器（寄存器 11-5 和寄存器 11-6）用于配置硬件（I/O 引脚）级别的模块操作。

PMSTAT 寄存器（寄存器 11-5 和寄存器 11-6）用于根据工作模式提供模块输入和输出缓冲区的状态标志。

寄存器 11-1: PMCONH: 并行端口控制寄存器高字节（位于普通存储区，地址 F5Fh）⁽¹⁾

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPEN	—	—	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 = POR 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 6-5	未实现: 读为 0
bit 4-3	ADRMUX<1:0>: 地址 / 数据复用选择位 11 = 保留 10 = 地址的所有 16 位与 PMD<7:0> 引脚复用 01 = 地址的低 8 位与 PMD<7:0> 引脚复用（在该模式下只能用地址的 8 位） 00 = 地址和数据出现在独立的引脚上（在该模式下只能用地址的 8 位）
bit 2	PTBEEN: 字节使能端口使能位（16 位主模式） 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

注 1: 该寄存器仅在 44 引脚器件上可用。

寄存器 11-2: PMCONL: 并行端口控制寄存器低字节（位于普通存储区，地址 F5Eh）⁽¹⁾

R/W-0	R/W-0	R/W-0 ⁽²⁾	R/W-0	R/W-0 ⁽²⁾	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	—	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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-6 **CSF<1:0>**: 片选功能位

11 = 保留

10 = 使能片选功能，PMCS 用作片选（在主模式下）。最多只能生成 13 个地址位。

01 = 保留

00 = 禁止片选功能（在主模式下）。可以生成所有 16 个地址位。

bit 5 **ALP**: 地址锁存器极性位 ⁽²⁾

1 = 高电平有效（PMALL 和 PMALH）

0 = 低电平有效（PMALL 和 PMALH）

bit 4 **未实现**: 读为 0

bit 3 **CS1P**: 片选极性位 ⁽²⁾

1 = 高电平有效（PMCS）

0 = 低电平有效（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

对于从模式和主模式 2（PMMODEH<1:0> = 00、01 和 10）:

1 = 写选通高电平有效（PMWR）

0 = 写选通低电平有效（PMWR）

对于主模式 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

对于从模式和主模式 2（PMMODEH<1:0> = 00、01 和 10）:

1 = 读选通高电平有效（PMRD）

0 = 读选通低电平有效（PMRD）

对于主模式 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: 该寄存器仅在 44 引脚器件上可用。

2: 这些位在相应引脚用作地址线时无效。

PIC18F46J50 系列

寄存器 11-3: PMMODEH: 并行端口模式寄存器高字节 (位于普通存储区, 地址 F5Dh) (1)

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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	BUSY: 忙状态位 (仅用于主模式) 1 = 端口处于忙状态 0 = 端口未处于忙状态
bit 6-5	IRQM<1:0>: 中断请求模式位 11 = 当对读缓冲区 3 执行读操作或对写缓冲区 3 执行写操作时产生中断 (缓冲 PSP 模式) 或者当 PMA<1:0> = 11 (仅用于可寻址 PSP 模式) 时的读 / 写操作时产生中断 10 = 不产生中断, 处理器停止工作 01 = 读 / 写周期末尾产生中断 00 = 不产生中断
bit 4-3	INCM<1:0>: 递增模式位 11 = PSP 读缓冲区和写缓冲区自动递增 (仅用于传统 PSP 模式) 10 = 每个读 / 写周期 ADDR<15,13:0> 递减 1 01 = 每个读 / 写周期 ADDR<15,13:0> 递增 1 00 = 无地址递增或递减
bit 2	MODE16: 8/16 位模式位 1 = 16 位模式: 数据寄存器为 16 位, 对数据寄存器执行读或写操作调用两次 8 位传输 0 = 8 位模式: 数据寄存器为 8 位, 对数据寄存器执行读或写操作调用一次 8 位传输
bit 1-0	MODE<1:0>: 并行端口模式选择位 11 = 主模式 1 (PMCS、PMRD/PMWR、PMENB、PMBE、PMA<x:0> 和 PMD<7:0>) 10 = 主模式 2 (PMCS、PMRD、PMWR、PMBE、PMA<x:0> 和 PMD<7:0>) 01 = 增强型 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>)

注 1: 该寄存器仅在 44 引脚器件上可用。

寄存器 11-4: **PMMODEL**: 并行端口模式寄存器低字节（位于普通存储区，地址 F5Ch）⁽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
WAITB1 ⁽²⁾	WAITB0 ⁽²⁾	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1 ⁽²⁾	WAITE0 ⁽²⁾
bit 7							bit 0

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

- bit 7-6

WAITB<1:0>: 数据建立读 / 写等待状态配置位 ⁽²⁾
11 = 数据等待 4 个 T_{CY} ; 复用地址阶段等待 4 个 T_{CY}
10 = 数据等待 3 个 T_{CY} ; 复用地址阶段等待 3 个 T_{CY}
01 = 数据等待 2 个 T_{CY} ; 复用地址阶段等待 2 个 T_{CY}
00 = 数据等待 1 个 T_{CY} ; 复用地址阶段等待 1 个 T_{CY}
- bit 5-2

WAITM<3:0>: 读字节使能选通等待状态配置位
1111 = 等待额外 15 个 T_{CY}
.
.
.
0001 = 等待额外 1 个 T_{CY}
0000 = 无额外等待周期（强制操作在 1 个 T_{CY} 内执行）
- bit 1-0

WAITE<1:0>: 选通后数据保持等待状态配置位 ⁽²⁾
11 = 等待 4 个 T_{CY}
10 = 等待 3 个 T_{CY}
01 = 等待 2 个 T_{CY}
00 = 等待 1 个 T_{CY}

注 1: 该寄存器仅在 44 引脚器件上可用。

2: 只要 WAITM<3:0> = 0000，WAITBx 和 WAITEx 位就被忽略。

PIC18F46J50 系列

寄存器 11-5: PMEH: 并行端口使能寄存器高字节（位于普通存储区，地址 F57h）⁽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
—	PTEN14	—	—	—	—	—	—
bit 7							bit 0

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

- bit 7 未实现：保持为 0
- bit 6 **PTEN14**: PMCS 端口使能位
 1 = PMCS 用作片选线
 0 = PMCS 用作端口 I/O
- bit 5-0 未实现：保持为 0

注 **1**: 该寄存器仅在 44 引脚器件上可用。

寄存器 11-6: PMEL: 并行端口使能寄存器低字节（位于普通存储区，地址 F56h）⁽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
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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7-2 **PTEN<7:2>**: PMP 地址端口使能位
 1 = PMA<7:2> 作为 PMP 地址线
 0 = PMA<7:2> 作为端口 I/O
- bit 1-0 **PTEN<1:0>**: PMALH/PMALL 选通使能位
 1 = PMA<1:0> 作为 PMA<1:0> 或 PMALH 和 PMALL
 0 = PMA<1:0> 引脚作为端口 I/O

注 **1**: 该寄存器仅在 44 引脚器件上可用。

寄存器 11-7: PMSTATH: 并行端口状态寄存器高字节 (位于普通存储区, 地址 F55h) (1)

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 = POR 时的值 1 = 置 1 0 = 清零 x = 未知

- bit 7 **IBF:** 输入缓冲区满状态位
 1 = 所有可写输入缓冲寄存器均已满
 0 = 部分或所有可写输入缓冲寄存器为空
- bit 6 **IBOV:** 输入缓冲区溢出状态位
 1 = 尝试对已满的输入字节寄存器进行写操作 (必须用软件清零)
 0 = 未发生溢出
- bit 5-4 **未实现:** 读为 0
- bit 3-0 **IB3F:IB0F:** 输入缓冲区 x 状态满位
 1 = 输入缓冲区包含尚未读取的数据 (读缓冲区将清零该位)
 0 = 输入缓冲区不包含任何未读数据

注 1: 该寄存器仅在 44 引脚器件上可用。

寄存器 11-8: PMSTATL: 并行端口状态寄存器低字节 (位于普通存储区, 地址 F54h) (1)

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 = POR 时的值 1 = 置 1 0 = 清零 x = 未知

- bit 7 **OBE:** 输出缓冲区空状态位
 1 = 所有可读输出缓冲寄存器均为空
 0 = 部分或所有可读输出缓冲寄存器已满
- bit 6 **OBUF:** 输出缓冲区下溢状态位
 1 = 对空输出字节寄存器执行了读操作 (必须用软件清零)
 0 = 未发生下溢
- bit 5-4 **未实现:** 读为 0
- bit 3-0 **OB3E:OB0E:** 输出缓冲区 x 状态空位
 1 = 输出缓冲区为空 (向缓冲区写数据将清零该位)
 0 = 输出缓冲区包含尚未发送的数据

注 1: 该寄存器仅在 44 引脚器件上可用。

PIC18F46J50 系列

11.1.2 数据寄存器

PMP 模块使用 8 个寄存器来将数据传输入和传输出单片机。它们构成 4 个寄存器对，用于支持 16 位数据操作：

- PMDIN1H 和 PMDIN1L
- PMDIN2H 和 PMDIN2L
- PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L
- PMDOUT2H 和 PMDOUT2L

PMDIN1 寄存器在从模式下用于存放传入的数据，在主模式下用于输入和输出数据。PMDIN2 寄存器在某些从模式下用于缓冲输入数据。

PMADDR/PMDOUT1 寄存器实际上是一个寄存器对；名称和功能由模块的工作模式决定。在主模式下，这两个寄存器用作 PMADDRH 和 PMADDRL 寄存器，包含所有传入或外发数据的地址。在从模式下，这两个寄存器用作 PMDOUT1H 和 PMDOUT1L，用于存放外发的数据。

PMADDRH 与 PMADDRL 的区别在于 PMADDRH 具有有限的 PMP 控制功能。当模块以选择从模式配置工作时，寄存器的高 2 位可用于决定片选信号的操作。如果未使用它们，则 PMADDR 仅仅用于保存地址的高 8 位。[寄存器 11-9](#) 列出了 PMADDRH 中各个位的功能。

PMDOUT2H 和 PMDOUT2L 寄存器仅用于缓冲从模式，用作外发数据的缓冲区。

11.1.3 焊盘配置控制寄存器

除了模块级别的配置选项之外，PMP 模块还可以在 I/O 引脚级别配置电气操作。这种选项使用户可以选择与 PMP 共用的数字 I/O 引脚上的常规施密特触发器输入缓冲器，或使用与 TTL 电平兼容的缓冲器。缓冲器配置由 PADCFG1 寄存器中的 PMPTTL 位控制。

寄存器 11-9: PMADDRH: 并行端口地址寄存器高字节（仅用于主模式）（位于快速操作存储区，地址 F6Fh）⁽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
—	CS1	并行主端口地址高字节 <13:8>					
bit 7							bit 0

图注:			
R = 可读位	W = 可写位	U = 未实现位，读为 0	r = 保留
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未实现:** 保留为 0

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 00:
此位作为 ADDR<14>。

bit 5-0 **并行主端口地址:** 高字节 <13:8> 位

注 1: 在增强型从模式下，PMADDRH 作为 PMDOUT1H（输出数据缓冲寄存器之一）。

寄存器 11-10: PMADDRL: 并行端口地址寄存器低字节（仅用于主模式）（位于快速操作存储区，地址 F6Eh）⁽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
并行主端口地址低字节 <7:0>							
bit 7							bit 0

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

bit 7-0 **并行主端口地址:** 低字节 <7:0> 位

注 1: 在增强型从模式下，PMADDRL 作为 PMDOUT1L（输出数据缓冲寄存器之一）。

11.2 从端口模式

模块的主要工作模式使用 `PMODEH` 寄存器中的 `MODE<1:0>` 位进行配置。设置会影响模块是工作于主模式还是从模式，并决定控制引脚的使用。

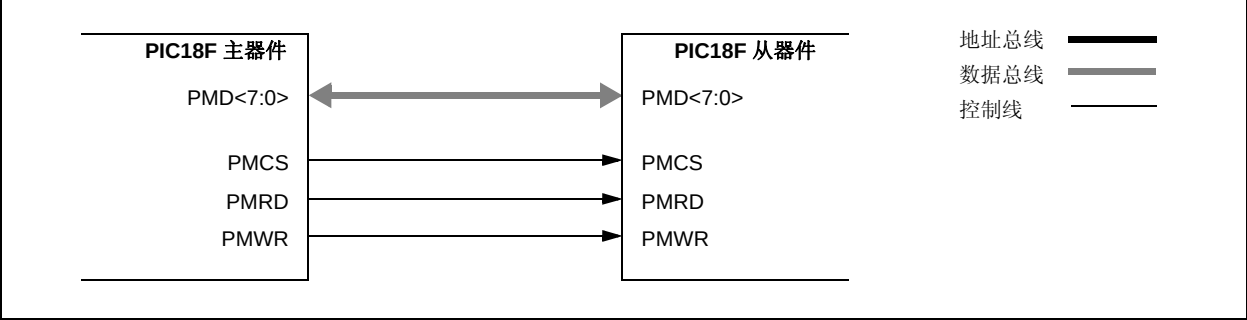
11.2.1 传统模式（PSP）

在传统模式下（`PMODEH<1:0> = 00` 且 `PMPEN = 1`），模块被配置为并行从端口（PSP），相关的使能模块引脚为模块所专用。在该模式下，例如其

他单片机或微处理器的外部器件可以使用 8 位数据总线（`PMD<7:0>`）、读（`PMRD`）、写（`PMWR`）和片选（`PMCS`）输入引脚异步地读写数据。它在总线上用作从器件，对读 / 写控制信号作出响应。

图 11-2 给出了 PSP 的连接图示。当片选有效并产生写选通（`PMCS = 1` 且 `PMWR = 1`）时，`PMD<7:0>` 中的数据被捕捉到 `PMDIN1L` 寄存器中。

图 11-2: 传统并行从端口示例



11.2.2 写从端口

当片选有效并产生写选通（PMCS = 1 且 PMWR = 1）时，PMD<7:0> 中的数据被捕捉到 PMDIN1L 低字节寄存器中。当写操作结束时，PMPIF 和 IBF 标志位会置 1。图 11-3 给出了写操作模式下的控制信号的时序。控制信号的极性是可配置的。

11.2.3 读从端口

当片选有效并产生读选通（PMCS = 1 且 PMRD = 1）时，来自 PMDOUT1L 寄存器（PMDOUT1L<7:0>）的数据被送到 PMD<7:0>。图 11-4 给出了读操作模式下的控制信号的时序。

图 11-3: 并行从端口写波形

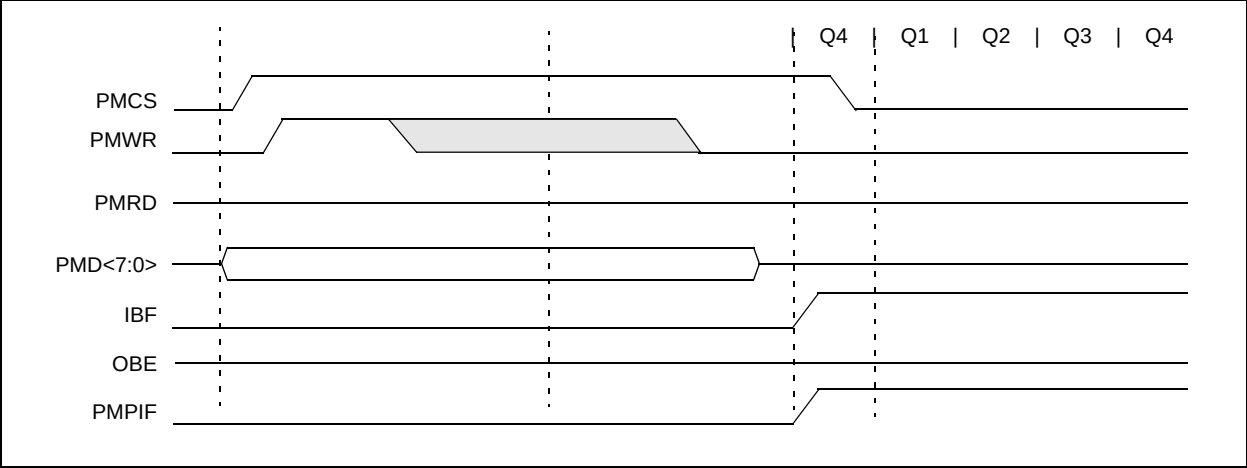
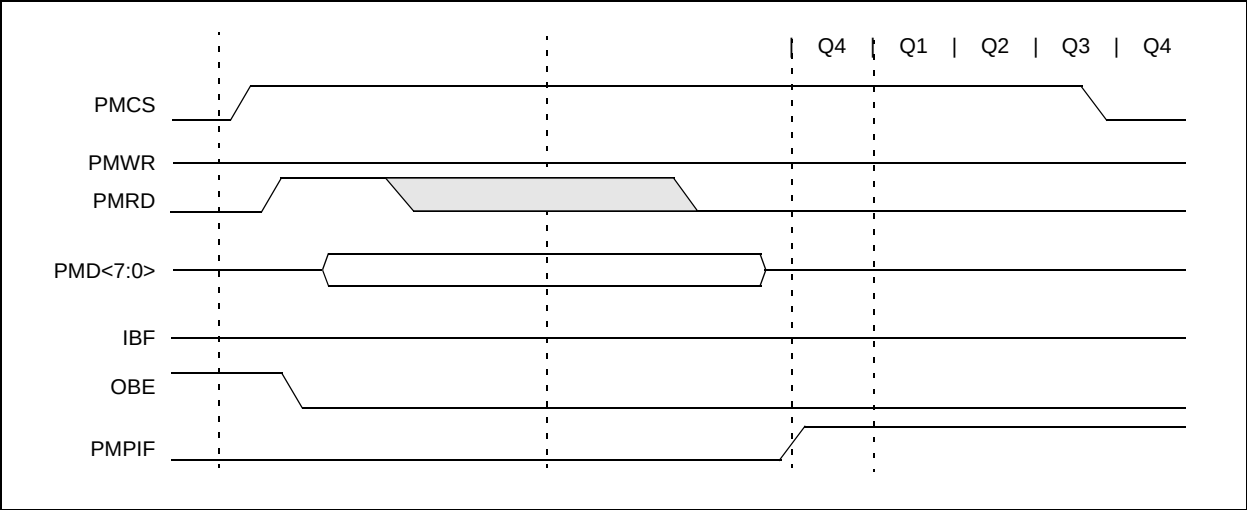


图 11-4: 并行从端口读波形



PIC18F46J50 系列

11.2.4 缓冲并行从端口模式

缓冲并行从端口模式在功能上等效于传统 PSP 模式，但有一点区别，即它实现了 4 级深的读写缓冲区。缓冲 PSP 模式通过设置 PMMODEH 寄存器中的 INCM 位进行使能。如果 INCM<1:0> 位设置为 11，PMP 模块将用作缓冲 PSP 模式。

当缓冲 PSP 模式有效时，PMDIN1L、PMDIN1H、PMDIN2L 和 PMDIN2H 寄存器成为写缓冲区，PMDOUT1L、PMDOUT1H、PMDOUT2L 和 PMDOUT2H 寄存器成为读缓冲区。缓冲区从 0 至 3 进行编号，从 PMDIN1L 低字节到 PMDIN2H 用作读缓冲区，从 PMDOUT1L 到 PMDOUT2H 用作写缓冲区。

11.2.4.1 读从端口

对于读操作，每次读选通后，字节按顺序送出，从缓冲区 0（PMDOUT1L<7:0>）开始，到缓冲区 3（PMDOUT2H<7:0>）结束。模块通过内部指针跟踪要读取哪个缓冲区。每个缓冲区在 PMSTATL 寄存器中都有一个对应的读状态位 OBxE。当缓冲区包含尚未写到总线的的数据时，该位被清零；当数据写入总线时则被置 1。如果当前所读取的缓冲区单元为空，则产生缓冲区下溢，并且缓冲区下溢标志位 OBUF 被置 1。如果所有 4 个 OBxE 状态位都被置 1，则输出缓冲区空标志（OBE）也将被置 1。

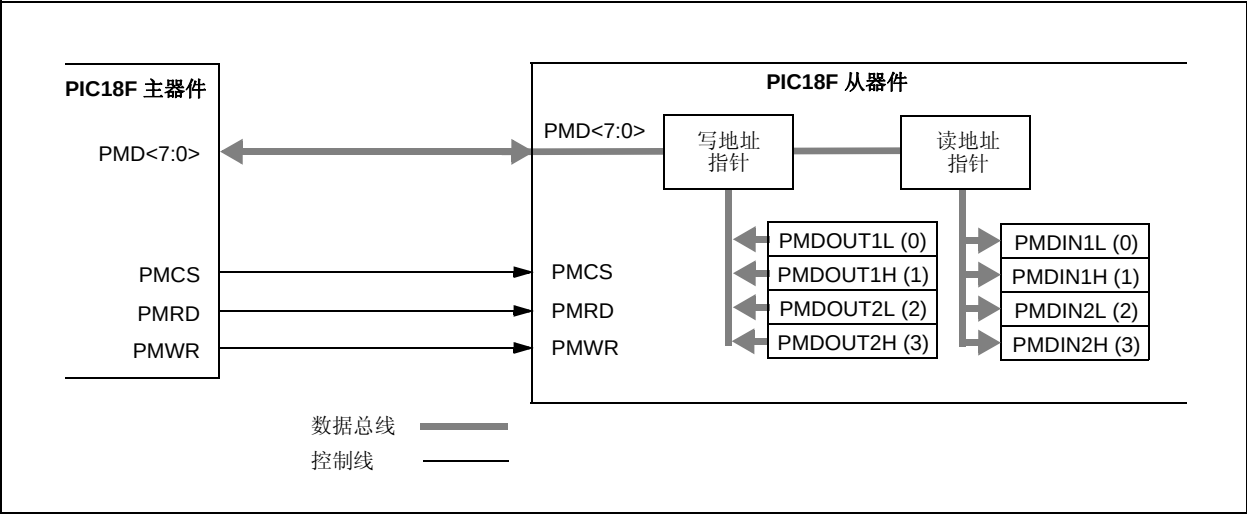
11.2.4.2 写从端口

对于写操作，数据按顺序存储，从缓冲区 0（PMDIN1L<7:0>）开始，到缓冲区 3（PMDIN2H<7:0>）结束。与读操作一样，模块内部指针保持指向下一次要写的缓冲区。

输入缓冲区在 PMSTATH 寄存器中有自己的写状态位 IBxF。当缓冲区包含未读的输入数据时，该位置 1；当数据已读取时，该位清零。标志位在写选通时置 1。如果相关的 IBxF 位置 1 时对缓冲区执行写操作，则缓冲区溢出标志位 IBOV 置 1；缓冲区中任何输入的数据将丢失。如果所有 4 个 IBxF 标志都被置 1，则输入缓冲区满标志（IBF）被置 1。

在缓冲从模式下，可以将模块配置为在每个读或写选通（IRQM<1:0> = 01）时产生中断。也可以配置为在对读缓冲区 3 执行读操作或对写缓冲区 3 执行写操作（RQM<1:0> = 11）时产生中断，这实质上是在每 4 个读或写选通时产生一次中断。当输入第 4 个字节数据产生中断时，应该读所有的输入缓冲寄存器来清零 IBxF 标志。如果不清零这些标志，则可能会导致产生溢出条件。

图 11-5: 并行主 / 从器件连接缓冲示例



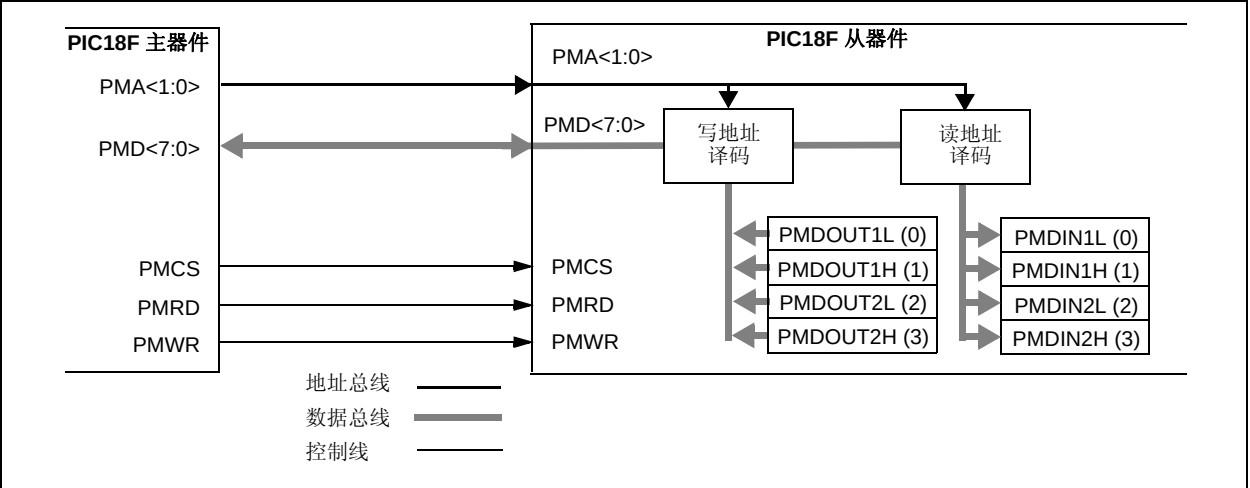
11.2.5 可寻址的并行从端口模式

在可寻址的并行从端口模式下（ $PMMODEH<1:0> = 01$ ），模块配置有两个额外的输入（ $PMA<1:0>$ ），它们是地址线 1 和 0。这使 4 字节的缓冲区空间可作为固定的读写缓冲区对进行寻址。类似于传统缓冲模式，数据从 $PMDOUT1L$ 、 $PMDOUT1H$ 、 $PMDOUT2L$ 和 $PMDOUT2H$ 输出，并读取到 $PMDIN1L$ 、 $PMDIN1H$ 、 $PMDIN2L$ 和 $PMDIN2H$ 。[表 11-1](#) 列出了传入地址与输入和输出寄存器的缓冲区寻址关系。

表 11-1: 从模式缓冲区寻址

$PMA<1:0>$	输出寄存器 (缓冲区)	输入寄存器 (缓冲区)
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 / 从器件连接寻址缓冲区示例

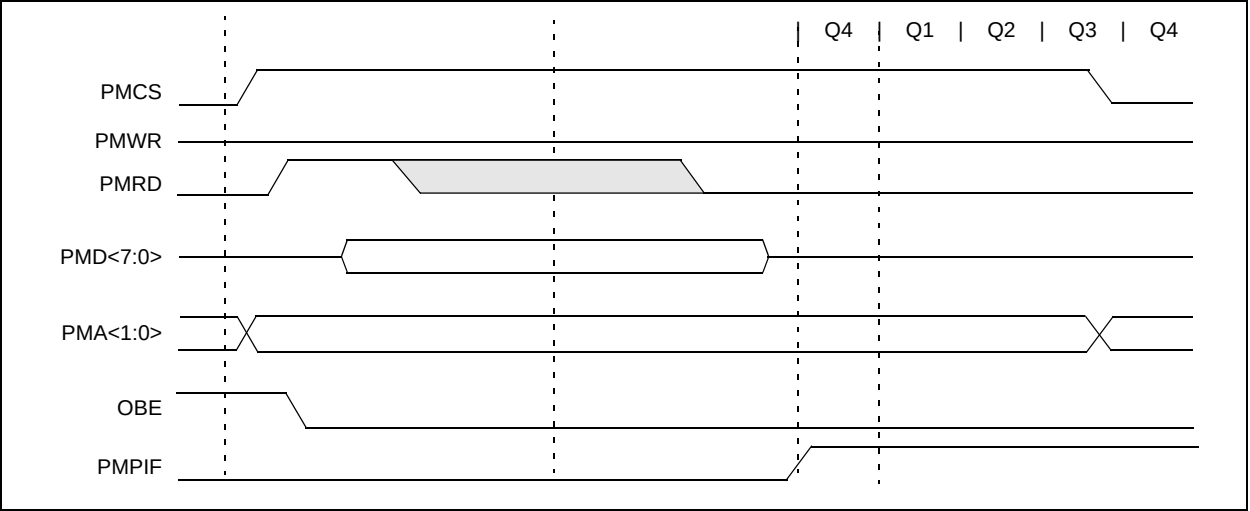


11.2.5.1 读从端口

当片选有效并产生读选通（ $PMCS = 1$ 且 $PMRD = 1$ ）时，4 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 11-1](#) 给出了

对应的输出寄存器及其相关地址。当读输出缓冲区时，对应的 $OBxE$ 位被置 1。 $OBxE$ 标志位在所有缓冲区为空时被置 1。如果所有缓冲区已为空， $OBxE = 1$ ，则对该缓冲区的下一次读操作将产生 $OBUF$ 事件。

图 11-7: 并行从端口读波形



PIC18F46J50 系列

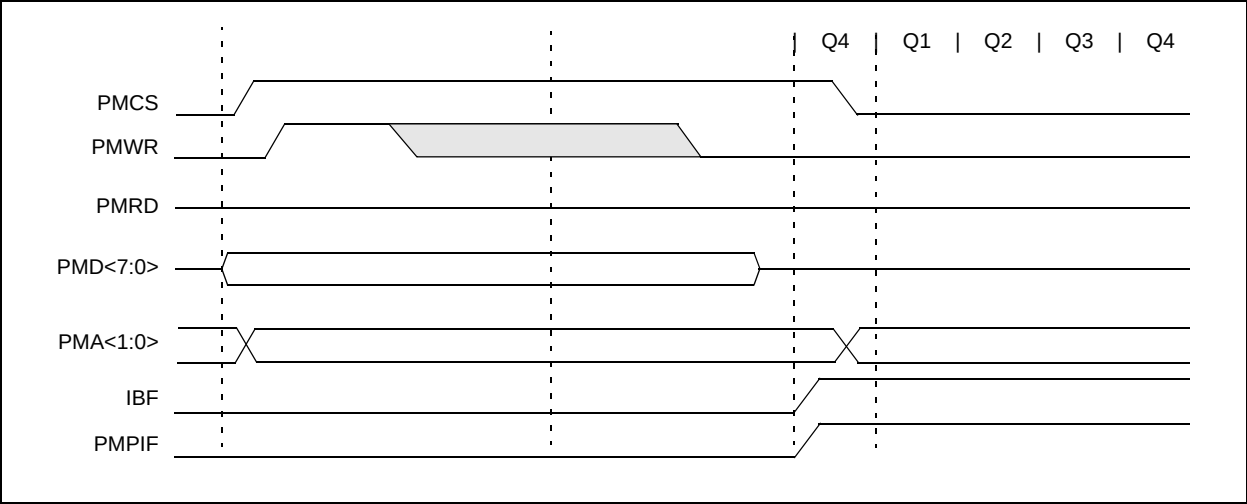
11.2.5.2 写从端口

当片选有效并产生写选通（PMCS = 1 且 PMWR = 1）时，PMD<7:0> 中的数据被捕捉到 4 个输入缓冲区字节之一。写入哪个字节取决于 ADDRL<1:0> 中的 2 位地址。

表 11-1 列出了相应的输入寄存器，以及与它们关联的地址。

当写输入缓冲区时，对应的 IBxF 位被置 1。写完所有缓冲区时，IBF 标志位被置 1。如果已写任何缓冲区（IBxF = 1），则对该缓冲区的下一次写选通将产生 OBUF 事件，并且字节将被丢弃。

图 11-8: 并行从端口写波形



11.3 主端口模式

在主模式下，PMP 模块提供了 8 位数据总线、最多 16 位地址以及操作各种外部并行器件（例如存储器件、外设和从单片机）的所有必需的控制信号。要使 PMP 工作在主模式下，必须使能模块（PMPEN = 1），并且必须设置为两个可用的主模式之一（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 位和 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控制和地址信号存在或不存在。这些位是 PTBEEN、PTWREN、PTRDEN 和 PTEN<15:0>。用户可以通过它们将引脚配置为其他功能，并可以灵活控制外部地址。当这些位中的任何位置 1 时，相关的引脚上具有相关的功能；当清零时，相关的引脚恢复为定义的 I/O 端口功能。

将 PTENx 位置 1 将使相关的引脚作为地址引脚并驱动 PMADDR 寄存器中包含的对应数据。清零 PTENx 位将强制引脚恢复为其原始的 I/O 功能。

对于配置为片选（PMCS），且相应的 PTENx 位置 1 的引脚，PTEN0 和 PTEN1 位还将控制 PMALL 和 PMALH 信号。当使用复用时，应该使能相关的地址锁存信号。

11.3.2 读 / 写控制

PMP 模块支持两种不同的读 / 写信号控制方式。在主模式 1 下，读选通和写选通组合为单根控制线 PMRD/PMWR 中。第二根控制线 PMENB 决定何时执行读操作或写操作。在主模式 2 下，在独立的引脚上提供独立的读选通和写选通（PMRD 和 PMWR）。

可以将所有控制信号（PMRD、PMWR、PMBE、PMENB、PMAL 和 PMCS）单独配置为正极性或负极性。配置由 PMCONL 寄存器中单独的位控制。请注意，共用同一输出引脚的控制信号的极性（例如，PMWR 和 PMENB）由同一个位控制；配置取决于使用的是哪种主端口模式。

11.3.3 数据宽度

PMP 支持 8 位和 16 位数据宽度。数据宽度通过 MODE16 位（PMMODEH<2>）进行选择。由于进出模块的数据路径仅为 8 位宽，16 位操作总是以复用形式处理，先送数据的最低有效字节（LSB）。为区分数据字节，通过使用字节使能控制选通 PMBE 在数据的最高有效字节（Most Significant Byte, MSB）送到数据线时发出信号。

11.3.4 地址复用

在任一主模式下（PMMODEH<1:0> = 1x），用户可以将地址总线配置为与数据总线复用。这是通过使用 ADRMUX<1:0> 位（PMCONH<4:3>）实现的。有三种地址复用模式可用；图 11-9、图 11-10 和图 11-11 给出了这些模式的典型引脚配置。

在解复用模式下（PMCONH<4:3> = 00），数据和地址信息是完全分开的。数据位送到 PMD<7:0>，地址位送到 PMADDRH<6:0> 和 PMADDRL<7:0>。

在部分复用的模式下（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<6:0>。PMA0 引脚用作地址锁存器，并且提供地址锁存器低字节（PMALL）使能选通。读和写序列在一个完整 CPU 周期内展开，其间地址位于 PMD<7:0> 引脚上。

在完全复用的模式下（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 引脚分别用于提供地址锁存器低字节（PMALL）使能选通和地址锁存器高字节（PMALH）使能选通。读和写序列在两个完整 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，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，PMALH 选通有效。如果将高地址位配置为片选引脚，则对应的地址位被自动强制为 0。

PIC18F46J50 系列

图 11-9: 解复用的寻址模式（独立读和写选通，带有片选）

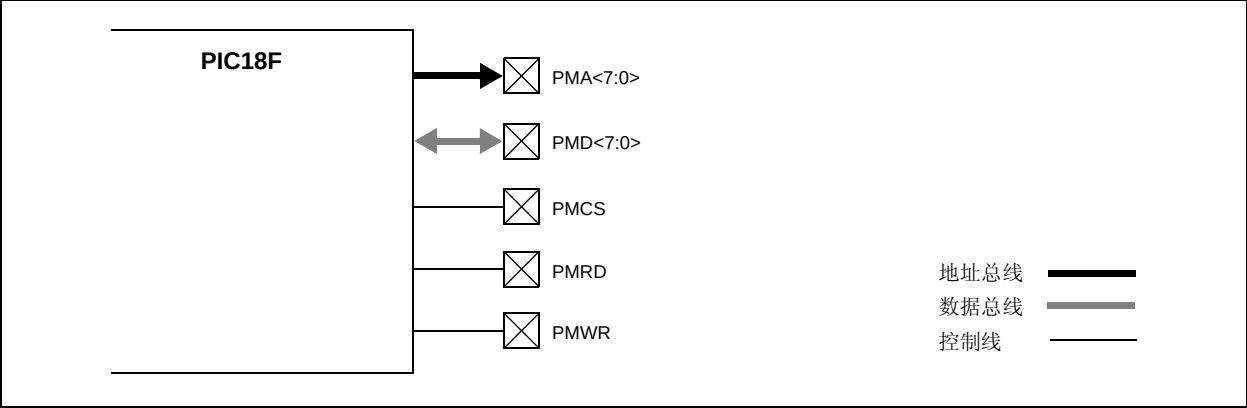


图 11-10: 部分复用的寻址模式（独立读和写选通，带有片选）

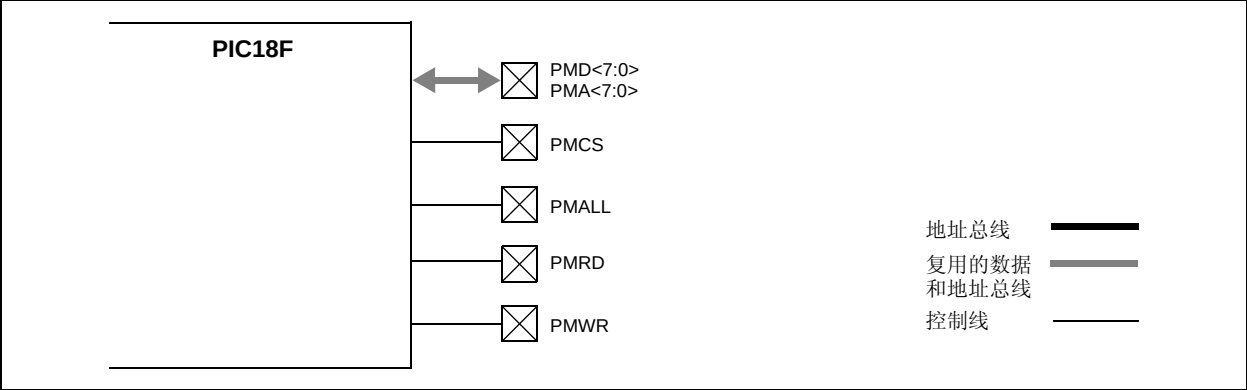
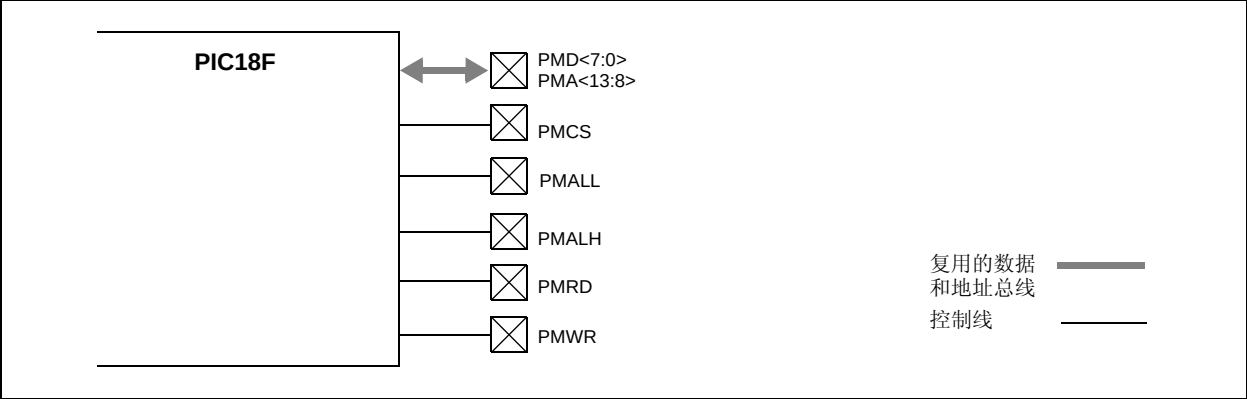


图 11-11: 完全复用的寻址模式（独立读和写选通，带有片选）



11.3.5 片选功能

PMP 的主模式下有一根片选信号线 **PMCS**。这根片选线与地址总线的次高位 (**MSb**) (**PMADDRH<6>**) 复用。当将引脚配置为片选时, 在地址自动递增 / 递减操作中不会包括 **PMADDRH<7:6>** 位。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

当模块工作在某种主模式下时, **INCMx** 位 (**PMMODEH<4:3>**) 控制地址值的行为。可以将地址配置为在每个读和写操作之后自动递增或递减。在每完成一次操作后地址递增一次, 并且 **BUSY** 位变为 0。如果禁止了片选信号并配置为地址位, 则这些位将进行递增和递减操作; 否则 **CS1** 位的值不受影响。

11.3.7 等待状态

在主模式下, 通过配置模块等待状态, 用户可以实现对读、写和地址周期期间的控制。周期的三个部分 (开始、中间和末尾) 通过 **PMMODEL** 寄存器中的对应 **WAITBx**、**WAITMx** 和 **WAITEx** 位进行配置。

WAITBx 位 (**PMMODEL<7:6>**) 设置在 **PMRD/PMWT** 选通之前 (模式 10) 或在 **PMENB** 选通之前 (模式 11) 数据建立的等待周期数。**WAITMx** 位 (**PMMODEL<5:2>**) 设置 **PMRD/PMWT** 选通 (模式 10) 或 **PMENB** 选通 (模式 11) 的等待周期数。当该等待状态设置为 0 时, **WAITB** 和 **WAITE** 不起任何作用。**WAITE** 位 (**PMMODEL<1:0>**) 定义在 **PMRD/PMWT** 选通之后 (模式 10) 或在 **PMENB** 选通之后 (模式 11) 数据保持时间的等待周期数。

11.3.8 读操作

要对 PMP 执行读操作, 用户可以读取 **PMDIN1L** 寄存器。这使 PMP 在片选线和地址总线上输出期望的值。然后读线 (**PMRD**) 被选通。读取的数据将放入 **PMDIN1L** 寄存器。

如果使能了 16 位模式 (**MODE16 = 1**), 读 **PMDIN1L** 寄存器的低字节将启动两次总线读操作。读取的第一个数据字节放入 **PMDIN1L** 寄存器, 读取的第二个数据字节放入 **PMDIN1H**。

请注意, 从 **PMDIN1L** 寄存器读取的数据实际上是前面读操作中读取的值。因此, 用户执行的第一次读操作是一次假读操作, 它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, 在背对背读操作中, 两次读操作中从寄存器读取的数据相同。下一次读取寄存器将产生新的值。

11.3.9 写操作

要对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到 **PMD<7:0>** 数据总线。然后写操作线 (**PMWR**) 被选通。如果使能了 16 位模式 (**MODE16 = 1**), 写 **PMDIN1L** 寄存器将启动两次总线写操作。第一个写操作的数据是 **PMDIN1L** 中包含的数据, 第二个写操作的数据是 **PMDIN1H** 中的数据。

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。该位仅在主模式下使用。当正在进行任何读或写操作时, 除了操作的最后一个 CPU 周期, **BUSY** 位都被置 1。事实上, 如果请求单周期读或写操作, 则 **BUSY** 位永远不会有效。这允许背对背传输。当该位置 1 时, 将忽略用户任何的启动新操作请求 (即, 读或写 **PMDIN1L** 寄存器的低字节不会启动读或写操作)。

11.3.10.2 中断

为主模式允许 PMP 模块中断时, 模块将在每次完成读周期或写周期时产生中断; 否则, 可以通过 **BUSY** 位来查询模块的状态。

PIC18F46J50 系列

11.3.11 主模式时序

本节包含与常用主模式配置选项相对应的许多时序示例。这些选项从 8 位数据到 16 位数据、从完全解复用的地址到完全复用的地址，以及等待状态。

图 11-12: 读和写时序，8 位数据，解复用的地址

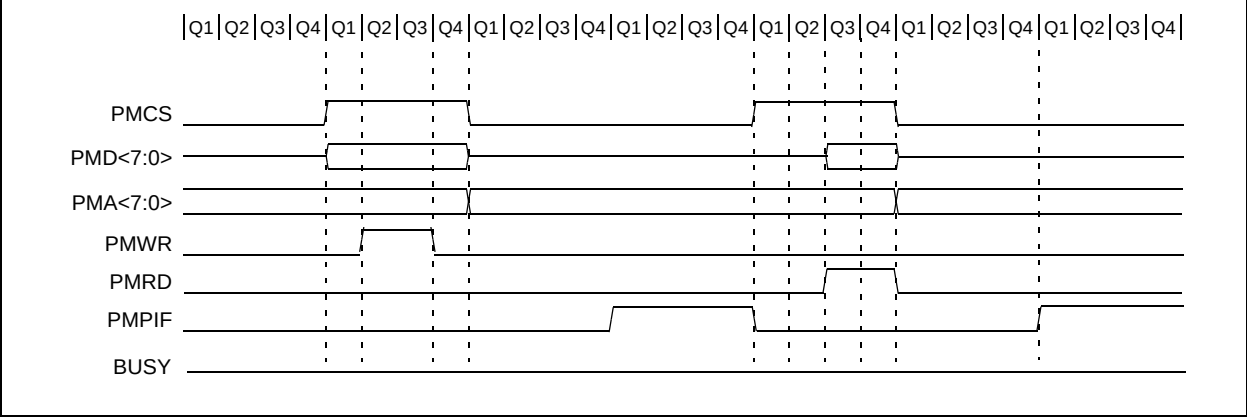


图 11-13: 读时序，8 位数据，部分复用的地址

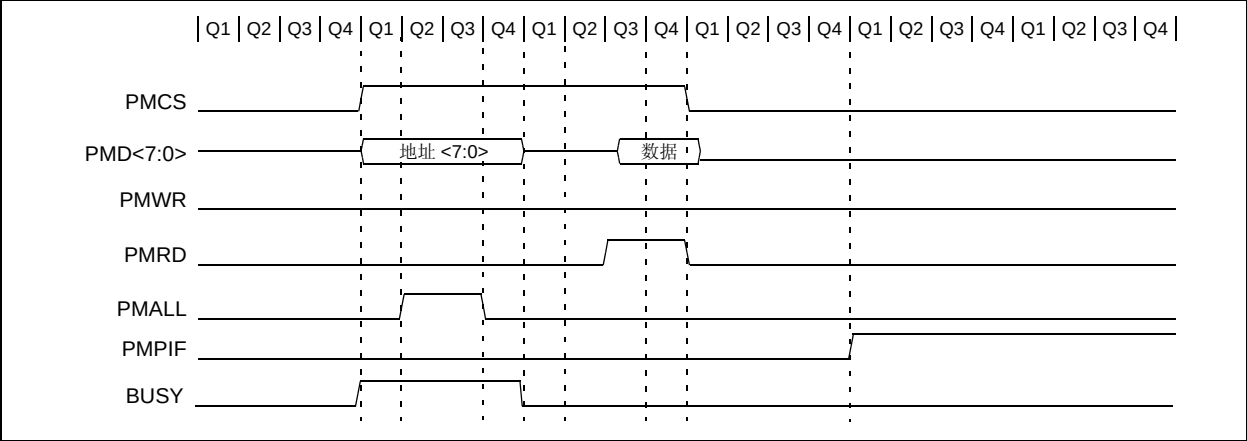


图 11-14: 读时序，8 位数据，使能等待状态，部分复用的地址

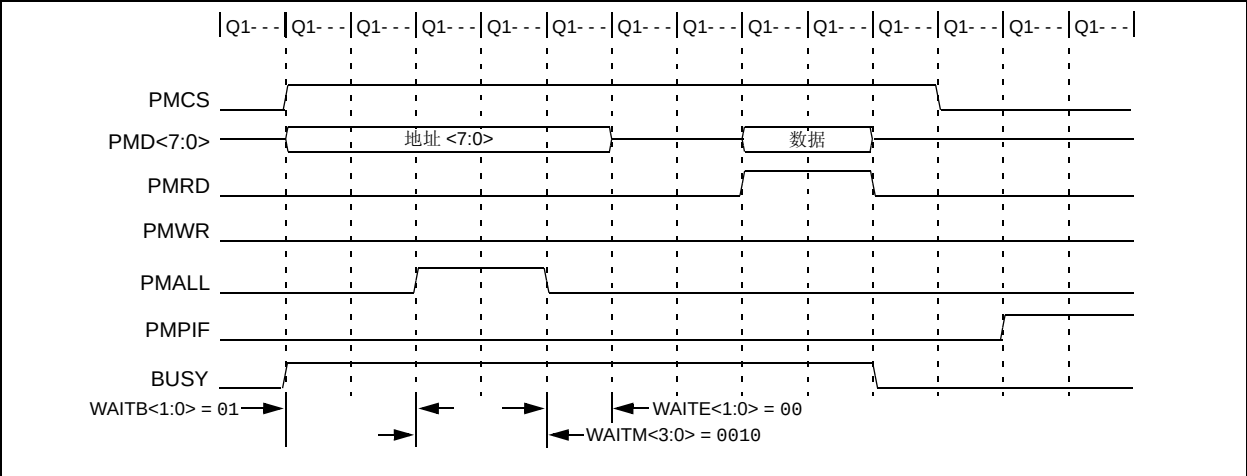


图 11-15: 写时序, 8 位数据, 部分复用的地址

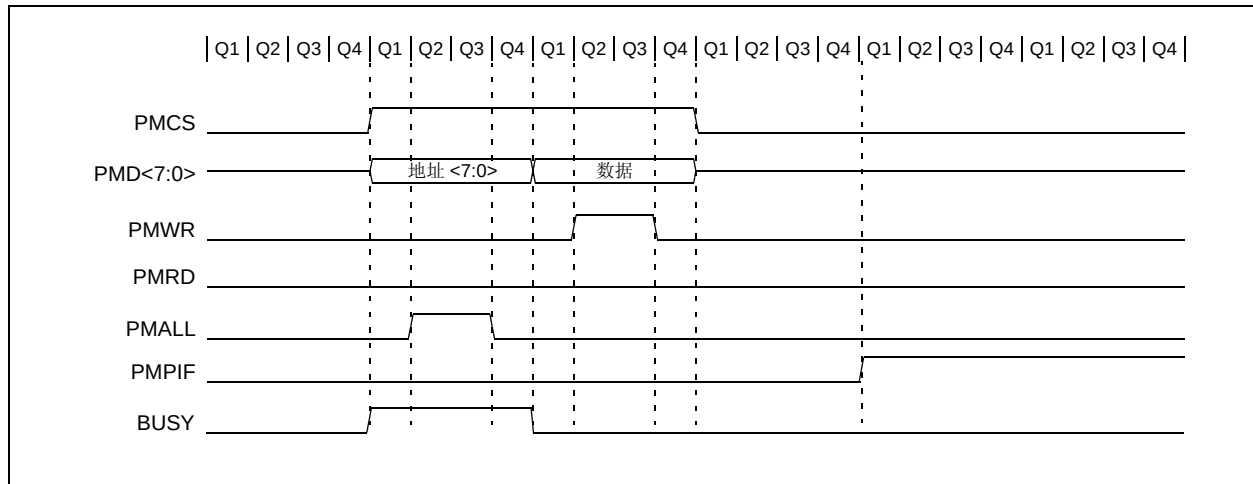


图 11-16: 写时序, 8 位数据, 使能等待状态, 部分复用的地址

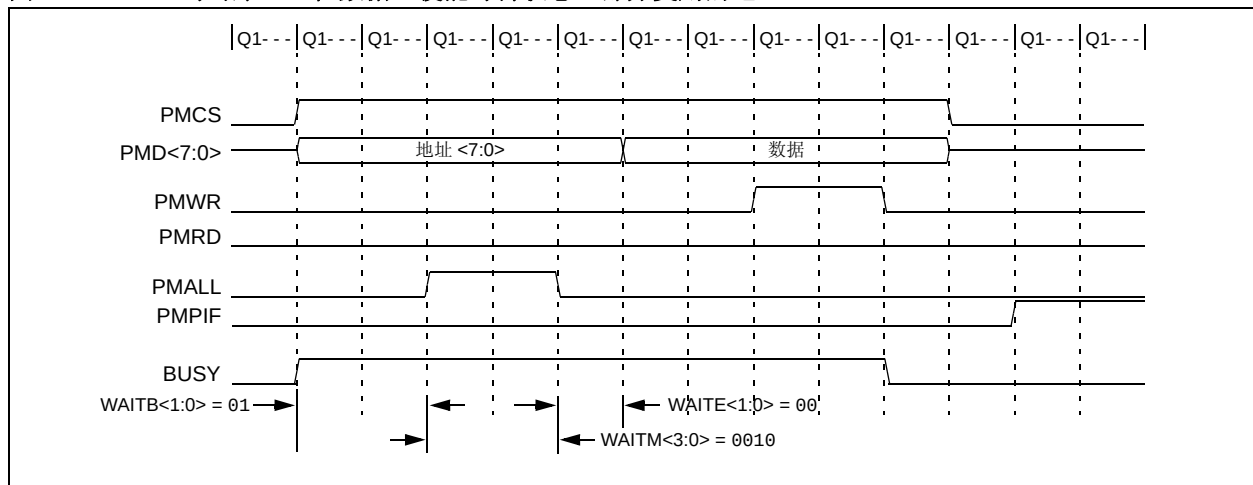
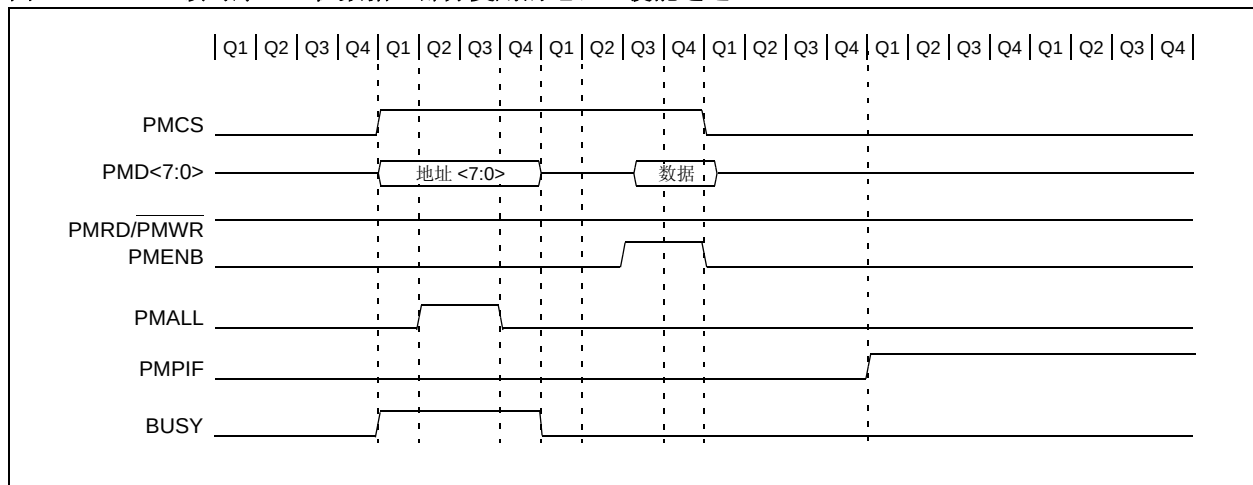


图 11-17: 读时序, 8 位数据, 部分复用的地址, 使能选通



PIC18F46J50 系列

图 11-18: 写时序，8 位数据，部分复用的地址，使能选通

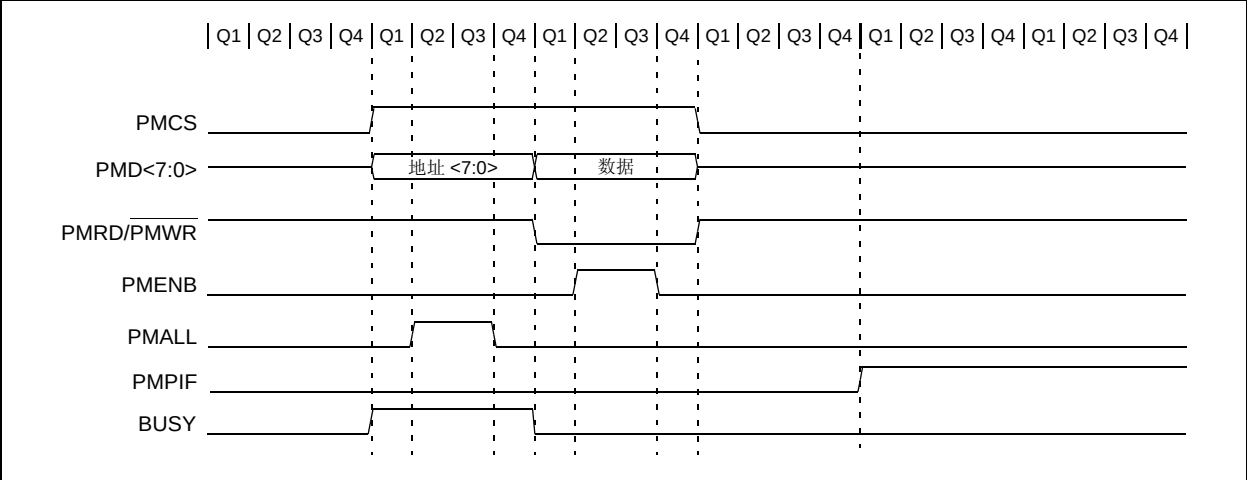


图 11-19: 读时序，8 位数据，完全复用的 16 位地址

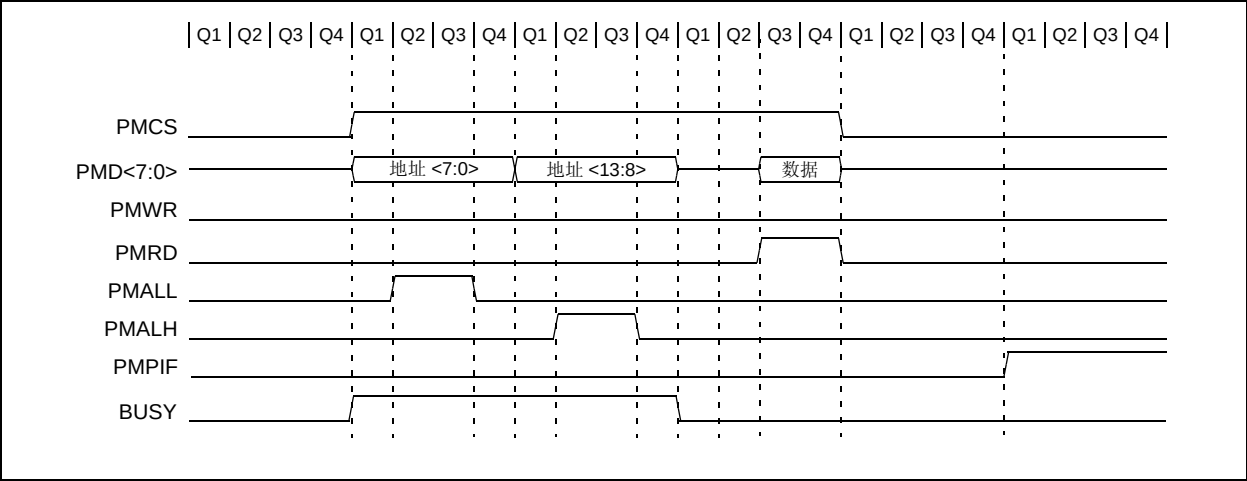


图 11-20: 写时序，8 位数据，完全复用的 16 位地址

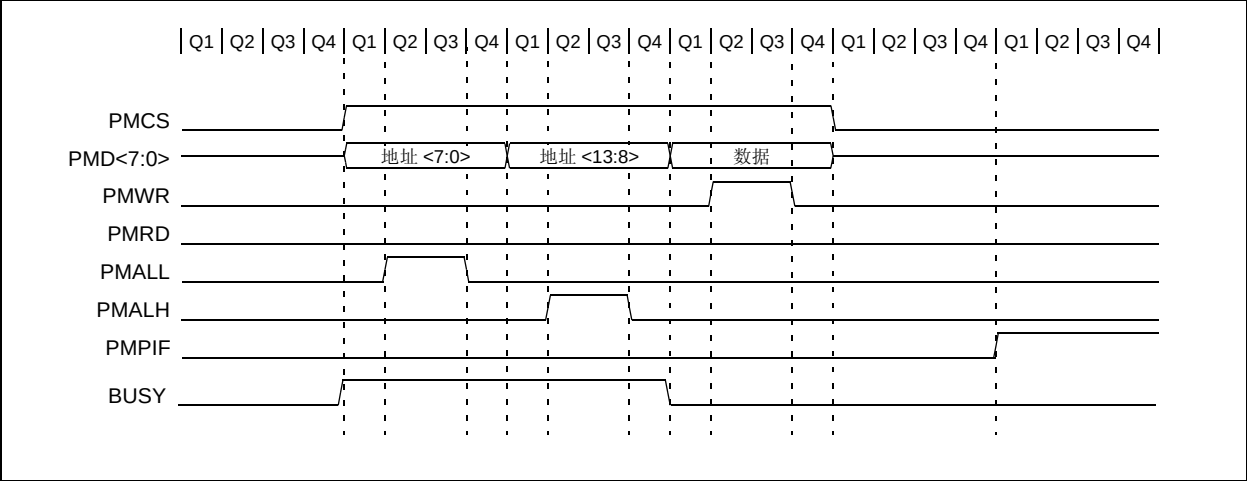


图 11-21: 读时序, 16 位数据, 解复用的地址

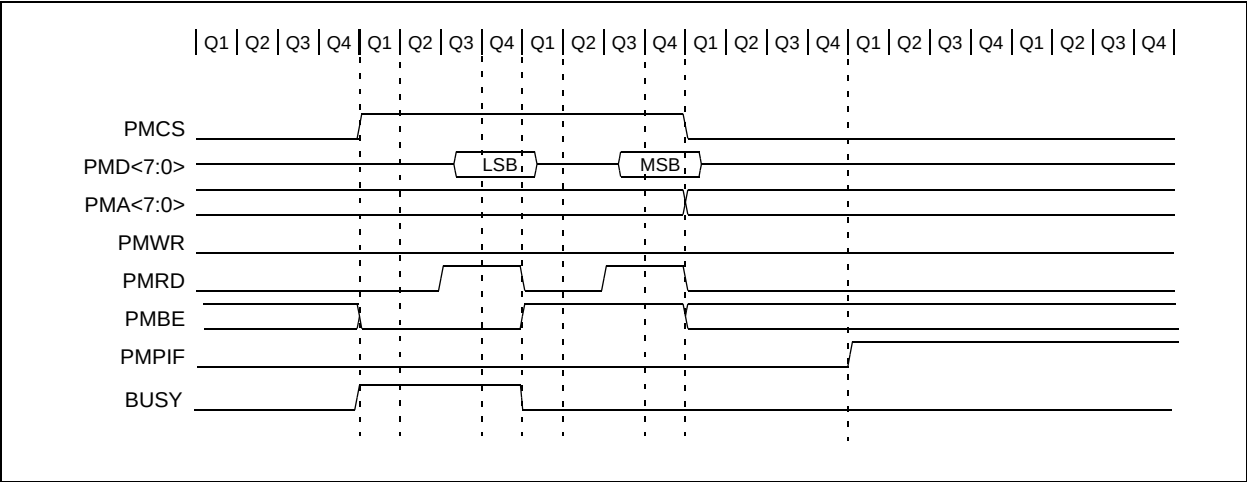


图 11-22: 写时序, 16 位数据, 解复用的地址

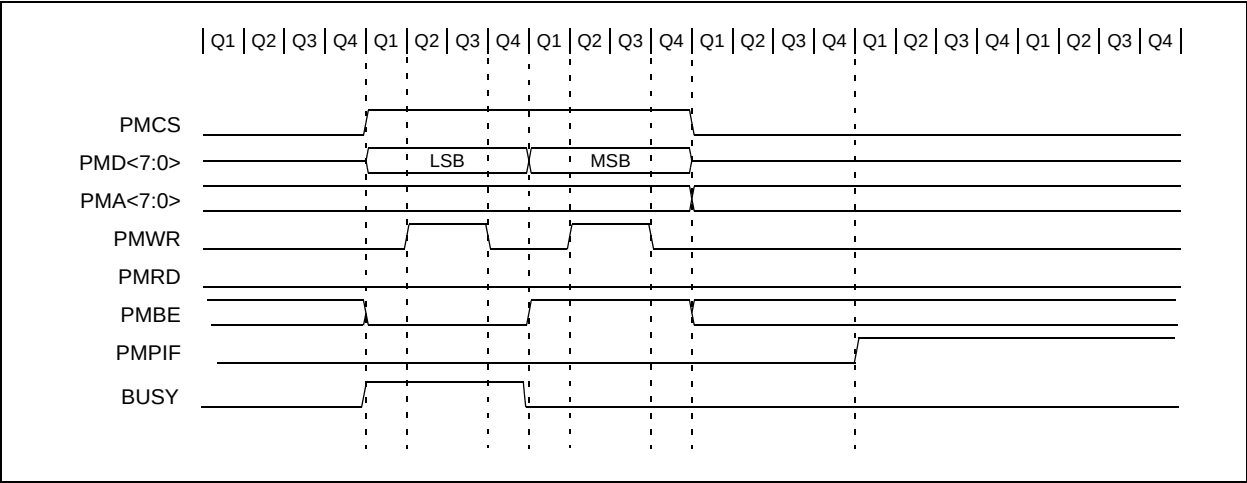
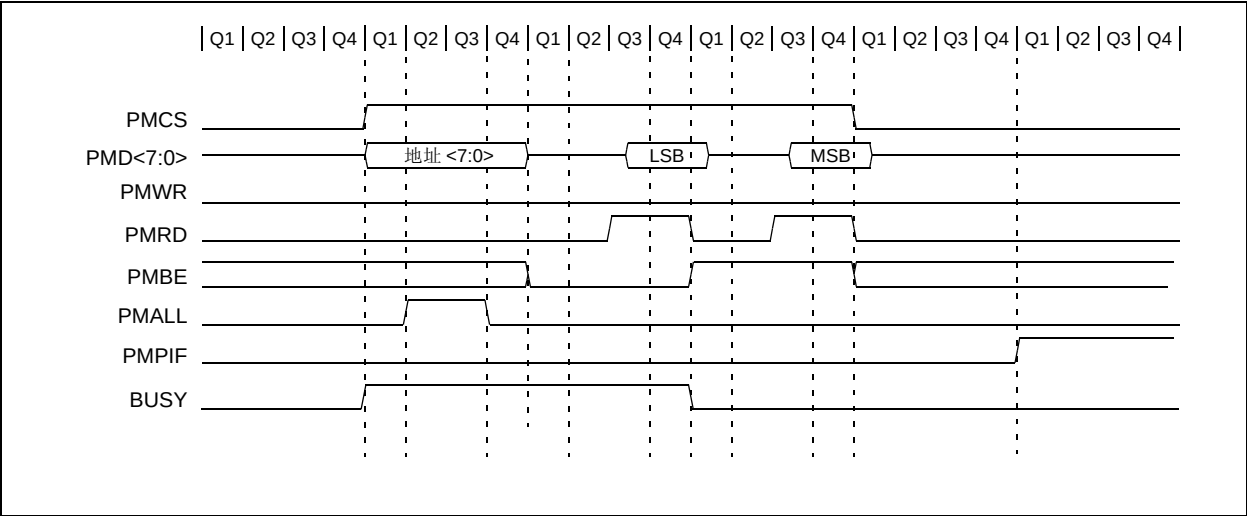


图 11-23: 读时序, 16 位复用的数据, 部分复用的地址



PIC18F46J50 系列

图 11-24: 写时序，16 位复用的数据，部分复用的地址

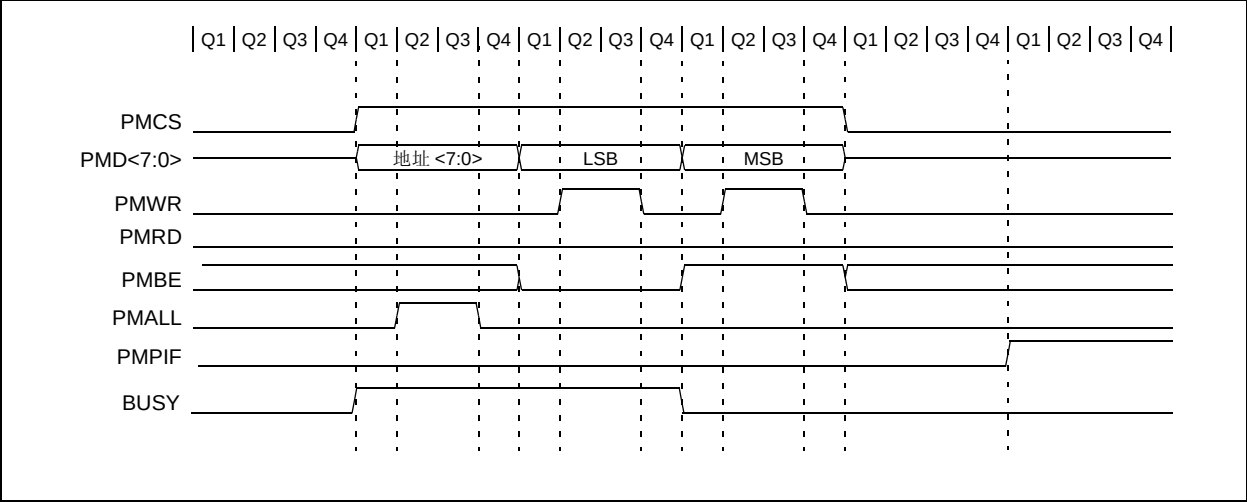


图 11-25: 读时序，16 位复用的数据，完全复用的 16 位地址

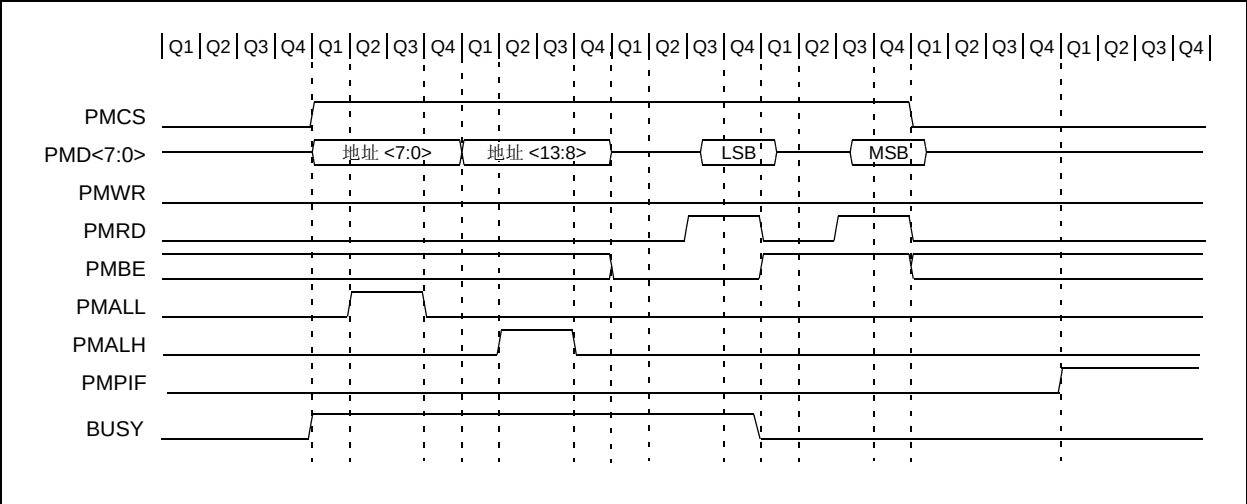
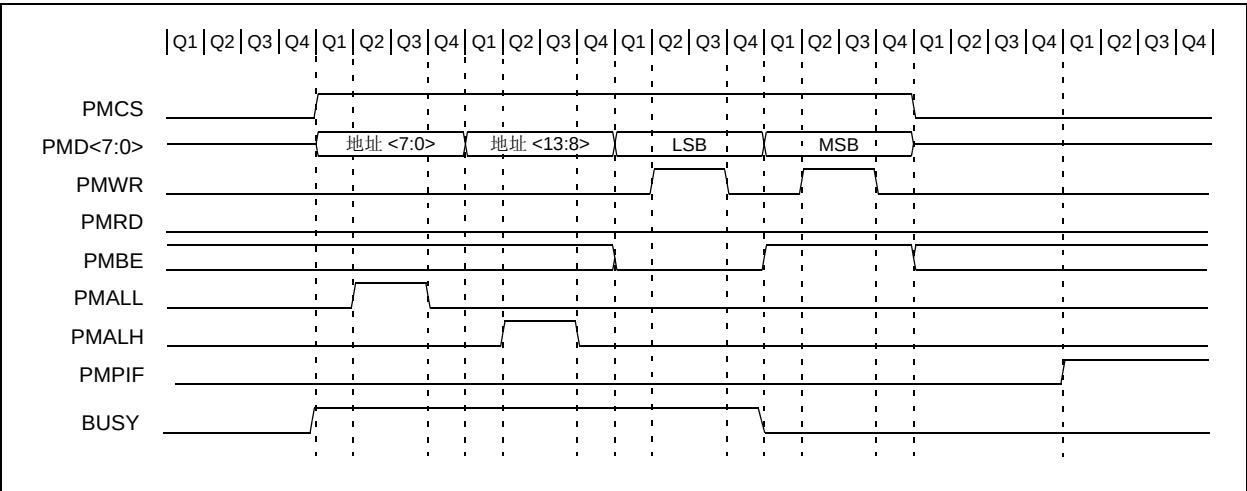


图 11-26: 写时序，16 位复用的数据，完全复用的 16 位地址



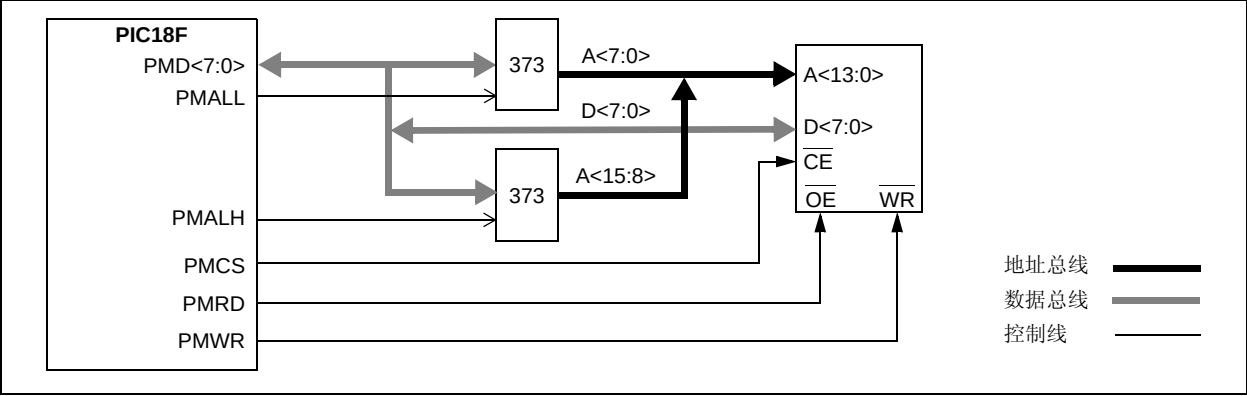
11.4 应用示例

本节介绍 PMP 模块的部分可能的应用。

11.4.1 复用的存储器或外设

图 11-27 展示了存储器或其他可寻址外设在全复用模式下的连接。因此，从单片机角度来看，这种模式最能节省引脚。但是，为了使用此配置，需要使用一些外部锁存器来保持地址。

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

部分复用需要使用更多的引脚；但是，通过几个额外引脚，可以实现额外的性能。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。如果外设具有内部

锁存器（如图 11-29 所示），则只需外设即可，无需额外的电路。

图 11-28: 部分复用的寻址应用示例

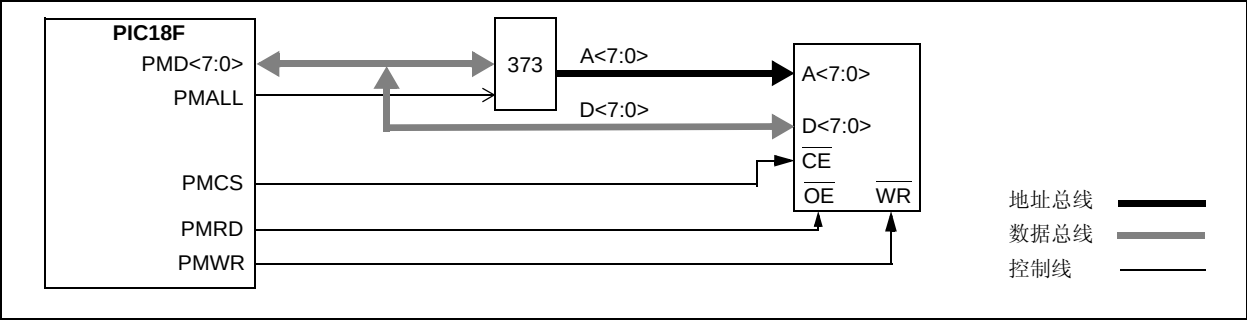


图 11-29: 8 位复用地址和数据应用示例



PIC18F46J50 系列

11.4.3 并行 EEPROM 示例

图 11-30 给出了一个将并行 EEPROM 与 PMP 连接的示例。图 11-31 给出了一个进行了稍许更改的示例，用于配置从单个 EEPROM 传输 16 位数据的连接。

图 11-30: 并行 EEPROM 示例（最多 15 位地址，8 位数据）

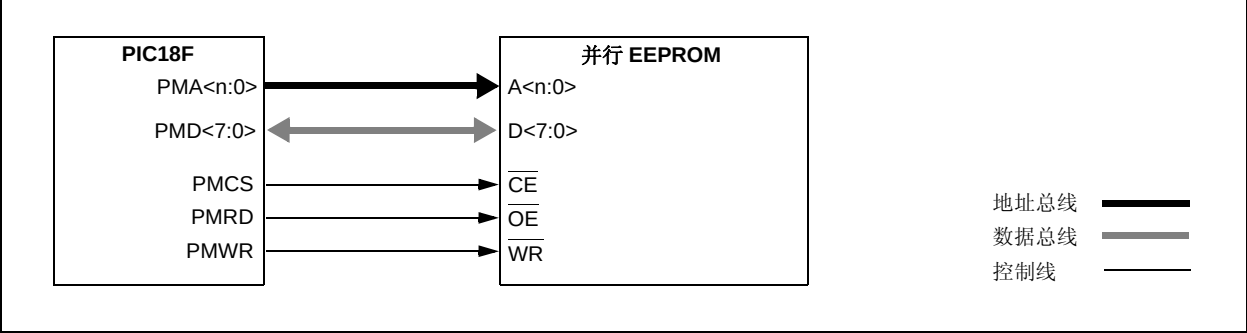
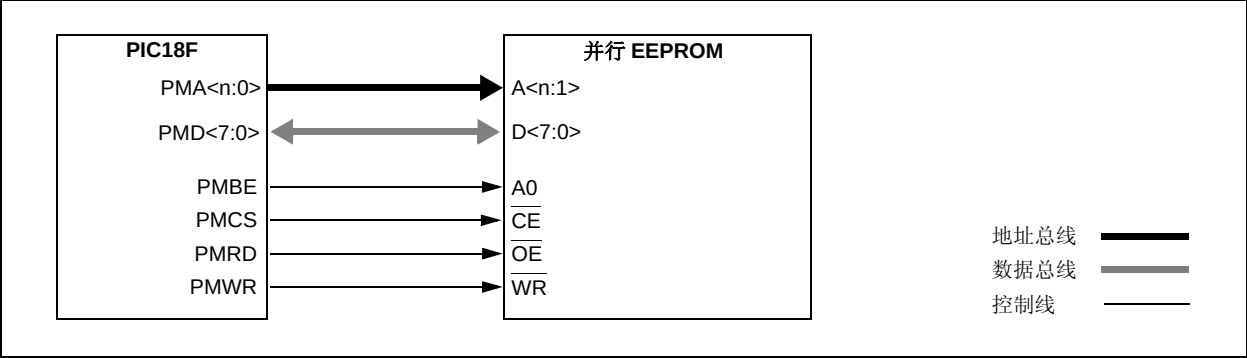


图 11-31: 并行 EEPROM 示例（最多 15 位地址，16 位数据）



11.4.4 LCD 控制器示例

可以将 PMP 模块配置为连接到典型的 LCD 控制器接口，如图 11-32 所示。在此例中，PMP 模块被配置为控制信号高电平有效，这是因为常见的 LCD 显示器需要高电平有效控制。

图 11-32: LCD 控制示例（字节模式操作）

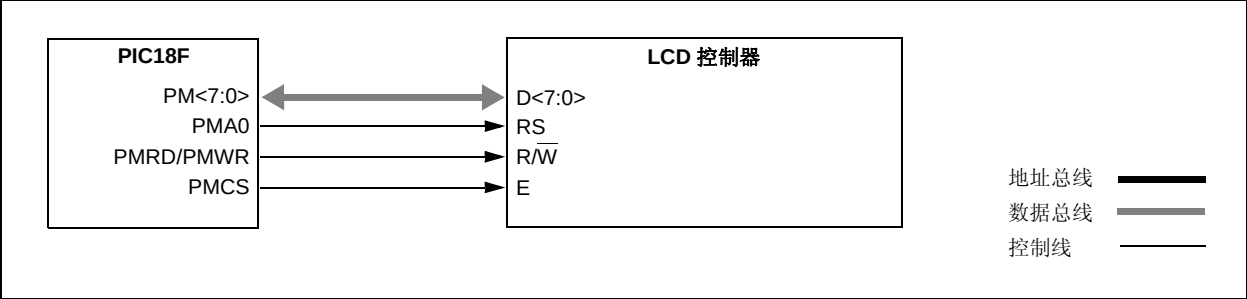


表 11-2: 与 PMP 模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽²⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽²⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP ⁽²⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PMCONH ⁽²⁾	PMPEN	—	—	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN	73
PMCONL ⁽²⁾	CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP	73
PMADDRH ^(1,2) /	—	CS1	并行主端口地址的高字节						73
PMDOUT1H ^(1,2)	并行端口输出数据的高字节（缓冲区 1）								73
PMADDRL ^(1,2) /	并行主端口地址的低字节								73
PMDOUT1L ^(1,2)	并行端口输出数据的低字节（缓冲区 0）								73
PMDOUT2H ⁽²⁾	并行端口输出数据的高字节（缓冲区 3）								74
PMDOUT2L ⁽²⁾	并行端口输出数据的低字节（缓冲区 2）								74
PMDIN1H ⁽²⁾	并行端口输入数据的高字节（缓冲区 1）								73
PMDIN1L ⁽²⁾	并行端口输入数据的低字节（缓冲区 0）								73
PMDIN2H ⁽²⁾	并行端口输入数据的高字节（缓冲区 3）								74
PMDIN2L ⁽²⁾	并行端口输入数据的低字节（缓冲区 2）								74
PMMODEH ⁽²⁾	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	74
PMMODEL ⁽²⁾	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	74
PMEH ⁽²⁾	—	PTEN14	—	—	—	—	—	—	74
PMEL ⁽²⁾	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	74
PMSTATH ⁽²⁾	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	74
PMSTATL ⁽²⁾	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	74
PADCFG1	—	—	—	—	—	RTSECSEL1	RTSECSEL0	PMPTTL	74

图注: — = 未实现, 读为 0。PMP 操作期间不使用阴影单元。

注 1: PMADDRH/PMDOUT1H 和 PMADDRL/PMDOUT1L 寄存器对共用物理寄存器和地址, 但具有由模块工作模式所决定的不同功能。

2: 这些位和 / 或寄存器仅在 44 引脚器件上可用。

PIC18F46J50 系列

注:

12.0 TIMER0 模块

Timer0 模块具有以下特性：

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

T0CON 寄存器（[寄存器 12-1](#)）控制该模块操作的所有方面，包括预分频比的选择。它是可读写的。

[图 12-1](#) 给出了 8 位模式下 Timer0 模块的简化框图。

[图 12-2](#) 给出了 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（位于快速操作存储区，地址 FD5h）

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 = POR 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 引脚输入信号的电平跳变沿（有效的边沿由 T0SE 位设置）

0 = 内部时钟（Fosc/4）

bit 4 **T0SE:** Timer0 时钟源边沿选择位

1 = 在 T0CKI 引脚信号从高至低跳变时，递增计数

0 = 在 T0CKI 引脚信号从低至高跳变时，递增计数

bit 3 **PSA:** Timer0 预分频器分配位

1 = 未分配 Timer0 预分频器。Timer0 时钟输入不经预分频器分频

0 = 已分配 Timer0 预分频器。Timer0 时钟输入来自预分频器的输出

bit 2-0 **T0PS<2:0>:** Timer0 预分频比选择位

111 = 1:256 预分频比

110 = 1:128 预分频比

101 = 1:64 预分频比

100 = 1:32 预分频比

011 = 1:16 预分频比

010 = 1:8 预分频比

001 = 1:4 预分频比

000 = 1:2 预分频比

PIC18F46J50 系列

12.1 Timer0 工作原理

Timer0既可用作定时器也可用作计数器。可以通过T0CS位（T0CON<5>）来选择模式。在定时器模式（T0CS = 0）下，该模块在每个时钟周期都会递增（默认情况下），除非选择了其他预分频值（见第12.3节“预分频器”）。在对TMR0寄存器执行写操作之后的两个指令周期内Timer0禁止递增。用户可通过将调整值写入TMR0寄存器来解决这一问题。

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

可以使用外部时钟源来驱动Timer0；但是，必须满足一定要求，以确保外部时钟和内部相位时钟（Tosc）保

持同步。在同步之后，定时器/计数器需要一定的延时才开始递增。

12.2 16位模式下Timer0的读写操作

TMR0H并不是16位模式下Timer0的实际高字节，而是Timer0实际高字节的缓存形式，不可以被直接读写（见图12-2）。在读TMR0L时使用Timer0高字节的内容更新TMR0H。这种方式使用户可以读取Timer0的全部16位，而不需要验证高字节和低字节读取的有效性。由于高字节和低字节连续读取之间的计满返回，可能会产生无效读取。

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

图 12-1: TIMER0 框图（8 位模式）

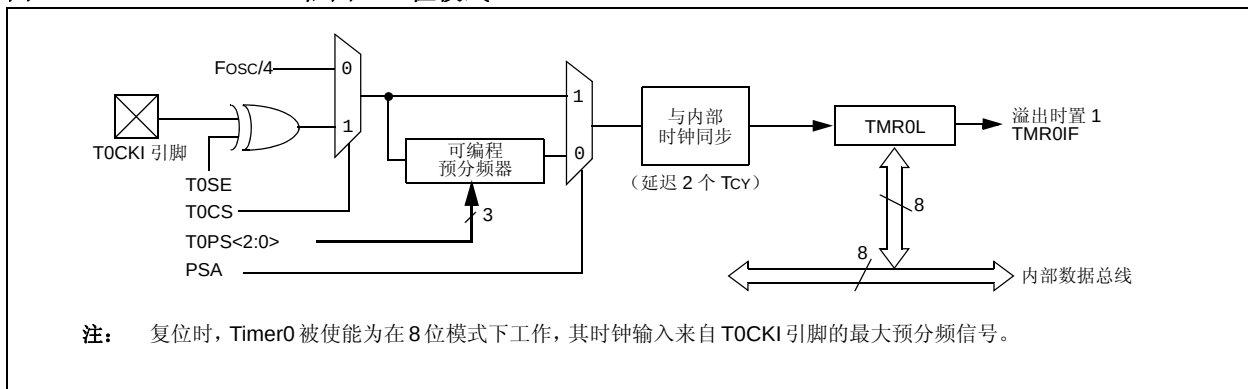
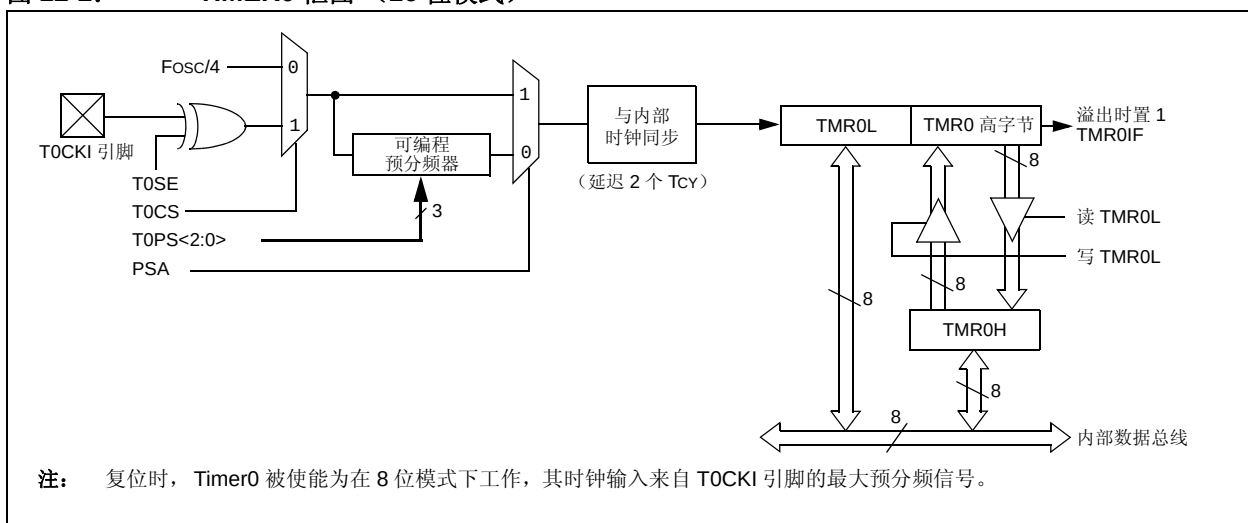


图 12-2: TIMER0 框图（16 位模式）



12.3 预分频器

Timer0 模块的预分频器为一个 8 位计数器。该预分频器不可直接读写；通过 PSA 和 T0PS<2:0> 位 (T0CON<3:0>) 进行预分频器的分配和设定预分频比。

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

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

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

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 位模式下的 TMR0 寄存器从 FFh 溢出到 00h，或 16 位模式下的 TMR0 从 FFFFh 溢出到 0000h 时，将产生 TMR0 中断。这种溢出会使 TMR0IF 标志位置 1。可以通过清零 TMR0IE 位 (INTCON<5>) 来屏蔽该中断。在重新允许该中断前，必须在中断服务程序 (ISR) 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 所在页:
TMR0L	Timer0 低字节寄存器								90
TMR0H	Timer0 高字节寄存器								90
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	89
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	90

图注： — = 未实现，读为 0。Timer0 不使用阴影单元。

PIC18F46J50 系列

注:

13.0 TIMER1 模块

Timer1 定时器 / 计数器模块具有以下特性：

- 可由软件选择作为 16 位定时器或计数器
- 可读写的 8 位寄存器（TMR1H 和 TMR1L）
- 可选择器件时钟或 Timer1 内部振荡器作为时钟源（内部或外部）
- 溢出时产生中断
- 在 ECCP 特殊事件触发时复位
- 器件时钟状态标志位（T1RUN）
- 带门控的定时器

图 13-1 给出了 Timer1 模块的简化框图。

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

Timer1 由 T1CON 控制寄存器（寄存器 13-1）控制。该寄存器还包含 Timer1 振荡器使能位（T1OSCEN）。可以通过将控制位 TMR1ON（T1CON<0>）置 1 或清零来使能或禁止 Timer1。

Fosc 时钟源（TMR1CS<1:0> = 01）不应与 ECCP 捕捉 / 比较功能一起使用。如果定时器要与捕捉或比较功能一起使用，总是选择其他定时器时钟源选项之一。

寄存器 13-1: T1CON: TIMER1 控制寄存器（位于快速操作存储区，地址 FCDh）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	RD16	TMR1ON
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-6 **TMR1CS<1:0>**: Timer1 时钟源选择位

10 = Timer1 时钟源为 T1OSC 引脚或 T1CKI 引脚

01 = Timer1 时钟源为系统时钟（Fosc）⁽¹⁾

00 = Timer1 时钟源为指令时钟（Fosc/4）

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

11 = 1:8 预分频比

10 = 1:4 预分频比

01 = 1:2 预分频比

00 = 1:1 预分频比

bit 3 **T1OSCEN**: Timer1 振荡器源选择位

当 TMR1CS<1:0> = 10 时：

1 = 给 Timer1 晶振驱动器上电，并依靠晶振的输出提供 Timer1 时钟

0 = Timer1 晶振驱动器断电，Timer1 时钟来自 T1CKI 输入引脚⁽²⁾

当 TMR1CS<1:0> = 0x 时：

1 = 给 Timer1 晶振驱动器上电

0 = Timer1 晶振驱动器断电⁽²⁾

bit 2 **T1SYNC**: Timer1 外部时钟输入同步选择位

TMR1CS<1:0> = 10:

1 = 不同步外部时钟输入

0 = 同步外部时钟输入

TMR1CS<1:0> = 0x:

该位为无关位。当 TMR1CS<1:0> = 0x 时，Timer1 使用内部时钟。

注 1: 如果定时器要与 ECCP 捕捉 / 比较功能一起使用，则不应选择 Fosc 时钟源。

2: 当 T1OSCEN（T1CON）或 T3OSCEN（T3CON）= 1 时，Timer1 振荡器晶振驱动器上电。这两个位的逻辑或决定是否使能驱动器电路。驱动器电路禁止时，反相器和反馈电阻被禁止以降低功耗。要使晶振驱动器上电，并不必使能 TMR1ON 和 TMR3ON 位。

PIC18F46J50 系列

寄存器 13-1: T1CON: TIMER1 控制寄存器（位于快速操作存储区，地址 FCDh）（续）

bit 1	RD16: 16 位读 / 写模式使能位
	1 = 使能 Timer1 通过一次 16 位操作进行寄存器读 / 写
	0 = 使能 Timer1 通过两次 8 位操作进行寄存器读 / 写
bit 0	TMR1ON: Timer1 使能位
	1 = 使能 Timer1
	0 = 停止 Timer1

- 注**
- 1:** 如果定时器要与 ECCP 捕捉 / 比较功能一起使用，则不应选择 Fosc 时钟源。
 - 2:** 当 T1OSCEN（T1CON）或 T3OSCEN（T3CON）= 1 时，Timer1 振荡器晶振驱动器上电。这两个位的逻辑或决定是否使能驱动器电路。驱动器电路禁止时，反相器和反馈电阻被禁止以降低功耗。要使晶振驱动器上电，并不必使能 TMR1ON 和 TMR3ON 位。

13.1 Timer1 门控控制寄存器

Timer1 门控控制寄存器（T1GCON）如寄存器 13-2 所示，用于控制 Timer1 门控信号。

寄存器 13-2: T1GCON: TIMER1 门控控制寄存器（位于快速操作存储区，地址 F9Ah）⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/T1DONE	T1GVAL	T1GSS1	T1GSS0
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位，读为 0
-n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7 **TMR1GE:** Timer1 门控使能位

如果 TMR1ON = 0:

该位为无关位。

如果 TMR1ON = 1:

1 = Timer1 计数由 Timer1 门控功能控制

0 = Timer1 计数与 Timer1 门控功能无关

bit 6 **T1GPOL:** Timer1 门控极性位

1 = Timer1 门控高电平有效（当门控为高电平时 Timer1 计数）

0 = Timer1 门控低电平有效（当门控为低电平时 Timer1 计数）

bit 5 **T1GTM:** Timer1 门控翻转模式位

1 = 使能 Timer1 门控翻转模式

0 = 禁止 Timer1 门控翻转模式并清除翻转触发器

Timer1 门控触发器在每个上升沿翻转。

bit 4 **T1GSPM:** Timer1 门控单脉冲模式位

1 = 使能 Timer1 门控单脉冲模式，控制 Timer1 门控

0 = 禁止 Timer1 门控单脉冲模式

bit 3 **T1GGO/T1DONE:** Timer1 门控单脉冲采集状态位

1 = Timer1 门控单脉冲采集就绪，正在等待一个边沿

0 = Timer1 门控单脉冲采集已经结束或尚未开始

当 T1GSPM 清零时，该位会自动清零。

bit 2 **T1GVAL:** Timer1 门控当前状态位

表示可提供给 TMR1H:TMR1L 的 Timer1 门控的当前状态；不受 Timer1 门控使能（TMR1GE）位的影响。

bit 1-0 **T1GSS<1:0>:** Timer1 门控源选择位

00 = Timer1 门控引脚

01 = Timer0 溢出输出

10 = TMR2 匹配 PR2 输出

注 1: 建议在设定 T1CON 之前先设定 T1GCON。

PIC18F46J50 系列

寄存器 13-3: TCLKCON: 定时器时钟控制寄存器（位于普通存储区，地址 F52h）

U-0	U-0	U-0	R-0	U-0	U-0	R/W-0	R/W-0
—	—	—	T1RUN	—	—	T3CCP2	T3CCP1
bit 7							bit 0

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

- bit 7-5 **未实现：** 读为 0
- bit 4 **T1RUN:** Timer1 运行状态位
 1 = 目前器件时钟来自 T1OSC/T1CKI
 0 = 系统时钟来自除 T1OSC/T1CKI 外的振荡器
- bit 3-2 **未实现：** 读为 0
- bit 1-0 **T3CCP<2:1>:** ECCP 定时器分配位
 10 = ECCP1 和 ECCP2 均使用 Timer3（捕捉 / 比较）和 Timer4（PWM）
 01 = ECCP1 使用 Timer1（比较 / 捕捉）和 Timer2（PWM）； ECCP2 使用 Timer3（捕捉 / 比较）和 Timer4（PWM）
 00 = ECCP1 和 ECCP2 均使用 Timer1（捕捉 / 比较）和 Timer2（PWM）

13.2 Timer1 工作原理

Timer1 模块是一个 8 位或 16 位递增计数器，可通过 TMR1H:TMR1L 寄存器对访问。

与内部时钟源一起使用时，模块为定时器并在每个指令周期递增。与外部时钟源一起使用时，模块可用作定时器或计数器，在外部时钟源的每个选定边沿上递增。

Timer1 通过分别配置 T1CON 和 T1GCON 寄存器中的 TMR1ON 和 TMR1GE 位使能。

当使能 Timer1 时，RC1/T1OSI/ $\overline{\text{UOE}}$ /RP12 和 RC0/T1OSO/T1CKI/RP11 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。

13.3 时钟源选择

T1CON 寄存器的 TMR1CS<1:0> 和 T1OSCEN 位用于选择 Timer1 的时钟源。寄存器 13-1 显示了时钟源选择。

13.3.1 内部时钟源

当选择内部时钟源时，TMR1H:TMR1L 寄存器对将在 Fosc 的整数倍（由 Timer1 预分频器决定）处递增。

13.3.2 外部时钟源

当选择外部时钟源时，Timer1 模块可以作为定时器或计数器工作。

计数时，Timer1 在外部时钟输入 T1CKI 或容性传感振荡器信号的上升沿递增。这些外部时钟源可以与单片机系统时钟同步，也可以异步运行。

依靠时钟振荡器作为定时器工作时，可以将外部 32.768 kHz 晶振与专用内部振荡器电路一起使用。

注：

在计数器模式下，发生以下任何一个或多个情况后，计数器在首个上升沿递增前，必须先经过一个下降沿：

• Timer1 在 POR 复位后被使能

• 写入 TMR1H 或 TMR1L

• Timer1 被禁止

• T1CKI 为高电平时 Timer1 被禁止（TMR1ON = 0），然后在 T1CKI 为低电平时 Timer1 被使能（TMR1ON = 1）。

表 13-1: TIMER1 时钟源选择

TMR1CS1	TMR1CS0	T1OSCEN	时钟源
0	1	x	时钟源（Fosc）
0	0	x	指令时钟（Fosc/4）
1	0	0	T1CKI 引脚上的外部时钟
1	0	1	T1OSI/T1OSO 引脚上的振荡器电路

13.4 Timer1 的 16 位读 / 写模式

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

写 Timer1 的高字节也必须通过 TMR1H 缓冲寄存器进行。在写入 TMR1L 的同时, 使用 TMR1H 的内容更新 Timer1 的高字节。这样允许用户将所有 16 位一次写入 Timer1 的高字节和低字节。

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

13.5 Timer1 振荡器

片上晶振电路连接在 T1OSI (输入) 引脚和 T1OSO (放大器输出) 引脚之间。可以通过将 Timer1 振荡器使能位 T1OSCEN (T1CON<3>) 置 1 来使能该振荡器电路。该振荡器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 13-2 所示为典型的 LP 振荡器电路。表 13-2 给出了供 Timer1 振荡器选择的电容值。

用户必须提供软件延时来确保 Timer1 振荡器的正常起振。

表 13-2: 定时器振荡器的电容选择 (2,3,4,5)

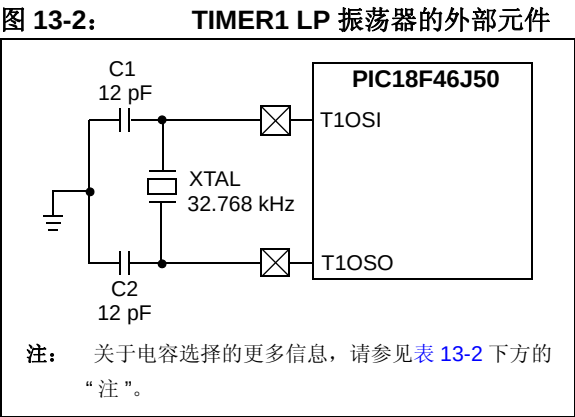
振荡器类型	频率	C1	C2
LP	32 kHz	12 pF ⁽¹⁾	12 pF ⁽¹⁾

- 注 1: Microchip 建议仅将这些值作为验证振荡器电路的起始点。
- 2: 电容越大, 振荡器越稳定, 但起振时间越长。
- 3: 因为每种谐振器 / 晶振都有其自身特性, 用户应当向谐振器 / 晶振制造厂商询问外部元件的适当值。
- 4: 上述电容值仅供设计参考。列出的值是 LPT1OSC = 1 时, 额定负载电容为 CL = 10 pF 的晶振的典型值。
- 5: 如果电容值不正确, 则可能导致频率不满足晶振制造商的容差规范。

Timer1 晶振驱动电平根据 LPT1OSC (CONFIG2L<4>) 配置位确定。高驱动电平模式 (LPT1OSC = 1) 用于驱动具有多种负载电容 (CL) 额定值的一系列广泛的 32.768 kHz 晶振。

低驱动电平模式则是针对超低功耗而高度优化。它不用于驱动任何类型的 32.768 kHz 晶振。在低驱动电平模式下, 如果 T1OSI 和 T1OSO 引脚上放置了极大的分立电容, 则晶振电路可能无法正确工作。该模式仅设计为在以下情况下工作: 每个引脚上的分立电容约为 3 pF-10 pF。

晶振制造商通常会为他们的晶振指定一个 CL (负载电容) 额定值。该值与图 13-2 中用于 C1 和 C2 的值相关, 但不一定相同。关于如何为给定晶振选择最佳 C1 和 C2 的更多详细信息, 请参见晶振制造商的应用信息。最佳值部分依赖于电路中的寄生电容, 该寄生电容通常是未知的。因此, 选择了电容值之后, 强烈建议您对振荡器执行全面的测试和验证。



PIC18F46J50 系列

13.5.1 使用 TIMER1 作为时钟源

在功耗管理模式下也可以将 Timer1 振荡器用作时钟源。通过将时钟选择位 $SCS<1:0>$ ($OSCCON<1:0>$) 设置为 01, 器件可以切换到 SEC_RUN 模式; CPU 和外设都可以用 Timer1 振荡器作为时钟源。如果 IDLEN 位 ($OSCCON<7>$) 被清零并且执行了 SLEEP 指令, 器件将进入 SEC_IDLE 模式。更多详细信息, 请参见第 4.0 节“低功耗模式”。

无论何时将 Timer1 振荡器用作时钟源, Timer1 系统时钟状态标志位 T1RUN ($TCLKCON<4>$) 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。如果使能了时钟监视器并且 Timer1 振荡器在提供时钟信号时发生了故障, 查询 T1RUN 位可以确定时钟源是 Timer1 振荡器还是其他时钟源。

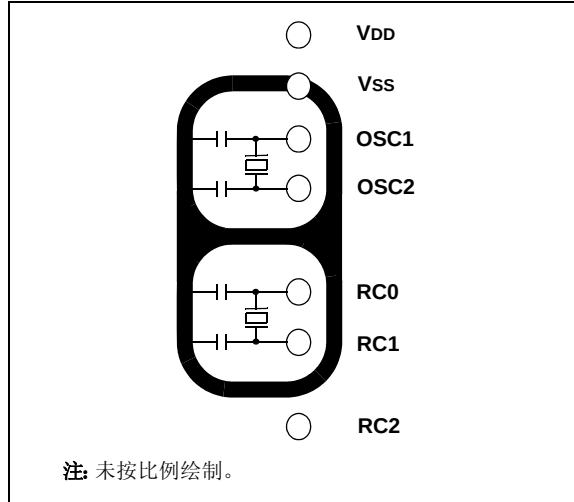
13.5.2 TIMER1 振荡器布线注意事项

Timer1 振荡器电路在工作期间仅消耗极小的电流。鉴于此振荡器的低功耗特性, 它对附近变化较快的信号比较敏感。在振荡器被配置为超低功耗模式 ($LPT1OSC = 0$) 时尤其如此。

如图 13-2 所示, 振荡器电路应该尽可能靠近单片机。除了 Vss 或 VDD 外, 在该振荡器电路边界内不应有其他电路通过。

对于单面 PCB, 如果必须要在该振荡器附近布高速电路 (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), 那么在该振荡器电路周围布接地保护环 (如图 13-3 所示), 或再加一个地平面可能会有帮助。

图 13-3: 带有接地保护环的振荡器电路



在低驱动电平模式下 ($LPT1OSC = 0$)，有一点非常关键，即使 RC2 I/O 引脚信号避开振荡器电路。如果将 RC2 配置为数字输出并不断翻转它的输出信号，则可能会影响振荡器电路，即使是对于相对较好的 PCB 布线设计。如果可能，建议将 RC2 保留为不用，或者将它用作输入引脚，但其信号源的斜率有一定限制。如果必须将 RC2 用作数字输出，则对于许多 PCB 布线设计，可能必须使用高驱动电平振荡器模式 ($LPT1OSC = 1$)。即使是在高驱动电平模式下，设计振荡器电路时，仍然应当谨慎布线。

除了要考虑 dV/dt 感应噪声之外，确保电路板洁净也非常重要。即使是极少量的导电焊剂残留物也会导致 PCB 泄漏电流，对振荡器电路产生很大的影响。

13.6 Timer1 中断

TMR1 寄存器对 (TMR1H:TMR1L) 从 0000h 递增到 FFFFh, 然后计满返回到 0000h 重新开始计数。如果允许了 Timer1 中断, 则溢出时会产生 Timer1 中断, 锁存到中断标志位 TMR1IF ($PIR1<0>$) 中。可以通过将 Timer1 中断允许位 TMR1IE ($PIE1<0>$) 置 1 或清零来允许或禁止该中断。

13.7 使用 ECCP 特殊事件触发信号复位 Timer1

如果 ECCP1 或 ECCP2 配置为使用 Timer1 并在比较模式下产生特殊事件触发信号（CCPxM<3:0> = 1011），该信号将复位 Timer1。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（更多信息，请参见第 18.3.4 节“特殊事件触发器”）。

要利用这一功能，必须将模块配置为定时器或同步计数器。在这种情况下，CCPRxH:CCPRxL 寄存器对实际上变成了 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信号同时发生，则写操作优先。

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

13.8 Timer1 门控

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

Timer1 门控也可由多个可选择源驱动。

13.8.1 TIMER1 门控计数使能

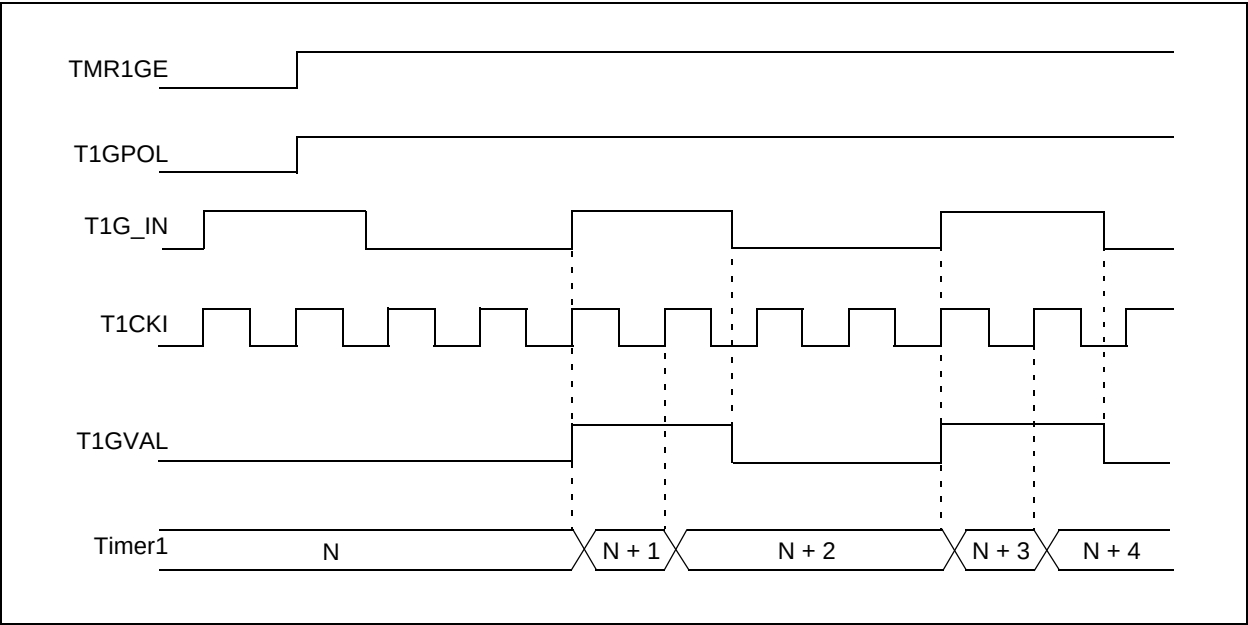
通过将 T1GCON 寄存器的 TMR1GE 位置 1 使能 Timer1 门控使能模式。使用 T1GCON 寄存器的 T1GPOL 位来配置 Timer1 门控使能模式的极性。

使能 Timer1 门控使能模式时，Timer1 将在 Timer1 时钟源的上升沿递增。禁止 Timer1 门控使能模式时，不会发生递增，Timer1 将保持当前计数。时序详细信息请参见图 13-4。

表 13-3: TIMER1 门控使能选择

T1CLK	T1GPOL	T1G	Timer1 工作
↑	0	0	计数
↑	0	1	保存计数
↑	1	0	保存计数
↑	1	1	计数

图 13-4: TIMER1 门控计数使能模式



PIC18F46J50 系列

13.8.2 TIMER1 门控源选择

Timer1 门控源可从四种不同源之中选择。源的选择由 T1GCON 寄存器的 T1GSSx 位控制。每个可用源的极性也是可选择的。极性的选择由 T1GCON 寄存器的 T1GPOL 位控制。

表 13-4: TIMER1 门控源

T1GSS<1:0>	Timer1 门控源
00	Timer1 门控引脚
01	Timer0 溢出 (TMR0 从 FFh 递增到 00h)
10	TMR2 匹配 PR2 (TMR2 递增以匹配 PR2)

13.8.2.1 T1G 引脚门控操作

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

13.8.2.2 Timer0 溢出门控操作

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

13.8.2.3 Timer2 匹配门控操作

TMR2 寄存器将递增到与 PR2 寄存器中的值匹配。在紧接着的下一个递增周期，TMR2 将复位为 00h。发生此复位时，将自动产生由低至高脉冲并在内部提供给 Timer1 门控电路。

该脉冲保持一个指令周期的高电平，然后恢复至低电平，直到下次匹配发生为止。

当 T1GPOL = 1 时，那么在 TMR2 与 PR2 匹配后，Timer1 在一个指令周期递增。

当 T1GPOL = 0 时，匹配发生后经过一个周期，Timer1 开始递增。

13.8.3 TIMER1 门控翻转模式

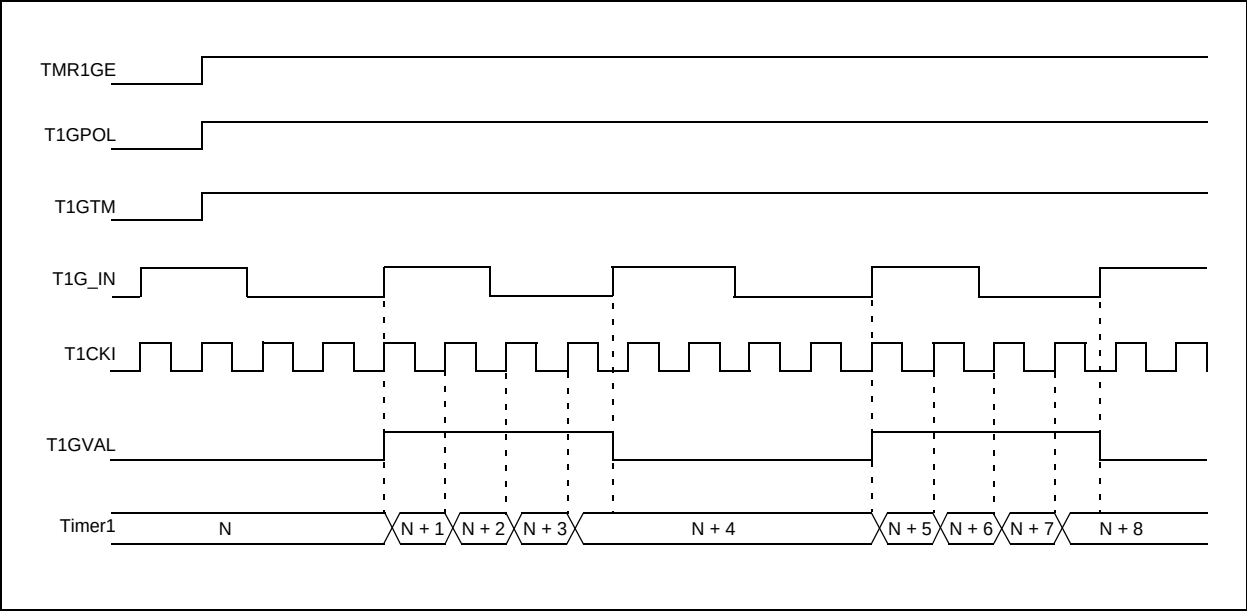
使能 Timer1 门控翻转模式时，可测量 Timer1 门控信号的全周期长度，而不是单个电平脉冲持续时间。

Timer1 门控源经由在信号的每个递增边沿改变状态的触发器进行路由。时序详细信息请参见图 13-5。

T1GVAL 位将指示翻转模式何时工作以及定时器何时计数。

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

图 13-5: TIMER1 门控翻转模式



13.8.4 TIMER1 门控单脉冲模式

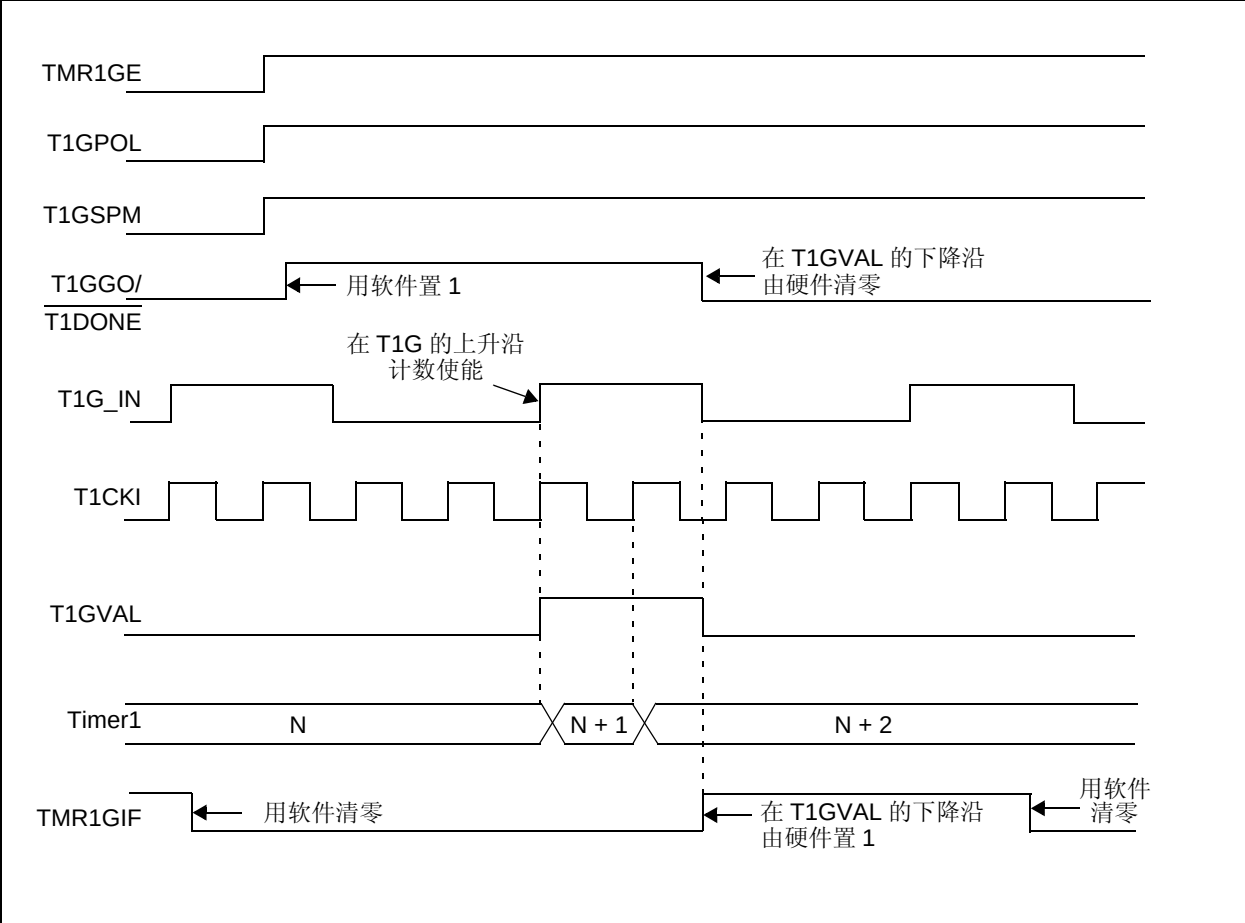
使能 Timer1 门控单脉冲模式时，可能会捕捉到一个单脉冲门控事件。首先通过将 T1GCON 寄存器中的 T1GSPM 位置 1 使能 Timer1 门控单脉冲模式。接下来必须将 T1GCON 寄存器中的 T1GGO/T1DONE 位置 1。Timer1 将在下一个递增边沿完全使能。在脉冲的下一个后边沿，将自动清零 T1GGO/T1DONE 位。不允许其他门控事件递增 Timer1，直到 T1GGO/T1DONE 位再次用软件置 1。

清零 T1GCON 寄存器的 T1GSPM 位也会清零 T1GGO/T1DONE 位。时序详细信息请参见图 13-6。同时使能翻转模式和单脉冲模式将允许两部分协同工作。这样就可以测量 Timer1 门控源的周期时间。时序详细信息请参见图 13-7。

13.8.5 TIMER1 门控值状态

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

图 13-6: TIMER1 门控单脉冲模式



PIC18F46J50 系列

图 13-7: TIMER1 门控单脉冲和翻转组合模式

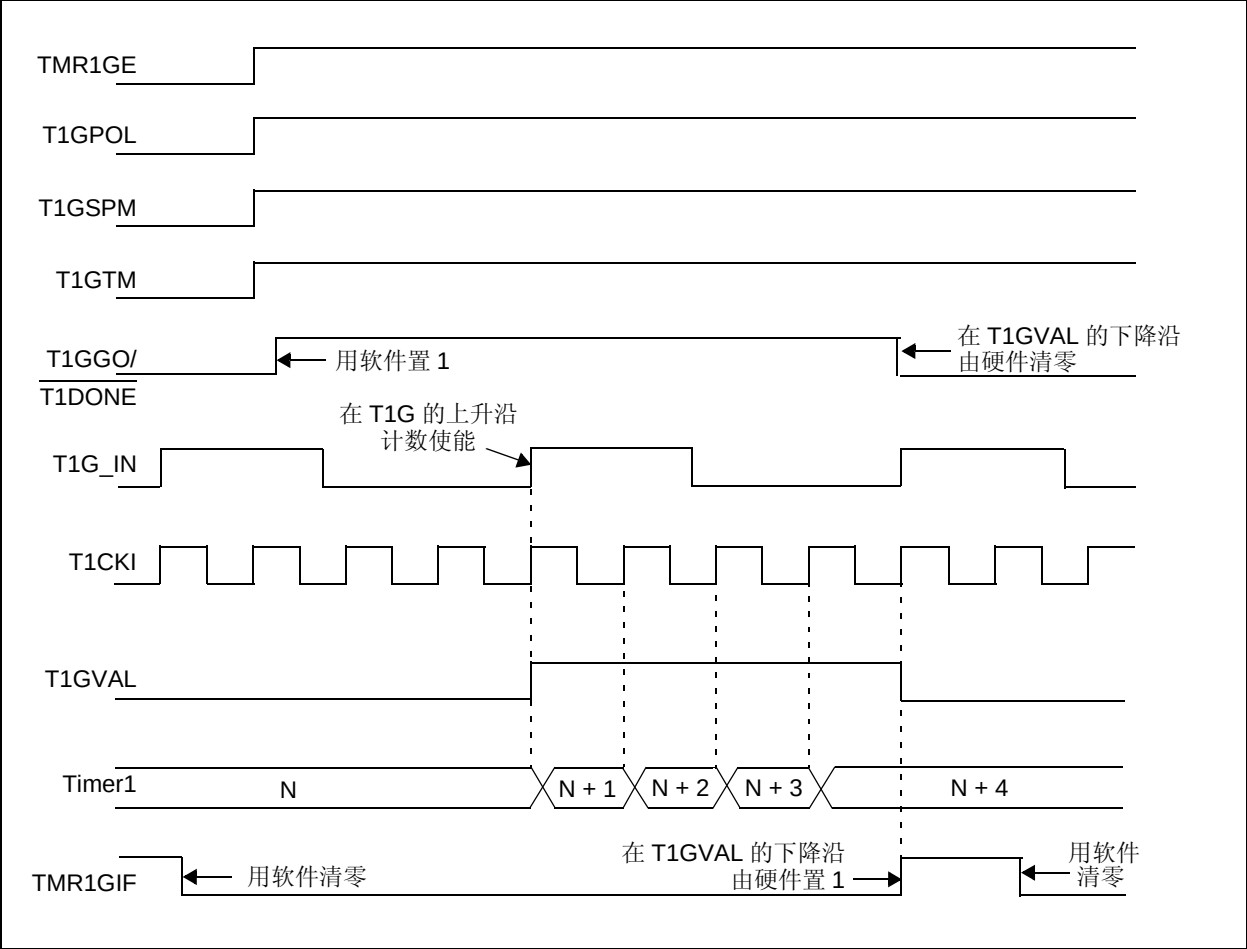


表 13-5: 与 TIMER1 作为定时器 / 计数器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	89
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	91
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	91
IPR1	PMPiP ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	91
TMR1L	Timer1 低字节寄存器								90
TMR1H	Timer1 高字节寄存器								90
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	RD16	TMR1ON	90
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ T1DONE	T1GVAL	T1GSS1	T1GSS0	91
TCLKCON	—	—	—	T1RUN	—	—	T3CCP2	T3CCP1	93

图注: Timer1 模块不使用阴影单元。

注 1: 这些位仅在 44 引脚器件上可用。

14.0 TIMER2 模块

Timer2 模块具有以下特性：

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

图 14-1 给出了此模块的简化框图。

14.1 Timer2 工作原理

在正常工作模式下，TMR2 从 00h 开始，每个时钟周期 (Fosc/4) 递增 1。4 位计数器 / 预分频器提供了对时钟输入不分频、4 分频和 16 分频三种预分频选项，可通过预分频比控制位 T2CKPS<1:0> (T2CON<1:0>) 进行选择。在每个时钟周期，TMR2 的值都会与周期寄存器 PR2 中的值进行比较。当两个值匹配时，由比较器产生匹配信号作为定时器的输出。此信号也会将 TMR2 的值在下一个周期复位为 00h，并驱动输出计数器 / 后分频器（见第 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。预分频器和后分频器计数器均会在发生以下事件时清零：

- 对 TMR2 寄存器进行写操作
 - 对 T2CON 寄存器进行写操作
 - 任何器件复位（上电复位 (POR)、MCLR 复位、看门狗定时器复位 (WDTR) 或欠压复位 (BOR)）
- 写 T2CON 时 TMR2 不会清零。

寄存器 14-1: T2CON: TIMER2 控制寄存器（位于快速操作存储区，地址 FCAh）

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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	未实现：读为 0
bit 6-3	T2OUTPS<3:0>: Timer2 输出后分频比选择位 0000 = 1:1 后分频比 0001 = 1:2 后分频比 • • • 1111 = 1:16 后分频比
bit 2	TMR2ON: Timer2 使能位 1 = 使能 Timer2 0 = 关闭 Timer2
bit 1-0	T2CKPS<1:0>: Timer2 时钟预分频比选择位 00 = 预分频比为 1 01 = 预分频比为 4 1x = 预分频比为 16

PIC18F46J50 系列

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。此计数器产生的 TMR2 匹配中断标志位为 TMR2IF（PIR1<1>）。可以通过将 TMR2 匹配中断允许位 TMR2IE（PIE1<1>）置 1 来允许此中断。

可以通过后分频比控制位 T2OUTPS<3:0>（T2CON<6:3>）在 16 个后分频比选项（从 1:1 到 1:16）中选择其一。

14.3 Timer2 输出

TMR2 的不经分频的输出主要用于 ECCP 模块，它用作 ECCP 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“主同步串行口（MSSP）模块”](#)中提供了更多信息。

图 14-1: TIMER2 框图

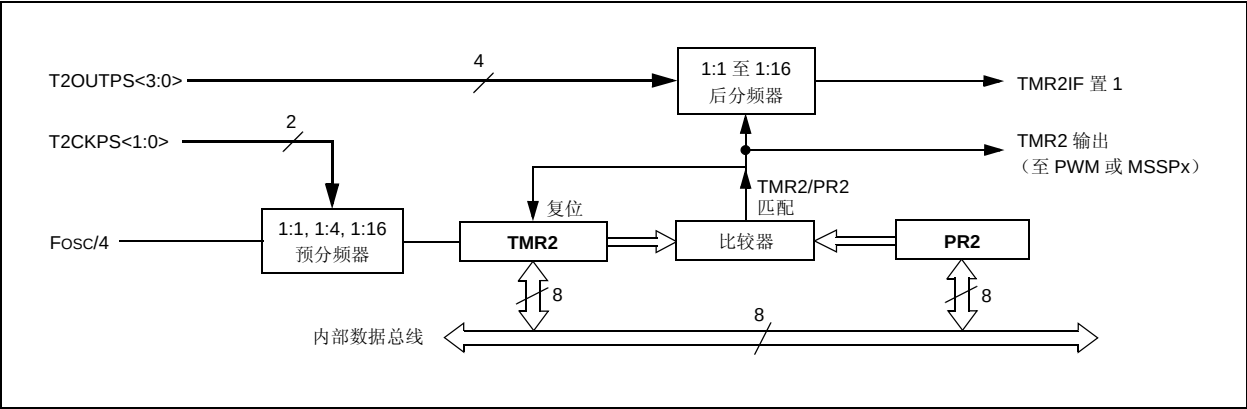


表 14-1: 与 TIMER2 作为定时器 / 计数器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	89
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	91
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	91
IPR1	PMPPI ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	91
TMR2	Timer2 寄存器								90
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	90
PR2	Timer2 周期寄存器								90

图注： — = 未实现，读为 0。Timer2 模块不使用阴影单元。

注 1： 这些位仅在 44 引脚器件上可用。

15.0 TIMER3 模块

Timer3 模块定时器 / 计数器具有以下特性：

- 可由软件选择作为 16 位定时器或计数器
- 可读写的 8 位寄存器（TMR3H 和 TMR3L）
- 可选择器件时钟或 Timer1 内部振荡器作为时钟源（内部或外部）
- 溢出时产生中断
- ECCP 特殊事件触发模块复位

图 15-1 给出了 Timer3 模块的简化框图。

Timer3 模块是通过 T3CON 寄存器（寄存器 15-1）来控制的。它还可以为 ECCP 模块选择时钟源；更多信息，请参见第 18.1.1 节“ECCP 模块和定时器资源”。

Fosc 时钟源（TMR3CS<1:0> = 01）不应与 ECCP 捕捉 / 比较功能一起使用。如果定时器要与捕捉或比较功能一起使用，总是选择其他定时器时钟源选项之一。

寄存器 15-1: T3CON: TIMER3 控制寄存器（位于快速操作存储区，地址 F79h）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	T3SYNC	RD16	TMR3ON
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7-6 **TMR3CS<1:0>**: Timer3 时钟源选择位
 10 = Timer3 时钟源是 Timer1 振荡器或 T3CKI 数字输入引脚（在 PPS 模块中指定）
 01 = Timer3 时钟源是系统时钟（Fosc）⁽¹⁾
 00 = Timer3 时钟源是指令时钟（Fosc/4）
- bit 5-4 **T3CKPS<1:0>**: Timer3 输入时钟预分频比选择位
 11 = 1:8 预分频比
 10 = 1:4 预分频比
 01 = 1:2 预分频比
 00 = 1:1 预分频比
- bit 3 **T3OSCEN**: Timer3 振荡器源选择位
当 TMR3CS<1:0> = 10 时:
 1 = 给 Timer1 晶振驱动器上电，并依靠晶振的输出提供 Timer3 时钟
 0 = Timer1 晶振驱动器断电，Timer3 时钟来自 PPS 模块中指定的 T3CKI 数字输入引脚⁽²⁾
当 TMR3CS<1:0> = 0x 时:
 1 = 给 Timer1 晶振驱动器上电（T1OSC）
 0 = Timer1 晶振驱动器断电⁽²⁾
- bit 2 **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<1:0> = 10 时:
 1 = 不同步外部时钟输入
 0 = 同步外部时钟输入
当 TMR3CS<1:0> = 0x 时:
 该位为无关位；Timer3 使用内部时钟。
- bit 1 **RD16**: 16 位读 / 写模式使能位
 1 = 使能 Timer3 通过一次 16 位操作进行寄存器读 / 写
 0 = 使能 Timer3 通过两次 8 位操作进行寄存器读 / 写
- bit 0 **TMR3ON**: Timer3 使能位
 1 = 使能 Timer3
 0 = 停止 Timer3

注 1: 如果定时器要与 ECCP 捕捉 / 比较功能一起使用，则不应选择 Fosc 时钟源。

注 2: 当 T1OSCEN（T1CON）或 T3OSCEN（T3CON）= 1 时，Timer1 振荡器晶振驱动器上电。这两个位的逻辑或决定是否使能驱动器电路。驱动器电路禁止时，反相器和反馈电阻被禁止以降低功耗。要使晶振驱动器上电，并不必使能 TMR1ON 和 TMR3ON 位。

PIC18F46J50 系列

15.1 Timer3 门控控制寄存器

Timer3 门控控制寄存器（T3GCON）如寄存器 15-2 所示，用于控制 Timer3 门控。

寄存器 15-2: T3GCON: TIMER3 门控控制寄存器（位于快速操作存储区，地址 F97h）⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/T3DONE	T3GVAL	T3GSS1	T3GSS0
bit 7							bit 0

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

bit 7	TMR3GE: Timer3 门控使能位 如果 TMR3ON = 0: 该位为无关位。 如果 TMR3ON = 1: 1 = Timer3 计数由 Timer3 门控功能控制 0 = Timer3 计数与 Timer3 门控功能无关
bit 6	T3GPOL: Timer3 门控极性位 1 = Timer3 门控高电平有效（当门控为高电平时 Timer3 计数） 0 = Timer3 门控低电平有效（当门控为低电平时 Timer3 计数）
bit 5	T3GTM: Timer3 门控翻转模式位 1 = 使能 Timer3 门控翻转模式 0 = 禁止 Timer3 门控翻转模式并清除翻转触发器 Timer3 门控触发器在每个上升沿翻转。
bit 4	T3GSPM: Timer3 门控单脉冲模式位 1 = 使能 Timer3 门控单脉冲模式，控制 Timer3 门控 0 = 禁止 Timer3 门控单脉冲模式
bit 3	T3GGO/T3DONE: Timer3 门控单脉冲采集状态位 1 = Timer3 门控单脉冲采集就绪，正在等待一个边沿 0 = Timer3 门控单脉冲采集已经结束或尚未开始 当 T3GSPM 清零时，该位会自动清零。
bit 2	T3GVAL: Timer3 门控当前状态位 表示可提供给 TMR3H:TMR3L 的 Timer3 门控的当前状态。不受 Timer3 门控使能位（TMR3GE）的影响。
bit 1-0	T3GSS<1:0>: Timer3 门控源选择位 10 = TMR2 匹配 PR2 输出 01 = Timer0 溢出输出 00 = Timer3 门控引脚（T3G）

注 1: 建议在设定 T3CON 之前先设定 T3GCON。

寄存器 15-3: **TCLKCON**: 定时器时钟控制寄存器 （位于快速操作存储区，地址 F52h）

U-0	U-0	U-0	R-0	U-0	U-0	R/W-0	R/W-0
—	—	—	T1RUN	—	—	T3CCP2	T3CCP1
bit 7						bit 0	

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

- bit 7-5 未实现：读为 0
- bit 4 **T1RUN**: Timer1 运行状态位
 1 = 目前器件时钟来自 T1OSC/T1CKI
 0 = 系统时钟来自除 T1OSC/T1CKI 外的振荡器
- bit 3-2 未实现：读为 0
- bit 1-0 **T3CCP<2:1>**: ECCP 定时器分配位
 10 = ECCP1 和 ECCP2 均使用 Timer3 （捕捉 / 比较）和 Timer4 （PWM）
 01 = ECCP1 使用 Timer1（比较 / 捕捉）和 Timer2（PWM）； ECCP2 使用 Timer3（捕捉 / 比较）和 Timer4（PWM）
 00 = ECCP1 和 ECCP2 均使用 Timer1 （捕捉 / 比较）和 Timer2 （PWM）

PIC18F46J50 系列

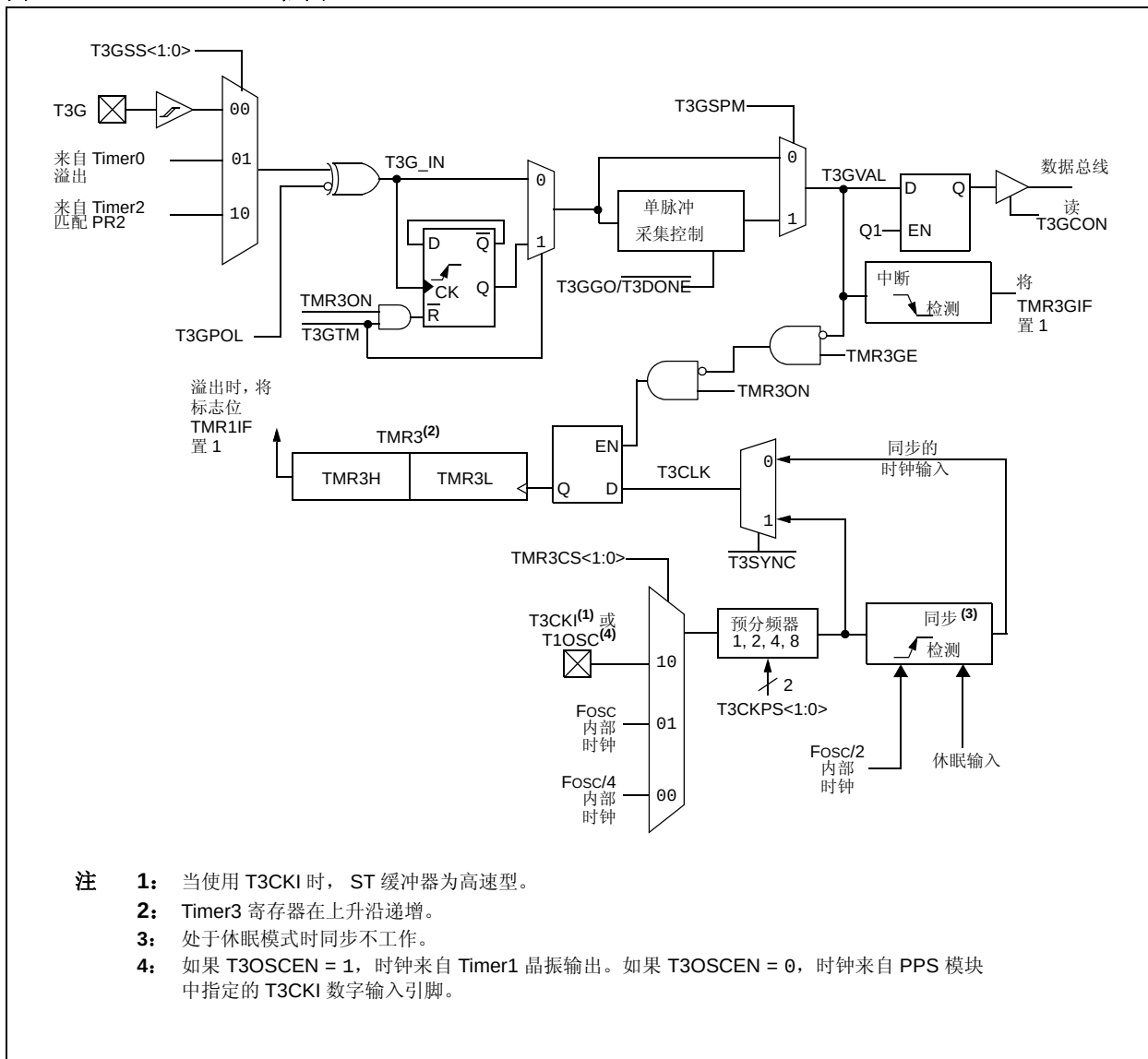
15.2 Timer3 工作原理

Timer3 可工作在以下三种模式之一：

- 定时器
- 同步计数器
- 异步计数器
- 带门控的定时器

工作模式由时钟选择位 TMR3CSx (T3CON<7:6>) 决定。当 TMR3CSx 位清零 (= 00) 时, Timer3 在每个内部指令周期 ($F_{osc}/4$) 递增。当 TMR3CSx = 01 时, Timer3 时钟源为系统时钟 (F_{osc})；当 TMR3CSx = 10 时, Timer3 用作计数器, 采用来自 T3CKI 引脚 (在第一个下降沿之后的上升沿) 或 Timer1 振荡器的外部时钟。

图 15-1: TIMER3 框图



15.3 Timer3 的 16 位读 / 写模式

可将Timer3配置为16位读写模式（见第15.3节“Timer3 的 16 位读 / 写模式”）。当RD16 控制位（T3CON<1>）置 1 时，TMR3H 的地址被映射到 Timer3 的高字节缓冲寄存器。读 TMR3L 将把 Timer3 的高字节的内容装入 Timer3 高字节缓冲寄存器。这种方式使用户可以精确地读取 Timer3 的全部 16 位，而不需要像先读高字节再读低字节那样，由于两次读取之间可能存在计满返回，而不得不验证读取的有效性。

写 Timer3 的高字节也必须通过 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 16 位一次写入 Timer3 的高字节和低字节。

在该模式下不能直接读写 Timer3 的高字节。所有读写都必须通过 Timer3 高字节缓冲寄存器来进行。

写入 TMR3H 不会清零 Timer3 预分频器。只有在写 TMR3L 时才会清零该预分频器。

15.4 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 振荡器进行了说明。

15.5 Timer3 门控

Timer3 可配置为自由计数或用 Timer3 门控电路使能和禁止计数。这也称为 Timer3 门控计数使能。

Timer3 门控也可由多个可选择源驱动。

15.5.1 TIMER3 门控计数使能

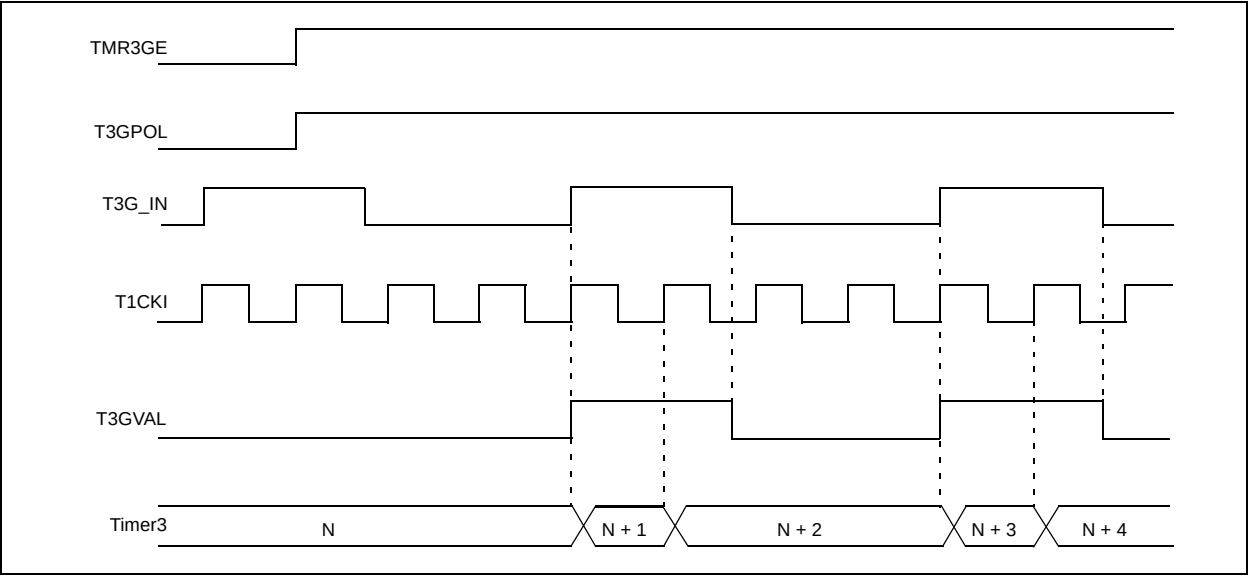
通过将 T3GCON 寄存器的 TMR3GE 位置 1 使能 Timer3 门控使能模式。使用 T3GCON 寄存器的 T3GPOL 位来配置 Timer3 门控使能模式的极性。

使能 Timer3 门控使能模式时，Timer3 将在 Timer3 时钟源的上升沿递增。禁止 Timer3 门控使能模式时，不会发生递增，Timer3 将保持当前计数。时序详细信息请参见图 15-2。

表 15-1: TIMER3 门控使能选择

T3CLK	T3GPOL	T3G	Timer3 工作
↑	0	0	计数
↑	0	1	保存计数
↑	1	0	保存计数
↑	1	1	计数

图 15-2: TIMER3 门控计数使能模式



15.5.2 TIMER3 门控源选择

Timer3 门控源可从四种不同源之中选择。源的选择由 T3GCON 寄存器的 T3GSSx 位控制。每个可用源的极性也是可选择的。极性的选择由 T3GCON 寄存器的 T3GPOL 位控制。

表 15-2: TIMER3 门控源

T3GSS<1:0>	Timer3 门控源
00	Timer3 门控引脚
01	Timer0 溢出 (TMR0 从 FFh 递增到 00h)
10	TMR2 匹配 PR2 (TMR2 递增以匹配 PR2)
11	保留

15.5.2.1 T3G 引脚门控操作

T3G 引脚是 Timer3 门控源之一。它可用于向 Timer3 门控电路提供外部源。

15.5.2.2 Timer0 溢出门控操作

Timer0 从 FFh 递增到 00h 时，将自动产生由低至高脉冲并在内部提供给 Timer3 门控电路。

15.5.2.3 Timer2 匹配门控操作

TMR2 寄存器将递增到与 PR2 寄存器中的值匹配。在紧接着的下一个递增周期，TMR2 将复位为 00h。发生复位时，将自动产生由低至高脉冲并在内部提供给 Timer3 门控电路。

15.5.3 TIMER3 门控翻转模式

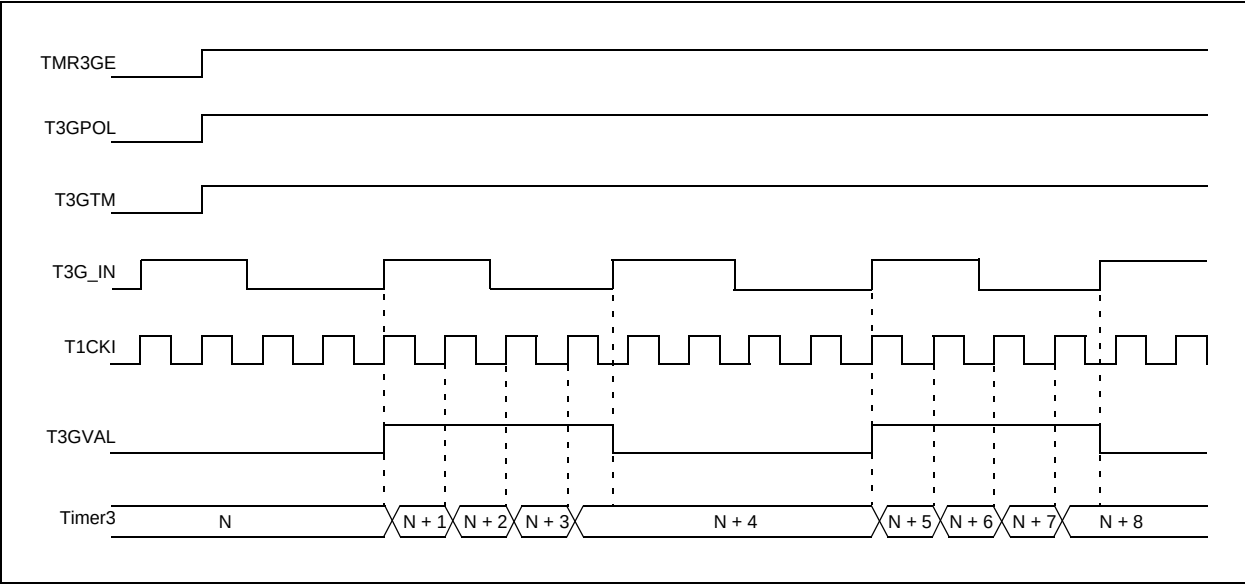
使能 Timer3 门控翻转模式时，可测量 Timer3 门控信号的全周期长度，而不是单个电平脉冲持续时间。

Timer3 门控源经由在信号的每个递增边沿改变状态的触发器进行路由。时序详细信息请参见图 15-3。

T3GVAL 位将指示翻转模式何时工作以及定时器何时计数。

Timer3 门控翻转模式通过将 T3GCON 寄存器的 T3GTM 位置 1 使能。T3GTM 位清零时，将清除触发器并保持清零。这对于控制测量哪个边沿是必需的。

图 15-3: TIMER3 门控翻转模式



15.5.4 TIMER3 门控单脉冲模式

使能 Timer3 门控单脉冲模式时，可能会捕捉到一个单脉冲门控事件。首先通过将 T3GCON 寄存器中的 T3GSPM 位置 1 使能 Timer3 门控单脉冲模式。接下来必须将 T3GCON 寄存器中的 T3GGO/T3DONE 位置 1。

Timer3 将在下一个递增边沿完全使能。在脉冲的下一个后边沿，将自动清零 T3GGO/T3DONE 位。不允许其他

门控事件递增 Timer3，直到 T3GGO/T3DONE 位再次用软件置 1。

清零 T3GCON 寄存器的 T3GSPM 位也会清零 T3GGO/T3DONE 位。时序详细信息请参见图 15-4。

同时使能翻转模式和单脉冲模式将允许两部分协同工作。这样就可以测量 Timer3 门控源的周期时间。时序详细信息请参见图 15-5。

图 15-4: TIMER3 门控单脉冲模式

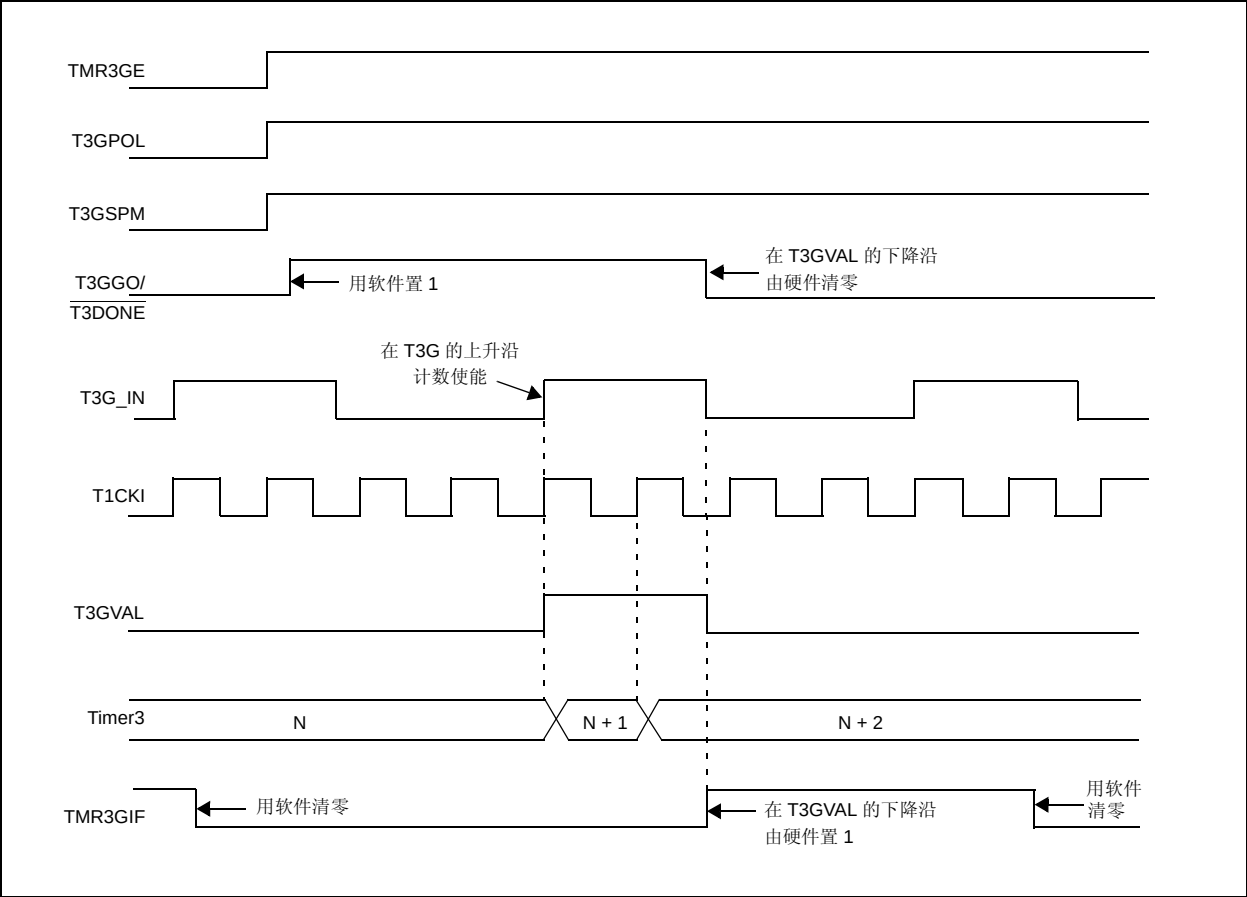
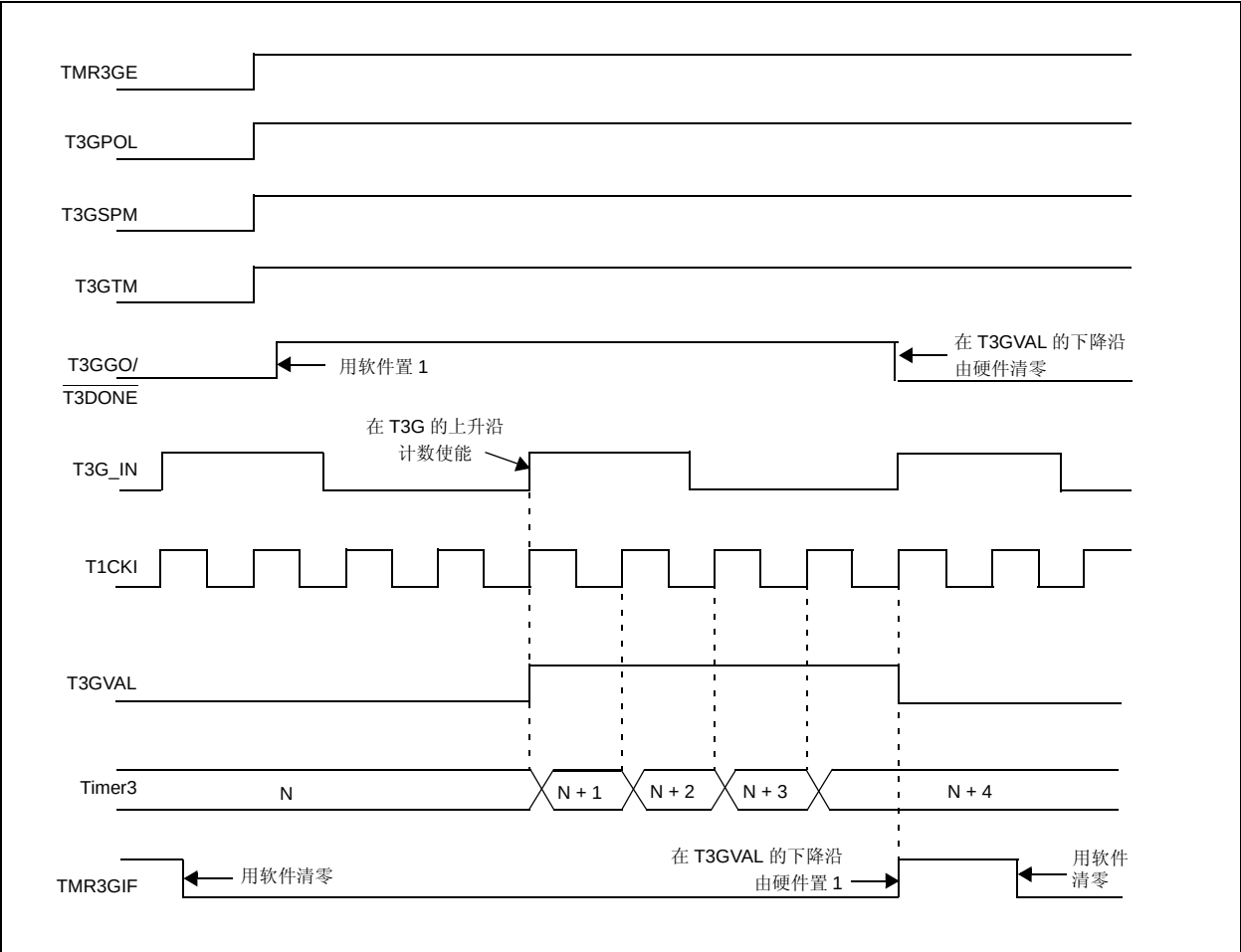


图 15-5: TIMER3 门控单脉冲和翻转组合模式



15.5.5 TIMER3 门控值状态

使用 Timer3 门控值状态时，可读取门控控制值的最新电平。该值保存在 T3GCON 寄存器的 T3GVAL 位中。即使 Timer3 门控未使能（TMR3GE 位清零），T3GVAL 位也是有效的。

15.5.6 TIMER3 门控事件中断

允许 Timer3 门控事件中断时，可在门控事件完成时产生一个中断。出现 T3GVAL 的下降沿时，PIR3 寄存器中的 TMR3GIF 标志位将置 1。如果 PIE3 寄存器中的 TMR3GIE 位置 1，则会识别出这个中断。

即使 Timer3 门控未使能（TMR3GE 位清零），TMR3GIF 标志位也能工作。

15.6 Timer3 中断

TMR3 寄存器对（TMR3H:TMR3L）从 0000h 递增到 FFFFh，然后计满返回到 0000h 重新开始计数。如果允许了 Timer3 中断，则溢出时会产生 Timer3 中断，锁存到中断标志位 TMR3IF（PIR2<1>）中。可以通过将 Timer3 中断允许位 TMR3IE（PIE2<1>）置 1 或清零来允许或禁止该中断。

15.7 使用 ECCP 特殊事件触发信号复位 Timer3

如果 ECCP1 或 ECCP2 配置为使用 Timer3 并在比较模式下产生特殊事件触发信号（CCPxM<3:0> = 1011），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（更多信息，请参见第 18.3.4 节“特殊事件触发器”）。

要使用这一功能，必须将模块配置为定时器或同步计数器。在这种情况下，CCPRxH:CCPRxL 寄存器对实际上变成了 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 Timer3 的写操作和来自 ECCP 模块的特殊事件触发同时发生，则写操作优先。

注： ECCPx 模块产生的特殊事件触发信号不会将 TMR3IF 中断标志位（PIR3<0>）置 1。

表 15-3: 与 TIMER3 作为定时器 / 计数器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	89
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	91
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	91
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	91
TMR3L	Timer3 低字节寄存器								92
TMR3H	Timer3 高字节寄存器								92
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\overline{C}$	RD16	TMR1ON	90
T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	T3SYN $\overline{C}$	RD16	TMR3ON	92
T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ T3DONE	T3GVAL	T3GSS1	T3GSS0	92
TCLKCON	—	—	—	T1RUN	—	—	T3CCP2	T3CCP1	93
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	91
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	91
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	91

图注： — = 未实现，读为 0。Timer3 模块不使用阴影单元。

PIC18F46J50 系列

注:

16.0 TIMER4 模块

Timer4 定时器模块具有以下特性：

- 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 到 1:16）
- TMR4 与 PR4 匹配时产生中断

Timer4 有一个控制寄存器，如寄存器 16-1 所示。可以通过清零控制位 TMR4ON（T4CON<2>）关闭 Timer4，以实现功耗最小。此寄存器还控制对 Timer4 的预分频比和后分频比的选择。图 16-1 所示为 Timer4 模块的简化框图。

16.1 Timer4 工作原理

Timer4 可以用作 ECCP 模块在 PWM 模式下的 PWM 时基。TMR4 寄存器是可读写的，在任何器件复位时都会被清零。输入时钟（Fosc/4）有由控制位 T4CKPS<1:0>（T4CON<1:0>）选择的预分频选项（1:1、1:4 或 1:16）。TMR4 的匹配输出通过一个 4 位后分频器（提供 1:1 到 1:16 的分频比）产生 TMR4 中断，锁存到标志位 TMR4IF（PIR3<3>）中。

预分频器和后分频器计数器均会在发生以下任意事件时清零：

- 对 TMR4 寄存器进行写操作
 - 对 T4CON 寄存器进行写操作
 - 任何器件复位（上电复位（POR）、MCLR 复位、看门狗定时器复位（WDTR）或欠压复位（BOR））
- 写 T4CON 时 TMR4 不会清零。

寄存器 16-1: T4CON: TIMER4 控制寄存器（位于快速操作存储区，地址 F76h）

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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	未实现：读为 0
bit 6-3	T4OUTPS<3:0>: 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 = 使能 Timer4 0 = 关闭 Timer4
bit 1-0	T4CKPS<1:0>: 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

PIC18F46J50 系列

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，然后在下一个递增周期复位为 00h。在复位时，PR4 寄存器初始化为 FFh。

16.3 TMR4 的输出

TMR4 的输出（在通过后分频器之前）只能用作 ECCP 模块的 PWM 时基。它不能像 Timer2 输出一样用作 MSSP 模块的波特率时钟。

图 16-1: TIMER4 框图

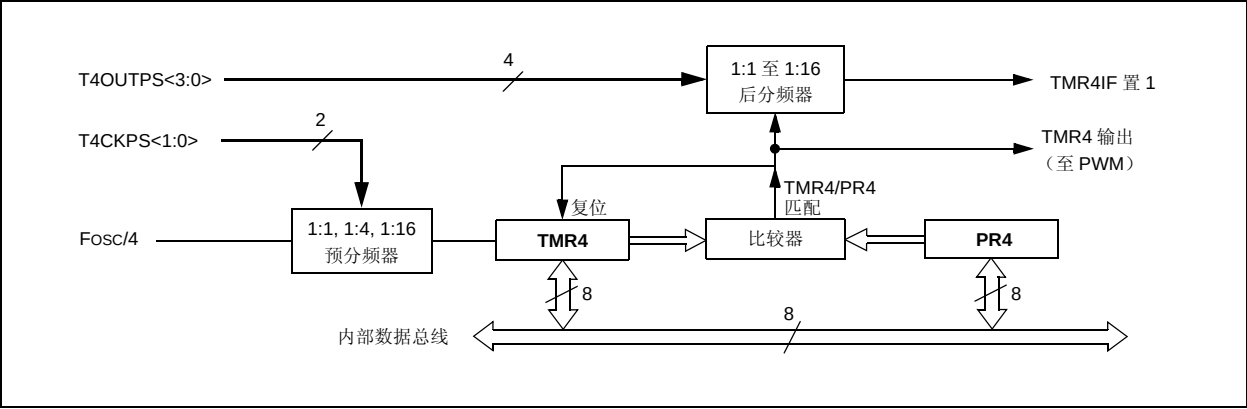


表 16-1: 与 TIMER4 作为定时器 / 计数器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	89
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	91
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	91
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	91
TMR4	Timer4 寄存器								92
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	92
PR4	Timer4 周期寄存器								92

图注： — = 未实现，读为 0。Timer4 模块不使用阴影单元。

17.0 实时时钟和日历 (RTCC)

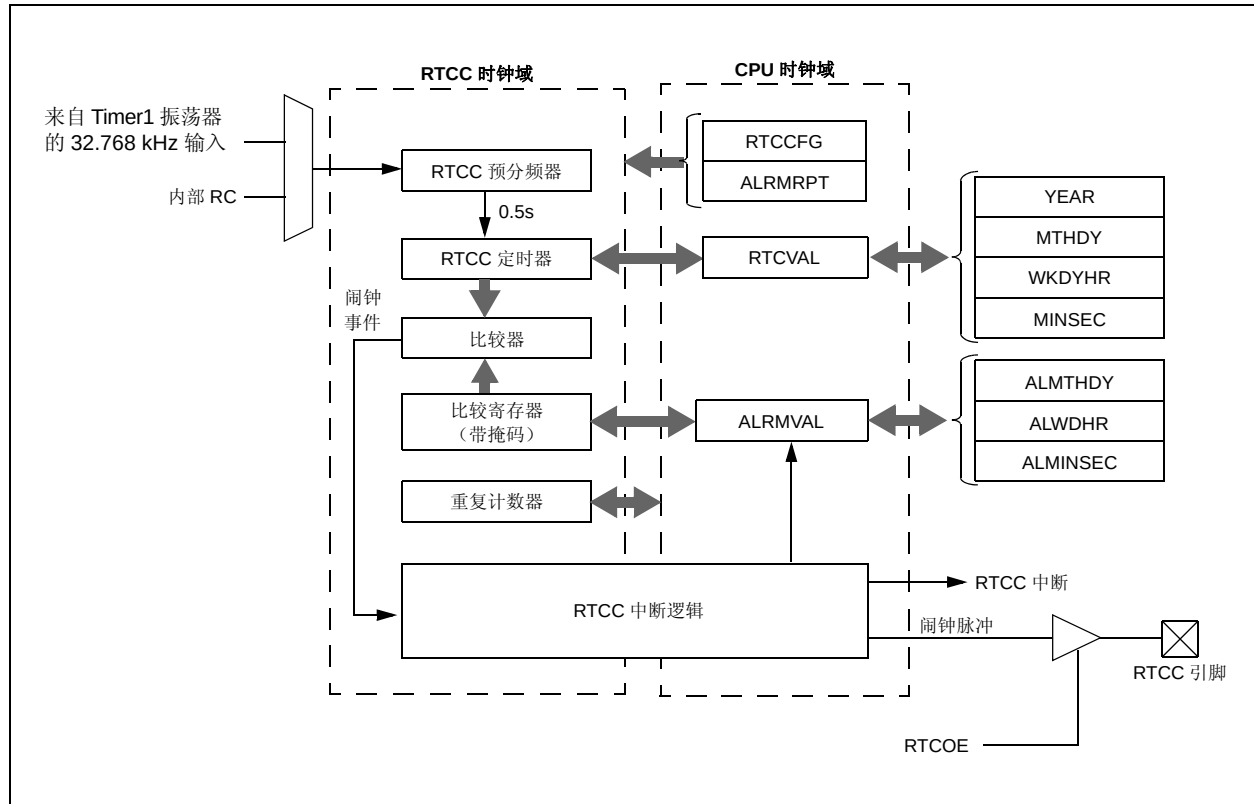
实时时钟和日历 (RTCC) 模块的主要特性有:

- 时间: 小时、分钟和秒钟
- 24 小时格式 (军用时间)
- 日历: 星期、日期、月和年
- 闹钟可配置
- 年份范围: 2000 至 2099
- 闰年修正
- 用于获得紧凑固体的 BCD 格式
- 为低功耗工作进行了优化
- 带自动调节的用户校准
- 校准范围: 每月 ± 2.64 秒误差
- 要求: 外部 32.768 kHz 时钟晶振
- RTCC 引脚上的闹钟脉冲或秒时钟输出

RTCC 模块是为需要长时间维持精确时间的应用设计的, 无需或很少需要 CPU 干预。该模块为低功耗使用进行了优化, 以便在跟踪时间时延长电池寿命。

该模块是百年时钟和日历, 能自动检测闰年。时钟范围为从 2000 年 1 月 1 日 00:00:00 (午夜) 到 2099 年 12 月 31 日 23:59:59。小时数采用 24 小时 (军用时间) 格式计量。该时钟提供一秒的时间粒度, 用户可看到半秒的时间间隔。

图 17-1: RTCC 框图



17.1 RTCC 模块寄存器

RTCC 模块寄存器可分为以下三类：

RTCC 控制寄存器

- RTCCFG
- RTCCAL
- PADCFG1
- ALRMCFG
- ALRMRPT

RTCC 值寄存器

- RTCVALH 和 RTCVALL——可访问以下寄存器：
 - YEAR
 - MONTH
 - DAY
 - WEEKDAY
 - HOUR
 - MINUTE
 - SECOND

闹钟值寄存器

- ALRMVALH和ALRMVALL——可访问以下寄存器：
 - ALRMMNTH
 - ALRMDAY
 - ALRMWD
 - ALRMHR
 - ALRMMIN
 - ALRMSEC

注： RTCVALH 和 RTCVALL 寄存器可通过 RTCRPT<1:0> 进行访问。ALRMVALH 和 ALRMVALL 可通过 ALRMPTR<1:0> 进行访问。

17.1.1 RTCC 控制寄存器

寄存器 17-1: **RTCCFG: RTCC 配置寄存器** (位于普通存储区, 地址 F3Fh) ⁽¹⁾

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN ⁽²⁾	—	RTCWREN	RTCSYNC	HALFSEC ⁽³⁾	RTCOE	RTCPTR1	RTCPTR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **RTCEN: RTCC 使能位** ⁽²⁾

1 = 使能 RTCC 模块

0 = 禁止 RTCC 模块

bit 6 **未实现:** 读为 0

bit 5 **RTCWREN: RTCC 值寄存器写使能位**

1 = RTCVALH 和 RTCVALL 寄存器可由用户写入

0 = RTCVALH 和 RTCVALL 寄存器已锁定, 不可由用户写入

bit 4 **RTCSYNC: RTCC 值寄存器读同步位**

1 = 由于计满返回, RTCVALH、RTCVALL 和 ALCFGRPT 寄存器在读操作过程中可能改变, 从而导致读取的数据无效

如果两次读取寄存器得到的数据相同, 可认为数据是有效的。

0 = RTCVALH、RTCVALL 或 ALCFGRPT 寄存器在读取时无需考虑计满返回

bit 3 **HALFSEC: 半秒状态位** ⁽³⁾

1 = 一秒的后半秒

0 = 一秒的前半秒

bit 2 **RTCOE: RTCC 输出使能位**

1 = 使能 RTCC 时钟输出

0 = 禁止 RTCC 时钟输出

bit 1-0 **RTCPTR<1:0>: RTCC 值寄存器窗口指针位**

读取 RTCVALH 和 RTCVALL 寄存器时, 指向相应的 RTCC 值寄存器; 每当读或写 RTCVALH 时 RTCPTR<1:0> 的值就递减 1, 直到达到 00。

RTCVAL<15:8>:

00 = 分钟数

01 = 星期

10 = 月

11 = 保留

RTCVAL<7:0>:

00 = 秒数

01 = 小时数

10 = 日

11 = 年

注 1: RTCCFG 寄存器只受 POR 的影响。

2: 仅当 RTCWREN = 1 时才允许写入 RTCEN 位。

3: 该位是只读的。写入 MINSEC 寄存器的低半部分时, 它被清零。

PIC18F46J50 系列

寄存器 17-2: RTCCAL: RTCC 校准寄存器 (位于普通存储区, 地址 F3Eh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7							bit 0

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

bit 7-0 **CAL<7:0>: RTC 漂移校准位**
01111111 = 最大正向调整; 每分钟增加 508 个 RTC 时钟脉冲
.
.
.
00000001 = 最小正向调整; 每分钟增加 4 个 RTC 时钟脉冲
00000000 = 无调整
11111111 = 最小负向调整; 每分钟减少 4 个 RTC 时钟脉冲
.
.
.
10000000 = 最大负向调整; 每分钟减少 512 个 RTC 时钟脉冲

寄存器 17-3: PADCFG1: 焊盘配置寄存器 (位于普通存储区, 地址 F3Ch)

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	RTSECSEL1 ⁽¹⁾	RTSECSEL0 ⁽¹⁾	PMPTTL
bit 7							bit 0

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

bit 7-3 **未实现: 读为 0**

bit 2-1 **RTSECSEL<1:0>: RTCC 秒时钟输出选择位 ⁽¹⁾**
11 = 保留。不要使用
10 = 选择 RTCC 引脚输出 RTCC 源时钟 (引脚可以是 INTRC 或 T1OSC, 取决于 RTCOSC (CONFIG3L<1>) 设置)
01 = 选择 RTCC 引脚输出 RTCC 秒时钟
00 = 选择 RTCC 引脚输出 RTCC 闹钟脉冲

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 模块使用施密特触发器输入缓冲器

注 1: 要能使实际 RTCC 输出, RTCOE (RTCCFG<2>) 位必须置 1。

寄存器 17-4: ALRMCFG: 闹钟配置寄存器（位于快速操作存储区，地址 F91h）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7 **ALRMEN:** 闹钟使能位
1 = 使能闹钟（每当 ARPT<7:0> = 0000 0000 且 CHIME = 0 时，发生闹钟事件后都自动清零）
0 = 禁止闹钟
- bit 6 **CHIME:** 响铃（Chime）使能位
1 = 使能响铃；ARPT<7:0> 位允许从 00h 返回到 FFh
0 = 禁止响铃；ARPT<7:0> 到达 00h 就停止
- bit 5-2 **AMASK<3:0>:** 闹钟掩码配置位
0000 = 每半秒
0001 = 每秒
0010 = 每 10 秒
0011 = 每分钟
0100 = 每 10 分钟
0101 = 每小时
0110 = 一天一次
0111 = 一周一次
1000 = 一月一次
1001 = 一年一次（配置在 2 月 29 日时除外，这种情况下每 4 年一次）
101x = 保留 —— 不要使用
11xx = 保留 —— 不要使用
- bit 1-0 **ALRMPTR<1:0>:** 闹钟值寄存器窗口指针位
读取 ALRMVALH 和 ALRMVALL 寄存器时，指向相应的闹钟值寄存器。每当读或写 ALRMVALH 时 ALRMPTR<1:0> 的值就递减 1，直到达到 00。
ALRMVAL<15:8>:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = 未实现
ALRMVAL<7:0>:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = 未实现

PIC18F46J50 系列

寄存器 17-5: **ALRMRPT**: 闹钟重复计数器寄存器（位于快速操作存储区，地址 **F90h**）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
bit 7							bit 0

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

bit 7-0 **ARPT<7:0>**: 闹钟重复计数器值位
11111111 = 闹钟将再重复 255 次
.
.
.
00000000 = 闹钟将不再重复
每当发生闹钟事件时计数器就递减 1。除非 CHIME = 1，否则计数器不能从 00h 返回到 FFh。

17.1.2 RTCVALH 和 RTCVALL 寄存器映射

寄存器 17-6: 保留的寄存器（位于快速操作存储区，地址 F99h，PTR 11b）

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-0

未实现：读为 0

寄存器 17-7: **YEAR:** 年份值寄存器（位于快速操作存储区，地址 F98h，PTR 11b）⁽¹⁾

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-4

YRTEN<3:0>: 年份的十位数的二 - 十进制码（Binary Coded Decimal，BCD）值为 0 到 9。

bit 3-0

YRONE<3:0>: 年份的个位数的 BCD 值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时才允许写入 YEAR 寄存器。

寄存器 17-8: **MONTH:** 月值寄存器（位于快速操作存储区，地址 F99h，PTR 10b）⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MHTTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-5

未实现：读为 0

bit 4

MHTTEN0: 月份的十位数的 BCD 值为 0 或 1。

bit 3-0

MTHONE<3:0>: 月份的个位数的 BCD 值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

PIC18F46J50 系列

寄存器 17-9: DAY: 日值寄存器（位于快速操作存储区，地址 F98h，PTR 10b）⁽¹⁾

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

- bit 7-6 未实现：读为 0
- bit 5-4 DAYTEN<1:0>: 日的十位数的 BCD 值为 0 到 3。
- bit 3-0 DAYONE<3:0>: 日的个位数的 BCD 值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

寄存器 17-10: WKDY: 星期值寄存器（位于快速操作存储区，地址 F99h，PTR 01b）⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 7							bit 0

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

- bit 7-3 未实现：读为 0
- bit 2-0 WDAY<2:0>: 星期的 BCD 值为 0 到 6。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

寄存器 17-11: HOURS: 小时值寄存器（位于快速操作存储区，地址 **F98h**，**PTR 01b**）⁽¹⁾

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-6 **未实现:** 读为 0

bit 5-4 **HRTEN<1:0>:** 小时的十位数的 BCD 值为 0 到 2。

bit 3-0 **HRONE<3:0>:** 小时的个位数的 BCD 值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

寄存器 17-12: MINUTES: 分钟值寄存器（位于快速操作存储区，地址 **F99h**，**PTR00b**）

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **未实现:** 读为 0

bit 6-4 **MINTEN<2:0>:** 分钟的十位数的 BCD 值为 0 到 5。

bit 3-0 **MINONE<3:0>:** 分钟的个位数的 BCD 值为 0 到 9。

寄存器 17-13: SECONDS: 秒值寄存器（位于快速操作存储区，地址 **F98h**，**PTR 00b**）

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **未实现:** 读为 0

bit 6-4 **SECTEN<2:0>:** 秒的十位数的 BCD 值为 0 到 5。

bit 3-0 **SECONE<3:0>:** 秒的个位数的 BCD 值为 0 到 9。

PIC18F46J50 系列

17.1.3 ALRMVALH 和 ALRMVALL 寄存器映射

寄存器 17-14: **ALRMMNTH**: 闹钟月值寄存器（位于快速操作存储区，地址 F8Fh，PTR 10b）⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 7							bit 0

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

- bit 7-5 未实现：读为 0
- bit 4 **MTHTEN0**: 月份的十位数的 BCD 值
值为 0 或 1。
- bit 3-0 **MTHONE<3:0>**: 月份的个位数的 BCD 值
值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

寄存器 17-15: **ALRMDAY**: 闹钟日值寄存器（位于快速操作存储区，地址 F8Eh，PTR 10b）⁽¹⁾

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

- bit 7-6 未实现：读为 0
- bit 5-4 **DAYTEN<1:0>**: 日的十位数的 BCD 值
值为 0 到 3。
- bit 3-0 **DAYONE<3:0>**: 日的个位数的 BCD 值
值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

寄存器 17-16: ALRMWD: 闹钟星期值寄存器（位于快速操作存储区，地址 F8Fh，PTR 01b）⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-3 **未实现:** 读为 0

bit 2-0 **WDAY<2:0>:** 星期的 BCD 值
值为 0 到 6。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

寄存器 17-17: ALRMHR: 闹钟小时值寄存器（位于快速操作存储区，地址 F8Eh，PTR 01b）⁽¹⁾

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-6 **未实现:** 读为 0

bit 5-4 **HRTEN<1:0>:** 小时的十位数的 BCD 值
值为 0 到 2。

bit 3-0 **HRONE3:HRONE0:** 小时的个位数的 BCD 值
值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

PIC18F46J50 系列

寄存器 17-18: ALRMMIN: 闹钟分钟值寄存器（位于快速操作存储区，地址 F8Fh，PTR 00b）

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 7							bit 0

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

- bit 7 未实现：读为 0
- bit 6-4 MINTEN<2:0>: 分钟的十位数的 BCD 值
值为 0 到 5。
- bit 3-0 MINONE<3:0>: 分钟的个位数的 BCD 值
值为 0 到 9。

寄存器 17-19: ALRMSEC: 闹钟秒值寄存器（位于快速操作存储区，地址 F8Eh，PTR 00b）

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

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

- bit 7 未实现：读为 0
- bit 6-4 SECTEN<2:0>: 秒的十位数的 BCD 值
值为 0 到 5。
- bit 3-0 SECONE<3:0>: 秒的个位数的 BCD 值
值为 0 到 9。

17.1.4 RTCEN 位写

RTCWREN = 0 时尝试写入 RTCEN 位将被忽略。
RTCWREN 必须置 1，然后才会发生对 RTCEN 的写操作。
像 RTCEN 位一样，RTCVALH 和 RTCVALL 寄存器只能在 RTCWREN = 1 时写入。RTCWREN = 0 时写入这些寄存器将被忽略。

17.2 工作原理

17.2.1 寄存器接口

RTCC 和闹钟值的寄存器接口是用二-十进制码 (BCD) 格式实现的。这在使用该模块时简化了固件，因为每个数字都包含在它自己的 4 位值中了 (见图 17-2 和图 17-3)。

图 17-2: 定时器位格式

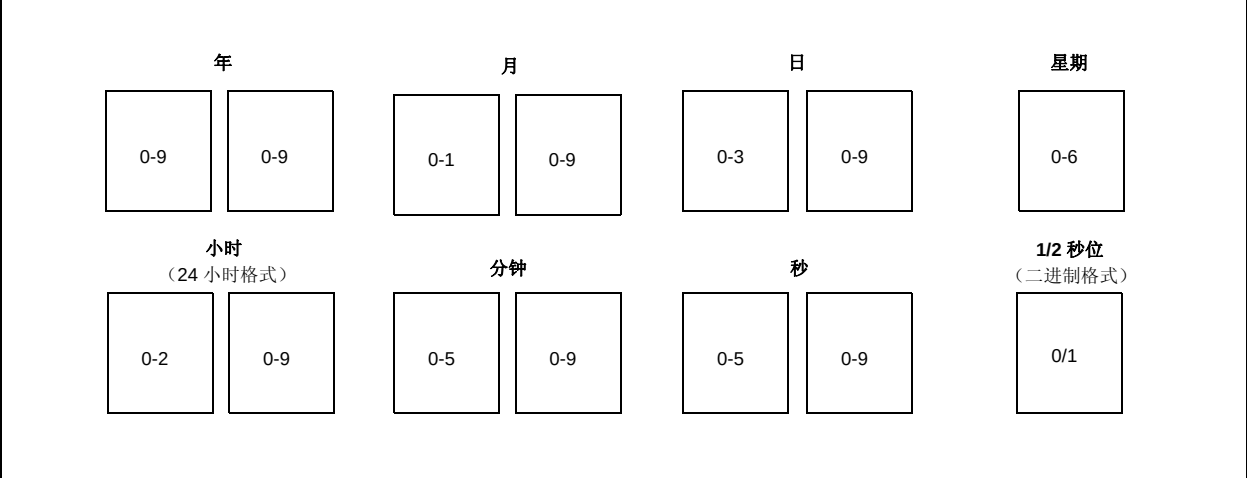
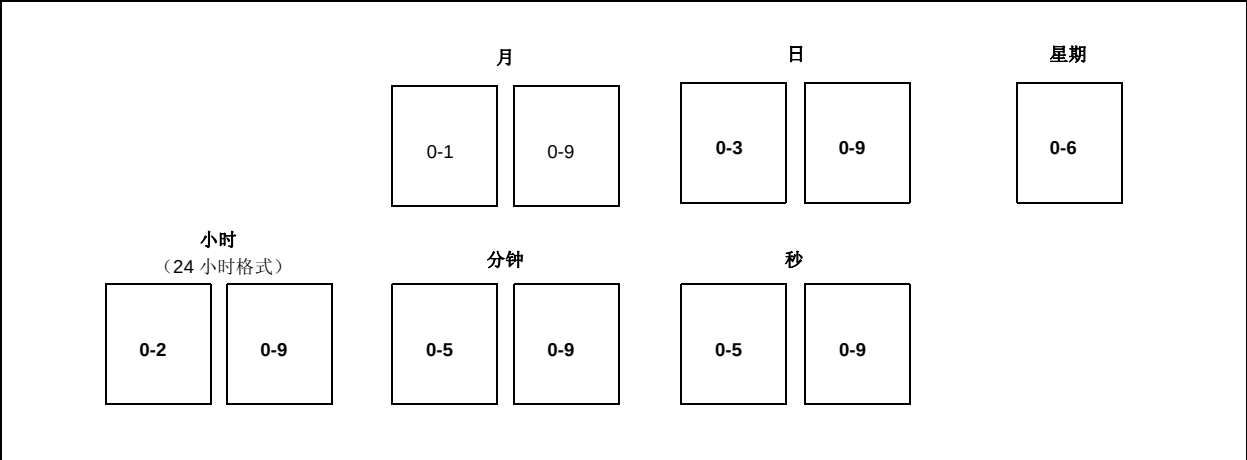


图 17-3: 闹钟位格式



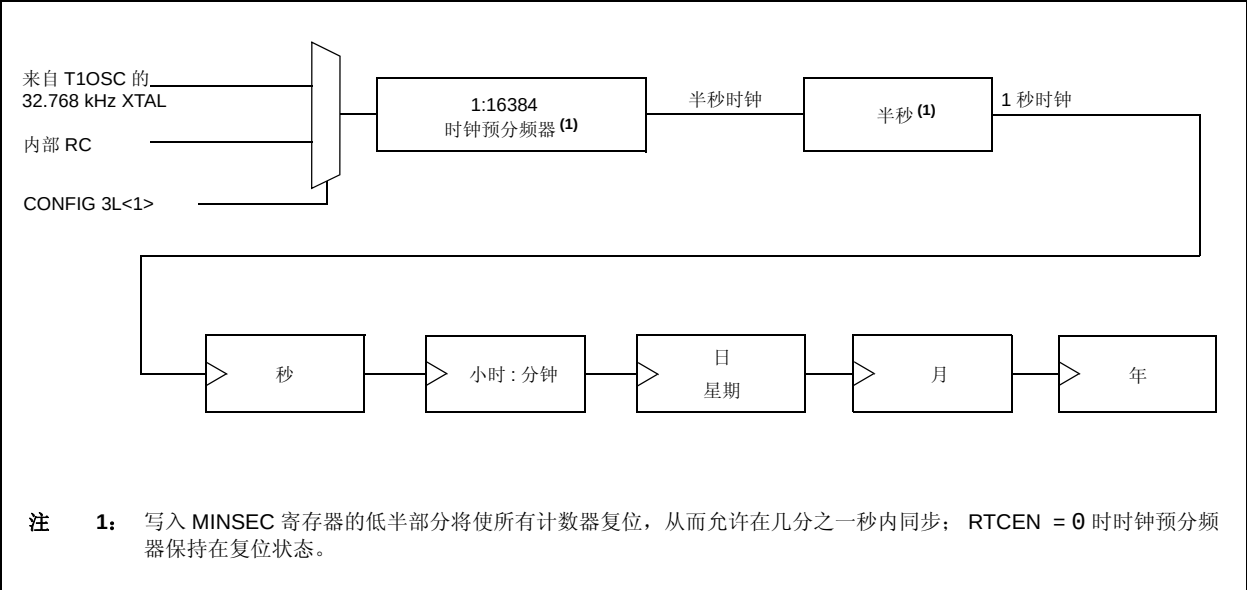
PIC18F46J50 系列

17.2.2 时钟源

正如之前提到的，RTCC 模块应由 32.768 kHz 的外部实时时钟（RTC）晶振提供时钟源，但也可以由 INTRC 提供时钟源。RTCC 时钟选择由 RTCOSC 位（CONFIG3L<1>）决定。

晶振校准可通过该模块实现，产生的误差为每月 3 秒或更小。（更多详细信息，请参见第 17.2.9 节“校准”。）

图 17-4: 时钟源复用



17.2.2.1 实时时钟使能

RTCC 模块可由外部 32.768 kHz 晶振（Timer1 振荡器或 T1CKI 输入）或 INTRC 振荡器提供时钟源，这通过 CONFIG3L<1> 位进行选择。

若将 Timer1 振荡器用作 RTCC 的时钟源，则应确保将 T1CON<3>（T1OSCEN）置 1 使能 Timer1。由 PADCFG 寄存器中的 RTSECSSEL<1:0> 位选定的 RTC 时钟，可输出至 RTCC 引脚。

17.2.3 进位规则

本节说明发生计满返回时，会影响哪些定时器值。

- 日时间：从 23:59:59 到 00:00:00，向日字段进位
- 月：从 12/31 到 01/01，向年字段进位
- 星期：从 6 到 0，无进位（见表 17-1）
- 年进位：从 99 到 00；这也超越 RTCC 的使用

关于日到月的计满返回设定，请参见表 17-2。

考虑到以下值是 BCD 格式，进位到 BCD 的高位将在计数为 10 时发生，而不是在计数为 16 时发生（SECONDS、MINUTES、HOURS、WEEKDAY、DAYS 和 MONTHS）。

表 17-1: 星期设定

星期	
星期日	0
星期一	1
星期二	2
星期三	3
星期四	4
星期五	5
星期六	6

表 17-2: 日到月计满返回设定

月	最大日字段
01 (一月)	31
02 (二月)	28 或 29 ⁽¹⁾
03 (三月)	31
04 (四月)	30
05 (五月)	31
06 (六月)	30
07 (七月)	31
08 (八月)	31
09 (九月)	30
10 (十月)	31
11 (十一月)	30
12 (十二月)	31

注 1: 请参见第 17.2.4 节“闰年”。

17.2.4 闰年

由于 RTCC 模块的年份范围是 2000 到 2099, 闰年是通过以上范围内的年份能否被 4 整除来确定的。闰年中唯一受影响的月份是二月。

闰年中二月有 29 天, 而其他年份中二月是 28 天。

17.2.5 一般功能

所有包含秒或更大时间值的定时器寄存器都可写。用户只需将要写入的年、月、日、小时、分钟和秒通过寄存器指针写入定时器寄存器, 就可以配置时间 (见第 17.2.8 节“寄存器映射”)。

随后定时器就会用新写入的值从所需起点继续计数。

通过将 RTCEN 位 (RTCCFGL<7>) 置 1 使能 RTCC。如果定时器在调节这些寄存器时是使能的, 则定时器仍将继续递增。但是, 任何时候只要写入 MINSEC 寄存器, 两个定时器预分频器都将复位为 0。这就允许在几分之一秒内同步。

定时器寄存器的更新周期与 CPU 执行写指令的周期相同。用户必须确保 RTCEN = 1 时, 更新的寄存器不会同时递增。这可以通过几种方式实现:

- 检查 RTCSYNC 位 (RTCCFG<4>)
- 检查前面可能发生进位的位
- 在秒脉冲 (或闹钟中断) 后立即更新寄存器

用户可看到计数器的半秒字段。该值是只读的, 只能通过写入 SECONDS 寄存器的低半部分复位。

17.2.6 寄存器读写安全窗口

RTCSYNC 位指示可安全读写 RTCC 时钟域寄存器、而无需担心计满返回的时间窗。当 RTCSYNC = 0 时, CPU 可安全访问这些寄存器。

无论 RTCSYNC = 1 还是 0, 用户都应采用固件方法确保数据读取没有发生在计满返回边界, 从而导致无效或部分读取。该固件方法由读取每个寄存器两遍, 然后比较两个值组成。如果两个值匹配, 则没有发生计满返回。

17.2.7 写锁定

为执行对任何 RTCC 定时器寄存器的写入, RTCWREN 位 (RTCCFG<5>) 必须置 1。

为避免意外写入 RTCC 定时器寄存器, 建议在不写入该寄存器其他任何时候 RTCWREN 位 (RTCCFG<5>) 都保持清零。要将 RTCWREN 位置 1, 在 55h/AA 序列和 RTCWREN 置 1 之间只允许一个指令周期的时间窗。因此, 建议用户遵循例 17-1 中的代码示例。

例 17-1: 将 RTCWREN 位置 1

```
movlb 0x0F          ;RTCCFG is banked
bcf    INTCON, GIE   ;Disable interrupts
movlw  0x55
movwf  EECON2
movlw  0xAA
movwf  EECON2
bsf    RTCCFG, RTCWREN
```

17.2.8 寄存器映射

为限制寄存器接口, RTCC 定时器和闹钟定时器寄存器通过相应的寄存器指针访问。RTCC 值寄存器窗口 (RTCVALH<15:8> 和 RTCVALL<7:0>) 使用 RTCPTR 位 (RTCCFG<1:0>) 选择所需的定时器寄存器对。

通过读或写 RTCVALH 寄存器, RTCC 指针值 (RTCPTR<1:0>) 递减 1, 直到达到 00。一旦达到 00, MINUTES 和 SECONDS 值可通过 RTCVALH 和 RTCVALL 访问, 直到手动更改指针值。

PIC18F46J50 系列

表 17-3: RTCVALH 和 RTCVALL 寄存器映射

RTCPTR<1:0>	RTCC 值寄存器窗口	
	RTCVAL<15:8>	RTCVAL<7:0>
00	分钟	秒
01	星期	小时
10	月	日
11	—	年

闹钟值寄存器窗口（ALRMVALH 和 ALRMVALL）使用 ALRMPTR 位（ALRMCFG<1:0>）选择所需的闹钟寄存器对。

通过读或写 ALRMVALH 寄存器，闹钟指针值 ALRMPTR <1:0> 递减 1，直到达到 00。一旦达到 00，ALRMMIN 和 ALRMSEC 值可通过 ALRMVALH 和 ALRMVALL 访问，直到手动更改指针值。

表 17-4: ALRMVAL 寄存器映射

ALRMPTR<1:0>	闹钟值寄存器窗口	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

17.2.9 校准

实时晶振输入可用周期性自动调节功能校准。正确校准后，RTCC 可提供小于每月 3 秒的误差。

为实现该校准，需要找到误差时钟脉冲数并将该值存储到 RTCCAL 寄存器的低半部分。装入 RTCCAL 的 8 位有符号值乘以 4，每分钟一次将其加到 RTCC 定时器中或从 RTCC 定时器中减去。

校准 RTCC 模块：

1. 使用器件上的其他定时器资源找出 32.768 kHz 晶振的误差。
2. 转换每分钟误差时钟脉冲数（见公式 17-1）。

公式 17-1: 转换误差时钟脉冲

(理想频率 (32,768) – 测得频率) *
60 = 每分钟误差时钟数

- 如果振荡器快于理想频率（从步骤 2 得出的负的结果），RCFGCALL 寄存器值必须为负。这会导致每分钟从定时器计数器减去指定的时钟脉冲数。
- 如果振荡器慢于理想频率（从步骤 2 得出的正的结果），RCFGCALL 寄存器值必须为正。这会导致每分钟在定时器计数器上加上指定的时钟脉冲数。

3. 将正确的值装入 RTCCAL 寄存器。

只有当定时器关闭或紧接秒脉冲的上升沿时，才会发生对 RTCCAL 寄存器的写操作。

注：在确定晶振误差值时，是否包含由温度或晶振老化所造成的漂移引起的初始误差，由用户自行决定。

17.3 闹钟

闹钟功能和特性如下所示：

- 可在半秒到一年的范围内配置
- 使用 ALRMEN 位（寄存器 17-4 的 ALRMCFG<7>）使能
- 提供一次性闹钟和重复闹钟选项

17.3.1 配置闹钟

闹钟功能用 ALRMEN 位使能。

发出闹钟后该位清零。如果 CHIME 位 = 1 或者 ALRMRPT ≠ 0，则不会清零该位。

闹钟的间隔时间选择通过 ALRMCFG 位（AMASK<3:0>）配置。（见图 17-5。）这些位决定了要触发闹钟，闹钟的哪些位以及多少位必须和时钟值匹配。

也可以配置闹钟使之根据预先配置的间隔时间重复。使能闹钟后发生的总次数存储在 ALRMRPT 寄存器中。

注： 使能闹钟（ALRMEN = 1）时，更改除 RTCCAL、ALRMCFG 和 ALRMRPT 寄存器以外的任何寄存器以及 CHIME 位，都会导致误闹钟事件，进而导致错误的闹钟中断。为避免这种情况，只应在禁止闹钟（ALRMEN = 0）时更改定时器值和闹钟值。建议在 RTCSYNC = 0 时更改 ALRMCFG 和 ALRMRPT 寄存器以及 CHIME 位。

图 17-5: 闹钟掩码设置

闹钟掩码设置 AMASK<3:0>	星期	月	日	小时	分钟	秒
0000 —— 每半秒						
0001 —— 每秒						
0010 —— 每 10 秒						s
0011 —— 每分钟						s
0100 —— 每 10 分钟					m	s
0101 —— 每小时				m	m	s
0110 —— 每日				h	h	s
0111 —— 每周	d			h	h	s
1000 —— 每月			d	h	h	s
1001 —— 每年 ⁽¹⁾		m	m	d	d	s

注 1： 每年，除非配置为 2 月 29 日。

PIC18F46J50 系列

当 $ALRMCFG = 00$ 且 $CHIME$ 位 = 0 ($ALRMCFG<6>$) 时, 重复功能被禁止, 只发生单次闹钟。通过将 FFh 装入 $ALMRPT$ 寄存器, 闹钟可重复最多 255 次。

每个闹钟发出后, $ALMRPT$ 寄存器都递减 1。一旦寄存器达到 00, 将最后一次发出闹钟。

最后一次发出闹钟后 $ALRMEN$ 位自动清零, 闹钟关闭。如果 $CHIME$ 位 = 1, 闹钟可能不断重复。

当 $CHIME = 1$ 时, $ALMRPT$ 寄存器达到 00 时不会禁止闹钟, 而是计满返回到 FF , 继续无限计数。

17.3.2 闹钟中断

每个闹钟事件发生时, 都会产生中断。此外会提供闹钟脉冲输出, 其频率是闹钟频率的一半。

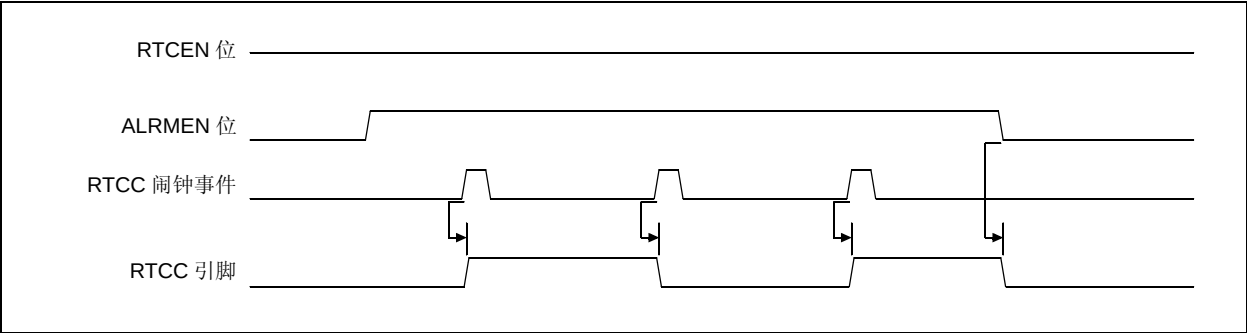
闹钟脉冲输出完全与 $RTCC$ 时钟同步, 可用作其他外设的触发时钟。可在 $RTCC$ 引脚上输出闹钟脉冲。输出脉冲是占空比为 50% 的时钟, 频率为闹钟事件频率的一半 (见图 17-6)。

$RTCC$ 引脚也能输出秒时钟。用户可在 $RTCC$ 模块生成的闹钟脉冲或秒时钟输出之间选择。

$RTSECSEL$ ($PADCFG1<2:1>$) 位在以下两个输出间选择:

- 闹钟脉冲 —— $RTSECSEL<2:1> = 00$
- 秒时钟 —— $RTSECSEL<2:1> = 0$

图 17-6: 定时器脉冲的产生



17.4 低功耗模式

可选择在休眠、空闲和深度休眠模式下继续运行定时器和闹钟。闹钟事件可用来唤醒低功耗模式下的单片机。

17.5 复位

17.5.1 器件复位

发生器件复位时, $ALRMCFG$ 和 $ALMRPT$ 寄存器强制进入其复位状态, 导致闹钟被禁止 (如果复位前是使能的)。如果 $RTCC$ 是使能的, 发生基本器件复位后将继续工作。

17.5.2 上电复位 (POR)

$RTCCFG$ 和 $ALMRPT$ 寄存器只在 POR 时复位。器件退出 POR 状态后, 时钟寄存器应重新装入所需值。

定时器预分频器值只能通过写入 $SECONDS$ 寄存器复位。器件复位不会影响预分频器。

17.6 寄存器映射

表 17-5、表 17-6 和表 17-7 总结了与 RTCC 模块相关的寄存器。

表 17-5: RTCC 控制寄存器

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
RTCCFG	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	0000
RTCCAL	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000
PADCFG1	—	—	—	—	—	RTSECSEL1	RTSECSEL0	PMPTTL	0000
ALRMCFG	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	0000
ALRMRPT	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCCIF	0000
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCCIE	0000
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCICIP	0000

图注: — = 未实现, 读为 0。44 引脚器件的复位值以十六进制显示。

表 17-6: RTCC 值寄存器

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
RTCVALH	RTCC 值寄存器窗口的高字节, 基于 RTCPTR<1:0>								XXXX
RTCVALL	RTCC 值寄存器窗口的低字节, 基于 RTCPTR<1:0>								XXXX
RTCCFG	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	0000
ALRMCFG	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	0000
ALRMVALH	闹钟值寄存器窗口的高字节, 基于 ALRMPTR<1:0>								XXXX
ALRMVALL	闹钟值寄存器窗口的低字节, 基于 ALRMPTR<1:0>								XXXX

图注: — = 未实现, 读为 0。44 引脚器件的复位值以十六进制显示。

表 17-7: 闹钟值寄存器

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
ALRMRPT	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
ALRMVALH	闹钟值寄存器窗口的高字节, 基于 ALRMPTR<1:0>								XXXX
ALRMVALL	闹钟值寄存器窗口的低字节, 基于 ALRMPTR<1:0>								XXXX
RTCCAL	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000
RTCVALH	RTCC 值寄存器窗口的高字节, 基于 RTCPTR<1:0>								XXXX
RTCVALL	RTCC 值寄存器窗口的低字节, 基于 RTCPTR<1:0>								XXXX

图注: — = 未实现, 读为 0。44 引脚器件的复位值以十六进制显示。

PIC18F46J50 系列

注:

18.0 增强型捕捉 / 比较 / PWM
(ECCP) 模块

PIC18F46J50系列器件具有两个增强型捕捉/比较/PWM (ECCP) 模块: ECCP1 和 ECCP2。这两个模块包含一个 16 位寄存器, 可用作 16 位捕捉寄存器、16 位比较寄存器或 PWM 主 / 从占空比寄存器。这两个 ECCP 模块与早期的许多 PIC16 和 PIC18 器件中的 CCP 模块兼容。

注: 引用两个 ECCP 模块之一的寄存器名称和位名称使用 “x” 来代替模块编号。例如, 寄存器 CCP1CON 和 CCP2CON 具有相同的定义, 通称为 CCPxCON。图形和框图中使用基于 ECCP1 的名称, 但那些名称也适用于 ECCP2, 只是需要使用 “2” 替换图示名称中的 “1”。
在编写固件时, 寄存器名称和位名称中的 “x” 必须替换为相应的模块编号。

ECCP1 和 ECCP2 实现为具有增强型 PWM 功能的标准 CCP 模块。这些功能包括:

- 提供 2 路或 4 路输出通道
- 输出转向模式
- 可编程极性
- 可编程死区控制
- 自动关闭和重启

第 18.5 节 “PWM (增强型模式)” 将详细讨论增强功能。

注: PxA、PxB、PxC 和 PxD 与可重映射引脚 (RPN) 关联。

PIC18F46J50 系列

寄存器 18-1: **CCPxCON: 增强型捕捉 / 比较 / PWMx 控制寄存器**（位于快速操作存储区，地址 FB4h 和 FB4h）

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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-6

PxM<1:0>: 增强型 PWM 输出配置位

如果 CCPxM<3:2> = 00、01 和 10:

xx = PxA 配置为捕捉 / 比较输入 / 输出; PxB、PxC 和 PxD 配置为端口引脚

如果 CCPxM<3:2> = 11:

00 = 单输出: PxA、PxB、PxC 和 PxD 通过转向控制（见第 18.5.7 节“脉冲转向模式”）

01 = 全桥正向输出: PxD 被调制; PxA 有效; PxB 和 PxC 无效

10 = 半桥输出: PxA 和 PxB 被调制, 带有死区控制; PxC 和 PxD 配置为端口引脚

11 = 全桥反向输出: PxB 被调制; PxC 有效; PxA 和 PxD 无效

bit 5-4

DCxB<1:0>: PWM 占空比 bit 1 和 bit 0

捕捉模式:

未使用。

比较模式:

未使用。

PWM 模式:

这两位是 10 位 PWM 占空比的低 2 位。占空比的高 8 位在 CCPRxL 中。

bit 3-0

CCPxM<3:0>: ECCPx 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，匹配时翻转输出

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，初始化 ECCPx 引脚为低电平，比较匹配时输出置 1（CCPxIF 置 1）

1001 = 比较模式，初始化 ECCPx 引脚为高电平，比较匹配时输出清零（CCPxIF 置 1）

1010 = 比较模式，仅产生软件中断，ECCPx 引脚回复到 I/O 状态

1011 = 比较模式，触发特殊事件（ECCPx 复位 TMR1 或 TMR3，启动 A/D 转换，CCxIF 位置 1）

1100 = PWM 模式; PxA 和 PxC 高电平有效; PxB 和 PxD 高电平有效

1101 = PWM 模式; PxA 和 PxC 高电平有效; PxB 和 PxD 低电平有效

1110 = PWM 模式; PxA 和 PxC 低电平有效; PxB 和 PxD 高电平有效

1111 = PWM 模式; PxA 和 PxC 低电平有效; PxB 和 PxD 低电平有效

除了可使用 CCPxCON 和 ECCPxAS 寄存器提供的扩展模式外，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 （增强型 PWM 控制）
- PSTRxCON （脉冲转向控制）

18.1 ECCP 输出和配置

取决于所选定的工作模式，增强型 CCP 模块最多有 4 路 PWM 输出。这些指定为 PxA 到 PxD 的输出通过外设引脚选择（PPS）模块进行映射。因此，各功能可被映射到任一可重映射的 I/O 引脚 RPn。输出根据选定的 ECCP 工作模式被激活。表 18-4 中总结了引脚分配。

若要将 I/O 引脚配置为 PWM 输出，必须通过设置 PxM<1:0> 和 CCPxM<3:0> 位来选择适当的 PWM 模式。端口引脚相应的 TRIS 方向位也必须设置为输出，并且需要在 PPS 模块中将输出功能分配给 I/O 引脚。（关于配置 PPS 模块的详细信息，请参见第 10.7 节“外设引脚选择（PPS）”。）

18.1.1 ECCP 模块和定时器资源

ECCP 模块根据选定的模式使用 Timer1、Timer2、Timer3 或 Timer4。模块可在捕捉或比较模式下使用 Timer1 和 Timer3，而在 PWM 模式下可使用 Timer2 和 Timer4。

表 18-1: ECCP 模式 —— 定时器资源

ECCP 模式	定时器资源
捕捉	Timer1 或 Timer3
比较	Timer1 或 Timer3
PWM	Timer2 或 Timer4

要将哪个特定的定时器分配给 ECCP 模块是由 TCLKCON 寄存器（寄存器 13-3）中的“ECCP 的定时器使能”位决定的。图 18-1 描述了这两种模块间的相互关系。捕捉操作设计为在定时器配置为同步计数器模式时使用。如果关联的定时器配置为异步计数器模式，捕捉操作可能不会按预期工作。

18.2 捕捉模式

在捕捉模式下，当相应的 ECCPx 引脚发生以下事件时，CCPRxH:CCPRxL 寄存器对捕捉 TMR1 或 TMR3 寄存器的 16 位值：

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

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

18.2.1 ECCP 引脚配置

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

此外，还需要通过外设引脚选择模块将 ECCPx 输入功能分配给 I/O 引脚。关于设置可重映射引脚的详细信息，请参见第 10.7 节“外设引脚选择 (PPS)”。

注：如果 ECCPx 引脚被配置为输出，则写端口将产生一次捕捉条件。

18.2.2 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，可能无法进行捕捉操作。可在 TCLKCON 寄存器中选择用于每个 ECCP 模块的定时器 (寄存器 13-3)。

18.2.3 软件中断

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

18.2.4 ECCP 预分频器

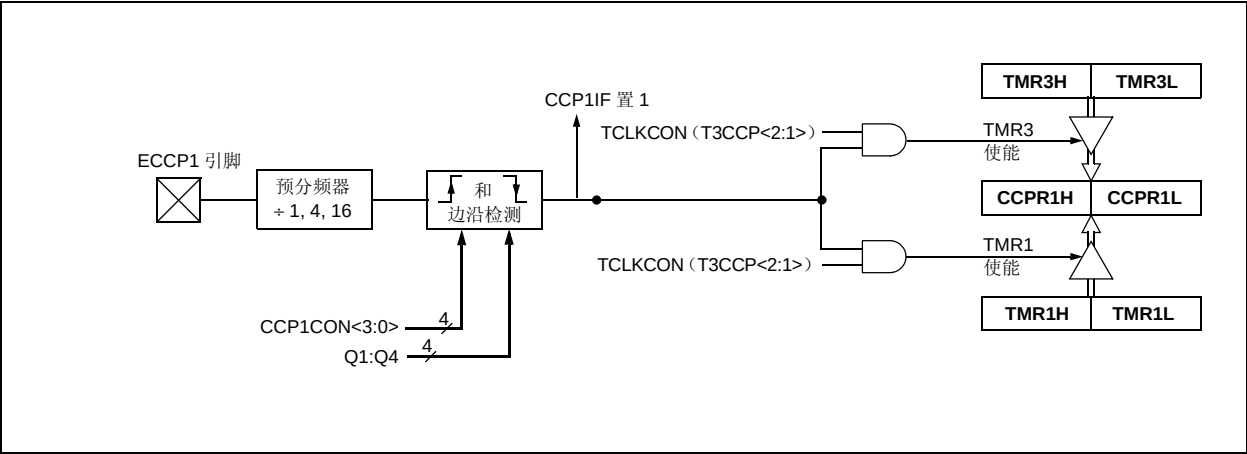
在捕捉模式下有 4 种预分频比设置；它们作为工作模式的一部分由模式选择位 (CCPxM<3:0>) 指定。只要关闭 ECCP 模块或禁止捕捉模式，预分频器计数器就会被清零。这意味着任何复位都会将预分频器计数器清零。

在两个捕捉预分频比之间切换可能会产生中断。而且，预分频器计数器不会被清零；因此，第一次捕捉可能来自于一个非零的预分频器。例 18-1 给出了切换捕捉预分频比时建议采用的方法。这个示例使预分频器计数器清零且不会产生错误中断。

例 18-1: 改变捕捉预分频比

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

图 18-1: 捕捉模式工作原理框图



18.3 比较模式

在比较模式下，16 位 CCPRx 寄存器的值不断与 TMR1 或 TMR3 寄存器对的值作比较。当两者匹配时，ECCPx 引脚将会：

- 驱动为高电平
- 驱动为低电平
- 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（CCPxM<3:0>）的值。同时，中断标志位 CCPxIF 置 1。

18.3.1 ECCP 引脚配置

用户必须通过将相应的 TRIS 位清零，将 ECCPx 引脚配置为输出。

注： 清零 CCPxCON 寄存器会将 ECCPx 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 PORTx I/O 数据锁存器。

18.3.2 TIMER1/TIMER3 模式选择

如果 ECCP 模块使用比较功能，则 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，可能无法进行比较操作。

18.3.3 软件中断模式

当选择了“产生软件中断”模式（CCPxM<3:0> = 1010）时，ECCPx 引脚不受影响；只会影响 CCPxIF 中断标志。

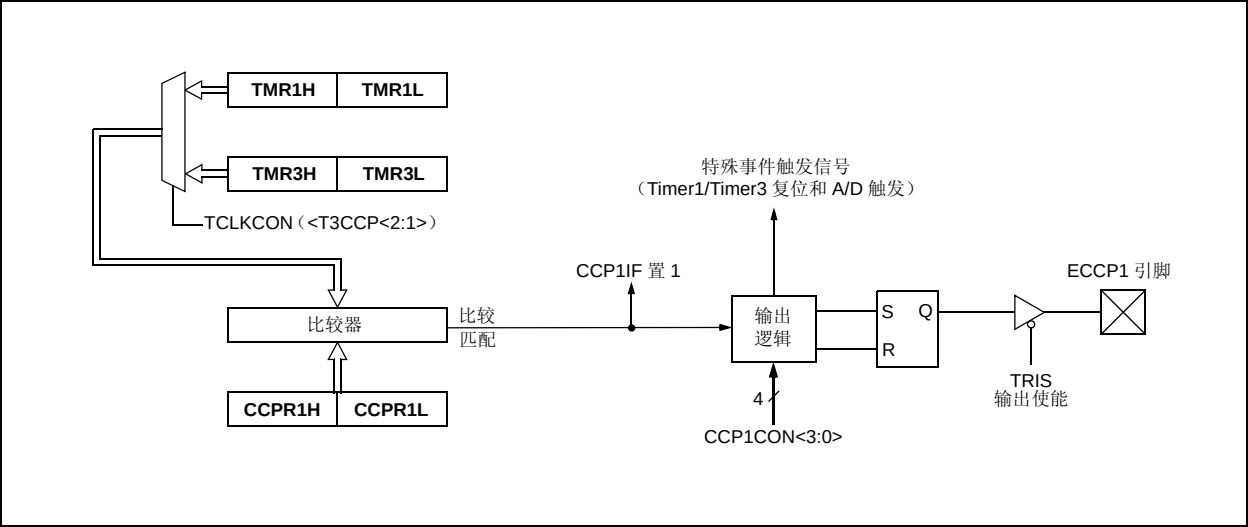
18.3.4 特殊事件触发器

ECCP 模块配备了一个特殊事件触发器。在比较模式下可产生内部硬件信号以触发其他模块动作。通过选择比较特殊事件触发模式（CCPxM<3:0> = 1011），使能特殊事件触发器。

无论当前使用哪个定时器资源作为模块的时基，特殊事件触发信号将对对应的定时器寄存器对复位。这样 CCPRx 寄存器可用作两个定时器中任一定时器的可编程周期寄存器。

特殊事件触发信号还能启动 A/D 转换。要实现此功能，必须首先使能 A/D 转换器。

图 18-2: 比较模式工作原理框图



PIC18F46J50 系列

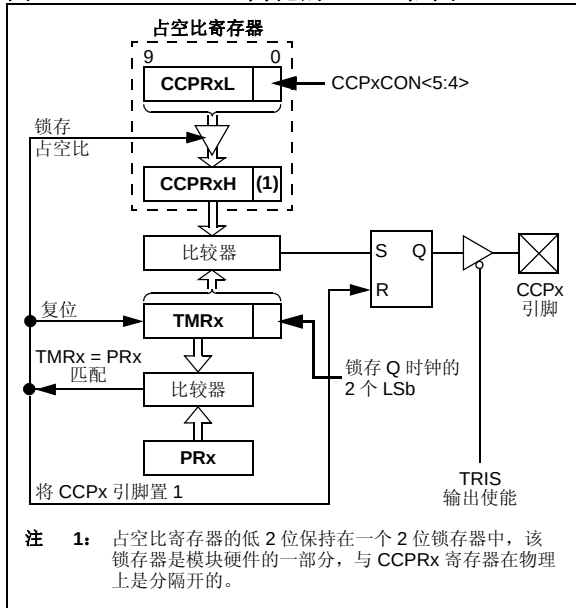
18.4 PWM 模式

在脉宽调制（Pulse-Width Modulation，PWM）模式下，CCPx 引脚产生的 PWM 输出信号的分辨率最高为 10 位。

图 18-3 给出了 PWM 模式下 CCP 模块的简化框图。

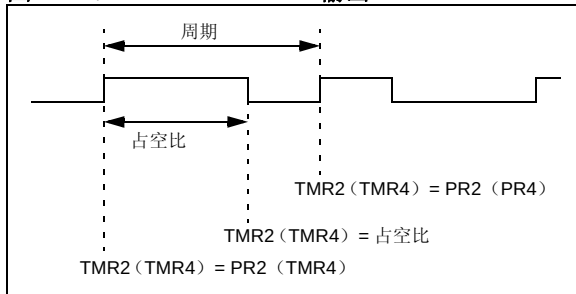
关于如何设置 CCP 模块使之在 PWM 模式下工作的步骤，请参见第 18.4.3 节“设置 PWM 工作”。

图 18-3: 简化的 PWM 框图



PWM 输出（图 18-4）具有一个时基（周期）和一段输出保持高电平的时间（占空比）。PWM 的频率是周期的倒数（1/周期）。

图 18-4: PWM 输出



18.4.1 PWM 周期

通过写入 PR2（PR4）寄存器指定 PWM 周期。可使用公式 18-1 计算 PWM 周期：

公式 18-1:

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

PWM 频率定义为 $1/[\text{PWM 周期}]$ 。

当 TMR2（TMR4）等于 PR2（PR4）时，在下一个计数器递增周期，会发生以下三个事件：

- TMR2（TMR4）清零
- CCPx 引脚置 1（例外情况：PWM 占空比 = 0% 时，CCPx 引脚不会置 1）
- PWM 占空比从 CCPRxL 锁存至 CCPRxH

注： 在确定 PWM 频率时不会使用 Timer2 和 Timer 4 后分频器（见第 15.0 节“Timer3 模块”和第 16.0 节“Timer4 模块”）。可使用不同于 PWM 输出频率的速率更新后分频器。

18.4.2 PWM 占空比

通过写入 CCPRxL 寄存器和 CCPxCON<5:4> 位指定 PWM 占空比。分辨率最高可为 10 位。CCPRxL 保存高 8 位，CCPxCON<5:4> 包含低 2 位。全部 10 位值由 CCPRxL:CCPxCON<5:4> 表示。公式 18-2 用于计算当前的 PWM 占空比。

公式 18-2:

$$\text{PWM 占空比} = (\text{CCPRxL:CCPxCON<5:4>}) \cdot T_{osc} \cdot (\text{TMR2 预分频值})$$

可在任意时间写 CCPRxL 和 CCPxCON<5:4>，但是在 PR2（PR4）和 TMR2（TMR4）匹配（即周期完成）之前，占空比值不会锁存至 CCPRxH。在 PWM 模式下，CCPRxH 是只读寄存器。

CCPRxH 寄存器与内部 2 位锁存器用来为 PWM 占空比提供双重缓冲。双重缓冲对于消除 PWM 操作中的毛刺至关重要。

当 CCPRxH 和 2 位锁存器的值与 TMR2（TMR4）结合内部 2 位 Q 时钟或 TMR2（TMR4）预分频器的 2 位匹配时，CCPx 引脚清零。

给定 PWM 频率的最大 PWM 分辨率（位）由公式 18-3 给定：

公式 18-3:

$$\text{PWM 分辨率 (最大)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ 位}$$

注： 如果 PWM 占空比值大于 PWM 周期，则 CCPx 引脚不会清零。

18.4.3 设置 PWM 工作

应执行以下步骤以将 CCP 模块配置为 PWM 工作：

- 1. 通过写入 PR2（PR4）寄存器设置 PWM 周期。
- 2. 通过写入 CCPRxL 寄存器和 CCPxCON<5:4> 位设置 PWM 占空比。
- 3. 通过将相应的 TRIS 位清零，将 CCPx 引脚设置为输出引脚。
- 4. 设置 TMR2（TMR4）预分频值，然后通过写 T2CON（T4CON）使能 Timer2（Timer4）。
- 5. 配置 CCPx 模块进行 PWM 工作。

表 18-2: 40 MHz 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 和 16）	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最大分辨率（位）	10	10	10	8	7	6.58

PIC18F46J50 系列

表 18-3: 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	69
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	70
PIR1	PMPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPPI	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	72
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	72
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	72
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	72
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	72
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	72
TCLKCON	—	—	—	T1RUN	—	—	T3CCP2	T3CCP1	74
TMR2	Timer2 寄存器								70
PR2	Timer2 周期寄存器								70
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	70
TMR4	Timer4 寄存器								73
PR4	Timer4 周期寄存器								73
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	73
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								71
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								71
CCPRL2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								71
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								71
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	73
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	73
ODCON1	—	—	—	—	—	—	ECCP2OD	ECCP1OD	74

图注: — = 未实现, 读为 0。PWM、Timer2 或 Timer4 不使用阴影单元。

18.5 PWM（增强型模式）

增强型 PWM 模式可以在最多 4 个输出引脚上产生 PWM 信号，最高可达 10 位分辨率。可通过 4 种 PWM 输出模式实现：

- 单 PWM
- 半桥 PWM
- 全桥 PWM, 正向模式
- 全桥 PWM, 反向模式

要选择增强型 PWM 模式，必须对 CCPxCON 寄存器的 PxM 位进行适当设置。

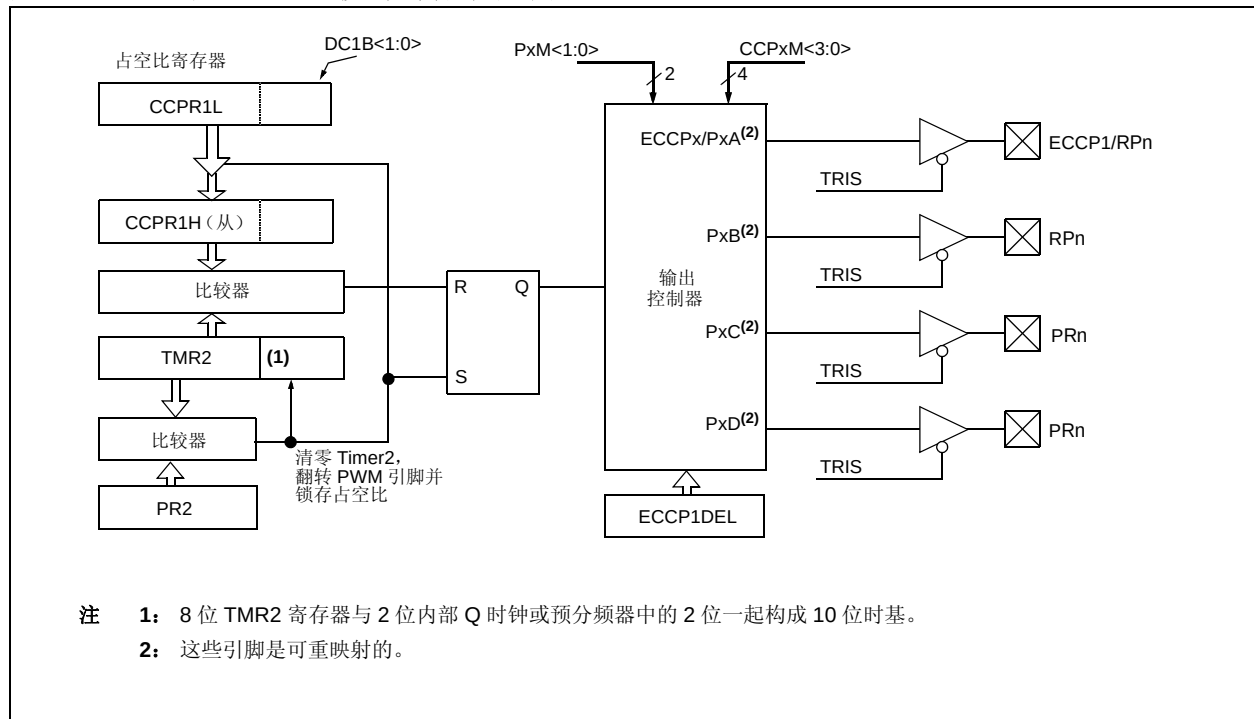
PWM 输出与 I/O 引脚复用，指定为 PxA、PxB、PxC 和 PxD。PWM 引脚的极性是可配置的，通过对 CCPxCON 寄存器中的 CCPxM 位适当设置来选择。

表 18-1 给出了每种增强型 PWM 模式的引脚分配。

图 18-5 给出了增强型 PWM 模块的简化框图的示例。

注: 为防止在开始使能PWM时产生不完整的波形, ECCP 模块在产生 PWM 信号前会等待直到新的 PWM 周期开始。

图 18-5: 增强型 PWM 模式的简化框图示例



注 1: 每个 PWM 输出的 TRIS 寄存器值必须进行适当配置。

2: 增强型 PWM 模式没有使用的任何引脚均可用于备用引脚功能。

PIC18F46J50 系列

表 18-4: 各种 PWM 增强型模式的引脚分配示例

ECCP 模式	PxM<1:0>	PxA	PxB	PxC	PxD
单 PWM	00	使用 (1)	使用 (1)	使用 (1)	使用 (1)
半桥	10	使用	使用	不使用	不使用
全桥, 正向	01	使用	使用	使用	使用
全桥, 反向	11	使用	使用	使用	使用

注 1: 在单输出模式下, 输出通过脉冲转向使能 (见寄存器 18-4)。

图 18-6: PWM (增强型模式) 输出关系示例 (高电平有效状态)

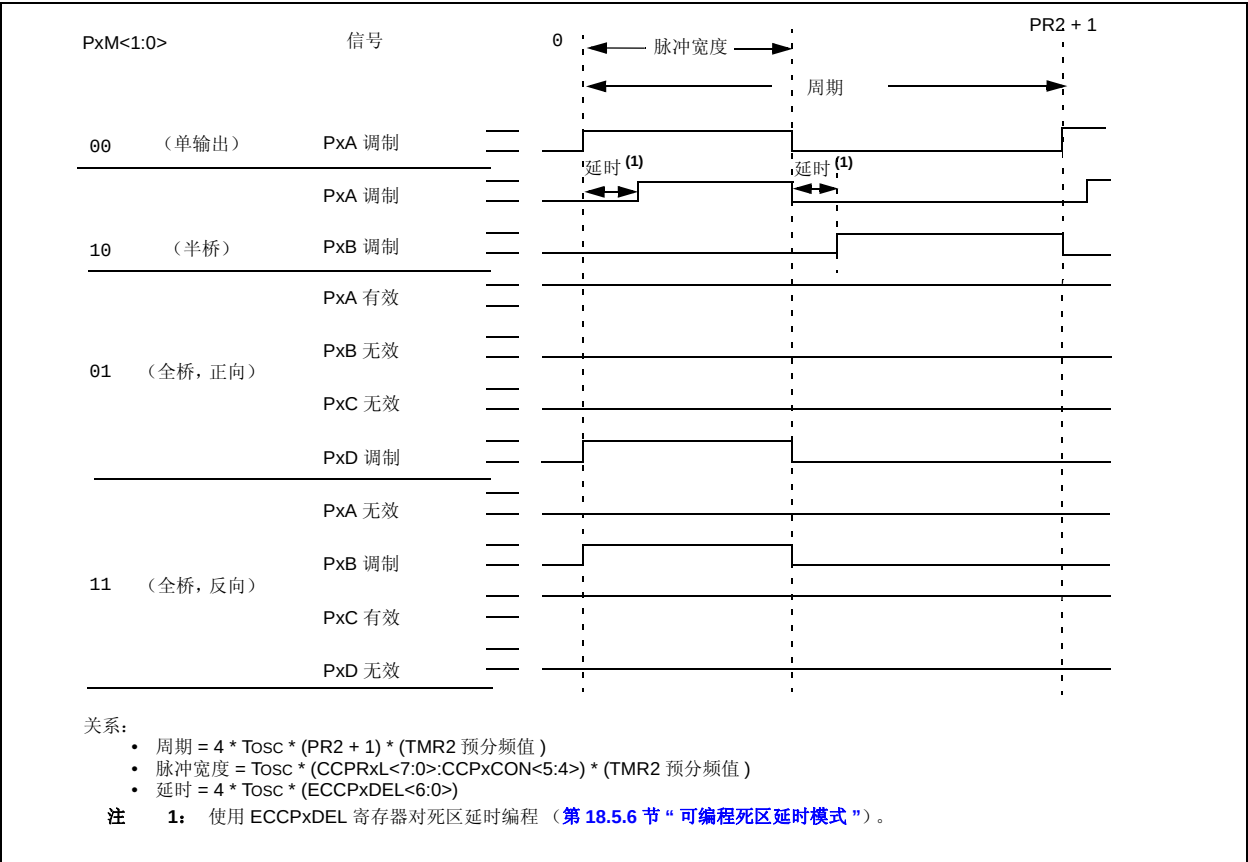
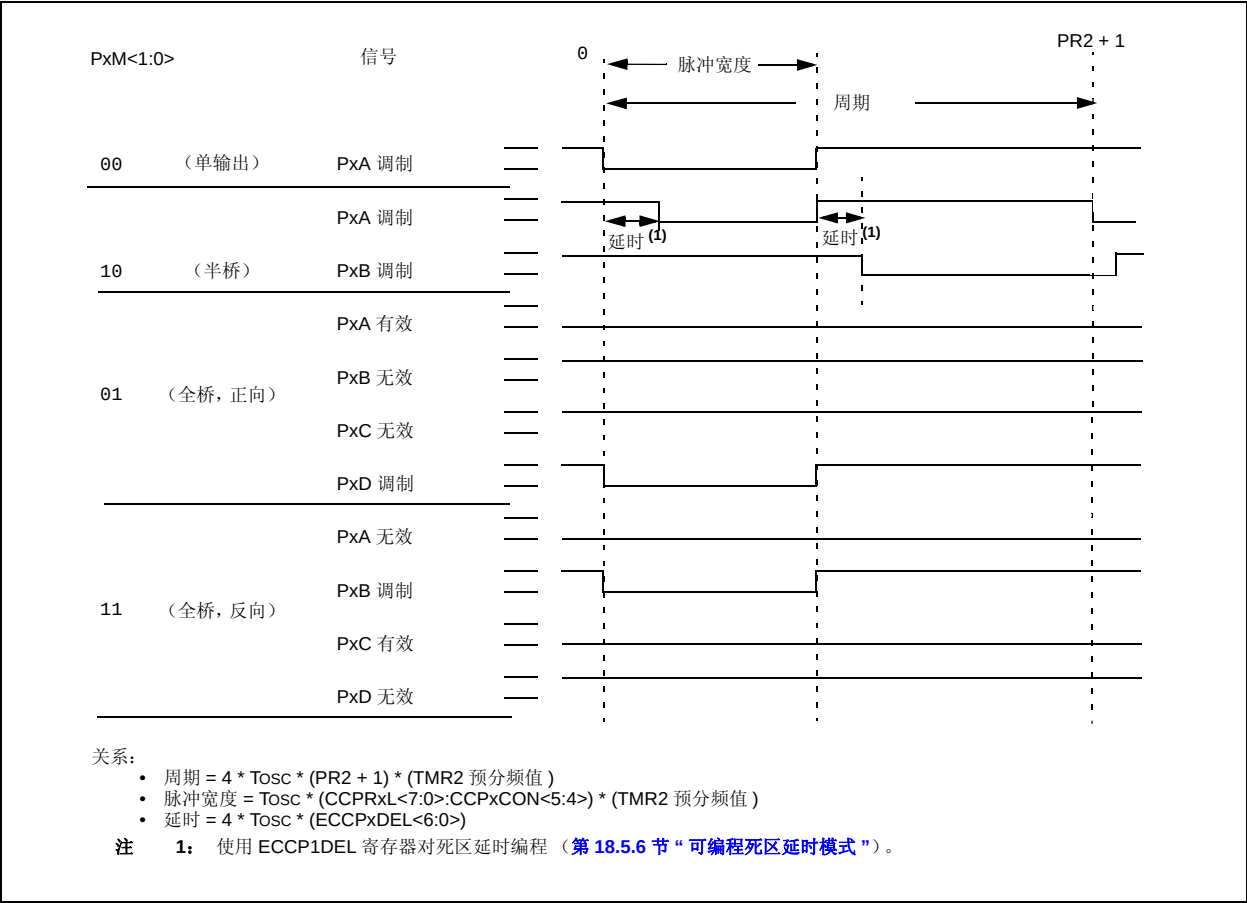


图 18-7: 增强型 PWM 输出关系示例（低电平有效状态）



PIC18F46J50 系列

18.5.1 半桥模式

在半桥模式下，有两个引脚用作输出驱动推挽式负载。PxA 引脚输出 PWM 输出信号，PxB 引脚输出互补的 PWM 输出信号（见图 18-8）。这种模式可用于半桥应用（如图 18-9 所示）；或者用于全桥应用，在全桥应用中使用两个 PWM 信号调制 4 个功率开关。

在半桥模式下，可编程死区延时可用来防止半桥功率器件中流过直通电流。ECCPxDEL 寄存器的 PxDC<6:0> 位的值设置在输出被驱动为有效之前的指令周期数。如果这个值比占空比大，则在整个周期中相应的输出保持为无效。关于死区延时操作的更多详细信息，请参见第 18.5.6 节“可编程死区延时模式”。

由于 PxA 和 PxB 输出与端口数据锁存器是复用的，相关的 TRIS 位必须清零，从而将 PxA 和 PxB 配置为输出。

图 18-8: 半桥 PWM 输出示例

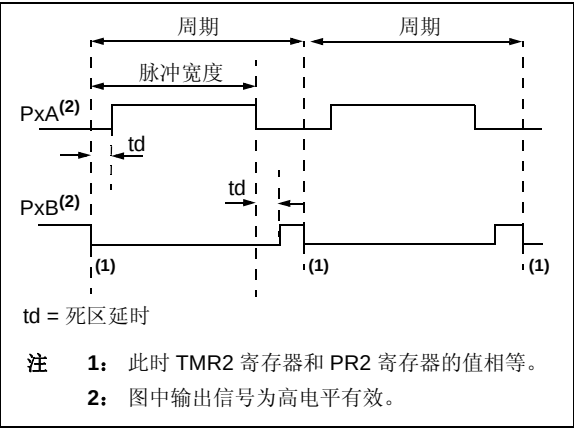
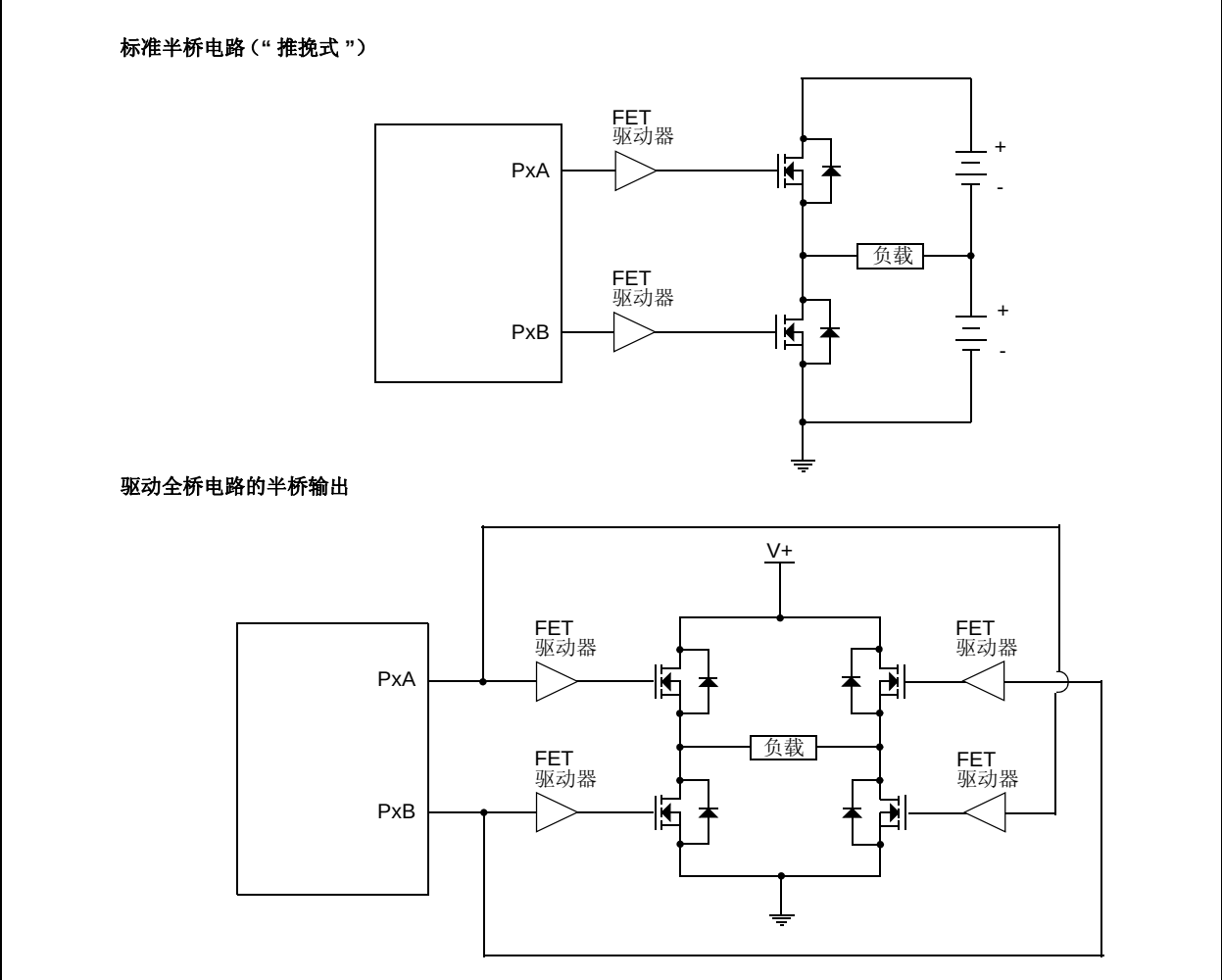


图 18-9: 半桥应用示例



18.5.2 全桥模式

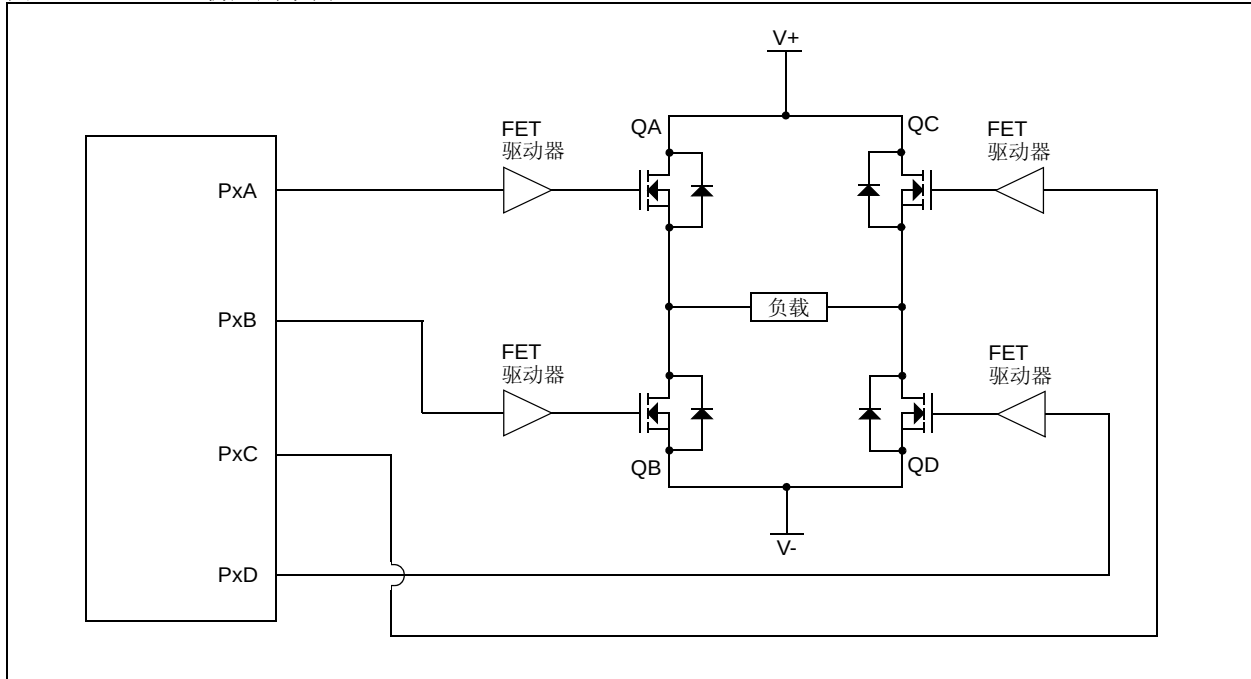
在全桥模式下，4 个引脚都用作输出。全桥应用的示例如图 18-10 所示。

在正向模式下，Px A 引脚被驱动为有效状态，Px D 引脚被调制，而 Px B 和 Px C 引脚将被驱动为无效状态，如图 18-11 所示。

在反向模式下，Px C 引脚被驱动为有效状态，Px B 引脚被调制，而 Px A 和 Px D 引脚将被驱动为无效状态，如图 18-11 所示。

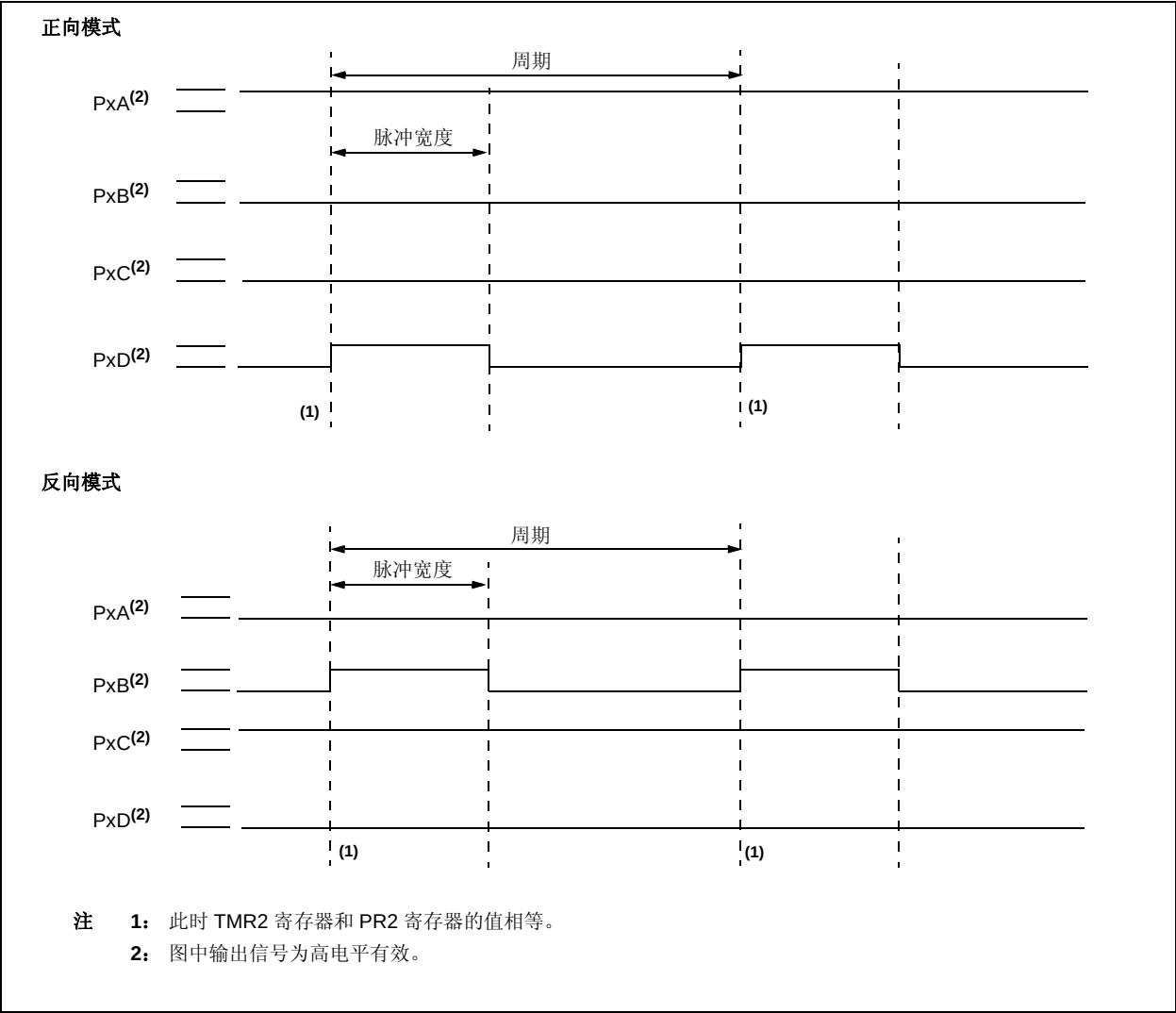
Px A、Px B、Px C 和 Px D 输出与端口数据锁存器复用。相关的 TRIS 位必须清零，从而将 Px A、Px B、Px C 和 Px D 引脚配置为输出。

图 18-10: 全桥应用示例



PIC18F46J50 系列

图 18-11: 全桥 PWM 输出示例



18.5.2.1 全桥模式中的方向改变

在全桥模式下，CCPxCON 寄存器中的 PxM1 位允许用户控制正/反方向。当应用软件改变这个方向控制位时，模块将在下一个 PWM 周期改用新的方向。

通过改变 CCPxCON 寄存器的 PxM1 位，可以用软件启动方向改变。以下序列在当前 PWM 周期结束前发生：

- 调制输出（Px_B 和 Px_D）进入无效状态。
- 相关的未调制输出（Px_A 和 Px_C）被切换到以相反的方向驱动。
- PWM 调制在下一个周期开始继续。

关于该序列的说明，请参见图 18-12。

全桥模式不提供死区延时。因为一次只有一个输出被调制，所以一般不需要死区延时。有一种情况需要死区延时，这一情况发生在以下两个条件同时满足时：

1. 当输出的占空比达到或者接近 100%，PWM 输出方向改变。
2. 功率开关（包括功率器件和驱动电路）的关断时间比导通时间要长。

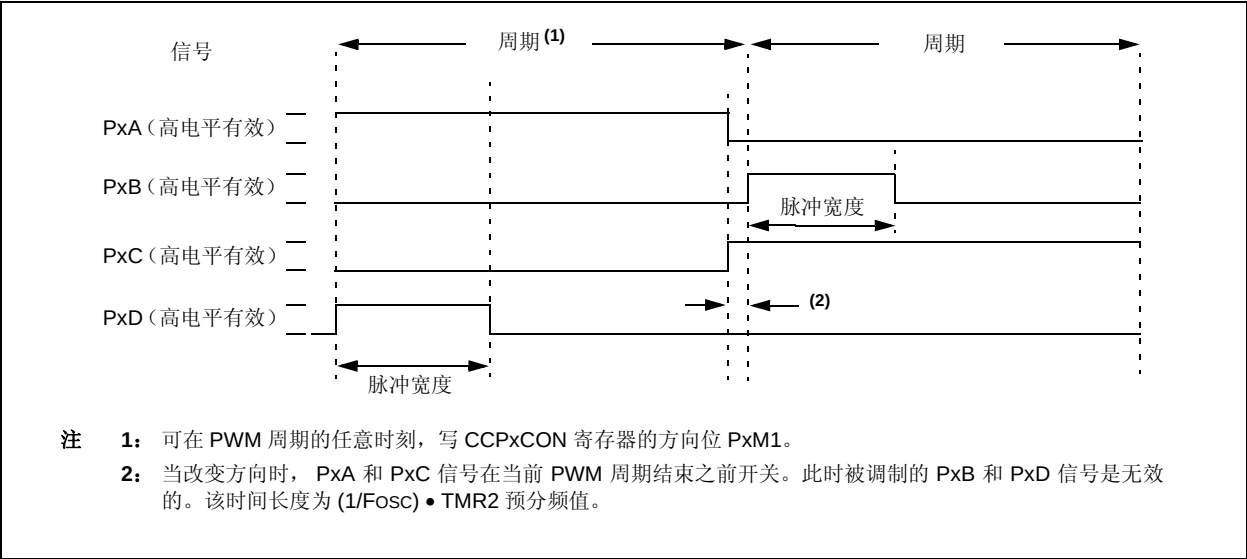
在图18-13所示的示例中，在占空比接近100%时，PWM 方向从正向改变到反向。在这个示例中，在时间 t₁，Px_A 和 Px_D 输出变为无效，而 Px_C 输出变为有效。因为功率器件的关断时间比导通时间要长，在“t”时间内，功率器件 QC 和 QD 中可能流过直通电流（见图 18-10）。当 PWM 方向从反向改变到正向时，功率器件 QA 和 QB 也将出现相同的现象。

如果应用中需要在高占空比时改变 PWM 方向，避免直通电流的两种可能方法是：

1. 在改变方向之前的一个 PWM 周期降低 PWM 占空比。
2. 使用开关驱动电路，使开关的关断时间比导通时间短。

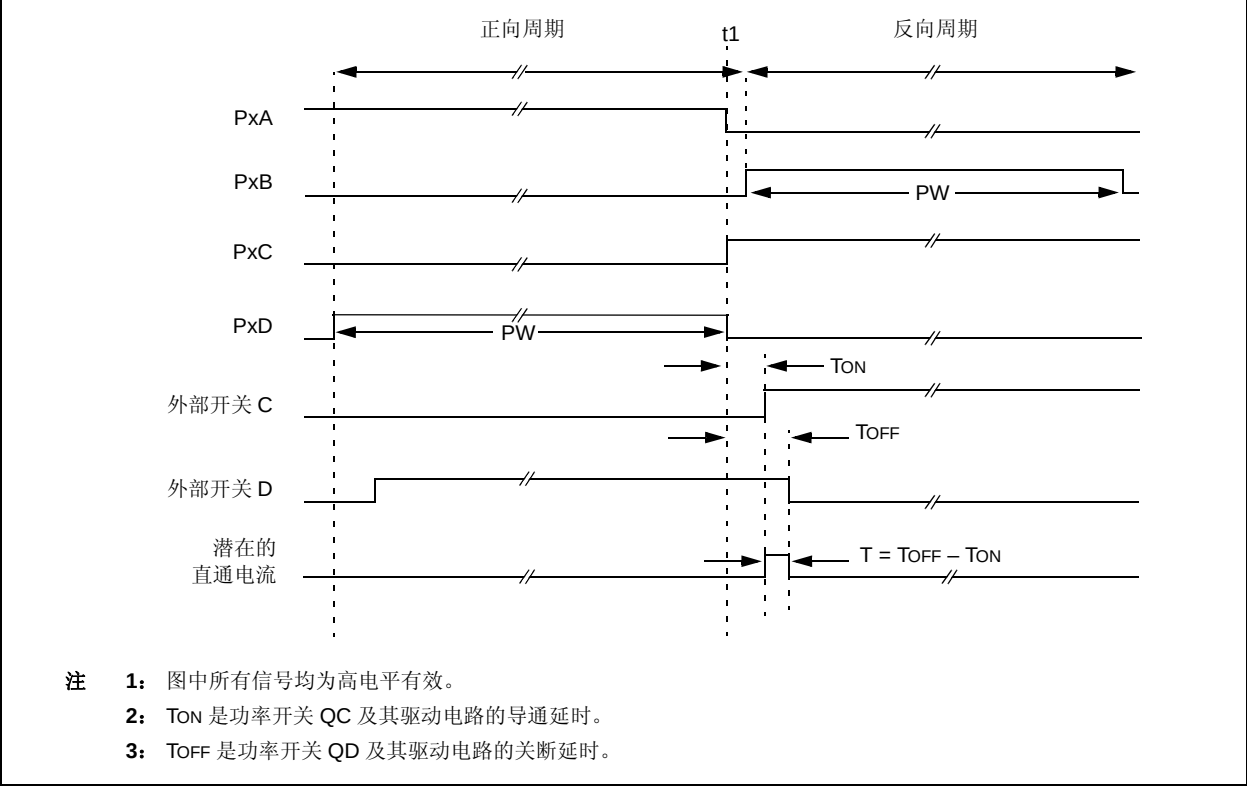
也可能存在其他避免直通电流的方案。

图 18-12: PWM 方向改变的示例



PIC18F46J50 系列

图 18-13: 在占空比接近 100% 时改变 PWM 方向的示例



18.5.3 启动注意事项

当使用 PWM 任何模式时，应用硬件必须在 PWM 输出引脚上外接适当的上拉和 / 或下拉电阻。

注： 当单片机退出复位状态时，所有 I/O 引脚呈高阻态。外部电路必须保持功率开关器件处于截止状态，直到单片机将 I/O 引脚驱动为适当的信号电平，或者激活 PWM 输出为止。

CCPxCON 寄存器的 CCPxM<1:0> 位允许用户为每一对 PWM 输出引脚（PxA/PxC 和 PxB/PxD）选择 PWM 输出信号是高电平有效还是低电平有效。PWM 输出极

性必须在使能 PWM 引脚输出驱动器之前选择。由于可能导致应用电路的损坏，因此不推荐在使能 PWM 引脚输出驱动器的同时改变极性配置。

当 PWM 模块初始化时，PxA、PxB、PxC 和 PxD 输出锁存器可能不在正确的状态。这样在使能增强型 PWM 模式的同时使能 PWM 引脚输出驱动器，可能损坏应用电路。应首先将增强型 PWM 模式配置为正确的输出模式并经过一个完整的 PWM 周期之后，再使能 PWM 引脚输出驱动器。当第二个 PWM 周期开始时，PIR1 或 PIR3 寄存器的 TMR2IF 或 TMR4IF 位置 1 表明一个完整的 PWM 周期结束了。

18.5.4 增强型 PWM 自动关闭模式

PWM 模式支持自动关闭模式，当外部关闭事件发生时将禁止 PWM 输出。自动关闭模式将 PWM 输出引脚置于预先确定的状态。该模式用于防止 PWM 损坏应用。

通过使用 ECCPxAS 寄存器的 ECCPxAS<2:0> 位来选择自动关闭源。关闭事件由以下条件产生：

- 分配为 FLT0 输入功能的引脚上的逻辑 0
- 比较器 C1
- 比较器 C2
- 用固件将 ECCPxASE 位置 1

关闭条件由 ECCPxAS 寄存器的 ECCPxASE（自动关闭事件状态）位指示。如果该位为 0，PWM 引脚正常工作。如果该位为 1，PWM 输出处于关闭状态。

当关闭事件发生时，会发生以下两个事件：

ECCPxASE 位被设为 1。ECCPxASE 将保持置 1 直到由固件清零或发生自动重启（见第 18.5.5 节“自动重启模式”）。

使能的 PWM 引脚被异步置为其关闭状态。PWM 输出引脚被分组为 [PxA/PxC] 和 [PxB/PxD] 对。每对引脚的状态由 ECCPxAS 寄存器的 PSSxAC 和 PSSxBD 位决定。每对引脚可以设置为以下 3 种状态之一：

- 驱动逻辑 1
- 驱动逻辑 0
- 三态（高阻态）

寄存器 18-2: ECCPxAS: ECCPx 自动关闭控制寄存器（位于快速操作存储区，地址 FBEh 和 FB8h）

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 = POR 时的值

1 = 置 1

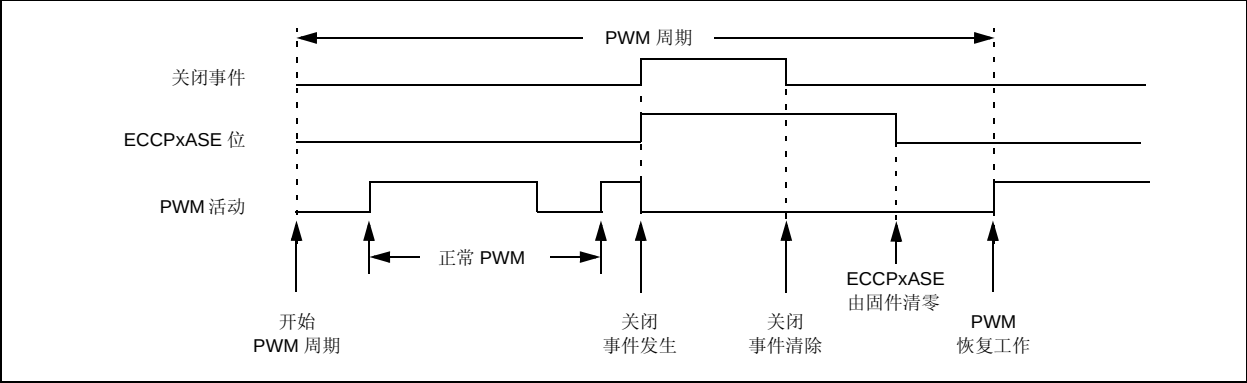
0 = 清零

x = 未知

bit 7	ECCPxASE: ECCP 自动关闭事件状态位 1 = 发生了关闭事件；ECCP 输出为关闭状态 0 = ECCP 输出工作
bit 6-4	ECCPxAS<2:0>: ECCP 自动关闭源选择位 000 = 禁止自动关闭 001 = 比较器 C1OUT 输出为高电平 010 = 比较器 C2OUT 输出为高电平 011 = 比较器 C1OUT 或 C2OUT 为高电平 100 = FLT0 引脚电压为 V _{IL} 101 = FLT0 引脚电压为 V _{IL} 或比较器 C1OUT 输出为高电平 110 = FLT0 引脚电压为 V _{IL} 或比较器 C2OUT 输出为高电平 111 = FLT0 引脚电压为 V _{IL} ，或者比较器 C1OUT 或比较器 C2OUT 为高电平
bit 3-2	PSSxAC<1:0>: 引脚 PxA 和 PxC 关闭状态控制位 00 = 驱动引脚 PxA 和 PxC 为 0 01 = 驱动引脚 PxA 和 PxC 为 1 1x = 引脚 PxA 和 PxC 为三态
bit 1-0	PSSxBD<1:0>: 引脚 PxB 和 PxD 关闭状态控制位 00 = 驱动引脚 PxB 和 PxD 为 0 01 = 驱动引脚 PxB 和 PxD 为 1 1x = 引脚 PxB 和 PxD 为三态

- 注**
- 1: 自动关闭条件是基于电平的信号，而不是基于边沿的信号。只要电平存在，自动关闭就将持续。
 - 2: 当自动关闭条件持续时，禁止写 ECCPxASE 位。
 - 3: 一旦自动关闭条件被移除并且 PWM 重启（通过固件或自动重启），PWM 信号将总是在下一个 PWM 周期重启。

图 18-14: PWM 自动关闭 (PxRSEN = 0, 用固件重启)



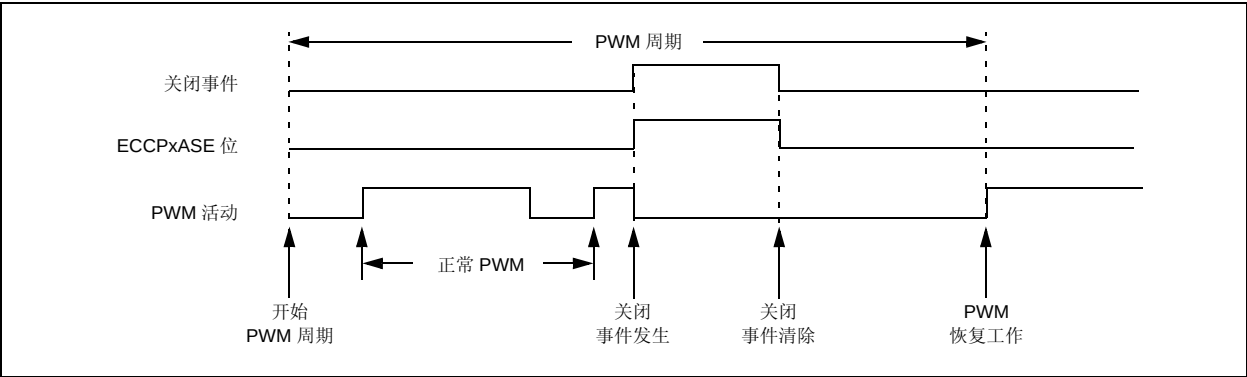
18.5.5 自动重启模式

一旦自动关闭条件被移除，增强型 PWM 可被配置为自动重启 PWM 信号。通过将 ECCPxDEL 寄存器中的 PxRSEN 位置 1 使能自动重启。

如果使能了自动重启，只要自动关闭条件有效，ECCPxASE 位将保持置 1。当自动关闭条件被移除时，ECCPxASE 位将由硬件清零并恢复正常工作。

但模块将一直等到下一个 PWM 周期开始，然后再重新使能输出引脚。这种行为使得可以在基于当前模式 PWM 控制的应用中使用自动关闭与自动重启功能。

图 18-15: PWM 自动关闭 (PxRSEN = 1, 使能自动重启)



18.5.6 可编程死区延时模式

在所有功率开关都以 PWM 频率调制的半桥应用中，功率开关关断通常比导通需要更多的时间。如果上下两个功率开关在同一时间开关（一个导通，另一个关断），那么在一段很短的时间里，两个开关可能同时导通，直到一个开关完全关断为止。在这短暂的时间中，两个功率开关中可能流过较高的电流（直通电流），将逆变桥的电源与地短路。为避免开关过程中可能会出现破坏性直通电流，通常需要延迟功率开关的导通，保证在另一个开关完全关断之后，再导通相应的功率开关。

在半桥模式下，可采用数字可编程死区延时来避免出现损坏逆变桥功率开关的直通电流。在信号从无效状态切换到有效状态时增加延时。请参见图 18-16。ECCPxDEL 寄存器（寄存器 18-3）的低 7 位以单片机指令周期（Tcy 或 4 Tosc）为单位设置延时。

图 18-16: 半桥 PWM 输出示例

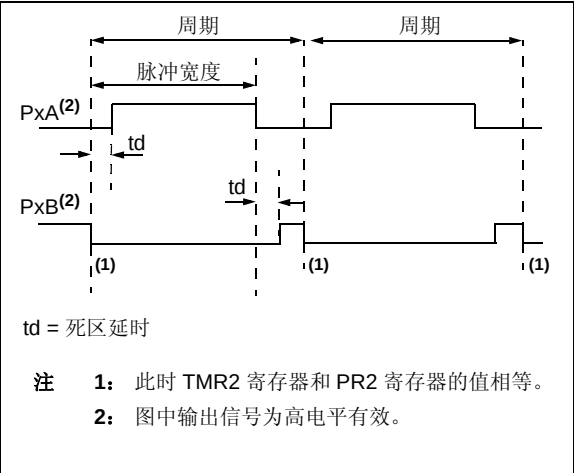
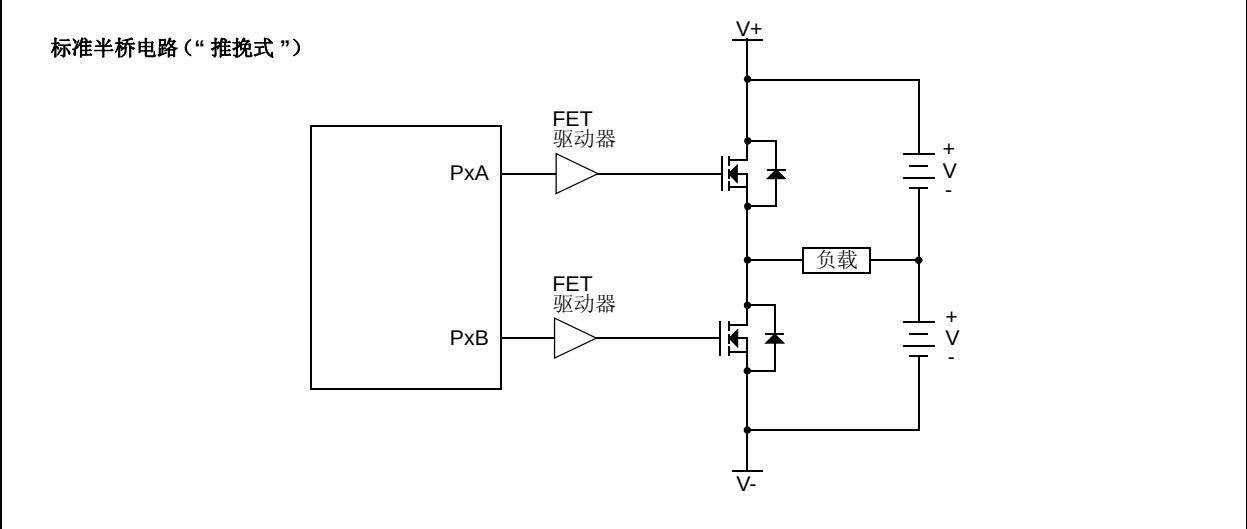


图 18-17: 半桥应用示例



PIC18F46J50 系列

寄存器 18-3: ECCPxDEL: 增强型 PWM 控制寄存器（位于快速操作存储区，地址 FBDh 和 FB7h）

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 = POR 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，一旦关闭事件消失，ECCPxASE 位自动清零；PWM 自动重启 0 = 自动关闭时，ECCPxASE 必须用软件清零以重启 PWM
bit 6-0	PxDC<6:0>: PWM 延时计数位 PxDCn = 在 PWM 信号应该转换为有效的预定时间和转换为有效的实际时间之间的 Fosc/4 (4 * Tosc) 周期数。

18.5.7 脉冲转向模式

在单输出模式下，脉冲转向允许任何 PWM 引脚为调制信号。此外，多个引脚上可以同时使用同一 PWM 信号。一旦选择了单输出模式（CCPxCON 寄存器的 CCPxM <3:2> = 11 和 PxM<1:0> = 00），通过将 PSTRxCON 寄存器的相应 STR<D:A> 位置 1，用户固件可将同一 PWM 信号加到 1、2、3 或 4 个输出引脚，如表 18-4 所示。

注:	必须将相关的 TRIS 位设为输出（0）以使能引脚输出驱动器，从而在引脚上看到 PWM 信号。

当 PWM 转向模式有效时，CCPxCON 寄存器的 CCPxM <1:0> 位将为 Px<D:A> 引脚选择 PWM 输出极性。PWM 自动关闭操作也适用于 PWM 转向模式，如第 18.5.4 节“增强型 PWM 自动关闭模式”中所述。自动关闭事件只对使能 PWM 输出的引脚有影响。

寄存器 18-4: **PSTRxCON: 脉冲转向控制寄存器**（位于快速操作存储区，地址 FBFh 和 FB9h）⁽¹⁾

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

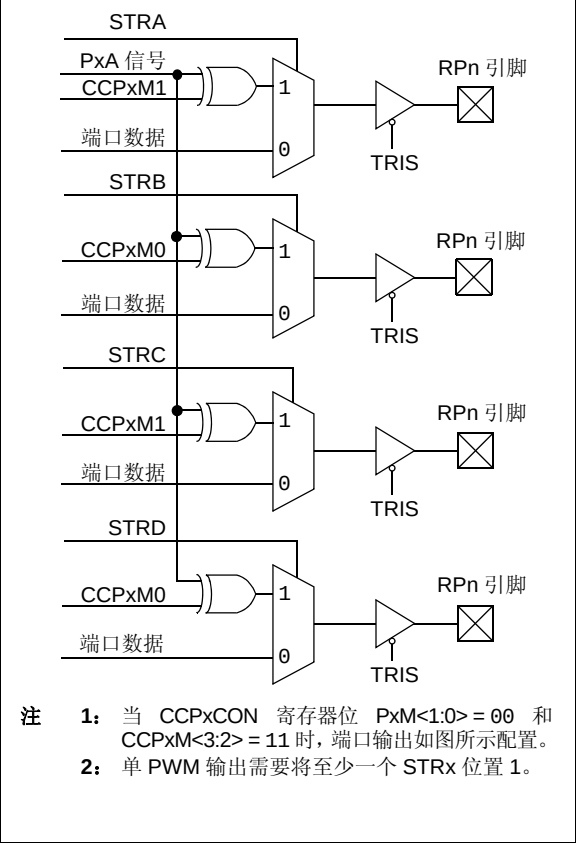
x = 未知

- bit 7-6 **CMPL<1:0>**: 互补模式输出分配转向同步位
 1 = 对于每个周期，调制输出引脚在 PxA 和 PxB 之间切换
 0 = 互补输出分配被禁止；STR<D:A> 位用于决定转向模式
- bit 5 **未实现**: 读为 0
- bit 4 **STRSYNC**: 转向同步位
 1 = 在下一个 PWM 周期发生输出转向更新
 0 = 在指令周期边界的开始发生输出转向更新
- bit 3 **STRD**: 转向使能位 D
 1 = PxD 引脚的 PWM 波形极性受 CCPxM<1:0> 控制
 0 = PxD 引脚被分配为端口引脚
- bit 2 **STRC**: 转向使能位 C
 1 = PxC 引脚的 PWM 波形极性受 CCPxM<1:0> 控制
 0 = PxC 引脚被分配为端口引脚
- bit 1 **STRB**: 转向使能位 B
 1 = PxB 引脚的 PWM 波形极性受 CCPxM<1:0> 控制
 0 = PxB 引脚被分配为端口引脚
- bit 0 **STRA**: 转向使能位 A
 1 = PxA 引脚的 PWM 波形极性受 CCPxM<1:0> 控制
 0 = PxA 引脚被分配为端口引脚

注 1: PWM 转向模式仅在 CCPxCON 寄存器位 CCPxM<3:2> = 11 和 PxM<1:0> = 00 时可用。

PIC18F46J50 系列

图 18-18: 转向简化框图



18.5.7.1 转向同步

当转向事件发生时，PSTRxCON 寄存器的 STRSYNC 位向用户提供两种选择。当 STRSYNC 位为 0 时，转向事件将发生在写 PSTRxCON 寄存器指令结束时。在这种情况下，Px<D:A> 引脚的输出信号可能是一个不完整的 PWM 波形。用户固件需要立即从引脚移除 PWM 信号时，该操作非常有用。

当 STRSYNC 位为 1 时，在下一个 PWM 周期的开始将发生有效的转向更新。此时，转向开 / 关 PWM 输出将始终产生一个完整的 PWM 波形。

图 18-19 和图 18-20 是基于 STRSYNC 设置的 PWM 转向时序图。

图 18-19: 指令结束时发生的转向事件的示例 (STRSYNC = 0)

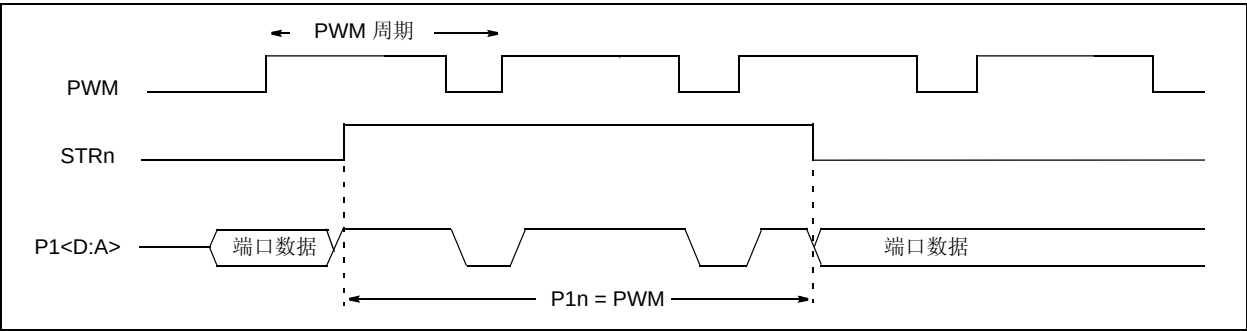
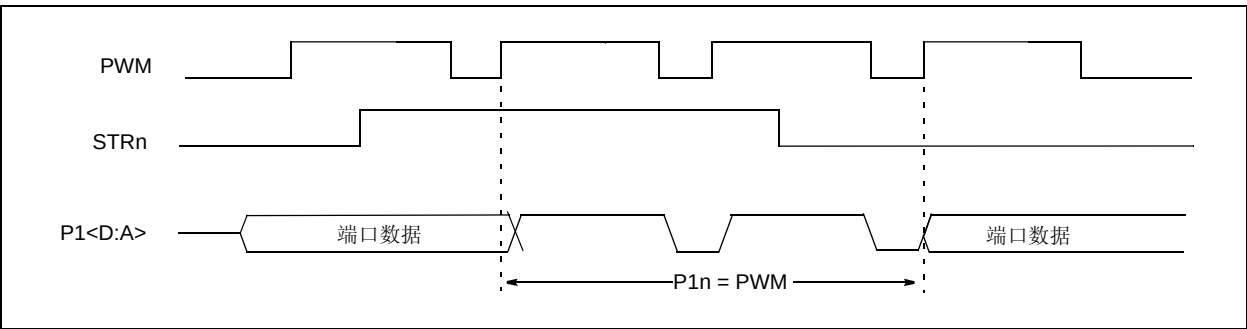


图 18-20: 指令开始时发生的转向事件的示例 (STRSYNC = 1)



18.5.8 在功耗管理模式下的操作

在休眠模式下，所有时钟源都被禁止。Timer2 不再递增，模块的状态也不会改变。如果 ECCPx 引脚正在驱动一个值，则会继续驱动该值。当器件被唤醒时，将从该状态继续。如果使能了双速启动，来自 HFINTOSC 和后分频器的初始启动频率可能不会立即稳定。

在 PRI_IDLE 模式下，主时钟将继续作为 ECCPx 模块的时钟源，保持不变。

18.5.8.1 故障保护时钟监视器（FSCM）相关操作

如果使能了故障保护时钟监视器（FSCM），时钟故障将强制器件进入 RC_RUN 功耗管理模式，并将 PIR2 寄

存器的 OSCFIF 位置 1。ECCPx 将从内部振荡器时钟源获取时钟信号，可能与主时钟的时钟频率不同。

18.5.9 复位的影响

上电复位及后续的复位都将强制所有端口为输入模式，并强制 ECCP 寄存器为复位状态。

这将强制 ECCP 模块复位为这种状态：与其他 PIC18 和 PIC16 器件中使用的早期版本非增强型 ECCP 模块兼容。

表 18-5: 与 ECCP1 模块和 TIMER1 到 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 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RABIE	TMR0IF	INT0IF	RABIF	87
RCON	IPEN	—	—	RI	TO	PD	POR	BOR	90
PIR1	PMPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	87
PIE1	PMPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	91
IPR1	PMPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	91
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	91
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	91
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	91
TCLKCON	—	—	—	T1RUN	—	—	T3CCP2	T3CCP1	93
TMR4	Timer4 寄存器								93
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0	93
PR4	Timer4 周期寄存器								93
TMR1L	Timer1 寄存器的低字节								87
TMR1H	Timer1 寄存器的高字节								87
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\overline{C}$	RD16	TMR1ON	87
TMR2	Timer2 寄存器								87
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	87
PR2	Timer2 周期寄存器								87
TMR3L	Timer3 寄存器的低字节								87
TMR3H	Timer3 寄存器的高字节								87
T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	T3SYN $\overline{C}$	RD16	TMR3ON	87
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								87
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								87
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	87
ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	87
ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	264

图注： — = 未实现，读为 0。ECCP 操作期间不使用阴影单元。

注 1： 这些位仅在 44 引脚器件上可用。

PIC18F46J50 系列

注:

19.0 主同步串行口（MSSP）模块

主同步串行口（MSSP）模块是用于同其他外设或单片机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、ADC、DAC 和许多其他类型的集成电路。

19.1 主 SSP（MSSP）模块概述

MSSP 模块有以下两种工作模式：

- 串行外设接口（SPI）
- I²C™
 - 完全的主模式
 - 从模式（支持广播地址呼叫）

I²C 接口硬件上支持以下模式：

- 主模式
- 多主模式
- 带 5 位和 7 位地址掩码的从模式（具有用于 10 位和 7 位寻址的地址掩码）

PIC18F46J50 系列的所有成员都有两个 MSSP 模块，称为 MSSP1 和 MSSP2。这两个模块独立工作：

- PIC18F4XJ50 器件——两个模块都可以配置为 I²C 或 SPI 通信
- PIC18F2XJ50 器件：
 - MSSP1 可用于 I²C 或 SPI 通信
 - MSSP2 只能用于 SPI 通信

所有与 MSSP1 模块相关的 SPI 和 I²C I/O 功能都硬映射到特定的 I/O 引脚。

对于 MSSP2 功能：

- SPI I/O 功能（SDO2、SDI2、SCK2 和 $\overline{SS}2$ ）全部都通过外设引脚选择（PPS）模块进行映射。这些功能可以配置为使用任意的 RPN 可重映射引脚，如第 10.7 节“**外设引脚选择（PPS）**”中所述。
- I²C 功能（SCL2 和 SDA2）的引脚位置是固定的。

在所有 PIC18F46J50 系列器件上，SPI DMA 功能只能与 MSSP2 一起使用。第 19.4 节“**SPI DMA 概述**”中描述了 SPI DMA 功能。

注： 在本节中，在所有工作模式下，通常指的 MSSP 模块都可以解释为 MSSP1 或 MSSP2。在需要时，寄存器名称和模块 I/O 信号使用通用标识符“x”（数字）来识别某个特定模块。控制位名称没有区别。

19.3.1 寄存器

MSSP 模块有四个寄存器用于 SPI 工作模式。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)
- MSSPx 移位寄存器 (SSPxSR) —— 不可直接访问

SSPxCON1 和 SSPxSTAT 是 SPI 工作模式下的控制寄存器和状态寄存器。SSPxCON1 寄存器是可读写的。SSPxSTAT 的低 6 位是只读的，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，可用于数据字节的写入或读出。

接收数据时，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之后，该字节会被送入 SSPxBUF，同时将中断标志位 SSPxIF 置 1。

在数据发送过程中，SSPxBUF 不是双重缓冲的，对 SSPxBUF 的写操作将同时写入 SSPxBUF 和 SSPxSR。

由于接收操作会双重缓冲 SSPxBUF 寄存器，因此以 SSPxBUF 作为操作目标的读-修改-写指令（每条指令执行两次操作）（诸如 BCF 和 COMF）无效。使用以该寄存器为操作目标的标准指令（MOVWF、MOVF（目标 = WREG）和 MOVFF）来读写 SSPxBUF（每条指令执行一次操作）。

同样道理，当使用在线调试器调试时，执行会读取 SSPxBUF 的操作（例如调试 Watch）时，读到的值有可能并非应用程序代码想要接收的数据。

寄存器 19-1: SSPxSTAT: MSSPx 状态寄存器 (SPI 模式) (位于快速操作存储区, 地址 FC7h 和 F73h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE ⁽¹⁾	D/A	P	S	R/W	UA	BF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **SMP:** 采样位
SPI 主模式:
 1 = 在数据输出时间的末端采样输入数据
 0 = 在数据输出时间的中间采样输入数据
SPI 从模式:
 当 SPI 工作在从模式时, 必须将 SMP 清零。

bit 6 **CKE:** SPI 时钟选择位 ⁽¹⁾
 1 = 时钟状态从有效转换到空闲时发送
 0 = 时钟状态从空闲转换到有效时发送

bit 5 **D/A:** 数据 / 地址位
 仅在 I²C™ 模式下使用。

bit 4 **P:** 停止位
 仅在 I²C 模式下使用; 当禁止 MSSP 模块 (SSPEN 清零) 时, 该位被清零。

bit 3 **S:** 启动位
 仅在 I²C 模式下使用。

bit 2 **R/W:** 读 / 写信息位
 仅在 I²C 模式下使用。

bit 1 **UA:** 更新地址位
 仅在 I²C 模式下使用。

bit 0 **BF:** 缓冲区满状态位
 1 = 接收完成, SSPxBUF 已满
 0 = 接收未完成, SSPxBUF 为空

注 1: 时钟状态的极性由 CKP 位 (SSPxCON1<4>) 设置。

PIC18F46J50 系列

寄存器 19-2: **SSPxCON1: MSSPx 控制寄存器 1 (SPI 模式)** (位于快速操作存储区, 地址 FC6h 和 F72h)

R/W-0	R/C-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	C = 可清零位	U = 未实现位, 读为 0
-n = POR 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 从模式:
1 = SSPxBUF 中仍保存前一数据时, 又接收到一个新的字节。如果发生溢出, SSPxSR 中的数据会丢失。溢出只会发生在从模式下发生。即使只是发送数据, 用户也必须读 SSPxBUF, 以避免将溢出标志位置 1 (该位必须用软件清零)。
0 = 无溢出
- bit 5 **SSPEN:** 主同步串口使能位 ⁽²⁾
1 = 使能串口并将 SCKx、SDOx、SDIx 和 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 **SSPM<3:0>:** 主同步串口模式选择位 ⁽³⁾
0101 = SPI 从模式, 时钟 = SCKx 引脚, 禁止 SSx 引脚控制, 可将 SSx 用作 I/O 引脚
0100 = SPI 从模式, 时钟 = SCKx 引脚, 使能 SSx 引脚控制
0011 = SPI 主模式, 时钟 = TMR2 输出 /2
0010 = SPI 主模式, 时钟 = Fosc/64
0001 = SPI 主模式, 时钟 = Fosc/16
0000 = SPI 主模式, 时钟 = Fosc/4

- 注** **1:** 在主模式下, 溢出位不会被置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSPxBUF 寄存器启动的。
2: 当使能时, 必须将该引脚正确地配置为输入或输出。
3: 在此未列出的位组合被保留或仅在 I²C™ 模式下实现。

19.3.2 工作原理

初始化 SPI 时需要指定几个选项。可以通过编程相应的控制位（SSPxCON1<5:0> 和 SSPxSTAT<7:6>）来指定这些选项。这些控制位用于指定以下选项：

- 主模式（SCKx 作为时钟输出）
- 从模式（SCKx 作为时钟输入）
- 时钟极性（SCKx 的空闲状态）
- 数据输入采样阶段（数据输出时间的中间或末尾）
- 时钟边沿（在 SCKx 的上升沿 / 下降沿输出数据）
- 时钟速率（仅用于主模式）
- 从选择模式（仅用于从模式）

每个 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，先移位 MSb。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，缓冲区满（BF）检测位（SSPxSTAT<0>）和中断标志位 SSPxIF 被置 1。这种双重缓冲数据接收方式（SSPxBUF），允许在 CPU 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，任何对 SSPxBUF 寄存器的写操作都将被忽略，并且写冲突检测位 WCOL（SSPxCON1<7>）将被置 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。

注： 为确保应用软件能接收有效数据，在下一个要发送的数据字节写入 SSPxBUF 之前，读取 SSPxBUF 中现有的数据。应用软件应遵循该过程，即使 SSPxBUF 当前的内容并不重要。

缓冲区满位 BF（SSPxSTAT<0>）用于表示何时 SSPxBUF 装入了接收到的数据（发送完成）。当 SSPxBUF 中的数据被读取后，BF 位即被清零。如果 SPI 仅作为一个发送器，则不必理会该数据。通常，可用 MSSP 中断来判断发送 / 接收是否已完成。如果不打算使用中断方法，用软件查询的方法同样可确保不会发生写冲突。

例 19-1 说明了如何为 SSPxBUF（SSPxSR）装入数据，以进行数据发送。

不能直接读写 SSPxSR 寄存器，只能通过寻址 SSPxBUF 寄存器来访问。此外，SSPxSTAT 寄存器用于指示各种状态条件。

19.3.3 漏极开路输出选项

用于 SDOx 输出和 SCKx 时钟引脚的驱动器可以有选择地配置为漏极开路输出。此功能使得可以通过外部上拉电阻将引脚上的电平上拉到较高电平，前提是 SDOx 或 SCKx 引脚不与 ANx 模拟功能复用。这使输出无需额外的电平转换器就可以与外部电路进行通信。更多详细信息，请参见第 10.1.4 节“漏极开路输出”。

漏极开路输出选项由 SPI2OD 和 SPI1OD 位（ODCON3<1:0>）控制。SPIxOD 位置 1 时，可以将 SDOx 和 SCKx 引脚配置为相应的漏极开路操作。

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

PIC18F46J50 系列

19.3.4 使能 SPI I/O

要使用 SPI，MSSP 使能位 **SSPEN** (**SSPxCON1<5>**) 必须置 1。要复位或重新配置 SPI 模式，要先将 **SSPEN** 位清零，重新初始化 **SSPxCON1** 寄存器，然后将 **SSPEN** 位置 1。这将把 **SDIx**、**SDOx**、**SCKx** 和 **SSx** 引脚配置为串口引脚。要将引脚用作串口功能，应当在将 **SSPEN** 位置 1 之前正确初始化相应的 **TRIS** 位、**ANCON/PCFG** 位和外设引脚选择寄存器（如果使用 **MSSP2**）。

典型的 SPI 串口初始化过程如下：

- 初始化 **ODCON3** 寄存器（可选的漏极开路输出控制）
- 初始化可重映射引脚功能（如果使用 **MSSP2**，请参见第 10.7 节“外设引脚选择（PPS）”）
- 将 **SCKx LAT** 值初始化为所需的空闲 **SCK** 电平（如果是主器件）
- 初始化 **SCKx ANCON/PCFG** 位（如果是从模式，并与 **ANx** 功能复用）
- 将 **SCKx TRIS** 位初始化为输出（主模式）或输入（从模式）
- 初始化 **SDIx ANCON/PCFG** 位（如果 **SDIx** 与 **ANx** 功能复用）
- 初始化 **SDIx TRIS** 位
- 初始化 **SSx ANCON/PCFG** 位（如果是从模式，并与 **ANx** 功能复用）
- 初始化 **SSx TRIS** 位（从模式）
- 初始化 **SDOx TRIS** 位
- 初始化 **SSPxSTAT** 寄存器
- 初始化 **SSPxCON1** 寄存器
- 将 **SSPEN** 位置 1 以使用模块

对于不需要的 **MSSP1** 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改写。如果不使用一些个别的 **MSSP2** 串口功能，则可以将它们保留为不映射。

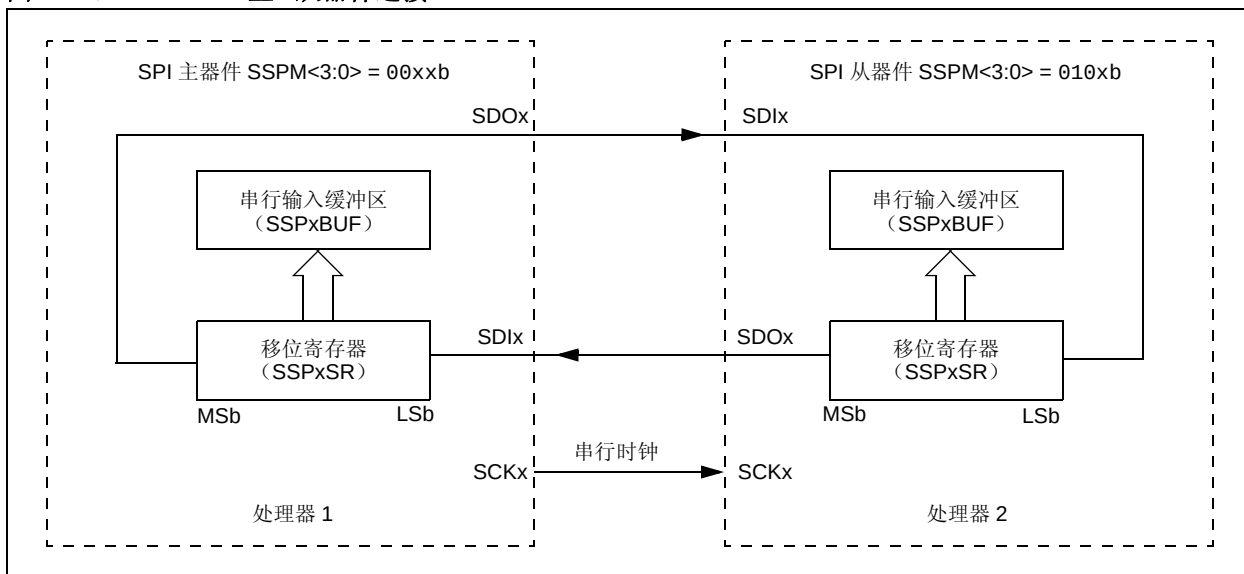
注： 在 SPI 主模式下使用 **MSSP2** 时，必须在 **PPS** 模块中将 **SCK2** 功能配置输出和输入。**SCK2** 必须初始化为输出引脚（通过向一个 **RPORx** 寄存器中写入 0x0A）。此外，还必须通过初始化 **RPINR22** 寄存器将 **SCK2IN** 映射到相同引脚。如果未能将 **SCK2/SCK2IN** 初始化为输出和输入，那么模块将无法在 **SDI2** 引脚上接收数据，因为模块使用 **SCK2IN** 信号来锁存接收到的数据。

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。主控制器（处理器 1）通过发送 **SCKx** 信号来启动数据传输。数据在编程设定的时钟边沿从两个移位寄存器移出，并在相反的时钟边沿锁存。必须将两个处理器的时钟极性（**CKP**）设置为相同，这样就可以同时收发数据。数据是否有意义（或无效数据），取决于应用软件。这就导致以下三种数据传输情形：

- 主器件发送有效数据——从器件发送无效（Dummy）数据
- 主器件发送有效数据——从器件发送有效数据
- 主器件发送无效数据——从器件发送有效数据

图 19-2: SPI 主 / 从器件连接



19.3.6 主模式

因为由主器件控制 SCK_x 信号，所以它可以在任意时刻启动数据传输。主器件根据软件协议确定从器件（图 19-2 中的处理器 2）在何时广播数据。

在主模式下，一写入 $SSPxBUF$ 寄存器就发送或接收数据。如果只打算将 SPI 作为接收器，则可以禁止 SDO_x 输出（将其编程设置为输入）。 $SSPxSR$ 寄存器按设置的时钟速率，对 SDI_x 引脚上的信号连续移入。每接收到一个字节，就将其装入 $SSPxBUF$ 寄存器，就像接收到普通字节一样（中断和状态位相应置 1）。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。

注： 为了避免在主模式下丢失数据，必须在两次发送之间执行 $SSPxBUF$ 读操作，以清零缓冲区满（BF）检测位。

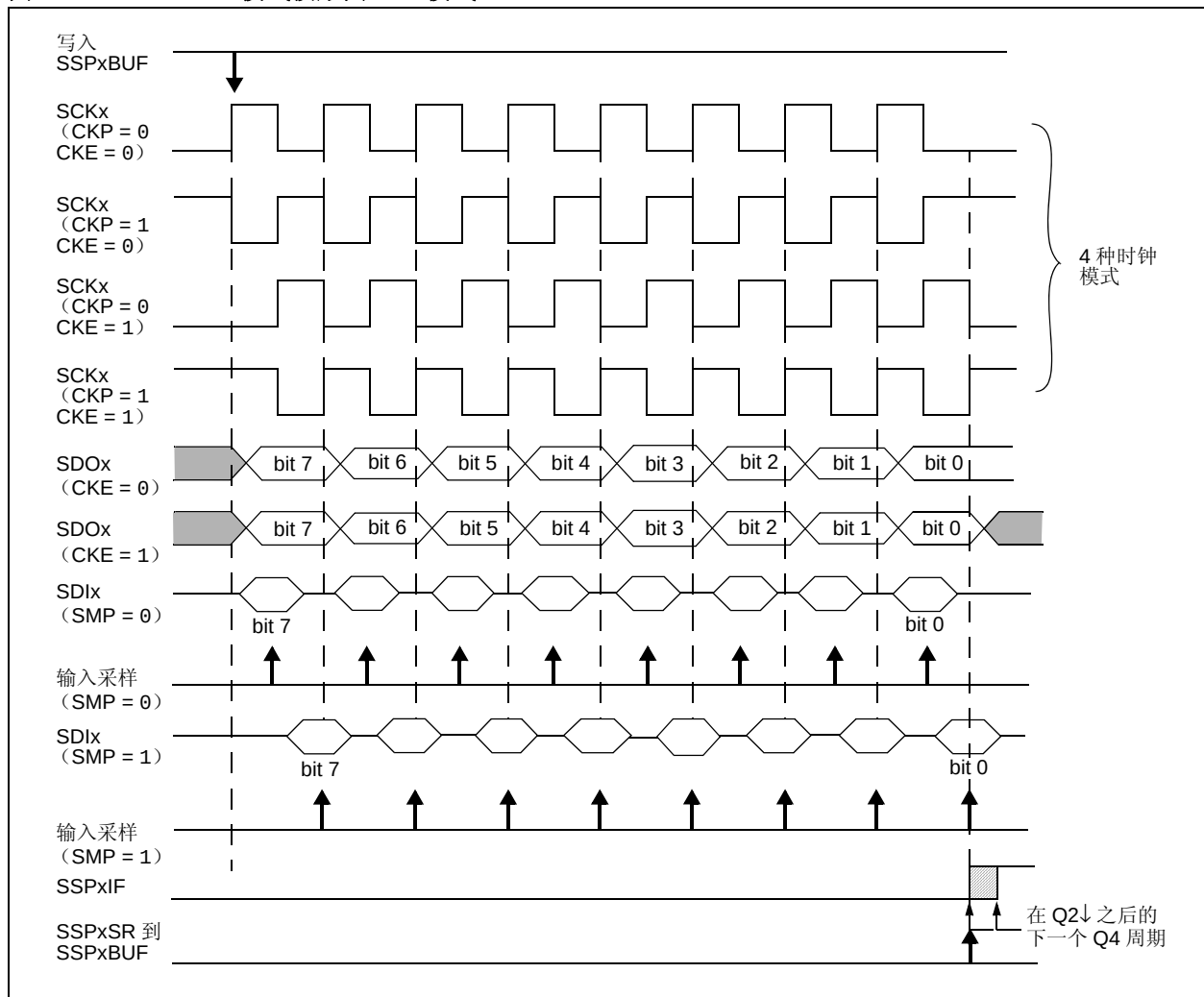
可通过对 CKP 位（ $SSPxCON1<4>$ ）进行适当的编程来选择 CKP 。图 19-3、图 19-5 和图 19-6 给出了 SPI 通信的波形图，其中最高有效字节（MSB）先发送。在主模式下，SPI 时钟速率（比特率）可由用户编程设定为以下几种之一：

- $F_{osc}/4$ （或 T_{CY} ）
- $F_{osc}/16$ （或 $4 \cdot T_{CY}$ ）
- $F_{osc}/64$ （或 $16 \cdot T_{CY}$ ）
- Timer2 输出 /2

使用 Timer2 输出 /2 选项时，周期寄存器 2（PR2）可用于确定 SPI 比特率。但是，在该模式下，只有 PR2 值 0x01 至 0xFF 是有效的。

图 19-3 给出了主模式的波形图。当 CKE 位置 1 时， SDO_x 数据在 SCK_x 上出现时钟边沿前一直有效。图中所示的输入采样的变化由 SMP 位的状态反映。图中给出了将接收到的数据装入 $SSPxBUF$ 的时刻。

图 19-3: SPI 模式波形图（主模式）



PIC18F46J50 系列

19.3.7 从模式

在从模式下，当 SCKx 引脚上出现外部时钟脉冲时发送和接收数据。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。

在休眠模式下，从器件仍可发送 / 接收数据。当接收到一个字节时，器件可配置为从休眠状态唤醒。

19.3.8 从选择同步

SSx 引脚允许器件工作于同步从模式。SPI 必须处于从模式，并使能 SSx 引脚控制 (SSPxCON1<3:0> = 04h)。当 SSx 引脚为低电平时，使能数据的发送和接收，同时 SDOx 引脚被驱动。当 SSx 引脚变为高电平时，即使是

在字节的发送过程中，也不再驱动 SDOx 引脚，而是变成悬空输出状态。可根据应用需要，在 SDOx 引脚上外接上拉 / 下拉电阻。

- 注 1:** 当 SPI 处于从模式且 SSx 引脚控制使能 (SSPxCON1<3:0> = 0100) 时，如果 SSx 引脚设置为 VDD，SPI 模块将会复位。

2: 如果 SPI 工作在从模式下并且 CKE 置 1，则必须使能 SSx 引脚控制。

当 SPI 模块复位后，位计数器被强制为 0。这是通过强制将 SSx 引脚拉为高电平或将 SSPEN 位清零来实现的。

可将 SDOx 引脚和 SDIx 引脚相连来仿真二线制通信。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。因为 SDIx 不会引起总线冲突，所以可以一直将其保留为输入 (SDIx 功能)。

图 19-4: 从同步波形图

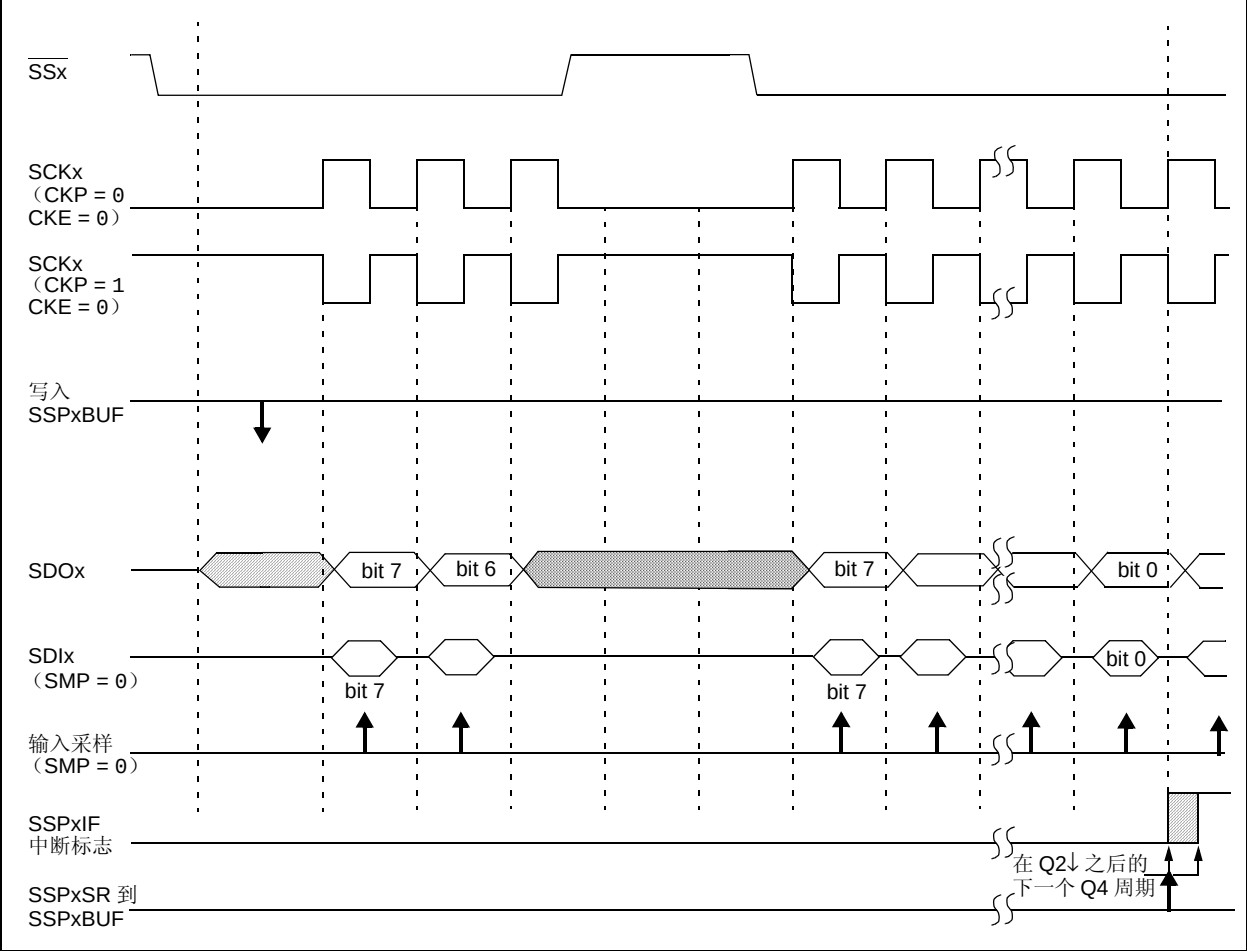


图 19-5: SPI 模式波形图 (从模式, CKE = 0)

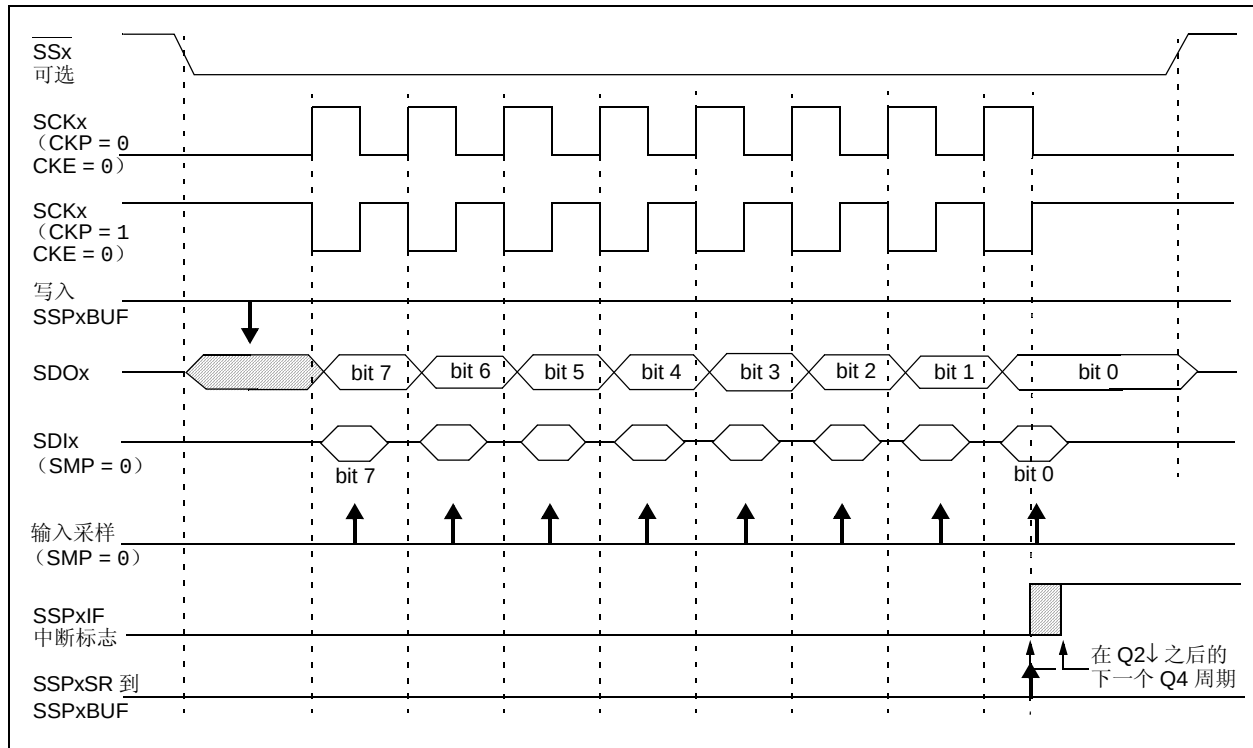
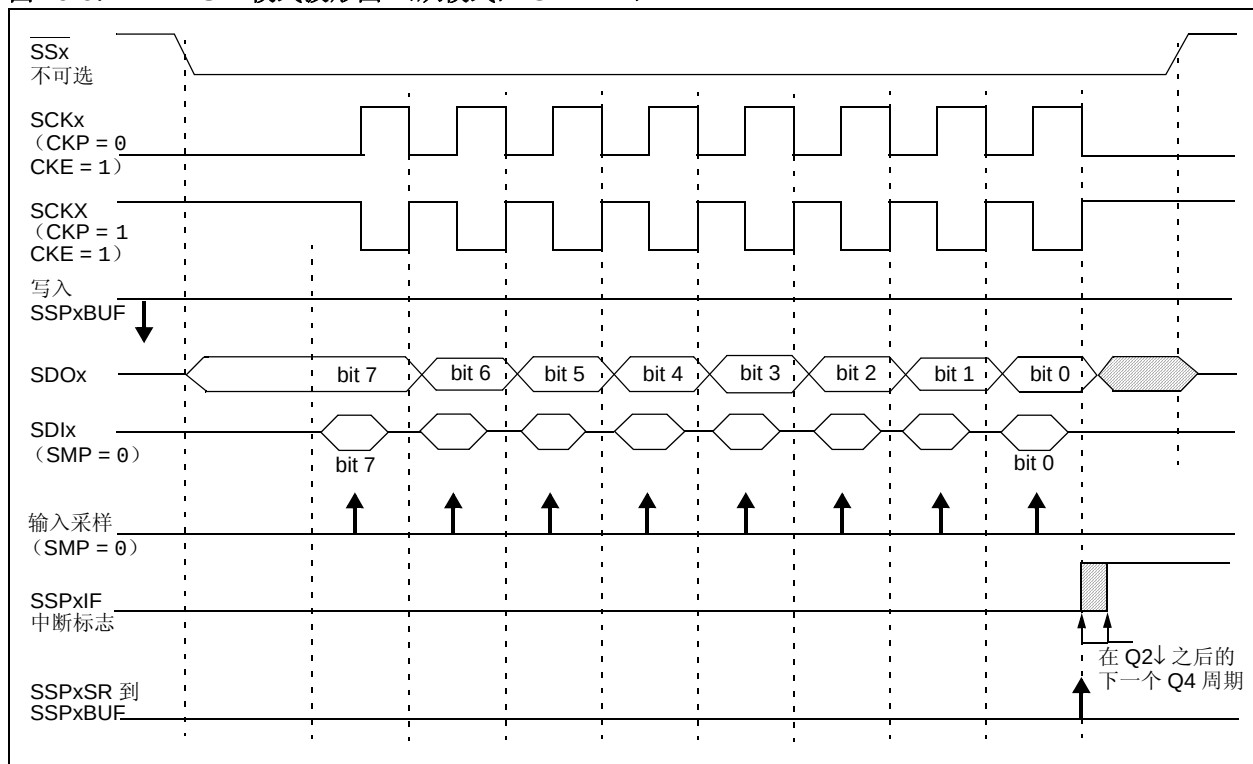


图 19-6: SPI 模式波形图 (从模式, CKE = 1)



PIC18F46J50 系列

19.3.9 在功耗管理模式下的操作

在 SPI 主模式下，模块时钟速度可能与全功耗模式下的不同；处于休眠模式时，所有时钟都停止。

在空闲模式下，需要为外设提供一个时钟。该时钟可能来自主时钟源、辅助时钟源（Timer1 振荡器）或 INTOSC 时钟源。更多信息，请参见第 3.5 节“时钟源与振荡器切换”。

在大多数情况下，主器件为 SPI 数据提供的时钟速度并不重要；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，应该禁止 MSSP 中断。

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 / 接收将保持此停止状态。当器件返回到运行模式后，MSSP 模块将重新开始发送和接收数据。

在 SPI 从模式下，SPI 发送 / 接收移位寄存器与器件异步工作。这可使器件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 位数据全部接收到后，MSSP 中断标志位将置 1，并且如果允许中断的话，器件被唤醒。

19.3.10 复位的影响

复位会禁止 MSSP 模块并终止当前的数据传输。

19.3.11 总线模式兼容性

表 19-1 给出了标准 SPI 模式与 CKP 和 CKE 控制位状态之间的兼容性。

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

注：还有一个 SMP 位用来控制数据何时被采样。

19.3.12 SPI 时钟速度和模块相互关系

因为 MSSP1 和 MSSP2 是彼此独立的模块，它们可以以不同的数据速率同时工作。将 SSPxCON1 寄存器的 SSPM<3:0> 位置 1 可确定对应模块的速率。

有一种例外情况就是两个模块都在主模式下使用 Timer2 作为时基的时候。在这种情况下，任何对 Timer2 模块工作的更改都会对两个 MSSP 模块造成相同的影响。如果每个模块需要不同的比特率，用户应该为一个模块选择其他三种时基选项之一。

表 19-2: 与 SPI 操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽²⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽²⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP ⁽²⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	71
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	72
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	72
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	72
SSP1BUF	MSSP1 接收缓冲区 / 发送寄存器								70
SSPxCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	70
SSPxSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	70
SSP2BUF	MSSP2 接收缓冲区 / 发送寄存器								73
ODCON3 ⁽¹⁾	—	—	—	—	—	—	SPI2OD	SPI1OD	74

图注: SPI 模式下的 MSSP 模块不使用阴影单元。

注 1: 配置 SFR 与默认 SFR 在此地址重叠；仅当 WDTCON<4> = 1 时可用。

2: 这些位仅在 44 引脚器件上可用。

PIC18F46J50 系列

19.4 SPI DMA 概述

SPI DMA 模块包含允许 MSSP2 模块执行 SPI 直接存储器访问传输的控制逻辑。这使模块可以在 CPU 干预相对较少的情况下，快速发送或接收大量数据。使用 SPI DMA 模块时，MSSP2 可以直接读写通用 SRAM。未使能 SPI DMA 模块时，MSSP2 可以正常工作，但没有 DMA 功能。

SPI DMA 模块包含控制逻辑、目标接收地址指针、发送源地址指针、中断管理器和用于设置每次 DMA 传输大小的字节计数寄存器。DMA 模块可用于所有 SPI 主模式和从模式，并支持半双工和全双工传输。

19.4.1 I/O 引脚注意事项

当使能时，SPI DMA 模块会使用 MSSP2 模块。与 MSSP2 相关的所有 SPI 相关输入和输出信号都通过外设引脚选择模块进行映射。使用 SPI DMA 模块之前，需要执行相应的初始化过程，如第 19.4.6 节“使用 SPI DMA 模块”中所述。分配给 SDO2 和 SCK2 功能的输出引脚可以选择配置为漏极开路输出，如用于本章中提到的电平转换操作。

19.4.2 RAM 至 RAM 复制操作

尽管 SPI DMA 模块主要的用途是 SPI 通信，但也可用来执行 RAM 至 RAM 复制操作。要执行这样的操作，应将模块配置为全双工主模式，但要将 SDO2 输出和 SDI2 输入功能分配至 PPS 模块中的同一个 RPn 引脚，将 SCK2 输出和 SCK2 输入页分配至同一个 RPn 引脚（不同于 SDO2 和 SDI2 使用的引脚）。这将允许模块在环回模式下工作，提供 RAM 复制功能。

19.4.3 空闲和休眠注意事项

在单片机处于空闲模式时，SPI DMA 模块保持完全正常工作。

在常规休眠期间，SPI DMA 模块不工作，所以不应使用它。为了避免破坏数据传输，在将单片机置为休眠模式之前，用户固件应当通过查询 DMACON1 寄存器中的 DMAEN 位，小心确保挂起的 DMA 操作已完成。

在 SPI 从模式下，处于休眠模式时，MSSP2 模块能够发送和/或接收一个字节的数据。这使得可以将 PIR3 寄存器中的 SSP2IF 标志用作唤醒源。当 DMAEN 位清零时，SPI DMA 模块实际上会被禁止，MSSP2 模块可以正常工作，但没有 DMA 功能。如果 DMAEN 位在进入休眠模式之前清零，则仍然可以使用 SSP2IF 作为唤醒源，而不会丢失数据。

在深度休眠时，MSSP2 和 SPI DMA 模块都不工作。从深度休眠中退出时，所有 I/O 引脚、与 MSSP2 和 SPI DMA 相关的寄存器将需要全部重新初始化，然后才能再次使用 SPI DMA 模块。

19.4.4 寄存器

SPI DMA 引擎通过以下特殊功能寄存器使能和控制：

• DMACON1	• DMACON2
• TXADDRH	• TXADDRL
• RXADDRH	• RXADDRL
• DMABCH	• DMABCL

19.4.4.1 DMACON1

DMACON1 寄存器用于选择 SPI DMA 模块的主工作模式。SSCON1 和 SSCON0 位用于控制从选择引脚。

在 SPI 主模式下，MSSP2 与 SPI DMA 模块一起使用时，SSDMA 可以由 DMA 模块控制，作为一个输出引脚。如果 MSSP2 用于与 SSx 引脚需要定期翻转的 SPI 从器件进行通信，SPI DMA 硬件可以自动用于每隔 1 个字节、2 个字节或 4 个字节将 SSx 置为无效。

或者，用户固件也可以使用常规的通用 I/O 引脚手动生成从选择信号（如果从器件要求）。

当 TXINC 位置 1 时，TXADDR 寄存器将在每次发送一个字节后自动递增。通过清零 TXINC 位，可以禁止自动发送地址递增。如果禁止了自动发送地址递增，则对于整个 DMA 事务，在 SDO2 上输出的每个字节（TXADDR 寄存器指向的 SRAM 的内容）将是相同的。

当 RXINC 位置 1 时，RXADDR 寄存器将在每次接收一个字节后自动递增。通过清零 RXINC 位，可以禁止自动接收地址递增。如果在全双工或半双工模式下禁止了 RXINC，则在 SDI2 上的所有传入数据字节将会覆盖 RXADDR 寄存器指向的同一存储单元。在 SPI DMA 事务完成之后，最后一个接收到的字节将驻留在 RXADDR 寄存器指向的存储单元中。

SPI DMA 模块可用于半双工仅接收通信、半双工仅发送通信，或全双工并发发送和接收操作。所有模式都可用于 SPI 主模式和 SPI 从模式配置。DUPLEX0 和 DUPLEX1 位可用于选择所需的工作模式。

DLYINTEN 位的行为会因 SPI 工作模式而存在极大差异。关于每种模式的示例行为，请参见图 19-3 至图 19-6。

SPI 从模式，DLYINTEN = 1：在该模式下，如果连续两次字节成功发送事件之间的时间大于 DMACON2 寄存器中的 DLYCYC<3:0> 位设置的值，则在传输期间将产生 SSP2IF 中断。该中断使从器件的固件可以知道主器件在两次字节发送之间占用了异常多的时间。例如，对于实现应用定义、且涉及到超时的通信协议，如果总线保持空闲的时间太长，该信息可能会很有用。当 DLYINTEN = 1 时，通常根据选定的设置产生 DLYLVL<3:0> 中断。

SPI 从模式，DLYINTEN = 0：在该模式下，禁止基于超时的中断。除了 DMACON2 寄存器中的 INTLVL<3:0> 位所指示的中断之外，SPI DMA 模块不产生任何其他 SSP2IF 中断事件。在该模式下，始终设置 DLYCYC<3:0> = 0000。

SPI 主模式，DLYINTEN = 0：DMACON2 寄存器中的 DLYCYC<3:0> 位决定附加字节间延时的时间大小，该延时是由 SPI DMA 模块在传输期间增加的。可以使用主模式 SS2 输出功能。

SPI 主模式，DLYINTEN = 1：在该模式下，硬件开销会稍稍降低，而最小字节间延时为 8 Tcy（对于 Fosc/4）、9 Tcy（对于 Fosc/16）和 15 Tcy（对于 Fosc/64）。使用该模式，可能可以获得稍高的有效 SPI 带宽。在该模式下，SS2 控制功能不能使用，并应始终禁止（DMACON1<7:6> = 00）。此外，中断产生硬件（在从模式下使用）保持在工作状态。要避免外来的 SSP2IF 中断事件，可以设置 DMACON2 延时位 DLYCYC<3:0> = 1111，并确保 SPI 串行时钟速率不低于 Fosc/64。

在 SPI 主模式下，DMAEN 位用于使能 SPI DMA 模块和启动 SPI DMA 事务。在用户固件将 DMAEN 位置 1 之后，DMA 硬件将根据所使用的配置开始发送和 / 或接收数据字节。在 SPI 从模式下，将 DMAEN 位置 1 即完成了准备 SPI DMA 模块进行通信所需要的初始化步骤（通信仍然必须由主器件启动）。

为了避免可能的数据损坏，一旦将 DMAEN 位置 1 之后，用户固件不应尝试修改与 MSSP2 或 SPI DMA 相关的任何寄存器，DMACON2 寄存器中的 INTLVL 位除外。

如果用户固件希望暂停正在进行的 DMA 事务，固件可以手动清零 DMAEN 位。如果当前正在发送一个字节，则清零 DMAEN 位时，不会立即暂停正在发送的字节。在 MSSP2 和 SPI DMA 模块恢复为它们的空闲条件之前，将会完成当前正在处理的任何字节。如果用户固件清零 DMAEN 位，则 TXADDR、RXADDR 和 DMABC 寄存器将不再更新，DMA 模块不再对 SRAM 进行任何读写；因此，状态信息可能会丢失。

PIC18F46J50 系列

寄存器 19-3: DMACON1: DMA 控制寄存器 1（位于快速操作存储区，地址 F88h）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSCON1	SSCON0	TXINC	RXINC	DUPLEX1	DUPLEX0	DLYINTEN	DMAEN
bit 7							bit 0

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

bit 7-6	SSCON<1:0> : $\overline{\text{SSDMA}}$ 输出控制位（仅用于主模式） 11 = $\overline{\text{SSDMA}}$ 置为有效的时间为 4 个字节的时间；DLYINTEN 始终复位为低电平 01 = $\overline{\text{SSDMA}}$ 置为有效的时间为 2 个字节的时间；DLYINTEN 始终复位为低电平 10 = $\overline{\text{SSDMA}}$ 置为有效的时间为 1 个字节的时间；DLYINTEN 始终复位为低电平 00 = $\overline{\text{SSDMA}}$ 不由 DMA 模块控制；DLYINTEN 位可软件编程
bit 5	TXINC : 发送地址递增使能位 允许在进行传输时递增发送地址。 1 = 发送地址从 TXADDR<11:0> 的初始值开始递增 0 = 发送地址始终设置为 TXADDR<11:0> 的初始值
bit 4	RXINC : 接收地址递增使能位 允许在进行传输时递增接收地址。 1 = 接收地址从 RXADDR<11:0> 的初始值开始递增 0 = 接收地址始终设置为 RXADDR<11:0> 的初始值
bit 3-2	DUPLEX<1:0> : 发送 / 接收工作模式选择位 10 = SPI DMA 以全双工模式工作，同时发送和接收数据 01 = DMA 以半双工模式工作，仅发送数据 00 = DMA 以半双工模式工作，仅接收数据
bit 1	DLYINTEN : 延时中断允许位 允许在以下情况下调用中断：最近一次完成传输之后，经过的 Tcy 周期数达到了 DLYCYC<2:0> 中指定的周期数。 1 = 允许中断，SSCON<1:0> 必须设置为 00 0 = 禁止中断
bit 0	DMAEN : DMA 操作启动 / 停止位 该位由用户软件置 1，用于启动 DMA 操作。当 DMA 操作完成或中止时，它由 DMA 引擎复位为零。 1 = DMA 正在进行操作 0 = DMA 不在进行操作

19.4.4.2 DMACON2

DMACON2 寄存器包含用于控制中断生成和字节间延时行为的控制位。INTLVL<3:0> 位用于选择何时产生 SSP2IF 中断。DLYCYC<3:0> 位的功能取决于 SPI 工作模式（主 / 从）和 DLYINTEN 设置。在 SPI 主模式下，DLYCYC<3:0> 位可用于控制在传输两个字节之间，模块

处于空闲的时间。默认情况下，硬件需要至少有以下延时：8 Tcy（对于 Fosc/4）、9 Tcy（对于 Fosc/16）和 15 Tcy（对于 Fosc/64）。还可以添加 DLYCYC 位指定的延时。在 SPI 从模式下，可以选择使用 DLYCYC<3:0> 位来触发基于附加超时的中断。

寄存器 19-4: DMACON2: DMA 控制寄存器 2（位于快速操作存储区，地址 F86h）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DLYCYC3	DLYCYC2	DLYCYC1	DLYCYC0	INTLVL3	INTLVL2	INTLVL1	INTLVL0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-4

DLYCYC<3:0>: 延时周期选择位

当 DLYINTEN = 0 时，这些位指定为进行下一次传输而再次写入 SSP2BUF 寄存器之前，以 Tcy 周期数表示的附加延时（硬件所需的基本延时外的额外延时）。当 DLYINTEN = 1 时，这些位指定从最近一次完成的传输开始，在调用 CPU 中断之前，以 Tcy 周期数表示的延时。在这种情况下，再次写入 SSP2BUF 寄存器之前的附加延时为 1 个 Tcy+（硬件所需的基本延时）。

1111 = 以指令周期数表示的延时为 2,048 个周期

1110 = 以指令周期数表示的延时为 1,024 个周期

1101 = 以指令周期数表示的延时为 896 个周期

1100 = 以指令周期数表示的延时为 768 个周期

1011 = 以指令周期数表示的延时为 640 个周期

1010 = 以指令周期数表示的延时为 512 个周期

1001 = 以指令周期数表示的延时为 384 个周期

1000 = 以指令周期数表示的延时为 256 个周期

0111 = 以指令周期数表示的延时为 128 个周期

0110 = 以指令周期数表示的延时为 64 个周期

0101 = 以指令周期数表示的延时为 32 个周期

0100 = 以指令周期数表示的延时为 16 个周期

0011 = 以指令周期数表示的延时为 8 个周期

0010 = 以指令周期数表示的延时为 4 个周期

0001 = 以指令周期数表示的延时为 2 个周期

0000 = 以指令周期数表示的延时为 1 个周期

bit 3-0

INTLVL<3:0>: 水印中断允许位

这些位指定待传输（发送和 / 或接收）的剩余数据量，在剩余数据量达到该值时将产生中断。

1111 = 待传输的剩余数据量为 576 字节

1110 = 待传输的剩余数据量为 512 字节

1101 = 待传输的剩余数据量为 448 字节

1100 = 待传输的剩余数据量为 384 字节

1011 = 待传输的剩余数据量为 320 字节

1010 = 待传输的剩余数据量为 256 字节

1001 = 待传输的剩余数据量为 192 字节

1000 = 待传输的剩余数据量为 128 字节

0111 = 待传输的剩余数据量为 67 字节

0110 = 待传输的剩余数据量为 32 字节

0101 = 待传输的剩余数据量为 16 字节

0100 = 待传输的剩余数据量为 8 字节

0011 = 待传输的剩余数据量为 4 字节

0010 = 待传输的剩余数据量为 2 字节

0001 = 待传输的剩余数据量为 1 字节

0000 = 传输完成

PIC18F46J50 系列

19.4.4.3 DMABCH 和 DMABCL

DMABCH 和 DMABCL 寄存器对构成一个 10 位的字节计数寄存器；对于每个 DMA 事务，SPI DMA 模块使用这个寄存器来发送 / 接收最多 1,024 个字节。当 DMA 模块在运行时（DMAEN = 1），DMA 字节计数寄存器会在发送 / 接收每个字节之后递减。在完成最后一个字节之后，DMA 事务将停止，硬件将自动清零 DMAEN 位。在 DMA 事务完成之后，DMABC 寄存器将读为 0x000。

在通过将 DMAEN 位置 1 而启动 DMA 事务之前，用户固件应在 DMABCH/DMABCL 寄存器中装入相应的值。DMABC 是从零开始计数计数器，所以将发送的实际字节数遵循公式 19-1。

例如，如果用户固件要在一个事务中发送 7 个字节，则 DMABC 中应装入 006h。类似地，如果用户固件要发送 1,024 个字节，DMABC 中应装入 3FFh。

公式 19-1: 对于给定 DMABC 发送的字节数

$$\text{Bytes}_{\text{XMIT}} \equiv (\text{DMABC} + 1)$$

19.4.4.4 TXADDRH 和 TXADDRL

TXADDRH 和 TXADDRL 寄存器对一起构成一个 12 位的发送源地址指针寄存器。在使用 TXADDR 的模式下（全双工和半双工发送），TXADDR 将在发送每个字节之后递增。发送的数据字节将从 TXADDR 寄存器指向的存储单元中获取。在发送期间，DMA 模块不会修改 TXADDR 指向的存储单元的内容。

SPI DMA 模块可以从器件上的所有通用存储器读取并发送数据，包括用于 USB 端点缓冲区的存储器。SPI DMA 模块不能用于读取 Bank 14 和 Bank 15 中包含的特殊功能寄存器（SFR）。

19.4.4.5 RXADDRH 和 RXADDRL

RXADDRH 和 RXADDRL 寄存器对一起构成一个 12 位的接收目标地址指针。在使用 RXADDR 的模式下（全双工和半双工接收），RXADDR 寄存器将在接收每个字节之后递增。接收的数据字节将存储在 RXADDR 寄存器指向的存储单元。

SPI DMA 模块可以将接收到的数据写入器件上的所有通用存储器，包括用于 USB 端点缓冲区的存储器。SPI DMA 模块不能用于修改 Bank 14 和 Bank 15 中包含的特殊功能寄存器。

19.4.5 中断

SPI DMA 模块会改变 SSP2IF 中断标志的行为。在常规 / 非 DMA 模式下，SSP2IF 在通过 MSSP2 模块发送 / 接收每个字节之后置 1。当 MSSP2 与 SPI DMA 模块配合使用时，SSP2IF 中断标志将根据在 DMACON2 寄存器中指定的 INTLVL<3:0> 用户选择值而置 1。当 SPI DMA 事务完全完成，且硬件将 DMAEN 位清零之后，还会产生 SSP2IF 中断条件。

当 DMA 字节计数值指示已达到所指定的 INTLVL 时，SSP2IF 标志会置 1。例如，如果 DMACON2<3:0> = 0101（剩余 16 个字节），SSP2IF 中断标志将在 DMABC 达到 00Fh 时置 1。如果用户固件之后清零 SSP2IF 中断标志，则只有所有字节全部发送且 DMA 事务完成之后，硬件才会再次将此中断标志置 1。

注： 用户固件可以在 DMA 事务正在进行时（DMAEN = 1）修改 INTLVL 位。如果选择的 INTLVL 值大于实际的剩余字节数（由 DMABC + 1 指示），SSP2IF 中断标志将立即置 1。

例如，如果 DMABC = 00Fh（表示剩余 16 个字节），且用户固件向 INTLVL<3:0> 中写入 1111（剩余 576 个字节时产生中断），SSP2IF 中断标志将立即置 1。如果用户固件清零该中断标志，则只有发生以下情况之一才会产生新的中断条件：用户固件向 INTLVL 中再次写入数据，且此时的中断级别高于实际的剩余数据级别；或者，DMA 事务完成，且 DMAEN 位清零。

注： 如果在 DMA 事务正在进行时修改了 INTLVL 位，则应当小心避免无意地更改 DLYCYC <3:0> 值。

19.4.6 使用 SPI DMA 模块

要使用并启用 SPI DMA 模块，通常需要执行以下步骤：

1. 配置将由 MSSP2 使用的 I/O 引脚。
 - a) 针对要使用的 SPI 模式，将 SCK2、SDO2、SDI2 和 SS2 功能分配给 RPn 引脚。只需要将要使用的功能分配给引脚。
 - b) 针对所需的空闲 SPI 总线状态，初始化关联的 LATx 寄存器。
 - c) 如果 SDO2 和 SCK2 上需要漏极开路输出模式（主模式），则将 ODCON3<1> 置 1。
 - d) 为所使用的每个 I/O 引脚配置相应的 TRISx 位。
2. 针对所需的 SPI 工作模式，配置并启用 MSSP2。
 - a) 通过写入 SSP2STAT 和 SSP2CON1 寄存器，选择所需的工作模式（主模式或从模式，SPI 模式 0、1、2 和 3）并配置模块。
 - b) 通过设置 SSP2CON1<5> = 1 启用 MSSP2。
3. 配置 SPI DMA 引擎。
 - a) 通过向 DMACON2 和 DMACON1 写入适当的值来选择所需的工作模式。
 - b) 初始化 TXADDRH/TXADDRL 指针（全双工或半双工仅发送模式）。
 - c) 初始化 RXADDRH/RXADDRL 指针（全双工或半双工仅接收模式）。
 - d) 使用要在下一次 SPI DMA 操作中传输的字节数初始化 DMABCH/DMABCL 字节计数器寄存器。
 - e) 将 DMAEN 位（DMACON1<0>）置 1。

在 SPI 主模式下，这将启动 DMA 事务。在 SPI 从模式下，这将完成初始化过程：模块现在已准备好，一旦主器件启动事务，模块就开始从主器件接收和 / 或向主器件发送数据。

4. 检测 SSP2IF 中断条件（PIR3<7>）。

- a) 如果配置为在 SPI DMA 事务完成时产生中断，DMAEN 位（DMACON1<0>）将清零。用户固件可以通过重复步骤 3.b 至 3.e 来为模块准备下一个事务。
- b) 如果配置为在 SPI DMA 事务完成之前产生中断，则 DMAEN 位可以仍置 1，指示事务仍然正在进行。用户固件通常使用该中断条件来开始为下一个 DMA 事务准备新数据。只有硬件清零 DMAEN 位之后（指示事务已完成），固件才能重复步骤 3.b. 至 3.e.。

例 19-2 给出了示例代码，说明使用 SPI DMA 模块执行 512 字节全双工、主模式传输的初始化过程和步骤。

PIC18F46J50 系列

例 19-2: 512 字节 SPI 主模式初始化和传输

```
                                ;For this example, let's use RP5(RB2) for SCK2,
                                ;RP4(RB1) for SD02, and RP3(RB0) for SDI2

                                ;Let's use SPI master mode, CKE = 0, CKP = 0,
                                ;without using slave select signalling.

InitSPIPins:
    movlb    0x0F                ;Select bank 15, for access to ODCON3 register
    bcf      ODCON3, SPI20D      ;Let's not use open drain outputs in this example

    bcf      LATB, RB2           ;Initialize our (to be) SCK2 pin low (idle).
    bcf      LATB, RB1           ;Initialize our (to be) SD02 pin to an idle state
    bcf      TRISB, RB1          ;Make SD02 output, and drive low
    bcf      TRISB, RB2          ;Make SCK2 output, and drive low (idle state)
    bsf      TRISB, RB0          ;SDI2 is an input, make sure it is tri-stated

                                ;Now we should unlock the PPS registers, so we can
                                ;assign the MSSP2 functions to our desired I/O pins.

    movlb    0x0E                ;Select bank 14 for access to PPS registers
    bcf      INTCON, GIE         ;I/O Pin unlock sequence will not work if CPU
                                ;services an interrupt during the sequence
    movlw    0x55                ;Unlock sequence consists of writing 0x55
    movwf    EECON2              ;and 0xAA to the EECON2 register.
    movlw    0xAA
    movwf    EECON2
    bcf      PPSCON, IOLOCK      ;We may now write to RPINRx and RPORx registers
    bsf      INTCON, GIE         ;May now turn back on interrupts if desired

    movlw    0x03                ;RP3 will be SDI2
    movwf    RPINR21             ;Assign the SDI2 function to pin RP3

    movlw    0x0A                ;Let's assign SCK2 output to pin RP4
    movwf    RPOR4               ;RPOR4 maps output signals to RP4 pin
    movlw    0x04                ;SCK2 also needs to be configured as an input on the
                                ;same pin
    movwf    RPINR22             ;SCK2 input function taken from RP4 pin
    movlw    0x09                ;0x09 is SD02 output
    movwf    RPOR5               ;Assign SD02 output signal to the RP5 (RB2) pin
    movlb    0x0F                ;Done with PPS registers, bank 15 has other SFRs

InitMSSP2:
    clrf     SSP2STAT            ;CKE = 0, SMP = 0 (sampled at middle of bit)
    movlw    b'00000000'        ;CKP = 0, SPI Master mode, Fosc/4
    movwf    SSP2CON1           ;MSSP2 initialized
    bsf      SSP2CON1, SSPEN     ;Enable the MSSP2 module

InitSPIDMA:
    movlw    b'00111010'        ;Full duplex, RX/TXINC enabled, no SSCON
    movwf    DMACON1            ;DLYINTEN is set, so DLYCYC3:DLYCYC0 = 1111
    movlw    b'11110000'        ;Minimum delay between bytes, interrupt
    movwf    DMACON2            ;only once when the transaction is complete
```

例 19-2: 512 字节 SPI 主模式初始化和传输 (续)

```
                                ;Somewhere else in our project, lets assume we have
                                ;allocated some RAM for use as SPI receive and
                                ;transmit buffers.

;                                udata      0x500
;DestBuf      res      0x200      ;Let's reserve 0x500-0x6FF for use as our SPI
;                                ;receive data buffer in this example
;SrcBuf       res      0x200      ;Lets reserve 0x700-0x8FF for use as our SPI
;                                ;transmit data buffer in this example

PrepareTransfer:
    movlw      HIGH(DestBuf)      ;Get high byte of DestBuf address (0x05)
    movwf      RXADDRH            ;Load upper four bits of the RXADDR register
    movlw      LOW(DestBuf)       ;Get low byte of the DestBuf address (0x00)
    movwf      RXADDRL           ;Load lower eight bits of the RXADDR register

    movlw      HIGH(SrcBuf)       ;Get high byte of SrcBuf address (0x07)
    movwf      TXADDRH            ;Load upper four bits of the TXADDR register
    movlw      LOW(SrcBuf)        ;Get low byte of the SrcBuf address (0x00)
    movwf      TXADDRL           ;Load lower eight bits of the TXADDR register

    movlw      0x01               ;Lets move 0x200 (512) bytes in one DMA xfer
    movwf      DMABCH             ;Load the upper two bits of DMABC register
    movlw      0xFF               ;Actual bytes transferred is (DMABC + 1), so
    movwf      DMABCL            ;we load 0x01FF into DMABC to xfer 0x200 bytes

BeginXfer:
    bsf        DMACON1, DMAEN     ;The SPI DMA module will now begin transferring
                                ;the data taken from SrcBuf, and will store
                                ;received bytes into DestBuf.

    ;Execute whatever            ;CPU is now free to do whatever it wants to
                                ;and the DMA operation will continue without
                                ;intervention, until it completes.

                                ;When the transfer is complete, the SSP2IF flag in
                                ;the PIR3 register will become set, and the DMAEN bit
                                ;is automatically cleared by the hardware.
                                ;The DestBuf (0x500-0x7FF) will contain the received
                                ;data. To start another transfer, firmware will need
                                ;to reinitialize RXADDR, TXADDR, DMABC and then
                                ;set the DMAEN bit.
```

PIC18F46J50 系列

19.5 I²C 模式

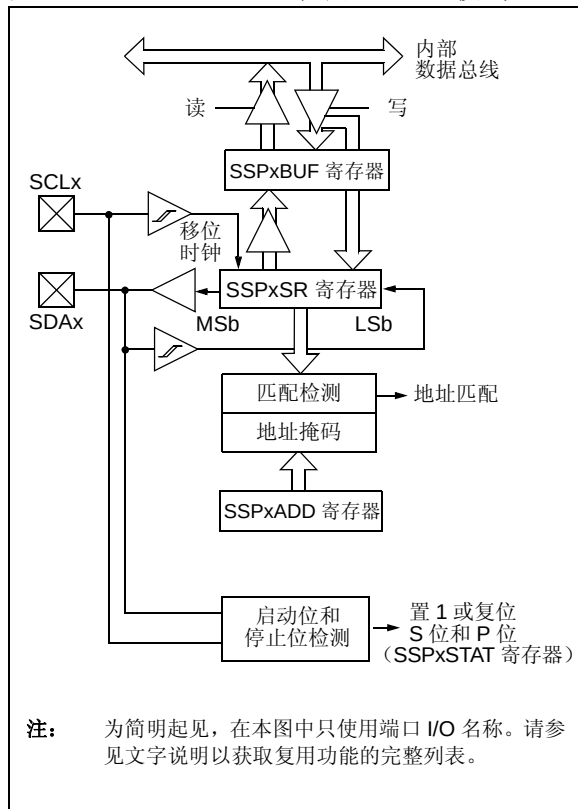
MSSP 模块工作在 I²C 模式时，可以实现所有的主和从功能（包括广播呼叫支持），并且硬件上提供启动位和停止位的中断来判断总线何时空闲（多主器件功能）。MSSP 模块实现了标准模式规范以及 7 位和 10 位寻址。

有两个引脚用于数据传输：

- 串行时钟（SCLx）——
RB4/PMA1/KBI0/SCK1/SCL1/RP7 或
RD0/PMD0/SCL2
- 串行数据（SDAx）——
RB5/PMA0/KBI1/SDI1/SDA1/RP8 或
RD1/PMD1/SDA2

用户必须通过设置相关的 TRIS 位将这些引脚配置为输入引脚。这些引脚最高能承受 5.5V 电压，因而在电压高于 V_{DD} 时，直接用于 I²C 总线工作。

图 19-7: MSSPx 框图 (I²C™ 模式)



19.5.1 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) —— 不可直接访问
- MSSPx 地址寄存器 (SSPxADD)
- MSSPx 7 位地址掩码寄存器 (SSPxMSK)

SSPxCON1、SSPxCON2 和 SSPxSTAT 是在 I²C 工作模式下的控制寄存器和状态寄存器。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、可写的。

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，可用于数据字节的写入或读出。

当 MSSP 被配置为工作在 I²C 从模式下时，SSPxADD 保存从器件的地址。当 MSSP 被配置为工作在主模式下时，SSPxADD 的低 7 位用作波特率发生器 (Baud Rate Generator, BRG) 的重载值。

当模块配置为 7 位地址掩码模式时，SSPxMSK 保存从地址掩码值。虽然它是独立的寄存器，它与 SSPxADD 共用相同的 SFR 地址；只有 SSPM<3:0> 位特别设置为允许访问时，才能访问它。更多详细信息，请参见第 19.5.3.4 节“7 位地址掩码模式”。

在接收操作中，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之后，该字节会被送入 SSPxBUF，同时将中断标志位 SSPxIF 置 1。

在数据发送过程中，SSPxBUF 不是双重缓冲的，对 SSPxBUF 的写操作将同时写入 SSPxBUF 和 SSPxSR。

寄存器 19-5: SSPxSTAT: MSSPx 状态寄存器 (I²C™ 模式) (位于快速操作存储区, 地址FC7h和F73h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P ⁽¹⁾	S ⁽¹⁾	R/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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7 **SMP:** 边沿斜率控制位
在主模式或从模式下:
 1 = 标准速度模式下禁止边沿斜率控制 (100 kHz 和 1 MHz)
 0 = 高速模式下使能边沿斜率控制 (400 kHz)
- bit 6 **CKE:** SMBus 选择位
在主模式或从模式下:
 1 = 使能 SMBus 特定输入
 0 = 禁止 SMBus 特定输入
- bit 5 **D/A:** 数据 / 地址位
在主模式下:
 保留。
在从模式下:
 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 = 表示上次检测到停止位
 0 = 上次未检测到停止位
- bit 3 **S:** 启动位 ⁽¹⁾
 1 = 表示上次检测到启动位
 0 = 上次未检测到启动位
- bit 2 **R/W:** 读 / 写信息位 ^(2,3)
在从模式下:
 1 = 读
 0 = 写
在主模式下:
 1 = 正在进行发送
 0 = 不在进行发送
- bit 1 **UA:** 更新地址位 (仅用于 10 位从模式)
 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:** 缓冲区满状态位
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 ACK 位和停止位)
 0 = SSPxBUF 为空 (不包括 ACK 位和停止位)

注 1: 该位在复位及 SSPEN 清零时被清零。

2: 该位保存最后一个地址匹配后的 R/W 位信息。该位仅在从地址匹配到下一个启动位、停止位或非 ACK 位之间有效。

3: 将该位与 SEN、RSEN、PEN、RCEN 或 ACKEN 进行逻辑或运算将指示 MSSPx 是否处于有效模式。

PIC18F46J50 系列

寄存器 19-6: SSPXCON1: MSSPX 控制寄存器 1 (I²C™ 模式) (位于快速操作存储区, 地址 FC6h 和 F72h)

R/W-0	R/C-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:	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7 **WCOL:** 写冲突检测位
在主发送模式下:
1 = 当 I²C 不满足启动发送数据的条件时, 试图向 SSPxBUF 寄存器写入数据 (必须用软件清零)
0 = 未发生冲突
在从发送模式下:
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 (主或从模式) 下:
该位是无关位。
- bit 6 **SSPOV:** 接收溢出指示位
在接收模式下:
1 = SSPxBUF 寄存器仍在保存前一字节时, 接收到一个新的字节 (必须用软件清零)
0 = 无溢出
在发送模式下:
在发送模式下, 该位是无关位。
- bit 5 **SSPEN:** 主同步串口使能位 ⁽¹⁾
1 = 使能串口并将 SDAx 和 SCLx 引脚配置为串口引脚
0 = 禁止串口并将上述引脚配置为 I/O 端口引脚
- bit 4 **CKP:** SCKx 释放控制位
在从模式下:
1 = 释放时钟
0 = 保持时钟低电平 (时钟延长); 用来确保数据建立时间
在主模式下:
在此模式下未使用。
- bit 3-0 **SSPM<3:0>:** 主同步串口模式选择位 ⁽²⁾
1111 = I²C 从模式, 10 位地址, 并允许启动位和停止位中断
1110 = I²C 从模式, 7 位地址, 并允许启动位和停止位中断
1011 = I²C 固件控制的主模式 (从器件空闲)
1001 = 装入 SSPxADD SFR 地址处的 SSPxMSK 寄存器 ^(3,4)
1000 = I²C 主模式, 时钟 = Fosc/(4 * (SSPxADD + 1))
0111 = I²C 从模式, 10 位地址
0110 = I²C 从模式, 7 位地址

- 注**
- 1: 当使能时, 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被保留或仅在 SPI 模式下实现。
 - 3: 当 SSPM<3:0> = 1001 时, 任何对 SSPxADD SFR 地址的读或写实际上访问 SSPxMSK 寄存器。
 - 4: 该模式仅在选择了 7 位地址掩码模式 (MSSPMASK 配置位为 1) 时可用。

寄存器 19-7: SSPxCON2: MSSPx 控制寄存器 2 (I²C™ 主模式) (位于快速操作存储区, 地址 FC5h 和 F71h)

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN ⁽³⁾	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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7 **GCEN:** 广播呼叫使能位 (仅用于从模式) ⁽³⁾
1 = 当 SSPxSR 中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 (仅用于主发送模式)
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 引脚上发出应答序列, 并发送 ACKDT 数据位; 由硬件自动清零
0 = 应答序列空闲
- bit 3 **RCEN:** 接收使能位 (仅用于主接收模式) ⁽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 引脚上发出停止条件; 由硬件自动清零
0 = 停止条件空闲
- bit 1 **RSEN:** 重复启动条件使能位 ⁽²⁾
1 = 在 SDAx 和 SCLx 引脚上发出重复启动条件; 由硬件自动清零
0 = 重复启动条件空闲
- bit 0 **SEN:** 启动条件使能位 ⁽²⁾
1 = 在 SDAx 和 SCLx 引脚上发出启动条件; 由硬件自动清零
0 = 启动条件空闲

- 注** **1:** 当用户在接收末尾发出一个应答序列时, 发送该值。
2: 如果 I²C 模块处于工作状态, 可能这些位不会被置 1 (无并行工作), 并且可能不会写入 SSPxBUF (或禁止写 SSPxBUF)。
3: 该位在 I²C 主模式下未实现。

PIC18F46J50 系列

寄存器 19-8: SSPxCON2: MSSPx 控制寄存器 2 (I²C™ 从模式) (位于快速操作存储区, 地址 FC5h 和 F71h)

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT ⁽²⁾	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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7 **GCEN:** 广播呼叫使能位 (仅用于从模式)
1 = 当 SSPxSR 中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 ⁽²⁾
在从模式下未使用。
- bit 5-2 **ADMSK<5:2>:** 从地址掩码选择位 (5 位地址掩码)
1 = 使能 SSPxADD 对应位的掩码
0 = 禁止 SSPxADD 对应位的掩码
- bit 1 **ADMSK1:** 从地址最低有效位掩码选择位
在 7 位寻址模式下:
1 = 仅使能 SSPxADD<1> 的掩码
0 = 仅禁止 SSPxADD<1> 的掩码
在 10 位寻址模式下:
1 = 使能 SSPxADD<1:0> 的掩码
0 = 禁止 SSPxADD<1:0> 的掩码
- bit 0 **SEN:** 启动条件使能 / 延长使能位 ⁽¹⁾
1 = 为从发送和从接收 (已使能时钟延长) 使能时钟延长
0 = 禁止时钟延长

- 注 1: 如果 I²C 模块处于工作状态, 可能这些位不会被置 1 (无并行工作), 并且可能不会写入 SSPxBUF (或禁止写 SSPxBUF)。
- 2: 该位在 I²C 从模式下未实现。

寄存器 19-9: SSPxMSK: I²C™ 从地址掩码寄存器 (7 位掩码模式) (位于快速操作存储区, 地址 FC8h 和 F74h) ⁽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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7-0 **MSK<7:0>:** 从地址掩码选择位
1 = 使能 SSPxADD 对应位的掩码
0 = 禁止 SSPxADD 对应位的掩码

- 注 1: 该寄存器与 SSPxADD 共用相同的 SFR 地址, 并且仅在某些 MSSP 工作模式下才可寻址该寄存器。更多信息, 请参见第 19.5.3.4 节 “7 位地址掩码模式”。
- 2: 在 7 位寻址时, MSK0 不用作掩码位。

19.5.2 工作原理

MSSP 模块功能通过将 MSSP 使能位 SSPEN (SSPxCON1<5>) 置 1 使能。

SSPxCON1 寄存器用于控制 I²C 工作模式。可通过设置 4 个模式选择位 (SSPxCON1<3:0>) 选择以下 I²C 模式之一：

- I²C 主模式，时钟
- I²C 从模式 (7 位地址)
- I²C 从模式 (10 位地址)
- I²C 从模式 (7 位地址)，允许启动位和停止位中断
- I²C 从模式 (10 位地址)，允许启动位和停止位中断
- I²C 固件控制的主模式，从器件空闲

如果通过将相应的 TRISB 或 TRISD 位置 1，将 SCLx 和 SDAx 引脚编程为输入引脚，则在 SSPEN 位置 1 时选择任何 I²C 模式，将强制上述引脚为漏极开路。要确保此模块正常工作，必须为 SCLx 和 SDAx 引脚外接上拉电阻。

19.5.3 从模式

在从模式下，SCLx 和 SDAx 引脚必须被配置为输入 (TRISB<5:4> 置 1)。必要时 MSSP 模块将用输出数据改写输入状态 (从发送器)。

I²C 从模式硬件总是在地址匹配时产生中断。地址掩码功能可使硬件在多个地址发生匹配时 (7 位寻址模式下最多 31 个，10 位寻址模式下最多 63 个) 产生中断。用户也可以通过模式选择位，选择使用启动位或停止位中断。

当地址匹配或在地址匹配后发送的数据被接收时，硬件会自动产生一个应答 (ACK) 脉冲，并把当前 SSPxSR 寄存器中接收到的值装入 SSPxBUF 寄存器。

只要满足以下条件之一，MSSP 模块就不会产生此 ACK 脉冲：

- 在接收到数据前，缓冲区满位 BF (SSPxSTAT<0>) 被置 1。
- 在接收到数据前，溢出位 SSPOV (SSPxCON1<6>) 被置 1。

在上述情况下，SSPxSR 寄存器的值不会装入 SSPxBUF，但 SSPxIF 位会置 1。BF 位是通过读取 SSPxBUF 寄存器清零的，而 SSPOV 位是通过软件清零的。

为确保正常工作，SCLx 时钟输入必须满足最小高电平和最小低电平时间要求。在时序参数 100 和参数 101 中给出了 I²C 规范的高低电平时间和对 MSSP 模块的具体要求。

19.5.3.1 寻址

一旦 MSSP 模块被使能，它就会等待启动条件出现。启动条件出现后，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。寄存器 SSPxSR<7:1> 的值会和 SSPxADD 寄存器的值比较，该比较是在第 8 个时钟 (SCLx) 脉冲下降沿进行的。如果地址匹配，并且 BF 位和 SSPOV 位都被清零，会发生以下事件：

1. SSPxSR 寄存器的值被装入 SSPxBUF 寄存器。
2. 缓冲区满标志位 BF 被置 1。
3. 产生 ACK 脉冲。
4. 在第 9 个 SCLx 脉冲的下降沿，MSSPx 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 (MSb) 将指定这是否是一个 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” 是该地址的高 2 位。10 位寻址模式的操作步骤如下，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 (SSPxIF、BF 和 UA 位在地址匹配时被置 1)。
2. 用地址的第二个 (低) 字节更新 SSPxADD 寄存器 (清零 UA 位并释放 SCLx 线)。
3. 读 SSPxBUF 寄存器 (清零 BF 位) 并清零标志位 SSPxIF。
4. 接收地址的第二个 (低) 字节 (SSPxIF、BF 和 UA 位置 1)。
5. 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 寄存器 (清零 BF 位) 并清零标志位 SSPxIF。
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 (清零 BF 位) 并清零标志位 SSPxIF。

PIC18F46J50 系列

19.5.3.2 地址掩码模式

将地址的某一位掩码使该位变为无关位，此时会响应两个地址并产生一个中断。可以一次掩码多个地址位，这可以极大地扩展被应答的地址的数量。

不管是否使用地址掩码，I²C 从器件的工作方式保持不变。但当使用地址掩码时，I²C 从器件能够应答多个地址并产生中断，此时需要通过检查 SSPxBUF 来判断是哪个地址引起中断。

在 I²C 从操作中，PIC18F46J50 系列器件能够使用两种不同的地址掩码模式：5 位地址掩码和 7 位地址掩码。掩码模式在器件配置中使用 MSSPMASK 配置位选择。默认的器件配置是 7 位地址掩码。

两种掩码模式都支持 7 位和 10 位地址的地址掩码。掩码模式和地址的组合为每种组合提供不同的可应答地址范围。

虽然两种掩码模式的工作方式大致相同，它们使用地址掩码的方式是不同的。

19.5.3.3 5 位地址掩码模式

正如名称所暗示，5 位地址掩码模式使用最多 5 位的地址掩码，用传入地址的 bit 5 至 bit 1 来产生可被应答的地址范围。使用 7 位寻址时，这使模块最多可以应答 31 个

地址；使用 10 位寻址时，这使模块最多可以应答 63 个地址（见例 19-3）。当编程 MSSPMASK 配置位时（0），即选择该掩码模式。

该地址模式下的地址掩码存储在 SSPxCON2 寄存器中，在 I²C 从模式下，该寄存器停止用作控制寄存器（寄存器 19-8）。在 7 位地址掩码模式下，地址掩码位 ADMSK<5:1>（SSPxCON2<5:1>）可用来掩码 SSPxADD 寄存器中对应的地址位。如果某个 ADMSK 位被置 1（ADMSK<n> = 1），则对应的地址位被忽略（SSPxADD<n> = x）。对于发出地址应答的模块，只要与没被掩码的地址位匹配就可以了。

在 10 位地址掩码模式下，ADMSK<5:2> 位可用来掩码 SSPxADD 寄存器中对应的地址位，而 ADMSK1 可以同时掩码地址的低 2 位（SSPxADD<1:0>）。如果某个 ADMSK 位是有效的（ADMSK<n> = 1），则对应的地址位被忽略（SPxADD<n> = x）。另外需要注意的是，尽管在 10 位地址掩码模式下，地址的高位也要用到 SSPxADD 寄存器中的某些位。地址掩码位对这些高位不起作用；地址掩码位只会影响地址的低位。

- 注 1: ADMSK1 掩码地址的低 2 位。

2: 地址的高 2 位不受地址掩码的影响。

例 19-3: 5 位掩码模式下的地址掩码示例

7 位寻址:

SSPxADD<7:1>= A0h (1010000) (SSPxADD<0> 假设为 0)

ADMSK<5:1> = 00111

可被应答的地址: A0h, A2h, A4h, A6h, A8h, AAh, ACh, AEh

10 位寻址:

SSPxADD<7:0>= A0h (10100000) (本例中地址高 2 位被忽略，因为它们不受掩码影响)

ADMSK<5:1> = 00111

可被应答的地址: A0h, A1h, A2h, A3h, A4h, A5h, A6h, A7h, A8h, A9h, AAh, ABh, ACh, ADh, AEh, AFh

19.5.3.4 7 位地址掩码模式

不同于 5 位地址掩码模式，7 位地址掩码模式使用最多 8 位的掩码（10 位寻址时），用传入地址的最低位来定义可被应答的地址范围。使用 7 位寻址时，这使模块最多可以应答 127 个不同地址；使用 10 位寻址时，这使模块最多可以应答 255 个地址（见例 19-4）。该模式是模块的默认配置，当 MSSPMSK 未编程时（1），即选择该模式。

7 位地址掩码模式的地址掩码存储在 SSPxMSK 寄存器中，而不是 SSPxCON2 寄存器中。SSPxMSK 是模块中一个独立的硬件寄存器，但它不能直接访问。它与 SSPxADD 寄存器共用 SFR 空间中的一个地址。要访问 SSPxMSK 寄存器，必须选择 MSSP 模式 1001（SSPxCON1<3:0>=1001），然后读取或写入 SSPxADD 单元。

要使用 7 位地址掩码模式，必须先使用一个值初始化 SSPxMSK，然后再选择 I²C 从寻址模式。所以，必需的事件序列为：

- 1. 选择 SSPxMSK 访问模式（SSPxCON2<3:0>=1001）。
- 2. 将掩码值写入相应的 SSPxADD 寄存器地址（对于 MSSP1 为 FC8h，对于 MSSP2 为 F6Eh）。
- 3. 设置相应的 I²C 从模式（对于 10 位寻址，SSPxCON2<3:0>=0111；对于 7 位寻址为 0110）。

置 1 或清零 SSPxMSK 中掩码位的作用与在 5 位地址掩码模式下置 1 或清零 ADMSK 位的作用相反。即，清零 SSPxMSK 中的某个位会导致相应的地址位被掩码；置 1 该位则要求相应的位单元匹配。任何复位后，SSPxMSK 都会复位到全 1 状态，因此，在写入掩码值之前对标准 MSSP 操作没有影响。

在 7 位地址掩码模式下，SSPxMSK<7:1> 位可用来掩码 SSPxADD 寄存器中对应的地址位。如果 SSPxMSK 的某位是有效的（SSPxMSK<n>=0），则对应的 SSPxADD 地址位被忽略（SSPxADD<n>=x）。对于发出地址应答的模块，只要与没被掩码的地址位匹配就可以了。

在 10 位地址掩码模式下，SSPxMSK<7:0> 位可用来掩码 SSPxADD 寄存器中对应的地址位。如果 SSPxMSK 的某位是有效的（=0），则对应的 SSPxADD 地址位被忽略（SSPxADD<n>=x）。

注：地址的高 2 位不受地址掩码的影响。

例 19-4: 7 位掩码模式下的地址掩码示例

7 位寻址：
SSPxADD<7:1>= 1010 000
SSPxMSK<7:1>= 1111 001
可被应答的地址 = ACh, A8h, A4h, A0h

10 位寻址：
SSPxADD<7:0> = 1010 0000 （本例中地址高 2 位被忽略，因为它们不受影响）
SSPxMSK<7:0> = 1111 0011
可被应答的地址 = ACh, A8h, A4h, A0h

19.5.3.5 接收

当地址字节的 $\overline{R/W}$ 位清零并发生地址匹配时，SSPxSTAT 寄存器的 $\overline{R/W}$ 位清零。接收的地址被装入 SSPxBUF 寄存器，且 SDAx 线保持低电平（ACK）。

当发生地址字节溢出时，则不会产生应答（ $\overline{ACK}$ ）脉冲。溢出条件定义为 BF 位（SSPxSTAT<0>）被置 1，或 SSPOV 位（SSPxCON1<6>）被置 1。

每传输一个数据字节都会产生一个 MSSP 中断。中断标志位 SSPxIF 必须用软件清零。通过 SSPxSTAT 寄存器可以确定该字节的状态。

如果 SEN 被使能（SSPxCON2<0> = 1），SCLx 将在每次数据传输后保持低电平（时钟延长）。必须通过将 CKP 位（SSPxCON1<4>）置 1 来释放时钟。更多信息，请参见第 19.5.4 节“时钟延长”。

19.5.3.6 发送

当输入的地址字节的 $\overline{R/W}$ 位置 1 并发生地址匹配时，SSPxSTAT 寄存器的 $\overline{R/W}$ 位置 1。接收到的地址被装入 SSPxBUF 寄存器。 $\overline{ACK}$ 脉冲在第 9 位发送，同时不管 SEN 的值如何，SCLx 引脚保持低电平（更多详细信息，请参见第 19.5.4 节“时钟延长”）。通过延长时钟，主器件只有在从器件准备好发送数据时，才发出另一个时钟脉冲。发送的数据必须被装入 SSPxBUF 寄存器，同时也被装入 SSPxSR 寄存器。然后，应通过将 CKP（SSPxCON1<4>）置 1 来使能 SCLx 引脚。8 个数据位在 SCLx 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（图 19-10）。

来自主接收器的 $\overline{ACK}$ 脉冲将在第 9 个 SCLx 输入脉冲的上升沿锁存。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 $\overline{ACK}$ ，则从器件会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。

每传输一个数据字节都会产生一个 MSSP 中断。SSPxIF 位必须用软件清零，SSPxSTAT 寄存器用于确定字节的状态。SSPxIF 位在第 9 个时钟脉冲的下降沿被置 1。

图 19-8: I²C™ 从模式接收时序 (SEN = 0, 7 位地址)

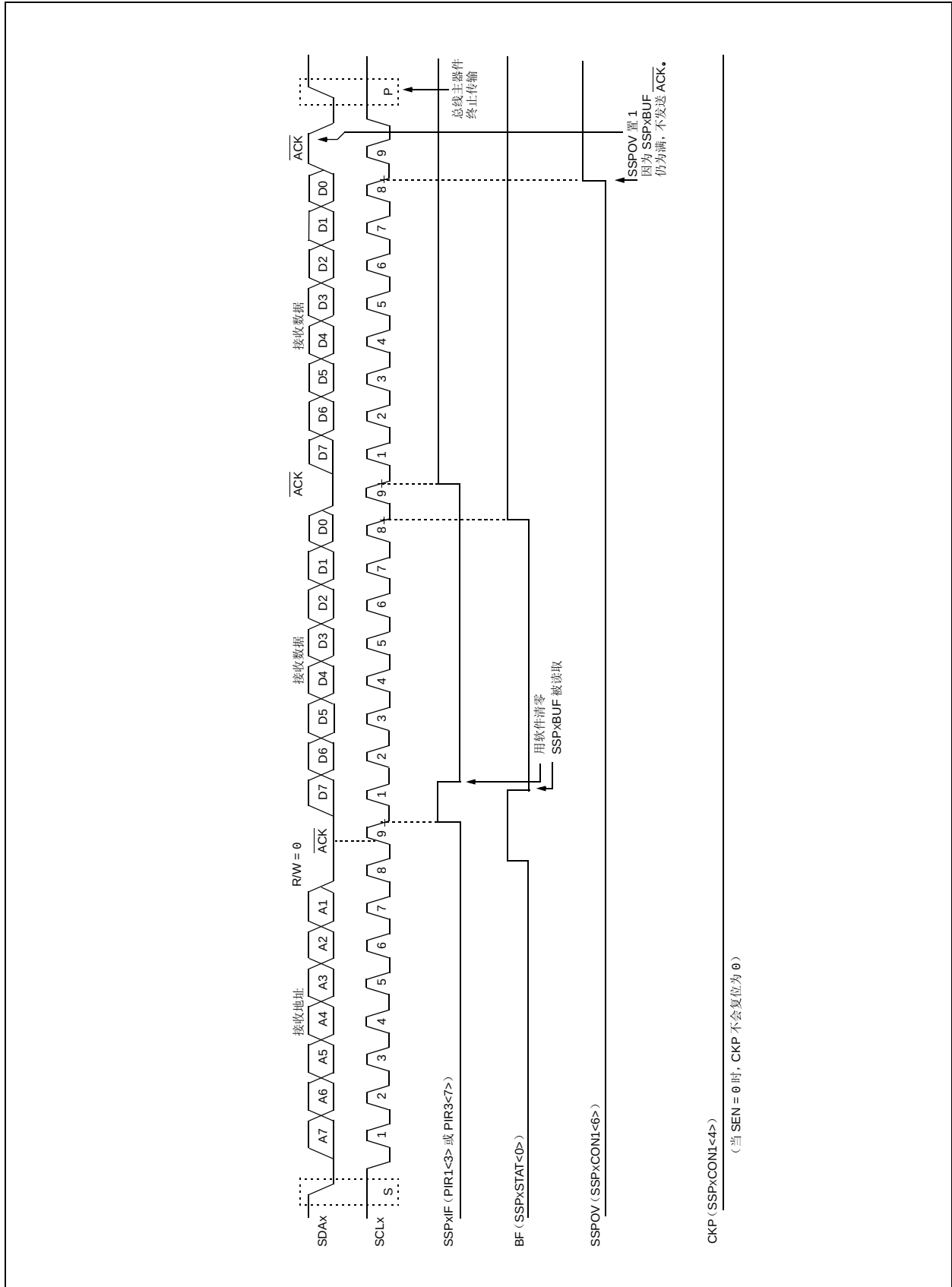


图 19-9: I²C™ 从模式接收时序 (SEN = 0 且 ADMSK<5:1> = 01011, 7 位地址)

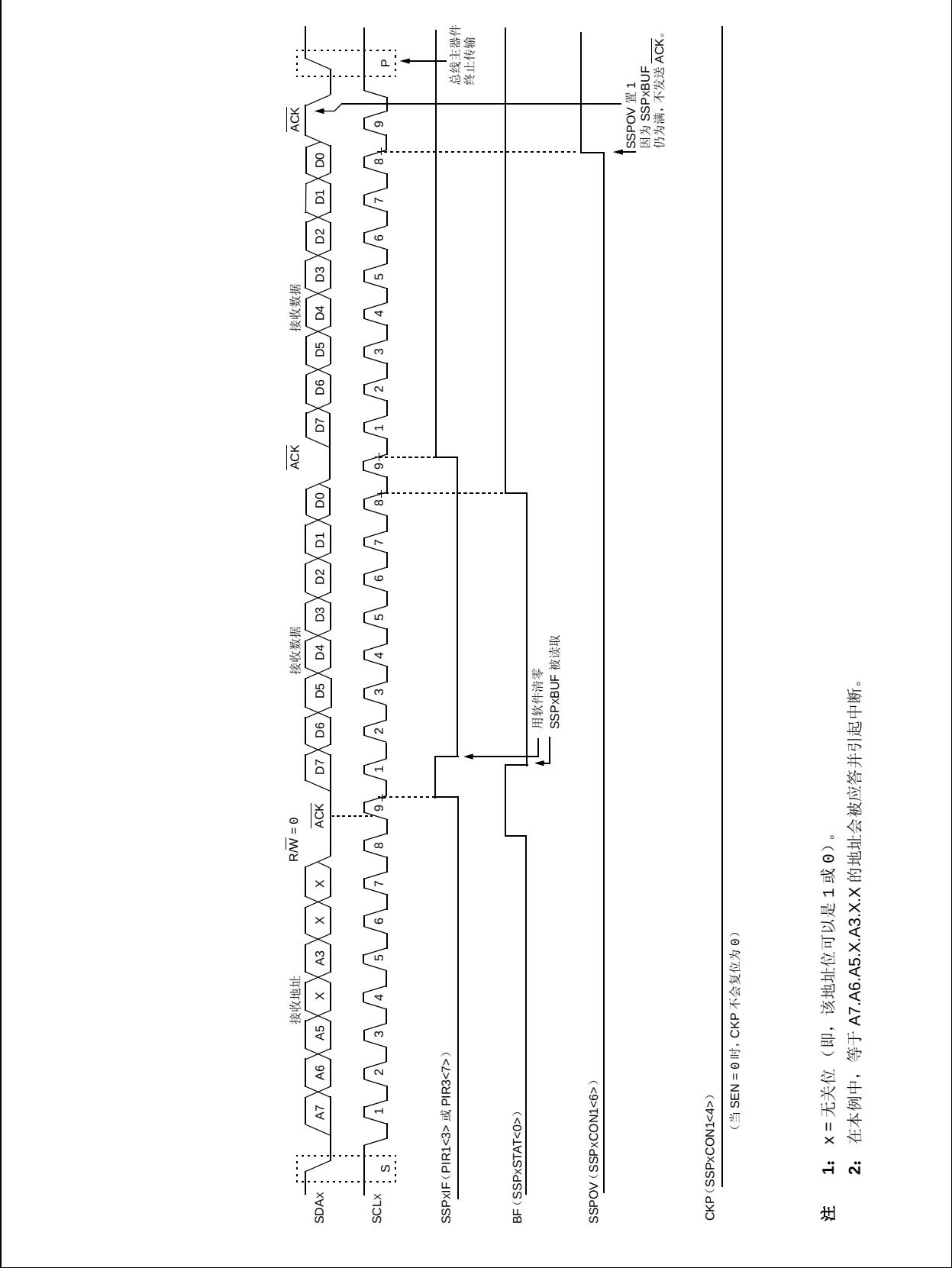


图 19-10: I²C™ 从模式发送时序 (7 位地址)

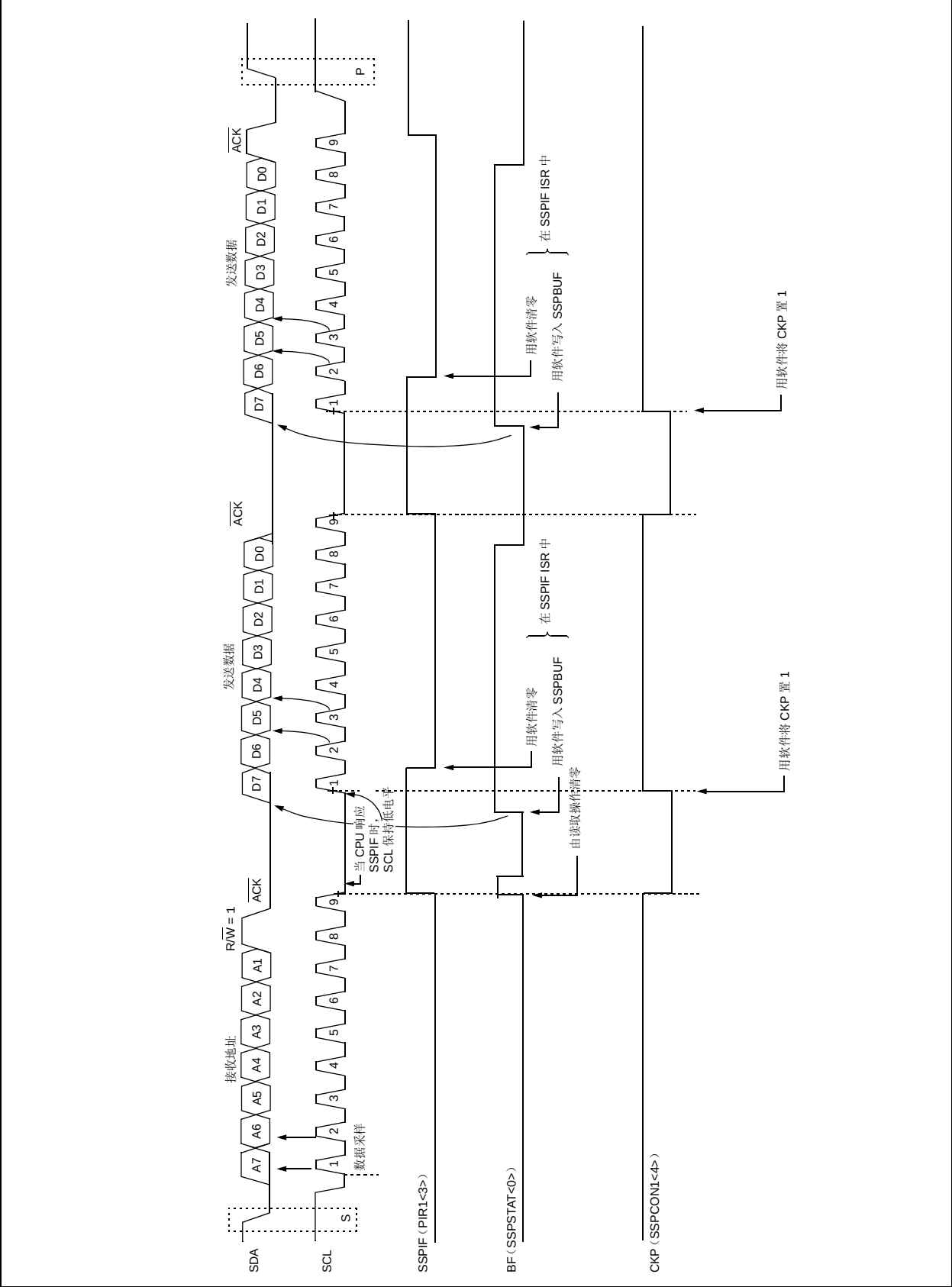
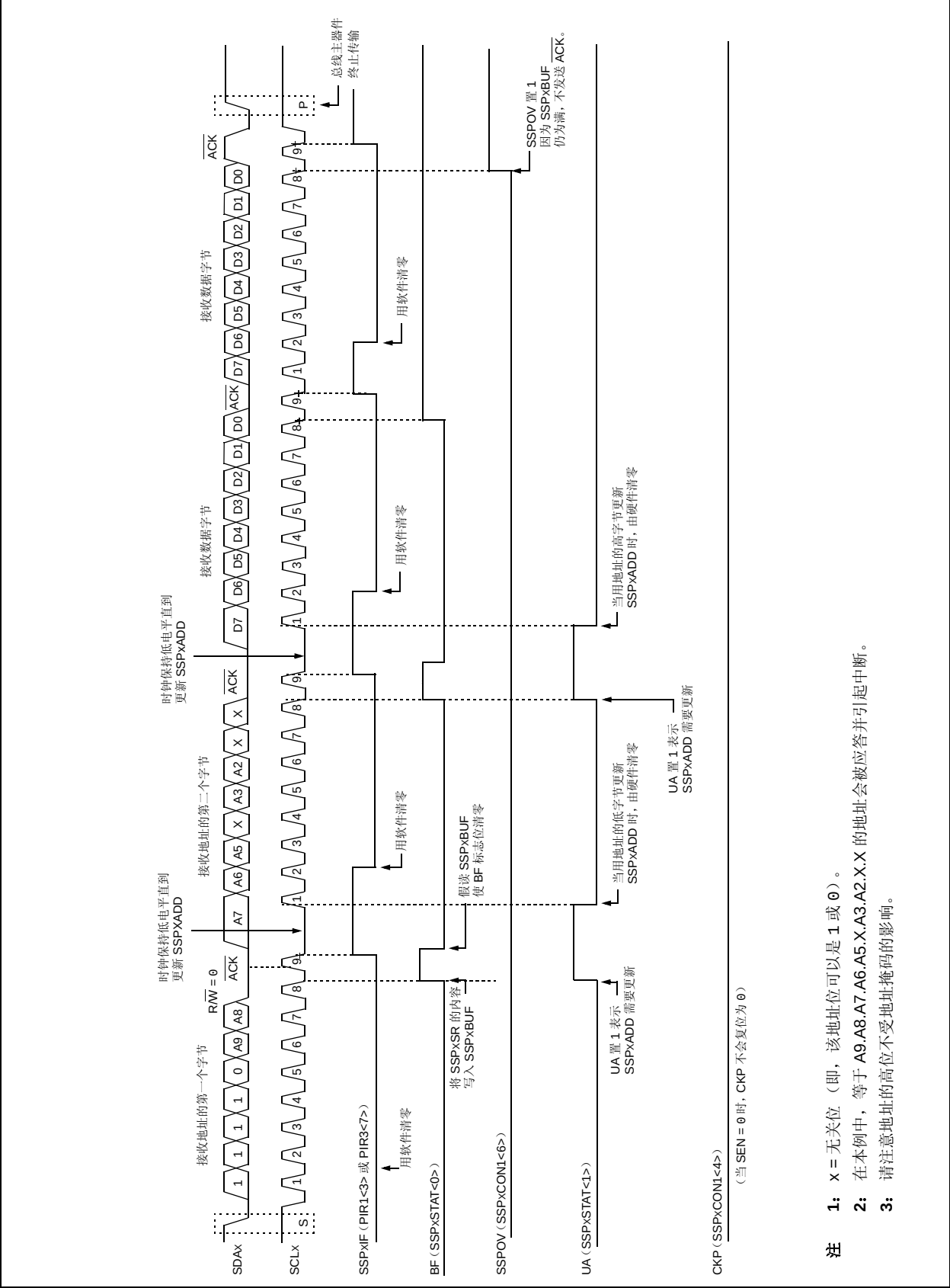


图 19-11: I²C™ 从模式接收时序 (SEN = 0 且 ADMSK<5:1> = 01001, 10 位地址)



- 注
- 1: x = 无关位 (即, 该地址位可以是 1 或 0)。
 - 2: 在本例中, 等于 A9.A8.A7.A6.A5.X.A3.A2.X.X 的地址会被应答并引起中断。
 - 3: 请注意地址的高位不受地址掩码的影响。

图 19-12: I²C™ 从模式接收时序 (SEN = 0, 10 位地址)

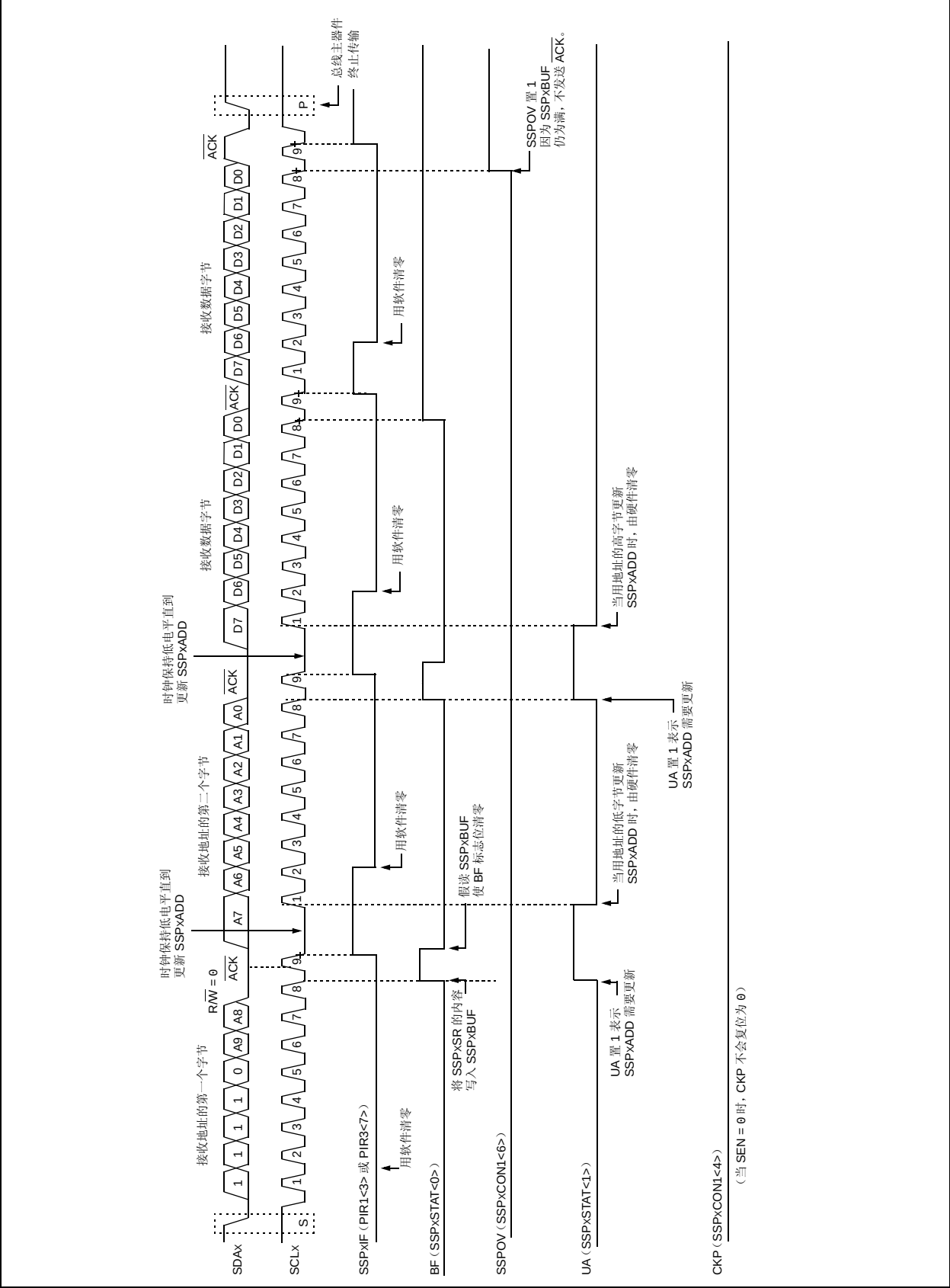
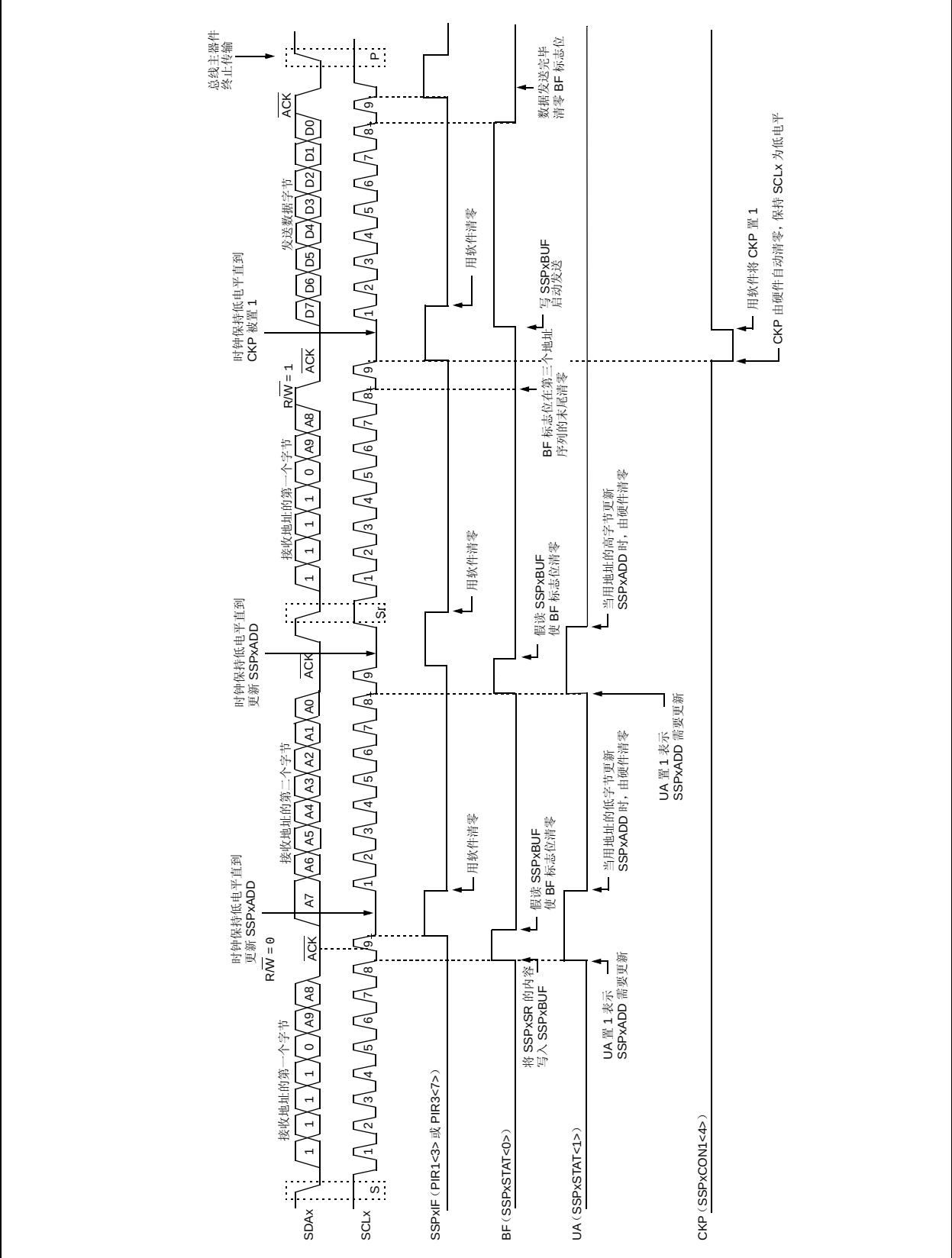


图 19-13: I²C™ 从模式发送时序 (10 位地址)



19.5.4 时钟延长

7 位和 10 位从模式均能在发送序列期间实现自动时钟延长。

SEN 位 (SSPxCON2<0>) 允许在接收期间使能时钟延长。将 SEN 置 1 将使 SCLx 引脚在每个数据接收序列的末尾保持低电平。

19.5.4.1 7 位从接收模式 (SEN = 1) 的时钟延长

在 7 位从接收模式下, 如果在 ACK 序列末尾的第 9 个时钟的下降沿将 BF 位置 1, 则 SSPxCON1 寄存器的 CKP 位就会自动清零, 强制 SCLx 输出保持在低电平。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 ISR 中将 CKP 位置 1。保持 SCLx 线为低电平, 用户可以在主器件发起另一个接收序列之前, 有时间执行 ISR 并读取 SSPxBUF 的内容。这将防止发生缓冲区溢出 (见图 19-15)。

- 注 1:** 如果用户在第 9 个时钟的下降沿到来之前读取了 SSPxBUF 的内容, 使得 BF 位被清零, 那么 CKP 位就不会被清零, 也不会发生时钟延长。
- 2:** 不管 BF 位的状态如何, CKP 位都可以用软件置 1。为避免溢出, 在下一个接收序列开始之前, 用户要注意在 ISR 中清零 BF 位。

19.5.4.2 10 位从接收模式 (SEN = 1) 的时钟延长

在 10 位从接收模式下, 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 个时钟之后置 1, 将启动时钟延长。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/W 位。在更新 SSPxADD 的时候释放时钟线。如同 7 位模式一样, 在每个数据接收序列中均会发生时钟延长。

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。

19.5.4.3 7 位从发送模式的时钟延长

如果 BF 位被清零, 7 位从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。上述情形与 SEN 位的状态无关。

用户的中断服务程序 (ISR) 必须先将 CKP 位置 1 才可以继续发送。在保持 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, 将有时间执行 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, 使得 BF 位置 1, 那么 CKP 位就不会被清零, 也不会发生时钟延长。
- 2:** 不管 BF 位的状态如何, CKP 位都可以用软件置 1。

19.5.4.4 10 位从发送模式的时钟延长

在 10 位从发送模式下, 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 位从接收模式一样。头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为 1 的 R/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, 由 BF 标志位控制时钟延长, 正如 7 位从发送模式一样 (见图 19-13)。

19.5.4.5 时钟同步和 CKP 位

当 CKP 位被清零时，SCLx 输出被强制为 0。然而，将 CKP 位清零不会将 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 位不会将 SCLx 线拉为低电平，除非外部 I²C 主器件将 SCLx 线拉低。

SCLx 输出将保持低电平，直到 CKP 位置 1 且 I²C 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

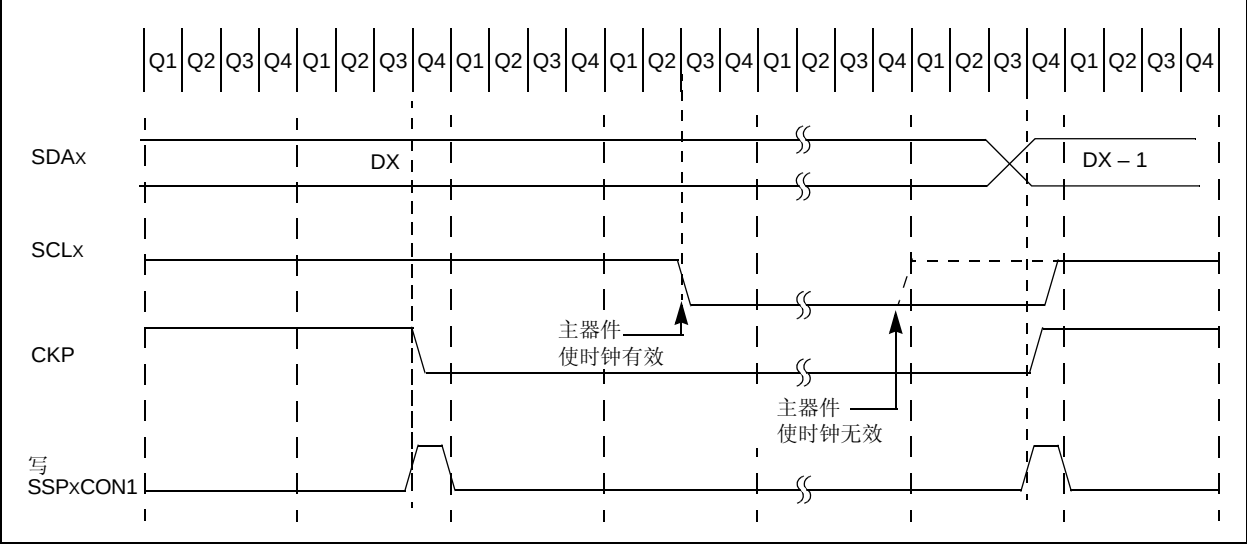


图 19-15: I²C™ 从模式接收时序 (SEN = 1, 7 位地址)

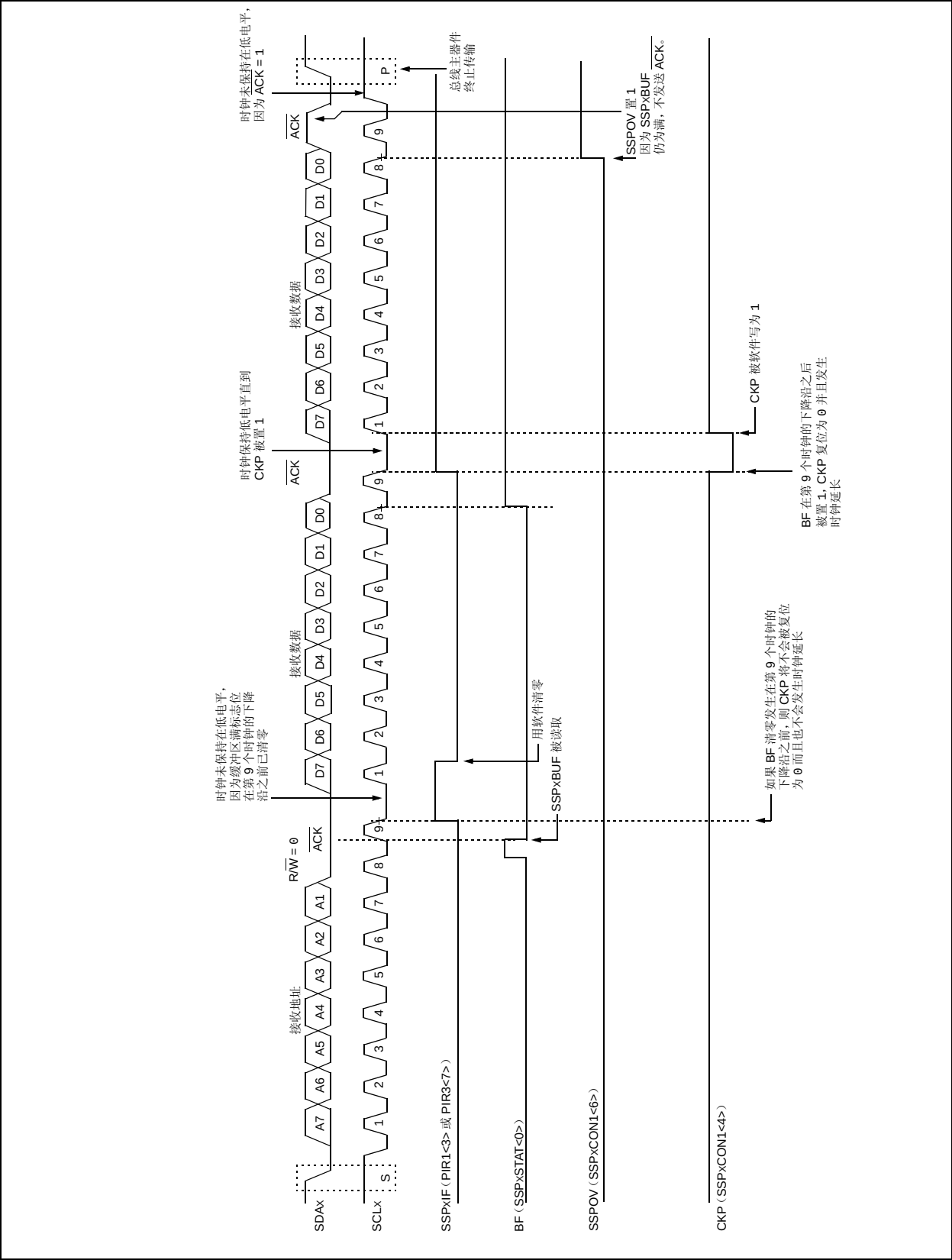
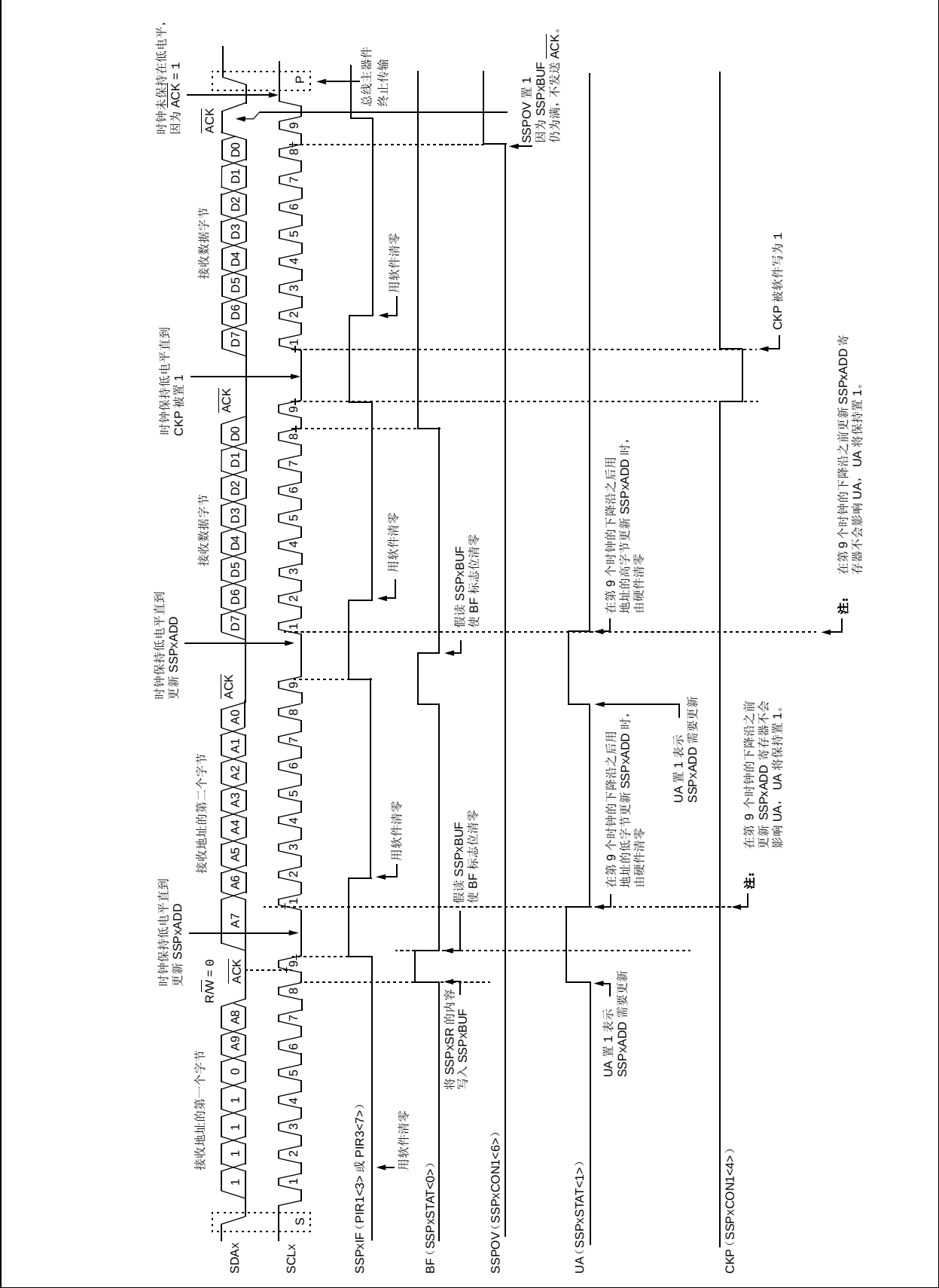


图 19-16: I²C™ 从模式接收时序 (SEN = 1, 10 位地址)



19.5.5 广播呼叫地址支持

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。当使用这个地址时，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 I²C 协议为特定目的保留的 8 个地址之一。它由全 0 组成，且 R/W = 0。

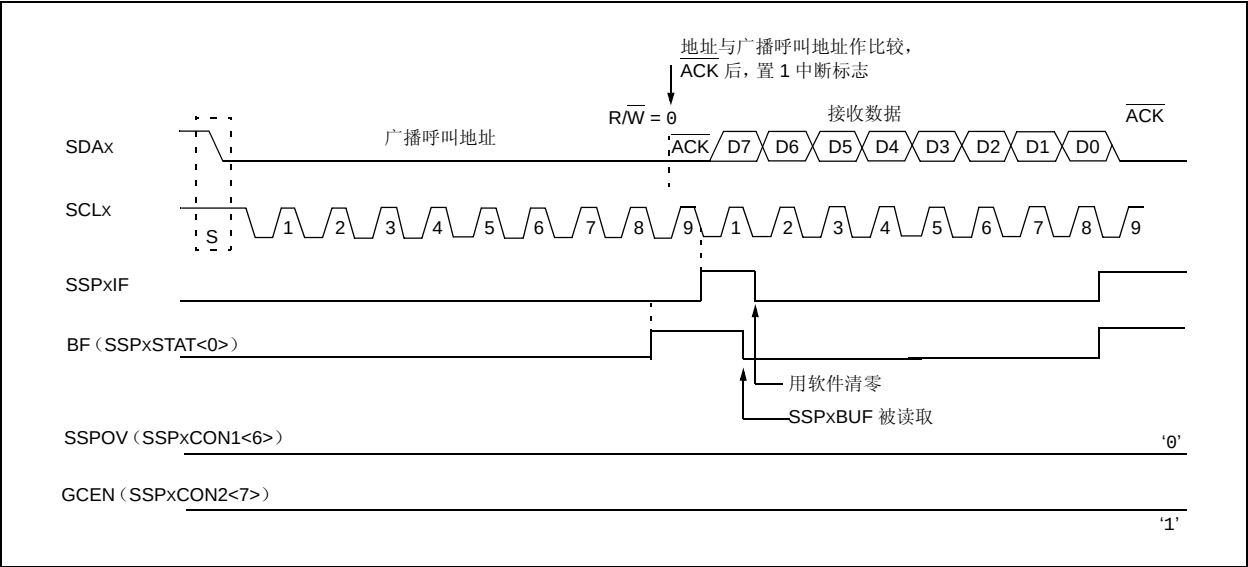
当使能广播呼叫使能位 (GCEN) (SSPxCON2<7> 置 1) 时，即可识别广播呼叫地址。检测到启动位后，8 位数据会被移入 SSPxSR，同时将地址与 SSPxADD 进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

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。

当中断得到响应时，可以通过读取 SSPxBUF 的内容来检查中断源。该值可用于判断是特定器件的地址还是一个广播呼叫地址。

在 10 位模式下，需要更新 SSPxADD 用来匹配地址的后半部分，同时 UA 位置 1 (SSPxSTAT<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 (图 19-17)。

图 19-17: 从模式广播呼叫地址序列 (7 位或 10 位寻址模式)



PIC18F46J50 系列

19.5.6 主模式

通过将 SSPxCON1 中的相应 SSPM 位置 1 和清零，同时将 SSPEN 位置 1，可以使能主模式。在主模式下，如果 TRIS 位被置 1，SCLx 和 SDAX 线由 MSSP 硬件控制。

主模式通过检测到启动条件和停止条件时产生中断来工作。启动(S)位和停止(P)位在复位时或禁止MSSP模块时清零。当停止位置1时,可以取得I²C总线的控制权;或者,总线处于空闲状态,启动位和停止位都清零。

在固件控制的主模式下，用户代码根据启动位和停止位条件执行所有的 I²C 总线操作。

一旦使能主模式，用户即可选择以下 6 项操作。

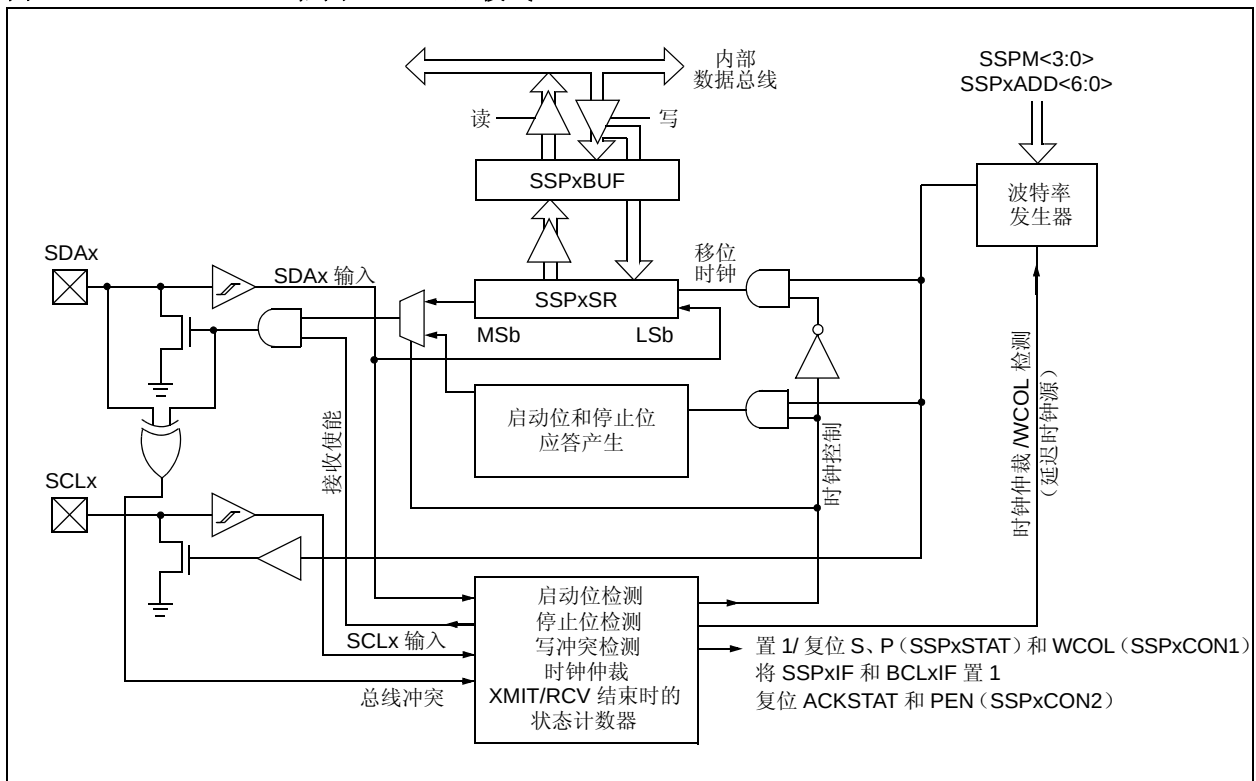
1. 在 SDAx 和 SCLx 上发出一个启动条件。
2. 在 SDAx 和 SCLx 上发出一个重复启动条件。
3. 写入 SSPxBUF 寄存器, 启动数据 / 地址的发送。
4. 配置 I²C 端口用于接收数据。
5. 在接收数据字节末尾产生应答信号。
6. 在 SDAx 和 SCLx 上产生一个停止条件。

注: 当配置为 I²C 主模式时, MSSP 模块不允许事件排队。例如, 在启动条件结束前, 不允许用户立即写 SSPxBUF 寄存器以启动传输。在这种情况下, 将不会执行写 SSPxBUF, WCOL 位将被置 1, 这表明没有发生对 SSPxBUF 的写操作。

以下事件会使 MSSP 中断标志位 SSPxIF 置 1（如果允许 MSSP 中断，则产生中断）：

- 启动条件
- 停止条件
- 数据字节发送 / 接收
- 应答发送
- 重复启动

图 19-18: MSSPx 框图 (I²C™ 主模式)



19.5.6.1 I²C 主模式工作原理

主器件产生所有的串行时钟脉冲、启动条件和停止条件。以停止条件或重复启动条件结束传输过程。因为重复启动条件也是下一次串行传输的开始，因此 I²C 总线不会被释放。

在主发送器模式下，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作为接收方的从器件地址（7 位）和读/写（R/W）位。在这种情况下，R/W 位将是逻辑 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 S 和 P 条件，表明串行传输的开始和结束。

在主接收模式下，发送的第一个字节包括作为发送方的从器件地址（7 位）和 R/W 位。在这种情况下，R/W 将是逻辑 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。一次接收 8 位串行数据。每接收到一个字节，都会发送一个应答位。S 和 P 条件表明代表发送的开始和结束。

在 I²C 模式下，将使用 SPI 工作模式中使用的 BRG 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 19.5.7 节“波特率”。

下面是一个典型的发送序列：

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。在进行任何其他操作前，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. 器件地址从 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 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 位数据。
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 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。

19.5.7 波特率

在 I²C 主模式下，BRG 的重载值存放在 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 计数器会在每个指令周期（Tcy）中的 Q2 和 Q4 时钟周期上进行两次递减计数。在 I²C 主模式下，会自动重载 BRG。

如果指定操作完成（即，在传输的最后一个数据位后面跟着 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 给出了不同的指令周期下的时钟频率以及装入 SSPxADD 的 BRG 值。不支持 SSPxADD 的值为 0x00，而应使用 ≥ 0x01 的值。

PIC18F46J50 系列

19.5.7.1 波特率和模块的相互关系

因为 MSSP1 和 MSSP2 是相互独立的模块，所以它们可以以不同的波特率同时在 I²C 主模式下工作。这是通过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将会同等程度地影响这两个模块。通过更改 BRG 重载值也许可以将其中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

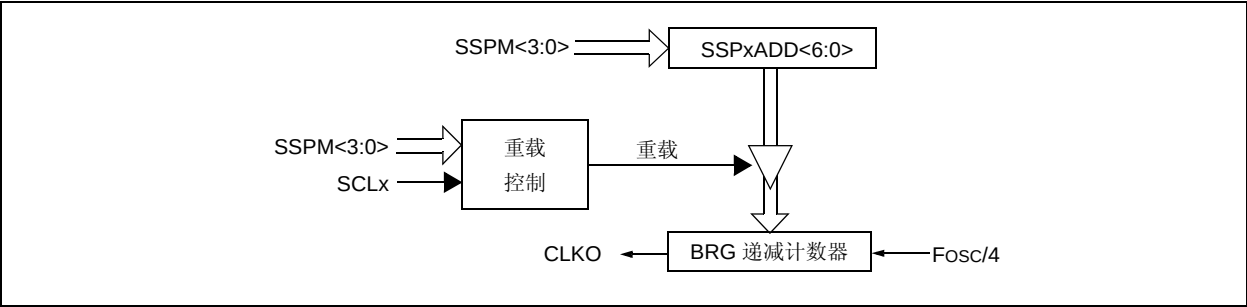


表 19-3: 使用 BRG 的 I²C™ 时钟频率

Fosc	Fcy	Fcy * 2	BRG 值	Fscl (两次 BRG 计满返回)
48 MHz	12 MHz	24 MHz	77h	100 kHz
40 MHz	10 MHz	20 MHz	18h	400 kHz ⁽¹⁾
40 MHz	10 MHz	20 MHz	63h	100 kHz
16 MHz	4 MHz	8 MHz	03h	1 MHz ⁽¹⁾
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

注 1: 虽然 I²C 接口各方面都不符合 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慎重使用。

PIC18F46J50 系列

19.5.9 I²C 主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, BRG 装入 SSPxADD<5:0> 的内容并开始计数。在该 BRG 计数周期 (TBRG) 内 SDAx 引脚被释放 (被拉高)。当 BRG 超时, 如果 SDAx 采样为高电平, SCLx 引脚将被拉高。当 SCLx 采样为高电平时, BRG 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 内采样为高电平。接下来, 在一个 TBRG 中, 将 SDAx 引脚驱动为低电平 (SDAx = 0), 同时 SCLx 保持高电平。随后 RSEN 位 (SSPxCON2<1>) 将自动清零, 这次 BRG 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, 启动位 (SSPxSTAT<3>) 将被置 1。直到 BRG 发生超时后, SSPxIF 位才会置 1。

- 注 1:** 有任何其他事件在进行时, 编程设置 RSEN 无效。
- 2:** 在重复启动条件期间, 以下事件将会导致总线冲突:
- 当 SCLx 由低电平变为高电平时, SDAx 采样为低电平。
 - 在 SDAx 被拉低之前, SCLx 变为低电平。这表明另一个主器件正试图发送一个数据 1。

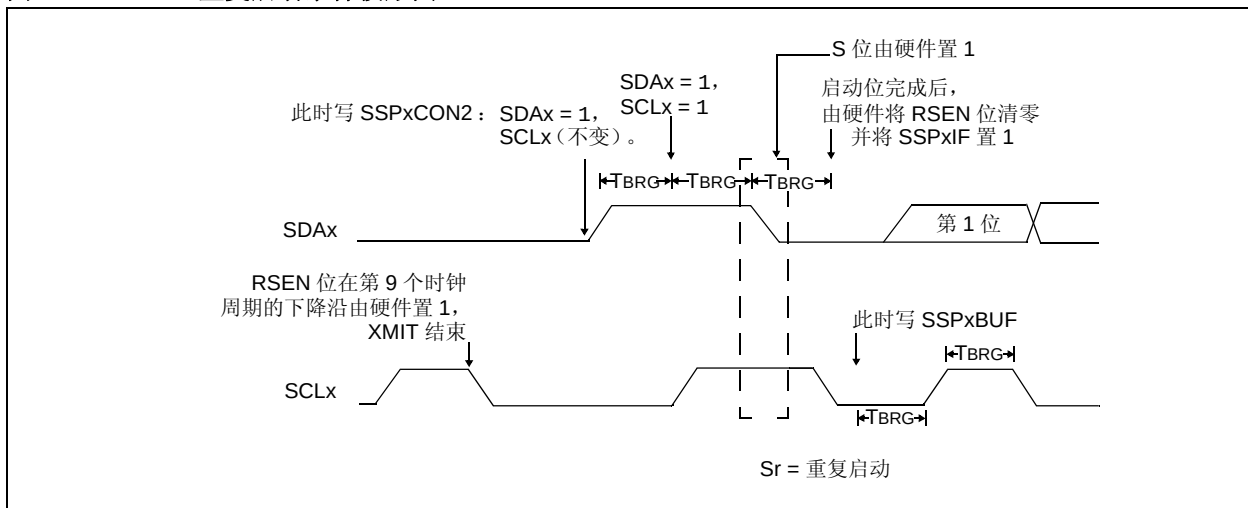
一旦 SSPxIF 位被置 1, 用户便可以在 7 位地址模式下将 7 位地址, 或者在 10 位地址模式下将默认的 8 位地址字节写入 SSPxBUF。当发送完第一个 8 位数据并接收到一个 ACK 后, 用户可以发送另外 8 位地址 (10 位模式) 或 8 位数据 (7 位模式)。

19.5.9.1 WCOL 状态标志

如果用户在重复启动序列过程中写 SSPxBUF, 则 WCOL 被置 1, 同时缓冲区内容不变 (写操作无效)。

注: 由于不允许事件排队, 在重复启动条件结束之前, 不能写 SSPxCON2 的低 5 位。

图 19-22: 重复启动条件波形图



19.5.10 I²C 主模式下的发送

发送一个数据字节、一个 7 位地址或一个 10 位地址的一半，都是通过写一个值到 SSPxBUF 寄存器来实现的。该操作将使缓冲区满标志位 BF 置 1，BRG 开始计数，同时开始下一次发送。在 SCLx 的下降沿有效后（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 BRG 计满返回计数周期（TBRG）内，SCLx 保持低电平。数据应该在 SCLx 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 引脚释放为高电平时，它将在一个 TBRG 内保持高电平状态。

在此期间以及 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 标志位被清零，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 个时钟周期发出一个 ACK 位作为响应。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，主时钟（BRG）暂停，直到下一个数据字节装入 SSPxBUF，SCLx 保持低电平，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 7 位地址位和 R/W 位都被移出。在第 8 个时钟的下降沿，主器件将 SDAx 引脚拉为高电平，以允许从器件发出一个应答响应。在第 9 个时钟的下降沿，主器件通过采样 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，BRG 关闭直到下一次写 SSPxBUF，且 SCLx 保持低电平，允许 SDAx 悬空。

19.5.10.1 BF 状态标志

在发送模式下，BF 位（SSPxSTAT<0>）在 CPU 写 SSPxBUF 时置 1，在所有 8 位数据移出后清零。

19.5.10.2 WCOL 状态标志

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 位被置 1，并且在写 SSPxBUF 之后的 2 Tcy 内缓冲区内容不变（写操作无效）。如果在 2 Tcy 内 SSPxBUF 被重新写入，则 WCOL 位被置 1 并且 SSPBUF 被更新。这可能导致传输被破坏。

用户应在每次写 SSPxBUF 后检查 WCOL 位是否清零，以确保传输正确。在所有情况下，WCOL 都必须用软件清零。

19.5.10.3 ACKSTAT 状态标志

在发送模式下，当从器件已发送应答响应（ACK = 0）时，ACKSTAT 位（SSPxCON2<6>）清零；当从器件没有应答（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。

19.5.11 I²C 主模式接收

通过编程接收使能位 RCEN（SSPxCON2<3>）使能主模式接收。

注： 将 RCEN 位置 1 前，MSSP 必须处于无效状态，否则对 RCEN 位置 1 将无效。

BRG 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，BRG 暂停计数，且 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 读缓冲区时，BF 标志位将自动清零。通过将应答序列使能位 ACKEN（SSPxCON2<4>）置 1，用户可以在接收结束时发送应答位。

19.5.11.1 BF 状态标志

接收数据过程中，把地址或数据字节从 SSPxSR 装入 SSPxBUF 时，BF 位置 1。在读 SSPxBUF 寄存器时将其清零。

19.5.11.2 SSPOV 状态标志

接收数据过程中，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。

19.5.11.3 WCOL 状态标志

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 位被置 1，同时缓冲区内容不变（写操作无效）。

图 19-23: I²C™ 主模式发送波形图 (7 位或 10 位地址)

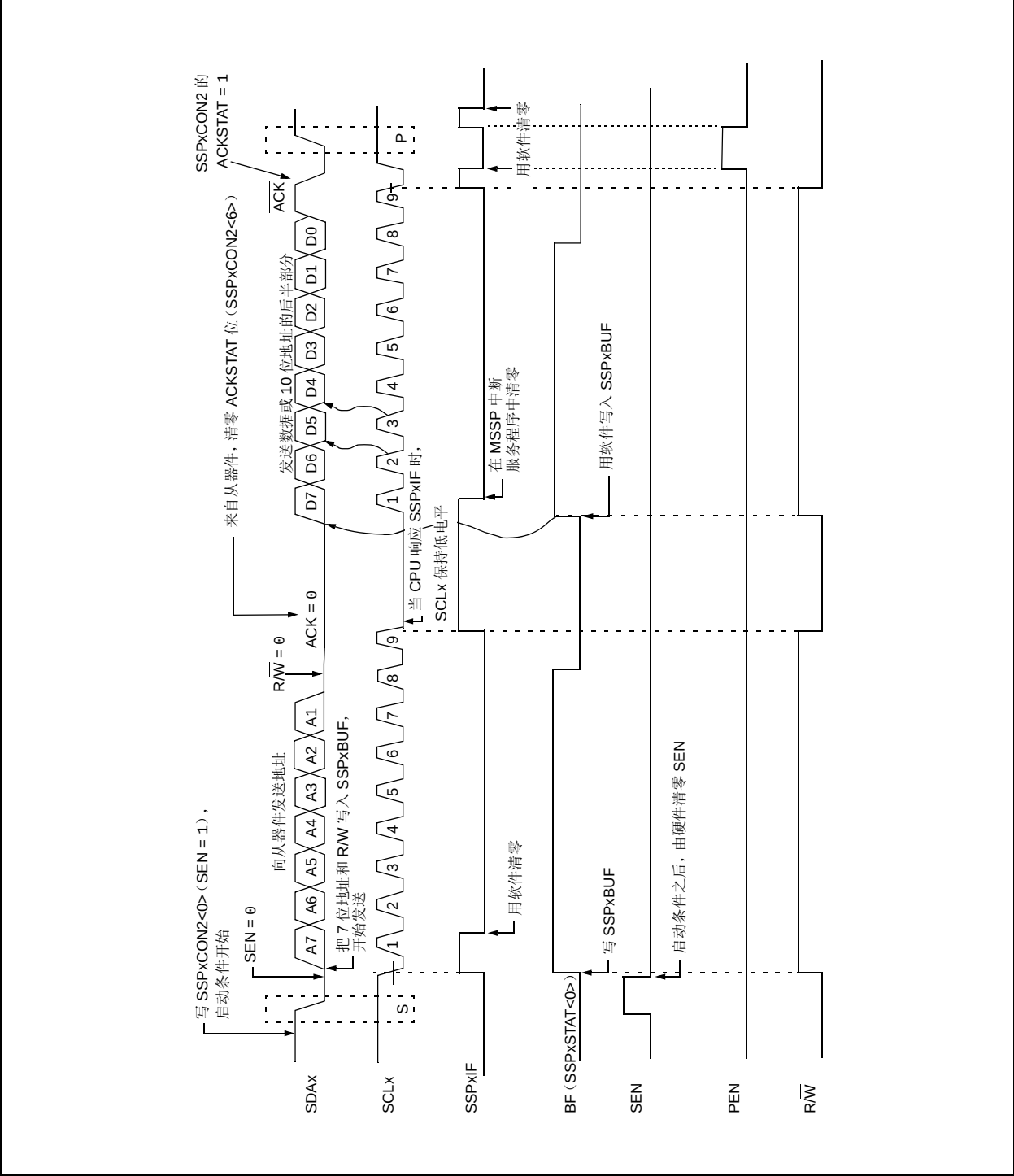
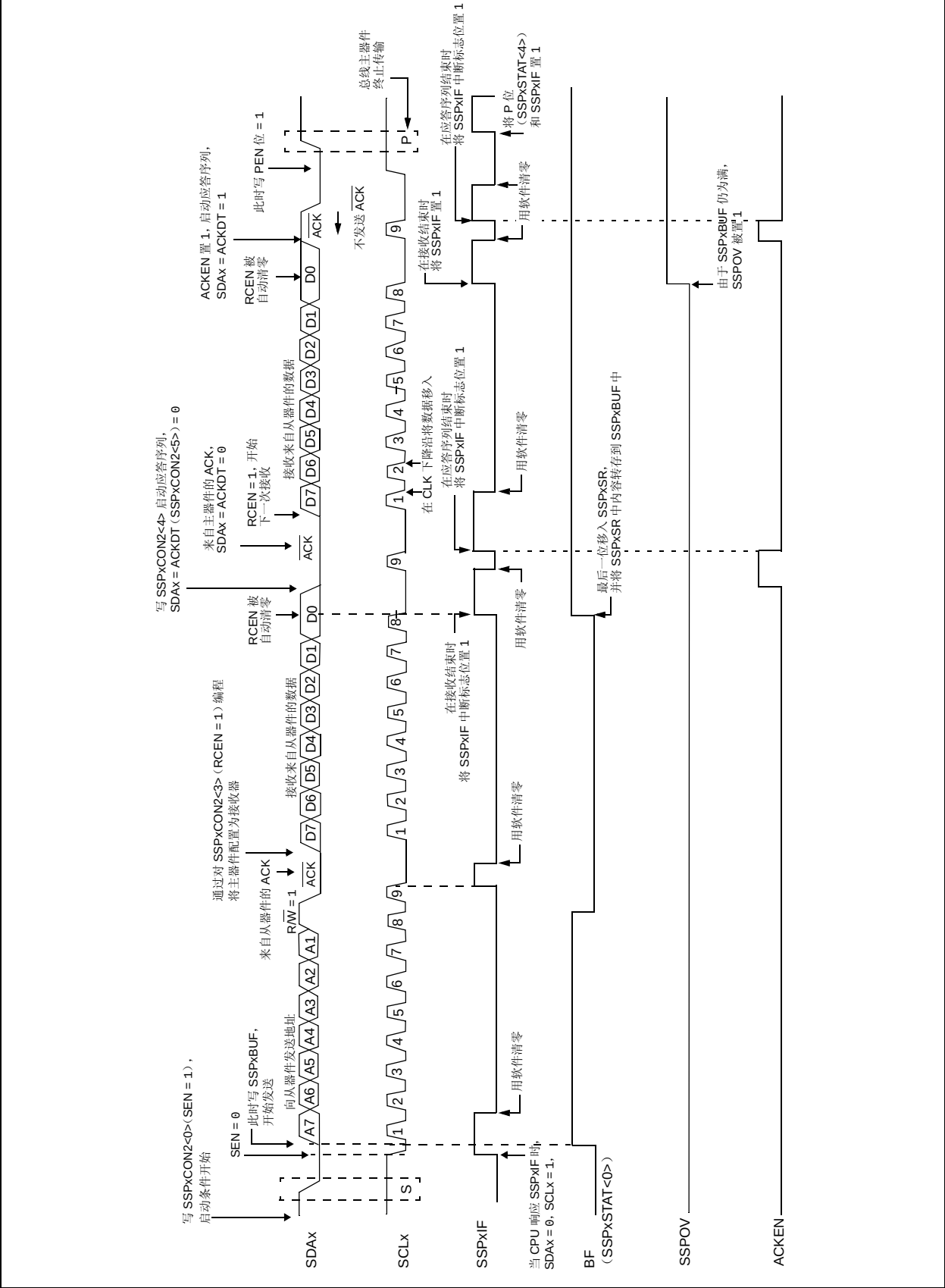


图 19-24: I²C™ 主模式接收波形图 (7 位地址)



PIC18F46J50 系列

19.5.12 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, 应答数据位的内容输出到 **SDAx** 引脚上。如果用户希望产生一个应答, 则应该将 **ACKDT** 位清零。否则, 用户要在应答序列开始前将 **ACKDT** 位置 1。然后 **BRG** 进行一个计满返回周期 (**TBRG**) 的计数, 随后 **SCLx** 引脚电平被拉高。当 **SCLx** 引脚采样为高电平 (时钟仲裁) 时, **BRG** 再进行一个 **TBRG** 周期的计数; 然后 **SCLx** 引脚被拉低。在这之后, **ACKEN** 位自动清零, **BRG** 关闭, **MSSP** 模块进入无效模式 (图 19-25)。

19.5.12.1 WCOL 状态标志

如果用户在应答序列进行过程中写 **SSPxBUF**, 则 **WCOL** 被置 1, 同时缓冲区内内容不变 (写操作无效)。

19.5.13 停止条件时序

如果将停止时序使能位 **PEN** (**SSPxCON2<2>**) 置 1, 则在接收 / 发送结束时, **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** 线采样为低电平时, **BRG** 被重载并递减计数至 0。当 **BRG** 发生超时, **SCLx** 引脚被拉为高电平, 在一个波特率发生器计满返回周期 (**TBRG**) 之后, **SDAx** 引脚将被拉高。当 **SDAx** 引脚采样为高电平且 **SCLx** 也是高电平时, 停止位 (**SSPxSTAT<4>**) 置 1。另一个 **TBRG** 之后, **PEN** 位被清零, 同时 **SSPxIF** 位被置 1 (图 19-26)。

19.5.13.1 WCOL 状态标志

如果用户在停止序列进行过程中写 **SSPxBUF**, 则 **WCOL** 位被置 1, 同时缓冲区内内容不变 (写操作无效)。

图 19-25: 应答序列波形图

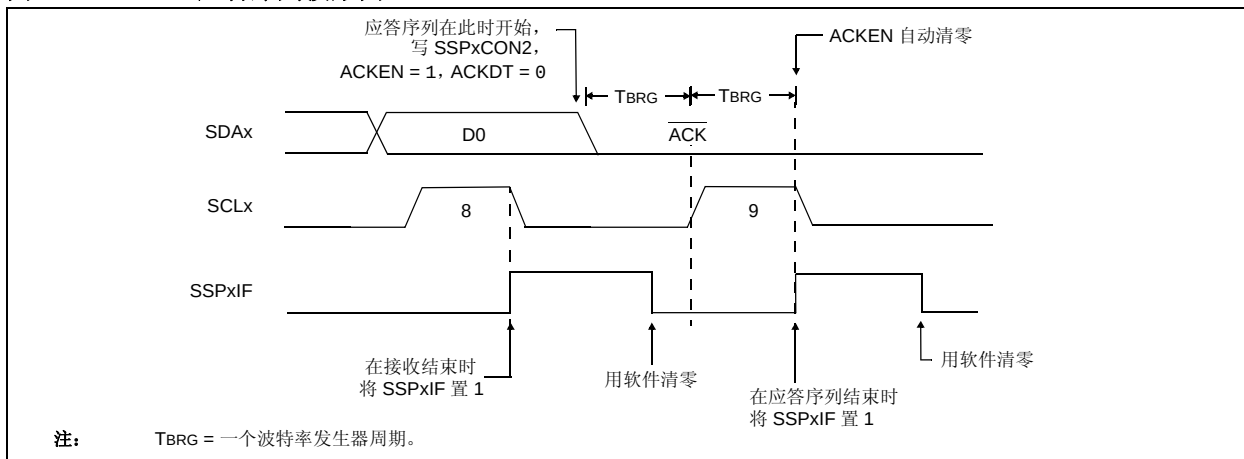
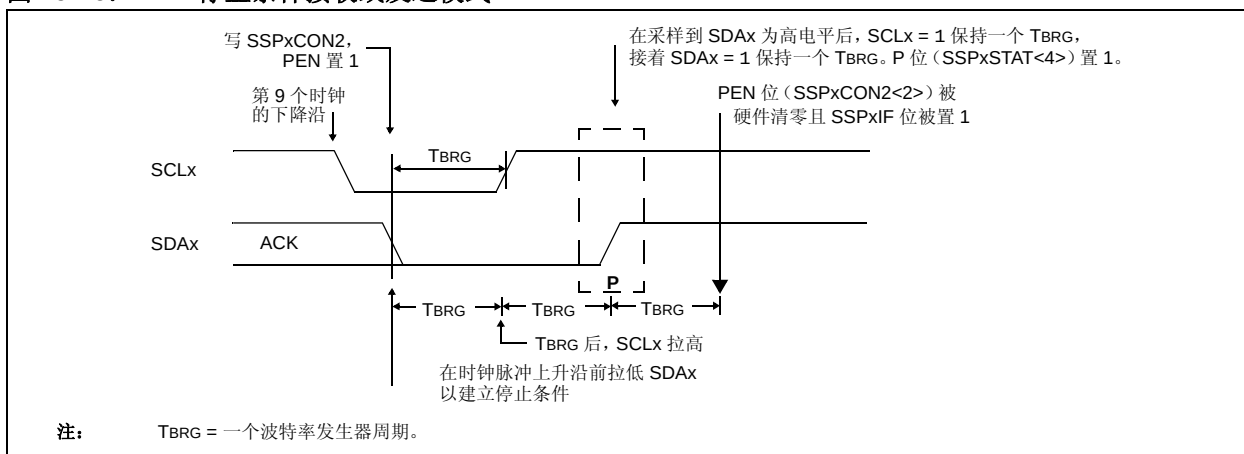


图 19-26: 停止条件接收或发送模式



19.5.14 休眠模式下的操作

在休眠模式下，I²C 模块能够接收地址或数据，并且当地址匹配或字节传输完成时，如果允许 MSSP 中断，将唤醒处理器。

19.5.15 复位的影响

复位会禁止 MSSP 模块并终止当前的数据传输。

19.5.16 多主器件模式

在多主器件模式下，在检测到启动和停止条件时将产生中断，这可用于判断总线是否空闲。启动位和停止位在复位或禁止 MSSP 模块时清零。当 P 位 (SSPxSTAT<4>) 置 1 时，可以取得 I²C 总线的控制权；或者，总线处于空闲状态，启动位和停止位都清零。当总线忙时，当发生停止条件时，将产生 MSSP 中断。

在多主器件模式下，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。此操作由硬件实现，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

19.5.17 多主器件通信、总线冲突和总线仲裁

多主器件模式是通过总线仲裁来支持的。当主器件将地址/数据位输出到 SDAx 引脚时，如果一个主器件在 SDAx 上输出 1（将 SDAx 引脚悬空为高电平），而另一个主器件输出 0，就会发生总线仲裁。当 SCLx 引脚悬空为高电平时，数据必须是稳定的。如果 SDAx 引脚上期望的数据是 1，而实际采样到的数据是 0，则发生了总线冲突。主器件将把总线冲突中断标志位 BCLxIF 置 1，并将 I²C 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，则发送操作停止，BF 标志位被清零，SDAx 和 SCLx 线被拉高，并且可写入 SSPxBUF。当执行总线冲突中断服务程序时，如果 I²C 总线空闲，用户可通过发出启动条件恢复通信。

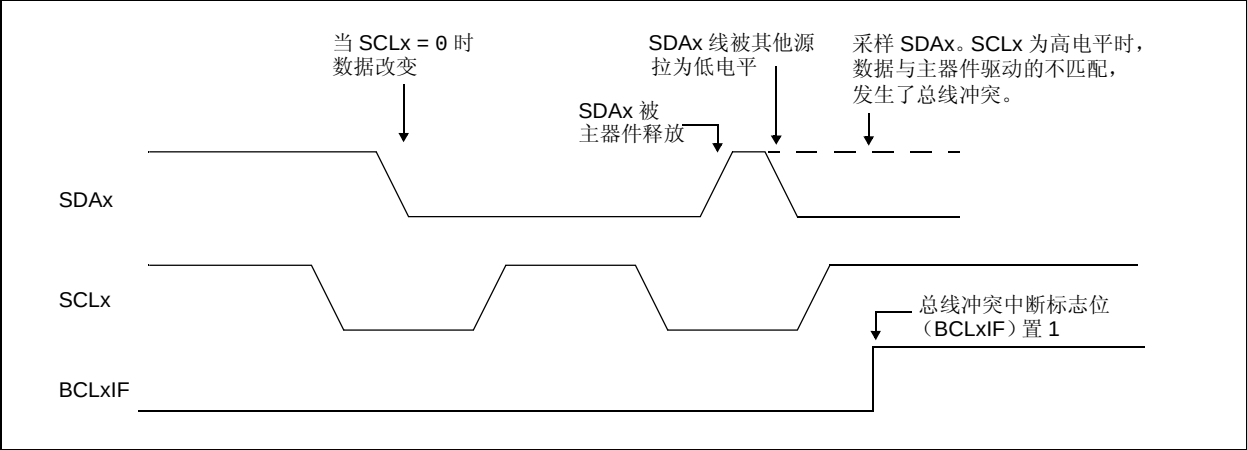
如果在启动、重复启动、停止或应答信号过程中发生总线冲突，则这种状态被中止，SDAx 和 SCLx 线被拉高，SSPxCON2 寄存器中的对应控制位清零。当执行完总线冲突中断服务程序 (ISR) 时，如果 I²C 总线空闲，用户可通过发出启动条件恢复通信。

主器件将继续监视 SDAx 和 SCLx 引脚。一旦出现停止条件，SSPxIF 位将被置 1。

发生总线冲突时无论发送的进度如何，写入 SSPxBUF 都会从第一个数据位开始发送数据。

在多主器件模式下，通过在检测到启动条件和停止条件时产生中断可以确定总线何时空闲。当 SSPxSTAT 寄存器中的停止位置 1 时，可以取得 I²C 总线的控制权；或者，总线处于空闲状态，启动位和停止位都清零。

图 19-27： 发送和应答时的总线冲突时序



PIC18F46J50 系列

19.5.17.1 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 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
- MSSP 模块复位为无效状态（图 19-28）

启动条件从 SDAx 和 SCLx 引脚被拉高开始。当 SDAx 引脚采样为高电平时，BRG 装入 SSPxADD<6:0> 的值并递减计数至 0。如果在 SDAx 为高电平时，SCLx 引脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图驱动一个数据 1。

如果 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，则在 BRG 计数结束时该引脚将被置为低电平。接着，BRG 被重载并递减计数至 0。在此期间，如果 SCLx 引脚采样到 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： 在启动条件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。但是上述情况不会引起总线冲突，因为两个主器件一定会对启动条件后的第一个地址进行仲裁。如果地址是相同的，必须继续对数据部分、重复启动条件或停止条件进行仲裁。

图 19-28： 启动条件期间的总线冲突（仅用于 SDAx）

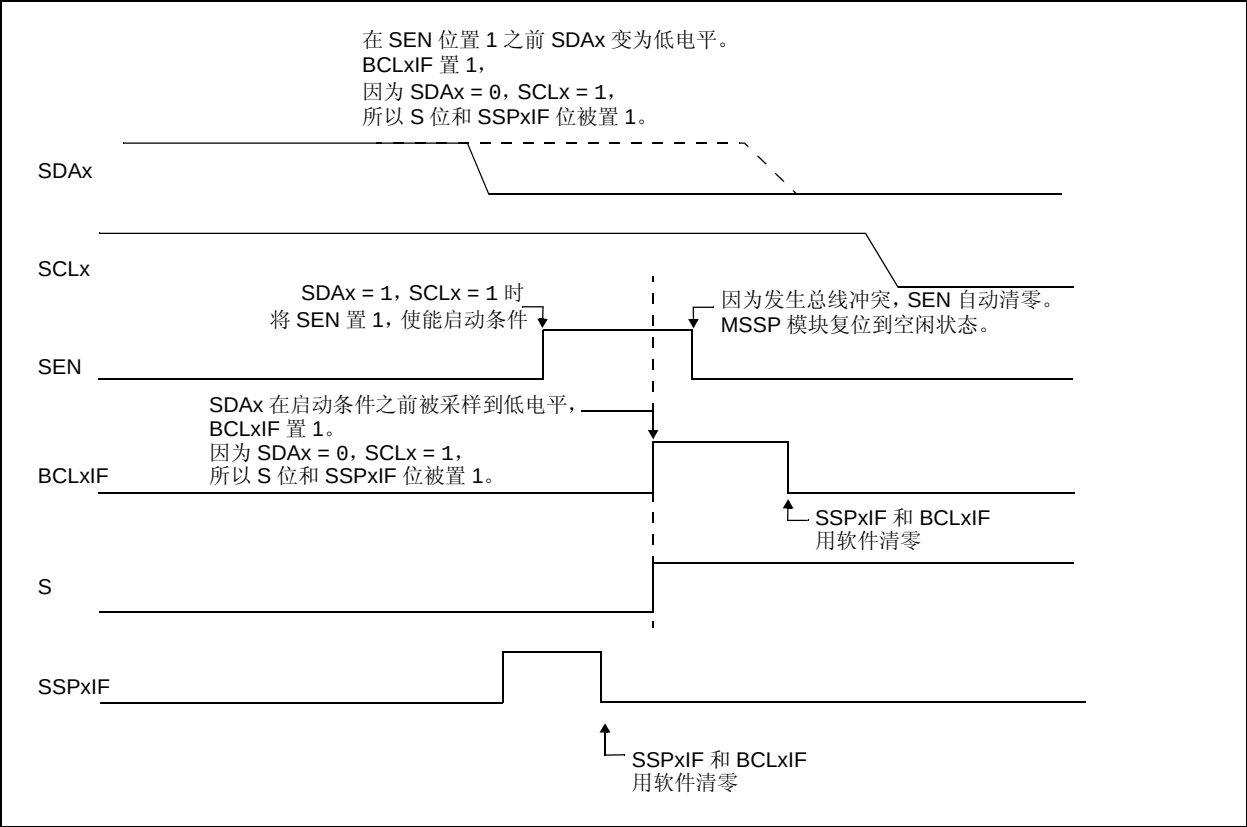


图 19-29: 启动条件期间的总线冲突 (SCLx = 0)

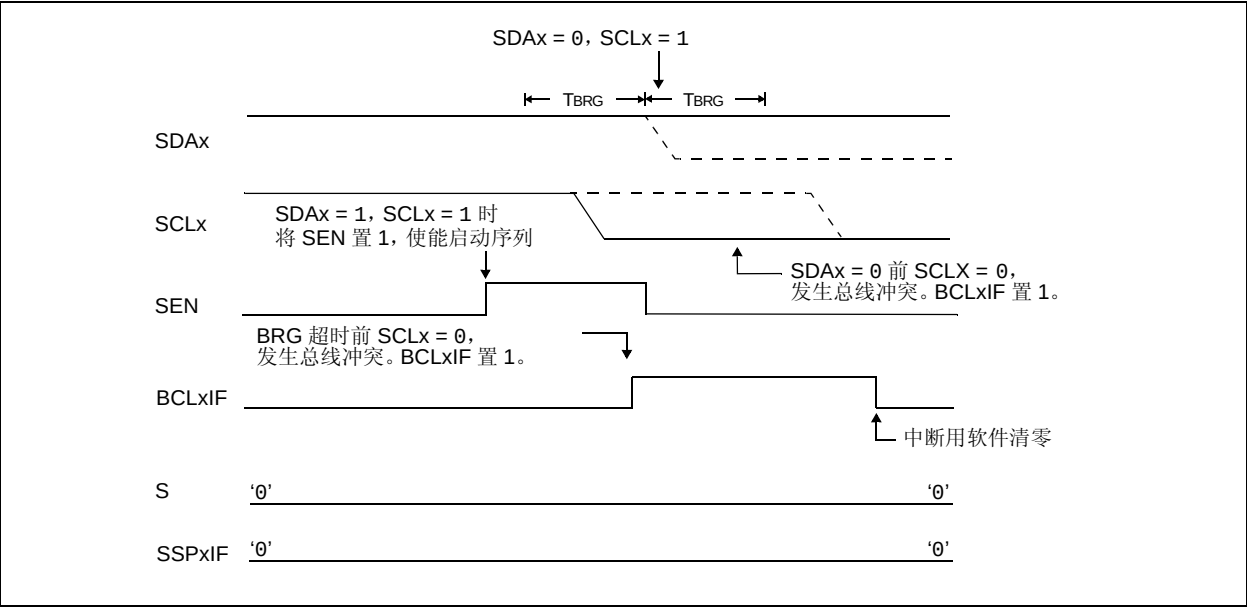
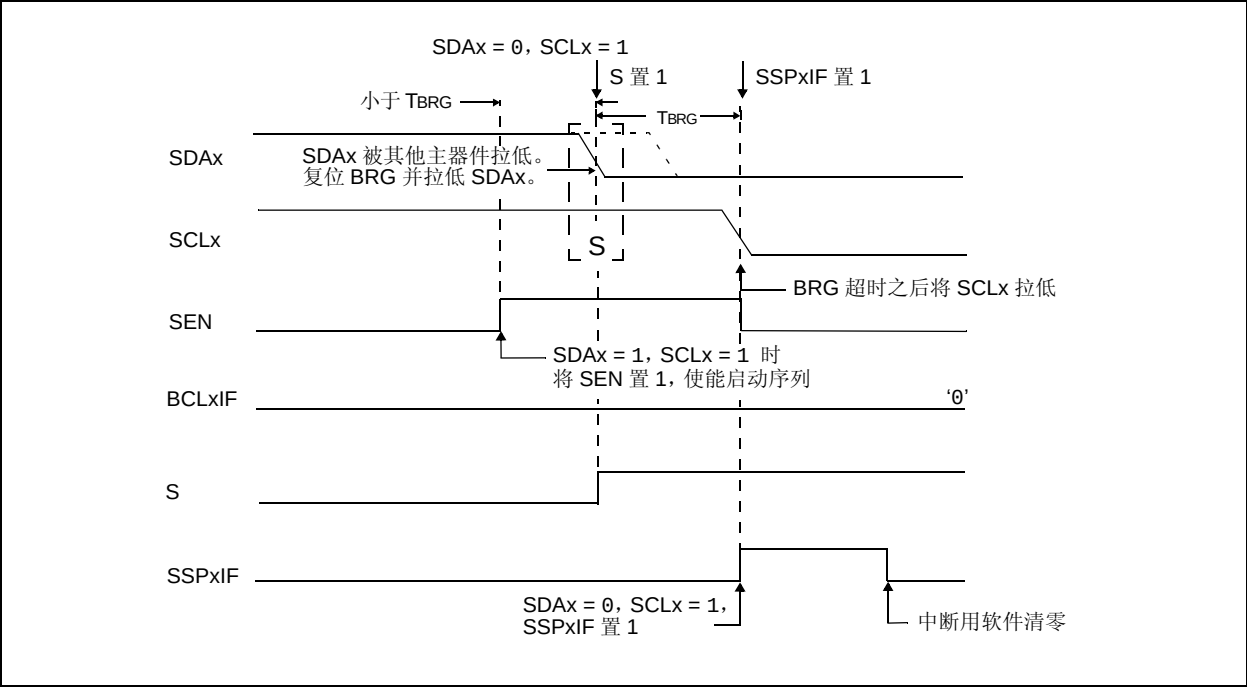


图 19-30: 启动条件期间由 SDAx 仲裁引起的 BRG 复位



PIC18F46J50 系列

19.5.17.2 重复启动条件期间的总线冲突

在以下情况中，重复启动条件期间会发生总线冲突：

- a) 在 SCLx 由低电平变为高电平期间，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一个数据 1。

当用户拉高 SDAx 并允许该引脚悬空时，BRG 装入 SSPxADD<6:0> 的值并递减计数至 0。接着 SCLx 引脚被拉高，当 SCLx 引脚采样到高电平时，对 SDAx 引脚进行采样。

如果 SDAx 为低电平，则已发生了总线冲突（即，另一个主器件正试图发送一个数据 0，见图 19-31）。如果 SDAx 采样到高电平，则 BRG 被重载并开始计数。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，且 SDAx 尚未被拉低，那么将发生总线冲突。在此情况下，另一个主器件在重复启动条件期间正试图发送一个数据 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，BRG 被重载并开始计数。在计数结束时，不管 SCLx 引脚的状态如何，SCLx 引脚都被拉低，重复启动条件结束。

图 19-31: 重复启动条件期间的总线冲突（情形 1）

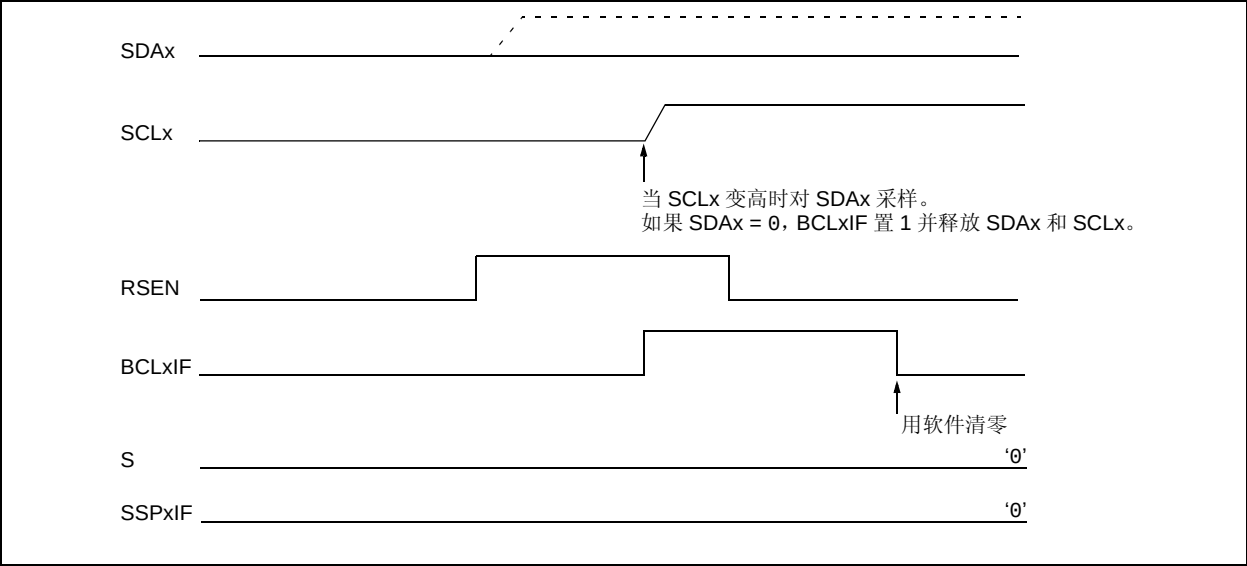
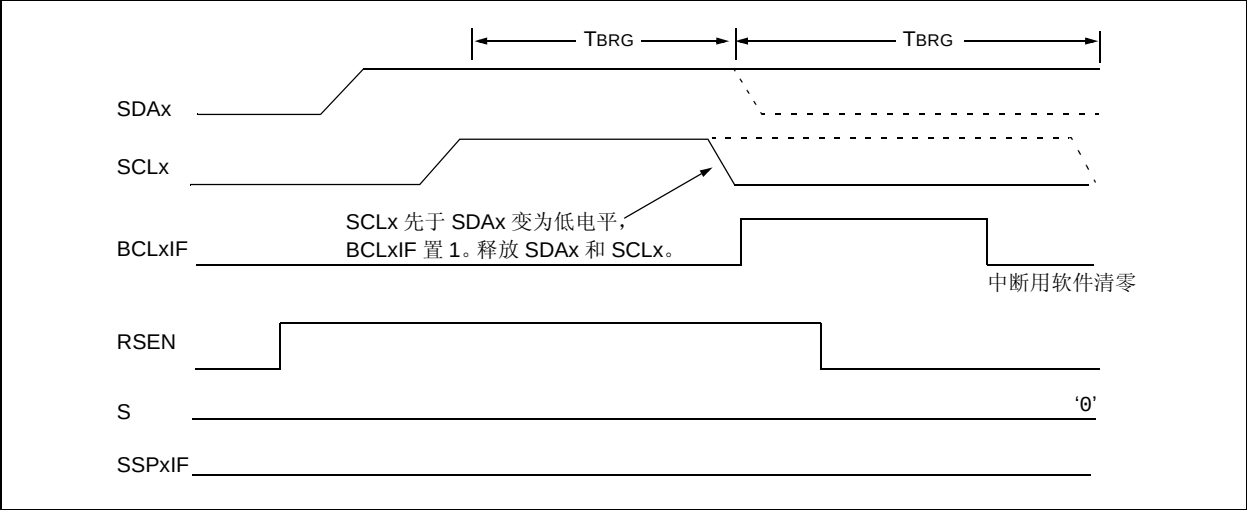


图 19-32: 重复启动条件期间的总线冲突（情形 2）



19.5.17.3 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 超时后被采样到低电平。
- b) SCLx 引脚被拉高之后，SCLx 在 SDAx 变成高电平之前被采样到低电平。

停止条件从 SDAx 被置成低电平开始。当 SDAx 采样为低电平时，SCLx 引脚被允许悬空。当 SDAx 采样为高电平（时钟仲裁）时，BRG 装入 SSPxADD<6:0> 的值并递减计数至 0。BRG 超时后，SDAx 被采样。如果 SDAx 采样为低电平，则已发生总线冲突。这是因为另一个主器件正试图发送一个数据 0（图 19-33）。如果 SCLx 引脚在允许 SDAx 悬空为高电平前被采样到低电平，也会发生总线冲突。这是另一个主器件正试图发送一个数据 0 的另外一种情况（图 19-34）。

图 19-33: 停止条件期间的总线冲突（情形 1）

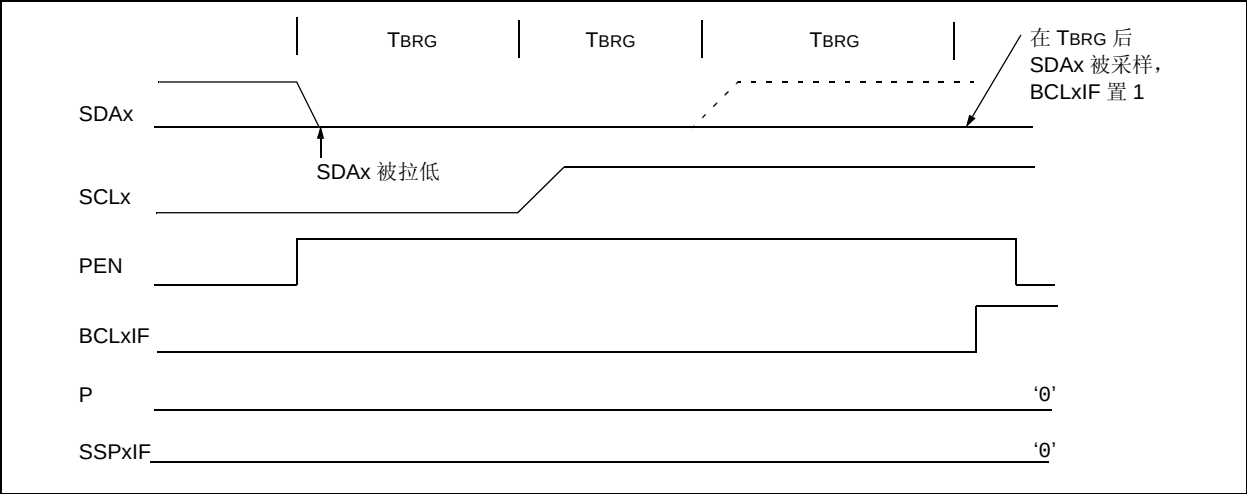
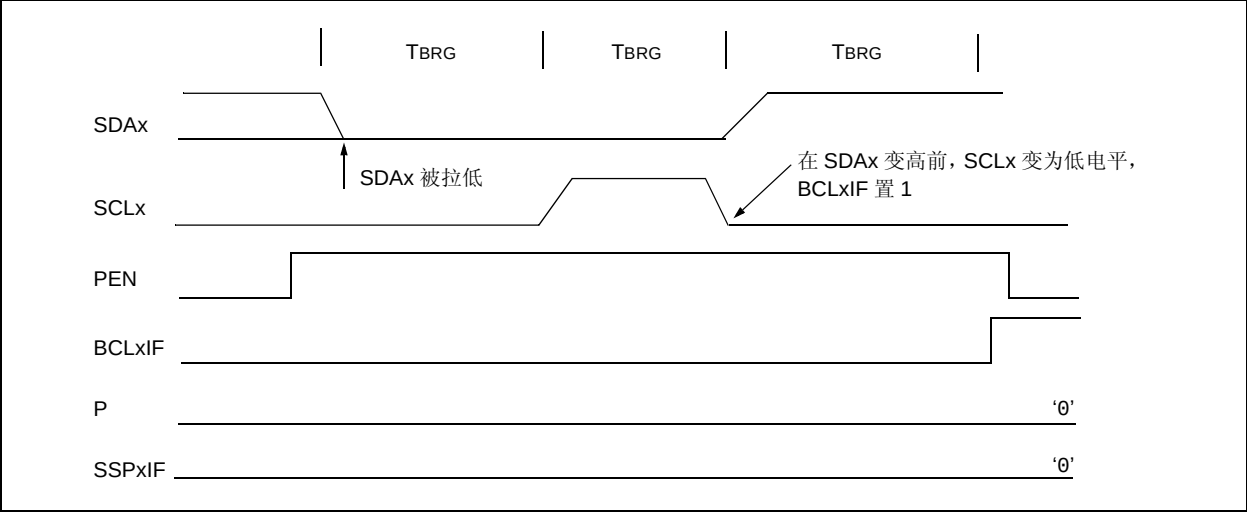


图 19-34: 停止条件期间的总线冲突（情形 2）



PIC18F46J50 系列

表 19-4: 与 I²C™ 操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽³⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽³⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP ⁽³⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	71
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	71
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	72
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	72
SSP1BUF	MSSP1 接收缓冲区 / 发送寄存器								70
SSPxADD	MSSP1 地址寄存器 (I ² C™ 从模式), MSSP1 波特率重载寄存器 (I ² C 主模式)								70, 73
SSPxMSK ⁽¹⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	70, 73
SSPxCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	70, 73
SSPxCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	70, 73
	GCEN	ACKSTAT	ADMSK5 ⁽²⁾	ADMSK4 ⁽²⁾	ADMSK3 ⁽²⁾	ADMSK2 ⁽²⁾	ADMSK1 ⁽²⁾	SEN	
SSPxSTAT	SMP	CKE	D/Ā	P	S	R/W	UA	BF	70, 73
SSP2BUF	MSSP2 接收缓冲区 / 发送寄存器								73
SSP2ADD	MSSP2 地址寄存器 (I ² C 从模式), MSSP2 波特率重载寄存器 (I ² C 主模式)								73

图注: — = 未实现, 读为 0。I²C™ 模式下的 MSSPx 模块不使用阴影单元。

- 注 1: SSPxMSK 与 SSPxADD 共用 SFR 空间中的相同地址, 但是只能在 7 位掩码模式下的某些 I²C 从模式操作中访问。更多详细信息, 请参见第 19.5.3.4 节 “7 位地址掩码模式”。
- 2: 仅在 I²C 从模式操作中使用的备用位定义。
- 3: 这些位仅在 44 引脚器件上可用。

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。(通常, EUSART 也称为“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。也可以将它配置为能与 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 模块还实现了其他特性, 包括自动波特率检测和校准、接收到同步间隔字符时自动唤醒和 12 位间隔字符发送。因为具有这些特性, 使其成为局域互联网 (Local Interconnect Network, LIN/J2602) 总线系统的理想选择。

PIC18F46J50 系列的所有产品都配备了两个独立的 EUSART 模块, 称为 EUSART1 和 EUSART2。可将这两个模块配置为以下几种工作模式:

- 异步模式 (全双工):
 - 接收到字符时自动唤醒
 - 自动波特率校准
 - 12 位间隔字符发送
- 同步 —— 主模式 (半双工), 时钟极性可选
- 同步 —— 从模式 (半双工), 时钟极性可选

EUSART1 和 EUSART2 的引脚分别与 PORTC (RC6/TX1/CK1/RP17 和 RC7/RX1/DT1/SDO1/RP18) 和可重映射引脚 (RPn1/TX2/CK2 和 RPn2/RX2/DT2) 的功能复用。要将这些引脚配置为 EUSART:

- 对于 EUSART1:
 - SPEN 位 (RCSTA1<7>) 必须置 1 (= 1)
 - TRISC<7> 位必须置 1 (= 1)
 - TRISC<6> 位必须清零 (= 0) 使该模块工作于异步和同步主模式
 - TRISC<6> 位必须置 1 (= 1) 使该模块工作于同步从模式
- 对于 EUSART2:
 - SPEN 位 (RCSTA2<7>) 必须置 1 (= 1)
 - 用于 RPn2/RX2/DT2 的 TRIS 位 = 1
 - 用于 RPn1/TX2/CK2 的 TRIS 位 = 0 使该模块工作于异步和同步主模式
 - TRISC<6> 位必须置 1 (= 1) 使该模块工作于同步从模式

注: EUSART 控制根据需要会自动将引脚从输入重新配置为输出。

TXx/CKx I/O 引脚具有可选的漏极开路输出功能。默认情况下, 当 EUSART 将该引脚用作输出时, 它将充当标准推挽式 CMOS 输出。通过将 ODCON2 寄存器中对应的 UxOD 位置 1 可使能 TXx/CKx I/O 引脚的漏极开路输出特性。更多详细信息, 请参见第 19.3.3 节“漏极开路输出选项”。

每个增强型 USART 模块的操作由以下 3 个寄存器控制:

- 发送状态和控制 (TXSTAx)
- 接收状态和控制 (RCSTAx)
- 波特率控制 (BAUDCONx)

这些寄存器的详细信息请分别参见寄存器 20-1、寄存器 20-2 和寄存器 20-3。

注: 在本章中, 与特定 EUSART 模块相关的寄存器名称和位名称的引用一般都采用以“x”代替特定模块编号的方式。因此, “RCSTAx”可能指 EUSART1 或 EUSART2 的接收状态寄存器。

PIC18F46J50 系列

寄存器 20-1: TXSTAx: 发送状态和控制寄存器 (位于快速操作存储区, 地址 FADh 和 FA8h)

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	SENDER	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 = POR 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:
无关位。

同步模式:
1 = 主模式 (时钟由内部 BRG 产生)
0 = 从模式 (时钟来自外部时钟源)
- bit 6

TX9: 9 位发送使能位

1 = 选择 9 位发送
0 = 选择 8 位发送
- bit 5

TXEN: 发送使能位 ⁽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

SENDER: 发送间隔字符位

异步模式:
1 = 在下一次发送时发送同步间隔字符 (完成时由硬件清零)
0 = 同步间隔字符发送完成

同步模式:
无关位。
- bit 2

BRGH: 高波特率选择位

异步模式:
1 = 高速
0 = 低速

同步模式:
在此模式下未使用。
- bit 1

TRMT: 发送移位寄存器状态位

1 = TSR 空
0 = TSR 满
- bit 0

TX9D: 发送数据的第 9 位

可以是地址 / 数据位或奇偶校验位。

注 1: 在同步模式下, SREN/CREN 可改写 TXEN。

寄存器 20-2: RCSTAx: 接收状态和控制寄存器 (位于快速操作存储区, 地址 FACH 和 F9Ch)

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 = POR 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 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 <u>异步模式:</u> 无关位。 <u>同步主模式:</u> 1 = 使能单字节接收 0 = 禁止单字节接收 此位在接收完成后清零。 <u>同步从模式:</u> 无关位。
bit 4	CREN: 连续接收使能位 <u>异步模式:</u> 1 = 使能接收器 0 = 禁止接收器 <u>同步模式:</u> 1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 的优先级高于 SREN) 0 = 禁止连续接收
bit 3	ADDEN: 地址检测使能位 <u>9 位异步模式 (RX9 = 1):</u> 1 = 当 RSR<8> 置 1 时, 使能地址检测, 允许中断并装入接收缓冲区 0 = 禁止地址检测, 接收所有字节并且第 9 位可作为奇偶校验位 <u>8 位异步模式 (RX9 = 0):</u> 无关位。
bit 2	FERR: 帧错误位 1 = 帧错误 (可以通过读 RCREGx 寄存器清零该位并接收下一个有效字节) 0 = 无帧错误
bit 1	OERR: 溢出错误位 1 = 溢出错误 (可以通过清零 CREN 位来清零该位) 0 = 无溢出错误
bit 0	RX9D: 接收数据的第 9 位 该位可以是地址 / 数据位或奇偶校验位, 并且必须由用户固件计算得到。

PIC18F46J50 系列

寄存器 20-3: **BAUDCONx: 波特率控制寄存器** (位于快速操作存储区, 地址 F7Eh 和 F7Ch)

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7 **ABDOVF:** 自动波特率采集计满返回状态位
1 = 在自动波特率检测模式下发生了 BRG 计满返回 (必须用软件清零)
0 = 没有发生 BRG 计满返回
- bit 6 **RCIDL:** 接收操作空闲状态位
1 = 接收操作处于空闲状态
0 = 接收操作处于活动状态
- bit 5 **RXDTP:** 数据 / 接收极性选择位
异步模式:
1 = 接收数据 (RXx) 反相 (低电平有效)
0 = 接收数据 (RXx) 未反相 (高电平有效)
同步模式:
1 = 数据 (DTx) 反相 (低电平有效)
0 = 数据 (DTx) 未反相 (高电平有效)
- bit 4 **TXCKP:** 同步时钟极性选择位
异步模式:
1 = 发送 (TXx) 的空闲状态为低电平
0 = 发送 (TXx) 的空闲状态为高电平
同步模式:
1 = 时钟 (CKx) 的空闲状态为高电平
0 = 时钟 (CKx) 的空闲状态为低电平
- 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 —— 仅限 SPBRGx (兼容模式), SPBRGHx 值被忽略
- 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 = 未监视 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 = 使能对下一个字符的波特率测量; 需要接收同步字段 (55h); 完成时由硬件清零
0 = 禁止波特率测量或测量已完成
同步模式:
在此模式下未使用。

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 寄存器对控制自由运行定时器的周期。在异步模式下，BRGH（TXSTAx<2>）和 BRG16（BAUDCONx<3>）也用于控制波特率。在同步模式下，BRGH 被忽略。

表 20-1 提供了不同 EUSART 模式下波特率的计算公式，仅适用于主模式（内部生成的时钟）。

给出目标波特率和 Fosc 值，就可以使用表 20-1 中的公式计算 SPBRGHx:SPBRGx 寄存器的最近似整数值。这样就可以确定波特率误差。例 20-1 给出了一个计算示例。表 20-2 中提供了不同异步模式下典型的波特率和误差值。使用高波特率（BRGH = 1）或 16 位 BRG 有助于降低波特率误差，或者在快速振荡器频率下取得较缓慢的波特率。

将新值写入 SPBRGHx:SPBRGx 寄存器会导致 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。

当工作在同步模式下时，SPBRGH:SPBRG 值 0000h 和 0001h 不受支持。在异步模式下，BRG 的所有值均可使用。

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一个不同的频率下。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/PMA4/RX1/DT1/SDO1/RP18 或 RPN/RX2/DT2）上的数据采样三次，以判断 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	$F_{osc}/[64 (n + 1)]$
0	0	1	8 位 / 异步	$F_{osc}/[16 (n + 1)]$
0	1	0	16 位 / 异步	
0	1	1	16 位 / 异步	$F_{osc}/[4 (n + 1)]$
1	0	x	8 位 / 同步	
1	1	x	16 位 / 同步	

图注： x = 无关位， n = SPBRGHx:SPBRGx 寄存器对的值

PIC18F46J50 系列

例 20-1: 计算波特率误差

器件工作在 Fosc = 16 MHz，目标波特率 = 9600，异步模式， 8 位 BRG:									
目标波特率 = Fosc/(64 ([SPBRGHx:SPBRGx] + 1))									
求解 SPBRGHx:SPBRGx:									
X	=	((Fosc/ 目标波特率)/64) - 1							
	=	((16000000/9600)/64) - 1							
	=	[25.042] = 25							
计算波特率 = 16000000/(64 (25 + 1))									
	=	9615							
误差	=	(计算波特率 - 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	复位值所在页:
TXSTAx	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								72
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								71

图注: — = 未实现，读为 0。BRG 不使用阴影单元。

表 20-3: 异步模式下的波特率

波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

PIC18F46J50 系列

表 20-3: 异步模式下的波特率 (续)

波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	0.300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1.201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2.403	-0.16	207
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.04	832	0.300	-0.16	415	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	0.300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1.200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2.400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9.615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19.230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57.142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117.64	-2.12	16

波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)	实际 波特率 (K)	% 误差	SPBRG 值 (十进制)
0.3	0.300	0.01	3332	0.300	-0.04	1665	0.300	-0.04	832
1.2	1.200	0.04	832	1.201	-0.16	415	1.201	-0.16	207
2.4	2.404	0.16	415	2.403	-0.16	207	2.403	-0.16	103
9.6	9.615	0.16	103	9.615	-0.16	51	9.615	-0.16	25
19.2	19.231	0.16	51	19.230	-0.16	25	19.230	-0.16	12
57.6	58.824	2.12	16	55.555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

20.1.3 自动波特率检测

增强型 USART 模块支持波特率自动检测和校准。此功能仅在异步模式下当 WUE 位清零时有效。

只要接收到启动位并且 ABDEN 位已置 1，就会开始自动波特率测量序列（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 的时钟反向。BRG 并不为进入的 RXx 信号提供时钟信号，而是由 RXx 信号为 BRG 定时。在 ABD 模式下，内部 BRG 用作计数器计算进入的串行字节流的位周期。

一旦 ABDEN 位置 1，状态机就会清零 BRG 并寻找启动位。要计算正确的比特率，ABD 必须接收到值为 55h 的字节（ASCII 码为“U”，也是 LIN/J2602 总线的同步字符）。为了尽量减少输入信号不对称造成的影响，测量时段内要包含一个高位和一个低位时间。在启动位后，SPBRGx 使用预先选择的时钟源在 RXx 的第一个上升沿开始递增计数。在 RXx 引脚传输了 8 个位，或在检测到第 5 个上升沿后，会将相应 BRG 周期内的累加值保存在 SPBRGHx:SPBRGx 寄存器对中。当第 5 个边沿出现时（应与停止位对应），ABDEN 位会自动清零。

如果发生了 BRG 计满返回（从 FFFFh 到 0000h 的溢出），会在 ABDOVF 状态位（BAUDCONx<7>）有所反映。该位可在 BRG 计满返回时由硬件置 1，也可以由用户通过软件置 1 或清零。在发生计满返回事件后，ABD 模式继续有效，ABDEN 位保持置 1（图 20-2）。

在校准波特率周期时，BRG 寄存器时钟频率为预配置时钟频率的 1/8。请注意 BRG 时钟将由 BRG16 和 BRGH 位配置。必须将 BRG16 位置 1，才能将 SPBRGx 和 SPBRGHx 都用作 16 位计数器。用户通过检查 SPBRGHx 寄存器中的值是否为 00h，可以验证 8 位模式下是否发生了进位。

表 20-4 所示为 BRG 计数器的时钟速率。

当发生 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 中的值以清零中断标志位 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:

需要由用户来判断输入字符的波特率是否处于所选 BRG 时钟源范围内。由于位错误率的原因，某些振荡器频率和 EUSART 波特率的组合是无法实现的。使用自动波特率检测功能时，必须综合考虑系统总的时序和通信波特率。
- 3:

要使波特率范围最大，建议在使用自动波特率功能时将 BRG16 位置 1。

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/64
1	1	Fosc/32

注:

在 ABD 序列期间，不管 BRG16 如何设置，SPBRGx 和 SPBRGHx 都被用作 16 位计数器。

20.1.3.1 ABD 和 EUSART 发送

由于 ABD 采集期间 BRG 时钟是反相的，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序列期间 ABDEN 不能为置 1 状态，否则可能会导致无法预料的 EUSART 操作。

图 20-1: 自动波特率计算

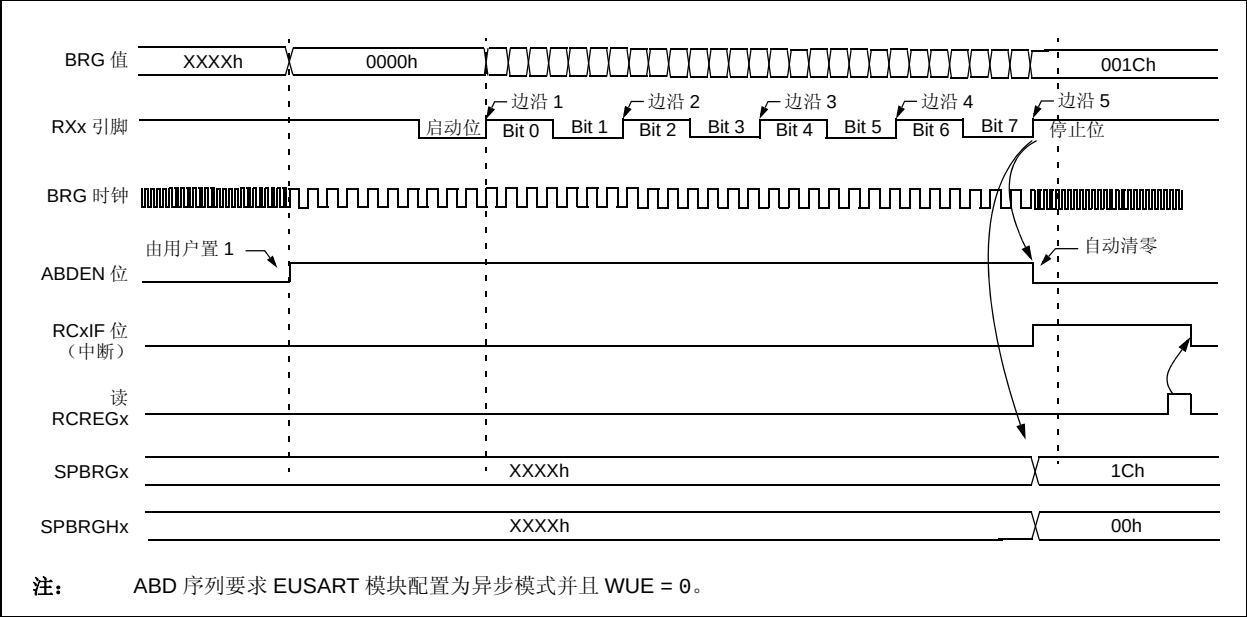
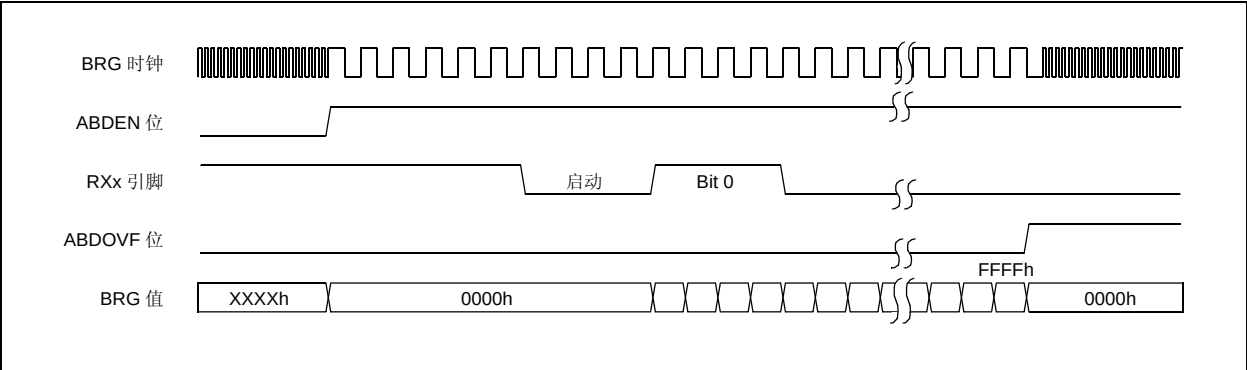


图 20-2: BRG 溢出序列



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 使用标准不归零码 (Non-Return-to-Zero, NRZ) 格式 (1 个启动位、8 个或 9 个数据位和 1 个停止位)。最常见的数据格式为 8 位。片上专用 8 位 /16 位 BRG 可用于从振荡器产生标准波特率频率。

EUSART 先发送和接收 LSB。EUSART 的发送器和接收器在功能上是相互独立的, 但使用相同的数据格式和波特率。根据 BRGH 和 BRG16 位 (TXSTAx<2> 和 BAUDCONx<3>) 的设置情况, BRG 会产生一个 16 倍或 64 倍数据移位速率的时钟信号。硬件不支持奇偶校验, 但可通过软件实现并作为第 9 个数据位存储。

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
- 接收到同步间隔字符时自动唤醒
- 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 给出了 EUSART 发送器框图。

发送器的核心是发送 (串行) 移位寄存器 (Transmit Shift Register, TSR)。移位寄存器从读 / 写发送缓冲寄存器 TXREGx 中获取数据。通过软件将数据装入 TXREGx 寄存器。在前一次装入数据的停止位发送前, 不会向 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) 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 TXxIF 标志位置 1。可以通过将中断允许位 TXxIE 置 1 或清零来允许或禁止该中断。不管 TXxIE 的状态如何, TXxIF 都将置 1; 不能用软件清零。TXxIF 也不会因 TXREGx 装入新数据时立即被清零, 而是在装入指令后的第二个指令周期被清零。因此在 TXREGx 装入新数据后立即查询 TXxIF, 将返回无效结果。

TXxIF 指示 TXREGx 寄存器的状态, 而另一个位 TRMT (TXSTAx<1>) 则指示 TSR 寄存器的状态。TRMT 是只读位, 它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断逻辑均无关联, 因此要确定 TSR 寄存器是否为空, 用户只能对此位进行查询。

注 1: TSR 寄存器不映射到数据存储中, 因此用户无法使用。

2: 当使能位 TXEN 置 1 时, 标志位 TXxIF 置 1。

设置异步发送操作的步骤如下:

1. 对 SPBRGHx:SPBRGx 寄存器进行初始化, 设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零, 获得所需的波特率。
2. 通过清零 SYNC 位并将 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. 如果需要 9 位发送, 将发送位 TX9 置 1。可以作为地址 / 数据位使用。
5. 通过将 TXEN 位置 1 使能发送, 这也将导致 TXxIF 位置 1。
6. 如果选择了 9 位发送, 应将第 9 位装入 TX9D 位。
7. 将数据装入 TXREGx 寄存器 (启动发送)。
8. 如果使用中断, 应确保 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

PIC18F46J50 系列

图 20-3: EUSART 发送框图

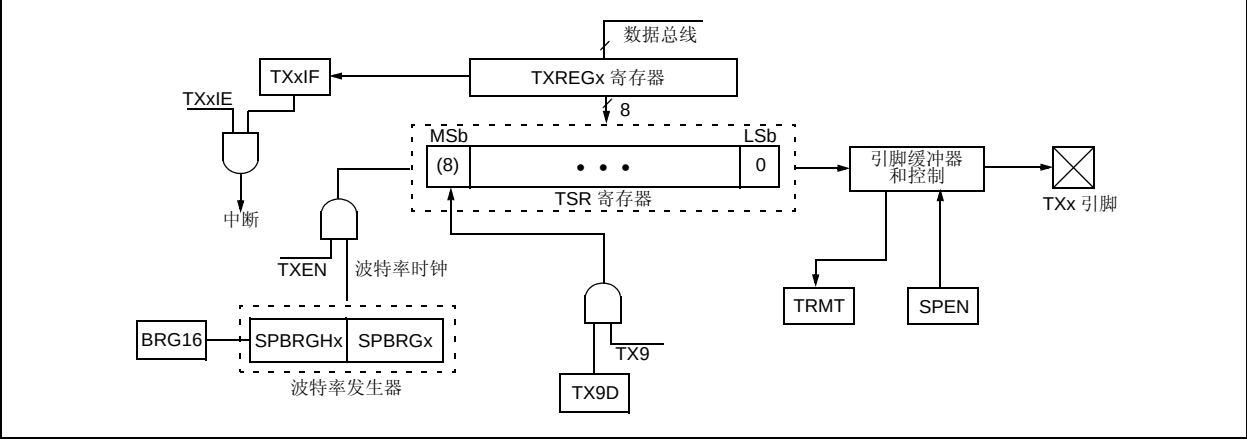


图 20-4: 异步发送

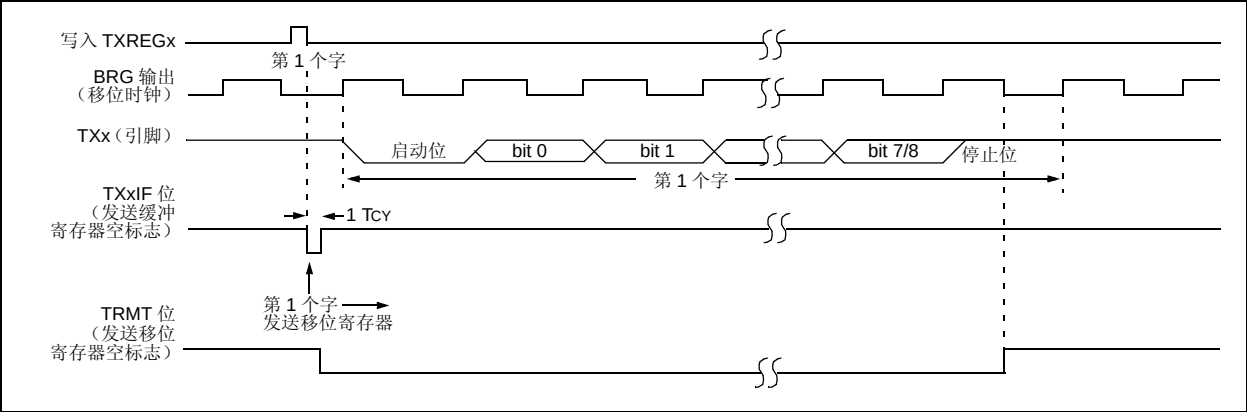


图 20-5: 异步发送（背对背）

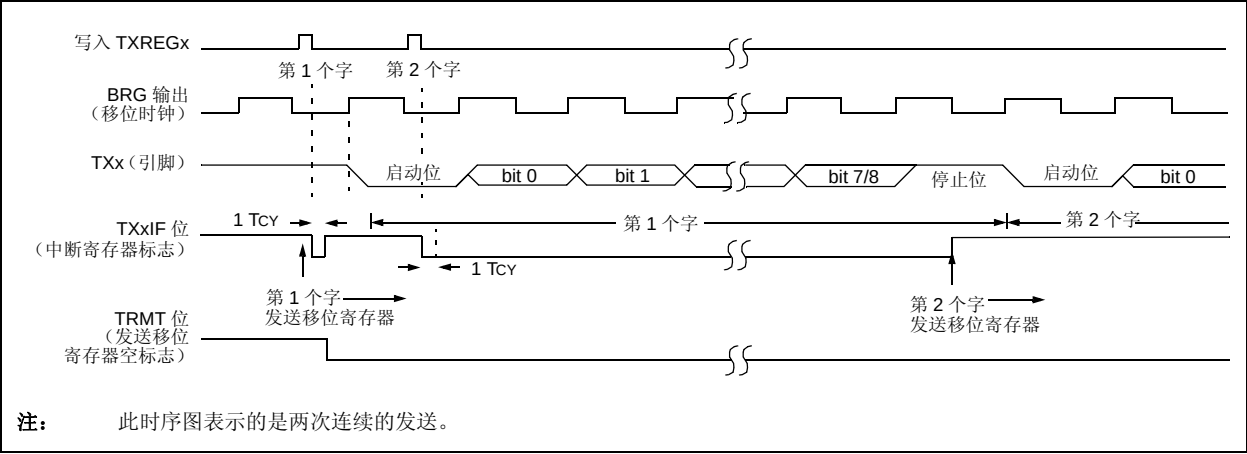


表 20-5: 与异步发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
TXREGx	EUSARTx 发送寄存器								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXDTP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								72
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								71
ODCON2	—	—	—	—	—	—	U2OD	U1OD	74

图注: — = 未实现位，读为 0。异步发送不使用阴影单元。

注 1: 这些位仅在 44 引脚器件上可用。

PIC18F46J50 系列

20.2.2 EUSART 异步接收器

图 20-6 给出了接收器框图。在 RXx 引脚上接收数据，并驱动数据恢复电路。数据恢复模块实际上是一个工作频率为 16 倍波特率的高速移位器，而主接收串行移位器的工作频率等于比特率或 Fosc。此模式通常用于 RS-232 系统中。

20.2.2.1 接收数据

接收器数据恢复电路在数据第一位的下降沿开始接收数据。在设置完 RXDTP 位之后传输的第一位也称为起始位，始终为零。起始位之后是待接收数据字符的最低位。接收到每一位后，都会对位值进行采样，然后将位移入接收移位寄存器（Receive Shift Register, RSR）。当字符的所有 8 或 9 个数据位（可由用户选择具体的位数）都被移入后，会测量最后一位的时间并采样其电平。最后一位为停止位，应始终为 1（在设置完 RXDTP 位之后）。如果数据恢复电路在停止位的位置采样到 0，那么就会将当前字符的帧错误（Framing Error, FERR）标志置 1，否则清零。

一旦字符的所有数据位和停止位都接收到后，RSR 中的数据位就会立即被传输到双字符先进先出（First-In-First-Out, FIFO）存储区。FIFO 缓冲机制允许在需要软件处理 EUSART 接收器之前接收两个完整的字符。软件无法直接访问 RSR 寄存器。固件可通过读取 RCREGx 寄存器读取来自 FIFO 的数据。每次固件开始读取 RCREGx 寄存器都会使 FIFO 前进至下一个字符，若 FIFO 中没有其他数据，就会将接收中断标志位（RCxIF）清零。

20.2.2.2 接收溢出错误

在 FIFO 已满时，如果用户固件允许在读取 RCREGx 前接收第三个字符，就会发生缓冲区溢出错误。在这样的情形下，硬件将会阻止 RSR 中的内容复制到接收 FIFO，后续字符将会丢失，并且 RCSTAx 寄存器中的 OERR 状态位将置 1。如果出现了 OERR 条件，那么固件必须通过先清零然后重新置 1 CREN 位清除该条件，从而成功接收更多的字符。

注： 如果接收 FIFO 溢出，那么在溢出条件清除前，不会接收更多字符。

20.2.2.3 设置异步接收

设置异步接收操作的步骤如下：

1. 对 SPBRGHx:SPBRGx 寄存器进行初始化，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，获得所需的波特率。
2. 通过清零 SYNC 位并将 SPEN 位置 1，使能异步串口。
3. 如果需要中断，将中断允许位 RCxIE 置 1。
4. 如果需要接收 9 位数据，将 RX9 位置 1。
5. 通过将 CREN 位置 1 使能接收。
6. 接收完成时标志位 RCxIF 将被置 1，此时如果中断允许位 RCxIE 已置 1，还将产生一个中断。
7. 读取 RCSTAx 寄存器取得第 9 位（如果已使能），并确定接收时是否发生了错误。
8. 读 RCREGx 寄存器来读取接收到的 8 位数据。
9. 如果发生错误，将使能位 CREN 清零以清除错误。
10. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

20.2.2.4 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。设置使能地址检测的异步接收的步骤如下：

1. 对 SPBRGHx:SPBRGx 寄存器进行初始化，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，获得所需的波特率。
2. 通过清零 SYNC 位并将 SPEN 位置 1，使能异步串口。
3. 如果需要中断，将 RCEN 位置 1 并用 RCxIP 位选择所需的优先级。
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. 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（如果适用）。
9. 读 RCREGx 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. 如果已寻址到器件，将 ADDEN 位清零以允许所有接收到的数据被送入接收缓冲区并发中断信号给 CPU。

图 20-6: EUSARTx 接收框图

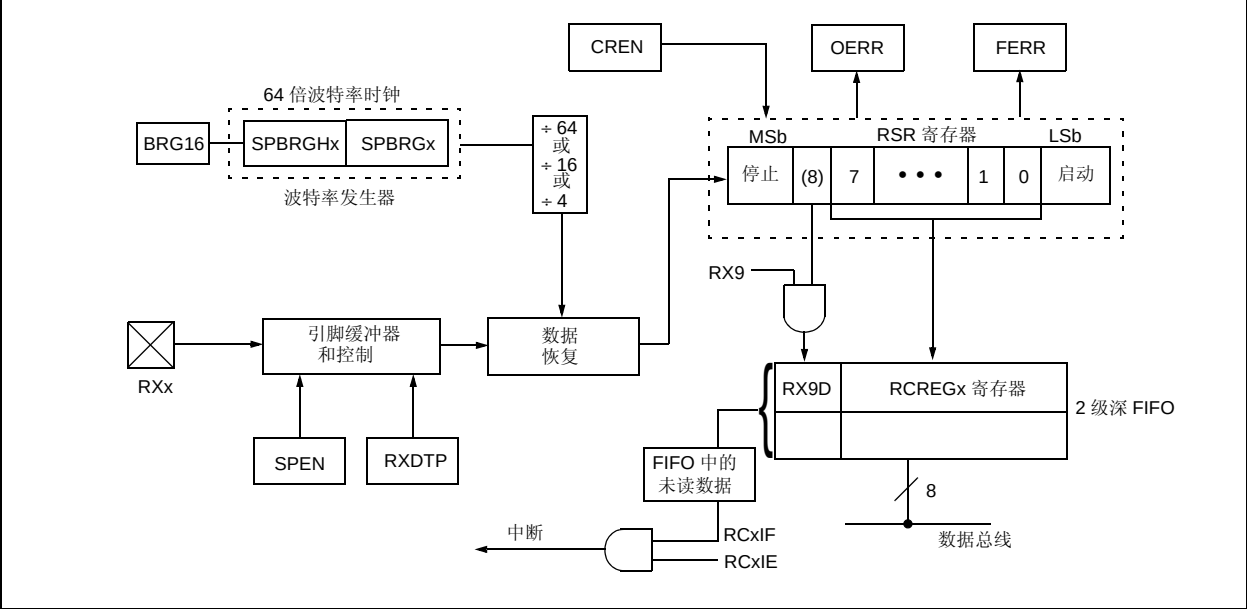
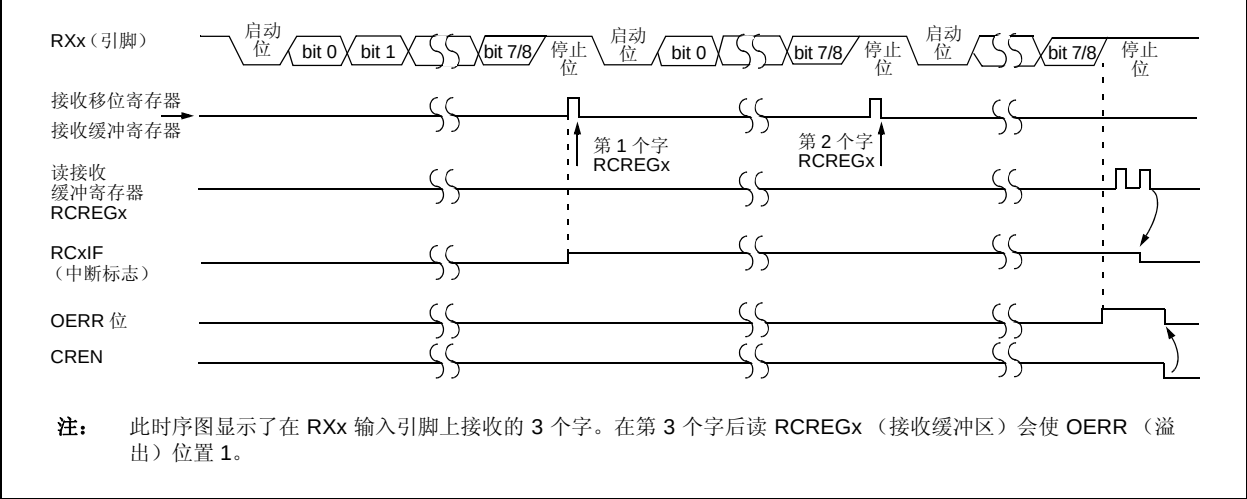


图 20-7: 异步接收



PIC18F46J50 系列

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 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIp ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	71
RCREGx	EUSARTx 接收寄存器								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								72
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								71

图注: — = 未实现位, 读为 0。异步接收不使用阴影单元。

注 1: 这些位仅在 44 引脚器件上可用。

20.2.3 接收到同步间隔字符时自动唤醒

在休眠模式下, EUSART 的所有时钟都会暂停。因此, BRG 处于不工作状态, 不能正常进行字节接收。当 EUSART 工作在异步模式下时, 自动唤醒功能使控制器可被 RXx/DTx 线上的活动唤醒。

通过将 WUE 位 (BAUDCONx<1>) 置 1 使能自动唤醒功能。一旦置 1, RXx/DTx 上的典型接收序列就被禁止, EUSART 保持在空闲状态, 监视与 CPU 模式无关的唤醒事件。唤醒事件包含 RXx/DTx 线上由高至低的跳变。(这与同步间隔字符的开始或 LIN/J2602 支持协议的唤醒信号字符一致。)

唤醒事件后, 模块产生一个 RCxIF 中断。在正常工作模式下, 中断产生与 Q 时钟同步 (图 20-8), 而器件处于休眠模式时则异步发生 (图 20-9)。通过读 RCREGx 寄存器可清除中断条件。

唤醒事件后, 一旦 RXx 线上出现由低至高的跳变, WUE 位自动清零。此时, EUSART 模块将从空闲模式返回正常工作模式。这向用户表明同步间隔事件结束。

20.2.3.1 使用自动唤醒功能的特殊注意事项

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会产生错误的结束字符 (End-Of-Character, EOC) 并导致数据或帧错误。因此, 为了确保正常的传输, 必须首先发送全 0 字符。对于标准的 RS-232 器件, 该字符是 00h (8 字节); 而对于 LIN/J2602 总线器件则是 000h (12 位)。

另外还必须考虑振荡器起振时间, 尤其在使用具有较长起振时间的振荡器 (即, HS 或 HSPLL 模式) 的应用中。同步间隔 (或唤醒信号) 字符必须足够长, 并随后有一个足够长的间隔时间, 以使所选的振荡器有足够的时间起振并正确初始化 EUSART。

20.2.3.2 使用 WUE 位的特殊注意事项

使用 WUE 和 RCxIF 事件的时序来判断接收数据的有效性时，有可能会引起一些混淆。如前所述，将 WUE 位置 1 会使 EUSART 进入空闲状态。唤醒事件会通过将 RCxIF 位置 1 产生一个接收中断。此后当 RXx/DTx 上出

现上升沿时，WUE 位被清零。然后通过读 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）同时 RCxIF 标志位置 1 并不能表明 RCREGx 中数据接收是完整的。用户应考虑

图 20-8: 正常操作时的自动唤醒位（WUE）时序

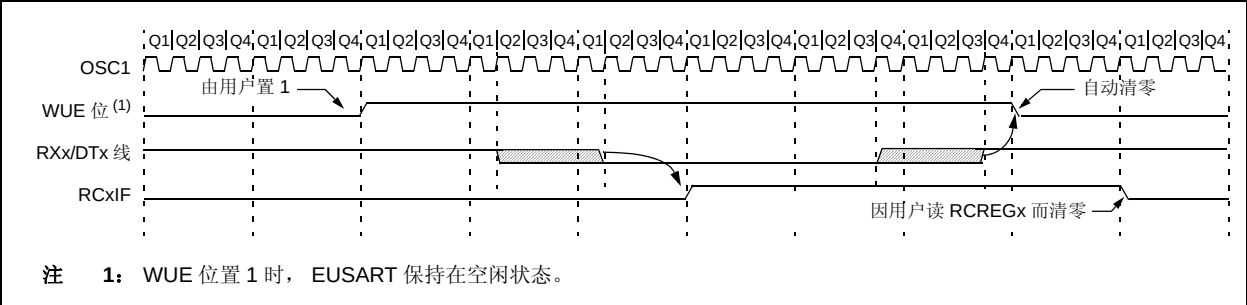
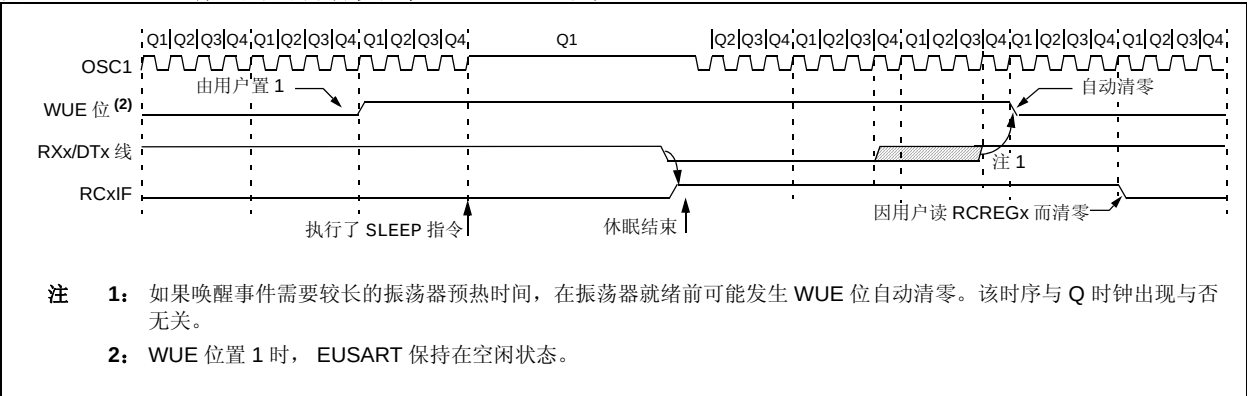


图 20-9: 休眠时的自动唤醒位（WUE）时序



20.2.4 间隔字符序列

EUSART 模块能够发送符合 LIN/J2602 总线标准的特殊间隔字符序列。发送的间隔字符包含 1 个启动位，随后的 12 个 0 位和 1 个停止位。当发送移位寄存器装入数据时，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发送全 0。

在发送了相应的停止位后，硬件会自动将 SENDB 位复位。这样用户可以在间隔字符 (在 LIN/J2602 规范中通常是同步字符) 后预先将下一个要发送字节装入发送 FIFO。

请注意在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。

正如在正常发送操作中一样，TRMT 位表明发送正在进行还是处于空闲状态。关于发送间隔字符的时序，请参见图 20-10。

20.2.4.1 间隔和同步发送序列

以下序列将发送一个报文帧头 (header)，它由一个间隔字符和其后的自动波特率同步字节组成。这是 LIN/J2602 总线主器件的典型序列。

- 1. 将 EUSART 配置为所需的模式。
- 2. 将 TXEN 和 SENDB 位置 1 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，以便将同步字符装入 FIFO 缓冲区。
- 5. 发送间隔字符后，SENDER 位被硬件复位。此时，同步字符会以预先配置的模式发送。

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

20.2.5 接收间隔字符

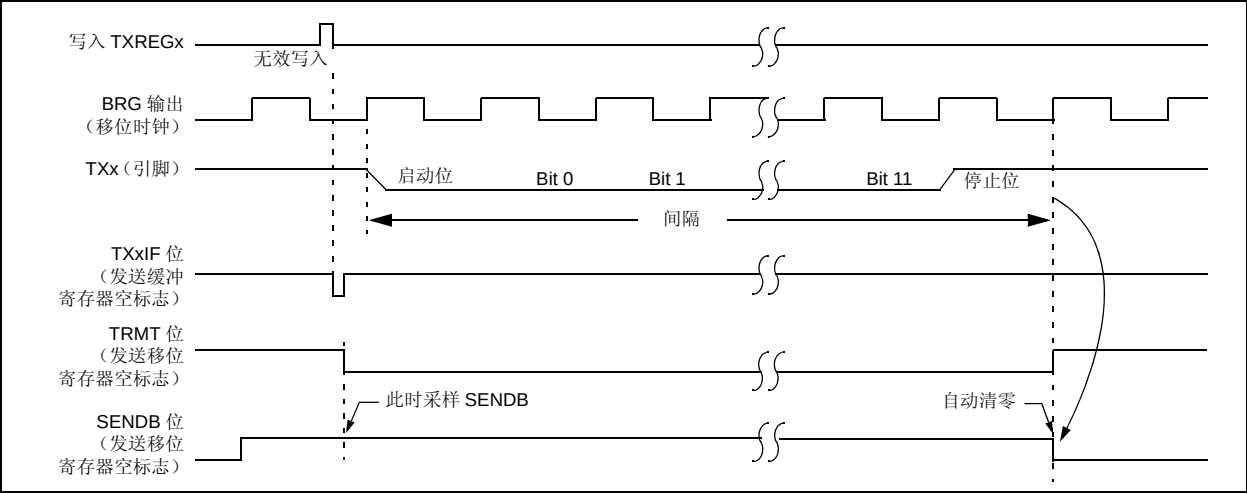
增强型 USART 模块接收间隔字符有两种方法。

第一种方法是强制将波特率配置为典型速度的 9/13。这可以使停止位在正确的采样点 (对于间隔字符为启动位之后的 13 位，对于典型数据则是 8 个数据位) 跳变。

第二种方法采用第 20.2.3 节 “接收到同步间隔字符时自动唤醒” 中所述的自动唤醒功能。通过使能此功能，EUSART 将采样 RXx/DTx 上的两次跳变，产生一个 RCxIF 中断，并接收下一个数据字节并再产生一次中断。

请注意在间隔字符后，用户通常希望使能自动波特率检测功能。采用这两种方法时，用户均可在检测到 TXxIF 中断时立即将 ABDEN 位置 1。

图 20-10: 发送间隔字符序列



20.3 EUSART 同步主模式

将 CSRC 位 (TXSTAx<7>) 置 1 可进入同步主模式。在此模式下, 数据以半双工方式发送 (即发送和接收不能同时进行)。发送数据时禁止接收, 反之亦然。将 SYNC 位 (TXSTAx<4>) 置 1 可进入同步模式。此外, 应将使能位 SPEN (RCSTAx<7>) 置 1, 分别将 TXx 和 RXx 引脚配置为 CKx (时钟) 和 DTx (数据) 线。

主模式意味着处理器在 CKx 线上发送主时钟信号。时钟极性通过 TXCKP 位 (BAUDCONx<4>) 选择。将 TXCKP 置 1 将 CKx 的空闲状态设为高电平, 将该位清零则将 CKx 的空闲状态设为低电平。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 给出了 EUSART 发送器框图。发送器的核心是发送 (串行) 移位寄存器 (TSR)。移位寄存器从读 / 写发送缓冲寄存器 TXREGx 中获取数据。用软件将数据装入 TXREGx 寄存器。在前一次装入数据的最后一位发送完成后, 才向 TSR 寄存器装入新数据。一旦最后一位发送完成, 就会将 TXREGx 寄存器的新数据 (如果有) 装入 TSR。

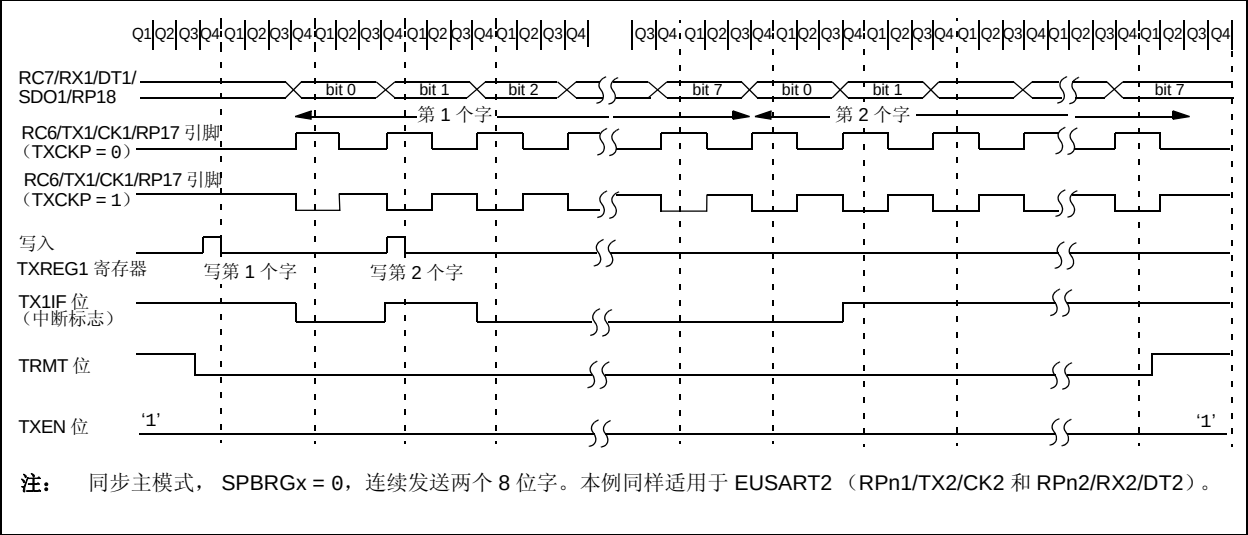
一旦 TXREGx 寄存器向 TSR 寄存器传输了数据 (在 1 个 Tcy 内发生), TXREGx 寄存器就为空, 同时 TXxIF 标志位被置 1。可以通过将中断允许位 TXxIE 置 1 或清零来允许或禁止该中断。TXxIF 的设置不受使能位 TXxIE 状态的影响, 且不能用软件清零。只有在新数据装入 TXREGx 寄存器时, TXxIF 才会复位。

标志位 TXxIF 表示 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 位是一个只读位, 当 TSR 为空时, TRMT 被置 1。TRMT 位与任何中断均无关联, 因此要确定 TSR 寄存器是否为空, 用户只能对此位进行查询。TSR 不映射到数据存储区中, 因此用户无法使用。

设置同步主发送操作的步骤如下:

- 1. 对 SPBRGHx:SPBRGx 寄存器进行初始化, 设置合适的波特率。按需要将 BRG16 位置 1 或清零, 获得所需的波特率。
- 2. 通过将 SYNC、SPEN 和 CSRC 位置 1, 使能同步主串口。
- 3. 如果需要中断, 将中断允许位 TXxIE 置 1。
- 4. 如果需要 9 位发送, 将 TX9 位置 1。
- 5. 通过将 TXEN 位置 1 使能发送。
- 6. 如果选择了 9 位发送, 应将第 9 位装入 TX9D 位。
- 7. 将数据装入 TXREGx 寄存器, 启动发送。
- 8. 如果使用中断, 应确保 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 20-11: 同步发送



PIC18F46J50 系列

图 20-12: 同步发送（由 TXEN 位控制）

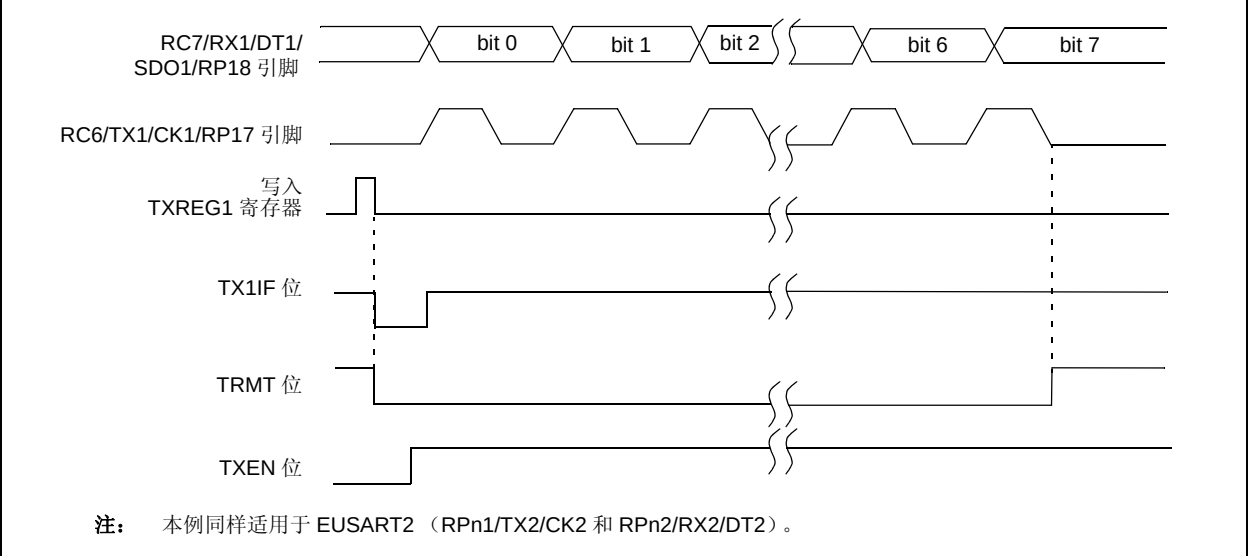


表 20-7: 与同步主发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPPI ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
TXREGx	EUSARTx 发送寄存器								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								72
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								71
ODCON2	—	—	—	—	—	—	U2OD	U1OD	74

图注： — = 未实现，读为 0。同步主发送不使用阴影单元。

注 1: 这些引脚仅在 44 引脚器件上可用。

20.3.2 EUSART 同步主接收

一旦选择了同步模式，可通过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 使能接收。在时钟的下降沿采样 RXx 引脚上的数据。

如果使能位 SREN 置 1，则只接收单个字。如果将使能位 CREN 置 1，则会连续接收数据，直到 CREN 位清零。如果两个位均被置 1，则 CREN 优先。

设置同步主接收操作的步骤如下：

1. 对 SPBRGHx:SPBRGx 寄存器进行初始化，设置合适的波特率。按需要将 BRG16 位置 1 或清零，获得所需的波特率。

2. 通过将 SYNC、SPEN 和 CSRC 位置 1，使能同步主串口。
3. 确保将 CREN 和 SREN 位清零。

4. 如果需要中断，将中断允许位 RCxIE 置 1。

5. 如果需要接收 9 位数据，将 RX9 位置 1。

6. 如果需要单字节接收，将 SREN 位置 1。如果需要连续接收，将 CREN 位置 1。

7. 接收完成时中断标志位 RCxIF 将被置 1，此时如果中断允许位 RCxIE 已置 1，还将产生一个中断。

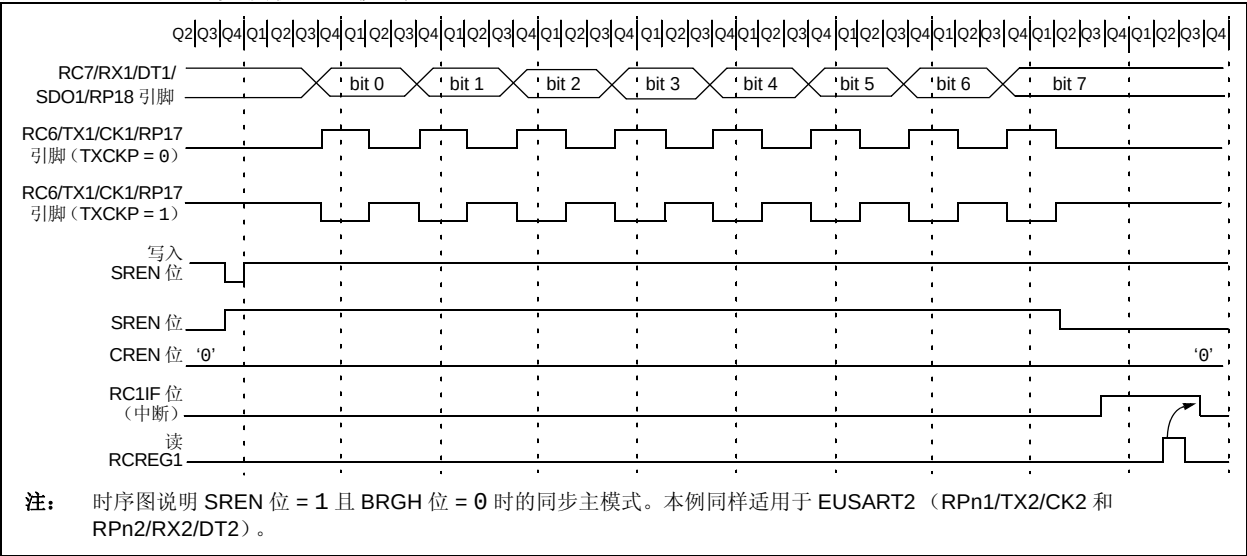
8. 读取 RCSTAx 寄存器取得第 9 位（如果已使能），并确定接收时是否发生了错误。

9. 读 RCREGx 寄存器来读取接收到的 8 位数据。

10. 如果发生错误，将 CREN 位清零以清除错误。

11. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 20-13: 同步接收（主模式，SREN）



PIC18F46J50 系列

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 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
RCREGx	EUSARTx 接收寄存器								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								72
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								71
ODCON2	—	—	—	—	—	—	U2OD	U1OD	74

图注: — = 未实现，读为 0。同步主接收不使用阴影单元。

注 1: 这些引脚仅在 44 引脚器件上可用。

20.4 EUSART 同步从模式

将 CSRC (TXSTAx<7>) 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 (主模式中由内部时钟提供)。这使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原理是相同的。

如果向 TXREGx 写两个字，然后执行 SLEEP 指令，则会发生以下事件：

- 第一个字将立即传送到 TSR 寄存器并发送。
- 第二个字将保留在 TXREGx 寄存器中。
- 标志位 TXxIF 不会被置 1。
- 第一个字移出 TSR 后，TXREGx 寄存器会将第二个字传送到 TSR，此时标志位 TXxIF 将置 1。

- 如果中断允许位 TXxIE 置 1，中断将把芯片从休眠状态唤醒。如果允许了全局中断，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：

- 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零，使能同步从串口。
- 清零 CREN 和 SREN 位。
- 如果需要中断，将中断允许位 TXxIE 置 1。
- 如果需要 9 位发送，将 TX9 位置 1。
- 通过将使能位 TXEN 置 1 使能发送。
- 如果选择了 9 位发送，应将第 9 位装入 TX9D 位。
- 将数据装入 TXREGx 寄存器，启动发送。
- 如果使用中断，应确保 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	复位值所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPiP ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMUIF	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMUIE	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMUIP	TMR3GIP	RTCCIP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
TXREGx	EUSARTx 发送寄存器								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								72
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								71

图注: — = 未实现，读为 0。同步从发送不使用阴影单元。

注 1: 这些引脚仅在 44 引脚器件上可用。

PIC18F46J50 系列

20.4.2 EUSART 同步从接收

除了休眠模式、任何空闲模式以及在从模式下忽略 SREN 位的情况以外，同步主模式和从模式的工作原理是相同的。

如果在进入休眠或任何空闲模式前将 CREN 位置 1 使能接收，那么在低功耗模式下可以接收到一个数据字。接收到该字后，RSR 寄存器将把数据传送到 RCREGx 寄存器。如果中断允许位 RCxIE 已置 1，产生的中断将把芯片从低功耗模式唤醒。如果允许了全局中断，程序将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，使能同步从串口。
2. 如果需要中断，将中断允许位 RCxIE 置 1。
3. 如果需要接收 9 位数据，将 RX9 位置 1。
4. 将使能位 CREN 置 1 以使能接收。
5. 接收完成时 RCxIF 位将被置 1。如果中断允许位 RCxIE 置 1，还将产生一个中断。
6. 读取 RCSTAx 寄存器取得第 9 位（如果已使能），并确定接收时是否发生了错误。
7. 读 RCREGx 寄存器来读取接收到的 8 位数据。
8. 如果发生错误，将 CREN 位清零以清除错误。
9. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）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 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMP1E ⁽¹⁾	AD1E	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMP1P ⁽¹⁾	AD1P	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	TMR4IF	CTMU1F	TMR3GIF	RTCCIF	71
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	TMR4IE	CTMU1E	TMR3GIE	RTCCIE	71
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	TMR4IP	CTMU1P	TMR3GIP	RTCCIP	71
RCSTAx	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	72
RCREGx	EUSARTx 接收寄存器								71
TXSTAx	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	71
BAUDCONx	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	73
SPBRGHx	EUSARTx 波特率发生器寄存器的高字节								72
SPBRGx	EUSARTx 波特率发生器寄存器的低字节								71

图注： — = 未实现，读为 0。同步从接收不使用阴影单元。

注 1: 这些引脚仅在 44 引脚器件上可用。

21.0 10 位模数转换器 (A/D) 模块

28 引脚器件的模数 (A/D) 转换器模块有 10 路输入，44 引脚器件的 A/D 转换器模块有 13 路输入。此外，还有两个内部通道，用于采样 VDDCORE 和 VBG 绝对参考电压。该模块能将一个模拟输入信号转换成相应的 10 位数字。

该模块有 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 0 (ANCON0)
- A/D 端口配置寄存器 1 (ANCON1)
- A/D 结果寄存器 (ADRESH 和 ADRESL)

寄存器 21-1 中的 ADCON0 寄存器用于控制 A/D 模块的工作。寄存器 21-2 中的 ADCON1 寄存器用于配置 A/D 时钟源、编程采集时间和对齐方式。

寄存器 21-3 和寄存器 21-4 中的 ANCON0 和 ANCON1 寄存器用于配置端口引脚的功能。

寄存器 21-1: ADCON0: A/D 控制寄存器 0 (位于快速操作存储区, 地址 FC2h)

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 ⁽²⁾	CHS2 ⁽²⁾	CHS1 ⁽²⁾	CHS0 ⁽²⁾	GO/DONE ⁽³⁾	ADON
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **VCFG1:** 参考电压配置位 (VREF- 电压源)
1 = VREF- (AN2)
0 = AVSS⁽⁴⁾

bit 6 **VCFG0:** 参考电压配置位 (VREF+ 电压源)
1 = VREF+ (AN3)
0 = AVDD⁽⁴⁾

bit 5-2 **CHS<3:0>:** 模拟通道选择位 ⁽²⁾
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 = 通道 05 (AN5) ⁽¹⁾
0110 = 通道 06 (AN6) ⁽¹⁾
0111 = 通道 07 (AN7) ⁽¹⁾
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)
1101 = (保留)
1110 = VDDCORE
1111 = VBG 绝对参考电压 (~1.2V) ⁽³⁾

bit 1 **GO/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

注 1: 这些通道在 28 引脚器件上未实现。

2: 对未实现通道执行转换将返回随机值。

3: 为了达到最佳精度，在该通道上执行转换之前，应先将带隙参考电路使能 (ANCON1<7> = 1) 至少 10 ms。

4: 在具有 AVDD 和 AVSS 引脚的封装类型中，这两个引脚应外部连接到电路板上的 VDD 和 VSS 电平。对于不含这两个引脚的封装类型，AVDD 和 AVSS 被内部连接到 VDD 和 VSS。

PIC18F46J50 系列

寄存器 21-1: ADCON0: A/D 控制寄存器 0 (位于快速操作存储区, 地址 FC2h) (续)

bit 0 **ADON:** A/D 使能位
1 = 使能 A/D 转换器模块
0 = 禁止 A/D 转换器模块

- 注 1: 这些通道在 28 引脚器件上未实现。
2: 对未实现通道执行转换将返回随机值。
3: 为了达到最佳精度, 在该通道上执行转换之前, 应先将带隙参考电路使能 (ANCON1<7> = 1) 至少 10 ms。
4: 在具有 AVDD 和 AVSS 引脚的封装类型中, 这两个引脚应外部连接到电路板上的 VDD 和 VSS 电平。对于不含这两个引脚的封装类型, AVDD 和 AVSS 被内部连接到 VDD 和 VSS。

寄存器 21-2: ADCON1: A/D 控制寄存器 1 (位于快速操作存储区, 地址 FC1h)

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 = POR 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 转换时执行校准
0 = 正常的 A/D 转换器操作

bit 5-3 **ACQT<2:0>:** A/D 采集时间选择位
111 = 20 T_{AD}
110 = 16 T_{AD}
101 = 12 T_{AD}
100 = 8 T_{AD}
011 = 6 T_{AD}
010 = 4 T_{AD}
001 = 2 T_{AD}
000 = 0 T_{AD}

bit 2-0 **ADCS<2:0>:** A/D 转换时钟选择位
110 = Fosc/64
101 = Fosc/16
100 = Fosc/4
011 = FRC (时钟来自 A/D RC 振荡器) (1)
010 = Fosc/32
001 = Fosc/8
000 = Fosc/2

- 注 1: 如果选择了 A/D FRC 时钟源, 在 A/D 时钟启动之前会加上一个 T_{CY} (指令周期) 的延时。这允许在启动转换之前执行 SLEEP 指令。

ANCON0 和 ANCON1 寄存器用于配置与每个模拟通道关联的 I/O 引脚的操作。通过将任意一个 PCFG 位置 1，可将相应的引脚配置为仅用作数字功能的 I/O。通过清零一个位，可将引脚配置为 A/D 转换器或比较器模块的模拟输入；所有数字外设将被禁止，数字输入读为 0。一般来说，在器件复位时，与模拟输入复用的 I/O 引脚默认设为执行模拟操作。

对于 VBG 带隙参考电压 (ADCON0<5:2> = 1111)，为了正确执行 A/D 转换，参考电路必须先上电。应使能 ANCON1 寄存器中的 VBGEN 位，以允许固件手动请求

使能带隙参考电路。为了达到最佳精度，使能带隙参考电压之后，在该通道上执行第一次采集之前，固件应当留出至少 10 ms 的稳定时间。

如果某个其他硬件模块（例如片上稳压器、比较器或 HLVD）已经请求了参考电路，参考电路可能已经开启。这种情况下，可能已经经过了初始的开启稳定时间，固件不需要在测量 VBG 之前等待很久。采集完成之后，如果没有其他模块仍在请求 VBG 参考电压，固件可以清零 VBGEN 位，这可以节约少量的功耗。

寄存器 21-3: ANCON0: A/D 端口配置寄存器 0 (位于普通存储区, 地址 F48h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7 ⁽¹⁾	PCFG6 ⁽¹⁾	PCFG5 ⁽¹⁾	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 = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7-0 **PCFG<7:0>**: 模拟端口配置位 (AN7-AN0)
 1 = 引脚配置为数字端口
 0 = 引脚配置为模拟通道 —— 禁止数字输入且读为 0

注 1: 这些位仅在 44 引脚器件上可用。

寄存器 21-4: ANCON1: A/D 端口配置寄存器 1 (位于普通存储区, 地址 F49h)

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VBGEN	r	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 7							bit 0

图注:

r = 保留位
 R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7 **VBGEN**: 1.2V 带隙参考电压使能位
 1 = 1.2V 带隙参考电压上电
 0 = 1.2V 带隙参考电压关闭, 以节约功耗 (如果没有其他模块在请求它)

bit 6 **保留**: 始终保持为 0, 以使功耗最低

bit 5 **未实现**: 读为 0

bit 4-0 **PCFG<12:8>**: 模拟端口配置位 (AN12-AN8)
 1 = 引脚配置为数字端口
 0 = 引脚配置为模拟通道 —— 禁止数字输入且读为 0

PIC18F46J50 系列

可通过软件选择将器件的正负电源电压（AVDD 和 AVSS）或 RA3/AN3/VREF+/C1INB 和 RA2/AN2/VREF-/CVREF/C2INB 引脚上的电压作为模拟参考电压。

A/D 转换器具有能在器件处于休眠模式时工作的特性。要在休眠模式下工作，A/D 转换时钟必须来自 A/D 的内部 RC 振荡器。

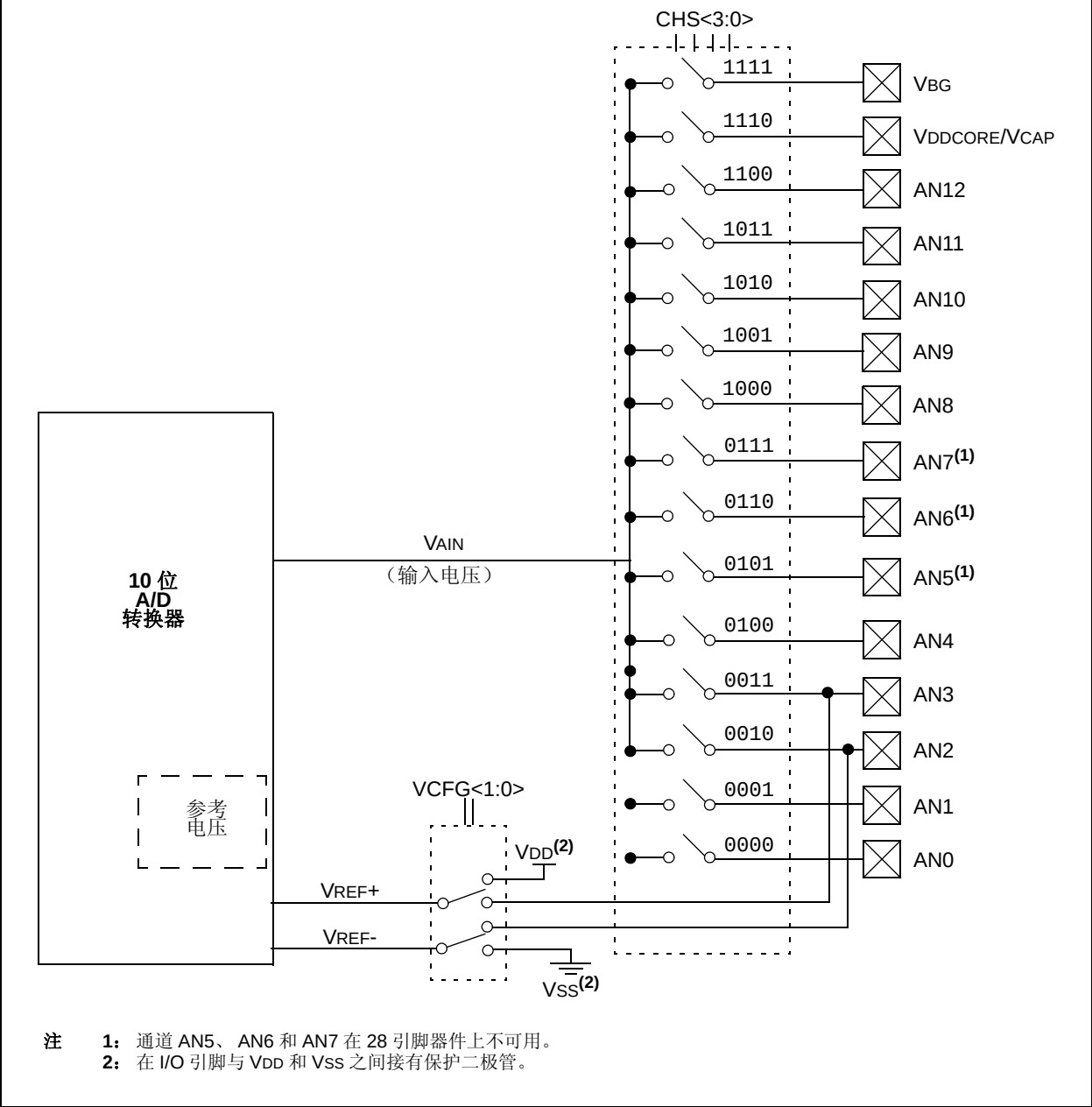
采样 / 保持电路的输出是转换器的输入，A/D 转换器采用逐次逼近法（Successive Approximation，SAR）得到转换结果。

与 A/D 转换器相关的每个端口引脚都可以被配置为模拟输入或数字 I/O。ADRESH 和 ADRESL 寄存器保存 A/D 转换的结果。当 A/D 转换完成时，结果被装入 ADRESH:ADRESL 寄存器对，GO/DONE 位（ADCON0<1>）被清零且 A/D 中断标志位 ADIF 被置 1。

器件复位将强制所有寄存器为复位状态。同时强制关闭 A/D 模块并中止任何正在进行的转换。上电复位（POR）时，不修改 ADRESH:ADRESL 寄存器对的值。POR 后，这些寄存器的值不确定。

图 21-1 给出了 A/D 模块的框图。

图 21-1: A/D 框图

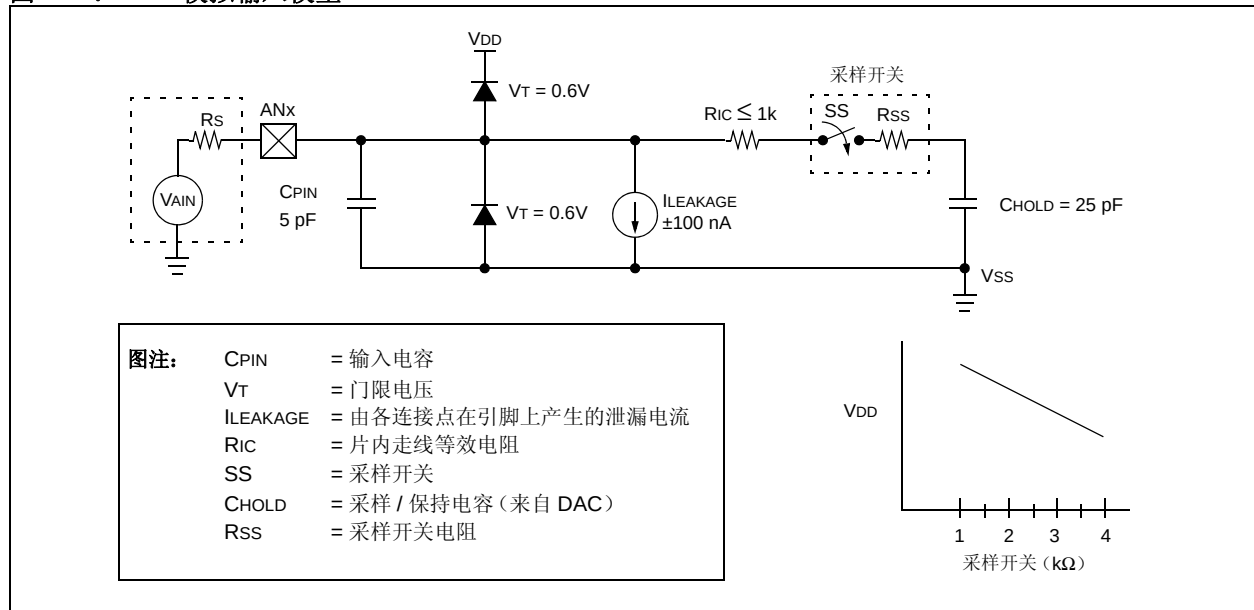


在根据需要配置好 A/D 模块之后，必须在转换开始之前对选定的通道进行采集。必须将模拟输入通道相应的 TRIS 位选择为输入。采集时间的确定，请参见第 21.1 节“A/D 采集要求”。采集时间一结束，即可启动 A/D 转换。可将采集时间编程设定在 GO/DONE 位置 1 和实际转换启动之间。

在执行 A/D 转换时应该遵循以下步骤：

1. 配置 A/D 模块：
 - 使用 ANCON0 和 ANCON1 将所需的 ADC 引脚配置为模拟引脚
 - 使用 ADCON0 设置参考电压
 - 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)
 - 开启 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，可通过以下两种方法之一判断转换是否完成：
 - 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)；需要时清零 ADIF 位。
7. 如需下一次进行 A/D 转换，返回步骤 1 或步骤 2。每位的 A/D 转换时间定义为 T_{AD} 。在下一次采集开始前需要等待至少 $2 T_{AD}$ 的时间。

图 21-2: 模拟输入模型



PIC18F46J50 系列

21.1 A/D 采集要求

为了使 A/D 转换器达到规定的精度，必须使充电保持电容（CHOLD）充满至输入通道的电压值。模拟输入模型见图 21-2。信号源阻抗（RS）和内部采样开关阻抗（RSS）直接影响为电容 CHOLD 充电所需的时间。采样开关阻抗值（RSS）随器件电压（VDD）不同而改变。信号源阻抗会影响模拟输入的失调电压（由于引脚泄漏电流的原因）。**模拟信号源的最大阻抗推荐值为 2.5 kΩ**。选择（改变）模拟输入通道后，必须对通道进行采样才能启动转换，采样时间必须大于最小采集时间。

注： 当启动转换时，要将保持电容与输入引脚断开。

可以使用公式 21-1 来计算最小采集时间。该公式假设误差为 1/2 LSB（A/D 转换需要 1024 步）。1/2 LSB 的误差是 A/D 达到规定分辨率所能允许的最大误差。

公式 21-3 给出了所需的最小采集时间 TACQ 的计算过程。计算结果基于以下对应用系统的假设：

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

TACQ	=	放大器稳定时间 + 保持电容充电时间 + 温度系数
	=	TAMP + TC + TCOFF

公式 21-2: A/D 最小充电时间

VHOLD	=	$(VREF - (VREF/2048)) \cdot (1 - e^{-(Tc/CHOLD)(Ric + RSS + Rs)})$
或		
Tc	=	$-(CHOLD)(Ric + RSS + Rs) \ln(1/2048)$

公式 21-3: 计算所需的最小采集时间

TACQ	=	TAMP + TC + TCOFF
TAMP	=	0.2 μs
TCOFF	=	(温度 - 25°C)(0.02 μs/°C) (85°C - 25°C)(0.02 μs/°C) 1.2 μs
只有在温度 > 25°C 时才需要温度系数。当温度低于 25°C 时，TCOFF = 0 μs。		
TC	=	$-(CHOLD)(Ric + RSS + Rs) \ln(1/2048) \mu s$ $-(25 pF)(1 k\Omega + 2 k\Omega + 2.5 k\Omega) \ln(0.0004883) \mu s$ 1.05 μs
TACQ	=	0.2 μs + 1.05 μs + 1.2 μs 2.45 μs

21.2 选择和配置自动采集时间

ADCON1 寄存器允许用户选择采集时间，该时间在每当 GO/DONE 位置 1 时发生。

当 GO/DONE 位置 1 时，采样停止并启动转换。用户应确保在选定所需的输入通道之后到 GO/DONE 位置 1 之间经过了所需的采集时间。这发生在 ACQT<2:0> 位 (ADCON1<5:3>) 保持在其复位状态 (000) 的情况下，与不提供可编程采集时间的器件相兼容。

如果需要，可设置 ACQT 位以便为 A/D 模块选择可编程采集时间。当 GO/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，然后自动启动转换。由于采集时间已被编程，因此在选择通道和 GO/DONE 位置 1 之间无需另外等待一个采集时间。

在这两种情况下，当转换完成时，GO/DONE 位被清零，ADIF 标志位被置 1 且 A/D 再次开始对当前选定的通道进行采样。如果采集时间已被编程，那么将不会有任何采集时间结束、转换开始的指示。

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。可通过软件选择 A/D 转换的时钟源。

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

为了实现正确的 A/D 转换，A/D 转换时钟 (TAD) 必须尽可能得小，但它必须大于最小 TAD（更多信息，请参见表 30-32 中的参数 130）。

表 21-1 给出了器件在不同工作频率下和选择不同的 A/D 时钟源时得到的 TAD。

表 21-1: 不同器件工作频率下的 TAD

A/D 时钟源 (TAD)		最高器件频率
工作时钟源	ADCS<2:0>	
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	45.71 MHz
64 TOSC	110	48.0 MHz
RC ⁽²⁾	011	1.00 MHz ⁽¹⁾

- 注 1: RC 时钟源的典型 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允许的范围。

21.4 配置模拟端口引脚

ANCON0、ANCON1 和 TRISA 寄存器用于控制 A/D 端口引脚的工作。必须将需要用作模拟输入的端口引脚的对应 TRIS 位置 1（输入）。如果将 TRIS 位清零（输出），则数字输出电平 (VOH 或 VOL) 将被转换。

A/D 转换操作与 CHS<3:0> 位和 TRIS 位的状态无关。

注 1: 当读取端口寄存器时，所有配置为模拟输入通道的引脚均读为零（低电平）。配置为数字输入的引脚将按模拟输入进行转换。配置为数字输入的引脚上的模拟电平将被精确转换。

2: 定义为数字输入引脚上的模拟电平可能会导致数字输入缓冲器消耗的电流超出器件规范。

21.5 A/D 转换

图 21-3 显示了在 $\overline{\text{GO/DONE}}$ 位置 1 且 $\text{ACQT}\langle 2:0 \rangle$ 位被清零后 A/D 转换器的工作。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 $\overline{\text{GO/DONE}}$ 位置 1， $\text{ACQT}\langle 2:0 \rangle$ 位被设置为 010，且在转换开始之前选择 4 TAD 采集时间后 A/D 转换器的工作。

在转换期间清零 $\overline{\text{GO/DONE}}$ 位将中止当前的 A/D 转换。不会用部分完成的 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完成的转换的结果（或上一次写入 ADRESH:ADRESL 寄存器的值）。

在 A/D 转换完成或中止后，需要等待 2 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：不应在开启 A/D 模块的指令中将 $\overline{\text{GO/DONE}}$ 位置 1。

21.6 ECCP2 触发信号的使用

ECCP2 模块的特殊事件触发信号可以启动 A/D 转换。这需要将 $\text{CCP2M}\langle 3:0 \rangle$ 位（ $\text{CCP2CON}\langle 3:0 \rangle$ ）编程为 1011，并且使能 A/D 模块（ $\overline{\text{ADON}}$ 位置 1）。发生触发事件时， $\overline{\text{GO/DONE}}$ 位将被置 1，启动 A/D 采集和转换并将 Timer1（或 Timer3）计数器复位为零。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 内容传送到所需存储单元）。特殊事件触发信号将 $\overline{\text{GO/DONE}}$ 位置 1（启动转换）之前，用户必须选择正确的模拟输入通道和最小采集周期，最小采集周期由用户设定或通过选择适当的 TACQ 时间来设定。

如果未使能 A/D 模块（ $\overline{\text{ADON}}$ 清零），则特殊事件触发信号将被 A/D 模块忽略，但它仍会将 Timer1（或 Timer3）计数器复位。

图 21-3: A/D 转换 TAD 周期 ($\text{ACQT}\langle 2:0 \rangle = 000$, $\text{TACQ} = 0$)

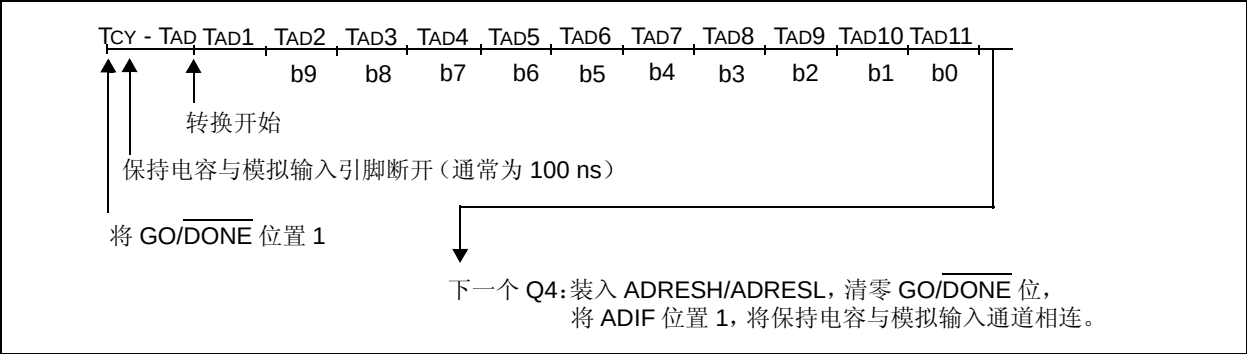
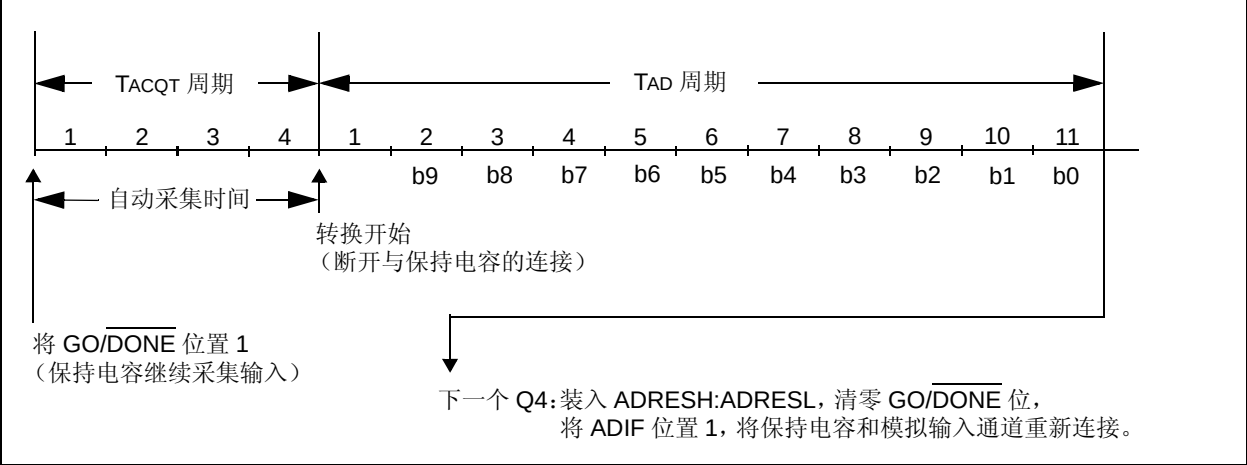


图 21-4: A/D 转换 TAD 周期 ($\text{ACQT}\langle 2:0 \rangle = 010$, $\text{TACQ} = 4 \text{ TAD}$)



21.7 A/D 转换器校准

PIC18F46J50 系列器件中的 A/D 转换器包括自校准功能，能补偿模块中产生的任何偏移。校准过程是通过将 ADCAL 位（ADCON1<6>）置 1 而自动启动的。GO/DONE 位下次置 1 时，模块将执行“假”转换（即不读输入通道），将结果值内部存储起来以补偿偏移。这样可实现后续偏移的补偿。

例 21-1 提供了校准程序的一个示例。

校准过程假定器件处于相对稳态工作条件下。如果使用了 A/D 校准功能，应在每次器件复位后或工作条件有重大变化时执行校准操作。

21.8 在功耗管理模式下的操作

在功耗管理模式下，自动采集时间和 A/D 转换时钟的选择一定程度上可由时钟源和频率决定。

如果希望器件处于功耗管理模式时 A/D 继续工作，就应该根据要在功耗管理模式下使用的时钟对 ADCON1 中的 ACQT<2:0> 和 ADCS<2:0> 位进行更新。在进入功耗管理模式后（两种功耗管理运行模式之一），就可以开始 A/D 采集或转换。采集或转换开始以后，器件应继续使用与功耗管理模式相同的时钟源直到转换完成。如果需要，在转换期间也可以将器件置于相应的功耗管理空闲模式。

如果功耗管理模式的时钟频率小于 1 MHz，就应该选择 A/D RC 时钟源。

在休眠模式下工作需要选择 A/D RC 时钟。如果将 ACQT<2:0> 位设置为 000 并启动转换，转换将延时一个指令周期以允许执行 SLEEP 指令并进入休眠模式。OSCCON 寄存器中的 IDLEN 和 SCS 位必须在启动转换之前被清零。

例 21-1: A/D 校准程序示例

BCF	ANCON0,PCFG0	;Make Channel 0 analog
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

PIC18F46J50 系列

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 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR1	PMPIF ⁽¹⁾	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	72
PIE1	PMPIE ⁽¹⁾	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	72
IPR1	PMPIP ⁽¹⁾	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	72
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	71
ADRESH	A/D 结果寄存器的高字节								70
ADRESL	A/D 结果寄存器的低字节								70
ADCON0	VCFG1	VCFG0	CHS3	CHS3	CHS1	CHS0	GO/DONE	ADON	70
ANCON0	PCFG7 ⁽¹⁾	PCFG6 ⁽¹⁾	PCFG5 ⁽¹⁾	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	74
ADCON1	ADFM	ADCAL	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	70
ANCON1	VBGEN	r	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	74
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	71
PORTA	RA7	RA6	RA5	—	RA3	RA2	RA1	RA0	72
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	72

图注: — = 未实现 (读为 0), r = 保留。A/D 转换不使用阴影单元。

注 1: 这些位仅在 44 引脚器件上可用。

22.0 通用串行总线（USB）

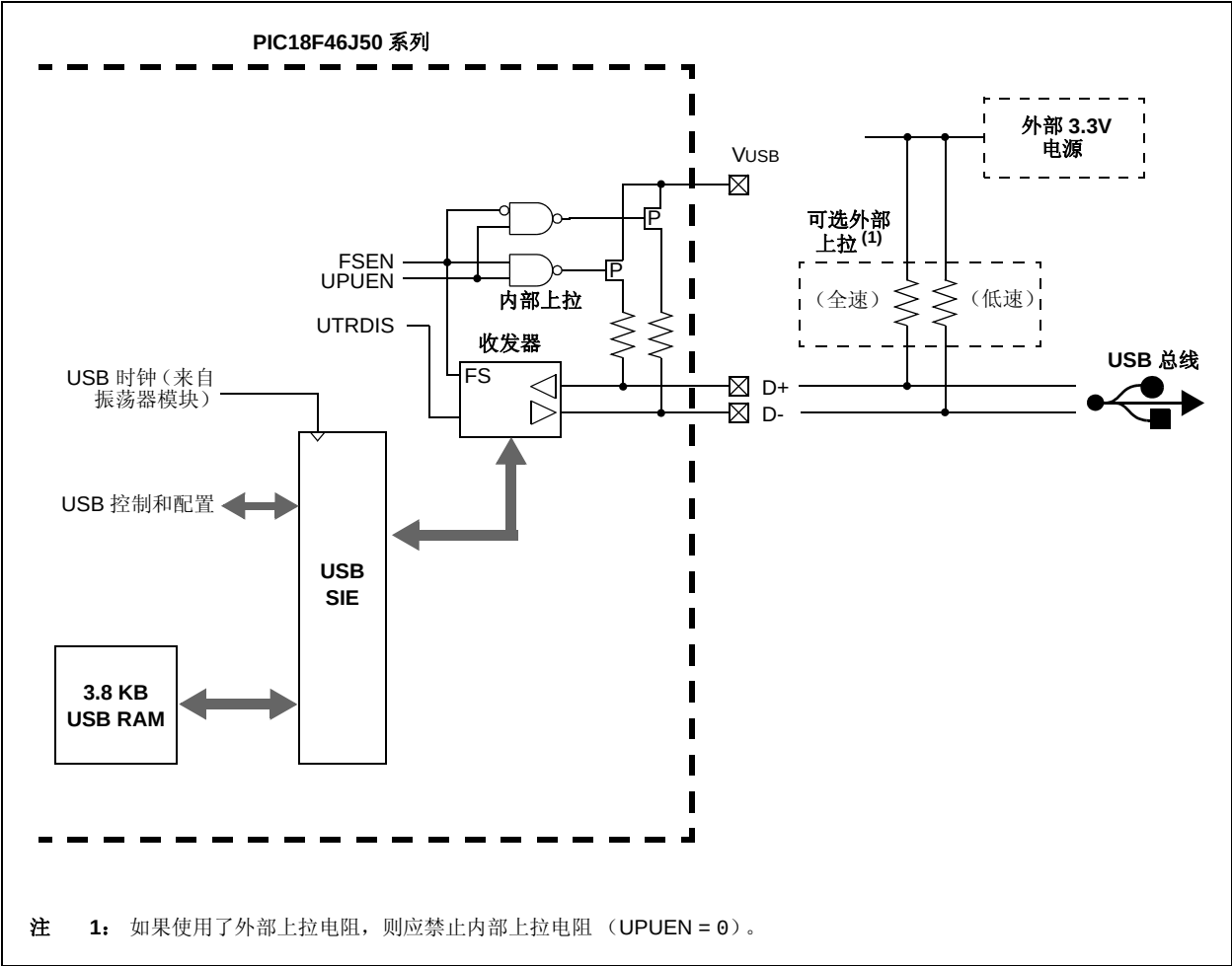
本节将详细介绍 USB 外设。由于本模块的特殊性质，要求读者具备 USB 方面的基础知识。在第 22.9 节“USB 概述”中提供了一些 USB 的概括信息，仅供应用设计参考。我们鼓励设计人员参考 USB Implementers Forum（USB-IF）出版的官方规范，以获得最新的信息。本文档发布时，USB 规范 2.0 版是最新的规范。

22.1 USB 外设概述

PIC18F46J50 系列器件包含全速和低速、兼容 USB 的串行接口引擎（SIE），它允许在 USB 主机和 PIC® MCU 之间实现快速通信。SIE 可使用内部收发器直接与 USB 接口。

该模块包括了一些特殊硬件功能来改善性能。在器件的数据存储空间（USB RAM）中提供了双端口存取存储器，用于单片机内核与 SIE 之间的直接存储器访问。还提供了缓冲区描述符，允许用户自由地设置 USB RAM 空间内对端点存储区的使用。图 22-1 提供了 USB 外设及其特性的一般概括。

图 22-1: USB 外设和选项



22.2 USB 状态和控制

USB 模块的操作是通过 3 个控制寄存器配置和管理的。此外，总共使用了 22 个寄存器管理实际 USB 事务。这些寄存器是：

- USB 控制寄存器（UCON）
- USB 配置寄存器（UCFG）
- USB 传输状态寄存器（USTAT）
- USB 器件地址寄存器（UADDR）
- 帧编号寄存器（UFRMH:UFRML）
- 端点使能寄存器 0 到 15（UEPn）

22.2.1 USB 控制寄存器（UCON）

USB 控制寄存器（寄存器 22-1）包含控制传输期间的模块行为所需的控制位。寄存器包含控制以下内容的位：

- 主 USB 外设使能
- 乒乓缓冲区指针复位
- 挂起模式的控制
- 数据包传输禁止

此外，USB 控制寄存器还包含状态位 SE0（UCON<5>），用于指示总线上是否出现了单端零状态。USB 模块使能

时，应监视该位，以确定差分数据线上是否出自单端零条件。这有助于区分初始上电状态和 USB 复位信号。

USB 模块的总体操作是由 USBEN 位（UCON<3>）控制的。将该位置 1 将激活模块，并将缓冲区描述符表（BDT）中的所有 PPBI 位复位为 0。该位还将激活内部上拉电阻（如果已使能）。因此，该位可用作 USB 的软件连接 / 断开控制。尽管清零该位时，所有状态和控制位都将被忽略，但是还是需要在置 1 该位前预先完全配置好 USB 模块。USB 时钟源应当已经配置为正确的频率，并已经在运行。如果使用了 PLL，则应先将它使能至少 2 ms（以便有足够的时间让 PLL 锁定），然后再尝试将 USBEN 位置 1。

注：	在禁止 USB 模块时，请确保在清零 USBEN 位之前，SUSPND 位（UCON<1>）已清零。在模块处于挂起状态时清零 USBEN 位可能会使模块无法完全掉电。

寄存器 22-1: UCON: USB 控制寄存器（位于快速操作存储区，地址 F65h）

U-0	R/W-0	R-x	R/C-0	R/W-0	R/W-0	R/W-0	U-0
—	PPBRST	SE0	PKTDIS	USBEN ⁽¹⁾	RESUME	SUSPND	—
bit 7							bit 0

图注:	C = 可清零位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	U = 未实现位，读为 0
	0 = 清零
	x = 未知

bit 7	未实现: 读为 0
bit 6	PPBRST: 乒乓缓冲区复位位 1 = 将所有乒乓缓冲区指针复位到偶编号缓冲区描述符（Buffer Descriptor, BD）存储区 0 = 乒乓缓冲区指针不复位
bit 5	SE0: 单端零出现标志位 1 = 在 USB 总线上出现单端零信号 0 = 没有检测到单端零信号
bit 4	PKTDIS: 数据包传输禁止位 1 = 禁止 SIE 令牌和数据包处理，收到 SETUP 令牌时自动置 1 0 = 使能 SIE 令牌和数据包处理
bit 3	USBEN: USB 模块使能位 ⁽¹⁾ 1 = 使能 USB 模块和支持电路（与设备连接） 0 = 禁止 USB 模块和支持电路（与设备断开连接）
bit 2	RESUME: 恢复信号传输使能位 1 = 激活恢复信号传输 0 = 禁止恢复信号传输
bit 1	SUSPND: 挂起 USB 位 1 = USB 模块和支持电路处于节能模式，SIE 时钟无效 0 = USB 模块和支持电路处于正常工作模式，SIE 时钟按配置的速率运行
bit 0	未实现: 读为 0

注 1: 在置 1 该位之前，请确保先正确配置 USB 时钟源。

PIC18F46J50 系列

PPBRST 位 (UCON<6>) 控制使用双重缓冲模式 (乒乓缓冲) 时的复位状态。PPBRST 位置 1 时, 所有乒乓缓冲区指针都被设置为指向偶编号缓冲区。PPBRST 必须由固件清零。该位在不使用乒乓缓冲模式时被忽略。

PKTDIS 位 (UCON<4>) 是指示 SIE 禁止数据包发送和接收的标志。该位在收到允许设置处理的 SETUP 令牌时由 SIE 置 1。单片机无法将该位置 1, 只能清零; 将其清零可使 SIE 继续发送和 / 或接收。缓冲区描述符表中的任何待处理事件仍有效, 由 USTAT 寄存器的 FIFO 缓冲区指示。

RESUME 位 (UCON<2>) 允许外设通过执行恢复信号传输执行远程唤醒。为了产生有效的远程唤醒, 固件必须将 RESUME 置 1 并保持 10 ms, 然后将该位清零。关于恢复信号传输的更多信息, 请参见 USB 2.0 规范中的第 7.1.7.5 节、第 11.4.4 节和第 11.9 节。

SUSPND 位 (UCON<1>) 将模块和支持电路置于低功耗模式。SIE 的输入时钟也将被禁止。在响应 IDLEIF 中断时, 该位应由软件置 1。它在发生 ACTVIF 中断后应由单片机固件复位。该位有效时, 设备保持与总线的连接, 但是收发器输出保持空闲。VUSB 引脚上的电压可能随该位值的不同而变化。在 IDLEIF 请求之前将该位置 1 将导致不可预测的总线行为。

注: 在挂起模式下, 典型的总线供电 USB 设备的平均电流上限是 2.5 mA。这是 PIC 器件及其支持电路消耗的全部电流。器件进入挂起模式后, 要小心确保消耗电流最低。

22.2.2 USB 配置寄存器 (UCFG)

通过 USB 进行通信前, 必须配置该模块的相关内部和 / 或外部硬件。多数配置都是通过 UCFG 寄存器 (寄存器 22-2) 执行的。UCFG 寄存器包含大多数控制 USB 模块系统级操作的控制位。这些操作包括:

- 总线速度 (全速与低速)
- 片上上拉电阻使能
- 片上收发器使能
- 乒乓缓冲区使用

UCFG 寄存器还包含辅助模块测试、调试和 USB 认证的两个位。这些位控制输出使能状态监视和眼图生成。

注: 只应在模块设置阶段配置 USB 速度、收发器和上拉电路。建议不要在使能模块时改变这些设置。

22.2.2.1 内部收发器

USB 外设具有一个内置的、USB 2.0 全速和低速兼容收发器, 从内部连接到 SIE。该功能部件对于低成本单芯片应用很有用。UTRDIS 位 (UCFG<3>) 控制收发器, 它默认为使能 (UTRDIS = 0)。FSEN 位 (UCFG<2>) 控制收发器速度; 将该位置 1 将使能全速工作。

片上 USB 上拉电阻通过 UPUEN 位 (UCFG<4>) 进行控制。只有使能片上收发器时, 才能选择它们。

内部 USB 收发器由 VUSB 引脚外接电源供电。为了满足 USB 信号电平规范, 必须在 VUSB 上提供介于 3.0V 和 3.6V 之间的电压源。使用 3.3V 电源, 并使用高品质陶瓷电容 (例如 0.1 μ F) 进行本地旁路时, 可以获得最佳的电气信号质量。电容应尽可能靠近 VUSB 和 VSS 引脚。

应始终保持 $V_{USB} \geq V_{DD}$ 。若不使用 USB 模块, 而将 RC4 或 RC5 用作通用输入引脚, 应仍然将 VUSB 连接至电源 (如 VDD)。RC4 和 RC5 引脚的输入门限值取决于 VUSB 的供电电压。

D+ 和 D- 信号线可以直接连接到 USB 连接器或电缆的相应引脚 (对于硬连线应用)。不需要额外使用其他电阻、电容或磁性元件, 因为 D+ 和 D- 驱动器具有受控制的压摆率, 以及与 USB 电缆的特性阻抗相匹配的输出阻抗。

为了使 USB 信号的质量最佳, 单片机和 USB 连接器 (或电缆) 之间的 D+ 和 D- 走线的长度应小于 19 cm。两条走线的长度应该相等, 并且布线时应使它们相互平行。理想情况下, 这些走线应设计为特性阻抗与 USB 电缆阻抗相匹配。

寄存器 22-2: UCFG: USB 配置寄存器（位于普通存储区，地址 F39h）

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UTEYE	UOEMON	—	UPUEN ^(1,2)	UTRDIS ^(1,3)	FSEN ⁽¹⁾	PPB1	PPB0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **UTEYE: USB 眼图测试使能位**

1 = 使能眼图测试

0 = 禁止眼图测试

bit 6 **UOEMON: USB OE 监视器使能位**

1 = UOE 信号有效；指示驱动 D+/D- 线的时间间隔

0 = UOE 信号无效

bit 5 **未实现:** 读为 0

bit 4 **UPUEN: USB 片上上拉使能位 ^(1,2)**

1 = 使能片上上拉（FSEN = 1 时，D+ 上拉；FSEN = 0 时，D- 上拉）

0 = 禁止片上上拉

bit 3 **UTRDIS: 片上收发器禁止位 ^(1,3)**

1 = 禁止片上收发器

0 = 使能片上收发器

bit 2 **FSEN: 全速使能位 ⁽¹⁾**

1 = 全速设备：控制收发器边沿速率；要求输入时钟运行在 48 MHz

0 = 低速设备：控制收发器边沿速率；要求输入时钟运行在 6 MHz

bit 1-0 **PPB<1:0>: 乒乓缓冲区配置位**

11 = 使能端点 1 到 15 的奇 / 偶乒乓缓冲区

10 = 使能所有端点的奇 / 偶乒乓缓冲区

01 = 使能 OUT 端点 0 的奇 / 偶乒乓缓冲区

00 = 禁止奇 / 偶乒乓缓冲区

注 1: UPUEN、UTRDIS 和 FSEN 位在 USB 模块使能时不要更改。必须在使能模块前预先配置这些值。

2: 该位仅在片上收发器有效（UTRDIS = 0）时有效；否则将被忽略。

3: 如果 UTRDIS 被置 1，UOE 信号将有效，与 UOEMON 位的设置无关。

PIC18F46J50 系列

22.2.2.2 内部上拉电阻

PIC18F46J50 系列器件具有内置的上拉电阻，设计为用来满足低速和全速 USB 的要求。UPUEN 位 (UCFG<4>) 用于使能内部上拉。图 22-1 给出了上拉及其控制的图示。

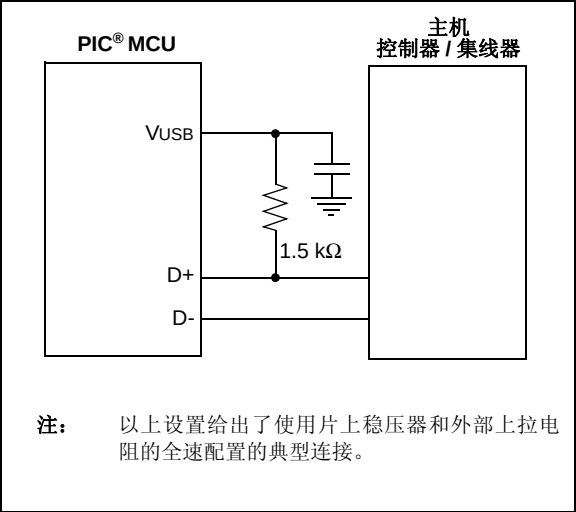
注：符合规范的 USB 设备永远不能在 USB 电缆的 +5V VBUS 线上引入任何电流。此外，每当 +5V VBUS 线的电压低于 1.17V 时，USB 设备不应在 D+ 和 D- 数据线上引入任何电流。为了满足 USB 规范，不是纯粹使用总线供电的应用应当监视 VBUS 线，并避免在 VBUS 大于 1.17V 之前开启 USB 模块和 D+ 或 D- 上拉电阻。可以将 VBUS 连接到任何可承受 5V 的 I/O 引脚并通过该引脚进行监视。若使用电阻分压器，则将 VBUS 连接至一个模拟引脚，并通过该引脚进行监视。

22.2.2.3 外部上拉电阻

此外，也可以使用外部上拉电阻。可用 VUSB 引脚上拉 D+ 或 D-。按照 USB 规范的要求，上拉电阻必须是 1.5 kΩ (±5%)。

图 22-2 给出了一个外部电路的示例。

图 22-2: 外部电路



22.2.2.4 乒乓缓冲区配置

乒乓缓冲区的使用是通过 PPB<1:0> 位配置的。关于乒乓缓冲区的完整说明，请参见第 22.4.4 节“乒乓缓冲”。

22.2.2.5 使能眼图测试

UCFG<7> 位置 1 时，可通过该模块生成自动的眼图测试。眼图输出的可观察性取决于模块设置，这意味着用户首先要负责配置 SIE 时钟设置、上拉电阻和收发器模式。此外，必须使能该模块。

一旦 UTEYE 被置 1，该模块将仿真从接收切换到发送状态，并开始发送 J-K-J-K 位序列（对于全速是 K-J-K-J）。当使能了眼图测试模式时，该序列将无限重复。

注意，该模块连接到实际 USB 系统时，不应将该位置 1。该测试模式旨在用于电路板检验，并帮助进行 USB 认证测试。它旨在向系统开发人员展示 USB 信号中的完整噪声，USB 信号噪声来自于电路板布线、阻抗不匹配和邻近其他系统元器件的影响。它不能正确测试从接收到发送状态的转换过程。尽管眼图不能代替更复杂的 USB 认证测试，但它在首轮系统调试中会有帮助。

22.2.3 USB 状态寄存器 (USTAT)

USB 状态寄存器报告 SIE 内的事务状态。SIE 发出 USB 传输完成中断时，应读取 USTAT 以确定传输的状态。USTAT 包含传输端点号、方向和乒乓缓冲区指针值（如果使用了）。

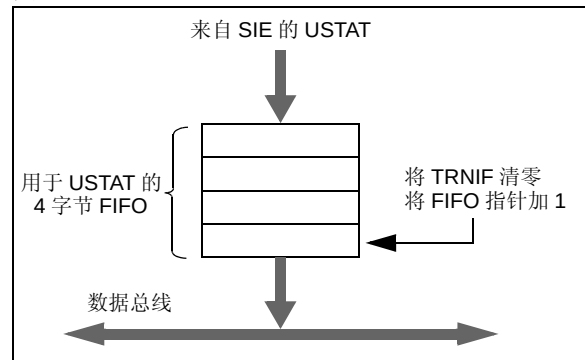
注： USB 状态寄存器中的数据仅在 TRNIF 中断标志被置 1 时有效。

USTAT 寄存器实际上是由 SIE 维护的四字节状态 FIFO 的读取窗口。它允许单片机在 SIE 处理其他端点时处理一次传输（图 22-3）。当 SIE 使用缓冲区完成读写数据时，它将更新 USTAT 寄存器。如果在响应事务完成中断前进行了另一次 USB 传输，SIE 将把下次传输的状态存储到状态 FIFO 中。

将传输完成中断标志位 TRNIF 清零将使 SIE 把 FIFO 指针加 1。如果 FIFO 保持寄存器中的下一个数据有效，则 SIE 将在清零 TRNIF 后的 5 个 Tcy 内将中断标志再次置 1。如果没有其他数据，TRNIF 将保持清零，USTAT 数据将不再可靠。

注： 如果 USTAT FIFO 已满时收到端点请求，SIE 将自动将 NAK 发送回主机。

图 22-3: USTAT FIFO



寄存器 22-3: USTAT: USB 状态寄存器（位于快速操作存储区，地址 F64h）

U-0	R-x	R-x	R-x	R-x	R-x	R-x	U-0
—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI ⁽¹⁾	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 未实现：读为 0

bit 6-3 **ENDP<3:0>**: 上次活动的端点的编码
(代表上次 USB 传输更新的 BDT 编号)

1111 = 端点 15

1110 = 端点 14

.

.

.

0001 = 端点 1

0000 = 端点 0

bit 2 **DIR**: 上次 BD 方向指示位

1 = 上次事务是 IN 令牌

0 = 上次事务是 OUT 令牌或 SETUP 令牌

bit 1 **PPBI**: 乒乓 BD 指针指示位 ⁽¹⁾

1 = 上次事务针对奇编号 BD 存储区

0 = 上次事务针对偶编号 BD 存储区

bit 0 未实现：读为 0

注 1: 该位仅对具有可用的偶编号和奇编号 BD 寄存器的端点有效。

PIC18F46J50 系列

22.2.4 USB 端点控制

16 个可能的双向端点中的每一个都有各自的独立控制寄存器 UEPn（其中“n”代表端点号）。每个寄存器内相应控制位的设置都是相同的。原型如寄存器 22-4 所示。

EPHSHK 位（UEPn<4>）控制端点的握手；将该位置 1 将使能 USB 握手。通常该位总是置 1 的，使用同步端点时除外。

EPCONDIS 位（UEPn<3>）用于使能或禁止通过端点进行 USB 控制操作（SETUP）。将该位清零将使能 SETUP 事务。注意，对应的 EPINEN 和 EPOUTEN 位

必须置 1，才能使能 IN 和 OUT 事务。对于端点 0，该位应总是清零的，因为 USB 规范将端点 0 视为默认的控制端点。

EPOUTEN 位（UEPn<2>）用于使能或禁止主机发起的 USB OUT 事务。将该位置 1 将使能 OUT 事务。类似地，EPINEN 位（UEPn<1>）使能或禁止主机发起的 USB IN 事务。

EPSTALL 位（UEPn<0>）用于指示端点的 STALL 状态。如果在特定端点上发出了 STALL，SIE 将把该端点对的 EPSTALL 位置 1。EPSTALL 位将保持置 1 状态直到固件将其清零，或 SIE 复位。

寄存器 22-4: UEPn: USB 端点 n 控制寄存器（UEP0 到 UEP15）（位于普通存储区，地址 F26h-F35h）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL
bit 7			bit 0				

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

bit 7-5	未实现：读为 0
bit 4	EPHSHK: 端点握手使能位 1 = 使能端点握手 0 = 禁止端点握手（通常用于同步端点）
bit 3	EPCONDIS: 双向端点控制位 如果 EPOUTEN = 1 并且 EPINEN = 1: 1 = 禁止端点 n 的控制传输；只允许 IN 和 OUT 传输 0 = 使能端点 n 的控制（SETUP）传输；同时允许 IN 和 OUT 传输
bit 2	EPOUTEN: 端点输出使能位 1 = 使能端点 n 输出 0 = 禁止端点 n 输出
bit 1	EPINEN: 端点输入使能位 1 = 使能端点 n 输入 0 = 禁止端点 n 输入
bit 0	EPSTALL: 端点停止指示位 1 = 端点 n 已发出一个或多个 STALL 数据包 0 = 端点 n 未发出任何 STALL 数据包

22.2.5 USB 地址寄存器（UADDR）

USB 地址寄存器包含唯一的 USB 地址，当此地址有效时，可被外设识别。在收到 URSTIF 指示的“USB 复位”或从单片机收到“复位”时，UADDR 将复位为 00h。USB 地址必须在 USB 设置阶段（枚举）由单片机写入，该操作受 Microchip USB 固件支持。

22.2.6 USB 帧编号寄存器（UFRMH:UFRML）

帧编号寄存器包含 11 位的帧编号。低位字节保存在 UFRML 中，而高 3 位保存在 UFRMH 中。每当收到 SOF 令牌时，就用当前帧编号更新寄存器对。对于单片机，这些寄存器是只读的。帧编号寄存器主要用于同步传输。仅当 48 MHz SIE 时钟有效时，UFRMH 和 UFRML 寄存器的内容才有效（即，当 SUSPND（UCON<1>）位 = 1 时，这两个寄存器的内容是不准确的）。

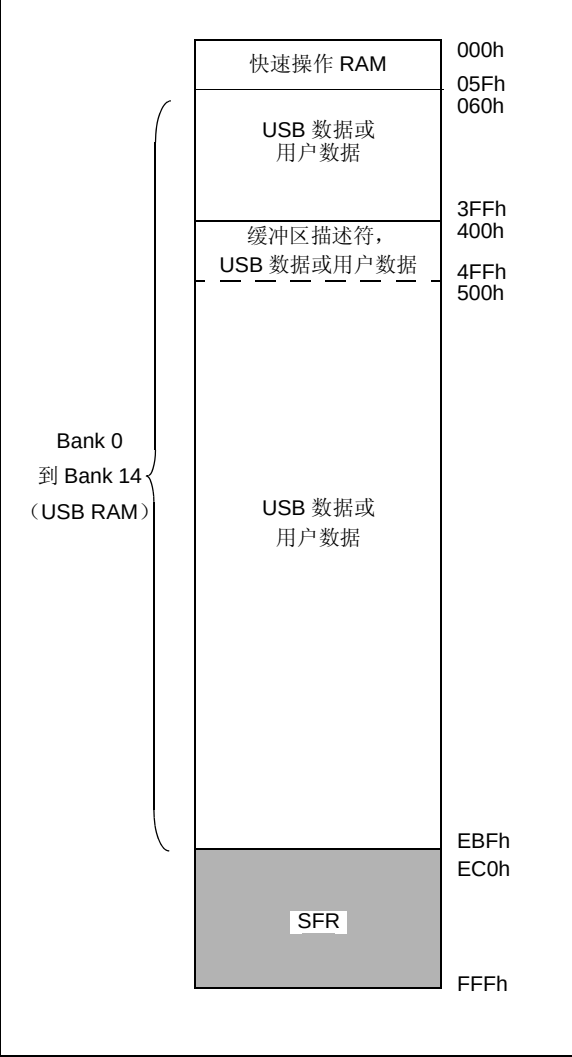
22.3 USB RAM

USB 数据通过称为 USB RAM 的存储空间在单片机内核与 SIE 之间传送。这是特殊的双端口存取存储器，被映射到 Bank 0 到 Bank 14（00h 到 EBFh）中的普通数据存储区，总共 3.8 KB（图 22-4）。

Bank 4（400h 到 4FFh）专门用于端点缓冲区控制，而 Bank 0 到 Bank 3 以及 Bank 5 到 Bank 14 用于存储 USB 数据。根据使用的缓冲类型，除 8 个字节外，Bank 4 的所有其他单元都可用作 USB 缓冲空间。

尽管单片机可使用 USB RAM 作为数据存储器，但是正由 SIE 访问的段不应被单片机访问。信号量机制被用来确定任意给定时间对特定缓冲区的访问。这将在第 22.4.1.1 节“缓冲区所有权”中讨论。

图 22-4: USB RAM 在数据存储区中的实现



22.4 缓冲区描述符和缓冲区描述符表

Bank 4 中的寄存器组成称为缓冲区描述符表（BDT）的结构，专门用于端点缓冲区控制。它为用户提供了灵活的方法，用于构造和控制不同长度及配置的端点缓冲区。

BDT 由缓冲区描述符（BD）组成，后者用于定义和控制 USB RAM 空间中的实际缓冲区。而每个 BD 由四个寄存器组成，其中 n 代表 64 个可能的 BD 中的一个（范围是 0 到 63）：

- BDnSTAT: BD 状态寄存器
- BDnCNT: BD 字节计数寄存器
- BDnADRL: BD 低位地址寄存器
- BDnADRH: BD 高位地址寄存器

BD 在序列中总是以一个四字节的數據块（BDnSTAT:BDnCNT:BDnADRL:BDnADRH）出现。BDnSTAT 的地址总是一个相对于 400h 的偏移量（ $4n - 1$ ，十六进制），其中 n 是缓冲区描述符编号。

根据使用的缓冲配置（第 22.4.4 节“乒乓缓冲”），最多可以有 32、33 或 64 组缓冲区描述符。BDT 至少要有 8 个字节长。这是因为 USB 规范规定，每个设备的初始设置都必须有输入和输出端点 0。根据端点和缓冲配置，BDT 可以长达 256 字节。

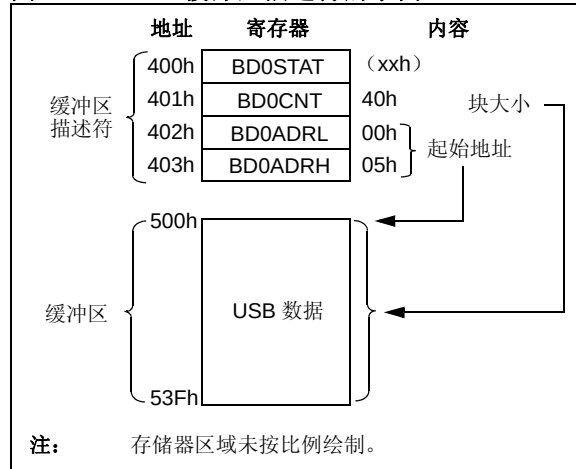
尽管缓冲区描述符状态和地址寄存器可被视为特殊功能寄存器，但它们和 Bank 15 中的常规单片机 SFR 不同，并不是硬件映射的。如果对应于特定 BD 的端点未被使用，其寄存器将不会被使用。它们将显示为可用的 RAM，而非未实现地址。仅当通过将 UEPn<1> 位置 1 使能了端点时，那些地址处的存储器才能用作 BD 寄存器。正如数据存储空间中的任何地址一样，BD 寄存器在任何器件复位时，包含的都是一个不确定的值。

图 22-5 所示为用于 64 字节缓冲区的 BD（从 500h 开始）的示例。特定的 BD 寄存器组仅当相应端点已使用 UEPn 寄存器使能时才有效。USB RAM 中的所有 BD 寄存器都可用。每个端点的 BD 都应在使能该端点前设置。

22.4.1 BD 状态和配置

缓冲区描述符不仅定义端点缓冲区的大小，还确定其配置和控制。多数配置是通过 BD 状态寄存器 BDnSTAT 完成的。每个 BD 都有其唯一的、对应编号的 BDnSTAT 寄存器。

图 22-5: 缓冲区描述符的示例



和其他控制寄存器不同，BDnSTAT 寄存器的位的配置取决于具体应用。有两种不同的配置，取决于在某个特定时间修改 BD 和缓冲区的是单片机还是 USB 模块。只有 3 个位定义是两种配置共用的。

22.4.1.1 缓冲区所有权

由于缓冲区及其 BD 是在 CPU 和 USB 模块之间共用的，因此采用了简单的信号量机制来区分允许 CPU 还是 USB 模块更新存储器中的 BD 和相关缓冲区。

这是通过使用 UOWN 位（BDnSTAT<7>）作为信号量来实现的。UOWN 是在 BDnSTAT 的两种配置之间共用的唯一一个位。

UOWN 清零时，BD 归单片机内核“所有”。UOWN 位置 1 时，BD 和缓冲存储区归 USB 外设“所有”。此时内核不应修改 BD 或其对应的数据缓冲区。注意，SIE 拥有缓冲区时，单片机内核仍可以读取 BDnSTAT，反之亦然。

根据寄存器更新源的不同，缓冲区描述符具有不同的意义。在分配给 USB 外设所有权之前，用户可以通过 BDnSTAT 位配置外设的基本操作。此时也可设置字节计数和缓冲区位置寄存器。

UOWN 置 1 时，用户不能再依赖写入 BD 的值。从此时起，SIE 根据需要更新 BD，覆盖原始 BD 值。BDnSTAT 寄存器由 SIE 用令牌 PID 更新，传输计数 BDnCNT 也被更新。

在准备配置端点时，BDT 的 BDnSTAT 字节应该总是最后更新的字节。SIE 在事务完成时将把 UOWN 位清零。

UOWN 位置 1 时，没有任何硬件机制来阻止单片机对存储器的访问。因此，如果单片机试图在 SIE 拥有存储器时修改该存储器，可能会出现无法预料的结果。同样，在 USB 外设将所有权归还单片机之前，读取该存储器可能会导致数据出错。

22.4.1.2 BDNSTAT 寄存器（CPU 模式）

当 UOWN = 0 时，单片机内核拥有 BD。此时，寄存器的其他 7 位担负控制功能。

数据翻转同步使能位 DTSEN（BDnSTAT<3>）控制数据翻转奇偶检验。将 DTSEN 置 1 将使能 SIE 数据翻转同步。使能时，会将数据包的奇偶检验位与 DTS 的值（BDnSTAT<6>）进行比对。如果数据包到达时同步不正确，数据基本上将被忽略。它不会被写入 USB RAM，USB 传输完成中断标志也不会被置 1。但是，SIE 将把 ACK 令牌发送回主机，对接收作出应答。表 22-1 中总结了 DTSEN 位对 SIE 的影响。

缓冲区停止位 BSTALL（BDnSTAT<2>）支持控制传输，通常在端点 0 停止一次。它还支持 USB 规范第 9 章中指定的 SET_FEATURE/CLEAR_FEATURE 命令；通常连续将 STALL 令牌传送给除了默认控制端点外的任何端点。

BSTALL 位使能缓冲区停止。将 BSTALL 置 1，SIE 会把 STALL 令牌归还主机（如果收到的令牌使用该地址处的 BD）。将 STALL 发送到主机时，对应的 UEPn 控制寄存器中的 EPSTALL 位置 1 并产生 STALL 中断。收到 SETUP 令牌前，UOWN 位将保持置 1 且 BD 不会更改。在这种情况下，STALL 状态将被清除，BD 的所有权将归还给单片机内核。

BD<9:8> 位（BDnSTAT<1:0>）存储 SIE 字节计数的高 2 位；低 8 位将存储在对应的 BDnCNT 寄存器中。更多信息，请参见第 22.4.2 节“BD 字节计数”。

表 22-1: DTSEN 位对奇 / 偶（DATA0/DATA1）数据包接收的影响

来自主机的 OUT 数据包	BDnSTAT 设置		收到数据包后的设备响应			
	DTSEN	DTS	握手	UOWN	TRNIF	BDnSTAT 和 USTAT 状态
DATA0	1	0	ACK	0	1	已更新
DATA1	1	0	ACK	1	0	未更新
DATA0	1	1	ACK	1	0	未更新
DATA1	1	1	ACK	0	1	已更新
两者中任一	0	x	ACK	0	1	已更新
两者中任一，有错误	x	x	（无）	1	0	未更新

图注： x = 无关位

注

- 1: 必须在使能 USB 模块之前, 由用户将该位初始化为所需值。
- 2: 除非 DTSEN = 1, 否则该位将被忽略。
- 3: 如果这些位被置 1, USB 可能无法进行通信。因此, 这些位应该始终保持为 0。

22.4.1.3 BDnSTAT 寄存器（SIE 模式）

BD 及其缓冲区由 SIE 所有时，BDnSTAT 中的大多数位都有了不同的含义。配置如寄存器 22-6 所示。一旦 UOWN 被置 1，先前用户写入的任何数据或控制设置都将被来自 SIE 的数据覆盖。

由 SIE 使用令牌数据包标识符（Packet Identifier, PID）更新 BDnSTAT 寄存器，该 PID 存储在 BDnSTAT<5:2> 中。对应 BDnCNT 寄存器中的传输计数值也会更新。该 8 位寄存器的值溢出到计数值的高 2 位，存储在 BDnSTAT<1:0> 中。

22.4.2 BD 字节计数

字节计数代表 IN 传输期间将发送的字节总数。IN 传输完成后，SIE 将返回发送到主机的字节数。

对于 OUT 传输，字节计数代表可接收并存储在 USB RAM 中的最大字节数。OUT 传输完成后，SIE 将返回接收到的实际字节数。如果接收到的字节数超过了设定的字节计数，数据包将被拒绝，并生成一个 NAK 握手信号。在这种情况下，不会更新字节计数。

10 位字节计数保存在两个寄存器中。计数的低 8 位保存在 BDnCNT 寄存器中，而高 2 位保存在 BDnSTAT<1:0> 中。这代表字节数的有效范围是 0 到 1023。

22.4.3 BD 地址验证

BD 地址寄存器对包含对应端点缓冲区的起始 RAM 地址。没有可以验证 BD 地址有效性的硬件机制。

如果 BD 地址的值没有指向 USB RAM 中的地址，或指向另一个端点缓冲区中的地址，数据很可能丢失或被覆盖。同样，如果接收缓冲区（OUT 端点）和使用的 BD 地址重叠，也会导致无法预料的后果。开发 USB 应用程序时，用户可能需要考虑在代码中加入基于软件的地址验证。

寄存器 22-6: BDnSTAT: 缓冲区描述符 n 状态寄存器（BD0STAT 到 BD63STAT），SIE 模式（从 SIE 返回数据到 MCU）（位于普通存储区，地址 4xxh）

R/W-x	r-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
UOWN	r	PID3	PID2	PID1	PID0	BC9	BC8
bit 7							bit 0

图注:	r = 保留位		
R = 可读位	W = 可写位		U = 未实现位，读为 0
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7 **UOWN:** USB 所有权位
1 = SIE 拥有 BD 及其对应缓冲区
- bit 6 **保留:** 不由 SIE 写入
- bit 5-2 **PID<3:0>:** 数据包标识符位
上次传输收到的令牌 PID 值（仅适用于 IN、OUT 或 SETUP 事务）。
- bit 1-0 **BC<9:8>:** 字节计数的 bit 9 和 bit 8
这两位由 SIE 更新，以便反映 OUT 传输中收到的和 IN 传输中发送的实际字节数。

PIC18F46J50 系列

22.4.4 乒乓缓冲

当端点有两组 BD 项时（一组用于偶数数据传输，一组用于奇数数据传输），则该端点被定义为具有乒乓缓冲区。这样就允许在 CPU 处理一组 BD 的同时，SIE 处理另一组 BD。以这种方式实现的双重缓冲 BD 能获得 CPU 和 USB 之间的最大数据吞吐量。

USB 模块支持 4 种工作模式：

- 不支持乒乓缓冲区
- 仅对 OUT 端点 0 提供乒乓缓冲区支持
- 对所有端点提供乒乓缓冲区支持
- 对除端点 0 外的所有其他端点提供乒乓缓冲区支持

乒乓缓冲区设置是用 UCFG 寄存器中的 PPB<1:0> 位配置的。

USB 模块跟踪每个端点的乒乓指针。该模块使能时，所有指针初始时都复位到偶编号 BD。事务完成后（UOWN

由 SIE 清零），指针转向奇编号 BD。下个事务完成后，指针指回偶编号 BD，以此类推。

上个事务的奇 / 偶状态存储在 USTAT 寄存器的 PPBI 位中。用户可以用 PPBRST 位将所有乒乓指针复位到偶编号 BD。

图 22-6 显示了 4 种不同的工作模式，以及 BD 是如何填充 USB RAM 的。

根据缓冲配置，BD 和特定端点有着固定的对应关系。表 22-2 给出了 BD 到端点的映射。这一关系也意味着如果端点不是连续使能的，BDT 中可能出现空白区域。已禁止端点的 BD 理论上可用作缓冲空间，但在实际应用中，用户应避免使用 BDT 中的这类空间，除非采取了验证 BD 地址的措施。

图 22-6: 不同缓冲模式的缓冲区描述符表映射

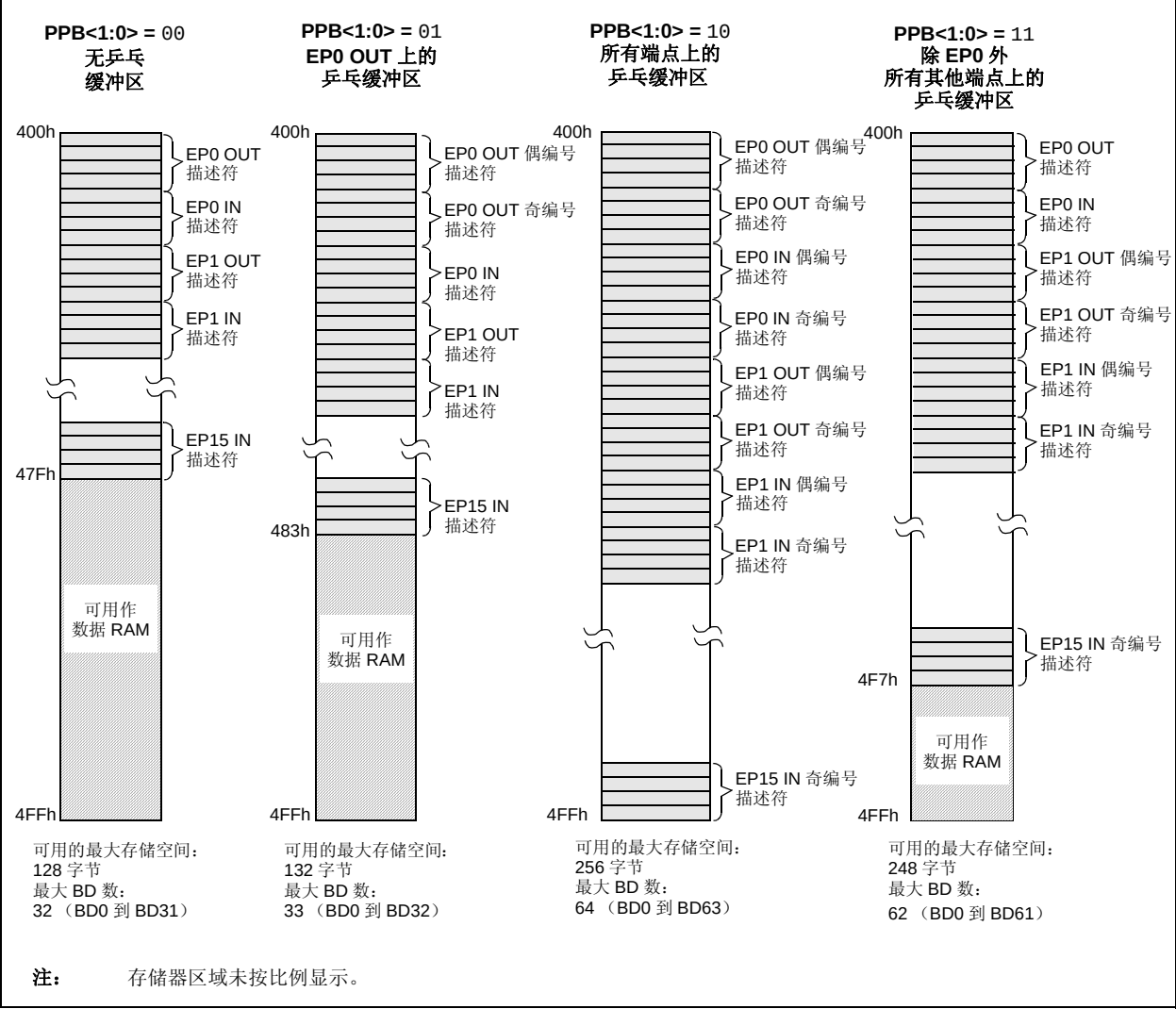


表 22-2: 不同缓冲模式的缓冲区描述符分配

端点	分配给端点的 BD							
	模式 0 (无乒乓缓冲区)		模式 1 (EP0 OUT 上的乒乓缓冲区)		模式 2 (所有端点上的乒乓缓冲区)		模式 3 (除 EP0 外所有其他端点上的乒乓缓冲区)	
	输出	输入	输出	输入	输出	输入	输出	输入
0	0	1	0 (E), 1 (O)	2	0 (E), 1 (O)	2 (E), 3 (O)	0	1
1	2	3	3	4	4 (E), 5 (O)	6 (E), 7 (O)	2 (E), 3 (O)	4 (E), 5 (O)
2	4	5	5	6	8 (E), 9 (O)	10 (E), 11 (O)	6 (E), 7 (O)	8 (E), 9 (O)
3	6	7	7	8	12 (E), 13 (O)	14 (E), 15 (O)	10 (E), 11 (O)	12 (E), 13 (O)
4	8	9	9	10	16 (E), 17 (O)	18 (E), 19 (O)	14 (E), 15 (O)	16 (E), 17 (O)
5	10	11	11	12	20 (E), 21 (O)	22 (E), 23 (O)	18 (E), 19 (O)	20 (E), 21 (O)
6	12	13	13	14	24 (E), 25 (O)	26 (E), 27 (O)	22 (E), 23 (O)	24 (E), 25 (O)
7	14	15	15	16	28 (E), 29 (O)	30 (E), 31 (O)	26 (E), 27 (O)	28 (E), 29 (O)
8	16	17	17	18	32 (E), 33 (O)	34 (E), 35 (O)	30 (E), 31 (O)	32 (E), 33 (O)
9	18	19	19	20	36 (E), 37 (O)	38 (E), 39 (O)	34 (E), 35 (O)	36 (E), 37 (O)
10	20	21	21	22	40 (E), 41 (O)	42 (E), 43 (O)	38 (E), 39 (O)	40 (E), 41 (O)
11	22	23	23	24	44 (E), 45 (O)	46 (E), 47 (O)	42 (E), 43 (O)	44 (E), 45 (O)
12	24	25	25	26	48 (E), 49 (O)	50 (E), 51 (O)	46 (E), 47 (O)	48 (E), 49 (O)
13	26	27	27	28	52 (E), 53 (O)	54 (E), 55 (O)	50 (E), 51 (O)	52 (E), 53 (O)
14	28	29	29	30	56 (E), 57 (O)	58 (E), 59 (O)	54 (E), 55 (O)	56 (E), 57 (O)
15	30	31	31	32	60 (E), 61 (O)	62 (E), 63 (O)	58 (E), 59 (O)	60 (E), 61 (O)

图注: (E) = 偶数事务缓冲区, (O) = 奇数事务缓冲区

表 22-3: USB 缓冲区描述符表寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BDnSTAT ⁽¹⁾	UOWN	DTS ⁽⁴⁾	PID3 ⁽²⁾	PID2 ⁽²⁾	PID1 ⁽²⁾ DTSEN ⁽³⁾	PID0 ⁽²⁾ BSTALL ⁽³⁾	BC9	BC8
BDnCNT ⁽¹⁾	字节计数							
BDnADRL ⁽¹⁾	缓冲区地址低字节							
BDnADRH ⁽¹⁾	缓冲区地址高字节							

- 注 1: 对于缓冲区描述符寄存器, n 是 0 到 63 之间的某个值。为了简明起见, 所有 64 个寄存器都显示为一个通用名称代号。所有寄存器都有不确定的复位值 (xxxx xxxx)。
- 2: 只要 BDnSTAT 寄存器为 SIE 所有 (UOWN 置 1), SIE 就可使用 BDnSTAT 寄存器的 bit 2 到 bit 5 返回 PID<3:0> 的值。一旦寄存器处于 SIE 控制下, 写入 DTSEN 和 BSTALL 的值就不再有效。
- 3: 将缓冲区描述符所有权移交给 SIE 前 (UOWN 位清零), BDnSTAT 寄存器的 bit 5 到 bit 2 用于配置 DTSEN 和 BSTALL 设置。
- 4: 除非 DTSEN = 1, 否则该位将被忽略。

22.5 USB 中断

USB 模块可产生多个中断条件。为了识别所有这些中断源，该模块配备有自身的中断逻辑结构（和单片机的中断逻辑结构相似）。USB 中断是用一组控制寄存器允许的，并用一组独立的标志寄存器来捕捉中断事件。在单片机的中断逻辑中，所有中断源都归为一个 USB 中断请求 USBIF（PIR2<4>）。

图 22-7 给出了 USB 模块的中断逻辑。USB 模块的中断寄存器分为两层。顶层全部由 USB 状态中断组成；这些中断分别是在 UIE 及 UIR 寄存器中允许和标记的。第

二层由 USB 错误条件中断组成，分别是在 UEIE 及 UEIR 寄存器中允许和标记的。上述任意一个中断条件都将触发顶层的 USB 错误中断标志（UERRIF）。

可使用中断来处理 USB 事务中的常规事件。图 22-8 展示了 USB 帧传输过程中的某些常见事件及其对应中断。

图 22-7: USB 中断逻辑

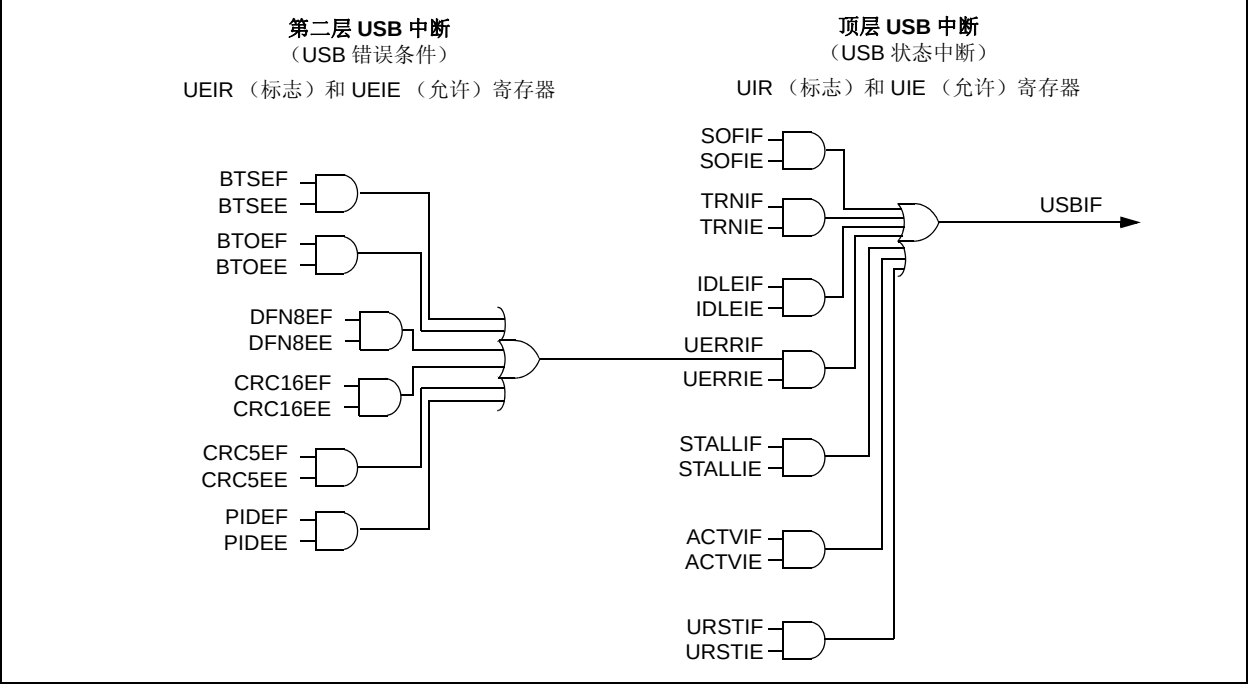
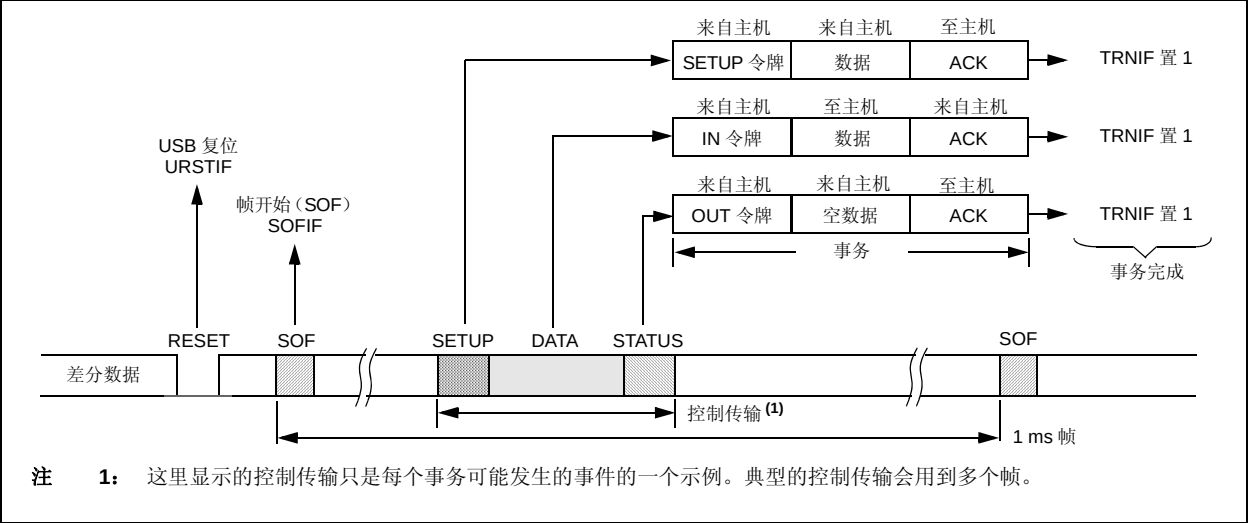


图 22-8: USB 事务和中断事件示例



注 1: 这里显示的控制传输只是每个事务可能发生的事件的一个示例。典型的控制传输会用到多个帧。

22.5.1 USB 中断状态寄存器 (UIR)

USB 中断状态寄存器（寄存器 22-7）包含每个 USB 状态中断源的标志位。这些中断源中的每一个在 UIR 寄存器中都有相应的中断允许位。所有 USB 状态标志一起作逻辑或运算，产生单片机中断逻辑中的 USBIF 中断标志。

一旦由 SIE 将中断位置 1 后，必须由软件通过写入 0 来清零。也可以用软件置 1 标志位，这有助于固件调试。

当 USB 模块处于低功耗挂起模式（UCON<1> = 1）时，将不会为 SIE 提供时钟。处于该状态时，SIE 无法处理数据包，因而无法检测除活动检测中断 ACTVIF 之外的新的中断条件。USB 固件通常使用 ACTVIF 位来检测何时单片机应使 USB 模块退出低功耗挂起模式（UCON<1> = 0）。

寄存器 22-7: UIR: USB 中断状态寄存器（位于快速操作存储区，地址 F62h）

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
—	SOFIF	STALLIF	IDLEIF ⁽¹⁾	TRNIF ⁽²⁾	ACTVIF ⁽³⁾	UERRIF ⁽⁴⁾	URSTIF
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位，读为 0
 -n = POR 时的值 1 = 置 1 0 = 清零 x = 未知

bit 7	未实现: 读为 0
bit 6	SOFIF: 帧起始令牌中断位 1 = SIE 收到的帧起始令牌 0 = SIE 没有收到帧起始令牌
bit 5	STALLIF: STALL 握手中断位 1 = SIE 发送了一个 STALL 握手 0 = 未发送 STALL 握手
bit 4	IDLEIF: 空闲检测中断位 ⁽¹⁾ 1 = 检测到空闲状态（3 ms 或更长的连续空闲状态） 0 = 未检测到空闲状态
bit 3	TRNIF: 事务完成中断位 ⁽²⁾ 1 = 待处理事务处理完成；从 USTAT 寄存器读取端点信息 0 = 未处理完待处理事务或没有待处理事务
bit 2	ACTVIF: 总线活动检测中断位 ⁽³⁾ 1 = D+/D- 线上检测到活动 0 = D+/D- 线上没有检测到活动
bit 1	UERRIF: USB 错误条件中断位 ⁽⁴⁾ 1 = 发生了未屏蔽的错误条件 0 = 未发生未屏蔽的错误条件
bit 0	URSTIF: USB 复位中断位 1 = 发生有效的 USB 复位；00h 装入 UADDR 寄存器 0 = 没有发生 USB 复位

- 注**
- 一旦检测到空闲状态，用户可能希望将 USB 模块置于挂起模式。
 - 将该位清零可导致 USTAT FIFO 指针加 1（仅对 IN、OUT 和 SETUP 令牌有效）。
 - 该位通常只在检测到 UIDLE 中断事件后是未被屏蔽的。
 - 只有通过 UEIE 寄存器使能的错误条件才能将该位置 1。该位只是状态位，用户不能将它置 1 或清零。

22.5.1.1 总线活动检测中断位（ACTVIF）

在USB模块从挂起模式唤醒后，或USB模块被挂起时，不能立即清零ACTVIF位。需要先用几个时钟周期来同步内部硬件状态机，然后固件才能清零ACTVIF位。在内部硬件同步之前清零ACTVIF位可能不会对ACTVIF值产生影响。此外，如果USB模块使用来自96 MHz

PLL源的时钟，那么在清零SUSPND位之后，USB模块需要等待96 MHz PLL锁定，不能立即开始工作。应用程序代码应按例22-1中所示，清零ACTVIF标志。

注：从USB总线空闲状态恢复工作时，只会产生一个ACTVIF中断。如果用户固件清零ACTVIF位，即使存在连续的总线通信，该位也不会立即再次置1。总线通信必须停止足够长的时间来产生另一个IDLEIF条件，然后才会产生另一个ACTVIF中断。

例 22-1: 清零 ACTVIF 位（UIR<2>）

汇编语言：

BCF UCON, SUSPND

LOOP：

BTFS UIR, ACTVIF

BRA DONE

BCF UIR, ACTVIF

BRA LOOP

DONE：

C 语言：

UCONbits.SUSPND = 0;

while (UIRbits.ACTVIF) { UIRbits.ACTVIF = 0; }

22.5.2 USB 中断允许寄存器（UIE）

该寄存器中的值只影响是否将一个中断条件传递给单片机中断逻辑。标志位仍是由其中断条件置 1 的，允许对它们进行查询和处理而无需实际产生中断。

USB 中断允许（UIE）寄存器（寄存器 22-8）包含用于 USB 状态中断源的中断允许位。将这些位中的任何一位置 1 将允许 UIR 寄存器中的对应中断源。

寄存器 22-8: UIE: USB 中断允许寄存器（位于普通存储区，地址 F36h）

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE
bit 7							bit 0

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

bit 7	未实现：读为 0
bit 6	SOFIE : 帧起始令牌中断允许位 1 = 允许帧起始令牌中断 0 = 禁止帧起始令牌中断
bit 5	STALLIE : STALL 握手中断允许位 1 = 允许 STALL 中断 0 = 禁止 STALL 中断
bit 4	IDLEIE : 空闲检测中断允许位 1 = 允许空闲检测中断 0 = 禁止空闲检测中断
bit 3	TRNIE : 事务完成中断允许位 1 = 允许事务中断 0 = 禁止事务中断
bit 2	ACTVIE : 总线活动检测中断允许位 1 = 允许总线活动检测中断 0 = 禁止总线活动检测中断
bit 1	UERRIE : USB 错误中断允许位 1 = 允许 USB 错误中断 0 = 禁止 USB 错误中断
bit 0	URSTIE : USB 复位中断允许位 1 = 允许 USB 复位中断 0 = 禁止 USB 复位中断

PIC18F46J50 系列

22.5.3 USB 错误中断状态寄存器（UEIR）

USB 错误中断状态寄存器（寄存器 22-9）包含 USB 外设中每个错误源的标志位。这些源中的每一个都由 UEIE 寄存器中相应的中断允许位控制。所有 USB 错误标志一起进行逻辑或运算，生成中断逻辑顶层的 USB 错误中断标志（UERRIF）。

一旦检测到错误条件，就会立即将对应的错误标志位置 1。因此，中断通常不会在令牌处理结束时产生。

一旦 SIE 将某中断位置 1，就必须通过软件写入 0 进行清零。

寄存器 22-9: UEIR: USB 错误中断状态寄存器（位于普通存储区，地址 F63h）

R/C-0	U-0	U-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
bit 7							bit 0

图注:			
R = 可读位	C = 可清零位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	BTSEF: 位填充错误标志位 1 = 检测到位填充错误 0 = 没有检测到位填充错误
bit 6-5	未实现: 读为 0
bit 4	BTOEF: 总线周转（Turnaround）超时错误标志位 1 = 发生总线周转超时（在上个 EOP 结束之后，已经过了 16 个位时间以上的空闲时间） 0 = 未发生总线周转超时
bit 3	DFN8EF: 数据字段大小错误标志位 1 = 数据字段的字节数不是整数 0 = 数据字段的字节数是整数
bit 2	CRC16EF: CRC16 失败标志位 1 = CRC16 失败 0 = CRC16 通过
bit 1	CRC5EF: CRC5 主机错误标志位 1 = 令牌包由于 CRC5 错误而被拒绝 0 = 令牌包被接受
bit 0	PIDEF: PID 检查失败标志位 1 = PID 检查失败 0 = PID 检查通过

22.5.4 USB 错误中断允许寄存器（UEIE）

USB 错误中断允许寄存器（寄存器 22-10）包含每个 USB 错误中断源的中断允许位。将这些位中的任何一位置 1，都将允许把 UEIR 寄存器中的对应错误中断源传递到中断逻辑顶层的 UERR 位中。

同 UIE 寄存器一样，该允许位只影响是否将中断条件传递给单片机中断逻辑。标志位仍是由其中断条件置 1 的，允许对它们进行查询和处理而无需实际产生中断。

寄存器 22-10: UEIE: USB 错误中断允许寄存器（位于普通存储区，地址 F37h）

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE
bit 7							bit 0

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

bit 7	BTSEE: 位填充错误中断允许位 1 = 允许位填充错误中断 0 = 禁止位填充错误中断
bit 6-5	未实现: 读为 0
bit 4	BTOEE: 总线周转超时错误中断允许位 1 = 允许总线周转超时错误中断 0 = 禁止总线周转超时错误中断
bit 3	DFN8EE: 数据字段大小错误中断允许位 1 = 允许数据字段大小错误中断 0 = 禁止数据字段大小错误中断
bit 2	CRC16EE: CRC16 失败中断允许位 1 = 允许 CRC16 失败中断 0 = 禁止 CRC16 失败中断
bit 1	CRC5EE: CRC5 主机错误中断允许位 1 = 允许 CRC5 主机错误中断 0 = 禁止 CRC5 主机错误中断
bit 0	PIDEE: PID 检查失败中断允许位 1 = 允许 PID 检查失败中断 0 = 禁止 PID 检查失败中断

PIC18F46J50 系列

22.6 USB 电源模式

许多 USB 应用都可能有多组不同的电源要求和配置。最常见的电源模式是仅总线供电（Bus Power Only）、仅自供电（Self-Power Only）和以自供电为主的双电源供电（Dual Power）模式。这里列出了最常见的情况。同时还提供了一种估算 USB 收发器电流消耗的方法。

22.6.1 仅总线供电

在仅总线供电模式中，应用的所有电源都来自 USB（图 22-9）。这是对设备供电的最简单方式。

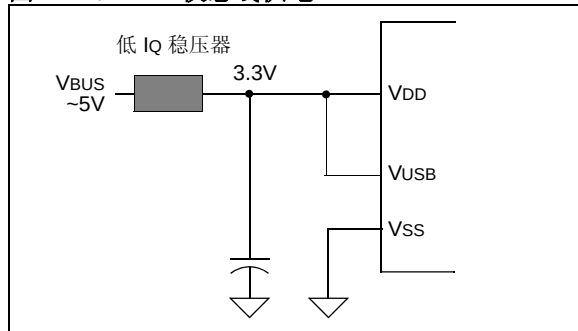
为了满足 USB 2.0 规范的浪涌电流要求，VBUS 和地之间的总有效电容不能超出 10 μF 。如果超出，则需要某种浪涌电流限制措施。更多详细信息，请参见 USB 2.0 规范的第 7.2.4 节。

根据 USB 2.0 规范，所有 USB 设备还必须支持低功耗挂起模式。在 USB 挂起模式下，设备从 USB 电缆的 5V VBUS 线消耗的电流不能超出 2.5 mA。

主机通过以下方式指示 USB 设备进入挂起模式：停止到该 USB 设备的所有 USB 通信，保持超过 3 ms 的时间。该条件会导致 UIR 寄存器中的 IDLEIF 位置 1。

在 USB 挂起模式下，D+ 或 D- 上拉电阻必须保持有效，这会消耗一部分允许的挂起电流：2.5 mA 的预算电流。

图 22-9： 仅总线供电



22.6.2 仅自供电

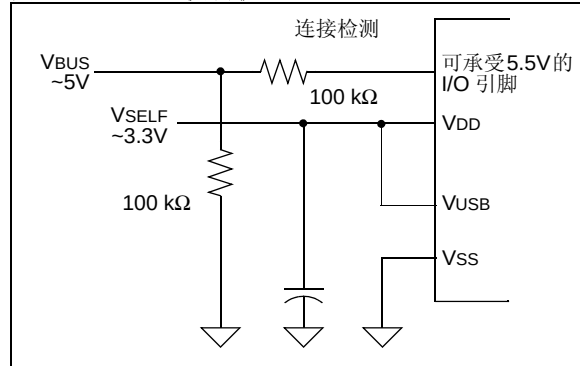
在仅自供电模式中，USB 应用为其自身提供电源，仅有很少一部分电源来自 USB。图 22-10 给出了一个示例。

请注意，其中添加了连接指示，用于指示何时 USB 已经连接，主机正在主动为 VBUS 供电。

为了满足兼容性规范，只有主机主动将 VBUS 驱动为高电平之后，才能使能 USB 模块（以及 D+ 或 D- 上拉电阻）。可以使用一个可承受 5.5V 的 I/O 引脚来实现此目的。

应用不能在 USB 电缆的 5V VBUS 引脚引入任何电流。

图 22-10： 仅自供电

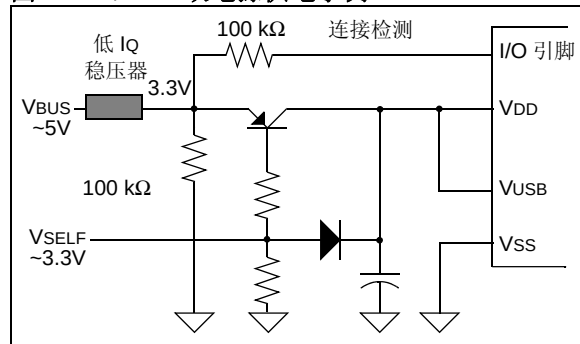


22.6.3 以自供电为主的双电源供电

某些应用可能需要双电源供电选项。在该供电模式下，应用主要使用内部电源，但无内部电源可用时，可切换到 USB 供电。图 22-11 所示为一个以自供电为主的双电源供电模式的示例，它可在仅自供电模式和仅 USB 总线供电模式之间自动切换。

双电源供电设备还必须满足对于浪涌电流和挂起模式电流的所有特殊要求，并且只有 VBUS 驱动为高电平之后才能使能 USB 模块。关于这些要求的说明，请参见第 22.6.1 节“仅总线供电”和第 22.6.2 节“仅自供电”。此外，双电源供电器件永远不能在 USB 电缆的 5V VBUS 引脚引入任何电流。

图 22-11： 双电源供电示例



注： 用户应牢记设备由 USB 总线供电的电流限制。根据 USB 规范 2.0，对每个低功耗设备，不能超过 100 mA；对每个高功耗设备，不能超过 500 mA。

22.6.4 USB 收发器的电流消耗

根据 USB 电缆的特性阻抗、电缆的长度、V_{USB} 供电电压和通过 USB 电缆的实际数据模式，USB 收发器会消耗不同的电流大小。电缆越长，电容就越大，在切换输出状态时，消耗的总电能就越多。

由“IN”（传入）通信组成的数据模式消耗的电流远高于“OUT”（外发）通信。IN 通信要求 PIC[®] MCU 驱动 USB 电缆，而 OUT 通信要求主机驱动 USB 电缆。

通过 USB 电缆发送的数据采用 NRZI 编码。在 NRZI 编码方案中，数据位为 0 时，收发器的输出状态会发生翻转（从“J”状态变为“K”状态，或者反之）。进行位填充时的情况除外，NRZI 编码的数据位为 1 时，收发器的

输出状态不会改变。因此，由值为 0 的数据位组成的 IN 通信会消耗最大的电流，因为收发器必须对 USB 电缆充电 / 放电，以便改变状态。

关于 NRZI 编码和位填充的更多详细信息，请参见 USB 规范的第 7.1 节，虽然使用 PIC18F46J50 系列单片机实现 USB 应用并不需要知道这些详细信息。此外，SIE 还使用硬件处理位填充 / 去填充、NRZI 编码 / 解码和 CRC 生成 / 校验。

收发器消耗的总电流将取决于具体应用。但是，为了帮助估算在全速应用中实际可能需要多少电流，可以使用 [公式 22-1](#)。

[公式 22-2](#) 说明了可以如何对理论应用使用该公式。

公式 22-1: 估算 USB 收发器的电流消耗

$$I_{XCVR} = \frac{(40\text{ mA} \cdot V_{USB} \cdot P_{ZERO} \cdot P_{IN} \cdot L_{CABLE})}{(3.3\text{V} \cdot 5\text{m})} + I_{PULLUP}$$

图注：

V_{USB}——施加到 V_{USB} 引脚的电压，以 V 为单位（应为 3.0V 至 3.6V）。

P_{ZERO}——PIC[®] MCU 发送的 IN 通信位中，值为 0 的数据位的百分比（以十进制表示）。

P_{IN}——总线总带宽中用于 IN 通信的百分比（以十进制表示）。

L_{CABLE}——USB 电缆的长度（以米为单位）。USB 2.0 规范要求全速应用使用的电缆不超出 5 米。

I_{PULLUP}——1.5 kΩ 标称上拉电阻（使能时）必须为 USB 电缆提供的电流。在 USB 电缆的主机或集线器端，存在 15 kΩ 的标称电阻（14.25 kΩ 至 24.8 kΩ），它将 D+ 和 D- 线拉到地。在总线空闲条件下（例如，在两个数据包之间或在 USB 挂起模式下），电压为 3.3V 时，这会产生最大 218 μA 的静态电流。

I_{PULLUP} 也取决于总线通信条件，当数据在大多数时间都将线路驱动为“K”状态，USB 带宽得到完全利用（IN 或 OUT 通信）时，电流最高可以达到 2.2 mA。

PIC18F46J50 系列

公式 22-2: 计算 USB 收发器电流†

对于该示例，应用存在以下假设：

- 为 VUSB 和 VDD 施加 3.3V 的电压，并使能内核稳压器。
- 这是一个全速应用，使用一个可以每 1 ms 发送一个 64 字节数据包的中断 IN 端点，并且对发送的字节值没有限制。在 OUT 端点上，应用可能有也可能没有另外的通信。
- 在应用电路板中，将使用常规的 USB“B”或“mini-B”连接器。

在此例中，PZERO = 100% = 1，因为对通过 IN 端点的数据值没有限制。所有 64 kbps 的数据都可能是值为 00h 的字节。因为数据位为 0 时，收发器的输出状态会发生翻转，所以它们会导致 USB 收发器消耗额外的电流来对电缆进行充电 / 放电。在此例中，所发送的数据位都可以为值 0。该值应视为“最大”值，因为通常情况下，数据将由一定比例的 1 和 0 组成。

对于 IN 通信，该应用使用 1.5 Mbps（12 Mbps）总线总带宽中的 64 kbps，因此：

$$\text{引脚} = \frac{64 \text{ kbps}}{1.5 \text{ Mbps}} = 4.3\% = 0.043$$

因为该应用使用常规的“B”或“mini-B”连接器，所以最终用户可以插入任意类型的电缆（最大允许长度为 5 米）。因此，使用最差情况的长度：

$$\text{LCABLE} = 5 \text{ 米}$$

假设 IPULLUP = 2.2 mA。IPULLUP 的实际值可能更接近 218 μA，但这样的假设可以容许最差情况。USB 带宽由插入根端口（通过集线器）的所有设备共用。如果将应用插入已插有其他设备的 USB 1.1 集线器，则用户的设备可能会在总线上检测到主机到设备的通信，即使通信不是发送给该设备的。因为不论来源如何，所有通信都会使 IPULLUP 电流升高于基本电流 218 μA，所以容许最差情况（2.2 mA 的电流）是最安全的。

因此：

$$\text{IXCVR} = \frac{(40 \text{ mA} \cdot 3.3\text{V} \cdot 1 \cdot 0.043 \cdot 5\text{m})}{(3.3\text{V} \cdot 5\text{m})} + 2.2 \text{ mA} = 3.9 \text{ mA}$$

† 计算得到的值应视为近似值，建议留有一定的安全余量或执行取决于具体应用的产品测试。收发器电流与 PIC18F46J50 系列器件为了运行内核、驱动其他 I/O 线、为各种模块供电等方面需要消耗的其他电流一起形成总电流消耗。

22.7 振荡器

USB 模块有特定的时钟要求。对于全速工作模式，时钟源必须是 48 MHz。虽然如此，单片机内核和其他外设不一定要以该时钟频率运行。可用的时钟选项在 [第 3.3 节“USB 的振荡器设置”](#) 中有详细描述。

22.8 USB 固件和驱动程序

Microchip 提供一系列针对特定应用的资源，例如 USB 固件和驱动程序。请访问 www.microchip.com 获得最新的固件和驱动程序。

表 22-4: 与 USB 模块操作相关的寄存器 ⁽¹⁾

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	详情请见 (页):
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	71
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	71
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	73
UCFG	UTEYE	UOEMON	—	UPUEN	UTRDIS	FSEN	PPB1	PPB0	74
USTAT	—	ENDP3	ENDP2	ENDP1	ENDP0	DIR	PPBI	—	73
UADDR	—	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	74
UFRML	FRM7	FRM6	FRM5	FRM4	FRM3	FRM2	FRM1	FRM0	73
UFRMH	—	—	—	—	—	FRM10	FRM9	FRM8	73
UIR	—	SOFIF	STALLIF	IDLEIF	TRNIF	ACTVIF	UERRIF	URSTIF	73
UIE	—	SOFIE	STALLIE	IDLEIE	TRNIE	ACTVIE	UERRIE	URSTIE	74
UEIR	BTSEF	—	—	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	73
UEIE	BTSEE	—	—	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	74
UEP0	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	75
UEP1	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	75
UEP2	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	75
UEP3	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	75
UEP4	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	75
UEP5	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	75
UEP6	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	75
UEP7	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP8	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP9	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP10	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP11	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP12	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP13	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP14	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74
UEP15	—	—	—	EPHSHK	EPCONDIS	EPOUTEN	EPINEN	EPSTALL	74

图注: — = 未实现，读为 0。USB 模块不使用阴影单元。

注 1: 该表只包含在数据存储空间 Bank 15 中硬件映射的 SFR。缓冲区描述符寄存器映射到 Bank 4，它们不是真正的 SFR，另外列在 [表 22-3](#) 中。

22.9 USB 概述

本节介绍 USB 的一些基本概念和设计 USB 设备所需的有用信息。尽管本节涵盖了很多内容，但 USB 规范和分类规范中提供了更为详尽的信息。因此，建议读者参考 USB 规范了解更多信息（www.usb.org）。如果您对 USB 非常熟悉，阅读本节可以帮您快速重温 USB 的相关知识。

22.9.1 分层架构

USB 设备的功能是以分层架构的形式实现的，如图 22-12 所示。每一层都和设备的某个功能层次关联。最高层是配置层。一个设备可有多种配置。例如，在仅自供电或仅总线供电模式的基础上，某个特定器件可能有多种供电要求。

对每种配置，可能有多个接口。每个接口支持该配置的某个特定模式。

接口之下是端点。数据直接在该层传送。最多可有 16 个双向端点。端点 0 总是控制端点，并且当设备在总线上时，默认情况下端点 0 必须可用于配置设备。

22.9.2 帧

在总线上通信的信息通常被划分为多个 1 ms 长的时段，称为“帧”。每个帧可包含多个设备和端点的许多事务。图 22-8 给出了帧内一个事务的示例。

22.9.3 传输

USB 规范中定义了四种传输类型。

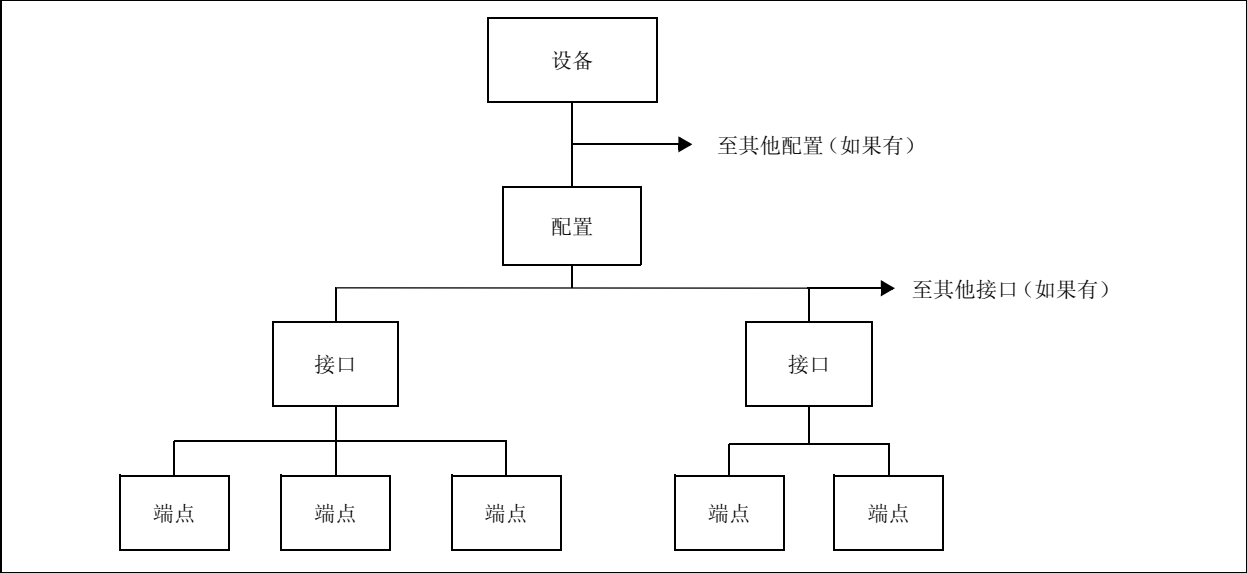
- **同步：**该类型可确保及时传递大量数据（最多可达 1023 字节）；但不能保证数据的完整性。这适用于少量数据丢失影响不大的数据流应用，例如音频。
- **批量：**这类传输方式允许传输大量数据并确保数据的完整性；但不能保证传输时限。
- **中断：**这类传输可确保小块数据的及时传递，并能保证数据的完整性。
- **控制：**这类传输是为设备启动控制提供的。

全速设备支持所有传输类型，而低速设备只限于中断和控制传输类型。

22.9.4 电源

电源可从 USB 获得。USB 规范定义了总线供电的要求。设备既可以使用自身电源，也可以通过总线供电。自供电设备从外部电源获得电能，而总线供电设备使用总线提供的电源。

图 22-12: USB 层



USB 规范对从总线的获得电能做出了限制。在电压大约为 5V 的条件下，每个设备确保能获得 100 mA 的电流（一个单位负载）。可能需要更多的电能，最大为 500 mA。

注意，可以请求供给超过一个单位负载的电能，但主机或集线器不一定会提供额外电流。因此，耗电量可能超过单位负载的设备必要时必须能够置于低功耗配置（一个单位负载及以下）下。

USB 规范还定义了挂起模式。在这种情况下，1 秒内的平均电流不能超过 500 μ A。总线上没有任何活动 3 ms（即 3 ms 内没有 SOF 令牌）后，设备必须进入挂起状态。进入挂起模式的设备必须在挂起后 10 ms 内降低电流消耗。类似地，在唤醒时，设备必须在 10 ms 内消耗电流超出挂起门限值以示唤醒。

22.9.5 枚举

设备最初连接到总线时，主机进入枚举过程，试图识别设备。基本上，主机会询问设备，收集类似功耗、数据传输速率和大小、协议及其他描述信息；描述符将包含这些信息。下面是一个典型的枚举过程：

1. **USB 复位** —— 将设备复位。因此，设备未配置，也没有地址（地址 0）。
2. **获取设备描述符**——主机请求设备描述符的一小部分。
3. **USB 复位** —— 再次将设备复位。
4. **设置地址** —— 主机将地址分配给设备。
5. **获取设备描述符** —— 主机检索设备描述符，收集类似制造商、设备类型、最大控制包大小等信息。
6. **获取配置描述符**。
7. **获取任何其他描述符**。
8. **设置配置**。

具体的枚举进程取决于主机。

22.9.6 描述符

有 8 种不同的标准描述符类型，其中 5 种对这种设备来讲非常重要。

22.9.6.1 设备描述符

设备描述符提供常规信息，例如制造商、产品编号、序列号、设备类别和配置数。只有一个设备描述符。

22.9.6.2 配置描述符

配置描述符提供设备功耗要求以及在该配置下支持多少不同接口的信息。一个设备可能有多种配置（即低功耗和高功耗配置）。

22.9.6.3 接口描述符

接口描述符详细描述该接口中使用的端点数，以及接口类型。一个配置可能有多个接口。

22.9.6.4 端点描述符

端点描述符确定传输类型（[第 22.9.3 节“传输”](#)）和方向，以及端点的一些其他定义。一个设备中可以有許多端点，端点可在不同配置间共用。

22.9.6.5 字符串描述符

许多前面的描述符都引用了一个或多个字符串描述符。字符串描述符提供关于它们描述的层（[第 22.9.1 节“分层架构”](#)）的可读信息。通常，这些字符串出现在主机中，帮助用户识别设备。为了节省存储空间，字符串描述符通常是可选的，采用 unicode 格式编码。

22.9.7 总线速度

每个 USB 设备都必须告知主机其是否连接在总线上以及总线速度。在连接时这是通过连接到总线的 1.5 k Ω 电阻实现的。

根据设备速度，电阻可将 D+ 或 D- 线上拉到 3.3V。对于低速设备，上拉电阻连接到 D- 线。对于全速设备，上拉电阻连接到 D+ 线。

22.9.8 分类规范和驱动程序

USB 规范包含操作系统供应商可选择支持的分类型规范。类别包括诸如音频、大容量存储、通信和人机界面（Human Interface, HID）。多数情况下，主机端需要驱动程序来与 USB 设备“对话”。在定制应用中，可能需要开发驱动程序。幸运的是，对于绝大多数常见类型的设备来说，常见主机系统中均配有驱动程序。因此，这些驱动程序可以被重用。

PIC18F46J50 系列

注:

23.0 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独立配置。输入可以从模拟输入和两个内部参考电压中选择。数字输出可从引脚电平读取，也可通过控制寄存器读取。此外，还提供多种输出和中断事件产生方式。图 23-1 给出了模块的通用单比较器的图示。

该模块的主要特性包括：

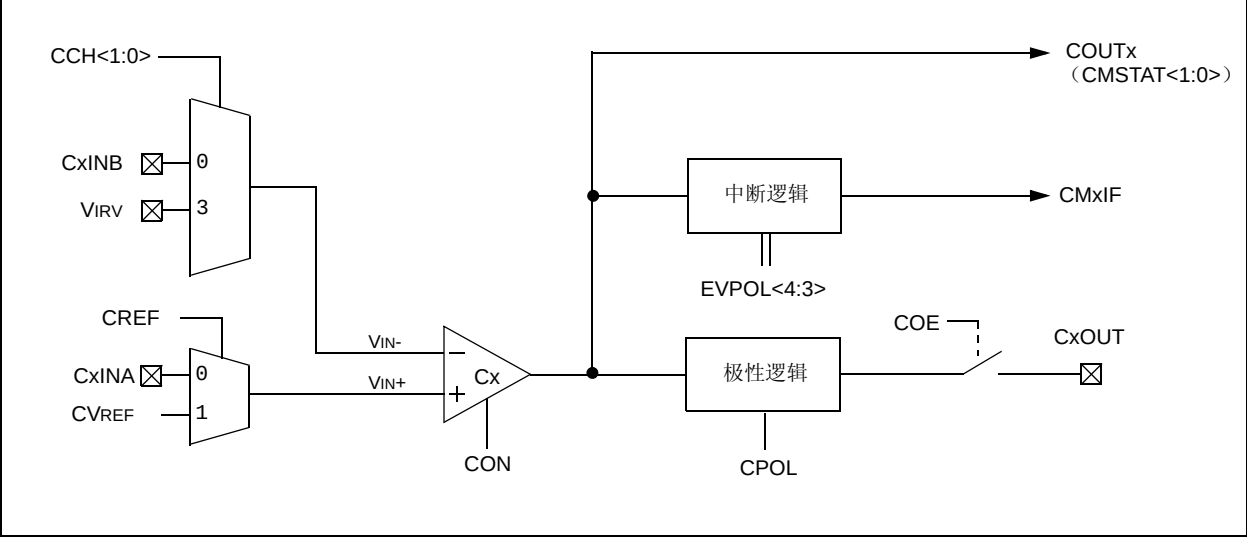
- 独立的比较器控制
- 可编程输入配置
- 输出到引脚和寄存器
- 可编程输出极性
- 每个比较器可独立产生中断，并可配置电平变化中断

23.1 寄存器

CMxCON 寄存器（寄存器 23-1）用于选择每个比较器的输入和输出配置，以及中断产生的设置。

CMSTAT 寄存器（寄存器 23-2）用于提供比较器的输出结果。该寄存器中的位是只读位。

图 23-1: 比较器的简化框图



PIC18F46J50 系列

寄存器 23-1: CMxCON: 比较器控制 x 寄存器 (位于快速操作存储区, 地址 FD2h 和 FD1h)

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 = POR 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 = 比较器输出出现在 CxOUT 引脚上 (在 PPS 模块中指定) 0 = 比较器输出仅在内部有效
bit 5	CPOL: 比较器输出极性选择位 1 = 比较器输出反相 0 = 比较器输出不反相
bit 4-3	EVPOL<1:0>: 中断极性选择位 11 = 在输出发生任何变化时产生中断 ⁽¹⁾ 10 = 仅在输出从高电平跳变为低电平时产生中断 01 = 仅在输出从低电平跳变为高电平时产生中断 00 = 禁止产生中断
bit 2	CREF: 比较器参考电压选择位 (同相输入) 1 = 同相输入连接至内部 CVREF 电压 0 = 同相输入连接至 CxINA 引脚
bit 1-0	CCH<1:0>: 比较器通道选择位 11 = 比较器的反相输入连接至 V _{IRV} (0.6V) 00 = 比较器的反相输入连接至 CxINB 引脚

注 1: 每次选择该模式时, CMxIF 会自动置 1, 在进行初始配置之后, 该位必须由应用程序清零。

寄存器 23-2: CMSTAT: 比较器状态寄存器 (位于快速操作存储区, 地址 F70h)

U-0	U-0	U-0	U-0	U-0	U-0	R-1	R-1
—	—	—	—	—	—	COUT2	COUT1
bit 7							bit 0

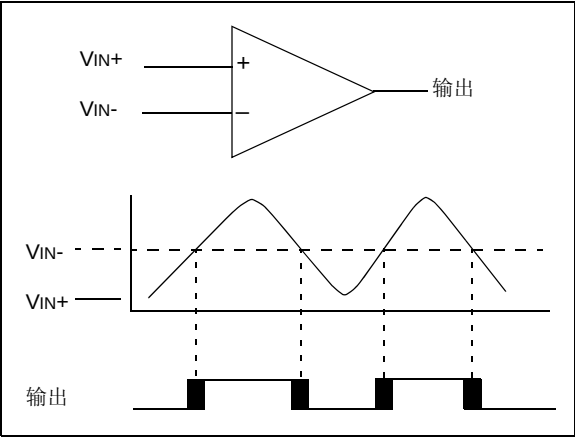
图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2	未实现: 读为 0
bit 1-0	COUT<2:1>: 比较器 x 状态位 <u>如果 CPOL = 0 (极性不反相):</u> 1 = 比较器 V _{IN+} > V _{IN-} 0 = 比较器 V _{IN+} < V _{IN-} <u>如果 CPOL = 1 (极性反相):</u> 1 = 比较器 V _{IN+} < V _{IN-} 0 = 比较器 V _{IN+} > V _{IN-}

23.2 比较器工作原理

图 23-2 所示为单比较器以及模拟输入电平与数字输出之间的关系。当 V_{IN+} 上的模拟输入电压小于 V_{IN-} 上的模拟输入电压时，比较器输出为数字低电平。当 V_{IN+} 上的模拟输入电压大于 V_{IN-} 上的模拟输入电压时，比较器输出为数字高电平。图 23-2 中比较器输出的黑色区域表示因输入失调和响应时间所造成的输出不确定区域。

图 23-2: 单比较器



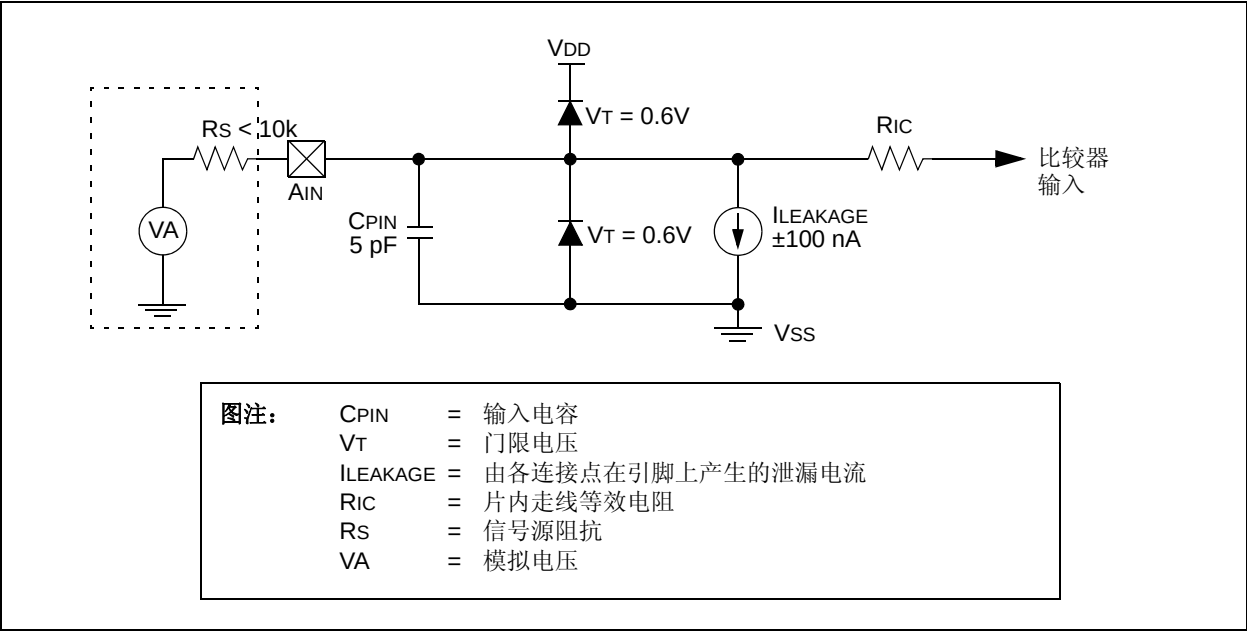
23.3 比较器响应时间

响应时间是指在选择新的参考电压或输入源后，比较器输出达到有效电平的最短时间。比较器的响应时间不同于参考电压的稳定时间。因此，在确定比较器输入改变的总响应时间时，必须同时考虑这两个时间。否则，应使用比较器的最大延时（见第 30.0 节“电气特性”）。

23.4 模拟输入连接注意事项

模拟输入的简化电路如图 23-3 所示。由于模拟引脚被连接到数字输出端，它们在 V_{DD} 和 V_{SS} 之间连有反向偏置的二极管。因此，模拟输入必须在 V_{SS} 和 V_{DD} 之间。如果输入电压与这一范围偏离的绝对值超过 $0.6V$ ，就可能发生一个二极管正向导通，从而可能导致锁死发生。模拟信号源的最大阻抗推荐值为 $10\text{ k}\Omega$ 。任何连接到模拟输入引脚的外部元件（如电容或齐纳二极管），要保证其泄漏电流极小。

图 23-3: 比较器模拟输入模型



23.5 比较器控制和配置

每个比较器具有最多 8 种可能的输入组合：最多 4 种外部模拟输入，以及两个内部参考电压之一。

两个比较器都允许选择来自引脚 CxINA 的信号，或者来自同相通道上的比较器参考电压（CVREF）。该输入与 CxINB、CTMU 或反相通道上的单片机固定内部参考电压（VIRV，标称值为 0.6V）进行比较。

表 23-1 列出了与固定 I/O 引脚关联的比较器输入和输出。

表 23-1: 比较器输入和输出

比较器	输入或输出	I/O 引脚
1	C1INA (VIN+)	RA0
	C1INB (VIN-)	RA3
	C1OUT	可重映射的 RPn
2	C2INA (VIN+)	RA1
	C2INB (VIN-)	RA2
	C2OUT	可重映射的 RPn

23.5.1 比较器使能和输入选择

将 CMxCON 寄存器的 CON 位（CMxCON<7>）置 1 可以使能比较器操作。清零 CON 位可以禁止比较器，以使电流消耗降至最低。

CMxCON 寄存器中的 CCH<1:0> 位（CMxCON<1:0>）指示 3 个模拟输入引脚之一或内部参考电压（VIRV）连接到比较器 VIN-。根据不同的比较器工作模式，可选择使用外部或内部参考电压。将 VIN- 上的模拟信号与 VIN+ 上的信号作比较，并相应地调整比较器的数字输出。

当 CREF = 0（CMxCON<2>），且 VIN+ 连接到 CxINA 引脚时，将使用外部参考电压。使用外部参考电压时，比较器模块可以配置为使用外部参考电压源。参考电压信号必须在 VSS 和 VDD 之间，并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内部比较器参考电压模块产生的参考电压（CVREF）。在第 23.0 节“比较器模块”中详细介绍了该模块。仅当 CREF = 1 时，来自比较器参考电压模块的参考电压才可用。在该模式下，内部参考电压被施加到比较器的 VIN+ 引脚上。

注：通过 CCH<1:0> 选择的比较器输入引脚必须配置为输入，即将 ANCON1 寄存器中相应的 TRIS 和 PCFG 位置 1。

23.5.2 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，而 CMSTAT<1> 读取比较器 2 的输出。这些位是只读的。

通过将 COE 位（CMxCON<6>）置 1，比较器输出也可以直接输出到 RPn I/O 引脚。在使能时，引脚输出路径中的多路开关将切换到比较器输出。

默认情况下，每当 VIN+ 上的电压高于 VIN- 上的电压时，比较器的输出为逻辑高电平。比较器输出的极性可以使用 CPOL 位（CMxCON<5>）进行反相。

每个比较器输出的不确定区域的大小与规范中给出的输入失调电压和响应时间有关，如第 23.2 节“比较器工作原理”中所讨论。

23.6 比较器中断

每当发生以下任意事件时，比较器中断标志都将被置 1

- 比较器输出从低电平跳变为高电平
- 比较器输出从高电平跳变为低电平
- 比较器输出发生任何变化

比较器中断选择通过 CMxCON 寄存器中的 EVPOL<1:0> 位（CMxCON<4:3>）完成。

为了提供最大的灵活性，比较器的输出可以使用 CMxCON 寄存器中的 CPOL 位（CMxCON<5>）进行反相。这在功能上等效于对于特定模式颠倒比较器的反相和同相输入。

在比较器输出从低电平跳变为高电平或从高电平跳变为低电平时产生中断。这种中断产生模式依赖于 CMxCON 寄存器中的 EVPOL<1:0> 位。当 EVPOL<1:0> = 01 或 10 时，在比较器输出从低电平跳变为高电平或从高电平跳变为低电平时产生中断。在产生中断后，需要用软件清零中断标志。

当 EVPOL<1:0> = 11 时，任一比较器的输出值发生变化，都会将该比较器的中断标志位置 1。需要用软件保存输出位的状态信息（从 CMSTAT<1:0> 读取），以确定实际发生的变化。CMxIF 位（PIR2<6:5>）是比较器中断标志。CMxIF 位必须通过清零复位。由于可以向该寄存器写入 1，因此可以模拟中断产生。

表 23-2 列出了对应于比较器输入电压和 EVPOL 位设置的中断产生情况。

必须将 CMxIE 位（PIE2<6:5>）和 PEIE 位（INTCON<6>）置 1 以允许中断。此外，还必须将 GIE 位（INTCON<7>）置 1。如果这些位中的任何一个被清零，将无法允许中断，尽管中断条件发生时仍会将 CMxIF 位置 1。

图 23-3 给出了中断部分的简化框图。

表 23-2: 比较器中断产生

CPOL	EVPOL<1:0>	比较器输入变化	COUTx 跳变	中断产生
0	00	VIN+ > VIN-	低电平到高电平	否
		VIN+ < VIN-	高电平到低电平	否
	01	VIN+ > VIN-	低电平到高电平	是
		VIN+ < VIN-	高电平到低电平	否
	10	VIN+ > VIN-	低电平到高电平	否
		VIN+ < VIN-	高电平到低电平	是
	11	VIN+ > VIN-	低电平到高电平	是
		VIN+ < VIN-	高电平到低电平	是
1	00	VIN+ > VIN-	高电平到低电平	否
		VIN+ < VIN-	低电平到高电平	否
	01	VIN+ > VIN-	高电平到低电平	否
		VIN+ < VIN-	低电平到高电平	是
	10	VIN+ > VIN-	高电平到低电平	是
		VIN+ < VIN-	低电平到高电平	否
	11	VIN+ > VIN-	高电平到低电平	是
		VIN+ < VIN-	低电平到高电平	是

PIC18F46J50 系列

23.7 休眠期间的比较器操作

当比较器处于工作状态而器件处于休眠模式时，比较器仍保持工作状态并可产生中断（如果中断被允许）。中断会将器件从休眠模式唤醒。每个工作比较器都将消耗额外电流。若要将休眠模式下的功耗降至最低，可在进入休眠模式前关闭比较器（CON = 0）。如果器件从休眠模式唤醒，CMxCON 寄存器的内容不受影响。

23.8 复位的影响

器件复位强制 CMxCON 寄存器为复位状态。这使比较器和参考电压被强制为“关闭”状态。

表 23-3: 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 所在页:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	71
CMxCON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	70
CVRCON	CVREN	CVROE	CVRR	r	CVR3	CVR2	CVR1	CVR0	74
CMSTAT	—	—	—	—	—	—	COUT2	COUT1	73
ANCON0	PCFG7 ⁽¹⁾	PCFG6 ⁽¹⁾	PCFG5 ⁽¹⁾	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	74
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	72

图注： — = 未实现（读为 0），r = 保留。比较器模块的工作不会使用阴影单元。

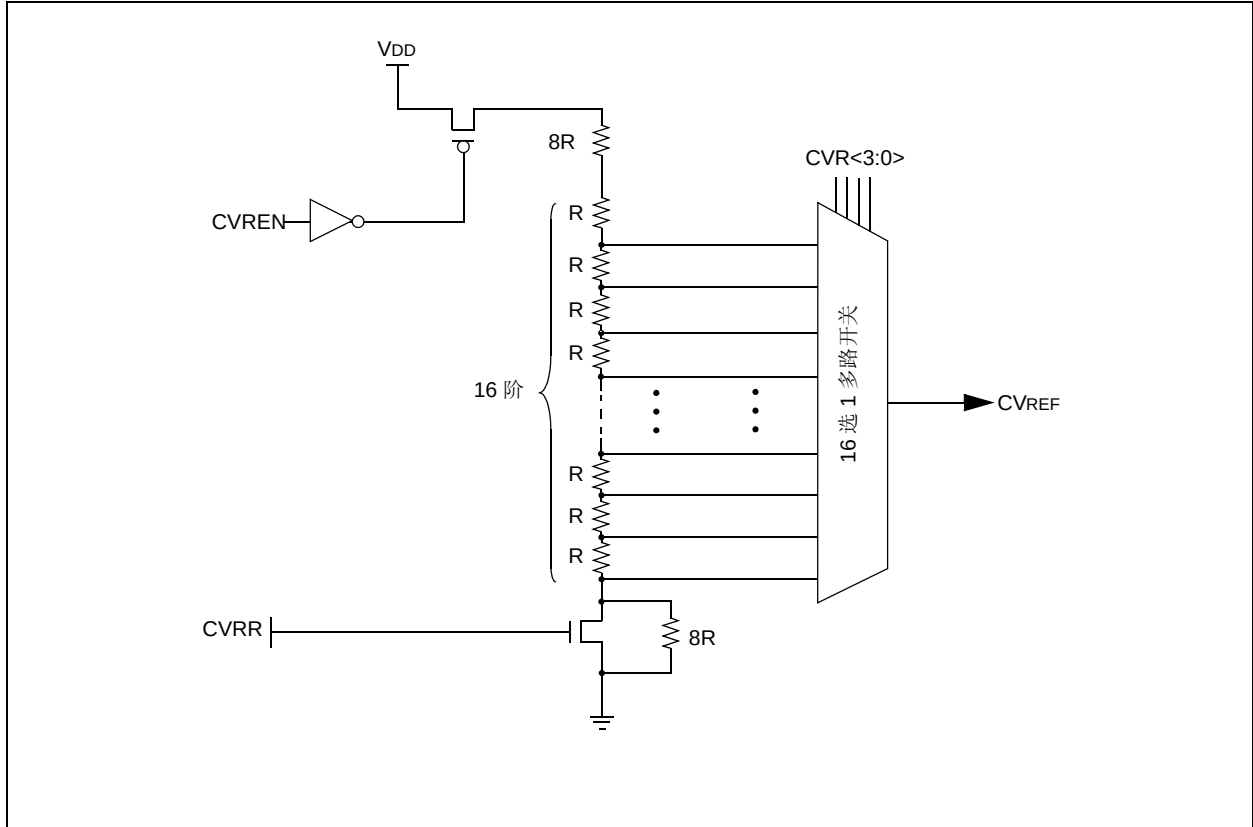
注 1: 这些位和 / 或寄存器在 28 引脚器件上未实现。

24.0 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，可提供多个参考电压供选择。虽然它的主要目的是为模拟比较器提供参考电压，但也可将它用于其他场合。

图 24-1 给出了该模块的框图。梯形电阻经过分段可提供两种范围的 CVREF 值，并且该网络还具有断电功能，可以在不使用参考电压的情况下节省功耗。该模块的参考电压源由 VDD/VSS 提供。

图 24-1: 比较器参考电压模块框图



PIC18F46J50 系列

24.1 配置比较器参考电压

比较器参考电压模块是通过 CVRCON 寄存器（寄存器 24-1）来控制的。比较器参考电压模块提供两种范围的输出电压，每种范围都具有 16 个不同的电压。通过 CVRR 位（CVRCON<5>）选择输出电压的范围。这两种范围的主要区别在于其电压值之间的步长不同（其中一种范围可提供较高的分辨率），该步长由 CVREF 选择位（CVR<3:0>）来决定。下面是计算比较器参考电压输出值的公式：

公式 24-1： 计算比较器参考电压输出值

当 CVRR = 1 时：
$CVREF = ((CVR<3:0>)/24) \times (VDD)$
当 CVRR = 0 时：
$CVREF = (VDD/4) + ((CVR<3:0>)/32) \times (VDD)$

在更改 CVREF 输出值时，必须考虑比较器参考电压的稳定时间（见第 30.0 节“电气特性”中的表 30-3）。

寄存器 24-1： CVRCON：比较器参考电压控制寄存器（位于普通存储区，地址 F53h）

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	r	CVR3	CVR2	CVR1	CVR0
bit 7				bit 0			

图注：	r = 保留位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 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 = CVREF 电路上电
0 = CVREF 电路断电
- bit 6 **CVROE：**比较器 VREF 输出使能位 ⁽¹⁾
1 = CVREF 电压也从 RA2/AN2/VREF-/CVREF/C2INB 引脚输出
0 = CVREF 电压从 RA2/AN2/VREF-/CVREF/C2INB 引脚断开
- bit 5 **CVRR：**比较器 VREF 范围选择位
1 = 0 至 0.667 VDD，相邻值之间的步长为 VDD/24（低电压范围）
0 = 0.25 VDD 至 0.75 VDD，相邻值之间的步长为 VDD/32（高电压范围）
- bit 4 **保留：**始终保持为 0
- bit 3-0 **CVR<3:0>：**比较器 VREF 值选择位（0 ≤ (CVR<3:0>) ≤ 15）
 当 CVRR = 1 时：
 $CVREF = ((CVR<3:0>)/24) \cdot (VDD)$
 当 CVRR = 0 时：
 $CVREF = (VDD/4) + ((CVR<3:0>)/32) \cdot (VDD)$

注 1： CVROE 改写 TRIS 位设置。

24.2 参考电压精度 / 误差

由于模块结构的限制，并不能实现整个参考电压范围的满量程输出。梯形电阻网络顶部和底部的晶体管（见图 24-1）使 CVREF 值不能达到参考电压源的满幅值。参考电压是由参考电压源分压而来的，因此 CVREF 输出随参考电压源的波动而变化。参考电压的精度，请参见第 30.0 节“电气特性”。

24.3 连接注意事项

参考电压模块的工作独立于比较器模块。如果 CVROE 位被置 1，那么参考电压发生器的输出可与 RA2 引脚相连。当 RA2 被配置为数字输入引脚时，将参考电压输出连接到 RA2 引脚，将会增加电流消耗。

RA2 引脚可被用作简单的 D/A 输出，但是其驱动能力有限。要提高电流驱动能力，VREF 参考电压输出端必须外接缓冲器。请参见图 24-2 了解缓冲技术示例。

24.4 休眠期间的操作

如果因中断或看门狗定时器超时将器件从休眠模式唤醒，CVRCON 寄存器的内容将不受影响。为了降低休眠模式下的电流消耗，应禁止参考电压模块。

24.5 复位的影响

器件复位时，CVREN 位（CVRCON<7>）将被清零从而禁止参考电压模块。复位还将 CVROE 位（CVRCON<6>）清零，使参考电压从 RA2 引脚断开；同时通过将 CVRR 位（CVRCON<5>）清零选择高电压范围。CVR 值选择位也将清零。

图 24-2: 参考电压输出缓冲示例

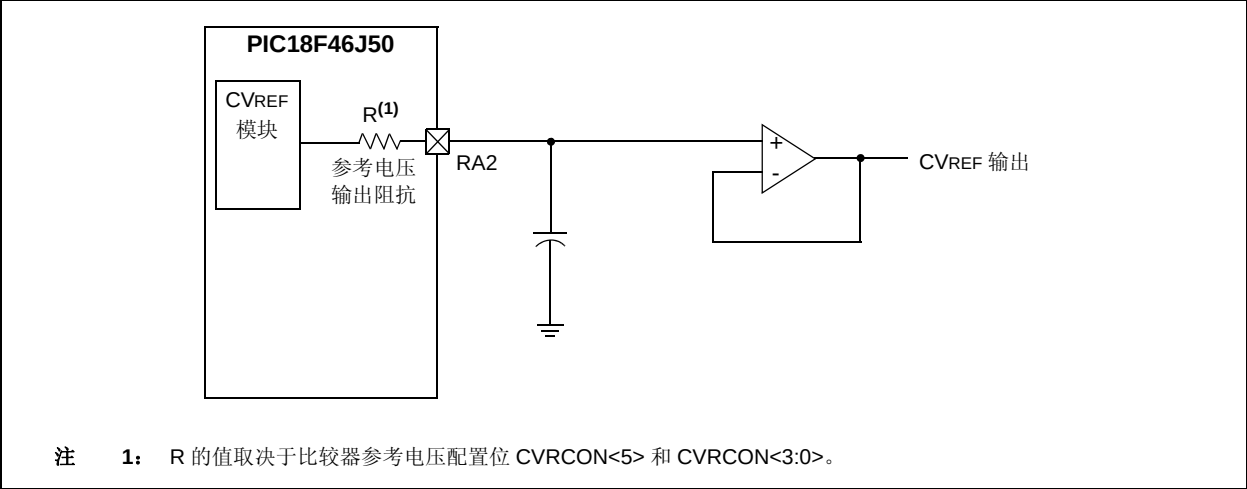


表 24-1: 与比较器参考电压相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
CVRCON	CVREN	CVROE	CVRR	r	CVR3	CVR2	CVR1	CVR0	74
CM1CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	70
CM2CON	CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0	70
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0	72
ANCON0	PCFG7 ⁽¹⁾	PCFG6 ⁽¹⁾	PCFG5 ⁽¹⁾	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	74
ANCON1	VBGEN	r	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	74

图注: — = 未实现（读为 0），r = 保留。比较器参考电压不使用阴影单元。

注 1: 这些位仅在 44 引脚器件上可用。

PIC18F46J50 系列

注:

25.0 高 / 低压检测（HLVD）

高 / 低电压检测（High/Low-Voltage Detect，HLVD）模块可用来监视 VDD 或 HLVDIN 引脚上的绝对电压。该模块是一个可编程的电路，它允许用户指定器件的电压跳变点和变化方向。

如果模块检测到电压按照指定的方向相对于该跳变点发生了偏离，就会将中断标志位置 1。如果允许了中断，程序将跳转到中断向量地址处执行，由软件响应该中断。

高 / 低压检测控制寄存器（寄存器 25-1）完全控制 HLVD 模块的工作。用户可通过软件控制该寄存器将电路“关闭”，从而使器件的电流消耗降至最低。

图 25-1 给出了 HLVD 模块的框图。

寄存器 25-1: HLVDCON: 高 / 低压检测控制寄存器（位于快速操作存储区，地址 F85h）

R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VDIRMAG	BGVST	IRVST	HLVDEN	HLVDL3 ⁽¹⁾	HLVDL2 ⁽¹⁾	HLVDL1 ⁽¹⁾	HLVDL0 ⁽¹⁾
bit 7							bit 0

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

bit 7	VDIRMAG: 电压方向大小选择位 1 = 当电压等于或超过跳变点（HLVDL<3:0>）时，事件发生 0 = 当电压等于或低于跳变点（HLVDL<3:0>）时，事件发生
bit 6	BGVST: 带隙参考电压稳定状态标志位 1 = 表示内部带隙参考电压稳定 0 = 表示内部带隙参考电压不稳定
bit 5	IRVST: 内部参考电压稳定标志位 1 = 表示电压检测逻辑在检测到指定的电压范围时，产生中断标志 0 = 表示电压检测逻辑在检测到指定的电压范围时，不会产生中断标志，并且 HLVD 中断不被允许
bit 4	HLVDEN: 高 / 低压检测模块使能位 1 = 使能 HLVD 0 = 禁止 HLVD
bit 3-0	HLVDL<3:0>: 电压检测限制位 ⁽¹⁾ 1111 = 使用外部模拟输入（输入来自于 HLVDIN 引脚） 1110 = 最大设置 . . . 1000 = 最小设置 0xxx = 保留

注 1: 具体规范请参见第 30.0 节“电气特性”中的表 30-8。

通过将 HLVDEN 位置 1 使能该模块。每次使能该模块时，电路需要一定时间才能稳定下来。IRVST 位是一个只读位，用于指示电路何时稳定。仅当电路稳定且 IRVST 位置 1 后，该模块才能产生中断。

VDIRMAG 位决定该模块的整体工作状态。当 VDIRMAG 清零时，模块监视 VDD 看它是否降到预先确定的设置点以下。当该位置 1 时，模块监视 VDD 看它是否上升到设置点以上。

PIC18F46J50 系列

25.1 工作原理

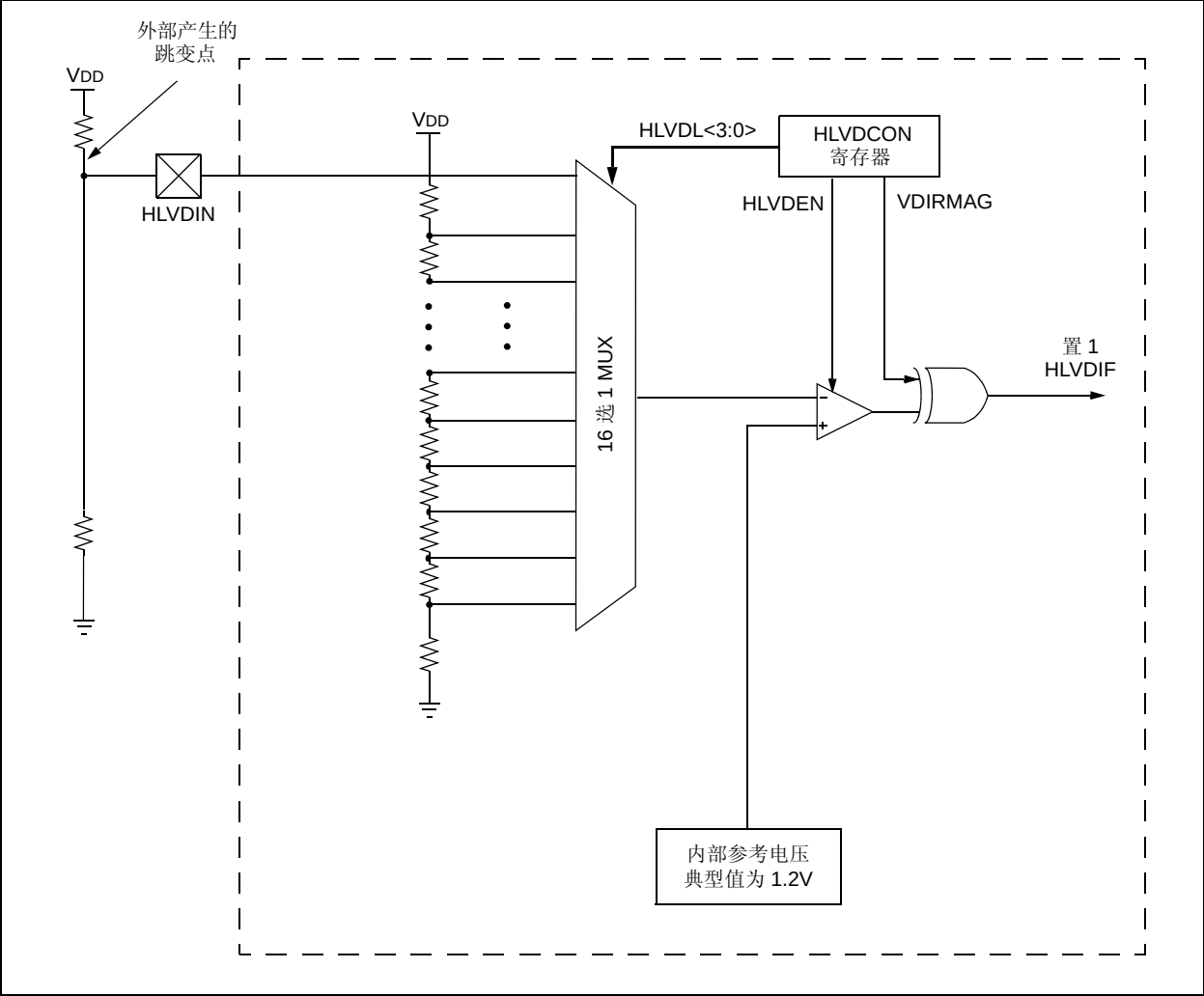
当使能了 HLVD 模块时，比较器使用内部产生的参考电压作为设置点。将设置点的电压与跳变点电压作比较，其中电阻分压器中的每个节点均代表一个电压跳变点。“跳变点”电压是被检测到的高压或低压事件的电压值，它取决于该模块的配置。

当供电电压等于跳变点电压时，电阻阵列的节点电压输出值等于由参考电压模块产生的内部参考电压。然后比较器通过将 HLVDIF 位置 1 产生一个中断信号。

可用软件编程设定跳变点电压为 8 个值中的任何一个。通过对 HLVDL<3:0> 位（HLVDCON<3:0>）进行编程选择跳变点。

此外，HLVD 模块允许用户通过外部电源向模块提供跳变电压。当 HLVDL<3:0> 位被设置为 1111 时，使能该模式。在此状态下，比较器输入与外部输入引脚 HLVDIN 复用。因此用户可以灵活地配置 HLVD 中断，使之可以在有效工作电压范围内的任何电压点上产生。

图 25-1: HLVD 模块框图（带外部输入）



25.2 HLVD 设置

设置 HLVD 模块：

1. 通过清零 HLVDEN 位（HLVDCON<4>），禁止该模块。
2. 将值写入 HLVDL<3:0> 位，选择所需的 HLVD 跳变点。
3. 设置 VDIRMAG 位以检测如下情况：
 - 高压（VDIRMAG = 1）
 - 低压（VDIRMAG = 0）
4. 通过将 HLVDEN 位置 1，使能 HLVD 模块。
5. 清零 HLVD 中断标志位 HLVDIF（PIR2<2>），该位可能被上次中断置 1。
6. 如果需要中断，通过将 HLVDIE 和 GIE/GIEH 位（PIE2<2> 和 INTCON<7>）置 1，允许 HLVD 中断。

直到 IRVST 位也置 1 时才会发生中断。

25.3 电流消耗

使能了该模块就使能了 HLVD 比较器和分压器，并将消耗静态电流。电气规范中的参数 D022B (ΔI_{HLVD})（[第 30.2 节“直流特性 PIC18F46J50 系列的掉电电流和供电电流（工业级）”](#)）给出了使能该模块时的电流总消耗。

HLVD 模块无需一直工作，工作与否取决于具体的应用。要降低电流消耗，只需要在检测电压时，短时间地使能 HLVD 电路，在检测完成之后马上禁止 HLVD 模块。

25.4 HLVD 启动时间

电气规范中的参数 D420（见[第 30.0 节“电气特性”](#)中的[表 30-8](#)）规定了 HLVD 模块的内部参考电压，该参考电压也可供其他内部电路（如可编程欠压复位（BOR）电路）使用。

如果禁止了 HLVD 或其他使用参考电压的电路以降低器件的电流消耗，则参考电压电路将需要一段时间稳定下来以后才能可靠地检测低压或高压条件。HLVD 启动时间 T_{IRVST} 与器件时钟速度无关。它由电气规范中的参数 36（[表 30-13](#)）规定。

直到 T_{IRVST} 结束并且参考电压达到稳定后才会允许 HLVD 中断标志。基于此原因，在此时间间隔期间，超出设置点的短暂偏离可能不会被检测到。请参见[图 25-2](#)或[图 25-3](#)。

图 25-2: 低压检测工作原理 (VDIRMAG = 0)

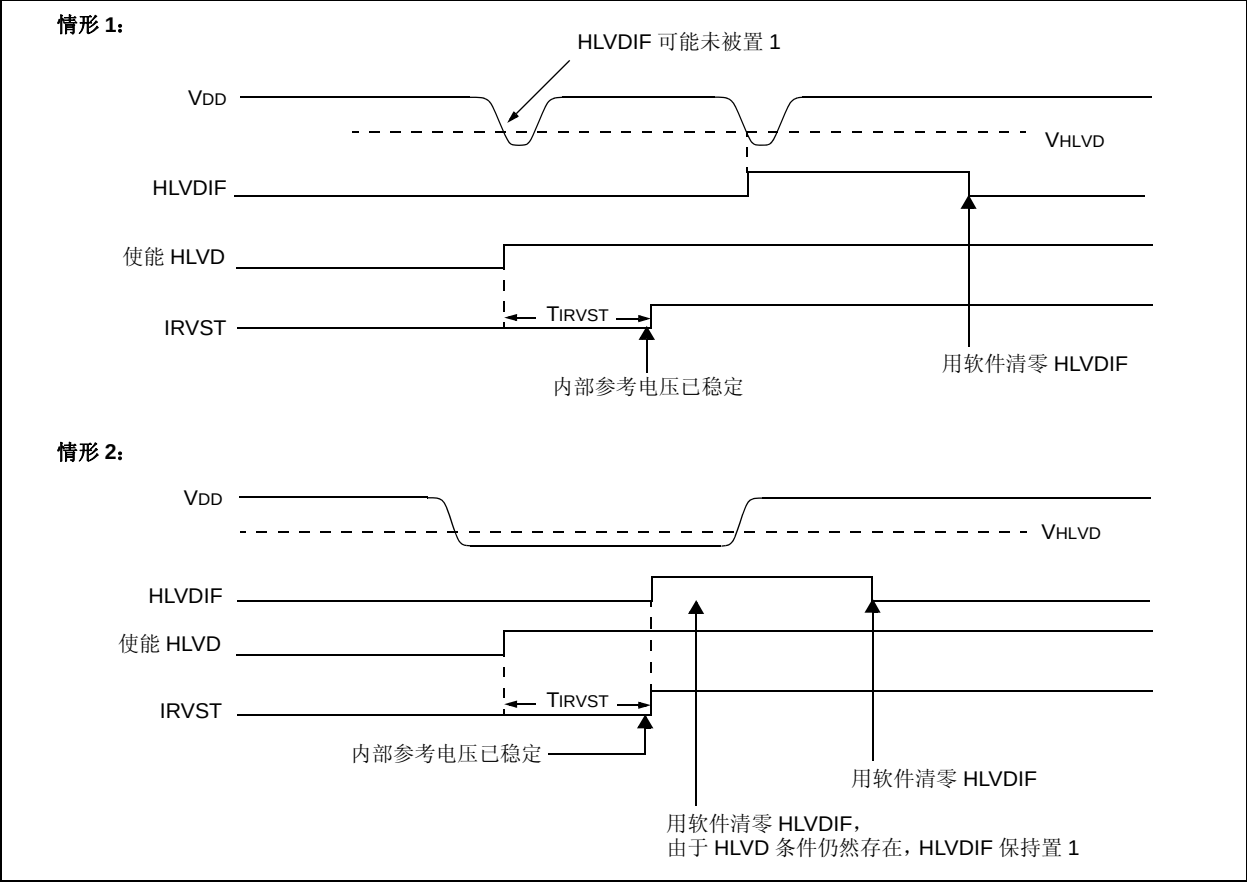
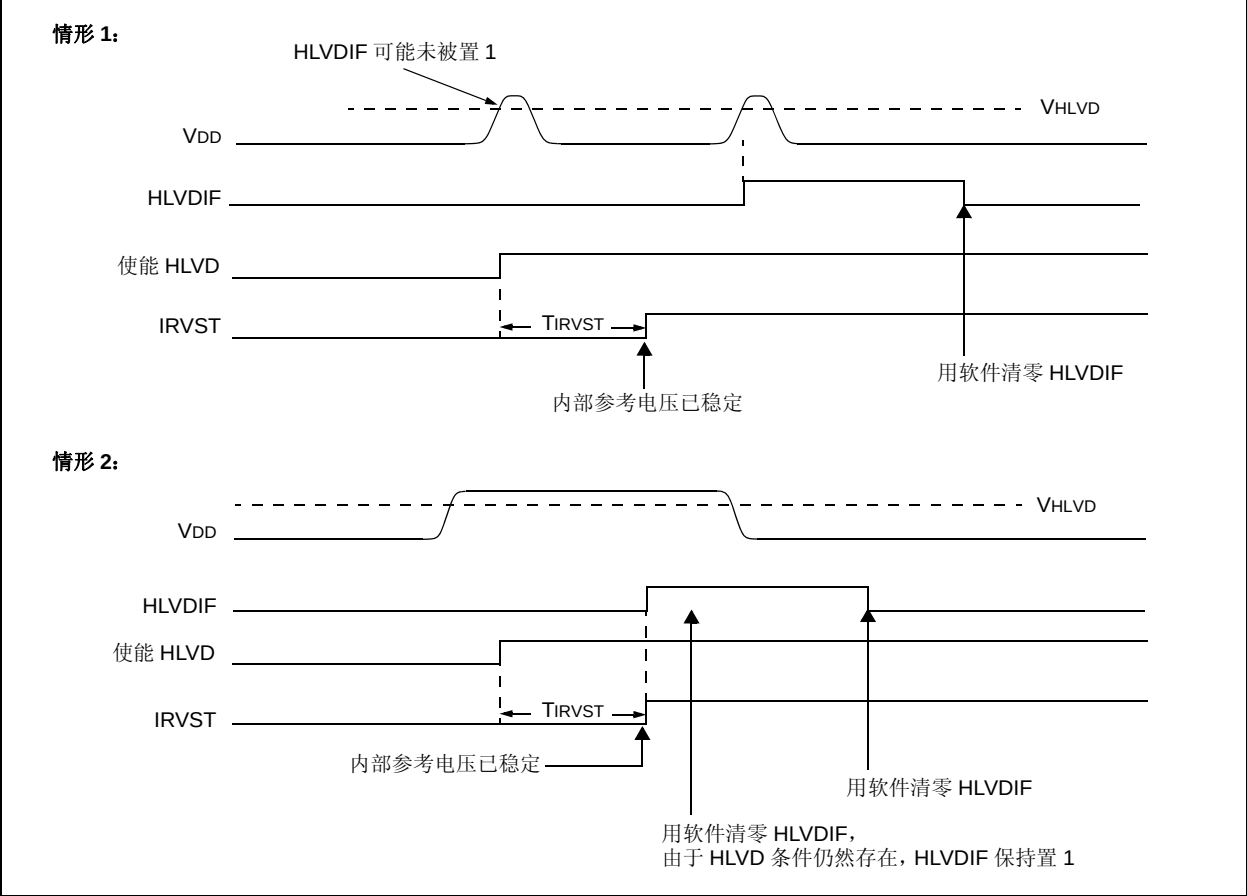


图 25-3: 高压检测工作原理 (VDIRMAG = 1)



25.5 应用

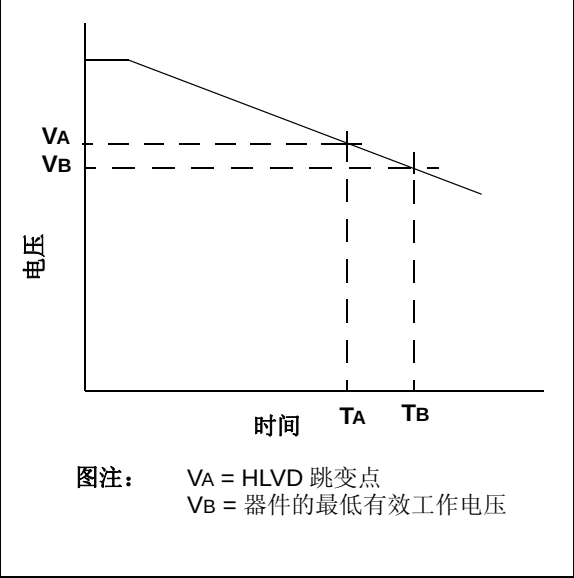
在许多应用中, 当电压低于或高于某个门限值时, 系统希望可以检测到这种事件。例如, 可以定期使能 HLVD 模块来检测是否连接通用串行总线 (USB)。

对于一般的电池应用, 图 25-4 给出了一个可能的电压曲线。

器件电压会随时间逐渐下降。当器件电压达到电压 V_A 时, HLVD 逻辑电路会在时间 T_A 产生中断。该中断将导致执行中断服务程序, 从而使应用程序能在器件电压退出有效工作范围 (对应的时间为 T_B) 之前执行“日常任务”, 并执行受控关闭。

因此, HLVD 将会提供一个时间窗 (表示为 T_A 和 T_B 的时间差) 使应用程序能安全地退出。

图 25-4: 典型高 / 低压检测应用



PIC18F46J50 系列

25.6 休眠期间的操作

如果使能了 HLVD 电路，则其在休眠期间将继续工作。如果器件电压越过了跳变点，HLVDIF 位将会被置 1 并且器件将从休眠状态中被唤醒。如果已经允许了全局中断，程序将跳转到中断向量地址处继续执行。

25.7 复位的影响

器件复位将强制所有寄存器为复位状态。这会强制关闭 HLVD 模块。

表 25-1: 与高 / 低压检测模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
HLVDCON	VDIRMAG	BGVST	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	72
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	69
PIR2	OSCFIF	CM2IF	CM1IF	USBIF	BCL1IF	HLVDIF	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CM2IE	CM1IE	USBIE	BCL1IE	HLVDIE	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CM2IP	CM1IP	USBIP	BCL1IP	HLVDIP	TMR3IP	CCP2IP	71

图注： — = 未实现，读为 0。HLVD 模块不使用阴影单元。

26.0 充电时间测量单元 (CTMU)

充电时间测量单元 (CTMU) 是一个灵活的模拟模块，它提供脉冲源之间的精确时间差测量，以及异步脉冲生成功能。CTMU 可与其他片上模拟模块配合使用，精确测量时间、电容、电容的相对变化，或生成具有特定延时的输出脉冲。CTMU 是与电容式传感器接口的理想选择。

该模块具有以下主要特性：

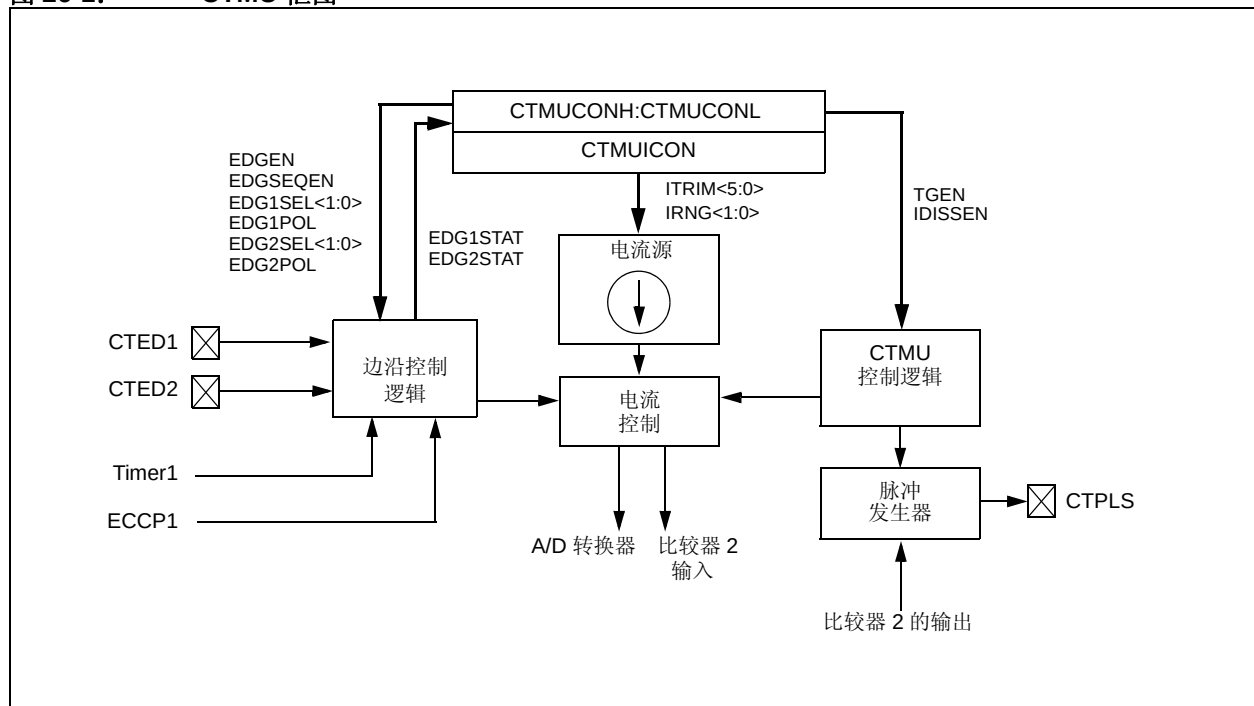
- 最多 13 路通道，可用于电容或时间测量输入
- 片上高精度电流源
- 4 个边沿输入触发源
- 每个边沿源的极性控制

- 边沿序列控制
- 控制对边沿的响应
- 1 纳秒的时间测量分辨率
- 高精度时间测量
- 与系统时钟异步的外部或内部信号的延时
- 适合测量电容的精确电流源

CTMU 与 A/D 转换器配合工作，根据具体器件和可用的 A/D 通道数，最多可提供 13 路通道用于时间或电荷测量。如果配置为产生延时，那么 CTMU 连接到其中一个模拟比较器。电平敏感输入边沿源可以从 4 个源中选择：两个外部输入、Timer1 或输出比较模块 1。

图 26-1 给出了 CTMU 的框图。

图 26-1: CTMU 框图



PIC18F46J50 系列

26.1 CTMU 工作原理

CTMU 的工作方式是使用固定电流源来对电路进行充电。电路的类型取决于要进行的测量的类型。在进行电荷测量的情况下，电流是固定的，向电路施加电流的时间也是固定的。这样，只要通过 A/D 测得电压就可以测得电路的电容。在进行时间测量的情况下，电流和电路的电容都是固定的。这种情况下，由 A/D 读取的电压可以代表从电流源开始对电路进行充电到停止充电经过的时间。

如果 CTMU 用于产生延时，那么电容和电流源，以及向比较器电路提供的电压都是固定的。信号的延时由将电压充电到比较器门限电压所需的时间决定。

26.1.1 工作原理

CTMU 的工作原理基于以下电荷公式：

$$I = C \cdot \frac{dV}{dT}$$

简单来说，在电路中测量的电荷（Q，以库仑为单位）定义为：以安培为单位的电流（I）乘以以秒为单位的电流流动时间（t）。电荷也可以定义为：以法拉为单位的电容（C）乘以电路的电压（V）。可得：

$$I \cdot t = C \cdot V.$$

CTMU 模块提供了恒定、已知的电流源。A/D 转换器用于测量公式中的电压（V），剩下两个未知量：电容（C）和时间（t）。以上公式可用于计算电容或时间，根据以下关系，使用电路的已知固定电容：

$$t = (C \cdot V) / I$$

或根据：

$$C = (I \cdot t) / V$$

使用电流源施加于电路的固定时间。

26.1.2 电流源

CTMU 的核心是高精度电流源，旨在提供用于测量的恒定基准。用户可以从三个范围或总共两个数量级的电流中选择电流等级，并可以按 $\pm 2\%$ 的增量（标称值）对输出进行微调。电流范围通过 IRNG<1:0> 位（CTMUICON<1:0>）进行选择，值 01 代表最低范围。

电流微调通过 ITRIM<5:0> 位（CTMUICON<7:2>）进行。这 6 个位使得可以按大约每步 2% 的步阶微调电流源。请注意，其中一半的范围用于正向调整电流源，另一半用于负向调整电流源。值 000000 是中性位置（无变化）。值 100001 代表最大负调整（大约 -62%），011111 代表最大正调整（大约 +62%）。

26.1.3 边沿选择和控制

CTMU 测量通过在模块的两路输入通道中发生的边沿事件进行控制。每路通道（称为边沿 1 和边沿 2）可以配置为接收来自一个边沿输入引脚（CTED1 和 CTED2）、Timer1 或输出比较模块 1 的输入脉冲。输入通道是电平敏感通道，响应通道中的瞬时电平，而不是电平跳变。输入使用 EDG1SEL 和 EDG2SEL 位对（CTMUCONL<3:2 和 6:5>）选择。

除了电流源之外，还可以使用 EDGE2POL 和 EDGE1POL 位（CTMUCONL<7,4>）配置每路通道的事件极性。还可以对输入通道进行过滤以选择边沿事件顺序（边沿 1 在边沿 2 之前发生），方法是将 EDGSEQEN 位（CTMUCONH<2>）置 1。

26.1.4 边沿状态

CTMUCONL 寄存器还包含两个状态位 EDG2STAT 和 EDG1STAT（CTMUCONL<1:0>）。它们的主要功能是显示在相应的通道中是否发生了边沿响应。当在通道中检测到边沿响应时，CTMU 会自动将特定的位置 1。输入通道的电平敏感特性也意味着，如果通道的配置或其电流状态发生改变，那么状态位会立即置 1。

模块使用边沿状态位来控制到外部模拟模块（如 A/D 转换器）的电流源输出。只有其中一个状态位置 1 而不是两个状态位同时置 1 时，才会向外部模块提供电流，如果两个位同时置 1 或同时清零，则会切断电流。这使 CTMU 可以仅测量两个边沿事件之间的电流。在两个状态位都置 1 后，必须先将它们清零，然后才能进行另一次测量。两个位应同时清零（如果可能），以避免重新使能 CTMU 电流源。

除了可以由 CTMU 硬件置 1 之外，边沿状态位也可以用软件置 1。也就是说可以在用户应用程序中手动使能或禁止电流源。将其中任意一位置 1（但不是同时置 1）即可使能电流源。将两位同时置 1 或清零即可立即禁止电流源。

26.1.5 中断

每当电流源先使能，然后禁止时，CTMU 就会将其中断标志位 (PIR3<2>) 置 1。只有相应的中断允许位 (PIE3<2>) 也置 1 时，才会产生中断。如果未使能边沿顺序 (即，边沿 1 必须在边沿 2 之前发生)，则需要监视边沿状态位，确定上次发生并导致中断的是哪一个边沿事件。

26.2 CTMU 模块初始化

以下过程是用于初始化 CTMU 模块的通用指南：

1. 使用 IRNG 位 (CTMUICON<1:0>) 选择电流源范围。
2. 使用 ITRIM 位 (CTMUICON<7:2>) 微调电流源。
3. 通过设置 EDG1SEL 和 EDG2SEL 位 (CTMUCONL<3:2 和 6:5>) 配置边沿 1 和边沿 2 的边沿输入源。
4. 使用 EDG1POL 和 EDG2POL 位 (CTMUCONL<4,7>) 配置边沿输入的输入极性。默认配置是使用负边沿极性 (从高至低跳变)。
5. 使用 EDGSEQEN 位 (CTMUCONH<2>) 使能边沿序列。默认情况下，将禁止边沿顺序。
6. 使用 TGEN 位 (CTMUCONH<4>) 选择工作模式 (测量或产生延时)。默认模式是时间 / 电容测量。
7. 通过将 IDISSEN 位 (CTMUCONH<1>) 置 1，对所连接电路放电；在等待足够时间，让电路完成放电之后，清零 IDISSEN。
8. 通过清零 CTMUEN 位 (CTMUCONH<7>) 禁止该模块。
9. 通过将 CTMUEN 位置 1 使能该模块。
10. 清零边沿状态位：EDG2STAT 和 EDG1STAT (CTMUCONL<1:0>)。如果可能的话，应同时清零这两个位，以避免重新使能 CTMU 电流源。
11. 通过将 EDGEN 位 (CTMUCONH<3>) 置 1 使能两个边沿输入。

根据要执行的测量或脉冲生成的类型，可能还需要再初始化和配置一个或更多其他模块，与 CTMU 模块配合使用：

- 边沿源生成：除了外部边沿输入引脚之外，Timer1 和输出比较 /PWM1 模块也可以用作 CTMU 的边沿源。
- 电容或时间测量：CTMU 模块使用 A/D 转换器来测量连接到一路模拟输入通道的电容两端的电压。
- 脉冲生成：在生成独立于系统时钟的输出脉冲时，CTMU 模块使用比较器 2 和关联的比较器参考电压。

26.3 校准 CTMU 模块

要精确测量电容和时间，以及产生精确延时，需要对 CTMU 进行校准。如果应用只需要测量电容或时间的相对变化，则通常不需要校准。此类应用的示例包括容性触摸开关，在这种应用中，触摸电路具有基本电容，所增加的人体电容会改变电路的总电容。

如果需要测量实际的电容或时间，则必须进行两项硬件校准：电流源需要进行校准，以使其提供精确的电流；要测量的电路也需要进行校准，以测量和 / 或抵消要测量电容之外的所有其他电容。

26.3.1 电流源校准

CTMU 模块随附的电流源具有三种电流范围，其中每种范围都可以在其标称值 $\pm 62\%$ 的范围内进行调节。因此，要进行精确测量，可以通过在未用模拟通道上放置一个高精度电阻 R_{CAL} ，测量并调整该电流源。图 26-2 给出了示例电路。电流源测量使用以下步骤执行：

1. 初始化 A/D 转换器。
2. 初始化 CTMU。
3. 通过将 EDG1STAT (CTMUCONL<0>) 置 1 使能电流源。
4. 启动一段延时，在这段时间内 R_{CAL} 两端的电压变为稳定，并且 ADC 采样 / 保持电容充电。
5. 执行 A/D 转换。
6. 使用 $I = V/R_{CAL}$ 计算有效的源电流；其中， R_{CAL} 是高精度电阻， V 通过执行 A/D 转换来测量。

PIC18F46J50 系列

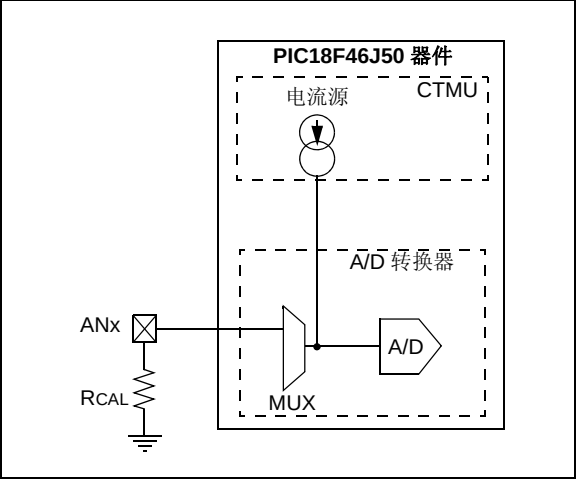
CTMU 电流源可以使用 CTMUICON 中的微调位进行微调，通过迭代过程来获取所需的精确电流。或者，也可以使用未经调整的标称值；可以由软件存储调整后的电流值，用于所有后续的电容或时间测量。

要计算 R_{CAL} 的最佳值，必须选择标称电流。例如，如果 A/D 转换器参考电压为 3.3V，使用满量程的 70%（或 2.31V）作为要由 A/D 转换器读取的所需近似电压。如果 CTMU 电流源的范围选择为 0.55 μA ，所需的电阻值使用 $R_{CAL} = 2.31V / 0.55 \mu A$ 计算，得到值为 4.2 M Ω 。类似地，如果电流源选择为 5.5 μA ， R_{CAL} 将为 420,000 Ω ；如果电流源设置为 55 μA ，则为 42,000 Ω 。

选择满量程电压 70% 的值，以确保 A/D 转换器处于充分高于基底噪声的范围。请记住，如果选择了某个需要结合使用 CTMUICON 的微调位的精确电流，则可能需要对 R_{CAL} 的电阻值进行相应调整。可能还需要再次调整 R_{CAL} ，以允许选择可用的电阻值。考虑将使用 CTMU 进行测量的电路所需的精度， R_{CAL} 应选择可用的最高精度。建议最小精度是允许 0.1% 的误差。

以下示例给出了执行 CTMU 电流校准的一种典型方法。[例 26-1](#) 演示如何初始化 A/D 转换器和 CTMU；该程序是同时使用两个模块的应用的典型程序。[例 26-2](#) 演示了实际校准程序的一种方法。

图 26-2: CTMU 电流源校准电路



例 26-1: CTMU 校准设置程序

```
#include <p18cxxx.h>
/*****
/*Setup CTMU *****/
/*****
void setup(void)

{ //CTMUCON - CTMU Control register

    CTMUCONH = 0x00;           //make sure CTMU is disabled
    CTMUCONL = 0x90;
    //CTMU continues to run when emulator is stopped,CTMU continues
    //to run in idle mode,Time Generation mode disabled, Edges are blocked
    //No edge sequence order, Analog current source not grounded
    //Edge2 polarity = positive level, Edge2 source =
    //source 0, Edge1 polarity = positive level, Edge1 source = source 0,

    //CTMUICON - CTMU Current Control Register
    CTMUICON = 0x01;           //0.55uA, Nominal - No Adjustment

/*****
//Setup AD converter;
/*****

    TRISA=0x04;                //set channel 2 as an input

    // Configured AN2 as an analog channel
    // ANCON0
    ANCON0 = 0xFB;
    // ANCON1
    ANCON1 = 0x1F;

    // ADCON1
    ADCON1bits.ADFM=1;          // Result format 1= Right justified
    ADCON1bits.ADCAL=0;         // Normal A/D conversion operation
    ADCON1bits.ACQT=1;          // Acquisition time 7 = 20TAD 2 = 4TAD 1=2TAD
    ADCON1bits.ADCS=2;          // Clock conversion bits 6= FOSC/64 2=FOSC/32

    ANCON1bits.VBGEN=1;         // Turn on the Bandgap (if not already on)

    // ADCON0
    ADCON0bits.VCFG0 =0;        // Vref+ = AVdd
    ADCON0bits.VCFG1 =0;        // Vref- = AVss
    ADCON0bits.CHS=2;           // Select ADC channel

    ADCON0bits.ADON=1;          // Turn on ADC

}
```

PIC18F46J50 系列

例 26-2: 电流校准程序

```
#include <p18cxxx.h>

#define COUNT 500 // @ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define RCAL .027 // R value is 4200000 (4.2M)
// scaled so that result is in
// 1/100th of uA

#define ADSCALE 1023 // for unsigned conversion 10 sig bits
#define ADREF 3.3 // Vdd connected to A/D Vr+

void main(void)
{
    int i;
    int j = 0; // index for loop
    unsigned int Vread = 0;
    double VTot = 0;
    float Vavg=0, Vcal=0, CTMUISrc = 0; // float values stored for calcs

    // assume CTMU and A/D have been setup correctly
    // see Example 25-1 for CTMU & A/D setup
    setup();

    CTMUCONHbits.CTMUEN = 1; // Enable the CTMU
    CTMUCONLbits.EDG1STAT = 0; // Set Edge status bits to zero
    CTMUCONLbits.EDG2STAT = 0;
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1; // drain charge on the circuit
        DELAY; // wait 125us
        CTMUCONHbits.IDISSEN = 0; // end drain of circuit

        CTMUCONLbits.EDG1STAT = 1; // Begin charging the circuit
        // using CTMU current source
        DELAY; // wait for 125us
        CTMUCONLbits.EDG1STAT = 0; // Stop charging circuit

        PIR1bits.ADIF = 0; // make sure A/D Int not set
        ADCON0bits.GO=1; // and begin A/D conv.
        while(!PIR1bits.ADIF); // Wait for A/D convert complete

        Vread = ADRES; // Get the value from the A/D
        PIR1bits.ADIF = 0; // Clear A/D Interrupt Flag
        VTot += Vread; // Add the reading to the total
    }

    Vavg = (float)(VTot/10.000); // Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL; // CTMUISrc is in 1/100ths of uA
}
```

26.3.2 电容校准

内部 A/D 转换器采样电容和电路板走线与焊盘的杂散电容虽然容值较小，但仍会影响电容测量的精度。在确保先取下期望测量的电容的情况下，可以对杂散电容进行测量。然后，测量使用以下步骤执行：

- 1. 初始化 A/D 转换器和 CTMU。
- 2. 将 EDG1STAT 置 1 (= 1)。
- 3. 等待固定延时 t 。
- 4. 清零 EDG1STAT。
- 5. 执行 A/D 转换。
- 6. 计算杂散电容和 A/D 采样电容：

$$C_{\text{OFFSET}} = C_{\text{STRAY}} + C_{\text{AD}} = (I \cdot t) / V$$

其中， I 从电流源测量步骤获知， t 是固定延时， V 通过执行 A/D 转换来测量。

然后，可以存储该测量值，用于时间测量时的计算，或在电容测量时减去该值。要进行校准，需要大致了解 $C_{\text{STRAY}} + C_{\text{AD}}$ 的电容值。 C_{AD} 约为 4 pF。

可能需要使用一个迭代过程来调整时间 t ，该时间是对电路进行充电，以从 A/D 转换器获得合理电压读数的时间。 t 的值可以通过将 C_{OFFSET} 设置为理论值，然后求解 t 来确定。例如，如果 C_{STRAY} 的理论计算值为 11 pF， V 预期为 V_{DD} 的 70%（或 2.31V），那么 t 为：

$$(4 \text{ pF} + 11 \text{ pF}) \cdot 2.31\text{V} / 0.55 \text{ mA}$$

或 63 μs 。

请参见例 26-3 了解 CTMU 电容校准的典型程序。

PIC18F46J50 系列

例 26-3: 电容校准程序

```
#include <p18cxxx.h>

#define COUNT 25 // @ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5 // time in uS
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023 // for unsigned conversion 10 sig bits
#define ADREF 3.3 // Vdd connected to A/D Vr+
#define RCAL .027 // R value is 4200000 (4.2M)
// scaled so that result is in
// 1/100th of uA

void main(void)
{
    int i;
    int j = 0; // index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

    // assume CTMU and A/D have been setup correctly
    // see Example 25-1 for CTMU & A/D setup
    setup();

    CTMUCONHbits.CTMUEN = 1; // Enable the CTMU
    CTMUCONLbits.EDG1STAT = 0; // Set Edge status bits to zero
    CTMUCONLbits.EDG2STAT = 0;
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1; // drain charge on the circuit
        DELAY; // wait 125us
        CTMUCONHbits.IDISSEN = 0; // end drain of circuit

        CTMUCONLbits.EDG1STAT = 1; // Begin charging the circuit
        // using CTMU current source
        DELAY; // wait for 125us
        CTMUCONLbits.EDG1STAT = 0; // Stop charging circuit

        PIR1bits.ADIF = 0; // make sure A/D Int not set
        ADCON0bits.GO=1; // and begin A/D conv.
        while(!PIR1bits.ADIF); // Wait for A/D convert complete

        Vread = ADRES; // Get the value from the A/D
        PIR1bits.ADIF = 0; // Clear A/D Interrupt Flag
        VTot += Vread; // Add the reading to the total
    }

    Vavg = (float)(VTot/10.000); // Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL; // CTMUISrc is in 1/100ths of uA
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
}
```

26.4 使用 CTMU 测量电容

使用 CTMU 测量电容有两种相对独立的方法。第一种是绝对方法，该方法需要测量实际电容值。第二种是相对方法，该方法不需要实际电容，只需要电容的变化量。

26.4.1 绝对电容测量

对于绝对电容测量，应遵循第 26.3 节“校准 CTMU 模块”中的电流和电容校准步骤。然后，电容测量使用以下步骤执行：

1. 初始化 A/D 转换器。
2. 初始化 CTMU。
3. 将 EDG1STAT 置 1。
4. 等待固定延时 T 。
5. 清零 EDG1STAT。
6. 执行 A/D 转换。
7. 计算总电容 $C_{TOTAL} = (I * T)/V$ ；其中， I 从电流源测量步骤（见第 26.3.1 节“电流源校准”）获知， T 是固定延时， V 通过执行 A/D 转换来测量。
8. 从 C_{TOTAL} 中减去杂散电容和 A/D 采样电容（COFFSET 来自第 26.3.2 节“电容校准”），确定被测电容的值。

26.4.2 相对电荷测量

有些应用可能并不需要精确的电容测量。例如，在检测电容式开关的有效按压时，只需要检测电容的相对变化。在此类应用中，当开关打开（未被触摸）时，总电容是电路板走线和 A/D 转换器等组合电容。此时 A/D 转换器将会测量到较大的电压。当开关关闭（被触摸）时，由于以上所列电容中增加了人体的电容，总电容增大，A/D 转换器将测量到较小的电压。

使用 CTMU 检测电容变化可以使用以下步骤简单实现：

1. 初始化 A/D 转换器和 CTMU。
2. 将 EDG1STAT 置 1。
3. 等待固定延时。
4. 清零 EDG1STAT。
5. 执行 A/D 转换。

通过执行 A/D 转换测量的电压可以指示相对电容。请注意，在这种情况下，不需要对电流源或电路电容测量进行校准。请参见例 26-4 了解容性触摸开关的软件程序示例。

PIC18F46J50 系列

例 26-4: 用于容性触摸开关的程序

```
#include <p18cxxx.h>

#define COUNT 500                //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define OPENSW 1000              //Un-pressed switch value
#define TRIP 300                 //Difference between pressed
                                //and un-pressed switch
#define HYST 65                  //amount to change
                                //from pressed to un-pressed

#define PRESSED 1
#define UNPRESSED 0

void main(void)
{
    unsigned int Vread;           //storage for reading
    unsigned int switchState;
    int i;

    //assume CTMU and A/D have been setup correctly
    //see Example 25-1 for CTMU & A/D setup
    setup();

    CTMUCONHbits.CTMUEN = 1;      // Enable the CTMU
    CTMUCONLbits.EDG1STAT = 0;    // Set Edge status bits to zero
    CTMUCONLbits.EDG2STAT = 0;

    CTMUCONHbits.IDISSEN = 1;     //drain charge on the circuit
    DELAY;                        //wait 125us
    CTMUCONHbits.IDISSEN = 0;     //end drain of circuit

    CTMUCONLbits.EDG1STAT = 1;    //Begin charging the circuit
    //using CTMU current source
    DELAY;                        //wait for 125us
    CTMUCONLbits.EDG1STAT = 0;    //Stop charging circuit

    PIR1bits.ADIF = 0;            //make sure A/D Int not set
    ADCON0bits.GO=1;              //and begin A/D conv.
    while(!PIR1bits.ADIF);        //Wait for A/D convert complete

    Vread = ADRES;                //Get the value from the A/D

    if(Vread < OPENSW - TRIP)
    {
        switchState = PRESSED;
    }
    else if(Vread > OPENSW - TRIP + HYST)
    {
        switchState = UNPRESSED;
    }
}
```

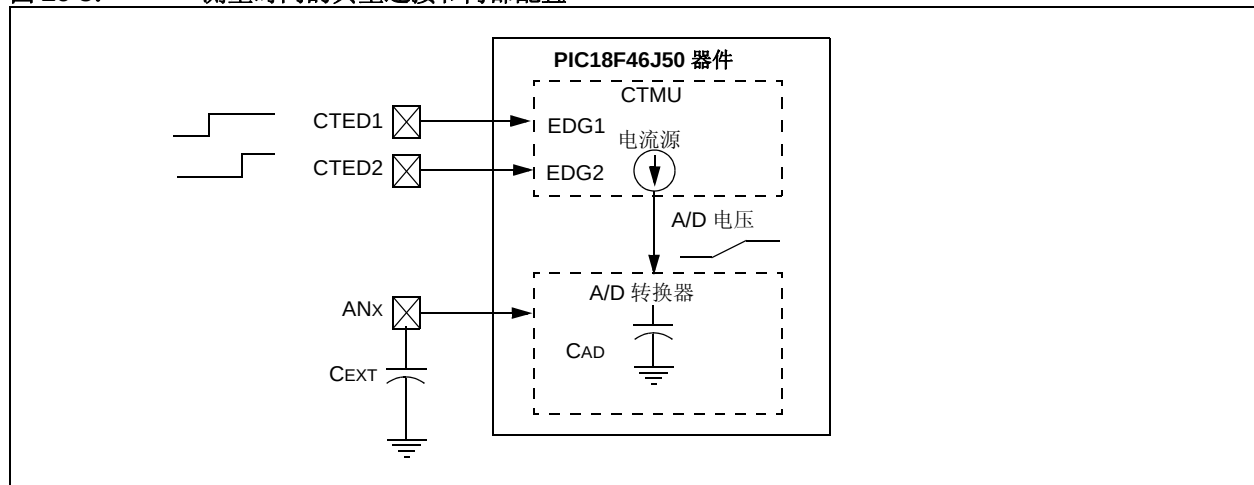
26.5 使用 CTMU 模块测量时间

通过电流和电容校准步骤测量比率 (C/I) 之后，可以使用以下步骤精确测量时间：

1. 初始化 A/D 转换器和 CTMU。
2. 将 EDG1STAT 置 1。
3. 将 EDG2STAT 置 1。
4. 执行 A/D 转换。
5. 根据 $T = (C/I) * V$ 计算边沿之间的时间；其中， I 在电流校准步骤（第 26.3.1 节“电流源校准”）中计算， C 在电容校准步骤（第 26.3.2 节“电容校准”）中计算， V 通过执行 A/D 转换来测量。

假定所测量的时间足够小，电容 $C_{AD} + C_{EXT}$ 可以向 A/D 转换器提供有效的电压。要进行最小的时间测量，请始终将 A/D 通道选择寄存器 (AD1CHS) 设置为未用的 A/D 通道；该通道的相应引脚不连接到任何电路板走线。这可以最大程度减小所增加的杂散电容，保持总电路电容接近于 A/D 转换器自身的电容。要测量较长的时间间隔，可以将一个外部电容连接到 A/D 通道，并在进行时间测量时选择该通道。

图 26-3: 测量时间的典型连接和内部配置



PIC18F46J50 系列

26.6 使用 CTMU 模块产生延时

CTMU 模块具有一种独特功能，即它可以根据外部电容值产生独立于系统时钟的输出脉冲。这通过使用内部比较器参考电压模块、比较器 2 输入引脚和外部电容实现。脉冲输出到 CTPLS 引脚上。要启用该模式，需将 TGEN 位置 1。

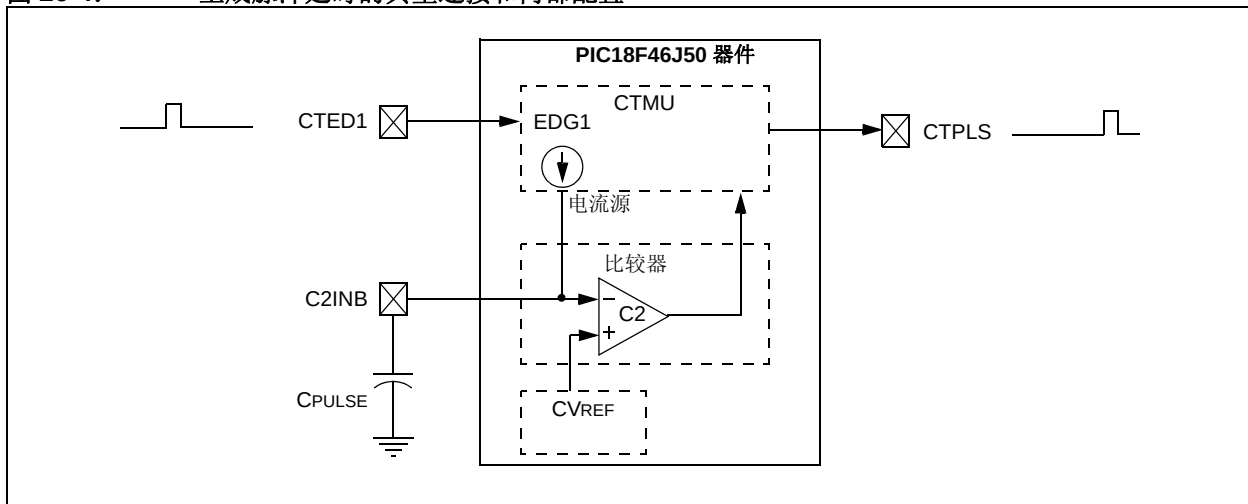
示例电路请参见图 26-4。CPULSE 由用户选择，用于确定 CTPLS 上的输出脉冲宽度。脉冲宽度根据 $T = (CPULSE/I) * V$ 计算；其中， I 从电流源测量步骤（第 26.3.1 节“电流源校准”）获知， V 是内部参考电压（CVREF）。

该功能的使用示例是与基于可变电容的传感器进行接口，例如湿度传感器。当湿度发生变化时，CTPLS 上的脉宽输出也会变化。CTPLS 输出引脚可以连接到输入捕捉引脚，通过测量变化的脉冲宽度来确定应用环境的湿度。

执行以下步骤来使用该功能：

1. 初始化比较器 2（CPOL = 1）。
2. 初始化比较器参考电压。
3. 初始化 CTMU，并通过将 TGEN 位置 1 来使能延时生成。
4. 将 EDG1STAT 置 1。
5. 当 CPULSE 充电到参考电压跳变点的值时，在 CTPLS 上会产生输出脉冲。

图 26-4: 生成脉冲延时的典型连接和内部配置



26.7 休眠 / 空闲模式期间的操作

26.7.1 休眠模式和深度休眠模式

当器件进入休眠模式时，CTMU 模块电流源将始终禁止。如果调用休眠模式时，CTMU 正在执行依赖于电流源的操作，则操作可能不会正确终止。电容和时间测量可能会返回错误值。

26.7.2 空闲模式

CTMU 在空闲模式下的行为由 CTMUSIDL 位（CTMUCONH<5>）决定。如果 CTMUSIDL 清零，在空闲模式下，模块将继续工作。如果 CTMUSIDL 置 1，则在器件进入空闲模式时，模块的电流源会被禁止。如

果调用空闲模式时，模块正在执行操作，这种情况下，结果将类似于休眠模式下的结果。

26.8 复位对 CTMU 的影响

在复位时，CTMU 的所有寄存器都会被清零。这使 CTMU 模块处于禁止状态，它的电流源被关闭，所有配置选项恢复为它们的默认设置。在任意复位之后，模块都需要重新初始化。

如果发生复位时，CTMU 正在进行测量，测量结果将丢失。正在测量的电路可能会存在部分充电的情况，在随后 CTMU 尝试进行测量之前，应正确进行放电。电路放电方法是，在 A/D 转换器连接到相应通道的同时，先将 IDISSEN 位（CTMUCONH<1>）置 1，然后再将其清零。

26.9 寄存器

有 3 个用于 CTMU 的控制寄存器：

- CTMUCONH
- CTMUCONL
- CTMUICON

CTMUCONH 和 CTMUCONL 寄存器（[寄存器 26-1](#) 和 [寄存器 26-2](#)）包含一些控制位，这些控制位用于配置 CTMU 模块边沿源选择、边沿源极性选择、边沿顺序、A/D 触发、模拟电路电容放电和使能。CTMUICON 寄存器（[寄存器 26-3](#)）包含一些用于选择电流源范围和电流源微调的位。

寄存器 26-1: CTMUCONH: CTMU 控制寄存器高字节（位于快速操作存储区，地址 FB3h）

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	r
bit 7							bit 0

图注：	r = 保留位		
R = 可读位	W = 可写位		U = 未实现位，读为 0
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	CTMUEN: CTMU 使能位 1 = 使能模块 0 = 禁止模块
bit 6	未实现: 读为 0
bit 5	CTMUSIDL: 空闲模式停止位 1 = 当器件进入空闲模式时，模块停止工作 0 = 在空闲模式下模块继续工作
bit 4	TGEN: 时间生成使能位 1 = 使能边沿延时生成 0 = 禁止边沿延时生成
bit 3	EDGEN: 边沿使能位 1 = 不阻止边沿 0 = 阻止边沿
bit 2	EDGSEQEN: 边沿顺序使能位 1 = 边沿 1 事件必须在边沿 2 事件发生前发生 0 = 无需边沿顺序
bit 1	IDISSEN: 模拟电流源控制位 1 = 模拟电流源输出接地 0 = 模拟电流源输出不接地
bit 0	保留: 写为 0

PIC18F46J50 系列

寄存器 26-2: CTMUCONL: CTMU 控制寄存器低字节（位于快速操作存储区，地址 FB2h）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x	R/W-x
EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 7							bit 0

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

bit 7	EDG2POL: 边沿 2 极性选择位 1 = 边沿 2 设定为正边沿响应 0 = 边沿 2 设定为负边沿响应
bit 6-5	EDG2SEL<1:0>: 边沿 2 源选择位 11 = CTED1 引脚 10 = CTED2 引脚 01 = ECCP1 输出比较模块 00 = Timer1 模块
bit 4	EDG1POL: 边沿 1 极性选择位 1 = 边沿 1 设定为正边沿响应 0 = 边沿 1 设定为负边沿响应
bit 3-2	EDG1SEL<1:0>: 边沿 1 源选择位 11 = CTED1 引脚 10 = CTED2 引脚 01 = ECCP1 输出比较模块 00 = Timer1 模块
bit 1	EDG2STAT: 边沿 2 状态位 1 = 已发生边沿 2 事件 0 = 未发生边沿 2 事件
bit 0	EDG1STAT: 边沿 1 状态位 1 = 已发生边沿 1 事件 0 = 未发生边沿 1 事件

寄存器 26-3: CTMUICON: CTMU 电流控制寄存器（位于快速操作存储区，地址 FB1h）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 7						bit 0	

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

bit 7-2

ITRIM<5:0>: 电流源微调位

011111 = 对标称电流的最大正向调整

011110

.

.

.

000001 = 对标称电流的最小正向调整

000000 = IRNG<1:0> 指定的标称电流输出

111111 = 对标称电流的最小负向调整

.

.

.

100010

100001 = 对标称电流的最大负向调整

bit 1-0

IRNG<1:0>: 电流源范围选择位

11 = 100 × 基本电流

10 = 10 × 基本电流

01 = 基本电流等级（标称值为 0.55 μA）

00 = 禁止电流源

表 26-1: 与 CTMU 模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页:
CTMUCONH	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	r	71
CTMUCONL	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	71
CTMUICON	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	71

图注: — = 未实现，读为 0。r = 保留位。CTMU 模块不使用阴影单元。

PIC18F46J50 系列

注:

27.0 CPU 的特殊功能

PIC18F46J50 系列器件包含的功能旨在最大限度地提高系统可靠性，并通过减少外部元件把系统成本降到最低。这些功能包括：

- 振荡器选择
- 复位：
 - 上电复位（POR）
 - 上电延时定时器（PWRT）
 - 振荡器起振定时器（OST）
 - 欠压复位（BOR）
- 中断
- 看门狗定时器（WDT）
- 故障保护时钟监视器（FSCM）
- 双速启动
- 代码保护
- 在线串行编程（ICSP）

要根据具体应用对频率、功耗、精度和成本的要求来选择振荡器。在[第 3.0 节“振荡器配置”](#)中详细讨论了所有的选项。

在本数据手册的前面几章中已完整地讨论了器件的复位和中断。除了为复位提供了上电延时定时器和振荡器起振定时器之外，PIC18F46J50 系列器件还有一个可配置的看门狗定时器（WDT），该定时器由软件控制。

器件自带的内部 RC 振荡器还提供了故障保护时钟监视器（FSCM）和双速启动这两个额外的功能。FSCM 对外设时钟进行后台监视，并在外设时钟发生故障时自动切换时钟源。双速启动使得几乎可在启动发生那一刻立即执行代码，同时主时钟源进行其起振延时。

通过设置相应的配置寄存器位可以使能和配置所有这些功能。

27.1 配置位

可以通过对配置位编程来选择各种器件配置。配置数据存储在闪存程序存储器的最后 4 个字中；[图 6-1](#) 对此进行了说明。配置数据装入到易失性配置寄存器 CONFIG1L 至 CONFIG4H 中，这些寄存器是可读寄存器，它们映射到程序存储器中从 300000h 开始的单元。

[表 27-2](#) 给出了完整的列表。从[寄存器 27-1](#)到[寄存器 27-6](#)详细说明了各配置位的功能。

27.1.1 配置 PIC18F46J50 系列器件的注意事项

与某些早期的 PIC18 单片机不同，PIC18F46J50 系列器件不再使用持久存储寄存器存储配置信息。配置寄存器 CONFIG1L 至 CONFIG4H 实现为易失性存储器。

在上电或器件复位之后，单片机硬件会立即自动向 CONFIG1L 至 CONFIG4L 寄存器中装入存储在非易失性闪存程序存储器中的配置数据。闪存程序存储器的最后 4 个字（称为闪存配置字，FCW）用于存储配置数据。

[表 27-1](#) 列出了这些闪存程序存储单元，其内容将装入相应的配置寄存器。

当为这些器件创建应用程序时，用户应该总是为配置数据特别分配 FCW 地址，以确保当编译代码时程序代码不会存储在该地址中。

对应于 CONFIG1H、CONFIG2H、CONFIG3H 和 CONFIG4H 的 FCW 的高 4 位（MSb）应始终设定为 1111。这使得在意外执行这些 FCW 单元的极少事件中，这些 FCW 执行为 NOP 指令。

CONFIG1H、CONFIG2H、CONFIG3H 和 CONFIG4H 寄存器的高 4 位未实现，所以向它们相应的 FCW 写入 1 对器件操作没有任何影响。

为了防止在代码执行期间，配置发生意外改变，配置寄存器 CONFIG1L 至 CONFIG4L 仅在上电或复位时装入一次。用户的固件仍然可以使用自行再编程来修改 FCW 的内容，通过这种方式来更改配置。

如果修改了 FCW，只有器件复位之后，在 CONFIG1L 至 CONFIG4H 寄存器中使用的有效内容才会更改。

PIC18F46J50 系列

表 27-1: 闪存配置字到配置寄存器的映射

配置寄存器（易失性）	配置寄存器地址	闪存配置字节地址
CONFIG1L	300000h	XXXF8h
CONFIG1H	300001h	XXXF9h
CONFIG2L	300002h	XXXFAh
CONFIG2H	300003h	XXXFBh
CONFIG3L	300004h	XXXFCh
CONFIG3H	300005h	XXXFDh
CONFIG4L	300006h	XXXFEh
CONFIG4H	300007h	XXXFFh

表 27-2: 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 / 未编程值 ⁽¹⁾
300000h	CONFIG1L	DEBUG	XINST	STVREN	—	PLLDIV2	PLLDIV1	PLLDIV0	WDTEN	111- 1111
300001h	CONFIG1H	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	—	CP0	CPDIV1	CPDIV0	1111 -111
300002h	CONFIG2L	IESO	FCMEN	—	LPT1OSC	T1DIG	FOSC2	FOSC1	FOSC0	11-1 1111
300003h	CONFIG2H	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	WDTPS3	WDTPS2	WDTPS1	WDTPS0	1111 1111
300004h	CONFIG3L	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0	DSWDTEN	DSBOREN	RTCOSC	DSWDTOSC	1111 1111
300005h	CONFIG3H	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	MSSPMSK	—	—	IOL1WAY	1111 1--1
300006h	CONFIG4L	WPCFG	WPEND	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0	1111 1111
300007h	CONFIG4H	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	— ⁽²⁾	—	—	—	WPDIS	1111 ---1
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxx0 0000 ⁽³⁾
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0100 00xx ⁽³⁾

图注: x = 未知, u = 不变, - = 未实现。阴影单元未实现, 读为 0。

注 1: 这些值反映出厂时和上电复位后的未编程状态。在所有其他复位状态中, 配置字节保持原先的编程状态。

2: 程序存储器中这些位的值应始终编程为 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: DEVID 值请参见寄存器 27-9 和寄存器 27-10。这些寄存器为只读寄存器, 用户不能对其进行编程。

寄存器 27-1: 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	R/WO-1	R/WO-1	R/WO-1	R/WO-1
DEBUG	XINST	STVREN	—	PLLDIV2	PLLDIV1	PLLDIV0	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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 7 **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
- 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 **未实现:** 读为 0
- bit 3-1 **PLLDIV<2:0>:** 振荡器选择位
 必须选择分频比以提供 4 MHz 输入给 96 MHz PLL。
 111 = 不分频 —— 直接使用振荡器 (4 MHz 输入)
 110 = 振荡器被 2 分频 (8 MHz 输入)
 101 = 振荡器被 3 分频 (12 MHz 输入)
 100 = 振荡器被 4 分频 (16 MHz 输入)
 011 = 振荡器被 5 分频 (20 MHz 输入)
 010 = 振荡器被 6 分频 (24 MHz 输入)
 001 = 振荡器被 10 分频 (40 MHz 输入)
 000 = 振荡器被 12 分频 (48 MHz 输入)
- bit 0 **WDTEN:** 看门狗定时器使能位
 1 = 使能 WDT
 0 = 禁止 WDT (由 SWDTEN 位控制)

PIC18F46J50 系列

寄存器 27-2: 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	R/WO-1	R/WO-1
—	—	—	—	—	CP0	CPDIV1	CPDIV0
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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7-4未实现: 将对应的闪存配置位编程为 1
- bit 3未实现: 保持为 0
- bit 2**CP0:** 代码保护位
1 = 程序存储器不受代码保护
0 = 程序存储器受代码保护
- bit 1-0**CPDIV<1:0>:** CPU 系统时钟选择位
11 = CPU 系统时钟不分频
10 = CPU 系统时钟被 2 分频
01 = CPU 系统时钟被 3 分频
00 = CPU 系统时钟被 6 分频

寄存器 27-3: 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	R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
IESO	FCMEN	—	LPT1OSC	T1DIG	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 = POR 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 **未实现:** 读为 0

bit 4 **LPT1OSC:** 低功耗 Timer1 振荡器使能位

1 = Timer1 振荡器配置为高功耗工作

0 = Timer1 振荡器配置为低功耗工作

bit 3 **T1DIG:** 辅助时钟源 T1OSCEN 强制位

1 = 无论 T1OSCEN (T1CON<3>) 的状态如何, 都可以选择辅助振荡器时钟源 (OSCCON<1:0> = 01)

0 = 除非 T1CON<3> = 1, 否则不能选择辅助振荡器时钟源

bit 2-0 **FOSC<2:0>:** 振荡器选择位

111 = 带软件控制 PLL 的 ECPLL 振荡器, RA6 用作 CLK0 引脚

110 = EC 振荡器, RA6 用作 CLK0 引脚

101 = 带软件控制 PLL 的 HSPLL 振荡器

100 = HS 振荡器

011 = INTOSCPLLO, 带软件控制 PLL 的内部振荡器, RA6 用作 CLK0 引脚, RA7 用作端口引脚

010 = INTOSCPLL, 带软件控制 PLL 的内部振荡器, RA6 和 RA7 均用作端口引脚

001 = INTOSCO 内部振荡器模块 (INTRC/INTOSC), RA6 用作 CLK0 引脚, RA7 用作端口引脚

000 = INTOSC 内部振荡器模块 (INTRC/INTOSC), RA6 和 RA7 均用作端口引脚

PIC18F46J50 系列

寄存器 27-4: 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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 7-4未实现: 将对应的闪存配置位编程为 1
- bit 3-0WDTPS<3:0>: 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 27-5: 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	R/WO-1	R/WO-1	R/WO-1
DSWDTPS3 ⁽¹⁾	DSWDTPS2 ⁽¹⁾	DSWDTPS1 ⁽¹⁾	DSWDTPS0 ⁽¹⁾	DSWDTEN ⁽¹⁾	DSBORN	RTCOSC	DSWDTOSC ⁽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 = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

bit 7-4 **DSWDTPS<3:0>**: 深度休眠看门狗定时器后分频比选择位 ⁽¹⁾

DSWDT 预分频比为 32。这产生一个约为 1 ms 的基本时间单位。

1111 = 1:2,147,483,648 (25.7 天)
 1110 = 1:536,870,912 (6.4 天)
 1101 = 1:134,217,728 (38.5 小时)
 1100 = 1:33,554,432 (9.6 小时)
 1011 = 1:8,388,608 (2.4 小时)
 1010 = 1:2,097,152 (36 分钟)
 1001 = 1:524,288 (9 分钟)
 1000 = 1:131,072 (135 秒)
 0111 = 1:32,768 (34 秒)
 0110 = 1:8,192 (8.5 秒)
 0101 = 1:2,048 (2.1 秒)
 0100 = 1:512 (528 ms)
 0011 = 1:128 (132 ms)
 0010 = 1:32 (33 ms)
 0001 = 1:8 (8.3 ms)
 0000 = 1:2 (2.1 ms)

bit 3 **DSWDTEN**: 深度休眠看门狗定时器使能位 ⁽¹⁾

1 = 使能 DSWDT
 0 = 禁止 DSWDT

bit 2 **DSBORN**: “F” 器件深度休眠 BOR 使能位, “LF” 器件 V_{DD} BOR 使能位

对于 “F” 器件:

1 = 在休眠模式下使能 V_{DD} BOR 检测
 0 = 始终禁止 V_{DD} BOR 检测电路

对于 “LF” 器件:

1 = 始终使能 V_{DD} BOR 检测电路
 0 = 始终禁止 V_{DD} BOR 检测电路

bit 1 **RTCOSC**: RTCC 参考时钟选择位

1 = RTCC 使用 T1OSC/T1CKI 作为参考时钟
 0 = RTCC 使用 INTRC 作为参考时钟

bit 0 **DSWDTOSC**: DSWDT 参考时钟选择位 ⁽¹⁾

1 = DSWDT 使用 INTRC 作为参考时钟
 0 = DSWDT 使用 T1OSC/T1CKI 作为参考时钟

注 1: “LF” 器件不具有这些功能。

PIC18F46J50 系列

寄存器 27-6: 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	U-0	U-0	R/WO-1
—	—	—	—	MSSPMSK	—	—	IOL1WAY
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 = POR 时的值 1 = 置 1 0 = 清零 x = 未知

- bit 7-4 **未实现:** 将对应的闪存配置位编程为 1
- bit 3 **MSSPMSK:** MSSP 7 位地址掩码模式使能位
1 = 使能 7 位地址掩码模式
0 = 使能 5 位地址掩码模式
- bit 2-1 **未实现:** 读为 0
- bit 0 **IOL1WAY:** IOLOCK 一次置 1 使能位
1 = IOLOCK 位 (PPSCON<0>) 可以置 1 一次, 前提是已经完成解锁序列。
置 1 后, 不能再次写入外设引脚选择寄存器。
0 = IOLOCK 位 (PPSCON<0>) 可以根据需要置 1 和清零, 前提是已经完成解锁序列

寄存器 27-7: CONFIG4L: 配置寄存器 4 的低字节 (字节地址为 300006h)

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
WPCFG	WPEND	WPFP5 ⁽²⁾	WPFP4 ⁽³⁾	WPFP3	WPFP2	WPFP1	WPFP0
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 = POR 时的值 1 = 置 1 0 = 清零 x = 未知

- bit 7 **WPCFG:** 配置字擦 / 写保护区域选择位
1 = 除非 WPEND 和 WPFP<5:0> 设置保护配置字页 (且 WPDIS = 0), 否则不对配置字页进行擦 / 写保护 ⁽¹⁾
0 = 无论 WPDIS、WPEND 和 WPFP<5:0> 位的设置如何, 均对配置字页执行擦 / 写保护 ⁽¹⁾
- bit 6 **WPEND:** 写 / 擦除保护区域选择位 (当 WPDIS = 0 时有效)
1 = 闪存页 WPFP<5:0> 至配置字页受到擦 / 写保护
0 = 闪存页 0 至 WPFP<5:0> 受到擦 / 写保护
- bit 5-0 **WPFP<5:0>:** 写 / 擦除保护页开始 / 结束单元位
与 WPEND 位配合确定闪存中的哪些页将受到擦 / 写保护。

- 注 1: “配置字页”中包含 FCW, 并且是在给定器件上实现的闪存的最后一页。每页包含 1,024 个字节。例如, 在具有 64 KB 闪存的器件上, 第一页是 0, 最后一页 (配置字页) 是 63 (3Fh)。
2: 在 64 KB 器件 (PIC18FX6J50) 器件中实现了该位。在 32 KB 和 16 KB 器件 (PIC18FX5J50 和 PIC18FX4J50) 中, 该位是保留的, 应始终被编程为 0, 以使器件正常工作。
3: 在 64 KB 和 32 KB 器件中实现了该位。在 16 KB 器件 (PIC18FX4J50) 中, 该位是保留的, 应始终被编程为 0, 以使器件正常工作。

寄存器 27-8: CONFIG4H: 配置寄存器 4 的高字节 (字节地址为 300007h)

U-1	U-1	U-1	U-1	U-0	U-0	U-0	R/WO-1
—	—	—	—	—	—	—	WPDIS
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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-4 **未实现:** 将对应的闪存配置位编程为 1

bit 3-1 **未实现:** 读为 0

bit 0 **WPDIS:** 写保护禁止位

1 = WPPFP<5:0> 和 WPEND 位被忽略; 选定的区域不受擦 / 写保护

0 = 使能 WPPFP<5:0> 和 WPEND 位; 擦 / 写保护对于选定区域有效

寄存器 27-9: DEVID1: PIC18F46J50 系列器件的器件 ID 寄存器 1 (字节地址为 3FFFFEh)

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 = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-5 **DEV<2:0>:** 器件 ID 位
这些位与器件 ID 寄存器 2 中的 DEV<10:3> 位一起用于标识部件编号。请参见[寄存器 27-10](#)。

bit 4-0 **REV<4:0>:** 版本 ID 位
这些位用于指明器件版本。

PIC18F46J50 系列

寄存器 27-10: DEVID2: PIC18F46J50 系列器件的器件 ID 寄存器 2（字节地址为 3FFFFFh）

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 = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 **DEV<10:3>**: 器件 ID 位
这些位与器件 ID 寄存器 1 中的 DEV<2:0> 位一起用于标识部件编号。

DEV<10:3> (DEVID2<7:0>)	DEV<2:0> (DEVID2<7:5>)	器件
0100 1100	101	PIC18F46J50
0100 1100	100	PIC18F45J50
0100 1100	011	PIC18F44J50
0100 1100	010	PIC18F26J50
0100 1100	001	PIC18F25J50
0100 1100	000	PIC18F24J50
0100 1101	011	PIC18LF46J50
0100 1101	010	PIC18LF45J50
0100 1101	001	PIC18LF44J50
0100 1101	000	PIC18LF26J50
0100 1100	111	PIC18LF25J50
0100 1100	110	PIC18LF24J50

27.2 看门狗定时器（WDT）

PIC18F46J50 系列器件同时具有传统的 WDT 电路和支持深度休眠的专用看门狗定时器。当使能时，传统的 WDT 将在正常的运行、空闲和休眠模式下工作。本数据手册小节将描述传统的 WDT 电路。

支持深度休眠的专用 WDT 只能在深度休眠模式下使能。此定时器在 [第 4.6.4 节“深度休眠看门狗定时器 \(DSWDT\)”](#) 中描述。

传统的 WDT 是由 INTRC 振荡器驱动的。当使能 WDT 时，时钟源也将被使能。WDT 周期的标称值为 4 ms，其稳定性与 INTRC 振荡器相同。

4 ms 的 WDT 周期被 16 位后分频器分频来得到更长的时间周期。通过配置寄存器 2H 中的 WDTPS 位来控制一个多路开关以对 WDT 后分频器的输出进行选择。可获得的定时周期范围为 4 ms 至 135 秒（2.25 分钟，取决于电压、温度和 WDT 后分频器）。当发生以下任一

事件时，WDT 和后分频器将被清零，这些事件包括：执行了 SLEEP 或 CLRWDT 指令，或者发生了时钟故障（主时钟或 Timer1 振荡器）。

- 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:

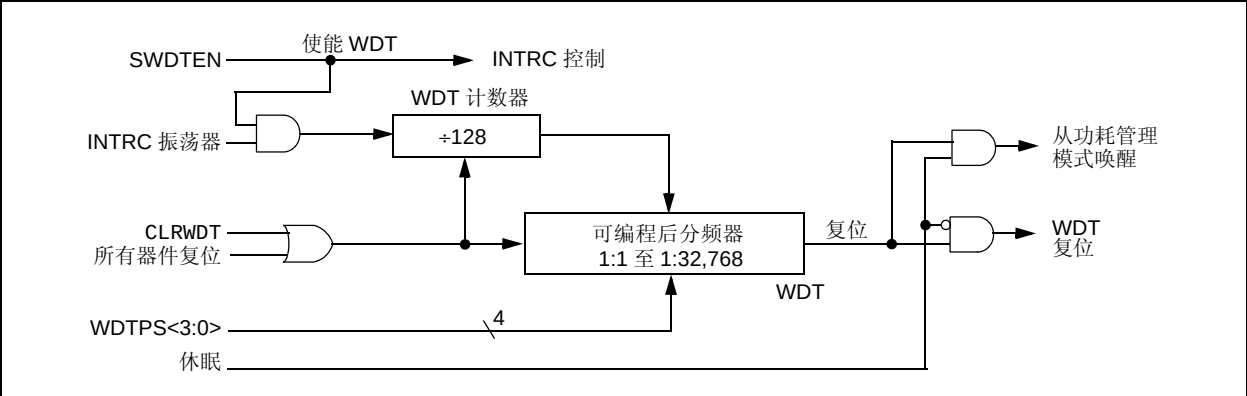
当执行 CLRWDT 指令时，后分频器的计数值将被清零。

27.2.1 控制寄存器

WDTCON 寄存器（[寄存器 27-11](#)）是可读写寄存器。SWDTEN 位使能或禁止 WDT 操作。仅当 WDT 被配置位禁止时，才允许使用软件改写 WDTEN 配置位并使能 WDT。

LVDSTAT 是只读状态位，它会不断更新，提供关于 VDDCORE 当前值的信息。只有使能片上稳压器时，该位才有效。

图 27-1: WDT 框图



PIC18F46J50 系列

寄存器 27-11: WDTCON: 看门狗定时器控制寄存器 (位于快速操作存储区, 地址 FC0h)

R/W-1	R-x	R-x	U-0	R-q	R/W-0	R/W-0	R/W-0
REGSLP	LVDSTAT ⁽²⁾	ULPLVL	—	DS	ULPEN	ULPSINK	SWDTEN ⁽¹⁾
bit 7							bit 0

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为 0	q = 视情况而定
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	REGSLP: 稳压器低功耗工作使能位 1 = 当器件进入休眠模式时, 片上稳压器进入低功耗工作 0 = 片上稳压器即使在休眠模式下仍保持工作状态
bit 6	LVDSTAT: 低压检测状态位 1 = VDDCORE > 2.45V 标称值 0 = VDDCORE < 2.45V 标称值
bit 5	ULPLVL: 超低功耗唤醒输出位 (除非 ULPEN = 1 时才有效) 1 = RA0 上的电压 > ~0.5V 0 = RA0 上的电压 < ~0.5V
bit 4	未实现: 读为 0
bit 3	DS: 深度休眠唤醒状态位 (与 RCON、POR 和 BOR 位一起用于确定复位源) ⁽²⁾ 1 = 如果上次从复位退出是从深度休眠中正常唤醒导致的 0 = 如果上次从复位退出不是从深度休眠中唤醒导致的
bit 2	ULPEN: 超低功耗唤醒模块使能位 1 = 使能超低功耗唤醒模块; ULPLVL 位指示比较器输出 0 = 禁止超低功耗唤醒模块
bit 1	ULPSINK: 超低功耗唤醒灌电流使能位 1 = 使能超低功耗唤醒灌电流 0 = 禁止超低功耗唤醒灌电流
bit 0	SWDTEN: 软件控制的看门狗定时器使能位 ⁽¹⁾ 1 = 看门狗定时器开启 0 = 看门狗定时器关闭

- 注 1: 当使能 WDTEN 配置位时该位不起作用。
2: 对于禁止片上稳压器的器件, 该位不可用 (对于 “LF” 器件)。

表 27-3: 看门狗定时器寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页:
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	70
WDTCON	REGSLP	LVDSTAT	ULPLVL	—	DS	ULPEN	ULPSINK	SWDTEN	70

图注: — = 未实现, 读为 0。看门狗定时器不使用阴影单元。

27.3 片上稳压器

注 1: 只有在名称标有 “F” 的器件（例如，PIC18F25J50）中，才提供片上稳压器。在器件编号中含有 “LF” 的器件中，禁止片上稳压器。

2: 始终禁止将 VDDCORE/VCAP 引脚悬空。在 “F” 器件中，必须将该引脚通过一个称为 CEFC 的电容接地。在 “LF” 器件中，必须将 VDDCORE/VCAP 连接到介于 2.0V 与 2.7V 之间的电源。

PIC18F46J50 系列器件的数字内核逻辑基于先进的生产工艺设计，需要使用 2.0V 至 2.7V 的电压。数字内核逻辑通过 VDDCORE/VCAP 电源引脚供电。

但在许多应用中，I/O 引脚使用相同的内核逻辑电压运行可能并不方便，因为这会限制器件与其他较高电压器件（例如使用标称电压 3.3V 运行的器件）进行接口的能力。因此，所有 PIC18F46J50 系列器件都实现了双电源轨拓扑。VDDCORE/VCAP 引脚向内核逻辑供电，通过单片机的 VDD 引脚向 I/O 引脚供电，供电电压可在 2.15V 至 3.6V（“F” 器件）或 2.0V 至 3.6V（“LF” 器件）之间。

这种双电源拓扑使单片机可以与标准的 3.3V 逻辑器件进行接口，同时内核逻辑使用标称值为 2.5V 的较低电压运行。

为了使单片机更便于使用，在 PIC18F46J50 系列器件内的裸片上集成了 2.5V 低压差、低静态电流的线性稳压器。该稳压器专门设计为用于为器件的内核逻辑供电。它使 PIC18F46J50 系列器件实际上可以使用单电源轨工作，而无需使用外部稳压器。

“F” 器件上的片上稳压器总是使能的。VDDCORE/VCAP 引脚同时用作稳压器输出引脚和内核逻辑电源输入引脚。VDDCORE/VCAP 引脚与地之间应连接一个电容，用于确保稳压器的稳定性。关于 PIC18F 和 PIC18LF 器件的示例连接，请参见图 27-2。

在 “LF” 系列器件上，片上稳压器始终被禁止。这使器件可以节省少量的静态电流消耗，而这对于某些类型的应用（例如那些将完全使用标称电压 2.5V 运行的应用）是非常有利的。在 “LF” 系列器件上，VDDCORE/VCAP 引

脚仍然用作内核逻辑电源输入引脚，因而必须在应用电路板级别连接到 2.0V 至 2.7V 的电源轨。在这些器件上，仍然可以选择为 I/O 引脚提供 2.0V 至 3.6V 之间的电压，前提是 VDD 始终大于等于 VDDCORE/VCAP。关于 PIC18F 和 PIC18LF 器件的示例连接，请参见图 27-2。

注: 在名称标有 “LF” 的器件中（例如 PIC18LF46J50），VDDCORE 绝对不能超出 VDD。

第 30.3 节 “直流特性 PIC18F46J50 系列（工业级）” 中列出了内核电压和电容的规范。

27.3.1 稳压器跟踪模式和低压检测

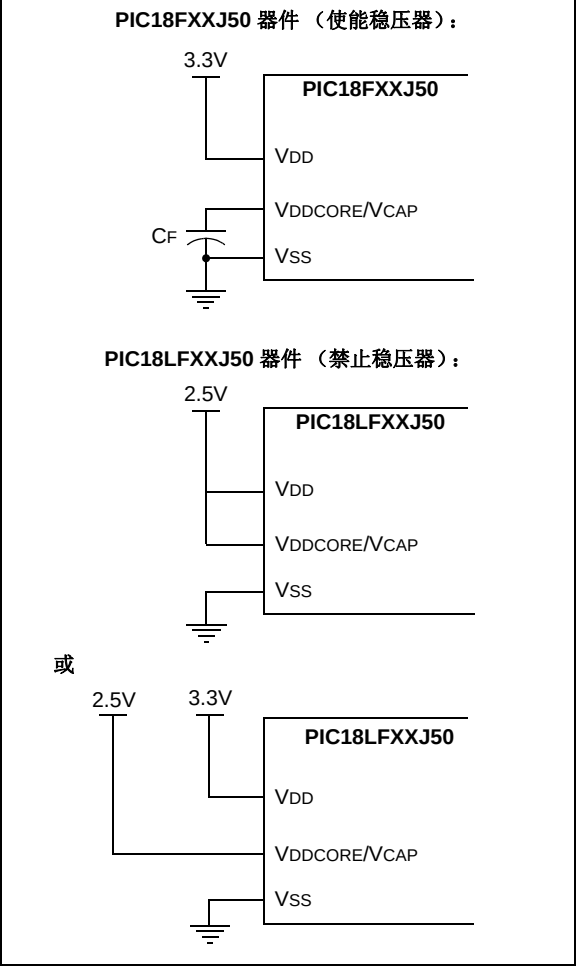
当片上稳压器被使能时，它可以向数字内核逻辑提供标称值为 2.5V 的恒压。稳压器可对大约 2.5V 直至器件 VDDMAX 的 VDD 进行稳压来提供此数字内核逻辑电压。VDD 电压降到 2.5V 以下时稳压器无法工作。当 VDD 输入电源电压降得太低，不足以提供 2.5V 稳压输出的情况下，稳压器进入跟踪模式。在跟踪模式下，稳压器输出跟随 VDD，有 100 mV 或更低的典型压降。

片上稳压器具有一个简单的低电压检测（Low-Voltage Detect, LVD）电路。该电路是一个独立的单元，与第 25.0 节 “高 / 低压检测（HLVD）” 中所述的高 / 低压检测（HLVD）模块无关。片上稳压器 LVD 电路持续监测 VDDCORE 电压，并据此更新 WDTCON 寄存器中的 LVDSTAT 位。设定的 LVD 检测门限值比片上稳压器的常规稳压设定点略低。

应用固件可以选择性地查询 LVDSTAT 位以确定何时可以安全地在最大额定频率下运行，而不违反图 30-1 提供的电压与频率关系要求。

仅在使能片上稳压器时，VDDCORE 监视 LVD 电路才能工作。在 “LF” 器件中，模数转换器和 HLVD 模块仍能向固件提供 VDD 和 VDDCORE 电压信息。

图 27-2: 片上稳压器连接



27.3.2 片上稳压器和 BOR

当使能片上稳压器时，PIC18F46J50 系列器件也会有一个简单的欠压保护功能。如果向稳压器提供的电压不足以维持最小输出电压，那么稳压器复位电路将产生欠压复位（BOR）。BOR 标志位（RCON<0>）会捕捉该事件。

第 5.4 节“欠压复位（BOR）”和第 5.4.1 节“检测 BOR”详细描述了 BOR 的工作原理。第 30.1 节“直流特性：PIC18F46J50 系列供电电压（工业级）”中指定了欠压电压值。

27.3.3 上电要求

片上稳压器是为了满足器件的上电要求而设计的。如果应用不使用稳压器，那就必须遵守严格的上电条件。在上电时，VDDCORE 超出 VDD 的幅度应小于 0.3V。

27.3.4 休眠模式下的操作

当片上稳压器使能时，它总是消耗比IDD多一点的电流。器件处于休眠模式时也是如此，即使内核数字逻辑功耗不高时。要在功率资源极其重要的应用中进一步节省功耗，可以将稳压器配置为每当器件进入休眠模式时就自动进入低静态电流待机模式。该功能由 REGSLP 位（寄存器 27-11 中的 WDTCON<7>）控制。如果在进入休眠模式时将该位置 1，稳压器将转换为较低功耗状态。在该状态下，稳压器仍然提供维持 SRAM 状态信息所需的稳定输出电压，但消耗的静态电流较低。

通过将 REGSLP 位置 1，在休眠模式下可以节省大量功耗，但器件的唤醒时间将增加，以确保稳压器有足够的时间进行稳定。

27.4 双速启动

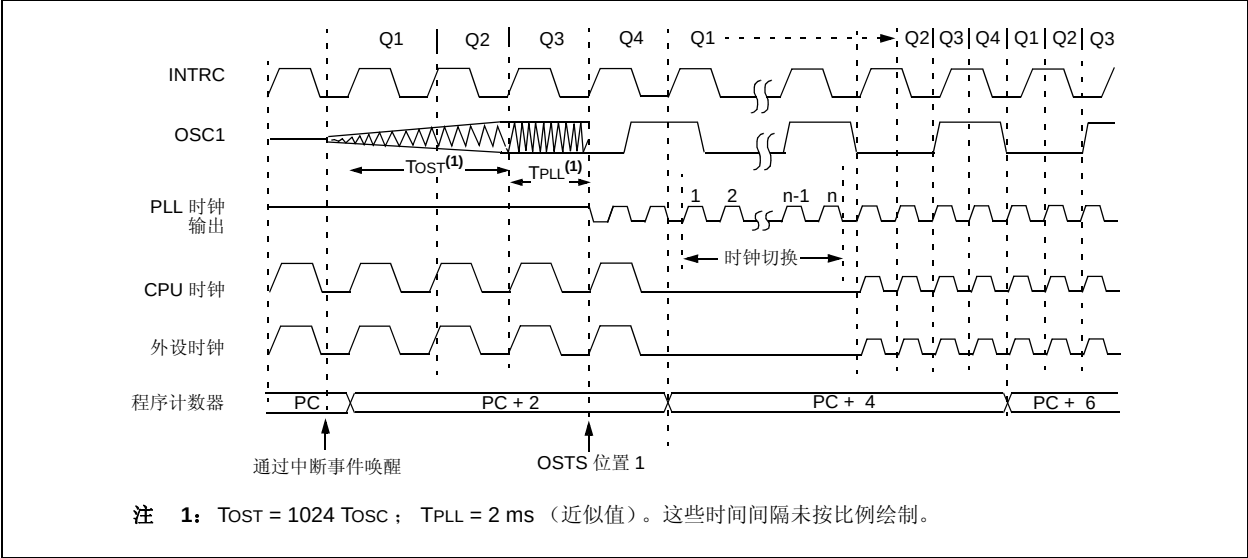
双速启动功能允许单片机在主时钟源稳定之前使用 INTRC 振荡器作为时钟源，从而帮助器件最大限度地缩短从振荡器起振到代码执行之间的延时。通过将 IESO 配置位置 1 可启用该功能。

仅当主振荡器模式为 HS 或 HSPLL（基于晶振）模式时才可使能双速启动。由于 EC 和 ECPLL 模式不需要振荡器起振定时器（OST）延时，因此应禁止双速启动。

一旦使能双速启动，当器件复位或从休眠模式唤醒时，在上电延时定时器发生超时（使能上电复位）后，器件将被配置为使用内部振荡器模块作为时钟源。这使得在主振荡器起振、OST 运行的同时，代码几乎立即开始执行。一旦 OST 超时，器件就自动切换到 PRI_RUN 模式。

在所有其他功耗管理模式下，不使用双速启动。器件将使用当前选定的时钟源直到主时钟源可用为止。IESO 位的设置被忽略。

图 27-3: 双速启动时钟切换的时序图（从 INTRC 切换到 HSPLL）



27.4.1 使用双速启动时的注意事项

当在双速启动模式中使用 INTRC 振荡器时，器件仍将遵守进入功耗管理模式的正常指令顺序，包括执行多条 SLEEP 指令（见第 4.1.4 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 $SCS<1:0>$ 位的设置或发出 SLEEP 指令。这就使应用程序能短暂地唤醒器件，执行“日常任务”，并在器件开始使用主时钟源前返回休眠状态。

用户代码还能通过检查 OSTS 位（OSCCON<3>）的状态来确定主时钟源是否正在为器件提供时钟。如果该位置 1，则表示主振荡器正在为器件提供时钟。否则，表示当器件复位或从休眠模式唤醒期间由内部振荡器模块为器件提供时钟。

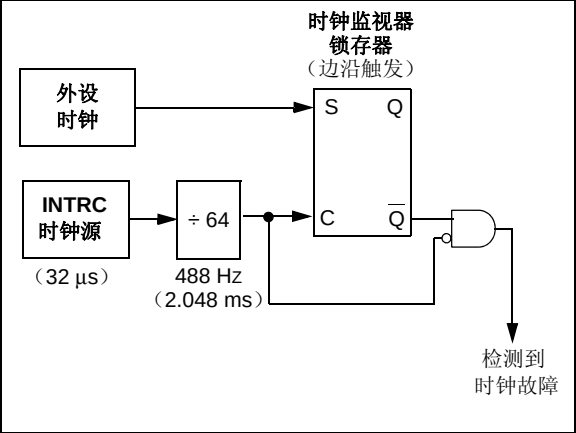
27.5 故障保护时钟监视器

故障保护时钟监视器（FSCM）可使单片机在外部振荡器发生故障时，自动将器件时钟切换到内部振荡器模块以保证器件能继续工作。将 FCMEN 配置位置 1 可启用 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 27-4 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边沿出现。外设时钟和采样时钟作为时钟监视锁存器的输入。时钟监视器在器件时钟源的下降沿被置 1，在采样时钟的上升沿被清零。

PIC18F46J50 系列

图 27-4: FSCM 框图



在采样时钟的下降沿检测时钟故障。如果在时钟监视器仍然设为使能时出现采样时钟下降沿，并检测到时钟故障（图 27-5），则会产生以下情况：

- 通过将 OSCFIF 位（PIR2<7>）置 1，由 FSCM 产生振荡器故障中断。
- 器件时钟源切换到内部振荡器模块（OSCCON 不会被更新，因此无法显示当前时钟源——这就是故障保护条件）。
- WDT 复位。

切换过程中，对于时序敏感的应用，内部振荡器模块的后分频器频率可能不够稳定。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关闭。更多详细信息，请参见第 4.1.4 节“多条 SLEEP 命令”和第 27.4.1 节“使用双速启动时的注意事项”。

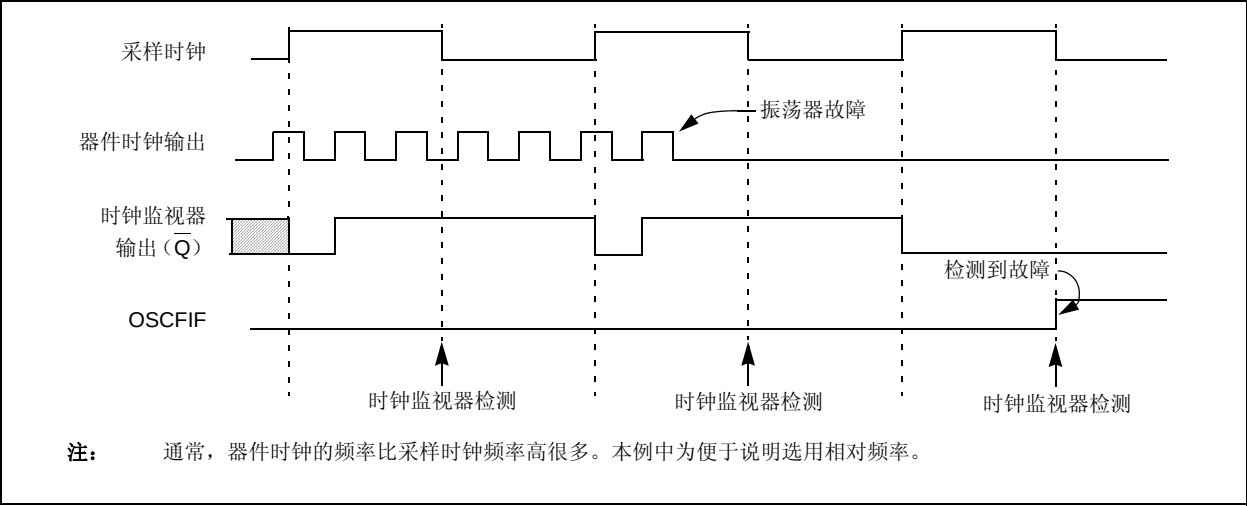
FSCM 只能检测出主时钟源或辅助时钟源的故障。如果内部振荡器模块发生故障，将不会检测到故障，当然也不可能采取任何措施。

27.5.1 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，使能 FSCM 时，禁止 WDT 对 INTRC 振荡器的操作没有任何影响。

如前所述，当检测到时钟故障时，时钟源将切换到 INTRC 时钟；这可能意味着代码执行速度会发生很大的变化。如果使能 WDT 时使用的小预分频值，时钟速度的下降将引起 WDT 超时，随后使器件复位。由于这个原因，故障保护时钟监视器事件也会使 WDT 和后分频器复位，使 WDT 从执行速度发生变化那一刻起开始重新计数，从而降低发生错误超时的可能。

图 27-5: FSCM 时序图



27.5.2 退出故障保护工作模式

器件复位或进入功耗管理模式均可终止故障保护时钟监视器条件。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（具有如 OST 或 PLL 定时器等振荡器模式所需的起振延时）。INTRC 振荡器将在主时钟源就绪之前提供系统时钟（类似于双速启动）。随后系统时钟源切换为主时钟（由 OSCCON 寄存器中的 OSTS 位置 1 指示）。然后，FSCM 恢复监视外设时钟。

在起振期间，主时钟源可能永远不能就绪。在这种情况下，器件工作将以 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。

27.5.3 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。在功耗管理模式下功耗管理时钟源的 FSCM 继续工作。

如果在功耗管理工作模式下发生了振荡器故障，后续事件取决于是否允许了振荡器故障中断。如果允许了该中断（OSCFIF = 1），代码执行将以 INTRC 多路开关的输出作为时钟源，并且不会自动转换回发生故障的时钟源。

如果禁止了该中断，空闲模式下产生的后续中断将使 CPU 以 INTRC 时钟源作为时钟源开始执行指令。

27.5.4 POR 或从休眠中唤醒

FSCM 设计为在器件退出上电复位（POR）或低功耗休眠模式后检测振荡器故障。当器件主时钟为 EC 或 INTRC 模式时，会在这些事件后立即开始故障保护时钟监视。

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 FSCM 采样时钟时间长很多，因此可能会检测到假的时钟故障。为避免这种情况，内部振荡器模块会被自动配置为器件时钟并一直工

作到主时钟稳定为止（OST 和 PLL 定时器发生超时）。这与双速启动模式相同。一旦主时钟稳定下来，INTRC 就将重新作为 FSCM 时钟源。

注： 用于防止在 POR 或从休眠状态唤醒时发生错误中断的逻辑，同样也将阻止随后对振荡器故障的检测。通过监视 OSTS 位，并使用定时程序来确定振荡器起振时间是否过长，可避免这个问题。即使如此，在检测到振荡器故障时也不会将振荡器故障中断标志位置 1。

正如第 27.4.1 节“使用双速启动时的注意事项”中所述，在等待主时钟稳定的过程中，可以选择另一种时钟配置和进入另一种功耗管理模式。当选择了新的功耗管理模式时，主时钟将被禁止。

27.6 程序校验和代码保护

对于 PIC18F46J50 系列中的所有器件，片上程序存储空间被视为一个统一的存储块。配置位 CP0 控制该存储块的代码保护。该位阻止外部对程序存储空间的读写操作。但对正常的代码执行没有直接影响。

27.6.1 配置寄存器保护

有两种方式保护配置寄存器使其免遭破坏性的更改或读取。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。为防止不可预见的事件造成损害，由于单个存储单元故障（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。这在用户看来就是配置不匹配（CM）复位。

配置寄存器的数据来自程序存储器中的 FCW。当 CP0 位置 1 时，也将保护器件配置的源数据。

PIC18F46J50 系列

27.7 在线串行编程（ICSP）

可以在最终的应用电路中对 PIC18F46J50 系列单片机进行串行编程。只需要 5 根线即可实现这一操作，其中时钟线、数据线各一根，其余 3 根分别是电源线、接地线和编程电压线。这允许用户在生产电路板时使用未编程器件，仅在产品交付之前才对单片机进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

27.8 在线调试器

将 `DEBUG` 配置位编程为 0，可使能在线调试功能。该功能允许与 MPLAB® IDE 配合使用进行简单的调试功能。当使能了单片机的这项功能时，有些资源就不能用于一般用途了。

表 27-4 列出了后台调试器所需的资源。

表 27-4: 调试器资源

I/O 引脚:	RB6 和 RB7
堆栈:	TOSx 寄存器保留

28.0 指令集汇总

PIC18F46J50 系列器件具有一个包含 75 条核心 PIC18 指令的标准指令集，和一个包含 8 条新指令的扩展指令集，扩展指令集用于优化递归代码或使用软件堆栈的代码。本章后面的部分将讨论扩展指令集。

28.1 标准指令集

标准的 PIC18 指令集与以前的 PIC® MCU 指令集相比，添加了很多增强功能，并保持了易于从这些 PIC MCU 指令集移植的特点。大部分指令为单程序存储字指令（16 位），只有 4 条指令需要两个程序存储单元。

每个单字指令都是一个 16 位字，由操作码（指明指令类型）和一个或多个操作数（指定指令操作）组成。

整个指令集具有高度的正交性，可以分为以下 4 种基本类型：

- 字节操作类指令
- 位操作类指令
- 立即数操作类指令
- 控制操作类指令

表 28-2 为 PIC18 指令集汇总，列出了上述四类指令。

表 28-1 给出了操作码字段的说明。

大部分字节操作类的指令都含有三种操作数：

1. 文件寄存器（由“f”指定）
2. 保存结果的目标寄存器（由“d”指定）
3. 被访问存储区（由“a”指定）

文件寄存器标识符“f”指定了指令将会使用哪一个文件寄存器。目标寄存器标识符“d”指定了操作结果的存放位置。如果 d 为 0，操作结果存入 WREG 寄存器中。如果 d 为 1，操作结果存入指令指定的文件寄存器中。

所有位操作类指令都含有三种操作数：

1. 文件寄存器（由“f”指定）
2. 文件寄存器中的位（由“b”指定）
3. 被访问存储区（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文件寄存器地址。

立即数操作类指令使用以下操作数：

- 要装入到文件寄存器中的立即数（由“k”指定）
- 要装入立即数的 FSR 寄存器（由“f”指定）
- 不需要操作数（由“—”指定）

控制类指令可以使用以下操作数：

- 程序存储器地址（由“n”指定）
- CALL 或 RETURN 指令的模式（由“s”指定）
- 表读和表写指令的模式（由“m”指定）
- 不需要操作数（由“—”指定）

除了 4 条双字指令外，其他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，它会执行为 NOP 指令。

除非条件测试结果为真或者指令执行改变了程序计数器（PC）的值，否则执行所有的单字指令都只需要一个指令周期。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条 NOP 指令。

执行双字指令需要两个指令周期。

每个指令周期由 4 个振荡器周期组成。因此，如果振荡器频率为 4 MHz，正常的指令执行时间为 1 μs。如果条件测试结果为真或者指令执行改变了程序计数器的值，则该指令的执行时间为 2 μs。双字跳转指令（如果为真）的执行则需要 3 μs。

图 28-1 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 28-2）列出了可被 Microchip MPASM™ 汇编器识别的标准指令。

第 28.1.1 节“标准指令集”中对每条指令进行了介绍。

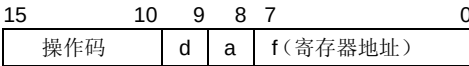
PIC18F46J50 系列

表 28-1: 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 存储单元 (BSR 寄存器被忽略) a = 1: 由 BSR 寄存器指定 RAM 存储区
bbb	8 位文件寄存器内的位地址 (0 到 7)
BSR	存储区选择寄存器。用于选择当前的 RAM 存储区
C、DC、Z、OV 和 N	ALU 状态位: 进位、半进位、全零、溢出和负标志位
d	目标寄存器选择位: d = 0: 结果保存至 WREG 寄存器 d = 1: 结果保存至文件寄存器 f
dest	目标寄存器: 可以是 WREG 寄存器或指定的文件寄存器地址
f	8 位文件寄存器地址 (00h 到 FFh), 或 2 位 FSR 标识符 (0h 到 3h)
f _s	12 位文件寄存器地址 (000h 到 FFFh)。这是源地址
f _d	12 位文件寄存器地址 (000h 到 FFFh)。这是目标地址
GIE	全局中断允许位
k	立即数、常数或者标号 (可能是 8 位、12 位或 20 位的值)
label	标号名称
mm	表读和表写指令的 TBLPTR 寄存器模式 只与表读和表写指令一起使用
*	不改变寄存器 (如用于表读和表写的 TBLPTR)
*+	后递增寄存器 (如用于表读和表写的 TBLPTR)
*-	后递减寄存器 (如用于表读和表写的 TBLPTR)
+*	预递增寄存器 (如用于表读和表写的 TBLPTR)
n	相对跳转指令的相对地址 (二进制补码形式), 或 Call/ 跳转和 Return 指令的直接地址
PC	程序计数器
PCL	程序计数器低字节
PCH	程序计数器高字节
PCLATH	程序计数器高字节锁存器
PCLATU	程序计数器最高字节锁存器
PD	掉电位
PRODH	乘积的高字节
PRODL	乘积的低字节
s	快速调用 / 返回模式选择位: s = 0: 不对影子寄存器进行更新, 也不用影子寄存器的内容更新其他寄存器 s = 1: 将寄存器的值装入影子寄存器或把影子寄存器中的值装入寄存器 (快速模式)
TBLPTR	21 位表指针 (指向程序存储器地址)
TABLAT	8 位表锁存器
T _O	超时位
TOS	栈顶
u	未使用或未改变
WDT	看门狗定时器
WREG	工作寄存器 (累加器)
x	无关位 (0 或 1)。汇编器将生成 x = 0 的代码; 为了与所有的 Microchip 软件工具兼容, 建议使用这种形式
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 指向的寄存器中的位 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)

例 28-1: 指令的通用格式

针对字节的文件寄存器操作指令

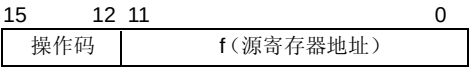


d = 0, 表示结果存入 WREG 寄存器
d = 1, 表示结果存入文件寄存器 (f)
a = 0, 强制使用快速操作存储区
a = 1, 根据 BSR 选择存储区
f = 8 位文件寄存器地址

指令示例

ADDWF MYREG, W, B

字节到字节的传送操作 (双字) 指令

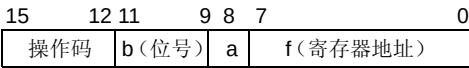


MOVFF MYREG1, MYREG2



f = 12 位文件寄存器地址

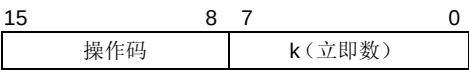
针对位的文件寄存器操作指令



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

立即数操作指令

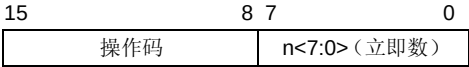


MOVLW 7Fh

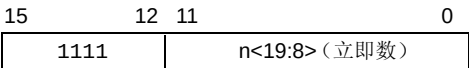
k = 8 位立即数的值

控制操作指令

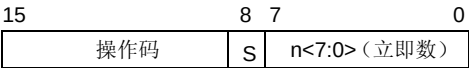
CALL、GOTO 和跳转操作类指令



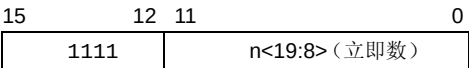
GOTO Label



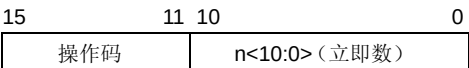
n = 20 位立即数的值



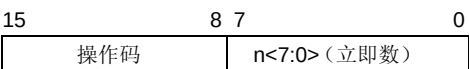
CALL MYFUNC



S = 快速位



BRA MYFUNC



BC MYFUNC

PIC18F46J50 系列

表 28-2: PIC18F46J50 系列指令集

助记符, 操作数	说明	周期数	16 位指令字		受影响的 状态位	注
			MSb	LSb		
针对字节的操作类指令						
ADDWF f, d, a	WREG 与 f 相加	1	0010	01da ffff ffff	C, DC, Z, OV, N	1, 2
ADDWFC f, d, a	WREG 与 f 带进位相加	1	0010	00da ffff ffff	C, DC, Z, OV, N	1, 2
ANDWF f, d, a	WREG 和 f 作逻辑与运算	1	0001	01da ffff ffff	Z, N	1, 2
CLRF f, a	将 f 清零	1	0110	101a ffff ffff	Z	2
COMF f, d, a	对 f 取反	1	0001	11da ffff ffff	Z, N	1, 2
CPFSEQ f, a	将 f 与 WREG 作比较, 相等则跳过	1 (2 或 3)	0110	001a ffff ffff	无	4
CPFSGT f, a	将 f 与 WREG 作比较, 大于则跳过	1 (2 或 3)	0110	010a ffff ffff	无	4
CPFSLT f, a	将 f 与 WREG 作比较, 小于则跳过	1 (2 或 3)	0110	000a ffff ffff	无	1, 2
DECf f, d, a	f 递减 1	1	0000	01da ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ f, d, a	f 递减 1, 为 0 则跳过	1 (2 或 3)	0010	11da ffff ffff	无	1, 2, 3, 4
DCFSNZ f, d, a	f 递减 1, 非 0 则跳过	1 (2 或 3)	0100	11da ffff ffff	无	1, 2
INCF f, d, a	f 递增 1	1	0010	10da ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ f, d, a	f 递增 1, 为 0 则跳过	1 (2 或 3)	0011	11da ffff ffff	无	4
INFSNZ f, d, a	f 递增 1, 非 0 则跳过	1 (2 或 3)	0100	10da ffff ffff	无	1, 2
IORWF f, d, a	WREG 和 f 作逻辑或运算	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	从 f _s (源) 送到 f _d (目标)	2	1100	ffff ffff ffff	无	
	第二个字		1111	ffff ffff ffff		
MOVWF f, a	将 WREG 内容传送到 f	1	0110	111a ffff ffff	无	
MULWF f, a	WREG 与 f 相乘	1	0000	001a ffff ffff	无	1, 2
NEGF f, a	对 f 取补	1	0110	110a ffff ffff	C, DC, Z, OV, N	
RLCF f, d, a	f 带进位循环左移	1	0011	01da ffff ffff	C, Z, N	1, 2
RLNCF f, d, a	f 循环左移 (不带进位)	1	0100	01da ffff ffff	Z, N	
RRCF f, d, a	f 带进位循环右移	1	0011	00da ffff ffff	C, Z, N	
RRNCF f, d, a	f 循环右移 (不带进位)	1	0100	00da ffff ffff	Z, N	
SETF f, a	将 f 的内容置为全 1	1	0110	100a ffff ffff	无	1, 2
SUBFWB f, d, a	WREG 减去 f (带借位)	1	0101	01da ffff ffff	C, DC, Z, OV, N	
SUBWF f, d, a	f 减去 WREG	1	0101	11da ffff ffff	C, DC, Z, OV, N	1, 2
SUBWFB f, d, a	f 减去 WREG (带借位)	1	0101	10da ffff ffff	C, DC, Z, OV, N	
SWAPF f, d, a	将 f 中的两个半字节进行交换	1	0011	10da ffff ffff	无	4
TSTFSZ f, a	测试 f, 为 0 则跳过	1 (2 或 3)	0110	011a ffff ffff	无	1, 2
XORWF f, d, a	WREG 和 f 作逻辑异或运算	1	0001	10da ffff ffff	Z, N	

- 注 1: 当 PORT 寄存器修改自身时 (例如, MOVF PORTB, 1, 0), 修改时使用的值是引脚上的当前值。例如, 如果将一引脚配置为输入, 其对应数据锁存器中的值将为 1, 但此时若有外部器件将该引脚驱动为低电平, 则被写回数据锁存器的数据值将是 0。
- 2: 当对 TMR0 寄存器执行该指令 (并且 d = 1) 时, 如果已为其分配了预分频器, 则将该预分频器清零。
- 3: 如果程序计数器 (PC) 被修改或者条件测试为真, 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 28-2: PIC18F46J50 系列指令集 (续)

助记符， 操作数	说明	周期数	16 位指令字				受影响的 状态位	注	
			MSb		LSb				
针对位的操作类指令									
BCF	f, b, a	将 f 中的某位清零	1	1001	bbba	ffff	ffff	无	1, 2
BSF	f, b, a	将 f 中的某位置 1	1	1000	bbba	ffff	ffff	无	1, 2
BTFSC	f, b, a	测试 f 中的某位，为 0 则跳过	1（2 或 3）	1011	bbba	ffff	ffff	无	3, 4
BTFSS	f, b, a	测试 f 中的某位，为 1 则跳过	1（2 或 3）	1010	bbba	ffff	ffff	无	3, 4
BTG	f, b, a	将 f 中的某位取反	1	0111	bbba	ffff	ffff	无	1, 2
控制类指令									
BC	n	进位则跳转	1（2）	1110	0010	nnnn	nnnn	无	4
BN	n	为负则跳转	1（2）	1110	0110	nnnn	nnnn	无	
BNC	n	无进位则跳转	1（2）	1110	0011	nnnn	nnnn	无	
BNN	n	不为负则跳转	1（2）	1110	0111	nnnn	nnnn	无	
BNOV	n	不溢出则跳转	1（2）	1110	0101	nnnn	nnnn	无	
BNZ	n	不为零则跳转	1（2）	1110	0001	nnnn	nnnn	无	
BOV	n	溢出则跳转	1（2）	1110	0100	nnnn	nnnn	无	
BRA	n	无条件跳转	2	1101	0nnn	nnnn	nnnn	无	
BZ	n	为零则跳转	1（2）	1110	0000	nnnn	nnnn	无	
CALL	n, s	调用子程序	2	1110	110s	kkkk	kkkk	无	
		第一个字		1111	kkkk	kkkk	kkkk		
		第二个字							
CLRWDT	—	将看门狗定时器清零	1	0000	0000	0000	0100	TO, PD	
DAW	—	对 WREG 进行十进制调整	1	0000	0000	0000	0111	C	
GOTO	n	跳转到地址	2	1110	1111	kkkk	kkkk	无	
		第一个字		1111	kkkk	kkkk	kkkk		
		第二个字							
NOP	—	空操作	1	0000	0000	0000	0000	无	
NOP	—	空操作	1	1111	xxxx	xxxx	xxxx	无	
POP	—	弹出返回堆栈栈顶（TOS）的内容	1	0000	0000	0000	0110	无	
PUSH	—	将数据压入返回堆栈栈顶（TOS）	1	0000	0000	0000	0101	无	
RCALL	n	相对调用	2	1101	1nnn	nnnn	nnnn	无	
RESET		软件器件复位	1	0000	0000	1111	1111	全部	
RETFIE	s	中断返回允许	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	返回时将立即数送入 WREG	2	0000	1100	kkkk	kkkk	无	
RETURN	s	从子程序返回	2	0000	0000	0001	001s	无	
SLEEP	—	进入待机模式	1	0000	0000	0000	0011	TO, PD	

- 注 1: 当 PORT 寄存器修改自身时 (例如, MOVF PORTB, 1, 0), 修改时使用的值是引脚上的当前值。例如, 如果将一引脚配置为输入, 其对应数据锁存器中的值将为 1, 但此时若有外部器件将该引脚驱动为低电平, 则被写回数据锁存器的数据值将是 0。
- 2: 当对 TMR0 寄存器执行该指令 (并且 d = 1) 时, 如果已为其分配了预分频器, 则将该预分频器清零。
- 3: 如果程序计数器 (PC) 被修改或者条件测试为真, 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。

PIC18F46J50 系列

表 28-2: PIC18F46J50 系列指令集 （续）

助记符, 操作数	说明	周期数	16 位指令字		受影响的状态位	注
			MSb	LSb		
立即数操作类指令						
ADDLW k	WREG 与立即数相加	1	0000	1111 kkkk kkkk	C, DC, Z, OV, N	
ANDLW k	WREG 和立即数进行逻辑与运算	1	0000	1011 kkkk kkkk	Z, N	
IORLW k	WREG 和立即数进行逻辑或运算	1	0000	1001 kkkk kkkk	Z, N	
LFSR f, k	传送立即数（12 位） 第二个字 到 FSR（f） 第一个字	2	1110	1110 00ff kkkk	无	
			1111	0000 kkkk kkkk		
MOVLB k	将立即数送入 BSR<3:0>	1	0000	0001 0000 kkkk	无	
MOVLW k	将立即数送入 WREG	1	0000	1110 kkkk kkkk	无	
MULLW k	WREG 和立即数相乘	1	0000	1101 kkkk kkkk	无	
RETLW k	返回时将立即数送入 WREG	2	0000	1100 kkkk kkkk	无	
SUBLW k	立即数减去 WREG	1	0000	1000 kkkk kkkk	C, DC, Z, OV, N	
XORLW k	WREG 和立即数进行逻辑异或运算	1	0000	1010 kkkk kkkk	Z, N	
数据存储器 ↔ 程序存储器操作						
TBLRD*	表读	2	0000	0000 0000 1000	无	
TBLRD*+	后递增表读	2	0000	0000 0000 1001	无	
TBLRD*-	后递减表读		0000	0000 0000 1010	无	
TBLRD+*	预递增表读		0000	0000 0000 1011	无	
TBLWT*	表写		0000	0000 0000 1100	无	
TBLWT*+	后递增表写		0000	0000 0000 1101	无	
TBLWT*-	后递减表写		0000	0000 0000 1110	无	
TBLWT+*	预递增表写		0000	0000 0000 1111	无	

- 注 1: 当 PORT 寄存器修改自身时（例如，**MOVF PORTB, 1, 0**），修改时使用的值是引脚上的当前值。例如，如果将一引脚配置为输入，其对应数据锁存器中的值将为 1，但此时若有外部器件将该引脚驱动为低电平，则被写回数据锁存器的数据值将是 0。
- 2: 当对 TMR0 寄存器执行该指令（并且 d = 1）时，如果已为其分配了预分频器，则将该预分频器清零。
- 3: 如果程序计数器（PC）被修改或者条件测试为真，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。

28.1.1 标准指令集

ADDLW		W 与立即数相加						
语法:	ADDLW k							
操作数:	$0 \leq k \leq 255$							
操作:	$(W) + k \rightarrow W$							
受影响的状态位:	N、OV、C、DC 和 Z							
机器码:	<table border="1"><tr><td>0000</td><td>1111</td><td>kkkk</td><td>kkkk</td></tr></table>				0000	1111	kkkk	kkkk
0000	1111	kkkk	kkkk					
说明:	将 W 寄存器的内容与 8 位立即数 k 相加，结果存储在 W 寄存器中。							
指令字数:	1							
指令周期数:	1							
Q 周期操作:								
	Q1	Q2	Q3	Q4				
	译码	读立即数 k	处理数据	写入 W				

示例: ADDLW 0x15

执行指令前
W = 10h

执行指令后
W = 25h

ADDWF		W 与 f 相加										
语法:	ADDWF f {,d {,a}}											
操作数:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]											
操作:	(W) + (f) → dest											
受影响的状态位:	N、OV、C、DC 和 Z											
机器码:	0010	01da	ffff	ffff								
说明:	将 W 的内容与 f 寄存器的内容相加。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。 如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。 如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。											
指令字数:	1											
指令周期数:	1											
Q 周期操作:	<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>读寄存器 f</td><td>处理数据</td><td>写入目标寄存器</td></tr></table>				Q1	Q2	Q3	Q4	译码	读寄存器 f	处理数据	写入目标寄存器
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，那么指令格式将变为: {label} 指令参数。

PIC18F46J50 系列

ADDWFC

W 与 f 带进位相加

语法:

ADDWFC f {,d {,a}}

操作数:

0 ≤ f ≤ 255

d ∈ [0,1]

a ∈ [0,1]

操作:

(W) + (f) + (C) → dest

受影响的状态位:

N、OV、C、DC 和 Z

机器码:

0010	00da	ffff	ffff

说明:

将 W 的内容、进位标志位与数据存储单元 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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例:	ADDWFC REG, 0, 1
执行指令前	
进位标志位	= 1
REG	= 02h
W	= 4Dh
执行指令后	
进位标志位	= 0
REG	= 02h
W	= 50h

ANDLW

立即数和 W 寄存器作逻辑与运算

语法:

ANDLW k

操作数:

0 ≤ k ≤ 255

操作:

(W) .AND. k → W

受影响的状态位:

N 和 Z

机器码:

0000	1011	kkkk	kkkk

说明:

将 W 的内容与 8 位立即数 k 进行逻辑与运算。结果存储在 W 寄存器中。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例:

ANDLW 0x5F

执行指令前

W = A3h

执行指令后

W = 03h

ANDWF

将 W 和 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) → dest

受影响的状态位:

N 和 Z

机器码:

0001	01da	ffff	ffff

说明:

将 W 的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

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，程序将跳转。

“2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数:

1

指令周期数:

1（2）

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:

HERE BC 5

执行指令前

PC = 地址（HERE）

执行指令后

如果进位标志位 = 1；
PC = 地址（HERE + 12）
如果进位标志位 = 0；
PC = 地址（HERE + 2）

PIC18F46J50 系列

BCF

将 f 中的某位清零

语法:

BCF f, b {,a}

操作数:

$0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$

操作:

$0 \rightarrow f$

受影响的状态位:

无

机器码:

1001	bbba	ffff	ffff

说明:

将寄存器 f 中的位 b 清零。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ （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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

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:	<table border="1"><tr><td>1110</td><td>0110</td><td>nnnn</td><td>nnnn</td></tr></table>					1110	0110	nnnn	nnnn
1110	0110	nnnn	nnnn						
说明:	如果负标志位为 1，程序将跳转。 “2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。								
指令字数:	1								
指令周期数:	1（2）								
Q 周期操作:	如果跳转:								
	Q1	Q2	Q3	Q4					
	<table border="1"><tr><td>译码</td></tr></table>	译码	<table border="1"><tr><td>读立即数 n</td></tr></table>	读立即数 n	<table border="1"><tr><td>处理数据</td></tr></table>	处理数据	<table border="1"><tr><td>写入 PC</td></tr></table>		写入 PC
译码									
读立即数 n									
处理数据									
写入 PC									
	<table border="1"><tr><td>空操作</td></tr></table>	空操作	<table border="1"><tr><td>空操作</td></tr></table>	空操作	<table border="1"><tr><td>空操作</td></tr></table>	空操作	<table border="1"><tr><td>空操作</td></tr></table>		空操作
空操作									
空操作									
空操作									
空操作									
如果不跳转:									
	Q1	Q2	Q3	Q4					
	<table border="1"><tr><td>译码</td></tr></table>	译码	<table border="1"><tr><td>读立即数 n</td></tr></table>	读立即数 n	<table border="1"><tr><td>处理数据</td></tr></table>	处理数据	<table border="1"><tr><td>空操作</td></tr></table>		空操作
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）

BNC

无进位则跳转

语法:

BNC n

操作数:

-128 ≤ n ≤ 127

操作:

如果进位标志位为 0,
(PC) + 2 + 2n → PC

受影响的状态位:

无

机器码:

1110	0011	nnnn	nnnn

说明:

如果进位标志位为 0，程序将跳转。

“2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数:

1

指令周期数:

1（2）

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:

HERE

BNC

Jump

执行指令前

PC

=

地址（HERE）

执行指令后

如果进位标志位

=

0；

PC

=

地址（Jump）

如果进位标志位

=

1；

PC

=

地址（HERE + 2）

BNN

不为负则跳转

语法:

BNN n

操作数:

-128 ≤ n ≤ 127

操作:

如果负标志位为 0,
(PC) + 2 + 2n → PC

受影响的状态位:

无

机器码:

1110	0111	nnnn	nnnn

说明:

如果负标志位为 0，程序将跳转。

“2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数:

1

指令周期数:

1（2）

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:

HERE

BNN

Jump

执行指令前

PC

=

地址（HERE）

执行指令后

如果负标志位

=

0；

PC

=

地址（Jump）

如果负标志位

=

1；

PC

=

地址（HERE + 2）

PIC18F46J50 系列

BNOV		不溢出则跳转															
语法:	BNOV n																
操作数:	$-128 \leq n \leq 127$																
操作:	如果溢出标志位为 0, $(PC) + 2 + 2n \rightarrow PC$																
受影响的状态位:	无																
机器码:	<table border="1"><tr><td>1110</td><td>0101</td><td>nnnn</td><td>nnnn</td></tr></table>				1110	0101	nnnn	nnnn									
1110	0101	nnnn	nnnn														
说明:	如果溢出标志位为 0, 程序将跳转。 “2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令, 所以新地址将为 $PC + 2 + 2n$ 。该指令为一条双周期指令。																
指令字数:	1																
指令周期数:	1（2）																
Q 周期操作:																	
如果跳转:	<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>读立即数 n</td><td>处理数据</td><td>写入 PC</td></tr><tr><td>空操作</td><td>空操作</td><td>空操作</td><td>空操作</td></tr></table>					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
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:	<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>读立即数 n</td><td>处理数据</td><td>空操作</td></tr></table>					Q1	Q2	Q3	Q4	译码	读立即数 n	处理数据	空操作				
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:	<table border="1"><tr><td>1110</td><td>0001</td><td>nnnn</td><td>nnnn</td></tr></table>				1110	0001	nnnn	nnnn									
1110	0001	nnnn	nnnn														
说明:	如果全零标志位为 0, 程序将跳转。 “2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令, 所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。																
指令字数:	1																
指令周期数:	1（2）																
Q 周期操作:																	
如果跳转:	<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>读立即数 n</td><td>处理数据</td><td>写入 PC</td></tr><tr><td>空操作</td><td>空操作</td><td>空操作</td><td>空操作</td></tr></table>					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
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:	<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>读立即数 n</td><td>处理数据</td><td>空操作</td></tr></table>					Q1	Q2	Q3	Q4	译码	读立即数 n	处理数据	空操作				
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)	

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:

“2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数:

1

指令周期数:

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 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

受影响的状态位:

无

机器码:

1000	bbba	ffff	ffff

说明:

将寄存器 f 的位 b 置 1。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

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

PIC18F46J50 系列

BTFSC 测试文件寄存器中的某位，为 0 则跳过

语法: BTFSC f, b {,a}

操作数: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$

操作: 如果 $(f < b) = 0$ ，则跳过

受影响的状态位: 无

机器码:

1011	bbba	ffff	ffff

说明: 如果寄存器 f 的位 b 为 0，则跳过下一条指令。即在 b 位为 0 时，丢弃在执行当前指令期间取的下一条指令，转而执行一条 NOP 指令，使该指令变成双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ (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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

指令字数: 1

指令周期数: 1 (2)
注: 如果跳过，且后面跟有 2 字指令，则执行 BTFSC 需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过，且后面跟有 2 字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

HERE	BTFSC	FLAG, 1, 0
FALSE	:	
TRUE	:	

执行指令前
PC = 地址 (HERE)

执行指令后
如果 FLAG<1> = 0:
PC = 地址 (TRUE)
如果 FLAG<1> = 1:
PC = 地址 (FALSE)

BTFSS 测试文件寄存器中的某位，为 1 则跳过

语法: BTFSS f, b {,a}

操作数: $0 \leq f \leq 255$
 $0 \leq b < 7$
 $a \in [0,1]$

操作: 如果 $(f < b) = 1$ ，则跳过

受影响的状态位: 无

机器码:

1010	bbba	ffff	ffff

说明: 如果寄存器 f 的位 b 为 1，则跳过下一条指令。即在 b 位为 1 时，丢弃在执行当前指令期间取的下一条指令，转而执行一条 NOP 指令，使该指令变成双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ (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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

指令字数: 1

指令周期数: 1 (2)
注: 如果跳过，且后面跟有 2 字指令，则执行 BTFSS 需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过，且后面跟有 2 字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

HERE	BTFSS	FLAG, 1, 0
FALSE	:	
TRUE	:	

执行指令前
PC = 地址 (HERE)

执行指令后
如果 FLAG<1> = 0:
PC = 地址 (FALSE)
如果 FLAG<1> = 1:
PC = 地址 (TRUE)

BTG

将 f 中的某位取反

语法:

BTG f, b {,a}

操作数:

$0 \leq f \leq 255$

$0 \leq b < 7$

$a \in [0,1]$

操作:

$(\overline{f\langle b \rangle}) \rightarrow f\langle b \rangle$

受影响的状态位:

无

机器码:

0111	bbba	ffff	ffff

说明:

将数据存储单元 f 中的位 b 取反。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ （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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

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

LATC,

4,

0

执行指令前:

LATC

=

0111 0101 [75h]

执行指令后:

LATC

=

0110 0101 [65h]

BOV

溢出则跳转

语法:

BOV n

操作数:

$-128 \leq n \leq 127$

操作:

如果溢出标志位为 1，
 $(PC) + 2 + 2n \rightarrow PC$

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。

“2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 $PC + 2 + 2n$ 。该指令为一条双周期指令。

指令字数:

1

指令周期数:

1（2）

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:

HERE

BOV

Jump

执行指令前

PC

=

地址（HERE）

执行指令后

如果溢出标志位

=

1；

PC

=

地址（Jump）

如果溢出标志位

=

0；

PC

=

地址（HERE + 2）

PIC18F46J50 系列

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:	<table><tr><td>1110</td><td>0000</td><td>nnnn</td><td>nnnn</td></tr></table>	1110	0000	nnnn	nnnn
1110	0000	nnnn	nnnn		
说明:	如果全零标志位为 1, 程序将跳转。				

“2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数: 1
指令周期数: 1 (2)
Q 周期操作:
如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:	HERE	BZ	Jump
执行指令前			
PC	=	地址 (HERE)	
执行指令后			
如果全零标志位	=	1;	
PC	=	地址 (Jump)	
如果全零标志位	=	0;	
PC	=	地址 (HERE + 2)	

CALL	调用子程序
语法:	CALL k {,s}
操作数:	0 ≤ k ≤ 1048575 s ∈ [0,1]
操作:	(PC) + 4 → TOS, k → PC<20:1>; 如果 s = 1, (W) → WS, (STATUS) → STATUSS, (BSR) → BSR

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 BSR。如果 s = 0，将不会进行任何更新（默认）。然后，将 20 位的值 k 装入 PC<20:1>。CALL 是一条双周期指令。

指令字数: 2
指令周期数: 2
Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	将 PC 压入堆栈	读立即数 k<19:8>, 写入 PC
空操作	空操作	空操作	空操作

示例:	HERE	CALL	THERE, 1
执行指令前			
PC	=	地址 (HERE)	
执行指令后			
PC	=	地址 (THERE)	
TOS	=	地址 (HERE + 4)	
WS	=	W	
BSRS	=	BSR	
STATUSS	=	STATUS	

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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例:

CLRf

FLAG_REG,1

执行指令前

FLAG_REG = 5Ah

执行指令后

FLAG_REG = 00h

CLRWDt

将看门狗定时器清零

语法:

CLRWDt

操作数:

无

操作:

000h → WDT,
000h → WDT 后分频器,
1 → $\overline{TO}$,
1 → $\overline{PD}$

受影响的状态位:

$\overline{TO}$ 和 $\overline{PD}$

机器码:

0000	0000	0000	0100

说明:

CLRWDt 指令复位看门狗定时器及其后分频器。状态位 $\overline{TO}$ 和 $\overline{PD}$ 置 1。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	处理数据	空操作

示例:

CLRWDt

执行指令前

WDT 计数器 = ?

执行指令后

WDT 计数器 = 00h

WDT 后分频器 = 0

$\overline{TO}$ = 1

$\overline{PD}$ = 1

PIC18F46J50 系列

COMF		对 f 取反									
语法:	COMF f {,d {,a}}										
操作数:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]										
操作:	$\bar{f} \rightarrow \text{dest}$										
受影响的状态位:	N 和 Z										
机器码:	<table border="1"><tr><td>0001</td><td>11da</td><td>ffff</td><td>ffff</td></tr></table>			0001	11da	ffff	ffff				
0001	11da	ffff	ffff								
说明:	将寄存器 f 的内容取反。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。 如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。 如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。										
指令字数:	1										
指令周期数:	1										
Q 周期操作:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>译码</td><td>读寄存器 f</td><td>处理数据</td><td>写入目标寄存器</td></tr></table>			Q1	Q2	Q3	Q4	译码	读寄存器 f	处理数据	写入目标寄存器
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:	<table border="1"><tr><td>0110</td><td>001a</td><td>ffff</td><td>ffff</td></tr></table>				0110	001a	ffff	ffff								
0110	001a	ffff	ffff													
说明:	通过执行无符号的减法，将数据存储单元 f 的内容与 W 的内容作比较。 如果 f = W，则所取的指令被丢弃并执行一条 NOP 指令，使该指令成为双周期指令。 如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。 如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。															
指令字数:	1															
指令周期数:	1（2） 注： 如果跳过，且后面跟有 2 字指令，则执行 CPFSEQ 需要 3 个周期。															
Q 周期操作:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>译码</td><td>读寄存器 f</td><td>处理数据</td><td>空操作</td></tr></table>				Q1	Q2	Q3	Q4	译码	读寄存器 f	处理数据	空操作				
Q1	Q2	Q3	Q4													
译码	读寄存器 f	处理数据	空操作													
如果跳过:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>空操作</td><td>空操作</td><td>空操作</td><td>空操作</td></tr></table>				Q1	Q2	Q3	Q4	空操作	空操作	空操作	空操作				
Q1	Q2	Q3	Q4													
空操作	空操作	空操作	空操作													
如果跳过，且后面跟有 2 字指令:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>空操作</td><td>空操作</td><td>空操作</td><td>空操作</td></tr><tr><td>空操作</td><td>空操作</td><td>空操作</td><td>空操作</td></tr></table>				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
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）

CPFSGT		比较 f 和 W，如果 f > W 则跳过						
语法：	CPFSGT f {,a}							
操作数：	0 ≤ f ≤ 255 a ∈ [0,1]							
操作：	(f) − (W)， 如果 (f) > (W)，则跳过（无符号比较）							
受影响的状态位：	无							
机器码：	<table border="1"><tr><td>0110</td><td>010a</td><td>ffff</td><td>ffff</td></tr></table>				0110	010a	ffff	ffff
0110	010a	ffff	ffff					
说明：	通过执行无符号的减法，将数据存储单元 f 的内容与 W 的内容作比较。 如果 f > W，则所取的指令被丢弃并执行一条 NOP 指令，使该指令成为双周期指令。 如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。 如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。							
指令字数：	1							
指令周期数：	1（2） 注： 如果跳过，且后面跟有 2 字指令，则执行 CPFSGT 需要 3 个周期。							
Q 周期操作：								
Q1		Q2		Q3		Q4		
译码		读寄存器 f		处理数据		空操作		
如果跳过：								
Q1		Q2		Q3		Q4		
空操作		空操作		空操作		空操作		
如果跳过，且后面跟有 2 字指令：								
Q1		Q2		Q3		Q4		
空操作		空操作		空操作		空操作		
空操作		空操作		空操作		空操作		

示例：	HERE CPFSGT REG, 0
	NGREATER :
	GREATER :
执行指令前	
PC	= 地址（HERE）
W	= ?
执行指令后	
如果 REG	> W；
PC	= 地址（GREATER）
如果 REG	≤ W；
PC	= 地址（NGREATER）

CPFSLT

比较 f 和 W，如果 f < W 则跳过

语法：

CPFSLT f{,a}

操作数：

0 ≤ f ≤ 255

a ∈ [0,1]

操作：

(f) − (W)，

如果 (f) < (W)，则跳过（无符号比较）

受影响的状态位：

无

机器码：

0110	000a	ffff	ffff

说明：

通过执行无符号的减法，将数据存储单元 f 的内容与 W 的内容作比较。

如果 f < W，则所取的指令被丢弃并执行一条 NOP 指令，使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）。

指令字数：

1

指令周期数：

1（2）

注：如果跳过，且后面跟有 2 字指令，则执行 CPFSLT 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过，且后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：	HERE CPFSLT REG, 1
	NLESS :
	LESS :
执行指令前	
PC	= 地址（HERE）
W	= ?
执行指令后	
如果 REG	< W；
PC	= 地址（LESS）
如果 REG	≥ W；
PC	= 地址（NLESS）

PIC18F46J50 系列

DAW	对 W 寄存器进行十进制调整
语法:	DAW
操作数:	无
操作:	如果 [W<3:0> > 9] 或 [DC = 1], 则 (W<3:0>) + 6 → W<3:0>; 否则 (W<3:0>) → W<3:0>
受影响的状态位:	如果 [W<7:4> > 9] 或 [C = 1], 则 (W<7:4>) + 6 → W<7:4>; C = 1; 否则 (W<7:4>) → W<7:4>
机器码:	C
说明:	DAW指令调整W寄存器内的8位值, 即之前两个压缩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:	
	Q1Q2Q3Q4
	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 递减 1
语法:	DECF f [{,d},{a}]
操作数:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
操作:	(f) − 1 → dest
受影响的状态位:	C、DC、N、OV 和 Z
机器码:	000001daffffffff
说明:	将寄存器 f 的内容递减 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。 如果 a 为 0 且使能了扩展指令集, 只要 f ≤ 95 (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见 第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令” 。
指令字数:	1
指令周期数:	1
Q 周期操作:	
	Q1Q2Q3Q4
	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

DECFSZ		f 递减 1, 为 0 则跳过					
语法:	DECFSZ f {,d {,a}}						
操作数:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]						
操作:	(f) − 1 → dest, 如果结果 = 0 则跳过						
受影响的状态位:	无						
机器码:	<table border="1"><tr><td>0010</td><td>11da</td><td>ffff</td><td>ffff</td></tr></table>			0010	11da	ffff	ffff
0010	11da	ffff	ffff				
说明:	将寄存器 f 的内容递减 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。 如果结果为 0，则丢弃已取的下一条指令转而执行一条 NOP 指令，使该指令成为双周期指令。 如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。 如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。						
指令字数:	1						
指令周期数:	1（2）						
	注：如果跳过，且后面跟有 2 字指令，则执行 DECFSZ 需要 3 个周期。						

Q 周期操作:	Q1	Q2	Q3	Q4
	译码	读寄存器 f	处理数据	写入目标寄存器
如果跳过:	Q1	Q2	Q3	Q4
	空操作	空操作	空操作	空操作
如果跳过，且后面跟有 2 字指令:	Q1	Q2	Q3	Q4
	空操作	空操作	空操作	空操作
	空操作	空操作	空操作	空操作

示例:	HERE	DECFSZ	CNT, 1, 1
		GOTO	LOOP
	CONTINUE		
执行指令前	PC	=	地址（HERE）
执行指令后	CNT	=	CNT − 1
	如果 CNT =	0；	
	PC	=	地址（CONTINUE）
	如果 CNT ≠	0；	
	PC	=	地址（HERE + 2）

DCFSNZ

f 递减 1, 非 0 则跳过

语法:

DCFSNZ f {,d {,a}}

操作数:

$0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作:

$(f) - 1 \rightarrow \text{dest},$
如果结果 $\neq 0$ 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 的内容递减 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果结果不为 0，则丢弃已取的下一条指令转而执行一条 NOP 指令，使该指令成为双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ （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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

指令字数:

1

指令周期数:

1（2）

注:

如果跳过，且后面跟有 2 字指令，则执行 DCFSNZ 需要 3 个周期。

Q 周期操作:	Q1	Q2	Q3	Q4
	译码	读寄存器 f	处理数据	写入目标寄存器
如果跳过:	Q1	Q2	Q3	Q4
	空操作	空操作	空操作	空操作
如果跳过，且后面跟有 2 字指令:	Q1	Q2	Q3	Q4
	空操作	空操作	空操作	空操作
	空操作	空操作	空操作	空操作

示例:	HERE	DCFSNZ	TEMP, 1, 0
	ZERO	:	
	NZERO	:	
执行指令前	TEMP	=	?
执行指令后	TEMP	=	TEMP − 1,
	如果 TEMP =	0；	
	PC	=	地址（ZERO）
	如果 TEMP ≠	0；	
	PC	=	地址（NZERO）

PIC18F46J50 系列

GOTO

无条件跳转

语法:

GOTO k

操作数:

$0 \leq k \leq 1048575$

操作:

$k \rightarrow PC<20:1>$

受影响的状态位:

无

机器码:

1110	1111	k_7kkk	$kkkk_0$
1111	$k_{19}kkk$	kkkk	$kkkk_8$

说明:

GOTO 指令允许无条件跳转到整个 2 MB 存储器范围中的任何位置。将 20 位值 k 装入 PC<20:1>。GOTO 始终为双周期指令。

指令字数:

2

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	空操作	读立即数 k<19:8>, 写入 PC
空操作	空操作	空操作	空操作

示例:

GOTO THERE

执行指令后
PC = 地址 (THERE)

INCF

f 递增 1

语法:

INCF f{,d{,a}}

操作数:

$0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作:

$(f) + 1 \rightarrow dest$

受影响的状态位:

C、DC、N、OV 和 Z

机器码:

0010	10da	ffff	ffff

说明:

将寄存器 f 的内容递增 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f (默认)。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区 (默认)。

如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ (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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

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

INCFSZ

f 递增 1，为 0 则跳过

语法：

INCFSZ f{,d{,a}}

操作数：

0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作：

(f) + 1 → dest,
如果结果 = 0 则跳过

受影响的状态位：

无

机器码：

0011	11da	ffff	ffff

说明：

将寄存器 f 的内容递增 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果返回寄存器 f（默认）。

如果结果为 0，则丢弃已取的下一条指令转而执行一条 NOP 指令，使该指令成为双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

指令字数：

1

指令周期数：

1（2）
注：如果跳过，且后面跟有 2 字指令，则执行 INCFSZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过，且后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

INFSNZ

f 递增 1，非 0 则跳过

语法：

INFSNZ f{,d{,a}}

操作数：

0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作：

(f) + 1 → dest,
如果结果 ≠ 0 则跳过

受影响的状态位：

无

机器码：

0100	10da	ffff	ffff

说明：

将寄存器 f 的内容递增 1。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果返回寄存器 f（默认）。

如果结果不为 0，则丢弃已取的下一条指令转而执行一条 NOP 指令，使该指令成为双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

指令字数：

1

指令周期数：

1（2）
注：如果跳过，且后面跟有 2 字指令，则执行 INFSNZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过，且后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

HERE
ZERO
ZERO

INFSNZ REG, 1, 0

执行指令前

PC = 地址（HERE）

执行指令后

REG = REG + 1
如果 REG ≠ 0：
PC = 地址（ZERO）
如果 REG = 0：
PC = 地址（ZERO）

PIC18F46J50 系列

IORLW 将立即数与 W 作逻辑或运算								
语法:	IORLW k							
操作数:	$0 \leq k \leq 255$							
操作:	(W) .OR. k $\rightarrow$ W							
受影响的状态位:	N 和 Z							
机器码:	<table border="1"><tr><td>0000</td><td>1001</td><td>kkkk</td><td>kkkk</td></tr></table>				0000	1001	kkkk	kkkk
0000	1001	kkkk	kkkk					
说明:	将 W 的内容与 8 位立即数 k 进行逻辑或运算。结果存储在 W 寄存器中。							
指令字数:	1							
指令周期数:	1							
Q 周期操作:								
	Q1	Q2	Q3	Q4				
	译码	读立即数 k	处理数据	写入 W				

示例:	IORLW	35h
执行指令前	W	= 9Ah
执行指令后	W	= BFh

IORWF 将 W 与 f 作逻辑或运算									
语法:	IORWF f {,d {,a}}								
操作数:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$								
操作:	$(W) .OR.(f) \rightarrow dest$								
受影响的状态位:	N 和 Z								
机器码:	<table><tr><td>0001</td><td>00da</td><td>ffff</td><td>ffff</td></tr></table>	0001	00da	ffff	ffff				
0001	00da	ffff	ffff						
说明:	将 W 的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。 如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。 如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$（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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。								
指令字数:	1								
指令周期数:	1								
Q 周期操作:									
<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>读寄存器 f</td><td>处理数据</td><td>写入目标寄存器</td></tr></table>		Q1	Q2	Q3	Q4	译码	读寄存器 f	处理数据	写入目标寄存器
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

LFSR

装入 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:	<table><tr><td>1110</td><td>1110</td><td>00ff</td><td>k₁₁kkk</td></tr><tr><td>1111</td><td>0000</td><td>k₇kkk</td><td>kkkk</td></tr></table>	1110	1110	00ff	k ₁₁ kkk	1111	0000	k ₇ kkk	kkkk			
1110	1110	00ff	k ₁₁ kkk									
1111	0000	k ₇ kkk	kkkk									
说明:	将 12 位立即数 k 装入 f 所指向的文件选择寄存器。											
指令字数:	2											
指令周期数:	2											
Q 周期操作:												
	Q1	Q2	Q3	Q4								
译码	读立即数 k 的 MSB	处理数据	将立即数 k 的 MSB 写入 FSRfH									
译码	读立即数 k 的 LSB	处理数据	将立即数 k 的 LSB 写入 FSRfL									

示例: LFSR 2, 0x3AB

执行指令后
FSR2H = 03h
FSR2L = ABh

MOVf	传送 f			
语法:	MOVf f {,d {,a}}			
操作数:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]			
操作:	f → dest			
受影响的状态位:	N 和 Z			
机器码:	0101	00da	ffff	ffff
说明:	根据 d 的状态, 将寄存器 f 的内容送入目标寄存器。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地址单元。 如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。 如果 a 为 0 且使能了扩展指令集, 只要 f ≤ 95 (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见 第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。			

示例: MOVF REG, 0, 0

执行指令前
REG = 22h
W = FFh
执行指令后
REG = 22h
W = 22h

PIC18F46J50 系列

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:

$0 \leq f_s \leq 4095$
 $0 \leq f_d \leq 4095$

操作:

$(f_s) \rightarrow f_d$

受影响的状态位:

无

机器码:

1100	ffff	ffff	ffff _s
1111	ffff	ffff	ffff _d

第一个字（源）

第二个字（目标）

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）中的任何单元，目标寄存器 f_d 也可以是 000h 到 FFFh 中的任何单元。

源或目标寄存器都可以是 W（这是个有用的特例）。

MOVFF 指令对于将数据存储单元中的内容送入外设寄存器（如发送缓冲区或 I/O 端口）的场合非常有用。

MOVFF 指令不能使用 PCL、TOSU、TOSH 或 TOSL 作为目标寄存器。

指令字数:

2

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f (源寄存器)	处理数据	空操作
译码	空操作 无假读	空操作	写寄存器 f (目标寄存器)

示例:

MOVFF REG1, REG2

执行指令前

REG1 = 33h

REG2 = 11h

执行指令后

REG1 = 33h

REG2 = 33h

MOVLB

将立即数送入 BSR 的低半字节

语法:

MOVLB k

操作数:

$0 \leq k \leq 255$

操作:

$k \rightarrow \text{BSR}$

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

MOVLW	将立即数送入 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 0x5A

 执行指令后

 W = 5Ah

MOVWF	将 W 的内容送入 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地址单元。			
	如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。			
	如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见 第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令” 。			
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

PIC18F46J50 系列

MULLW

将立即数与 W 中的内容相乘

语法:

MULLW k

操作数:

$0 \leq k \leq 255$

操作:

$(W) \times k \rightarrow \text{PRODH:PRODL}$

受影响的状态位:

无

机器码:

0000	1101	kkkk	kkkk

说明:

将 W 的内容与 8 位立即数 k 进行无符号的乘法运算。16 位的结果存储在 PRODH:PRODL 寄存器对中，其中 PRODH 用于存储高字节。

W 的内容不改变。

所有状态标志位都不受影响。

请注意此操作不可能发生溢出或进位。结果有可能为零，但不会被检测到。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写寄存器 PRODH: PRODL

示例:

MULLW 0xC4

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:

$0 \leq f \leq 255$
 $a \in [0,1]$

操作:

$(W) \times (f) \rightarrow \text{PRODH:PRODL}$

受影响的状态位:

无

机器码:

0000	001a	ffff	ffff

说明:

将 W 的内容与寄存器单元 f 的内容执行无符号的乘法运算。运算的 16 位结果保存在 PRODH:PRODL 寄存器对中，其中 PRODH 用于存储高字节。W 和 f 的内容都不改变。

所有状态标志位都不受影响。

请注意此操作不可能发生溢出或进位。结果有可能为零，但不会被检测到。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ (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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

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

NEGF

对 f 取补

语法:

NEGF f{,a}

操作数:

$0 \leq f \leq 255$
 $a \in [0,1]$

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:

用二进制补码对存储单元 f 取补, 结果存储在数据存储单元 f 中。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。

如果 a 为 0 且使能了扩展指令集, 只要 f ≤ 95 (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

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:	<table><tr><td>0000</td><td>0000</td><td>0000</td><td>0000</td></tr><tr><td>1111</td><td>xxxx</td><td>xxxx</td><td>xxxx</td></tr></table>				0000	0000	0000	0000	1111	xxxx	xxxx	xxxx
0000	0000	0000	0000									
1111	xxxx	xxxx	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。

PIC18F46J50 系列

POP

弹出返回堆栈栈顶的内容

语法:POP

操作数:无

操作:(TOS) → 位桶（即丢弃）

受影响的状态位:无

机器码:

0000	0000	0000	0110

说明:

从返回堆栈弹出 TOS 值并丢弃。然后，前一个压入返回堆栈的值成为 TOS 值。此指令可以让用户正确管理返回堆栈，从而实现软件堆栈。

指令字数:1

指令周期数:1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	弹出 TOS 值	空操作

示例:	POP		
	GOTO	NEW	
执行指令前			
TOS	=	0031A2h	
堆栈 (下一级)	=	014332h	
执行指令后			
TOS	=	014332h	
PC	=	NEW	

PUSH		将数据压入返回堆栈栈顶											
语法:	PUSH												
操作数:	无												
操作:	(PC + 2) → TOS												
受影响的状态位:	无												
机器码:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0101</td></tr></table>				0000	0000	0000	0101					
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:	<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>将 PC + 2 压入返回堆栈</td><td>空操作</td><td>空操作</td></tr></table>					Q1	Q2	Q3	Q4	译码	将 PC + 2 压入返回堆栈	空操作	空操作
Q1	Q2	Q3	Q4										
译码	将 PC + 2 压入返回堆栈	空操作	空操作										

示例:	PUSH		
执行指令前			
TOS	=	345Ah	
PC	=	0124h	
执行指令后			
PC	=	0126h	
TOS	=	0126h	
堆栈 (下一级)	=	345Ah	

RCALL

相对调用

语法:

RCALL n

操作数:

-1024 ≤ n ≤ 1023

操作:

(PC) + 2 → TOS,
(PC) + 2 + 2n → PC

受影响的状态位:

无

机器码:

1101	1nnn	nnnn	nnnn

说明:

从当前地址跳转（最多 1K）来调用子程序。首先，将返回地址（PC + 2）压入堆栈。然后，将“2n”（以二进制补码表示）与 PC 相加。由于 PC 将递增以便取出下一条指令，所以新地址将为 PC + 2 + 2n。该指令为一条双周期指令。

指令字数:

1

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 n 将 PC 压入堆栈	处理数据	写入 PC
空操作	空操作	空操作	空操作

示例: HERE RCALL Jump

执行指令前
PC = 地址（HERE）

执行指令后
PC = 地址（Jump）
TOS = 地址（HERE + 2）

RESET

复位

语法:

RESET

操作数:

无

操作:

将所有受 $\overline{\text{MCLR}}$ 复位影响的寄存器和标志位复位。

受影响的状态位:

全部

机器码:

0000	0000	1111	1111

说明:

此指令可实现用软件执行 $\overline{\text{MCLR}}$ 复位。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	开始复位	空操作	空操作

示例:

RESET

执行指令后
寄存器 = 复位值
标志位 * = 复位值

© 2009 - 2011 Microchip Technology Inc. DS39931D_CN 第 465 页

PIC18F46J50 系列

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, (STATUS) → 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:	<table border="1"><tr><td>0000</td><td>0000</td><td>0001</td><td>000s</td></tr></table>			0000	0000	0001	000s
0000	0000	0001	000s				
说明:	从中断返回。执行出栈操作，将栈顶（TOS）的内容装入 PC。通过将高或低优先级全局中断允许位置 1，来允许中断。如果 s = 1，则影子寄存器 WS、STATUS 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	= STATUS
GIE/GIEH, PEIE/GIEL	= 1

RETLW		将立即数返回到 W	
语法:	RETLW k		
操作数:	$0 \leq k \leq 255$		
操作:	$k \rightarrow W$, (TOS) $\rightarrow$ 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。高字节地址锁存器（PCLATH）内容保持不变。		
指令字数:	1		
指令周期数:	2		
Q 周期操作:			
Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	从堆栈弹出 PC 值, 写入 W
空操作	空操作	空操作	空操作

示例:	
	CALL TABLE ; W contains table ; offset value ; W now has ; table value
:	
TABLE	
ADDWF PCL	; W = offset
RETLW k0	; Begin table
RETLW k1	;
:	
:	
RETLW kn	; End of table
执行指令前	
W	= 07h
执行指令后	
W	= kn 的值

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,
(STATUS) → 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）的内容装入程序计数器。如果 s = 1，则影子寄存器 WS、STATUS 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	处理数据	从堆栈弹出 PC 值
空操作	空操作	空操作	空操作

示例: RETURN
 执行指令后:
 PC = TOS

RLCF

f 带进位循环左移

语法:

RLCF f{,d{,a}}

操作数:

0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作:

(f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

受影响的状态位:

C、N 和 Z

机器码:

0011	01da	ffff	ffff

说明:

将寄存器 f 的内容连同进位标志位一起循环左移 1 位。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

指令字数:

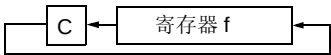
1

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

PIC18F46J50 系列

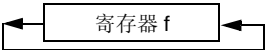
RLNCF	f 循环左移（不带进位）			
语法:	RLNCF f {,d {,a}}			
操作数:	$0 \leq f \leq 255$ $d \in [0,1]$ $a \in [0,1]$			
操作:	$(f<n>) \rightarrow dest<n + 1>$, $(f<7>) \rightarrow dest<0>$			
受影响的状态位:	N 和 Z			
机器码:	0100	01da	ffff	ffff

说明:

将寄存器 f 的内容循环左移 1 位。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入 目标寄存器

示例: RLNCF REG, 1, 0

执行指令前
REG = 1010 1011
执行指令后
REG = 0101 0111

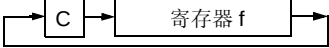
RRCF	f 带进位循环右移				
语法:	RRCF f{,d{,a}}				
操作数:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]				
操作:	(f<n>) → dest<n - 1>, (f<0>) → C, (C) → dest<7>				
受影响的状态位:	C、N 和 Z				
机器码:	<table><tr><td>0011</td><td>00da</td><td>ffff</td><td>ffff</td></tr></table>	0011	00da	ffff	ffff
0011	00da	ffff	ffff		

说明:

将寄存器 f 的内容连同进位标志位一起循环右移 1 位。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。



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

RRNCF

f 循环右移（不带进位）

语法:

RRNCF f{,d{,a}}

操作数:

0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作:

(f<n>) → dest<n - 1>,
(f<0>) → dest<7>

受影响的状态位:

N 和 Z

机器码:

0100	00da	ffff	ffff

说明:

将寄存器 f 的内容循环右移 1 位。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果 a 为 0，选择快速操作存储区，忽略 BSR 的值。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

寄存器 f

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

SETF

将 f 的内容置为全 1

语法:

SETF f{,a}

操作数:

0 ≤ f ≤ 255
a ∈ [0,1]

操作:

FFh → f

受影响的状态位:

无

机器码:

0110	100a	ffff	ffff

说明:

将指定寄存器的内容置为 FFh。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

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

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

PIC18F46J50 系列

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:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0011</td></tr></table>			0000	0000	0000	0011
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, 则该位将被清零。

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:

$(W) - (f) - (\overline{C}) \rightarrow \text{dest}$

受影响的状态位:

N、OV、C、DC 和 Z

机器码:

0101	01da	ffff	ffff

说明:

将 W 的内容减去 f 寄存器的内容和进位标志位（借位）（通过二进制补码方式进行运算）。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

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
N = 0 ; 结果为全零

SUBLW

立即数减去 W 的内容

语法:

SUBLW k

操作数:

$0 \leq k \leq 255$

操作:

$k - (W) \rightarrow W$

受影响的状态位:

N、OV、C、DC 和 Z

机器码:

0000	1000	kkkk	kkkk

说明:

用 8 位立即数 k 减去 W。结果存储在 W 寄存器中。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

例 1:

SUBLW 0x02

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 0x02

执行指令前

W = 02h

C = ?

执行指令后

W = 00h

C = 1 ; 结果为全零

Z = 1

N = 0

例 3:

SUBLW 0x02

执行指令前

W = 03h

C = ?

执行指令后

W = FFh ; (二进制补码)

C = 0 ; 结果为负

Z = 0

N = 1

SUBWF

f 减去 W

语法:

SUBWF f{,d{,a}}

操作数:

$0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作:

$(f) - (W) \rightarrow \text{dest}$

受影响的状态位:

N、OV、C、DC 和 Z

机器码:

0101	11da	ffff	ffff

说明:

用寄存器 f 中的内容减去 W 寄存器的内容 (通过二进制补码方式进行运算)。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区 (默认)。

如果 a 为 0 且使能了扩展指令集, 只要 $f \leq 95$ (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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。

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 ; 结果为全零

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

PIC18F46J50 系列

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) − (C̄) → dest										
受影响的状态位：	N、OV、C、DC 和 Z										
机器码：	<table border="1"><tr><td>0101</td><td>10da</td><td>ffff</td><td>ffff</td></tr></table>			0101	10da	ffff	ffff				
0101	10da	ffff	ffff								
说明：	用 f 寄存器的内容减去 W 的内容和进位标志位（借位）（通过二进制补码方式进行运算）。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。 如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。 如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。										
指令字数：	1										
指令周期数：	1										
Q 周期操作：	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>译码</td><td>读寄存器 f</td><td>处理数据</td><td>写入目标寄存器</td></tr></table>			Q1	Q2	Q3	Q4	译码	读寄存器 f	处理数据	写入目标寄存器
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 ; 结果为全零
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>) → dest<7:4>, (f<7:4>) → dest<3:0>						
受影响的状态位：	无						
机器码：	<table border="1"><tr><td>0011</td><td>10da</td><td>ffff</td><td>ffff</td></tr></table>			0011	10da	ffff	ffff
0011	10da	ffff	ffff				
说明：	寄存器 f 的高半字节和低半字节相互交换。如果 d 为 0，结果存储在 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 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”。						
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

TBLRD		表读					
语法:	TBLRD (*; *+; *-; +*)						
操作数:	无						
操作:	如果执行 TBLRD *, (程序存储单元 (TBLPTR)) → TABLAT, TBLPTR 不改变; 如果执行 TBLRD *+, (程序存储单元 (TBLPTR)) → TABLAT, (TBLPTR) + 1 → TBLPTR ; 如果执行 TBLRD *-, (程序存储单元 (TBLPTR)) → TABLAT, (TBLPTR) - 1 → TBLPTR ; 如果执行 TBLRD +*, (TBLPTR) + 1 → TBLPTR, (程序存储单元 (TBLPTR)) → TABLAT						
受影响的状态位:	无						
机器码:	<table><tr><td>0000</td><td>0000</td><td>0000</td><td>10nn nn=0 * =1 *+ =2 *- =3 +*</td></tr></table>			0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*
0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*				
说明:	该指令用于读取程序存储单元（P.M.）的内容。使用表指针（TBLPTR）对程序存储单元进行寻址。 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: 程序存储字的低有效字节 TBLPTR<0> = 1: 程序存储字的高有效字节 TBLRD 指令可用如下方法修改 TBLPTR 的值: • 不变 • 后递增 • 后递减 • 预递增						
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

PIC18F46J50 系列

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:	<table><tr><td>0000</td><td>0000</td><td>0000</td><td>11nn nn=0 * =1 *+ =2 *- =3 +*</td></tr></table>			0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*								
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单元 (P.M.) 的内容编程。(关于对闪存程序存储器编程的更多详细信息, 请参见第 6.0 节“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 选择要访问程序存储单元的哪个字节。 TBLPTR<0> = 0: 程序存储字的低有效字节 TBLPTR<0> = 1: 程序存储字的高有效字节 TBLWT 指令可用如下方法修改 TBLPTR 的值: • 不变 • 后递增 • 后递减 • 预递增														
指令字数:	1														
指令周期数:	2														
Q 周期操作:	<table><tr><td>Q1</td><td>Q2</td><td>Q3</td><td>Q4</td></tr><tr><td>译码</td><td>空操作</td><td>空操作</td><td>空操作</td></tr><tr><td>空操作</td><td>空操作 (读 TABLAT)</td><td>空操作</td><td>空操作 (写保持 寄存器)</td></tr></table>			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)
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

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 = 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，丢弃执行当前指令期间已取的下一条指令并执行一条 NOP 指令，使该指令成为双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

指令字数：

1

指令周期数：

1（2）
注：如果跳过，且后面跟有 2 字指令，则执行 TSTFSZ 需要 3 个周期。

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过，且后面跟有 2 字指令：

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例：

HERE

TSTFSZ

CNT, 1

NZERO

:

ZERO

:

执行指令前

PC

=

地址（HERE）

执行指令后

如果 CNT

=

00h.

PC

=

地址（ZERO）

如果 CNT

≠

00h.

PC

=

地址（NZERO）

XORLW

将立即数与 W 作逻辑异或运算

语法：

XORLW k

操作数：

0 ≤ k ≤ 255

操作：

(W) .XOR. k → W

受影响的状态位：

N 和 Z

机器码：

0000	1010	kkkk	kkkk

说明：

将 W 的内容与 8 位立即数 k 进行逻辑异或运算。结果存储在 W 寄存器中。

指令字数：

1

指令周期数：

1

Q 周期操作：

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例：

XORLW

0xAF

执行指令前

W

=

B5h

执行指令后

W

=

1Ah

PIC18F46J50 系列

XORWF

将 W 与 f 作逻辑异或运算

语法:

XORWF f{,d{,a}}

操作数:

0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

操作:

(W) .XOR.(f) → dest

受影响的状态位:

N 和 Z

机器码:

0001	10da	ffff	ffff

说明:

将 W 的内容与寄存器 f 的内容进行逻辑异或运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f（默认）。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区（默认）。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见[第 28.2.3 节“立即数变址寻址模式中针对字节和针对位的指令”](#)。

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

28.2 扩展指令集

除了PIC18指令集的75条标准指令之外，PIC18F46J50系列器件还提供了针对核心CPU功能的可选扩展指令。这些新增的功能包括8条额外的指令，增加了间接和变址寻址操作，并使得许多标准PIC18指令可以实现立即数变址寻址。

扩展指令集的额外功能在默认情况下对未编程器件是使能的。用户必须在编程期间将XINST配置位正确置1或清零，从而使能或禁止这些功能。

扩展指令集中的指令可以全部被归为立即数操作类指令，它们既可以对文件选择寄存器（FSR）进行操作，也可以使用文件选择寄存器进行变址寻址。还为其中两条指令ADDFSR和SUBFSR提供了使用FSR2的特例形式，即ADDULNK和SUBULNK，这两条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语言编写的可重入程序代码（也就是递归或使用软件堆栈的代码）。此外，它们使用户能更高效地用高级语言对数据结构执行某些操作。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
- 函数指针调用
- 对软件堆栈指针进行操作
- 对软件堆栈中的变量进行操作

表 28-3 提供了扩展指令集中的指令汇总。第 28.2.2 节“扩展指令集”对这些指令进行了详细说明。表 28-1（第 436 页）提供了标准和扩展的 PIC18 指令集的操作码字段说明。

注： 扩展指令集和立即数变址寻址模式是专为优化用 C 语言编写的应用程序而设计的，用户可能不会在汇编程序中直接使用这些指令。对于那些需要查看编译器生成代码的用户，这些命令的语法可作为参考。

28.2.1 扩展指令的语法

大部分扩展指令都使用变址参数，使用一个 FSR 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。这用于表示此参数用作变址或偏移量。如果 MPASM™ 汇编器发现一个变址或偏移量没有被括起来，它就会给出出错信息。

当使能扩展指令集时，方括号也用于表示针对字节和针对位的指令中的变址参数。这是对指令语法的额外更改。更多详细信息，请参见第 28.2.3.1 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，可选参数将用大括号 “[]” 表示。

表 28-3: PIC18 指令集的扩展

助记符, 操作数	说明	周期数	16 位指令字				受影响的状态位
			MSb		LSb		
ADDFSR f, k	将立即数加到 FSR	1	1110	1000	ffkk	kkkk	无
ADDULNK k	将立即数加到 FSR2 并返回	2	1110	1000	11kk	kkkk	无
CALLW	使用 WREG 调用子程序	2	0000	0000	0001	0100	无
MOVSF Z _S , f _d	将 Z _S (源) 移入 (第一个字) f _d (目标) (第二个字)	2	1110	1011	0zzz	zzzz	无
MOVSS Z _S , Z _d	将 Z _S (源) 移入 (第一个字) Z _d (目标) (第二个字)	2	1110	1011	1zzz	zzzz	无
PUSHL k	将立即数保存到 FSR2, FSR2 递减 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

PIC18F46J50 系列

28.2.2 扩展指令集

ADDFSR		将立即数加到 FSR		
语法:	ADDFSR f, k			
操作数:	$0 \leq k \leq 63$ $f \in [0, 1, 2]$			
操作:	$FSR(f) + k \rightarrow FSR(f)$			
受影响的状态位:	无			
机器码:	1110	1000	ffkk	kkkk
说明:	将 6 位立即数 k 加到由 f 指定的 FSR 的内容。			
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, 0x23
执行指令前	
FSR2	= 03FFh
执行指令后	
FSR2	= 0422h

ADDULNK		将立即数加到 FSR2 并返回	
语法:	ADDULNK k		
操作数:	0 ≤ k ≤ 63		
操作:	FSR2 + k → FSR2, (TOS) → PC		
受影响的状态位:	无		
机器码:	1110	1000	11kk kkkk
说明:	将 6 位立即数 k 加到 FSR2 的内容。 然后通过将 TOS 的值装入 PC，执行 RETURN。 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 0x23
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，那么指令格式将变为：{label} 指令参数。

CALLW

使用 WREG 调用子程序

语法:

CALLW

操作数:

无

操作:

(PC + 2) → TOS,
(W) → PCL,
(PCLATH) → PCH,
(PCLATU) → PCU

受影响的状态位:

无

机器码:

0000	0000	0001	0100

说明

首先，返回地址（PC + 2）被压入返回堆栈。接下来，将 W 寄存器的内容写入 PCL，PCL 现有的值被丢弃。然后，PCLATH 和 PCLATU 的内容被分别锁存到 PCH 和 PCU。第二个周期执行一条 NOP 指令，并同时取下一条新指令。

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送入 f

语法:

MOVSF [z_s], f_d

操作数:

0 ≤ z_s ≤ 127
0 ≤ f_d ≤ 4095

操作:

((FSR2) + z_s) → f_d

受影响的状态位:

无

机器码:

1110	1011	0zzz	zzzz _s
1111	ffff	ffff	ffff _d

说明:

将源寄存器的内容送入目标寄存器 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）中的任何存储单元。

MOVSF 指令中的目标寄存器不能是 PCL、TOSU、TOSH 或 TOSL。

如果计算得到的源地址指向间接寻址寄存器，将返回 00h。

指令字数:

2

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	读源寄存器
译码	空操作 无假读	空操作	写寄存器 f (目标寄存器)

示例:

MOVSF

[0x05], REG2

执行指令前

FSR2

=

80h

85h 单元的内容

=

33h

REG2

=

11h

执行指令后

FSR2

=

80h

85h 单元的内容

=

33h

REG2

=

33h

PIC18F46J50 系列

MOVSS

在变址寻址单元之间传送数据

语法:

MOVSS [z_s], [z_d]

操作数:

0 ≤ z_s ≤ 127
0 ≤ z_d ≤ 127

操作:

((FSR2) + z_s) → ((FSR2) + z_d)

受影响的状态位:

无

机器码:

1110	1011	1zzz	zzzz _s
1111	xxxx	xzzz	zzzz _d

第一个字 (源)

第二个字 (目标)

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 和 z_d 来确定源寄存器和目标寄存器的地址。两个寄存器都可以是 4096 字节数据存储空间 (000h 到 FFFh) 中的任意存储单元。

MOVSS 指令不能使用 PCL、TOSU、TOSH 或 TOSL 作为目标寄存器。

如果计算得到的源地址指向间接寻址寄存器, 将返回 00h。如果计算得到的目标地址指向间接寻址寄存器, 将执行一条 NOP 指令。

指令字数:

2

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	读源寄存器
译码	确定目标地址	确定目标地址	写目标寄存器

示例:

MOVSS [0x05], [0x06]

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

将立即数保存到 FSR2, FSR2 递减 1

语法:

PUSHL k

操作数:

0 ≤ k ≤ 255

操作:

k → (FSR2),
FSR2 − 1 → FSR2

受影响的状态位:

无

机器码:

1110	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 递减 1。

此指令允许用户将值压入软件堆栈。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写入目标寄存器

示例:

PUSHL 0x08

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

SUBFSR

FSR 减去立即数

语法:

SUBFSR f, k

操作数:

0 ≤ k ≤ 63
f ∈ [0, 1, 2]

操作:

FSRf – k → FSRf

受影响的状态位:

无

机器码:

1110	1001	ffkk	kkkk

说明:

用 f 指定的 FSR 的内容减去 6 位立即数 k。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例:

SUBFSR 2, 0x23

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 ≤ k ≤ 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 FSR 的内容减去 6 位立即数 k，然后将 TOS 的值装入 PC，执行 RETURN。

执行该指令需要两个指令周期，第二个指令周期执行一条 NOP 指令。

该指令可以被认为是 SUBFSR 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 0x23

执行指令前

FSR2 = 03FFh

PC = 0100h

执行指令后

FSR2 = 03DCh

PC = (TOS)

PIC18F46J50 系列

28.2.3 立即数变址寻址模式中针对字节和针对位的指令

注： 使能 PIC18 扩展指令集可能导致常规应用程序运行不正常或完全失败。

一旦使能扩展指令集，除了可以使用扩展指令集中的8条新命令之外，还将使能立即数变址寻址模式（[第 6.6.1 节“使用立即数偏移量进行变址寻址”](#)）。这将导致标准 PIC18 指令集中大部分指令的地址解析方法有很大变化。

当禁止扩展指令集时，嵌入在操作码中的地址被视为立即数存储单元：可以是快速操作存储区中的存储单元（ $a = 0$ ），或由 BSR 指定的 GPR 存储区中的存储单元（ $a = 1$ ）。当使能扩展指令集且 $a = 0$ 时，地址小于或等于 5Fh 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，即所有针对字节或针对位的指令，或者几乎半数的核心 PIC18 指令，在使能了扩展指令集时操作都会有所不同。

当 FSR2 的内容为 00h 时，快速操作 RAM 的边界会被重新映射到它们的原始值。这对于编写向下兼容的代码很有用处。如果使用此技术，有必要在 C 程序调用汇编子程序时保存 FSR2 的值并在返回时将它恢复，这样做的目的是保护堆栈指针。用户还必须记住扩展指令集的语法要求（见 [第 28.2.3.1 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操作很有用处，但是如果不小心对错误的寄存器进行了简单的算术运算也会非常麻烦。已经习惯使用 PIC18 编程的用户必须记住，在使能了扩展指令集后，地址小于或等于 5Fh 的寄存器用于立即数变址寻址。

下页提供了在立即数变址寻址模式中，一些针对字节和位的指令的代表示例，通过这些示例可以看出指令执行如何受到影响。示例中的操作数条件适用于所有这些类型的指令。

28.2.3.1 标准 PIC18 命令的扩展指令语法

当使能了扩展指令集时，立即数偏移量 “k” 被用来替换标准的针对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，偏移量必须用方括号 “[]” 标出。因为在扩展指令集中，编译器将方括号中的值解析为变址地址或偏移量。省略方括号，或在方括号内使用大于 5Fh 的值会在 MPASM 汇编器中产生错误。

如果变址参数已被正确加上了方括号，那么就不再需要指定快速操作 RAM 参数；此参数被自动假定为 0。这与标准操作（禁止扩展指令集时）刚好相反，在标准操作中，“a” 基于目标地址被置 1。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。

28.2.4 使能扩展指令集时的注意事项

需要注意的是，并非所有用户都有必要使用扩展指令集，尤其是那些不使用软件堆栈的用户。

此外，立即数变址寻址模式可能会给写入 PIC18 汇编器的常规应用程序带来问题。这是因为常规的指令会尝试寻址快速操作存储区中地址低于 5Fh 的寄存器。当使能了扩展指令集时，这些地址被解析为相对于 FSR2 的立即数偏移量，所以应用程序可能会读或写错误的地址。

将应用程序移植到 PIC18F46J50 系列器件时，考虑代码的类型是非常重要的。在使用扩展指令集时，用 C 语言编写的代码较长的可重入应用程序会运行得很好，而大量使用快速操作存储区的常规应用程序不会获得任何益处。

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 ≤ k ≤ 95 d ∈ [0,1]			
操作:	(W) + ((FSR2) + k) → dest			
受影响的状态位:	N、OV、C、DC 和 Z			
机器码:	0010	01d0	kkkk	kkkk
说明:	将 W 的内容与由 FSR2 加上偏移量 k 指定的寄存器的内容相加。 如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (默认)。			
指令字数:	1			
指令周期数:	1			
Q 周期操作:				
	Q1	Q2	Q3	Q4
	译码	读取 k	处理数据	写入目标寄存器

示例: ADDWF [0FST], 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 ≤ f ≤ 95 0 ≤ b ≤ 7			
操作:	1 → ((FSR2) + k)			
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 ≤ k ≤ 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 [0FST]

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

PIC18F46J50 系列

28.2.5 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 PIC18F46J50 系列器件的扩展指令集。软件工具包括 MPLAB C18 C 编译器、MPASM 汇编器和 MPLAB 集成开发环境 (Integrated Development Environment, IDE)。

在选择了目标器件进行软件开发后，MPLAB IDE 将自动按默认模式设置该器件的配置位。XINST 配置位的默认设置是 1，使能扩展指令集和立即数变址寻址模式。在编程时必须将 XINST 位置 1 才能确保使用扩展指令集开发的应用程序能够正确执行。

要使用扩展指令集开发软件，用户必须设置他们的语言工具以实现对扩展指令和变址寻址模式的支持。根据所使用的环境，可以通过以下几种方法：

- 开发环境中的菜单选项或对话框，允许用户配置项目的语言工具及其设置
- 命令行选项
- 源代码中的伪指令

这些选项在不同的编译器、汇编器和开发环境中将有所不同。建议用户在其开发系统所附带的文档中查询相应的文档。

29.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机和 dsPIC® 数字信号控制器提供支持：

- 集成开发环境
 - MPLAB® IDE 软件
- 编译器 / 汇编器 / 链接器
 - 适用于各种器件系列的 MPLAB C 编译器
 - 适用于各种器件系列的 HI-TECH C 编译器
 - MPASM™ 汇编器
 - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
 - 适用于各种器件系列的 MPLAB 汇编器 / 链接器 / 库管理器
- 模拟器
 - MPLAB SIM 软件模拟器
- 仿真器
 - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
 - MPLAB ICD 3
 - PICkit™ 3 Debug Express
- 器件编程器
 - PICkit™ 2 编程器
 - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包

29.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16/32 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）
 - 在线仿真器（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
- 鼠标停留在变量上进行查看的功能
- 将变量从源代码窗口拖放到 Watch（观察）窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件（C 语言或汇编语言）
- 点击一次即可完成编译或汇编，并将代码下载到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（C 语言或汇编语言）
 - 混合 C 语言和汇编语言
 - 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强大的工具时的学习时间。

PIC18F46J50 系列

29.2 适用于各种器件系列的 MPLAB C 编译器

MPLAB C 编译器代码开发系统是完整的 ANSI C 编译器，适用于 Microchip 的 PIC18、PIC24 和 PIC32 系列单片机及 dsPIC30 和 dsPIC33 系列数字信号控制器。这些编译器提供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，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

29.3 适用于各种器件系列的 HI-TECH C 编译器

HI-TECH C 编译器代码开发系统是完整的 ANSI C 编译器，适用于 Microchip 的 PIC 系列单片机及 dsPIC 系列数字信号控制器。这些编译器提供强大的集成功能和全知代码生成能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

编译器包括一个宏汇编器、链接器、预处理程序和单步驱动程序，可以在多种平台上运行。

29.4 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于 PIC10/12/16/18 MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特性：

- 集成在 MPLAB 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

29.5 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程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

29.6 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB 汇编器为 PIC24、PIC32 和 dsPIC 器件从符号汇编语言生成可重定位机器码。MPLAB 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性：

- 支持整个器件指令集
- 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

29.7 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC[®] DSC 进行模拟，可在 PC 主机环境下进行代码开发。对于任何给定的指令，都可以对数据区进行检查或修改，并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中，以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

29.8 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[®] 闪存 DSC 进行调试和编程。IDE 是随每个工具包一起提供的。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器和 (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，对该仿真器进行现场升级。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。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：低成本、全速仿真、运行时变量查看、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。

29.9 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 闪存数字信号控制器 (DSC) 和单片机 (MCU) 器件。结合 MPLAB 集成开发环境 (IDE) 所具有的功能强大但易于使用的图形用户界面，该调试器可对 PIC[®] 闪存单片机和 dsPIC[®] DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器和 (RJ-11) 与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 转接器。

29.10 PICkit 3 在线调试器 / 编程器及 PICkit 3 Debug Express

结合 MPLAB 集成开发环境 (IDE) 所具有的功能强大的图形用户界面，MPLAB PICkit 3 可对 PIC[®] 闪存单片机和 dsPIC[®] 数字信号控制器进行调试和编程，且价位较低。MPLAB PICkit 3 通过全速 USB 接口与设计工程师的 PC 相连，并利用 Microchip 调试 (RJ-11) 连接器（与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容）与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程。

PICkit 3 Debug Express 包括 PICkit 3、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器 and MPLAB IDE 软件）。

PIC18F46J50 系列

29.11 PICKit 2 开发编程器 / 调试器及 PICKit 2 Debug Express

PICKit™ 2 开发编程器 / 调试器是一款低成本开发工具，具有易于使用的界面，适用于对 Microchip 的闪存系列单片机进行编程和调试。这一全功能的 Windows® 编程界面支持低档（PIC10F、PIC12F5xx 和 PIC16F5xx）、中档（PIC12F6xx 和 PIC16F）、PIC18F、PIC24、dsPIC30、dsPIC33 和 PIC32 系列的 8 位、16 位及 32 位单片机，以及许多 Microchip 串行 EEPROM 产品。结合 Microchip 功能强大的 MPLAB 集成开发环境（IDE），PICKit 2 可对大多数 PIC® 单片机进行在线调试。即使 PIC 单片机已嵌入应用，在线调试功能仍可以运行、暂停和单步执行程序。在断点处暂停时，可以检查和修改文件寄存器。

PICKit 2 Debug Express 包括 PICKit 2、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器 and MPLAB IDE 软件）。

29.12 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器（128 x 64），以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡，用于文件存储及数据应用。

29.13 演示 / 开发板、评估工具包及入门工具包

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

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

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

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

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

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

30.0 电气特性

绝对最大值 †)

环境温度.....	-40°C 至 +125°C
储存温度.....	-65°C 至 +150°C
任一仅用作数字功能的 I/O 引脚或 $\overline{\text{MCLR}}$ 相对于 VSS 的电压（当 VDD ≥ 2.0V 时）.....	-0.3V 至 6.0V
任一仅用作数字功能的 I/O 引脚或 $\overline{\text{MCLR}}$ 相对于 VSS 的电压（当 VDD < 2.0V 时）.....	-0.3V 至 (VDD + 4.0V)
任一数模组合引脚相对于 VSS 的电压（VDD 除外）.....	-0.3V 至 (VDD + 0.3V)
VDDCORE 引脚相对于 VSS 的电压.....	-0.3V 至 2.75V
VDD 引脚相对于 VSS 的电压.....	-0.3V 至 4.0V
VUSB 引脚相对于 VSS 的电压.....	(VDD - 0.3V) 至 4.0V
总功耗（注 1）.....	1.0W
流出 VSS 引脚的最大电流.....	300 mA
流入 VDD 引脚的最大电流.....	250 mA
任一 PORTB、PORTC 和 RA6 I/O 引脚的最大输出灌电流.....	25 mA
任一 PORTA（RA6 除外）、PORTD 和 PORTE I/O 引脚的最大输出灌电流.....	4 mA
任一 PORTB、PORTC 和 RA6 I/O 引脚的最大输出拉电流.....	25 mA
任一 PORTA（RA6 除外）、PORTD 和 PORTE I/O 引脚的最大输出拉电流.....	4 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：
$$\text{PDIS} = \text{VDD} \times \{\text{IDD} - \sum \text{IOH}\} + \sum \{(\text{VDD} - \text{VOH}) \times \text{IOH}\} + \sum (\text{VOL} \times \text{IOL})$$

† 注：如果运行条件超过了上述“绝对最大值”，即可能对器件造成永久性损坏。这仅是极限参数，我们不建议器件工作在极限值甚至超过上述极限值。器件长时间工作在最大值条件下，其稳定性可能受到影响。

PIC18F46J50 系列

图 30-1: PIC18F46J50 系列 VDD 与频率关系图（工业级）

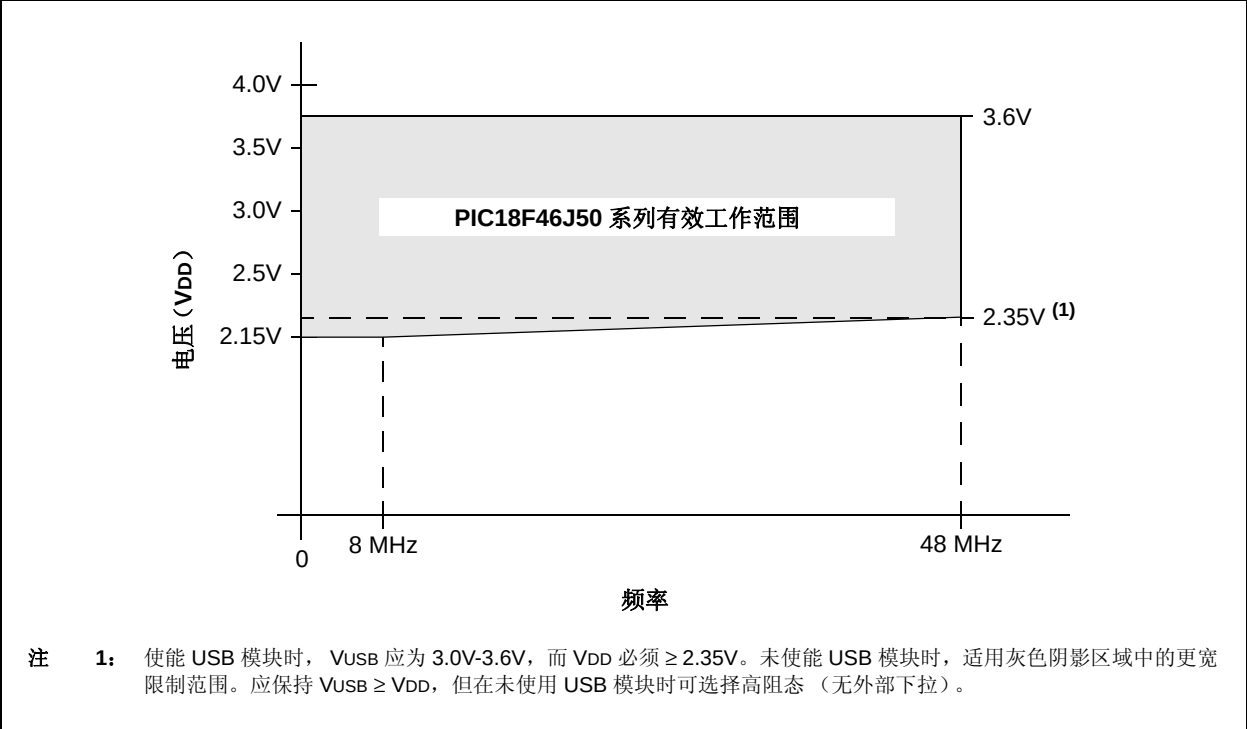
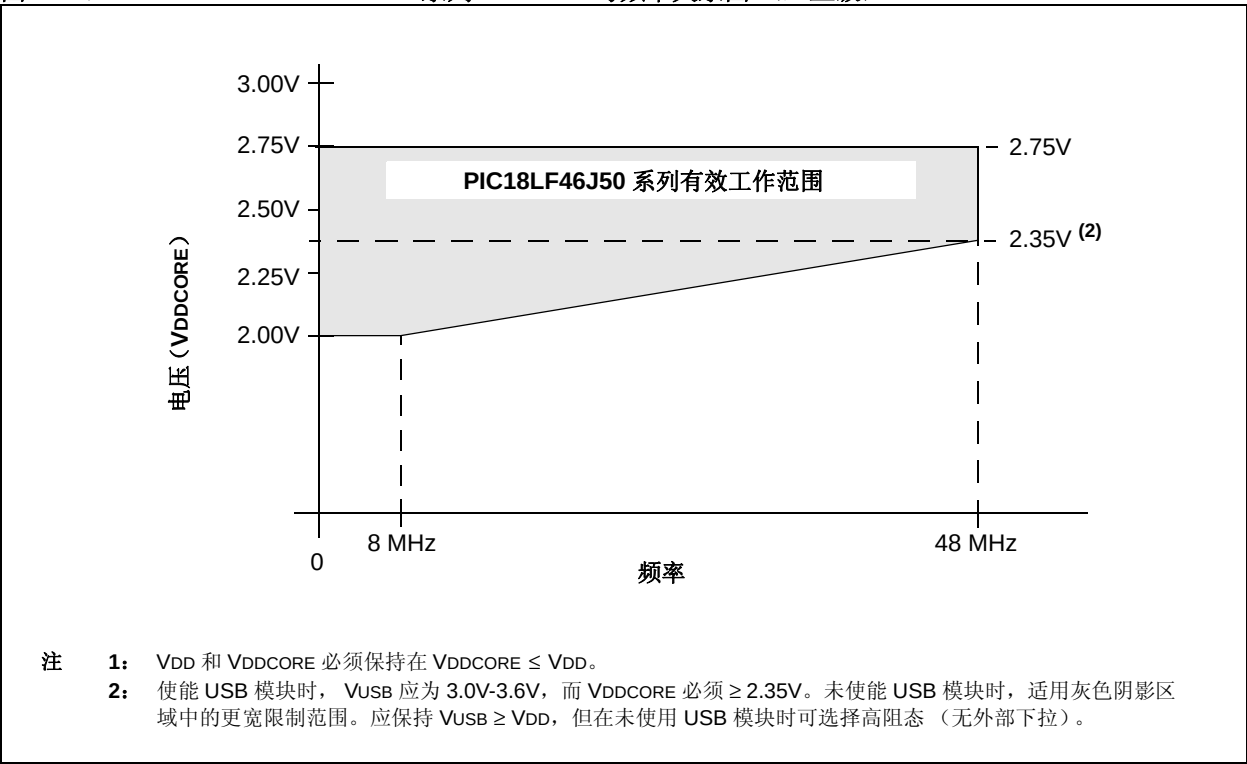


图 30-2: PIC18LF46J50 系列 VDDCORE 与频率关系图（工业级）⁽¹⁾



30.1 直流特性: PIC18F46J50 系列供电电压（工业级）

PIC18F46J50 系列			标准工作条件（除非另外声明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ （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	2.15	—	3.6	V	PIC18F4XJ50 和 PIC18F2XJ50
D001A	VDD	供电电压	2.0	—	3.6	V	PIC18LF4XJ50 和 PIC18LF2XJ50
D001B	VDDCORE	单片机内核的外部电源	2.0	—	2.75	V	PIC18LF4XJ50 和 PIC18LF2XJ50
D001C	AVDD	模拟供电电压	$V_{DD} - 0.3$	—	$V_{DD} + 0.3$	V	
D001D	AVSS	模拟地电位	$V_{SS} - 0.3$	—	$V_{SS} + 0.3$	V	
D001E	VUSB	USB 供电电压	3.0	3.3	3.6	V	USB 模块使能 ⁽²⁾
D002	VDR	RAM 数据保持电压 ⁽¹⁾	1.5	—	—	V	
D003	VPOR	确保内部上电复位信号的 VDD 启动电压	—	—	0.7	V	详情请参见第 5.3 节“上电复位 (POR)”
D004	SVDD	确保内部上电复位信号的 VDD 上升速率	0.05	—	—	V/ms	详情请参见第 5.3 节“上电复位 (POR)”
D005	VBOR ⁽³⁾	VDDCORE 欠压复位电压	1.9	2.0	2.2	V	仅限 PIC18F4XJ50 和 PIC18F2XJ50
D006	VDSBOR	VDD 欠压复位电压	—	1.8	—	V	“LF” 器件上的 DSBOR = 1 或处于深度休眠模式的“F” 器件

- 注 1: 这是在不丢失 RAM 数据的前提下，休眠模式或器件复位期间 VDDCORE 所能降到的最小电压值。
- 2: 应始终保持 $V_{USB} \geq V_{DD}$ ，但在 USB 模块被禁止时可能会悬空（高阻态，无外部下拉），RC4/RC5 将不会用作通用输入。
- 3: 在发生欠压复位之前，器件将正常工作，即使 V_{DD} 可能低于 V_{DDMIN} 。

PIC18F46J50 系列

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
PIC18F46J50 系列		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⁾ —— 休眠模式						
	PIC18LFXXJ50	0.01	1.4	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	休眠模式， REGSLP = 1
		0.06	1.4	μA	+25°C		
		0.52	6.0	μA	+60°C		
		1.8	10.2	μA	+85°C		
	PIC18LFXXJ50	0.035	1.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		0.13	1.5	μA	+25°C		
		0.63	8.0	μA	+60°C		
		2.2	12.6	μA	+85°C		
	PIC18FXXJ50	2.4	6.0	μA	-40°C	VDD = 2.15V Vddcore = 10 μF 电容	
		3.0	6.0	μA	+25°C		
		3.8	8.0	μA	+60°C		
		5.6	16	μA	+85°C		
	PIC18FXXJ50	3.5	7.0	μA	-40°C	VDD = 3.3V Vddcore = 10 μF 电容	
		3.2	7.0	μA	+25°C		
		4.2	10	μA	+60°C		
		6.4	19	μA	+85°C		
	掉电电流（IPD） ⁽¹⁾ —— 深度休眠模式						
	PIC18FXXJ50	1	25	nA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	深度休眠模式
		15	100	nA	+25°C		
		115	250	nA	+60°C		
		0.46	1.0	μA	+85°C		
	PIC18FXXJ50	3	50	nA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容	
		33	150	nA	+25°C		
		191	389	nA	+60°C		
		0.65	2.0	μA	+85°C		

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 VDD/VSS；
MCLR = VDD；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 +70°C。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1，SUSPND = 1，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度-40°C ≤ Ta ≤ +85°C（工业级）						
PIC18F46J50 系列		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)							
	PIC18LFXXJ50	5.3	14.2	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 31 kHz (RC_RUN 模式, 内部 RC 振荡器, INTSRC = 0)	
		6.2	14.2	μA	+25°C			
		8.5	19.0	μA	+85°C			
	PIC18LFXXJ50	8.0	16.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V		
		8.7	16.5	μA	+25°C			
		11.3	22.4	μA	+85°C			
	PIC18FXXJ50	37	77	μA	-40°C	VDD = 2.15V VDDCORE = 10 μF 电容		
		48	77	μA	+25°C			
		60	93	μA	+85°C			
	PIC18FXXJ50	45	84	μA	-40°C	VDD = 3.3V VDDCORE = 10 μF 电容		
		54	84	μA	+25°C			
		65	108	μA	+85°C			
	PIC18LFXXJ50	1.1	1.5	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0		FOSC = 4 MHz, RC_RUN 模式, 内部 RC 振荡器
		1.1	1.5	mA	+25°C			
		1.2	1.6	mA	+85°C			
	PIC18LFXXJ50	1.5	1.7	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V		
		1.6	1.7	mA	+25°C			
1.6		1.9	mA	+85°C				
PIC18FXXJ50	1.3	2.6	mA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容			
	1.4	2.6	mA	+25°C				
	1.4	2.8	mA	+85°C				
PIC18FXXJ50	1.6	2.9	mA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容			
	1.6	2.9	mA	+25°C				
	1.6	3.0	mA	+85°C				

- 注 1: 在休眠模式下, 掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 V_{DD} 或 V_{SS} , 禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素, 如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下, 所有 I_{DD} 测量值的测试条件为:
 $\text{OSC1} = \text{外部方波, 轨到轨满幅; 所有 I/O 引脚均为三态, 拉至 } V_{DD}/V_{SS};$
 $\text{MCLR} = V_{DD};$ 除非另外指定, 否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 $+70^{\circ}\text{C}$ 。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块, 使用 48 MHz 的时钟, 但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时, 消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1, SUSPND = 1, 总线处于空闲状态）, USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息, 集成上拉电阻使用“电阻切换”技术, 因此在空闲条件下, 电阻可低至 900 Ω 。

PIC18F46J50 系列

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）						
PIC18F46J50 系列		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)							
	PIC18LFXXJ50	1.9	3.6	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 8 MHz, RC_RUN 模式, 内部 RC 振荡器	
		2.0	3.8	mA	+25°C			
		2.0	3.8	mA	+85°C			
	PIC18LFXXJ50	2.8	4.8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V		
		2.8	4.8	mA	+25°C			
		2.8	4.9	mA	+85°C			
	PIC18FXXJ50	2.3	4.2	mA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容		
		2.3	4.2	mA	+25°C			
		2.4	4.5	mA	+85°C			
	PIC18FXXJ50	2.8	5.1	mA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容		
		2.8	5.1	mA	+25°C			
		2.8	5.4	mA	+85°C			
	PIC18LFXXJ50	1.9	9.4	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V		FOSC = 31 kHz, RC_IDLE 模式, 内部 RC 振荡器, INTSRC = 0
		2.3	9.4	μA	+25°C			
		4.5	17.2	μA	+85°C			
	PIC18LFXXJ50	2.4	10.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V		
		2.8	10.5	μA	+25°C			
5.4		19.5	μA	+85°C				
PIC18FXXJ50	33.3	75	μA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容			
	43.8	75	μA	+25°C				
	55.3	92	μA	+85°C				
PIC18FXXJ50	36.1	82	μA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容			
	44.5	82	μA	+25°C				
	56.3	105	μA	+85°C				

- 注 1: 在休眠模式下, 掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 V_{DD} 或 V_{SS} , 禁止所有会带来新增电流的功能部件 (如 WDT、Timer1 振荡器和 BOR 等) 时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素, 如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件 (USB 模块和 WDT 等)。有效工作模式下, 所有 I_{DD} 测量值的测试条件为:
 $\text{OSC1} =$ 外部方波, 轨到轨满幅; 所有 I/O 引脚均为三态, 拉至 V_{DD}/V_{SS} ;
 $\text{MCLR} = V_{DD}$; 除非另外指定, 否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 $+70^{\circ}\text{C}$ 。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块, 使用 48 MHz 的时钟, 但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时, 消耗的电流可能会高得多 (见第 22.6.4 节 “USB 收发器的电流消耗”)。在 USB 挂起模式下 ($\text{USBEN} = 1$, $\text{SUSPND} = 1$, 总线处于空闲状态), USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据 “USB 2.0 规范” 的 resistor_ecn 补充信息, 集成上拉电阻使用 “电阻切换” 技术, 因此在空闲条件下, 电阻可低至 900Ω 。

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
PIC18F46J50 系列		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)						
	PIC18LFXXJ50	0.531	0.980	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 4 MHz, RC_IDLE 模式, 内部 RC 振荡器
		0.571	0.980	mA	+25°C		
		0.608	1.12	mA	+85°C		
	PIC18LFXXJ50	0.625	1.14	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		0.681	1.14	mA	+25°C		
		0.725	1.25	mA	+85°C		
	PIC18FXXJ50	0.613	1.21	mA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	
		0.680	1.21	mA	+25°C		
		0.730	1.30	mA	+85°C		
	PIC18FXXJ50	0.673	1.27	mA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容	
		0.728	1.27	mA	+25°C		
		0.779	1.45	mA	+85°C		
	PIC18LFXXJ50	0.750	1.4	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 8 MHz, RC_IDLE 模式, 内部 RC 振荡器
		0.797	1.5	mA	+25°C		
		0.839	1.6	mA	+85°C		
	PIC18LFXXJ50	0.91	2.4	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		0.96	2.4	mA	+25°C		
1.01		2.5	mA	+85°C			
PIC18FXXJ50	0.87	2.1	mA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容		
	0.93	2.1	mA	+25°C			
	0.98	2.3	mA	+85°C			
PIC18FXXJ50	0.95	2.6	mA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容		
	1.01	2.6	mA	+25°C			
	1.06	2.7	mA	+85°C			

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 V_{DD} 或 V_{SS} ，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 I_{DD} 测量值的测试条件为：
 $\overline{OSC1}$ = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 V_{DD}/V_{SS} ；
 $\overline{MCLR}$ = V_{DD} ；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 $+70^{\circ}\text{C}$ 。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（ $USBEN = 1$ ， $SUSPND = 1$ ，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

PIC18F46J50 系列

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
PIC18F46J50 系列		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)					FOSC = 4 MHz, PRI_RUN 模式, EC 振荡器	
	PIC18LFXXJ50	0.879	1.25	mA	-40°C		VDD = 2.0V, VDDCORE = 2.0V
		0.881	1.25	mA	+25°C		
		0.891	1.36	mA	+85°C		
	PIC18LFXXJ50	1.35	1.70	mA	-40°C		VDD = 2.5V, VDDCORE = 2.5V
		1.30	1.70	mA	+25°C		
		1.27	1.82	mA	+85°C		
	PIC18FXXJ50	1.09	1.60	mA	-40°C		VDD = 2.15V, VDDCORE = 10 μF 电容
		1.09	1.60	mA	+25°C		
		1.11	1.70	mA	+85°C		
	PIC18FXXJ50	1.36	1.95	mA	-40°C		VDD = 3.3V, VDDCORE = 10 μF 电容
		1.36	1.89	mA	+25°C		
		1.41	1.92	mA	+85°C		
	PIC18LFXXJ50	10.9	14.8	mA	-40°C		VDD = 2.5V, VDDCORE = 2.5V
		10.6	14.8	mA	+25°C		
		10.6	15.2	mA	+85°C		
	PIC18FXXJ50	12.9	23.2	mA	-40°C		VDD = 3.3V, VDDCORE = 10 μF 电容
		12.8	22.7	mA	+25°C		
		12.7	22.7	mA	+85°C		

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 VDD/VSS；
MCLR = VDD；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 +70°C。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1，SUSPND = 1，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）					
PIC18F46J50 系列		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)						
	PIC18LFXXJ50	0.28	0.70	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	FOSC = 4 MHz PRI_IDLE 模式, EC 振荡器
		0.30	0.70	mA	+25°C		
		0.34	0.75	mA	+85°C		
	PIC18LFXXJ50	0.37	1.0	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		0.40	1.0	mA	+25°C		
		0.50	1.1	mA	+85°C		
	PIC18FXXJ50	0.36	0.85	mA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	
		0.38	0.85	mA	+25°C		
		0.41	0.90	mA	+85°C		
PIC18FXXJ50	0.45	1.3	mA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容		
	0.48	1.2	mA	+25°C			
	0.55	1.2	mA	+85°C			
	PIC18LFXXJ50	4.5	6.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	FOSC = 48 MHz PRI_IDLE 模式, EC 振荡器
		4.5	6.5	mA	+25°C		
		4.6	6.5	mA	+85°C		
PIC18FXXJ50	4.8	12.4	mA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容		
	4.9	11.5	mA	+25°C			
	5.1	11.5	mA	+85°C			

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 VDD/VSS；
MCLR = VDD；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 +70°C。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1，SUSPND = 1，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

PIC18F46J50 系列

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
PIC18F46J50 系列		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)						
	PIC18LFXXJ50	8.2	11	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V FOSC = 24 MHz PRI_RUN 模式, ECPLL 振荡器 (4 MHz 输入)	
		8.1	11	mA	+25°C		
		8.0	10	mA	+85°C		
	PIC18FXXJ50	8.1	15	mA	-40°C		VDD = 3.3V, VDDCORE = 10 μF 电容
		8.1	14	mA	+25°C		
		8.1	14	mA	+85°C		
	PIC18LFXXJ50	12	14	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V FOSC = 48 MHz PRI_RUN 模式, ECPLL 振荡器 (4 MHz 输入)	
		12	14	mA	+25°C		
		11	14	mA	+85°C		
	PIC18FXXJ50	14	24	mA	-40°C		VDD = 3.3V, VDDCORE = 10 μF 电容
		14	23	mA	+25°C		
		14	23	mA	+85°C		

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 VDD/VSS；
MCLR = VDD；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 +70°C。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1，SUSPND = 1，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
PIC18F46J50 系列		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)						
	PIC18LFXXJ50	9.9	45	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	FOSC = 32 kHz ⁽³⁾ SEC_RUN 模式, LPT1OSC = 0
		11	45	μA	+25°C		
		13	61	μA	+85°C		
	PIC18FXXJ50	39	95	μA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	
		50	95	μA	+25°C		
		57	105	μA	+85°C		
	PIC18FXXJ50	42	110	μA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容	
		54	110	μA	+25°C		
		57	150	μA	+85°C		
	PIC18LFXXJ50	3.5	31	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	FOSC = 32 kHz ⁽³⁾ SEC_IDLE 模式, LPT1OSC = 0
		3.8	31	μA	+25°C		
		4.3	50	μA	+85°C		
	PIC18FXXJ50	34	87	μA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	
		45	89	μA	+25°C		
		56	97	μA	+85°C		
	PIC18FXXJ50	35	100	μA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容	
		46	100	μA	+25°C		
		56	140	μA	+85°C		

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 VDD/VSS；
MCLR = VDD；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 +70°C。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1，SUSPND = 1，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

PIC18F46J50 系列

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
PIC18F46J50 系列		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		
	模块增加电流（ΔI _{WDT} 、ΔI _{HLVD} 、ΔI _{OSCB} 、ΔI _{AD} 和 ΔI _{USB} ）						
	看门狗定时器	0.84	8.0	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	PIC18LFXXJ50
		0.96	8.0	μA	+25°C		
		0.97	10.4	μA	+85°C		
		0.65	7.0	μA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	PIC18FXXJ50
		0.78	7.0	μA	+25°C		
		0.77	10	μA	+85°C		
		1.3	12.1	μA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容	
		1.3	12.1	μA	+25°C		
	1.3	13.6	μA	+85°C			
D022B (ΔI _{HLVD})	高 / 低压检测	3.9	8.0	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	PIC18LFXXJ50
		4.7	8.0	μA	+25°C		
		5.4	9.0	μA	+85°C		
		2.6	6.0	μA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	PIC18FXXJ50
		3.1	6.0	μA	+25°C		
		3.5	8.0	μA	+85°C		
		3.5	9.0	μA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容	
		4.1	9.0	μA	+25°C		
		4.5	12	μA	+85°C		
D025 (ΔI _{OSCB})	带低功耗 Timer1 振荡器的 实时时钟 / 日历	0.80	4.0	μA	-40°C	VDD = 2.15V, VDDCORE = 10 μF 电容	PIC18FXXJ50 32.768 kHz ⁽³⁾ , T1OSCEN = 1, LPT1OSC = 0
		0.83	4.5	μA	+25°C		
		0.95	4.5	μA	+60°C		
		1.2	4.5	μA	+85°C		
		0.75	4.5	μA	-40°C	VDD = 2.5V, VDDCORE = 10 μF 电容	
		0.92	5.0	μA	+25°C		
		1.1	5.0	μA	+60°C		
		1.1	5.0	μA	+85°C		
		0.95	6.5	μA	-40°C	VDD = 3.3V, VDDCORE = 10 μF 电容	
		1.1	6.5	μA	+25°C		
		1.2	8.0	μA	+60°C		
		1.4	8.0	μA	+85°C		

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 VDD 或 VSS，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 IDD 测量值的测试条件为：
OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 VDD/VSS；
MCLR = VDD；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 +70°C。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1，SUSPND = 1，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

30.2 直流特性: PIC18F46J50 系列的掉电电流和供电电流（工业级）（续）

PIC18LF46J50 系列		标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）			
PIC18F46J50 系列		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
D026 (ΔIAD)	模块增加电流 (ΔI _{WDT} 、ΔI _{HLVD} 、ΔI _{OSCB} 、ΔI _{AD} 和 ΔI _{USB})				
	A/D 转换器	3.0	10	μA	-40°C
		3.0	10	μA	+25°C
		3.0	10	μA	+85°C
		3.0	10	μA	-40°C
		3.0	10	μA	+25°C
		3.0	10	μA	+85°C
		3.2	11	μA	-40°C
		3.2	11	μA	+25°C
		3.2	11	μA	+85°C
D027 (ΔI _{USB})	USB 模块	1.6	3.2	mA	-40°C
		1.6	3.2	mA	+25°C
		1.5	3.2	mA	+85°C

- 注 1: 在休眠模式下，掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式、所有 I/O 引脚处于高阻态并且连接到 V_{DD} 或 V_{SS}，禁止所有会带来新增电流的功能部件（如 WDT、Timer1 振荡器和 BOR 等）时测得的。
- 2: 供电电流主要受工作电压、频率和模式的影响。其他因素，如 I/O 引脚负载和开关速率、振荡器类型和电路、内部代码执行模式和温度也会对电流消耗产生影响。禁止所有会带来新增电流的功能部件（USB 模块和 WDT 等）。有效工作模式下，所有 I_{DD} 测量值的测试条件为：
OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，拉至 V_{DD}/V_{SS}；
MCLR = V_{DD}；除非另外指定，否则禁止 WDT。
- 3: 采用标准低成本 32 kHz 晶振的低功耗 Timer1 的工作温度范围为 -10°C 至 +70°C。扩展级温度晶振的成本则高很多。
- 4: 这是在使能 USB 模块，使用 48 MHz 的时钟，但未连接 USB 电缆时的模块增加电流。连接 USB 电缆或发送数据时，消耗的电流可能会高得多（见第 22.6.4 节“USB 收发器的电流消耗”）。在 USB 挂起模式下（USBEN = 1，SUSPND = 1，总线处于空闲状态），USB 模块电流将主要是 D+ 或 D- 上拉电阻消耗的电流。根据“USB 2.0 规范”的 resistor_ecn 补充信息，集成上拉电阻使用“电阻切换”技术，因此在空闲条件下，电阻可低至 900Ω。

PIC18F46J50 系列

30.3 直流特性：PIC18F46J50 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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ （工业级）			
参数编号	符号	特性	最小值	最大值	单位	条件
D030 D030A D031 D031A D031B D032 D033 D033A D034	V_{IL}	输入低电压 所有 I/O 端口： 带 TTL 缓冲器 ⁽⁴⁾ 带 TTL 缓冲器 ⁽⁴⁾ 带施密特触发器缓冲器 SDAx/SCLx SDAx/SCLx MCLR OSC1 OSC1 T1OSI	V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} V_{SS}	$0.15 V_{DD}$ 0.8 $0.2 V_{DD}$ $0.3 V_{DD}$ 0.8 $0.2 V_{DD}$ $0.3 V_{DD}$ $0.2 V_{DD}$ 0.3	V V V V V V V V V	$V_{DD} \leq 3.3\text{V}$ $3.3\text{V} < V_{DD} < 3.6\text{V}$ 使能 I^2C^{TM} 使能 SMBus HS 和 HSPLL 模式 EC 和 ECPLL 模式 T1OSCEN = 1
D040 D040A D041 Dxxx DxxxA Dxxx D041A D041B D042 D043 D043A D044	V_{IH}	输入高电压 非 5.5V 耐压的 I/O 端口： 带 TTL 缓冲器 ⁽⁴⁾ 带 TTL 缓冲器 ⁽⁴⁾ 带施密特触发器缓冲器 5.5V 耐压的 I/O 端口： ⁽⁵⁾ 带 TTL 缓冲器 带施密特触发器缓冲器 SDAx/SCLx SDAx/SCLx MCLR OSC1 OSC1 T1OSI	$0.25 V_{DD} + 0.8\text{V}$ 2.0 $0.8 V_{DD}$ $0.25 V_{DD} + 0.8\text{V}$ 2.0 $0.8 V_{DD}$ $0.7 V_{DD}$ 2.1 $0.8 V_{DD}$ $0.7 V_{DD}$ $0.8 V_{DD}$ 1.6	V_{DD} V_{DD} V_{DD} 5.5 5.5 5.5 5.5 5.5 V_{DD} V_{DD} V_{DD}	V V V V V V V V V V V V	$V_{DD} < 3.3\text{V}$ $3.3\text{V} < V_{DD} < 3.6\text{V}$ $V_{DD} < 3.3\text{V}$ $3.3\text{V} \leq V_{DD} \leq 3.6\text{V}$ 使能 I^2C^{TM} 使能 SMBus, $V_{DD} \geq 3\text{V}$ HS 和 HSPLL 模式 EC 和 ECPLL 模式 T1OSCEN = 1
D060 D061 D063	I_{IL}	输入泄漏电流 ^(1,2) I/O 端口 MCLR OSC1	— — —	± 0.2 ± 0.2 ± 0.2	μA μA μA	$V_{SS} \leq V_{PIN} \leq V_{DD}$, 引脚处于高阻态 $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$
D070	I_{PU} I_{PURB}	弱上拉电流 PORTB、PORTD ⁽³⁾ 和 PORTE ⁽³⁾ 弱上拉电流	80	400	μA	$V_{DD} = 3.3\text{V}$, $V_{PIN} = V_{SS}$

- 注 1: MCLR 引脚上的泄漏电流主要取决于所施加的电压。规定电压为正常工作条件下的电压。在不同的输入电压下可测得更高的泄漏电流。
- 2: 负电流定义为引脚的拉电流。
- 3: 仅在 44 引脚器件上可用。
- 4: 当 RC4 和 RC5 用作通用输入时，它们的阈值参照 V_{USB} ，而不是 V_{DD} 。
- 5: 请参见表 10-2 了解引脚的耐压信息。

30.3 直流特性：PIC18F46J50 系列（工业级）（续）

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
D080	VOL	输出低电压				
		I/O 端口： PORTA（RA6 除外）、 PORTD 和 PORTE	—	0.4	V	IO _L = 2 mA，V _{DD} = 3.3V， -40°C 至 +85°C
		PORTB、PORTC 和 RA6	—	0.4	V	IO _L = 8.5 mA，V _{DD} = 3.3V， -40°C 至 +85°C
D090	VOH	输出高电压				
		I/O 端口： PORTA（RA6 除外）、 PORTD 和 PORTE	2.4	—	V	IO _H = -2 mA，V _{DD} = 3.3V， -40°C 至 +85°C
		PORTB、PORTC 和 RA6	2.4	—	V	IO _H = -6 mA，V _{DD} = 3.3V， -40°C 至 +85°C
D101	C _{IO}	输出引脚上的容性负载规范				
		所有 I/O 引脚和 OSC2	—	50	pF	满足交流时序规范
D102	C _B	SCLx 和 SDAx	—	400	pF	I ² C™ 规范

- 注 1: MCLR 引脚上的泄漏电流主要取决于所施加的电压。规定电压为正常工作条件下的电压。在不同的输入电压下可测得更高的泄漏电流。
- 2: 负电流定义为引脚的拉电流。
- 3: 仅在 44 引脚器件上可用。
- 4: 当 RC4 和 RC5 用作通用输入时，它们的阈值参照 V_{USB}，而不是 V_{DD}。
- 5: 请参见表 10-2 了解引脚的耐压信息。

PIC18F46J50 系列

表 30-1: 存储器编程要求

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	典型值 †	最大值	单位	条件
D130	EP	闪存程序存储器					
		单元耐擦写能力	10K	—	—	E/W	-40°C 至 +85°C
D131	VPR	读操作时的 VDDcore	VMIN	—	2.75	V	VMIN = 最小工作电压
D132B	VPEW	自定时擦除或写操作时的 VDDCORE	2.25	—	2.75	V	
D133A	TIW	自定时写周期时间	—	2.8	—	ms	64 字节
D133B	TIE	自定时块擦除周期时间	—	33.0	—	ms	
D134	TRETD	特性保持时间	20	—	—	年	假设没有违反其他规范
D135	IDDP	编程时的供电电流	—	3	—	mA	

† 除非另外声明，否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。

表 30-2: 比较器规范

工作条件: 3.0V < VDD < 3.6V, -40°C < TA < +85°C（除非另外声明）							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D300	VIOFF	输入失调电压	—	+/-5	+/-25	mV	
D301	VICM	输入共模电压	0	—	VDD	V	
	VIRV	内部参考电压	0.57	0.60	0.63	V	
D302	CMRR	共模抑制比	55	—	—	dB	
D303	TRESP	响应时间 (1)	—	150	400	ns	
D304	TMC2OV	比较器模式改变到输出有效的 时间	—	—	10	µs	

注 1: 响应时间是在比较器的一个输入端电压为 VDD/2，而另一个输入端从 VSS 跳变到 VDD 时测得的。

表 30-3: 参考电压规范

工作条件: $3.0V < V_{DD} < 3.6V$, $-40^{\circ}C < T_A < +85^{\circ}C$ (除非另外声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D310	VRES	分辨率	$V_{DD}/24$	—	$V_{DD}/32$	LSb	
D311	VRAA	绝对精度	—	—	1/2	LSb	
D312	VRUR	单位电阻值 (R)	—	2k	—	Ω	
D313	TSET	稳定时间 ⁽¹⁾	—	—	10	μs	

注 1: 稳定时间是在 $CVRR = 1$ 并且 $CVR<3:0>$ 位从 0000 跳变到 1111 时测得的。

表 30-4: 内部稳压器规范

工作条件: $-40^{\circ}C < T_A < +85^{\circ}C$ (除非另外声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	VRGOUT	稳压器输出电压	2.35	2.5	2.7	V	使能稳压器, $V_{DD} = 3.0V$
	CEFC	外部滤波电容值 ⁽¹⁾	5.4	10	18	μF	建议 $ESR < 3\Omega$ ESR 必须 $< 5\Omega$

注 1: CEFC 适用于系列中的 PIC18F 器件。对于系列中的 PIC18LF 器件, 虽然仍然应当使用适当的电源轨旁路, 但对于 V_{DDCORE} , 不存在具体的最小或最大电容。

表 30-5: ULPWU 规范

直流特性			标准工作条件 (除非另外声明)				
			工作温度 $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (工业级)				
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D100	IULP	超低功耗唤醒电流	—	60	—	nA	I/O 净泄漏电流和 1.6V 时引脚上的灌电流, $V_{DD} = 3.3V$ 请参见应用笔记 AN879, “Using the Microchip Ultra Low-Power Wake-up Module” (DS00879)

† 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。

表 30-6: CTMU 电流源规范

直流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明)				
			工作温度 $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (工业级)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
	IOUT1	CTMU 电流源, 基本范围	—	550	—	nA	$CTMUICON<1:0> = 01$
	IOUT2	CTMU 电流源, 10x 范围	—	5.5	—	μA	$CTMUICON<1:0> = 10$
	IOUT3	CTMU 电流源, 100x 范围	—	55	—	μA	$CTMUICON<1:0> = 11$

注 1: 电流微调范围的中点为标称值 ($CTMUICON<7:2> = 000000$)。

PIC18F46J50 系列

表 30-7: USB 模块规范

工作条件: -40°C < TA < +85°C (除非另外声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D313	VUSB	USB 电压	3.0	—	3.6	V	VUSB 引脚上的电压必须在此范围内以确保 USB 正常工作
D314	IIL	D+ 或 D- 上的输入泄漏电流	—	—	+/-0.5	µA	VSS ≤ VPIN ≤ VUSB
D315	VILUSB	USB 缓冲区的输入低电压	—	—	0.8	V	用于 VUSB 范围
D316	VIHUSB	USB 缓冲区的输入高电压	2.0	—	—	V	用于 VUSB 范围
D318	VDIFS	差分输入灵敏度	—	—	0.2	V	当达到 VCM 时, D+ 和 D- 之间的电压差必须超出此值
D319	VCM	差分共模范围	0.8	—	2.5	V	
D320	ZOUT	驱动器输出阻抗 ⁽¹⁾	28	—	44	Ω	
D321	VOL	输出低电压	0.0	—	0.3	V	1.5 kΩ 负载连接到 3.6V
D322	VOH	输出高电压	2.8	—	3.6	V	1.5 kΩ 负载接地

注 1: D+ 和 D- 信号线具有内置的阻抗匹配电阻。在 PIC18F46J50 系列器件和 USB 电缆之间的 D+/D- 信号路径上不需要外接任何电阻、电容或磁性元件。

图 30-3: 高 / 低压检测特性

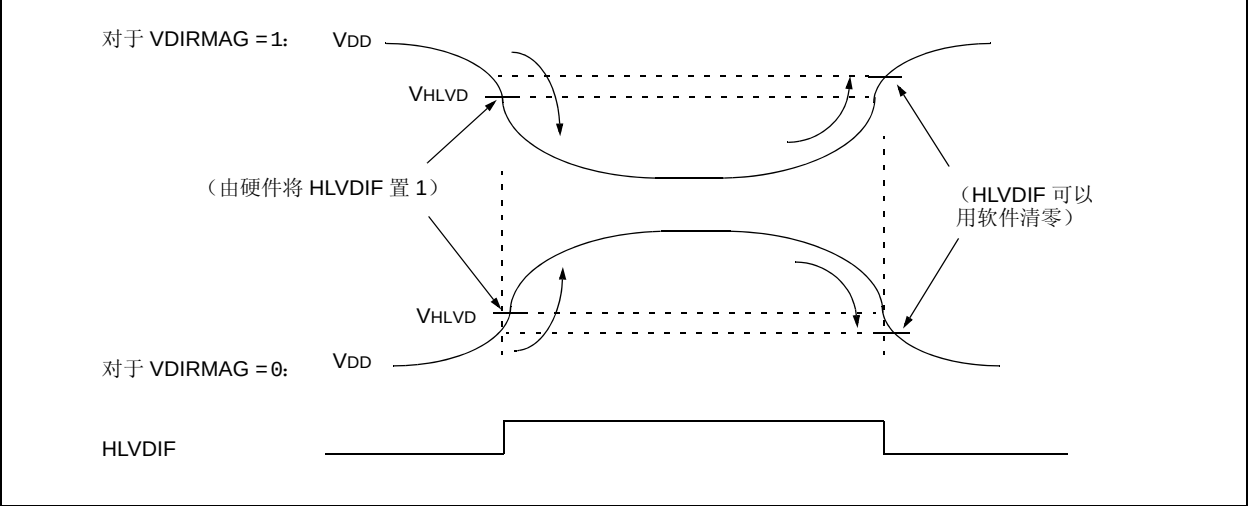


表 30-8: 高 / 低压检测特性

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	典型值	最大值	单位	条件
D420		VDD 由高转变为低时的 HLVD 电压	HLVDL<3:0> = 1000	2.33	2.45	2.57	V
			HLVDL<3:0> = 1001	2.47	2.60	2.73	V
			HLVDL<3:0> = 1010	2.66	2.80	2.94	V
			HLVDL<3:0> = 1011	2.76	2.90	3.05	V
			HLVDL<3:0> = 1100	2.85	3.00	3.15	V
			HLVDL<3:0> = 1101	2.97	3.13	3.29	V
			HLVDL<3:0> = 1110	3.23	3.40	3.57	V
D421	TIRVST	内部参考电压稳定时间	—	20	—	μs	
D422	TLVD	高 / 低压检测脉冲宽度	200	—	—	μs	

PIC18F46J50 系列

30.4 交流（时序）特性

30.4.1 时序参数符号体系

时序参数符号采用以下格式之一进行创建：

1. TppS2ppS
2. TppS
3. Tcc:ST（仅用于 I²C 规范）
4. Ts（仅用于 I²C 规范）

T		T	
F	频率	T	时间

小写字母（pp）及其含义：

pp		osc	OSC1
cc	CCP1	rd	$\overline{RD}$
ck	CLKO	rw	$\overline{RD}$ 或 $\overline{WR}$
cs	$\overline{CS}$	sc	SCK
di	SDI	ss	$\overline{SS}$
do	SDO	t0	T0CKI
dt	数据输入	t1	T13CKI
io	I/O 端口	wr	$\overline{WR}$
mc	MCLR		

大写字母及其含义：

S		P	周期
F	下降	R	上升
H	高	V	有效
I	无效（高阻）	Z	高阻
L	低		
仅用于 I ² C		High	高
AA	输出访问	Low	低
BUF	总线空闲		

Tcc:ST（仅用于 I²C 规范）

CC		SU	建立
HD	保持		
ST		STO	停止条件
DAT	数据输入保持		
STA	启动条件		

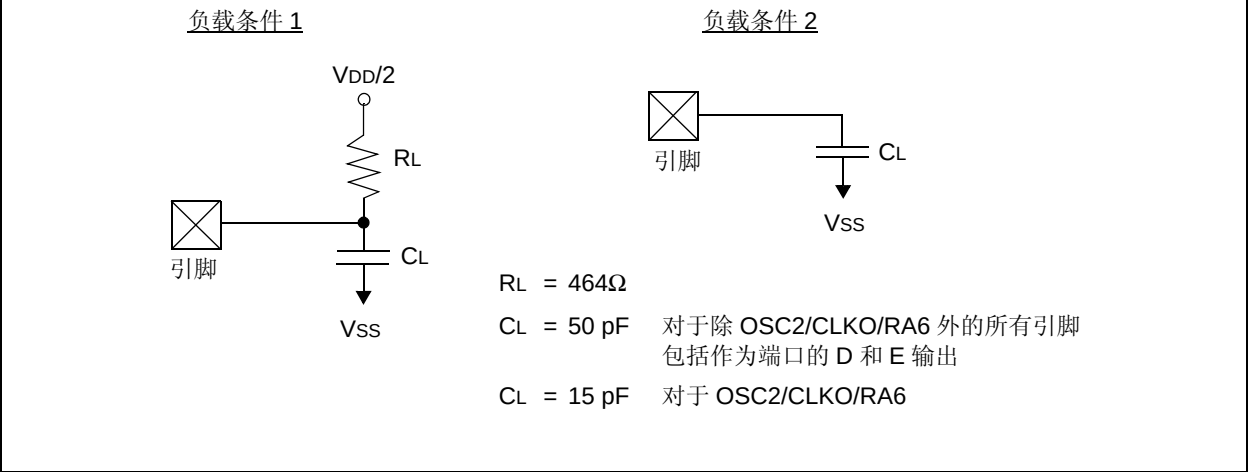
30.4.2 时序条件

表 30-9 中指定的温度和电压适用于所有的时序规范，除非另外声明。图 30-4 规定了时序规范的负载条件。

表 30-9: 温度和电压规范 —— 交流

交流特性	标准工作条件（除非另外声明）	
	工作温度	-40°C ≤ TA ≤ +85°C（工业级）
	工作电压 VDD	范围如第 30.1 节和第 30.3 节中所述。

图 30-4: 器件时序规范的负载条件



PIC18F46J50 系列

30.4.3 时序图和规范

图 30-5: 外部时钟时序

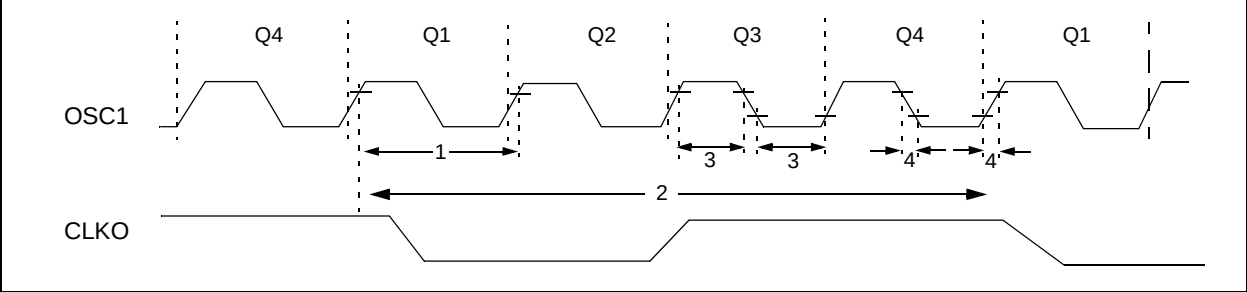


表 30-10: 外部时钟时序要求

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
		振荡器频率 ⁽¹⁾	DC	48	MHz	ECPLL 振荡器模式 ⁽²⁾
			4	16	MHz	HS 振荡器模式
			4	16 ⁽⁴⁾	MHz	HSPLL 振荡器模式 ⁽³⁾
1	TOSC	外部 CLKI 周期 ⁽¹⁾	20.8	—	ns	EC 振荡器模式
		振荡器周期 ⁽¹⁾	20.8	—	ns	ECPLL 振荡器模式 ⁽²⁾
			62.5	250	ns	HS 振荡器模式
			62.5 ⁽⁴⁾	250	ns	HSPLL 振荡器模式 ⁽³⁾
2	TCY	指令周期时间 ⁽¹⁾	83.3	DC	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	EC 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	EC 振荡器模式

- 注 1: 对于除 PLL 外的所有配置, 指令周期 (TCY) 等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型, 器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值, 可能导致振荡器运行不稳定和 / 或导致电流消耗超出预期值。所有器件在测试 “最小值” 时, 都在 OSC1/CLKI 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的 “最大” 周期时间限制为 “DC” (无时钟)。
- 2: 若要使用 PLL, 外部时钟频率必须为 4、8、12、16、20、24、40 或 48 MHz。
- 3: 若要使用 PLL, 晶振 / 谐振器必须产生 4、8、12 或 16 MHz 的频率。
- 4: 这是最高晶振 / 谐振器驱动频率。从 PLL 运行时的内部 FOSC 频率最高可达 48 MHz。

表 30-11: PLL 时钟时序规范 (VDDCORE = 2.35V 至 2.75V)

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
F10	FPLLIN	PLL 输入频率范围	—	4 ⁽¹⁾	—	MHz	
F11	FPLLO	PLL 输出频率 (24x FPLLIN)	—	96	—	MHz	
F12	t _{rc}	PLL 起振时间 (锁定时间)	—	—	2	ms	

注 1: PLL 设计为使用 4 MHz 的输入频率, 但通过使用 PLL 输入预分频器, 它可以接受 4 MHz 至 48 MHz 的输入。

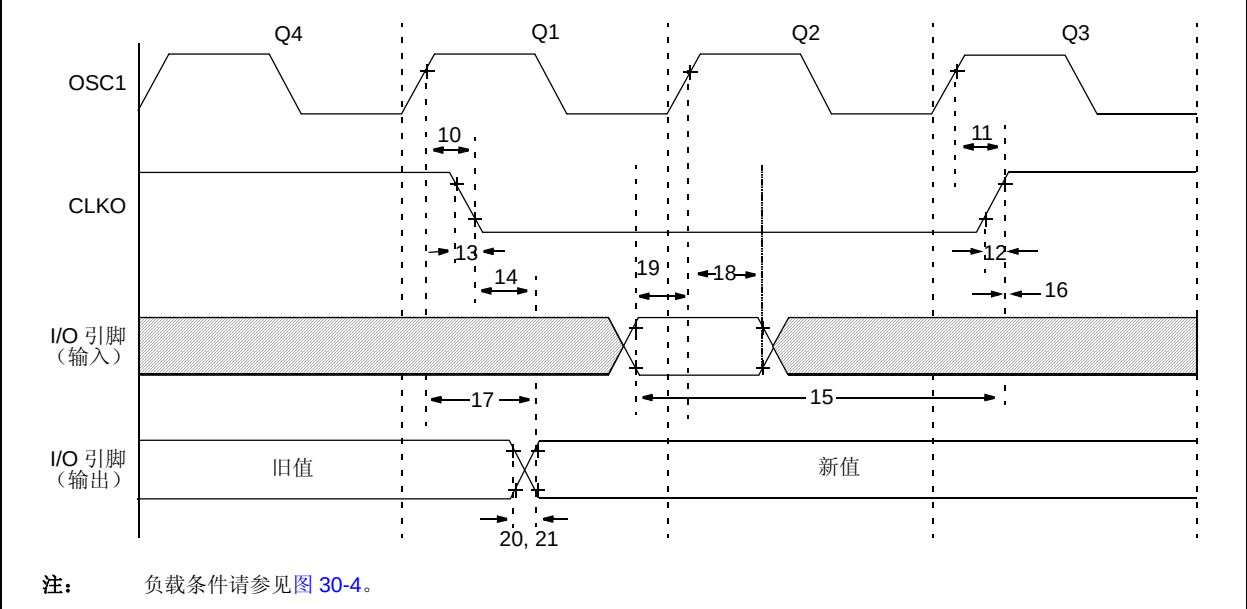
表 30-12: 内部 RC 精度 (INTOSC 和 INTRC 时钟源)

参数编号	器件	最小值	典型值	最大值	单位	条件	
	在频率为 8 MHz、4 MHz、2 MHz、1 MHz、500 kHz、250 kHz、125 kHz 和 31 kHz 时的 INTOSC 精度 ⁽¹⁾						
	所有器件	-1	+/-0.15	+1	%	0°C 至 +85°C	VDD = 2.4V-3.6V VDDCORE = 2.3V-2.7V
		-1	+/-0.25	+1	%	-40°C 至 +85°C	VDD = 2.0V-3.6V VDDCORE = 2.0V-2.7V
	在频率为 31 kHz 时的 INTRC 精度 ⁽¹⁾						
	所有器件	20.3	—	42.2	kHz	-40°C 至 +85°C	VDD = 2.0V-3.6V VDDCORE = 2.0V-2.7V

注 1: 31 kHz 时钟的精度规范由在给定时间内为其提供时钟的时钟源决定。当 INTSRC (OSCTUNE<7>) 为 1 时, 使用 INTOSC 精度规范。当 INTSRC 为 0 时, 使用 INTRC 精度规范。

PIC18F46J50 系列

图 30-6: CLKO 和 I/O 时序



注： 负载条件请参见图 30-4。

表 30-13: 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 ↓ 的时间	—	75	200	ns	(注 1)
11	TosH2ckH	OSC1 ↑ 到 CLKO ↑ 的时间	—	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 ↓ 到端口输出有效的时间	—	—	0.5 Tcy + 20	ns	
15	TioV2ckH	CLKO ↑ 之前端口输入有效的时间	0.25 Tcy + 25	—	—	ns	
16	TckH2ioI	CLKO ↑ 之后端口输入保持的时间	0	—	—	ns	
17	TosH2ioV	OSC1 ↑ (Q1 周期) 到端口输出有效的时间	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 周期) 到端口输入无效的时间 (I/O 输入保持时间)	100	—	—	ns	
19	TioV2osH	端口输入有效到 OSC1 ↑ 的时间 (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 引脚高电平或低电平时间	Tcy	—	—	ns	
23	TRBP	RB<7:4> 电平变化 INTx 高电平或低电平时间	Tcy	—	—	ns	

† 这些参数是与任何内部时钟边沿无关的异步事件。

注 1: 测量是在 EC 模式下进行的, 其中 CLKO 输出为 4 x Tosc。

图 30-7: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序

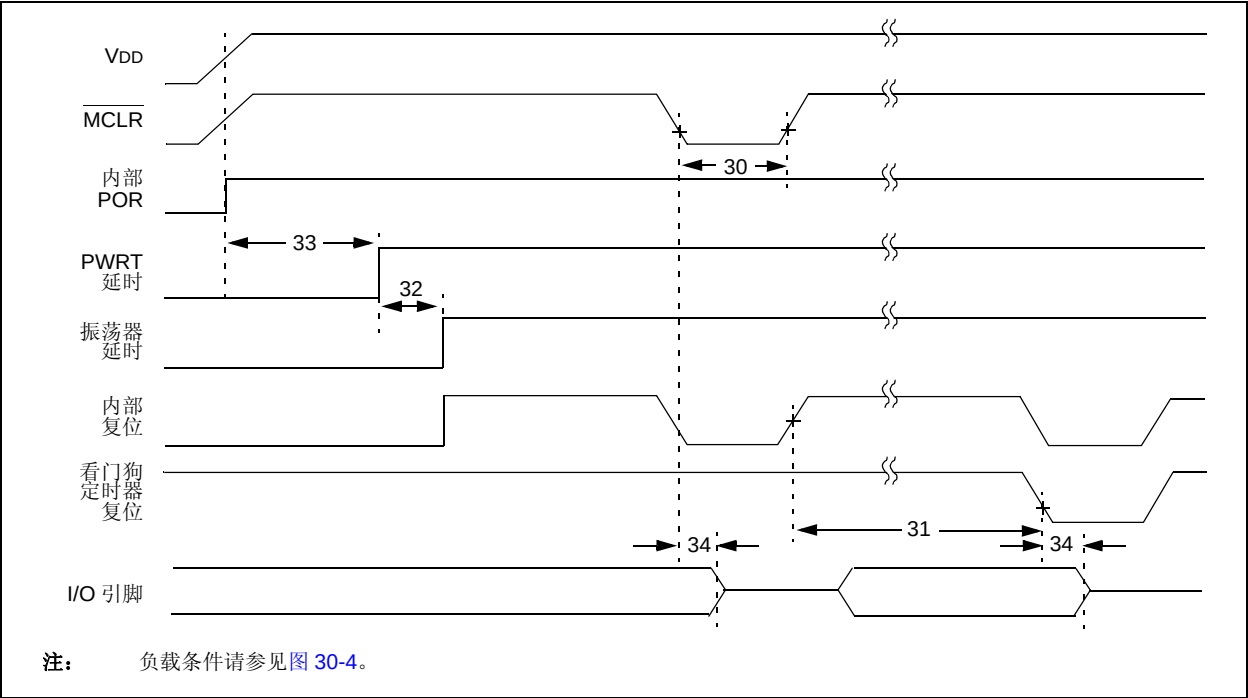


表 30-14: 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
30	TMCLR	MCLR 脉冲宽度（低电平）	2	—	—	μs	（注 3）
31	TWDT	看门狗定时器超时周期（无后分频器）	2.67	4.0	5.53	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	—	1.0	—	ms	
34	TIOZ	自 MCLR 低电平或看门狗定时器复位起 I/O 处于高阻态的时间	—	—	3 TCY + 2	μs	（注 1）
36	TIRVST	内部参考电压稳定时间	—	20	—	μs	
37	TLVD	高 / 低压检测脉冲宽度	—	200	—	μs	
38	TCSD	CPU 启动时间	—	200	—	μs	（注 2）

- 注 1: 最大 TIOZ 取 (3 TCY + 2 μs) 和 700 μs 之间的较小值。
- 注 2: MCLR 上升沿到代码开始执行, 假定 TPWRT (及 TOST, 如适用) 已结束。
- 注 3: MCLR 输入具有一个内部噪声滤波器, 以避免产生不期望的复位。在有意复位单片机时, MCLR 引脚保持低电平的时间必须至少达到该时间量, 以确保触发复位序列。

PIC18F46J50 系列

表 30-15: 低功耗唤醒时间

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
W1	WDS	深度休眠	—	1.5 ms	—	ms	REGSLP = 1
W2	WSLEEP	休眠	—	300 μs	—	μs	REGSLP = 1, PLEN = 0, Fosc = 8 MHz INTOSC
W3	WDOZE1	休眠	—	12 μs	—	μs	REGSLP = 0, PLEN = 0, Fosc = 8 MHz INTOSC
W4	WDOZE2	休眠	—	1.1 μs	—	μs	REGSLP = 0, PLEN = 0, Fosc = 8 MHz EC
W5	WDOZE3	休眠	—	250 ns	—	ns	REGSLP = 0, PLEN = 0, Fosc = 48 MHz EC
W6	WIDLE	空闲	—	300 ns	—	ns	Fosc = 48 MHz EC

图 30-8: TIMER0 和 TIMER1 外部时钟时序

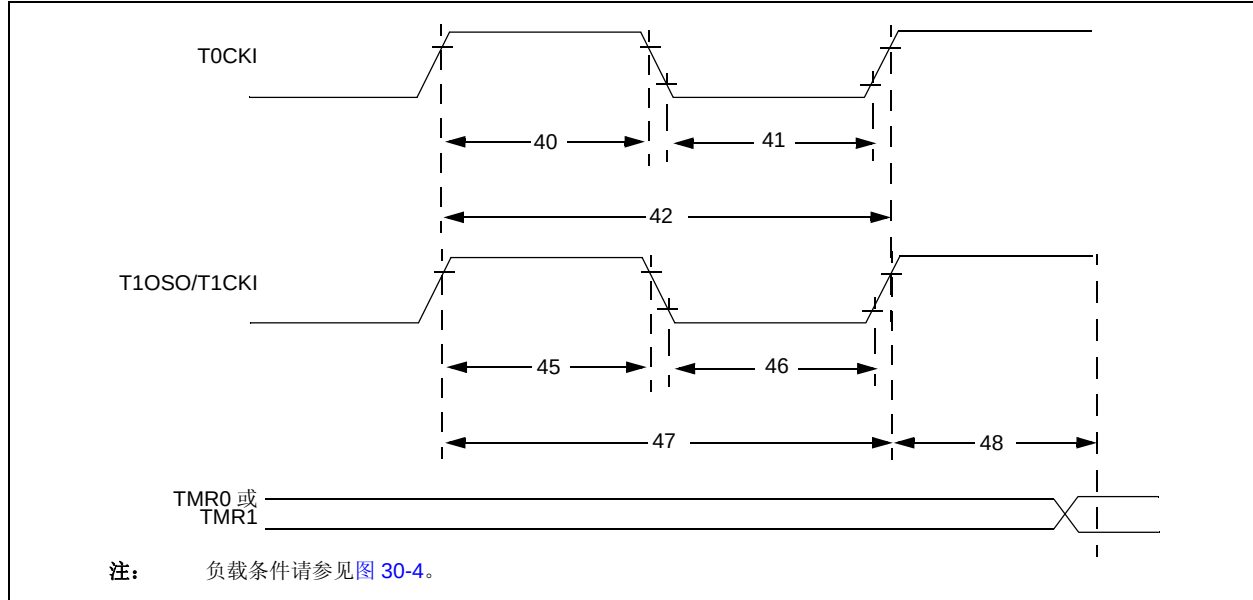


表 30-16: TIMER0 和 TIMER1 外部时钟要求

参数编号	符号	特性		最小值	最大值	单位	条件
40	T _{T0H}	T0CKI 高电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns	
			带预分频器	10	—	ns	
41	T _{T0L}	T0CKI 低电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns	
			带预分频器	10	—	ns	
42	T _{T0P}	T0CKI 周期	无预分频器	$T_{CY} + 10$	—	ns	
			带预分频器	取如下二者中较大值: 20 ns 或 $(T_{CY} + 40)/N$	—	ns	
45	T _{T1H}	T1CKI/T3CKI 高电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	ns	
			同步, 带预分频器	10	—	ns	
			异步	30	—	ns	
46	T _{T1L}	T1CKI/T3CKI 低电平时间	同步, 无预分频器	$0.5 T_{CY} + 5$	—	ns	
			同步, 带预分频器	10	—	ns	
			异步	30	—	ns	
47	T _{T1P}	T1CKI/T3CKI 输入周期	同步	取如下二者中较大值: 20 ns 或 $(T_{CY} + 40)/N$	—	ns	N = 预分频值 (1, 2, 4, 8)
			异步	83	—	ns	
	F _{T1}	T1CKI 输入频率范围 (1)		DC	12	MHz	
48	T _{CKE2TMR1}	从外部 T1CKI 时钟边沿到定时器递增的延时		2 T _{osc}	7 T _{osc}	—	

注 1: Timer1 振荡器设计为驱动 32.768 kHz 晶振。当 T1CKI 用作数字输入时, 可支持最高 12 MHz 的频率。

PIC18F46J50 系列

图 30-9: 增强型捕捉 / 比较 / PWM 时序

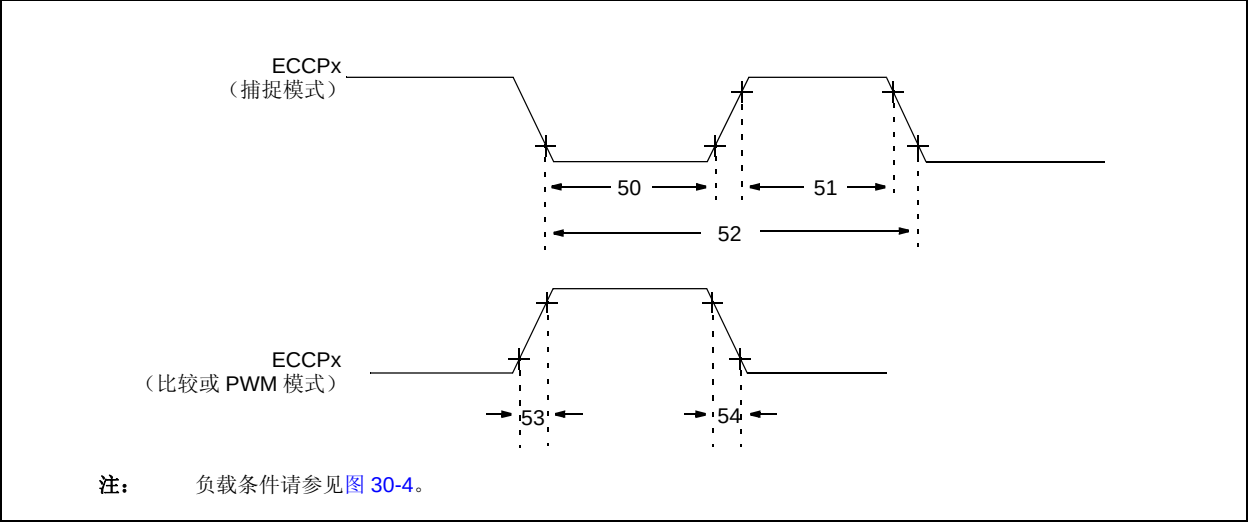


表 30-17: 增强型捕捉 / 比较 / PWM 要求

参数编号	符号	特性	最小值	最大值	单位	条件
50	TccL	ECCPx 输入低电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns
			带预分频器	10	—	ns
51	TccH	ECCPx 输入高电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns
			带预分频器	10	—	ns
52	TccP	ECCPx 输入周期	$\frac{3 T_{CY} + 40}{N}$	—	ns	N = 预分频值 (1、4 或 16)
53	TccR	ECCPx 输出上升时间	—	25	ns	
54	TccF	ECCPx 输出下降时间	—	25	ns	

图 30-10: 并行主端口读时序图

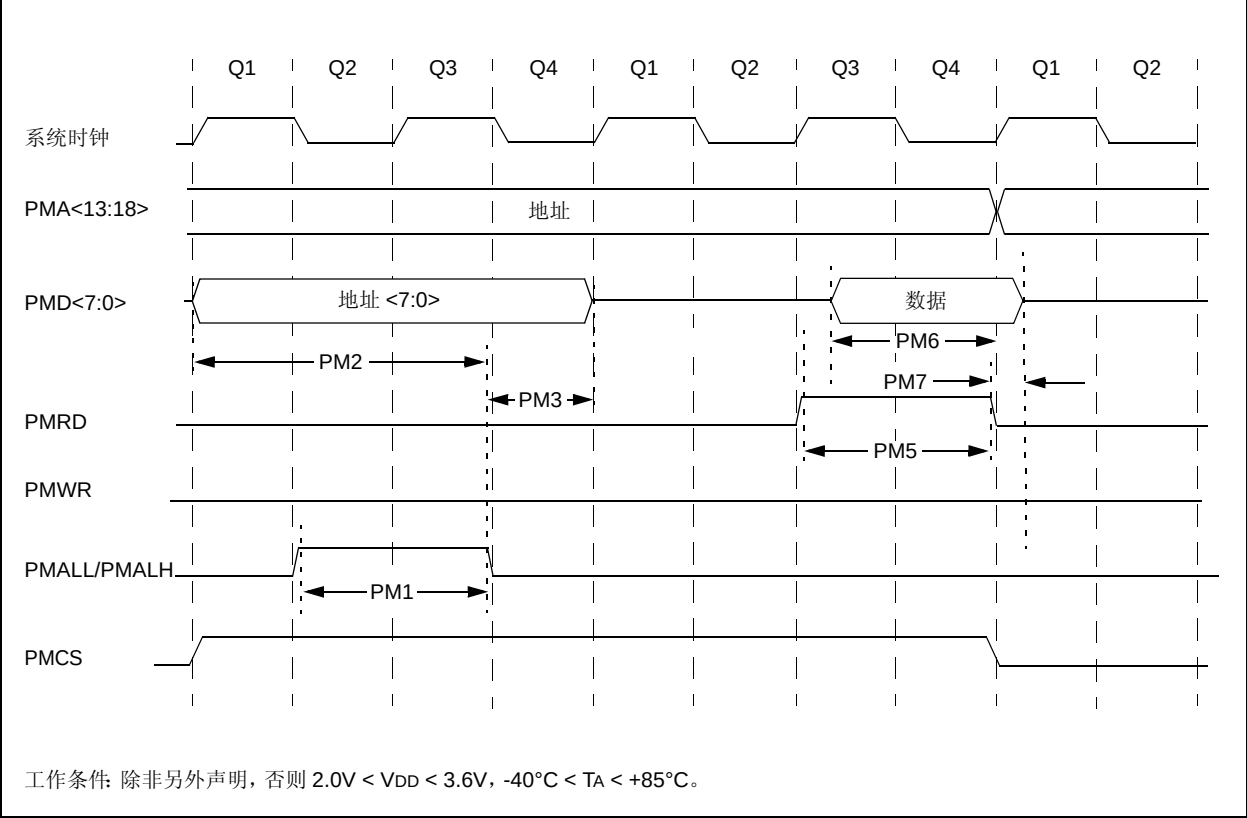


表 30-18: 并行主端口读时序要求

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		地址输出有效到 PMALL/PMALH 无效的时间 (地址建立时间)	—	0.75 Tcy	—	ns
PM3		PMALL/PMALH 无效到地址输出无效的时间 (地址保持时间)	—	0.25 Tcy	—	ns
PM5		PMRD 脉冲宽度	—	0.5 Tcy	—	ns
PM6		数据输入有效到 PMRD 或 PMENB 无效的时间 (数据建立时间)	—	—	—	ns
PM7		PMRD 或 PMENB 无效到数据输入无效的时间 (数据保持时间)	—	—	5	ns

PIC18F46J50 系列

图 30-11: 并行主端口写时序图

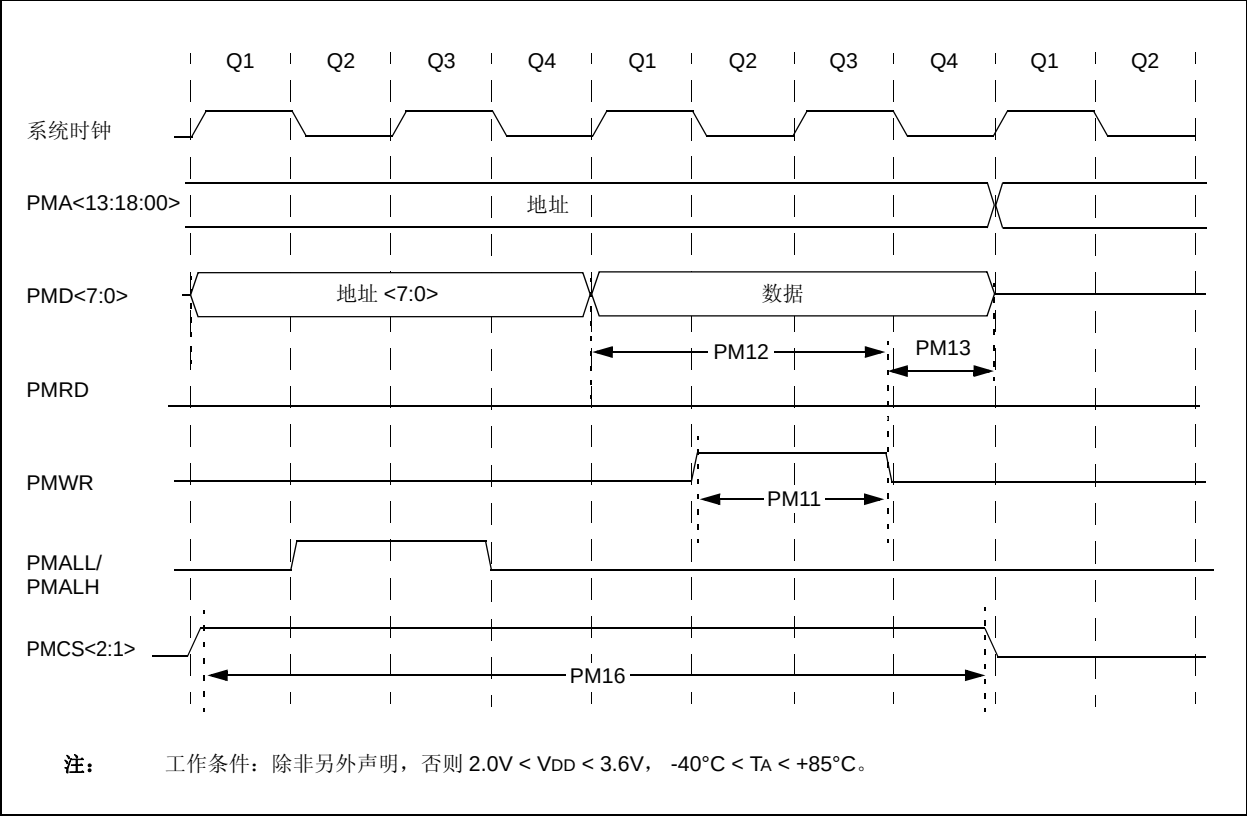


表 30-19: 并行主端口写时序要求

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		PMWR 或 PMENB 变为无效之前数据输出有效的时间（数据建立时间）	—	0.75 Tcy	—	ns
PM13		PMWR 或 PMEMB 无效到数据输出无效的时间（数据保持时间）	—	0.25 Tcy	—	ns
PM16		PMCS 脉冲宽度	Tcy - 5	—	—	ns

图 30-12: 并行从端口时序

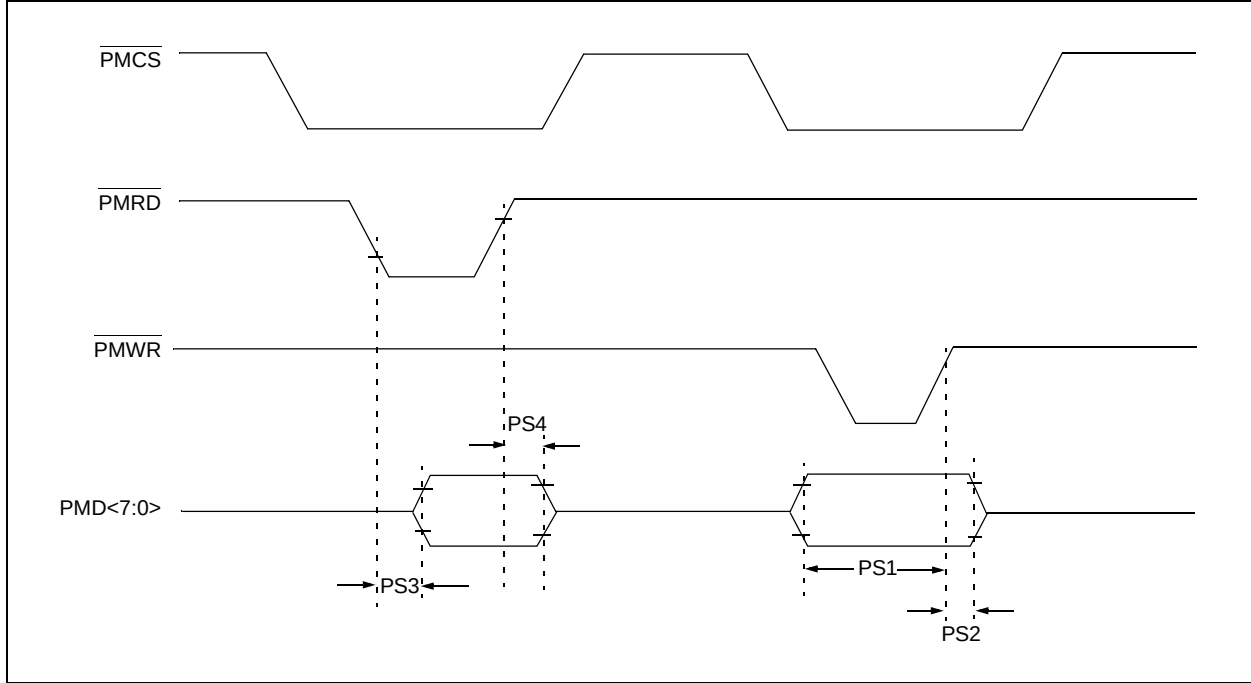


表 30-20: 并行从端口要求

交流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
PS1	TdtV2wrH	$\overline{\text{PMWR}}$ 或 $\overline{\text{PMCS}}$ 无效之前数据输入有效的时间 (建立时间)	20	—	—	ns	
PS2	TwrH2dtl	$\overline{\text{PMWR}}$ 或 $\overline{\text{PMCS}}$ 无效到数据输入无效的时间 (保持时间)	20	—	—	ns	
PS3	TrdL2dtV	$\overline{\text{PMRD}}$ 和 $\overline{\text{PMCS}}$ 有效到数据输出有效的时间	—	—	80	ns	
PS4	TrdH2dtl	$\overline{\text{PMRD}}$ 无效或 $\overline{\text{PMCS}}$ 无效到数据输出无效的时间	10	—	30	ns	

PIC18F46J50 系列

图 30-13: SPI 主模式时序示例 (CKE = 0)

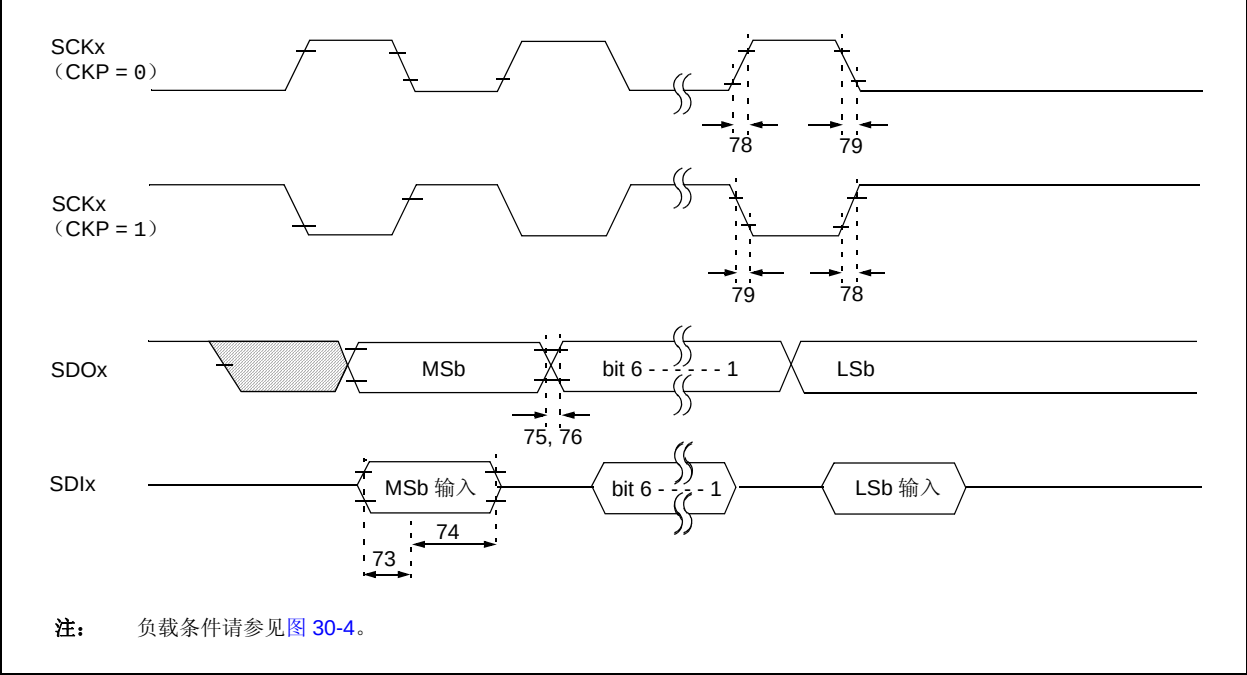


表 30-21: SPI 模式要求示例 (主模式, CKE = 0)

参数编号	符号	特性	最小值	最大值	单位	条件
73	TdiV2sCH, TdiV2sCL	SDIx 数据输入到 SCKx 边沿的建立时间	35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			100	—	ns	VDD = 2.15V, VDDCORE = 2.15V
74	TsCH2diL, TsCL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	30	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			83	—	ns	VDD = 2.15V
75	TdoR	SDOx 数据输出上升时间	—	25	ns	PORTB 或 PORTC
76	TdoF	SDOx 数据输出下降时间	—	25	ns	PORTB 或 PORTC
78	TsCR	SCKx 输出上升时间 (主模式)	—	25	ns	PORTB 或 PORTC
79	TsCF	SCKx 输出下降时间 (主模式)	—	25	ns	PORTB 或 PORTC

图 30-14: SPI 主模式时序示例 (CKE = 1)

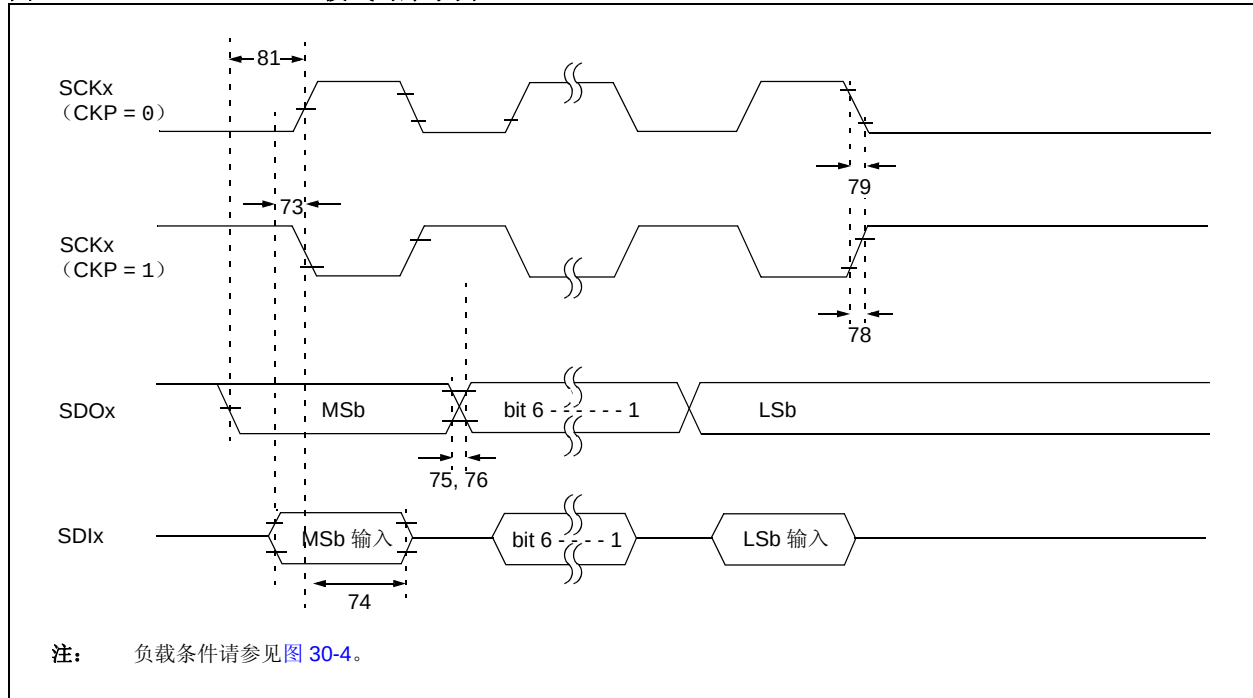


表 30-22: SPI 模式要求示例 (主模式, CKE = 1)

参数编号	符号	特性	最小值	最大值	单位	条件
73	T _{DiV2scH} , T _{DiV2scL}	SDIx 数据输入到 SCKx 边沿的建立时间	35	—	ns	V _{DD} = 3.3V, V _{DDCORE} = 2.5V
			100	—	ns	V _{DD} = 2.15V, V _{DDCORE} = 2.15V
74	T _{sch2diL} , T _{scl2diL}	SDIx 数据输入到 SCKx 边沿的保持时间	30	—	ns	V _{DD} = 3.3V, V _{DDCORE} = 2.5V
			83	—	ns	V _{DD} = 2.15V
75	T _{DoR}	SDOx 数据输出上升时间	—	25	ns	PORTB 或 PORTC
76	T _{DoF}	SDOx 数据输出下降时间	—	25	ns	PORTB 或 PORTC
78	T _{scR}	SCKx 输出上升时间 (主模式)	—	25	ns	PORTB 或 PORTC
79	T _{scF}	SCKx 输出下降时间 (主模式)	—	25	ns	PORTB 或 PORTC
81	T _{DoV2scH} , T _{DoV2scL}	SDOx 数据输出建立到出现 SCKx 边沿的时间	T _{cy}	—	ns	

PIC18F46J50 系列

图 30-15: SPI 从模式时序示例 (CKE = 0)

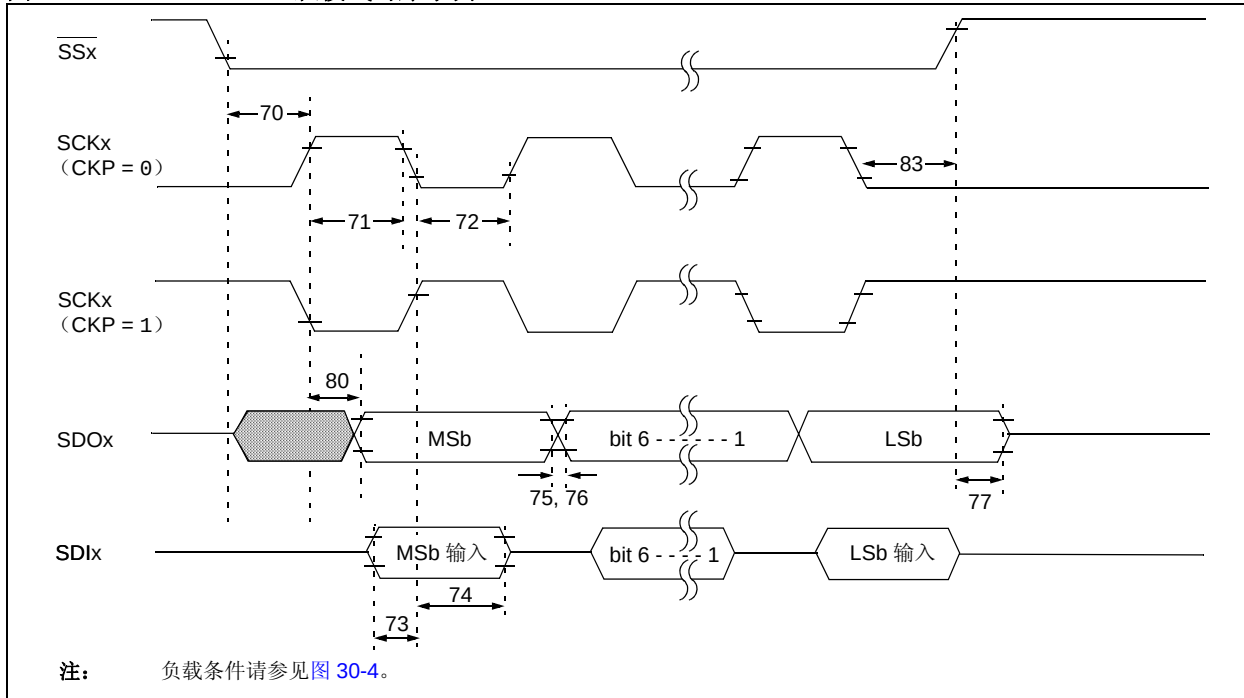


表 30-23: SPI 模式要求示例 (从模式时序, CKE = 0)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2sCH, TssL2sCL	SSx ↓ 到 SCKx ↓ 或 SCKx ↑ 输入的时间	3 Tcy	—	ns	
70A	TssL2WB	SSx ↓ 到写入 SSPxBUF 的时间	3 Tcy	—	ns	
71	Tsch	SCKx 输入高电平时间 (从模式)	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 (从模式)	连续	1.25 Tcy + 30	ns	
72A		单字节	40	—	ns	(注 1)
73	TdiV2sCH, TdiV2sCL	SDIx 数据输入到 SCKx 边沿的建立时间	25	—	ns	
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿之间的时间	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			100	—	ns	VDD = 2.15V
75	TdoR	SDOx 数据输出上升时间	—	25	ns	PORTB 或 PORTC
76	TdoF	SDOx 数据输出下降时间	—	25	ns	PORTB 或 PORTC
77	TssH2doZ	SSx ↑ 到 SDOx 输出高阻态的时间	10	70	ns	
80	Tsch2boV, TscL2boV	SCKx 边沿之后 SDOx 数据输出有效的的时间	—	50	ns	VDD = 3.3V, VDDCORE = 2.5V
			—	100	ns	VDD = 2.15V
83	Tsch2ssH, TscL2ssH	SCKx 边沿之后出现 SSx ↑ 的时间	1.5 Tcy + 40	—	ns	

注 1: 要求使用参数 73A。

注 2: 仅当使用参数 71A 和 72A 时。

图 30-16: SPI 从模式时序示例 (CKE = 1)

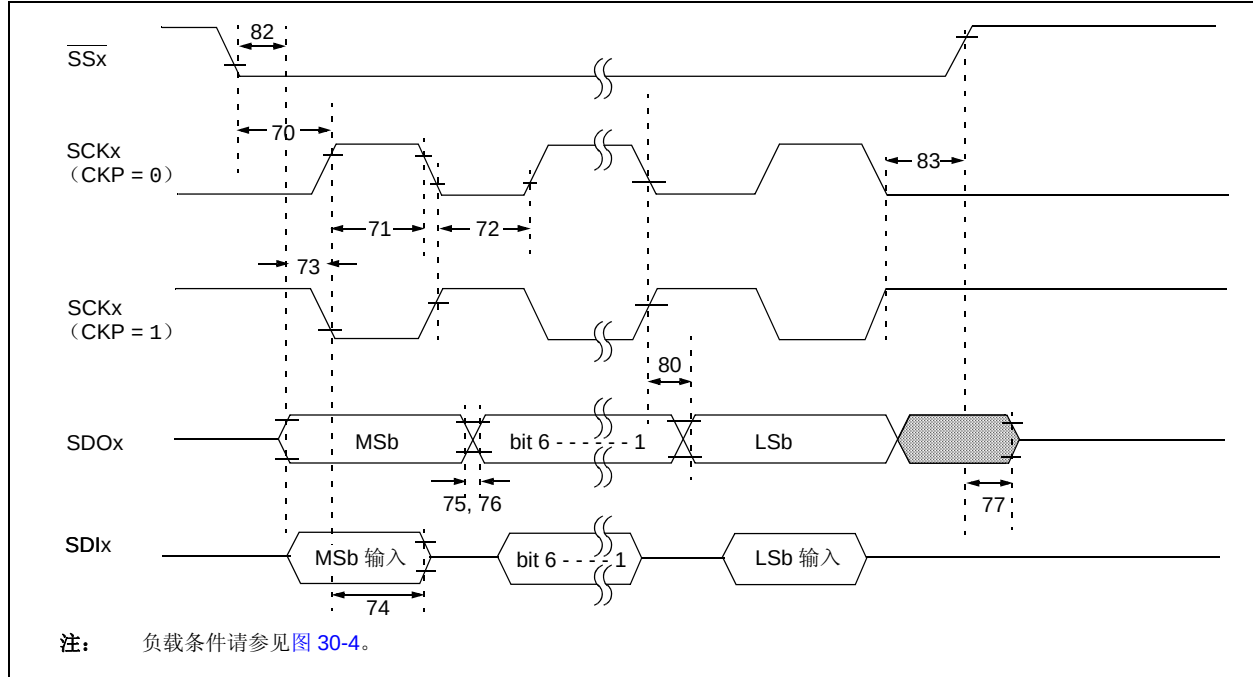


表 30-24: SPI 从模式要求示例 (CKE = 1)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2sCH, TssL2sCL	$\overline{SSx}$ ↓ 到 SCKx ↓ 或 SCKx ↑ 输入的时间	3 Tcy	—	ns	
70A	TssL2WB	$\overline{SSx}$ ↓ 到写入 SSPxBUF 的时间	3 Tcy	—	ns	
71	Tsch	SCKx 输入高电平时间 (从模式)	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 (从模式)	连续	1.25 Tcy + 30	ns	
72A		单字节	40	—	ns	(注 1)
73	TdIV2sCH, TdIV2sCL	SDIx 数据输入到 SCKx 边沿的建立时间	25	—	ns	
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿之间的时间	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2DiL, Tscl2DiL	SDIx 数据输入到 SCKx 边沿的保持时间	35	—	ns	VDD = 3.3V, VDDCORE = 2.5V
			100	—	ns	VDD = 2.15V
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	$\overline{SSx}$ ↑ 到 SDOx 输出高阻态的时间	10	70	ns	
80	Tsch2DoV, Tscl2DoV	SCKx 边沿之后 SDOx 数据输出有效的的时间	—	50	ns	VDD = 3.3V, VDDCORE = 2.5V
			—	100	ns	VDD = 2.15V
81	TDoV2sCH, TDoV2sCL	SDOx 数据输出建立到出现 SCKx 边沿的时间	Tcy	—	ns	
82	TssL2DoV	$\overline{SSx}$ ↓ 边沿之后 SDOx 数据输出有效的的时间	—	50	ns	
83	Tsch2ssH, Tscl2ssH	SCKx 边沿之后出现 $\overline{SSx}$ ↑ 的时间	1.5 Tcy + 40	—	ns	

注 1: 要求使用参数 73A。

注 2: 仅当使用参数 71A 和 72A 时。

PIC18F46J50 系列

图 30-17: I²C™ 总线启动位 / 停止位时序

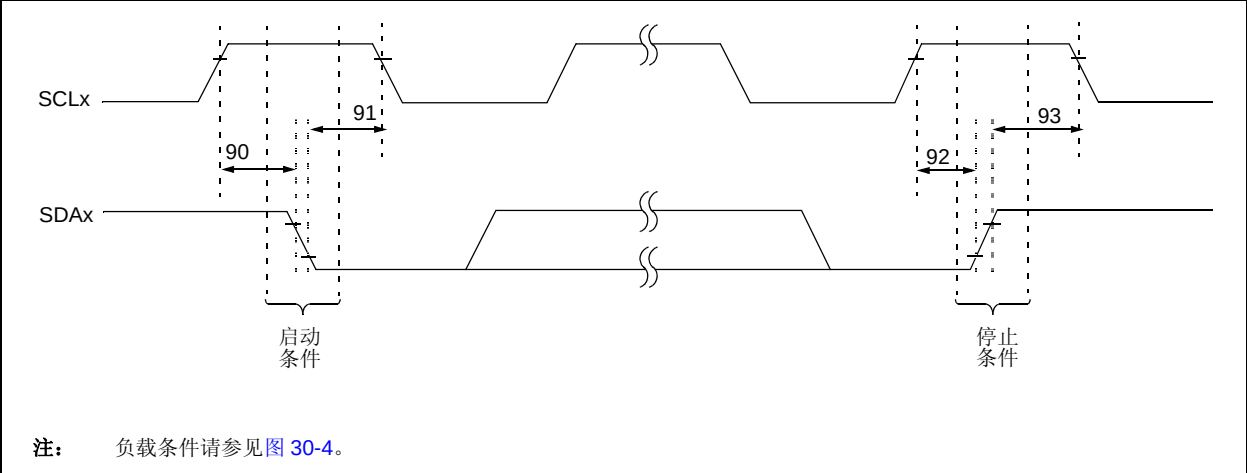


表 30-25: I²C™ 总线启动位 / 停止位要求（从模式）

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 模式	4700	—	ns	仅与重复启动条件相关
			400 kHz 模式	600	—		
91	THD:STA	启动条件保持时间	100 kHz 模式	4000	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz 模式	600	—		
92	TSU:STO	停止条件建立时间	100 kHz 模式	4700	—	ns	
			400 kHz 模式	600	—		
93	THD:STO	停止条件保持时间	100 kHz 模式	4000	—	ns	
			400 kHz 模式	600	—		

图 30-18: I²C™ 总线数据时序

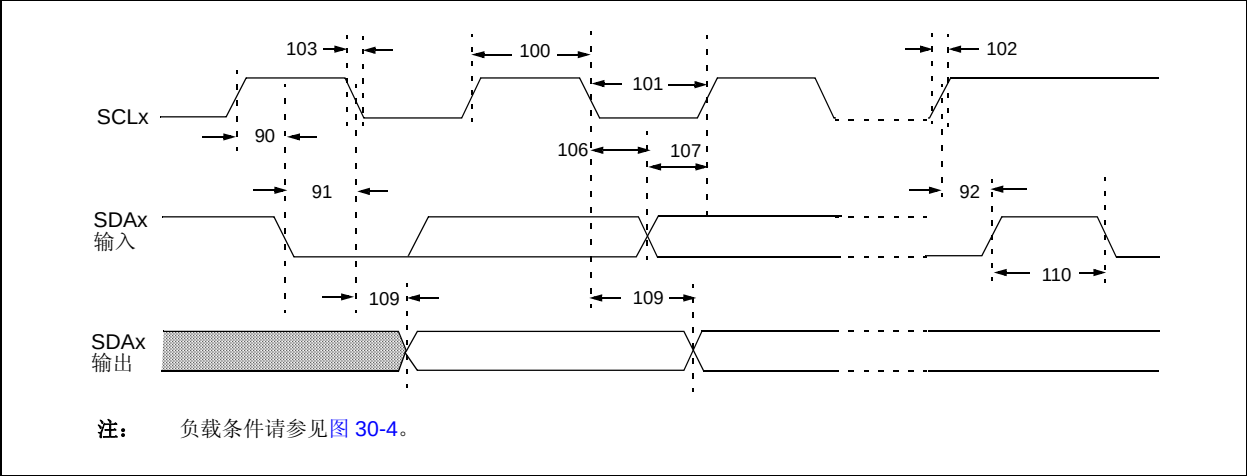


表 30-26: I²C™ 总线数据要求 (从模式)

参数编号	符号	特性	最小值	最大值	单位	条件
100	T _{HIGH}	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 T _{CY}	—	
101	T _{LOW}	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 T _{CY}	—	
102	T _R	SDA _x 和 SCL _x 上升时间	100 kHz 模式	—	1000	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns C _B 值规定在 10 至 400 pF 之间
103	T _F	SDA _x 和 SCL _x 下降时间	100 kHz 模式	—	300	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns C _B 值规定在 10 至 400 pF 之间
90	T _{SU: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	T _{HD: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	T _{HD:DAT}	数据输入保持时间	100 kHz 模式	0	—	ns
			400 kHz 模式	0	0.9	μs
107	T _{SU:DAT}	数据输入建立时间	100 kHz 模式	250	—	ns (注 2)
			400 kHz 模式	100	—	ns
92	T _{SU:STO}	停止条件建立时间	100 kHz 模式	4.7	—	μs
			400 kHz 模式	0.6	—	μs
109	T _{AA}	从时钟有效到输出有效的 时间	100 kHz 模式	—	3500	ns (注 1)
			400 kHz 模式	—	—	ns
110	T _{BUF}	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必须保持空闲的时间
			400 kHz 模式	1.3	—	μs
D102	C _B	总线容性负载	—	—	400	pF

注 1: 为避免产生意外的启动或停止条件，作为发送器的器件必须提供这个内部最小延时以补偿 SCL_x 下降沿的未定义区域（最小值 300 ns）。

2: 快速模式的 I²C™ 总线器件也可在标准模式的 I²C 总线系统上使用，但必须满足 T_{SU:DAT} ≥ 250 ns 的要求。如果快速模式器件没有延长 SCL_x 信号的低电平周期，则必然满足此条件。如果该器件延长了 SCL_x 信号的低电平周期，则在 SCL_x 线被释放前，它必须将下一个数据位输出到 SDA_x 线。根据标准模式 I²C 总线规范，T_R max. + T_{SU:DAT} = 1000 + 250 = 1250 ns。

PIC18F46J50 系列

图 30-19: MSSPx I²C™ 总线启动位 / 停止位时序波形

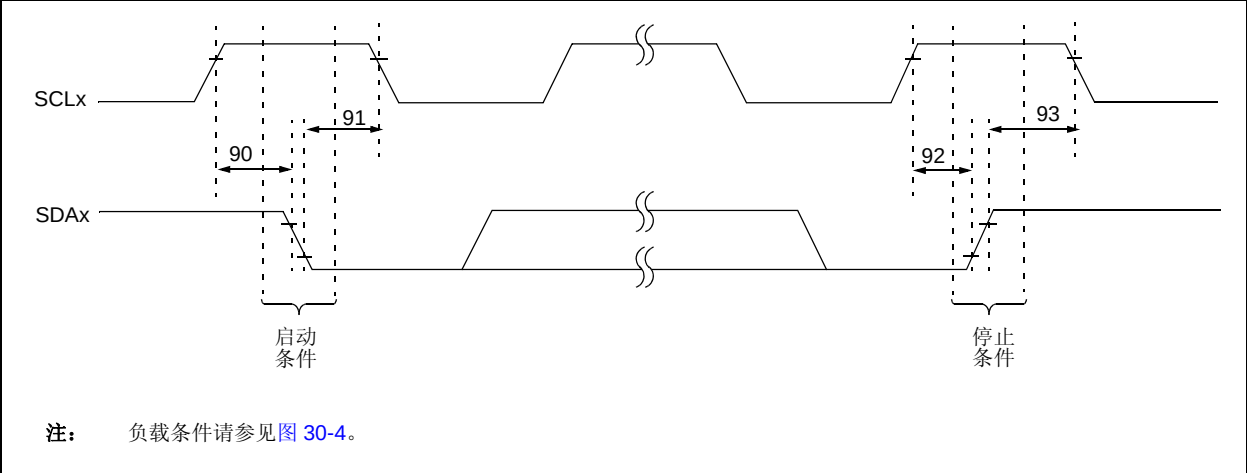


表 30-27: MSSPx I²C™ 总线启动位 / 停止位要求

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 模式	2(Tosc)(BRG + 1)	—	ns	仅与重复启动条件相关
			400 kHz 模式	2(Tosc)(BRG + 1)	—		
91	THD:STA	启动条件保持时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz 模式	2(Tosc)(BRG + 1)	—		
92	TSU:STO	停止条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ns	
			400 kHz 模式	2(Tosc)(BRG + 1)	—		
93	THD:STO	停止条件保持时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ns	
			400 kHz 模式	2(Tosc)(BRG + 1)	—		

图 30-20: MSSPx I²C™ 总线数据时序

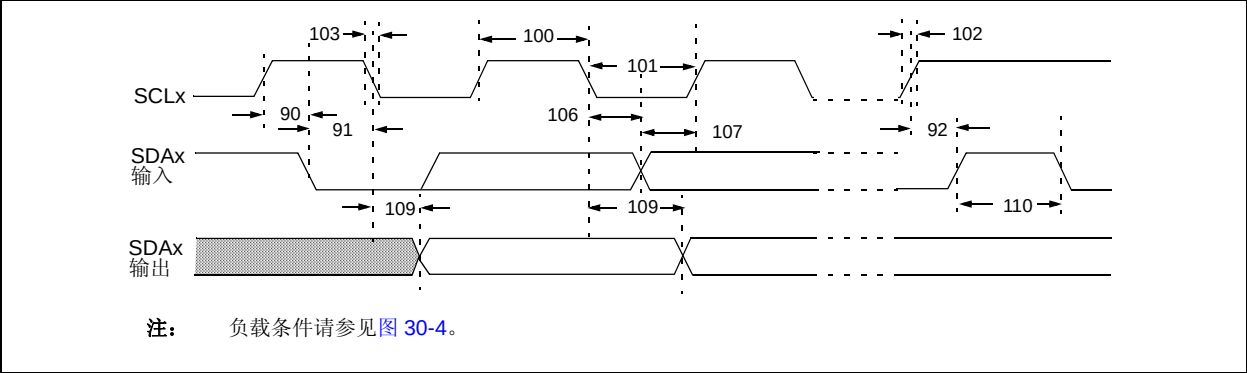


表 30-28: MSSPx I²C™ 总线数据要求

参数编号	符号	特性	最小值	最大值	单位	条件
100	T _{HIGH}	时钟高电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	μs
			400 kHz 模式	2(Tosc)(BRG + 1)	—	μs
101	T _{LOW}	时钟低电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	μs
			400 kHz 模式	2(Tosc)(BRG + 1)	—	μs
102	T _R	SDAx 和 SCLx 上升时间	100 kHz 模式	—	1000	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns
103	T _F	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns
90	T _{SU:STA}	启动条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	μs
			400 kHz 模式	2(Tosc)(BRG + 1)	—	μs
91	T _{HD:STA}	启动条件保持时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	μs
			400 kHz 模式	2(Tosc)(BRG + 1)	—	μs
106	T _{HD:DAT}	数据输入保持时间	100 kHz 模式	0	—	ns
			400 kHz 模式	0	0.9	μs
107	T _{SU:DAT}	数据输入建立时间	100 kHz 模式	250	—	ns
			400 kHz 模式	100	—	ns
92	T _{SU:STO}	停止条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	μs
			400 kHz 模式	2(Tosc)(BRG + 1)	—	μs
109	T _{AA}	从时钟有效到输出有效的时间	100 kHz 模式	—	3500	ns
			400 kHz 模式	—	1000	ns
110	T _{BUF}	总线空闲时间	100 kHz 模式	4.7	—	μs
			400 kHz 模式	1.3	—	μs
D102	C _B	总线容性负载	—	400	pF	

注 1: 快速模式的 I²C 总线器件也可在标准模式的 I²C 总线系统上使用, 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, 则必然满足此条件。如果该器件延长了 SCLx 信号的低电平周期, 则在 SCLx 线被释放前, 它必须将下一个数据位输出到 SDAx 线。在 100 kHz 模式下, 参数 102 + 参数 107 = 1000 + 250 = 1250 ns。

PIC18F46J50 系列

图 30-21: EUSARTx 同步发送（主 / 从）时序

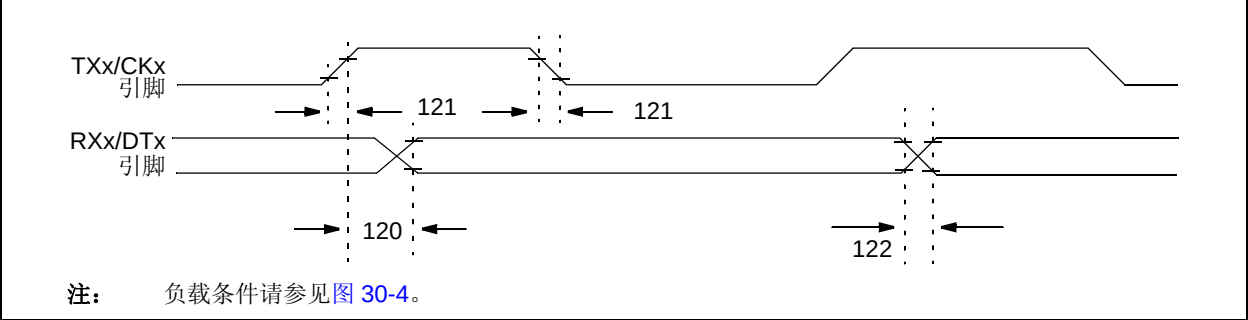


表 30-29: EUSARTx 同步发送要求

参数编号	符号	特性	最小值	最大值	单位	条件
120	TckH2DTV	同步发送（主 / 从） 时钟高电平到数据输出有效的的时间	—	40	ns	
121	TckRF	时钟输出上升时间和下降时间（主模式）	—	20	ns	
122	TdTRF	数据输出上升时间和下降时间	—	20	ns	

图 30-22: EUSARTx 同步接收（主 / 从）时序

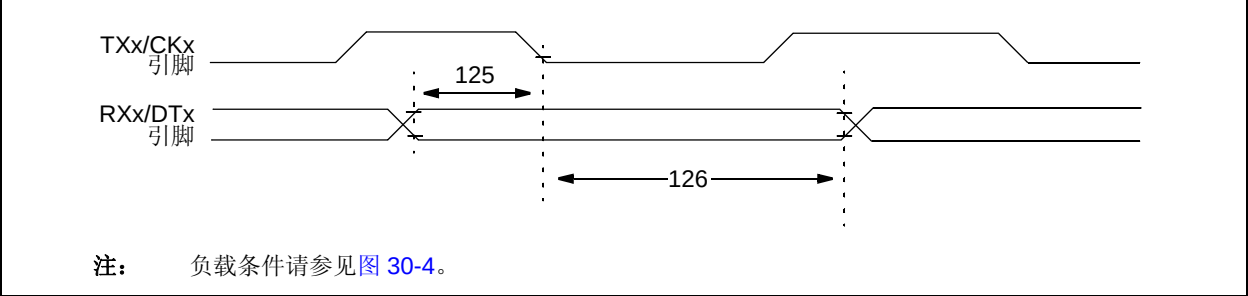


表 30-30: EUSARTx 同步接收要求

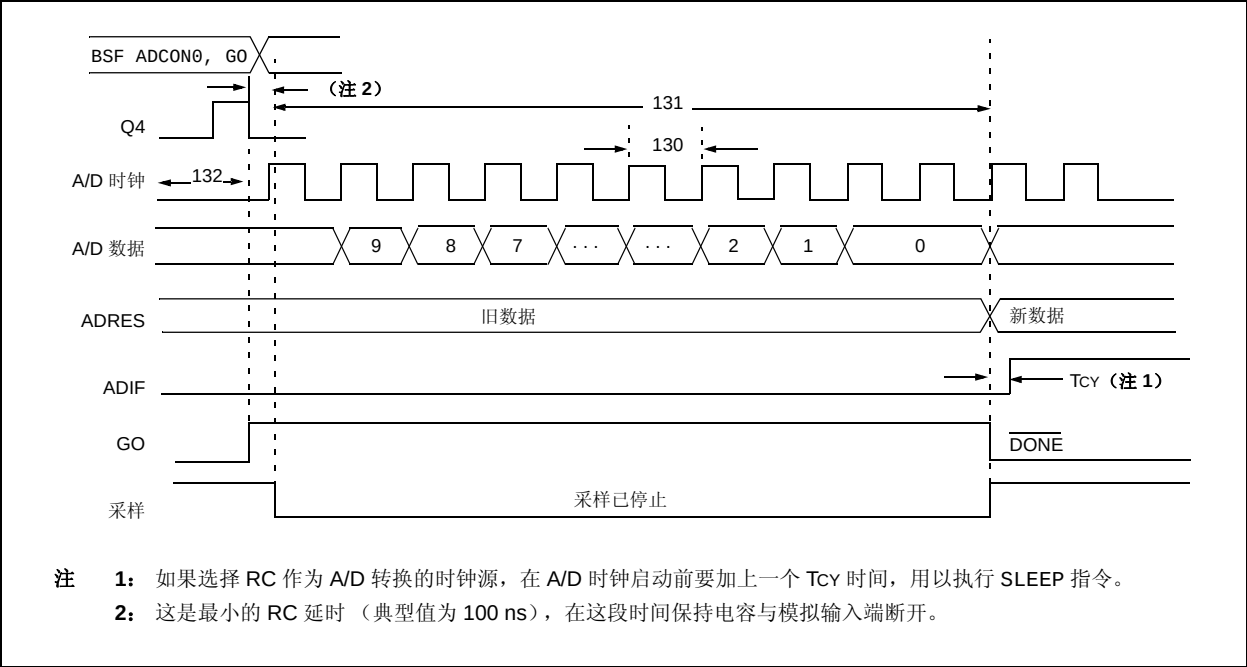
参数编号	符号	特性	最小值	最大值	单位	条件
125	TdTV2CKL	同步接收（主 / 从） CKx ↓ 之前数据的保持时间（DTx 保持时间）	10	—	ns	
126	TckL2DTL	CKx ↓ 之后数据的保持时间（DTx 保持时间）	15	—	ns	

表 30-31: A/D 转换器特性: PIC18F46J50 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	$\Delta V_{REF} \geq 3.0V$
A03	EIL	积分线性误差	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A04	EDL	微分线性误差	—	—	$<\pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A06	EOFF	失调误差	—	—	$<\pm 3$	LSb	$\Delta V_{REF} \geq 3.0V$
A07	EGN	增益误差	—	—	$<\pm 3.5$	LSb	$\Delta V_{REF} \geq 3.0V$
A10		单调性	保证 ⁽¹⁾			—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	ΔV_{REF}	参考电压范围 ($V_{REFH} - V_{REFL}$)	2.0	—	—	V	$V_{DD} < 3.0V$
			3	—	—	V	$V_{DD} \geq 3.0V$
A21	V_{REFH}	参考电压高电压	V_{REFL}	—	$V_{DD} + 0.3V$	V	
A22	V_{REFL}	参考电压低电压	$V_{SS} - 0.3V$	—	V_{REFH}	V	
A25	V_{AIN}	模拟输入电压	V_{REFL}	—	V_{REFH}	V	
A30	Z_{AIN}	模拟信号源的推荐阻抗	—	—	2.5	k Ω	
A50	I_{REF}	V_{REF} 输入电流 ⁽²⁾	—	—	5	μA	在采集 V_{AIN} 期间。 在 A/D 转换周期期间。
			—	—	150	μA	

- 注 1: A/D 转换结果不会因输入电压的增加而减小, 并且不会丢失编码。
- 注 2: V_{REFH} 电流来自选择作为 V_{REFH} 源的 $RA3/AN3/V_{REF+}/C1INB$ 引脚或 V_{DD} 。
 V_{REFL} 电流来自选择作为 V_{REFL} 源的 $RA2/AN2/V_{REF-}/C2INB$ 引脚或 V_{SS} 。

图 30-23: A/D 转换时序



PIC18F46J50 系列

表 30-32: 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
131	TCNV	转换时间（不包括采集时间） ⁽²⁾	11	12	TAD	
132	TACQ	采集时间 ⁽³⁾	1.4	—	μs	-40°C 至 +85°C
135	TSWC	转换 → 采样的切换时间	—	(注 4)		
137	TDIS	电容放电时间	0.2	—	μs	

- 注 1: A/D 时钟周期取决于器件频率和 TAD 时钟分频比。
- 2: ADRES 寄存器可在下一个 Tcy 周期被读取。
- 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。在输入通道上的信号源阻抗（Rs）为 50Ω。
- 4: 在器件时钟的下一个周期。

图 30-24: USB 信号时序

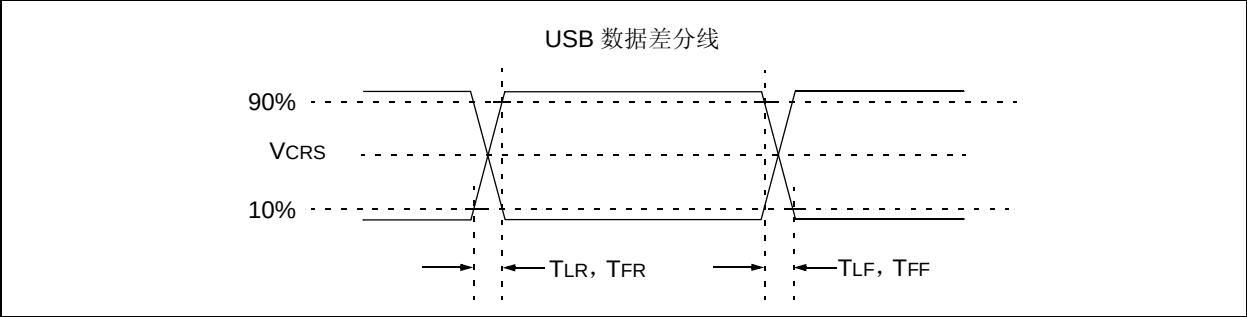


表 30-33: USB 低速时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
	TLR	信号跳变上升时间	75	—	300	ns	CL = 200 至 600 pF
	TLF	信号跳变下降时间	75	—	300	ns	CL = 200 至 600 pF
	TLRFM	上升 / 下降时间匹配	80	—	125	%	

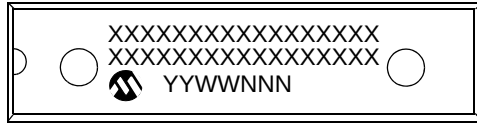
表 30-34: USB 全速要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
	TFR	信号跳变上升时间	4	—	20	ns	CL = 50 pF
	TFF	信号跳变下降时间	4	—	20	ns	CL = 50 pF
	TFRFM	上升 / 下降时间匹配	90	—	111.1	%	

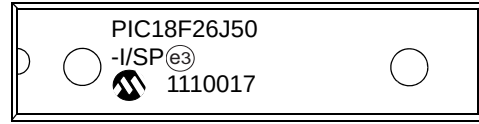
31.0 封装信息

31.1 封装标识信息

28 引脚 SPDIP



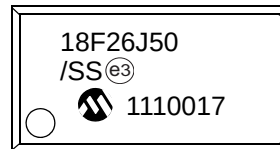
示例



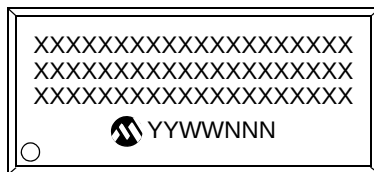
28 引脚 SSOP



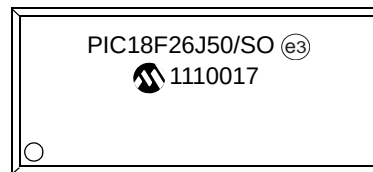
示例



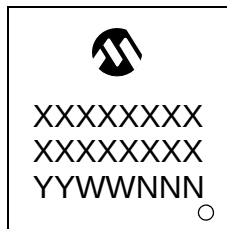
28 引脚 SOIC (0.300 英寸)



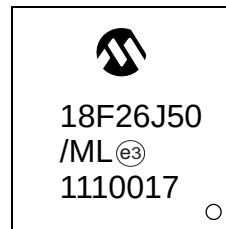
示例



28 引脚 QFN



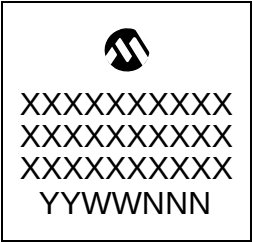
示例



图注:	XX...X 客户指定信息 Y 年份代码 (日历年的最后一位数字) YY 年份代码 (日历年的最后两位数字) WW 星期代码 (一月一日的星期代码为“01”) NNN 以字母数字排序的追踪代码 * 雾锡 (Matte Tin, Sn) 的 JEDEC 无铅标志 本封装为无铅封装。JEDEC 无铅标志 (e3) 标示于此种封装的外包装上。
注:	Microchip 元器件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

PIC18F46J50 系列

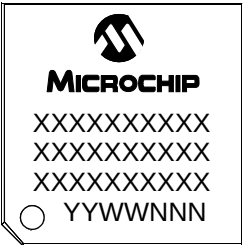
44 引脚 QFN



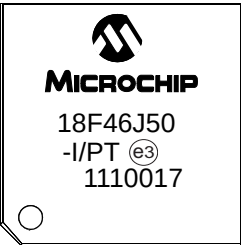
示例



44 引脚 TQFP



示例

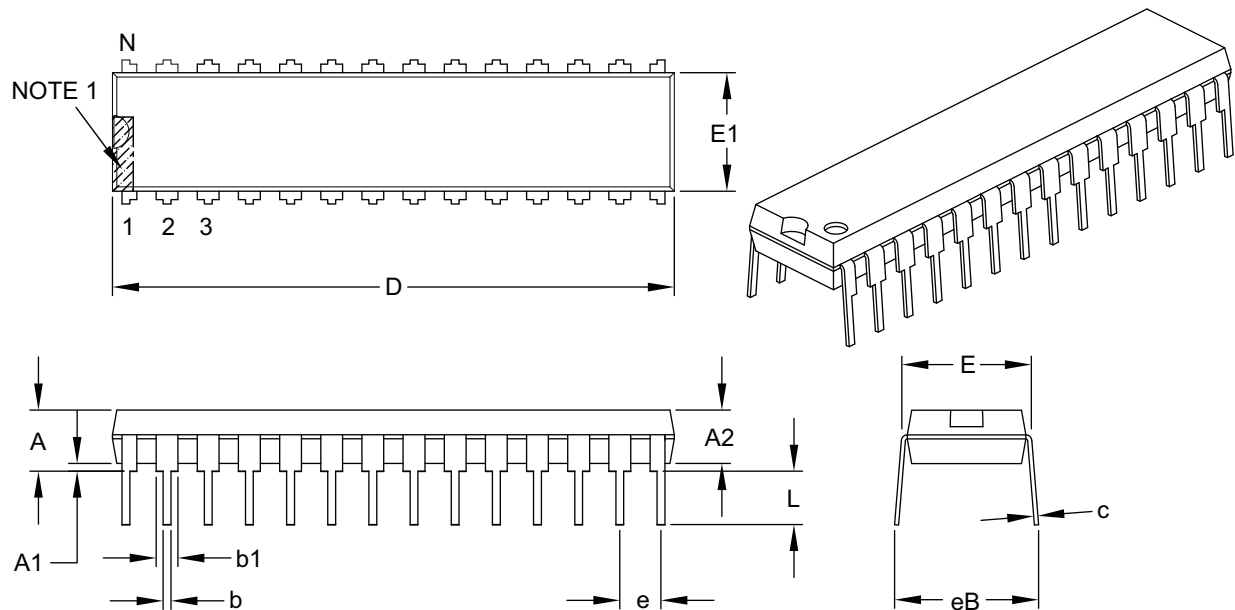


31.2 封装详细信息

以下部分将介绍各种封装的技术细节。

28引脚窄型塑封双列直插式封装（SP）——主体300 mil [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	—	—	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	—	—
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	—	—	.430

Notes:

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

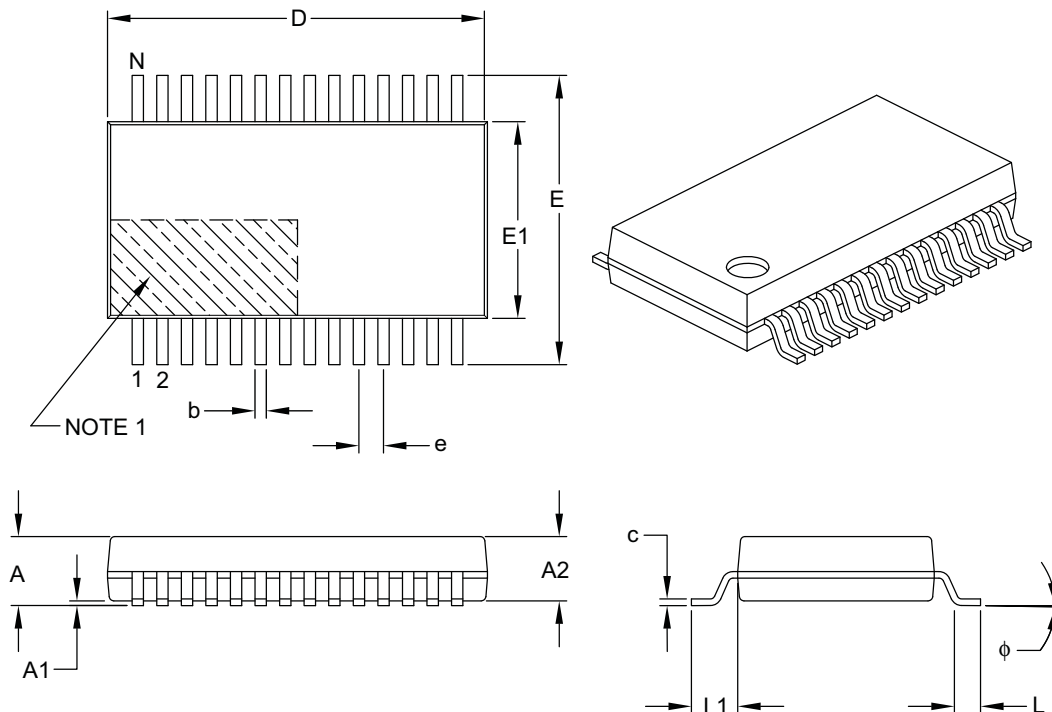
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

PIC18F46J50 系列

28引脚塑封缩小型小外形封装（SS）——主体5.30 mm [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	—	—	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	—	—
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	—	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	—	0.38

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

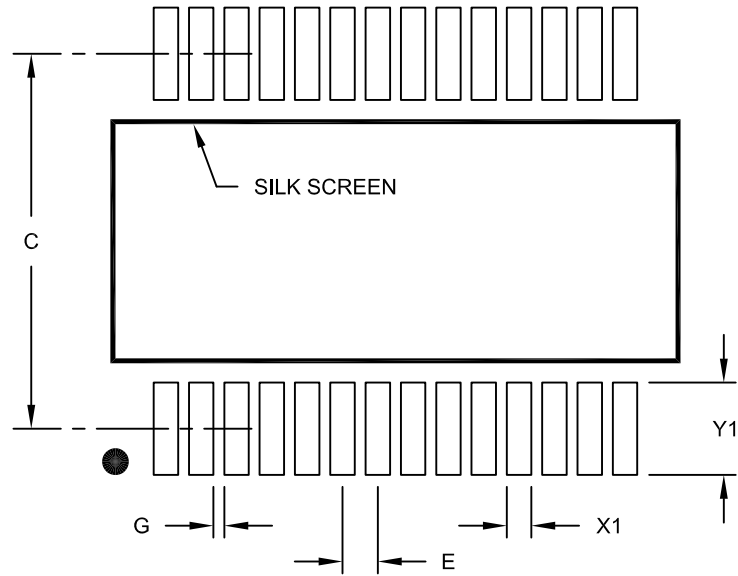
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-073B

28 引脚塑封缩小型小外形封装 (SS) —— 主体 5.30 mm [SSOP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

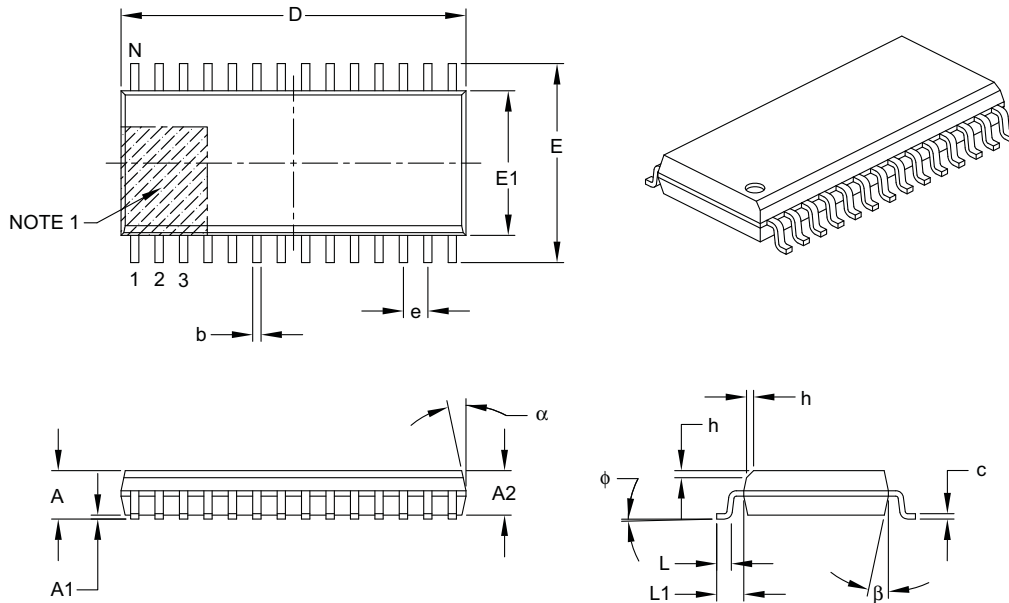
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

PIC18F46J50 系列

28引脚塑封宽条小外形封装（SO）——主体7.50 mm [SOIC]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	1.27 BSC		
Overall Height	A	—	—	2.65
Molded Package Thickness	A2	2.05	—	—
Standoff §	A1	0.10	—	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (optional)	h	0.25	—	0.75
Foot Length	L	0.40	—	1.27
Footprint	L1	1.40 REF		
Foot Angle Top	φ	0°	—	8°
Lead Thickness	c	0.18	—	0.33
Lead Width	b	0.31	—	0.51
Mold Draft Angle Top	α	5°	—	15°
Mold Draft Angle Bottom	β	5°	—	15°

Notes:

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

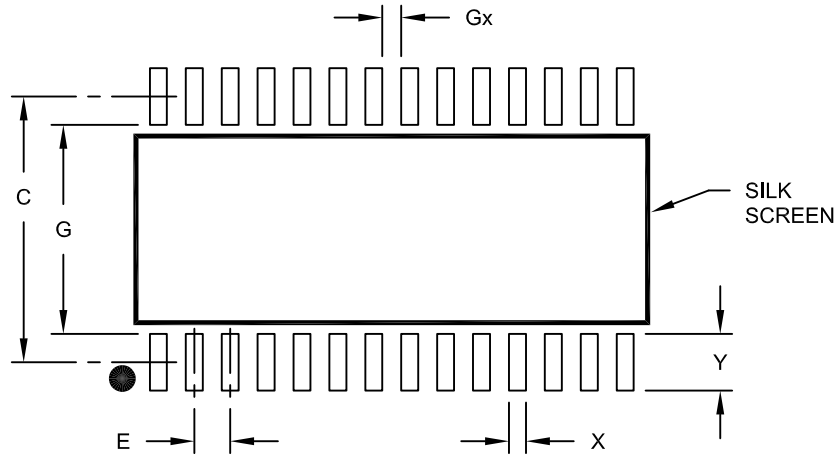
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-052B

28 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

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



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

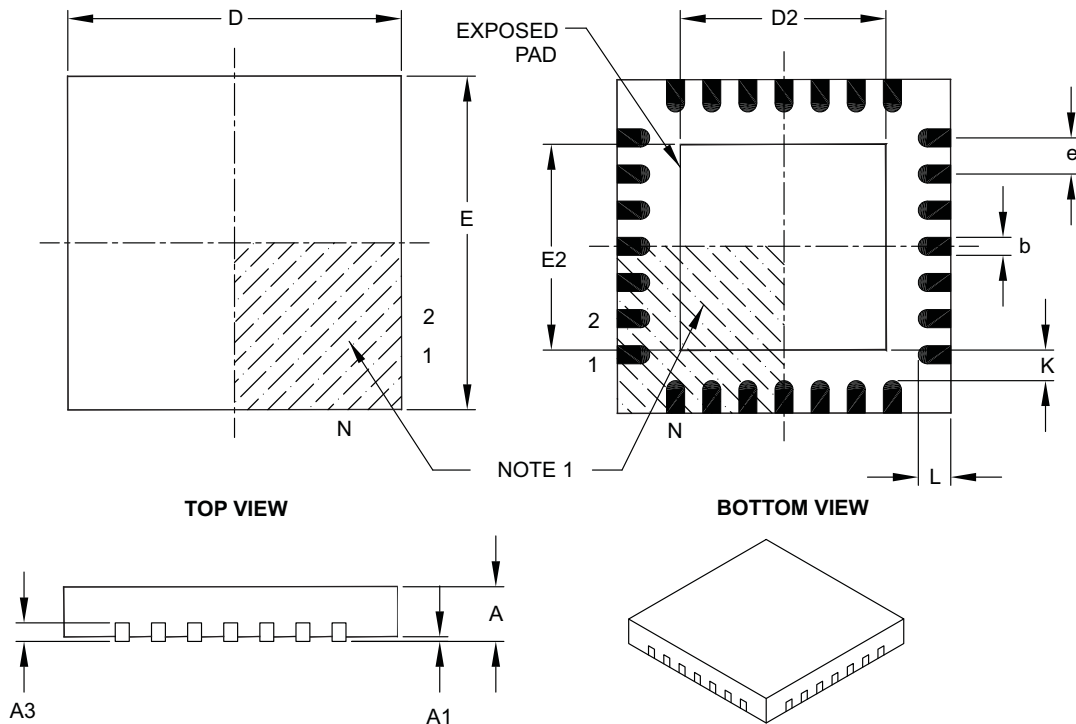
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

PIC18F46J50 系列

28引脚塑封正方扁平无脚封装（ML）——主体6x6 mm [QFN]，触点长度为0.55 mm

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	6.00 BSC		
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D	6.00 BSC		
Exposed Pad Length	D2	3.65	3.70	4.20
Contact Width	b	0.23	0.30	0.35
Contact Length	L	0.50	0.55	0.70
Contact-to-Exposed Pad	K	0.20	—	—

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

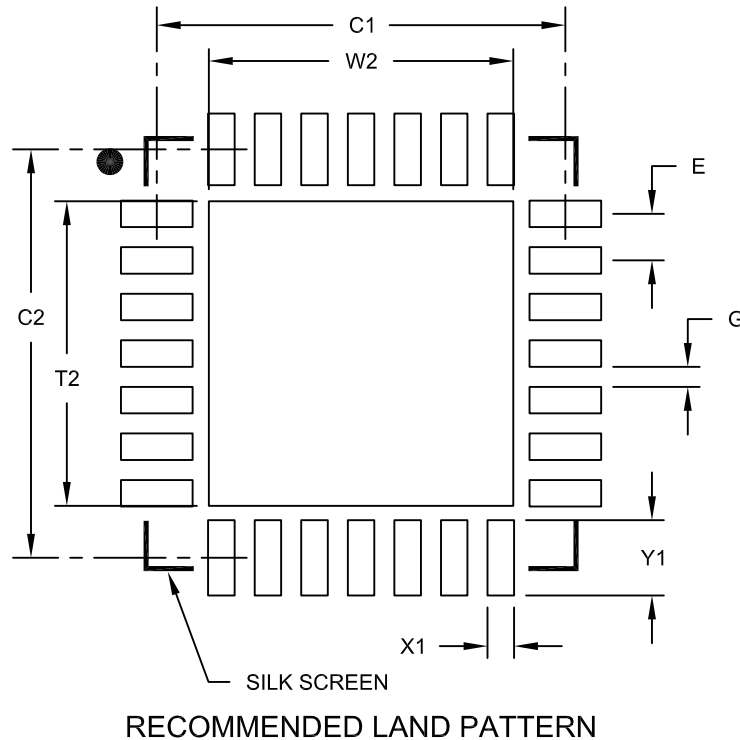
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-105B

28引脚塑封正方扁平无脚封装（ML）——主体6x6 mm [QFN]，触点长度为0.55 mm

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

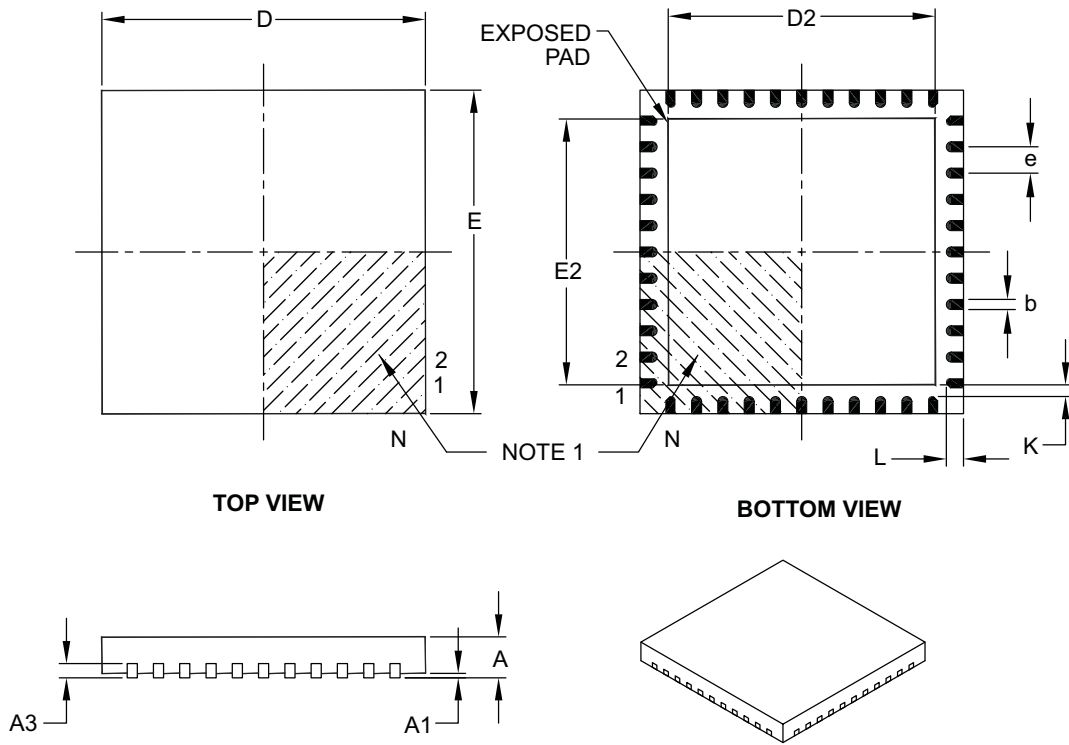
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

PIC18F46J50 系列

44引脚塑封正方扁平无脚封装（ML）——主体8x8 mm [QFN]

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



	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	44		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	8.00 BSC		
Exposed Pad Width	E2	6.30	6.45	6.80
Overall Length	D	8.00 BSC		
Exposed Pad Length	D2	6.30	6.45	6.80
Contact Width	b	0.25	0.30	0.38
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	—	—

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

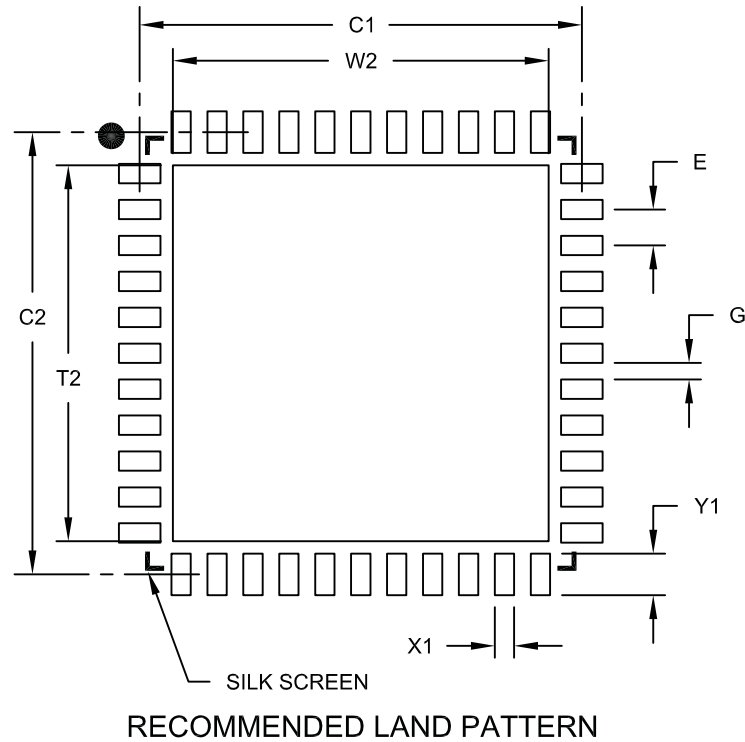
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-103B

44引脚塑封正方扁平无脚封装（ML）——主体8x8 mm [QFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			6.80
Optional Center Pad Length	T2			6.80
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.80
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

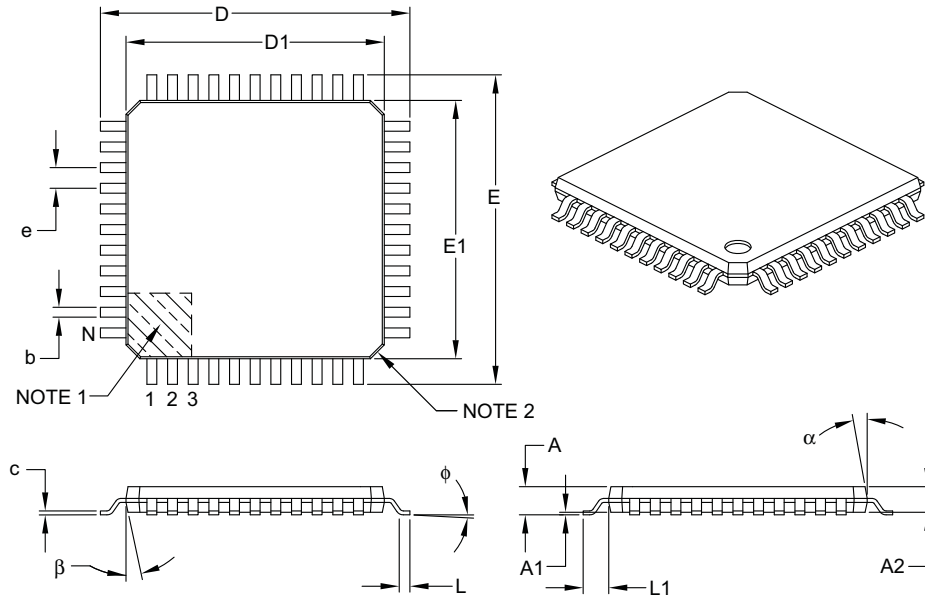
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103A

PIC18F46J50 系列

44引脚塑封薄型正方扁平封装（PT）——主体10x10x1 mm， 2.00 mm [TQFP]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Leads	N	44		
Lead Pitch	e	0.80 BSC		
Overall Height	A	—	—	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	—	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E	12.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Thickness	c	0.09	—	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

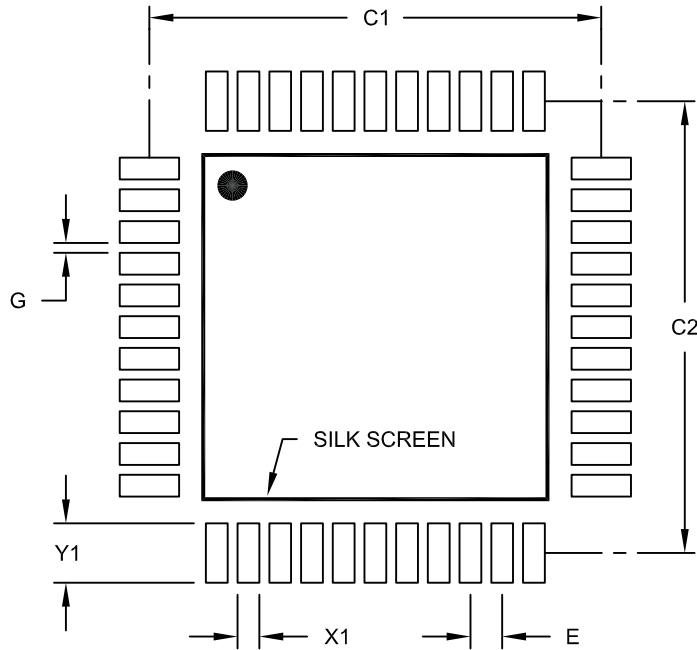
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

Microchip Technology Drawing C04-076B

44引脚塑封薄型正方扁平封装（PT）——主体10x10x1 mm， 2.00 mm [TQFP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.80 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2076A

PIC18F46J50 系列

注:

附录 A： 版本历史

版本 A（2008 年 9 月）

PIC18F46J50 系列器件的原始数据手册。

版本 B（2009 年 3 月）

对电气特性部分进行了修改，并通篇做了少量文字编辑。

版本 C（2009 年 10 月）

删除了“初稿”标记。

版本 D（2011 年 3 月）

添加了第 2.0 节“PIC18FJ 单片机入门指南”。将 CTEDG1 和 CTEDG2 引脚功能分别更改为 CTED1 和 CTED2。通篇阐明了部分内容并做了少量文字编辑。

附录 B： 器件差异

表 B-1 为本数据手册中所述器件间的差异。

表 B-1： PIC18F46J50 系列中各器件间差异

特性	PIC18F24J50	PIC18F25J50	PIC18F26J50	PIC18F44J50	PIC18F45J50	PIC18F46J50
程序存储器	16K	32K	64K	16K	32K	64K
程序存储器（指令）	8,192	16,384	32,768	8,192	16,384	32,768
I/O 端口（引脚）	端口 A、B 和 C			端口 A、B、C、D 和 E		
10 位 ADC 模块	10 路输入通道			13 路输入通道		
封装	28 引脚 QFN、SOIC、SSOP 和 SPDIP（300 mil）			44 引脚 QFN 和 TQFP		

PIC18F46J50 系列

注:

索引

A

A/D	347
A/D 转换器中断, 配置	351
ADCAL 位	355
ADRESH 寄存器	350
采集要求	352
ECCP2 触发信号的使用	354
模拟端口引脚, 配置	353
配置模块	351
特殊事件触发信号 (ECCPx)	354
相关的寄存器	356
校准	355
在功耗管理模式下的操作	355
转换	354
转换器特性	529
转换时钟 (TAD)	353
转换要求	530
转换状态 (GO/DONE 位)	350
自动采集时间	353
ACKSTAT	313
ACKSTAT 状态标志	313
ADCAL 位	355
ADCON0 寄存器 GO/DONE 位	350
ADDFSR	478
ADDLW	441
ADDWF	441
ADDWFC	442
ADDULNK	478
ADRESL 寄存器	350
ANDLW	442
ANDWF	443
B	
BC	443
BCF	444
BF	313
BF 状态标志	313
BN	444
BNC	445
BNN	445
BNOV	446
BNZ	446
BOR. 请参见欠压复位。	
BOV	449
BRA	447
BRG. 请参见波特率发生器。	
BSF	447
BTFSC	448
BTFSS	448
BTG	449
BZ	450
版本历史	545
比较 (ECCP 模块)	249
CCPRx 寄存器	249
软件中断	249
Timer1/Timer3 模式选择	249
特殊事件触发信号	221, 249
引脚配置	249
比较 (ECCPx 模块) 特殊事件触发信号	354

比较器	385
复位的影响	390
工作原理	387
寄存器	385
控制	388
模拟输入连接注意事项	387
配置	388
使能和输出选择	388
使能和输入选择	388
相关的寄存器	390
响应时间	387
休眠期间的操作	390
中断	389
比较器参考电压	391
复位的影响	393
精度和误差	393
连接注意事项	393
配置	392
相关的寄存器	393
休眠期间的操作	393
比较器规范	504
编程, 器件指令	435
变更通知客户服务	558
表读 / 表写	81
表指针操作 (表)	106
并行主端口 (PMP)	169
从端口模式	178
模块寄存器	170
数据寄存器	176
相关的寄存器	193
应用示例	191
主端口模式	183
波特率发生器	309
捕捉 (ECCP 模块)	248
CCPRxH:CCPRxL 寄存器	248
ECCP 引脚配置	248
软件中断	248
Timer1/Timer3 模式选择	248
预分频器	248

C

CALL	450
CALLW	479
C 编译器	
MPLAB C18	486
CLRF	451
CLRWDT	451
COMF	452
CPFSEQ	452
CPFSGT	453
CPFSLT	453
CPU 的特殊功能	417
CTMU	
测量电容	409
测量时间	411
产生延时	412
初始化	403
复位的影响	412
工作原理	402
空闲模式期间的操作	412
相关的寄存器	415
校准	403
休眠模式期间的操作	412
CTMU 电流源规范	505

PIC18F46J50 系列

参考电压规范	505	双字指令	83
参考时钟输出	44	写闪存程序存储器	110
产品标识体系	560	与 +5V 系统通信	133
超低功耗唤醒	60	装载 SSP1BUF	
程序存储器		(SSP1SR) 寄存器	273
ALU		低功耗模式	47,242
状态位	96	超低功耗唤醒	60
查找表	81	汇总 (表)	48
存储器硬件编码向量	78	空闲模式	52
存储器映射	77	PRI_IDLE	52
硬件编码向量和配置字	78	RC_IDLE	53
复位向量	78	SEC_IDLE	52
扩展指令集	99	SLEEP 命令的多种功能	48
闪存配置字	78	深度休眠模式	54
指令数	83	典型序列	56
双字	83	和 RTCC 外设	56
中断向量	78	唤醒源	55
程序计数器	79	寄存器	57
PCL、PCH 和 PCU 寄存器	79	看门狗定时器 (DSWDT)	55
PCLATH 和 PCLATU 寄存器	79	欠压复位 (DSBOR)	56
程序校验和代码保护	433	准备	54
串行时钟, SCKx	270	时钟转换和状态指示	48
串行数据输出 (SDOx)	270	退出空闲和休眠模式	53
串行数据输入 (SDIx)	270	通过复位	53
串行外设接口。请参见 SPI 模式。		通过 WDT 超时	53
从选择 (SSx)	270	通过中断	53
存储器编程要求	504	在没有振荡器起	
存储器构成	77	振延时的情况下	54
程序存储器	77	休眠模式	51
返回地址堆栈	79	运行模式	48
数据存储器	84	PRI_RUN	48
D		RC_RUN	50
DAW	454	SEC_RUN	48
DCFSNZ	455	电气特性	489
DECf	454	绝对最大额定值	489
DECFSZ	455	直流特性	491 - 502
代码保护	417	读者反馈表	559
代码示例		对标准 PIC MCU 指令的影响	482
16 x 16 无符号乘法程序	114	堆栈满 / 下溢复位	81
16 x 16 有符号乘法程序	114	E	
512 字节 SPI 主模式初始化和传输	286	EUSART	323
8 x 8 无符号乘法程序	113	波特率发生器	
8 x 8 有符号乘法程序	113	在功耗管理模式下的操作	327
A/D 校准程序	355	波特率发生器 (BRG)	327
擦除闪存程序存储器	108	波特率, 异步模式	329
CTMU 校准设置程序	405	采样	327
超低功耗唤醒初始化	61	高波特率选择	
初始化 PORTA	136	(BRGH 位)	327
初始化 PORTB	139	公式	327
初始化 PORTC	143	相关的寄存器	328
初始化 PORTD	146	自动波特率检测	331
初始化 PORTE	148	同步从模式	345
电流校准程序	406	发送	345
电容校准程序	408	接收	346
读闪存程序存储器的一个字	107	相关的寄存器, 发送	345
改变捕捉预分频比	248	相关的寄存器, 接收	346
计算波特率误差	328	同步主模式	341
将 STATUS、WREG 和 BSR		发送	341
寄存器的值保存在 RAM 中	130	接收	343
快速寄存器堆栈	81	相关的寄存器, 发送	342
配置 EUSART2 输入和输出功能	154	相关的寄存器, 接收	344
容性触摸开关程序	410	异步模式	333
使用间接寻址清零 RAM		12 位间隔字符发送和接收	340
(Bank 1)	97	发送器	333
使用偏移量值的计算 GOTO	81	接收器	336

设置带有地址检测		故障概述	
功能的 9 位模式	336	对比表	4
同步间隔自动唤醒	338	H	
相关的寄存器，发送	335	汇编器	
相关的寄存器，接收	338	MPASM 汇编器	486
F		J	
FSCM。请参见故障保护时钟监视器。		I/O 端口	131
返回地址堆栈	79	漏极开路输出	133
相关的寄存器	79	TTL 输入缓冲器选项	133
封装		引脚功能	132
标识	531	I/O 引脚说明	
详细信息	533	PIC18F2XJ50（28 引脚）	16
复位	63,417	PIC18F4XJ50（44 引脚）	22
堆栈满复位	63	I ² C 模式	288
堆栈下溢复位	63	I ² C 模式（MSSP）	
寄存器的状态	68	波特率发生器	309
看门狗定时器（WDT）复位	63	串行时钟（SCLx 引脚）	296
快速寄存器堆栈	81	从模式	293
MCLR	65	地址掩码模式	
MCLR 复位，功耗管理模式	63	5 位	294
MCLR 复位，正常工作	63	7 位	295
配置不匹配（CM）	63	发送	296
配置不匹配复位	66	接收	296
欠压复位	65	寻址	293
欠压复位（BOR）	63,417	读 / 写位信息（R/W 位）	293, 296
RESET 指令	63	多主器件模式	317
上电复位	65	多主器件通信、总线冲突和仲裁	317
上电复位（POR）	63,417	复位的影响	317
上电延时定时器	66	工作原理	293
上电延时定时器（PWRT）	417	广播呼叫地址支持	307
深度休眠	63	寄存器	288
振荡器起振定时器（OST）	417	使用 BRG 的 I ² C 时钟频率	310
G		时钟同步和 CKP 位	304
GOTO	456	时钟延长	303
高 / 低压检测	395	10 位从发送模式	303
电流消耗	397	10 位从接收模式（SEN = 1）	303
典型应用	399	7 位从发送模式	303
复位的影响	400	7 位从接收模式（SEN = 1）	303
工作原理	396	时钟仲裁	311
休眠期间	400	停止条件时序	316
启动时间	397	相关的寄存器	322
设置	397	休眠模式下的操作	317
特性	507	应答序列时序	316
相关的寄存器	400	主模式	308
应用	399	发送	313
功耗管理模式		工作原理	309
和 EUSART 操作	327	接收	313
和 PWM 操作	267	启动条件时序	311
和 SPI 操作	278	重复启动条件时序	312
进入	47	总线冲突	
时钟源	47	停止条件期间	321
选择	47	重复启动条件期间	320
公式		寄存器	
A/D 采集时间	352	ADCON0（A/D 控制 0）	347
A/D 最小充电时间	352	ADCON1（A/D 控制 1）	348
对于给定 DMABC 发送的字节数	284	ALRMCFG（闹钟配置）	229
估算 USB 收发器的电流消耗	379	ALRMDAY（闹钟日值）	234
计算所需的最小采集时间	352	ALRMHR（闹钟小时值）	235
计算 USB 收发器电流	380	ALRMMIN（闹钟分钟值）	236
固件指令	435	ALRMMNTH（闹钟月值）	234
故障保护时钟监视器	417, 431	ALRMRPRT（闹钟校准）	230
功耗管理模式下的中断	433	ALRMSEC（闹钟秒值）	236
POR 或从休眠中唤醒	433	ALRMWD（闹钟星期值）	235
振荡器故障期间的 WDT	432	ANCON0（A/D 端口配置 0）	349

PIC18F46J50 系列

ANCON1 (A/D 端口配置 1)	349	PMADDRH (并行端口地址高字节)	177
BAUDCONx (波特率控制)	326	PMADDRL (并行端口地址低字节)	177
BDnSTAT	367	PMCONH (并行端口控制高字节)	170
BDnSTAT		PMCONL (并行端口控制低字节)	171
(缓冲区描述符 n 状态, CPU 模式)	368	PMEH (并行端口使能高字节)	174
BDnSTAT		PMEL (并行端口使能低字节)	174
(缓冲区描述符 n 状态, SIE 模式)	369	PMMODEH (并行端口模式高字节)	172
BDnSTAT (SIE 模式)	369	PMMODEL (并行端口模式低字节)	173
保留	231	PMSTATH (并行端口状态高字节)	175
并行主端口	170	PMSTATL (并行端口状态低字节)	175
CCPxCON		PORTE	148
(增强型捕捉 / 比较 / PWM x 控制)	246	PPSCON (外设引脚选择输入 0)	155
CMSTAT (比较器状态)	386	PSTRxCON (脉冲转向控制)	265
CMxCON (比较器控制 x)	386	RCON (复位控制)	64,129
CONFIG1H (配置 1 高字节)	420	RCSTAx (接收状态和控制)	325
CONFIG1L (配置 1 低字节)	419	REF0CON (参考时钟输出)	44
CONFIG2H (配置 2 高字节)	422	RPINR1 (外设引脚选择输入 1)	156
CONFIG2L (配置 2 低字节)	421	RPINR12 (外设引脚选择输入 12)	158
CONFIG3H (配置 3 高字节)	424	RPINR13 (外设引脚选择输入 13)	158
CONFIG3L (配置 3 低字节)	423	RPINR16 (外设引脚选择输入 16)	159
CONFIG4H (配置 4 高字节)	425	RPINR17 (外设引脚选择输入 17)	159
CONFIG4L (配置 4 低字节)	424	RPINR2 (外设引脚选择输入 2)	156
CTMUCONH (CTMU 控制高字节)	413	RPINR21 (外设引脚选择输入 21)	159
CTMUCONL (CTMU 控制低字节)	414	RPINR22 (外设引脚选择输入 22)	160
CTMUICON (CTMU 电流控制)	415	RPINR23 (外设引脚选择输入 23)	160
CVRCON (比较器参考电压控制)	392	RPINR24 (外设引脚选择输入 24)	160
DAY (日值)	232	RPINR3 (外设引脚选择输入 3)	156
DEVID1 (器件 ID 1)	425	RPINR4 (外设引脚选择输入 4)	157
DEVID2 (器件 ID 2)	426	RPINR6 (外设引脚选择输入 6)	157
DMACON1 (DMA 控制 1)	282	RPINR7 (外设引脚选择输入 7)	157
DMACON2 (DMA 控制 2)	283	RPINR8 (外设引脚选择输入 8)	158
DSCONH (深度休眠控制高字节)	57	RPOR0 (外设引脚选择输出 0)	161
DSCONL (深度休眠控制低字节)	57	RPOR1 (外设引脚选择输出 1)	161
DSGPR0 (深度休眠持久通用 0)	58	RPOR10 (外设引脚选择输出 10)	164
DSGPR1 (深度休眠持久通用 1)	58	RPOR11 (外设引脚选择输出 11)	164
DSWAKEH (深度休眠唤醒高字节)	59	RPOR12 (外设引脚选择输出 12)	165
DSWAKEL (深度休眠唤醒低字节)	59	RPOR13 (外设引脚选择输出 13)	165
ECCPxAS (ECCPx 自动关闭控制)	261	RPOR17 (外设引脚选择输出 17)	165
ECCPxDEL (增强型 PWM 控制)	264	RPOR18 (外设引脚选择输出 18)	166
EECON1 (EEPROM 控制 1)	105	RPOR19 (外设引脚选择输出 19)	166
HLVDCON (高 / 低压检测控制)	395	RPOR2 (外设引脚选择输出 2)	161
HOURS (小时值)	233	RPOR20 (外设引脚选择输出 20)	166
缓冲区描述符, 汇总	371	RPOR21 (外设引脚选择输出 21)	167
I ² C 模式 (MSSP)	288	RPOR22 (外设引脚选择输出 22)	167
INTCON (中断控制)	117	RPOR23 (外设引脚选择输出 23)	167
INTCON2 (中断控制 2)	118	RPOR24 (外设引脚选择输出 24)	168
INTCON3 (中断控制 3)	119	RPOR3 (外设引脚选择输出 3)	162
IPR1 (外设中断优先级 1)	126	RPOR4 (外设引脚选择输出 4)	162
IPR2 (外设中断优先级 2)	127	RPOR5 (外设引脚选择输出 5)	162
IPR3 (外设中断优先级 3)	128	RPOR6 (外设引脚选择输出 6)	163
MINUTES (分钟值)	233	RPOR7 (外设引脚选择输出 7)	163
MONTH (月值)	231	RPOR8 (外设引脚选择输出 8)	163
ODCON1 (外设漏极开路控制 1)	134	RPOR9 (外设引脚选择输出 9)	164
ODCON2 (外设漏极开路控制 2)	134	RTCCAL (RTCC 校准)	228
ODCON3 (外设漏极开路控制 3)	135	RTCCFG (RTCC 配置)	227
OSCCON (振荡器控制)	43	SECONDS (秒值)	233
OSCTUNE (振荡器调节)	40	SPI 模式 (MSSP)	271
PADCFG1 (焊盘配置)	228	SSPxCON1	
PADCFG1 (焊盘配置控制 1)	135	(MSSPx 控制 1, SPI 模式)	272
PIE1 (外设中断允许 1)	123	SSPxCON2	
PIE2 (外设中断允许 2)	124	(MSSPx 控制 2, I ² C 从模式)	292
PIE3 (外设中断允许 3)	125	SSPxCON2	
PIR1 (外设中断请求 (标志) 1)	120	(MSSPx 控制 2, I ² C 主模式)	291
PIR2 (外设中断请求 (标志) 2)	121	SSPxMSK (I ² C 从地址掩码)	292
PIR3 (外设中断请求 (标志) 3)	122	SSPxSTAT	

(MSSPx 状态, I ² C 模式)	289
SSPxSTAT	
(MSSPx 状态, SPI 模式)	271
STATUS	96
STKPTR (堆栈指针)	80
T0CON (Timer0 控制)	195
T1CON (Timer1 控制)	199
T1GCON (Timer1 门控控制)	201
T2CON (Timer2 控制)	211
T3CON (Timer3 控制)	213
T3GCON (Timer3 门控控制)	214
T4CON (Timer4 控制)	223
TCLKCON (定时器时钟控制)	202, 215
TXSTAx (发送状态和控制)	324
UADDR	365
UCFG (USB 配置)	361
UCON (USB 控制)	359
WDTCON (看门狗定时器控制)	428
UEIR (USB 错误中断状态)	376
UEPn (USB 端点 n 控制)	364
UFRMH:UFRML	365
UIE (USB 错误中断允许)	377
UIE (USB 中断允许)	375
UIR (USB 中断状态)	373
WKDY (星期值)	232
USTAT (USB 状态)	363
文件汇总	89
YEAR (年值)	231
与比较器相关	385
与看门狗定时器相关	428
与 PMP 模块相关	193
与闪存程序存储器相关	112
寄存器文件	86
寄存器文件汇总	89 - 95
INCF	456
INCFSZ	457
INFSNZ	457
INTCON	117
INTCON 寄存器	117 - 119
INTOSC 和 INTRC. 请参见内部振荡器模块。	
INTOSC 频率漂移	39
IORLW	458
IORWF	458
IPR 寄存器	126 - 128
计算 GOTO	81
间隔字符 (12 位) 发送和接收	340
间接寻址	98
交流 (时序) 特性	508
参数符号体系	508
器件时序规范的负载条件	509
时序条件	509
温度和电压规范	509
接收到同步间隔字符时自动唤醒	338
晶振 / 陶瓷谐振器	37
绝对最大额定值	489
K	
开发支持	485
看门狗定时器 (WDT)	417, 427
编程注意事项	427
控制寄存器	427
相关的寄存器	428
振荡器故障期间	432
勘误表	9
客户通知服务	558
客户支持	558

快速寄存器堆栈	81
框图	
+5V 系统硬件接口	133
8 位复用地址和数据应用	191
A/D	350
比较器参考电压	391
比较器模拟输入模型	387
比较器输出	385
表读操作	103
表写操作	104
并行 EEPROM	
(最多 15 位地址, 16 位数据)	192
并行 EEPROM	
(最多 15 位地址, 8 位数据)	192
并行主 / 从器件连接缓冲	180
并行主 / 从器件连接寻址	
缓冲区示例	181
波特率发生器	310
部分复用的寻址模式	184
部分复用的寻址应用	191
捕捉模式工作原理	248
CTMU	401
CTMU 电流源校准电路	404
参考电压输出缓冲示例	393
测量时间的 CTMU 典型	
连接和内部配置	411
带外部输入的高 / 低压检测	396
单比较器	387
读闪存程序存储器	107
对闪存程序存储器的表写操作	109
EUSART 发送	334
EUSARTx 接收	337
复用的寻址应用	191
故障保护时钟监视器	432
简化的转向	266
解复用的寻址模式	184
看门狗定时器	427
LCD 控制	192
MSSP (SPI 模式)	270
MSSPx (I ² C 模式)	288
MSSPx (I ² C 主模式)	308
模拟输入模型	351
PIC18F4XJ50 (44 引脚)	15
PMP 模块	169
PWM (增强型)	253
片上复位电路	63
器件时钟	36
RTCC	225
生成脉冲延时的 CTMU	
典型连接和内部配置	412
使用漏极开路输出	133
Timer0 (16 位模式)	196
Timer0 (8 位模式)	196
Timer2	212
Timer3	216
Timer4	224
Timer1	204
通用 I/O 端口的工作原理	131
USB 外设和选项	357
USB 中断逻辑	372
USTAT FIFO	363
完全复用的寻址模式	184
中断逻辑	116
传统并行从端口	178
扩展指令	

PIC18F46J50 系列

使能时的注意事项	482
扩展指令集	
ADDFSR	478
ADDULNK	478
CALLW	479
MOVSF	479
MOVSS	480
PUSHL	480
SUBFSR	481
SUBULNK	481

L

LFSR	459
立即数变址模式	482
立即数变址寻址 和标准 PIC18 指令	482

M

Microchip 因特网网站	558
MOV	459
MOVFF	460
MOVLB	460
MOVLW	461
MOVSF	479
MOVSS	480
MOVWF	461
MPLAB ASM30 汇编器、 链接器和库管理器	486
MPLAB PM3 器件编程器	488
MPLAB REAL ICE 在线仿真器系统	487
MPLAB 集成开发环境软件	485
MPLINK 目标链接器 /MPLIB 目标库管理器	486
MSSP	
ACK 脉冲	293,296
I ² C 模式。请参见 I ² C 模式。 模块概述	269
SPI 主 / 从器件连接	274
TMR4 时钟输出移位	224
MULLW	462
MULWF	462
脉冲转向	264
模数转换器。请参见 A/D。	

N

NEGF	463
NOP	463
内部参考电压规范	505
内部集成电路。请参见 I ² C。 内部 RC 振荡器 与 WDT 一起使用	427
内部振荡器 频率漂移。请参见 INTOSC 频率漂移。	
内部振荡器模块	38
调整	39
OSCTUNE 寄存器	39
内核特性	
扩展存储器	11
扩展指令集	12
纳瓦技术	11
通用串行总线（USB）	11
移植方便	12
振荡器选项和特性	11

P

P1A/P1B/P1C/P1D。	

请参见增强型

捕捉 / 比较 /PWM（ECCP）。	253
PIE 寄存器	123 - 125
PIR 寄存器	120
PLL 倍频器	38
POP	464
POR。请参见上电复位。	

PORTA

LATA 寄存器	136
PORTA 寄存器	136
TRISA 寄存器	136
相关的寄存器	138
引脚的其他功能 超低功耗唤醒	60

PORTB

LATB 寄存器	139
PORTB 寄存器	139
RB7:RB4 电平变化中断标志 （RBIF 位）	139
TRISB 寄存器	139
相关的寄存器	142

PORTC

LATC 寄存器	143
PORTC 寄存器	143
TRISC 寄存器	143
相关的寄存器	145

PORTD

LATD 寄存器	146
PORTD 寄存器	146
TRISD 寄存器	146
相关的寄存器	147

PORTE

LATE 寄存器	148
PORTE 寄存器	148
TRISE 寄存器	148
相关的寄存器	149

PRI_IDLE 模式

PRI_RUN 模式

PWM（CCP 模块）

相关寄存器	252
占空比	250
周期	250
TMR4 与 PR4 匹配	223

PWM（ECCP 模块）

复位的影响	267
脉冲转向	264
使用故障保护时钟监视器操作	267
在功耗管理模式下的操作	267
转向同步	266

PWM 模式。

请参见增强型捕捉 / 比较 /PWM。

PUSH

PUSH 和 POP 指令

PUSHL

配置不匹配（CM）复位

配置寄存器

位和器件 ID	418
映射闪存配置字	418
配置寄存器保护	433
配置位	417

Q

器件差异	545
器件概述	11
其他特殊性能	12

特性 (28 引脚器件)	13
特性 (44 引脚器件)	13
系列中各产品的详细说明	12
欠压复位 (BOR)	65
和片上稳压器	430
检测	65
在休眠模式下禁止	65

R

RAM。请参见数据存储器。

RBIF 位	139
RCALL	465
RC_IDLE 模式	53
RCON 寄存器	
初始化时位的状态	68
RC_RUN 模式	50
RESET	465
RETFIE	466
RETLW	466
RETURN	467
RLCF	467
RLNCF	468
RRCF	468
RRNCF	469
RTCC	
复位	242
器件	242
上电复位 (POR)	242
工作原理	
寄存器读写安全窗口	239
寄存器映射	239
ALRMVAL	240
进位规则	238
闰年	239
时钟源	238
校准	240
写锁定	239
一般功能	239
寄存器接口	237
寄存器映射	243
控制寄存器	227
闹钟	241
配置	241
掩码设置	241
中断	242
闹钟值寄存器 (ALRMVAL)	234
外设模块禁止 (PMD) 寄存器	242
值寄存器 (RTCVAL)	231
RTCEN 位写	237
入门指南	29, 30
电源引脚	30
ICSP 引脚	32
连接要求	29
外部振荡器引脚	33
未用 I/O	33
稳压器引脚	
(VCAP/VDDCORE)	31
软件模拟器 (MPLAB SIM)	487

SLEEP	470
SPI 模式 (MSSP)	
DMA 模块	
RAM 至 RAM 复制	280
SPI 模式 (MSSP)	270
串行时钟	270
串行数据输出	270
串行数据输入	270
从模式	276
从选择	270
从选择同步	276
DMA 概述	280
典型连接	274
复位的影响	278
工作原理	273
漏极开路输出选项	273
寄存器	271
SPI 时钟	275
SSPxBUF 寄存器	275
SSPxSR 寄存器	275
使能 SPI I/O	274
时钟速度, 相互关系	278
相关的寄存器	279
在功耗管理模式下的操作	278
主 / 从器件连接	274
主模式	275
总线模式兼容性	278
SSPOV	313
SSPOV 状态标志	313
SSPxSTAT 寄存器	
R/W 位	293, 296
SSx	270
SWAPF	472
SUBFSR	481
SUBFWB	470
SUBLW	471
SUBWF	471
SUBWFB	472
SUBULNK	481
闪存程序存储器	103
表读与表写	103
表指针	
基于操作的边界	106
表指针边界	106
擦除	108
擦除序列	108
代码保护期间的操作	112
读	107
控制寄存器	104
EECON1 和 EECON2	104
TABLAT (表锁存) 寄存器	106
TBLPTR (表指针) 寄存器	106
相关的寄存器	112
写	109
写校验	112
意外终止	112
写序列	109
上电复位 (POR)	65
上电延时	45
上电延时定时器 (PWRT)	45, 66
延时时序	66
实时时钟和日历 (RTCC)	225
工作原理	237
寄存器	226
时序图	

PIC18F46J50 系列

A/D 转换	529	(7 位接收, SEN = 1)	305
BRG 溢出序列	332	i ² C 从模式广播呼叫地址序列	
半桥 PWM 输出	256, 263	(7 位或 10 位寻址模式)	307
并行从端口	519	i ² C 停止条件接收或发送模式	316
并行从端口读	179, 181	i ² C 应答序列	316
并行从端口写	179, 182	i ² C 总线启动位 / 停止位	524
并行主端口读	517	进入空闲模式的转换	52
并行主端口写	518	进入 SEC_RUN 模式的转换	49
CLKO 和 I/O	512	进入休眠模式的转换	51
从空闲模式唤醒		MSSPx i ² C 总线启动位 / 停止位	526
进入运行模式的转换	53	MSSPx i ² C 总线数据	526
从 RC_RUN 模式到		PWM 方向改变	259
PRI_RUN 模式的转换	50	PWM 输出 (低电平有效)	255
从 SEC_RUN 模式到		PWM 输出 (高电平有效)	254
PRI_RUN 模式的转换 (HSPLL)	49	PWM 自动关闭, 带固件重启	262
从同步	276	PWM 自动关闭, 使能自动重启	262
从休眠模式唤醒的转换 (HSPLL)	51	启动条件期间的总线冲突	
带有时钟仲裁的波特率发生器	311	(仅用于 SDAx)	318
到 RC_RUN 模式的转换	50	启动条件期间的总线冲突	
低压检测 (VDIRMAG = 0)	398	(SCLx = 0)	319
第一个启动位	311	启动条件期间由 SDAx	
定时器脉冲的产生	242	仲裁引起的 BRG 复位	319
读, 16 位复用的数据, ...		全桥 PWM 输出	258
部分复用的地址	189	SPI 从模式示例 (CKE = 0)	522
读, 16 位复用的数据, ...		SPI 从模式示例 (CKE = 1)	523
完全复用的 16 位地址	190	SPI 模式 (从模式, CKE = 0)	277
读, 16 位数据, ...		SPI 模式 (从模式, CKE = 1)	277
解复用的地址	189	SPI 模式 (主模式)	275
读, 8 位数据, ...		SPI 主模式示例 (CKE = 0)	520
部分复用的地址	186	SPI 主模式示例 (CKE = 1)	521
读, 8 位数据, ...		上电延时时序 (MCLR 连接到 VDD, ...	
部分复用的地址, 使能选通	187	VDD 电压上升时间 < TPWRT)	66
读, 8 位数据, ...		上电延时时序	
使能等待状态, 部分复用的地址	186	(MCLR 未连接到 VDD), ...	
读, 8 位数据, ...		情形 1	67
完全复用的 16 位地址	188	上电延时时序	
读和写, 8 位数据, 解复用的地址	186	(MCLR 未连接到 VDD), ...	
EUSARTx 同步发送 (主 / 从)	528	情形 2	67
EUSARTx 同步接收 (主 / 从)	528	时钟 / 指令周期	82
发送和应答时的总线冲突	317	时钟同步	304
发送间隔字符序列	340	双速启动转换	
复位、看门狗定时器 (WDT)、		(从 INTRC 切换到 HSPLL)	431
振荡器起振定时器 (OST)		Timer0 和 Timer1 外部时钟	515
和上电延时定时器 (PWRT)	513	Timer3 门控单脉冲 / 翻转组合模式	220
高 / 低压检测特性	507	Timer3 门控单脉冲模式	219
高压检测 (VDIRMAG = 1)	399	Timer3 门控翻转模式	218
故障保护时钟监视器	432	Timer3 门控计数使能模式	217
缓慢上升时间 (MCLR 连接到 VDD, ...		Timer1 门控单脉冲 / 翻转组合模式	210
VDD 电压上升时间 > TPWRT)	67	Timer1 门控单脉冲模式	209
i ² C 总线数据	524	Timer1 门控翻转模式	208
i ² C 从模式 (10 位发送)	302	Timer1 门控计数使能模式	207
i ² C 从模式		停止条件期间的总线冲突 (情形 1)	321
(10 位接收, SEN = 0, ...		停止条件期间的总线冲突 (情形 2)	321
ADMSK = 01001)	300	同步发送	341
i ² C 从模式		同步发送 (由 TXEN 位控制)	342
(10 位接收, SEN = 0)	301	同步接收 (主模式, SREN)	343
i ² C 从模式		USB 信号	530
(10 位接收, SEN = 1)	306	外部时钟	510
i ² C 从模式 (7 位发送)	299	写, 16 位复用的数据, ...	
i ² C 从模式		部分复用的地址	190
(7 位接收, SEN = 0, ...		写, 16 位复用的数据, ...	
ADMSK = 01011)	298	完全复用的 16 位地址	190
i ² C 从模式		写, 16 位数据, ...	
(7 位接收, SEN = 0)	297	解复用的地址	189
i ² C 从模式		写, 8 位数据, ...	

部分复用的地址.....	187	扩展指令集.....	99
写, 8 位数据, 部分复用的地址, 使能选通.....	188	特殊功能寄存器.....	87
写, 8 位数据, 使能等待状态, 部分复用的地址.....	187	上下文定义的 SFR.....	89
写, 8 位数据, 完全复用的 16 位地址.....	188	通用寄存器.....	86
休眠时的自动唤醒位 (WUE).....	339	USB RAM.....	84
异步发送.....	334	数据寻址模式.....	97
异步发送 (背对背).....	334	固有和立即数.....	97
异步接收.....	337	间接.....	97
在占空比接近 100% 时		立即数变址寻址.....	100
改变 PWM 方向.....	260	BSR.....	102
增强型捕捉 / 比较 / PWM.....	516	受影响的指令.....	100
正常操作时的自动唤醒位 (WUE).....	340	映射快速操作存储区.....	102
指令结束时发生的转向事件		使能了扩展指令集的寻址模式对比.....	101
(STRSYNC = 0).....	266	直接.....	97
指令开始时发生的转向事件		所有寄存器的初始化状态.....	69 - 76
(STRSYNC = 1).....	266	双速启动.....	417, 431
重复启动条件.....	312	双字指令	
重复启动条件期间的总线冲突 (情形 1).....	320	示例情形.....	83
重复启动条件期间的总线冲突 (情形 2).....	320	T	
自动波特率计算.....	332	TAD.....	353
时序图和规范		TBLRD.....	473
并行从端口要求.....	519	TBLWT.....	474
并行主端口读要求.....	517	Timer0.....	195
并行主端口写要求.....	518	16 位模式下的读写操作.....	196
CLKO 和 I/O 要求.....	512	工作原理.....	196
低功耗唤醒时间.....	514	时钟源边沿选择 (T0SE 位).....	196
EUSARTx 同步发送要求.....	528	时钟源选择 (T0CS 位).....	196
EUSARTx 同步接收要求.....	528	相关的寄存器.....	197
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.....	513	溢出中断.....	197
I ² C 总线启动位 / 停止位要求		预分频器.....	197
(从模式).....	524	切换分配.....	197
I ² C 总线数据要求		预分频器分配 (PSA 位).....	197
(从模式).....	525	预分频器选择 (T0PS2:T0PS0 位).....	197
交流特性		Timer2.....	211
内部 RC 精度.....	511	工作原理.....	211
MSSPx I ² C 总线启动位 / 停止位要求.....	526	输出.....	212
MSSPx I ² C 总线数据要求.....	527	相关的寄存器.....	212
PLL 时钟.....	511	中断.....	212
SPI 从模式要求示例 (CKE = 1).....	523	Timer3.....	213
SPI 模式要求示例		16 位读 / 写模式.....	217
(从模式, CKE = 0).....	522	工作原理.....	216
SPI 模式要求示例、		门控.....	217
(主模式, CKE = 0).....	520	TMR3H 寄存器.....	213
SPI 模式要求示例		TMR3L 寄存器.....	213
(主模式, CKE = 1).....	521	特殊事件触发信号 (ECCP).....	221
Timer0 和 Timer1 外部时钟要求.....	515	相关的寄存器.....	221
USB 低速要求.....	530	溢出中断.....	213, 221
USB 全速要求.....	530	振荡器.....	213, 217
外部时钟要求.....	510	Timer4.....	223
增强型捕捉 / 比较 / PWM 要求.....	516	工作原理.....	223
时钟源.....	42	后分频器。请参见后分频器, Timer4。.....	224
功耗管理模式的影响.....	45	MSSP 时钟移位.....	224
使用 OSCCON 寄存器选择.....	42	PR4 寄存器.....	223
选择 31 kHz 时钟源.....	42	输出.....	224
数据存储器.....	84	TMR4 到 PR4 匹配中断.....	224
存储器映射		TMR4 寄存器.....	223
非快速操作存储区的特殊功能寄存器.....	88	TMR4 与 PR4 匹配时中断.....	223
快速操作存储区的特殊功能寄存器.....	87	相关的寄存器.....	224
PIC18F46J50 系列器件.....	85	预分频器。请参见预分频器, Timer4。.....	224
存储区选择寄存器 (BSR).....	84	中断.....	224
快速操作存储区.....	86	Timer1.....	199
		16 位读 / 写模式.....	205
		复位, 使用 ECCP 特殊事件触发信号.....	207
		工作原理.....	203
		门控.....	207

PIC18F46J50 系列

时钟源选择	203
TMR1H 寄存器	199
TMR1L 寄存器	199
相关的寄存器	210
振荡器	199, 205
布线注意事项	206
中断	206
作为时钟源	206
TSTFSZ	475
TXSTAX 寄存器	
BRGH 位	327
特殊事件触发信号。请参见比较（ECCP 模式）。	
特性概述	3
通用串行总线	357
地址寄存器（UADDR）	365
电源模式	378
仅自供电	378
仅总线供电	378
收发器的电流消耗	379
以自供电为主的双电源供电	378
端点控制	364
概述	357, 382
电源	382
分层架构	382
分类规范和驱动程序	383
枚举	383
描述符	383
速度	383
帧	382
传输类型	382
固件和驱动程序	381
缓冲区描述符	366
BDnSTAT 寄存器	
（CPU 模式）	367
BDnSTAT 寄存器	
（SIE 模式）	369
不同缓冲模式中的分配	371
存储器映射	370
地址验证	369
寄存器汇总	371
乒乓缓冲	370
示例	366
所有权	366
状态和配置	366
字节计数	369
缓冲区描述符表	366
内部上拉电阻	362
内部收发器	360
乒乓缓冲区配置	362
RAM	365
存储器映射	365
使能眼图测试	362
UFRMH:UFRML 寄存器	365
外部上拉电阻	362
相关的寄存器	381
帧编号寄存器	365
振荡器要求	381
中断	372
和 USB 事务	372
状态和控制	358

W

WCOL	311, 312, 313, 316
WCOL 状态标志	311, 312, 313, 316
ULPWU 规范	505
USB。请参见通用串行总线。	

USB 的振荡器设置	40
配置选项	41
USB 规范	506
WWW 地址	558
外部时钟输入	38
外设引脚选择（PPS）	150
外设引脚选择寄存器	155 - 168
稳压器（片上）	429
休眠模式下的操作	430

X

XORLW	475
XORWF	476
校准（A/D 转换器）	355

Y

引脚功能

AVDD1	28
AVDD2	28
AVSS1	28
MCLR	16, 22
OSC1/CLKI/RA7	16, 22
OSC2/CLKO/RA6	16, 22
RA0/AN0/C1INA/ULPWU/PMA6/RP0	23
RA0/AN0/C1INA/ULPWU/RP0	17
RA1/AN1/C2INA/PMA7/RP1	23
RA1/AN1/C2INA/RP1	17
RA2/AN2/VREF-/CVREF/C2INB	17, 23
RA3/AN3/VREF+/C1INB	17, 23
RA5/AN4/SS1/HLVDIN/RCV/RP2	23
RA6	17, 23
RA7	17, 23
RB0/AN12/INT0/RP3	18, 24
RB1/AN10/PMBE/RTCCS/RP4	24
RB1/AN10/RTCC/RP4	18
RB2/AN8/CTED1/PMA3/VMO/REFO/RP5	24
RB2/AN8/CTED1/VMO/REFO/RP5	18
RB3/AN9/CTED2/PMA2/VPO/RP6	24
RB3/AN9/CTED2/VPO/RP6	18
RB4/KBI0/SCK1/SCL1/RP7	19
RB4/PMA1/KBI0/SCK1/SCL1/RP7	25
RB5/KBI1/SDI1/SDA1/RP8	19
RB5/PMA0/KBI1/SDI1/SDA1/RP8	25
RB6/KBI2/PGC/RP9	19, 25
RB7/KBI3/PGD/RP10	19, 25
RC0/T1OSO/T1CKI/RP11	20, 26
RC1/T1OSI/UOE/RP12	20, 26
RC2/AN11/CTPLS/RP13	20, 26
RC4/D-/VM	20, 26
RC5/D+/VP	20, 26
RC6/PMA5/TX1/CK1/RP17	26
RC6/TX1/CK1/RP17	20
RC7/PMA4/RX1/DT1/SDO1/RP18	26
RC7/RX1/DT1/SDO1/RP18	20
RD0/PMD0/SCL2	27
RD1/PMD1/SDA2	27
RD2/PMD2/RP19	27
RD3/PMD3/RP20	27
RD4/PMD4/RP21	27
RD5/PMD5/RP22	27
RD6/PMD6/RP23	27
RD7/PMD7/RP24	27
RE0/AN5/PMRD	28
RE1/AN6/PMWR	28
RE2/AN7/PMCS	28
VDD	21

VDD1	28
VDD2	28
VDDCORE/VCAP	21, 28
VSS1	21, 28
VSS2	21, 28
VUSB	21, 28
引脚图	5 - 7
因特网地址	558
硬件乘法器	113
8 x 8 乘法算法	113
工作原理	113
性能比较 (表)	113
预分频器, Timer0	197
Z	
在线串行编程 (ICSP)	417, 434
在线调试器	434
增强型捕捉 / 比较 / PWM (ECCP)	245
比较模式。请参见比较。	
捕捉模式。请参见捕捉。	
ECCP 模式和定时器资源	247
输出和配置	247
相关的寄存器	267
增强型 PWM 模式	253
半桥模式	256
半桥应用	256
半桥应用示例	263
可编程死区延时	263
启动注意事项	260
全桥模式	257
全桥输出模式中的方向改变	259
全桥应用	257
输出关系	
(高电平有效和低电平有效)	254
输出关系图	255
直通电流	263
自动关闭	261
自动重启	262
增强型通用同步 / 异步收发器 (EUSART)。	
请参见 EUSART。	
振荡器, Timer3	213
振荡器, Timer1	199, 205, 217
振荡器配置	35
内部振荡器模块	38
振荡器控制	35
振荡器类型	35
振荡器模式	35
振荡器模式和 USB 操作	36
振荡器起振定时器 (OST)	45
振荡器切换	42
振荡器选择	417
振荡器转换	43
直接寻址	98
指令集	435
ADDLW	441
ADDWF	441
ADDWF (立即数变址寻址模式)	483
ADDWFC	442
ANDLW	442
ANDWF	443
BC	443
BCF	444
BN	444
BNC	445
BNN	445
BNOV	446

BNZ	446
BOV	449
BRA	447
BSF	447
BSF (立即数变址寻址模式)	483
BTFSC	448
BTFSS	448
BTG	449
BZ	450
标准指令	435
CALL	450
CLRF	451
CLRWDI	451
COMF	452
CPFSEQ	452
CPFSGT	453
CPFSLT	453
操作码字段说明	436
DAW	454
DCFSNZ	455
DECf	454
DECFSZ	455
GOTO	456
INCF	456
INCFSZ	457
INFSNZ	457
IORLW	458
IORWF	458
扩展指令	477
使能时的注意事项	482
使用 MPLAB IDE 工具	484
语法	477
LFSR	459
MOVF	459
MOVFF	460
MOVLB	460
MOVLW	461
MOVWF	461
MULLW	462
MULWF	462
NEGF	463
NOP	463
POP	464
PUSH	464
RCALL	465
RESET	465
RETFIE	466
RETLW	466
RETURN	467
RLCF	467
RLNCF	468
RRCF	468
RRNCF	469
SETF	469
SETF (立即数变址寻址模式)	483
SLEEP	470
SWAPF	472
SUBFWB	470
SUBLW	471
SUBWF	471
SUBWFB	472
TBLRD	473
TBLWT	474
TSTFSZ	475
通用格式	437

PIC18F46J50 系列

XORLW	475
XORWF	476
指令周期	82
时钟分配	82
指令流 / 流水线	82
直流特性	502
掉电和供电电流	492
供电电压	491
直通电流	263
中断	115
INTx 引脚	130
控制寄存器。请参见 INTCON 寄存器。	
控制位	115
PORTB, 电平变化中断	130
期间, 现场保护	130
RCON 寄存器	129
TMR0	130
中断, 标志位	
电平变化中断 (RB7:RB4)	
标志 (RBIF 位)	139
中断源	417
A/D 转换完成	351
比较完成 (ECCP)	249
捕捉完成 (ECCP)	248
电平变化中断 (RB7:RB4)	139
TMR0 溢出	197
TMR1 溢出	206
TMR3 溢出	213, 221
TMR4 到 PR4 匹配	224
TMR4 与 PR4 匹配 (PWM)	223
主复位 (MCLR)	65
主同步串行口 (MSSP)。请参见 MSSP。	

MICROCHIP 网站

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

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

变更通知客户服务

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

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

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

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

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

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

PIC18F46J50 系列

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: PIC18F46J50 系列 文献编号: DS39931D_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?

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

<u>器件编号</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
器件	温度范围	封装	模式
器件	PIC18F24J50 PIC18F25J50 PIC18F26J50 PIC18F44J50 PIC18F45J50 PIC18F46J50 PIC18LF24J50 PIC18LF25J50 PIC18LF26J50 PIC18LF44J50 PIC18LF45J50 PIC18LF46J50		
温度范围	I = -40°C 至 +85°C（工业级）		
封装	SP = 窄型 PDIP SS = SSOP SO = SOIC ML = QFN PT = TQFP（薄型正方扁平封装）		
模式	QTP、SQTP、代码或特殊要求 （其他情况空白）		

示例：

a) PIC18F46J50-I/PT 301 = 工业级温度， TQFP 封装， QTP 模式 #301。

b) PIC18F46J50T-I/PT = 卷带式， 工业级温度， TQFP 封装。

全球销售及服务网点

美洲

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

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

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

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

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

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

底特律 Detroit
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 Santa Clara
Santa Clara, CA
Tel: 1-408-961-6444
Fax: 1-408-961-6445

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

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

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

中国 - 杭州
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

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 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-330-9305

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-2566-1512
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